UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DESIGNING RELIABLE MACHINE LEARNING ALGORITHMS FOR EARLY PREDICTION OF PREECLAMPSIA

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

Master of Science

By
Rachel Bennett
Norman, Oklahoma

2021

DESIGNING RELIABLE MACHINE LEARNING ALGORITHMS FOR EARLY PREDICTION OF PREECLAMPSIA

A THESIS APPROVED FOR THE
GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Talayeh Razzaghi, Chair

Dr. Dean Hougen

Dr. Charles Nicholson

Abstract

Known as a pregnancy complication due to high blood pressure and may be accompanied by damage to another organ system, preeclampsia afflicts between 3 and 6 percent of US pregnancies each year. Studies have shown the importance of early detection of preeclampsia to prevent further complications that are detrimental to both mother and infant. In this work, we develop an algorithmic modification of Deep Neural Networks to identify high-risk patients in preeclampsia diagnosis using imbalanced datasets in the presence of missing values. We identify the most influential set of clinical features relevant to preeclampsia and train a classifier that can be embedded within a clinical decision support system. Our results provide evidence in favor of increased consideration of patient race/ethnicity in preeclampsia prediction, and for more personalized medicine in general.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Preeclampsia spectrum disorders occurs in pregnant women that are generally linked by new onset hypertension and proteinuria after week 20 of gestation. Preeclampsia affects 2–8% of pregnancies worldwide (Duley, 2009), and is responsible for between 50,000–100,000 deaths annually. Of those women who survive, preeclampsia is associated with longterm health effects, such as increased risk of heart disease, stroke, and diabetes (Bellamy et al., 2007). Children of women with preeclampsia also have increased risk of long-term cardiovascular illness (Sacks et al., 2018). Studies have shown that the administration of low-dose aspirin early in pregnancy can reduce the occurrence of preeclampsia in pregnant women (Bujold et al., 2010). Early indication of preeclampsia would then allow clinicians to provide treatment to the most at-risk women.

Because prediction would allow women to be treated more effectively, multiple predictive models have been developed in the past to accomplish this task. Kenny et al. (2014) and Sandström et al. (2019) applied logistic regression to predict preeclampsia in nulliparous women. Moreira et al. (2017) and Sufriyana et al. (2020) have succesfully used Random Forest models to predict preeclampsia, however the Random Forest algorithm tends to show bias in the presence categorical variables with many levels (Strobl et al., 2007). Marić et al. (2020) applied the Elastic Net model to preeclampsia prediction. However, their

data included only a single high-risk referral hospital, thus having a higher occurance of preeclampsia than appears in a more general population.

Furthermore, the mothers' race and ethnicity are among significant risk factors for preeclampsia (Johnson, Louis, 2020). Maternal morbidity is significantly higher among women in particular racial and ethnic groups (Admon et al., 2018). For example, African American women are more likely to experience preeclampsia (Breathett et al., 2014), and when they do, the rate of mortality is higher among them as an outcome of this disease (Shahul et al., 2015). In addition, African American and American Indian women are at increased risk of recurrence of preeclampsia (Boghossian et al., 2014). Moreover, when preeclampsia reoccurs, they are at higher risk of having a baby with low birth weight and pre-term birth (Mbah et al., 2011).

There has been little study on the occurrence of preeclampsia and its related risk factors among Native American women. Hypertensive disorders of pregnancy accounted for significantly higher proportion of Native American than Caucasian counterparts, and in particular, Native American women had a 17% increased risk for preeclampsia compared to white women according to a recent study (Heck et al., 2020). Existing analysis of preeclampsia using machine leaning methods have rarely studied race and ethnicity into their predictions. To our knowledge, there is no study that investigated the risk factors of preeclampsia among African American and Native American populations using advanced machine learning algorithms.

Analyzing health care datasets presents its own set of challenges. Converting a large number of categorical variables to numerical values in the data in this study, resuting in extremely sparse feature matrices. Additionally, several samples contained missing values that are needed to be dropped or imputed. By far the largest issue however was the highly imbalanced nature of preeclampsia since only a small set of women (minority class) develops preeclampsia in a given year. Traditional machine learning algorithms often produce biased results in favor of the majority class (non-preeclamptic individuals) (Khan et al., 2017).

The contribution of this study are: 1) we propose the use of a cost-sensitive learning-based

Deep Neural Network in order to deal with the highly imbalanced nature of this problem. To the best of our knowledge, this algorithm has not been used in any previous study to predict the development of preeclampsia; 2) this study aims at the development of prediction models among minority groups, particularly with seperate analysis performed on Native American and African American datasets.

The rest of this thesis is organized as follows: In chapter 2, we describe the related work that has been performed on preeclampsia prediction. In chapter 3, we go into the methodology used for this thesis. In chapter 4, we describe the datasets our method was used on. In chapter 5, we describe the results of our methods and analyze the output. Finally, in chapter 6 we discuss conclusions and future research directions.

Chapter 2

Related Works

Due to the need for early detection of preeclampsia, a number of machine learning methods have been applied to the problem in the literature. The most basic machine learning models are the linear and generalized linear models, in which the relationship between the input variables and the target variables is assumed to be linear. Among these methods, Logistic Regression is the most common one, which is well-suited to binary classification problems (Marić et al., 2020; Sufriyana et al., 2020; Sandström et al., 2019; Kenny et al., 2014). Other linear methods have been applied to this problem such as Generalized Linear and Elastic Net models (Marić et al., 2020), and Linear Support Vector Machine (Sufriyana et al., 2020). Furthermore, methods that do not assume a linear relationship have been utilized for early detection of preeclampsia. Sufriyana et al. (2020) employed decision trees, which use a series of learned binary rules in order to classify observations. These methods, while easy for a clinician to interpret, suffer from over-fitting and high variability which makes them difficult to apply to healthcare datasets. An improvement on these are random forests, which aggregate the results of multiple decision trees in order to reduce the variability. This method was used successfully by Sufriyana et al. (2020), and Moreira et al. (2017). However, random forest also suffers from bias in datasets containing large amounts of categorical data (Strobl et al., 2007). Clinical datasets often include many categorical features which makes

the use of random forest challenging. Since the performance of random forest technique is deteriorated when there are many levels in categorical features [I will add reference]. The use of Neural Network method is still limited in preeclampsia prediction. Sufriyana et al. (2020) employed neural network method for early detection of preeclampsia. The advantage of this method is that it explores the nonlinear relationship among features. Also, their work is capable of handling both categorical and numeric features and is also well-suited to large datasets.

On the other hand, preeclampsia is a rare disease and only occurs in such a small subset of the population. Thus, it suffers from a class imbalance issue. A class imbalance occurs when the majority of samples in a datasets belongs to one class (Leevy et al., 2018a), which makes traditional machine learning algorithms learn from the minority class. A number of methods have been developed to deal with this issue in the literature. The most common techniques are undersampling and oversampling. In undersampling, a subset of the majority class is used for training while in Oversampling, the minority class is resampled with replacement in a way that the number of samples in both classes become equal. Synthetic Minority Oversampling (SMOTE) (Chawla et al., 2002) is the most popular oversampling approach in which new minority class samples are generated from the existing data by drawing samples that are close in feature space, then fit a line between them and generate a new synthetic data point from along that line (Chawla et al., 2002). However, these resampling methods have the downside of changing the distribution of the dataset. In undersampling, bias is introduced by removing information from the data while oversampling increases variance by creating redundant data points and runs the risk of overfitting, and thus, it reduces its ability to generalize to the new data samples. Finally, SMOTE tends to create ambiguous samples if there is considerable overlap in features between the majority and minority class samples (Fernández et al., 2018). Due to these issues reliable methods that do not alter the distribution of data are required for handling the class imbalanced problems. For this purpose, cost-sensitive learning algorithms are developed which assigns different weights to

the samples of each class based on their importance (Kukar, Kononenko, 1998). We note that the standard machine learning algorithms assume equal wights for each data samples and are prone to generate many misclassified samples. This will cause a huge loss in many real-world applications. In a healthcare setting, there is frequently a much higher cost to misdiagnosing someone who does have a disease than someone who doesn't. Kukar Kukar, Kononenko (1998) has proposed cost-sensitive neural networks to circumvent this issue. Cost-sensitive neural networks have been successfully applied in other imbalanced data problems both in healthcare (Du et al., 2021) and other domains, such as fraud detection (Dastile et al., 2020), fault detection (Fuqua, Razzaghi, 2020), and image recognition Khan et al. (2017). To our knowledge, none of the previous works have studied preeclampsia prediction using this method. In fact, very few researchers studied the preeclampsia prediction problem using imbalanced classification methods. Recently, Sufriyana et al. (2020) has employed an oversampling method to deal with class imbalance problem in early detection of preeclampsia.

In addition to the lack of reliable methods for addressing imbalanced data, there is no study that has examined the preeclampsia prediction problem within different ethnic groups. This is despite the fact that race is a known risk-factor for preeclampsia (Boghossian et al., 2014), with African American women in particular being more at risk for developing preeclampsia and being more likely to die once it develops (Admon et al., 2018). None of the previous studies developed a granular prediction model for each race/ethnic groups individually. We hypothesize this might be because there is limited number of patients in these ethnic groups (African American/American Indian) in hospitals' datasets, which makes it difficult to train machine learning algorithms and generate meaningful results. Sandström et al. (2019) has developed their model on a dataset which includes 90% white women. Developing personalized prediction models for each race/ethnicity provides more accurate models that can be used by policy makers for resource allocation and better healthcare management.

	Imputation Method			Feature Selection			Class imbalance Method		
	Removal	EM	Mean Imput.	No FS	FFS	BFS	No CLM	OS	DS
Maric et al. (2020)			✓		✓		✓		
Sufriyana et al. (2020)	✓				✓			✓	
Sandstrom et al. (2019)	✓					✓	✓		
Moreira et al. (2017)	✓			✓			✓		
Kenny et al. (2014)	✓					✓	✓		
Simbolon et al. (2021)							✓		
Yu et al. (2005)						✓	✓		
Poon et al. (2009/2010)				✓			✓		
Odibo et al. (2011)				✓			✓		
Caradeux et al (2013)						✓	✓		
Parra-Cordero et al. (2013)				✓			✓		
Scazzocchio et al. (2013)	✓				✓		✓		
Wright et al. (2019)				✓			✓		
North et al. (2011)		✓				✓	✓		

Table 2.1: Previous Studies and their Imputation, Feature Selection, and Class Imbalance Methods.

	Supervised Learning Model									
	LR	EN	GB	DT	RF	SVM	ANN	NB	GM	EL
Maric et al. (2020)	✓	✓	✓							
Sufriyana et al. (2020)	✓			✓	✓	✓	✓			✓
Sandstrom et al. (2019)	✓				✓					
Moreira et al. (2017)				✓	✓			✓		
Kenny et al. (2014)	✓									
Simbolon et al. (2021)										✓
Yu et al. (2005)	✓									
Poon et al. (2009/2010)	✓									
Odibo et al. (2011)	✓									
Caradeux et al (2013)	✓									
Parra-Cordero et al. (2013)	✓									
Scazzocchio et al. (2013)	✓									
Wright et al. (2019)									✓	
North et al. (2011)	✓									

Table 2.2: Summary of early-onset PE prediction models. MImp.: Missing imputation technique, R: Removal technique, EM: Expected maximization, M: Mean Imputation, Feature Selec.: Feature selection method, Forward Feature Selection (FFS), Backward Feature Selection (BFS), No: No feature selection is used, Imb.: class imbalance method, No: No class imbalance method is used, OS: oversampling, Supervised Learn.: Supervised learning model, LR: Logistic regression, EN: Elastic Net, DT: Decision Tree, RF: Random Forest, SVM: Support vector machine, ANN: Artificial Neural Network, GB: Gradient Boosting, EL: Ensemble Learning, NB: Naïve Bayes, GM: Gaussian Model

	Performance Measures							
	ACC	AUC	PR	SN	G-mean	SP	FM	KS
Maric et al. (2020)		✓		✓				
Sufriyana et al. (2020)		✓	✓	✓		✓		
Sandstrom et al. (2019)		✓		✓				
Moreira et al. (2017)		✓	✓	✓			✓	✓
Kenny et al. (2014)		✓	✓	✓				
Simbolon et al. (2021)	✓						✓	
Yu et al. (2005)		✓		✓		✓		
Poon et al. (2009/2010)								
Odibo et al. (2011)		✓		✓				
Caradeux et al (2013)								
Parra-Cordero et al. (2013)				✓				
Scazzocchio et al. (2013)		✓						
Wright et al. (2019)		✓		✓				
North et al. (2011)		✓						

Table 2.3: Previous Studies and Performance Metrics Used.

Chapter 3

Methods

3.1 Artificial Neural Networks

Artificial Neural Networks (ANN) have their origin in the 1940s, with the McCulloch-Pitts Neuron (McCulloch, Pitts, 1943). The idea of "artificial neurons" is inspired by the human brain, in which a neuron takes "input" in the form of signals from surrounding cells, and will only activate in the form of an electrical spike if the combined signals passes a threshold level. An artificial neuron mimics this behavior by taking a series of features x, multiplying each by an individually chosen weight w, and then adds it to a bias term b before summing them together to calculate if a pre-defined threshold is met, which allows for classification.

$$h = \sum_{i=1}^{n} x_i w_i \tag{3.1}$$

$$f(x, w) = \begin{cases} 1 & \text{if } h \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

This early ANN suffers from a few major drawbacks. The trained model is essentially just a linear model, meaning that it cannot learn more complicated functions than a more basic model (e.g., linear regression or logistic regression). The second drawback is that the

weights and bias terms need to be selected by the programmer, leading to a model that would need to be hand-tuned for every problem.

Later versions of ANN adapted the artificial neuron to represent more complicated functions by linking them together into an multiplayer perceptron (MLP) or feedforward ANN (Gardner, Dorling, 1998). The MLP is typically composed of multiple layers, each layer containing a pre-defined number of neurons, or nodes. These layers can be subdivided into three separate types: the input layer, which takes each feature x as input; a number of hidden layers (the number of layers here denotes the depth of the network), which performs the previously seen linear computation on each input before passing the output to the next layer; and finally, the output layer, which returns the final prediction. Each node in a layer is connected to every node in the next layer, making a fully-connected neural network where the final prediction is made up of a functional composition of each layer. For example, a network with three layers could be described as:

$$\hat{y} = f^{(3)}(f^{(2)}(f^{(1)}(x)) \tag{3.3}$$

Where $\hat{y}$ is the predicted output, $f^{(1)}$ is the function of the first layer, $f^{(2)}$ is the function of the second layer, and $f(3)$ is the function of the third layer. These functions each take the form of:

$$f(x; w, b) = w * x + b \tag{3.4}$$

Where x are the input features, w are the weights of each node in the layer, and b are accompanying bias term. Furthermore, more modern versions of neural networks add non-linearity through the use of activation functions (Sharma, 2017) that will be discussed in section 3.1.2.

3.1.1 Backpropogation and Gradient Descent

The goal of a neural network is to approximate some function f^* that can map a series of inputs x to appropriate outputs y (Goodfellow et al., 2016). This is done by finding weights and biases such that minimize a cost function that represents the error between a predicted output and the true output. The earliest ANNs required the programmer to set the weights and biases. Later extensions of ANN, such as the perceptron, could learn better weights automatically from data (Rosenblatt, 1958, 1961). This process has been expanded upon by the addition of the back-propagation algorithm (Rumelhart et al., 1986), which allows for the weights to be updated by taking information from a given cost function and flowing backwards through the network in order to compute the gradient with respect to each of the weights and biases. This leads to the following update:

$$\theta = \theta - \eta \nabla Cost(\theta) \tag{3.5}$$

Where θ denotes the parameters w or b and η is the step size that controls the speed of learning. In this version of gradient descent, the cost is taken with respect to each sample in the dataset.

Updating the parameters based on the cost of misclassifying every sample can be computationally expensive however. In practice the model's parameters are usually updated on a random subset of data called a minibatch, the size of which is a hyperparameter chosen by the experimenter. If the batch size is less than the size of the dataset then the resulting method is known as Stochastic Gradient Descent (SGD) (Bottou, 1998).

Algorithm 1: Stochastic gradient descent (SGD)

Input: Learning rate ϵ
Input: Initial parameter θ
while *Stopping criterion not met* **do**
  Sample a minibatch of m examples from the training set $x^{(1)}, ..., x^{(m)}$ with corresponding targets $y^{(i)}$
  Compute gradient: $\hat{g} \leftarrow \frac{1}{m} \nabla_\theta \sum_i C(f(x^{(i)}; \theta), y^{(i)})$
  Update: $\theta \leftarrow \theta - \eta \hat{g}$
end

A possible disadvantage of SGD is that the magnitudes of the gradients can be very different for each parameter, making it difficult to choose a learning rate that will work for all of them. Root mean square propogation (RMSprop) (Tieleman, Hinton, 2012) improves this method by adapting the step size to the individual parameters and changing the gradient accumulation into an exponentially weighted moving average (Goodfellow et al., 2016).

Algorithm 2: RMSProp algorithm

Input: Global learning rate ϵ, decay rate ρ
Input: Initial parameter θ
Input: Small constant δ, usually 10^{-6}, used to stabilize division by small numbers
Initialize accumulation variables $r = 0$
while *Stopping criterion not met* **do**
  Sample a minibatch of m examples from the training set $x^{(1)}, ..., x^{(m)}$ with corresponding targets $y^{(i)}$
  Compute gradient: $g \leftarrow \frac{1}{m} \nabla_\theta \sum_i C(f(x^{(i)}; \theta), y^{(i)})$
  Accumulate squared gradient: $r \leftarrow \rho r + (1 - \rho) g \odot g$
  Compute parameter update: $\Delta\theta = -\frac{\epsilon}{\sqrt{\delta + r}} \odot g$.
  Apply update: $\theta \leftarrow \theta + \Delta\theta$
end

Adaptive Moment Estimation (ADAM) (Kingma, Ba, 2014) is another extended version of vanilla gradient descent, which takes momentum directly through estimating the first order moment of the gradient (Goodfellow et al., 2016). The method of the adaptive moments in calculation of the gradient will result in better classification accuracy as well as higher computational efficiency and convergence speed.

Algorithm 3: Adam algorithm

Input: Step size ϵ
Input: Exponential decay rates for moment estimates, ρ_1 and ρ_2 in the range $[0, 1)$
Input: Small constant δ, used to stabilize division by small numbers
Input: Initial parameters θ
Initialize 1st and 2nd moment variables $s = 0, r = 0$ Initialize time step $t = 0$ **while** *Stopping criterion not met* **do**
  Sample a minibatch of m examples from the training set $x^{(1)}, ..., x^{(m)}$ with corresponding targets $y^{(i)}$
  Compute gradient: $g \leftarrow \frac{1}{m} \nabla_\theta \sum_i C(f(x^{(i)}; \theta), y^{(i)})$
  $t \leftarrow t + 1$
  Update biased first moment estimate: $s \leftarrow \rho_1 s + (1 - \rho_1) g$
  Update biased second moment estimate: $r \leftarrow \rho_2 r + (1 - \rho_2) g \odot g$
  Correct bias in first moment: $\hat{s} \leftarrow \frac{s}{1 - p_1^t}$
  Correct bias in second moment: $\hat{r} \leftarrow \frac{r}{1 - p_2^t}$
  Compute update: $\Delta\theta = -\epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}$
  Update: $\theta \leftarrow \theta + \Delta\theta$
end

Adam can also be improved with the addition of Nesterov momentum (Dozat, 2016). In this version, rather than taking the gradient being computed from the current term, it takes a projected position calculated from changes in the last iteration and uses the derivative of this projection instead. This in effect minimizes the possibility of overshooting minimums caused by using a simple momentum.

Algorithm 4: NAdam algorithm

Input: Learning Rate: $\alpha_0, ..., \alpha_T$;Decay Factors: $\mu_0, ..., \mu_T$; v; ϵ: Hyperparameters
Input: Initial parameters θ
Initialize 1st and 2nd moment variables $s = 0, r = 0$ **while** *Stopping criterion not met* **do**
  Compute gradient: $g_t \leftarrow \nabla_{\theta_{t-1}} f_t(\theta_{t-1})$
  Update biased first moment estimate: $s \leftarrow \mu_t m_{t-1} + (1 - \mu_t)$
  Update biased second moment estimate: $r_t \leftarrow v r_{t-1} + (1 - v) g_t^2$
  Correct bias in first moment:
  $\hat{s} \leftarrow (\mu_{t+1} s_t / (1 - \prod_{i=1}^{t+1} \mu_i)) + ((1 - \mu_t) g_t / (1 - \prod_{i=1}^{t} \mu_i))$
  Correct bias in second moment: $\hat{r} \leftarrow v n_t / (1 - v^t)$
  Update: $\theta_t \leftarrow \theta_{t-1} - \frac{\alpha_t}{\sqrt{\hat{n}_t + \epsilon}} \hat{s}_t$
end

After one batch is fed into the network, the parameters are updated, and the next batch is fed in until all of the data has been seen by the model. One full iteration through the neural network by all the data is considered as an epoch. The number of epochs used in the model is another hyperparameter. We note that too few epochs may lead into the fact that the model does not effectively learn the data, and too many epochs may result into the risk of overfitting and reducing the model's ability to generalize to new unseen data (Goodfellow et al., 2016). However, this risk is mitigated when the size of data is increased.

3.1.2 Activation Functions

Whereas the first artificial neuron applied a simple weighted sum to determine if it would fire or not, a modern neuron applies a non-linear activation function instead. These activation functions are applied after the inputs are transformed by the weights and biases and before passing them onto the next layer's nodes. The most commonly-used activation function is the sigmoid activation given by

$$a(z) = \frac{1}{1 + e^{-z}} \tag{3.6}$$

Where z represents the linear output of the node. This transformation squashes the output to a value between 0 and 1 in a continuously differentiable smooth curve, with small outputs moving closer to 0 and large ones moving closer to 1. In this way, sigmoid activation functions mimic the all-or-nothing approach of earlier neurons. The downside of this particular activation function is that it can saturate, meaning that if the output is too large or small the gradient can become close to 0 which negatively affects the ability of the network to update the parameters. To overcome this issue, the sigmoid function has been improved through using the closely related hyperbolic tangent function given by

$$a(z) = \frac{2}{1 + e^{-2z}} - 1 \tag{3.7}$$

This function outputs values between -1 and 1, and has a significantly steeper gradient which makes it easier for training than using the sigmoid function. Another common activation function is the rectified linear unit (ReLU) function as follows

$$a(z) = max(0, z) \tag{3.8}$$

The ReLU activation function is represented by a threshold value at 0 which sets any output $z \leq 0$ value, while any output greater than 0 is linearly represented. ReLU has been found to converge faster than sigmoid or tanh (Krizhevsky et al., 2012) which makes the learning of a neural network more efficient. Thus, due to the ability to learn complex non-linear functions, neural networks have been used successfully in a variety of machine learning problems, such as image recognition (He et al., 2015; LeCun et al., 1998), machine translation (Cho et al., 2014; Bahdanau et al., 2014), speech recognition (Graves et al., 2013; Abdel-Hamid et al., 2014), weather forcasting (Maqsood et al., 2004), credit scoring (Blanco et al., 2013), cancer detection (Joshi et al., 2010; Karabatak, Ince, 2009; Yavuz et al., 2017), and more.

3.2 Cost-Sensitive Neural Networks

Despite the success of neural networks in a variety of applications, their use might be challenging due to the distribution of the given dataset. In classification, many machine learning algorithms, including neural networks, assume that the distribution of classes is roughly the same. When this assumption is violated, the neural network can best reduce the misclassification cost by simply outputting the majority class in every case. This results in a model with a high accuracy but with no ability to truly distinguish between classes (Leevy et al., 2018b).

