UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

NBA TICKET PACKAGE OPTIMIZATION, A CASE STUDY OF THE CLEVELAND CAVALIERS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of
MASTERS OF SCIENCE

By
Benjamin Levicki
Norman, Oklahoma
2024

NBA TICKET PACKAGE OPTIMIZATION, A CASE STUDY OF THE CLEVELAND CAVALIERS

A THESIS APPROVED FOR THE
GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Charles Nicholson, Chair

Dr. Randa L. Shehab

Dr. Matthew Beattie

To Grandpa Bill,

who introduced me to the world of sports and statistics

Acknowledgements

This thesis is really a testament to both the astounding support from both Oklahoma University and the Cleveland Cavaliers. From Dr. Shehab who set the wheels in motion to make this thesis a possibility for me, to Dr. Nicholson, who graciously let me take on this feat remotely and wanted to be my chair. And finally, Dr. Beattie, who I'm honored to be the first student he's been on the committee for. The feedback I've received on this work has helped me learn valuable lessons when it comes to this style of writing.

Then, taking this to the Cavaliers as a possibility to conduct research on their behalf was welcomed with open arms. When I brought this idea to my team lead, Kevin O'Toole, he jumped on the opportunity to allow me to pursue my passions in research. Then, with support from our legal team, including non-other than Kerry Barrett, I was able to produce this work in the publishable state. And finally, thanks to Canaan Campo and Quinn Spangler for being the subject matter experts who helped me inject the business specific knowledge to take this from a purely theoretical idea, to one that has some legs in the business.

I mean if I were to tell myself over two and half years ago when I decided on this program that this would be how this degree is capped off, I would've been in disbelief. Where this program has brought me is something that I once dreamed about and to be where I'm at today is what I'm the most thankful of.

I wouldn't have been here without the support of my family. From the moment I brought up the idea of doing grad school, to the day I decided to pick up everything and leave home to move to Cleveland to work for the Cavs in a two-week span, they have been there for me throughout. I couldn't have been here without their council throughout many of my life decisions.

Table of Contents

List of Tables

List of Figures

Abstract

The purpose of this study is to create a genetic algorithm to further enhance the current half season ticket packaging process in the event industry through a case study with the Cleveland Cavaliers of the National Basketball Association. The focus of this study is on the integration of machine learning and heuristic methods to simulate the human decision making currently taking place across the industry. This study will cover the methodologies being proposed for the overall, integrated approach. The methods that we cover in this study surround the tiering of events using K-Medoids and the makeup of the genetic algorithm that was implemented to solve for optimal half season packages using the Cleveland Cavaliers home schedule. Then, using the interaction between these two methodologies, we analyze the results in collaboration with domain experts from the Cavaliers. This study will show how the usage of machine learning paired with a genetic algorithm can affectively simulate and improve upon the current process for determining half season ticket packages. Furthermore, future improvements, such as the addition of predictive analytics and fan behavior are explored to supplement this work and lead to future areas of research and development.

Chapter 1: Introduction

The entertainment and event industry has been a hallmark of recreation for people worldwide. From large venue concerts to sporting events, there exists a wide array of events that have varying appeal to consumers. Some patrons have their preferences for entertainment and choose to purchase tickets in bundles to save money and secure their spot at their favorite events. The National Basketball Association (NBA) is a multi-billion dollar-industry, with ticket sales serving as a crucial revenue stream for teams (Forbes, 2023). As one of the most successful franchises in the league, the Cleveland Cavaliers (Cavs) provide an excellent case study for examining the intricacies of ticket sales and the potential for optimization. In an increasingly competitive market, sports organizations must adapt and innovate to maximize efficiency and enhance the fan buying experience.

Currently, NBA teams employ various ticketing strategies, such as season tickets, split season packages, and single-game sales. However, the strategies to create the split season packages often face limitations in terms of efficiency leading to suboptimal time to market after the season schedule is released. The process is rather manual, which opens the door for subjective opinions to be put into consideration rather than objective decisions driven by previous buying behavior. The approach fails to account for the holistic nature of the schedule due to the complexity and wide array of decisions that need to be made for a 40-game schedule. With each decision being multidimensional, it can become tough to analyze each one in depth in a standard fashion. The approach is collaborative, which should be celebrated in any business process, but with many decisions to be made, this causes an inefficient process that can take one to two weeks to make a final decision. While the process is important to improve upon, the end product is the true focus.

A season split package is when the home schedule for a team is split into 2 or more offerings for purchase by fans. Each package contains games that are dispersed throughout the entire season. When creating split season packages in half and quarter seasons, it is paramount to provide an equivalent value to consumers. Viewing the packages as separate inventory items, the goal is to make each package as appealing as the other to equalize consumer demand. The consequences of not doing so could lead to an underselling of certain packages, which can impact the revenue gained for the games within those packages. Additionally, if the higher demand value packages sell out it can leave consumers feeling slighted, leading to consumer dissatisfaction. Optimizing ticket packages is crucial for teams to maximize attendance, revenue, and fan satisfaction.

The advent of technology presents a unique opportunity for organizations to leverage advanced methods, such as machine learning and genetic algorithms, to dynamically optimize ticketing packages. By adapting to this new paradigm, teams can gain a significant competitive advantage and better serve their fans.

One key aspect of the proposed solution is tiering the games. This involves categorizing games based on their expected demand, such as high-demand marquee matchups and lower-demand regular-season games. Ticket tiering allows teams to price tickets more effectively, aligning with the perceived value of each game. Machine learning, a branch of artificial intelligence, can be applied to analyze historical data, such as past attendance, opponent strength, and day of the week, to predict demand and automatically assign games to appropriate tiers.

By harnessing the power of machine learning, NBA teams can make data-driven decisions and optimize their ticketing strategies. This benefits the organization through increased revenue and provides fans with more tailored and appealing ticket options. While machine learning offers a way to tier games effectively, the genetic algorithm brings a unique optimization solution to the table, enabling teams to create the most attractive and balanced ticket packages.

Genetic algorithms, inspired by the principles of natural selection, have proven to be powerful tools for solving complex optimization problems. These algorithms have found applications in various domains, such as supply chain management (Altiparmak et al., 2006), financial portfolio optimization (Metaxiotis & Liagkouras, 2012), and transportation routing (Jozefowiez et al., 2008). In the context of the NBA, a genetic algorithm can be employed to optimize the scheduling of ticket packages, ensuring an even distribution of high-demand and low-demand games within each package.

The interplay between machine learning for game tiering and genetic algorithms for package optimization offers a sophisticated and adaptive approach to solving the ticket package optimization problem in the NBA. This powerful combination has the potential to revolutionize ticketing strategies and set a new standard for this time-consuming process.

The genetic algorithm, as applied in this context, works by encoding potential package combinations as "chromosomes" and iteratively evolving them through selection, crossover, and mutation operations. The fitness of each package combination is evaluated based on criteria such as the balance of game tiers and the spacing between games, which contribute to the overall attractiveness to fans. At the time of this writing, the fitness evaluation metric presented in this research represents a novel approach in the field of optimization, offering a unique and comprehensive method for maximizing equivalence. Through successive generations, the algorithm converges towards an optimal set of ticket packages that helps maximize the revenue potential by providing equal package splits.

The significance of this study extends beyond the Cavs and the NBA, as the findings and methodologies presented here can be adapted and applied to other sports leagues and the broader event and entertainment industry. By showcasing the benefits of advanced optimization techniques, this study encourages organizations to embrace data-driven decision-making and innovative ticketing strategies. The successful implementation of these methods can lead to a more sustainable and efficient process, benefiting both organizations and their patrons. This application of a genetic algorithm, paired with a machine learning-based game tiering, presents a novel approach to optimizing ticket package combinations in the event and entertainment industry, offering opportunities for operational efficiencies and customer satisfaction, as demonstrated through a case study of the Cleveland Cavaliers' 2023-24 half season ticket packages.

Chapter 2: Literature Review

2.1 Sports Ticketing Analytics

2.1.1 Broader purposes and methodology

The act of searching for the most favorable deal while buying tickets for sporting events over a period of time is fundamentally comparable to a behavior observed in behavioral ecology known as foraging and patch exploitation (Ødegaard et al., 2023; Drayer et al., 2022). Optimization models have been utilized by behavioral ecologists to examine how animals seek out natural food resources at regular intervals while considering uncertain time and environmental constraints (Reese and Bennett, 2013). Similarly, consumers engage and search to make predictions and decisions on when to purchase tickets based on various indicators and evaluations of uncertainty in the pre-sale ticketing environment (Ødegaard et al., 2023).

Research in the field of sports has highlighted the significant level of uncertainty associated with pre-sale ticketing, which makes it inherently difficult to predict demand and prices for sports events (Jee and Hyun, 2023). This uncertainty is influenced not only by factors related to time and environmental variables (such as temperature, precipitation, time of the event, part of season, weekday/weekend, days before event, etc.) (Marquez, 2020), but also by dynamic team and individual performance factors (such as star player injuries, home and team winning percentage, season rankings, playoff contention, etc.) (Arslan et al., 2020; Solanellas et al. 2022), which have been found to have a significant impact on game attendance and ticket prices. Sports fans value the unpredictability of the product, which can change rapidly and frequently throughout the season, and this ultimately affects their perception of the product's usefulness.

Jee and Hyun (2023) suggest that given complete information, consumers would have access to all the relevant context they need to make decisions about when to purchase sports event tickets. However, due to inherent time and resource constraints, as well as natural cognitive limitations, consumers often use quick and automatic decision-making that are influenced by their intuition.

Drayer et al. (2022) note that the value of a sport ticket changes constantly, making it challenging for consumers to accurately determine the optimal time to purchase tickets for a sporting event. This dynamic pricing environment in sports ticketing introduces additional complexity and uncertainty for consumers.

The different consumers of sports products are further divided into various segments of sports enthusiasts who possess a comprehensive understanding of the product elements and exhibit a significant psychological and emotional connection to it . Passionate sports fans closely track daily game statistics, player salaries, and injury reports and have a profound emotional attachment to their team and players. This creates a fanatic culture in which die-hard sports fans consistently follow and discuss the latest updates and real-time information through dedicated sports media channels and social media platforms online.

Reese and Bennett (2013) found that the degree of fan involvement with their preferred sports team can impact their risk perceptions. A risk to the sports fan can be a star player sitting out or their favorite team not winning the game they choose to attend. High levels of fan involvement generally led to lower risk perceptions and higher levels of risk-taking behavior. This is frequently attributed to the emotional attachment that fans develop with their team, which can lead to a sense of invulnerability and a willingness to overlook potential risks.

2.1.2 Analytical Methods Used in Sports Business

The industry of sports ticketing has witnessed a significant transformation in recent years, with the advent of advanced analytical methods and technologies. These innovative approaches have empowered organizations to optimize their ticketing strategies, enhance revenue generation, and improve the overall fan experience. In this section, we will look into the various analytical methods employed in sports ticketing, including data mining, predictive modeling, and machine learning, and explore their applications in the industry.

Data mining has emerged as a crucial tool in the sports ticketing landscape, enabling organizations to uncover valuable insights from vast amounts of data. By leveraging data mining techniques, teams can identify patterns, trends, and correlations in ticket sales, customer behavior, and market dynamics (Singh, 2020). This knowledge allows them to make data-driven decisions, such as identifying high-demand games, optimizing pricing strategies, and targeting specific customer segments. Sacha et al. (2014) demonstrates the effectiveness of feature-driven visual analytics in soccer data, highlighting the potential of data mining in sports analytics.

Predictive modeling is another powerful analytical method that has gained traction in sports ticketing. By utilizing historical data and machine learning algorithms, predictive models can forecast future ticket demand, revenue, and customer behavior (Brooks et al., 2016). These models consider a wide range of factors, such as team performance, opponent strength, weather conditions, and promotional activities, to generate accurate predictions. Predictive modeling enables organizations to proactively adjust their ticketing strategies, optimize inventory management, and enhance dynamic pricing capabilities.

Machine learning, a subset of artificial intelligence, has revolutionized the way sports organizations approach ticketing analytics. Machine learning algorithms can automatically learn from data, identify complex patterns, and make intelligent predictions without being explicitly programmed (Bhatnagar & Babbar, 2019). In the context of sports ticketing, machine learning can be applied to various tasks, such as demand forecasting, price optimization, and customer segmentation. By continuously learning from new data, machine learning models can adapt to changing market conditions and consumer preferences, enabling organizations to stay ahead of the curve.

The concept of variable ticket pricing (VTP) has gained traction in the sports industry as a means to optimize revenue by adjusting ticket prices based on factors such as opponent quality, day of the week, and special events. Rascher et al. (2007) investigate the application of VTP in Major League Baseball (MLB), showcasing how teams can leverage this strategy to increase ticket revenue by aligning prices with demand fluctuations. Their findings indicate that optimal VTP implementation can yield substantial revenue gains for MLB teams, highlighting the importance of data-driven strategies in sports ticketing.

Specific techniques within these analytical methods have proven to be particularly effective in sports ticketing. Clustering, for instance, allows organizations to group customers based on their purchasing behavior, preferences, and demographics (Bhatnagar & Babbar, 2019). This segmentation enables targeted marketing campaigns, personalized offers, and tailored ticketing packages. Regression analysis, on the other hand, helps in understanding the relationship between ticket sales and various influencing factors, such as team performance, ticket prices, and promotional activities (Brooks et al., 2016). Time series forecasting techniques, such as ARIMA and exponential smoothing, are employed to predict future ticket demand based on historical sales data, enabling organizations to optimize inventory and pricing decisions (Singh, 2020).

The application of analytical methods in sports ticketing has seen significant advancements in recent years, with organizations leveraging these techniques to optimize revenue management and enhance fan engagement. One such area of focus is the bundling of tickets and the scheduling of league games to maximize revenue generation. Duran et al. (2012) explore the interplay between league scheduling and game bundling decisions in a double round robin tournament setting. By utilizing a heuristic method that incorporates approximate expected revenue values based on revenue increase and decrease patterns of bundled tickets, they demonstrate the potential for significant revenue enhancements through strategic scheduling and bundling practices.

The use of evolutionary algorithms has emerged as a promising approach for tackling the complex challenges of sports scheduling. Barone et al. (2006) propose a multi-objective evolutionary algorithm for fixture scheduling in the Australian Football League (AFL), this method takes into account various factors such as competition fairness, revenue expectations, and venue availability. By simultaneously optimizing multiple objectives, their approach generates a range of alternative solutions that trade off different criteria, offering flexibility to decision-makers. The successful application of evolutionary algorithms in this context underscores the potential of these techniques to address the intricate requirements and constraints inherent in sports scheduling problems.

As the sports industry continues to evolve, the integration of advanced analytical methods and innovative ticketing strategies will become increasingly crucial for organizations seeking to stay ahead of the curve. By using data mining, predictive modeling, and machine learning, teams can uncover valuable insights, optimize pricing and bundling decisions, and create better fan experiences that drive long-term success.

2.2 Clustering and Tiering

2.2.1 Overall methods

Clustering is a fundamental technique in data analysis and pattern recognition that plays a pivotal role in uncovering hidden structures and relationships within datasets (Xu & Tian, 2015). By grouping similar objects together based on their inherent characteristics, clustering enables people to gain valuable insights and make data-driven decisions.

The origins of clustering algorithms can be traced back to the seminal work of Hugo Steinhaus in 1956. In his paper "Sur la division des corps matériels en parties," Steinhaus formulated the problem of partitioning a set of points into K clusters in a finite-dimensional space, with the objective of minimizing the sum of squared distances between each point and its assigned cluster centroid (Steinhaus, 1956). This formulation laid the foundation for the development of the K-means algorithm and its variants, which have become indispensable tools in the field of data clustering.

K-means stands out as a simple yet powerful method for partitioning data into K clusters (MacQueen, 1967). The algorithm iteratively assigns each data point to the nearest cluster centroid and updates the centroids based on the mean of the assigned points. Despite its simplicity, K-means has proven to be effective in various domains, including customer segmentation and product categorization (Punj & Stewart, 1983). Another popular clustering algorithm is K-medoids, also known as Partitioning Around Medoids (PAM) (Kaufman & Rousseeuw, 1990). Unlike K-means, which uses the mean of the data points as the cluster centroid, K-medoids selects actual data points as cluster representatives (medoids). This property makes K-medoids more robust to outliers and noise compared to K-means, as the medoids are less sensitive to extreme values (Park & Jun, 2009).

Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester et al., 1996), offer a different perspective on clustering. These algorithms define clusters as dense regions separated by areas of lower density. DBSCAN can discover clusters of arbitrary shape and is particularly effective in handling datasets with noise and outliers. By setting appropriate parameters for density thresholds, DBSCAN can uncover meaningful clusters that may be overlooked by other algorithms (Schubert et al., 2017). The choice of clustering algorithm depends on various factors, including the nature of the data, the desired number of clusters, and the presence of noise or outliers. In the context of tiering, the selected algorithm should be able to capture the underlying patterns and similarities among customers or products, enabling the creation of meaningful and actionable segments.

K-means, K-medoids (PAM), and density-based clustering algorithms (e.g., DBSCAN) are three prominent clustering techniques, each with its own strengths and weaknesses. K-means and K-medoids tend to produce spherical clusters, while density-based algorithms like DBSCAN can identify clusters of arbitrary shapes and are less sensitive to outliers K-means is relatively efficient but sensitive to the initial choice of cluster centers, while K-medoids is more robust to outliers but less scalable. Density-based algorithms require careful parameter tuning and may struggle with high-dimensional data (Saxena, A., Mittal, M., & Goyal, L. M., 2015).

