THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

PROBABILISTIC CHARACTERIZATION OF FLOODS FROM CATCHMENT-SCALE PRECIPITATION MOMENTS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By
JORGE A. DUARTE GARCÍA
Norman, Oklahoma

2019

PROBABILISTIC CHARACTERIZATION OF FLOODS FROM CATCHMENT-SCALE PRECIPITATION MOMENTS

A THESIS APPROVED FOR THE

GALLOGLY COLLEGE OF ENGINEERING

BY

Dr. Charles D. Nicholson, Chair

Dr. Pierre E. Kirstetter

Dr. Randa L. Shehab

©Copyright by Jorge A. Duarte García 2019

All rights reserved.

Abstract

Floods are one of the most devastating natural hazards across the world, accounting for roughly one third of all global geophysical hazards. The ability to predict and characterize floods is increasingly important, and in order to achieve effective flash flood characterization (due to their short lead times and distinct localization), the need to account for rainfall spatial variability arises.

Spatial precipitiation moments offer a concise yet resourceful set of abstractions, which condense and expose intrinsic geophysical interactions between rainfall and basin. By leveraging the richness of these dimensionless statistics, this research aims to construct supervised machine learning models which could offer a probabilistic characterization of flood conditions over gauged locations accross the Contigous United States (CONUS). These models are trained on a real, historical, event-based flood database, which contains precipitation moment data (pre-generated), as well as hydrological, morphological and bioclimatic information for each of the flooding events, and the basins over which they occurred.

Three different machine lerning techniques (MARS, Random Forest and Support Vector Machines) are used to characterize and explore three different aspects of floods: basin response time (lag time), flood stage threshold exceedance and the moment of relative peak discharge - a proposed indicator which describes the peak streamflow behavior of a stream with respect to the duration of the flooding event. Both classification and regression models are built for these responses using the same techniques. Variable importance analysis is also performed in order to determine the relevat factors that influence each of the modeled response. A probabilistic characterization of flood stage threshold exceedance is also achieved by extracting classification probabilities from these models, which are presented and analyzed by using reliability diagrams and other statistical tools.

Contents

List of Tables

List of Figures

Listings

Preface

As part of the University of Oklahoma's (OU) Cooperative Institute for Mesoscale Meteorological Studies (CIMMS), I have worked with the National Oceanic and Atmospheric Administration's (NOAA) National Severe Storms Laboratory (NSSL), where I conduct collaborative research on various topics related to hydrology and hydrometeorology as part of the Warning Research and Development Division (WRDD). One of WRDD's main functions is to design, test and transition new warning and decision-making tools and technologies to the National Weather Service (NWS). Within this collaborative context, surrounded by expert hydrologists, meteorologists and engineers, I've had the opportunity of working on flash flood and debris flow characterization, modeling, monitoring and alerting projects. The present study is an example of this kind of research, which is ultimately aimed towards innovating and/or improving existing tools and technologies, which enable relevant actors and organizations to better understand complex processes, make better decisions and issue warnings in the face of natural hazards.

This study also extends the work done for the course DSA 5900 - Professional Practice, titled *Characterizing Basin Response Time in the United States*, which revovled around characterizing Lag Time over the Contiguous United States, using a similar dataset to the one used in the present project. Even though spatial moments of precipitation were part of the dataset as well, this was a catchment-based dataset: all rainfall event data had been averaged, and that study pretended to characterize basin response time at a climatological scale. Conversely, the present study aims to characterize floods at an event scale, and furthermore, extend this characterization beyond Lag Time.

I would like to acknowledge the help and support provided by Dr. Pierre E. Kirstetter throughtout the execution of this project; thank you for your patience and guidance. I also acknowledge Dr. Manabendra Saharia's arduous labor of calculating, compiling and quality-controlling the data used for this work; this was the foundation for this project's development. Special thanks to Dr. J.J. Gourley and Dr. Humberto J. Vergara, for the trust they have placed in me since the beginning, their unconditional suppport and for the opportunity to pursue a research carreer. Lastly, I thank all of those who have supported me along this road: family, friends and colleages. None of this work would have been possible without your advice and encouragement.

Chapter 1

Introduction

Floods are one of the most devastating natural hazards that occur across all of our planet, and they accounting for roughly one third of all global geophysical hazards. Flash floods are floods that follow the causative storm event in a short period of time, with water levels in the drainage network reaching a crest within minutes to a few hours after the onset of the rain event. These stand out to be one of the most dangerous phenomena, as they leave extremely short times for warnings to be emitted [1]. In the United States, $2.86 billion dollars of direct flood damages occurred in 2014 alone, there were 55 flood-related deaths, of which 39 where flash-flood related [2].

The ability to predict and characterize floods is increasingly important, and in order to achieve effective flash flood characterizations a better understanding of contributing factors must be achieved. This has been approached by incorporating new techniques, sources of information and new representations of data which concisely describe complex geophysical, meteorological and climatological processes into existing hydrological models [3] [4] [5] [6] [7]. However, all of these approaches have strictly relied on pre-conceived conceptual, mathematical or even speculative relationships between the phenomena in question and the available data. In the age of *Big Data*, where computing resources are made available (nearly) instantaneously and Machine Learning has never been more within our reach, a data-driven approach towards the characterization of floods can perhaps provide an alternative, suitable way of approaching these types of problem. Not only providing modeling robustness and efficiency (*i.e.* when making predictions), but also allowing for a different data-centric perspective when exploring the underlying relationships which characterize flooding. Ultimately, these relationships can (and should) be compared and contrasted with the systematically-built models, that Hydrologists and Hydrometeorologists employ regularly. For these reasons, the proposal for these data-driven approaches should not exclusively be result oriented (*i.e.* black-boxes), but also -and most importantly- process oriented, so that experts and interested actors are able to understand how these phenomena are characterized from the input data.

The current study is rooted in the need of incorporating the spatial variability of rainfall into hydrological models, in order to account for the spatially-distributed interactions of terrain and precipitation [3] [4]. Rainfall is a highly heterogeneous process both spatially and temporally, but through *Spatial Moments of rainfall*, precipitation spatial variability can be described through concise quantities, that can be easily assimilated into hydrological models to better characterize hydrologic phenomena (such as flooding). The present approach aims to be significant not only in the exploration and proposal of alternatives for characterizing floods by incorporating said *precipitation moments*, but also in doing so in a data-driven way.

The notion of watershed (basin, or catchment) is the basic unit used in hydrology, to denote a finite, contiguous area, such that the net rainfall or runoff over that area will contribute water to its outlet (see Figure 1.1). Bounds for a given basin can be defined by topography, where runoff will travel from higher to lower elevation, and rainfall that falls outside of these boundary will not contribute to runoff at the outlet [8].

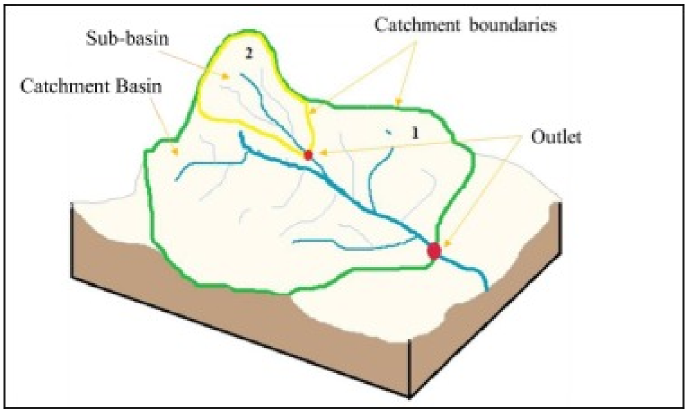

Figure 1.1: A general diagram of a watershed or basin [9]

Gauge stations are usually placed at these outlets to register the behavior of a stream, as it responds to the hydrologic processes affecting the watershed itself. They typically record data regarding the stream's stage (water level), velocity and discharge (streamflow). By using meteorological RADAR data as well as rain gauge networks hydrologists are able to measure and estimate the spatial and temporal distribution of precipitation over a basin, and then perform hydrological analyses of how the water inputs over the basin (*i.e.* precipitation) relate to the outputs being measured at the outlet. This can be represented by plotting these data over time, which generates a hydrograph. Typically, a streamflow hydrograph is presented in conjunction with the basin-averaged precipitation estimation data (hyetograph), which allows to appreciate the properties of this input-output relationship over time. An example of this is shown in Figure 1.2.

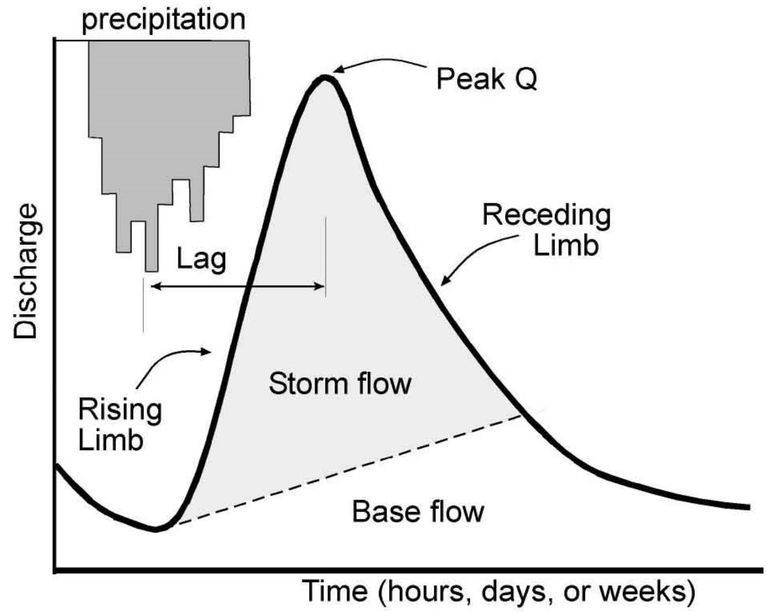

Figure 1.2: Parts and properties of a typical streamflow hydrograph [10]

The time difference between the precipitation's center of mass to the peak discharge (Q) in the streamflow response is defined as the lag time. This property of catchments is classically modeled as a relationship of basin characteristics, most prominently the catchment area [8]. Characterization of Lag Time is of interest in hydrology given its implications during extreme or heavy rainfall events which may trigger catastrophic flash floods downstream, as it is generally an indicator of lead time for issuing warnings, evacuation and risk assessment planning (among other applications).

Among gauged basins maintained by the United States Geological Survey (USGS), some have flood stage definitions defined and maintained by the National Weather Service (NWS). Flood stage is the level at which inundation is caused on areas that are not normally covered by water [11]. These are heights of water level associated with flooding conditions at a given channel, defined by historical records. Four flood stage levels are defined: *ACTION*, *MINOR*, *MODERATE* and *MAJOR*. These all refer to the potential severity of flooding associated with each threshold. Any value below the *ACTION* threshold is not considered as flood stage. Figure 1.3 shows an example of these flood stage definitions can be seen from a hydrograph taken from the NWS Advanced Hydrologic Prediction Service website, for a Mississippi River gauge in Baton Rouge, LA.

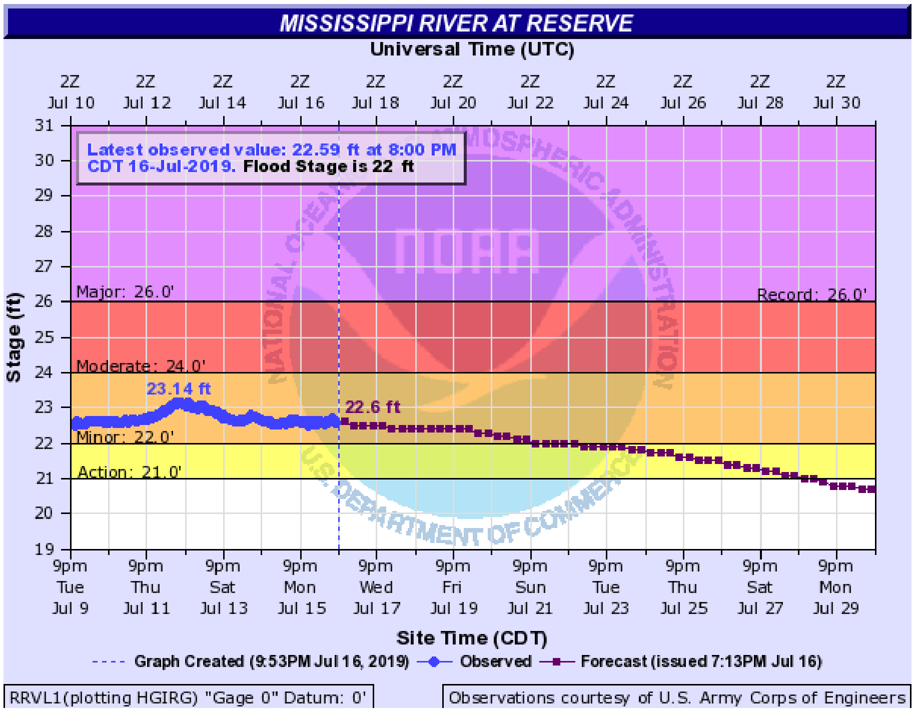

Figure 1.3: USGS Gauge Data: Stage of the Mississippi River at Reserve, Jul9-Jul15 2019 observations with NWS flood stage thresholds [12]

Characterizing flood stage conditions across the US is of interest as well, given that these are directly related to impacts in surrounding areas. This could dramatically improve a forecaster's abilities to issue more precise flood watches, warnings and evacuations, as well as improve flood inundation mapping efforts at ungauged locations. Additionally, providing probabilistic information for a given event of exceeding these threshold levels could dramatically improve guidance for forecasters, as well as risk managers and public service officials.

Rainfall estimation and measurement techniques over basins have evolved over time from simple measuring buckets into rain gauges, and from rain gauge networks into automated distributed RADAR networks. This evolution has brought the ability to measure not only the temporal variability of rainfall, but also its spatial variability. Instead of relying on a handful of geographically distributed data-points over which rainfall data was measured, averaged and assumed to be uniformly distributed across the terrain, modern RADAR technology now enables us to capture sub-kilometer gridded rainfall fields.

The spatial distribution of hydrology in general has been a continuous evolution process during the past decades. Distributed models were designed as the first approach to integrate spatially distributed information (elevation, soil moisture, land use, *etc.*). Slowly, as our ability to capture spatial variability of rainfall improved over time, these 'lumped' (spatially aggregated/averaged) hydrologic models became 'distributed' hydrologic models in a way. However, the process of transferring the effects of a distributed rainfall field into a streamflow response means that modeling efforts have been refocused over distributed runoff-generation processes and water transfer (routing) processes within watersheds. This is by all means a logical and coherent effort in hydrologic modeling, however, as the need for precision increases, the capability for increased resolution and sampling increases too; this means that modeling these processes accurately becomes a cumbersome challenge.

Because of this, ways to characterize the spatial distribution of precipitation in a comprehensive and usable way were sought after. Ideally, these new characterizations would allow existing hydrological models to account for rainfall spatial variability while keeping the assimilation process simple, as well as improving model accuracy and performance. Examples of these were proposed by Smith *et al.* [3], Zoccatelli *et al.* [4] [5], Douinot *et al.* [6] and Emmanuel *et al.* [7]. In essence, these measures of spatial variability relate to characteristics of a storm event over a catchment. Figure 1.4 shows the interpretation of Zoccatelli's δ_1 and δ_2 spatial moments of catchment rainfall, as presented by Douinot *et al.* [6].

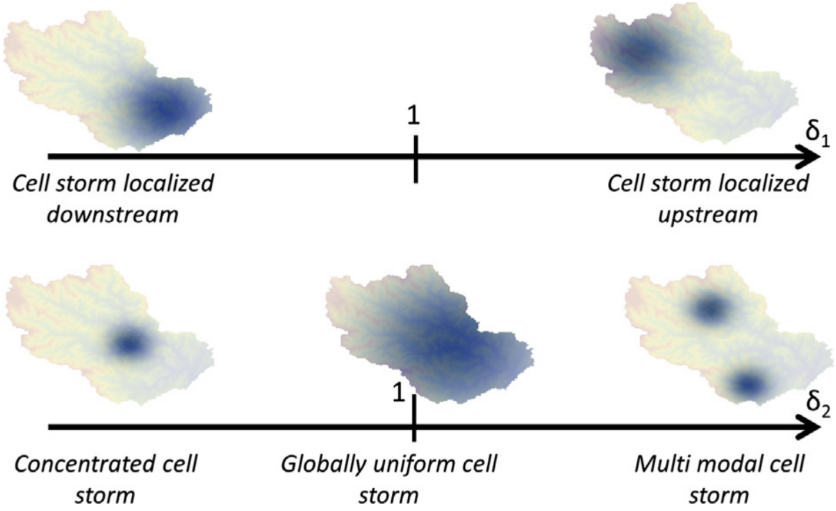
1
δ_1
Cell storm localized downstream
Cell storm localized upstream
1
δ_2
Concentrated cell storm
Globally uniform cell storm
Multi modal cell storm

Figure 1.4: Spatial moments of catchment rainfall: range of values and meaning of δ_1 and δ_2 [6]

As an example of one of these moments of catchment-scale precipitation, Zoccatelli's first moment of spatial rainfall states that: when $\delta_1 < 1$ the storm cell is localized downstream from the basin's centroid (near the outlet), and when $\delta_1 > 1$ the storm cell is localized upstream of the basin's centroid (near the head waters). As can be seen, these dimensionless statistics can be quite powerful in characterizing the behavior of a storm event, by reducing complex spatial interactions to a single indicator. Several of these quantities have been proposed by different authors [3] [4] [5] [7], and several of them were included in the working dataset for this research, in hopes of leveraging their usefulness to represent complex behaviors in a rich, concise way. Precipitation moments will be covered in more detail on the next chapter, which includes a thorough review of the relevant literature.

The USGS has over 10,000 gauge stations located all across the CONUS, each corresponding to a given catchment or basin. These gauges report data pertaining streamflow (discharge), water level (stage) and velocity(surface, or mean channel velocity), which is readily available online and through various distribution services. In addition to gauge information, morphological, bioclimatic and climatological data is available for most of the gauged basins occupying the CONUS. Observations of NEXRAD-based radar rainfall rates are available through NOAA's Multi-RADAR Multi-Sensor project, as well as a compilation of Flash Flood events made available through NOAA's Flooded Locations And Simulated Hydrographs (FLASH) project. Taking advantage of this abundance of data, a Spatial Precipitation Moment Flood event database was constructed by Dr. Manabendra Saharia, Dr. Pierre E. Kirstetter and several other collaborators, which integrated data from these aforementioned diverse resources, as well as others.

This dataset includes an enormous amount of attributes that describe historical precipitation events over various catchments across the CONUS, most of which triggered a flooding event. This dataset includes storm, streamflow and catchment information for each of the flooding events, including event lag times, peak flows and also each basin's USGS flood stage thresholds. Additionally, Dr. Saharia has computed an assortment of catchment-scale precipitation moments for each of the events.

Given the existence of this comprehensive dataset, and taking into consideration the matters discussed previously in this chapter, the following research questions arise: can an effective characterization of floods be achieved by using machine learning techniques and incorporating catchment-scale precipitation moments? If so: 1) can the relevant factors that characterize floods be determined? and 2) can distinct flooding conditions be characterized probabilistically?

In order to answer these questions and fulfill these objectives, this project explores the construction of supervised machine learning models that could offer a probabilistic characterization of flood conditions over gauged locations across the CONUS. Consequently, variable importance analyses were performed in order to determine the factors that influence flood characterization. Given that these models were trained and tested on the available real, historical, event-based rainfall moments, hydrological, meteorological, climatological and morphological data, it is expected that they should also be easily transferable to ungauged locations in future works.

The characterization of floods is by no means a novel idea, and it has transformed the way hydrology is applied in real life everyday. However, enhancing, building on top of these previous efforts, and incorporating new technologies into these types of problem will surely continue having enormous impacts on existing real-time hydrological modeling systems.

Chapter 2

Catchment-Scale Precipitation Moments

Lumped parameter hydrological models provide punctual outputs (usually at the basin's outlet), while distributed hydrologic modeling approaches offer the opportunity to model processes and discharge at points upstream the basin outlet. As mentioned before, Hydrology has struggled with the benefits and compromises of both alternatives for several decades.

In Smith *et al.* [3], the authors analyze observed rainfall and streamflow to describe the spatial variability of rainfall and corresponding basin outflow response in order to make inferences about model applicability (concerning lumped vs distributed models). It should be noted that the effects of model error as well as data and parameter uncertainty were intentionally excluded from this study.

The authors recognize that by accounting for spatial variability of rainfall and physical features within the basin (*i.e.* soil composition, morphology, *etc.*), better simulations can be achieved at the outlet. However, the nonlinearities and computational elements in distributed hydrological models could propagate and magnify errors when using high-resolution data. For this reason, distributed models can underperform when compared to a *well-calibrated* lumped model in cases of uniform precipitation. This means that distributed approached may not always yield improved outlet simulations.

The authors based their work on previous studies which evidenced that, for some cases, runoff volumes and peak flows can vary considerably between spatially uniform rainfall and spatially distributed rainfall patterns. However, they do recognize that there are circumstances where spatial variability might not be great enough to produce variability on the observed basin response. This can occur due to intrinsic smoothing and dampening properties of basins, as well as different types of storm event (see Figure 2.1). *Convective* storms are characterized by tall, towering cloud formations, product of intense heating at ground level, which can yield intense and highly focalized rainfall. *Stratiform* storms exhibit layered, extensively horizontal cloud formations which usually present continuous and uniformly intense rainfall.

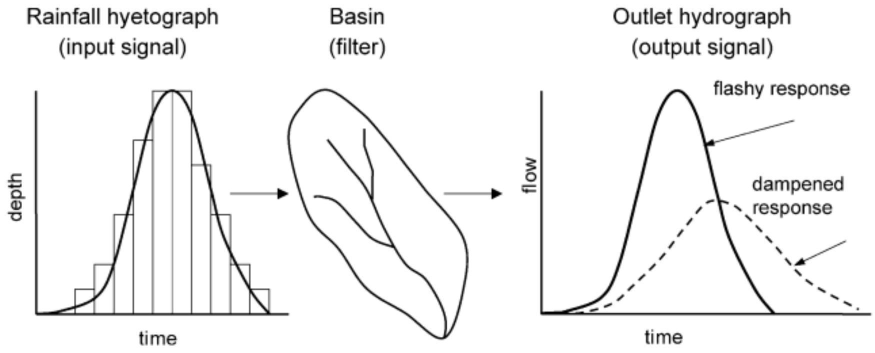

Figure 2.1: Effect of basin filtering on outflow response [3]

Their main hypothesis is that *Basins characterized by (1) marked spatial variability in precipitation, and (2) less of a filtering effect of the input rainfall signal will show improved outlet simulations from distributed versus lumped models.* In order to test this, the authors propose several indices for qualifying the observed basin outflow sensitivity, and spatially variable precipitation. These diagnostic indicators, which are derived from the observed data, allowed to formulate inferences to assess the dynamic characteristics of a basin's response.

First, the *index of rainfall location* I_L quantifies the generalized location of storms over the basin: if $I_L < 1$, rainfall is localized closer to the basin's outlet; if $I_L > 1$ the center of rainfall is located closer to the headwaters of the basin; if $I_L = 1$ indicates that rainfall is concentrated around the basin's center of mass (centroid). Secondly, the *index of general rainfall variability* I_σ quantifies the *instrastorm* rainfall variability for a given event.

In order to characterize and quantify measures of basin dampening, these indices were paired with extensive outflow hydrograph variability analysis using signal processing techniques. This variability was defined was defined in terms of filtering or dampening performed on the input rainfall signal, as measured in the basins outlet. These effects are portrayed in Figure 2.1, as the transformation of a input signal into an output signal, in which the shape of the resulting hydrograph is product of the combined effects of all of the basin's processes. Additionally, the effects of rainfall spatial variability are implicitly present in the transformation. Ultimately, this study was able to concretely tie and relate the gains in performance of distributed models over lumped models to specific characteristics in each basin, and spatial properties of precipitation events which took place in these basins.

In Zoccatelli *et al.* [4], the authors present a thorough analytical approach towards further characterizing spatial variability of rainfall for flash flood modeling. Their approach is based concretely on the spatial variability of rainfall-excess, measured over the distance from certain point in the catchment to the outlet (flow distance), along the flow direction. This approach derives from previously existing efforts referred to as the WS method, which was developed by Woods and Sivapalan (1999). The WS Method revealed that the impact of spatial variability of rainfall excess on simulated hydrograph shapes is controlled by the averaging of space-time rainfall excess fields across locations with equal flow distances. These results suggest that the sensitivity of hydrograph shapes to rainfall excess spatial variability is related to the mean and variance (first two statistical moments) of the distribution of rainfall-excess weighted flow distance.

The authors modify the WS methodology framework to derive two spatial rainfall statistics that condense the rainfall spatial patterns, aiming to improve runoff modeling. Fist, the *normalized time distance* θ_1 provides a notion of whether the spatial distribution of rainfall is concentrated towards the outlet ($\theta_1 < 1$), the headwaters ($\theta_1 > 1$), or the centroid of the catchment ($\theta_1 = 1$) (case which can be also understood as uniformly distributed rainfall). This is achieved by comparing the mean flow routing time with the averaged time it takes to route runoff from the basin's centroid to the outlet (similarly as proposed by Smith *et al.* [3]). Secondly, the *normalized time dispersion* θ_2 expresses how the rainfall is concentrated over the catchment: unimodal spatial distribution (rainfall localized somewhere over the catchment, $\theta_2 < 1$), bimodal spatial distribution (rainfall localized both at headwaters and outlet, $\theta_2 < 1$), uniform spatial distribution ($\theta_2 = 1$). This is expressed as the ratio between the variances of the flow routing time and the travel time.

Having readily prepared and analyzed the spatial variability indices, the authors performed an analysis of runoff model sensitivity to spatial rainfall variability. First, a baseline was established by computing the indices by assuming a uniform runoff coefficient, which was later compared to the ones obtained on the event-accumulated rainfall fields. These results showed that both statistics (θ_1 and θ_2) show a good correlation, and they seem to behave in a consistent way across most of the data. Subsequently, the effects of neglecting the spatial distribution of rainfall were tested by simulating each case with the actual rainfall and contrasting the results with simulations using spatially uniform precipitations.

Overall results show that neglecting spatial variability results in a considerable loss of simulation efficiency, which elucidates some of the influence of rainfall spatial variability on runoff modeling. An additional analysis was performed on the above results, by using a general rainfall spatial variability index I_σ, based on the one proposed by Smith *et al.* [3].

In Zoccatelli *et al.* [5], the authors build upon previous work [4] in order to redefine and describe a set of spatial rainfall statistics which describe rainfall spatial organization in terms of concentration and dispersion, as a function of the distance measured along the flow routing coordinate. Spatial organization is understood as the systematic spatial variation of rainfall with respect to certain basin geomorphic properties which directly control the runoff response. This updated approach uses rainfall spatial organization measured along the river network by using the flow distance coordinate: distance measured along the runoff flow path from a given point to the outlet [3] [4].

Still based on the WS methodology, but now including the developments by Viglione *etc.* (2010), the authors reformulate the spatial moments of catchment rainfall, aiming to provide a synthesis of the the interaction between the space-time variation of rainfall and basin morphologic properties (runoff coefficient, hillslope and channel routing, *etc.*), as well as quantifying their impact (delay and spread) on the resulting flood hydrograph. Firstly, the moments of catchment rainfall (p_0, p_1, p_2) and flow distance (g_1 and g_2) are introduced (smimilar to θ_1 and θ_2 used in [4]) as means for calculating δ_1 and δ_2. Similar and familiar formulations for δ_1 and δ_2 are presented as *scaled* moments of catchment rainfall, with the distinction of clarifying that values of $\delta_2 > 1$ (which are rare) indicate cases of multimodal rainfall distributions. Refer to Figure 1.4 for an illustration of these two indices. Additionally, temporally-averaged (event-based) version of these moments are introduced: Δ_1 and Δ_2.

The statistic Δ_1 measures the *hydrograph timing shift*, relative to the position of the rainfall centroid over the catchment. This statistics is also an indicator of mean time shift between hydrographs produced using the actual rainfall pattern for an event compared to the uniform precipitation baseline. Less-than-one values of Δ_1 intricate an anticipation of the mean hydrograph time with respect to the case of spatially uniform data; values larger than 1 represent the opposite. Δ_2 represent the ratio between the differential variance in runoff timing generated by rainfall spatial distribution and the variance of the catchment response time. Values of Δ_2 equal to 1 implies spatially uniform rainfall, and values lower 1 indicate that the precipitation is concentrated somewhere over the basin. Cases for values greater than 1 are rare, and indicate a bimodal (or multimodal) concentration of the rainfall (both at the headwaters and the outlet). As stated by the authors, in general the parameter Δ_1 is expected to influence the runoff timing, while Δ_2 affects the shape of the hydrograph and the value of the flood peak.

Ultimately, these renewed spatial rainfall statistics assess the dependence of the catchment flood response on the space-time interaction between rainfall and the spatial organization of catchment flow pathways. The first two spatial moments (δ_1 and δ_2) allowed to quantify the impact of rainfall spatial organization on two fundamental properties of the flood hydrograph: timing and amplitude. They also effectively allowed to describe the degree of spatial organization and quantify the relevance of rainfall spatial variability (in terms of timing error), which impact runoff modeling and flood modeling respectively. The main strength of this approach was a better understanding of the linkages between the characteristics of rainfall spatial patterns with the shape and magnitude of the catchment flood response, which was applicable across basins and scales (due to the scaling of moments).

In Douinot *et al.*, the authors present a new approach based on the Flash Flood Guidance (FFG) methodology (Mogil et al., 1978) which is widely used for flash flood forecasting throughout the US. It's defined as "the threshold rainfall [L] over accumulation periods of 1, 3 and 6 hours required to initiate flooding on small streams that respond to rainfall within a few hours" (Georgakakos, 1986; Sweeney, 1992). The term flash flood refers to sudden floods having high peak charges in a short response time. This short and rapid flood response is usually associated with watershed characteristics such as small catchments or steep slopes. Generally, the rapidity of these hydrological responses (within a few hours, up to a day) reduces the forecast time, and short lead times often prevent real-time observations of discharge and rainfall from being accurately assimilated into models. Therefore, forecasting methods should be achieved over small scales in both space and time.

The authors propose a new method for forecasting flash floods, named Spatialized Flash Flood Guidance Method (SFFG), aiming to improve the performance of the current FFG method while retaining its operational simplicity. Given that distributed hydrological models had shown significant improvements after including the local aspects of precipitation, a physically-based distributed hydrological model was used for both FFG and SFFG. In order to incorporate spatial information from rainfall data, the authors resort to Zoccatelli's spatial moments of precipitation [5] (δ_1, δ_2), which provide a description of the interaction between spatial rainfall organization and basin morphology. It should be noted that the authors took the liberty to rewrite Zoccatelli's formulation in a simpler, more straightforward way by redefining the *flow distance average*.

In order to calculate threshold intensities that integrate rainfall spatial information, rainfall forcing is was assumed to be spatially uniform anymore. This newly defined SFFG method accounted for global spatial variability of forecasted storms through δ_1 and δ_2 (it should be noted that the temporal dimension is ignored). This way, rainfall spatial distribution with specific (δ_1, δ_2) values were used to force the distributed hydrologic model and calculate threshold intensities. Overall, the spatial distribution of rainfall events was found to have a significant effect on the calculation of threshold intensities, and flash flood forecasting was found to be sensitive to upstream-downstream location of storms. This was consistent with Zoccatelli *et al.* [4] [5] and other authors which show the significant influence of δ_1) on flash and moderate flood response timing. The spreading index δ_2) was found to have a major effect on the amplitude of the flood, but almost negligible effects in terms of the timing of the hydrological response; so it doesn't significantly impact flood rising alerts. Also, the authors highlight that the interaction between the spatial distribution of rainfall and the spatial distribution of the storage capacity of the catchment could lead to either an attenuation or an amplification of the hydrological response, as stated by Smith *et al.* [3].

In conclusion, the proposed SFFG method provided encouraging improvements when compared with the FFG method: it offered the potential to analyze the sensitivity of hydrological responses to the spatial characteristics of the precipitation events as a function of the forecast lead time. However, any improvement in calculating the threshold intensity using SFFG should not be taken for granted, given that the effect of spatial variability of rainfall events was only significant for events of large amplitude. Factors other than the spatial distribution of rainfall probably influenced the results, and thus the effect of its interaction with other spatial distributions such as soil properties should be taken into account.

In Emmanuel *et al.* [7], the authors begin by acknowledging that the link between rainfall space-time variability and hydrological modeling is still an open issue in hydrology. Studies have compared the performance of hydrological models obtained through several rainfall estimation scenarios which include only rain gauge data, weather radar data or a combination of both. By doing so, different levels of rainfall spatial variability are corresponded, however, the influence of rainfall measurement errors are also indirectly introduced. Even though, most of these studies confirm the benefit of a spatially-detailed representation of adjusted radar images (bias correction using rain gauges), the influence of rainfall measurement errors on runoff modeling can still be significant.

The impact of rainfall spatial variability on runoff modeling at the catchment scale depends on the combined influence of several factors: rainfall patterns, catchment characteristics, and runoff generation processes. Studies on the topic (like the ones mentioned above) generally compare observed hydrographs to modeled hydrographs, which were obtained by forcing (precipitation through) a distributed hydrologic model using various spatial resolutions (high resolution radar images to catchment-averaged rainfall), have provided results and conclusions which shown contrasts and differences among them. These studies have also highlighted the difficulties involved in evaluating said influence. These include rainfall and outflow measurement errors, as well as modeling errors which can not be distinguished from the influence of spatial variability.

The authors state that by relying on a simulation approach can be helpful in: (1) deriving a better understanding of the way rainfall spatial variability propagates in the catchment; (2) exploring various and contrasted situations and controlling catchment characteristics; and (3) proposing a procedure to evaluate the influence of rainfall spatial variability on runoff modeling at the catchment-scale. More importantly, by proceeding by simulation would allow to control and eliminate error sources intrinsically present within streamflow and precipitation measurements. For this purpose, a simulation chain was developed, capable of simulating rainfall, stream networks and model hydrological processes product of their interaction (distributed hydrological model). Using this simulation chain, a simulated event database was created, which grouped contrasted simulation scenarios composed from combinations of four different simulated catchments and 6 distinct rainfall configurations.

In this study, the authors use this simulation chain-generated dataset to test the pertinence of the spatial variability indices proposed by Zoccatelli *et al.* ($\Delta_1, \Delta_2,$) [4] [5], and improve upon them. Results confirm the findings exposed by Zoccatelli *et al.*, in that for a given catchment, the influence of spatial variability of precipitation on basin response depends on the contrast of rainfall amount between upstream and downstream areas. Furthermore, given these results, the authors propose two new additional indices to represent rainfall spatial organization relative to the distance along the stream network from the outlet.

For these new moments, they relied on the concept of a *width function* $w(x)$, usually defined as the portion of the basin area at flow distance x of the outlet. A new *precipitation width function* $w_p(x)$ was proposed, as the proportion of rainfall on the catchment falling at a flow distance x from the outlet. Thus, by comparing $w(x)$ to $w_p(x)$ the influence of rainfall spatial organization on basin's response could be assessed. For this comparison, the authors propose to compare cumulative distribution functions of these width functions by using two criteria: the first index *vertical gap* (VG), is the absolute value of the maximum vertical difference between $w(x)$ and $w_p(x)$; the second index *horizontal gap* (HG), is the corresponding difference between $w(x)$ and $w_p(x)$, divided by the length of the longest hydrological path of the catchment. VG values close to 0 indicate weak spatial rainfall variability over the catchment, and the higher these values are, the more concentrated the rainfall is over a small portion of the catchment. HG values close to 0 indicate that rainfall is either distributed close to the catchment centroid or distributed uniformly. Values of HG other than 0 indicate rainfall concentration downstream ($HG < 0$) or upstream ($HG > 0$) of the catchment centroid. Figure 2.2 illustrates the comparison of $w(x)$ and $w_p(x)$ accumulations, showing the presence of VG and HG.

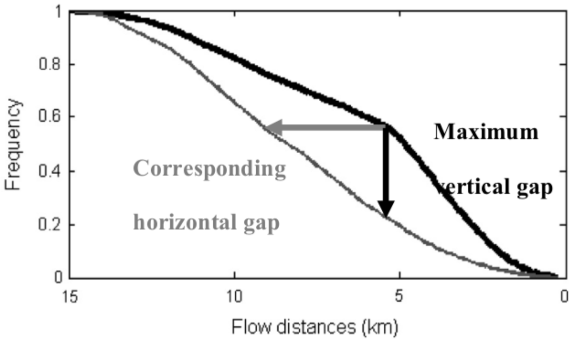

Figure 2.2: Distribution of $w_p(x)$ (black) and $w(x)$ (gray) rainfall accumulations [7]

Ultimately, the authors found that Δ_1 and HG appear to be highly correlated, and Δ_2 does not appear to hold significant correlation with the other indices. Moreover, a combination of VG and HG seem to hold strong explanatory power for the catchment response. Therefore, these indices yet again, through a rigorous simulation approach, prove useful in characterizing basin response. They also note that these newly proposed indices may explain better the impact of rainfall variability on hydrograph amplitude, than the ones proposed by Zoccatelli *et al.* and Smith *et al.*.

From the above literature review, it can be seen that these catchment-scale precipitation moments have been proven to encapsulate and describe the spatial variability of rainfall events, as well as their interactions with each catchment. Because of these properties they are natural candidates for the data-driven approach proposed in this current study. Table 2.1 describes the spatial moments of catchment rainfall, and associated indices included in the Spatial Precipitation Moment Flood event database.

Variable	**Name**	**Index**	**Source**
P0	0-th order moment of catchment precipitation	p_0	Smith / Zoccatelli
P1	1st order moment of catchment precipitation	p_1	Smith / Zoccatelli
P2	2nd order moment of catchment precipitation	p_2	Smith / Zoccatelli
G1	1st order moment of flow distance	g_1	Smith / Zoccatelli
G2	2nd order moment of flow distance	g_2	Smith / Zoccatelli
delta1	Catchment-averaged flow distance with respect to the catchment centroid	Δ_1	Smith / Zoccatelli
delta2	Rainfall field dispersion with respect to its mean position	Δ_2	Smith / Zoccatelli
EcartVertical	Vertical Gap: vertical difference between $w(x)$ and $w_p(x)$	VG	Emmanuel
EcartHorizontal	Horizontal Gap: corresponding difference between $w(x)$ and $w_p(x)$, divided by the length of the longest hydrological path of the catchment	HG	Emmanuel
precip_mean	Mean of precipitation accumulated during the centroid lag time period over the activated basin	μ_p	Saharia & Kirstetter
precip_sdev	Standard deviation of precipitation accumulated during the centroid lag time period over the activated basin	σ_p	Saharia & Kirstetter
precip_skew	Skewness of precipitation accumulated during the centroid lag time period over the activated basin	γ_p	Saharia & Kirstetter
precip_kurt	Kurtosis of precipitation accumulated during the centroid lag time period over the activated basin	κ_p	Saharia & Kirstetter
flowdist_mean	Mean of flow distance of the activated basin	μ_f	Saharia & Kirstetter
flowdist_sdev	Standard deviation of flow distance of the activated basin	σ_f	Saharia & Kirstetter
flowdist_skew	Skewness of flow distance of the activated basin	γ_f	Saharia & Kirstetter
flowdist_kurt	Kurtosis of flow distance of the activated basin	κ_f	Saharia & Kirstetter

Table 2.1 continued from previous page

Variable	Name	Index	Source
prod_mean	Mean of the product of accumulated precipitation and flow distance of the activated basin	μ_{pf}	Saharia & Kirstetter
prod_sdev	Standard deviation of the product of accumulated precipitation and flow distance of the activated basin	σ_{pf}	Saharia & Kirstetter
prod_skew	Skewness of the product of accumulated precipitation and flow distance of the activated basin	γ_{pf}	Saharia & Kirstetter
prod_kurt	Kurtosis of the product of accumulated precipitation and flow distance of the activated basin	κ_{pf}	Saharia & Kirstetter

Table 2.1: Catchment-scale precipitation moments

Notice that within these available catchment-scale precipitation moments, the first four statistical moments (mean, standard deviation, skewness and kurtosis) were also calculated by Dr. Saharia and Dr. Kirstetter for each event's precipitation, flow distance and their product. This was done as an effort to propose precipitation moments that are comparable and generalizable in a broader sense than the ones proposed by the literature. Traditional hydrology approaches rely on the characterization of phenomena and events over a select group of basins, under the assumption that these characterizations are generalizable to other cases. Conversely, a more generalized, systematic and data-driven approach is sought after by characterizing spatial variability with these statistical moments.

Chapter 3

Methodology

The methodology followed during the execution of this project derives from the Cross-Industry Standard Process for Data Mining (CRISP-DM) [13]. CRISP-DM is a data-centric, standardized, iterative knowledge discovery process composed of six distinct phases: project understanding, data understanding, data preparation, modeling, evaluation and deployment. Figure 3.1 shows a descriptive diagram of the process.

The project understanding phase was fulfilled over the first two chapters of this document (Introduction and Literature Review), where the problem at hand is introduced and the project objectives are defined. This chapter will cover the the phases corresponding to data understanding, data preparation and modeling (partially), whereas Chapter 4 will deal with the outcomes of modeling and the evaluation phase. Finally, Chapter 5 will treat aspects of the last CRISP-DM phase (deployment), but the extent of these conclusions will pertain to the exploratory nature of this study.

3.1 Data

The complete dataset provided by Dr. Manabendra Saharia for the development of this study was comprised of 21,143 observations for 133 variables. These variables include morphological, bioclimatic, climatological, precipitation and gauge data from 17,491 rainfall events across 902 different basins over the Contiguous United States (CONUS). Among these variables, various *precipitation moments* are present as well, such as the ones proposed by Zocattelli *et. al* [5] [6] [3] [7], as well as others proposed by Dr. Saharia and Dr. Pierre E. Kirstetter: first three statistical moments of spatial rainfall distribution, normalized flow distance and their products (9 in total) for each event. These variables are all summarized and defined in the **Appendix** item Table 5.1, and Table 3.1 shows the names and types of the 57 variables which were selected through the process described in the remainder of this chapter.

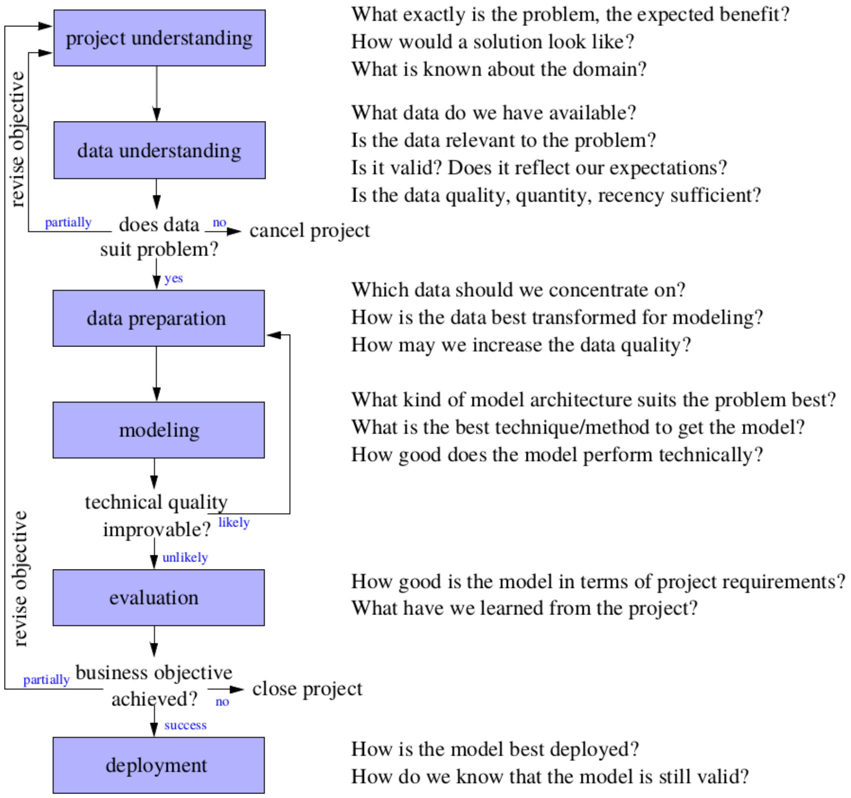
project understanding
data understanding
partially
does data suit problem?
no
cancel project
yes
data preparation
modeling
technical quality improvable?
likely
unlikely
evaluation
partially
business objective achieved?
no
close project
success
deployment
revise objective
revise objective
What exactly is the problem, the expected benefit?
How would a solution look like?
What is known about the domain?
What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?
Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?
What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How good does the model perform technically?
How good is the model in terms of project requirements?
What have we learned from the project?
How is the model best deployed?
How do we know that the model is still valid?

Figure 3.1: Diagram of the CRISP-DM methodology [14]

VARIABLE	TYPE
est_area	Morphological
rl	Morphological
rr	Morphological
si	Morphological
slopeoutlet	Morphological
precip	Climatological
temp	Climatological
cnbasin	Morphological
cncell	Morphological
coemcell	Morphological
imperviousbasin	Morphological
imperviouscell	Morphological
kfact	Morphological
rockdepth	Morphological
rockvolume	Morphological
bpartexture	Morphological
lbm	Morphological

Table 3.1 continued from previous page

VARIABLE	TYPE
ruggedness	Morphological
rt	Streamflow
mf.event	Streamflow
tp	Streamflow
activatedBasinPixels	Streamflow
totalBasinPixels	Morphological
precip_mean	Precipitation Moment
precip_sdev	Precipitation Moment
precip_skew	Precipitation Moment
precip_kurt	Precipitation Moment
flowdist_mean	Precipitation Moment
flowdist_sdev	Precipitation Moment
flowdist_skew	Precipitation Moment
flowdist_kurt	Precipitation Moment
prod_mean	Precipitation Moment
prod_sdev	Precipitation Moment
prod_skew	Precipitation Moment
prod_kurt	Precipitation Moment
G1	Precipitation Moment
G2	Precipitation Moment
delta1	Precipitation Moment
delta2	Precipitation Moment
EcartVertical	Precipitation Moment
EcartHorizontal	Precipitation Moment
snowpercent	Morphological
bio_1	Bioclimatic
bio_2	Bioclimatic
bio_3	Bioclimatic
bio_4	Bioclimatic
bio_7	Bioclimatic
bio_8	Bioclimatic
bio_10	Bioclimatic
bio_11	Bioclimatic
bio_12	Bioclimatic
bio_15	Bioclimatic
bio_17	Bioclimatic
bio_18	Bioclimatic
lag_centroid_peak_event	Response
peakq_moment	Response
exceeds_threshold	Response

Table 3.1: Selected Variables

Preliminarily, the target variables of interest in this study were *lag time*, and each event's *peak discharge* with respect to the *flood stage exceedance thresholds* previously established for each basin outlet (action, minor, moderate, major). *Lag time* was calculated based on MRMS quantitative precipitation estimates (QPE) and USGS stream gauge observations, and was provided as part of the dataset by Dr. Saharia. The dataset also contained non-relevant attributes for the objectives of this study (*i.e.* IDs, flags, tags and arbitrary control/reference variables) which will be removed. A detailed account of this process and further dataset preparations will be provided in the following sections.