Our proposed method removes this assumption which is based on cost-sensitive learning approach. In this approach which is originally proposed by Kukar (Kukar, Kononenko, 1998),

the cost function is modified such that different costs are associated with the true value of any given sample. In particular, we used two specific loss functions including weighted cross entropy and focal loss functions that we will explain them in the following sections 3.2.1 and 3.2.2

3.2.1 Weighted Cross Entropy Loss

In neural networks, the cross entropy (CE) loss function is usually used for binary classification problems which is defined by

$$CE(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{3.9}$$

Where $y \in \{\pm 1\}$ is the ground-truth class, $\hat{y}$ is the model's estimation of the class with label y = 1. This basic loss function can be modified by multiplying the cost of each individual sample by a class specific weight (Lin et al., 2017), which is so-called as the weighted cross entropy (WCE) defined by:

$$WCE(y, \hat{y}) = \begin{cases} -C^{+} \log(\hat{y}) & \text{if } y = 1 \\ -C^{-} \log(1 - \hat{y}) & \text{otherwise.} \end{cases} \tag{3.10}$$

Where $C^{+} = \frac{N}{2N^{+}}$ and $C^{-} = \frac{N}{2N^{-}}$ scales each cost by the number of samples within each class, and N^{+} and N^{-} is the size of the positive and negative class. In fact, different "importance" are given by the parameters C^{+} and C^{-} to the misclassification of samples in the minority (positive) and majority (negative) class. Accordingly, the error cost function will be

$$J(w, b) = -\frac{1}{N}(C^{+} \sum_{\{i|y_i=1\}}^{N^{+}} y_i \log(\hat{y_i}) + C^{-} \sum_{\{j|y_j=-1\}}^{N^{-}} (1 - y_j) \log(1 - \hat{y_j})) \tag{3.11}$$

3.2.2 Focal Loss

Recently Focal Loss (FL) has been proposed by Lin et al. (2017) which is another cost-sensitive CE loss function for binary classification. The main idea behind the FL is to focus training on hard samples while reducing the loss contribution from well-classified and easy samples through adding a modulating factor to the sigmoid CE loss.

Suppose the predicted output from the model for both classes are $\hat{y} = [\hat{y}_1, \hat{y}_2]^T$. The sigmoid function calculates the probability distribution for minority and majority classes as $p_t = \text{sigmoid}(\hat{y}_t) = 1/(1+ \exp(-\hat{y}_t))$ where p_t is defined as

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise.} \end{cases} \tag{3.12}$$

The focal loss can be formulated as:

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t) \tag{3.13}$$

or equivalently,

$$FL(y, \hat{y}) = -((1 - p)^{\gamma} \log(p) + p^{\gamma} \log(1 - p)) \tag{3.14}$$

where $y \in \{\pm 1\}$ is the ground-truth class and $p_t \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$. The parameter $\gamma \geq 0$ should be tuned. The modulating factor $(1 - p_t)^{\gamma}$ is added which reduces the loss contribution from easy examples. We note that FL is equivalent to CE, when $\gamma = 0$. The effect of the modulating factor increases as the γ parameter increases (Lin et al., 2017).

In addition, an α-balanced variant of the original focal loss has been developed to further

focus on the effective number of samples. The parameter $\alpha_t \in [0, 1]$ is defined as

$$\alpha_t = \begin{cases} \alpha & \text{if } y = 1 \\ 1 - \alpha & \text{otherwise.} \end{cases} \tag{3.15}$$

The α-balanced CE loss is then written as

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{3.16}$$

Or equivalently

$$FL(y, \hat{y}) = -(\alpha(1 - p)^\gamma \log(p) + (1 - \alpha) p^\gamma \log(1 - p)) \tag{3.17}$$

In this work, We use weighted cross entropy and focal loss functions.

3.3 Balanced-Batches

For comparison, a final method of balancing the batches was applied to each of the loss functions. As the neural network is fed batches of data the batch is checked to see how imbalanced it is, and a sampling technique is applied to the batch to ensure that the classes are equal. In this work we applied random oversampling, which samples from the preeclamptic patients with replacement. Balanced Batches were applied using the imblearn library. (Lemaître et al., 2017)

3.4 Hyperparameter Optimization Strategies

Neural Networks contain a large number of hyperparameters that are needed to be set prior training such as learning rate, depth, the number of nodes per layer, activation functions, weight initialization strategy, and more. Therefore, it is beneficial to apply an algorithmic

strategy to find the best performing combination of hyperparameters. In this work, three hyperparameter strategies are tested: Random Search, Bayesian Optimization, and Hyperband Optimization. All hyperparameter tuning is performed using the Keras Tuner library (O'Malley et al., 2019).

3.4.1 Random Search

Random search (Bergstra, Bengio, 2012) is the most common hyperparameter search method in the literature of deep learning. which consists of finding a search space, randomly sampling points within the space, and testing out various configurations of the neural network regardless of the results from previous iterations. However, this method is useful for space explorations and often results into good performance in several real-world problems, but it does not offer computationally efficient solutions. Furthermore, random search does not take into account the history of previous searches or surrounding search space, thus it does not avoid the risk of being stuck in local minima points and does not necessarily obtain an architecture that is close to the global minimum (Bergstra, Bengio, 2012).

Algorithm 5: Random Search

Input: Number of trials to test N
Initialize with random hyperparameter configuration T
Initialize best cost, C_{best} with cost of current T
$T_{Best} = T$
$n = 0$
while $n < N$ **do**
    Calculate cost C given T
    if $C < C_{Best}$ **then**
        $C_{Best} = C$
        $T_{Best} = T$
    end
    Sample new T
    Increment n
end

3.4.2 Bayesian Optimization

Bayesian optimization (Snoek et al., 2012) takes a more efficient approach in order to find the maximum or minimum of an objective function, unlike random search. The objective function consists of the hyperparameters that minimizes the neural network loss function. Since evaluating the objective function can be expensive, Bayesian optimization instead approximates the objective function using a probabilistic model, called a surrogate function. Most often, a Gaussian Process is used as the surrogate function, which is used to place a prior on our objective function f.

$$p(f) = GP(f; \mu, K) \tag{3.18}$$

Where μ is the mean function and K is a covariance function or kernel. A few initial hyperparameters are first chosen and evaluated to provide the Gaussian Process with data, which then provides a posterior probability distribution for potential $f(x)$ given candidate hyperparameters x.

$$p(f|Data) = GP(f; \mu_{f|Data}, K_{f|Data}) \tag{3.19}$$

An acquisition function is used to determine where to evaluate the surrogate function. The outcomes are used to update the surrogate function. The acquisition function used in our work is set as the Upper Confidence Bound. When this function is maximized, the sampling of hyperparameters is determined.

$$a_{UCB}(x; \beta) = \mu(x) + \beta\sigma(x) \tag{3.20}$$

Where $\beta > 0$ is a hyperparameter that defines how aggressively to explore the search space and $\sigma(x)$ is the marginal standard deviation of $f(x)$. In this way, Bayesian Optimization keeps tracks of previous searches and evaluates only the permutation of hyperparameters which are likely to improve the model.

Using notation inspired from Frazier (2018), the Bayesian Optimization that we used in our work is represented in Algorithm 5.

Algorithm 6: Bayesian Optimization

Input: Budget R, reduction proportion η
Initialize Gaussian process by calculating cost for randomly selected hyperparameters T
while *stopping criterion not met* **do**
    Update posterior probability distribution on cost using all available data
    Use acquisition function to acquire T most likely to minimize cost
    Calculate cost function given T
    Increment n
end

3.4.3 Hyperband

Hyperband optimization algorithm has been recently presented by Li et al. (2018) for hyperparameter optimization. This method is appropriate for a variety of deep-learning and kernel-based learning problems due to its accelerated speedup compared to the existing search algorithms. It utilizes successive halving algorithms to allocate a budget to a set of hyperparameter configuration, then samples an assortment of randomly chosen hyperparameters, and finally evaluates the results and discards the worst performing subset before continuing training. This process is repeated until only one model is left. However, successive halving suffers from a resource allocation problem. Given a finite budget B (for example, amount of training time), it is unclear whether it should consider many configurations (large n) with a small training time, or a smaller number of configurations (small n) with a longer average training time. Therefore, Hyperband addresses this problem by considering several possible values of n for a fixed B and performs a grid search over feasible values of n. Since Hyperband performs early stopping on worse performing selections of features, it is a much more efficient way to explore the large feature space.

Algorithm 7: Hyperband Algorithm

Input: Budget R, reduction proportion η
Initialize $s_{max} = \lfloor \log_\eta(R) \rfloor, B = (s_{max} + 1)R$
for $s\{s_{max}, s_{max} - 1, ..., 0\}$ **do**
  $n = \lceil \frac{B}{R} \frac{\eta^s}{s+1} \rceil, r = R\eta^{-s}$
  // Begin successive halving with (n,r) inner loop
  T = Randomly selected hyperparameter configuration
  for $i\{0, ..., s\}$ **do**
    $n_i = \lfloor n\eta^{-i} \rfloor$
    $r_i = r\eta^i$
    Run algorithm and return associated costs C
    T = Top k performing hyperparameters given $(T, C, \lfloor n_i/\eta \rfloor)$
  end
end

In this study, η is set to 3. The resource budget consists of the maximum number of iterations through algorithm and the maximum number of epochs. In our solution setting, this is set to 5 iterations with 100 epochs.

3.5 Performance Measures

The most commonly-used performance measures in binary classification tasks are calculated from the confusion matrix (Table 3.1).

Table 3.1: Confusion matrix for binary classification problems

		Predicted Value	
		PE	**Non-PE**
Actual Value	**PE**	True Positive	False Negative
	Non-PE	False Positive	True Negative

The numbers of true positives (TP) represents as the number of preeclamptic (PE) pa-

tients correctly classified while true Negatives (TN) is the number of non-preeclamptic (Non-PE) patients classified as Non-PE. The numbers of false positives (FP) is defined as the Non-PE patients classified as PE and false negatives (FN) represents PE patients classified as Non-PE.

The most common metric in classification tasks is accuracy, which measures the total correctly classified samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.21}$$

However, this metric is very sensitive to the size of majority class (Non-PE) and is likely to obtain a misleadingly high accuracy dominated by the majority class pattern while the minority class samples are most likely misclassified. Since accuracy alone does not take into account the imbalanced nature of the problem, we relied on several additional metrics such as precision, recall (sensitivity), specificity, G-mean, and area under the curve (AUC).

Precision measures the amount of positive values that are actually positive, while recall (or sensitivity) measures what percentage of the positive cases were captured by the model. Specificity refers to the percentage of the negative examples that are truly negative. Additionally, we report G-mean, which takes into account both the specificity and sensitivity, as well as the area under the curve (AUC), which measures the balance between the correctly classified positive samples (TP) and incorrectly classified negative samples (FP). The performance metrics are written as

$$Precision = \frac{TP}{TP + FP} \tag{3.22}$$

$$Recall/Sensitivity = \frac{TP}{TP + FN} \tag{3.23}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.24}$$

$$G - mean = \sqrt{Sensitivity * Specificity} \tag{3.25}$$

Chapter 4

Data

4.1 Texas Data Exploration

In this work, we used the 2013 Texas Inpatient Public Use Data File (PUDF), which includes a combination of both demographic and clinical information for inpatients in the Texas hospital system.

In particular, the PUDF contains demographic and clinical information of patients who were discharged from state-licensed facilities and hence it was required to report their data to the Texas Health Care Information Collection. Each record in the PUDF consists of a series of diagnosis codes. These codes come from the *International Classification of Diseases, Ninth revision, Clinical Modification* (ICD-9-CM)World Health Organization (1978). The admitting, principal, and 24 other possible diagnoses were examined for each patient. The records of women who delivered in-hospital were identified by searching each of the selected diagnosis fields for an ICD-9-CM code beginning with V27 (Outcome of Delivery). Table 4.1 provides the statistics of demographic features used in our model. This set of features consists of each patient's ethnicity (Hispanic and Non-Hispanic), race (White, African American, Native American, or Other), insurance (Medicaid, Medicare, Self-pay or Charity, or Other), age (in years), and whether or not the patient lives in a county in the border of Mexico. We

note that the definition of border county comes from the Texas Department of State Health Services (2021). The frequency (the number of patients) of each feature's values along with percentage of the population inside the parenthesis are shown in this table.

Table 4.1: Patient Demographic Attributes in the Texas Dataset

Feature	**Value**	**Frequency**
Ethnicity	Hispanic	150,031 (41.570%)
	Non-Hispanic	207,494 (57.490%)
Race	White	195,149 (54.100%)
	African American	41,168 (11.400%)
	Native American	1,214 (0.340%)
	Asian or Pacific Islander	13,139 (3.640%)
	Other	109,395 (30.300%)
Border County	Yes	44,989 (12.460%)
	No	315,954 (87.540%)
Insurance	Medicaid	185,010 (51.250%)
	Medicare	2,543 (0.700%)
	Self-pay or Charity	31,903 (8.84%)
	Other	176,312 (48.840%)
Discharge Date	Quarter 1	85,161(0.236%)
	Quarter 2	85,768(0.238%)
	Quarter 3	95,992(0.266%)
	Quarter 4	94,022(0.260%)
Age (years)	10-14	505 (0.140%)
	15-17	11,120 (3.08%)
	18-19	24,317 (6.740%)
	20-24	91,287 (25.290%)
	25-29	101,109 (28.010%)
	30-34	84,728 (23.470%)
	35-39	38,760 (10.740%)
	40-44	8,593 (2.380%)
	45-49	484 (0.130%)
	50-54	40 (0.010%)

Figure 4.1 shows that the majority of the patients are in the age range between 20 and 34. However the prevalence of preeclampsia across age groups shows a u-shaped distribution with the most at-risk patients in the range 45-49, followed by patients of ages 40-44 and 10-14.

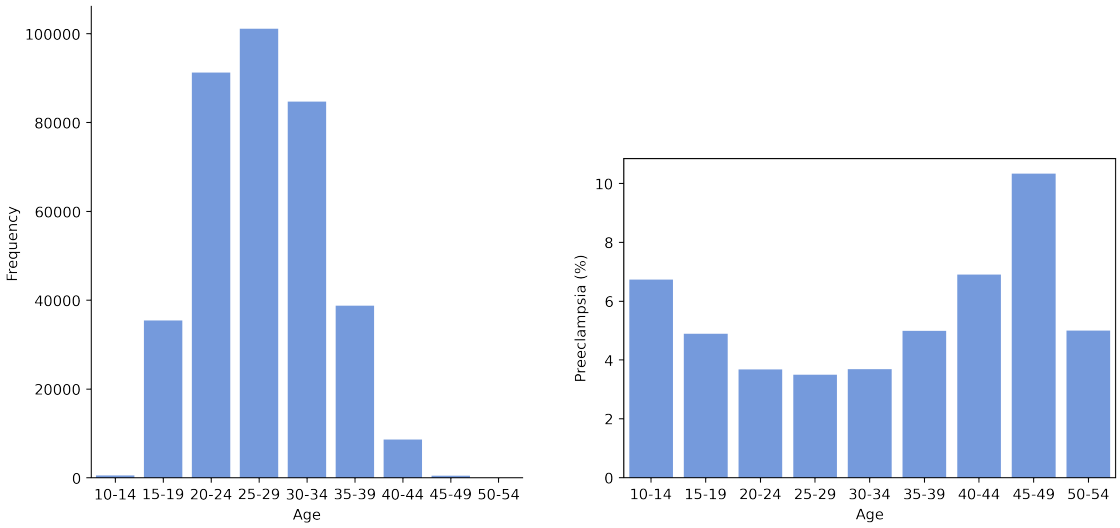


Figure 4.1: Left: The Distribution of Age Groups in the Texas Dataset; Right: The Prevalence of Preeclampsia Among Each of these Age Groups

Table 4.2 shows the breakdown of ethnicity by race. The majority of the Hispanic population was identified as either White or "Other Race." Asians were the least likely to be identified as Hispanic, followed by African Americans. The most likely ethnicity to have missing race information was the Non-Hispanic population. Additionally, if a patient had missing race value, they were also more likely to have missing values in ethnicity attribute.

Table 4.2: Race and Ethnicity Characteristics in the Texas Dataset [Frequency (Percentage)]

Race	**Hispanic**	**Non-Hispanic**	**Missing Ethnicity**
African American	1,125 (0.027%)	39,743 (0.965%)	300 (0.007%)
Native American	390 (0.321%)	805 (0.663%)	19 (0.016%)
Asian or Pacific Islander	333 (0.025%)	12,676 (0.965%)	130 (0.010%)
White	59,500 (0.305%)	134,246 (0.688%)	1,403 (0.007%)
Other Race	88,505 (0.809%)	19,384 (0.177%)	1,506 (0.014%)
Missing Race	178 (0.203%)	640 (0.729%)	60 (0.068%)

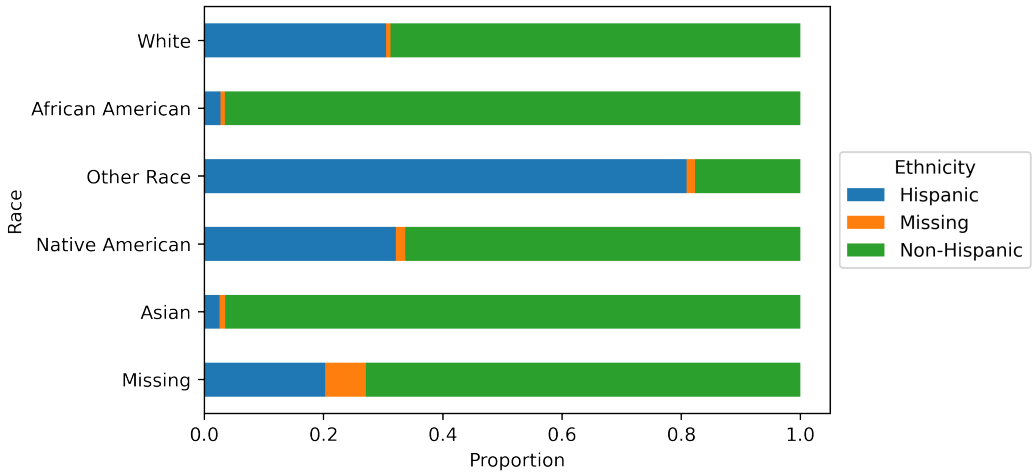


Figure 4.2: Breakdown of Ethnicity by Race in the Texas Dataset

The Texas dataset contained 360,943 women in total who delivered at hospital. Of those, 14,375 (3.98%) had preeclampsia. Table 4.3 breaks down the occurrence of preeclampsia by race. Notably, African American Hispanic patients had a higher incidence of preeclampsia with a frequency of 9.51% (as a proportion of population).

Table 4.3: Distribution of Preeclamptic Patient among Race/Ethnic Groups in the Texas Dataset

Race	**Ethnicity**	**Total Preeclamptic**
White	Hispanic	2461 (4.14%)
	Non-Hispanic	5117 (3.81%)
African American	Hispanic	107 (9.51%)
	Non-Hispanic	2118 (5.33%)
Native American	Hispanic	16 (4.10%)
	Non-Hispanic	25 (3.10%)
Asian or Pacific Islander	Hispanic	4(1.20%)
	Non-Hispanic	289 (2.28%)
Other Race	Hispanic	3464 (3.91%)
	Non-Hispanic	665 (3.431%)

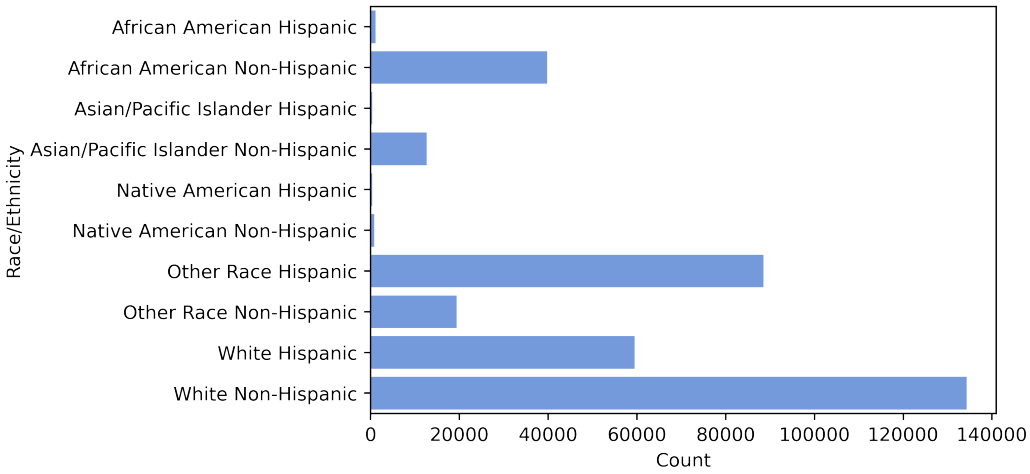


Figure 4.3: The Number of Patients of Each Race and Ethnicity Present in the Texas Dataset

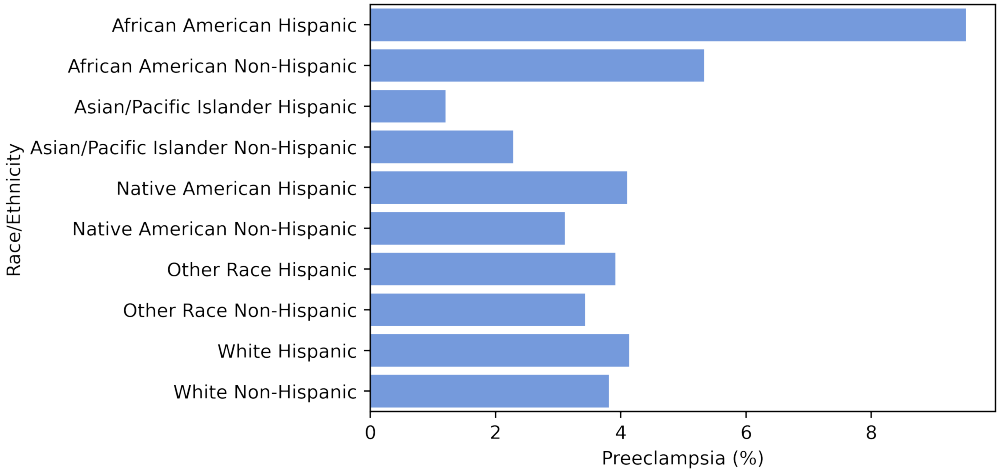


Figure 4.4: The Rate of Preeclampsia per Race in the Texas Dataset

According to this dataset, the preeclamptic women were more likely to have prolonged lengths of stay. The majority of women without preeclampsia stayed in the hospital only 2.5 days on average, while the women with preeclampsia stayed longer in the hospital, 3 to 4 days on average.