2.2.2 Determining Number of Clusters

At the core of the clustering algorithms is figuring out how many clusters is optimal for the dataset. First proposed by Robert L. Thorndike was a version of what is now referred to as the elbow method. Thorndike, R. L. (1953). In this work, he speculated about the use of plotting the explained variance as a function of the number of clusters and identifying the "elbow" of the curve as the optimal number of clusters to use. This approximation, while computationally efficient, has its limitations.

As Schubert (2023) points out, the elbow plot lacks a clear theoretical foundation and is highly sensitive to the scaling of the axes and the range of k values considered. The author argues that the heuristic approaches used to formalize the notion of an "elbow" are often based on visual interpretation and fail to take into account the underlying process that generates the data. Furthermore, the elbow method has been found to perform poorly on datasets with overlapping clusters, non-convex shapes, or varying cluster sizes (Schubert, 2023). These limitations suggest that relying solely on the elbow method for determining the number of clusters can lead to suboptimal results and that alternative methods should be considered instead.

The gap statistic proposed by Tibshirani et al. (2001) offers a more principled approach to estimating the optimal number of clusters. The gap statistic compares the observed within-cluster dispersion to the expected dispersion under a null reference distribution, typically generated by sampling uniformly from the range of the data. By standardizing the graph of $log(Wk)$ against this reference distribution, the gap statistic aims to identify the value of k for which the observed dispersion falls furthest below the reference curve. This approach effectively addresses some of the shortcomings of the elbow method, such as the lack of a clear theoretical foundation and sensitivity to the scaling of the axes. However, as Schubert (2023) notes, the gap statistic's performance can be sensitive to the choice of reference distribution and may exhibit instability with default sample sizes on challenging data sets.

2.2.3 Tiering

In the context of customer segmentation and product categorization, the concept of tiering emerges as a powerful application of clustering techniques. This method involves the strategic splitting up of customers or products into distinct groups based on their value, preferences, or other relevant attributes (Zeithaml et al., 2001). This approach allows organizations to tailor their offerings, pricing strategies, and marketing efforts to specific segments, thereby enhancing customer satisfaction and optimizing revenue potential. The process of tiering relies heavily on the effective implementation of clustering algorithms, which can identify meaningful patterns and similarities within the data.

Currently, the Cleveland Cavaliers tier their games using a weighted k-means cluster algorithm to form hierarchical tiers based on a cumulative value. For instance, the Boston Celtics on a Saturday, would be weighted higher than a Houston Rockets game on a Wednesday. This is due to the Celtics having a better record and being on a weekend, which historically has been understood to be a better revenue generating day of the week. This approach has proven to be an effective means for the organization to understand the relative value of games compared to one another. Using a method like this has led to better interpretability and understanding of the method for the business's stakeholders. However, this method does have its drawbacks. The weights given to each metric are arbitrary on how individuals feel the variable should rank in terms of importance on the model. While weights are a great strategy to further reduce the decision down to a single point, more work should be done to determine the true importance of variables outside of this study. Overall, it is a great step in the right direction by integrating machine learning techniques into the current business process (Quinn Spangler, personal communication, June 13, 2024).

The application of machine learning in ticket tiering is just one example of its potential in the sports industry. In various other sectors, such as retail and finance, machine learning has been successfully employed for customer segmentation (Neptune.ai, 2023), fraud detection (Ravelin Technology, n.d.), and predictive maintenance (SCW.AI, 2023).

2.3 Heuristics & Optimizations

2.3.1 Purpose

Metaheuristics have emerged as a powerful class of algorithms that can effectively navigate vast search spaces and find near-optimal solutions in a reasonable amount of time (Yang 2020). These techniques have gained significant attention due to their ability to adapt to various problem domains and their potential to tackle real-world challenges that traditional optimization methods struggle with.

At their core, metaheuristics are high-level problem-independent algorithmic frameworks that guide the search process through a complex solution space (Sörensen et al. 2017). They combine heuristics, which are problem-specific strategies, with overarching frameworks to efficiently explore and exploit promising regions of the search space. Heuristics play a crucial role in metaheuristics by providing domain knowledge and guiding the search towards feasible solutions (Gendreau and Potvin 2005). These heuristics are often inspired by natural phenomena, such as evolution, swarm intelligence, and physical processes, leading to the development of a diverse range of metaheuristic algorithms.

The primary goal of optimization techniques is to find the best solution among a set of alternatives while satisfying given constraints (Talbi 2009). Traditional optimization methods, such as linear programming and gradient-based techniques, have been successfully applied to well-defined problems with specific properties. However, many real-world optimization problems are characterized by non-linearity, multi-modality, and high dimensionality, rendering these classical methods ineffective or computationally intractable (Mahdavi et al. 2015). Metaheuristics have emerged as a powerful alternative, capable of handling the complexities and uncertainties inherent in these challenging problems.

One of the key advantages of metaheuristics is their flexibility and adaptability. Unlike problem-specific algorithms, metaheuristics provide a general-purpose optimization framework that can be easily tailored to suit different problem domains (Molina et al. 2020). This versatility has led to the successful application of metaheuristics across a wide range of fields, including engineering, finance, healthcare, and logistics. Moreover, metaheuristics can effectively handle problems with discrete, continuous, or mixed-integer variables, as well as those with multiple objectives and constraints (Tzanetos and Dounias 2021). The development of metaheuristics has been driven by the need to solve increasingly complex optimization problems in a computationally efficient manner. As the size and complexity of these problems grow, exact methods become impractical due to their exponential time complexity (Yang 2020). Metaheuristics provide a pragmatic approach by striking a balance between exploration and exploitation of the search space. Exploration refers to the ability to broadly search the solution space and identify promising regions, while exploitation focuses on refining and intensifying the search within these regions (Črepinšek et al. 2013). By carefully balancing these two aspects, metaheuristics can efficiently navigate the search space and converge towards high-quality solutions.

2.3.2 Broader Methods

Over the past few decades, numerous metaheuristic algorithms have been proposed, each with its unique characteristics and inspired by different natural phenomena. Some of the most well-known metaheuristics include Genetic Algorithms (Holland 1991), Particle Swarm Optimization (Eberhart and Kennedy 1995), Ant Colony Optimization (Dorigo et al. 2006), and Simulated Annealing (Kirkpatrick et al. 1983). These algorithms have been extensively studied and applied to a wide range of optimization problems, demonstrating their effectiveness and robustness.

More recently, there has been a surge in the development of new metaheuristic algorithms, with over 500 new algorithms proposed to date (Rajwar et al. 2023). This proliferation of metaheuristics has led to concerns about the novelty and originality of some of these algorithms, as many of them share substantial similarities with existing techniques (Tzanetos and Dounias 2021). Despite these concerns, the continued interest in metaheuristics highlights their potential to address the ever-growing complexity of optimization problems in various domains.

The field of methods has emerged as a powerful class of optimization techniques capable of tackling complex, real-world problems. By combining problem-specific heuristics with overarching frameworks, metaheuristics provide a flexible and adaptable approach to optimization. As the field continues to evolve, the development and application of metaheuristic algorithms will remain a critical area of research, driving innovation and enabling the solution of increasingly challenging optimization problems with efficient computational methods.

2.4 Genetic Algorithms

2.4.1 Overview

Genetic algorithms (GAs) are a nature-inspired optimization technique that draws upon the principles of evolution to solve complex problems. John Holland first showcased this in the 1970s. GAs have since been widely applied across various domains, including engineering, economics, and artificial intelligence (Holland, 1992). The power of genetic algorithms lies in their ability to efficiently explore solution spaces and find near-optimal solutions without requiring explicit knowledge of the problem structure. An overview of the process of a genetic algorithm can be seen in Figure 1 below.

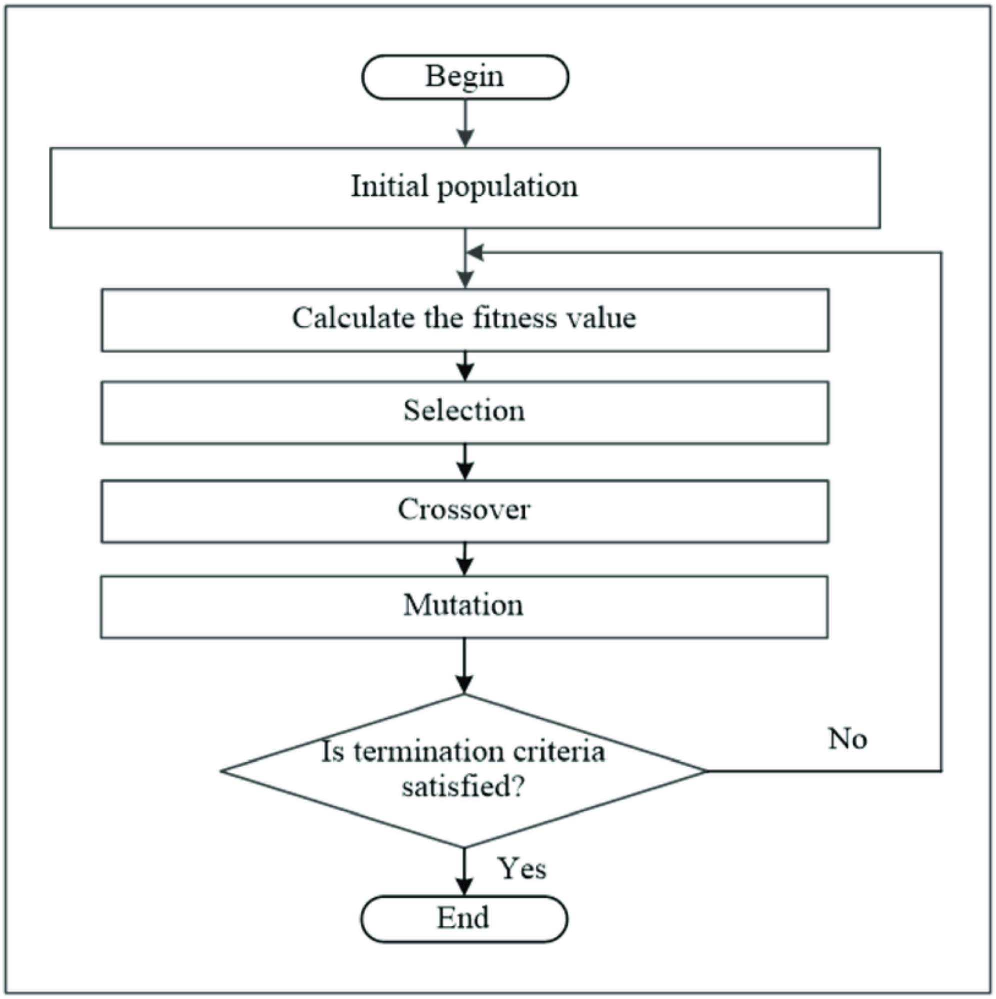
Begin
Initial population
Calculate the fitness value
Selection
Crossover
Mutation
Is termination criteria satisfied?
No
Yes
End

Figure 1: Process Diagram for a Genetic Algorithm (Albadr et al. 2020)

At the core of genetic algorithms are the key components that mimic the processes of natural evolution. These steps include population initialization, fitness evaluation, selection, crossover, and mutation (Mitchell, 1998). The GA begins by initializing a population of candidate solutions. Most of the time these are represented as binary strings or other encodings suitable for the problem at hand. Every individual in the population is then evaluated based on a fitness function, which shows its quality in context of the problem.

The selection process in genetic algorithms favors individuals with higher fitness values, allowing them to pass their genetics to the next generation. Common selection methods include roulette wheel selection, tournament selection, and rank-based selection (Goldberg & Deb, 1991). By giving preference to fitter individuals, the algorithm gradually improves the overall quality of the population over successive generations, while preserving the fittest individuals over successive populations.

Crossover and mutation are the primary genetic operators responsible for creating new offspring and introducing diversity into the population. Crossover involves exchanging genetic material between two parent individuals, typically by swapping segments of their encodings at one or more randomly chosen points. This process allows the algorithm to combine promising features from different solutions and explore new regions of the search space. On the other hand, mutation introduces random modifications to the genetic material of individuals, helping to maintain diversity and prevent premature convergence to suboptimal solutions (Eiben & Smith, 2015).

One of the key advantages of genetic algorithms is their ability to tackle complex optimization problems that are difficult to solve using traditional methods. GAs are particularly well-suited for problems with large, high-dimensional search spaces, where exhaustive enumeration of all possible solutions is infeasible. Using the principles of evolution genetic algorithms can efficiently search spaces and find almost optimal solutions in a reasonable amount of time (Goldberg, 1989).

Overall, genetic algorithms are versatile and can be adapted to a wide range of problems. They have been successfully applied to diverse domains, including function optimization, machine learning, scheduling, and design optimization (Coello et al., 2007).

The flexibility of GAs allows them to handle both continuous and discrete variables, as well as linear and nonlinear objective functions, making them a powerful optimization tool. This research will dive a bit deeper into two applications: scheduling and inventory optimization.

2.4.2 Scheduling Problems

Scheduling problems are a class of optimization problems that involve allocating resources to tasks over time with the goal of optimizing certain objectives. These problems are commonly encountered in various domains, including manufacturing, transportation, and project management.

One application of genetic algorithms in scheduling is the job shop scheduling problem. In this problem, a set of jobs needs to be processed on a set of machines. Each job consists of a sequence of operations that must be performed in order. The objective is to minimize the total time required to complete all jobs. Ding et al. proposed a hybrid genetic algorithm for solving this problem. Their approach combined a genetic algorithm with a local search technique to improve the quality of solutions. The experimental results demonstrated the effectiveness of the hybrid algorithm in finding near-optimal solutions in regards to their benchmark instances (Ding et al. 2023).

Another important scheduling problem is the resource-constrained project scheduling problem. This involves scheduling a set of activities subject to precedence constraints and resource availability constraints. The objective is to minimize the project duration while respecting the constraints of the problem. Wang and Song developed a GA for solving the problem. They introduced a new encoding scheme and designed specialized operators to handle the constraints effectively. An algorithm was tested on their set of benchmark instances and showed promising results in terms of solution quality and efficiency (Wang and Song 2023).

A third application has been in the vehicle routing problem, which is a problem in transportation and logistics. This involves designing optimal routes for a fleet of vehicles to serve a set of customers while minimizing the total travel distance or cost. Xiong and Xu proposed a fish swarm algorithm, which is a variant of genetic algorithms, for solving this problem. Their approach utilized the collective intelligence of fish swarms to explore the search space and find high-quality solutions. The experimental results demonstrated the effectiveness of the fish swarm algorithm in quickly converging to the shortest path and outperforming traditional methods (Xiong & Xu 2021).

Overall, GAs have been a very useful technique for solving a wide range of scheduling problems. Their ability to handle complex constraints, optimize multiple objectives, and adapt to different problem variants has made them a popular in the field. As the complexity of scheduling problems continues to increase, it’s safe to say GAs and their variants are set up to play an important role in developing efficient and effective solutions.

2.4.3 Inventory Optimization

Various inventory management problems have used GAs successfully. In the context of inventory optimization, GAs can be used to determine optimal product assortment, stock levels, and strategies to maximize efficiency and profitability.

One of the key advantages of using GAs for inventory optimization is their ability to handle complex, multi-objective problems. As discussed by Valova et al. (2014), GAs can be implemented with variable-length chromosomes, allowing for dynamic optimization of inventory-related decisions. This flexibility enables GAs to adapt to changing market conditions and consumer preferences, making them well-suited for inventory management in dynamic environments. This becomes prevalent in the NBA packaging problem since most years have 41 home games.

Al-Ashhab and Alghamdi (2017) use GAs in solving university course timetabling problems, which share similarities with inventory optimization. Both problems involve the allocation of limited resources (e.g., time slots, products) to maximize a specific objective (e.g., student satisfaction, profitability). The authors' successful usage of a two-stage GA model highlights the potential for applying GAs to inventory optimization, where the algorithm can be designed to optimize product assortment and stock levels in a multi-stage process.

When applying GAs to inventory optimization, the chromosomes can represent various aspects of the inventory management strategy, such as product selection, order quantities, and reorder points (Valova et al., 2014). The fitness function can be designed to evaluate the performance of each solution based on metrics such as revenue, customer satisfaction, and inventory turnover. By iteratively evolving the population of solutions through selection, crossover, and mutation operators, GAs can explore a wide range of inventory strategies and converge towards optimal solutions.

Also, GAs can be combined with other techniques, such as data analytics, to further enhance their effectiveness in inventory optimization. For example, historical sales data and demand forecasts can be used to initialize the GA population with promising solutions (Al-Ashhab & Alghamdi, 2017). This hybrid approach can lead to more robust and adaptive inventory management strategies that can cope with the uncertainties and challenges of real-world supply chains.

GAs prove useful got optimizing inventory management. By leveraging their ability to handle complex, multi-objective problems, GAs can help businesses determine the best product assortment, stock levels, and strategies to maximize efficiency and profitability. The integration of GAs with other techniques, such as data analytics, holds great promise for developing more sophisticated and effective inventory management solutions.

Chapter 3: Methodology

This chapter presents the methodology. I describe data preparation in section 3.1, the tiering process in section 3.2, and the method to optimize the ticket packages in Section 3.3. An outline of the entire process can be seen in Figure 2 on the following page.