3.2 Preliminary Variable Selection

The dataset, as originally obtained, contained several variables that were vestigial from a quality control process performed in the selection of the rainfall events, gauges and basins affected by these events. These 34 variables were immediately identified upon inspection, and were removed from the dataset. Table 3.1 lists these variables and their reason for removal.

Variable	**Reason for removal**
fips	ID
gauge	ID
lat	Non-relevant for model
lon	Non-relevant for model
HUC	ID
agency	Non-relevant for model
gname	Non-relevant for model
cc	Quality control variable used while constructing the dataset
area	Same as usgs_area
regulation	All basins are 'regulated'
error	Quality Control variable used by the provider of the dataset
ldd	Only four distinct values were present in the data; for which only 3 basins have values different than 0
Group.1	ID
county	Non-relevant for model
prop	Non-relevant for model
state	Non-relevant for model
month	Non-relevant for model
year	Non-relevant for model
start	Not needed, as peak flow and flow duration times are provided through other variables
end	Not needed, as peak flow and flow duration times are provided through other variables

Table 3.2 continued from previous page

Variable	Reason for removal
fness	Categorical; basins with an f.ecdf value higher than 0.5 are considered ‘Flashy’
eventID	ID
gaugenum	ID
lag_start_peak_event	Non-relevant for model; we’re interested in lag time measured from the center of mass of rainfall to the peak flow
lag_max_peak_event	Non-relevant for model; we’re interested in lag time measured from the center of mass of rainfall to the peak flow
casetag	ID
mean	Quality Control variable used by the provider of the dataset
season	Categorical and non-relevant for model
maxseason	Categorical and non-relevant for model
class	Categorical and non-relevant for model
std	Quality Control variable used by the provider of the dataset
a1	Quality Control variable used by the provider of the dataset
a12	Quality Control variable used by the provider of the dataset
a2	Quality Control variable used by the provider of the dataset

Table 3.2: Preliminary Variable Removals

After removing these 34 variables, the working dataset was left with 21,143 observations for 99 variables. Further analysis and feature engineering of these remaining attributes will be presented in the following sections.

3.3 Feature Engineering

Having retained 99 variables from the original dataset, additional features were constructed in order to be explored as target variables. One of them was designed to simplify the relationship between event peak discharge with respect to the exceedance of flood thresholds (aiming to characterize the probability of exceeding pre-existing flood stages). The other proposed feature to be modeled was designed to describe the temporal distribution of peak discharge with respect to its corresponding rainfall event, in a generalized and comparable way. These two features will be described in detail below.

3.3.1 Moment of Relative Peak Discharge

The Moment of Relative Peak Discharge is proposed and defined as a scalar quantity, which characterizes whether an rainfall event’s peak discharge occurred near the beginning, middle or end of the precipitation event. It was conceptualized as:

$$\begin{aligned}\tau_{pq} &= 1 - \frac{(End\,of\;Event - Start\,of\;Event) - (Time\,of\;Peak\,Flow\; - Start\,of\;Event)}{(End\,of\;Event - Start\,of\;Event)} \\ &= 1 - \frac{fd - dt}{fd}\end{aligned} \tag{3.1}$$

such that:

$$\{\tau_{pq} | 0 \leq \tau_{pq} \leq 1\} = \begin{cases} \text{peak occurs near the beggining of the event,} & \text{if } 0 \leq \tau_{pq} \leq 0.33 \\ \text{peak occurs near the middle of the event,} & \text{if } 0.33 < \tau_{pq} \leq 0.66 \\ \text{peak occurs near the end of the event,} & \text{if } 0.66 < \tau_{pq} \leq 1 \end{cases} \tag{3.2}$$

This additional feature *peakq_moment* was computed by using the variables *fd* (flow duration) and *dt* (time difference between the start of the event and peak flow), and then added back to the dataset. The distribution of resulting *peakq_moment* values is shown in Figure 3.2.

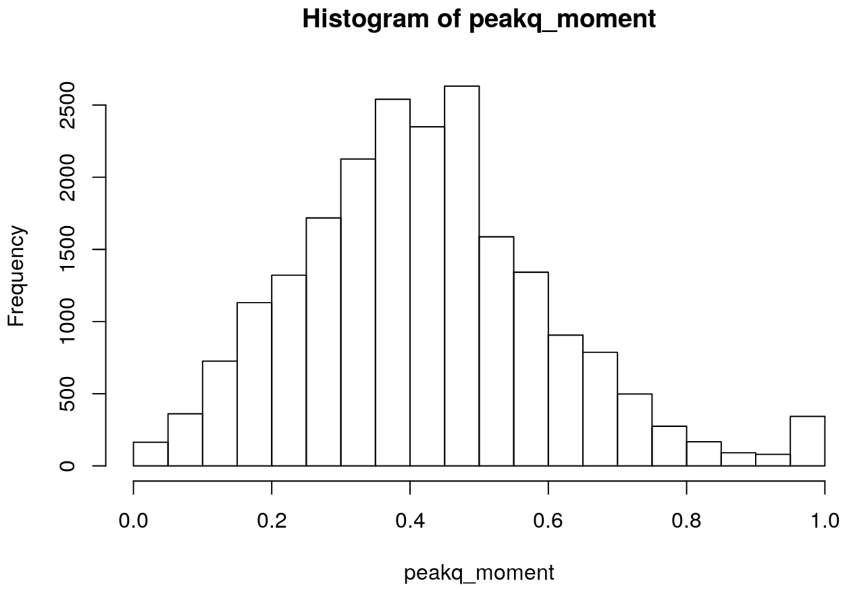

Figure 3.2: Moment of relative peak discharge histogram

3.3.2 Exceedance of Flood Stage Thresholds

In order to concisely characterize flood stage threshold exceedance, an encoding for when the peak discharge of a given event exceeded any of the defined thresholds (action, minor, moderate, major) for the basin over which it occurred was defined. In order to achieve this, four temporary new variables were created in the dataset: ‘Exceeded Major‘, ‘Exceeded Moderate‘, ‘Exceeded Minor‘ and ‘Exceeded Action‘. By assigning a binary value (yes/no, 1/0, True/False) to each of these columns, according to whether a given peak flow exceeded any of the aforementioned thresholds, and then collapsing these occurrences into a 4-bit binary number, final ’class’ labels were defined. These allowed to identify for each event whether any of the flood stage exceedance thresholds were exceeded, as well as identifying which ones. Table 3.3 illustrates the logic behind this encoding and class labels.

Class Label	**Exceeded Major**	**Exceeded Moderate**	**Exceeded Minor**	**Exceeded Action**
No Exceedance (0)	N	N	N	N
Exceeds Action (1)	N	N	N	Y
Exceeds Minor (2)	N	N	Y	Y
Exceeds Moderate (4)	N	Y	Y	Y
Exceeds Major (8)	Y	Y	Y	Y

Table 3.3: Flood Stage Exceedance Class Encoding

The histogram in Figure 3.3 shows the class label distribution of the whole working dataset:

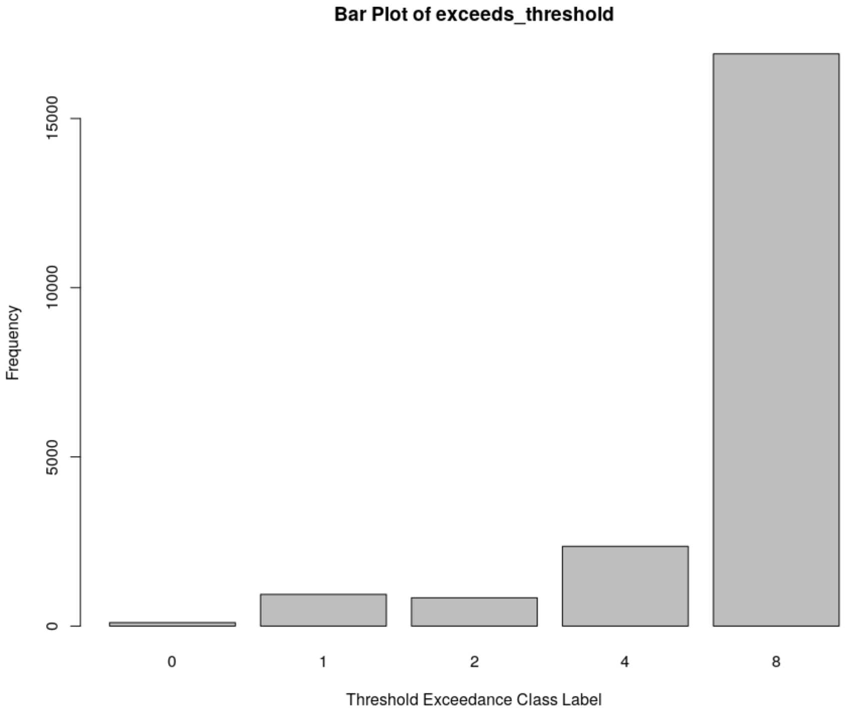

Figure 3.3: Flood Stage exceedance class label distribution

Lastly, having engineered these two new features, the original and temporal variables created used to construct them were removed from the dataset. Additionally, after consulting both Dr. Kirstetter and Dr. Saharia on the remaining pool of attributes, 29 additional variables were removed. These are listed in the **Appendix** Table 5.2.

At this point, the dataset was reduced to 21,143 observations for 78 variables. Both the Moment of Relative Peak Discharge and the Exceedance of Flood Stage Thresholds were then selected as target variables (attributes to be modeled from the rest), in addition to the originally selected Lag Time.

3.4 Data Transformation

The 78 selected variables from the original data set were further explored in terms of their density distributions. Histograms were plotted for each of the selected attributes, and their shape was observed and analyzed. Almost all of the available predictors exhibited a pronounced skewness in their distributions, and a wide range of value scales was observed in them: some variables include negative values, others include a large number of zeroes, and a few vary within very small or extremely large ranges of values. Because of the above, the decision to normalize the dataset was made. The normalization process was performed in order to maximize the efficiency of modeling techniques and algorithms which might be sensitive to skewness and scaling [15] [16]. Because of a large presence of zero-values, a logarithmic transformation was deemed inadequate, and because of the prominent presence of negative values in some of the predictors, a Box-Cox transformation would be unsuitable. Fortunately the Yeo-Johnson transformation provides a comparable method to the Box-Cox or Logarithmic transformations, but allowing for zeros and negative values to be transformed. The Yeo-Johnson transformation implementation available in the *bestNormalize* R package also allowed to compute the optimal parameter (λ) for the centering and scaling of the data. Each predictors density was then plotted alongside its optimal transformed counterpart in order to supervise the data standardization process. Figure 3.4 shows an example of this for the estimated area for each basin.

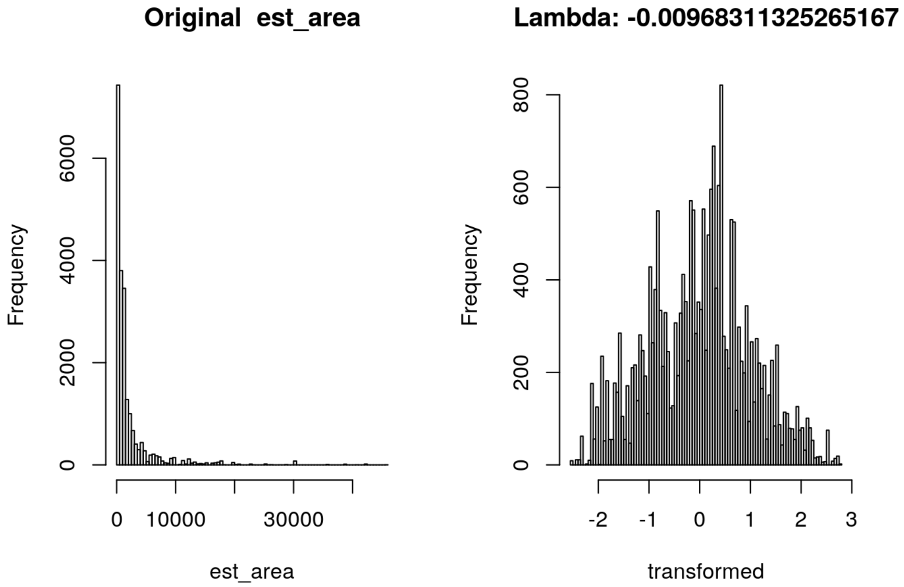

Figure 3.4: Example of variable standardization using the Yeo-Johnson transformation

It must be noted that the *exceeds_threshold* variable was not transformed given that it is the only categorical feature in the dataset. Through this transformation and inspection process, it was noticed that the variables *lbm* and *lfocf* had identical values and distributions, and therefore one of them was discarded (*lfocf*) reducing the number of variables to 77. Close inspection also revealed that certain variables were not scaled correctly by the Yeo-Johnson transformation (*rr*, *si*, *slopeoutlet* and *precip_mean*), and thus were log-transformed first, given that none of them held negative values, and then were transformed using Yeo-Johnson. This way, all continuous variables were normalized and held values within an order of magnitude of each other.

3.5 Correlation Analysis

Efforts were also made to explore the dataset with hopes to reduce the number of predictors that were to be used for the modeling phase. Two distinct correlation analyses were performed on the data set: a pairwise correlation analysis between predictors, and a correlation analysis between each predictor and each of the continuous responses (lag time, moment of relative peak discharge). The first analysis was performed through the construction of a correlogram (see Figure 3.5) which allowed to explore the strength of overall correlations between all feature pairs. It should be noted that the underlying structure of the correlogram was used to explore the correlations, and not the visualized diagram itself.

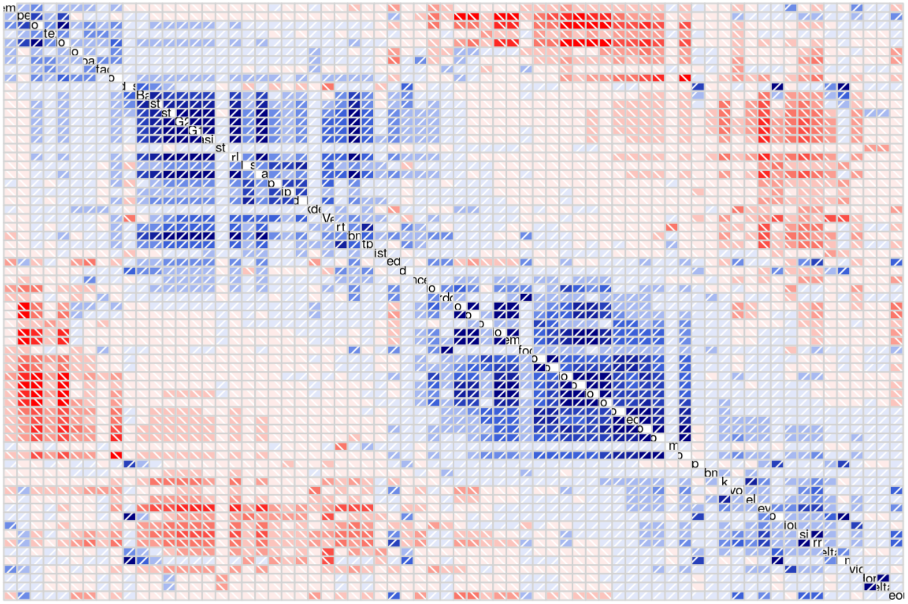

Figure 3.5: Correlogram built for the correlation analysis of the transformed dataset. Though not really useful for comparing this many variables, it highlights the high dimensionality of the working dataset

3.5.1 Pairwise Correlation

Highly correlated predictors were selected from the correlogram in order to determine if any of them could be further removed from the data set (given redundant explanatory power in highly correlated variables)[16] [15]. An absolute correlation of 0.75 was chosen as a diagnostic indicator of strong linear correlation between features. Strong correlations were defined as values between $[0.75, 0.80)$, values between $[0.80, 0.90)$ indicated high correlation, and values in the range $[0.90, 1.00]$ showed very high correlation. Evaluating these ranges on the results obtained revealed the following strong correlations:

- *bio_17* is very highly correlated to *bio_14*
- *prod_sdev* is very highly correlated to *prod_mean*
- *est_area* is very highly correlated to *rbm*
- *totalBasinPixels* is very highly correlated to *rbm*
- *EcartHorizontal* is very highly correlated to *delta1*
- *rdd* is very highly correlated to *rfocf*
- *bio_11* is very highly correlated to *bio_9*
- *G1* is very highly correlated to *rbm*
- *precip_mean* is very highly correlated to *prod_mean*
- *rl* is very highly correlated to *rbm*
- *G2* is very highly correlated to *rbm*
- *bio_6* is very highly correlated to *bio_9*
- *bio_3* is highly correlated to *bio_9*
- *flowdist_mean* is highly correlated to *rbm*
- *si* is highly correlated to *rr*
- *bio_19* is highly correlated to *bio_14*
- *temp* is highly correlated to *bio_9*
- *bio_1* is highly correlated to *bio_9*
- *bio_12* is highly correlated to *bio_14*
- *activatedBasinPixels* is highly correlated to *flowdist_sdev*
- *bio_16* is highly correlated to *precip*
- *bio_7* is highly correlated to *bio_19*
- *snowpercent* is highly correlated to *bio_6*
- *bio_15* is highly correlated to *bio_14*

- *prod_skew* is strongly correlated to *prod_kurt*
- *precip* is strongly correlated to *bio_14*
- *bio_10* is strongly correlated to *bio_6*
- *bio_9* is strongly correlated to *bio_19*
- *bio_13* is strongly correlated to *precip*
- *k* is strongly correlated to *el*
- *flowdist_sdev* is strongly correlated to *rbm*
- *bio_5* is strongly correlated to *bio_11*
- *rt* is strongly correlated to *tp*
- *bio_4* is strongly correlated to *bio_19*

Thus, the following 43 variables became candidates for removal: *activatedBasinPixels*, *delta1*, *EcartHorizontal*, *el*, *est_area*, *flowdist_mean*, *flowdist_sdev*, *G1*, *G2*, *k*, *precip_mean*, *precip*, *prod_kurt*, *prod_mean*, *prod_sdev*, *prod_skew*, *rbm*, *rdd*, *rfocf*, *rl*, *rr*, *rt*, *si*, *snowpercent*, *temp*, *totalBasinPixels*, *tp*, *bio_1*, *bio_3*, *bio_4*, *bio_5*, *bio_6*, *bio_7*, *bio_9*, *bio_10*, *bio_11*, *bio_12*, *bio_13*, *bio_14*, *bio_15*, *bio_16*, *bio_17*, *bio_19*. However, these will only be removed if they also lack any meaningful correlation with any of the target variables.

3.5.2 Response Correlations

The correlation analysis between the 74 attributes and the two continuous target variables lag time and the moment of relative peak discharge, was performed by calculating both Pearson's and Spearman's correlation in order to address both linear and ranked correlations. For this correlation analysis, an absolute linear correlation threshold of $|0.15|$ was defined in order to identify those variables that exhibit a quantifiable correlation, and this value was set to such a low number given the non-linear nature of these relationships.

Lag Time

Regarding lag time, the analysis revealed that 22 predictors exhibit correlation with *lag_centroid_peak*, which were deemed to hold some explaining power for building models:

- *est_area*: Estimated Area
- *rl*: River length

- *rr*: Relief ratio
- *si*: Slope index
- *rbm*: Basin magnitude, total number of first-order streams
- *imperviousbasin*: Basin total surface imperviousness
- *rt*: Recession time; peakq-to-end time
- *mf*: Basin median Flashiness
- *tp*: Rise time; start-to-peakq time
- *activatedBasinPixels*: Total number of 1km x 1km gridcells in a basin that received rainfall from centroid of precipitation to flow peak
- *totalBasinPixels*: Total number of 1km x 1km gridcells in a basin
- *precip_mean*: Mean of precipitation accumulated during the centroid lag time period over the activated basin(part of the basin where rainfall falls)
- *precip_sdev*: Standard deviation of precipitation accumulated during the centroid lag time period over the activated basin(part of the basin where rainfall falls)
- *flowdist_mean*: Mean of flow distance of the activated basin(part of the basin where rainfall falls)
- *flowdist_sdev*: Standard deviation of flow distance of the activated basin(part of the basin where rainfall falls)
- *prod_mean*: Mean of the product of accumulated precipitation and flow distance of the activated basin(part of the basin where rainfall falls)
- *prod_sdev*: Standard deviation of the product of accumulated precipitation and flow distance of the activated basin(part of the basin where rainfall falls)
- *prod_skew*: Skewness of the product of accumulated precipitation and flow distance of the activated basin(part of the basin where rainfall falls)
- *G1*: First-order Moment of flow distance (Catchment averaged flow distance)
- *G2*: Second-order Moment of flow distance
- *delta2*: Rainfall field dispersion (with respect to its mean position) relative to the dispersion of the flow distances
- *EcartVertical*: Vertical Gap, the higher the VG value, the more concentrated the rainfall over a small part of the catchment

These 22 variables become now candidates for selection (being kept instead of discarded due to quantifiable correlation with the response). Figures 3.6 and 3.7 show bar plots of these correlations, with relationship to the defined thresholds.

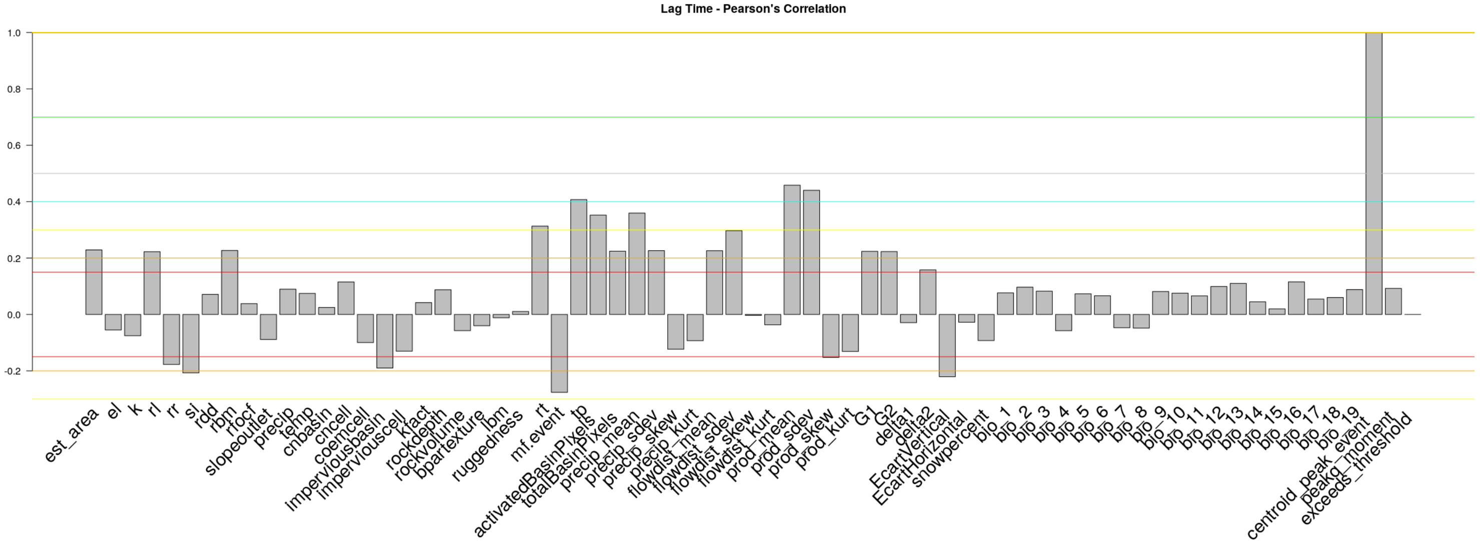

Figure 3.6: Bar plot: Pearson's Correlation for Lag time

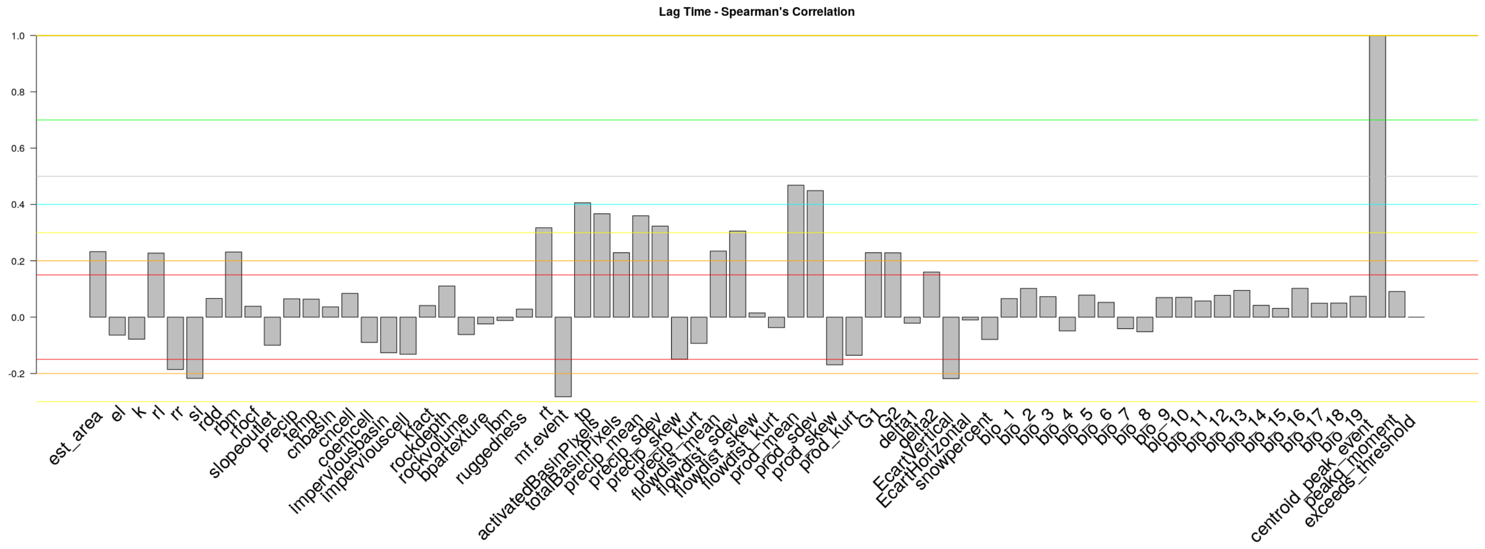

Figure 3.7: Bar plot: Spearman's Correlation for Lag time

Moment of relative Peak Discharge

The same analysis performed for Lag Time was done for the 74 attributes and *peakq_moment*. In this case, correlations between predictors and this response appeared to be extremely low. So low that only the variables *tp* and *rt* show any noticeable correlation, given that they are intrinsically related with how the moment of relative peak discharge was calculated, as they all describe the flow hydrograph for each event. Thus, they are both candidates for selection for modeling lag time and flood stage threshold exceedance, but should be discarded to model *peakq_moment*. Figures 3.8 and 3.9 show bar plots of these correlations, with relationship to the defined thresholds.

Having performed these correlation tests for all attributes and two of the response variables, a final analysis of these results lead to the definition of a final predictor set, which was used in modeling all three target variables.

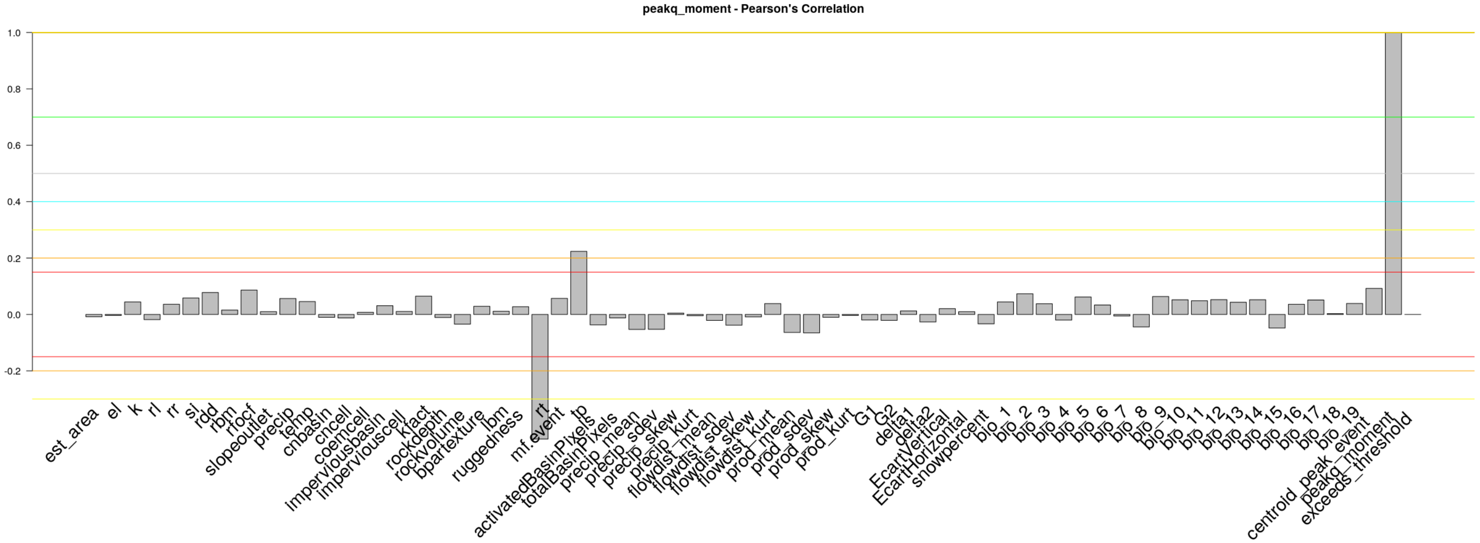

Figure 3.8: Bar plot: Pearson's Correlation for the moment of relative peak discharge

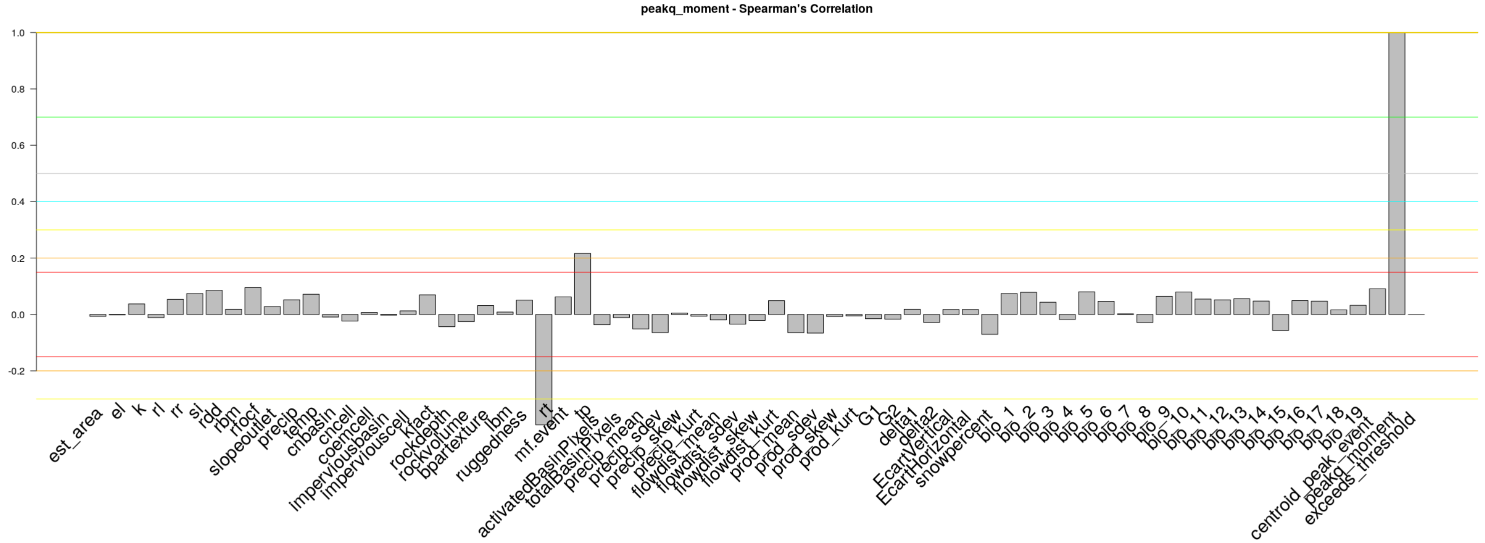

Figure 3.9: Bar plot: Spearman's Correlation for the moment of relative peak discharge

3.6 Final Data Selection and Partitioning

3.6.1 Final Predictor Set

Following the above analysis, some of the candidate variables for removal were expunged from the previously selected attributes: *el*, *k*, *rdd*, *rfocf*, *bio_5*, *bio_6*, *bio_9*, *bio_13*, *bio_14*, *bio_16* and *bio_19*. The following variables were revindicated by the second correlation analysis: *activatedBasinPixels*, *est_area*, *flowdist_mean*, *flowdist_sdev*, *G1*, *G2*, *precip_mean*, *prod_mean*, *prod_sdev*, *prod_skew*, *rl*, *rr*, *rt*, *si*, *totalBasinPixels*, *tp*,*bio_1*, *bio_3*, *bio_4*, *bio_7*, *bio_10*, *bio_11*, *bio_12*, *bio_15* and *bio_17*. At this point, it should be noted that *totalBasinPixels* and *est_area* are very highly correlated, and are analogous. Due to a mistake in the construction of this final predictor dataset, both of them were kept and used for modeling and this fact should be kept in mind when analyzing the results in the next section.

Even though *rbm* was revindicated by the second correlation analysis, it was expunged as well because it had high correlation with 7 other variables. Conversely, the following variables were kept regardless of having been selected for removal and not being revindicated by the second correlation analysis because they are of particular interest to this study: *delta1*, *EcartHorizontal*, *precip*, *prod_kurt*, *snowpercent* and *temp*. Variables that did not *pop up* in the correlation analyses were kept as well. These remaining variables will be further studied through predictor importance analyses to determine how they contribute to the prediction of the response variables.

After having performed the aforementioned correlation-supported variable selection/removal from the transformed dataset, 57 variables were left in the working dataset (54 predictors and 3 target variables), which still held 21,143 observations. These 57 variables are detailed in Table 3.1, shown at the beginning of this chapter.

3.6.2 Data Partitioning

Even though the training processes were carried out implementing cross-validation, an additional validation hold-out set was extracted from the working dataset. This allowed for a robust assessment and validation of the models constructed in this study, by examining their performance on previously unseen data. Given the large number of instances in the dataset, an 80-20 split was chosen: 80% of the data was going to be used for training and testing (using cross-validation) machine learning models, and the remaining 20% was used as validation of said trained models. This allows for training the best possible model on a large portion of the data, and also test its performance on a smaller but representative set of unseen data as a way to establish feasible realistic performance estimates [16] [15].

This split left the training dataset with 16,914 observations, and the validation dataset with 4,229 observations. Figures 3.10, 3.11 and 3.12 illustrate the split frequency of each response variables, for both datasets. These were used to validate that the value distributions remained similar/representative across both datasets.

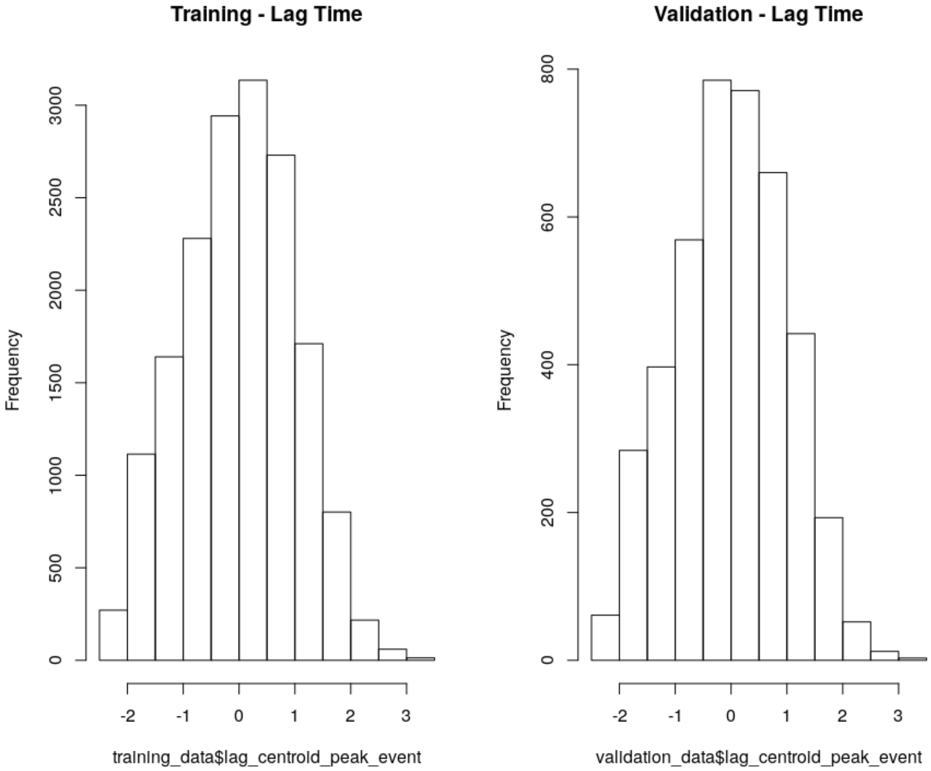

Figure 3.10: Training/validation dataset split - distribution of lag time

The Figures 3.10, 3.11 and 3.12 show that the distributions for the studied responses remained similar for both the training and validation datasets. Therefore, validation dataset was apt for verifying models built on the training dataset.

3.7 Modeling

Given the multidimensional and non-linear nature of the phenomena this project aims to model, three non-linear regression approaches based on diverse statistical, computational and learning techniques were selected to be explored during the modeling phase:

- **MARS** Multilinear Adaptive Regression Splines - multidimensional, segmented spline-based method using piecewise linear-like regressions to model non-linear problems in n-dimensional spaces. It is highly efficient, and it's able to rank and select variables that build an optimal model. [16][15][17].

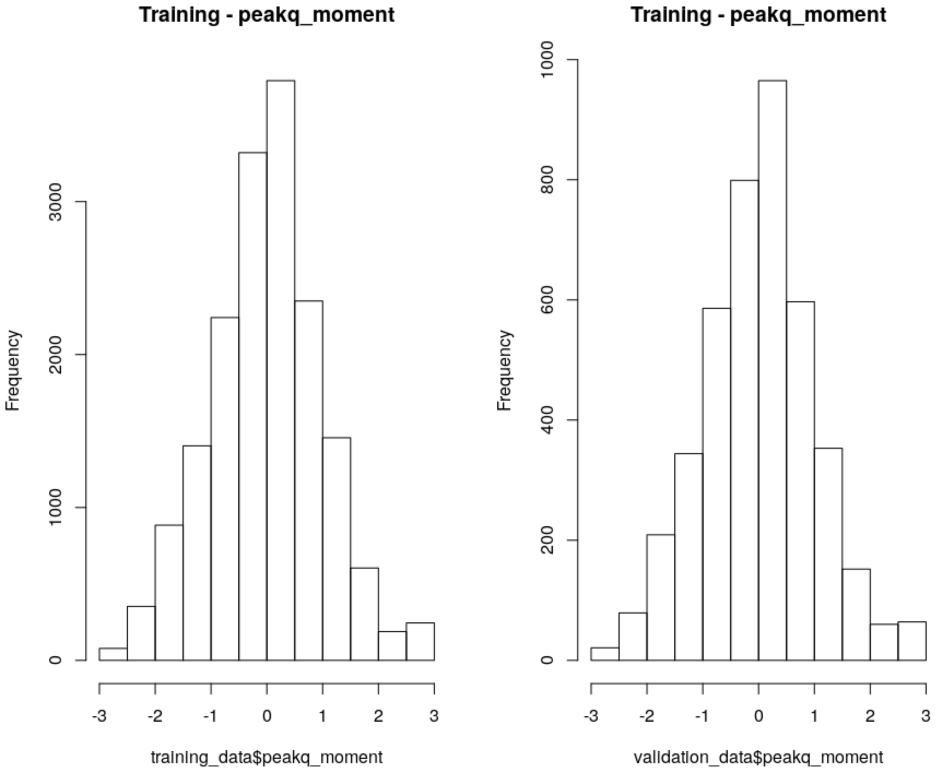

Figure 3.11: Training/validation dataset split - distribution of moment of relative peak discharge

- **Support Vector Machines** non-linear, multidimensional technique, which is a able of performing classification and regression. Support vectors are critical boundary instances derived from each class of the dataset. This technique relies on the use of kernel functions to perform higher-order spatial transformations, where high-order decision boundaries are established in order to separate said support vectors. Requires extensive parameter tuning, but provides accurate results while maintaining the interpretability of the model (unlike, for example, neural networks)[16][15][17].
- **Random Forest** versatile bagged decision tree approach, mainly intended for classification, which can also be used for multidimensional non-linear regression models. It is highly robust to outliers in the data, as well as scaling and non-normalized predictors [16][15][17].

Given that both MARS and Random Forest incorporate automatic variable selection, ranking and importance capabilities, contrasting their outputs will provide an interesting and robust assessment of the relevance of the selected predictors in the dataset, for characterizing the selected target variables.

Regarding the MARS approach, the models were parameterized to perform an importance evaluation of the input variables regarding their contribution to the minimization

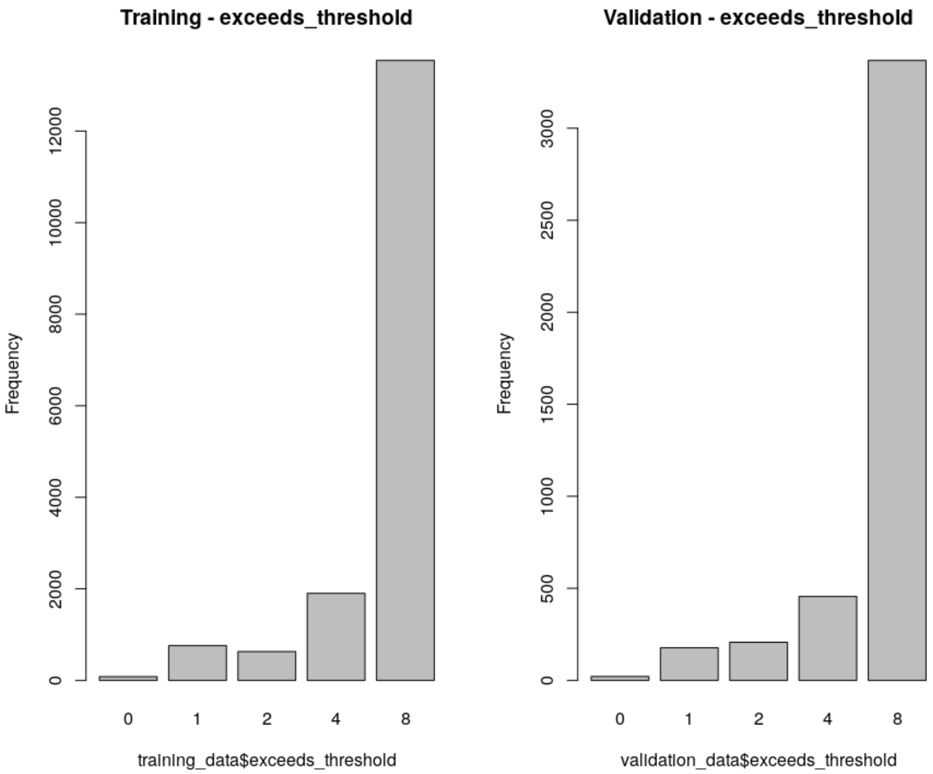

Figure 3.12: Training/validation dataset split - distribution of flood stage threshold exceedance

of errors (or increase in accuracy) of the final model. Training was also configured to perform a grid search-based parameter tuning for the optimal number of model terms to retain (from 2 up to 54) in the final model, as well as the optimal degree of interaction between predictors (from 1 up to 5). Residual plots and training analyses were performed in order to check the modeled response for possible outliers and other artifacts.

For the random forest approach, variable importance analysis was performed based on their contribution to the minimization of errors (or decrease in accuracy), but also to their relevance in making splits (decisions to characterize the response) in each of the tree's nodes. The number of trees to train was chosen to be 100 in order to allow sufficient variability in models throughout the training process, and a tuning grid was configured to find out what the optimal number of variables available for splitting at each tree node should be. This bagged tree approach also provided us with sensible metrics on the amount of variance explained by the model, as a proxy measure of fitness, as well as Out of Bag error rates.

Finally, regarding the support vector machines approach, a radial basis function kernel ($e^{-\sigma|x-C|^2}$) was chosen to fit a multidimensional non-linear regression model. During training, a grid search-based tuning was performed in order to estimate the kernel function parameters σ and C. Values for both C and σ were allowed to vary greatly, between 0 and 5.

In the k-fold cross validation approach, the data is randomly divided into k subsets, such that each time, one of these subsets is used as the test set and the other remaining $k-1$ of this sets are put together as the training dataset. After having tested on all of the k subsets, the error estimation is averaged over all k trials, to estimate the total effectiveness of a model. This process is usually repeated for an additional n number of times (n repeats of). All models were trained using 10 repeats of 10-fold cross-validation in order to mitigate overfitting on the training dataset (by averaging error estimations for all 10 repeats, and for each 10 folds), and once trained these were also tested to predict known outputs on a validation (holdout, not included in training) dataset. For all three approaches standard error metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MPAE) were calculated, in order to provide a means for comparing the performance between these different approaches. Additional modeling metrics such as R^2, RSS, $Accuracy$, Cohen's $Kappa$ coefficient and linear Correlation Coefficients are provided as outputs from running and fitting each model. The following chapter will present the results obtained after training these models, and their respective performances will be analyzed.

All these models were trained under similar circumstances (similarly-spec'd hardware), making use of the parallel capabilities of R packages such as *caret* and its integration with the *doParallel* library. All training processes were executed using a pool of 8 dedicated cores for building each model, and training times are reported in Table 3.3. All models were trained on the same machine using an Intel® Xeon® E5-2687W v4 CPU, with 24 hyper-threaded cores (48 threads) running at a base clock of 3.00GHz.

Target	**Model**	**Problem Type**	**Training Time (hours)**
Lag Time	MARS	Regression	2.8
Moment of Relative Peak Discharge	MARS	Regression	3.6
Flood stage Threshold Exceedance	MARS	Classification	9.5
Lag Time	Random Forest	Regression	16.8
Moment of Relative Peak Discharge	Random Forest	Regression	16.3
Flood stage Threshold Exceedance	Random Forest	Classification	3.0
Lag Time	SVM	Regression	~200
Moment of Relative Peak Discharge	SVM	Regression	~230
Flood stage Threshold Exceedance	SVM	Classification	~150

Table 3.4: Model training times per target variable

Chapter 4

Results

After having analyzed, selected, transformed and partitioned the working dataset, three different machine learning models (MARS , Random Forest and Support Vector Machines) were built for each of the three target variables selected and constructed for this study: Lag Time, Moment of Relative Peak Discharge and Flood Stage Threshold Exceedance.