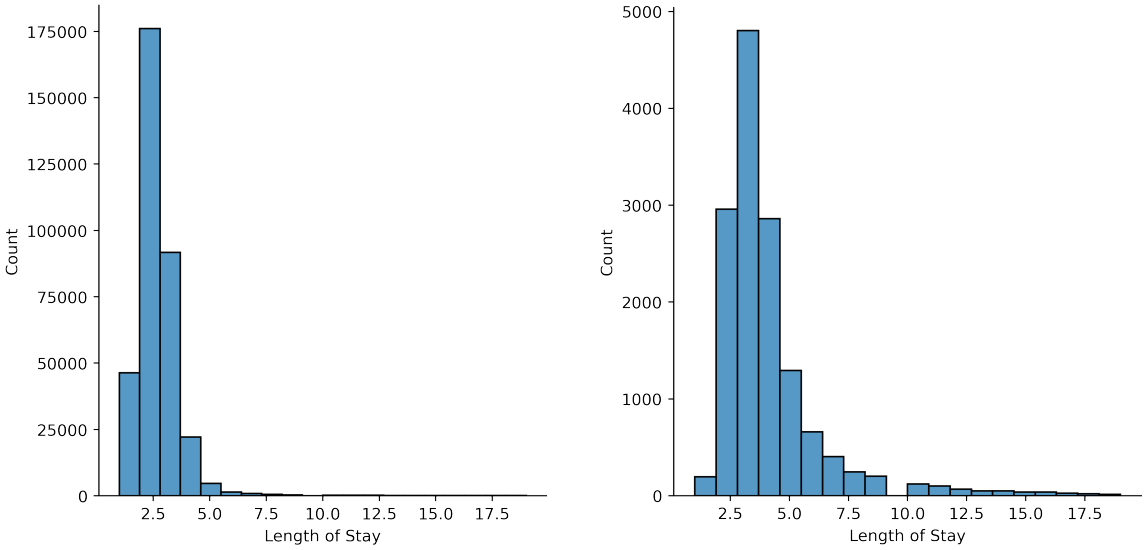


Figure 4.5: Distribution of Length of Stay - Left: Women without Preeclampsia; Right: Women with Preeclampsia

When the length of stay is broken down by race, we observed that African American patients (both Hispanic and non-Hispanic) had longer average stays in the hospital. African American Hispanic patients with preeclampsia stayed almost a full day longer on average than African American Non-Hispanic, and more than two days longer than White Hispanic patients.

Table 4.4: Length of stay (days) by race/ethnicity for patients without preeclampsia. We report the average (Avg), standard deviation (SD), minimum (min), first quartile (Q1), median, third quartile (Q3), and maximum (max) values.

Race	**Avg**	**SD**	**Min**	**Q1**	**Median**	**Q3**	**Max**
African American Hispanic	3.2	3.8	1.0	2.0	3.0	3.0	62.0
African American Non-Hispanic	2.8	3.1	1.0	2.0	2.0	3.0	113.0
Asian/Pacific Islander Hispanic	2.2	0.9	1.0	2.0	2.0	3.0	7.0
Asian/Pacific Islander Non-Hispanic	2.7	2.8	1.0	2.0	2.0	3.0	96.0
Native American Hispanic	2.7	1.9	1.0	2.0	2.0	3.0	19.0
Native American Non-Hispanic	2.6	1.6	1.0	2.0	2.0	3.0	26.0
Other Race Hispanic	2.4	2.1	1.0	2.0	2.0	3.0	106.0
Other Race Non-Hispanic	2.5	1.9	1.0	2.0	2.0	3.0	61.0
White Hispanic	2.2	1.9	1.0	2.0	2.0	3.0	95.0
White Non-Hispanic	2.6	2.9	1.0	2.0	2.0	3.0	365.0

Table 4.5: Length of stay (days) by race/ethnicity for patients with preeclampsia. We report the average (Avg), standard deviation (SD), minimum (min), first quartile (Q1), median, third quartile (Q3), and maximum (max) values.

Race	**Avg**	**SD**	**Min**	**Q1**	**Median**	**Q3**	**Max**
African American Hispanic	5.8	4.8	2.0	3.0	4.0	6.0	37.0
African American Non-Hispanic	5.0	6.0	1.0	3.0	4.0	5.0	107.0
Asian/Pacific Islander Hispanic	3.5	1.7	2.0	2.8	3.0	3.8	6.0
Asian/Pacific Islander Non-Hispanic	5.0	5.7	1.0	3.0	3.0	5.0	58.0
Native American Hispanic	4.3	3.7	1.0	2.0	3.0	4.3	13.0
Native American Non-Hispanic	4.0	3.8	1.0	2.0	3.0	4.0	21.0
Other Race Hispanic	4.1	4.4	1.0	2.0	3.0	4.0	93.0
Other Race Non-Hispanic	4.2	3.9	1.0	3.0	3.0	4.0	37.0
White Hispanic	3.8	3.0	1.0	2.0	3.0	4.0	44.0
White Non-Hispanic	4.6	4.7	1.0	3.0	3.0	5.0	105.0

4.2 Oklahoma Data Exploration

The next datasets we used are the 2017 and 2018 Oklahoma Inpatient PUDF. Unlike the 2013 Texas dataset, this dataset has employed the updated ICD-10-CM diagnosis codes instead of the ICD-9-CM codesWorld Health Organization (2004). The women who delivered in hospital are filtered based on the presence of codes beginning with Z37 (Delivery outcome). These datasets contained a total of 84,632 women who delivered at hospital, of which 4721 (4.48%) had preeclampsia. Table 4.6 shows the demographic attributes of the Oklahoma dataset. The frequency (the number of patients) of each feature's values along with percentage of the population inside the parenthesis are shown in this table. Unlike the Texas dataset, no data on ethnicity was collected for each patient, but there are additional attributes such as marital status and month of admission. There were no records indicating the delivery date for each patient in this data. So, we used the month of admission to estimate the month of delivery for each patient.

Table 4.6: Patient Demographic Attributes in the Oklahoma Dataset

Feature	**Value**	**Frequency**	**Feature**	**Value**	**Frequency**
Race	White	55,815 (65.950%)	Month of Delivery	Jan	7,148 (8.446%)
	African American	8,510 (10.055%)		Feb	6,418 (7.583%)
	Native American	5,443 (6.431%)		Mar	6,947 (8.208%)
	Other	14,864 (17.563%)		Apr	6,537 (7.724%)
Marital Status	Married	37,038 (43.764%)		May	7,242 (8.557%)
	Unmarried	32,579 (38.495%)		Jun	7,031 (8.308%)
Age group	10-14	71 (0.0838%)		Jul	7299 (8.624%)
	15-19	6,192 (7.316%)		Aug	7,699 (9.097%)
	20-24	21,831 (25.795%)		Sep	7,183 (8.487%)
	25-29	26,708 (31.559%)		Oct	7,371 (8.709%)
	30-34	20,115 (23.768%)		Nov	6,872 (8.120%)
	35-39	8,164 (9.646%)		Dec	6,885 (8.135%)
	40-44	1,458 (1.723%)			
	45-49	84 (0.099%)			
	50-54	9 (0.011%)			
Insurance	Medicaid	42,192 (0.499%)			
	Medicare	450 (0.005%)			
	Self-Pay	916 (0.011%)			
	Other Insurance	41,071 (0.485%)			

Figure 4.6 shows the distribution of patients' age groups and the prevalence of preeclampsia among them. Similar to the Texas dataset, most of the patients were in the age range of 20-34. The prevalence represents a U-shaped curve, with the youngest and oldest patients being as the most at-risk patients.

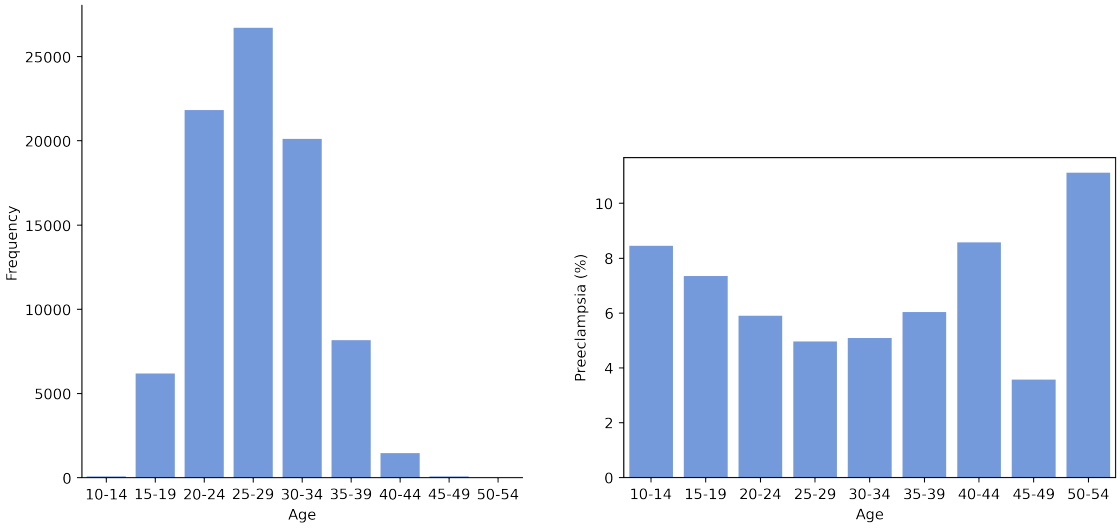


Figure 4.6: Left: The distribution of age groups in the Oklahoma dataset; Right: The prevalence of Preeclampsia among each of these age groups

Table 4.7 shows the frequency of preeclampsia between each racial groups in the Oklahoma dataset, while figure 4.8 shows the absolute number of patients in each racial category and their respective prevalence of preeclampsia. Despite White patients contributing the overwhelming majority of patients in the dataset, we observed that Native Americans and African Americans have the highest prevalence. In particular, Native Americans are almost twice the "Other" race.

Table 4.7: Rate of Preeclampsia by Race in the Oklahoma Dataset

Race	**Total Preeclamptic**
White	3008 (5.39%)
African American	551 (6.57%)
Native American	446 (8.19%)
Other Race	716 (4.82%)

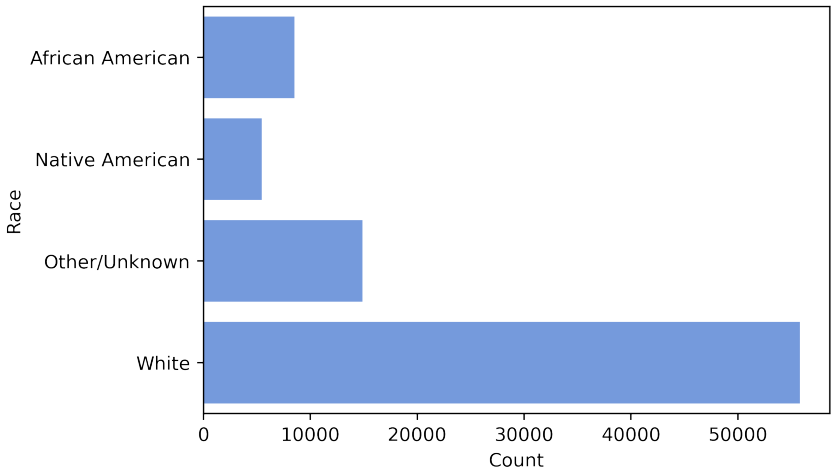


Figure 4.7: The Number of Patients within Each Race in the Oklahoma Dataset

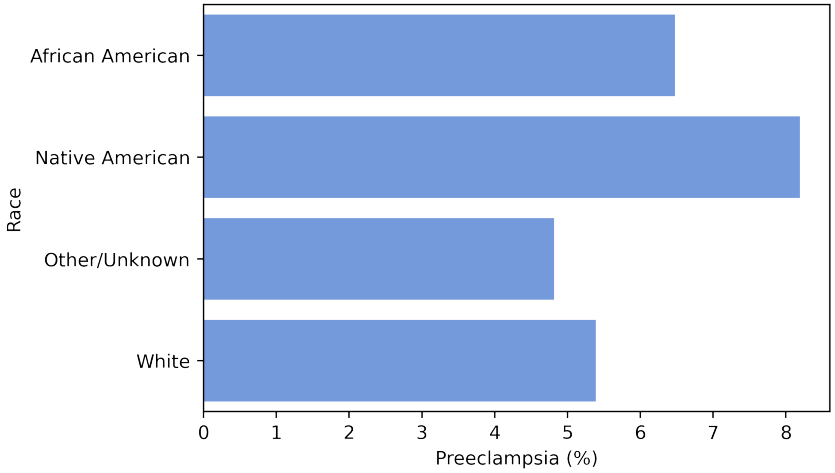


Figure 4.8: The Number of Preeclamptic Patients Within Each Race in the Oklahoma Dataset

Similar to the Texas dataset, the average length of stay is longer for patients with preeclampsia compared to those without preeclampsia. The average length of stay for those without preeclampsia is 2.4 days while for those with preeclampsia, the average length of stay is 4.0 days.

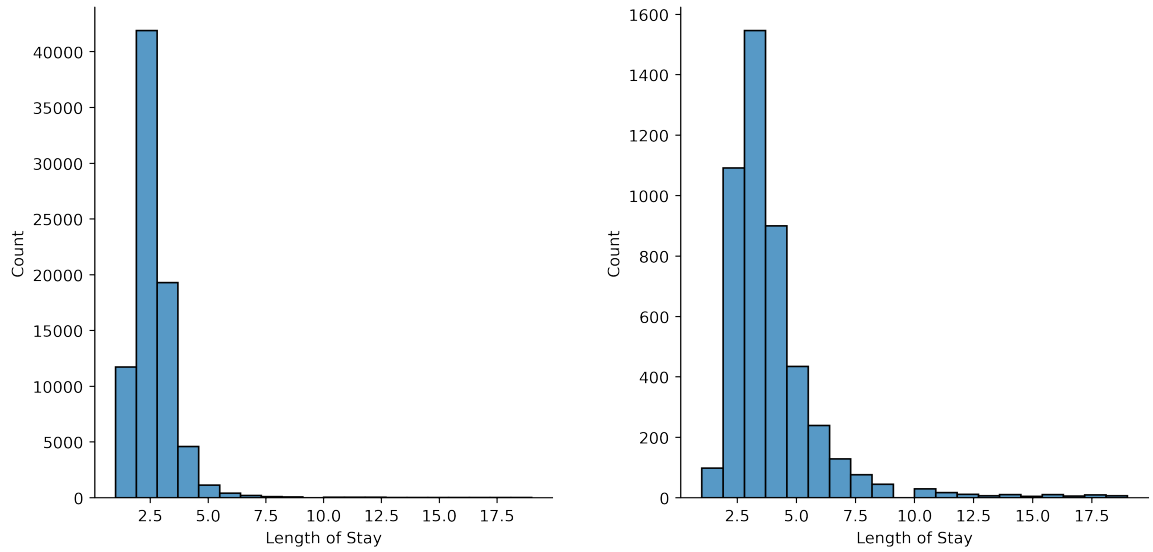


Figure 4.9: Distribution of Length of Stay - Left: Women without Preeclampsia; Right: Women with Preeclampsia

Th length of stay is also varied greatly among patients depending on their race. We observed that the African and Native Americans stay longer in the hospital in contrast to their white and other racial counterparts.

Table 4.8: Length of stay by race for patients without preeclampsia. We report the average (Avg), standard deviation (SD), minimum (min), first quartile (Q1), median, third quartile (Q3), and maximum (max) values.

Race	**Avg**	**SD**	**Min**	**Q1**	**Median**	**Q3**	**Max**
African American	2.6	2.7	1.0	2.0	2.0	3.0	96.0
Native American	2.6	2.8	1.0	2.0	2.f0	3.0	81.0
Other/Unknown	2.4	1.9	1.0	2.0	2.0	3.0	60.0
White	2.4	2.5	1.0	2.0	2.0	3.0	367.0

Table 4.9: Length of stay by race for patients with preeclampsia. We report the average (Avg), standard deviation (SD), minimum (min), first quartile (Q1), median, third quartile (Q3), and maximum (max) values.

Race	**Avg**	**SD**	**Min**	**Q1**	**Median**	**Q3**	**Max**
African American	4.2	3.3	1.0	3.0	3.0	5.0	35.0
Native American	4.5	4.4	1.0	3.0	3.0	5.0	57.0
Other/Unknown	3.8	3.7	1.0	2.0	3.0	4.0	57.0
White	4.0	3.9	1.0	2.0	3.0	4.0	84.0

4.3 Clinical Features and Feature Selection

The initial set of clinical features is selected based on the literature as well as our clinical collaborator opinion, Dr. Zuber Mulla. We modelled each clinical feature of each patient as a binary feature based on the presence of corresponding ICD-9-CM/ICD-10-CM codes in any of the diagnosis columns. So, we set the value of the feature equal to 1 if the corresponding ICD-9-CM/ICD-10-CM codes are present in the diagnosis columns otherwise the value would be set to zero. Table 4.10 shows the total samples and percentage of the population which contains the selected clinical features. The overwhelming majority of patients do not have many of the clinical diagnoses which leads to an incredibly sparse dataset.

Table 4.10: Patient Clinical Characteristics in the Texas and Oklahoma Datasets

Feature	**Frequency**	
	Texas	**Oklahoma**
Obesity	19,208 (5.322%)	7,136 (8.432%)
PRA*	615 (0.170%)	32 (0.038%)
Cocaine dependence	0 (0.000%)	67 (0.079%)
Amphetamine dependence	0 (0.000%)	962 (1.137%)
Gestational diabetes mellitus	21,658 (6.000%)	5,025 (5.938%)
Pre-existing diabetes mellitus	4,065 (1.126%)	1,159 (1.370%)
Anxiety	2,709 (0.751%)	3,148 (3.720%)
Anemia NOS	29,280 (8.112%)	11 (0.013%)
Iron deficiency anemia	3,937 (1.091%)	1246 (1.472%)
Other anemia	94 (0.03%)	12,784 (15.105%)
Depression	3,157 (0.875%)	2,752 (3.252%)
Primigravida*	4,969 (1.377%)	1796 (2.122%)
Hemorrhagic Disorders*	6 (0.002%)	0 (0.0%)
Systemic lupus erythematosus	366 (0.101%)	141 (0.167%)
Lupus erythematosus	20 (0.006%)	35 (0.041%)
Autoimmune Disease*	18 (0.005%)	9 (0.012%)
Pure hypercholesterolemia	108 (0.030%)	25 (0.030%)
Unspecified vitamin D deficiency	227 (0.063%)	189 (0.223%)
Proteinuria	21 (0.006%)	166 (0.196%)
Tobacco use disorder	6,140 (1.701%)	-
History of tobacco use	3,226 (0.894%)	-
Current Smoker	-	5,438 (6.426%)
Hypertension	2,424 (0.672%)	10276 (12.142%)

Continued on next page

Table 4.10 – *Continued from previous page*

Hypertensive heart disease	16 (0.004%)	5 (0.006%)
Chronic venous hypertension	1 (0.0003%)	1 (0.001%)
Unspecified renal disease*	644 (0.178%)	546 (0.645%)
Chronic kidney disease	173 (0.048%)	73 (0.086%)
Hypertensive kidney disease	96 (0.027%)	24 (0.028%)
Hypertensive heart and CKD*	6 (0.002%)	2 (0.002%)
Renal failure not elsewhere classified	6 (0.002%)	0 (0.000%)
Infections of GU* tract in pregnancy	3299 (0.914%)	618 (0.730%)
UTI*	1838 (0.509%)	175 (0.207%)
History of Trophoblastic Disease	0 (0.000%)	390 (0.461%)
Supervision of pregnancy, trophoblastic*	28 (0.008%)	11 (0.013%)
Thrombophilia	1,073 (0.297%)	271 (0.320%)
History of premature delivery	180 (0.050%)	149 (0.176%)
Hemorrhage in early pregnancy	216 (0.060%)	22 (0.026%)
Congenital abnormalities of the uterus*	1,184 (0.328%)	17,082 (20.184%)
Multiple gestations	5,871 (1.627%)	1,393 (1.646%)
Fetal growth restriction	3 (0.001%)	1 (0.001%)
Asthma	7,124 (1.974%)	3,547 (4.1911%)
Obstructive sleep apnea	106 (0.029%)	58 (0.0685%)
Other cardiovascular diseases*	1,372 (0.380%)	46 (0.0544%)
Sickle cell disease	75 (0.021%)	284 (0.3356%)
Thyroid disease	8,880 (2.460%)	2,750 (3.249%)
Inadequate prenatal care	8,959 (2.482%)	767 (0.906%)
Periodontal disease	35 (0.010%)	2 (0.002%)
Preeclampsia/Eclampsia	14,375 (3.983%)	4,721 (5.578%)

*PRA:Pregnancy resulting from assisted reproductive technology, UTI: Urinary Tract Infection, Unspecified Renal Disease: Unspecified renal disease in pregnancy without mention of hypertension, Supervision of pregnancy, trophoblastic: Supervision of high-risk pregnancy with history of trophoblastic disease, Congenital Abnormalities of the Uterus: Congenital abnormalities of the uterus including those complicating pregnancy, childbirth, or the puerperium, Other Cardiovascular Diseases: Other cardiovascular disease complicating pregnancy and childbirth, or the puerperium, CKD: Chronic Kidney Disease, GU: Genitourinary, Primigravida: Primigravida at the extremes of maternal age,Hemorrhagic Disorders: Hemorrhagic Disorders due to intrinsic circulating antibodies, Autoimmune Disease: Autoimmune Disease not elsewhere classified

Table 4.11 shows the total number of patients with preeclampsia/eclampsia in the dataset. We observed that around 4% of the Texas patients and 5% of the Oklahoma patients developed preeclampsia/eclampsia, meaning all datasets are highly imbalanced.

Table 4.11: Preeclampsia/Eclampsia and their Frequency Among Patients under Study

	Texas	**Oklahoma**
Preeclamptic	14,376 (3.98%)	4,721 (5.58%)
Non-Preeclamptic	346,567 (96.02%)	79,911 (94.42%)

The ranking of the top 20 features of the Texas dataset is given by the Chi-squared feature selection which is shown in Table 4.12. There is overlap in many of the features, especially Hypertension, which is the highest ranking feature in the Full and African American datasets and second highest in the Native American. Additionally obesity and pre-existing diabetes appear as the second and third most important features respectively in the Full and African American datasets. However, there are several features in the sub-populations that are not seen as important as in the Full dataset, such as renal disease, thyroid disease, anemia, etc.

Table 4.12: Feature rankings derived by the Chi-squared scores in the Texas dataset in general (full dataset) as well as among only African American and only Native American populations. Each column consists of feature ranking based on their importance with the associated Chi-squared scores in parenthesis.