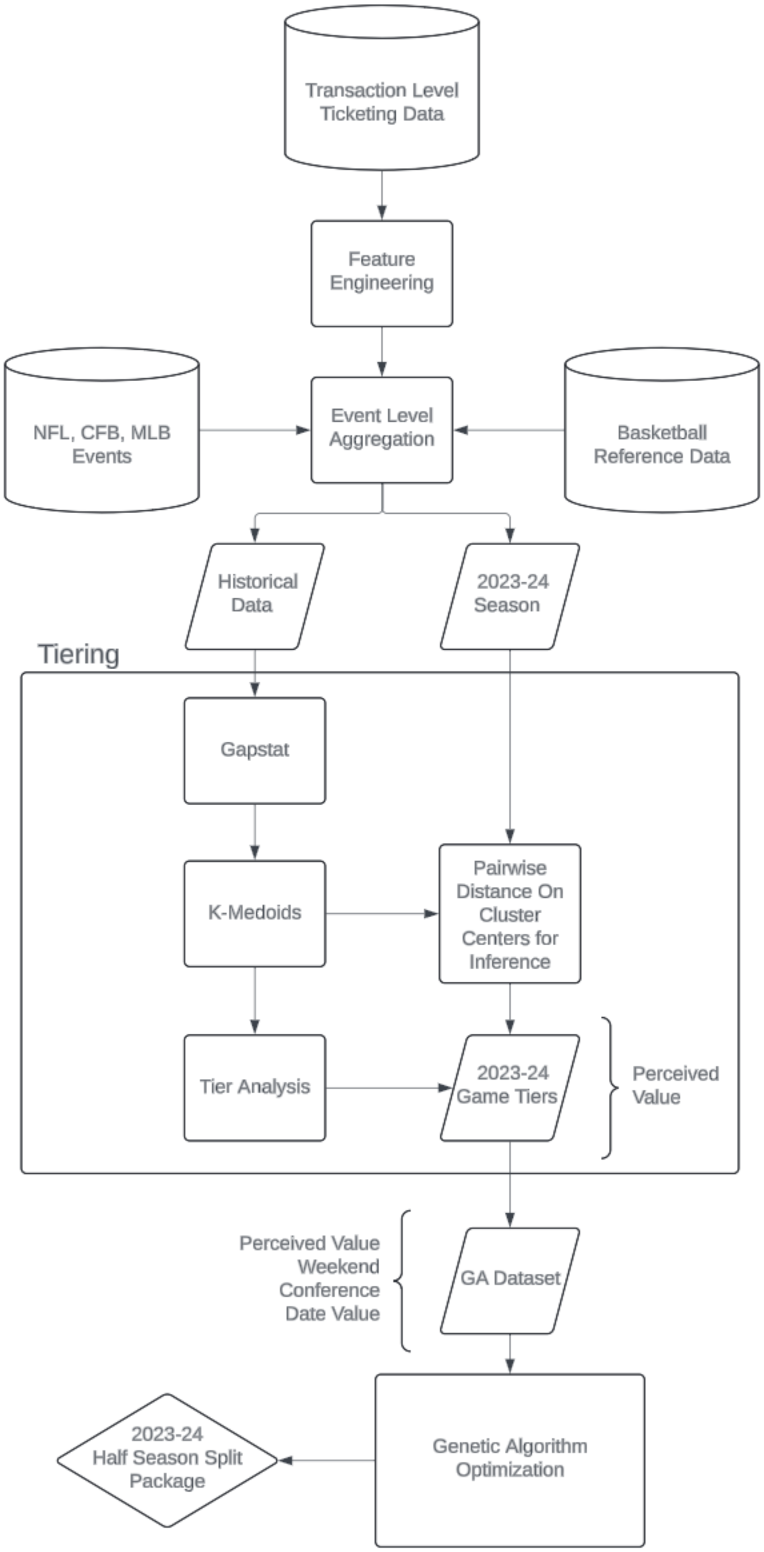

Figure 2: Methodology Process Diagram

3.1 Data Preparation

3.1.1 Data Sources

The data for this study is drawn from two primary sources: a ticketing database and basketball reference. These two sources combined help paint the picture of how events are vied from a high level. By using these in tandem, fan purchase behavior with opposing team performance and marquee factor can be understood together.

The ticketing database describes the fan ticket purchase behavior via a breakdown of inventory purchased on the seat level. The database is a well modeled ticketing manifest that outlines each seat for an event, sold or available. To query this information from the database, there are filters and functions needed.

Due to the Cavalier's recent success and the COVID-19 pandemic impacting in person events, the ticketing information will be filtered from the 2021-22 season onward. This is to account for the shift in behavior from the two potential sources of bias in the dataset. Next, high-level filters are used to make sure that the dataset only includes admissions tickets for Cavaliers games. The exclusion of any tickets that are contracted out to partners is necessary, as those are bought in a separate process by our sponsor companies as part of a partnership deal. The last filter needed on this database is to remove all playoff and preseason games hosted, since the season ticket packages only include regular season matchups.

When pulling down the information from this data source, there are functions built into SQL that will help smooth out data preprocessing. First, the event data timestamp is split into three columns: time of the game, day of the week, and month. The sold status of the seat is reduced to a binary where 1 is the sold and 0 is available. If a fan chooses to resell their ticket, this is marked in binary as 1 and 0 signifies the ticket was not resold. When a fan attends an event, or scanned in, is reduced to a binary where 1 is scanned and 0 is unscanned. Finally, there is a concept in ticketing called a price type groups (PTGs). This is a logical method of grouping different types of ticketing inventories together. For this study, a simplified breakdown was used of five categories. Season is the grouping of tickets that are bought in quarter half, or full season packages. Individual is when the fan buys one or more tickets for a single game. Complimentary, or comp, are tickets that have been given out free of charge. Group tickets are defined as groups of 10 or more tickets purchased together. Partial plans are special packages available for a small selection of games, such as games around Christmas. The resulting query yields a dataset with no null or missing values, as seen in the sample in Table 1 below.

date	day_of_week	game_time	paid_amount	ptg	scanned	resold	seat_sold
3/31/23	Friday	19:30:00	76	group	0	0	1
3/15/23	Wednesday	19:30:00	61	member	1	1	1
3/28/22	Monday	19:00:00	33	member	1	1	1
3/28/22	Monday	19:00:00	33	member	1	1	1
3/15/23	Wednesday	19:30:00	61	member	1	1	1

Table 1: Ticketing Database Sample Data

Basketball reference is a repository for statistics. This study will use the team level statistics to account for the fan's perception of the opposing team based on performance from the previous year. The two metrics used in this study are the rank in conference and number of all stars. Rank in the respective conference is used as an aggregation of team performance relative to their competition. Interpretability of this variable is the advantage of including it. Not only are the top teams from the entire season are ranked high, it also helps determine if the team were in the playoffs. The duality of this variable accounts for both the fans that are locked into the regular season and the recency bias of the playoffs occurring in the early summer. To account for the marquee matchups, the number of all-stars from the previous season present on the team is derived from this source. While some teams may perform poorly in the previous year, they may have the star power that drives fans to watch the superstars of the league. Additionally, this accounts for any offseason moves our opponents make to improve their roster. These two factors will be used to access the on-court product. A sample of this data can be seen in Table 2 below.

season	Opponent	conference	win_perc	rank_in_conference	num_all_stars
2021-22	Atlanta Hawks	East	0.569444	4	0
2021-22	Boston Celtics	East	0.506849	7	2
2021-22	Brooklyn Nets	East	0.666667	2	2
2021-22	Chicago Bulls	East	0.430556	11	1
2021-22	Charlotte Hornets	East	0.452055	10	0

Table 2: Basketball Reference Sample Data

An additional data point is determined from baseball reference. This data source, along with other derived fields outlined in section 3.1.3, determines the competing events in the sports marketplace. This source will inform what the Guardians home schedule is. Since these events compete for fans, it will be important to include it as an external factor in the tiering model.

By using these three sources, an understanding of how fans perceive the value of events using the method of tiering will be determined. From there, the perceived value tier combined with certain event data derived from the ticketing database will be used for the genetic algorithm. In the next section, an outline of the aggregations and additional preprocessing will be addressed.

3.1.2 Feature Engineering

The creation of new features is crucial in preparing the ticketing dataset for tiering the games. This process involves aggregating the data at the event level, which provides a comprehensive view of each game's performance. By using Python and the Pandas library, we can efficiently transform the raw data into a structured format that enables us to derive meaningful insights.

The process begins by creating a new data frame which contains a row for each Cavaliers event with event-related information such as the event date, day of the week, month, game time, and event name. This data frame serves as the foundation for the subsequent aggregations.

Several key metrics are calculated and added to the data frame. These include the total revenue, total tickets sold, total attendance, median paid amount, and minimum paid amount for each event. Additionally, the number of tickets sold for different PTGs are determined and included in the aggregated data frame. These metrics show the performance of a game from a direct financial perspective. Finally, the number of tickets resold for each event is calculated and added to the data frame. This provides insights into the secondary market activity for each event.

The aggregation process reduces the granularity of the ticketing data from the individual ticket level to the event level. By doing so, it enables a more manageable and meaningful analysis of ticket sales and attendance patterns across different events. The resulting data frame contains a wide range of event-level metrics that can be used for the game tiering decision-making process. The resulting dataset consists of 1 row for each of the 122 home games of the past 3 seasons.

Then, the information on the teams from the previous year is pulled down from Basketball Reference via their API. By joining on the season and team, the number of all-stars, the team's conference, and respective rank in the conference is added to the overarching data frame. Now, the model will be able to account for the level of competition faced in each game.

In a similar fashion, the Guardian's schedule is pulled from the Baseball Reference API and joined onto the data frame based on the date. Then, to take into account for college and National Football League (NFL) games, a binary variable is added for each where true is the existence of a game, and false is the absence of a game. For college football, the defined schedule is each Saturday from September through December. In the NFL, the schedule is defined as each Sunday from September through January. Each of these variables is added onto the data frame by joining on the date. By adding this variable, the model is now informed about the competitive sports market in Cleveland.

An additional feature is added to account for Lebron James of the Los Angelos Lakers. The gravity of his presence when he returns to Cleveland is immense since he won a championship with the team in 2016. There remain many fans in the Cleveland area who still revere the NBA superstar as the best Cavaliers player of all time. To account for this, a binary variable is set to true for the three games played in Cleveland against the Lakers.

The feature engineering step is crucial in transforming the raw ticketing data into a structured and aggregated format that facilitates the tiering methods outlined in section 3.2. The aggregated event-level metrics provide a comprehensive view of the ticketing landscape, allowing for a deeper understanding of games from the perspectives of ticket sales, opponent quality, and competing events in the sports market.

3.1.3 Data Exploration and Cleaning

When the original data was pulled from the sources, there existed no null values. However, post-aggregation, there exists a need to clean up the information within the new data model since the left joins produced null values in the dataset. First, the missing values in the data will be explored.

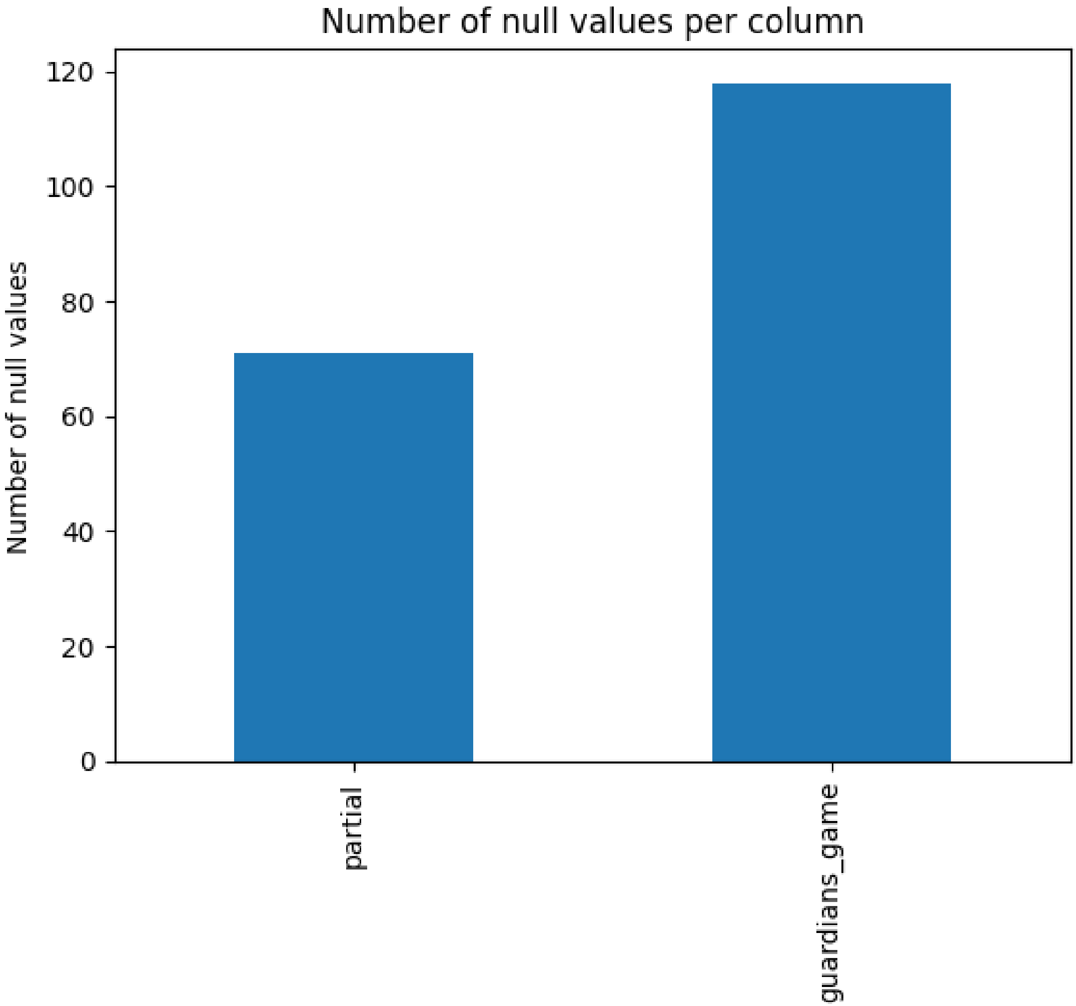

Figure 1: Number of Null Values in Tiering Dataset

According to figure 1, there exists two columns with null values. The missing values can be categorized as not missing at random. For the partial column, this means the game was not a part of a partial package, and therefore there were no tickets sold in this category. Since this affects a large subset of the games, this column should be removed to focus on primary ticketing channels to help the interpretability of the model. In the Guardians game column, the null values represent the games where there were no Guardian's games. Thus, the null values can be imputed to 0 to indicate the occurrence of a Guardian's game being false.

Currently, the dataset holds a variable for the month of a game. To decrease the bias of the date on the resulting model, this variable will be factor reduced. In place of one-hot encoding the variable, creating a variable for each month of the season, a logical breakup of the NBA calendar is the All-Star break. This milestone of the season is considered the halfway point of the season. This will help create less of a weight on the time of the year on the model's decision making.

Another time-based variable exists in the form of game time. This variable will be converted to a binary variable called late night game. The instance of true encapsulates the games that start at a time greater than or equal to 7pm. By reducing this variable, it can now influence the model based on the time of game without having a disproportionate explanation of the model.

Day of the week is another time-based variable that will be factor reduced. This variable is converted to explain if the game exists on a weekend. The weekend games for the NBA are Friday, Saturday, and Sunday. The value of true is assigned to the weekend games and false is assigned to the weekday games. Again, this factor reduction is used to prevent a disproportionate impact on the model.

The resulting data set for tiering is 82 rows by 18 columns. Before the scaling, the 40 games for the 2023-24 season were split off, where they will be run against the model for inference. For the remaining 40 games, the columns of total revenue, show rate, comp, individual, member, group, and secondary market sales will be removed to simulate the schedule being released without previous knowledge of game performance. This split off data set will later be used to run inference on the model using a pairwise comparison.

The scale of the variables of total revenue, show rate, median paid amount, opponent win percentage, comp, individual, member, group, and secondary market sales must be addressed for the 82 previous games. Each variable is put on a scale relative to the mean of the distribution of the column, with each of the values representing the distance from the mean in standard deviations. The use of the standard scaler is for these variables to in the model's ability to interpret the variables. Additionally, the inclusion of these data points in the training of the model will aide with the interpretability of the model's results.

Now that the tiering dataset is prepared (refer to Appendix A for a sample of the data), an analysis of the correlations between variables will be conducted. A correlation matrix is crucial for understanding relationships between variables and identifying potential multicollinearity issues that can impact model performance and interpretability. The correlation heatmap in Figure 2 provides a visual representation of these relationships, informing decisions regarding variable selection. By looking at the visualization, we can detect strong linear relationships between variables, positive or negative correlations, and potential redundancies. Interpreting the correlation heatmap while considering domain knowledge is essential for model development.