These models were trained on several servers with similar configurations and specifications, where more processing power was available, and dedicated scripts were built to execute and save model states and outputs. These serialized model objects were then downloaded and unpacked for analysis and validation in a workstation. Training and validation results for each of these models will be presented, analyzed and discussed in this chapter. A copy of these scripts can be found in the **Appendix** section *Model training scripts.*

4.1 MARS

First, Lag time (*lag_centroid_peak_event*), the Moment of Relative Peak Discharge (*peakq_moment*) and the Flood Stage Threshold Exceedance (*exceeds_threshold*) were modeled by fitting parameter-tuned MARS models, which explored combinations of parameters (degrees of interaction x number of terms to retain) using a tuning grid. This way, optimal parameter settings were found for a model which would minimize error measures, or maximize performance measures. Additionally, these models were trained using 10 times 10-fold cross-validation in order to mitigate overfitting on the training dataset, and once trained these were also tested to predict known outputs on a validation dataset (holdout, not included in training).

4.1.1 Lag Time Modeling

This Lag Time model was trained using 8 dedicated cores, and took ~3 hours to complete. Parameter tuning was performed from 1 up to 5 degrees of interaction (model terms could be composed of products of up to 5 predictors), and from 2 up to 54 model terms (up to one term per predictor). The parameter tuning results during training are shown in Figure 4.1.

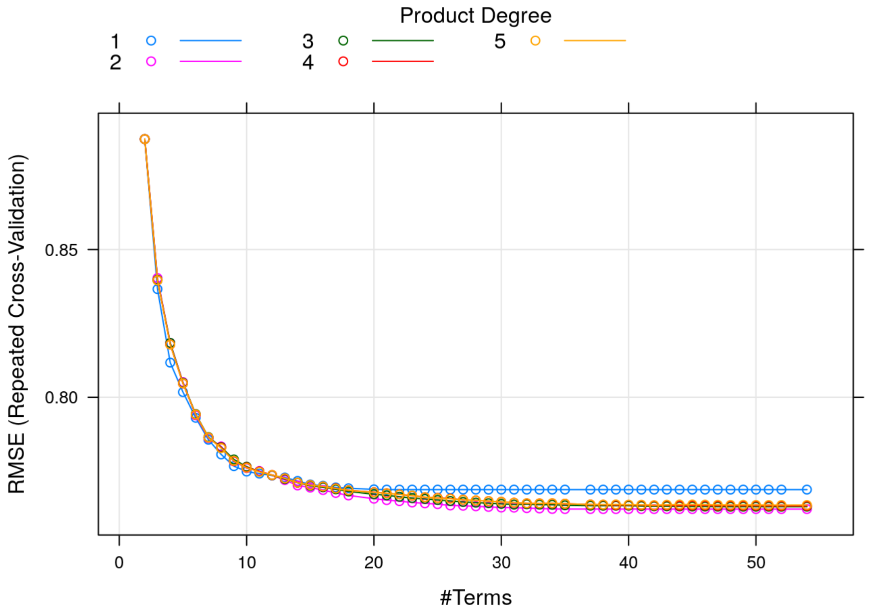

Figure 4.1: MARS: Lag Time Training - Parameter Tuning results

From this tuning grid results, the best fit was found to be a model with up to 39 terms, each with up to 2 degrees of interaction. Root Mean Squared Error was used to determine the model fitness throughout training. The structure and output of the best model found is shown in the **Appendix** on Listing 5.1.

The final model was constructed with 34 terms (17 of which where 2nd degree interaction terms) and using only 18 of of the 52 possible predictors. This model shows R^2 values ranging from 0.42 to 0.43 which indicate an estimate of ~42% - ~43% of the variance explained. MARS also provided a variable importance ranking for this model, which can be seen in Table 4.1.

Variable	**nsubsets**	**gcv**	**rss**
prod_mean	33	100.0	100.0
mf.event	32	70.5	70.9
precip	29	44.0	44.9

Table 4.1 continued from previous page

Variable	**nsubsets**	**gcv**	**rss**
flowdist_mean	26	31.7	33.1
precip_sdev	25	30.0	31.4
bio_2	25	30.0	31.4
snowpercent	24	27.6	29.0
prod_sdev	23	33.1>	34.2>
precip_mean	23	25.7	27.2
bio_18	20	21.3	22.9
rr	17	18.0	19.6
imperviousbasin	17	18.0	19.6
bio_15	17	18.0	19.6
flowdist_sdev	14	14.5	16.1
flowdist_skew	13	13.2	14.9
prod_skew	13	13.2	14.9
bio_3	12	12.0	13.7
kfact	7	6.2	8.1

Table 4.1: MARS Variable importance - Lag Time

MARS assesses variable importance based on the reduction of error estimates in the Generalized Cross-Validation (gcv), as well as in the change of Residual Sum of Squares obtained by including each variable in the model.

According to this variable importance ranking, the most important variables to characterize Lag Time seem to be *prod_mean*, *mf.event*, *precip*, *flowdist_mean*, *precip_sdev*, *bio_2* and *snowpercent*. In this case, MARS seems to acknowledge the importance of bioclimatic, morphological variables but overall statistical rainfall moments to characterize lag time. Curiously, the only variables directly related to catchment area are the firtst three statistical moments of flow distance, as well as relief ratio.

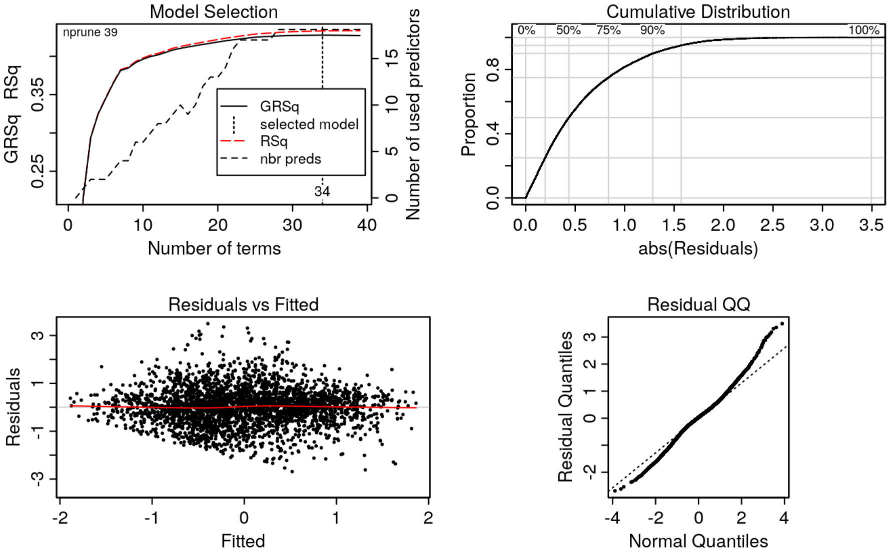

Figure 4.2: MARS: Lag Time training metrics and residual plots

The above plot shows a detailed portrait of the training process which led to the final model. A chart showing the increment in R^2 with respect to the tuned parameters summarizes the model's construction. The residual vs fitted plot shows a slight pattern (indicating some underlying unexplained variance), and the normality plot shows a slight deviation from normal behavior, particularly on the right tail of the distribution (large lag times).

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results are shown in Figure 4.3 and Table 4.2.

Baseline	**Metrics**
CC	0.658
MAE	0.574
MSE	0.569
MPE	0.673
MAPE	2.361
Rsq	0.433

Table 4.2: MARS Baseline Error Metrics - Lag Time

It's noteworthy that the model's performance on previously seen data seems to be consistent with the model's expected explanatory power. This baseline shows a correlation coefficient between the expected and predicted values of 0.658. Error metrics and r^2 values for this fit are consistent with the training metrics. All this likely means that the use of cross-validation during training succesfully avoided overfitting on the training data.

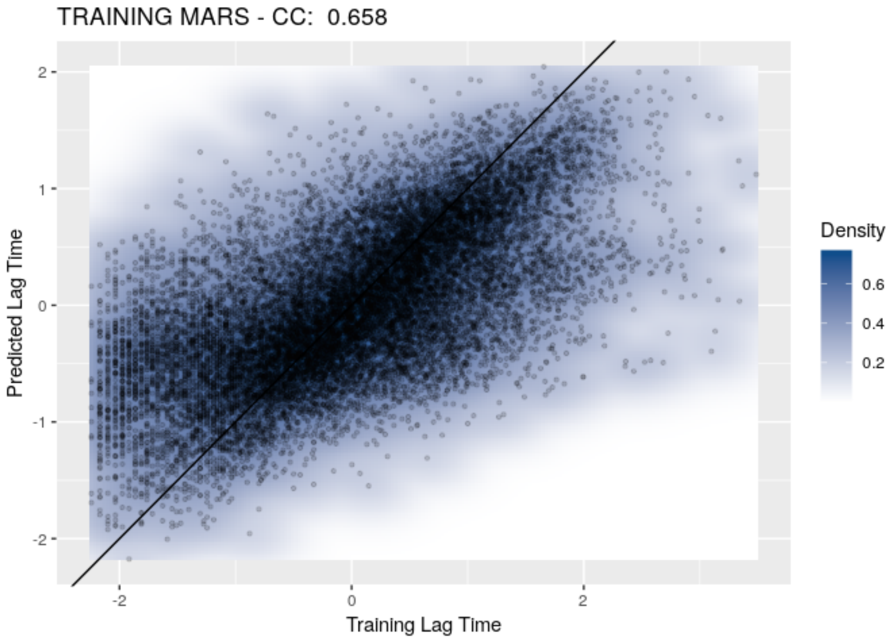

Figure 4.3: MARS: Lag Time fit using training data

Having constructed this baseline, now the trained model was used to predict the response values from the validation dataset, which where not part of the training data. These results are shown in Figure 4.4 and Table 4.3.

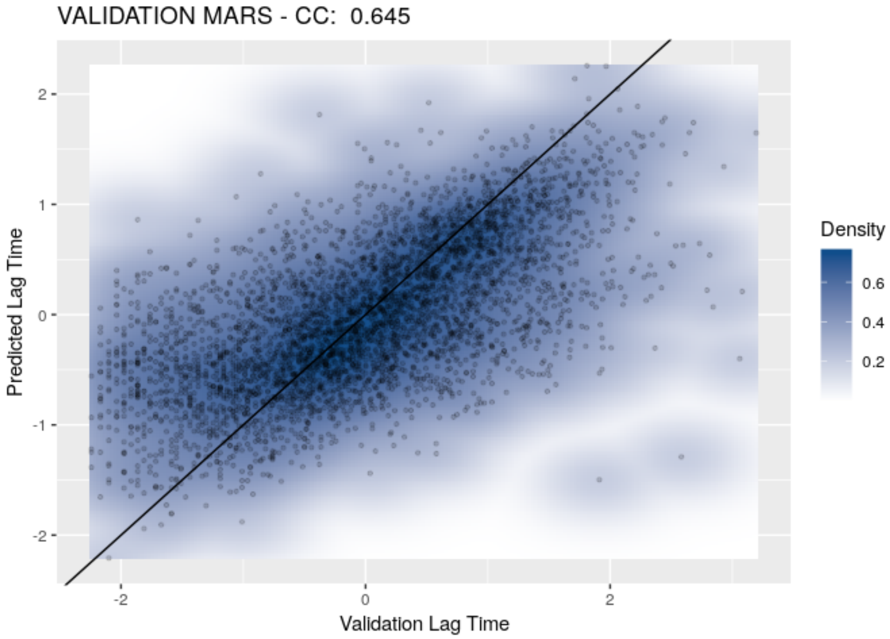

Figure 4.4: MARS: Lag Time fit using validation data

Validation Metrics	
CC	0.645
MAE	0.578
MSE	0.577
MPE	1.009
MAPE	2.188
Rsq	0.416

Table 4.3: MARS Validation Error Metrics - Lag Time

This validation shows a correlation coefficient between the expected and predicted values of 0.645, which remains consistent with the baseline previously established on the training dataset. Error metrics for this fit lie within the expected ranges as well, and so does the R^2 value. These results suggest that the trained model performs with solid consistently when predicting on previously unseen data.

4.1.2 Moment of Relative Peak Discharge Modeling

This Moment of Relative Peak Discharge model took ∼4 hours to train. Parameter tuning was performed from 1 up to 5 degrees of interaction (products of up to 5 predictors), and from 2 up to 54 model terms (two terms over the total amount of predictors). The parameter tuning results during training can be seen Figure 4.5.

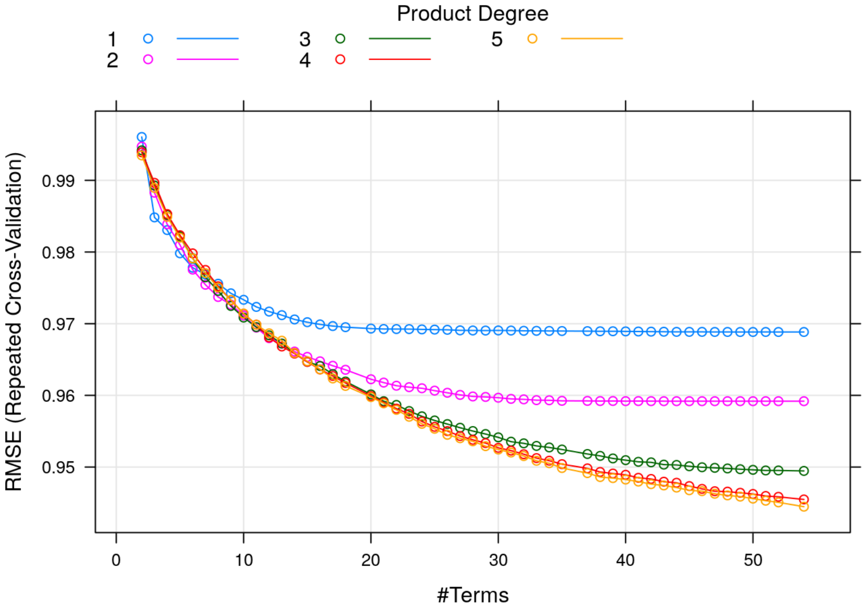

Figure 4.5: MARS: Moment of Relative Peak Discharge Training - Parameter Tuning results

From the parameter tuning, the best fit was found to be a model with up to 54 terms, each with up to 5 degrees of interaction (the maximum allowed for both parameters). The structure and output of the best model found is shown in the **Appendix** on Listing 5.2. The final model was constructed with 54 terms (7 of which where 1st degree interaction terms) and using only 26 of of the 52 possible predictors. This model shows R^2 values ranging from 0.13 to 0.15 which indicate an estimate of ∼13% - ∼15% of the variance explained. Table 4.4 presents the results for this model's variable importance analysis.

variable	**nsubsets**	**gcv**	**rss**
bio_10	53	100.0	100.0
ruggedness	51	92.4	92.9
slopeoutlet	49	85.0	86.0
imperviouscell	48	82.7	83.8
precip_sdev	48	82.7	83.8
bio_1	48	82.7	83.8
bio_7	48	82.7	83.8
bio_8	48	82.7	83.8
si	46	78.2	79.6
snowpercent	44	75.0	76.5
rr	44	73.7	75.3
kfact	42	69.1	71.0
bio_2	40	63.7	66.0
cncell	39	61.1	63.6
bio_3	39	61.1	63.6
lbm	38	58.7	61.3
bio_15	38	58.7	61.3
G1	34	53.3	56.0
G2	34	53.3	56.0
mf.event	30	47.6	50.5
bio_18	29	46.2	49.1
rockdepth	27	43.6	46.5
cnbasin	24	41.8	44.4
bio_12	12	27.8	29.9
precip_mean	11	26.3	28.4
coemcell	10	24.7	26.7

Table 4.4: MARS Variable Importance - peakq_moment

According to MARS, the most important variables to characterize the Moment of Relative Peak Discharge seem to be *bio_10*, *ruggedness*, *slopeoutlet*, *imperviouscell*, *precip_sdev*, *bio_1* and *bio_7*. This points to a clear influence of bioclimatic and morphological variables. Additionally, statistical rainfall moments as well as the catchment-scale moments of flow distance seem to be relevant as well. These make sense, due to this target variable's dependency on the basin's flow response, and the influence of these variables on it.

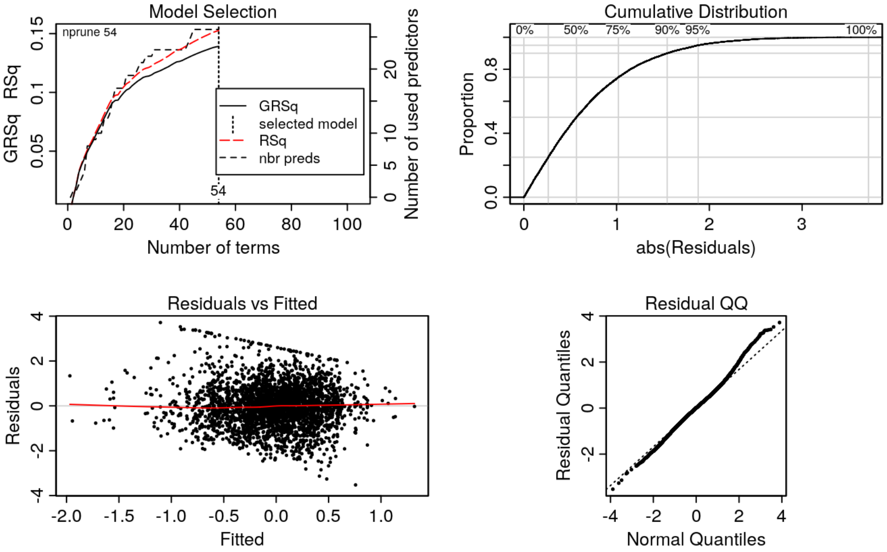

Figure 4.6: MARS: Moment of Relative Peak Discharge training metrics and residual plots

Figure 4.6 shows a the same MARS training statistics presented for lag time. Even though the normality plot seems to be behaving better than in the case of lag time, the distribution of residuals vs fitted show clear signs of unexplained variance, as well as apparent artifacts. This is expected due to the low skill presented by the model, as well as the very low correlations between predictors and the target variable.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results can be seen in Figure 4.7 and Table 4.5.

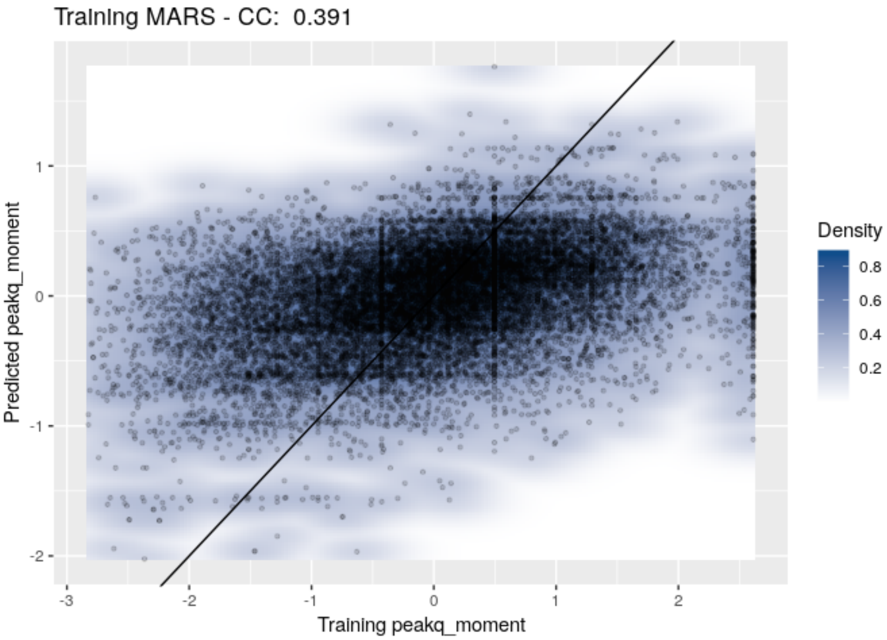

Figure 4.7: MARS: Moment of Relative Peak Discharge fit using training data

Baseline	**Metrics**
CC	0.391
MAE	0.713
MSE	0.846
MPE	0.436
MAPE	2.571
Rsq	0.153

Table 4.5: MARS Baseline Error Metrics - peakq_moment

This baseline shows a correlation coefficient between the expected and predicted values of 0.391 and error metrics for this fit lie in the neighborhood of what is expected from training (0.85). The R^2 value also points towards a consistent explanatory power according to training results. Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. These results are shown in Figure 4.8 and Table 4.6.

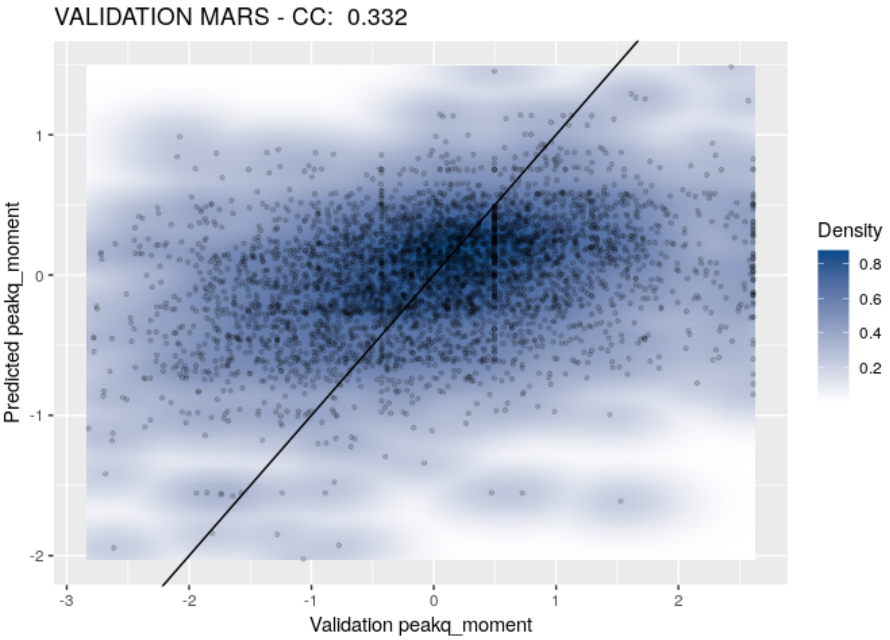

Figure 4.8: MARS: Moment of Relative Peak Discharge fit using validation data

Validation Metrics	
CC	0.332
MAE	0.731
MSE	0.898
MPE	0.932
MAPE	2.224
Rsq	0.11

Table 4.6: MARS Validation Error Metrics - peakq_moment

This validation shows a correlation coefficient between the expected and predicted values of 0.332, which remains consistent with the baseline previously established on the training dataset. Error metrics for this fit lie between 0.731 and 0.898. Remarkably, these results suggest that the trained model for the moment of relative peak discharge performs consistently when predicting on previously unseen data.

4.1.3 Flood Stage Threshold Exceedance Modeling

This Flood Stage Threshold Exceedance model took ~10 hours to train. Parameter tuning was performed from 1 up to 5 degrees of interaction (products of up to 5 predictors), and from 2 up to 52 model terms (up to one term for each predictor). Note that the number of predictors is two less than the other models, given that for this case variables *tp* and *rt* were not used. Note that in this instance MARS will be used to perform classification instead of regression. Figure 4.9 shows training accuracy curves for the parameter tuning process.

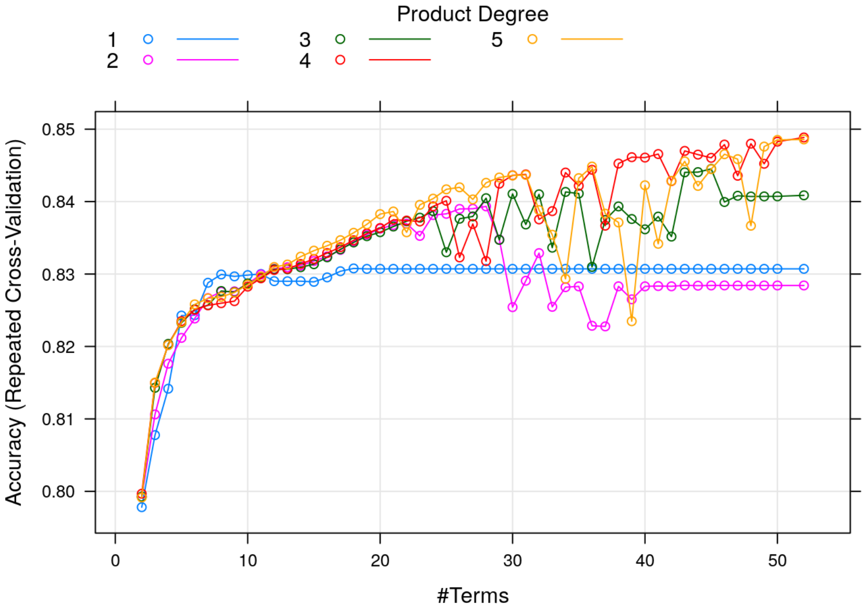

Figure 4.9: MARS: Flood Stage Threshold Exceedance Training - Parameter Tuning results

From the parameter tuning, the best fit was found to be a model with up to 52 terms (the maximum allowed), each with up to 4 degrees of interaction. Also note that given that this is a classification model, the training metric used was accuracy. The structure of the best model found is shown in the **Appendix** on Listing 5.3. As can be seen from these model results, MARS is able to perform classification by generating a model for each of the response classes. Some generalized training metrics were extracted from this model, as well as per-class metrics. These can be seen on Table 4.7 and 4.8.

Class	Error Metric	Value
ALL	RSS	3471.72
ALL	Rsq	0.401
ALL	GRsq	0.392

Table 4.7: MARS: Flood Threshold Exceedance - Generalized Error Metrics

Class	Label	RSS	Rsq	GRsq
No Exceedance	0	59.65	0.259	0.248
Exceeds Action	1	411.39	0.433	0.424
Exceeds Minor	2	509.87	0.158	0.145
Exceeds Moderate	4	1300.42	0.229	0.217
Exceeds Major	8	1190.37	0.558	0.552

Table 4.8: MARS: Flood Threshold Exceedance - Per-class Error Metrics

The final model was constructed with 52 terms (the maximum number possible), 5 of which where 1st degree interaction terms and only 19 out of the 52 possible predictors were used. This model shows R^2 values ranging from 0.39 to 0.40 which indicate an estimate of ∼39% - ∼40% of the variance explained. Table 4.9 present the results for this model's variable importance.

variable	nsubsets	gcv	rss
est_area	51	100.0	100.0
mf.event	50	82.4	82.8
prod_mean	48	65.1	66.0
slopeoutlet	41	48.8	50.1
totalBasinPixels	41	48.8	50.1
bio_10	39	45.1	46.4
G1	38	43.5	44.8
imperviousbasin	37	41.7	43.1
imperviouscell	36	40.1	41.6
G2	36	40.1	41.6
precip	35	38.7	40.1
cnbasin	34	37.6	39.0
rl	32	35.0	36.6
rockdepth	28	30.8	32.4
snowpercent	28	30.8	32.4
bio_3	25	28.1	29.6

Table 4.9 continued from previous page

variable	nsubsets	gcv	rss
bio_17	25	28.1	29.6
ruggedness	17	20.5	22.0
si	16	20.0	21.4

Table 4.9: MARS Variable Importance - exceeds_threshold

According to MARS' variable importance ranking, the most important variables to characterize the Exceedance of Flood Stage Thresholds seem to be *est_area*, *mf.event*, *prod_mean*, *slopeoutlet*, *totalBasinPixels*, *bio_10*, *G1* and *imperviousbasin*. Note that both *est_area* and *totalBasinPixels* appear to be very relevant, which is expected as they are evidently highly correlated (one is a direct function of the other), and once could anticipate them both to appear together when the catchment's area is relevant. However, the fact that their contribution to minimizing gcv errors differs by over 50% also show how other morphological and bioclimatic factors, as well as precipitation moments and moments of flow distance play a role in characterizing this response.

Given that this MARS training generated 5 different models (one per response class), there are five sets of training metrics and residual plots. These are shown in Figures 4.10 through 4.14.

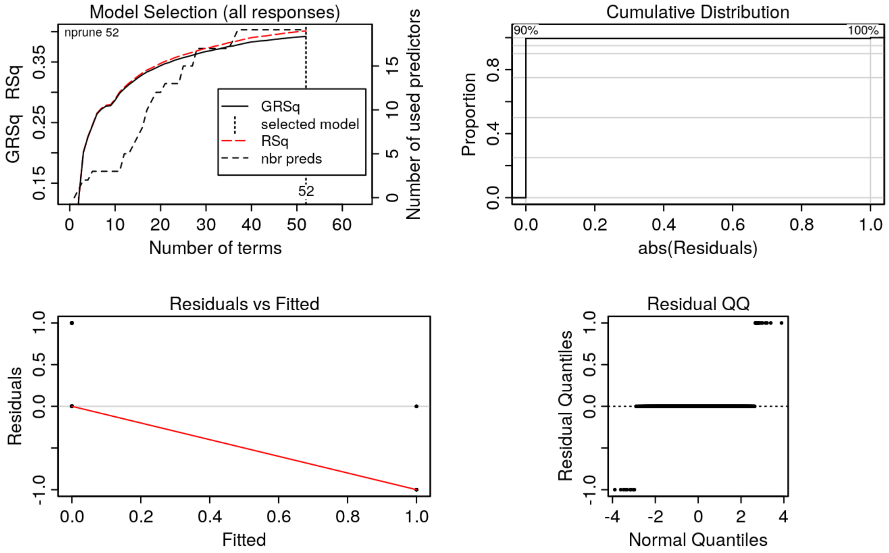

Figure 4.10: MARS: Flood Stage Threshold Exceedance training metrics and residual plots for No-Exceedance

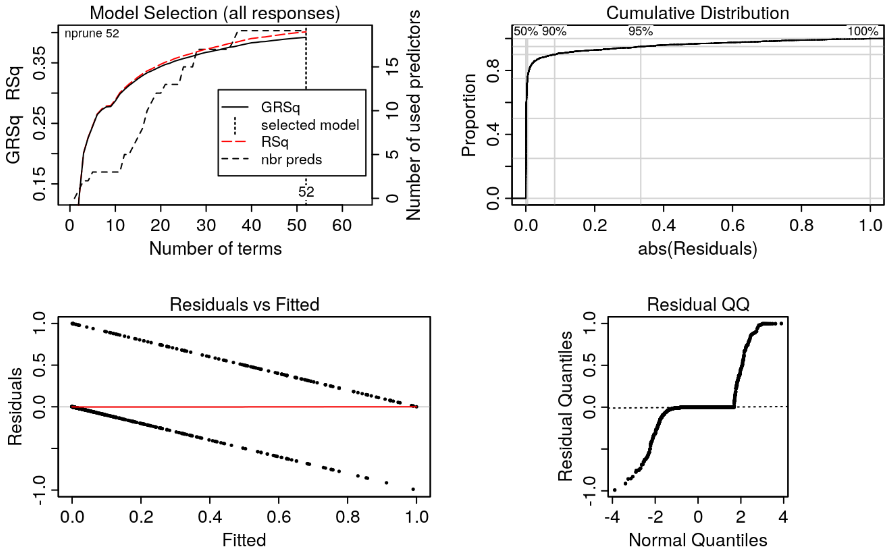

Figure 4.11: MARS: Flood Stage Threshold Exceedance training metrics and residual plots for Exceeds Action

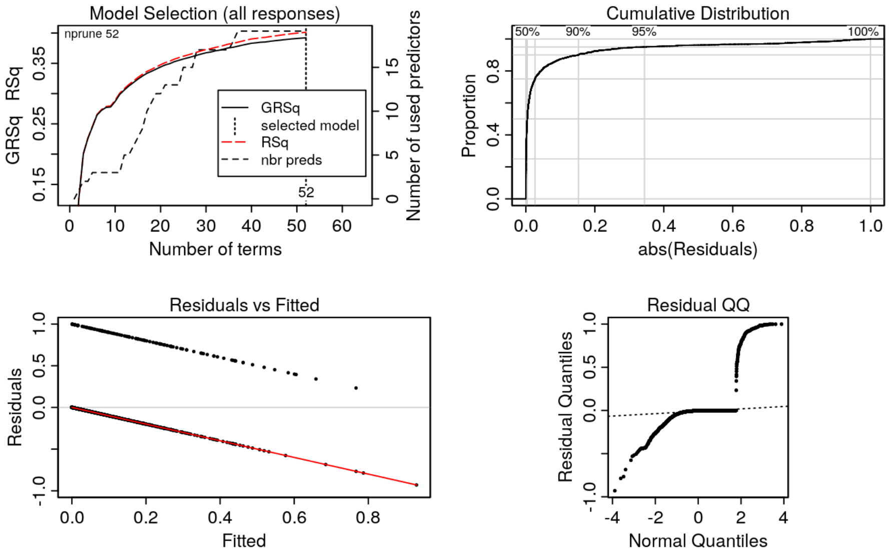

Figure 4.12: MARS: Flood Stage Threshold Exceedance training metrics and residual plots for Exceeds Minor

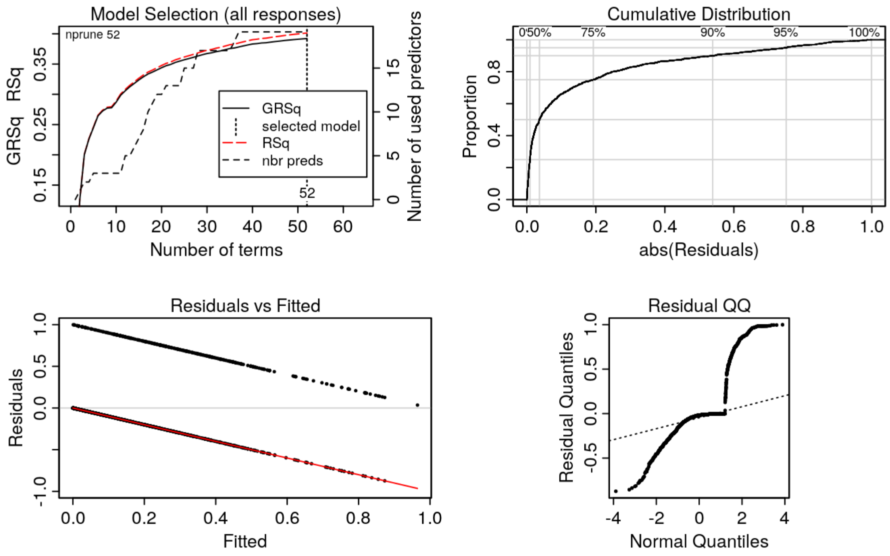

Figure 4.13: MARS: Flood Stage Threshold Exceedance training metrics and residual plots for Exceeds Moderate

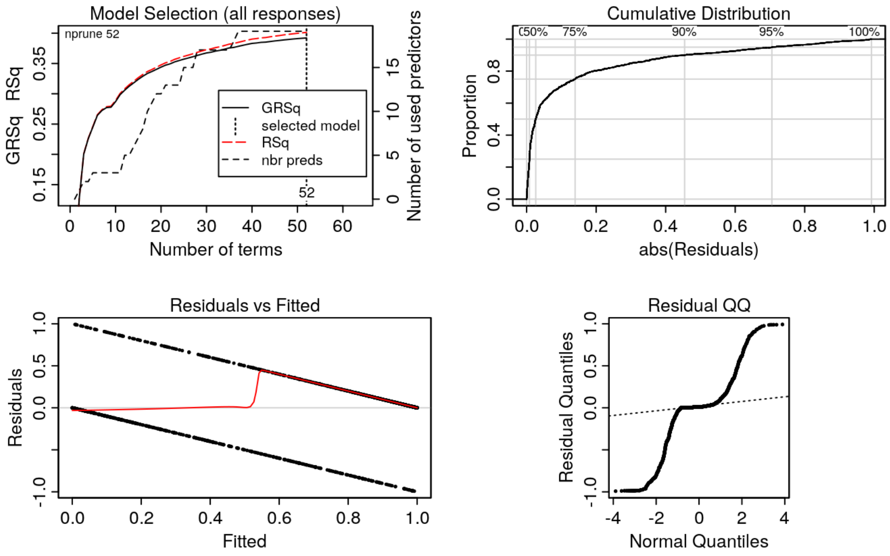

Figure 4.14: MARS: Flood Stage Threshold Exceedance training metrics and residual plots for Exceeds Major

Overall, the same chart for R^2 is presented in all cases. As classes move from No-Exceedance to major threshold exceedance, the cumulative distribution of absolute residuals tends to exhibit a softer *attack*, which reflects the lower amount of cases for all classes with respect to *Exceeds Major*. Given that each of these represent the model partially, the normality plots as well as the residual vs fitted plots exhibit various fragmentations and step-like behaviors, corresponding to the binary nature of whether a sample is classified with each label or not. In other words, they reflect each class' bimodal nature, as well as evidence of underlying unexplained variance.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline classification results and metrics are shown in Tables 4.10, 4.11 and 4.12.

	Reference				
prediction	**0**	**1**	**2**	**4**	**8**
0	28	19	1	5	3
1	44	529	121	124	71
2	3	24	66	51	14
4	3	127	275	702	308
8	3	61	166	1019	13147

Table 4.10: MARS Baseline Confusion Matrix - exceeds_threshold

MARS - exceeds_threshold	
Accuracy	0.8556
95% CI	(0.8502, 0.8609)
No. of information Rate	0.8007
P-value [Acc >NIR]	<2.2e-16
Kappa	0.5288
Mcnemar's Test P-Value	<2.2e-16

Table 4.11: MARS Baseline Overall Statistics - exceeds_threshold

This baseline shows that accuracy metrics for this fit lie between 0.85 and 0.86, and the Kappa statistic establishes a baseline value of 0.52. The kappa statistic is a measure of how closely the instances classified by the machine learning classifier matched the data labeled as ground truth. Per-class statistics reflect once more the effect of training on unbalanced classes, where accuracy for classifying Exceeds Action (Class 1) and Exceeds Major (Class 8) are much higher than the other classes; particularly No-Exceedance.

Statistic	**Class: 0**	**Class: 1**	**Class: 2**	**Class: 4**	**Class: 8**
Sensitivity	0.345679	0.69605	0.104928	0.36928	0.9708
Specificity	0.998337	0.97771	0.994351	0.95251	0.6295
Pos Pred Value	0.500000	0.59505	0.417722	0.49611	0.9132

Table 4.12 continued from previous page

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Neg Pred Value	0.996856	0.98559	0.966400	0.92264	0.8427
Prevalence	0.004789	0.04493	0.037188	0.11239	0.8007
Detection Rate	0.001655	0.03128	0.003902	0.04150	0.7773
Detection Prevalence	0.003311	0.05256	0.009341	0.08366	0.8511
Balanced Accuracy	0.672008	0.83688	0.549640	0.66089	0.8001

Table 4.12: MARS Baseline Class Statistics - exceeds_threshold

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation classification metrics and results are presented in Tables 4.13, 4.14 and 4.15.

	Reference				
prediction	**0**	**1**	**2**	**4**	**8**
0	5	5	0	3	0
1	13	123	45	35	16
2	1	5	21	16	8
4	2	30	86	164	86
8	0	14	55	238	3258

Table 4.13: MARS Validation Confusion Matrix - exceeds_threshold

MARS - exceeds_threshold	
Accuracy	0.8444
95% CI	(0.8331, 0.8552)
No. of information Rate	0.7964
P-value [Acc >NIR]	7.076e-16
Kappa	0.5082
Mcnemar's Test P-Value	NA

Table 4.14: MARS Validation Validation Statistics - exceeds_threshold

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Sensitivity	0.238095	0.69492	0.101449	0.35965	0.9673
Specificity	0.998099	0.97310	0.992541	0.94593	0.6434

Table 4.15 continued from previous page

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Pos Pred Value	0.384615	0.53017	0.411765	0.44565	0.9139
Neg Pred Value	0.996205	0.98649	0.955481	0.92437	0.8343
Prevalence	0.004966	0.04185	0.048948	0.10783	0.7964
Detection Rate	0.001182	0.02908	0.004966	0.03878	0.7704
Detection Prevalence	0.003074	0.05486	0.012060	0.08702	0.8430
Balanced Accuracy	0.618097	0.83401	0.546995	0.65279	0.8054

Table 4.15: MARS Validation Class Statistics - exceeds_threshold

This validation shows Accuracy metrics for this fit lie between 0.83 and 0.85 for unseen data, which remains consistent with the baseline previously established on the training dataset. The Kappa statistic is also at 0.508, which resembles closely the baseline results. These results suggest that the trained model performs consistently when predicting on previously unseen data, still favoring Exceeds Action (Class 1) and Exceeds Major (Class 8).

4.2 Random Forest

In second instance, Lag time, the Moment of Relative Peak Discharge and the Flood Stage Threshold Exceedance were also modeled by fitting bagged, parameter-tuned Random Forest models, which explored the number of terms to retain at each split using a tuning grid, and a bag of 100 trees. This way, optimal parameter settings were found for a model which would minimize error measures, or maximize performance measures. Like MARS, these models were trained using 10 times 10-fold cross-validation in order to mitigate overfitting on the training dataset, and once trained these were also tested to predict known outputs on a validation (holdout, not included in training) dataset.

4.2.1 Lag Time Modeling

This Lag Time model took ∼16 hours to train. Parameter tuning was performed from 1 up to 54 variables to retain per split in each tree (all variables could be considered to perform a split at a given node), and 100 trees were used. Parameter tuning results for this model are shown in Figure 4.15, and model outputs are shown in Table 4.16.

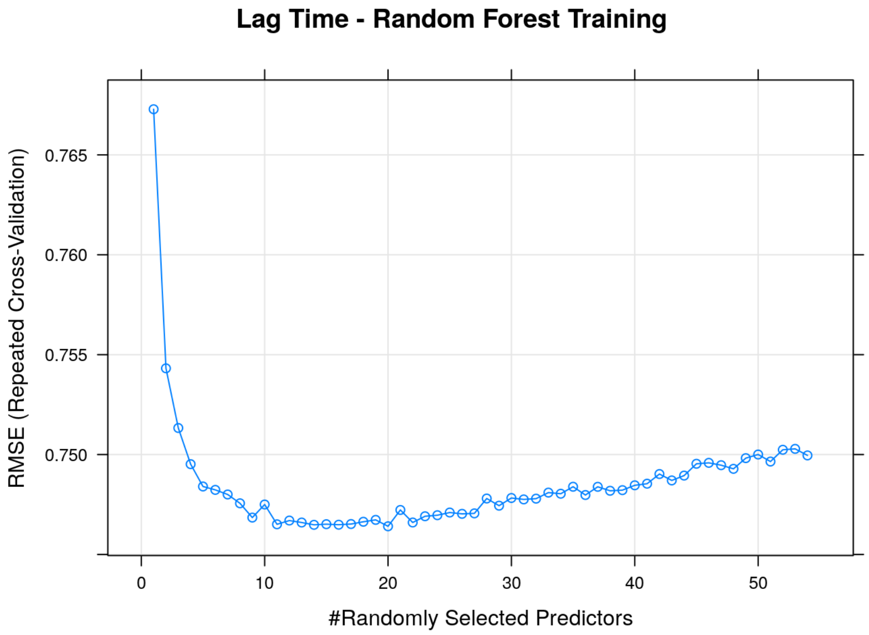

Figure 4.15: Random Forest: Lag Time Training - Parameter Tuning results

Random Forest - Lag time

Random Forest Type	Regression
No. of Trees	100
No. of of variables tried at each split	20
Mean Squared Residuals	0.565
% Var. Explained	43.66

Table 4.16: Random Forest Best Fit - Lag Time

The final model produced by the tuning process, was achieved by using 20 variables at each split and 100 trees. The mean RSS for the bagged tree model was around 0.56, and the final model explains around 43% of the variance in the training data. Figure 4.16 shows the results for variable importance calculated for this model.

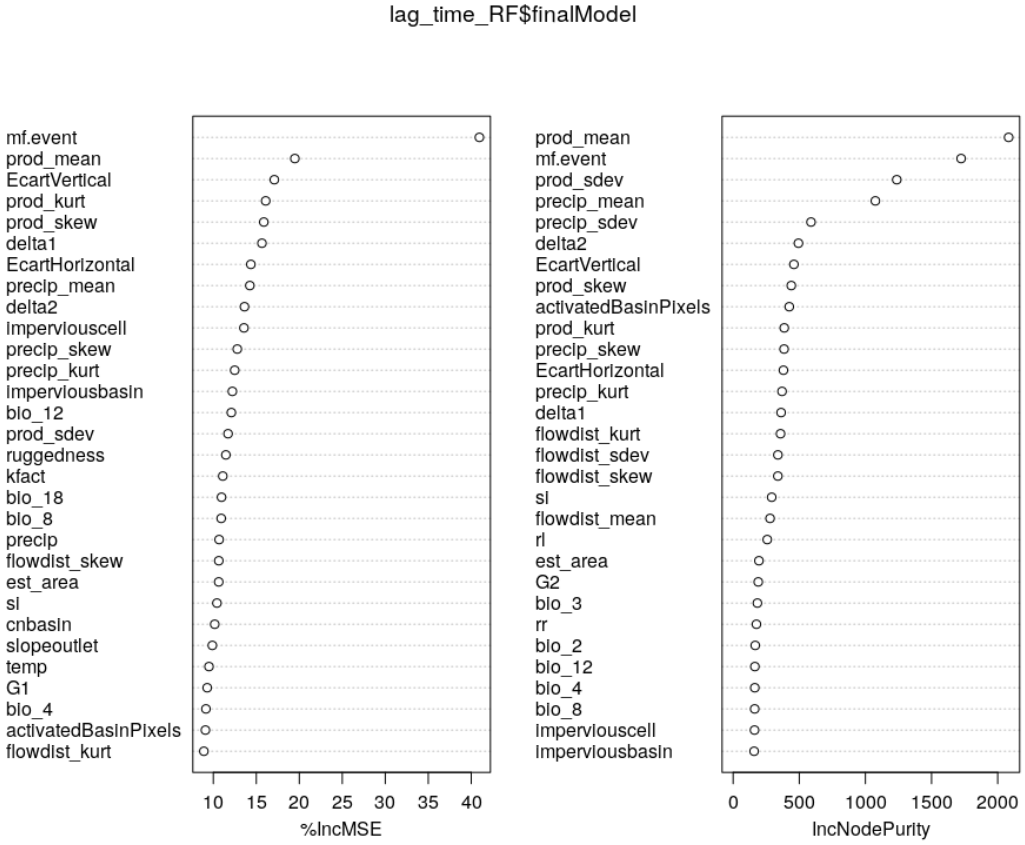

Figure 4.16: Random Forest: Lag Time Training - Variable importance results

According to this Random Forest model, *mf.event*, *prod_mean*, *EcartVertical*, *prod_sdev*, *precip_mean*, *precip_sdev* and *delta2* are some of the most significant factors for characterizing Lag Time. This variable importance assessment is done with respect to each variable's contribution to reducing the MSE during training (%IncMSE), and with respect to how much the presence of each variable at any given split reduces node impurity (IncNodePurity, pure nodes make splits according to values of a single predictor). In a similar and consistent fashion with MARS' results, Random Forest highlights the importance of statistical precipitation moments, as well as morphological and bioclimatic variables. Both models seem to agree on the importance of moments of flow distance, however a different one is selected between MARS and Random Forest.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results are shown in Figure 4.17 and Table 4.17.