Features	**Full**	**African American**	**Native American**
Hypertension	1 (22,024.9)	1 (3733.5)	2 (24.6)
Obesity	2 (1,292.5)	2 (182.4)	-
Pre-existing Diabetes Mellitus	3 (1180.0)	3 (133.6)	6 (5.5)
Multiple Gestations	4 (741.0)	5 (64.2)	5 (8.6)
Gestation Diabetes Mellitus	5 (528.8)	4 (65.5)	12 (1.3)
UTI	6 (237.1)	10 (30.8)	-
Obstructive Sleep Apnea	7 (224.7)	11 (23.4)	4 (13.0)
Infections of Genitourinary Tract in Pregnancy	8 (222.0)	9 (37.9)	20 (0.3)
Chronic Kidney Disease	9 (215.2)	6 (57.3)	-
Hypertensive Kidney Disease	10 (207.8)	-	-
Ages 40+	11 (197.4)	7 (47.9)	-
Primigravida*	12 (149.4)	18 (11.9)	18 (0.3)
African American Non-Hispanic	13 (144.4)	-	-
Anemia NOS	14 (122.9)	-	-
Other cardiovascular diseases*	15 (122.6)	8 (47.2)	-
Asthma	16 (96.6)	12 (23.0)	-
Anxiety	17 (86.3)	-	8 (4.1)
African American Hispanic	18 (82.6)	-	-
Asian/Pacifc Islander Non-Hispanic	19 (81.0)	-	-
Ages 10-19	20 (66.1)	-	10 (1.9)
Hispanic	-	13 (22.8)	-
Unspecified Renal Disease*	-	14 (22.8)	-
Thyroid Disease	-	15 (18.1)	8 (4.1)
Renal Failure*	-	16 (17.4)	-
Hypertensive Heart and Chronic Kidney Disease	-	17 (17.4)	-
Ages 20-29	-	19 (10.7)	11 (1.6)
Iron Deficiency Anemia	-	20 (8.8)	3 (13.5)
Pure Hypercholesterolemia	-	-	1 (27.9)
On Border	-	-	9 (2.5)
Inadequate Prenatal Care	-	-	14 (0.8)
Tobacco Use Disorder	-	-	15 (0.7)
Ages 30-39	-	-	16 (0.7)
Discharge: 2013Q1	-	-	17 (0.3)
Self-Pay or Charity	-	-	19 (0.3)

*Unspecified Renal Disease: Unspecified Renal Disease without mention of hypertension, Primigravida: Primagravida at the extremes of maternal age, Other Cardiovascular Disease: Other cardiovascular diseases complicating pregnancy and childbirth or the puerperium, Renal Failure: Renal failure not elsewhere classified

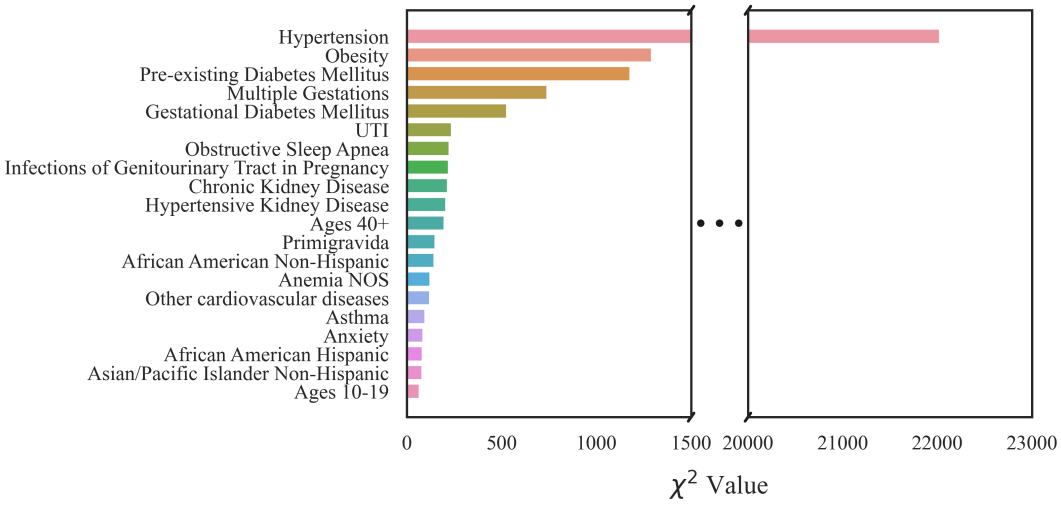


Figure 4.10: The Feature Ranking for the Full Texas Dataset

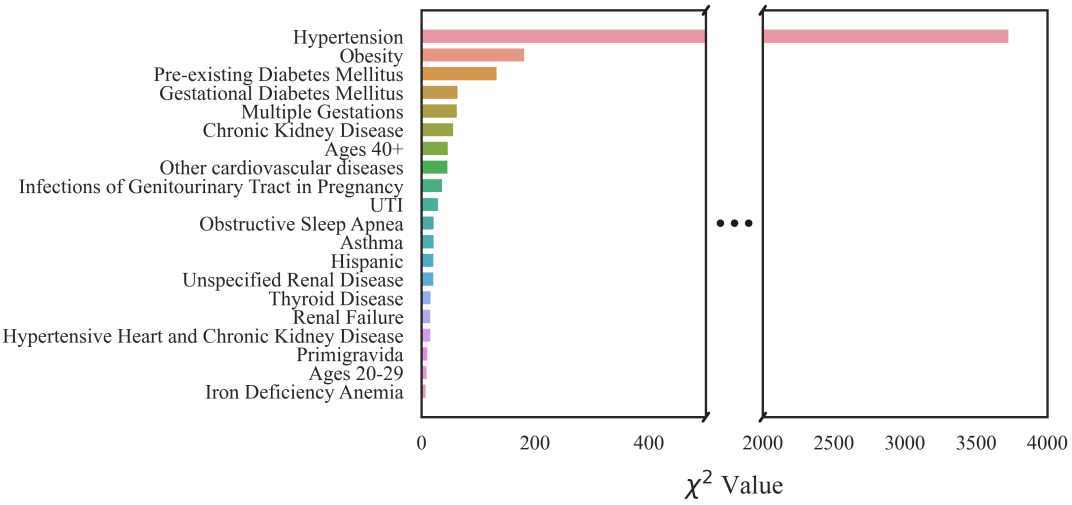


Figure 4.11: The Feature Ranking for the Texas African American Dataset

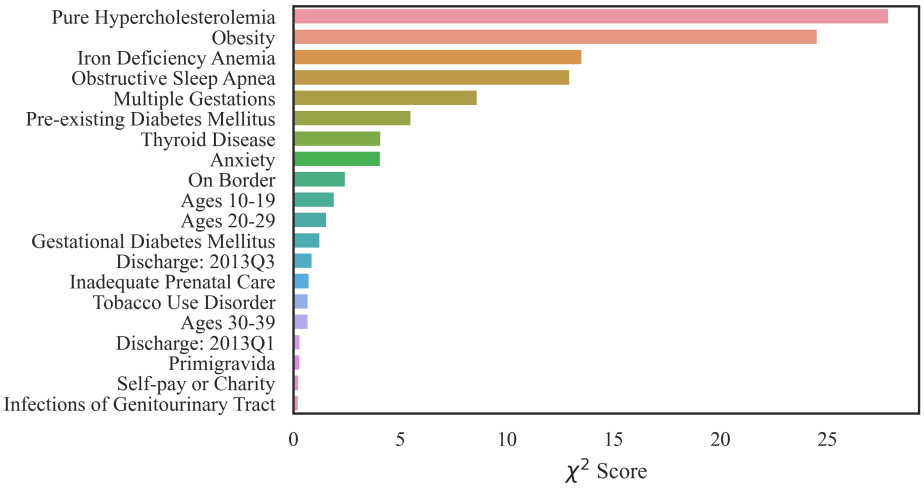


Figure 4.12: The Feature Ranking for the Texas Native American Dataset

The ranking of the top 20 features of the Oklahoma dataset is given by the Chi-squared feature selection which is shown in table 4.13. The Chi-squared scores are given in the parenthesis. Although there are differences in which features are chosen among the various groups, there is also a considerable amount of overlap. For example, Obesity is the highest, second highest, and third highest ranked feature in the full, only African American, and only Native American populations datasets respectively. In the African American dataset, there are 7 features that are indicated as important that do not appear in the full population's most important features. These are primagravida, Month of Delivery, Ages 20-29, Ages 30-39, Medicare, Unspecified Vitamin D Deficiency, and History of Premature Delivery. In the Native American dataset, there are even more specific features that do not overlap with the general population. These features are primagrivada at the extremes of maternal age, ages 20-29, thyroid disease, ages 30-39, self-pay, medicare, unspecified vitamin D deficiency, history of premature delivery, and iron deficiency anemia.

Table 4.13: Feature rankings derived by the Chi-squared scores in the Oklahoma dataset in general (full dataset) as well as among only African American and only Native American populations. Each column consists of feature ranking based on their importance with the associated Chi-squared scores in parenthesis.

Features	**Full**	**African American**	**Native American**
Obesity	1 (426.024)	2 (55.110)	3 (16.946)
Pre-existing Diabetes Mellitus	2 (289.184)	1 (65.243)	5 (7.258)
Multiple Gestations	3 (201.018)	7 (6.913)	1 (20.120)
Proteinuria	4 (153.201)	4 (14.106)	2 (16.234)
Native American	5 (66.940)	-	-
Gestational Diabetes Mellitus	6 (65.720)	3 (26.303)	16 (1.890)
Unspecified Renal Disease*	7 (63.480)	12 (4.611)	9 (5.652)
Infections of Genitourinary Tract in Pregnancy	8 (44.670)	6 (9.790)	14 (2.562)
Anxiety	9 (42.012)	-	15 (1.990)
Other Anemia	10 (35.197)	-	18 (1.286)
Hypertension	11 (33.132)	-	-
Ages 10-19	12 (26.343)	11 (4.814))	-
Ages 40+	13 (22.588)	5 (10.949)	-
Depression	14 (19.192)	-	7 (5.853)
Amphetamine Dependence	15 (18.947)	18 (1.451)	8 (5.843)
Other/Unknown Race	16 (18.136)	-	-
Marital Status	17 (11.625)	-	4 (7.543)
African American	18 (10.984)	-	-
Iron Deficiency Anemia	19 (10.763)	19 (1.372)	-
Self-Pay	20 (8.614)	14 (2.380)	-
Hypertensive Kidney Disease	-	-	6 (7.155)
Ages 20-29	-	8 (6.751)	-
Primigravida*	-	9 (5.706)	11 (3.393)
Month of Delivery	-	10 (4.868)	-
Current Smoker	-	-	17 (1.536)
Cocaine Dependence	-	-	18 (1.263)
Ages 30-39	-	13 (4.611)	-
Medicare	-	15 (2.022)	-
Unspecified Vitamin D Deficiency	-	16 (1.962)	-
History of Premature Delivery	-	17 (1.855)	20 (1.107)
UTI	-	-	12 (3.092)
Intrauterine Death	-	-	13 (2.884)

*Unspecified Renal Disease: Unspecified Renal Disease without mention of hypertension, Primigravida: Primagravida at the extremes of maternal age

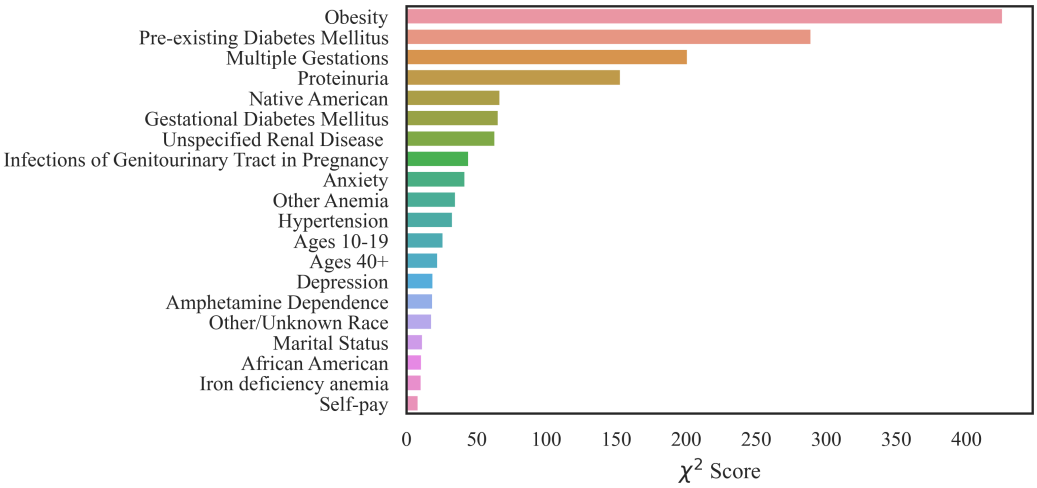


Figure 4.13: The Feature Ranking for the Full Oklahoma Dataset

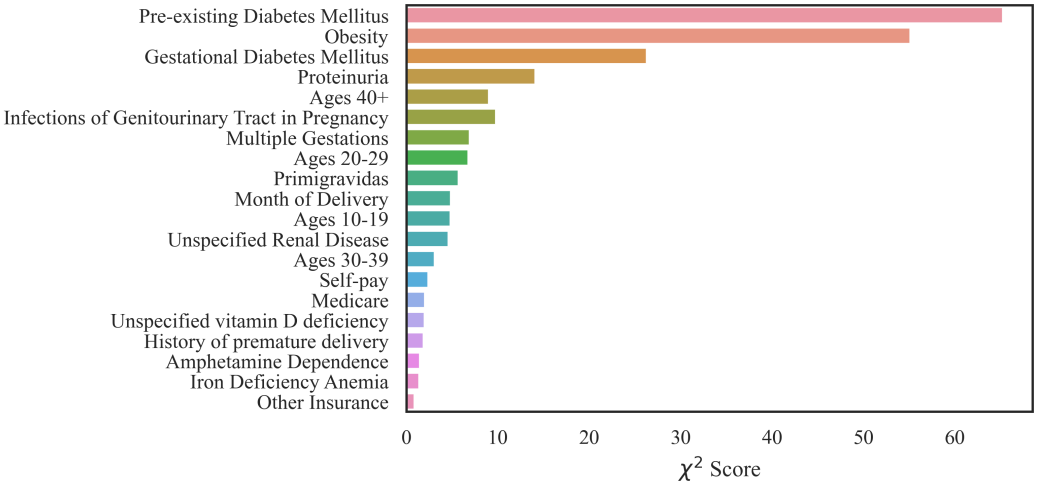


Figure 4.14: The Feature Ranking for the Oklahoma African American Dataset

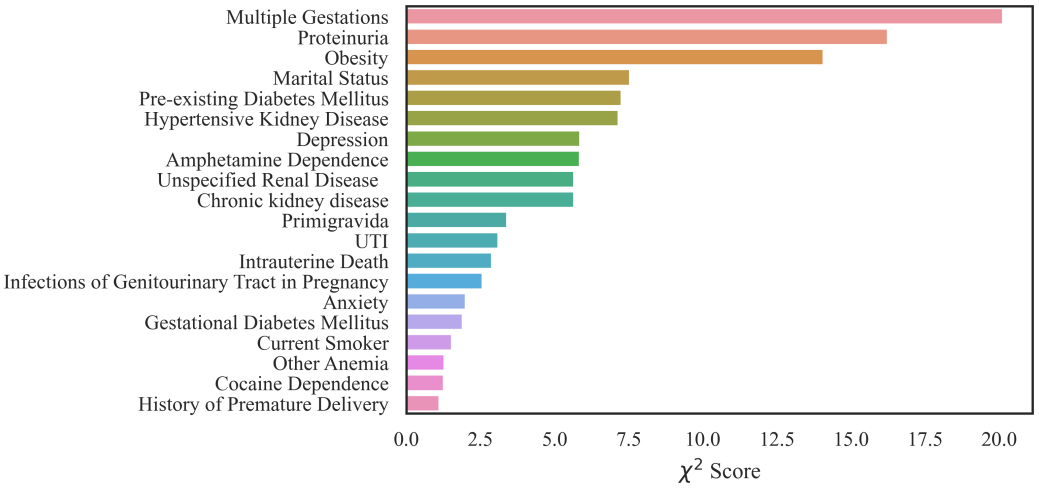


Figure 4.15: The Feature Ranking for the Oklahoma Native American Dataset

4.4 Missing Data

Table 4.14 shows the list of features with missing values with their associated missing rate in the Texas dataset. The most frequently missing feature is the County information, followed by the patients' ethnicity and race, followed by what kind of insurance is used by patients. We note that there are no missing variables in the age category.

Table 4.14: The List of Features with Missing Values in the Texas Dataset

Feature	**Missing Ratio**
Race	878 (0.243%)
Ethnicity	3,418 (0.947%)
County	9,018 (2.498 %)
Insurance	149 (0.041%)

Table 4.15 shows the amount missing features in the Oklahoma dataset. The most commonly missing feature in this dataset is marital status, with a total of 15,015 missing values. The next two features are county and insurance type each in the single digits.

Table 4.15: The List of Features with Missing Values in the Oklahoma Dataset

Feature	**Missing Ratio**
Race	0 (0.000%)
County	3 (0.004%)
Marital Status	15,015 (17.742%)
Month of Delivery	0 (0.000%)
Age	0 (0.000%)
Insurance	3 (0.004%)

In order handle the missing values, we are required to determine if there is a pattern behind in the missing values. Missing data can be categorized in three groups: missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) (Rubin, 1976). If the data is MCAR, there is no pattern or reason behind why a value is missing; the data does not depend on the observed or missing values. In this case, it is usually safe to drop the missing values if there are few of them. If the data is MAR, then the data is missing based on a pattern in the observed values, and dropping them would remove information from the model. If the data is MNAR, then the data is missing based on a pattern within the missing data.

In order to determine how to handle the missing values, we first performed Little's test to check whether data is missing completely at random, or MCAR (Little, 1988). This test checks the likelihood of data being MCAR. Table 4.16 shows the results of Little's test on both Texas and Oklahoma datasets.

Table 4.16: MCAR results

Dataset	**Chi-Squared**	**Degrees of Freedom**	**p-value**
Texas	16058.77	61	$\ll 0.0000$
Oklahoma	11482	17	$\ll 0.0000$

Since these results show very small p-values at 5% the significance level. So, the null hypothesis of MCAR (Missing Completely at Random) is rejected at the 5% significance level. Since data is not MCAR, we can perform imputation algorithms to estimate missing values. We selected Multiple Imputation technique (Buuran van, Groothuis-Oudshoorn,

2010) to estimate missing values. This imputation method is performed in a "round robin fashion", which first chooses the feature with the least missing values as a target variable and then builds a predictive model using all the other non-missing features. This process is repeated on the next least missing feature until all missing values are estimated. In this work, we used Bayesian Ridge Regression (Tipping, 2001) as the model for imputation.

Chapter 5

Results

In this section, we present results of DNN and the proposed CSDNN algorithms on Oklahoma and Texas datasets. The performance of these algorithms are compared based on the evaluation measures described in Chapter 3. We implement both DNN and CSDNN algorithms in Python version 3.6 with Keras (Chollet, others, 2015) and TensorFlow libraries (Martín et al., 2015). Statistical testing was performed using SciPy (Virtanen et al., 2020). All experiments and data processing are performed on an AMD Ryzen 5 3.6 GHz 6-Core processor and 16GB of Ram in a 64-bit platform. Our source code is available at the online repository mentioned in Appendix A. We used Multiple Imputation technique (Buuran van, Groothuis-Oudshoorn, 2010) to estimate missing values. For the Multiple Imputation implementation, we used Bayesian Ridge Regression (Tipping, 2001) within 5-fold cross-validation. In all models, the 20% dropout rate was applied in order to reduce overfitting.

5.1 CSDNN architecture

The optimal hyperparameters of DNN and CSDNN models are determined using Random Search (RS), Hyperband (HB), and Bayesian optimization (BO) model selection approaches each time before DNN and CSDNN training. The initial range of each hyperparameter is summarized in Table 5.1 for the model selection algorithms. These hyperparamters are batch

size, the number of epochs, the number of hidden layers (h), the number of neurons in hidden layers (k), and the learning rate (LR). In general, model selection algorithm is performed on each dataset using 10-fold cross validation repeated 5 times in order to increase robustness of results, however when looking at the smaller sub-populations datasets cross validation was repeated 35 times to increase the robustness of the results. The best set of hyperparameters is selected based on the model selection that yields the highest G-mean. Tables 5.2-5.7 show the best architecture of the DNN and CSDNN with WCE and FL functions, plus the hybrid models that additionally balanced the batches with oversampling with replacement. We observe that Hyperband model selection both DNN and CSDNN performs well on all datasets consistently for both DNN and CSDNNs. We note that in the following tables, Tanh is abbreviated as TH and ReLU is abbreviated as RL.

Table 5.1: Summary of hyperparameter ranges for DNN and CSDNN, where k is the number of hidden units in a layer, h is the number of hidden layers, and LR is the learning rate.

	Batch size	**Epochs**	**k**	**h**	**LR**
Range	64-8096	10-200	32-64	2-8	0.01-0.0001

Hyperband most consistently found the best architecture, with it's final selections making up 22 of the models examined. Most models consisted of only 3-4 layers, with the largest in terms of layers belonging to the Texas Full dataset using CE loss with 8 layers. Most models used larger learning rates of 0.001, but a few of the smaller datasets (Texas Native, Oklahoma African, Oklahoma Native) had smaller learning rates chosen, particularly in the CE loss function and in the cases where the batches were balanced with random oversampling.

Table 5.2: DNN (with CE loss) architecture of Texas and Oklahoma PUDF datasets. We note that h_i and a_i refer to the number of neurons and activation function in the hidden layer i, respectively, where $i = 1, 2, 3, ..., 8$.

Dataset	h_1, a_1	h_2, a_2	h_3, a_3	h_4, a_4	h_5, a_5	h_6, a_6	h_7, a_7	h_8, a_8	**Opt**	**LR**	**Batch**	**Tuner**
TX Full	30, RL	30, TH	60, TH	60, RL	60, TH	45, RL	60, TH	30, RL	NAdam	0.001	8192	HB
TX AA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSProp	0.001	8192	HB
TX NA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	8192	HB
OK Full	60, TH	60, RL	41, TH	-	-	-	-	-	RMSProp	0.001	8192	HB
OK AA	30, TH	60, RL	45, TH	45, RL	41, TH	-	-	-	SGD	0.0001	8192	HB
OK NA	45, TH	36, TH	30, TH	-	-	-	-	-	Adam	0.0001	8192	RA

Table 5.3: CSDNN (with WCE loss) architecture of Texas and Oklahoma PUDF datasets. We note that h_i and a_i refer to the number of neurons and activation function in the hidden layer i, respectively, where $i = 1, 2, 3, ..., 8$.

Dataset	h_1, a_1	h_2, a_2	h_3, a_3	h_4, a_4	h_5, a_5	h_6, a_6	h_7, a_7	h_8, a_8	**Opt**	**LR**	**Batch**	**Tuner**
TX Full	45, TH	30, RL	60, TH	30, TH TH	30, TH	60, TH	30, TH	-	Adam	0.001	8192	HB
TX AA	36, RL	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	8192	HB
TX NA	41, RL	30, RL	36, TH	45, TH	-	-	-	-	RMSprop	0.001	8192	RA
OK Full	30, TH	60, TH	41, TH	-	-	-	-	-	RMSprop	0.001	8192	HB
OK AA	60, RL	60, TH	60, TH	30, TH	-	-	-	-	NAdam	0.001	8192	RA
OK NA	60, TH	36, TH	36, TH	30, TH	-	-	-	-	NAdam	0.001	8192	BA

Table 5.4: CSDNN (with FL) architecture of Texas and Oklahoma PUDF datasets. We note that h_i and a_i refer to the number of neurons and activation function in the hidden layer i, respectively, where $i = 1, 2, 3, ..., 8$.

Dataset	h_1, a_1	h_2, a_2	h_3, a_3	h_4, a_4	h_5, a_5	h_6, a_6	h_7, a_7	h_8, a_8	**Opt**	**LR**	**Batch**	**Tuner**	α	γ
TX Full	60, TH	30, RL	45, RL	-	-	-	-	-	RMSProp	0.001	8192	HB	0.97	1.25
TX AA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSProp	0.001	8192	HB	0.96	1.75
TX NA	60, TH	30, RL	45, RL	-	-	-	-	-	NAdam	0.001	8192	HB	0.97	1
OK Full	60, TH	30, RL	45, RL	-	-	-	-	-	RMSProp	0.001	8192	HB	0.95	1.0
OK AA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	8192	HB	0.92	0.25
OK NA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	8192	HB	0.94	0.25

Table 5.5: CSDNN (with FL and Balanced Batches) architecture of Texas and Oklahoma PUDF datasets. We note that h_i and a_i refer to the number of neurons and activation function in the hidden layer i, respectively, where $i = 1, 2, 3, ..., 8$.