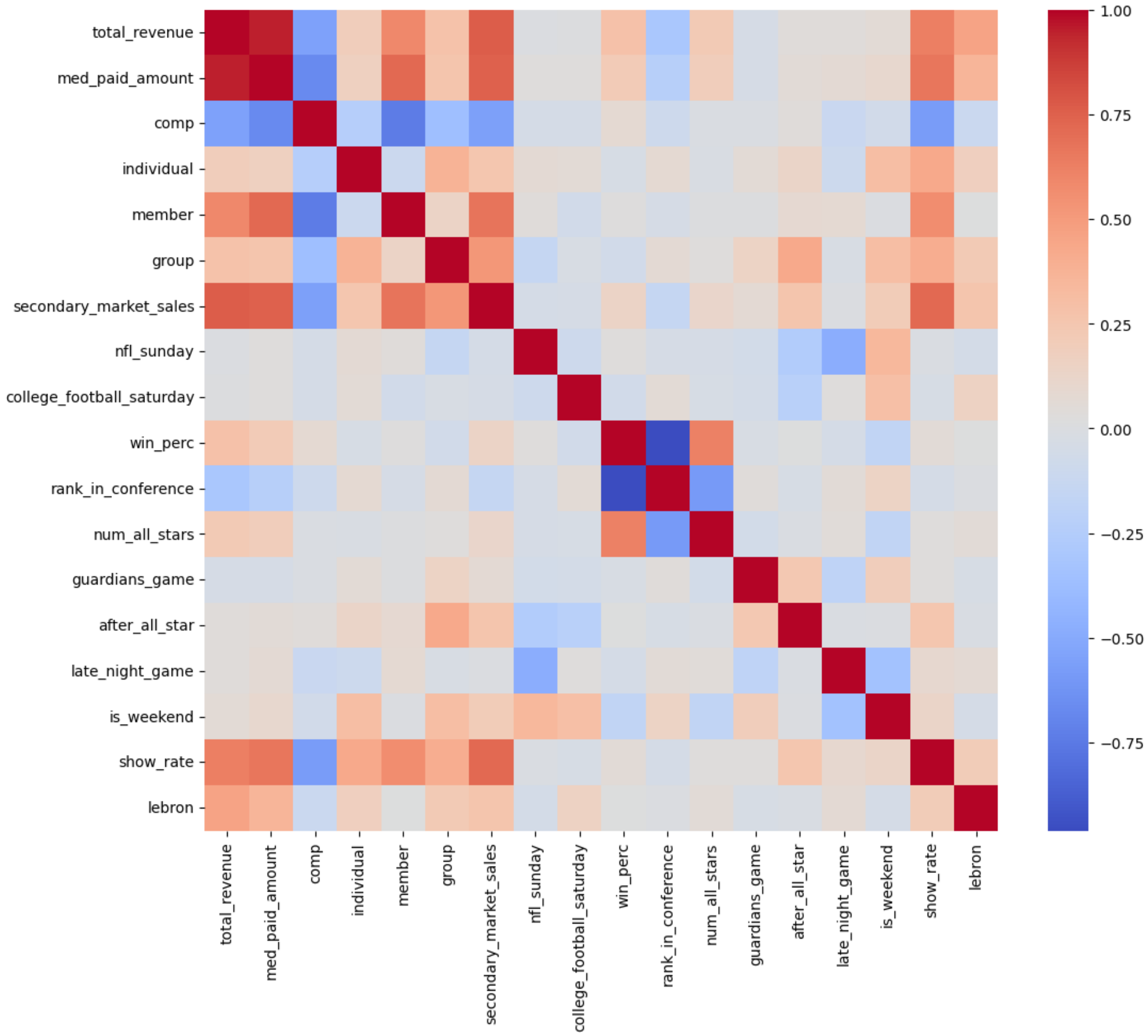

Figure 2: Correlation Heatmap

By examining the correlation heatmap in Figure 2, several notable patterns and relationships emerge. The intensity of the color scale immediately draws attention to a cluster of highly correlated variables, including total revenue and median paid amount This strong positive correlation suggests potential multicollinearity, which is understandable since the higher the median ticket price is for a game, more revenue would be produced. By looking at the intensity of correlations with other variables, notice that median paid amount has more intense correlations with other variables in the data set over total revenue. The removal of median paid amount from the dataset is then executed to avoid giving too much weight to the factor of revenue on the model. Total revenue encapsulates both tickets sold and paid amount, giving it strong interpretability for applying domain knowledge to the results.

Additionally, the intense blue, or negative, coloring draws eyes to win percentage and rank in conference. Since a higher rank in conference is determined by a team's win percentage, this heavy correlation makes sense. The rank in the dataset is on a scale with 1 being the highest and 15 being the lowest, therefore a team with a high winning percentage would have a low rank, resulting in this negative correlation. The removal of win percentage is then executed to avoid multicollinearity that could weigh the model too heavily on team performance since conference rank's playoff logic is favorable for interpretability of the model's results.

Interestingly, the variable comp tickets exhibit a moderate negative correlation with total revenue, secondary market sales, individual, member, and group tickets. This indicates an inverse relationship. In the context of these tickets being freely given away, this observation aligns with the expected behavior, as there are more tickets given for free when there is available inventory in the other categories which then negatively impacts the total revenue. These games are generally less desirable, making the inclusion of this variable, and its relationships, important to preserve.

However, it is crucial to interpret these correlations cautiously, as they may not necessarily imply causation. For instance, the weak correlation exists between show rate and many of the other variables. This could be attributed to fans being more likely to show up to an event they spend their discretionary income on. In a prior case study on the

Cleveland Cavalier’s show rate (Levicki 2024), it was noticed that generally fans who buy tickets show up to a given game.

Overall, the correlation heatmap provides valuable insights into the relationships within the dataset, guiding our feature selection and modeling strategies. By integrating domain knowledge and contextual understanding, these insights can be effectively leveraged to tier games for the genetic algorithm’s evaluation metric. The resulting dataset used for training can be referred to in Appendix B for data definitions of the dataset.

3.2 Game Tiering

3.2.1 Overall Tiering Methodology

Game tiering is a crucial step in optimizing ticket packages by categorizing games based on their demand and revenue potential. This creates a groups game of similar appeal to the fans. This section will outline a proposed method for determining the perceived value of a game to feed into the genetic algorithm.

The chosen algorithm for the tiering method in this study is K-medoids. This method is chosen since the sensitivity to outliers and noise in the dataset. This allows the model to not be as sensitive to overperforming games. The implications of misclassifying a game would negatively affect the packaging of games since there could be an imbalance in fan demand where they would favor one package over another. Then, to determine the optimal number of clusters, the gap stat method will be employed. Using this methodology, on a non-complex data set of a small size fits the algorithm well. In doing so, this will help give an exact number of tiers for the analysis. Finally, we will use UMAP to reduce the dimensions of the clusters for visual analysis for separation of the clusters.

3.2.2 Optimizing the Number of Tiers: Gap Stat

The estimation of the optimal number of clusters in a dataset is a crucial step in cluster analysis. One widely used approach to determine the appropriate number of clusters is the "gap statistic" method, proposed by Tibshirani, Walther, and Hastie (2001). This method provides a statistical procedure to formalize the heuristic of identifying the "elbow" in the plot of the within-cluster dispersion against the number of clusters.

The gap statistic compares the observed within-cluster dispersion to its expected value under an appropriate null reference distribution. The optimal number of clusters is then estimated as the value of k for which the observed within-cluster dispersion falls the farthest below the reference curve.

3.2.2.1 Mathematical Formulation

Let $\{x_{ij}\}, i = 1,2,\ldots,n, j = 1,2,\ldots,p$, be the data consisting of p features measured on n independent observations. Let dii' denote the distance between observations i and i', typically the squared Euclidean distance $\sum_j (x_{ij} - x_i j)^2$

Suppose we have clustered the data into k clusters C1, C2, ..., Ck, with Cr denoting the indices of observations in cluster r, and nr = |Cr|. Let Dr be the sum of the pairwise distances for all points in cluster r, and define the within-cluster dispersion as:

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} * Dr$$

The gap statistic is then defined as:

$$\mathrm{Gap}_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

Where E_n^* denotes the expectation under a sample of size n from the reference distribution.

3.2.2.2 Computational Implementation

The choice of the reference distribution is crucial for the gap statistic method. Tibshirani et al. (2001) proposed two options. The first entails generating each reference feature uniformly over the range of the observed values for that feature. And the second is to generate the reference features from a uniform distribution over a box aligned with the principal components of the data.

The implementation of the gap statistic involves the following steps:

1. Cluster the observed data, varying the total number of clusters from $k = 1,2,\ldots,K$, giving within-dispersion measures $W_k, k = 1,2,\ldots,K$.
2. Generate B reference data sets, using the chosen reference distribution (uniform or principal component-based), and cluster each one, giving within-dispersion measures $W^*_{kb}, b = 1,2,\ldots,B, k = 1,2,\ldots,K$.
3. Compute the estimated gap statistic:

$$Gap(k) \ = \ (1/B)\sum_b log(W^*_{kb}) \ - \ log(W_k).$$

4. Let $\bar{l} = \left(\frac{1}{B}\right)\sum_b \log(W^*_{kb})$, Compute the standard deviation

$$sd_k = [(1/B)\sum_b log(W^*_{kb}) \ - \ log(W_k) - \bar{l}\,\}^2]^{1/2}$$

5. Define $s_k \ = \ sd_k\sqrt{}((1 + 1/B))$.
6. Choose the number of clusters k̂ as the smallest k such that

$$Gap(k) \ \geq \ Gap(k + 1) \ - \ s_{\{k+1\}}$$

Using this as the baseline methodology

3.2.2.3 Code Implementation

The gap stat method will be put into place using an imported Python created on Github (Maloney, 2019). Not only will the computations of the gap statistic be enabled with this, but visualizations will also be provided. The results of using this heuristic method will be presented in the results section.

3.2.3 Tiering Algorithm: K-Medoids

The k-medoids method is a robust and effective clustering technique that partitions a dataset into k clusters. Unlike traditional clustering methods like k-means, which use centroids (mean vectors) to represent clusters, the k-medoids method employs actual data points, called medoids, as cluster representatives (Kaufman & Rousseeuw, 1990). This approach offers several advantages, particularly in handling outliers and noise within the dataset. With some observed noise in the correlation heatmap, this method is a great fit for the given situation.

The objective of the k-medoids method is to minimize the sum of dissimilarities between each object and its closest medoid. Mathematically, the objective function can be expressed as:

$$min \sum_{i=1}^{n} \sum_{j=1}^{k} d(x_i, m_j) u_{ij}$$

where:

- n is the number of objects
- k is the number of clusters
- $d(x_i, m_j)$ is the dissimilarity between object x_i and medoid m_j

- u_{ij} is a binary variable indicating whether object x_i belongs to the cluster represented by medoid m_j

This algorithm follows an iterative approach to find the optimal set of medoids and cluster assignments. Following the general steps below:

1. Randomly select k objects as the initial medoids.
2. Assign each object to the cluster represented by its closest medoid.
3. For each cluster, find the k objects that have the smallest sum of dissimilarities to all other objects in the cluster. Replace the current medoid with this new object.
4. Repeat steps 2 and 3 until convergence, where there would be no change in medoids or cluster assignments.

3.2.4 Tier Evaluation and Validation

This method will be implemented within Python using the scikit learn package. In doing so, we will assign games from the previous two years into logical clusters, which will be analyzed by their medoids as well as visually. The visualizations that will be used in the results section are produced by applying a UMAP dimension reduction technique to allow the clusters to be viewed in 3 dimensions to apply a visual inspection of the separation of clusters. Statistical measures such as silhouette score were used to assess the clustering performance as well. Then, assigning tiers will be conducted by assigning value to each cluster based on the medoids according to domain knowledge and consulting with subject matter experts with in-depth understanding of the business's requirements.

After the determination of tiers, each tier's characteristics and implications for ticket packages will be interpreted and discussed. Examples of games assigned to each tier will be provided, highlighting their distinguishing features. Then, a comparison with the game tiers used for 2023-24 season will ensue. By leveraging the opinions of several subject matter experts, a comparison of the advantages and disadvantages of the two results will be discussed.

Then, the 2023-24 season's schedule, which only includes information that would be included upon the release of the schedule, will be compared to the cluster centers using the pairwise distance of each data point against each of the clusters. This process simulates the real-world decision needed to be made upon the release of the schedule. The data points that will be included in this analysis are NFL Sunday, College Football Sunday, Rank in Conference, Number of All-Stars, Guardians Game, After All-Star, Is Weekend, and Lebron.

3.2.5 Integration with Genetic Algorithm

The game tiers for the 2023-24 season will be fed into a genetic algorithm to optimize the composition of half and quarter season ticket packages. The other data points that are derive from each of the games include a binary weekend variable, day number, and conference. Day number is a numeric variable where the event lies on the calendar's distribution where the first day is zero and the final day is the days from the first game of the season. This data set will be the basis of the genetic algorithm and will be used in evaluating the packages as a metric that considers all four variables. Refer to Appendix C for definitions of the data. It is important to note that these tiers are not necessary for the evaluation function, as it is agnostic to the value given to the clusters. Tiers exist as a business construct to better interpret and understand the relative value for a group of games compared to another.

3.3 Ticket Package Optimization

3.3.1 Evaluation Function

The evaluation function is the heart of the ticket package optimization process. It assesses the quality of ticket package solutions by comparing them against global metrics. The function takes in the ticket package solutions and optional weights for different attributes (perceived value, weekend/weekday, conference, and date range). It then calculates global metrics for each attribute across all ticket packages, providing a benchmark for comparison. By setting this benchmark, it makes way for the comparison of each packages distributions to be compared to the entire schedule. Each individual ticket package is then compared to the global metrics. Then, similarity scores for each attribute are normalized and weighted based on the provided weights, allowing for customization and fine tuning of the evaluation process using domain expertise. Finally, an overall similarity score is calculated as the weighted sum of the attribute similarity scores, providing a single metric to assess the quality of the ticket package solutions. Below is a breakdown of the evaluation algorithm's process. Then, calculation of this metric will be broken down in the following subsections.

1. **Input Collection**: The function receives ticket package solutions and optional weights for the various attributes.
2. **Global Metric Calculation**: Global metrics are computed for each attribute across all ticket packages, establishing benchmarks for comparison. This creates a baseline for which individual splits can be evaluated against.
3. **Attribute Breakdown**: Individual components such as perceived value, weekend/weekday distribution, conference distribution, and date distribution are

each given similarity scores. This provides insight into specific aspects of equivalence between half season splits.

4. **Normalization and Weighting**: Similarity scores for each attribute are normalized and weighted based on provided inputs. This allows for customization of the evaluation process using domain expertise.
5. **Overall Similarity Calculation**: A weighted sum of attribute similarity scores is computed, producing a single metric representing package quality. This consolidates multiple factors into one comprehensive score from 0 to 1. This final step yields a comprehensive assessment of how well each equivalent each half season split is to its counterpart.

3.3.1.1 Perceived Value

Let's define the following variables:

- X: The perceived value distribution in the array being analyzed
- Y: The global perceived value distribution
- n: The number of elements in the array being analyzed
- N: The total number of elements in the global distribution
- O_i: The observed frequency of the i-th perceived value in the array
- E_i: The expected frequency of the i-th perceived value based on the global distribution

The perceived value metric is calculated using the chi-square goodness-of-fit test, which compares the observed distribution (X) with the expected distribution (Y). The formula for the chi-square statistic is:

$$\chi^2 = \Sigma ((O_i - E_i)^2 / E_i)$$

Where the summation is taken over all possible perceived values. The expected frequency E_i is calculated as:

$$E_i = (\mathrm{n} / \mathrm{N}) * Y_i$$

Where Y_i is the count of the i-th perceived value in the global distribution. The perceived value metric is then calculated as:

$$s_1 = p\_value$$

Where the p_value is the probability of obtaining the observed chi-square statistic or a more extreme value, assuming that the null hypothesis, or the observed distribution matches the expected distribution, is true.

3.3.1.2 Weekend/weekday

Let's define the variables:

- X: The weekend/weekday distribution in the array being analyzed (0 for weekday, 1 for weekend)
- Y: The global weekend/weekday distribution
- n: The number of elements in the array being analyzed
- N: The total number of elements in the global distribution
- O_0: The observed frequency of weekdays in the array
- O_1: The observed frequency of weekends in the array
- E_0: The expected frequency of weekdays based on the global distribution
- E_1: The expected frequency of weekends based on the global distribution

Now, The weekend metric is calculated using the chi-square goodness-of-fit test for a binary distribution:

$$\chi^2 = ((O_0 - E_0)^2 / E_0) + ((O_1 - E_1)^2 / E_1)$$

Then, the expected frequencies E_0 and E_1 are calculated as:

$$E_0 = (n / N) * Y_0$$

$$E_1 = (n / N) * Y_1$$

Where Y_0 and Y_1 are the counts of weekdays and weekends, respectively, in the global distribution. The weekend metric is then calculated as:

$$s_2 = p\_value$$

Where the p_value is the probability of obtaining the observed chi-square statistic or a more extreme value, assuming that the null hypothesis, or the observed distribution matches the expected distribution, is true.

3.3.1.3 Conference

The conference distribution is calculated in a similar way to the weekend/weekday distribution, using the chi-square goodness-of-fit test for a binary distribution.