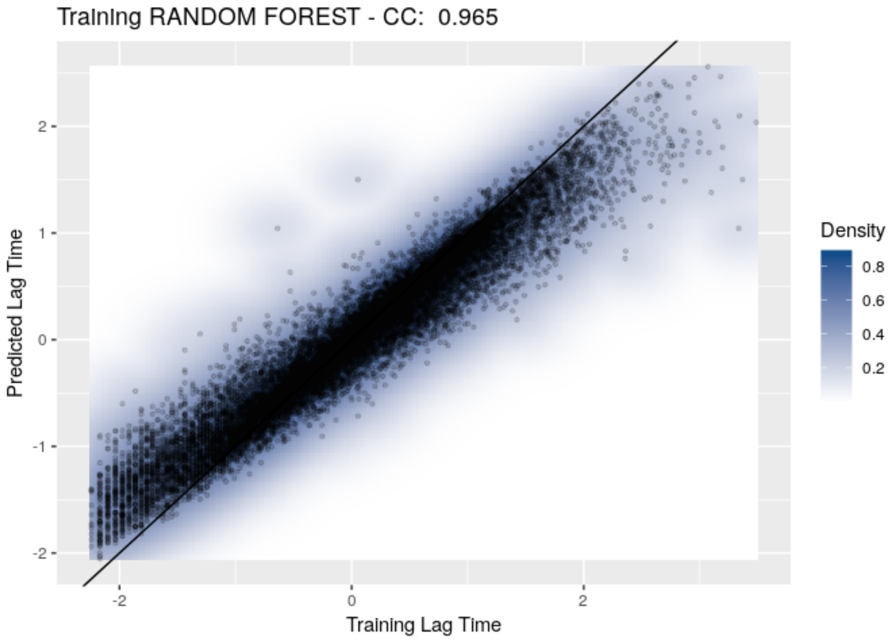

Figure 4.17: Random Forest: Lag Time fit using training data

Baseline	**Metrics**
CC	0.965
MAE	0.225
MSE	0.095
MPE	0.214
MAPE	0.957
Rsq	0.931

Table 4.17: Random Forest Baseline Error Metrics - Lag Time

This baseline shows a correlation coefficient between the expected and predicted values of 0.964, and error metrics for this fit lie between 0.09 and 0.22. Given the above plot this model exhibits a high correlation between the fitted model and the original response variable which could be an indication of overfitting. However, given the implementation of a bagged tree approach and 10x10-fold cross-validation, the performance of this model on unseen data should still be able to explain around 43% of the variance of the new data (according to training metrics).

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation results and error metrics are shown in Figure 4.18 and Table 4.18.

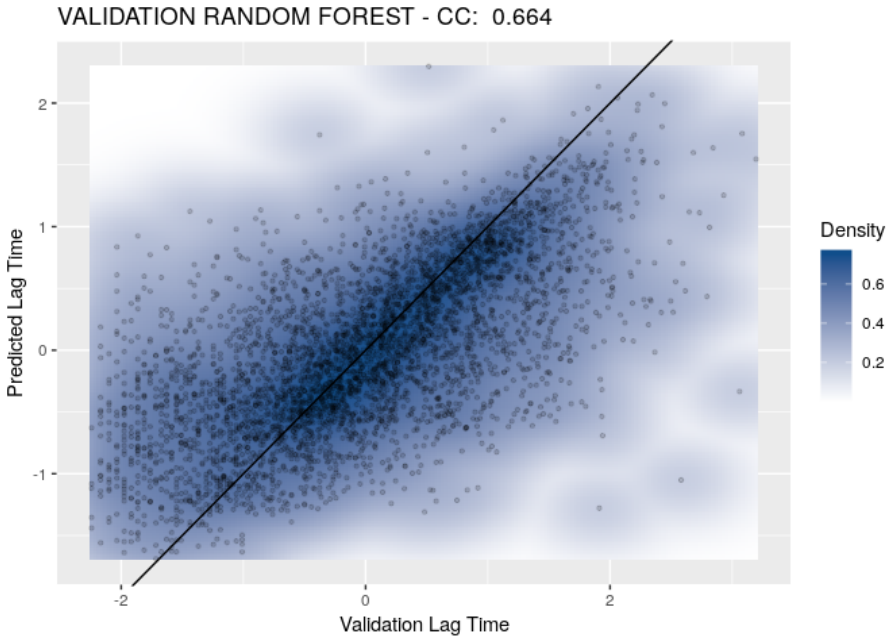

Figure 4.18: Random Forest: Lag Time fit using training data

Validation Metrics	
CC	0.664
MAE	0.553
MSE	0.552
MPE	1.033
MAPE	2.187
Rsq	0.441

Table 4.18: Random Forest Validation Error Metrics - Lag Time

This validation shows a correlation coefficient between the expected and predicted values of 0.664, which is considerably lower than the baseline previously established on the training dataset. However, the error metrics for this fit lie around 0.55, which is consistent with the explanatory power of the constructed model according to training metrics. These results suggest that, even though the Random Forest model tends to overfit when presented with it's own training data, the trained model performs as expected when predicting on previously unseen data.

4.2.2 Moment of Relative Peak Discharge Modeling

This Moment of Relative Peak Discharge model took ∼16 hours to train. Parameter tuning was performed from 1 up to 54 variables to retain per split in each tree, and 100 trees were used. Parameter tuning results for this model are shown in Figure 4.19 and model outputs in Table 4.19.

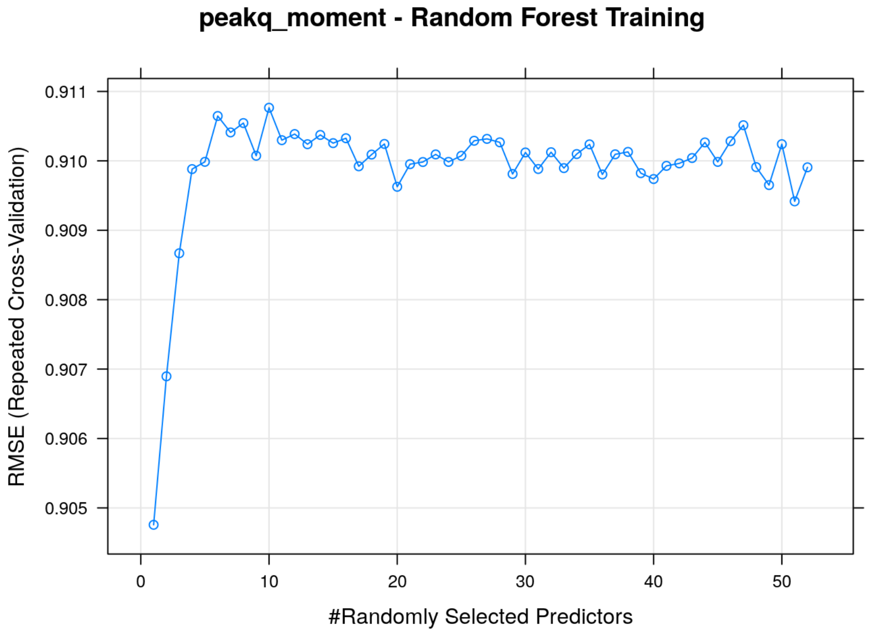

Figure 4.19: Random Forest: Moment of Relative Peak Discharge Training - Parameter Tuning results

Random Forest - peakq_moment	
Random Forest Type	Regression
No. of Trees	100
No. of of variables tried at each split	1
Mean Squared Residuals	0.820
% Var. Explained	17.85

Table 4.19: Random Forest Best Fit - peakq_moment

The final model produced by the tuning process, was achieved by using 1 variable at each split and 100 trees (note how error quickly rises the more predictors are selected). The mean RSS for the bagged tree model is around 0.82, and the final model explains only around 18% of the variance in the training data. Figure 4.20 shows the variable importance results calculated for this model.

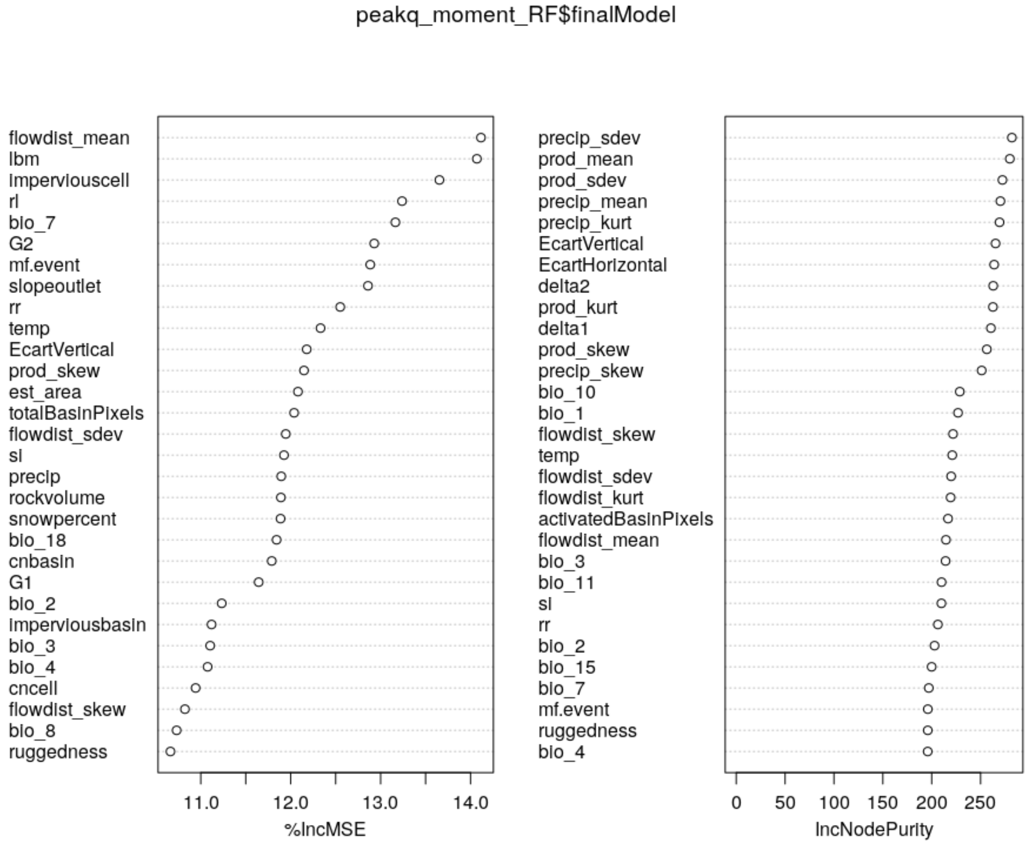

Figure 4.20: Random Forest: Moment of Relative Peak Discharge Training - Variable importance results

Regarding variable importance, this Random Forest model shows *flowdist_mean*, *lbm*, *imperviouscell*, *rl*, *bio_7*, *G2* and *precip_sdev* to be some of the most influential factors for characterizing the Moment of Relative Peak Discharge. Notice that the overall contribution for each variable on the importance metrics is rather small, which is a reflection of the low correlation of the predictors on this target variable. These agree partially with MARS' assessment, and even though different morphological variables are highlighted by Random Forest, these still hold a close relationship with the basin's flow response.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results and error metrics are presented in Figure 4.21 and Table 4.20.

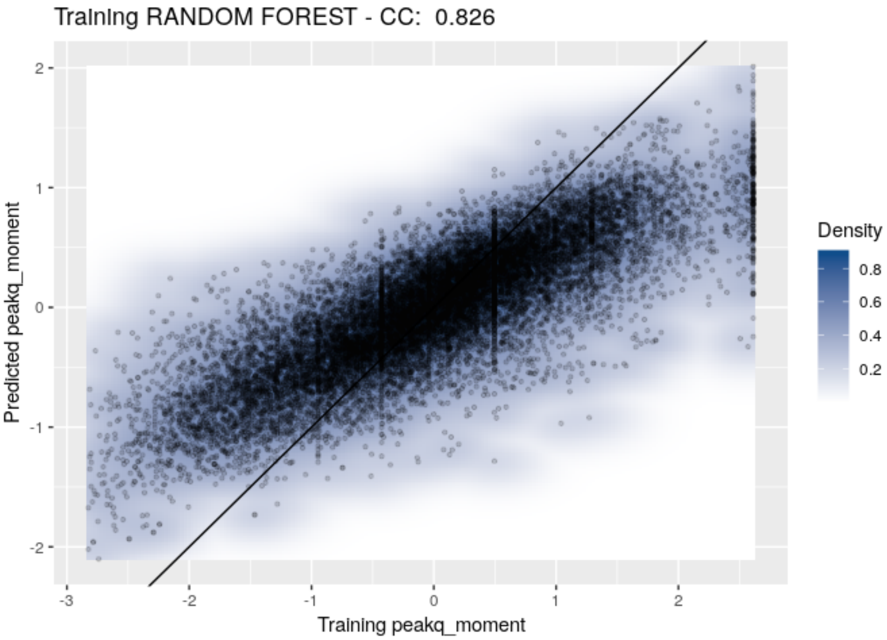

Figure 4.21: Random Forest: Moment of Relative Peak Discharge fit using training data

Baseline	**Metrics**
CC	0.862
MAE	0.482
MSE	0.4
MPE	0.25
MAPE	1.863
Rsq	0.683

Table 4.20: Random Forest Baseline Error Metrics - peakq_moment

This baseline shows a correlation coefficient between the expected and predicted values of 0.826, and error metrics for this fit lie between 0.4 and 0.5. Given the above plot this model exhibits a moderately high correlation between the fitted model and the original response variable which could be an indication of overfitting. However, given the implementation of a bagged tree approach and 10x10-fold cross-validation, the performance of this model on unseen data should still be able to explain at least 17% of the variance of the new data (according to training metrics). Even though this model's predictive power doesn't seem to be high, it is of interest due to this research's exploratory nature.

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation results are shown in Figure 4.22 and Table 4.21.

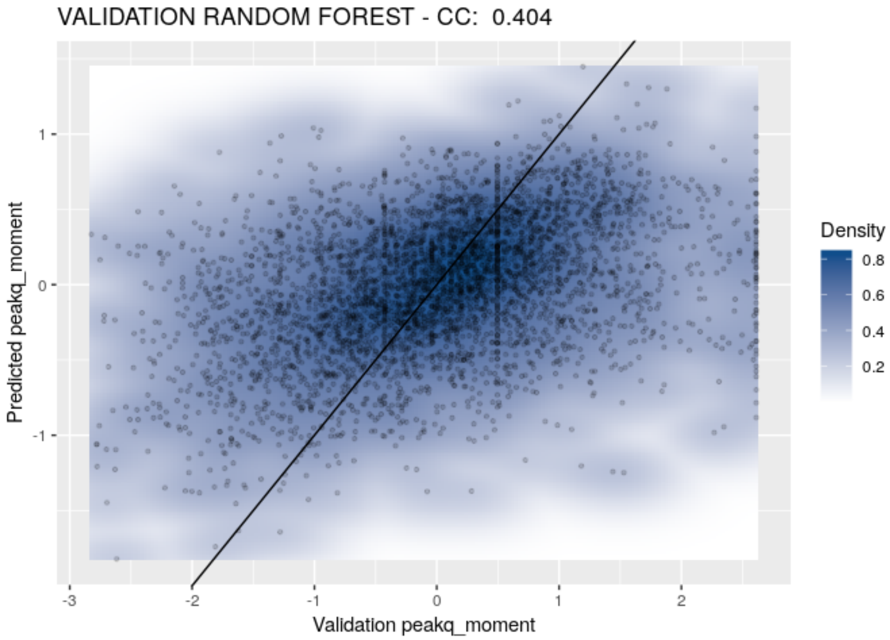

Figure 4.22: Random Forest: Moment of Relative Peak Discharge fit using validation data

Validation Metrics	
CC	0.404
MAE	0.697
MSE	0.843
MPE	1.001
MAPE	2.309
Rsq	0.163

Table 4.21: Random Forest Validation Error Metrics - peakq_moment

This validation shows a correlation coefficient between the expected and predicted values of 0.404, which is considerably lower than the baseline previously established on the training dataset. The error metrics for this fit lie between 0.69 and 0.85. The R^2 value for this fit is consistent with the explanatory power of the constructed model according to training metrics. These results suggest that the trained model performs consistently when predicting on previously unseen data, however it should be noted that predictive power is low. Regardless, valuable information was be collected from this model, which can help better understand which variables hold relevance for modeling the Moment of Relative Peak Discharge.

4.2.3 Flood Stage Threshold Exceedance Modeling

This Exceedance of Flood Stage Thresholds model took ∼3 hours to train. Parameter tuning was performed from 1 up to 52 variables to retain per split in each tree (*tp* and *rt* were excluded), and 100 trees were used. Note that this Random Forest will be used to build a classification model. Figure 4.23 shows training accuracy for the parameter tunning, and Tables 4.22 and 4.23 show model training results.

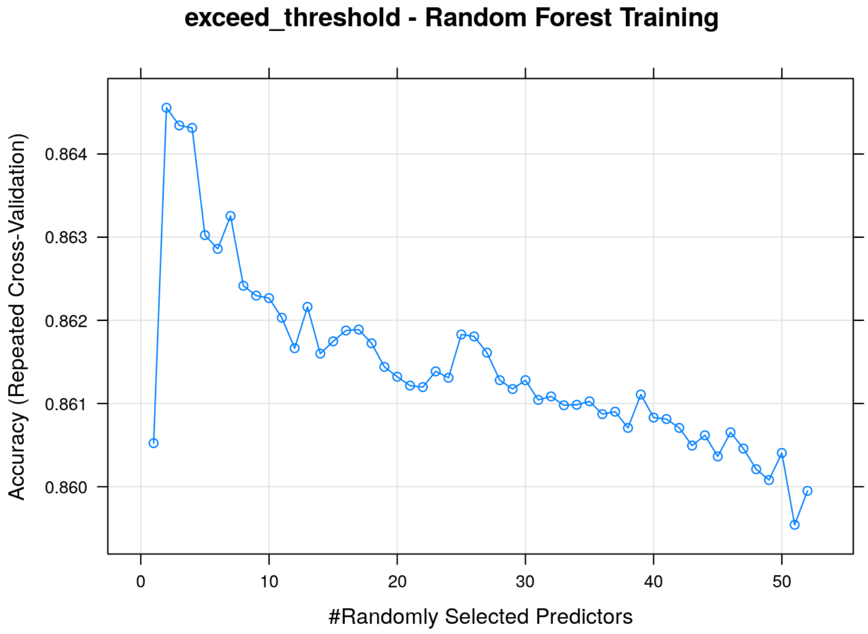

Figure 4.23: Random Forest: Flood Stage Threshold Exceedance Training - Parameter Tuning results

Random Forest - exceeds_threshold	
Random Forest Type	Classification
Number of Trees	100
No. of variables tried at each split	2
OOB estimate of error rate	13.59%

Table 4.22: Random Forest Best Fit - exceeds_threshold

Label	**0**	**1**	**2**	**4**	**8**	**class error**
0	41	37	0	2	1	0.49382716
1	23	581	61	54	41	0.23552632
2	2	151	173	226	77	0.72496025
4	2	147	115	855	782	0.55023672

Table 4.23 continued from previous page

Label	**0**	**1**	**2**	**4**	**8**	**class error**
8	1	82	52	443	12965	0.04267887

Table 4.23: Random Forest Best Fit: Confusion Matrix - exceeds_threshold

The final classification model produced by the tuning process, was achieved by using 2 variables at each split (note the stark dip in accuracy at around 3) and 100 trees. The out of bag estimated error for this model is around 13%, and class errors range widely from 72% to 4%. These error discrepancies are a reflection of the imbalance of the training classes (more training samples for a given class than another). Variable importance analysis was also calculated for this model, and its results are shown in Figure 4.24.

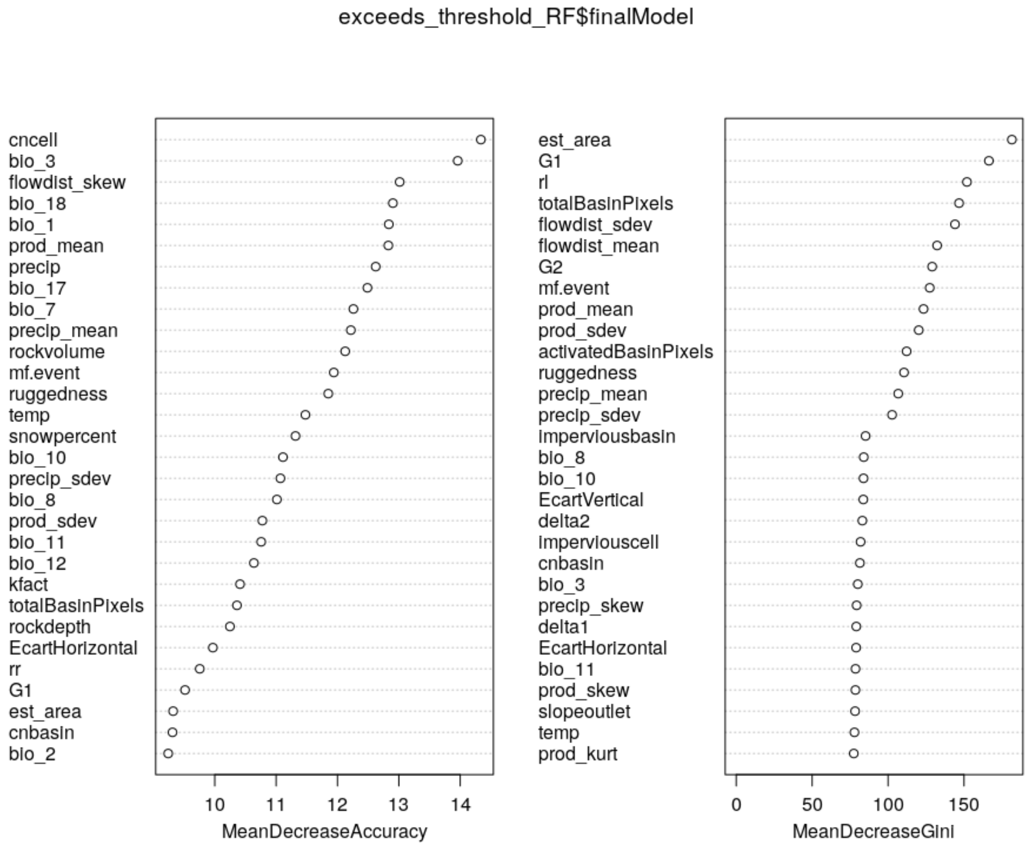

Figure 4.24: Random Forest: Flood Stage Threshold Exceedance Training - Variable importance results

Regarding variable importance, this Random Forest model shows *cncell*, *bio_3*, *flowdist_skew*, *bio_18*, *est_area*, *G1*, *rl*, *totalBasinPixels*, *flowdist_sdev* and *flowdist_mean* to be some of the most influential factors for characterizing flood stage threshold exceedance. Random Forest's variable importance for classification are slightly different than for regression, in that each variable is asses by how much mean decrease in accuracy they reduce by being included,instead of each variable's contribution to reducing MSE. Similarly each variables contribution to the Mean Decrease of Gini coefficient (a measure of inequality among values in each class), instead of their contribution to node impurity. Once again, *est_area* and *totalBasinPixels* appear close to one another, as expected due to their similitude. However, the later only appears in the one of the two importance metrics. Conversely, even though this model and MARS agree on the relevance of moments of flow distance, bioclimatic and morphological variables, different sets seem to be highlighted by each method.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Basline results and classification metrics are presented in Tables 4.24, 4.25 and 4.26.

	Reference				
Prediction	**0**	**1**	**2**	**4**	**8**
0	59	10	1	3	1
1	22	692	111	96	55
2	0	30	389	26	27
4	0	20	102	1487	141
8	0	8	26	289	13319

Table 4.24: Random Forest Baseline Confusion Matrix - exceeds_threshold

RF - exceeds_threshold	
Accuracy	0.9428
95% CI	(0.9392, 0.9462)
No. of information Rate	0.8007
P-value [Acc >NIR]	<2.2e-16
Kappa	0.8311
Mcnemar's Test P-Value	<2.2e-16

Table 4.25: Random Forest Baseline Validation Statistics - exceeds_threshold

Statistic	**Class: 0**	**Class: 1**	**Class: 2**	**Class: 4**	**Class: 8**
Sensitivity	0.728395	0.91053	0.61844	0.78222	0.9835

Table 4.26 continued from previous page

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Specificity	0.999109	0.98242	0.99490	0.98248	0.9042
Pos Pred Value	0.797297	0.70902	0.82415	0.84971	0.9763
Neg Pred Value	0.998694	0.99573	0.98540	0.97270	0.9315
Prevalence	0.004789	0.04493	0.03719	0.11239	0.8007
Detection Rate	0.003488	0.04091	0.02300	0.08792	0.7875
Detection Prevalence	0.004375	0.05770	0.02791	0.10346	0.8066
Balanced Accuracy	0.863752	0.94647	0.80667	0.88235	0.9438

Table 4.26: Random Forest Baseline Class Statistics - exceeds_threshold

This baseline shows a accuracy between 93% and 94%, with a kappa statistic of 0.83. Given the above results, this model exhibits a very high correlation between the fitted model and the original response variable which could be an indication of overfitting. However, given the implementation of a bagged tree approach and 10x10-fold cross-validation, the performance of this model on unseen data should still be consistent.

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation results are shown in Tables 4.27, 4.28 and 4.29.

	Reference				
Prediction	**0**	**1**	**2**	**4**	**8**
0	9	7	0	0	0
1	12	132	63	33	28
2	0	16	56	30	13
4	0	16	65	210	110
8	0	6	23	183	3217

Table 4.27: Random Forest Validation Confusion Matrix - exceeds_threshold

RF - exceeds_threshold	
Accuracy	0.8569
95% CI	(0.846, 0.8674)
No. of information Rate	0.7964
P-value [Acc >NIR]	<2.2e-16
Kappa	0.5793
Mcnemar's Test P-Value	NA

Table 4.28 continued from previous page
RF - exceeds_threshold

Table 4.28: Random Forest Validation Validation Statistics - exceeds_threshold

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Sensitivity	0.428571	0.74576	0.27053	0.46053	0.9552
Specificity	0.998337	0.96644	0.98533	0.94938	0.7538
Pos Pred Value	0.562500	0.49254	0.48696	0.52369	0.9382
Neg Pred Value	0.997152	0.98864	0.96330	0.93574	0.8113
Prevalence	0.004966	0.04185	0.04895	0.10783	0.7964
Detection Rate	0.002128	0.03121	0.01324	0.04966	0.7607
Detection Prevalence	0.003783	0.06337	0.02719	0.09482	0.8108
Balanced Accuracy	0.713454	0.85610	0.62793	0.70495	0.8545

Table 4.29: Random Forest Validation Class Statistics - exceeds_threshold

This validation shows an accuracy of 0.85, and metrics for this fit that resemble closely the trained model's kappa statistic, therefore we can say it is consistent with the explanatory power of the constructed model according to training metrics. These results suggest that the trained model performs well when predicting on previously unseen data.

4.3 Support Vector Machines

Lastly, Lag time (*lag_centroid_peak_event*), the Moment of Relative Peak Discharge (*peakq_moment*) and the Flood Stage Threshold Exceedance (*exceeds_threshold*) were modeled by fitting parameter-tuned Support Vector Machine (SVM) models, using a tuning grid to find the most optimal parameters (σ and C) for the radial basis kernel that was used. This way, optimal parameter settings were found for a model which would minimize error measures, or maximize performance measures. Additionally, these models were trained using 10 times 10-fold cross-validation in order to mitigate overfitting on the training dataset, and once trained these were also tested to predict known outputs on a validation (holdout, not included in training) dataset.

4.3.1 Lag Time Modeling

This Lag Time model was took ∼200 hours to train. Parameter tuning was performed for ten evenly-spaced values of σ and C, both ranging from 0 to 5. Parameter tuning results are presented in Figure 4.25.

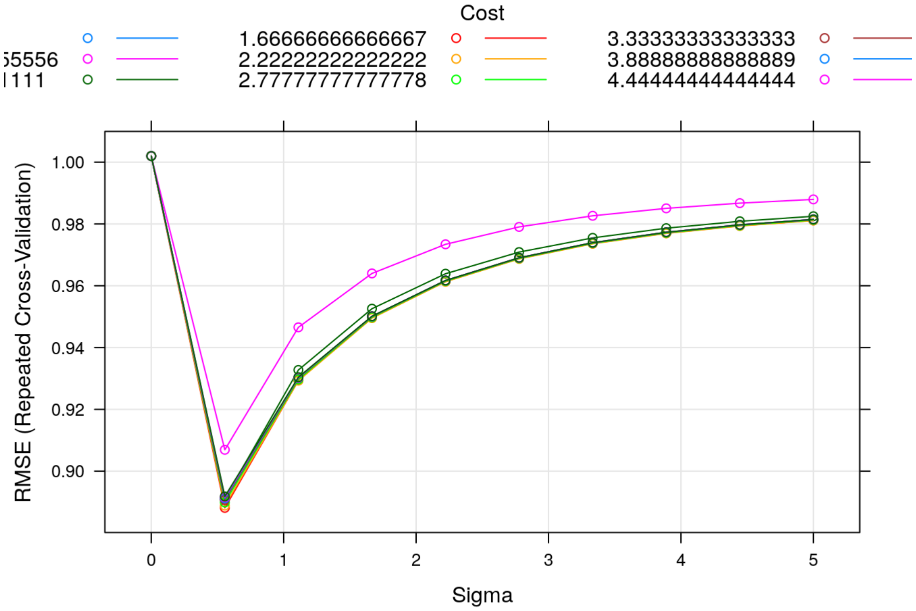

Figure 4.25: SVM: Lag Time Training - Parameter Tuning results

The final model produced by the tuning process, was achieved by using values sigma = 0.5555556 and C = 1.666667, where RMSE dropped at around 0.89. The structure and output of the best model found is shown in the **Appendix** on Listing 5.4.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results are shown in Figure 4.26 and Table 4.30.

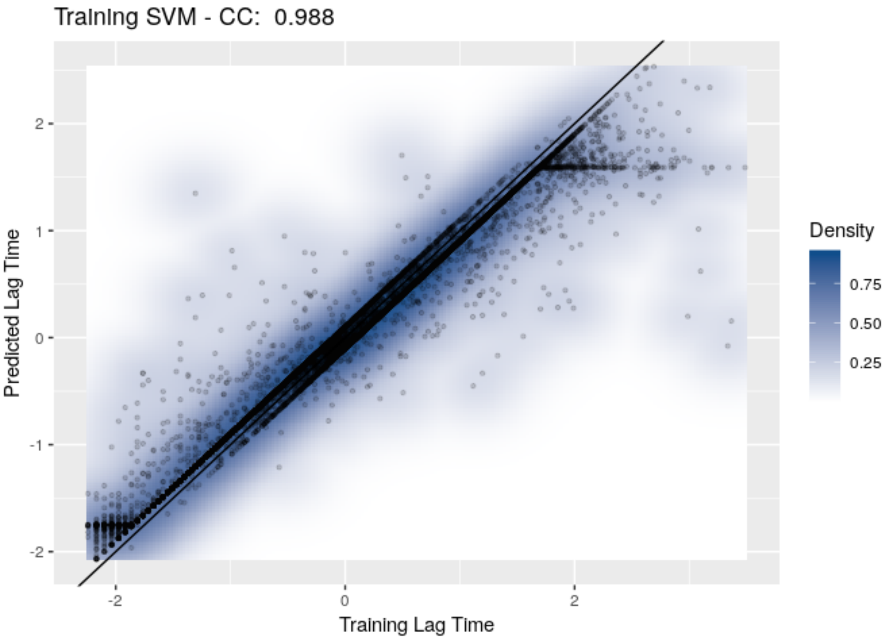

Figure 4.26: SVM: Lag Time fit using training data

Baseline	**Metrics**
CC	0.988
MAE	0.123
MSE	0.034
MPE	0.216
MAPE	0.603
Rsq	0.977

Table 4.30: SVM Basline Error Metrics - Lag Time

This baseline shows a correlation coefficient between the expected and predicted values of 0.988, and error metrics for this fit lie between 0.03 and 0.6. Given the above plot this model exhibits a extremely high correlation between the fitted model and the original response variable which could be an indication of overfitting. However, given the implementation of a bagged tree approach and 10x10-fold cross-validation, the performance of this model on unseen data should be consistent with a training R^2 of 0.97.

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation results and error metrics are presented in Figure 4.27 and Listing 4.31.

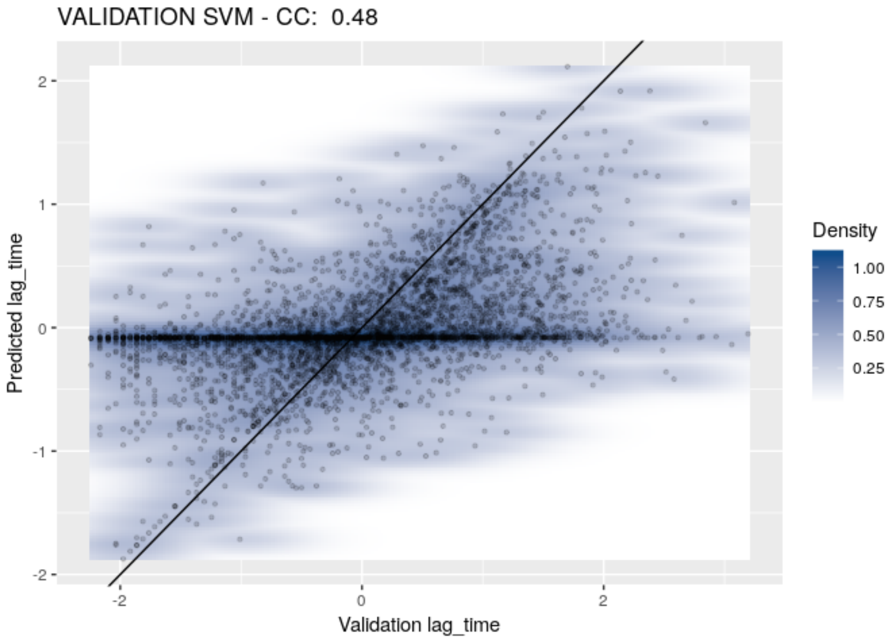

Figure 4.27: SVM: Lag Time fit using validation data

Validation Metrics	
CC	0.48
MAE	0.673
MSE	0.763
MPE	1.234
MAPE	1.714
Rsq	0.23

Table 4.31: SVM Validation Error Metrics - Lag Time

This validation shows a correlation coefficient between the expected and predicted values of 0.48, which is considerably lower than the baseline previously established on the training dataset. The error metrics for this fit lie between 0.67 and 0.76. The R^2 value for this fit diverges drastically from the explanatory power of the constructed model according to training metrics. These results suggest that the trained model underperforms dramatically when predicting on previously unseen data.

4.3.2 Moment of Relative Peak Discharge Modeling

This Moment of Relative Peak Discharge Modeling model took ∼230 hours to train. Parameter tuning was performed for ten evenly-spaced values of σ and C, both ranging from 0 to 5. Parameter tuning results are presented in Figure 4.28.

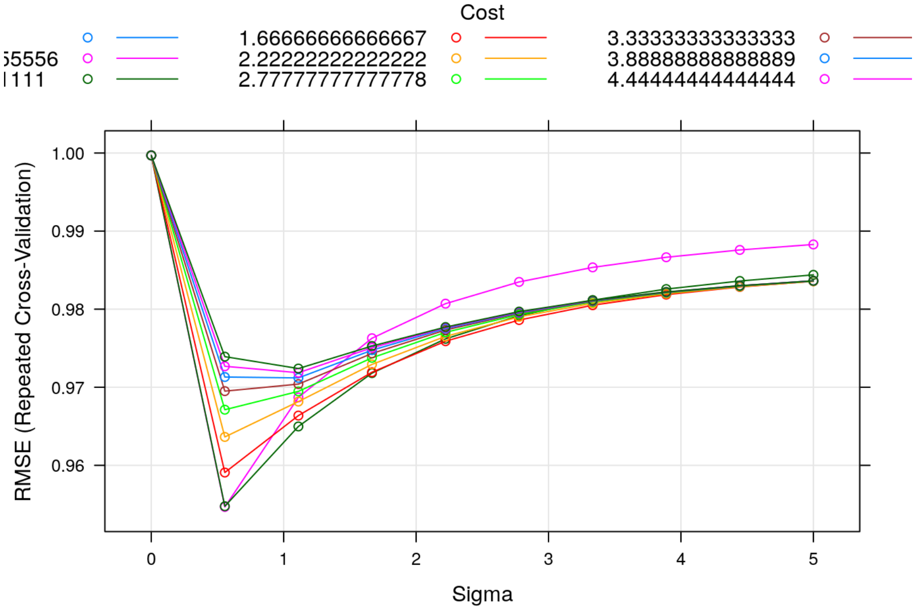

Figure 4.28: SVM: Moment of Relative Peak Discharge Training - Parameter Tuning results

The final model produced by the tuning process, was achieved by using values sigma = 0.5555556 and C = 0.5555556, where RMSE dropped at around 0.95. The structure and output of the best model found is shown in the **Appendix** on Listing 5.5.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results are shown in Figure 4.29 and Listing 4.32.

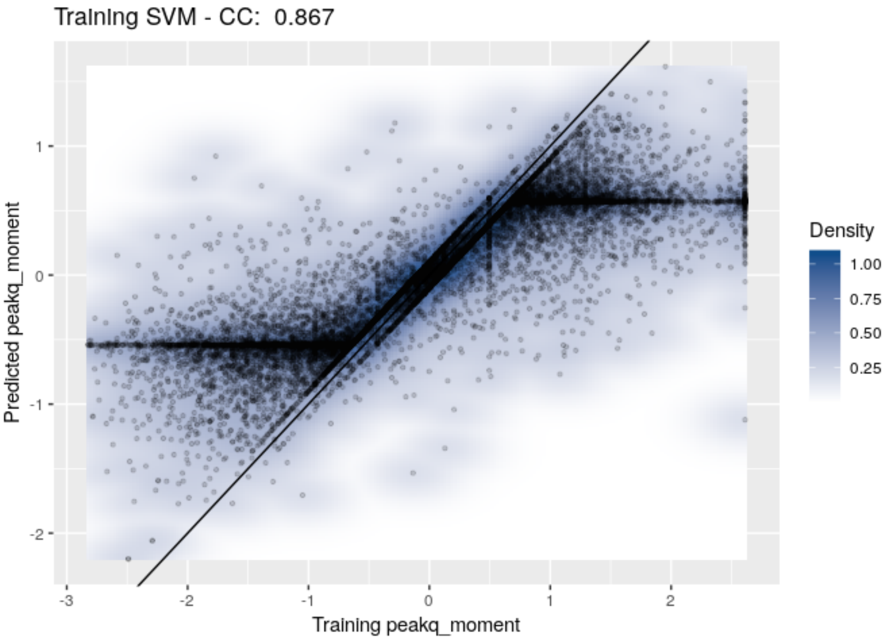

Figure 4.29: SVM: Moment of Relative Peak Discharge fit using training data

Baseline	**Metrics**
CC	0.867
MAE	0.401
MSE	0.39
MPE	0.307
MAPE	0.881
Rsq	0.751

Table 4.32: SVM Baseline Error Metrics - peakq_moment

This baseline shows a correlation coefficient between the expected and predicted values of 0.867, and error metrics for this fit lie between 0.3 and 0.4. Given the above plot this model exhibits a extremely high correlation between the fitted model and the original response variable which could be an indication of overfitting. However, given the implementation of a bagged tree approach and 10x10-fold cross-validation, the performance of this model on unseen data should be consistent with a training RMSE of 0.95 for unseen data.

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation results and error metrics are presented in Figure 4.30 and Listing 4.33.

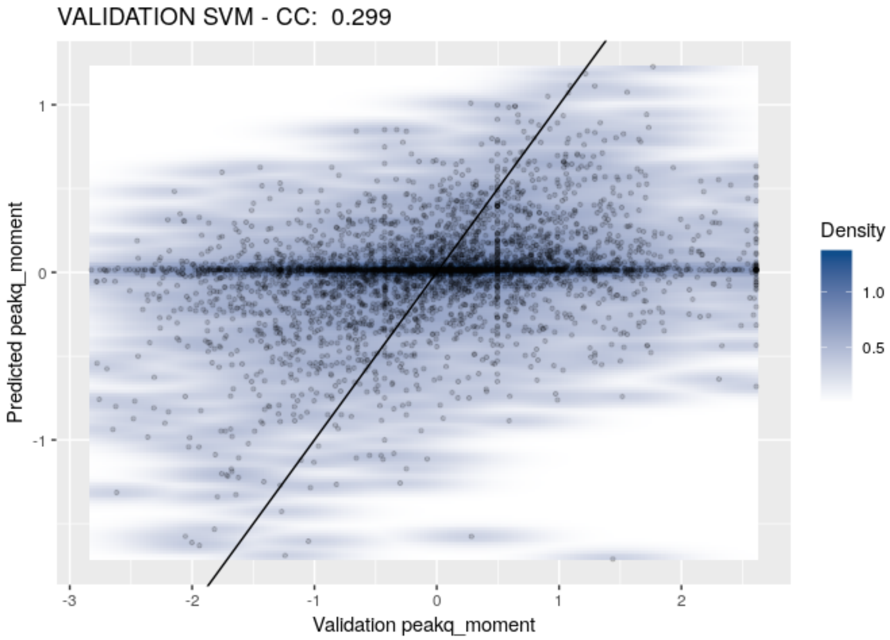

Figure 4.30: SVM: Moment of Relative Peak Discharge fit using validation data

Validation Metrics	
CC	0.299
MAE	0.737
MSE	0.915
MPE	0.996
MAPE	1.491
Rsq	0.089

Table 4.33: SVM Validation Error Metrics - peakq_moment

This validation shows a correlation coefficient between the expected and predicted values of 0.299, which is considerably lower than the baseline previously established on the training dataset. The error metrics for this fit lie between 0.73 and 0.91. The R^2 value for this fit is consistent with the explanatory power of the constructed model according to training metrics. These results suggest that the trained model underperforms dramatically when predicting on previously unseen data, however given the high training error figure, this is a consistent behavior.

4.3.3 Flood Stage Threshold Exceedance Modeling

This Flood Stage Threshold Exceedance model took ~150 hours to train. Parameter tuning was performed for ten evenly-spaced values of σ and C, both ranging from 0 to 5. Parameter tuning results are presented in Figure 4.31.

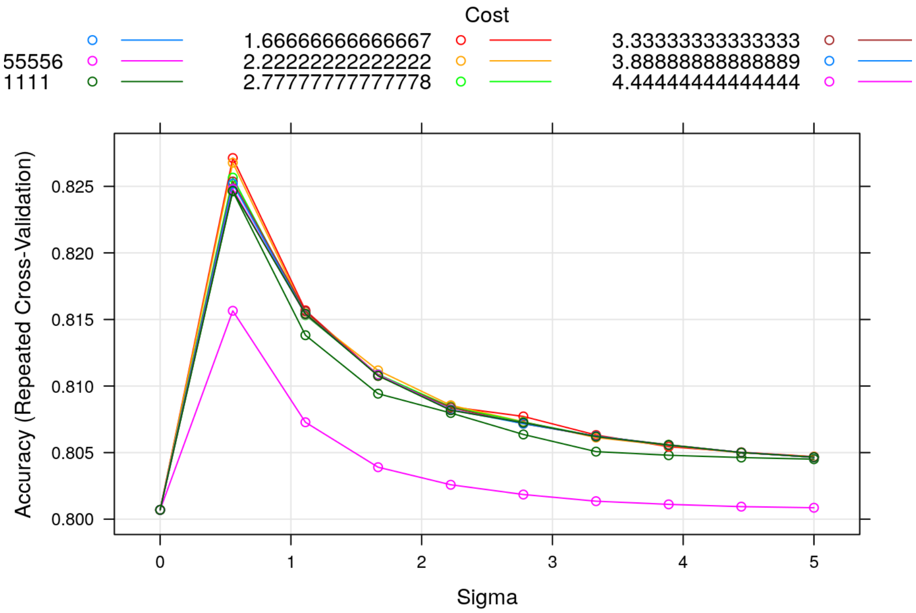

Figure 4.31: SVM: Flood Stage Threshold Exceedance Training - Parameter Tuning results

The final model produced by the tuning process, was achieved by using values sigma = 0.5555556 and C = 1.666667, where Accuracy peaked at around 0.827. The structure and output of the best model found is shown in the **Appendix** on Listing 5.6.

In order to establish a baseline, the trained model was tested against the expected results from the samples in the training dataset. Baseline results are shown in Tables 4.34, 4.35 and 4.36.

	Reference				
Prediction	**0**	**1**	**2**	**4**	**8**
0	71	1	0	0	0
1	10	729	25	20	8
2	0	8	556	6	3
4	0	16	31	1770	26
8	0	6	17	105	13506

Table 4.34: SVM Basline Confusion Matrix - exceeds_threshold

SVM - exceeds_threshold	
Accuracy	0.9833

Table 4.35 continued from previous page

SVM - exceeds_threshold	
95% CI	(0.9813, 0.9852)
No. of information Rate	0.8007
P-value [Acc >NIR]	<2.2e-16
Kappa	0.9508
Mcnemar's Test P-Value	NA

Table 4.35: SVM Baseline Statistics - exceeds_threshold

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Sensitivity	0.876543	0.95921	0.88394	0.9311	0.9973
Specificity	0.999941	0.99610	0.99896	0.9951	0.9620
Pos Pred Value	0.986111	0.92045	0.97033	0.9604	0.9906
Neg Pred Value	0.999406	0.99808	0.99553	0.9913	0.9887
Prevalence	0.004789	0.04493	0.03719	0.1124	0.8007
Detection Rate	0.004198	0.04310	0.03287	0.1046	0.7985
Detection Prevalence	0.004257	0.04683	0.03388	0.1090	0.8061
Balanced Accuracy	0.938242	0.97766	0.94145	0.9631	0.9796

Table 4.36: SVM Baseline Class Statistics - exceeds_threshold

This baseline shows an accuracy of around 0.98%, with a Kappa statistic of 0.95. Given the above results, this models exhibits a very high correlation between the fitted model and the original response variable with could be a indication of overfitting. However, given the implementation of 10x10-fold cross-validation throughout the parameter tuning process, the performance of this model on unseen data should still be consistent with the training accuracy of 0.82.

Having constructed this baseline, now the trained model will be used to predict the expected values from the validation dataset, which where not part of the training data. Validation results are shown in Tables 4.37, 4.38 and 4.39.