Dataset	h_1, a_1	h_2, a_2	h_3, a_3	h_4, a_4	h_5, a_5	h_6, a_6	h_7, a_7	h_8	**Opt**	**LR**	**Batch**	**Tuner**	α	γ
TX Full	60, RL	60, TH	41, TH	-	-	-	-	-	Adam	0.001	8192	BA	0.5	1.75
TX AA	60, RL	60, TH	60, TH	-	-	-	-	-	Adam	0.001	8192	HB	0.5	1.25
TX NA	60, TH	36, TH	41, RL	41, RL	36, TH	30, TH	-	-	SGD	0.0001	2048	RA	0.5	1.25
OK Full	60, TH	60, TH	-	-	-	-	-	-	Adam	0.001	8192	BA	0.5	1.25
OK AA	60, TH	60, RL	45, RL	30, TH	-	-	-	-	Adam	0.0001	1024	HB	0.5	1.25
OK NA	30, TH	30, RL	45, RL	36, RL	30, RL	41, TH	-	-	NAdam	0.001	1024	HB	0.5	1.25

Table 5.6: CSDNN (with WCE and Balanced Batches) architecture of Texas and Oklahoma PUDF datasets. We note that h_i and a_i refer to the number of neurons and activation function in the hidden layer i, respectively, where $i = 1, 2, 3, ..., 8$.

Dataset	h_1, a_1	h_2, a_2	h_3, a_3	h_4, a_4	h_5, a_5	h_6, a_6	h_7, a_7	h_8, a_8	**Opt**	**LR**	**Batch**	**Tuner**
TX Full	41, TH	60, RL	41, RL	30, RL	30, RL	-	-	-	Adam	0.001	8192	BA
TX AA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	8192	HB
TX NA	41, RL	30, RL	36, TH	45, TH	-	-	-	-	RMSprop	0.00001	2048	RA
OK Full	30, RL	30, TH	60, TH	45, RL	30, RL	60, RL	-	-	RMSprop	0.001	1024	BA
OK AA	30, TH	30, TH	45, RL	60, TH	-	-	-	-	Adam	0.0001	1024	HB
OK NA	30, TH	60, RL	-	-	-	-	-	-	SGD	0.00001	1024	HB

Table 5.7: DNN (with Balanced Batches) architecture of Texas and Oklahoma PUDF datasets. We note that h_i and a_i refer to the number of neurons and activation function in the hidden layer i, respectively, where $i = 1, 2, 3, ..., 8$.

Dataset	h_1, a_1	h_2, a_2	h_3, a_3	h_4, a_4	h_5, a_5	h_6, a_6	h_7, a_7	h_8	**Opt**	**LR**	**Batch**	**Tuner**
TX Full	60, RL	36, RL	-	-	-	-	-	-	Adam	0.001	8192	HB
TX AA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	8192	HB
TX NA	60, TH	30, RL	45, RL	-	-	-	-	-	RMSprop	0.001	2048	HB
OK Full	60, TH	60, TH	60, TH	60, TH	-	-	-	-	Adam	0.001	1024	BA
OK AA	30, RL	41, TH	-	-	-	-	-	-	RMSprRLop	0.0001	1024	HB
OK NA	30, TH	41, TH	36, TH	45, TH	41, RL	36, TH	45, TH	41, TH	RMSprop	8192	RA	

5.2 Predictive Accuracy of the CSDNN on population-specific features

Most of the previously examined studies have built their models using a general population. This works well for the populations that make up a large proportion of the dataset, but less well for smaller groups like racial and ethnic minorities. This difference can have clinical consequences. For example, imagine an African American woman going into a clinic where one of these models is employed to analyse her risk of getting preeclampsia. Looking at figure 5.1, this would correspond to Model 1 — she would be getting a recommendation that was built for populations she is not a member of. Moreover, the model that is built for the woman was more than likely made using features that had been previously indicated as significant in the literature. As mentioned previously however, many studies have not been trained on race dis-aggregated data, meaning they suffer from the same issue of fitting to the majority group, which would correspond to the use of Model 2 in figure 5.1. This difference can be seen in the previous chapter, where anxiety was indicated as significant in the general population, but not in the African Americans. Again, the same kind of error would occur - the symptoms she is being tested for may or may not be applicable to her subpopulation. A potentially better solution then would be to use a model that is trained on her specific subpopulation using features that have been indicated as important to her subpopulation. This refers to Model 3 figure 5.1.

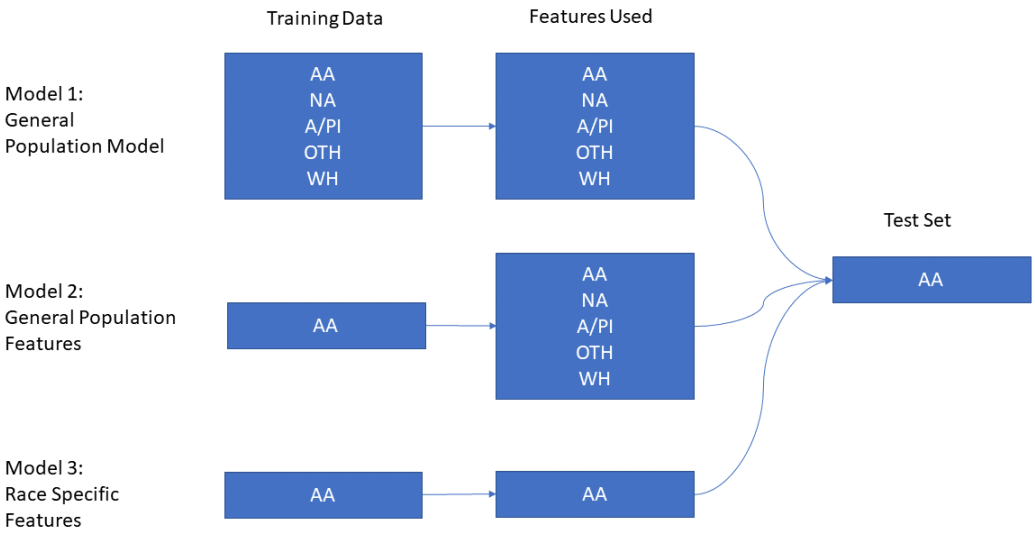


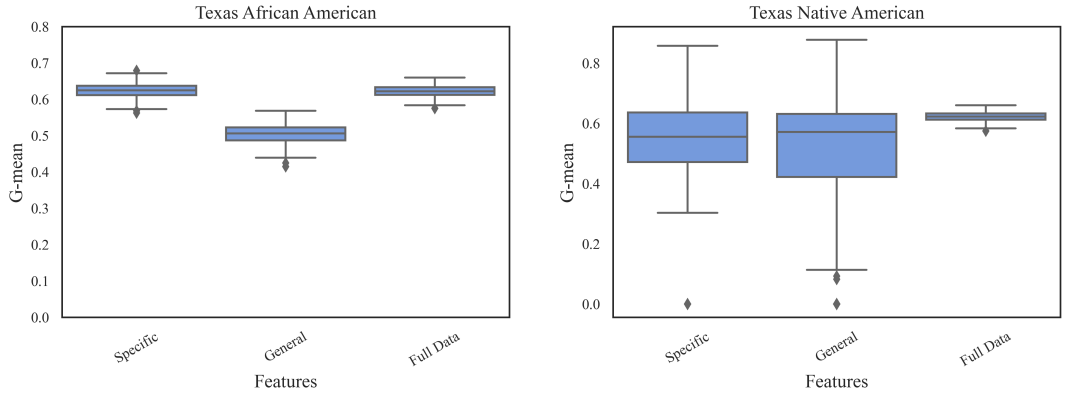
Figure 5.1: The three models tested in this section. Model 1 refers to models that were trained on the full dataset using the most significant features population in general. Model 2 refers to the models that were trained using only the patients of the subpopulation, but still using the features of the general population. Model 3 refers to the models that were trained using only the patients in the subpopulation and using only the features significant to that specific population. AA: African American, NA:Native American, A/PI: Asian or Pacific Islander, OTH: Other, WH: White

In order to test these scenarios, we built 3 unique models: One model that was trained on the full population and used the top 20 features of that population; one trained on the subpopulation only but used the top 20 features from the general population; and one trained on the subpopulation only and used the top 20 features from that specific subpopulation.

The results are reported in Tables 5.8 and Figures 5.2-5.3. In most cases, the models built with population specific features performs better in terms of G-mean and had a smaller spread of results than with the models trained on the full dataset with the exception of the Texas Native population.

Table 5.8: Results of general vs. specific feature selection in Texas NA and AA populations. The highest sensitivity, specificity, G-mean, AUC, precision, and accuracy values between general and specific feature selection are denoted in bold.

Dataset	Accuracy	AUC	G-mean	Precision	Recall	Specificity
Texas AA - General	**0.788**	0.654	0.610	**0.121**	0.461	**0.807**
Texas AA - Specific	0.748	0.667	**0.624**	0.110	0.512	0.762
Texas AA - Full Data	0.619	**0.670**	0.622	0.086	**0.625**	0.619
Texas NA - General	**0.706**	0.559	0.491	0.052	0.423	**0.716**
Texas NA - Specific	0.544	0.571	0.535	0.044	**0.582**	0.543
Texas NA - Full Data	0.699	**0.611**	**0.575**	**0.061**	0.533	0.706
Oklahoma AA - General	0.509	**0.635**	0.477	0.119	**0.679**	0.494
Oklahoma AA - Specific	**0.643**	0.619	**0.578**	**0.124**	0.529	**0.653**
Oklahoma AA - Full Data	0.346	0.599	0.451	0.073	**0.757**	0.319
Oklahoma NA - General	0.622	0.593	0.552	0.085	**0.491**	**0.631**
Oklahoma NA - Specific	**0.649**	0.595	**0.553**	0.088	0.471	0.661
Oklahoma NA - Full Data	0.364	**0.656**	0.459	**0.098**	0.792	0.327

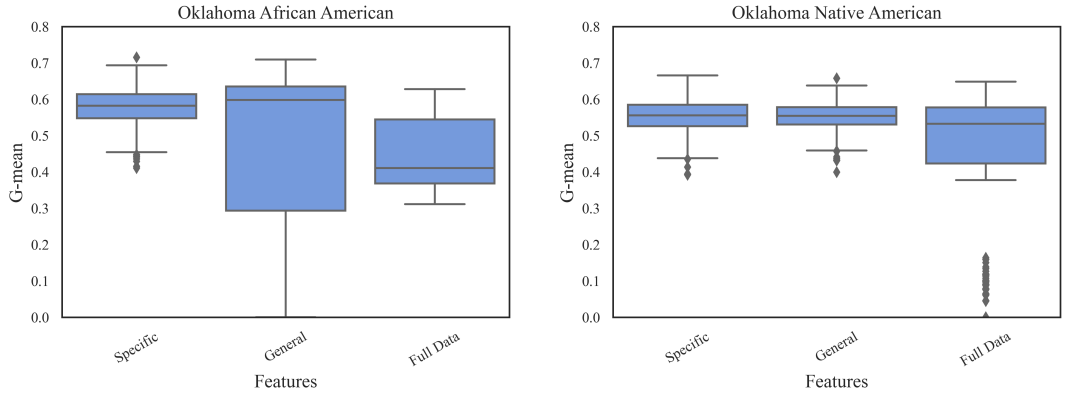


a) African American dataset b) Native American dataset

Figure 5.2: G-means of the models using features selected from the population-specific feature set vs. general vs. full population feature set for Texas data - Left: African American dataset; Right: Native American dataset

Figures 5.3 shows the G-means of of the African American and Native American datasets

when general population and population specific features are applied. For the Native American population, there is a slight improvement in the mean G-mean, but a larger distribution.



a) African American dataset b) Native American dataset

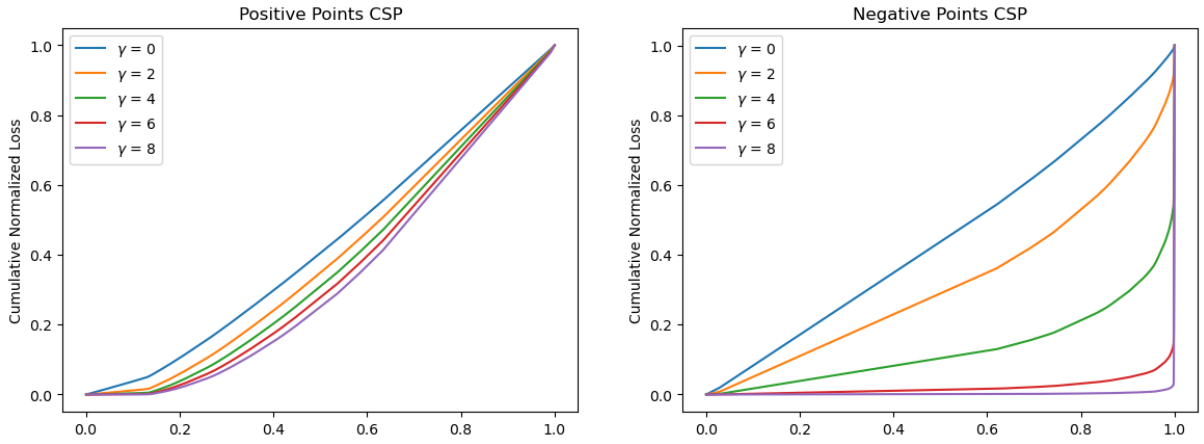
Figure 5.3: G-means of the models using features selected from the population-specific feature set vs. general population feature set for Oklahoma data - Left: African American dataset; Right: Native American dataset

5.3 Comparative Analysis of CSDNN with FL versus parameters γ and α

Since preeclampsia prediction is a highly imbalanced problem, most machine learning model's loss functions will be overwhelmed by the large number of negative samples - most of which will likely be easy to predict, since the majority of women in each dataset are healthy and do not suffer from any of the health conditions examined. Focal loss's advantage then is not only in weighting the samples according to class, but also by down-weighting the negative samples that are easy to predict, reducing their impact on the loss function. This can be demonstrated by the figures shown below.

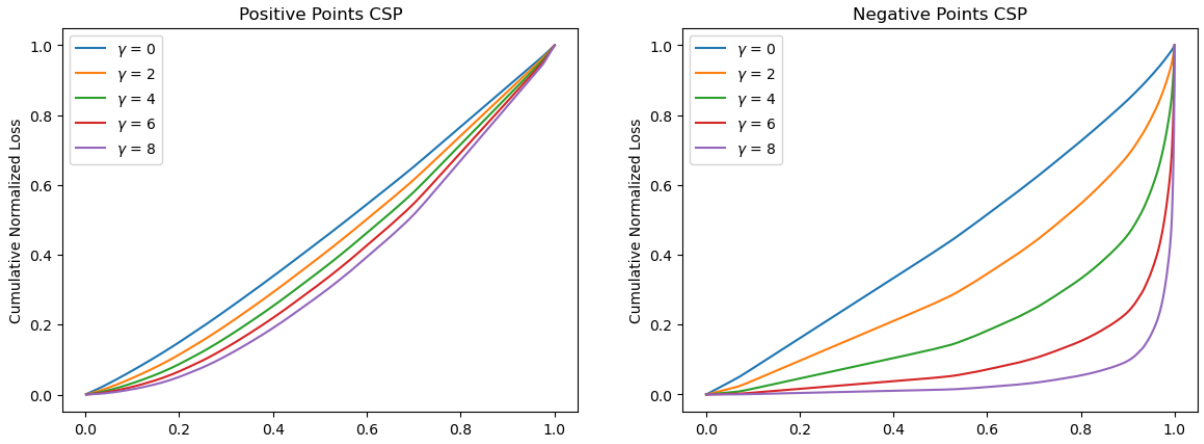
Inspired by the original paper by Lin et al. (2017), Figs. 5.4 and 5.5 are created by training CSDNN model with an α of 0.5. The test data samples are split into the positive and negative samples, and the loss is calculated for each samples using different values

of γ. The plots are then created by ordering the normalized loss from lowest to highest and projecting the cumulative distribution function (CDF) for both positive and negative classes for various γ (Figs. 5.4 and 5.5). The effect of γ on positive samples (cases with preeclampsia) is not as noticeable, however the effect of γ on negative samples (cases without preeclampsia) is substantially different. Both positive and negative CFDs look relatively analogous when $\gamma = 0$. We observe that increasing the γ has a large effect on down-weighting the easy negative samples, as FL focuses learning on hard negative examples. The results are consistent with earlier literature on FL Lin et al. (2017).



a) positive (preeclamptic) samples b) negative (non-preeclamptic) samples

Figure 5.4: Cumulative distribution functions of the normalized loss for positive and negative samples for various γ values for Texas data



a) Positive (preeclamptic) samples b) Negative (non-preeclamptic) samples

Figure 5.5: Cumulative distribution functions of the normalized loss for positive and negative samples for various γ values for Oklahoma data

Tables 5.9, and 5.10, which used the best performing α, show that an increase of γ would highly affect the specificity and recall. The higher γ results in lower specificity and higher recall.

Table 5.9: Comparative analysis of CSDNN versus γ using Texas Full dataset with $\alpha = 0.97$. The highest sensitivity, specificity, G-mean, AUC, precision, and accuracy values are denoted in bold.

γ	0	2	4	6	8
Accuracy	**0.775**	0.759	0.759	0.759	0.698
G-mean	**0.573**	0.561	0.560	0.561	0.515
AUC	**0.634**	**0.634**	0.633	0.633	0.633
Specificity	**0.789**	0.772	0.772	0.772	0.706
Recall	0.438	0.449	0.449	**0.450**	0.497
Precision	**0.084**	0.083	0.083	0.083	0.079

Table 5.10: Comparative analysis of CSDNN versus γ using Oklahoma Full dataset with $\alpha = 0.95$. The highest sensitivity, specificity, G-mean, AUC, precision, and accuracy values are denoted in bold.

γ	0	2	4	6	8
Accuracy	0.636	0.685	0.674	0.657	**0.735**
G-mean	0.603	0.613	0.611	**0.614**	0.593
AUC	**0.658**	**0.658**	0.650	0.648	0.647
Specificity	0.640	0.693	0.681	0.662	**0.750**
Recall	0.568	0.542	0.549	**0.570**	0.468
Precision	0.085	0.094	0.092	**0.090**	0.010

Comparative analysis of CSDNN equipped with FL versus both α and γ parameters using full Oklahoma and Texas datasets are shown in Tables 5.11-5.12. According to these results, we observe that α tended to have a greater effect on the outcome than γ. Additionally, α was usually found to be most effective the closer it was to the percentage of non-preeclamptic patients, meaning that the best performing cost function was the one that balanced the weights of the positive and negative samples before downweighting any of the easily classified ones.

Table 5.11: Sensitivity analysis of CSDNN equipped with FL versus both α and γ parameters using Texas Full dataset based on G-mean. The highest G-mean is denoted in bold.

		γ								
		0	0.25	0.5	0.75	1	1.25	1.5	01.75	2
α	0.9	0.432	0.434	0.423	0.414	0.412	0.413	0.415	0.416	0.409
	0.91	0.449	0.448	0.446	0.436	0.437	0.427	0.445	0.43	0.428
	0.92	0.489	0.492	0.477	0.492	0.485	0.452	0.474	0.44	0.447
	0.93	0.502	0.499	0.500	0.502	0.495	0.500	0.502	0.501	0.492
	0.94	0.526	0.516	0.526	0.52	0.521	0.536	0.511	0.516	0.505
	0.95	0.593	0.559	0.584	0.594	0.572	0.562	0.559	0.563	0.557
	0.96	0.609	0.608	0.608	0.610	0.609	0.609	0.608	0.610	0.608
	0.97	0.626	0.626	0.625	0.626	0.626	**0.631**	0.625	0.625	0.626
	0.98	0.169	0.174	0.170	0.170	0.170	0.168	0.161	0.168	0.169
	0.99	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 5.12: Sensitivity analysis of CSDNN equipped with FL versus both α and γ parameters using Oklahoma Full dataset based on G-mean. The highest G-mean is denoted in bold.

		γ								
		0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
α	0.90	0.432	0.442	0.434	0.421	0.437	0.413	0.43	0.44	0.408
	0.91	0.483	0.470	0.476	0.473	0.471	0.483	0.468	0.471	0.475
	0.92	0.501	0.499	0.500	0.492	0.493	0.501	0.495	0.494	0.489
	0.93	0.533	0.524	0.530	0.518	0.53	0.515	0.515	0.519	0.521
	0.94	0.577	0.575	0.574	0.575	0.577	0.562	0.564	0.566	0.559
	0.95	0.605	0.613	0.615	0.613	0.608	0.614	**0.616**	0.613	0.608
	0.96	0.588	0.583	0.586	0.604	0.586	0.588	0.603	0.585	0.588
	0.97	0.254	0.252	0.253	0.255	0.253	0.252	0.253	0.265	0.095
	0.98	0.000	0.000	0.000	0.000	0.000	0.019	0.034	0.000	0.000
	0.99	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

5.4 Comparative Analysis of different loss functions

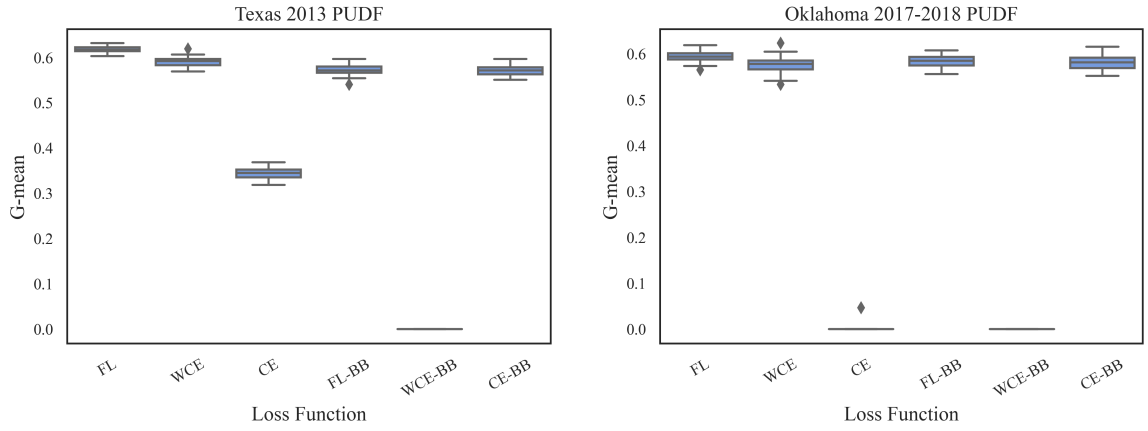
In most cases, the FL function outperformed all other methods in terms of AUC and G-mean, with the exception of the Oklahoma Full dataset, in which AUC was largest for the CE function, the Oklahoma African American dataset, in which the best performing algorithm was WCE. In all cases, the cost-sensitive loss function improved the recall of the model, meaning that there was a greater number of preeclamptic cases being predicted than in the more traditional CE models. This did come at a cost to specificity, meaning that there were a greater number of false positives in these algorithms.

Table 5.13: Comparison of CSDNN with FL and WCE versus CE loss function on the full Texas and Oklahoma datasets as well as AA and NA population datasets.