Let's define the variables for conference:

- X: The conference distribution in the array being analyzed (0 for non-conference, 1 for in conference)
- Y: The global conference distribution
- n: The number of elements in the array being analyzed
- N: The total number of elements in the global distribution
- O_0: The observed frequency of non-conference games in the array
- O_1: The observed frequency of in conference games in the array
- E_0: The expected frequency of non-conference games based on global distribution
- E_1: The expected frequency of in conference games based on the global distribution

Using the same methodology used in the above section (3.3.1.2), where the chi-square statistic and expected frequencies are calculated for a binary distribution, the conference metric is calculated as:

$$s_3 = p\_value$$

Where the p_value is the probability of obtaining the observed chi-square statistic or a more extreme value, assuming that the null hypothesis, or the observed distribution matches the expected distribution, is true.

3.3.1.4 Date

Let's define the following variables:

- D: The set of dates in the ticket package being analyzed, $\{d_1, d_2, \ldots, d_n\}$
- n: The number of dates in the set D
- d_i: The i-th date in the set D
- R: The maximum date range in the global distribution
- w_d: The weight value assigned to the distance metric
- w_r: The weight value assigned to the range metric

The date score is calculated as a weighted average of two components: normalized average distance and normalized date range. This is calculated by

$$s_4 = (normalized\ avgerage\ distance\ *\ 0.5) + (normalized\ date\ range\ *\ 0.5)$$

3.3.1.4.1 Normalized Average Distance

This metric is used to understand the distance between games. The inclusion of this metric is to avoid clustering of games together within packages and ensure an even spacing of games within the same package consistent with the overall schedule.

The list of distances between consecutive dates is given by

$$\varDelta d_i = d_{\{i+1\}} - d_i \,, for\ i = 1,2,\ldots,n-1$$

The average distance between dates is calculated as

$$\mu_{\varDelta d} = (1 / (n-1)) * \sum_{i=1}^{n-1} \varDelta d_i$$

The maximum possible average distance is

$$\mu_{max} = R / (n-1)$$

The mean absolute deviation (MAD) of distances is calculated as

$$MAD = (1 / (n-1)) * \sum_{i=1}^{n-1} |\varDelta d\_i - \mu\_\varDelta d|$$

The normalized MAD is then defined as

$$MAD_{norm} = 1 - (MAD / \mu_{max})$$

3.3.1.4.2 Normalized Date Range

This metric is used to understand the overall range of games in each section. The inclusion of this metric is to analyze for an even span of games within the same package compared to the overall schedule.

The minimum and maximum dates in the set D are given by

$$d_{min} = min(D)$$

$$d_{max} = max(D)$$

The date range in the set D is

$$r_D = d_{max} - d_{min}$$

Finally, the normalized date range is defined as

$$r_{norm} = r_D / R$$

3.3.1.5 Overall Similarity Score

The overall similarity score is calculated as a weighted average of the individual metric scores. This is done to take into account each variable in the evaluation of the ticket packages as a holistic look at how close the packages are to the overall schedule distributions.

$$overall\_similarity = \Sigma \left(s_i * w_i \right)$$

Where s_i is the similarity score for the i-th metric (perceived value, weekend, conference, date score) and w_i is the weight assigned to the i-th metric.

The weights are normalized to sum up to 1

$$w_t = \Sigma\, w_i$$

$$w_i = weight\ / w_t$$

Where w_t summation is taken over all metrics and $weight$ is the given weight assigned to a metric. Using the metrics shown in the previous sections, here is the formula:

$$overall\_similarity = (s_1 * w_1) + (s_2 * w_2) + (s_3 * w_3) + (s_4 * w_4)$$

The evaluation metric presented through these formal equations plays a crucial role in the context of a genetic algorithm, serving as the fitness function that quantifies the quality of each proposed ticket package combination. By incorporating various components such as perceived value distribution, weekend/weekday distribution, conference distribution, and date distribution, and utilizing statistical measures like the chi-square goodness-of-fit test and normalization techniques, this evaluation metric provides a comprehensive and robust assessment of ticket packages. The similarity score guides the genetic algorithm in selecting the fittest solutions for evolution while being an interpretable metric on the scale of 0 to 1.

3.3.2 Initializing Population

In the context of ticket package optimization, chromosomes represent potential solutions. The creation of chromosomes in this problem instance is to create the different combinations of tickets for each member of the population. Each chromosome in the population is split into the desired number of packages, representing each split of the season schedule. Thus, each member of the population is a pre-selected breeding pair since open cross breeding of chromosomes could lead to duplicate games being included in chromosomes. Upon initialization of the population, each member of the population contains a pre-selected number of chromosomes for inner breeding.

The function takes in the dataset described in section 3.2.4, the desired number of chromosome pairs (n), and the number of splits (s) for each pair. It starts by determining the equal-sized splits for each chromosome pair, ensuring a balanced distribution of games across the packages. Then, games are randomly selected for each split without replacement, creating diverse chromosome pairs. Then, the evaluation function is then used to evaluate each chromosome pair, calculating overall fitness of the solutions. Finally, the chromosome pairs are sorted based on their overall similarity scores in descending order, prioritizing the best solutions. By organizing the solutions, the preservation of the fittest chromosome can be easily accessible by the algorithm.

3.3.3 Tournament Selection

For genetic algorithms, tournament selection is a technique for selecting individuals from the population to participate in the breeding process. This method mimics the natural selection process, where the fittest individuals are more likely to forward their genetic material to the next generation. The function enters a loop that continues until the desired number of parents is obtained. Within each iteration, the steps for this function are as follows:

1. **Random Selection**: A random subset of k individuals is chosen from the population. This subset represents the participants in the tournament.
2. **Fitness Evaluation**: The fitness score is used to evaluate the quality of each ticket package combination in the tournament.
3. **Selection of the Fittest**: The fittest individual from the tournament is identified. This individual is considered the winner of the tournament.
4. **Mating Pool Inclusion**: The winning individual is added to the mating pool, which will eventually contain the selected parents for breeding.

3.3.4 Breeding

Creating offspring in the generic algorithm is a key operation, allowing the exchange of genetic information between parent chromosomes to create offspring. With the given population, the selection of breeding pairs is inherent in each member. Each member will then produce one new offspring.

3.3.4.1 Crossover

The crossover function takes the parent chromosomes in the member of the population and a crossover rate as input. If the parent chromosomes have the same length, a regular crossover is performed by randomly selecting games and exchanging genetic material between the parents. If the parent chromosomes have different lengths, the crossover is performed based on the shorter length to ensure compatibility. Then, the following steps will ensue for each scenario:

1. **Determine Crossover Amount**: Based on the predetermined cross over rate and the length of the chromosomes, the function calculates the number of crossover points.
2. **Select Crossover Indices**: The function then randomly selects the number of indices found in the previous step from the range of chromosome lengths.
3. **Perform Crossover**: The selected indices in the previous step are then swapped with each other, creating a new combination of the ticket packages.

The crossover operation allows for the exchange of genetic information between parent chromosomes, creating new combinations of solutions that may potentially lead to better fitness scores.

3.3.4.2 Mutation

The mutate function introduces random variations in the offspring chromosomes by swapping elements at randomly selected indices. The function takes in the newly created off spring from crossover with a mutation rate parameter that determines the probability of mutation occurring. Then, the following steps are taken:

1. **Determine Number of Swaps**: Based on the mutation rate and the length of the offspring chromosomes, the function calculates the number of swaps to be performed.
2. **Perform Swaps**: For each swap, the function randomly selects two indices, one from each ticket package, and swaps the elements at those indices between the two offspring chromosomes.

The mutation operation introduces random variations in the offspring chromosomes, helping to explore new regions of the solution space and potentially escape local optima.

3.3.5 Insertion

After breeding, the offspring chromosomes are evaluated using the fitness score function to calculate the similarity scores of the new packages produced. Now, the insertion process involves integrating the offspring into the existing population. In this function, the goal is to create a new population by combining the fittest individuals from both the parent and offspring populations.

The way this function preserves the fittest members of the population by selecting elite solutions, in a process call elitism. First, the function preserves the top individuals from the parent population. This ensures that the best solutions are not lost during the selection process and move on to the successive generation. Then, the function removes the same number of solutions from children with the lowest fitness scores. This step ensures that the new population maintains its size by replacing the least fit individuals with the offspring. The two groups are then put together to create the newest generation of ticket schedule combinations. This new list, now containing a combination of offspring and elite solutions, is sorted in descending order based on the fitness scores. This sorting ensures that the fittest individuals are prioritized for selection and easy retrieval.

Insertion aims to create a new generation of solutions that combines the fittest individuals from both the parent and offspring populations. This approach helps maintain diversity while preserving the best ticket packages, ultimately driving the genetic algorithm towards optimal solutions.

Chapter 4: Results

4.1 Game Tiering

4.1.1 GAP stat Analysis

One of the primary goals in this research was to determine the ideal number of tiers that exist in the event dataset. To achieve this objective, several techniques were employed in order to evaluate their performance across a range from 2 to 10 clusters (denoted as k). The results of this clustering analysis are portrayed within the three plots presented below, each offering a perspective on the data's structure.

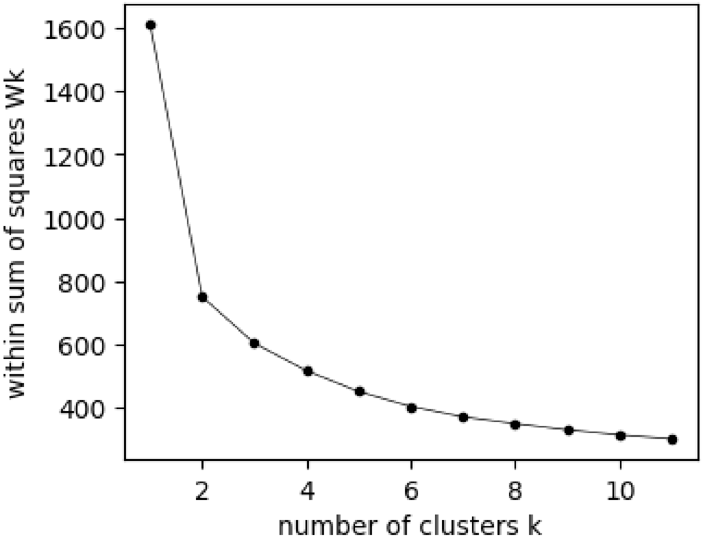

Figure 5: Number of Clusters k with Sum of Squares W_k

Figure 3 shows the within-sum of squares (WSS) plotted against the number of clusters (k). However, in Figure 3, it is not very clear where this elbow point might be. The graph shows a steady and gradual decrease in the WSS as the number of clusters increases from 2 to 10, without a distinct inflection point or elbow.

This lack of a clear elbow in the WSS plot makes it challenging to determine the optimal number of clusters based solely on this visualization. The WSS continues to decrease with each additional cluster, and there is no obvious point where the rate of decrease slows down significantly, which could indicate the appropriate number of clusters. When this happens, it becomes necessary to consider other cluster metrics or techniques to help determine the optimal number of clusters.

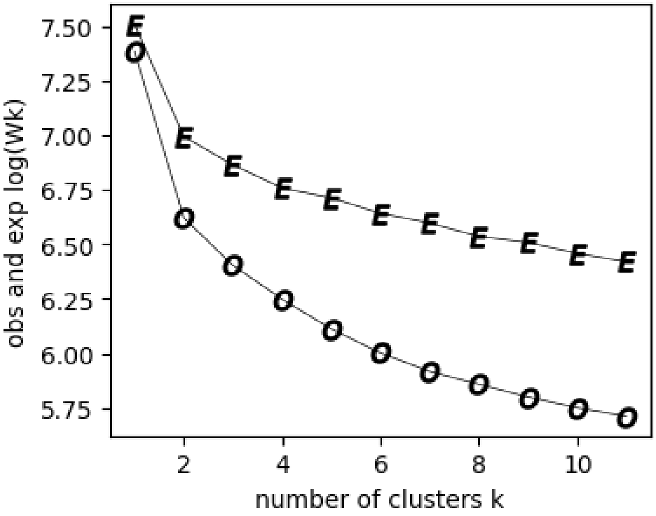

Figure 6: Number of Clusters k with Observed vs. Expected log (W_k)

Figure 4 displays the relationship between the number of clusters (k) and the observed vs. expected $\log(W_k)$ values in a clustering analysis. The plot contains two curves: one represented by the letter "E" for the expected $\log(W_k)$ values, and another represented by the letter "O" for the observed log (W_k)values. The x-axis represents the number of clusters ranging from 2 to 10, while the y-axis shows the corresponding log(Wk) values.

As the number of clusters increases from 2 to 10, there are several patterns that emerge. The expected $\log(W_k)$ curve decreases consistently and appears to level off or flatten out as k increases. In contrast, the observed $\log(W_k)$ curve decreases more rapidly.

The gap between the observed and expected curves grows as k goes from 2 to around 5 to 7 clusters, suggesting that the observed clustering structure becomes more distinct from the expected null distribution. However, after k equals 7, the gap between the curves starts to decrease slightly, indicating that the observed clustering may not be as pronounced compared to the expected values for higher k. Hence, according to figure 4, the optimal number of clusters is potentially in the range of 5 to 7, where the gap between the observed and expected $\log(W_k)$ values is maximized. It is crucial to consider this with further cluster validation metrics to make a more informed decision on the optimal number of clusters.

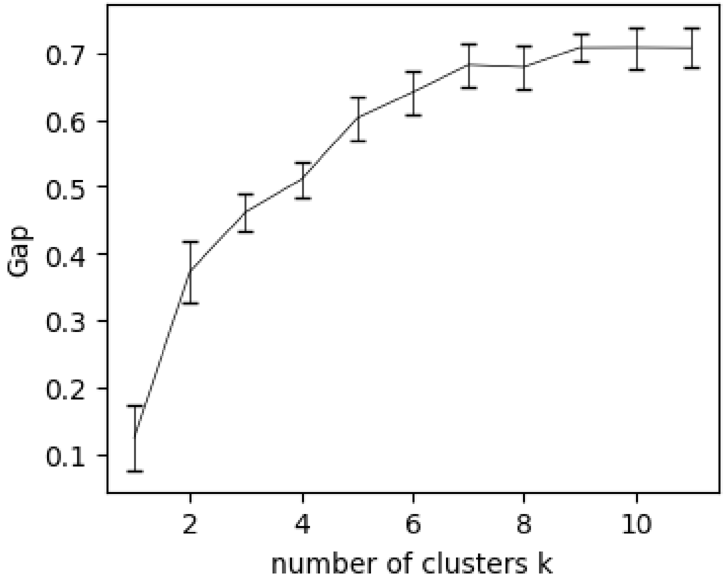

Figure 7: Number of Clusters k with GAP Score

Figure 5 depicts the relationship between the number of clusters (k) and the corresponding GAP statistic values, a metric used to determine an optimal number of clusters. As we scrutinize the plot, a few salient observations emerge.

The GAP statistic graph shows a steady increase as the number of clusters increases from 2 to 10. This upward trend suggests that increasing the number of clusters generally improves the clustering solution's quality. However, the rate of increase is not uniform – it's steeper from 2 to 4 clusters, indicating a significant improvement in the clustering solution when moving from 2 to 4 clusters.

Beyond 4 clusters, the GAP statistic continues to rise, but at a more gradual pace. This flattening of the curve implies that the benefit of adding more clusters shrinks after a certain point. The elbow, or the point where the curve starts to level off or drop, signifies the optimal number of clusters for the event data set. In the graph, the first and only drop point is after 7 clusters, indicating that there is not much improvement after k is 7.

When the results of figure 5 are paired with an automated analysis of the gap statistic provided in the code from John Maloney (2019), the optimal number of clusters was determined to be 7. Meaning that there are 7 buckets for classifying the games into.

4.1.2 Cluster Visualization

Taking the results of the GAP statistic, the k-medoids method was then run using 7 clusters. In figure 6 below, a the UMAP method is used to reduce the high dimensional space of the clusters to better visualize the data.

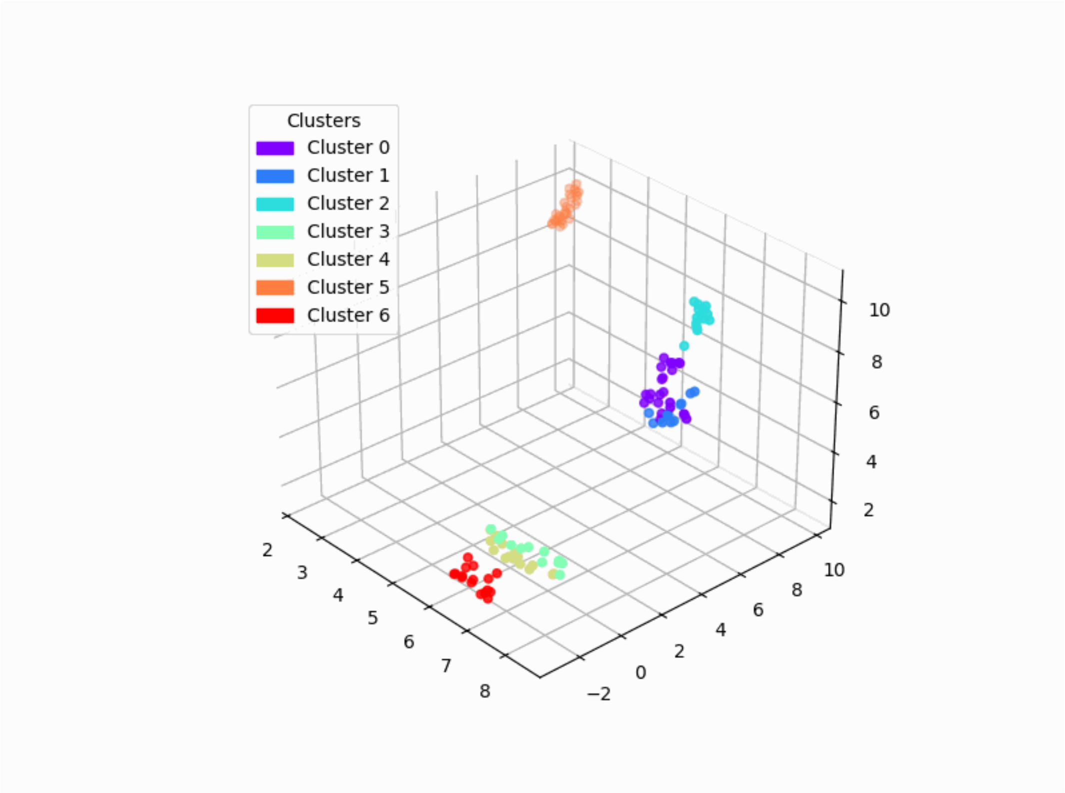

Figure 8: UMAP Visualization of K-Medoids Clusters

At first glance, the clusters appear as distinct groups, each occupying its own well-defined territory within the three-dimensional space. This visual separation is due to the algorithm's ability to identify and group data points based on their similarities. By looking closer, the distinctiveness of each cluster becomes even more apparent. The games within a single cluster are close knit communities that are clearly distinguishable from their neighboring clusters. This level of separation indicated effective clustering, where the algorithm has successfully identified the patterns and structures between the games.