	Reference				
Prediction	**0**	**1**	**2**	**4**	**8**
0	4	4	0	1	0
1	5	61	19	11	6
2	0	8	27	17	9
4	0	11	35	97	43
8	12	93	126	330	3310

Table 4.37 continued from previous page

	Reference

Table 4.37: SVM Validation Confusion Matrix - exceeds_threshold

SVM - exceeds_threshold	
Accuracy	0.8274
95% CI	(0.8156, 0.8387)
No. of information Rate	0.7964
P-value [Acc >NIR]	1.879e-07
Kappa	0.3475
Mcnemar's Test P-Value	NA

Table 4.38: SVM Validation Statistics - exceeds_threshold

Statistic	Class: 0	Class: 1	Class: 2	Class: 4	Class: 8
Sensitivity	0.1904762	0.34463	0.130435	0.21272	0.9828
Specificity	0.9988118	0.98988	0.991546	0.97641	0.3484
Pos Pred Value	0.4444444	0.59804	0.442623	0.52151	0.8551
Neg Pred Value	0.9959716	0.97189	0.956814	0.91120	0.8380
Prevalence	0.0049657	0.04185	0.048948	0.10783	0.7964
Detection Rate	0.0009459	0.01442	0.006384	0.02294	0.7827
Detection Prevalence	0.0021282	0.02412	0.014424	0.04398	0.9153
Balanced Accuracy	0.5946440	0.66726	0.560991	0.59457	0.6656

Table 4.39: SVM Validation Class Statistics - exceeds_threshold

This validation shows an accuracy of 0.82, and metrics for this fit that resemble closely the trained model's kappa statistic as well as the training accuracy. Therefore, we can say that it is consistent with the explanatory power of the constructed model according to training metrics. These results suggest that the trained model performs well when predicting previously unseen data. Once more, clear signs of unbalanced training classes can be seen in the per-class statistics. However, these results seem to project a much homogeneous class accuracy than the ones observed for MARS and Random Forest.

4.4 Model Performance Summary

Overall, the MARS approach seems to be able to produce models that avoid overfitting on the training data. Even though predictive power appears to be in a modest range, its performance is maintained very consistently when presented with unseen samples. This is true for both regression as well as classification models. Additionally, from all three models, MARS took the least amount of time to train continuous target variables; time is extended when performing classification, as a model must be fit for each class in the response.

Concerning Random Forest, even though it's predictive power is generally similar to what was achieved with MARS (slightly higher, but not really significant), it tends to overfit dramatically on its training dataset. Even though training times were higher for regression, Random Forest really shines for classification, showing near MARS-based regression performance.

Lastly, Support Vector Machines appear to produce both classification and regression models, that in spite of requiring substantially more time to train, and a more rigorous and extensive parameter tuning, offer no significant overall performance increase. Additionally, SVMs are prone not only to overfitting on the training data, but also introduce strange artifacts on verification data, likely due to unexplained variance and their reliance on higher order spatial transformations. However, improvement was seen regarding the characterization of unbalanced classes. Furthermore, SVM offers no built-in assessments for variable importance, which would require the implementation of alternatives like randomized stepwise variable selection into the training process, but given the extensive aforementioned training times, this would only add up to them.

Table 4.40 presents a summary of training and validation statistics for all the models built. Note that in order to present this table as a whole, column names had to be abbreviated. **VAL.** represents validation results, while **TRN.** represents Training results. **CC** stands for Correlation Coefficient, *Rsq* represent the Coefficient of Determination R^2, **MSE** is Mean Squared Error and **ACC.** stands for Accuracy.

MODEL	TARGET	TYPE	TRN. CC	VAL. CC	TRN. ACC.	VAL. ACC.	TRN. KAPPA	VAL. KAPPA	TRN. MSE	VAL. MSE	TRN. Rsq	VAL. Rsq
MARS	lag time	Regression	0.658	0.645	-	-	-	-	0.569	0.577	0.433	0.416
MARS	peakq moment	Regression	0.391	0.332	-	-	-	-	0.846	0.898	0.153	0.110
MARS	exceeds threshold	Classification	-	-	0.856	0.844	0.5288	0.5082	-	-	-	-
Random Forest	lag time	Regression	0.965	0.664	-	-	-	-	0.095	0.552	0.931	0.441
Random Forest	peakq moment	Regression	0.826	0.404	-	-	-	-	0.400	0.843	0.683	0.163
Random Forest	exceeds threshold	Classification	-	-	0.943	0.857	0.8322	0.5813	-	-	-	-
Support Vector Machines	lag time	Regression	0.988	0.480	-	-	-	-	0.034	0.763	0.977	0.23
Support Vector Machines	peakq moment	Regression	0.867	0.299	-	-	-	-	0.390	0.915	0.751	0.089
Support Vector Machines	exceeds threshold	Classification	-	-	0.9833	0.8274	0.9508	0.3475	-	-	-	-

Table 4.40: Model Performance and Error Metrics

4.5 Variable Importance Summary

Regarding insights gained about variable importance for modeling each of the proposed responses, table 4.41 summarizes each model's assessment for characterizing each response. In cases where both *est_area* and *totalBasinPixels* were selected together, only the highest ranking one will be shown.

For characterizing lag time, common variables between models are *prod_mean*, *mf.event* and *precip_sdev*. In the case of the moment of relative peak discharge, *imperviouscell*, *bio_7* and *precip_sdev* are common choices between MARS and Random Forest. Lastly, characterization of flood stage threshold exceedance seems to be commonly associated with *est_area* and *G1* by both techniques. Statistical and catchment-based precipitation and flow distance moments, as well as morphological variables are common to all of these characterizations.

Variable Importance					
MARS			**Random Forest**		
Lag time	**peakq_moment**	**exceeds_threshold**	**Lag Time**	**peakq_moment**	**exceeds_threshold**
prod_mean	bio_10	est_area	mf.event	flowdist_mean	cncell
mf.event	ruggedness	mf.event	prod_mean	lbm	bio_3
precip	slopeoutlet	prod_mean	EcartVertical	imperviouscell	flowdist_skew
flowdist_mean	imperviouscell	slopeoutlet	prod_sdev	rl	bio_18
precip_sdev	precip_sdev	bio_10	precip_mean	bio_7	est_area
bio_2	bio_1	G1	precip_sdev	G2	G1
snowpercent	bio_7	imperviousbasin	delta_2	precip_sdev	rl

Table 4.41: Variable Importance Summary

4.6 Probability of Flood Stage Threshold Exceedance

Classification models were built for characterizing the exceedance of flood stage thresholds, and these were also used for prediction using the validation dataset. Effectively forecasts were made for each of the threshold exceedance classes, which means class probabilities were able to be extracted from these models and forecasts. By doing so, we were able to compare and contrast the skill of each classification model to predict (or forecast) the probability of each verification sample for each of the threshold exceedance classes. In order to assess this skill in a comprehensive but straightforward way, reliability diagrams were built for each model's per-class skill.

Reliability diagrams are commonly used statistical tools in the atmospheric sciences, used to represent the performance of probability forecasts of dichotomous events. These diagrams consist of only the plot of observed relative frequency as a function of forecast probability, where the 1:1 diagonal line implies perfect reliability. Additionally, a summary of the frequency distribution of forecast values is shown, given that the plotted points represent the conditional distribution of observations. This allows for a compact display of the full distribution of forecasts and observations [18].

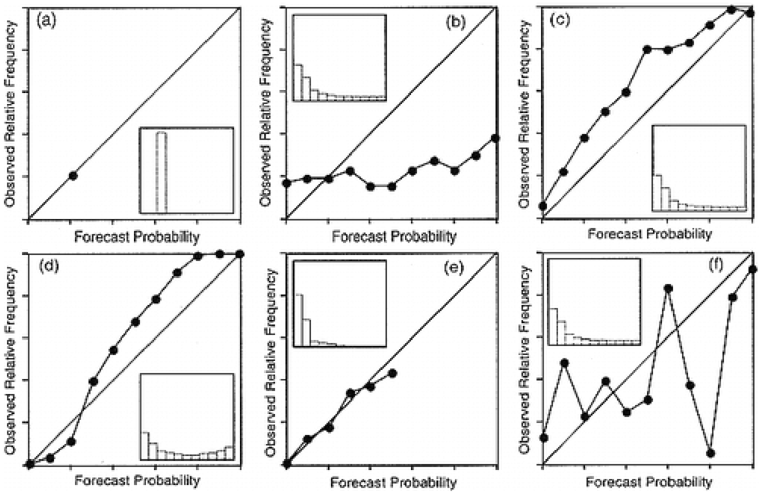

Figure 4.32: Examples of hypothetical reliability diagrams [18]

Figure 4.32 shows the stacked reliability diagrams for the forecast of the class No-Exceedance, product of all three models for floodstage threshold exceedance. Given the skewed distribution of samples in the histograms and the sparse distribution of points, we could say that all three forecasts are product of a limited dataset, which due to it's small sample size of observations for this class lead to unreliable forecasts for this class. Of the three, Random Forest appears to underestimate consistently throughout the distribution.

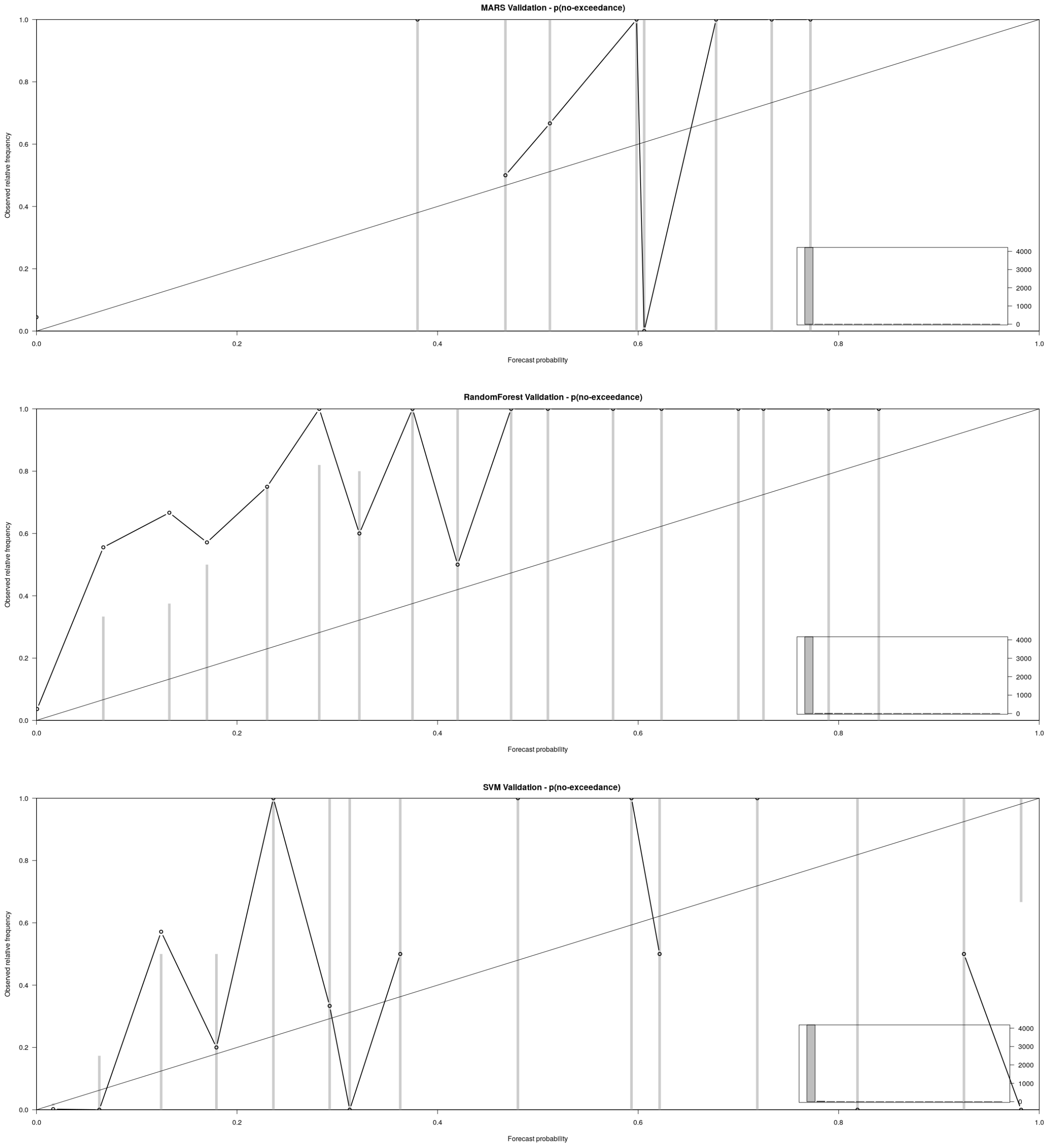

Figure 4.33: No-Exceedance Reliability Diagrams: MARS (top), Random Forest (middle), SVM (bottom)

Figure 4.33 shows the stacked reliability diagrams for the forecast of the class Exceeds Action, product of all three models for floodstage threshold exceedance. All three diagrams show a similar behavior along the perfect reliability diagram, generally overestimating on higher probability values, and underestimating towards lower values. All models appear to behave similarly, but Support Vector Machines shows a smoother overall behavior near the 1:1 line. Random Forest shows a consistent overestimation trend past probability values of 0.3.

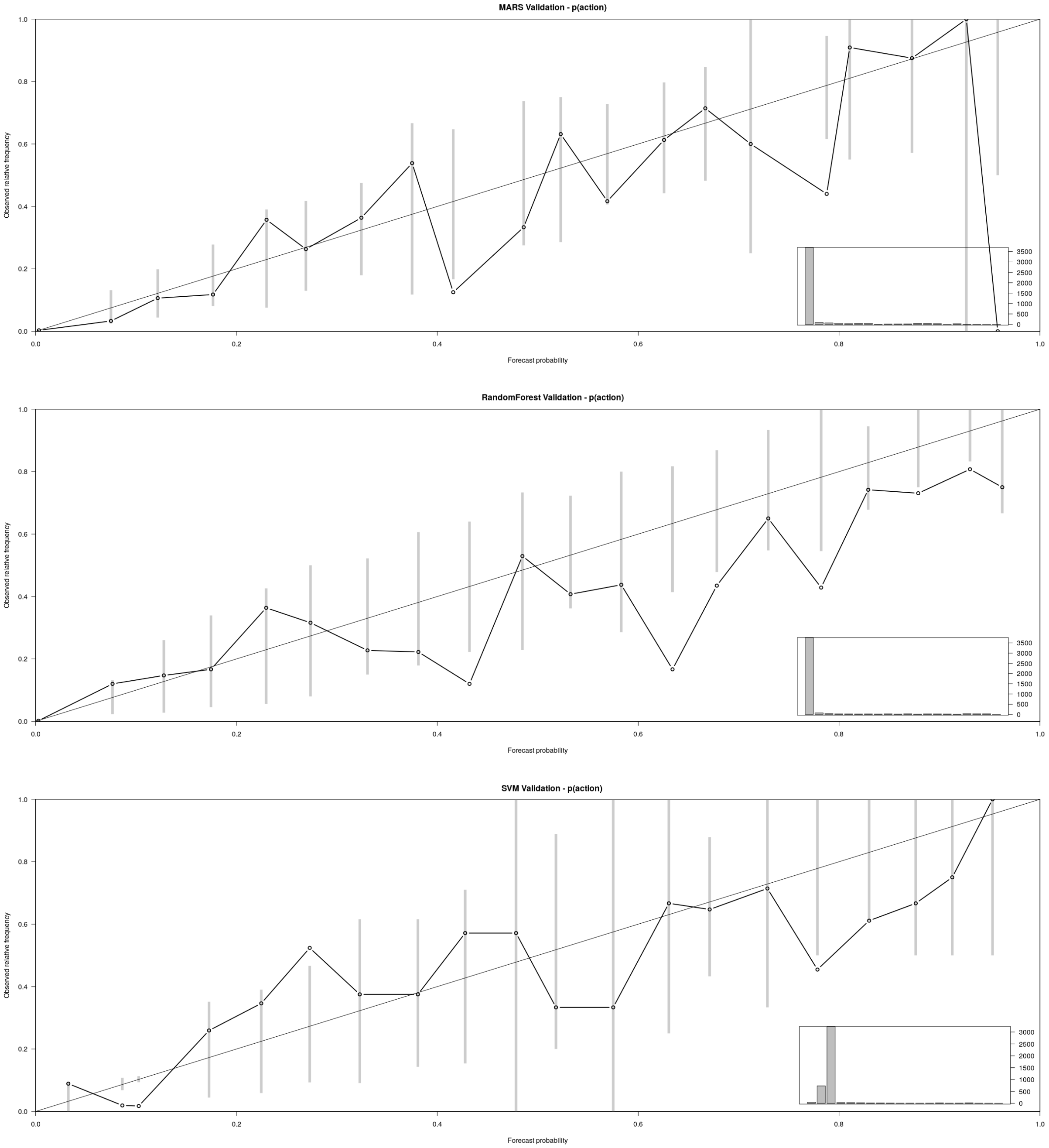

Figure 4.34: Exceeds Action Reliability Diagrams: MARS (top), Random Forest (middle), SVM (bottom)

Figure 4.34 shows the stacked reliability diagrams for the forecast of the class Exceeds Minor, product of all three models for floodstage threshold exceedance. MARS seems to perform poorly compared to the other two models, however, Random Forest seems to be the best performer of them all. The overall trend for the three models is to underestimate low, and overestimate high probabilities.

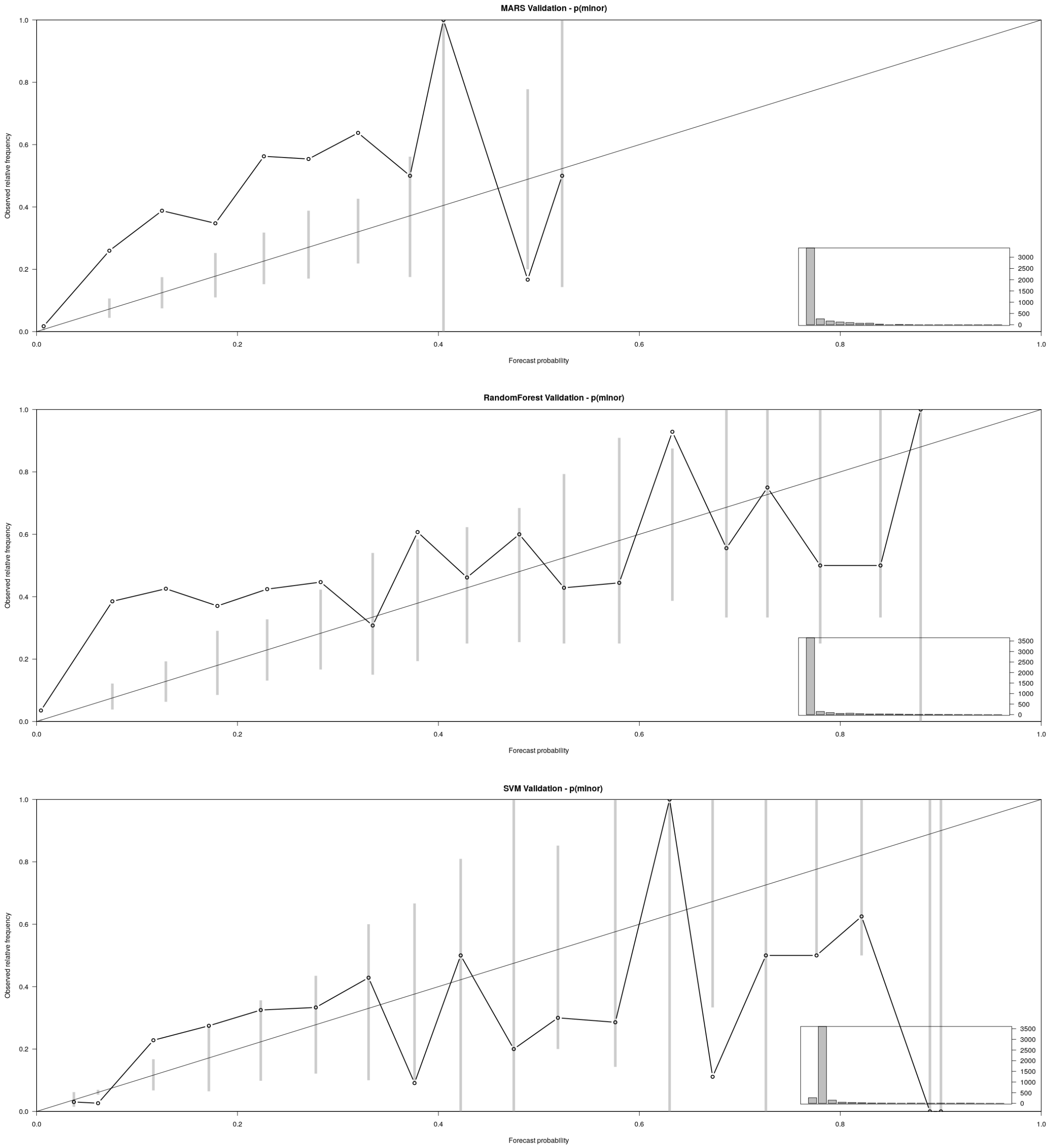

Figure 4.35: Exceeds Minor Reliability Diagrams: MARS (top), Random Forest (middle), SVM (bottom)

Figure 4.35 shows the stacked reliability diagrams for the forecast of the class Exceeds Moderate, product of all three models for floodstage threshold exceedance. Regarding the forecast of this class, even though MARS shows no frequency of observed probabilities beyond 0.85, it appears to have a smoother distribution across the perfect reliability line. In general, all three models struggle with high probability values, which seem to be scarce for this class.

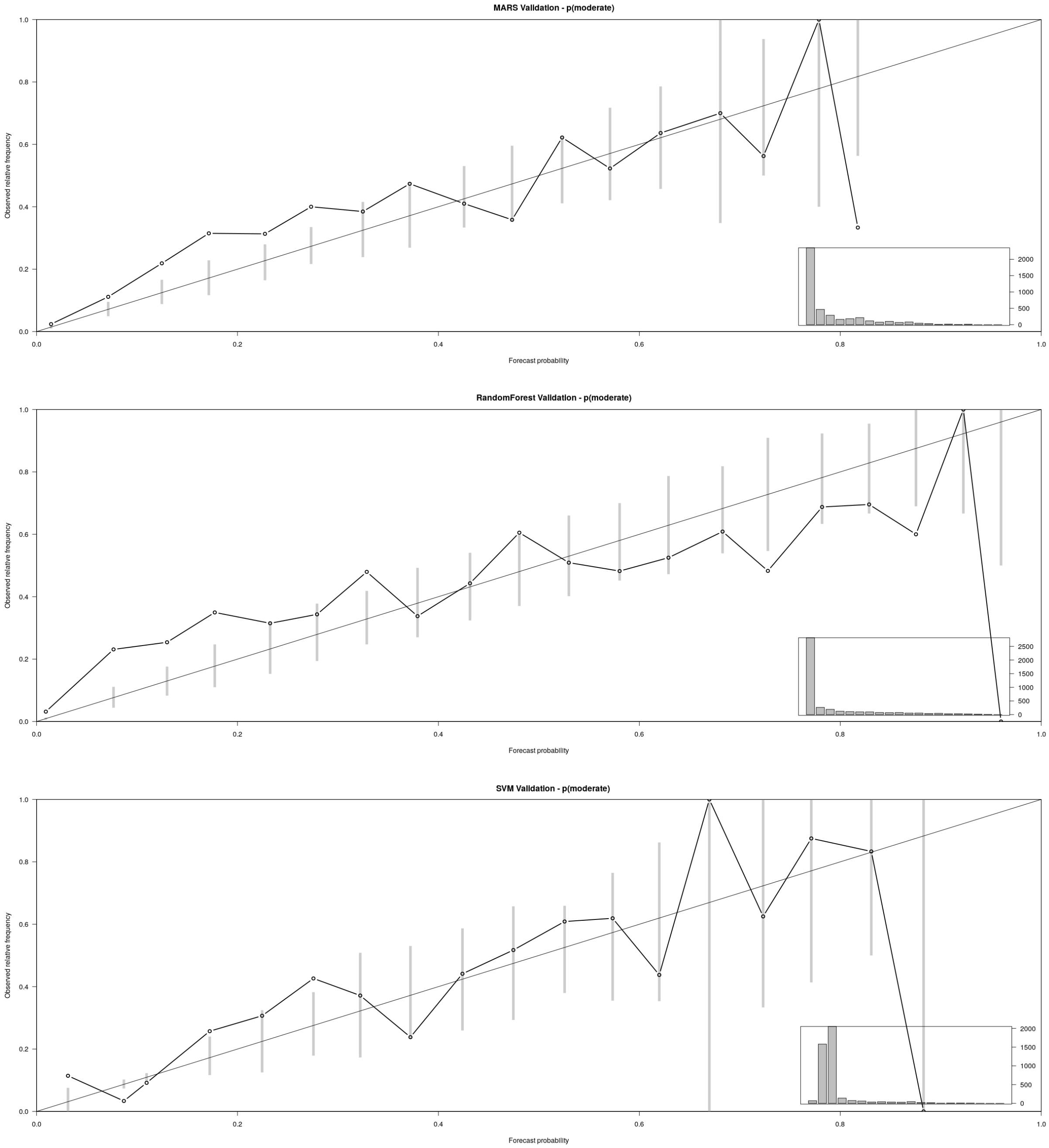

Figure 4.36: Exceeds Moderate Reliability Diagrams: MARS (top), Random Forest (middle), SVM (bottom)

Lastly, Figure 4.36 shows the stacked reliability diagrams for the forecast of the class Exceeds Major, product of all three models for floodstage threshold exceedance. This is clearly the best performing class for all three models, as they all exhibit remarkable skill to forecast this flood stage exceedance. Random Forest and MARS show similar underestimation for low probability values, which is smaller in Random Forest, and seems to be entirely mitigated when using Support Vector Machines. However, as values move higher, SVM's performance deteriorates. MARS' performance is really good staying near the perfect reliability line beyond values of 0.3, and Random Forest exhibits unparalleled skill beyond values of 0.4.

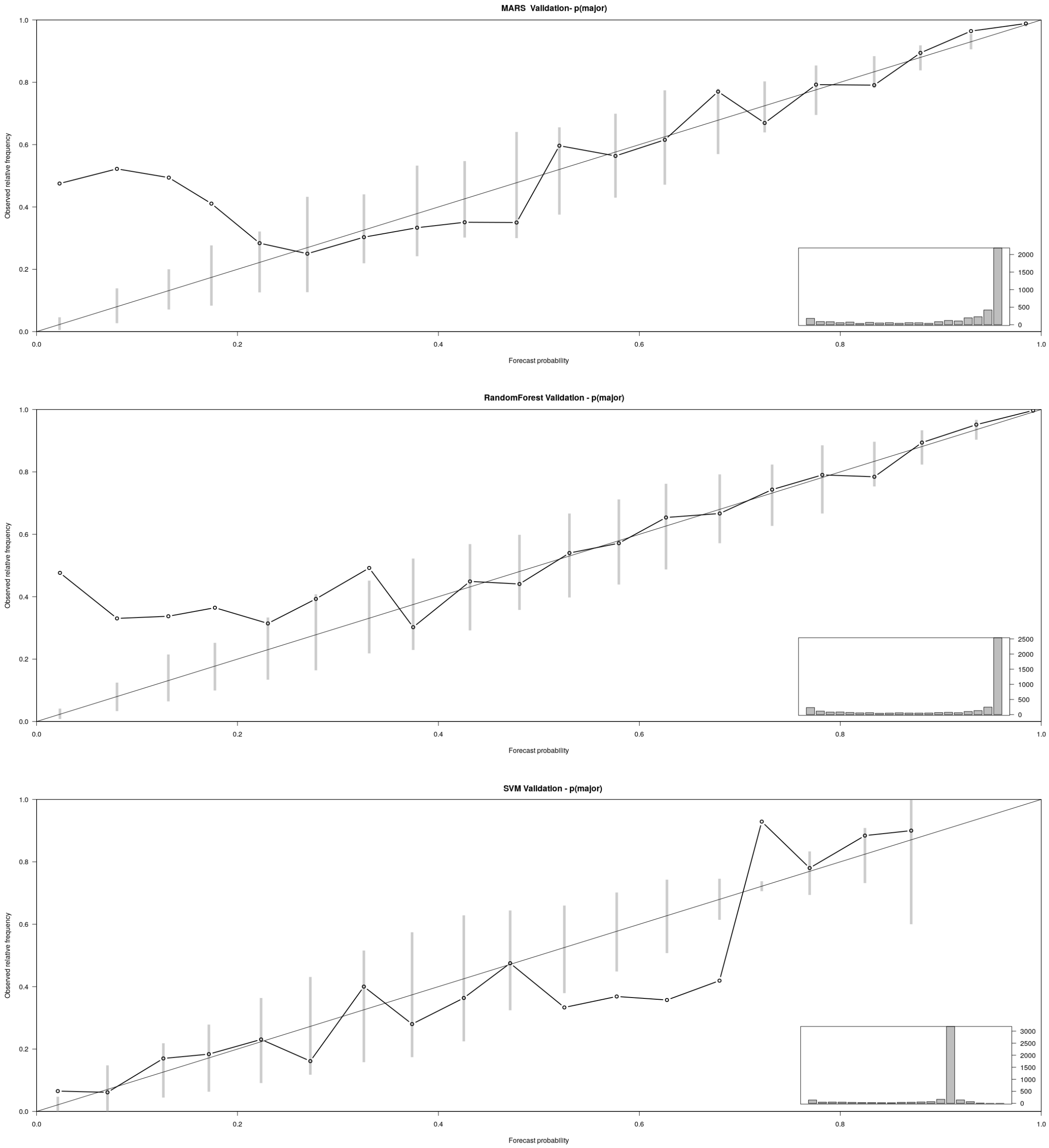

Figure 4.37: Exceeds Major Reliability Diagrams: MARS (top), Random Forest (middle), SVM (bottom)

In general, models show better skill for forecasting the Exceeds Action and the Exceeds Major classes; particularly the later. This is also a product of the class imbalance present in the data, and these model's performance on the Exceeds Major class is a clear example of the types of skill that can be expected of each of these types of models for this particular case. There are clear trade-offs between the different types of models, particularly when comparing overall performance with training time and model complexity. Also, reliability diagrams prove to be a concise way of assessing and comparing model skills for classification problems such as this one.

Chapter 5

Conclusions

Through various analyses, three distinct machine learning models, based on three fundamentally different learning techniques were objectively compared in terms of their training times, performance, variable importance and forecast skill, leading to a comprehensive case study for future research pertaining the use of catchment-scale rainfall moments in the characterization of flood-related responses.

Having applied the CRISP-DM methodology on a physically-based spatial precipitation moment flood event database, this study has effectively performed a data-driven statistical analysis, leading to the characterization of floods by using machine learning techniques. The models built from this dataset which included catchment-scale precipitation moments were presented and analyzed, showing that effective characterization of flood-related phenomena such as Lag Time and Flood Stage Threshold Exceedance is possible.

Additionally, through variable importance analysis the relevant factors that characterize these responses were able to be determined, described and compared between models. Furthermore, even though the newly proposed Moment of Relative Peak Discharge showed little correlation with the available predictors, and model errors were considerably high across model types, it was possible to determine the factors that contribute to the characterization of this flow response index.

Lastly, by training and validating classification models for the flood stage threshold exceedance, probabilistic class forecasts were able to be produced, which led to the probabilistic characterization of model skills for predicting specific flood stage exceedance classes. These class forecasts were successfully compared between models by using reliability diagrams to assess their skill in characterizing each of these classes.

MARS has proven to be the most consistent performer among all three models, as well as the most efficient one for continuous responses. It consistently demonstrated resistance to overfitting, overall training times were the lowest. Random Forest offers a marginal improvement over MARS in terms of regression performance, however it swiftly outperformed MARS when used for classification. Random Forest exhibited prominent tendencies toward overfitting on training data, regardless of implementing 10 times 10-fold cross-validation, but training metrics provided sane predictive power estimates for unseen samples. Lastly, Support Vector Machines represented a cumbersome exercise in parameter tuning. Even though interesting results were evidenced regarding class-specific probabilistic forecasting skill for flood stage threshold exceedance, their overall performance was not significantly better compared to the other two alternatives.

Future research looking to build upon the present study should bear in mind the following recommendations. First, a more robust variable selection exercise and methodology could be implemented. Second, even though statistical moments of precipitation and flow distance appear to be relevant for modeling these hydrological responses, the fourth statistical moment Kurtosis, appears to hardly hold any relevant influence over the target variables explored here. Perhaps a thorougher assessment of whether high order moments hold significant value is in order.

Bibliography

[1] Guy Delrieu, John Nicol, Eddy Yates, Pierre-Emmanuel Kirstetter, Jean-Dominique Creutin, Sandrine Anquetin, Charles Obled, Georges-Marie Saulnier, Véronique Ducrocq, Eric Gaume, Olivier Payrastre, Hervé Andrieu, Pierre-Alain Ayral, Christophe Bouvier, Luc Neppel, Marc Livet, Michel Lang, Jacques Parent du Châtelet, Andrea Walpersdorf, and Wolfram Wobrock. The catastrophic flash-flood event of 8–9 september 2002 in the gard region, france: A first case study for the cévennes–vivarais mediterranean hydrometeorological observatory. *Journal of Hydrometeorology*, 6(1):34–52, February 2005. doi: 10.1175/jhm-400.1. URL `https://doi.org/10.1175/jhm-400.1`.

[2] Manabendra Saharia, Pierre-Emmanuel Kirstetter, Humberto Vergara, Jonathan J. Gourley, Yang Hong, and Marine Giroud. Mapping flash flood severity in the united states. *Journal of Hydrometeorology*, 18(2):397–411, February 2017. doi: 10.1175/jhm-d-16-0082.1. URL `https://doi.org/10.1175/jhm-d-16-0082.1`.

[3] Michael B. Smith, Victor I. Koren, Ziya Zhang, Seann M. Reed, Jeng-J. Pan, and Fekadu Moreda. Runoff response to spatial variability in precipitation: an analysis of observed data. *Journal of Hydrology*, 298(1-4):267–286, October 2004. doi: 10.1016/j.jhydrol.2004.03.039. URL `https://doi.org/10.1016/j.jhydrol.2004.03.039`.

[4] D. Zoccatelli, M. Borga, F. Zanon, B. Antonescu, and G. Stancalie. Which rainfall spatial information for flash flood response modelling? A numerical investigation based on data from the Carpathian range, Romania. *Journal of Hydrology*, 394(1-2):148–161, November 2010. doi: 10.1016/j.jhydrol.2010.07.019. URL `https://doi.org/10.1016/j.jhydrol.2010.07.019`.

[5] D. Zoccatelli, M. Borga, A. Viglione, G. B. Chirico, and G. Blöschl. Spatial moments of catchment rainfall: rainfall spatial organisation, basin morphology, and flood response. *Hydrology and Earth System Sciences*, 15(12):3767–3783, December 2011. doi: 10.5194/hess-15-3767-2011. URL `https://doi.org/10.5194/hess-15-3767-2011`.

[6] Audrey Douinot, Hélène Roux, Pierre-André Garambois, Kévin Larnier, David Labat, and Denis Dartus. Accounting for rainfall systematic spatial variability in flash flood forecasting. *Journal of Hydrology*, 541:359–370, October 2016. doi: 10.1016/j.jhydrol.2015.08.024. URL `https://doi.org/10.1016/j.jhydrol.2015.08.024`.

[7] I. Emmanuel, H. Andrieu, E. Leblois, N. Janey, and O. Payrastre. Influence of rainfall spatial variability on rainfall–runoff modelling: Benefit of a simulation approach? *Journal of Hydrology*, 531:337–348, December 2015. doi: 10.1016/j.jhydrol.2015.04.058. URL https://doi.org/10.1016/j.jhydrol.2015.04.058.

[8] Philip B. Bedient, Wayne C. Huber, and Baxter E. Vieux. *Hydrology and Floodplain Analysis*. Pearson, third edition, 2007. ISBN 0131745891.

[9] Catchment basins. http://basins.ghkates.com/catchment-basins/, Jun 2003.

[10] L is for lag time. https://snowhydro1.wordpress.com/2012/04/13/l-is-for-lag-time/, Mar 2013.

[11] US Department of Commerce and NOAA. High water level terminology, Aug 2016. URL https://www.weather.gov/aprfc/terminology.

[12] LA Weather Forecast Office NWS New Orleans/Baton Rouge. USGS gauge - Mississippi River at Reserve - Baton Rouge, LA. https://water.weather.gov/ahps2/hydrograph.php?wfo=lix&gage=rrvl1, 2019.

[13] IBM. Cross-industry standard process for data mining. https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm, 2012.

[14] Volkan Vural. Introduction: CRISP-DM. https://piazza-resources.s3.amazonaws.com/jsw5nwuudmj15n/ju3accscvig3f1/1_Intro.pdf, 2019.

[15] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 2011. ISBN 0123748569.

[16] Max Kuhn. *Applied Predictive Modeling*. Springer, May 2013.

[17] Alan J. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning (Springer Texts in Statistics)*. Springer, 2013. ISBN 0387781889.

[18] Daniel S. Wilks. *Statistical Methods in the Atmospheric Sciences, Volume 59: An Introduction (International Geophysics)*. Academic Press, feb 1995. ISBN 0127519653.

Appendix

5.1 Variable tables

VARIABLE	**DEFINITION**
fips	Federal Information Processing Standard - State and County codes
gauge	USGS gauge ID
lat	Latitude of gauge location
lon	Longitude of gauge location
start	Event start time
end	Event end time
peakq	Peak Flow
peakt	Time of peak flow
dt	Time difference start of the event and peak flow
HUC	Hydrological Unit Code - Watershed ID
agency	Agency who made the streamflow measurement: USGS
gname	Gauge Name
area	Quality-controlled USGS basin area
carea	Corrected (using basin delineation?) area
q2	Q Return Period - 2y
q5	Q Return Period - 5y
q10	Q Return Period - 10y
q25	Q Return Period - 25y
q50	Q Return Period - 50y
q100	Q Return Period - 100y
q200	Q Return Period - 200y
q500	Q Return Period - 500y
action	Flood stage threshold - ACTION
minor	Flood stage threshold - MINOR
moderate	Flood stage threshold - MODERATE
major	Flood stage threshold - MAJOR
regulation	Regulated or unregulated streams (discrete)
alpha	Kinematic Wave parameter: ALPHA

Table 5.1 continued from previous page

VARIABLE	DEFINITION
beta	Kinematic Wave parameter: BETA
cc	Correlation coefficient for the fit of alpha and beta
usgs_area	True (reported by USGS) drainage area
est_area	Estimated Area (from Digital Elevation Model; flow grids)
error	Relative error of estimated drainage area
el	Elongation Ratio; a measure of basin shape
k	Shape factor; a measure of basin shape
rl	River length
rr	Relief ratio; R divided by Basin Length (highly correlated with drainage area)
si	Slope index
rdd	Drainage density; number of streams divided by drainage area
rbm	Basin magnitude; the total number of first-order streams (streams whose only input is overland flow)
rfocf	Frequency of first-order channels; the basin magnitude divided by drainage area
slopeoutlet	Outlet Slope
precip	Climatological precipitation
temp	Climatological Average temperature
cnbasin	Basin average curve number
cncell	Outlet Cell average curve number
coemcell	Surface Roughness (function of Manning's roughness)
imperviousbasin	Basin total surface imperviousness
imperviouscell	Outlet cell surface imperviousness
kfact	K-factor; relative index of susceptibility of bare; cultivated soil to particle detachment and transport by rainfall
rockdepth	Depth to bedrock at the outlet
rockvolume	Volume of rock; similar to rock depth
bpartexture	A parameter related to soil
dc	Diameter of Circle with same Drainage Area as basin
ldd	Local Drainage Density at the outlet
lbm	Local Basin Magnitude
lfocf	Local Frequency of the First-Order Channels
ruggedness	Ruggedness expressed as drainage density multiplied by relief
fd	Flow duration; duration of the entire event
rt	Recession time; peak-to-end time
nfd	Normalized (Unit) Flood Duration = Flood Duration/Area
ntp	Normalized (Unit) Time to Peak
nrt	Normalized (Unit) Recession Time
nq	Normalized (Unit) Peak Discharge
f	Flashiness
Group.1	Auxiliary dataset merging variable

Table 5.1 continued from previous page

VARIABLE	DEFINITION
county	COUNTY
class	Koppen Geiger Climate Class
prop	Auxiliary dataset merging variable
state	State
month	Month
year	Year
season	Season in which the flood happened
maxseason	Season in which the Maximum Flood Peak was recorded in this Gauge
mf	Basin median Flashiness
mf.event	Event Flashiness
fness	Flashiness (discrete): a cutoff of 0.75 on f.ecdf means flashy; lower is categorized as non-flashy
f.ecdf	Empirical Cumulative Distribution Function values of event-based flashiness
eventID	Event ID
gaugenum	Gauge number
tp	Rise time; start-to-peak time
lag_start_peak_event	Time from start of rainfall to peak of flood based on MRMS
lag_centroid_peak_event	Time from centroid of rainfall to peak of flood based on MRMS
lag_max_peak_event	Time from maximum of rainfall to peak of flood based on MRMS
activatedBasinPixels	Total number of 1km x 1km gridcells in a basin that received rainfall from centroid of precipitation to flow peak
totalBasinPixels	Total number of 1km x 1km gridcells in a basin
precip_mean	Mean of precipitation accumulated during the centroid lag time period over the activated basin(=part of the basin where rainfall falls)
precip_sdev	Standard deviation of precipitation accumulated during the centroid lag time period over the activated basin(=part of the basin where rainfall falls)
precip_skew	Skewness of precipitation accumulated during the centroid lag time period over the activated basin(=part of the basin where rainfall falls)
precip_kurt	Kurtosis of precipitation accumulated during the centroid lag time period over the activated basin(=part of the basin where rainfall falls)
flowdist_mean	Mean of flow distance of the activated basin(=part of the basin where rainfall falls)
flowdist_sdev	Standard deviation of flow distance of the activated basin(=part of the basin where rainfall falls)

Table 5.1 continued from previous page

VARIABLE	DEFINITION
flowdist_skew	Skewness of flow distance of the activated basin(=part of the basin where rainfall falls)
flowdist_kurt	Kurtosis of flow distance of the activated basin(=part of the basin where rainfall falls)
prod_mean	Mean of the product of accumulated precipitation and flow distance of the activated basin(=part of the basin where rainfall falls)
prod_sdev	Standard deviation of the product of accumulated precipitation and flow distance of the activated basin(=part of the basin where rainfall falls)
prod_skew	Skewness of the product of accumulated precipitation and flow distance of the activated basin(=part of the basin where rainfall falls)
prod_kurt	Kurtosis of the product of accumulated precipitation and flow distance of the activated basin(=part of the basin where rainfall falls)
P0	Zero-th order moment of precipitation (Catchment-averaged rainfall)
P1	First-order moment of precipitation
P2	Second-order moment of precipitation
G1	First-order Moment of flow distance (Catchment averaged flow distance)
G2	Second-order Moment of flow distance
delta1	Delta 1 (Distance of the catchment rainfall centroid with respect to the catchment centroid. Values of d1 close to 1 reflect a rainfall distribution either concentrated close to the catchment centroid position or else spatially homogeneous. Values less than 1 (or greater than 1) indicate that rainfall is distributed downstream (or upstream).)
delta2	Delta 2 (Rainfall field dispersion (with respect to its mean position) relative to the dispersion of the flow distances. Values of d2 close to 1 reflect a uniform-like rainfall distribution, whereas values less (greater) than 1 indicate that rainfall is characterized by a uni- modal (multimodal) distribution along the flow distance)
EcartVertical	Vertical Gap (VG values close to zero indicate a rainfall distribution over the catchment revealing weak spatial variability. The higher the VG value; the more concentrated the rainfall over a small part of the catchment.)