Dataset	Loss Function	ACC	AUC	GM	SN	SP	PR
TX Full	FL	0.619	**0.663**	**0.617**	0.616	0.619	0.063
	WCE	0.813	0.663	0.590	0.420	0.830	0.093
	CE	0.963	0.658	0.344	0.118	0.998	0.689
	FL-BB	0.831	0.634	0.572	0.385	0.850	0.096
	WCE-BB	0.040	0.633	0.000	1.000	0.000	0.040
	CE-BB	0.832	0.634	0.571	0.384	0.851	0.096
TX AA	FL	0.748	**0.667**	**0.623**	0.512	0.762	0.110
	WCE	0.795	0.667	0.605	0.450	0.815	0.123
	CE	0.951	0.665	0.414	0.173	0.996	0.689
	FL-BB	0.778	0.666	0.612	0.472	0.795	0.117
	WCE-BB	0.054	0.667	0.000	1.000	0.000	0.054
	CE-BB	0.789	0.667	0.608	0.458	0.808	0.121
TX NA	FL	0.544	**0.571**	**0.535**	0.582	0.542	0.044
	WCE	0.658	0.563	0.484	0.413	0.666	0.043
	CE	0.965	0.535	0.000	0.000	1.000	0.167
	FL-BB	0.502	0.500	0.285	0.498	0.502	0.047
	WCE-BB	0.426	0.492	0.282	0.584	0.420	0.045
	CE-BB	0.706	0.571	0.466	0.368	0.718	0.046
OK Full	FL	0.622	0.635	**0.594**	0.566	0.626	0.082
	WCE	0.706	0.620	0.575	0.461	0.720	0.089
	CE	0.944	**0.636**	0.000	0.000	1.000	0.000
	FL-BB	0.702	0.635	0.583	0.476	0.716	0.090
	WCE-BB	0.056	0.619	0.000	1.000	0.000	0.056
	CE-BB	0.691	0.621	0.580	0.480	0.704	0.088
OK AA	FL	0.642	0.619	0.578	0.529	0.653	0.124
	WCE	0.589	**0.623**	**0.582**	0.588	0.589	0.115
	CE	0.478	0.501	0.172	0.527	0.475	0.070
	FL-BB	0.710	0.594	0.479	0.374	0.740	0.128
	WCE-BB	0.082	0.554	0.000	1.000	0.000	0.0819
	CE-BB	0.582	0.581	0.551	0.533	0.586	0.105
OK NA	FL	0.649	0.595	**0.553**	0.471	0.661	0.088
	WCE	0.653	**0.597**	0.551	0.462	0.666	0.088
	CE	0.641	0.576	0.540	0.467	0.653	0.087
	FL-BB	0.794	0.579	0.485	0.288	0.829	0.105
	WCE-BB	0.303	0.503	0.386	0.691	0.276	0.062
	CE-BB	0.641	0.576	0.540	0.467	0.653	0.087

Figures 5.6 - 5.8 show the results of the repeated cross-validations of each model on each dataset. In all cases, there is much smaller variation between each run of the FL model.

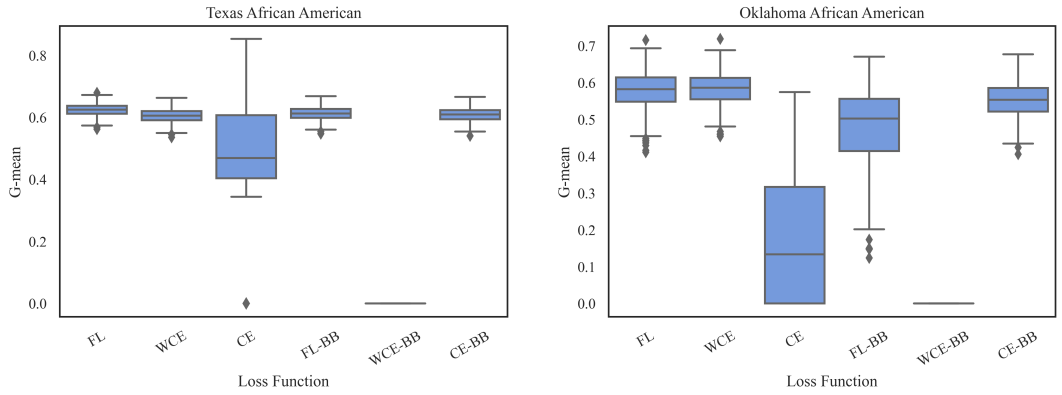
Additionally, the oversampled cases more frequently had greater variation between runs, showing that it was much more prone to overfitting. The difference in variation tended to shrink with the size of the dataset however, though even in the full Texas and Oklahoma datasets this trend is observable.



a) Texas dataset b) Oklahoma dataset

Figure 5.6: Comparison of CSDNN with FL and WCE versus CE loss function (in terms of G-mean) on the full dataset - Left: Texas dataset; Right: Oklahoma dataset

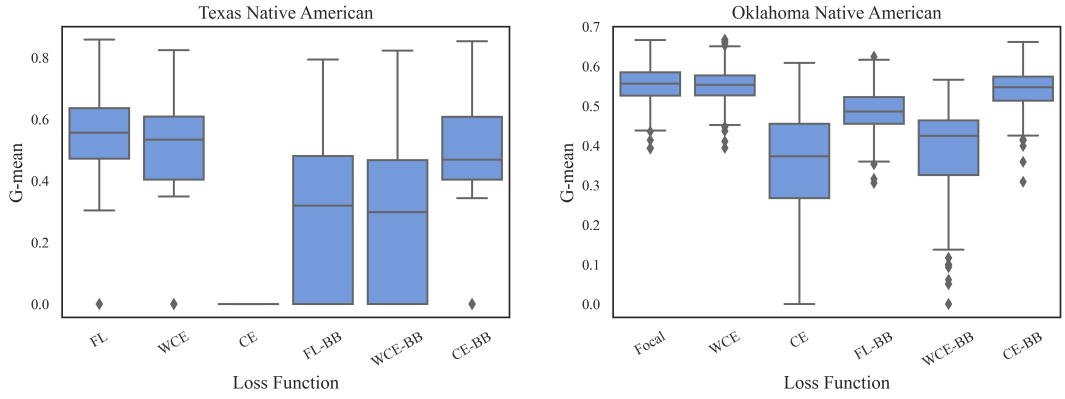
The African American datasets showed more variation than in the larger sets, with the CE model having the greatest variation. The African American sub-population was the one with the highest prevalance of preeclampsia, which could have influenced the more varied results.



a) Texas dataset b) Oklahoma dataset

Figure 5.7: Comparison of CSDNN with FL and WCE versus CE loss function (in terms of G-mean) on the African American population - Left: Texas dataset; Right: Oklahoma dataset

The Native American datasets had the largest spread of gmeans among any of the others, likely due to being the smallest sub-populations. This variation could be limited by the inclusion of more samples from the Native American population.



a) Texas dataset b) Oklahoma dataset

Figure 5.8: Comparison of CSDNN with FL and WCE versus CE loss function (in terms of G-mean) on the Native American population - Left: Texas dataset; Right: Oklahoma dataset

5.4.1 Comparative Analysis of model behavior while training

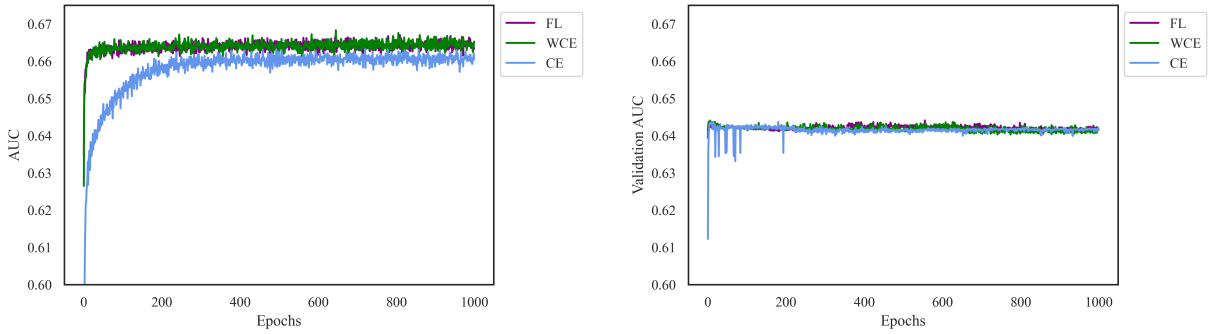
Figure 5.9 shows the AUC over the course of 1000 epochs for the Texas Full dataset. You can see on the figure on the left that the crossentropy loss function takes the longest to stabilize, and its validation AUC is subject to more extreme changes over the course of training. The Focal Loss and Weighted Crossentropy loss functions perform roughly the same, stabilizing very early on. All three loss functions result in slight overfitting, with the training AUC averaging between 0.65-0.67 and the validation AUC staying around 0.64.

Figure 5.11 shows the accuracy of all three loss functions over the course of 1000 epochs. The highest accuracy loss function is crossentropy, which remains consistent in all datasets. This is likely due to its tendency to predict only the majority classes, resulting in an accuracy that closely reflects the distribution of preeclamptic and non-preeclamptic patients. The validation accuracy shows some variation over the course of the epochs, with focal loss having a few spikes in accuracy while weighted crossentropy oscillates between 0.7 and 0.8 accuracy. Focal Loss is more stable than weighted crossentropy here, but is not quite as accurate.

Finally figure 5.13 shows the losses between each of the loss functions. The loss functions for this dataset are all fairly stable and do not change much between training and validation with the exception of crossentropy, which has a sharp drop early on before stabilizing. The highest loss was from weighted crossentropy, which applied a multiplicative factor depending on the class of the sample, while the lowest loss was from focal loss, which actively downweighted easily classifiable samples in the dataset. Focal Loss also downweighted the negative (non-preeclamptic) samples due to the α parameter, meaning that the majority of the samples did not contribute strongly to the loss. The Oklahoma Full dataset's AUC, Accuracy, and Loss are shown in figures 5.10-5.14. The training data shows trends similar to the Texas data, however there seems to be less overfitting in the case of the AUC, with crossentropy having a more consistent AUC and loss. The only other difference seems to be that the Accuracy of focal loss and weighted crossentropy seem to be less stable in this

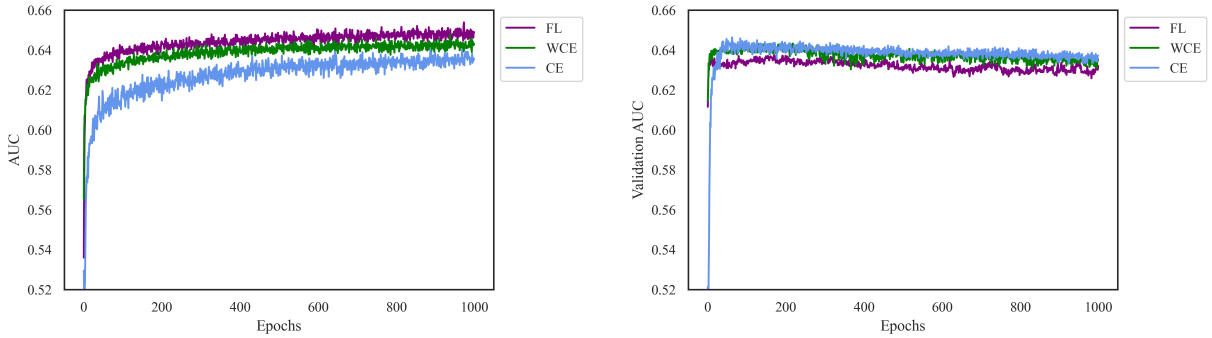
dataset than in the Texas dataset.

Figures 5.9-5.10 show the AUCs over the course of 1000 epochs for the training and validation data for each full dataset. In both cases, crossentropy took the longest to stabilize, although it also was the slowest to overfit to the data. In the Texas dataset crossentropy was also the most unstable, and all three loss functions overfit more than in the Oklahoma dataset.



a) Texas Full AUC b) Texas Full Validation AUC

Figure 5.9: The AUC over 1000 Epochs for the Texas Full Datasets

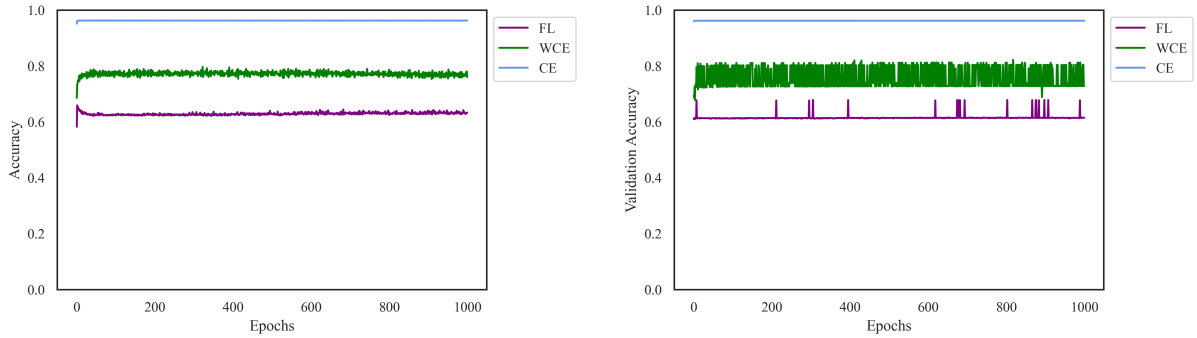


a) Oklahoma Full AUC b) Oklahoma Full Validation AUC

Figure 5.10: The AUC over 1000 Epochs for the Oklahoma Full Datasets

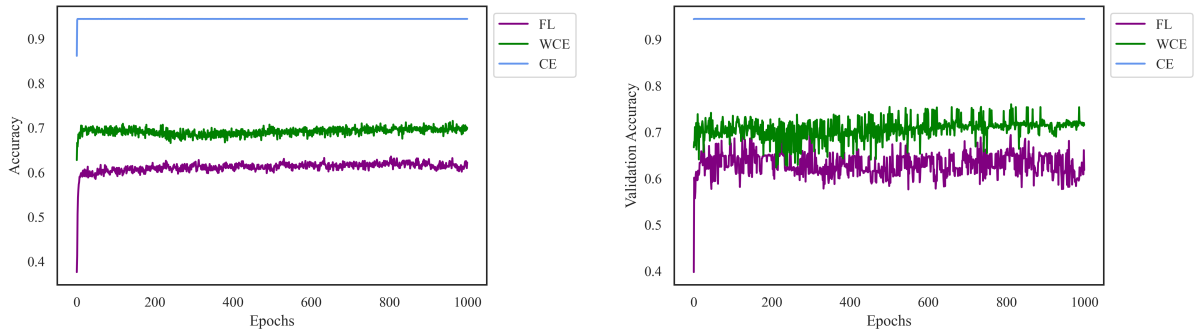
Figures 5.11-5.12 show the accuracy of each of the datasets. Crossentopy consistently had the highest and most stable accuracy, which can be explained by its tendency to predict only the most common class; since no learning was needed to improve the accuracy, it's accuracy

remained consistnet throughout the process. Both Focal Loss and weighted crossentropy were more unstable, likely caused by more emphasis on the smaller, more difficult to predict classes. In both datasets, focal loss had the lowest accuracy.



a) Texas Full Accuracy b) Texas Full Validation Accuracy

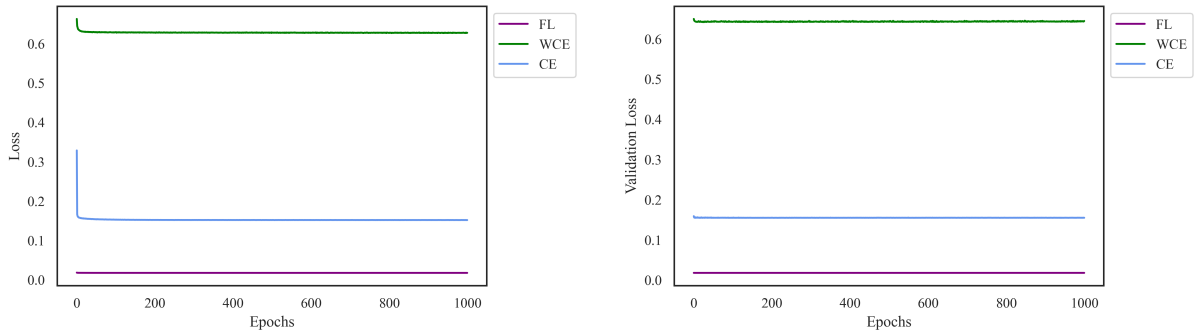
Figure 5.11: The Accuracy over 1000 Epochs for the Texas Full Datasets



a) Oklahoma Full Accuracy b) Oklahoma Full Validation Accuracy

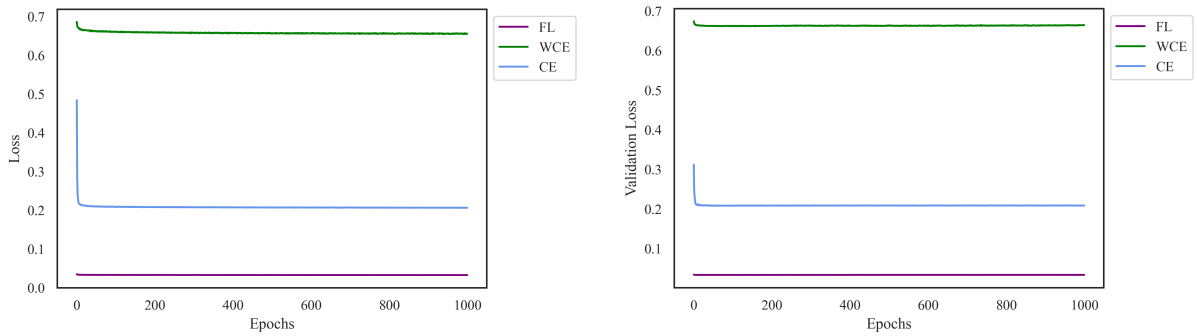
Figure 5.12: The Accuracy over 1000 Epochs for the Oklahoma Full Datasets

Figures 5.13-5.14 show the loss for each of the datasets. In the Texas and Oklahoma datasets, loss remained fairly constant throughout training, although crossentropy tended to start at a much higher loss before stabilizing.



a) Texas Full Loss b) Texas Full Validation Loss

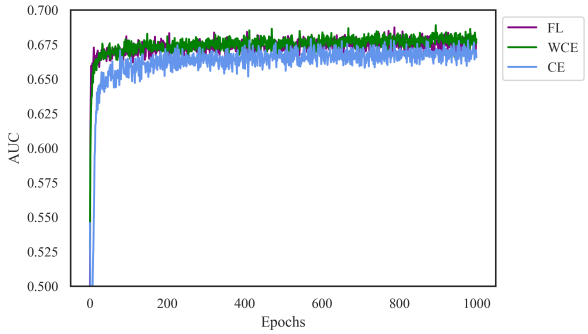
Figure 5.13: The Loss over 1000 Epochs for the Texas Full Datasets



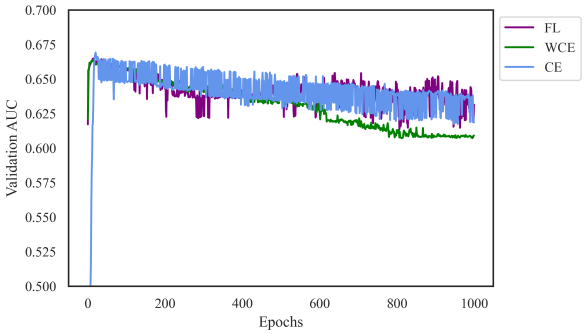
a) Oklahoma Full Loss b) Oklahoma Full Validation Loss

Figure 5.14: The Loss over 1000 Epochs for the Oklahoma Full Datasets

Figures 5.15-5.16 show the AUC of the African American subpopulation datasets. It largely follows the same trend as the larger groups with the exception of the Oklahoma African American dataset with crossentropy, which seems to get caught in a local minimum which it oscillates around. This is also the only case where Stochastic Gradient Descent was used as the optimizing function, which could be causing the model to be stuck in a local minimum.

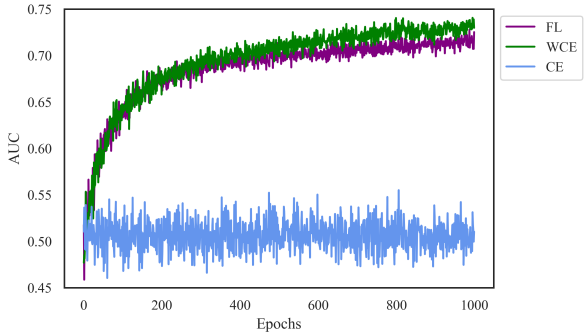


a) Texas African AUC

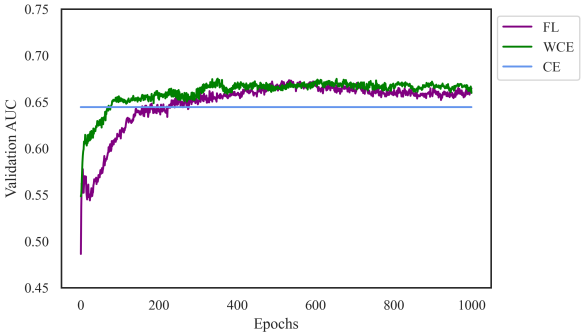


b) Texas African Validation AUC

Figure 5.15: The AUC over 1000 Epochs for the Texas African American Datasets



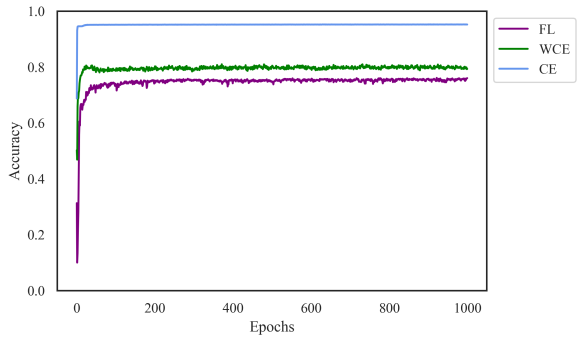
a) Oklahoma African AUC



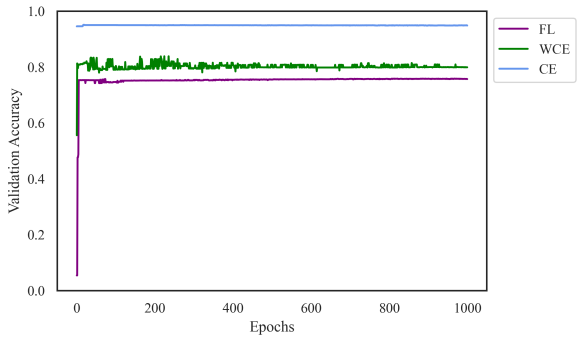
b) Oklahoma African Validation AUC

Figure 5.16: The AUC over 1000 Epochs for the Oklahoma African Datasets

Figure 5.17-5.18 show the accuracy of each of the African sub-population models. The trends seem to be the same as in the larger datasets with the exception of the Oklahoma African population, which has a lower CE accuracy.