Figure 6 also shows in the differences between the clusters themselves. While some clusters are densely populated, others are more sparsely distributed, creating a visual contrast. This variation adds a layer depth and complexity to the overall representation, reflecting the nuances of differing perceived value of events.

Overall, the Figure 6 presents an informative representation of the data's structure. The visual separation and distinctiveness of the clusters shows the effectiveness of the K-medoids algorithm in grouping the games into distinct categories. Due to these results, an analysis of the medoids cluster centers will ensue to better understand the distinct properties of these clusters.

4.1.3 Cluster Characterization

The characteristics of the medoid cluster centers reveals an inside look on the properties of the clusters. Table 3 and 4 below will show the numeric and categorical descriptions of each of the clusters.

Cluster	Total Revenue	Comp	Individual Member Group	Group	Secondary Market Sales	Show Rate
0	0	-0.23	0.21	0.58	-0.01	0.17
1	1	0.09	-0.85	-0.25	0.31	0.58
2	2	-0.06	2.43	-1.11	-1.29	-1.2
3	3	-0.29	-1.01	-0.6	1.01	-0.5
4	4	-0.11	-0.78	-0.19	1.07	0.95
5	5	0.61	-0.64	-1.11	1.29	-0.5
6	6	-0.92	1.12	0.49	-1.35	0.14

Table 3: Numeric Observations of Cluster Centers (Standard Deviations from Mean)

Cluster	NFL Sunday	College Football Saturday	Rank in Conference	Number of All-Stars	Guardians Game	After All-Star	Late Night Game	Weekend	LeBron
0	0	0	8	1	0	0	1	0	0
1	1	0	8	0	0	0	1	1	0
2	0	0	3	1	0	0	1	0	0
3	0	0	12	1	0	0	1	0	0
4	0	0	13	0	0	1	1	0	0
5	0	0	3	1	0	1	1	0	0
6	0	0	12	0	0	1	1	1	0

Table 4: Categorical Observations of Cluster Centers

Cluster 0 – Mid-tier games with moderate individual and group ticket sales, average attendance, and involving playoff teams with an All-Star(s) before the all-star break.

Cluster 1 – Games with higher group and member ticket sales, low comp tickets, involving playoff teams without All-Stars on weekends before the all-star break, and average revenue and attendance.

Cluster 2 – Games with high comp tickets, low individual, member, and group ticket sales, involving elite teams with an All-Star(s) before the all-star break, but low attendance.

Cluster 3 – Games with high member ticket sales, low comp, individual, and group ticket sales, involving low-tier teams with All-Star(s) before the all-star break, and above-average attendance.

Cluster 4 – Games with high member and group ticket sales, high secondary market sales, involving the lowest-tier teams without All-Stars after the all-star break, and above-average attendance.

Cluster 5 – High-revenue games with high member ticket sales, low individual and group ticket sales, involving elite teams with at least All-Star after the all-star break, and high attendance.

Cluster 6 – Low-revenue games with high comp and individual ticket sales, low member and secondary market sales, involving low-tier teams without All-Stars on weekends after the all star break, and below-average attendance.

4.1.4 Cluster Tiering

Additional contextual analysis is conducted on these clusters to form them into tiers using domain expertise. By consulting with Quinn Spangler, the Quantitative Data Analyst that oversees the current tiering process at the Cleveland Cavaliers, the tiers shown in table 5 were then determined.

Tier	Cluster
A	5
B	0
C	1
D	4
F	3
G	2
H	6

Table 5: The Organizing of Clusters into Hierarchical Tiers

By organizing the games into hierarchical tiers, we can better understand which types of games are more appealing to be able to better group them into rankings based on their group's attributes. This vastly reduces the complexity of the cluster attributes into a more consumable format after domain specific logic was applied. Now, this will allow the genetic algorithm to better understand the perceived value of the games and be able to conduct the chi square distribution test for part of the evaluation function.

A comparison analysis between each tier and the tier below it was then conducted. (Quinn Spangler, personal communication, June 12, 2024)

4.1.4.1 A Tier (Cluster 5) vs. B Tier (Cluster 0)

Cluster 5 has a much higher level of performance compared to Cluster 0 across multiple metrics. It demonstrates above-average total revenue (0.61), while Cluster 0 has below-average total revenue (-0.65). Also, Cluster 5 experiences well above-average member ticket sales (1.29), significantly outperforming Cluster 0's member ticket sales (-0.01). This suggests a stronger demand from members for events in Cluster 5 and signifies that more members have these events included in their selected packages. Furthermore, events in Cluster 5 have a higher show rate (0.53) compared to those in Cluster 0 (0.37), implying a greater attendance and demand. Both clusters involve elite or playoff teams, but Cluster 5's elite team (rank 3) is more attractive than Cluster 0's playoff team (rank 8), potentially contributing to the observed differences in revenue and ticket sales. Overall, the large interest and high quality of the opponents in Cluster 5 show why it is the top tier of games.

4.1.4.2 B Tier (Cluster 0) vs. C Tier (Cluster 1)

Cluster 0 has below-average total revenue (-0.65), while Cluster 1 demonstrates average total revenue (0.09), suggesting that events in Cluster 1 generate higher overall sales. However, Cluster 0 has above-average individual ticket sales (0.58), outperforming Cluster 1's below-average individual ticket sales (-0.25). This indicates a stronger demand for individual tickets for events in Cluster 0, which is a great pulse of fan demand overall as this represents fans who are selecting this game in particular. Conversely, Cluster 1 experiences higher group ticket sales (0.58) compared to Cluster 0 (0.17), suggesting a greater appeal for group attendance at events in Cluster 1. Group ticketed events are generally stronger for lower tier games as a cost-effective way for companies or organizations to provide a perk for their employees since these games would be priced lower. Events in Cluster 0 have a higher show rate (0.37) compared to those in Cluster 1 (0.01), implying a greater attendance and demand for events in the former cluster. Both clusters involve playoff teams, but Cluster 0 has one All-Star, making it slightly more attractive in terms of team performance. The addition of the star power is what gives Cluster 0 the edge as the higher tier game.

4.1.4.3 C Tier (Cluster 1) vs. D Tier (Cluster 4)

Cluster 1 produced about average total revenue (0.09), while Cluster 4 demonstrates slightly below-average total revenue (-0.11), suggesting a marginal difference in overall sales between the two clusters. Cluster 4 experiences higher member ticket sales (1.07) compared to Cluster 1 (0.31), indicating a stronger demand from members for events in Cluster 4. However, as the tiers get lower, member demand begins to be less of a separator between tiers since these games are often paired with high tiers games to balance out the packages for evenness of value. Additionally, Cluster 4 has higher group ticket sales (0.95) compared to Cluster 1 (0.58), suggesting a greater appeal for group attendance at events in Cluster 4. Events in Cluster 4 also see higher secondary market sales (0.90) compared to those in Cluster 1 (-0.14), implying a greater demand and resale activity for tickets in the former cluster. The higher sales indicate that the original ticket buyers are not looking to attend the game. Furthermore, Cluster 4 has a higher show rate (0.34) compared to Cluster 1 (0.01), indicating a greater attendance and demand for events in the former cluster. However, Cluster 1 involves playoff teams on the weekend, while Cluster 4 involves the lowest-tier teams on weekdays, making Cluster 1 more attractive on face value despite lower ticket sales.

4.1.4.4 D Tier (Cluster 3) vs. F Tier (Cluster 4)

Cluster 3 has below-average total revenue (-0.29), while Cluster 4 has more average total revenue (-0.11). Cluster 3 has well below-average complimentary tickets (-1.01) compared to Cluster 4's below-average complimentary tickets (-0.78). Additionally, Cluster 3 has below-average individual ticket sales (-0.60) compared to Cluster 4's below-average sales (-0.19). Both clusters have high member ticket sales, but Cluster 4 (1.07) is slightly higher than Cluster 3 (1.01). Cluster 4 experiences higher group ticket sales (0.95) compared to Cluster 3 (-0.50). Furthermore, Cluster 4 exhibits higher secondary market sales (0.90) compared to Cluster 3 (-0.07). Cluster 3 involves low-tier teams (rank 12), while Cluster 4 involves the lowest-tier teams (rank 13). Cluster 3 has one All-Star, while Cluster 4 has none. Events in Cluster 3 have a higher show rate (0.38) compared to those in Cluster 4 (0.34). Given the better marquee matchup that cluster 3 has with the all-star, this what ultimately provides the separation needed to rank Cluster 3 above Cluster 4.

4.1.4.5 F Tier (Cluster 4) vs. G Tier (Cluster 2)

Cluster 4 exhibits average total revenue (-0.11), while Cluster 2 demonstrates average total revenue (-0.06). However, Cluster 2 has well above-average complimentary tickets (2.43) compared to Cluster 4's below-average complimentary tickets (-0.78). Cluster 2 also has well below-average individual ticket sales (-1.11) compared to Cluster 4's below-average sales (-0.19). Conversely, Cluster 4 experiences higher member ticket sales (1.07) compared to Cluster 2's well below-average sales (-1.29). Additionally, Cluster 4 has higher group ticket sales (0.95) compared to Cluster 2's well below-average sales (-

1.20). Furthermore, Cluster 4 has higher secondary market sales (0.90) compared to Cluster 2 (-0.09). Cluster 2 involves elite teams (rank 3), while Cluster 4 involves the lowest-tier teams (rank 13). Events in Cluster 4 have a higher show rate (0.34) compared to those in Cluster 2 (-0.71). Even though the events in Cluster 2 have all stars and elite teams, the sales and show rate results show that these games are not in high demand for purchasing or attending the games. With sales and attendance being the primary driver of the creation of tiers, this is what sets Cluster 4 ahead of Cluster 2.

4.1.4.5 G Tier (Cluster 2) vs. H Tier (Cluster 6)

Cluster 2 has about average total revenue (-0.06), while Cluster 6 underperforms with well below-average total revenue (-0.92). Additionally, Cluster 6 experiences above-average individual ticket sales (0.49) compared to Cluster 2's well below-average sales (-1.11). Both clusters have well below-average member ticket sales, but Cluster 6 (-1.35) is slightly lower than Cluster 2 (-1.29). Cluster 6 also has slightly above-average group ticket sales (0.14) compared to Cluster 2's well below-average sales (-1.20). Both clusters involve low-tier teams, but Cluster 6 (rank 12) is far worse than Cluster 2 (rank 3). Cluster 2 has one All-Star, while Cluster 6 has none. Events in Cluster 6 have a below-average show rate (-0.08) compared to Cluster 2's well below-average show rate (-0.71). The decision to rank Cluster 2 above Cluster 6 comes down to the revenue numbers. With the revenue of Cluster 2 far outperforming Cluster 6, this ultimately drives the decision along with the better quality of the opponent in comparison.

4.1.5 Pairwise Determination of 2023-24 Schedule

By using the pairwise distance of each game's pre-season statistics outlined in Section 3.2.4, the games of this past season can be categorized into the tiers described above. The following table outlines the schedule in order of home game with the respective tier assigned to it.

Tier	Date	Team
A	10/31/23	New York Knicks
A	11/5/23	Golden State Warriors
A	11/19/23	Denver Nuggets
A	12/29/23	Milwaukee Bucks
A	1/17/24	Milwaukee Bucks
A	1/29/24	Los Angeles Clippers
A	2/5/24	Sacramento Kings
A	2/12/24	Philadelphia 76ers
A	3/3/24	New York Knicks
A	3/5/24	Boston Celtics
A	3/10/24	Brooklyn Nets
A	3/11/24	Phoenix Suns
A	3/20/24	Miami Heat
A	3/29/24	Philadelphia 76ers
A	4/10/24	Memphis Grizzlies
B	11/22/23	Miami Heat
B	1/15/24	Chicago Bulls
B	2/14/24	Chicago Bulls
C	11/25/23	Los Angeles Lakers
C	11/26/23	Toronto Raptors
C	11/28/23	Atlanta Hawks
C	12/16/23	Atlanta Hawks
C	12/21/23	New Orleans Pelicans
C	1/5/24	Washington Wizards
C	3/8/24	Minnesota Timberwolves
D	10/27/23	Oklahoma City Thunder
D	10/28/23	Indiana Pacers
D	11/30/23	Portland Trail Blazers
D	12/6/23	Orlando Magic
D	12/18/23	Houston Rockets
D	12/20/23	Utah Jazz
D	1/3/24	Washington Wizards
D	2/27/24	Dallas Mavericks
F	11/17/23	Detroit Pistons
F	1/7/24	San Antonio Spurs
F	1/31/24	Detroit Pistons
F	2/22/24	Orlando Magic
F	3/25/24	Charlotte Hornets
F	4/12/24	Indiana Pacers
F	4/14/24	Charlotte Hornets

Table 6: 2023-24 Cleveland Cavaliers Game Tiers

Looking closer at the distribution of games between tiers in the below figure 9, there are interesting patterns that emerge,

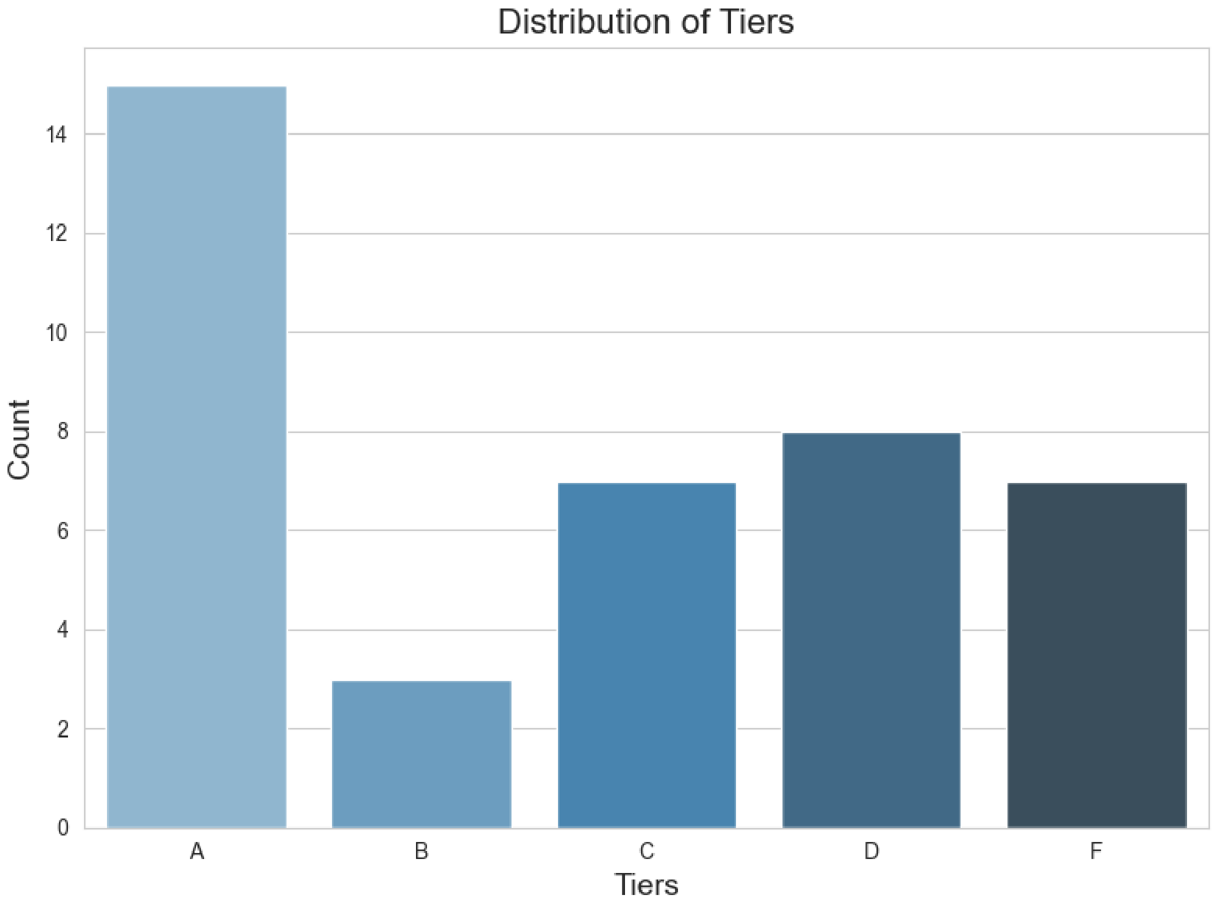

Figure 9: Distribution of Games into Tiers (Games)

First, notice the exclusion of the G and H tiers. No games in the 2023-24 season are identified in these tiers. Since these games are regarded as having lower stature, this is allowable as the business teams at the Cavaliers would still have a logical model to go off. This can be attributed to the variation in both the schedule and opponents year over year. Additionally, the games are heavily favoring the top tier. This can be attributed to off-season moves of some of the top teams in the league. Teams that had an elite ranking the previous year have acquired new talent putting them in a higher tier with a higher expected perceived value. Moves like Kevin Love to the Miami heat, Kristaps Porzingis to the Boston Celtics, and Damion Lillard to the Milwaukee Bucks also, the emergence of All-Star talent like Tyrese Maxey of the 76ers and Jalen Brunson of the New York Knicks can contribute to teams belonging in the higher tier of games based on their composition. We also notice some bias from the Cluster Centers favoring better rankings for the games in the second half of the season for the imbalance of the A versus B tier. Overall, having a logical breakdown of the quality of games creates a logical breakdown of the schedule to not only better inform ticket package creation, but to inform pricing of tickets for individual and group pricing structures.