Table 5.1 continued from previous page

VARIABLE	DEFINITION
EcartHorizontal	Horizontal Gap (HG values close to 0 reflect a rainfall distribution either concentrated close to the catchment centroid position or spatially homogeneous. Values less than 0 (greater than 0) indicate that rain- fall is distributed downstream (or upstream).)
casetag	Auxiliary spatial moment calculation variable - Flood event case tag
mean	Auxiliary dataset merging variable
std	Auxiliary dataset merging variable
a1	Auxiliary dataset merging variable
a12	Auxiliary dataset merging variable
a2	Auxiliary dataset merging variable
snowpercent	Percentage of Snow in the Gauge
bio1	Annual Mean Temperature
bio2	Mean Diurnal Range (Mean of monthly (max temp - min temp))
bio3	Isothermality (BIO2/BIO7) (* 100)
bio4	Temperature Seasonality (standard deviation *100)
bio5	Max Temperature of Warmest Month
bio6	Min Temperature of Coldest Month
bio7	Temperature Annual Range (BIO5-BIO6)
bio8	Mean Temperature of Wettest Quarter
bio9	Mean Temperature of Driest Quarter
bio10	Mean Temperature of Warmest Quarter
bio11	Mean Temperature of Coldest Quarter
bio12	Annual Precipitation
bio13	Precipitation of Wettest Month
bio14	Precipitation of Driest Month
bio15	Precipitation Seasonality (Coefficient of Variation)
bio16	Precipitation of Wettest Quarter
bio17	Precipitation of Driest Quarter
bio18	Precipitation of Warmest Quarter
bio19	Precipitation of Coldest Quarter

Table 5.1: Table of all variables

Expertly Removed Variables			
$peakt	$q200	$class	$ntp
$q2	$q500	$season	$nrt
$q5	$alpha	$maxseason	$nq
$q10	$beta	$P0	$peakq

Table 5.2 continued from previous page

Expertly Removed Variables			
$q25	$usgs_area	$P1	$f
$q50	$carea	$P2	$mf
$q100	$dc	$nfd	$f.ecdf

Table 5.2: Expertly Removed Variables

5.2 Model results and outputs

```
Call: earth(x=data.frame[16914,52], y=c(-0.403,0.396,...), keepxy=TRUE,
            degree=2, nprune=39)

                                                     coefficients
(Intercept)                                            -0.2146015
h(1.59977-precip)                                      -0.1391982
h(-0.291407-imperviousbasin)                            0.1327861
h(0.700318-mf.event)                                    0.4103090
h(mf.event-0.700318)                                   -0.7494138
h(0.688744-precip_sdev)                                 0.2991444
h(precip_sdev-0.688744)                                -0.0441179
h(0.543984-flowdist_mean)                               0.2619192
h(flowdist_mean-0.543984)                              -0.4650195
h(1.77731-flowdist_sdev)                               -0.0358267
h(flowdist_sdev-1.77731)                                0.5161853
h(0.0922843-prod_mean)                                 -1.6017554
h(prod_mean-0.0922843)                                  1.5408134
h(0.262394-prod_sdev)                                   0.6274666
h(prod_sdev-0.262394)                                  -0.4811582
h(2.63582-prod_skew)                                   -0.0650089
h(prod_skew-2.63582)                                    0.3841768
h(-0.442446-rr) * h(0.543984-flowdist_mean)            -0.0939906
h(rr- -0.442446) * h(0.543984-flowdist_mean)           -0.0885677
h(imperviousbasin- -0.291407) * h(bio_15- -2.39842)    -0.0251479
h(imperviousbasin- -0.291407) * h(-2.39842-bio_15)     -4.1795172
h(0.0094663-kfact) * h(prod_sdev-0.262394)             -0.1064142
h(kfact-0.0094663) * h(prod_sdev-0.262394)             -0.1028072
h(0.700318-mf.event) * h(bio_2-0.917908)               -0.0828962
h(0.700318-mf.event) * h(0.917908-bio_2)               -0.0796223
h(0.700318-mf.event) * h(bio_18-1.83243)               -0.0749972
h(0.700318-mf.event) * h(1.83243-bio_18)                0.0231948
h(-0.564008-precip_mean) * h(0.688744-precip_sdev)      0.3973578
h(0.503667-precip_mean) * h(prod_mean-0.0922843)       -1.0087215
h(precip_mean-0.503667) * h(prod_mean-0.0922843)       -0.1389018
h(0.688744-precip_sdev) * h(-1.45447-bio_3)             0.6394830
h(-1.17561-flowdist_skew) * h(2.63582-prod_skew)        0.0424978
h(0.0922843-prod_mean) * h(snowpercent-2.33135)         0.7048619
h(0.0922843-prod_mean) * h(2.33135-snowpercent)         0.0809701

Selected 34 of 39 terms, and 18 of 52 predictors
Termination condition: RSq changed by less than 0.001 at 39 terms
Importance: prod_mean, mf.event, precip, flowdist_mean, precip_sdev, ...
Number of terms at each degree of interaction: 1 16 17
GCV 0.5742863    RSS 9617.81    GRSq 0.4275755    RSq 0.4331463
```

Listing 5.1: MARS Best Fit - lag time

```
Call: earth(x=data.frame[16914,52], y=c(0.0818,-0.127...), keepxy=TRUE,
degree=5, nprune=54)
coefficients
(Intercept)                                                                                                              0.4093
h(si-0.879128)                                                                                                          -0.6231
h(imperviouscell-1.45799)                                                                                                0.8206
h(0.972409-precip_sdev)                                                                                                  0.2339
h(snowpercent-2.30136)                                                                                                   1.2845
h(0.471711-bio_10)                                                                                                      -0.3664
h(bio_10-0.471711)                                                                                                      -0.1736
h(si-0.879128) * h(slopeoutlet-1.0149)                                                                                   0.3237
h(si-0.879128) * h(1.0149-slopeoutlet)                                                                                   0.5176
h(0.879128-si) * h(-0.964212-bio_2)                                                                                      0.4699
h(si-0.559212) * h(bio_10-0.471711)                                                                                     -1.2451
h(0.879128-si) * h(bio_15- -1.99507)                                                                                     0.0258
h(0.879128-si) * h(-1.99507-bio_15)                                                                                     -3.4713
h(slopeoutlet-0.540864) * h(1.45799-imperviouscell)                                                                     -0.1402
h(0.876816-kfact) * h(2.30136-snowpercent)                                                                              -0.0673
h(kfact-0.876816) * h(2.30136-snowpercent)                                                                              -0.0834
h(-0.263271-ruggedness) * h(0.471711-bio_10)                                                                            -0.4047
h(ruggedness- -0.263271) * h(0.471711-bio_10)                                                                            0.0764
h(mf.event- -0.382882) * h(0.972409-precip_sdev)                                                                        -0.1243
h(0.972409-precip_sdev) * h(-2.11377-bio_8)                                                                             -3.4991
h(-0.719769-bio_3) * h(-0.838173-bio_10)                                                                                 0.8678
h(-0.719769-bio_3) * h(bio_18-0.489383)                                                                                 -3.7545
h(-0.719769-bio_3) * h(0.489383-bio_18)                                                                                 -0.4173
h(rr-1.1255) * h(si-0.879128) * h(imperviouscell-0.855871)                                                              -1.6205
h(rr-1.52852) * h(0.879128-si) * h(bio_2- -0.964212)                                                                   -14.3362
h(0.879128-si) * h(slopeoutlet- -1.44706) * h(2.05235-ruggedness)                                                       -0.0428
h(0.879128-si) * h(-1.44706-slopeoutlet) * h(2.05235-ruggedness)                                                        -0.0364
h(0.879128-si) * h(imperviouscell-1.6277) * h(bio_2- -0.964212)                                                         -9.9579
h(si-0.559212) * h(imperviouscell-1.45799) * h(bio_10-0.471711)                                                         16.3127
h(0.4259-si) * h(0.876816-kfact) * h(2.30136-snowpercent)                                                                0.0219
h(si-0.4259) * h(0.876816-kfact) * h(2.30136-snowpercent)                                                                0.0610
h(0.879128-si) * h(bio_2- -0.643945) * h(bio_15- -1.99507)                                                              -0.0362
h(0.879128-si) * h(-0.643945-bio_2) * h(bio_15- -1.99507)                                                               -0.1924
h(slopeoutlet-0.540864) * h(1.45799-imperviouscell) * h(ruggedness-1.86612)                                              0.2795
h(slopeoutlet- -0.602161) * h(-0.263271-ruggedness) * h(0.471711-bio_10)                                                 0.7753
h(cncell- -1.20704) * h(-0.719769-bio_3) * h(bio_8-0.749662)                                                            -6.0882
h(kfact-0.876816) * h(2.30136-snowpercent) * h(0.866336-bio_3)                                                           0.1160
h(ruggedness- -0.263271) * h(mf.event-0.629614) * h(0.471711-bio_10)                                                     0.1258
h(mf.event- -0.382882) * h(precip_mean-0.97798) * h(0.972409-precip_sdev)                                                0.4539
h(si-0.559212) * h(1.45799-imperviouscell) * h(bio_1-1.48759) * h(bio_10-0.471711)                                       1.2493
h(0.4259-si) * h(0.876816-kfact) * h(2.30136-snowpercent) * h(-1.48289-bio_15)                                           0.4477
h(0.879128-si) * h(lbm-1.01123) * h(bio_2- -0.643945) * h(bio_15- -1.99507)                                             -0.4188
h(slopeoutlet-0.540864) * h(1.45799-imperviouscell) * h(-0.712177-rockdepth) * h(1.86612-ruggedness)                     0.1645
h(slopeoutlet-0.540864) * h(1.45799-imperviouscell) * h(bio_1- -0.554413) * h(-0.370714-bio_8)                          -0.2196
h(slopeoutlet- -0.602161) * h(-0.263271-ruggedness) * h(G2-1.37429) * h(0.471711-bio_10)                                24.4934
h(cncell- -1.20704) * h(-0.719769-bio_3) * h(bio_8-0.749662) * h(bio_12- -0.40832)                                   -9641.1416
h(kfact-0.876816) * h(-0.979421-mf.event) * h(2.30136-snowpercent) * h(0.866336-bio_3)                                  -1.4561
h(0.879128-si) * h(lbm-1.01123) * h(snowpercent- -0.196804) * h(bio_2- -0.643945) * h(bio_15- -1.99507)                  1.1140
h(0.879128-si) * h(lbm-1.01123) * h(-0.196804-snowpercent) * h(bio_2- -0.643945) * h(bio_15- -1.99507)                   0.2653
h(slopeoutlet-0.540864) * h(cnbasin-0.701198) * h(1.45799-imperviouscell) * h(rockdepth- -0.712177) * h(1.86612-ruggedness)     0.6945
h(slopeoutlet-0.540864) * h(0.701198-cnbasin) * h(1.45799-imperviouscell) * h(rockdepth- -0.712177) * h(1.86612-ruggedness)     0.1111
h(slopeoutlet-0.540864) * h(coemcell-0.507571) * h(1.45799-imperviouscell) * h(rockdepth- -0.712177) * h(1.86612-ruggedness)   -2.4639
h(slopeoutlet-0.540864) * h(1.45799-imperviouscell) * h(bio_1- -0.554413) * h(bio_7-0.497885) * h(-0.370714-bio_8)      -123.6701
h(slopeoutlet- -0.602161) * h(-0.263271-ruggedness) * h(G1-0.297868) * h(1.37429-G2) * h(0.471711-bio_10)                 -5.2698

Selected 54 of 104 terms, and 26 of 52 predictors
Termination condition: Reached nk 105
Importance: bio_10, ruggedness, slopeoutlet, imperviouscell, ...
Number of terms at each degree of interaction: 1 6 16 16 8 7
GCV 0.8599431    RSS 14316.38    GRSq 0.1391669    RSq 0.152602
```

Listing 5.2: MARS Best Fit - peakq_moment

```
Call: earth(x=data.frame[16914,52], y=factor.object, keepxy=TRUE, glm=list(family=function.object), degree=4, nprune=52)

Number of Terms:  52

Earth coefficients
0           1           2
(Intercept)                                                                                                          -0.01889439   -0.3289523   -0.3178187
h(-0.0295668-est_area)                                                                                                0.04206606    1.4293902    0.5807953
h(est_area- -0.0295668)                                                                                               0.00982552   -1.1746520   -0.1623361
h(mf.event- -0.881797)                                                                                                0.00848579    0.1456778    0.1357694
h(1.36709-mf.event)                                                                                                   0.00439856    0.7860767    0.2278130
h(bio_10-1.89634)                                                                                                     0.02860260    0.3647857    0.0147854
h(-0.0295668-est_area) * h(imperviouscell-1.45833)                                                                    0.00868986   -0.2567235   -0.2442683
h(-0.0295668-est_area) * h(rockdepth- -0.658523)                                                                      0.01473341    0.1046184   -0.0019797
h(-0.0295668-est_area) * h(-0.658523-rockdepth)                                                                       0.00985447    0.0497935   -0.0172209
h(-0.0295668-est_area) * h(mf.event- -0.841039)                                                                      -0.02222288   -0.6097675   -0.1807684
h(-0.0295668-est_area) * h(-0.841039-mf.event)                                                                       -0.09723022    2.1877683   -0.6710246
h(est_area- -0.0295668) * h(mf.event- -0.875855)                                                                     -0.00393567    0.5257515    0.0781979
h(est_area- -0.0295668) * h(-0.875855-mf.event)                                                                      -0.00577377   -0.6950907   -0.2455270
h(1.1894-est_area) * h(1.36709-mf.event)                                                                              0.00724689   -0.5278320   -0.0640533
h(est_area-1.1894) * h(1.36709-mf.event)                                                                             -0.00549740    0.5278783    0.0773333
h(-0.0295668-est_area) * h(prod_mean- -0.0563436)                                                                    -0.01287315   -0.0859060   -0.0629748
h(est_area- -0.0295668) * h(rl-0.119428) * h(-0.875855-mf.event)                                                      0.00244754    0.0455346    0.0525931
h(est_area- -0.0295668) * h(0.119428-rl) * h(-0.875855-mf.event)                                                      0.04277813   -2.6108713    5.1169982
h(-0.0295668-est_area) * h(imperviousbasin-2.32647) * h(imperviouscell-1.45833)                                      -0.03816319   -0.3704926    0.0917310
h(-0.0295668-est_area) * h(imperviousbasin-2.78995) * h(mf.event- -0.841039)                                          0.01553757    0.1126510   -0.0568983
h(-0.0295668-est_area) * h(1.45833-imperviouscell) * h(G2-0.0426209)                                                 -0.09630320    3.0640702   -0.1998077
h(-0.0295668-est_area) * h(1.45833-imperviouscell) * h(0.0426209-G2)                                                  0.00861713    0.0085905   -0.0124182
h(-0.0295668-est_area) * h(rockdepth- -0.658523) * h(totalBasinPixels- -2.37288)                                     -0.01289690   -0.0746898   -0.0177004
h(-0.0295668-est_area) * h(rockdepth- -0.658523) * h(-2.37288-totalBasinPixels)                                      -0.07166972    3.3818037   -1.4665285
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(snowpercent- -0.804155)                                          -0.00647698    0.0563937    0.0098326
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(-0.804155-snowpercent)                                           -0.01331149    0.1485824    0.0489924
h(1.1894-est_area) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels)                                               0.08070480   -0.0034784    0.1556097
h(-0.0295668-est_area) * h(mf.event- -0.841039) * h(G1- -2.01108)                                                     0.00563719   -0.0000899   -0.0415395
h(-0.0295668-est_area) * h(mf.event- -0.841039) * h(-2.01108-G1)                                                     -0.00808890    0.1948620   -0.0851707
h(1.1894-est_area) * h(1.36709-mf.event) * h(bio_17- -0.971404)                                                      -0.00401160    0.0034044   -0.0074419
h(-0.0295668-est_area) * h(rl- -1.87538) * h(1.45833-imperviouscell) * h(0.0426209-G2)                               -0.01325597   -0.0512800   -0.0041663
h(-0.0295668-est_area) * h(-1.87538-rl) * h(1.45833-imperviouscell) * h(0.0426209-G2)                                -0.00652696   -0.1797040    0.0717488
h(1.1894-est_area) * h(si- -1.30303) * h(1.36709-mf.event) * h(-0.971404-bio_17)                                      0.06909241    0.0265249   -0.0049363
h(1.1894-est_area) * h(-1.30303-si) * h(1.36709-mf.event) * h(-0.971404-bio_17)                                       0.04887898   -0.0484546   -0.0526416
h(1.1894-est_area) * h(slopeoutlet-0.667119) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels)                    -0.06994020    0.0323312   -0.0425788
h(1.1894-est_area) * h(0.667119-slopeoutlet) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels)                    -0.04835213    0.3040940   -0.1677995
h(1.1894-est_area) * h(slopeoutlet-0.800896) * h(1.36709-mf.event) * h(bio_17- -0.971404)                             0.00074944   -0.0036052    0.0253166
h(1.1894-est_area) * h(0.800896-slopeoutlet) * h(1.36709-mf.event) * h(bio_17- -0.971404)                             0.00056814   -0.0055044    0.0008786
h(1.1894-est_area) * h(slopeoutlet- -1.88766) * h(1.36709-mf.event) * h(-0.971404-bio_17)                             0.01735466   -0.0201474   -0.0321455
h(1.1894-est_area) * h(-0.1689-precip) * h(1.36709-mf.event) * h(totalBasinPixels- -1.02251)                         -0.00076632   -0.0073306    0.0049836
h(1.1894-est_area) * h(cnbasin- -1.16277) * h(1.36709-mf.event) * h(totalBasinPixels- -1.02251)                      -0.00067858   -0.0012146   -0.0027366
h(1.1894-est_area) * h(-1.16277-cnbasin) * h(1.36709-mf.event) * h(totalBasinPixels- -1.02251)                        0.00057814   -0.0084972    0.0419881
h(-0.0295668-est_area) * h(imperviousbasin-2.78995) * h(mf.event- -0.841039) * h(totalBasinPixels- -0.654315) -0.18757762  -18.3349073  -14.0922715
h(1.1894-est_area) * h(imperviousbasin-0.767002) * h(1.36709-mf.event) * h(-0.971404-bio_17)                         -0.23036964   -0.1141491    0.8757871
h(1.1894-est_area) * h(0.767002-imperviousbasin) * h(1.36709-mf.event) * h(-0.971404-bio_17)                         -0.12681097   -0.0323958    0.0607548
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(totalBasinPixels- -2.11809) * h(snowpercent- -0.804155)    -0.00705134   -0.0628761   -0.0143853
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(-2.11809-totalBasinPixels) * h(snowpercent- -0.804155)      0.72157961   -0.2282225   -0.3449676
h(1.1894-est_area) * h(ruggedness-2.05431) * h(1.36709-mf.event) * h(-0.971404-bio_17)                               -0.58027419    0.3280408    0.2250235
h(1.1894-est_area) * h(2.05431-ruggedness) * h(1.36709-mf.event) * h(-0.971404-bio_17)                                0.02642500    0.0317992   -0.0083922
h(1.1894-est_area) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels) * h(G1- -0.84282)                             1.28573660   -2.7049560   -0.9903670
h(1.1894-est_area) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels) * h(-0.84282-G1)                             -0.03265023   -0.0518211   -0.1336644
h(1.1894-est_area) * h(1.36709-mf.event) * h(-0.829714-bio_3) * h(bio_17- -0.971404)                                 -0.03483394    0.4537545    0.0818949

(continuation)
Earth coefficients
4         8
(Intercept)                                                                                                          -0.921986    2.587651
```

```
h(-0.0295668-est_area)                                                                                   1.669199  -3.721450
h(est_area- -0.0295668)                                                                                 -1.342634   2.669797
h(mf.event- -0.881797)                                                                                   0.378956  -0.668889
h(1.36709-mf.event)                                                                                      1.112345  -2.130633
h(bio_10-1.89634)                                                                                       -0.026745  -0.381429
h(-0.0295668-est_area) * h(imperviouscell-1.45833)                                                      -0.381562   0.873864
h(-0.0295668-est_area) * h(rockdepth- -0.658523)                                                        -0.040695  -0.076677
h(-0.0295668-est_area) * h(-0.658523-rockdepth)                                                         -0.115384   0.072957
h(-0.0295668-est_area) * h(mf.event- -0.841039)                                                         -0.549671   1.362429
h(-0.0295668-est_area) * h(-0.841039-mf.event)                                                          -1.224801  -0.194713
h(est_area- -0.0295668) * h(mf.event- -0.875855)                                                         0.637734  -1.237748
h(est_area- -0.0295668) * h(-0.875855-mf.event)                                                         -0.876957   1.823349
h(1.1894-est_area) * h(1.36709-mf.event)                                                                -0.506377   1.091015
h(est_area-1.1894) * h(1.36709-mf.event)                                                                 0.636742  -1.236456
h(-0.0295668-est_area) * h(prod_mean- -0.0563436)                                                       -0.074937   0.236691
h(est_area- -0.0295668) * h(rl-0.119428) * h(-0.875855-mf.event)                                         0.038294  -0.138869
h(est_area- -0.0295668) * h(0.119428-rl) * h(-0.875855-mf.event)                                         2.507178  -5.056083
h(-0.0295668-est_area) * h(imperviousbasin-2.32647) * h(imperviouscell-1.45833)                          1.968567  -1.651642
h(-0.0295668-est_area) * h(imperviousbasin-2.78995) * h(mf.event- -0.841039)                            -0.586520   0.515230
h(-0.0295668-est_area) * h(1.45833-imperviouscell) * h(G2-0.0426209)                                    -1.118469  -1.649490
h(-0.0295668-est_area) * h(1.45833-imperviouscell) * h(0.0426209-G2)                                    -0.080132   0.075343
h(-0.0295668-est_area) * h(rockdepth- -0.658523) * h(totalBasinPixels- -2.37288)                        -0.013478   0.118765
h(-0.0295668-est_area) * h(rockdepth- -0.658523) * h(-2.37288-totalBasinPixels)                         -5.187357   3.343751
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(snowpercent- -0.804155)                              0.093291  -0.153041
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(-0.804155-snowpercent)                               0.086506  -0.270769
h(1.1894-est_area) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels)                                  0.138700  -0.371536
h(-0.0295668-est_area) * h(mf.event- -0.841039) * h(G1- -2.01108)                                       -0.064729   0.100721
h(-0.0295668-est_area) * h(mf.event- -0.841039) * h(-2.01108-G1)                                        -0.345906   0.244303
h(1.1894-est_area) * h(1.36709-mf.event) * h(bio_17- -0.971404)                                         -0.025059   0.033108
h(-0.0295668-est_area) * h(rl- -1.87538) * h(1.45833-imperviouscell) * h(0.0426209-G2)                   0.123723  -0.055021
h(-0.0295668-est_area) * h(-1.87538-rl) * h(1.45833-imperviouscell) * h(0.0426209-G2)                    0.436386  -0.321904
h(1.1894-est_area) * h(si- -1.30303) * h(1.36709-mf.event) * h(-0.971404-bio_17)                        -0.062096  -0.028585
h(1.1894-est_area) * h(-1.30303-si) * h(1.36709-mf.event) * h(-0.971404-bio_17)                          0.280510  -0.228293
h(1.1894-est_area) * h(slopeoutlet-0.667119) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels)       -0.090765   0.170952
h(1.1894-est_area) * h(0.667119-slopeoutlet) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels)       -0.363738   0.275796
h(1.1894-est_area) * h(slopeoutlet-0.800896) * h(1.36709-mf.event) * h(bio_17- -0.971404)                0.013895  -0.036356
h(1.1894-est_area) * h(0.800896-slopeoutlet) * h(1.36709-mf.event) * h(bio_17- -0.971404)                0.013556  -0.009498
h(1.1894-est_area) * h(slopeoutlet- -1.88766) * h(1.36709-mf.event) * h(-0.971404-bio_17)                0.001197   0.033742
h(1.1894-est_area) * h(-0.1689-precip) * h(1.36709-mf.event) * h(totalBasinPixels- -1.02251)             0.045959  -0.042846
h(1.1894-est_area) * h(cnbasin- -1.16277) * h(1.36709-mf.event) * h(totalBasinPixels- -1.02251)         -0.009954   0.014584
h(1.1894-est_area) * h(-1.16277-cnbasin) * h(1.36709-mf.event) * h(totalBasinPixels- -1.02251)           0.076625  -0.110694
h(-0.0295668-est_area) * h(imperviousbasin-2.78995) * h(mf.event- -0.841039) * h(totalBasinPixels- -0.654315) 87.567353 -54.952596
h(1.1894-est_area) * h(imperviousbasin-0.767002) * h(1.36709-mf.event) * h(-0.971404-bio_17)             0.444837  -0.976105
h(1.1894-est_area) * h(0.767002-imperviousbasin) * h(1.36709-mf.event) * h(-0.971404-bio_17)             0.147321  -0.048869
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(totalBasinPixels- -2.11809) * h(snowpercent- -0.804155)   -0.084211   0.168524
h(-0.0295668-est_area) * h(-0.658523-rockdepth) * h(-2.11809-totalBasinPixels) * h(snowpercent- -0.804155)    -1.114200   0.965811
h(1.1894-est_area) * h(ruggedness-2.05431) * h(1.36709-mf.event) * h(-0.971404-bio_17)                  -0.677201   0.704411
h(1.1894-est_area) * h(2.05431-ruggedness) * h(1.36709-mf.event) * h(-0.971404-bio_17)                  -0.102778   0.052946
h(1.1894-est_area) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels) * h(G1- -0.84282)               -0.014336   2.423922
h(1.1894-est_area) * h(1.36709-mf.event) * h(-1.02251-totalBasinPixels) * h(-0.84282-G1)                -0.125198   0.343334
h(1.1894-est_area) * h(1.36709-mf.event) * h(-0.829714-bio_3) * h(bio_17- -0.971404)                    -0.463165  -0.037651
```

Listing 5.3: MARS Best Fit - exceeds_threshold

```
Support Vector Machines with Radial Basis Function Kernel

16914 samples
52 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 15223, 15224, 15224, 15222, 15222, 15223, ...
Resampling results across tuning parameters:

sigma      C          RMSE       Rsquared    MAE
0.0000000  0.0000000        NaN         NaN        NaN
0.0000000  0.5555556  1.0019820         NaN  0.8203102
0.0000000  1.1111111  1.0019820         NaN  0.8203102
0.0000000  1.6666667  1.0019820         NaN  0.8203102
0.0000000  2.2222222  1.0019820         NaN  0.8203102
0.0000000  2.7777778  1.0019820         NaN  0.8203102
0.0000000  3.3333333  1.0019820         NaN  0.8203102
0.0000000  3.8888889  1.0019820         NaN  0.8203102
0.0000000  4.4444444  1.0019820         NaN  0.8203102
0.0000000  5.0000000  1.0019820         NaN  0.8203102
0.5555556  0.0000000        NaN         NaN        NaN
0.5555556  0.5555556  0.9069105  0.20170954  0.7089658
0.5555556  1.1111111  0.8910200  0.21739448  0.6920289
0.5555556  1.6666667  0.8881179  0.21845658  0.6885907
0.5555556  2.2222222  0.8888401  0.21596645  0.6887640
0.5555556  2.7777778  0.8898763  0.21363715  0.6895134
0.5555556  3.3333333  0.8907036  0.21184671  0.6901420
0.5555556  3.8888889  0.8911733  0.21081937  0.6905901
0.5555556  4.4444444  0.8915411  0.21002139  0.6909359
0.5555556  5.0000000  0.8918132  0.20943263  0.6912014
1.1111111  0.0000000        NaN         NaN        NaN
1.1111111  0.5555556  0.9465507  0.13384277  0.7522765
1.1111111  1.1111111  0.9327454  0.14799341  0.7371557
1.1111111  1.6666667  0.9293912  0.14965589  0.7333388
1.1111111  2.2222222  0.9292803  0.14850389  0.7329020
1.1111111  2.7777778  0.9296879  0.14718329  0.7331136
1.1111111  3.3333333  0.9301097  0.14593516  0.7333451
1.1111111  3.8888889  0.9302298  0.14556080  0.7334772
1.1111111  4.4444444  0.9303316  0.14524826  0.7335959
1.1111111  5.0000000  0.9304163  0.14499746  0.7336814
1.6666667  0.0000000        NaN         NaN        NaN
1.6666667  0.5555556  0.9639936  0.10063307  0.7727052
1.6666667  1.1111111  0.9525738  0.11129210  0.7595715
1.6666667  1.6666667  0.9496357  0.11227707  0.7561124
1.6666667  2.2222222  0.9495232  0.11106998  0.7556410
1.6666667  2.7777778  0.9498106  0.11000545  0.7557640
1.6666667  3.3333333  0.9500606  0.10917012  0.7558762
1.6666667  3.8888889  0.9500925  0.10904334  0.7559108
1.6666667  4.4444444  0.9501209  0.10894015  0.7559399
1.6666667  5.0000000  0.9501397  0.10886632  0.7559646
2.2222222  0.0000000        NaN         NaN        NaN
2.2222222  0.5555556  0.9734253  0.08012466  0.7841209
2.2222222  1.1111111  0.9639010  0.08764243  0.7727083
2.2222222  1.6666667  0.9614090  0.08814069  0.7696350
2.2222222  2.2222222  0.9613030  0.08699285  0.7691570
2.2222222  2.7777778  0.9615110  0.08620073  0.7691966
2.2222222  3.3333333  0.9617029  0.08556148  0.7692554
2.2222222  3.8888889  0.9617114  0.08551921  0.7692704
2.2222222  4.4444444  0.9617152  0.08549820  0.7692769
2.2222222  5.0000000  0.9617145  0.08549787  0.7692759
2.7777778  0.0000000        NaN         NaN        NaN
2.7777778  0.5555556  0.9790510  0.06619722  0.7910229
2.7777778  1.1111111  0.9709040  0.07164672  0.7809426
2.7777778  1.6666667  0.9688027  0.07180034  0.7781813
2.7777778  2.2222222  0.9686979  0.07081045  0.7776854
```

```
  2.7777778  2.7777778  0.9688812  0.07014041  0.7777009
  2.7777778  3.3333333  0.9690588  0.06960066  0.7777475
  2.7777778  3.8888889  0.9690591  0.06959695  0.7777477
  2.7777778  4.4444444  0.9690591  0.06959691  0.7777475
  2.7777778  5.0000000  0.9690592  0.06959679  0.7777474
  3.3333333  0.0000000        NaN         NaN        NaN
  3.3333333  0.5555556  0.9826492  0.05638909  0.7954569
  3.3333333  1.1111111  0.9754990  0.06054270  0.7863526
  3.3333333  1.6666667  0.9736847  0.06054990  0.7838365
  3.3333333  2.2222222  0.9735889  0.05970640  0.7833381
  3.3333333  2.7777778  0.9737650  0.05911600  0.7833412
  3.3333333  3.3333333  0.9739425  0.05863749  0.7833884
  3.3333333  3.8888889  0.9739425  0.05863742  0.7833880
  3.3333333  4.4444444  0.9739425  0.05863740  0.7833878
  3.3333333  5.0000000  0.9739425  0.05863735  0.7833876
  3.8888889  0.0000000        NaN         NaN        NaN
  3.8888889  0.5555556  0.9850595  0.04934814  0.7984523
  3.8888889  1.1111111  0.9786421  0.05269541  0.7900797
  3.8888889  1.6666667  0.9770332  0.05268721  0.7877127
  3.8888889  2.2222222  0.9769526  0.05195633  0.7872200
  3.8888889  2.7777778  0.9771357  0.05140484  0.7872306
  3.8888889  3.3333333  0.9773131  0.05096689  0.7872787
  3.8888889  3.8888889  0.9773131  0.05096677  0.7872784
  3.8888889  4.4444444  0.9773131  0.05096672  0.7872782
  3.8888889  5.0000000  0.9773132  0.05096667  0.7872780
  4.4444444  0.0000000        NaN         NaN        NaN
  4.4444444  0.5555556  0.9867407  0.04420155  0.8005941
  4.4444444  1.1111111  0.9808704  0.04703744  0.7927424
  4.4444444  1.6666667  0.9794155  0.04706076  0.7904672
  4.4444444  2.2222222  0.9793495  0.04642191  0.7899779
  4.4444444  2.7777778  0.9795390  0.04589944  0.7899962
  4.4444444  3.3333333  0.9797154  0.04549437  0.7900446
  4.4444444  3.8888889  0.9797154  0.04549429  0.7900443
  4.4444444  4.4444444  0.9797154  0.04549423  0.7900441
  4.4444444  5.0000000  0.9797154  0.04549417  0.7900439
  5.0000000  0.0000000        NaN         NaN        NaN
  5.0000000  0.5555556  0.9879571  0.04036538  0.8021905
  5.0000000  1.1111111  0.9824997  0.04286573  0.7946900
  5.0000000  1.6666667  0.9811587  0.04294924  0.7924973
  5.0000000  2.2222222  0.9811062  0.04238221  0.7920100
  5.0000000  2.7777778  0.9813000  0.04188335  0.7920338
  5.0000000  3.3333333  0.9814756  0.04150415  0.7920819
  5.0000000  3.8888889  0.9814756  0.04150407  0.7920816
  5.0000000  4.4444444  0.9814756  0.04150403  0.7920814
  5.0000000  5.0000000  0.9814756  0.04150400  0.7920812

  RMSE was used to select the optimal model using the smallest value.
  The final values used for the model were sigma = 0.5555556 and C = 1.666667.
```

Listing 5.4: SVM Best Fit - lag time

```
Support Vector Machines with Radial Basis Function Kernel

16914 samples
52 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 15222, 15222, 15224, 15222, 15224, 15222, ...
Resampling results across tuning parameters:

sigma      C          RMSE       Rsquared    MAE
0.0000000  0.0000000        NaN        NaN        NaN
0.0000000  0.5555556  0.9996931        NaN  0.7878128
0.0000000  1.1111111  0.9996931        NaN  0.7878128
0.0000000  1.6666667  0.9996931        NaN  0.7878128
0.0000000  2.2222222  0.9996931        NaN  0.7878128
```

```
0.0000000  2.7777778  0.9996931         NaN  0.7878128
0.0000000  3.3333333  0.9996931         NaN  0.7878128
0.0000000  3.8888889  0.9996931         NaN  0.7878128
0.0000000  4.4444444  0.9996931         NaN  0.7878128
0.0000000  5.0000000  0.9996931         NaN  0.7878128
0.5555556  0.0000000        NaN         NaN        NaN
0.5555556  0.5555556  0.9546673  0.08889842  0.7381761
0.5555556  1.1111111  0.9547582  0.08975652  0.7360130
0.5555556  1.6666667  0.9590769  0.08726597  0.7384933
0.5555556  2.2222222  0.9636342  0.08361430  0.7419653
0.5555556  2.7777778  0.9671220  0.08073861  0.7446485
0.5555556  3.3333333  0.9695016  0.07878516  0.7465794
0.5555556  3.8888889  0.9713087  0.07730301  0.7480234
0.5555556  4.4444444  0.9726840  0.07619749  0.7491306
0.5555556  5.0000000  0.9739173  0.07519939  0.7501012
1.1111111  0.0000000        NaN         NaN        NaN
1.1111111  0.5555556  0.9687017  0.06531231  0.7539174
1.1111111  1.1111111  0.9649787  0.06808458  0.7484700
1.1111111  1.6666667  0.9663706  0.06643798  0.7486826
1.1111111  2.2222222  0.9681664  0.06433412  0.7499722
1.1111111  2.7777778  0.9694721  0.06270459  0.7509825
1.1111111  3.3333333  0.9704042  0.06149992  0.7517690
1.1111111  3.8888889  0.9711958  0.06048239  0.7524128
1.1111111  4.4444444  0.9718580  0.05964648  0.7529646
1.1111111  5.0000000  0.9723960  0.05898539  0.7534025
1.6666667  0.0000000        NaN         NaN        NaN
1.6666667  0.5555556  0.9762777  0.05219004  0.7620276
1.6666667  1.1111111  0.9718086  0.05530265  0.7562807
1.6666667  1.6666667  0.9719265  0.05466063  0.7554996
1.6666667  2.2222222  0.9729272  0.05306291  0.7560434
1.6666667  2.7777778  0.9737701  0.05172058  0.7566023
1.6666667  3.3333333  0.9743277  0.05085082  0.7570460
1.6666667  3.8888889  0.9747489  0.05019710  0.7573869
1.6666667  4.4444444  0.9750609  0.04971878  0.7576470
1.6666667  5.0000000  0.9752789  0.04938857  0.7578204
2.2222222  0.0000000        NaN         NaN        NaN
2.2222222  0.5555556  0.9807067  0.04418632  0.7667156
2.2222222  1.1111111  0.9761966  0.04715300  0.7612304
2.2222222  1.6666667  0.9758957  0.04683532  0.7601901
2.2222222  2.2222222  0.9765173  0.04562890  0.7603925
2.2222222  2.7777778  0.9770321  0.04469763  0.7606501
2.2222222  3.3333333  0.9773329  0.04416744  0.7608863
2.2222222  3.8888889  0.9775136  0.04385598  0.7610223
2.2222222  4.4444444  0.9776261  0.04366170  0.7611123
2.2222222  5.0000000  0.9777085  0.04352241  0.7611807
2.7777778  0.0000000        NaN         NaN        NaN
2.7777778  0.5555556  0.9834920  0.03898680  0.7696787
2.7777778  1.1111111  0.9791649  0.04153570  0.7643987
2.7777778  1.6666667  0.9786091  0.04154264  0.7631445
2.7777778  2.2222222  0.9790206  0.04060957  0.7631348
2.7777778  2.7777778  0.9793207  0.04000679  0.7632387
2.7777778  3.3333333  0.9794763  0.03971695  0.7633581
2.7777778  3.8888889  0.9795587  0.03956539  0.7634388
2.7777778  4.4444444  0.9796292  0.03943827  0.7634987
2.7777778  5.0000000  0.9796918  0.03932685  0.7635531
3.3333333  0.0000000        NaN         NaN        NaN
3.3333333  0.5555556  0.9853527  0.03540761  0.7716609
3.3333333  1.1111111  0.9811620  0.03771318  0.7665391
3.3333333  1.6666667  0.9805078  0.03783193  0.7651982
3.3333333  2.2222222  0.9807514  0.03717056  0.7650557
3.3333333  2.7777778  0.9809314  0.03677264  0.7650658
3.3333333  3.3333333  0.9810132  0.03661416  0.7651409
3.3333333  3.8888889  0.9810758  0.03649567  0.7652015
3.3333333  4.4444444  0.9811155  0.03642140  0.7652391
3.3333333  5.0000000  0.9811420  0.03637092  0.7652657
3.8888889  0.0000000        NaN         NaN        NaN
3.8888889  0.5555556  0.9866488  0.03284584  0.7730312
3.8888889  1.1111111  0.9825741  0.03498439  0.7680488
```

```
3.8888889  1.6666667  0.9818580  0.03518321  0.7666670
3.8888889  2.2222222  0.9819769  0.03474361  0.7664269
3.8888889  2.7777778  0.9820926  0.03445915  0.7663926
3.8888889  3.3333333  0.9821446  0.03435613  0.7664390
3.8888889  3.8888889  0.9821770  0.03429339  0.7664730
3.8888889  4.4444444  0.9821974  0.03425216  0.7664952
3.8888889  5.0000000  0.9822076  0.03423002  0.7665099
4.4444444  0.0000000        NaN         NaN        NaN
4.4444444  0.5555556  0.9875870  0.03092437  0.7740287
4.4444444  1.1111111  0.9836090  0.03296000  0.7691565
4.4444444  1.6666667  0.9828445  0.03324370  0.7677617
4.4444444  2.2222222  0.9828795  0.03295823  0.7674664
4.4444444  2.7777778  0.9829499  0.03275672  0.7673993
4.4444444  3.3333333  0.9829841  0.03268857  0.7674280
4.4444444  3.8888889  0.9830006  0.03265431  0.7674481
4.4444444  4.4444444  0.9830082  0.03263724  0.7674619
4.4444444  5.0000000  0.9830096  0.03263378  0.7674655
5.0000000  0.0000000        NaN         NaN        NaN
5.0000000  0.5555556  0.9882883  0.02941879  0.7747823
5.0000000  1.1111111  0.9843947  0.03140382  0.7700057
5.0000000  1.6666667  0.9835823  0.03178933  0.7685878
5.0000000  2.2222222  0.9835635  0.03160442  0.7682676
5.0000000  2.7777778  0.9836039  0.03146042  0.7681748
5.0000000  3.3333333  0.9836237  0.03141974  0.7681880
5.0000000  3.8888889  0.9836306  0.03140409  0.7682005
5.0000000  4.4444444  0.9836314  0.03140184  0.7682038
5.0000000  5.0000000  0.9836314  0.03140184  0.7682038

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were sigma = 0.5555556 and C
= 0.5555556.
```

Listing 5.5: SVM Best Fit - peakq_moment

```
Support Vector Machines with Radial Basis Function Kernel

16914 samples
   52 predictor
    5 classes: ’0’, ’1’, ’2’, ’4’, ’8’

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 15223, 15222, 15222, 15224, 15223, 15223, ...
Resampling results across tuning parameters:

  sigma      C          Accuracy   Kappa
  0.0000000  0.0000000        NaN          NaN
  0.0000000  0.5555556  0.8006977  0.000000000
  0.0000000  1.1111111  0.8006977  0.000000000
  0.0000000  1.6666667  0.8006977  0.000000000
  0.0000000  2.2222222  0.8006977  0.000000000
  0.0000000  2.7777778  0.8006977  0.000000000
  0.0000000  3.3333333  0.8006977  0.000000000
  0.0000000  3.8888889  0.8006977  0.000000000
  0.0000000  4.4444444  0.8006977  0.000000000
  0.0000000  5.0000000  0.8006977  0.000000000
  0.5555556  0.0000000        NaN          NaN
  0.5555556  0.5555556  0.8156498  0.177197309
  0.5555556  1.1111111  0.8246184  0.281215198
  0.5555556  1.6666667  0.8271254  0.323088856
  0.5555556  2.2222222  0.8267708  0.330661634
  0.5555556  2.7777778  0.8256533  0.328785786
  0.5555556  3.3333333  0.8253636  0.328923874
  0.5555556  3.8888889  0.8251980  0.329360306
  0.5555556  4.4444444  0.8248018  0.328384143
  0.5555556  5.0000000  0.8246600  0.328517604
  1.1111111  0.0000000        NaN          NaN
```

```
1.1111111  0.5555556  0.8072781  0.080977036
1.1111111  1.1111111  0.8138170  0.164954911
1.1111111  1.6666667  0.8156853  0.204881403
1.1111111  2.2222222  0.8154606  0.212904338
1.1111111  2.7777778  0.8153246  0.214013336
1.1111111  3.3333333  0.8156084  0.216269661
1.1111111  3.8888889  0.8154133  0.215633819
1.1111111  4.4444444  0.8153778  0.215721790
1.1111111  5.0000000  0.8154133  0.216166382
1.6666667  0.0000000        NaN          NaN
1.6666667  0.5555556  0.8039022  0.041991006
1.6666667  1.1111111  0.8094241  0.110361094
1.6666667  1.6666667  0.8108136  0.141018536
1.6666667  2.2222222  0.8111742  0.148925970
1.6666667  2.7777778  0.8108786  0.148782490
1.6666667  3.3333333  0.8107544  0.148645058
1.6666667  3.8888889  0.8108254  0.149275500
1.6666667  4.4444444  0.8108550  0.149833794
1.6666667  5.0000000  0.8107722  0.149559594
2.2222222  0.0000000        NaN          NaN
2.2222222  0.5555556  0.8025837  0.024088026
2.2222222  1.1111111  0.8079756  0.084512234
2.2222222  1.6666667  0.8084427  0.108290962
2.2222222  2.2222222  0.8085550  0.113586618
2.2222222  2.7777778  0.8084249  0.113976153
2.2222222  3.3333333  0.8083836  0.113929468
2.2222222  3.8888889  0.8082298  0.113426993
2.2222222  4.4444444  0.8081944  0.113280184
2.2222222  5.0000000  0.8081766  0.113172692
2.7777778  0.0000000        NaN          NaN
2.7777778  0.5555556  0.8018506  0.014005483
2.7777778  1.1111111  0.8063615  0.065537857
2.7777778  1.6666667  0.8077154  0.089674564
2.7777778  2.2222222  0.8073193  0.091512966
2.7777778  2.7777778  0.8072957  0.091988335
2.7777778  3.3333333  0.8071656  0.091481363
2.7777778  3.8888889  0.8071715  0.091584206
2.7777778  4.4444444  0.8072602  0.092052738
2.7777778  5.0000000  0.8072484  0.091966729
3.3333333  0.0000000        NaN          NaN
3.3333333  0.5555556  0.8013481  0.007998257
3.3333333  1.1111111  0.8050726  0.051140205
3.3333333  1.6666667  0.8063201  0.071619624
3.3333333  2.2222222  0.8061250  0.074819429
3.3333333  2.7777778  0.8062255  0.075667678
3.3333333  3.3333333  0.8062373  0.076194549
3.3333333  3.8888889  0.8062373  0.076232602
3.3333333  4.4444444  0.8061959  0.076106770
3.3333333  5.0000000  0.8062018  0.076148576
3.8888889  0.0000000        NaN          NaN
3.8888889  0.5555556  0.8011116  0.004791918
3.8888889  1.1111111  0.8048007  0.044531051
3.8888889  1.6666667  0.8054393  0.059947210
3.8888889  2.2222222  0.8055220  0.063497916
3.8888889  2.7777778  0.8055929  0.064024719
3.8888889  3.3333333  0.8055811  0.063931818
3.8888889  3.8888889  0.8055575  0.063785417
3.8888889  4.4444444  0.8055634  0.063813911
3.8888889  5.0000000  0.8055634  0.063813911
4.4444444  0.0000000        NaN          NaN
4.4444444  0.5555556  0.8009401  0.003094190
4.4444444  1.1111111  0.8046292  0.039652012
4.4444444  1.6666667  0.8050254  0.051899570
4.4444444  2.2222222  0.8049781  0.054326753
4.4444444  2.7777778  0.8050017  0.054561548
4.4444444  3.3333333  0.8050077  0.054592083
4.4444444  3.8888889  0.8050077  0.054592083
4.4444444  4.4444444  0.8050077  0.054592083
```

```
  4.4444444  5.0000000  0.8050077  0.054590062
  5.0000000  0.0000000        NaN          NaN
  5.0000000  0.5555556  0.8008573  0.001949617
  5.0000000  1.1111111  0.8045050  0.036778279
  5.0000000  1.6666667  0.8047002  0.046133249
  5.0000000  2.2222222  0.8046943  0.048116338
  5.0000000  2.7777778  0.8046529  0.048211807
  5.0000000  3.3333333  0.8046529  0.048211807
  5.0000000  3.8888889  0.8046529  0.048211800
  5.0000000  4.4444444  0.8046529  0.048211800
  5.0000000  5.0000000  0.8046529  0.048211800

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.5555556 and C
 = 1.666667.
```