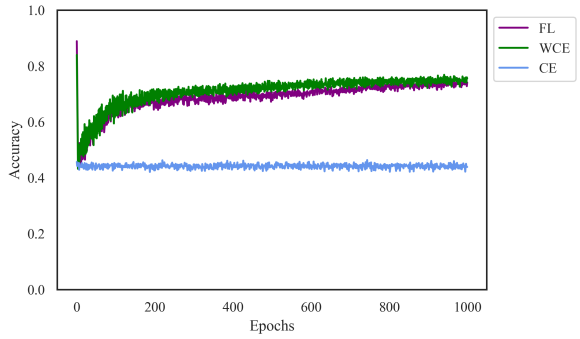


a) Texas African Accuracy

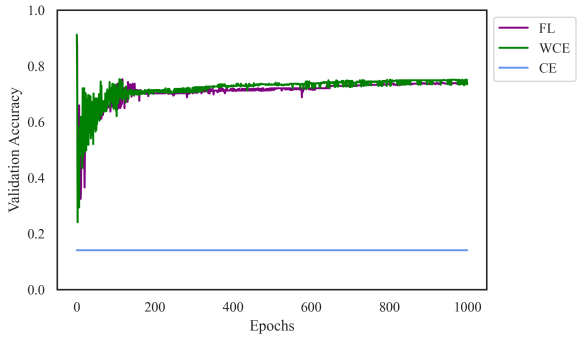


b) Texas African Validation Accuracy

Figure 5.17: The Accuracy over 1000 Epochs for the Texas African American Datasets



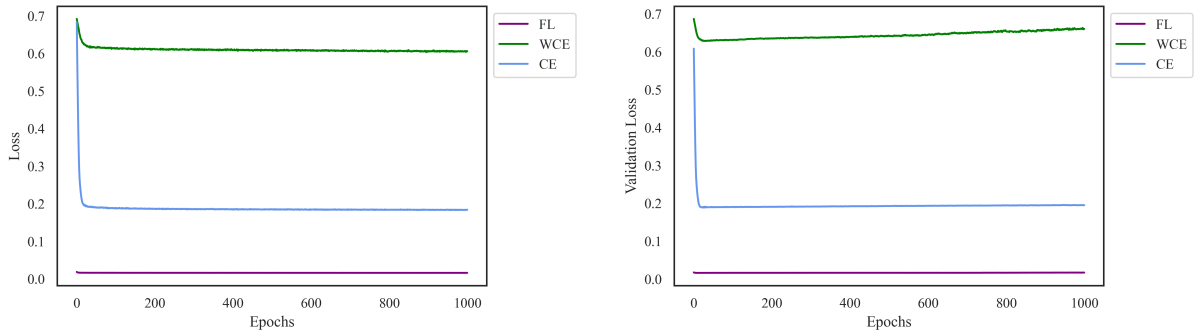
a) Oklahoma African Accuracy



b) Oklahoma African Validation Accuracy

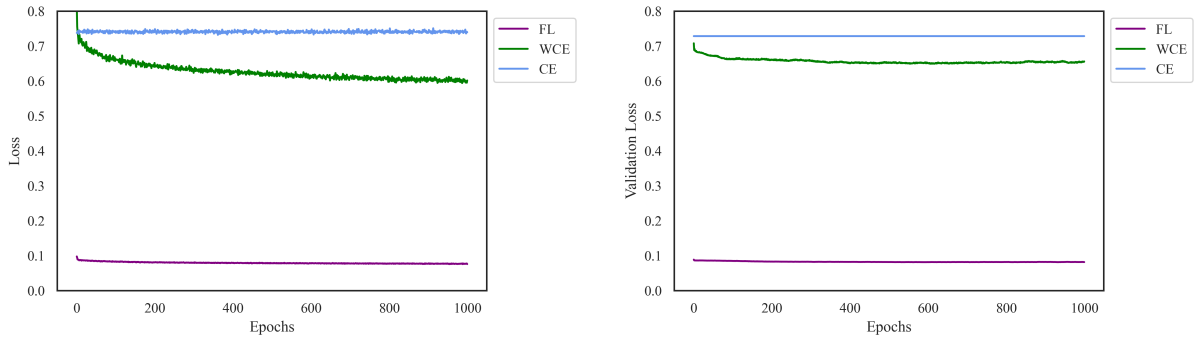
Figure 5.18: The Accuracy over 1000 Epochs for the Oklahoma African Datasets

Figures 5.19-5.20 show the losses of the African sub-population models. These show similar patterns to the full population losses, however the loss of the CE function in the Oklahoma dataset is much higher, likely again due to being stuck in a local minimum.



a) Texas African Loss b) Texas African Validation Loss

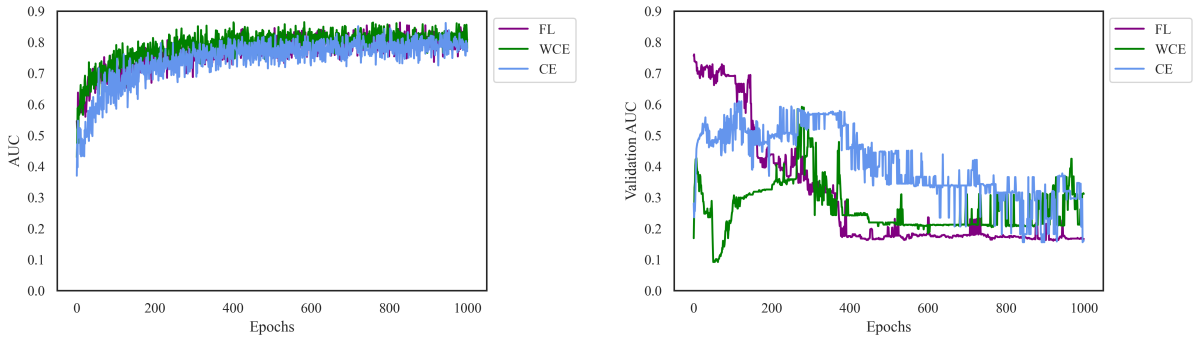
Figure 5.19: The Loss over 1000 Epochs for the Texas African American Datasets



a) Oklahoma African Loss b) Oklahoma African Validation Loss

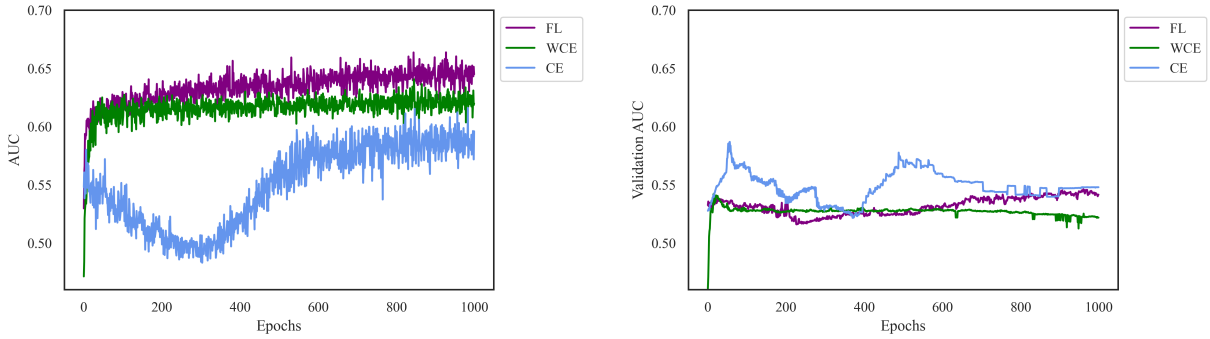
Figure 5.20: The Loss over 1000 Epochs for the Oklahoma African Datasets

The Native American sub-populations had the most variation out of any of the datasets, which can be attributed to their comparatively smaller sizes. Figures 5.21-5.22 show the AUC of the Texas and Oklahoma Native sub-populations. The Oklahoma Native CE training AUC seems to be stuck in a local minimum for over 200 epochs before correcting, and the validation graph shows a much higher variation than in larger populations. The Texas Native sub-population does not have this dip, but does seem to overfit more significantly than in larger populations. The focal loss function in this population seems to outperform the other two loss functions at least initially before dropping to the worst performing function, showing again a tendency to overfit.



a) Texas Native AUC b) Texas Native Validation AUC

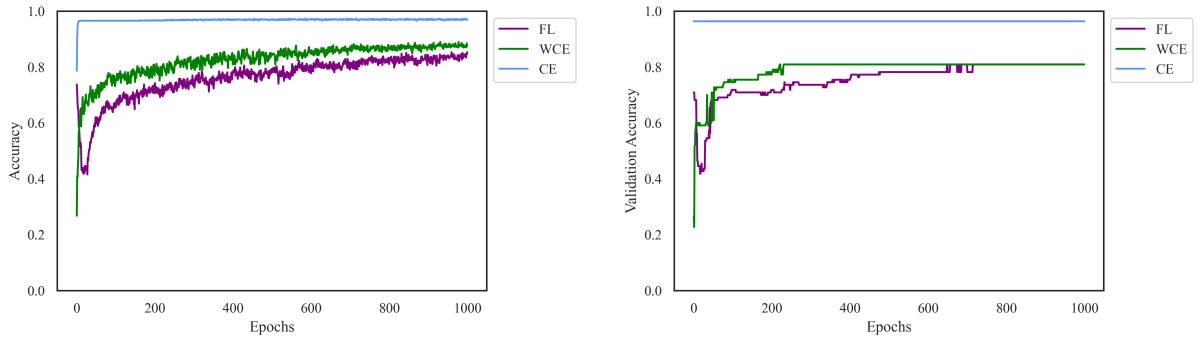
Figure 5.21: The AUC over 1000 Epochs for the Texas Native American Datasets



a) Oklahoma Native AUC b) Oklahoma Native Validation AUC

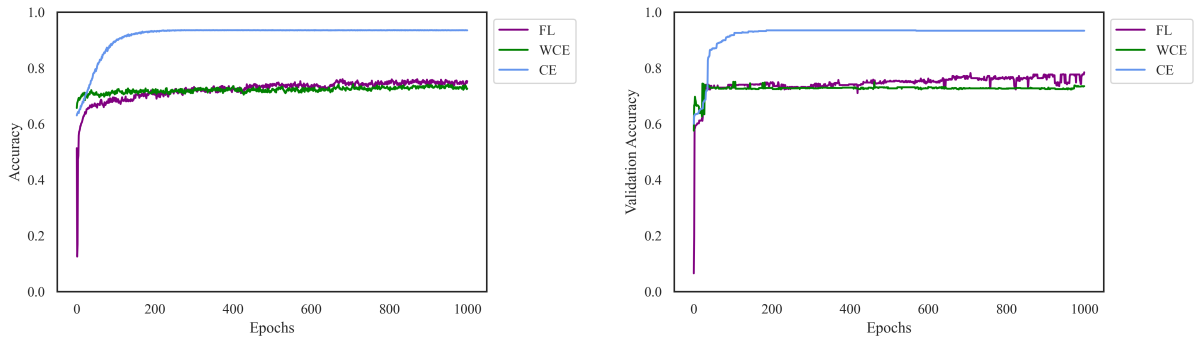
Figure 5.22: The AUC over 1000 Epochs for the Oklahoma Native Datasets

Accuracy among the Native American sub-populations also show a similar trend to the larger populations, although there seems to be a larger amount of time before the CE loss function stabilizes.



a) Texas Native Accuracy b) Texas Native Validation Accuracy

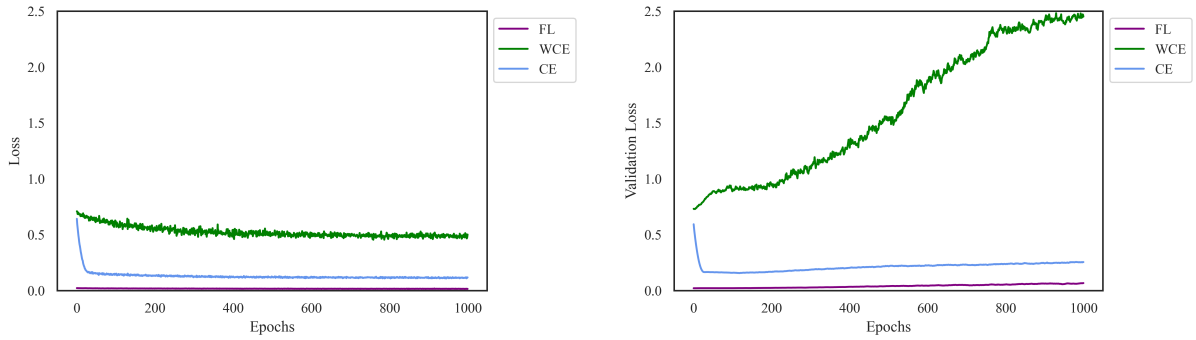
Figure 5.23: The Accuracy over 1000 Epochs for the Texas Native Datasets



a) Oklahoma Native Accuracy b) Oklahoma Native Validation Accuracy

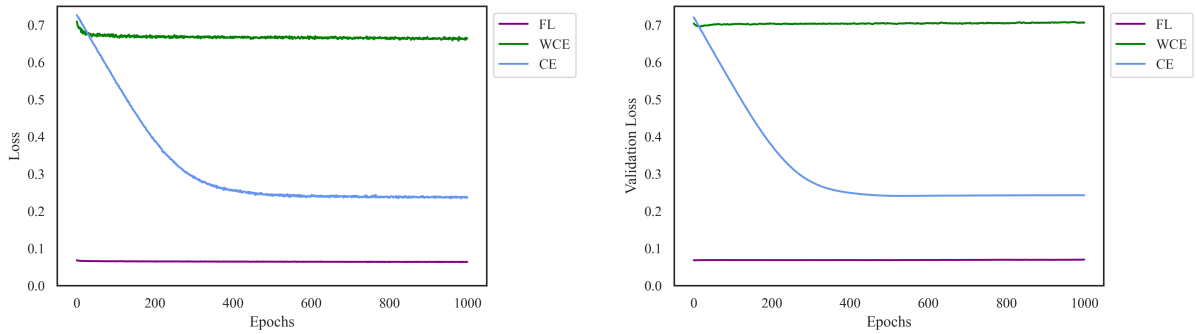
Figure 5.24: The Accuracy over 1000 Epochs for the Oklahoma Native Datasets

The loss of the Texas Native American sub-population has more significant overfitting, at least in the case of the weighted CE loss. The Oklahoma Native American loss differs from the larger populations in the length of time before stabilizing, taking around 400 epochs to stabilize.



a) Texas Native Loss b) Texas Native Validation Loss

Figure 5.25: The Loss over 1000 Epochs for the Texas Native Datasets



a) Oklahoma Native Loss b) Oklahoma Native Validation Loss

Figure 5.26: The Loss over 1000 Epochs for the Oklahoma Native Datasets

5.5 Comparative analysis with traditional ML algorithms

Tables 5.14 and 5.15 show the results of different traditional machine learning algorithms on the Texas and Oklahoma datasets respectively. The models tested were logistic regression, support vector machines with a linear kernal and radial basis function, and weighted versions of each of those models. In all cases the weighted versions outperformed in terms of both AUC and G-mean, however the best performing model was the cost sensitive neural network with Focal Loss. Additionally, focal loss once more had the smallest variation in g-means among the best performing models, showing that it is more robust than other machine learning algorithms.

Figure 5.29 shows the ROC curve for each of these models. These graphs show a slight improvement over other traditional algorithms, although in all datasets neural networks tend to perform similarly regardless of the loss function used (in terms of ROC-AUC).

Table 5.14: Mean g-means AUCs of Texas data using Logistic Regression (LR), Weighted LR, Support Vector Machine (SVM-Lin), Weighted SVM-Lin, SVM with Radial Basis Function (SVM-RBF), Weighted SVM-RBF, Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), and CSDNN with Focal Loss (CSDNN-Focal)

	LR	WLR	SVM-Lin	WSVM-Lin	SVM-RBF	WSVM-RBF	DNN	CSDNN-WCE	CSDNN (Focal)
G-mean	0.013	0.579	0.000	0.523	0.329	0.607	0.344	0.590	**0.663**
AUC	0.500	0.596	0.500	0.605	0.553	0.621	0.661	0.663	**0.663**

Table 5.15: Mean g-means and AUCs of Oklahoma data using of Logistic Regression (LR), Weighted LR, Support Vector Machine (SVM-Lin), Weighted SVM-Lin, SVM with Radial Basis Function (SVM-RBF), Weighted SVM-RBF, Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), and CSDNN with Focal Loss (CSDNN-Focal)

	LR	WLR	SVM-Lin	WSVM-Lin	SVM-RBF	WSVM-RBF	DNN	CSDNN-WCE	CSDNN (Focal)
G-mean	0.012	0.576	0.000	0.515	0.000	0.561	0.001	0.575	**0.594**
AUC	0.500	0.596	0.500	0.579	0.500	0.582	0.661	0.620	**0.635**

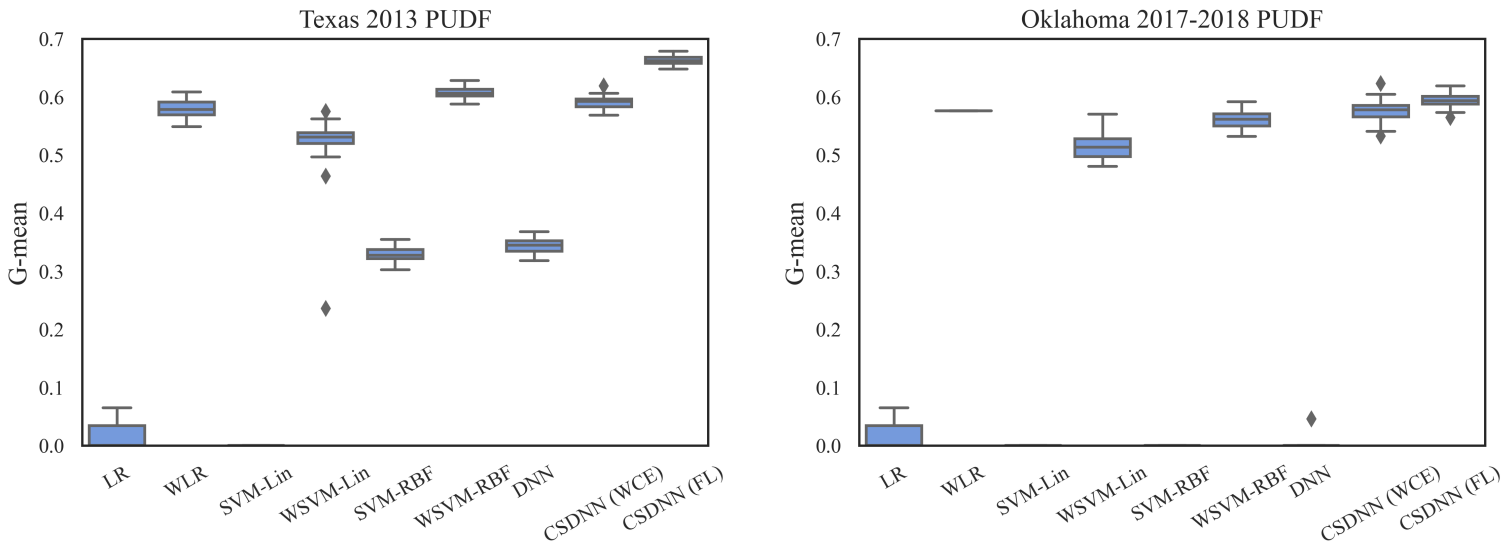


Figure 5.27: - Left: distribution of G-means in Texas dataset; Right: distribution of G-means in Oklahoma dataset

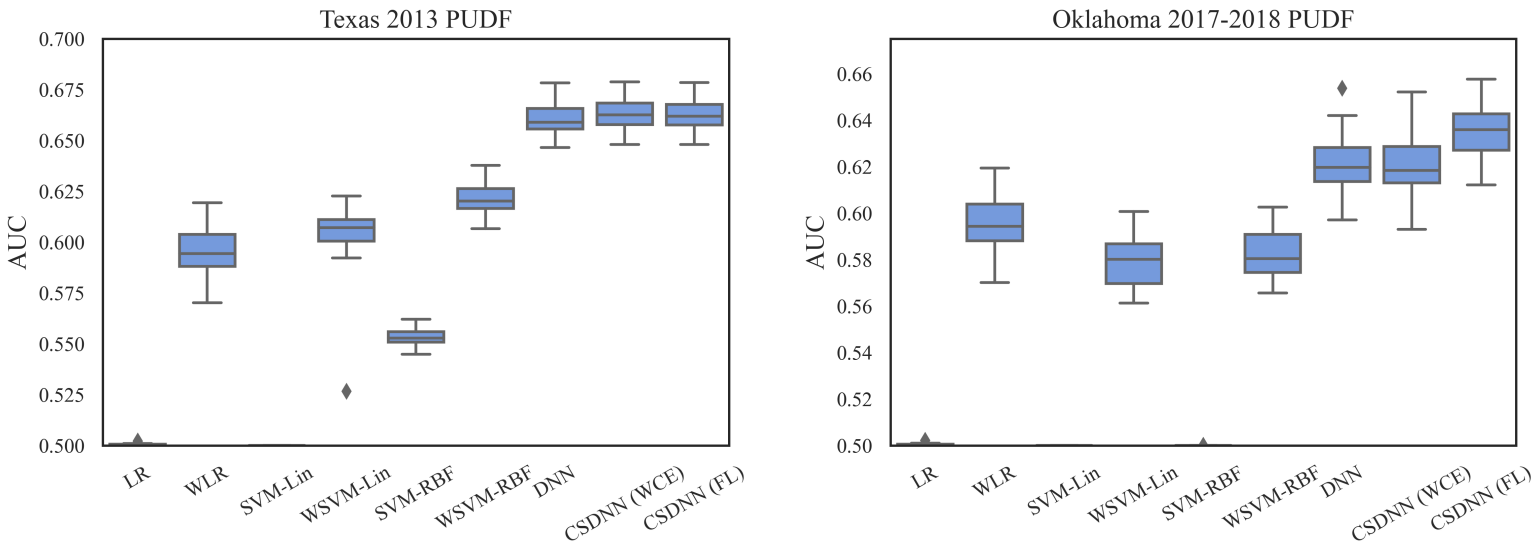


Figure 5.28: - Left: distribution of AUCs in Texas dataset; Right: distribution of AUCs in Oklahoma dataset

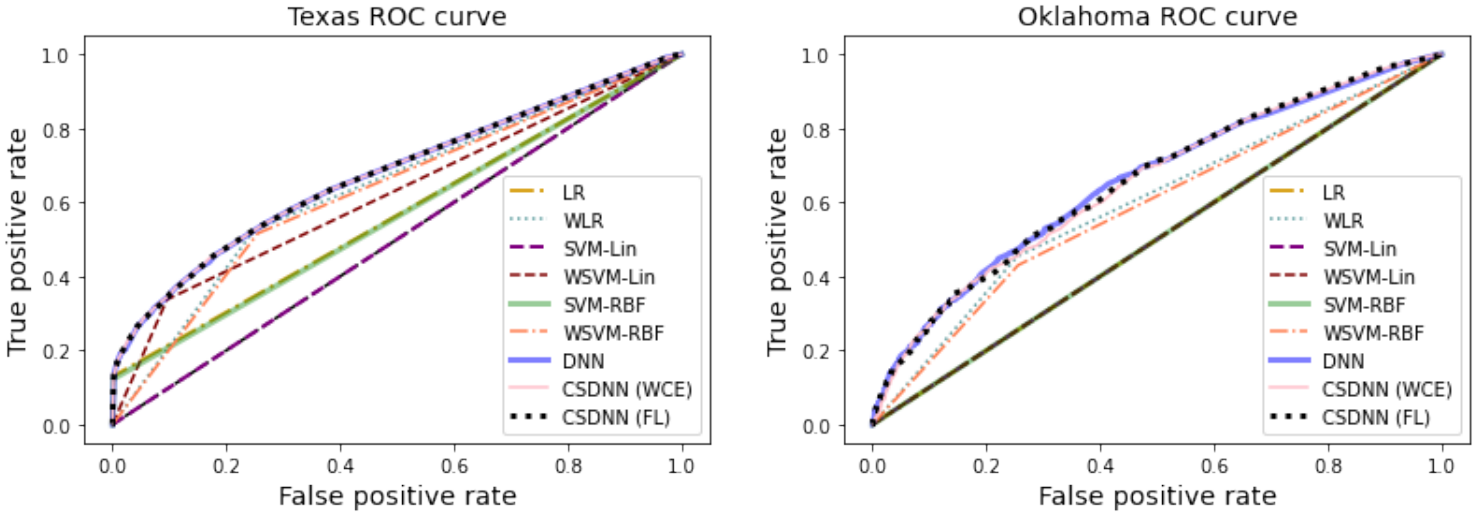


Figure 5.29: - Left: ROC curve for the Texas dataset; Right: ROC curve for the Oklahoma dataset

5.6 Statistical Analysis of Results

In order to test whether there was a statistical difference between any of the models, a Kruskal-Wallis test was performed for each dataset. In both cases, a statistical difference was found with an extremely low p-value, meaning we can reject the null at the 5% significance level.