Through consulting with the domain experts at the Cavaliers, the tiers produced above have their advantages and disadvantages. Increased depth of knowledge about games in each tier through the analysis of the medoids allows each tier to tell a story. Since the analysis was conducted subjectively with expert input, it allows for the proposal to be flushed out with not just a score based on monetary value, but additional reasoning and context to business stakeholders. However, they did add that having the weighted model did help facilitate conversations and boil down the algorithm’s decision-making process into a more interpretable measure (Quinn Spangler, personal communication, June 1, 2024). An area of improvement on this method could look to do away with the arbitrary approach to create this value. Additionally, they noted this continues to allow games and packages to be marketed given the tier’s medoid values. By analyzing fan behavior, these tiers can be used to market games effectively and determine relative price points between the tiers.

One game was found to be out of place in the result was surrounding the Los Angeles Lakers falling into tier C. This is generally the highest demand game on the schedule each year due to Lebron James being an NBA Champion with the Cleveland Cavaliers. Although the Lebron input variable tried to account for this, the model decided it was not significant enough. Further work to refine this should be conducted. In further talks with Quinn Spangler, he said this is a common occurrence with model outputs and that working out the movement of games between tiers is a yearly occurrence with business stakeholders. The important part, he noted, is that the algorithm gets the decision close to the finish line, only leaving a couple adjustments to be made (Quinn Spangler, personal communication, June 2, 2024).

4.2 Genetic Algorithm Evaluation

By consulting the Cavalier's resident expert on ticket packages and Director of Business Intelligence, Canaan Campo, he provided the weights for the evaluation function based on his prior experience in building these packages. After deliberation with Quinn Spangler, he decided that perceived value would be assigned the value of 1, weekend would be assigned 3, conference would be assigned 1, and date score would be assigned 5. His thought process behind this is that he wants the algorithm to put more importance on the areas of difficulty when creating the half season packages (Canaan Campo, personal communication, June 14, 2024). These weights mean that perceived value, weekend, conference, and date score account for 10%, 30%, 10%, and 50% of the evaluation metric respectively. The advantage of the weights allows for domain experts to have a level of influence on the decision making of the algorithm and can fine tune the process by applying their expertise.

4.2.1 Sensitivity Analysis

Tuning the hyperparameters of the genetic algorithm helps gain insights into how quickly and robust the algorithm is. By performing a sensitivity analysis, the best combinations to find global maxima can be determined. For this process, the algorithm will be tested in 108 combinations, including 4 population sizes, 3 generation amounts, 3 crossover rates, and 3 mutation rates. The population sizes selected were 250, 500, 750, and 1000. The generation amounts selected were 200, 400, and 600. The crossover rates selected were 20%, 40%, and 60%. Finally, the mutation rates selected were 5%, 10%, and 15%.

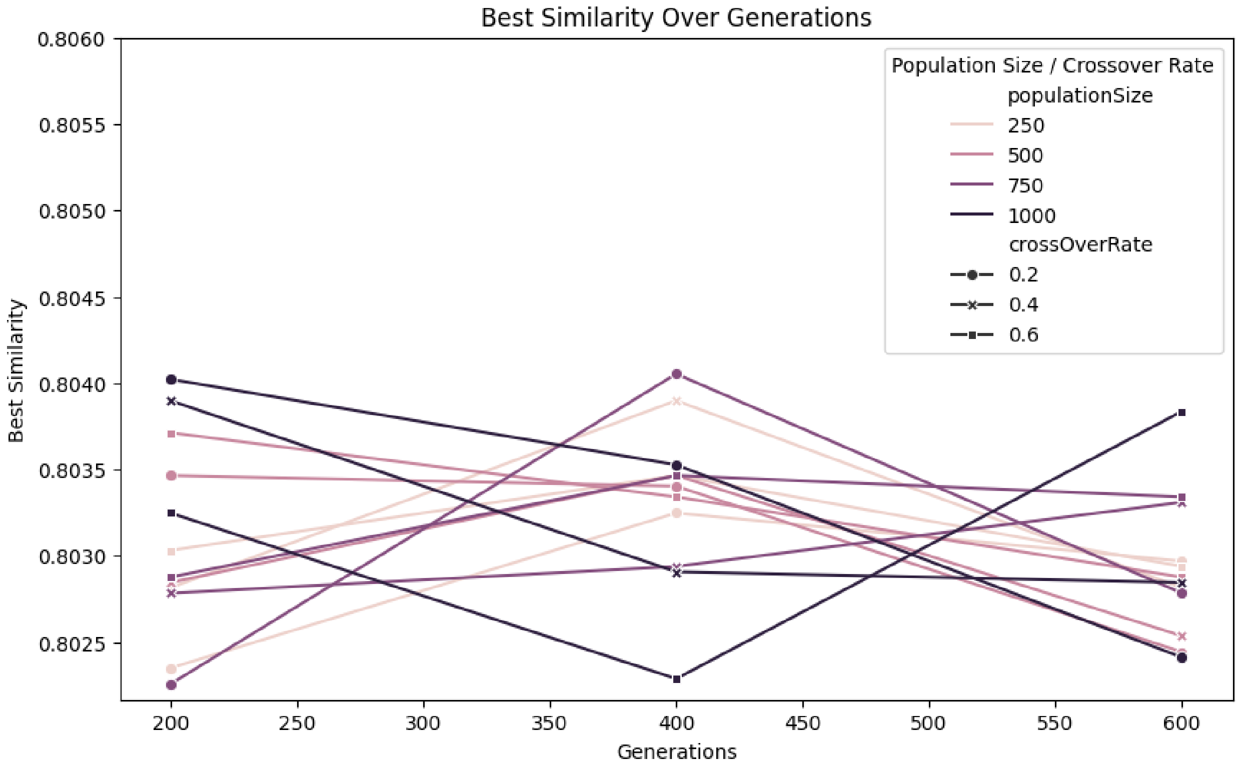

Figure 10: Sensitivity Analysis: Generations, Population Size, and Crossover Rate

In figure 10, first notice the y-axis is separated by 3 thousandths. This demonstrates the stability and consistency of the genetic algorithm's results under different parameter settings. This means the algorithm is rather stable under changing conditions at finding near optimal solutions that vary slightly from each other. The derived hyperparameters from this visual that will be used for the remainder of this study are from the peak in the center at 400 generations where the population is 750 and the crossover rate is 20%. This set of parameters shows the best performance of all combinations ran. Now, the final hyperparameter to derive from this analysis is the mutation rate.

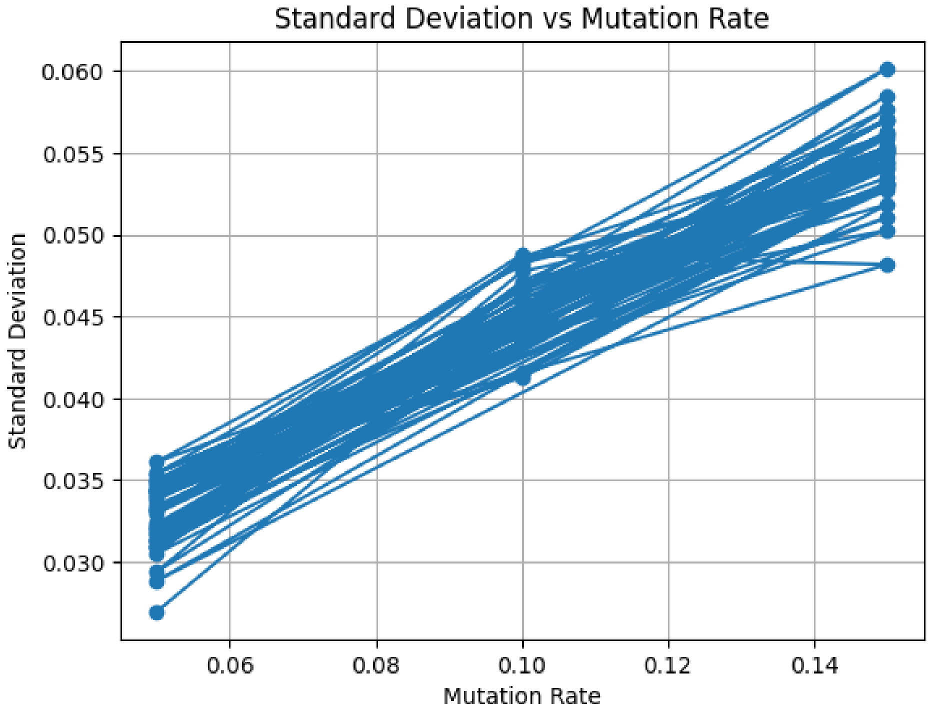

Figure 11: Mutation Rate vs. Standard Deviation of Population Solutions

In figure 11, it is demonstrated that by raising the mutation rate, it increases the diversity of packages present in the population. This graph shows an expected behavior from the mutation rate and demonstrates that it is useful in escaping local maxima as the algorithm searches through the solution space. This diversity is an advantage of the algorithm as it add a variability to the population. The mutation rate of the optimal solution from the previous section is 0.05, so the remainder of this study will utilize this rate as it was used to find the best solution from the sensitivity analysis. Table 7 below details the optimal hyperparameters that will be used:

Population Size	750
Generations	400
Crossover Rate	0.2
Mutation Rate	0.05
Elite Solutions	75
Tournament Size	75
Mating Pool Size	750

Table 7: Optimal Hyperparameters

4.2.2 Comparison to Internal 23-24 Season Packages

The below table shows the differences in the ticket packages when the new tiers are inserted into the genetic algorithm for optimization versus the original ticket splits. It is important to note that the

Date	Opponent	Tier	Original Package	New Package
10/27/23	Oklahoma City	D	A	B
10/28/23	Indiana	D	B	A
10/31/23	New York	A	A	B
11/5/23	Golden State	A	A	A
11/17/23	Detroit	F	B	B
11/19/23	Denver	A	B	A
11/22/23	Miami	B	A	A
11/25/23	L.A. Lakers	C	B	B
11/26/23	Toronto	C	A	A
11/28/23	Atlanta	C	B	A
11/30/23	Portland	D	A	B
12/8/23	TBD	D	B	A
12/16/23	Atlanta	C	A	A
12/18/23	Houston	D	A	B
12/20/23	Utah	D	B	B
12/21/23	New Orleans	C	A	A
12/29/23	Milwaukee	A	B	B
1/3/24	Washington	D	A	A
1/5/24	Washington	C	B	B
1/7/24	San Antonio	F	A	A
1/15/24	Chicago	B	B	B
1/17/24	Milwaukee	A	A	A
1/29/24	LA Clippers	A	B	A
1/31/24	Detroit	F	A	B
2/5/24	Sacramento	A	B	A
2/12/24	Philadelphia	A	A	B
2/14/24	Chicago	B	A	A
2/22/24	Orlando	F	B	B
2/27/24	Dallas	D	A	A
3/3/24	New York	A	B	B
3/5/24	Boston	A	B	A
3/8/24	Minnesota	C	A	B
3/10/24	Brooklyn	A	B	A
3/11/24	Phoenix	A	A	B
3/20/24	Miami	A	B	B
3/25/24	Charlotte	F	A	A
3/29/24	Philadelphia	A	B	B
4/10/24	Memphis	A	B	B
4/12/24	Indiana	F	A	A
4/14/24	Charlotte	F	B	A

Table 8: New Tier Optimization vs Original Ticket Package

By looking at the two packages above, there exists very little difference in the date spread of the two packages. By consulting the Cavalier's resident expert on ticket packages and Director of Business Intelligence, Canaan Campo, this spread is attractive since it does not include 3 games in a row for any one package (Canaan Campo, personal communication, June 14, 2024). He added that the advantage of the algorithm taking into the account of the range of dates is beneficial since usually business leaders will ask for 10 games in each half of the season, which is present in the new packages as well. Based on the evaluation function's date score metric, the new packages (0.75) narrowly improve on the original's score (0.70). Given slight improvement equitable comparison along the date spread, the two package combinations are then compared on the weekend level.

Original Package		
Day of Week	A	B
Weekend	7	10
Weekday	13	10

New Package		
Day of Week	A	B
Weekend	9	8
Weekday	11	12

Table 9 & 10: Original Packages vs New Packages on Weekend

As previously stated in section 4.2, this is one of the pain points in the process and therefore it was given the second highest weight in the evaluation function (Canaan Campo, personal communication, June 14, 2024). In the original package, there exists an imbalance that is greater than the new package. This shows the improvement of the algorithms ability to strike this balance between the packages on weekend over human decision making. This is also evident in the Chi-Squared tests where the original package has a p-value of 0.61 and the new package has a p-value of 0.82 compared to the total schedule's distribution. This means that the distribution of weekday to weekend games in the new packages is closer to the overall schedule's distribution than that of the original package. The next part of the evaluation function of the algorithm is the comparison on the distribution of conferences.

Original Package		
Conference	A	B
East	14	11
West	6	9

New Package		
Conference	A	B
East	11	14
West	9	6

Tables 11 & 12: Original Packages vs New Packages on Conference

Based on the above distributions, both package combinations are equivalent on their distribution of conferences. This can be attributed to the weights of the algorithm not prioritizing the conferences as much as the other variables, which were deemed more important by the subject matter experts at the Cleveland Cavaliers. (Quinn Spangler, personal communication, June 14, 2024)

Now, since the two package combinations were determined using different tiering structures, it is not equitable to compare performance on tiers. While the new tiering method laid out in this research provides value to the organization, in order to conduct a proper analysis of the algorithm compared to the original ticket package, the tiers used for the original packages are plugged into the algorithm. This also showcases the flexibility of the evaluation function since it can act as a “bring your own tier” system, allowing for flexibility depending on how an organization determines its tiers. Table 9 showcases the usage of the original tiers in the genetic algorithm versus the original packages.

Date	Opponent	Tier	New Packages	Original Packages
10/27/23	Oklahoma City	C	A	A
10/28/23	Indiana	C	B	B
10/31/23	New York	B	A	A
11/5/23	Golden State	A	B	A
11/17/23	Detroit	C	A	B
11/19/23	Denver	B	B	B
11/22/23	Miami	B	B	A
11/25/23	L.A. Lakers	A	A	B
11/26/23	Toronto	D	B	A
11/28/23	Atlanta	D	B	B
11/30/23	Portland	D	A	A
12/8/23	TBD	D	B	B
12/16/23	Atlanta	B	B	A
12/18/23	Houston	D	A	A
12/20/23	Utah	D	B	B
12/21/23	New Orleans	C	A	A
12/29/23	Milwaukee	A	A	B
1/3/24	Washington	D	B	A
1/5/24	Washington	C	B	B
1/7/24	San Antonio	B	A	A
1/15/24	Chicago	C	B	B
1/17/24	Milwaukee	B	A	A
1/29/24	LA Clippers	B	B	B
1/31/24	Detroit	D	A	A
2/5/24	Sacramento	D	A	B
2/12/24	Philadelphia	B	B	A
2/14/24	Chicago	C	A	A
2/22/24	Orlando	D	A	B
2/27/24	Dallas	B	B	A
3/3/24	New York	B	A	B
3/5/24	Boston	B	B	B
3/8/24	Minnesota	B	B	A
3/10/24	Brooklyn	B	A	B
3/11/24	Phoenix	A	B	A
3/20/24	Miami	B	A	B
3/25/24	Charlotte	D	B	A
3/29/24	Philadelphia	B	A	B
4/10/24	Memphis	B	A	B
4/12/24	Indiana	C	B	A
4/14/24	Charlotte	D	A	B

Table 13: Original Tier Optimization vs Original Ticket Packages Schedule

Now since each package has been wholly generated using the same foundation, the comparison between the two becomes clearer cut since each uses an identical data set for the evaluation function and the overall similarity scores given by both combinations. The genetic algorithm ticket package scored 0.8, while the original ticket package scored 0.715, which is about a 12% improvement in performance accuracy wise overall. In Table 10, the differences along each of the metrics are presented.