Listing 5.6: SVM Best Fit - exceeds_threshold

5.3 Training Reliability Diagrams

5.3.1 MARS

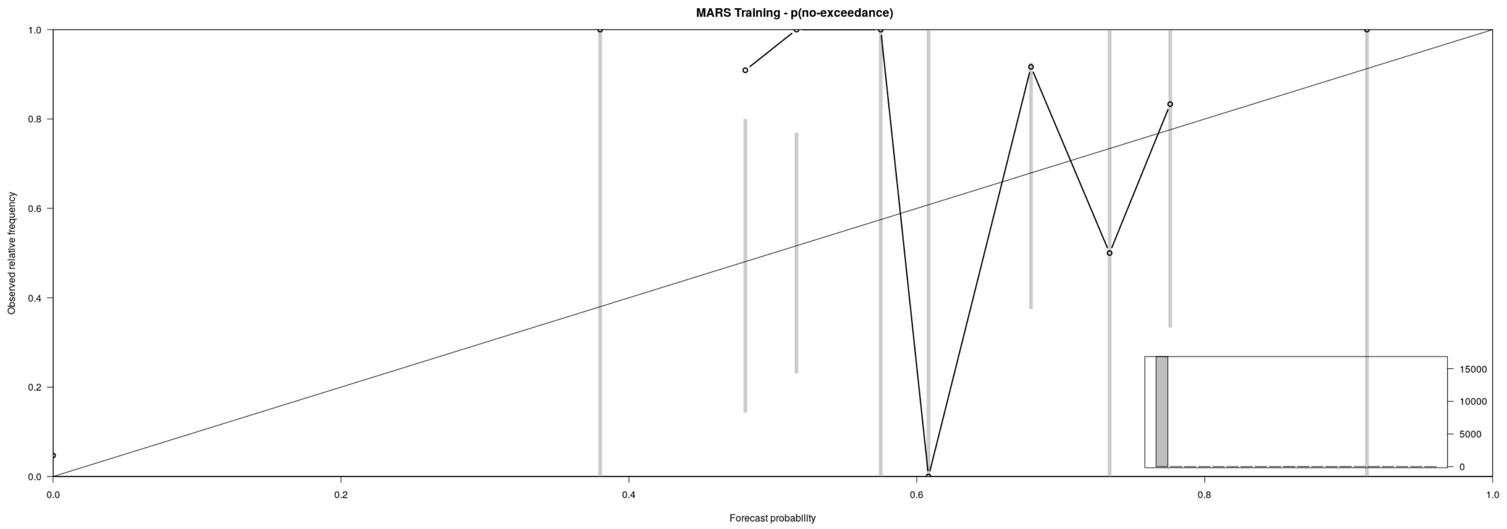

Figure 5.1: MARS Training: No-Exceedance Reliability Diagram

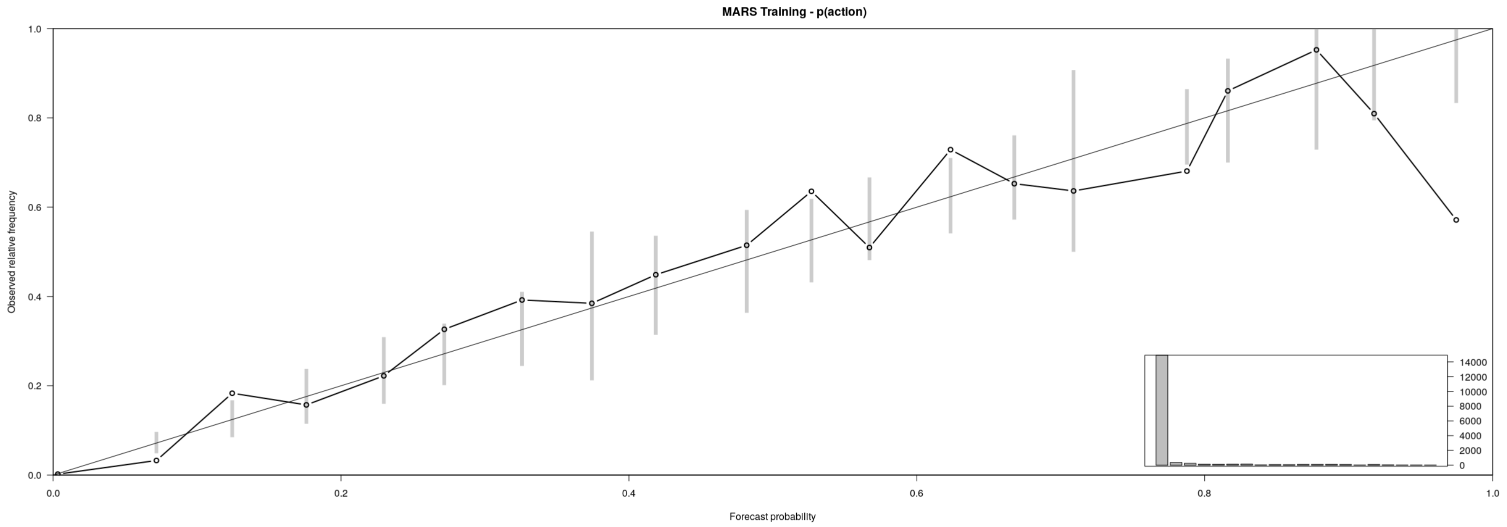

Figure 5.2: MARS Training: Exceeds Action Reliability Diagram

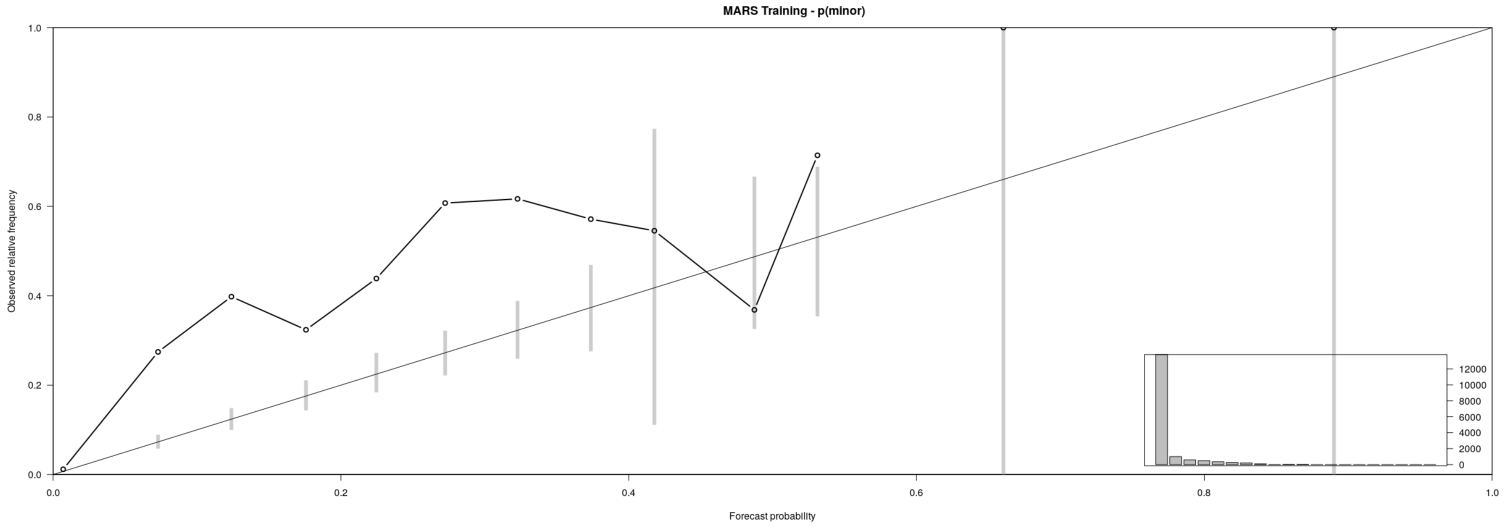

Figure 5.3: MARS Training: Exceeds Minor Reliability Diagram

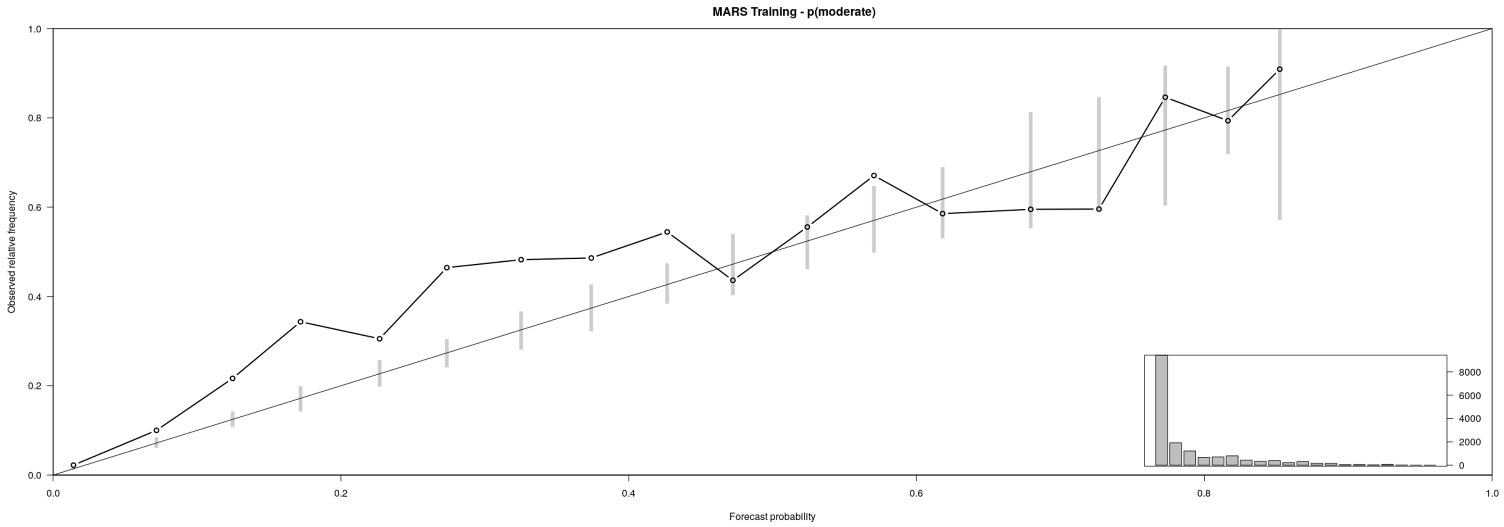

Figure 5.4: MARS Training: Exceeds Moderate Reliability Diagram

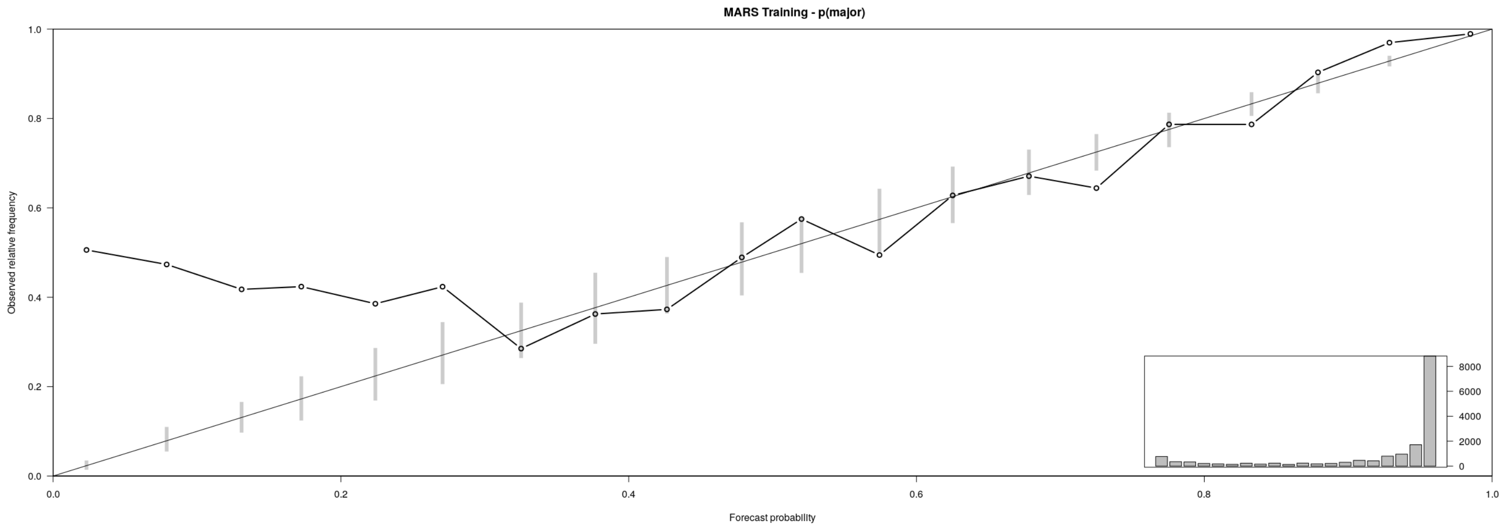

Figure 5.5: MARS Training: Exceeds Major Reliability Diagram

5.3.2 Random Forest

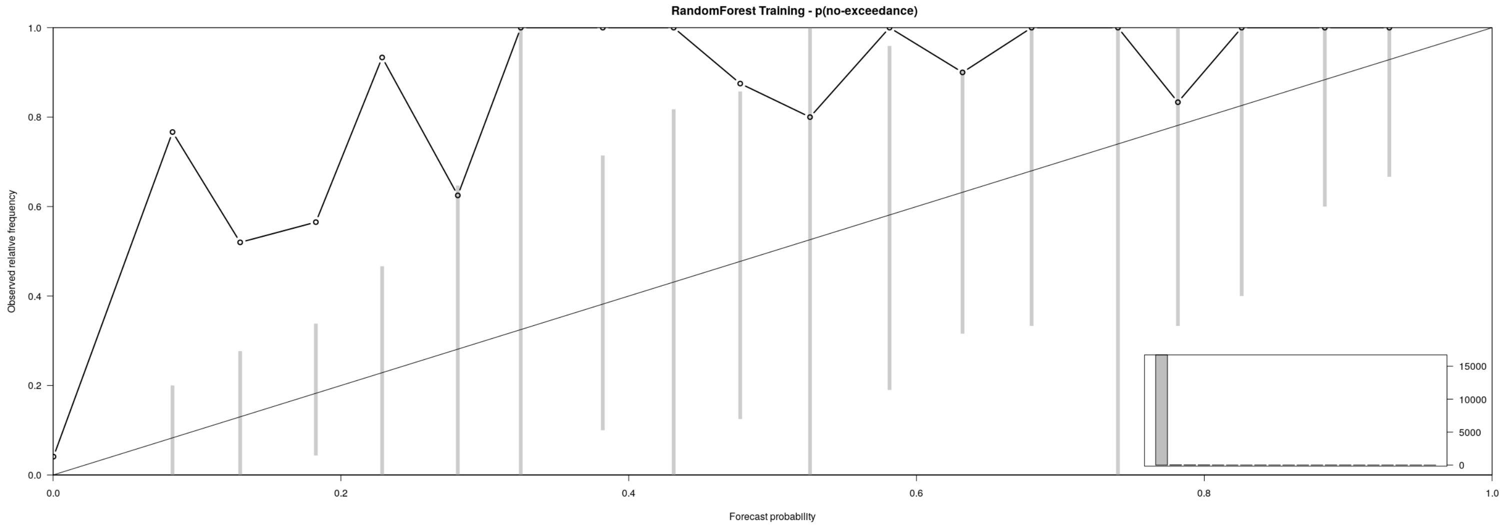

Figure 5.6: Random Forest Training: No-Exceedance Reliability Diagram

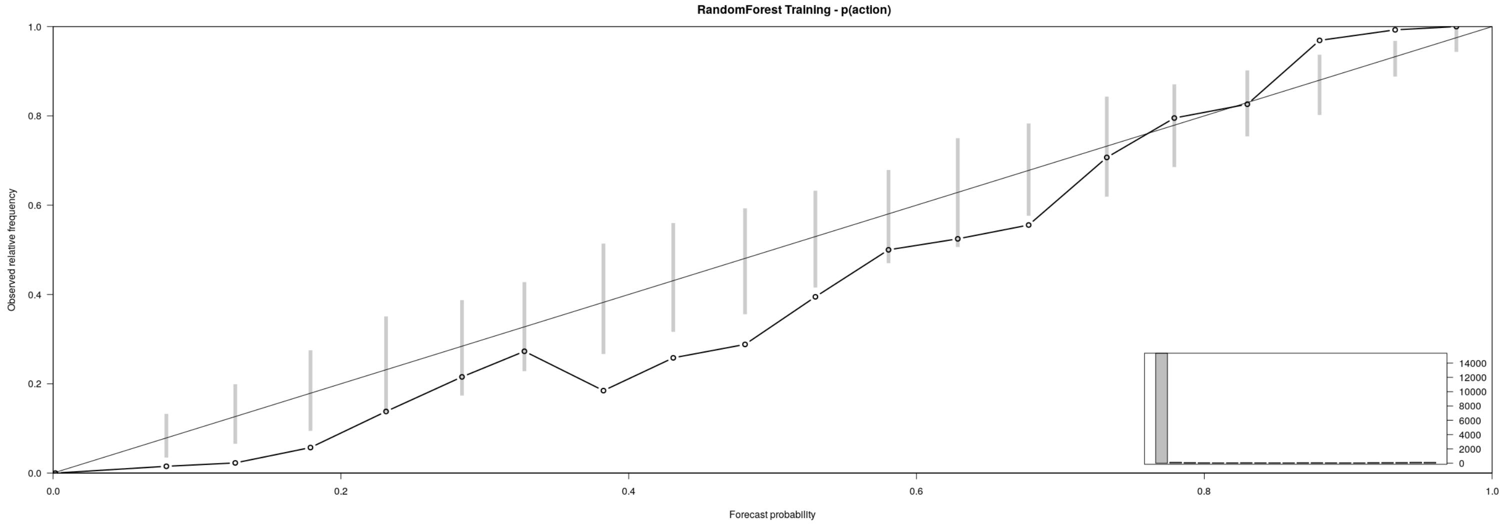

Figure 5.7: Random Forest Training: Exceeds Action Reliability Diagram

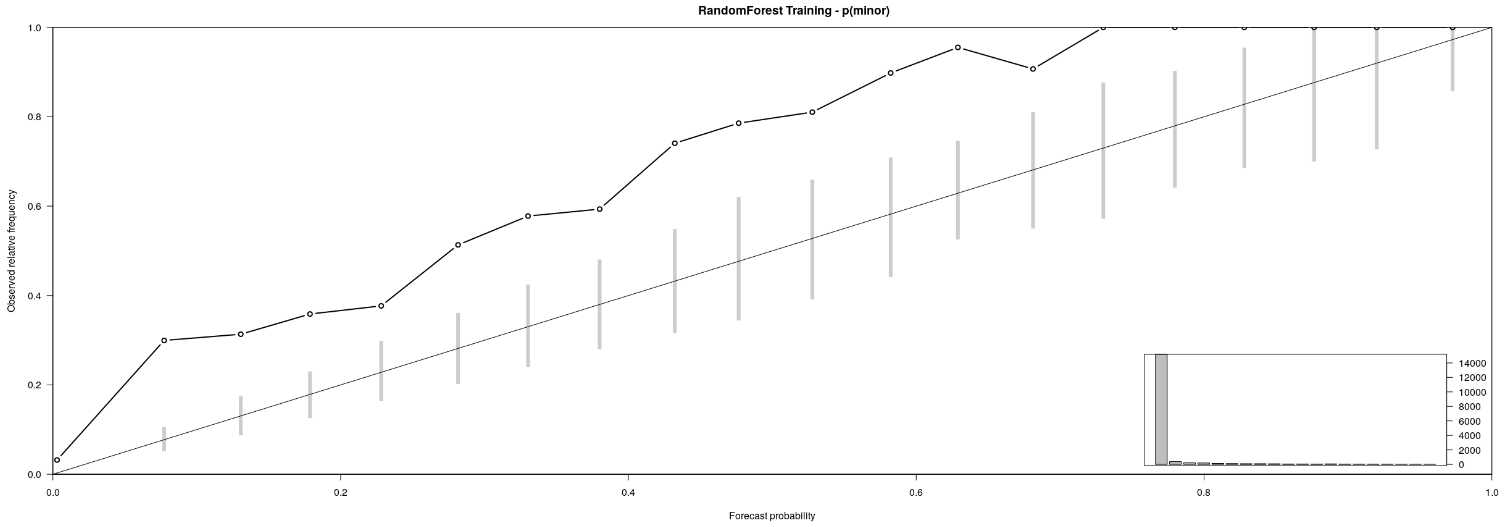

Figure 5.8: Random Forest Training: Exceeds Minor Reliability Diagram

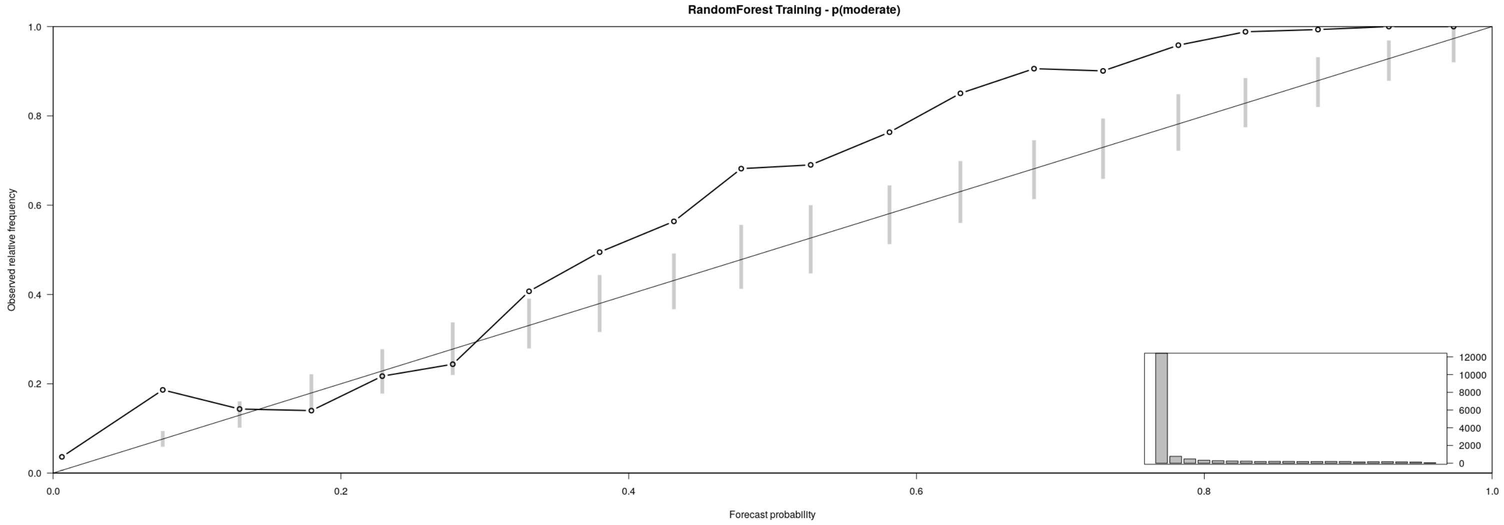

Figure 5.9: Random Forest Training: Exceeds Moderate Reliability Diagram

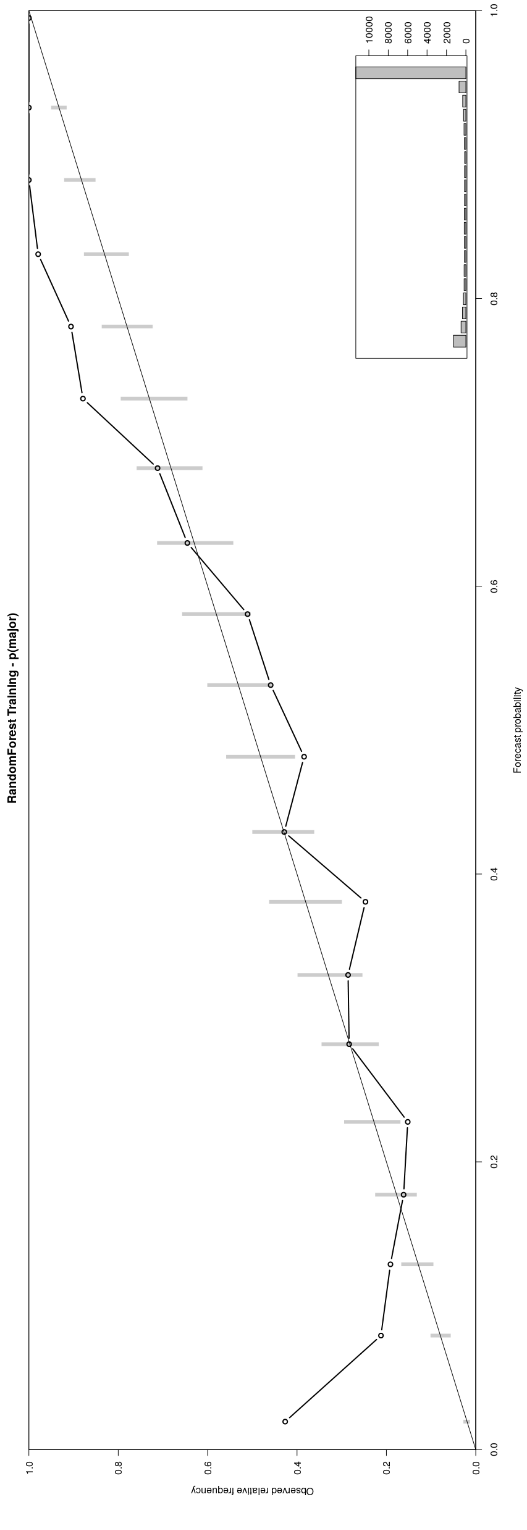

Figure 5.10: Random Forest Training: Exceeds Major Reliability Diagram

5.3.3 Support Vector Machines

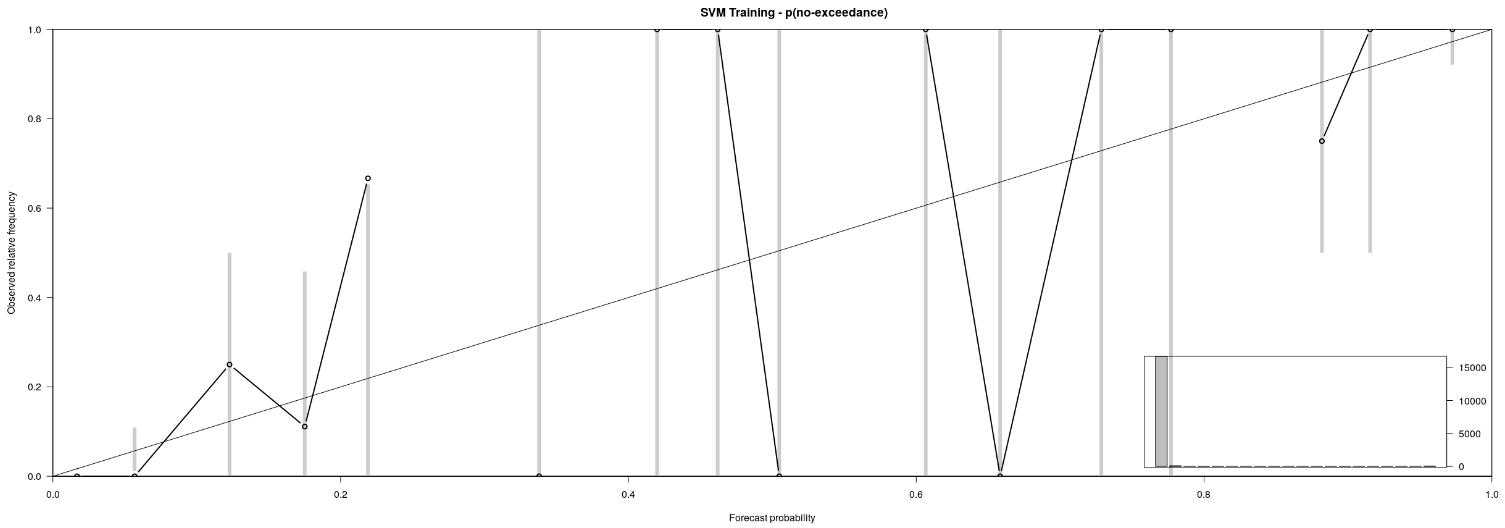

Figure 5.11: SVM Training: No-Exceedance Reliability Diagram

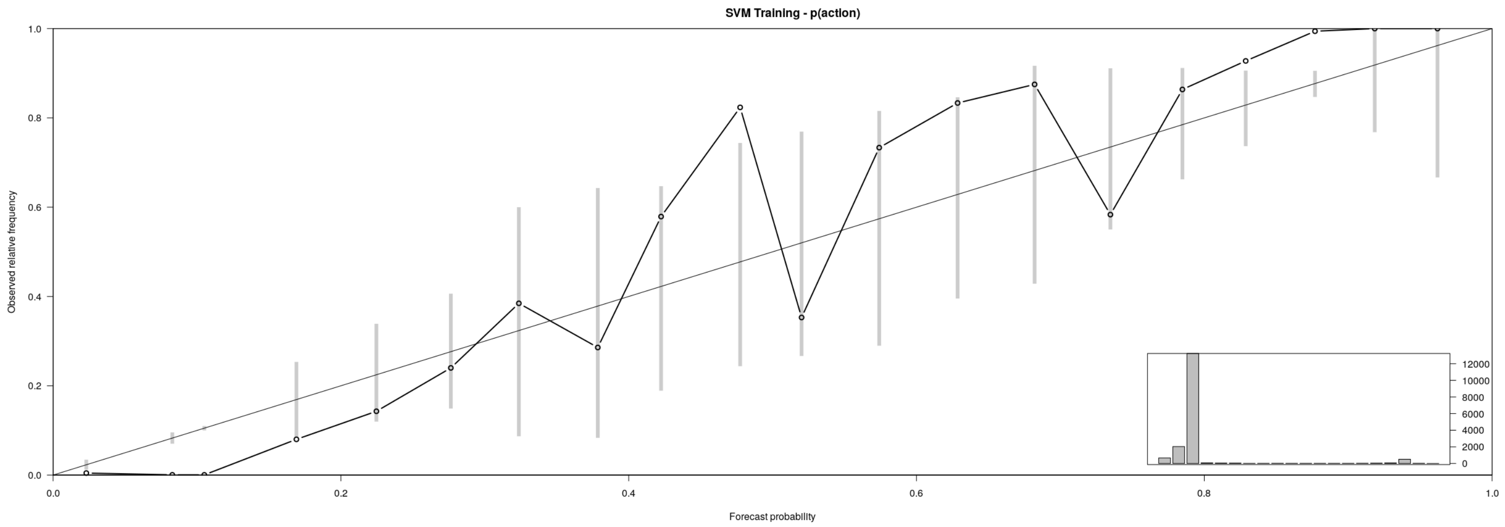

Figure 5.12: SVM Training: Exceeds Action Reliability Diagram

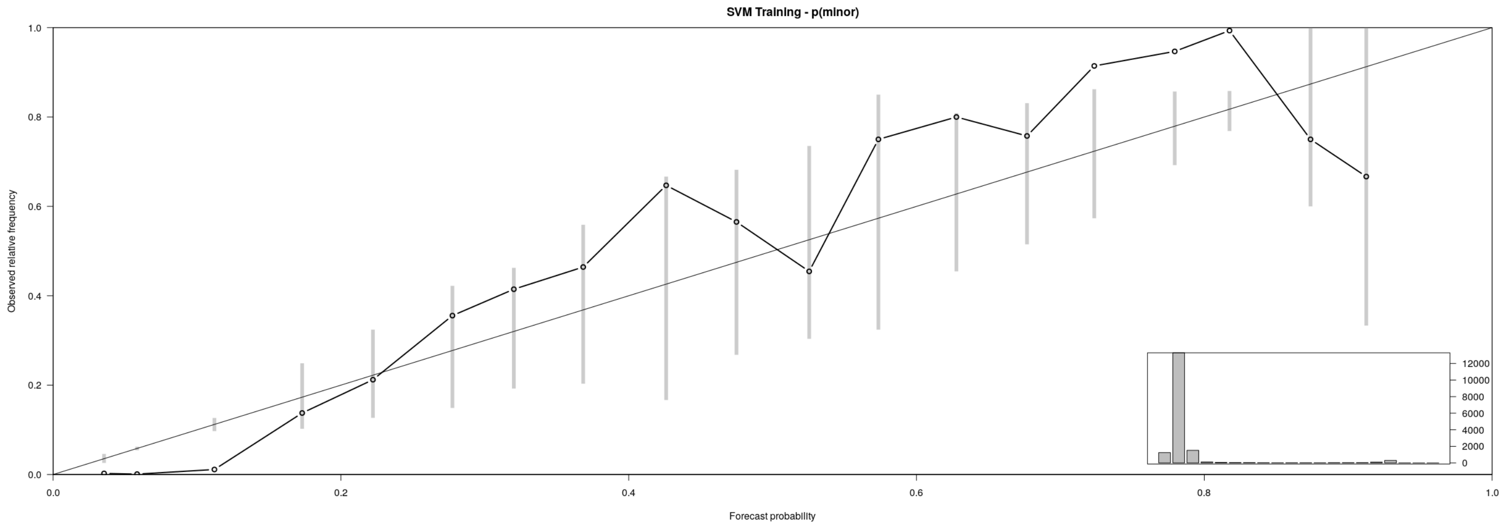

Figure 5.13: SVM Training: Exceeds Minor Reliability Diagram

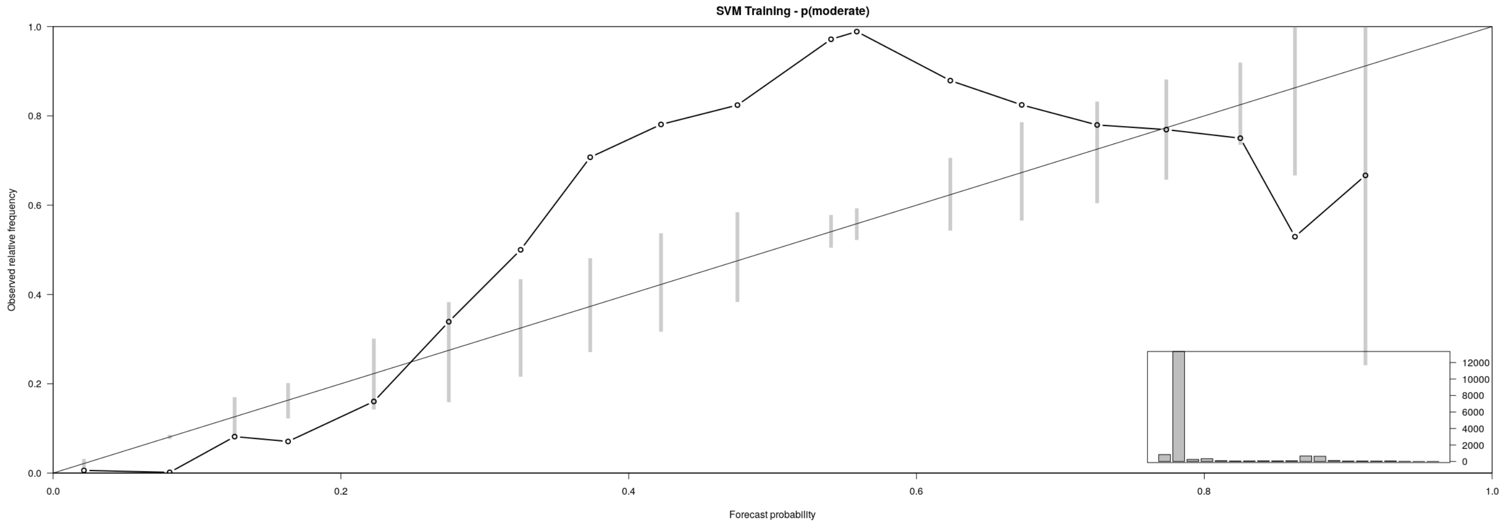

Figure 5.14: SVM Training: Exceeds Moderate Reliability Diagram

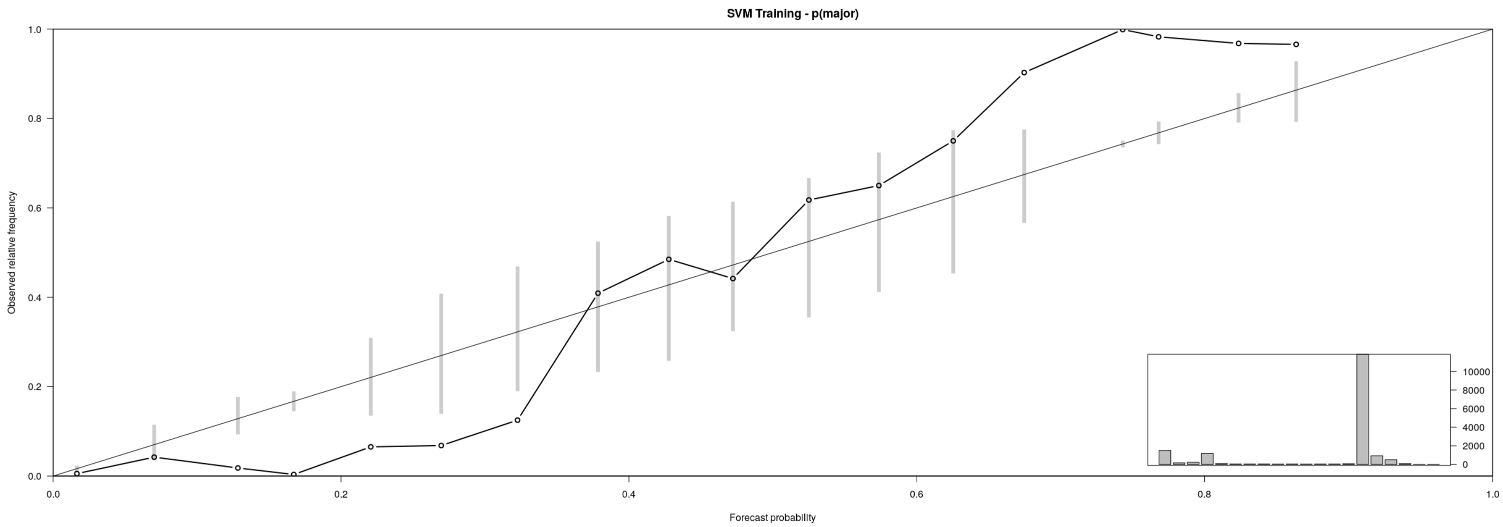

Figure 5.15: SVM Training: Exceeds Major Reliability Diagram

5.4 Validation Reliability Diagrams

5.4.1 MARS

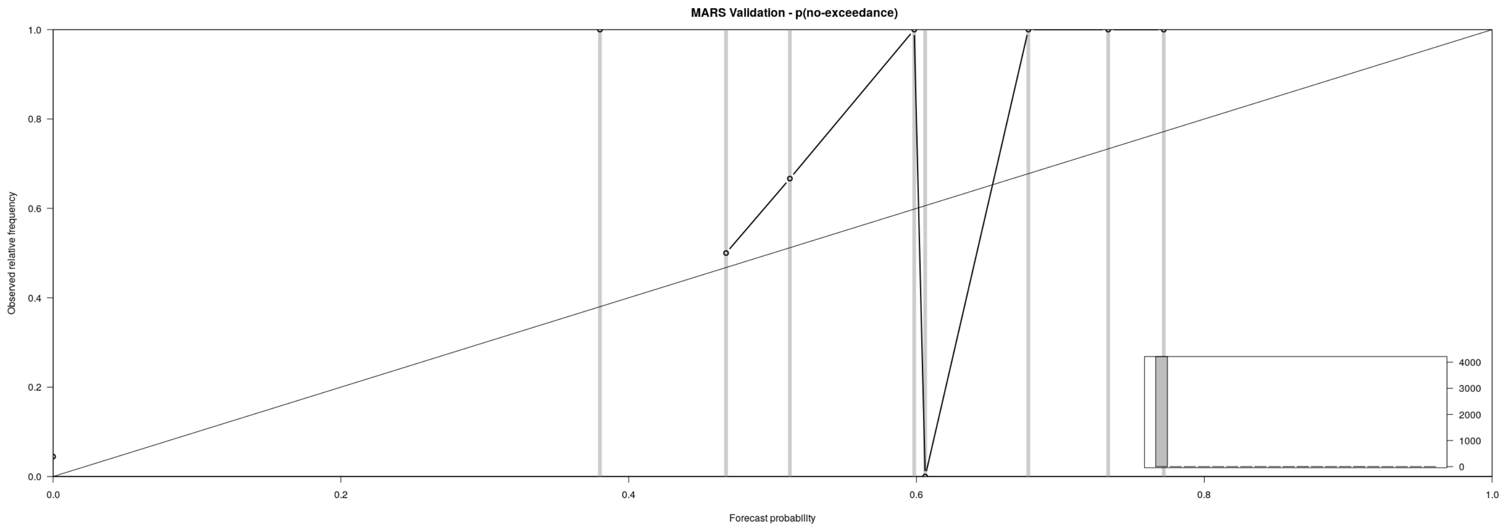

Figure 5.16: MARS Validation: No-Exceedance Reliability Diagram

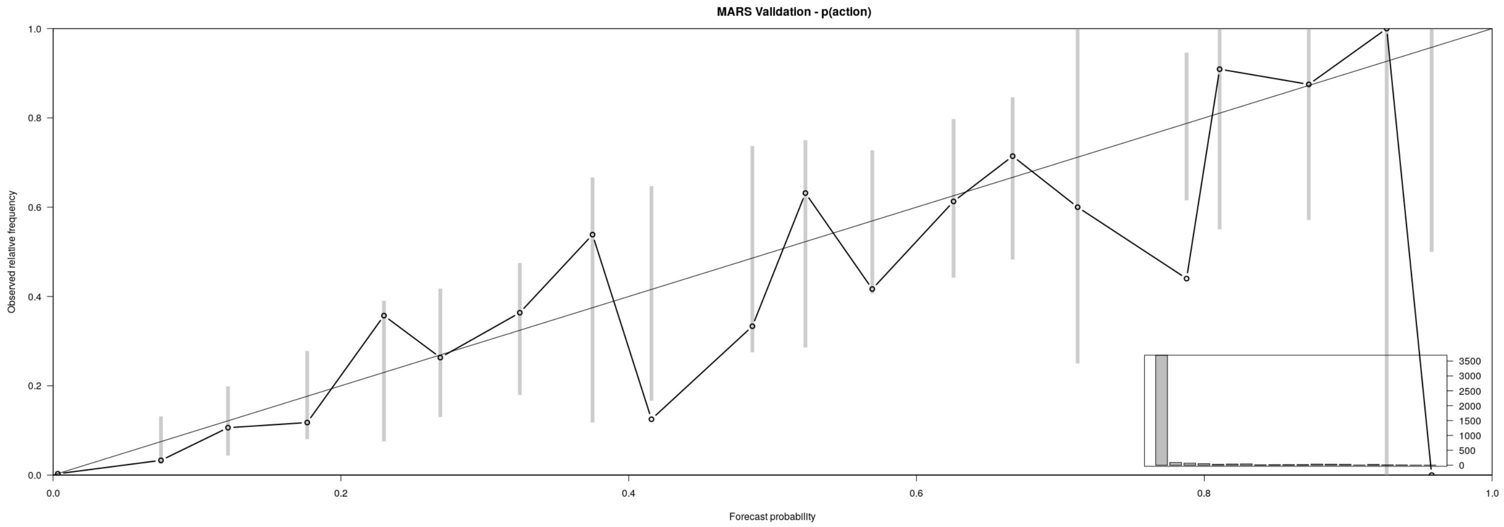

Figure 5.17: MARS Validation: Exceeds Action Reliability Diagram

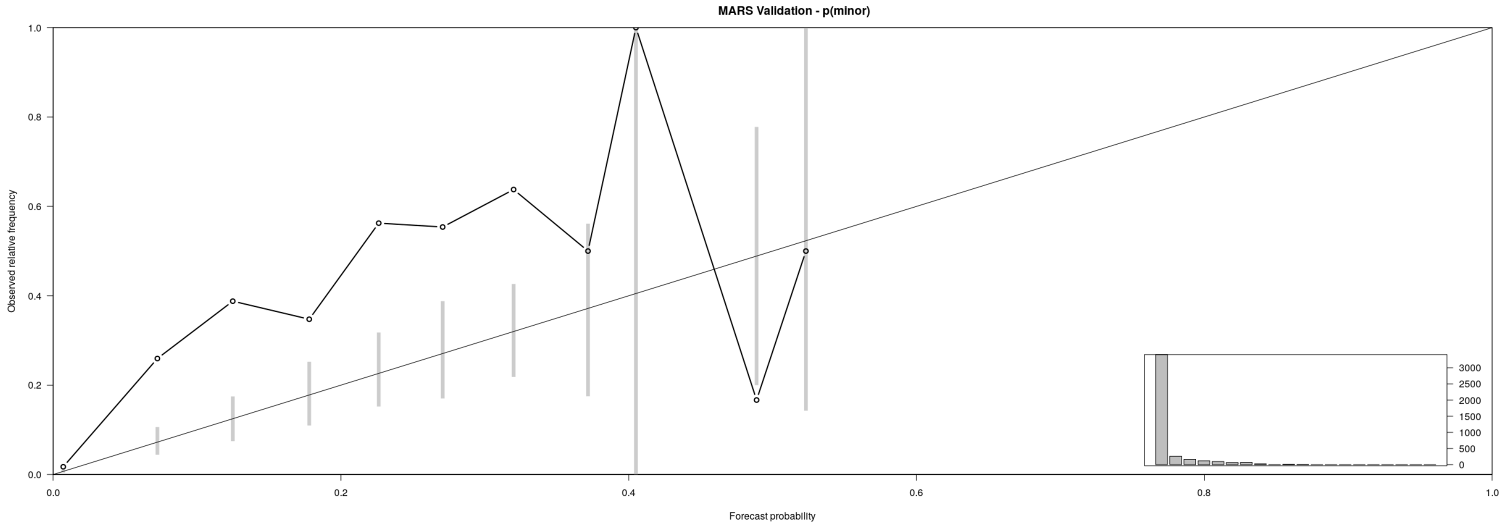

Figure 5.18: MARS Validation: Exceeds Minor Reliability Diagram

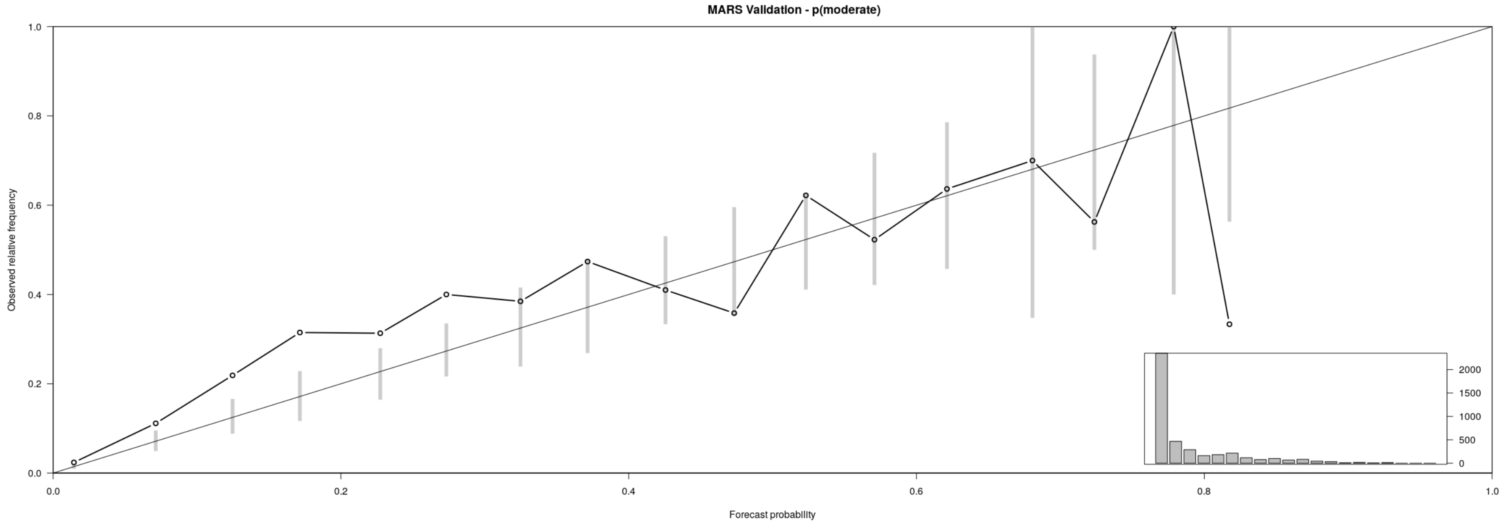

Figure 5.19: MARS Validation: Exceeds Moderate Reliability Diagram

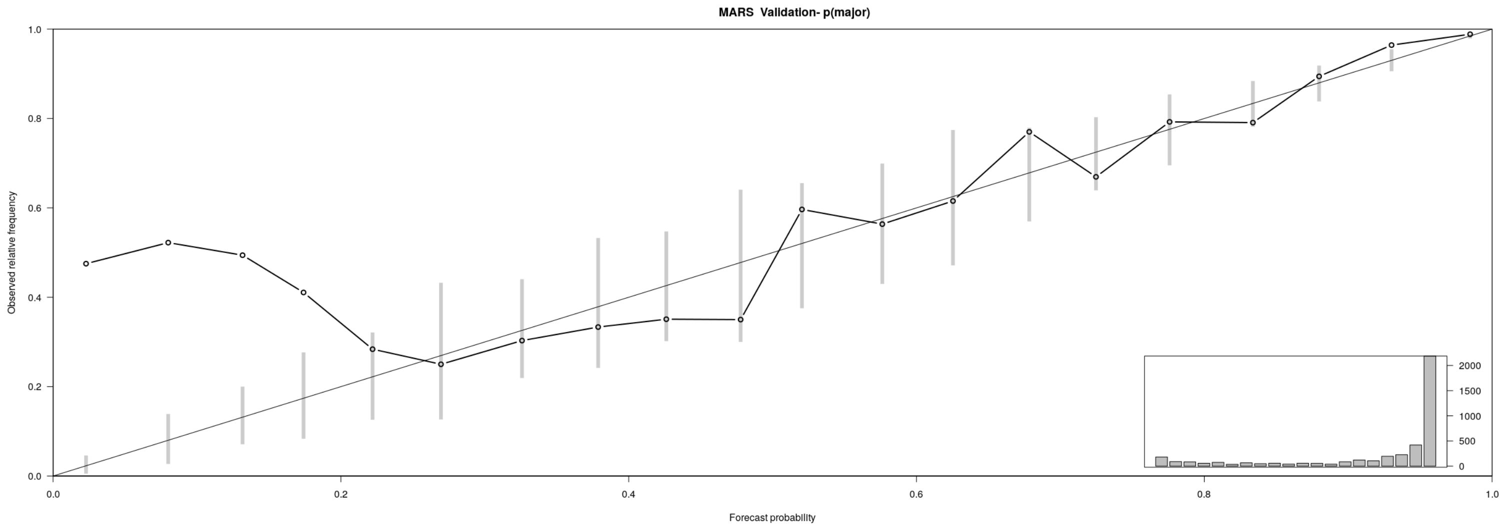

Figure 5.20: MARS Validation: Exceeds Major Reliability Diagram

5.4.2 Random Forest

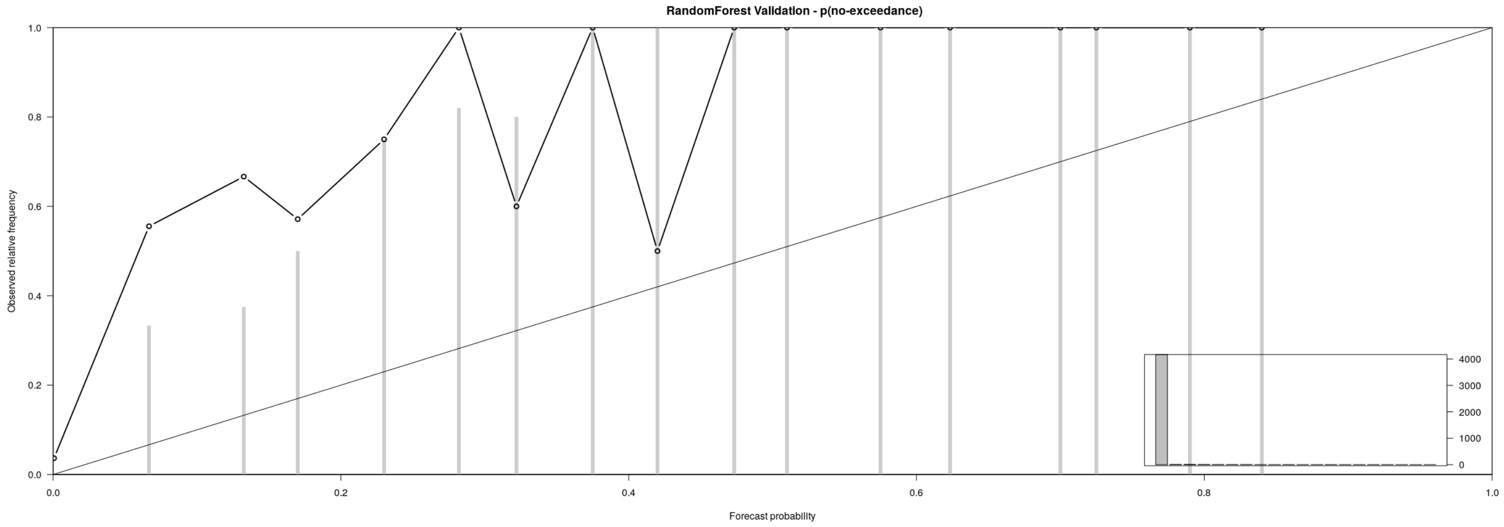

Figure 5.21: Random Forest Validation: No-Exceedance Reliability Diagram

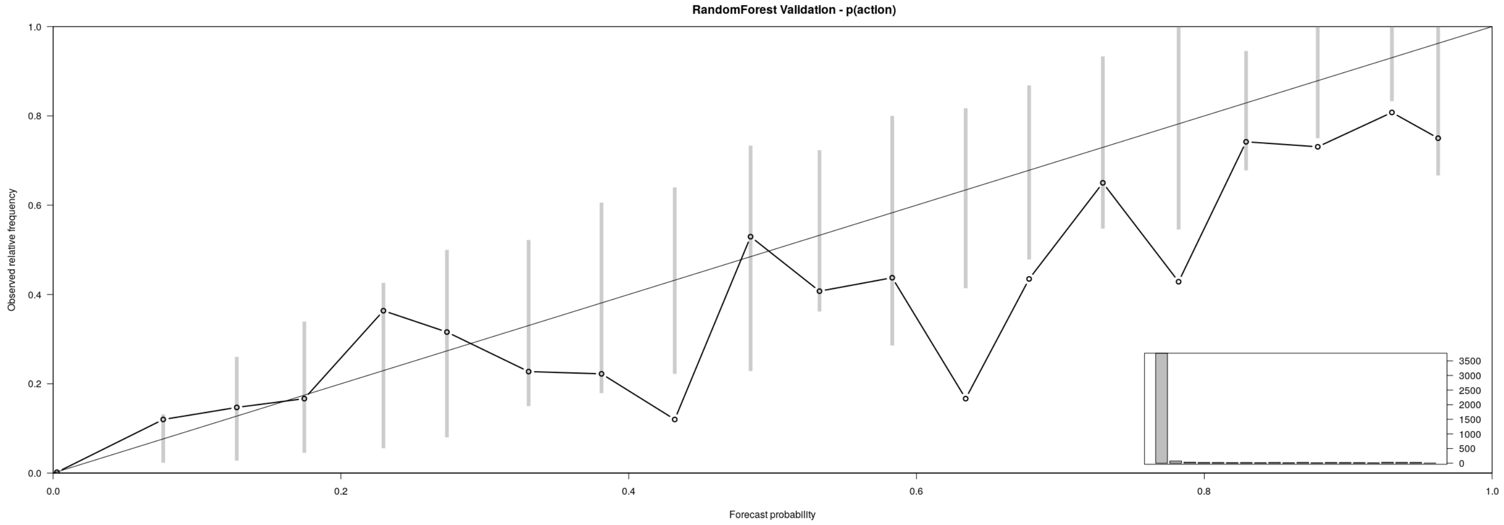

Figure 5.22: Random Forest Validation: Exceeds Action Reliability Diagram

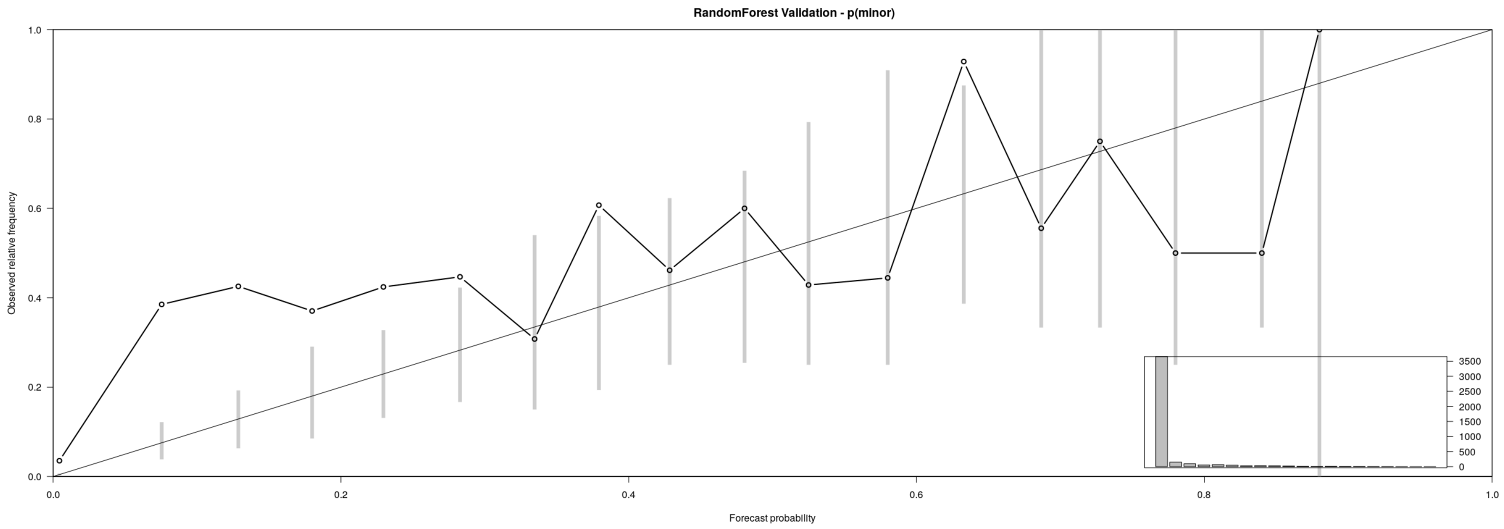

Figure 5.23: Random Forest Validation: Exceeds Minor Reliability Diagram

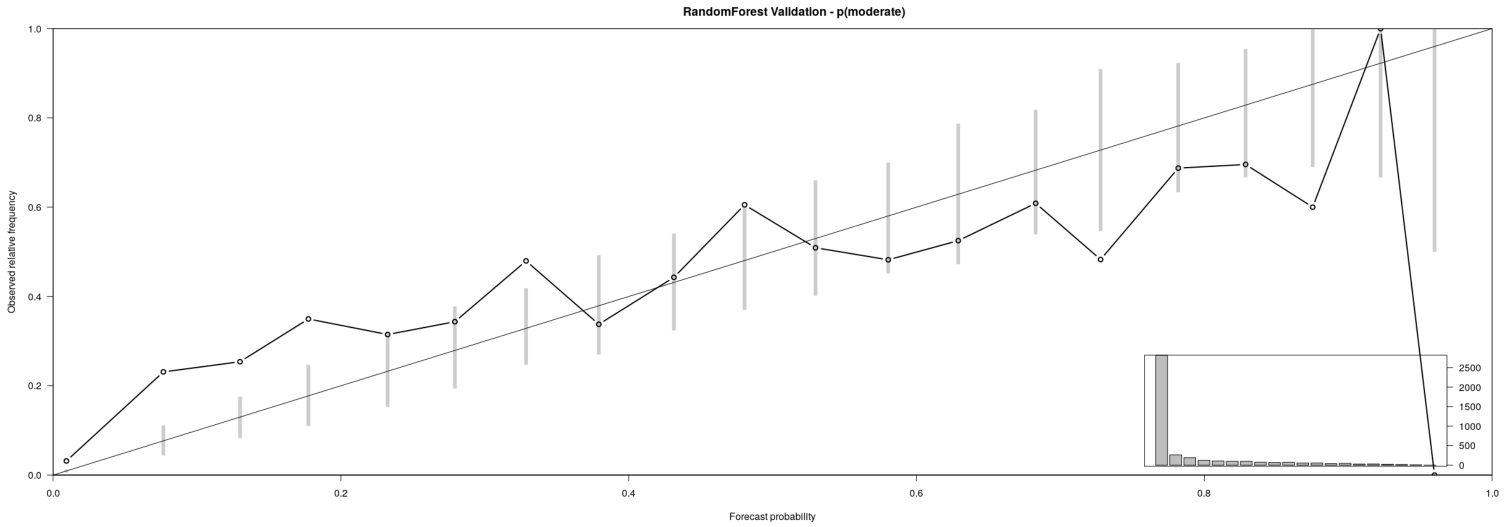

Figure 5.24: Random Forest Validation: Exceeds Moderate Reliability Diagram

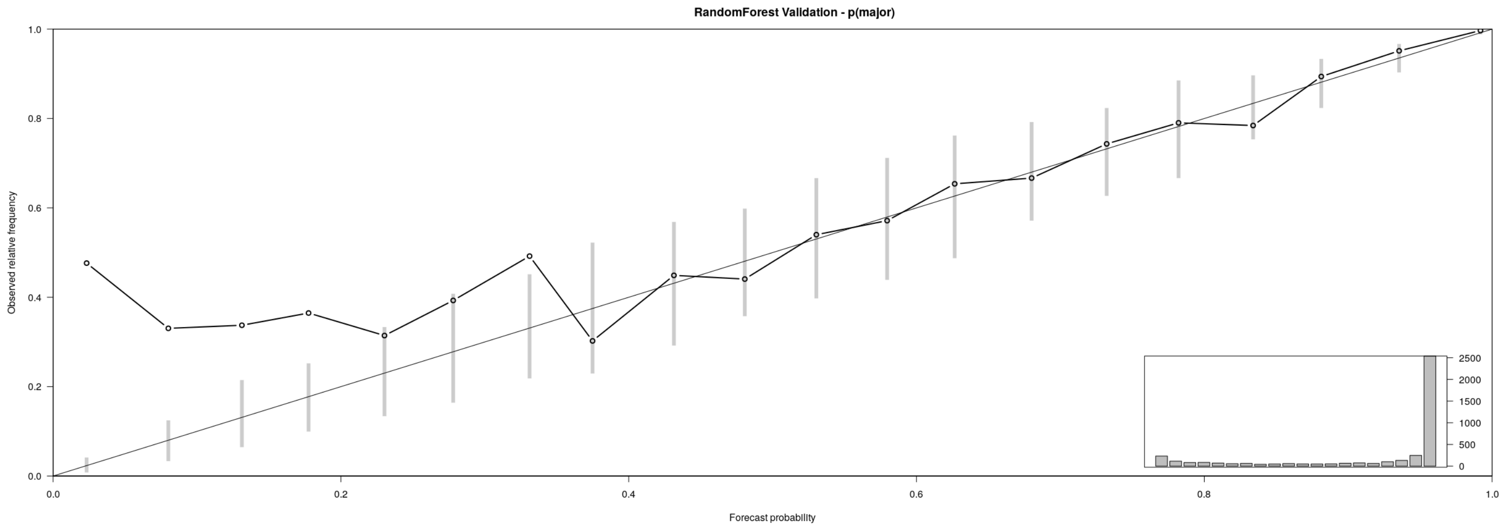

Figure 5.25: Random Forest Validation: Exceeds Major Reliability Diagram

5.4.3 Support Vector Machines

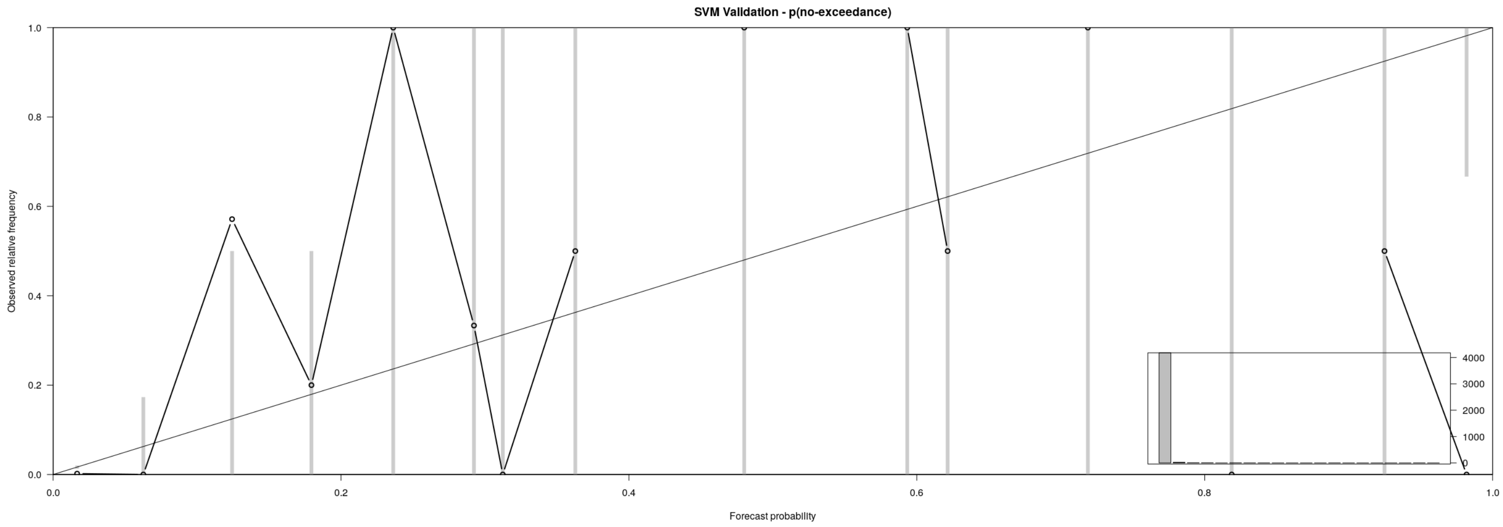

Figure 5.26: SVM Validation: No-Exceedance Reliability Diagram

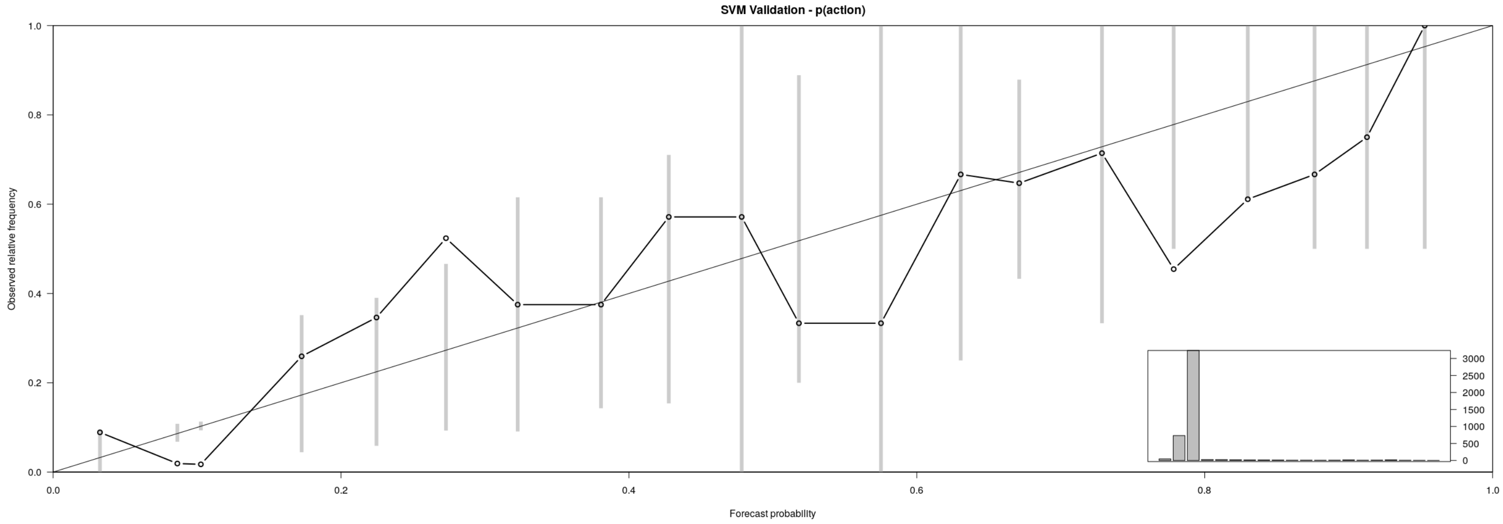

Figure 5.27: SVM Validation: Exceeds Action Reliability Diagram

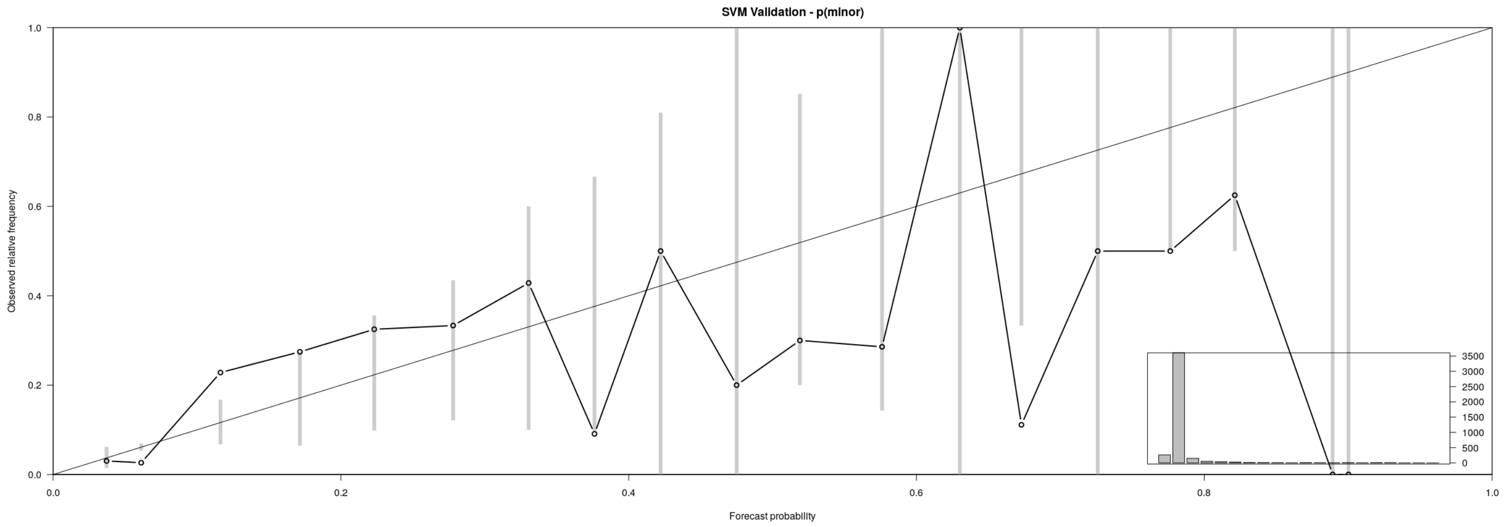

Figure 5.28: SVM Validation: Exceeds Minor Reliability Diagram

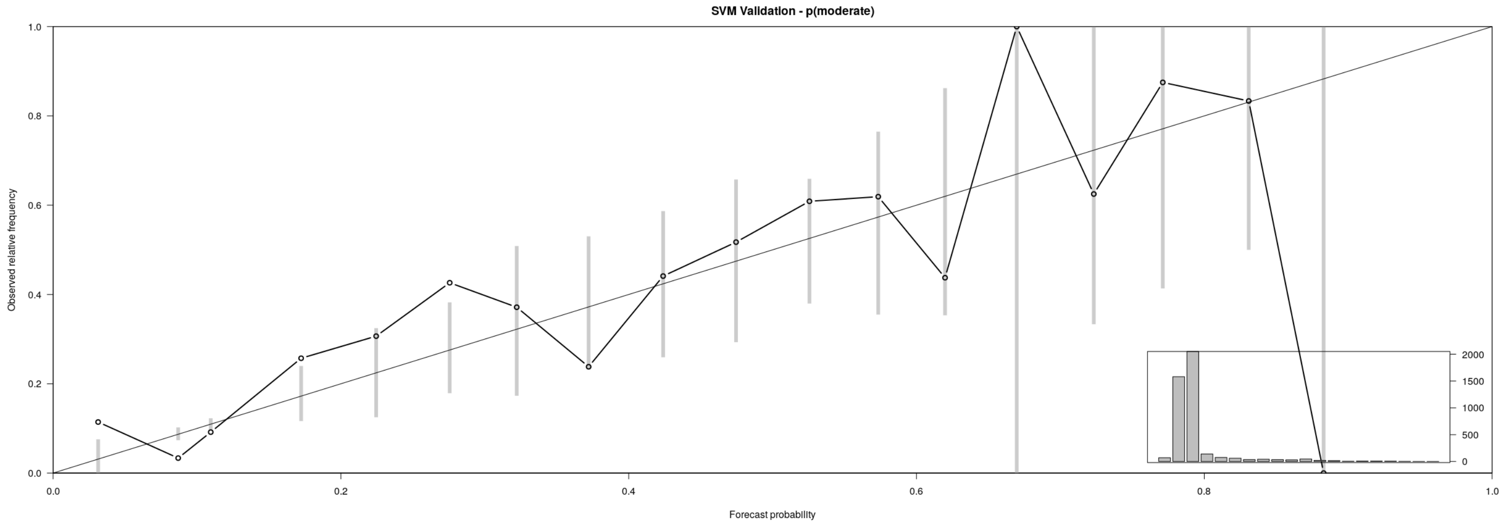

Figure 5.29: SVM Validation: Exceeds Moderate Reliability Diagram

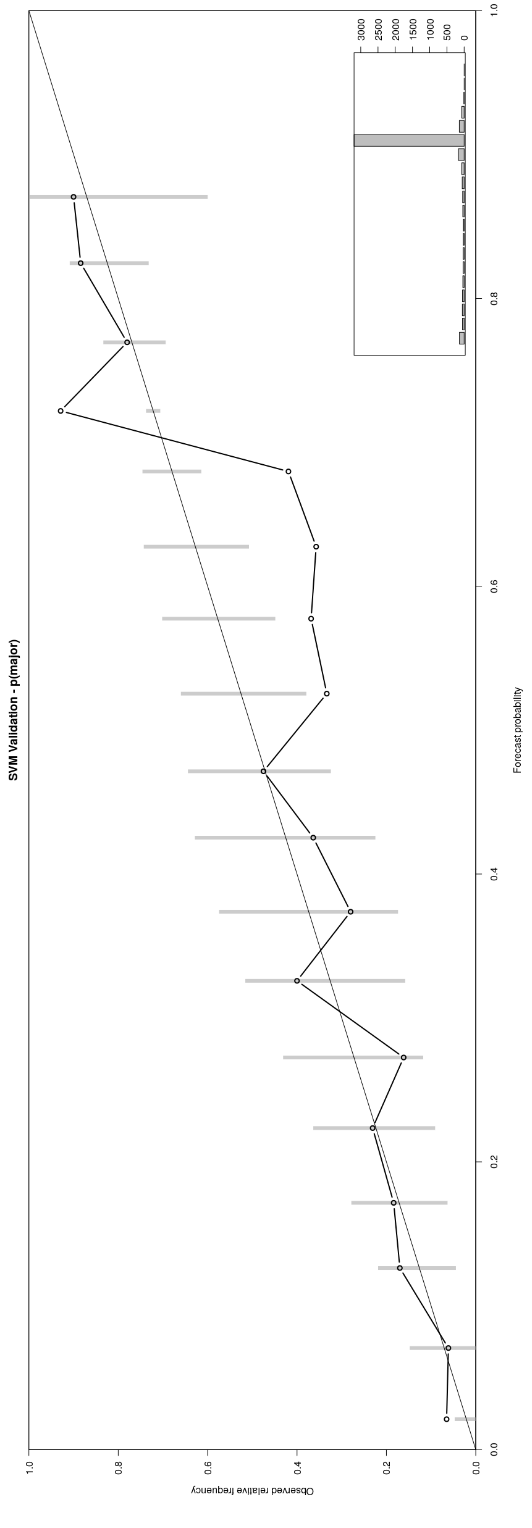

Figure 5.30: SVM Validation: Exceeds Major Reliability Diagram

5.5 Model training scripts

```
#############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM       #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS           #
#                           by                              #
#            Jorge A. Duarte G. - jduarte@ou.edu            #
#               The University of Oklahoma                  #
#                       Summer 2019                         #
#-----------------------------------------------------------#
#       Script: MARS MODELING - Lag_centroid_peak_event     #
#                         V.1.0                             #
#############################################################

# Number of cores to run with
NUM_CORES <- 8

# Library Imports
library("smooth")        # simulation metrics
library("ggplot2")       # plotting
library("earth")         # fit MARS models
library("caret")         # automating the tuning process
library("vip")           # variable importance
library("pdp")           # variable relationships
library("doParallel")    # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
validation_data <- read.csv("../../Data/final_validation_data.csv")

print("Imported Datasets")

# Create a tuning grid for MARS
hyper_grid <- expand.grid(
degree = 1:5, # Interaction effect degrees
nprune = seq(2, 54, length.out = 50) %>% floor() # Number of terms to retain
)

print("Created Tuning Grid")

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training MARS for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-validated model training
tuned_mars <- train(
x = subset(training_data, select = -c(lag_centroid_peak_event, peakq_moment, exceeds_
    threshold)),
y = training_data$lag_centroid_peak,
method = "earth",
metric = "RMSE",
trControl = trainControl(method = "repeatedcv", repeats = 10, number = 10, p = 0.25,
    allowParallel = TRUE),
tuneGrid = hyper_grid
)
```

```
# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(tuned_mars, "lag_time-MARS_10x10CV_tuned.rds")

print("Saved Model Object")

# Write console outputs to log file
sink("./lag_time-MARS_10x10CV_training_log.txt")

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
tuned_mars$results
tuned_mars$bestTune
summary(tuned_mars)

# Return variable importance
evimp(tuned_mars$finalModel, trim = FALSE)
sink()

print("DONE!")
```

Listing 5.7: R Script - MARS model: lag time

```
##############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM        #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS            #
#                           by                               #
#            Jorge A. Duarte G. - jduarte@ou.edu             #
#               The University of Oklahoma                   #
#                       Summer 2019                          #
#------------------------------------------------------------#
#       Script: MARS MODELING - peakq_moment                 #
#                          V.1.0                             #
##############################################################

# Number of cores to run with
NUM_CORES <- 8

# Library Imports
library("smooth")       # simulation metrics
library("ggplot2")      # plotting
library("earth")        # fit MARS models
library("caret")        # automating the tuning process
library("vip")          # variable importance
library("pdp")          # variable relationships
library("doParallel")   # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
validation_data <- read.csv("../../Data/final_validation_data.csv")

print("Imported Datasets")


# Create a tuning grid for MARS
hyper_grid <- expand.grid(
degree = 1:5, # Interaction effect degrees
nprune = seq(2, 54, length.out = 50) %>% floor() # Number of terms to retain
)

print("Created Tuning Grid")

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training MARS for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-validated model training
tuned_mars <- train(
x = subset(training_data, select = -c(lag_centroid_peak_event, peakq_moment, exceeds_
    threshold)),
y = training_data$peakq_moment,
method = "earth",
metric = "RMSE",
trControl = trainControl(method = "repeatedcv", repeats = 10, number = 10, p = 0.25,
    allowParallel = TRUE),
tuneGrid = hyper_grid
)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(tuned_mars, "peakq_moment-MARS_10x10CV_tuned.rds")

print("Saved Model Object")

# Write console outputs to log file
sink("./peakq_moment-MARS_10x10CV_training_log.txt")

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
tuned_mars$results
tuned_mars$bestTune
summary(tuned_mars)

# Return variable importance
evimp(tuned_mars$finalModel, trim = FALSE)
sink()

print("DONE!")
```

Listing 5.8: R Script - MARS model: peakq_moment

```
##################################################################
#        PROBABILISTIC CHARACTERIZATION OF FLOODS FROM           #
#             CATCHMENT-SCALE PRECIPITATION MOMENTS              #
#                               by                               #
#            Jorge A. Duarte G. - jduarte@ou.edu             #
#               The University of Oklahoma                   #
#                       Summer 2019                          #
#------------------------------------------------------------#
#       Script: MARS MODELING - exceeds_threshold            #
#                          V.1.0                             #
##############################################################

# Number of cores to run with
NUM_CORES <- 8

# Library Imports
library("smooth")       # simulation metrics
library("ggplot2")      # plotting
library("earth")        # fit MARS models
library("caret")        # automating the tuning process
library("vip")          # variable importance
library("pdp")          # variable relationships
library("doParallel")   # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
training_data$exceeds_threshold <- as.factor(training_data$exceeds_threshold)
validation_data <- read.csv("../../Data/final_validation_data.csv")
validation_data$exceeds_threshold <- as.factor(validation_data$exceeds_threshold)


print("Imported Datasets")


# Create a tuning grid for MARS
hyper_grid <- expand.grid(
degree = 1:5, # Interaction effect degrees
nprune = seq(2, 52, length.out = 50) %>% floor() # Number of terms to retain
)

print("Created Tuning Grid")

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training MARS for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-validated model training
tuned_mars <- train(
x = subset(training_data, select = -c(lag_centroid_peak_event, peakq_moment, exceeds_
    threshold)),
y = training_data$exceeds_threshold,
method = "earth",
metric = "Accuracy",
trControl = trainControl(method = "repeatedcv", repeats = 10, number = 10, p = 0.25,
    allowParallel = TRUE),
tuneGrid = hyper_grid
)

# Record end time
end_time <- Sys.time()
```

```r
# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(tuned_mars, "exceeds_threshold-MARS_10x10CV_tuned.rds")

print("Saved Model Object")

# Write console outputs to log file
sink("./exceeds_threshold-MARS_10x10CV_training_log.txt")

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
tuned_mars$results
tuned_mars$bestTune
summary(tuned_mars)

# Return variable importance
evimp(tuned_mars$finalModel, trim = FALSE)
sink()

print("DONE!")
```

Listing 5.9: R Script - MARS model: exceedsthreshold

```r
##############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM       #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS           #
#                           by                              #
#            Jorge A. Duarte G. - jduarte@ou.edu            #
#                The University of Oklahoma                 #
#                        Summer 2019                        #
#-----------------------------------------------------------#
# Script: RandomForest MODELING - lag_centroid_peak_event   #
#                          V.1.0                            #
##############################################################

# Number of cores to run with
NUM_CORES <- 8
NUM_TREES <- 100

# Library Imports
library("smooth")       # simulation metrics
library("ggplot2")      # plotting
library("randomForest") # fit Random Forest models
library("caret")        # automating the tuning process
library("vip")          # variable importance
library("pdp")          # variable relationships
library("doParallel")   # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
validation_data <- read.csv("../../Data/final_validation_data.csv")

print("Imported Datasets")

# Create a tuning grid for MARS
tunegrid <- expand.grid(.mtry=c(1:54))

# Create a training control vector for Random Forest
control <- trainControl(method="repeatedcv", number=10, repeats=10, search="grid")
```

```
# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training RandomForest for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-valiadted model training
rf_gridsearch <- train(lag_centroid_peak_event~., data=subset(training_data, select = -c(
    peakq_moment, exceeds_threshold)), method="rf", metric="RMSE", ntree = NUM_TREES,
    importance = TRUE, do.trace=F, tuneGrid=tunegrid, trControl=control)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(rf_gridsearch, "lag_time-RF_10x10CV_gridsearch.rds")

print("Saved Model Object")

# Write console outputs to log file
sink("./lag_time-RF_10x10CV_training_log.txt")

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
print(rf_gridsearch)
rf_gridsearch$finalModel
summary(rf_gridsearch)
sink()

print("DONE!")
```

Listing 5.10: R Script - Random Forest model: lag time

```
#############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM       #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS           #
#                           by                              #
#            Jorge A. Duarte G. - jduarte@ou.edu            #
#               The University of Oklahoma                  #
#                       Summer 2019                         #
#-----------------------------------------------------------#
# Script: RandomForest MODELING - peakq_moment              #
#                          V.1.0                            #
#############################################################