Table 5.16: Kruskal-Wallis Test Results for Oklahoma Models

Comparison	**p-value**	**Hypothesis** ($\alpha = 0.05$)
Texas Models	$\ll 0.05$	Rejected H_0
Oklahoma Models	$\ll 0.05$	Rejected H_0

In order to then test if our cost-sensitive models (CSDNN-Focal and CSDNN-WCE) outperformed the others with statistical significance, we performed the one-tailed Wilcoxon rank-sum test to compare the g-means collected through our 10-fold cross validation repeated 5 times. Since these tests needed to be performed multiple times, the family-wise error rate was taken into account by reducing the significance level to 0.0005.

Tables 5.17-5.18 show the results of a one-tailed Wilcoxon rank-sum test on the Texas and Oklahoma full datasets respectively using each of the previously tested methods. The results of these tests show that CSDNN-Focal outperforms every other method with statistical significance in both datasets, while CSDNN-WCE outperforms most methods, with the exception of WSVM-RBF in the Texas dataset and WLR, FL-BB, and CE-BB in the Oklahoma dataset.

Table 5.17: Wilcoxon Test Results for Texas data using Logistic Regression (LR), Weighted LR, Support Vector Machine (SVM-Lin), Weighted SVM-Lin, SVM with Radial Basis Function (SVM-RBF), Weighted SVM-RBF, Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), and CSDNN with Focal Loss (CSDNN-Focal)

Comparison	**p-value**	**Hypothesis** ($\alpha = 0.0005$)
CSDNN-Focal $>$ **CSDNN-WCE**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **DNN**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **LR**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **WLR**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **SVM-Lin**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **WSVM-Lin**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **SVM-RBF**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **WSVM-RBF**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **FL-BB**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **WCE-BB**	$\ll 0.0005$	Rejected H_0
CSDNN-Focal $>$ **CE-BB**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **DNN**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **LR**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **WLR**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **SVM-Lin**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **WSVM-Lin**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **SVM-RBF**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **WSVM-RBF**	0.999	Did not reject H_0
CSDNN-WCE $>$ **FL-BB**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **WCE-BB**	$\ll 0.0005$	Rejected H_0
CSDNN-WCE $>$ **CE-BB**	$\ll 0.0005$	Rejected H_0

Table 5.18: Wilcoxon Test Results for Oklahoma data using Logistic Regression (LR), Weighted LR, Support Vector Machine (SVM-Lin), Weighted SVM-Lin, SVM with Radial Basis Function (SVM-RBF), Weighted SVM-RBF, Deep Neural Network (DNN), Cost-Sensitive DNN with weighted cross-entropy (CSDNN-WCE), CSDNN with Focal Loss (CSDNN-Focal), CSDNN with Focal Loss and balanced batches (FL-BB), CSDNN with weighted cross-entropy and balanced batches (WCE-BB), and DNN with weighted cross-entropy and balanced batches (CE-BB)

Comparison	**p-value**	$\alpha = 0.0005$
CSDNN-Focal > CSDNN-WCE	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > LR	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WLR	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > SVM-Lin	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WSVM-Lin	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > SVM-RBF	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WSVM-RBF	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > CE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > LR	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > WLR	0.630	Did not reject H_0
CSDNN-WCE > SVM-Lin	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > WSVM-Lin	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > SVM-RBF	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > WSVM-RBF	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > FL-BB	0.999	Did not reject H_0
CSDNN-WCE > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > CE-BB	0.989	Did not reject H_0

Tables 5.19-5.20 show the results of the one-tailed Wilcoxon rank test for the Texas and

Oklahoma African American datasets respectively. In these groups, CSDNN-Focal outperformed every other method tested with statistical significance with the exception of the Oklahoma African American dataset, in which we could not reject the null that CSDNN-Focal did not perform better than CSDNN-WCE. As CSDNN-WCE, we were able to show that it outperformed the other methods tested with the exception of FL-BB and CE-BB in the Texas African American dataset.

Table 5.19: Wilcoxon Test Results for Texas African American data using Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), CSDNN with Focal Loss (CSDNN-Focal), CSDNN with Focal Loss and balanced batches (FL-BB), CSDNN with weighted cross-entropy and balanced batches (WCE-BB), and DNN with weighted cross-entropy and balanced batches (CE-BB)

Comparison	**p-value**	$\alpha = 0.0005$
CSDNN-Focal > CSDNN-WCE	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > CE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > FL-BB	0.999	Did not reject H_0
CSDNN-WCE > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > CE-BB	0.999	Did not reject H_0

Table 5.20: Wilcoxon Test Results for Oklahoma African American data using Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), CSDNN with Focal Loss (CSDNN-Focal), CSDNN with Focal Loss and balanced batches (FL-BB), CSDNN with weighted cross-entropy and balanced batches (WCE-BB), and DNN with weighted cross-entropy and balanced batches (CE-BB)

Comparison	**p-value**	$\alpha = 0.0005$
CSDNN-Focal > CSDNN-WCE	0.862	Did not reject H_0
CSDNN-Focal > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > CE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > CE-BB	$\ll 0.0005$	Rejected H_0

The results of the one-tailed Wilcoxon rank-sum tests are presented in tables 5.21-5.22 for the Texas and Oklahoma Native American datasets respectively. In these cases, the null was only rejected in two instances. In the Texas Native American population, we were unable to reject the null that CSDNN-WCE did not perform better than CE-BB. In the Oklahoma Native American population, we were unable to reject the null that CSDNN-Focal did not perform better than CSDNN-WCE. In both datasets however, CSDNN-Focal outperformed all the other methods with statistical significance, holding to the same trend present in every other dataset.

Table 5.21: Wilcoxon Test Results for Texas Native American data using Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), CSDNN with Focal Loss (CSDNN-Focal), CSDNN with Focal Loss and balanced batches (FL-BB), CSDNN with weighted cross-entropy and balanced batches (WCE-BB), and DNN with weighted cross-entropy and balanced batches (CE-BB)

Comparison	**p-value**	$\alpha = 0.0005$
CSDNN-Focal > CSDNN-WCE	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > CE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > CE-BB	0.141	Did not reject H_0

Table 5.22: Wilcoxon Test Results for Oklahoma Native American data using Deep Neural Network (DNN), Cost-Sensitive DNN with weighed cross-entropy (CSDNN-WCE), CSDNN with Focal Loss (CSDNN-Focal), CSDNN with Focal Loss and balanced batches (FL-BB), CSDNN with weighted cross-entropy and balanced batches (WCE-BB), and DNN with weighted cross-entropy and balanced batches (CE-BB)

Comparison	**p-value**	$\alpha = 0.0005$
CSDNN-Focal > CSDNN-WCE	0.007	Did not reject H_0
CSDNN-Focal > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-Focal > CE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > DNN	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > FL-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > WCE-BB	$\ll 0.0005$	Rejected H_0
CSDNN-WCE > CE-BB	0.0004	Rejected H_0

The results of the one-tailed Wilcoxon rank-sum tests for the African American populations are presented in tables 5.23 and 5.24. These show that at the 5% significance level, the specific models outperform the general and full population models in all cases except for the Oklahoma African American population.

Table 5.23: Wilcoxon Test results for Specific Features vs. General Features and Full population models in the Texas African Dataset

Comparison	**p-value**	**Hypothesis** ($\alpha = 0.0005$)
Specific $>$ **General**	$\ll 0.0005$	Rejected H_0
Specific $>$ **Full Pop**	0.044	Did not reject H_0

Table 5.24: Wilcoxon Test results for Specific Features vs. General Features and Full population models in the Oklahoma African Dataset

Comparison	**p-value**	**Hypothesis** ($\alpha = 0.0005$)
Specific $>$ **General**	0.433	Did not reject H_0
Specific $>$ **Full Pop**	$\ll 0.0005$	Rejected H_0

The results for the Native American subpopulations are shown in tables 5.25 and 5.26. These results are less significant, where the only case we were able to reject the null hypothesis was in the case of the Oklahoma Native American Specific model when compared to the Full Population model.

Table 5.25: Wilcoxon Test results for Specific Features vs. General Features and Full population models in the Texas Native Dataset

Comparison	**p-value**	**Hypothesis** ($\alpha = 0.0005$)
Specific $>$ **General**	0.005	Did not reject H_0
Specific $>$ **Full Pop**	0.999	Did not reject H_0

Table 5.26: Wilcoxon Test results for Specific Features vs. General Features and Full population models in the Oklahoma Native Dataset

Comparison	**p-value**	**Hypothesis** ($\alpha = 0.0005$)
Specific $>$ **General**	0.149	Did not reject H_0
Specific $>$ **Full Pop**	$\ll 0.0005$	Rejected H_0

Chapter 6

Conclusions

This work explores the use of cost-sensitive neural networks in the case of preeclampsia prediction, and although the results are not quite accurate enough for use in a clinical setting, they show that there is an improvement in the results when either focal loss or weighted cross entropy are used as the loss functions. Focal Loss in particular, which prior to this study had only been used on image data, was shown to outperform any other method tested in almost all studied datasets, and there was significantly less variation in its results than in any other methods, showing that it can be a much more robust model and its performance is less dependent on how a dataset is split.

Additionally, we have tested the use of models built for specific sub-populations and compared them to more traditional models that are built using the entire dataset but used on specific minority populations. Although we were unable to show a statistically significant improvement in all of the cases tested, our results show an improvement by employing these methods, therefore, we conclude that it could be worth exploring the specific sub-population datasets if a larger dataset or a higher quality data were available.

The work in this thesis can be generalized to other highly imbalanced problems. As stated earlier, a large proportion of real world problems are highly imbalanced, and other techniques such as over and undersampling run the risk of changing the distribution of the

dataset. The work here provides researchers with ways of addressing the imbalancedness in their datasets without changing the distribution, resulting in models that generalize better to unseen data. Additionally, the research into minority specific models can be extended towards other health problems in which disparity in outcomes is an issue.

Acknowledgements

This work was supported by the Vice President for Research and Partnerships of the University of Oklahoma. We acknowledge the source of the data in this project: Oklahoma Discharge Public Use Data file, Health Care Information Division, Oklahoma State Department of Health and the Texas Department of State Health Services (DSHS).

Bibliography

Abdel-Hamid Ossama, Mohamed Abdel-rahman, Jiang Hui, Deng Li, Penn Gerald, Yu Dong. Convolutional neural networks for speech recognition // IEEE/ACM Transactions on audio, speech, and language processing. 2014. 22, 10. 1533–1545.

Admon Lindsay K, Winkelman Tyler NA, Zivin Kara, Terplan Mishka, Mhyre Jill M, Dalton Vanessa K. Racial and ethnic disparities in the incidence of severe maternal morbidity in the United States, 2012–2015 // Obstetrics & Gynecology. 2018. 132, 5. 1158–1166.

Bahdanau Dzmitry, Cho Kyunghyun, Bengio Yoshua. Neural machine translation by jointly learning to align and translate // arXiv preprint arXiv:1409.0473. 2014.

Bellamy Leanne, Casas Juan-Pablo, Hingorani Aroon D, Williams David J. Pre-eclampsia and risk of cardiovascular disease and cancer in later life: systematic review and meta-analysis // Bmj. 2007. 335, 7627. 974.

Bergstra James, Bengio Yoshua. Random search for hyper-parameter optimization // The Journal of Machine Learning Research. 2012. 13, 1. 281–305.

Blanco Antonio, Pino-Mejías Rafael, Lara Juan, Rayo Salvador. Credit scoring models for the microfinance industry using neural networks: Evidence from Peru // Expert Systems with applications. 2013. 40, 1. 356–364.

Boghossian Nansi S, Yeung Edwina, Mendola Pauline, Hinkle Stefanie N, Laughon S Katherine, Zhang Cuilin, Albert Paul S. Risk factors differ between recurrent and incident

preeclampsia: a hospital-based cohort study // Annals of epidemiology. 2014. 24, 12. 871–877.

Bottou Léon. Online learning and stochastic approximations // On-line learning in neural networks. 1998. 17, 9. 142.

Breathett Khadijah, Muhlestein David, Foraker Randi, Gulati Martha. Differences in preeclampsia rates between African American and Caucasian women: trends from the National Hospital Discharge Survey // Journal of women's health. 2014. 23, 11. 886–893.

Bujold Emmanuel, Roberge Stephanie, Lacasse Yves, Bureau Marc, Audibert Francois, Marcoux Sylvie, Forest Jean-Claude, Giguere Yves. Prevention of preeclampsia and intrauterine growth restriction with aspirin started in early pregnancy: a meta-analysis // Obstetrics & Gynecology. 2010. 116, 2. 402–414.

Buuran Stef van, Groothuis-Oudshoorn Karin. MICE: Multivariate Imputation by Chained Equations in R // J. Stat. Softw. 2010. 1–68.

Chawla Nitesh V, Bowyer Kevin W, Hall Lawrence O, Kegelmeyer W Philip. SMOTE: synthetic minority over-sampling technique // Journal of artificial intelligence research. 2002. 16. 321–357.

Cho Kyunghyun, Van Merriënboer Bart, Bahdanau Dzmitry, Bengio Yoshua. On the properties of neural machine translation: Encoder-decoder approaches // arXiv preprint arXiv:1409.1259. 2014.

Chollet François, others . Keras. 2015.

Dastile Xolani, Celik Turgay, Potsane Moshe. Statistical and machine learning models in credit scoring: A systematic literature survey // Applied Soft Computing. 2020. 91. 106263.

Incorporating nesterov momentum into adam. // . 2016.

Du Guodong, Zhang Jia, Li Shaozi, Li Candong. Learning from class-imbalance and heterogeneous data for 30-day hospital readmission // Neurocomputing. 2021. 420. 27–35.

Duley Lelia. The global impact of pre-eclampsia and eclampsia // Seminars in perinatology. 33, 3. 2009. 130–137.

Fernández Alberto, Garcia Salvador, Herrera Francisco, Chawla Nitesh V. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary // Journal of artificial intelligence research. 2018. 61. 863–905.

Frazier Peter I. A tutorial on Bayesian optimization // arXiv preprint arXiv:1807.02811. 2018.

Fuqua Donovan, Razzaghi Talayeh. A cost-sensitive convolution neural network learning for control chart pattern recognition // Expert Systems with Applications. 2020. 150. 113275.

Gardner Matt W, Dorling SR. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences // Atmospheric environment. 1998. 32, 14-15. 2627–2636.

Goodfellow Ian, Bengio Yoshua, Courville Aaron, Bengio Yoshua. Deep learning. 1, 2. 2016.

Graves Alex, Mohamed Abdel-rahman, Hinton Geoffrey. Speech recognition with deep recurrent neural networks // 2013 IEEE international conference on acoustics, speech and signal processing. 2013. 6645–6649.

He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification // Proceedings of the IEEE international conference on computer vision. 2015. 1026–1034.

Heck Jennifer L, Jones Emily J, Bohn Diane, McCage Shondra, Parker Judy Goforth, Parker Mahate, Pierce Stephanie L, Campbell Jacquelyn. Maternal mortality among American Indian/Alaska Native women: A scoping review // Journal of Women's Health. 2020.

Johnson Jasmine D, Louis Judette M. Does race or ethnicity play a role in the origin, pathophysiology, and outcomes of preeclampsia? An expert review of the literature // American journal of obstetrics and gynecology. 2020.

Joshi Dipali M, Rana NK, Misra VMi. Classification of brain cancer using artificial neural network // 2010 2nd International Conference on Electronic Computer Technology. 2010. 112–116.

Karabatak Murat, Ince M Cevdet. An expert system for detection of breast cancer based on association rules and neural network // Expert systems with Applications. 2009. 36, 2. 3465–3469.

Kenny Louise C., Black Michael A., Poston Lucilla, Taylor Rennae, Myers Jenny E., Baker Philip N., McCowan Lesley M., Simpson Nigel A.B., Dekker Gus A., Roberts Claire T., Rodems Kelline, Noland Brian, Raymundo Michael, Walker James J., North Robyn A. Early pregnancy prediction of preeclampsia in nulliparous women, combining clinical risk and biomarkers: The Screening for Pregnancy Endpoints (SCOPE) international cohort study // Hypertension. 2014. 64, 3. 644–652.

Khan Salman H, Hayat Munawar, Bennamoun Mohammed, Sohel Ferdous A, Togneri Roberto. Cost-sensitive learning of deep feature representations from imbalanced data // IEEE transactions on neural networks and learning systems. 2017. 29, 8. 3573–3587.

Kingma Diederik P, Ba Jimmy. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. 2014.

Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. 25. 1097–1105.

Kukar M, Kononenko Igor. Cost-sensitive learning with neural networks // 13th Eur. Conf. Artif. Intell. 1998. 445–449.

LeCun Yann, Bottou Léon, Bengio Yoshua, Haffner Patrick. Gradient-based learning applied to document recognition // Proceedings of the IEEE. 1998. 86, 11. 2278–2324.

Leevy Joffrey L, Khoshgoftaar Taghi M, Bauder Richard A, Seliya Naeem. A survey on addressing high-class imbalance in big data // J. Big Data. 2018a. 5, 1.

Leevy Joffrey L, Khoshgoftaar Taghi M, Bauder Richard A, Seliya Naeem. A survey on addressing high-class imbalance in big data // Journal of Big Data. 2018b. 5, 1. 1–30.

Lemaître Guillaume, Nogueira Fernando, Aridas Christos K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning // Journal of Machine Learning Research. 2017. 18, 17. 1–5.

Li Lisha, Jamieson Kevin, DeSalvo Giulia, Rostamizadeh Afshin, Talwalkar Ameet. Hyperband: A novel bandit-based approach to hyperparameter optimization // J. Mach. Learn. Res. 2018. 18. 1–52.

Lin Tsung-Yi, Goyal Priya, Girschick Ross, He Kaiming, Dollar Piotr. Retinane // roceedings IEEE Int. Conf. Comput. Vis. 2017. 2980–2988.

Little Roderick JA. A test of missing completely at random for multivariate data with missing values // Journal of the American statistical Association. 1988. 83, 404. 1198–1202.

Maqsood Imran, Khan Muhammad Riaz, Abraham Ajith. An ensemble of neural networks for weather forecasting // Neural Computing & Applications. 2004. 13, 2. 112–122.

Marić Ivana, Tsur Abraham, Aghaeepour Nima, Montanari Andrea, Stevenson David K., Shaw Gary M., Winn Virginia D. Early prediction of preeclampsia via machine learning // Am. J. Obstet. Gynecol. MFM. 2020. 2, 2. 100100.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado S., Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz,

Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015.

Mbah Alfred K, Alio Amina P, Marty Phillip J, Bruder Karen, Wilson R, Salihu Hamisu M. Recurrent versus isolated pre-eclampsia and risk of feto-infant morbidity outcomes: racial/ethnic disparity // European Journal of Obstetrics & Gynecology and Reproductive Biology. 2011. 156, 1. 23–28.

McCulloch Warren S, Pitts Walter. A logical calculus of the ideas immanent in nervous activity // The bulletin of mathematical biophysics. 1943. 5, 4. 115–133.

Moreira Mario W.L., Rodrigues Joel J.P.C., Oliveira Antonio M.B., Saleem Kashif, Neto Augusto J.Venancio. Predicting hypertensive disorders in high-risk pregnancy using the random forest approach // IEEE Int. Conf. Commun. 2017. October. 5081–5085.

O'Malley Tom, Bursztein Elie, Long James, Chollet François, Jin Haifeng, Invernizzi Luca, others . Keras Tuner. 2019.

Rosenblatt Frank. The perceptron: a probabilistic model for information storage and organization in the brain. // Psychological review. 1958. 65, 6. 386.

Rosenblatt Frank. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. 1961.

Rubin Donald B. Inference and missing data // Biometrika. 1976. 63, 3. 581–592.

Rumelhart David E, Hinton Geoffrey E, Williams Ronald J. Learning representations by back-propagating errors // nature. 1986. 323, 6088. 533–536.

Sacks Kira Nahum, Friger Michael, Shoham-Vardi Ilana, Spiegel Efrat, Sergienko Ruslan, Landau Daniella, Sheiner Eyal. Prenatal exposure to preeclampsia as an independent risk factor for long-term cardiovascular morbidity of the offspring // Pregnancy hypertension. 2018. 13. 181–186.

Sandström Anna, Snowden Jonathan M, Höijer Jonas, Bottai Matteo, Wikström Anna-Karin. Clinical risk assessment in early pregnancy for preeclampsia in nulliparous women: A population based cohort study // PloS one. 2019. 14, 11. e0225716.

Shahul Sajid, Tung Avery, Minhaj Mohammed, Nizamuddin Junaid, Wenger Julia, Mahmood Eitezaz, Mueller Ariel, Shaefi Shahzad, Scavone Barbara, Kociol Robb D, others . Racial disparities in comorbidities, complications, and maternal and fetal outcomes in women with preeclampsia/eclampsia // Hypertension in pregnancy. 2015. 34, 4. 506–515.

Sharma Sagar. Activation functions in neural networks // towards data science. 2017. 6.

Snoek Jasper, Larochelle Hugo, Adams Ryan P. Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. 2012. 25. 2951–2959.

Strobl Carolin, Boulesteix Anne-Laure, Zeileis Achim, Hothorn Torsten. Bias in random forest variable importance measures: Illustrations, sources and a solution // BMC bioinformatics. 2007. 8, 1. 1–21.

Sufriyana Herdiantri, Wu Yu Wei, Su Emily Chia Yu. Artificial intelligence-assisted prediction of preeclampsia: Development and external validation of a nationwide health insurance dataset of the BPJS Kesehatan in Indonesia // EBioMedicine. 2020. 54.

Texas Department of State Health Services . Department of State Health Services Map of Border Area. 2021. [Online]. Available from: `https://www.dshs.texas.gov/borderhealth/border_health_map.shtm`.

Tieleman Tijmen, Hinton Geoffrey. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning // COURSERA Neural Networks Mach. Learn. 2012.

Tipping Michael E. Sparse Bayesian learning and the relevance vector machine // Journal of machine learning research. 2001. 1, Jun. 211–244.

Virtanen Pauli, Gommers Ralf, Oliphant Travis E., Haberland Matt, Reddy Tyler, Cournapeau David, Burovski Evgeni, Peterson Pearu, Weckesser Warren, Bright Jonathan, van der Walt Stéfan J., Brett Matthew, Wilson Joshua, Millman K. Jarrod, Mayorov Nikolay, Nelson Andrew R. J., Jones Eric, Kern Robert, Larson Eric, Carey C J, Polat İlhan, Feng Yu, Moore Eric W., VanderPlas Jake, Laxalde Denis, Perktold Josef, Cimrman Robert, Henriksen Ian, Quintero E. A., Harris Charles R., Archibald Anne M., Ribeiro Antônio H., Pedregosa Fabian, van Mulbregt Paul, SciPy 1.0 Contributors . SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python // Nature Methods. 2020. 17. 261–272.

World Health Organization . International classification of diseases : [9th] ninth revision, basic tabulation list with alphabetic index. 1978. [Online]. Available from: `https://apps.who.int/iris/handle/10665/39473`.

World Health Organization . ICD-10 : international statistical classification of diseases and related health problems : tenth revision, 2nd ed. 2004. [Online]. Available from: `https://apps.who.int/iris/handle/10665/42980`.

Yavuz Erdem, Eyupoglu Can, Sanver Ufuk, Yazici Rifat. An ensemble of neural networks for breast cancer diagnosis // 2017 International Conference on Computer Science and Engineering (UBMK). 2017. 538–543.