# Number of cores to run with
NUM_CORES <- 8
NUM_TREES <- 100

# Library Imports
library("smooth")       # simulation metrics
library("ggplot2")      # plotting
library("randomForest") # fit Random Forest models
library("caret")        # automating the tuning process
library("vip")          # variable importance
library("pdp")          # variable relationships
library("doParallel")   # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
validation_data <- read.csv("../../Data/final_validation_data.csv")

print("Imported Datasets")

# Auxiliary Functions
# Common Error Metrics Function
error_metrics <- function(obs, pred){
outcomes <- data.frame(obs, pred)
print(paste("MAE", round(MAE(obs, pred), 3)))
print(paste("MSE", round(ModelMetrics::mse(obs, pred), 3)))
print(paste("MPE", round(MPE(obs, pred), 3)))
print(paste("MAPE", round(MAPE(obs, pred), 3)))
}

# Create a tuning grid for MARS
tunegrid <- expand.grid(.mtry=c(1:52))

# Create a training control vector for Random Forest
control <- trainControl(method="repeatedcv", number=10, repeats=10, search="grid")

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training RandomForest for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-valiadted model training
rf_gridsearch <- train(peakq_moment˜., data=subset(training_data, select = -c(lag_
    centroid_peak_event, exceeds_threshold)), method="rf", metric="RMSE", ntree = NUM_
    TREES,  importance = TRUE, do.trace=F, tuneGrid=tunegrid, trControl=control)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(rf_gridsearch, "peakq_moment-RF_10x10CV_gridsearch.rds")

print("Saved Model Object")

# Write console outputs to log file
sink("./peakq_moment-RF_10x10CV_training_log.txt")

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
```

```
print(rf_gridsearch)
rf_gridsearch$finalModel
summary(rf_gridsearch)
sink()

print("DONE!")
```

Listing 5.11: R Script - Random Forest model: peakq_moment

```
##############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM        #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS            #
#                           by                               #
#            Jorge A. Duarte G. - jduarte@ou.edu             #
#               The University of Oklahoma                   #
#                      Summer 2019                           #
#------------------------------------------------------------#
# Script: RandomForest MODELING - exceeds_threshold          #
#                         V.1.0                              #
##############################################################

# Number of cores to run with
NUM_CORES <- 8
NUM_TREES <- 100
REPS_CV <- 10
FOLDS_CV <- 10

# Library Imports
library("smooth")       # simulation metrics
library("ggplot2")      # plotting
library("randomForest") # fit Random Forest models
library("caret")        # automating the tuning process
library("vip")          # variable importance
library("pdp")          # variable relationships
library("doParallel")   # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
training_data$exceeds_threshold <- as.factor(training_data$exceeds_threshold)
validation_data <- read.csv("../../Data/final_validation_data.csv")
validation_data$exceeds_threshold <- as.factor(validation_data$exceeds_threshold)

print("Imported Datasets")


# Create a tuning grid for MARS
tunegrid <- expand.grid(.mtry=c(1:52))

# Create a training control vector for Random Forest
control <- trainControl(method="repeatedcv", number=FOLDS_CV, repeats=REPS_CV, search="
    grid")

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training RandomForest for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)
```

```
# Cross-valiadted model training
rf_gridsearch <- train(exceeds_threshold~., data=subset(training_data, select = -c(lag_
    centroid_peak_event, peakq_moment)), method="rf", metric="Accuracy", ntree = NUM_
    TREES,  importance = TRUE, do.trace=F, tuneGrid=tunegrid, trControl=control)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(rf_gridsearch, paste("exeeds_threshold-RF_",REPS_CV,"x",FOLDS_CV,"CV_gridsearch.
    rds", sep=""))

print("Saved Model Object")

# Write console outputs to log file
sink(paste("./exeeds_threshold-RF_",REPS_CV,"x",FOLDS_CV,"CV_training_log.txt", sep=""))

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
print(rf_gridsearch)
rf_gridsearch$finalModel
summary(rf_gridsearch)
sink()

print("DONE!")
```

Listing 5.12: R Script - Random Forest model: exceeds_threshold

```
#############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM       #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS           #
#                           by                              #
#            Jorge A. Duarte G. - jduarte@ou.edu            #
#               The University of Oklahoma                  #
#                       Summer 2019                         #
#-----------------------------------------------------------#
# Script: Suport Vector Machines - lag_centroid_peak_event  #
#                          V.1.0                            #
#############################################################

# Number of cores to run with
NUM_CORES <- 8
REPS_CV <- 10
FOLDS_CV <- 10

# Library Imports
library("smooth")        # simulation metrics
library("ggplot2")       # plotting
library("kernlab")       # kernel-based learning utilities
library("caret")         # automating the tuning process
library("vip")           # variable importance
library("pdp")           # variable relationships
library("doParallel")    # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
training_data$exceeds_threshold <- as.factor(training_data$exceeds_threshold)
validation_data <- read.csv("../../Data/final_validation_data.csv")
```

```
validation_data$exceeds_threshold <- as.factor(validation_data$exceeds_threshold)

print("Imported Datasets")


# Create a tuning grid for SVM parameters sigma and C
tunegrid <- expand.grid(sigma = seq(0,5,length=10), C = seq(0,5,length=10))

# Create a training control vector for SVM
control <- trainControl(method="repeatedcv", number=FOLDS_CV, repeats=REPS_CV)

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training SVM for lag_time")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-valiadted model training
svm.tune <- train(lag_centroid_peak_event~., data=subset(training_data, select = -c(
    exceeds_threshold, peakq_moment)), method = "svmRadial",  tuneGrid = tunegrid,
    trControl=control)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(svm.tune, paste("lag_time-SVM_",REPS_CV,"x",FOLDS_CV,"CV_gridsearch.rds", sep="")
    )

print("Saved Model Object")

# Write console outputs to log file
sink(paste("./lag_time-SVM_",REPS_CV,"x",FOLDS_CV,"CV_training_log.txt", sep=""))

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
print(svm.tune)
summary(svm.tune)
svm.tune$finalModel
summary(svm.tune$finalModel)
sink()

print("DONE!")
```

Listing 5.13: R Script - SVM model: lag time

```
##############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM        #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS            #
#                           by                               #
#            Jorge A. Duarte G. - jduarte@ou.edu             #
#               The University of Oklahoma                   #
#                       Summer 2019                          #
```

```
#------------------------------------------------------------#
#       Script: Suport Vector Machines - peakq_moment        #
#                          V.1.0                             #
##############################################################

# Number of cores to run with
NUM_CORES <- 8
REPS_CV <- 10
FOLDS_CV <- 10

# Library Imports
library("smooth")        # simulation metrics
library("ggplot2")       # plotting
library("kernlab")       # kernel-based learning utilities
library("caret")         # automating the tuning process
library("vip")           # variable importance
library("pdp")           # variable relationships
library("doParallel")    # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../Data/final_training_data.csv")
training_data$exceeds_threshold <- as.factor(training_data$exceeds_threshold)
validation_data <- read.csv("../../Data/final_validation_data.csv")
validation_data$exceeds_threshold <- as.factor(validation_data$exceeds_threshold)

print("Imported Datasets")


# Create a tuning grid for SVM parameters sigma and C
tunegrid <- expand.grid(sigma = seq(0,5,length=10), C = seq(0,5,length=10))

# Create a training control vector for SVM
control <- trainControl(method="repeatedcv", number=FOLDS_CV, repeats=REPS_CV)

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training SVM for peakq_moment")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-valiadted model training
svm.tune <- train(peakq_moment~., data=subset(training_data, select = -c(lag_centroid_
    peak_event, exceeds_threshold)), method = "svmRadial",  tuneGrid = tunegrid,
    trControl=control)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(svm.tune, paste("peakq_moment-SVM_",REPS_CV,"x",FOLDS_CV,"CV_gridsearch.rds", sep
    =""))

print("Saved Model Object")
```

```
# Write console outputs to log file
sink(paste("./peakq_moment-SVM_",REPS_CV,"x",FOLDS_CV,"CV_training_log.txt", sep=""))

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
print(svm.tune)
summary(svm.tune)
svm.tune$finalModel
summary(svm.tune$finalModel)
sink()

print("DONE!")
```

Listing 5.14: R Script - SVM model: peakq_moment

```
#############################################################
#       PROBABILISTIC CHARACTERIZATION OF FLOODS FROM       #
#           CATCHMENT-SCALE PRECIPITATION MOMENTS           #
#                           by                              #
#            Jorge A. Duarte G. - jduarte@ou.edu            #
#               The University of Oklahoma                  #
#                       Summer 2019                         #
#-----------------------------------------------------------#
#    Script: Suport Vector Machines - exceeds_threshold     #
#                          V.1.0                            #
#############################################################

# Number of cores to run with
NUM_CORES <- 8
REPS_CV <- 10
FOLDS_CV <- 10

# Library Imports
library("smooth")        # simulation metrics
library("ggplot2")       # plotting
library("kernlab")       # kernel-based learning utilities
library("caret")         # automating the tuning process
library("vip")           # variable importance
library("pdp")           # variable relationships
library("doParallel")    # CPU Parallelization

print("Imported Libraries")

# Data Imports
training_data <- read.csv("../../../Data/final_training_data.csv")
training_data$exceeds_threshold <- as.factor(training_data$exceeds_threshold)

levels(training_data$exceeds_threshold)[levels(training_data$exceeds_threshold) == ’0’]
    <- ’NoExceedance’
levels(training_data$exceeds_threshold)[levels(training_data$exceeds_threshold) == ’1’]
    <- ’Action’
levels(training_data$exceeds_threshold)[levels(training_data$exceeds_threshold) == ’2’]
    <- ’Minor’
levels(training_data$exceeds_threshold)[levels(training_data$exceeds_threshold) == ’4’]
    <- ’Moderate’
levels(training_data$exceeds_threshold)[levels(training_data$exceeds_threshold) == ’8’]
    <- ’Major’

validation_data <- read.csv("../../../Data/final_validation_data.csv")
validation_data$exceeds_threshold <- as.factor(validation_data$exceeds_threshold)

levels(validation_data$exceeds_threshold)[levels(validation_data$exceeds_threshold) == ’0
    ’] <- ’NoExceedance’
levels(validation_data$exceeds_threshold)[levels(validation_data$exceeds_threshold) == ’1
    ’] <- ’Action’
```

```
levels(validation_data$exceeds_threshold)[levels(validation_data$exceeds_threshold) == '2
    '] <- 'Minor'
levels(validation_data$exceeds_threshold)[levels(validation_data$exceeds_threshold) == '4
    '] <- 'Moderate'
levels(validation_data$exceeds_threshold)[levels(validation_data$exceeds_threshold) == '8
    '] <- 'Major'

print("Imported Datasets")

# Create a tuning grid for SVM parameters sigma and C
#tunegrid <- expand.grid(sigma = seq(0,5,length=10), C = seq(0,5,length=10))
tunegrid <- expand.grid(sigma = 0.5555556, C = 1.666667)

# Create a training control vector for SVM
control <- trainControl(method="repeatedcv", number=FOLDS_CV, repeats=REPS_CV, classProbs
     =  TRUE)

# Set seed for reproducibiity
set.seed(123)

# Instantiate parallelization socket cluster (# of cores to use)
cl <- makePSOCKcluster(NUM_CORES)
registerDoParallel(cl)

print("Started training SVM for exceeds_threshold")

# Record start time
start_time <- Sys.time()
paste("Start time: ", start_time)

# Cross-valiadted model training
svm.tune <- train(exceeds_threshold~., data=subset(training_data, select = -c(lag_
    centroid_peak_event, peakq_moment)), method = "svmRadial",  tuneGrid = tunegrid,
    trControl=control)

# Record end time
end_time <- Sys.time()

# Stop the socket cluster and free up the cores
stopCluster(cl)

print("Finished Training")

# Save tuned MARS model object
saveRDS(svm.tune, paste("exceeds_threshold-SVM_",REPS_CV,"x",FOLDS_CV,"CV_gridsearch.rds"
    , sep=""))

print("Saved Model Object")

# Write console outputs to log file
sink(paste("./exceeds_threshold-SVM_",REPS_CV,"x",FOLDS_CV,"CV_training_log.txt", sep="")
    )

# Record training time
paste("Run time: ", end_time - start_time)

# Return results, best model and summary
print(svm.tune)
summary(svm.tune)
svm.tune$finalModel
summary(svm.tune$finalModel)
sink()

print("DONE!")
```

Listing 5.15: R Script - SVM model: exceeds_threshold