	Optimized Package	Original Package
Perceived Value	1	1
Weekend	0.82	0.61
Conference	0.82	0.49
Date Score	0.74	0.70

Table 14: Original Tier Optimized Packages vs Original Packages Evaluation Score

Both packages are balanced along the tiers perfectly with the distribution. According to Canaan Campo, this is a must have in any half season ticket package since balance of tiers is key to the business. The improvement over the original solution in the other 3 categories is substantial, given the struggles each year to strike this multivariable balance manually. Overall, along these evaluation metrics, it has been shown that the usage of a genetic algorithm for half season ticket package optimization for the NBA schedule improves both the quality and speed of producing said packages.

Chapter 5: Conclusion

Through the case study of the Cleveland Cavaliers' 2023-24 half season ticket packages, this research has demonstrated the effectiveness of applying a genetic algorithm, coupled with machine learning-based game tiering, in optimizing ticket package combinations within the event and entertainment industry. The approach presented offers significant potential for enhancing operational efficiencies and product quality, as evidenced by the results of this study. The successful implementation of this methodology highlights the value of leveraging advanced computational techniques to address complex challenges in the ticketing industry.

Being able to tier games based on demand is an effective use of machine learning, specifically K-medoids, increases the depth of knowledge about games. By understanding games in detail, the model output aides the decision-making process by providing additional context that drive decisions through data. Then, applying domain knowledge to the clusters, a relative demand value of each game can be reduced into a more consumable ranking. The benefits of this span outside of just the ticket packaging aspect of sports business, but to ticketing in general. In providing the relative value of a game, it can help price different types of inventories, like individual and group ticket sales. Through the use of machine learning in this application, it allows for high dimensional decisions to be broken down into consumable and actionable outputs for the business to apply strategy towards.

The genetic algorithm proposed in this study mimics the process of the business intelligence teams of the NBA by making swaps of games in the schedule until the optimal solution emerges that aligns with the business logic. By embedding domain expertise from business intelligence professionals from the Cavaliers into a weighted evaluation function, this allows the algorithm to judge the quality of the solutions just as they would. Unlike the manual process, however, the genetic algorithm cuts down the process to just under 10 minutes compared to the week or two that it currently takes most teams.

The speed and accuracy that this solution uses has several advantages. The business can now leverage the momentum of the schedule release in August to go to market with ticket packages sooner. By doing so, a team can capitalize on the excitement of the schedule release with their members that are more to buy the half season packages. Given the timing of this release lining up right before the start of the NFL schedule, it is advantageous for teams that share a home city with NFL teams to go to market before a sports fans gaze turns to the NFL regular season.

Additionally, the teams can benefit from the quality of the product. By creating equitable demand for each of the half season packages, it creates an overall higher combined product offering. The implications of this could lead to higher sales of packages since inventory is limited for each. If there exists an imbalance, fans may favor one package over another, leaving inventory sold out for one half and unsold for other half of the games in the schedule. The impact on revenue for a team could be substantial in being able to more easily sell through both forms of inventory, therefore increasing sales. Another implication of undersold games is the impact on revenue of a fan showing up to a game. There are more ways that fans contribute to revenue rather than just by buying a ticket. When a fan enters the arena, they are presented with options to better their experience by purchasing team apparel or concessions. By getting more consistent and higher attendance for all games, multiple fan-generated revenue streams can be boosted in the process.

With the application of different business logic into the evaluation function, the impact of this approach stretches beyond the sports industry. Any event driven business could optimize their events into different packages making for interesting and appealing products and newly available inventory. For instance, a concert venue could place their events into this algorithm to create new season long ticket packages that could appeal to consumers. The advantage of this sales method would be to pair the big ticket shows with shows that usually don't sell through based on the perceived value of past performances for similar acts. By using this method to aide in the creation of this inventory, the venue can put out higher quality or new inventory. This also helps the consumer not having to deal with the secondary market prices for each show that has plagued the concert industry. Opening solutions like this up to drive revenue and customer satisfaction in industries outside of the NBA and sports is why this algorithm is much more than a one-off use, but rather a new method for the entirety of the event industry to benefit from.

While there are benefits from the development of this methodology, there exist limitations that must be addressed. The data restriction in running inference in the clustering method could use improvement. While the results were serviceable and provided insights on the type of games in the schedule, the inference dataset was limited. One way to improve this is to run predictive algorithms to impute the numerical variables of an event, such as total revenue and show rate. Given a high enough level of accuracy from these models, the results could help solidify the inference for the tiers. Furthermore, to improve tier determination, a weight could be assigned to each of the variables. In a similar lane as imputing values for machine learning, an analysis on an individual variable level on the effect on demand should be conducted. Then, by creating coefficients for each variable in the medoid centers, an overarching metric for the medoids can be ranked to form a hierarchical tier list. With these improvements, the tiering methodology could be greatly improved.

Additionally, a future improvement to the genetic algorithm is to add in a constraint to not allow a team to be in the same package twice. This could be done during the creation of the population by only selecting individuals that fit this criterion. Additionally, this constraint would need to be enforced in the breeding process, either by only allowing offspring post crossover that fit the criterion to live on, or by retrying the crossover until a valid offspring is created.

The next logical application of this method would be the analyze quarter season and eighth season packages. The methodology used in this approach to half season packages directly applies to smaller packages as well. The algorithm was built to accommodate any specified number of splits less than or equal to half the number of events, allowing for the optimization of a multitude of splits. This higher quantity of packages poses a new challenge for the robustness of the algorithm

Additionally, using fan behavior to determine the perceived value tiers as well as the weights for the evaluation function could create an avenue for personalization of packages. For instance, we can study the behavior of different segments of the fan base that attend several games a year to determine which games they value as well as their behavior around the number of games they go to, weekends, conferences, and the date spread of the games they go to. Curating packages can help aide the migration of non-members to members based on their preferences and needs according to the parameters of the algorithm, ultimately providing a broader reach of packages for fans.

The adoption of innovative solutions, such as machine learning and genetic algorithms presented in this study, is crucial for sports organizations seeking to remain competitive and deliver exceptional fan experiences. As demonstrated through the successful application of these techniques in optimizing ticket package combinations for the Cleveland Cavaliers, the potential benefits can help with both speed and quality of decision making. By embracing data-driven approaches, sports organizations can improve operational efficiency, increase revenue generation, and provide more value to their fans.

As the sports industry continues to evolve, the integration of cutting-edge technologies and analytical techniques will become increasingly vital. The insights gained from advanced analytics can help businesses better understand their fans' preferences and tailor their offerings to meet those needs. The success of the genetic algorithm and machine learning game tiering in optimizing ticket packages for the Cleveland Cavaliers serves as a testament to the immense potential of these approaches. By staying at the forefront of these developments and embracing data-driven decision-making, sports organizations can position themselves for long-term success, fostering stronger connections with their fans and driving the industry forward in exciting new directions. The work presented in this thesis serves as a steppingstone towards a future where advanced analytics and cutting-edge technologies are seamlessly integrated into every aspect of sports business decision making, driving unparalleled growth, fan engagement, and competitive advantage as the sports industry continues to embrace innovation and harness the power of data.

Appendix A: Tiering Data Sample

There are 4 anonymized data samples provided, each row represents a game. This dataset is derived from the tiering dataset before features were removed from the dataset following the correlation heatmap analysis.

total_revenue	med_paid_amount	comp	individual	member	group
-0.11	0.20	-1.27	0.67	1.47	1.93
0.72	0.67	-0.88	0.96	1.34	0.56
-0.85	-0.78	0.75	0.78	-0.81	-0.25
-0.98	-0.88	0.40	0.06	-1.23	-1.14

secondary_market_sales	nfl_sunday	college_football_saturday
2.17	FALSE	FALSE
0.62	FALSE	FALSE
-0.83	FALSE	FALSE
-1.34	FALSE	FALSE

win_perc	rank_in_conference	num_all_stars	guardians_game
0.45	10.00	0.00	TRUE
0.62	2.00	1.00	FALSE
0.29	13.00	1.00	FALSE
0.32	13.00	0.00	FALSE

after_all_star	late_night_game	is_weekend	lebron	show_rate
TRUE	TRUE	TRUE	FALSE	1.30
TRUE	TRUE	FALSE	FALSE	1.30
TRUE	TRUE	FALSE	FALSE	-0.58
TRUE	TRUE	FALSE	FALSE	-0.66

Appendix B: List of Included Attributes for K-medoids Algorithm

total_revenue **(Numeric)**	The total revenue of a game is the culmulative ticket generated revenue produced by an event. This encompasses all ticket sales of all types of product. This is a key performance indicator (KPI) of the success of a game on the aggregate level
Comp **(Numeric)**	Complimentary tickets are the tickets that are given away at no charge. These are often given out to employees and family members of the players.
Individual **(Numeric)**	Individual tickets are tickets that are bought for a single game. These can range in quantity for any particular game. Typically, these are purchased through the team's ticketing provider and are bought online directly from the team.
Member **(Numeric)**	Member tickets are those who have bought full season, half season, or quarter season packages with multiple games included in the package.
Group **(Numeric)**	Group tickets are bought through a sales rep in larger quantities defined by the business.
secondary_market_sales **(Numeric)**	These are a type of individual ticket that is not being bought directly from the business, but from a person who has already purchased a ticket and has chosen not to attend the event
nfl_Sunday **(Binary)**	This is true for every Sunday from September through January, when the NFL regular season and post season is taking place.
college_football_Saturday **(Binary)**	This is true for every Saturday from August through December, when the College football regular season is taking place.
rank_in_conference **(Integer)**	The respective ranking based on win percentage of the team from the previous season, with 1 being the best, and 15 being the worst in the respective conference. The Wast and West Conferences of the NBA are separated for interpretability of the variable to also indicate playoff participation.
num_all_stars **(Integer)**	This is the number of all stars on the team from the previous year at the point of schedule release. For example, when a team acquires an all-star player from the previous year that was

	not on their roster, they would be accounted for in this variable.
guardians_game **(Binary)**	This is true if there is an with the Cleveland Guardians game scheduled on the same day as a Cavs game.
after_all_star **(Binary)**	This is true if the game occurred before or after the NBA All-Star break that typically occurs in the middle of February.
late_night_game **(Binary)**	This is true if the tip-off of the game was at 7:00PM ET or later
is_weekend **(Binary)**	This is true if the game occurred on a Friday, Saturday, or Sunday
Lebron **(Binary)**	This is true if the former Cleveland Cavalier, Lebron James, is on the opposing team (currently on the Los Angelos Lakers)
show_rate **(Numeric)**	This represents the percentage of ticket purchasers who show up to a game. Also referred to as scans.

Appendix C: List of Included Attributes for Genetic Algorithm

Perceived Value (Factor)	This is the relative group the event falls in according to business logic for determining the demand and quality of matchup for fan appeal.
Weekend (Binary)	This is true if the game is on a weekend or not. Defined as Friday, Saturday, or Sunday.
day number (Integer)	This is how many days past the opening night the game occurs, ranging from opening night until the final home game.
In Conference (Binary	This is true if the game occurred on a Friday, Saturday, or Sunday

References

Al-Ashhab, M. S., & Alghamdi, A. (2017). Two-stage multi-objective university courses timetabling using genetic algorithms. International Journal of Engineering & Technology, 7(4.30), 30-37. https://doi.org/10.14419/ijet.v7i4.30.22008

Albadr, Musatafa & Tiun, Sabrina & Ayob, Masri & Al-Dhief, Fahad. (2020). Genetic Algorithm Based on Natural Selection Theory for Optimization Problems. Symmetry. 12. 1-31. 10.3390/sym12111758.

Altiparmak, F., Gen, M., Lin, L., & Paksoy, T. (2006). A genetic algorithm approach for multi-objective optimization of supply chain networks. Computers & Industrial Engineering, 51(1), 196-215. https://doi.org/10.1016/j.cie.2006.07.011

Arslan, H. A., Easley, R. F., Wang, R., & Yilmaz, O. (2021). Data-driven sports ticket pricing for multiple sales channels with heterogeneous customers. Manufacturing & Service Operations Management, 23(6), 1371-1389. https://doi.org/10.1287/msom.2021.1005

Banciu, M., Hinterhuber, A., & Ødegaard, F. (2023). Revenue management in sports, live entertainment and arts. Journal of Revenue and Pricing Management, 22(3), 185-187. https://doi.org/10.1057/s41272-023-00432-y

Bhatnagar, R., & Babbar, M. (2022). A systematic review of sports analytics. International Journal of Technology Transfer and Commercialization, 19, 393. https://doi.org/10.1504/IJTTC.2022.127574

Brooks, J., Kerr, M., & Guttag, J. (2016). Developing a data-driven player ranking in soccer using predictive model weights. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 49-55). ACM. https://doi.org/10.1145/2939672.2939695

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems (2nd ed.). Springer.

Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. ACM Computing Surveys, 45(3), 1-33. https://doi.org/10.1145/2480741.2480752

Ding, C., Bi, J., & Wang, Y. (2023). A hybrid genetic algorithm based on imitation learning for the airport gate assignment problem. Entropy, 25(4), 565. https://doi.org/10.3390/e25040565

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory.

Eiben, A. E., & Smith, J. E. (2015). Introduction to evolutionary computing (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-44874-8

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, & U. Fayyad (Eds.), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96) (pp. 226-231). AAAI Press.

Forbes. (2023, October 26). NBA valuations 2023. https://www.forbes.com/lists/nba-valuations/

Gendreau, M., & Potvin, J. Y. (2005). Metaheuristics in combinatorial optimization. Annals of Operations Research, 140(1), 189-213. https://doi.org/10.1007/s10479-005-3971-7

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins (Ed.), Foundations of genetic algorithms (Vol. 1, pp. 69-93). Morgan Kaufmann.

Holland, J. H. (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. MIT Press.

Jee, W. F., & Hyun, M. (2023). Blinded by attachment: Examining the overconfidence bias of sports fans' intertemporal ticket purchase decisions. Behavioral Sciences, 13(5), Article 405. https://doi.org/10.3390/bs13050405

Jozefowiez, N., Semet, F., & Talbi, E. G. (2008). Multi-objective vehicle routing problems. European Journal of Operational Research, 189(2), 293-309. https://doi.org/10.1016/j.ejor.2007.05.055

Kaufman, L., & Rousseeuw, P. J. (1990). Finding groups in data: An introduction to cluster analysis. John Wiley & Sons. https://doi.org/10.2307/2532178

Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220 (4598), 671-680. https://doi.org/10.1126/science.220.4598.671

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, pp. 281-297). University of California Press.

Mahdavi, S., Shiri, M. E., & Rahnamayan, S. (2015). Metaheuristics in large-scale global continues optimization: A survey. Information Sciences, 295, 407-428. https://doi.org/10.1016/j.ins.2014.10.042

Maloney, J. (2019). gapstat (Version 1.0) [Computer software]. GitHub. https://github.com/jmmaloney3/gapstat

Marquez, A. A. (2020). The effects of partitioned pricing on event ticket purchasers [Doctoral dissertation, Georgia State University]. ScholarWorks @ Georgia State University. https://doi.org/10.57709/17623952

Metaxiotis, K., & Liagkouras, K. (2012). Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. Expert Systems with Applications, 39(14), 11685-11698. https://doi.org/10.1016/j.eswa.2012.04.053

Mitchell, M. (1998). An introduction to genetic algorithms. MIT Press.

Molina, D., Poyatos, J., Del Ser, J., García, S., Hussain, A., & Herrera, F. (2020). Comprehensive taxonomies of nature- and bio-inspired optimization: Inspiration versus algorithmic behavior, critical analysis and recommendations. Cognitive Computation, 12(5), 897-939. https://doi.org/10.1007/s12559-020-09730-8

Neptune.ai. (2023, December 19). Implementing customer segmentation using machine learning. https://neptune.ai/blog/customer-segmentation-using-machine-learning

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications, 36(2), 3336-3341. https://doi.org/10.1016/j.eswa.2008.01.039

Punj, G., & Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. Journal of Marketing Research, 20(2), 134-148. https://doi.org/10.1177/002224378302000204

Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. Natural Computing. https://doi.org/10.1007/s11047-023-09950-3

Ravelin Technology. (n.d.). Machine learning for fraud detection. https://www.ravelin.com/insights/machine-learning-for-fraud-detection

Reese, J. D., & Bennett, G. (2013). Satisfaction with the season ticket sales process. Journal of Contemporary Athletics, 7(2), 103-117.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. PloS One, 14(1), Article e0210236. https://doi.org/10.1371/journal.pone.0210236

Sacha, D., Stein, M., Schreck, T., Keim, D. A., & Deussen, O. (2014). Feature-driven visual analytics of soccer data. In 2014 IEEE Conference on Visual Analytics Science and Technology (VAST) (pp. 13-22). IEEE. https://doi.org/10.1109/VAST.2014.7042477

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN.... In Proceedings of the Sixth International Symposium on Micro Machine and Human Science (pp. 39-43). IEEE. https://doi.org/10.1109/MHS.1995.494215

SCW.AI. (2023, November 28). Predictive maintenance with machine learning in 2024. https://scw.ai/blog/predictive-maintenance-with-machine-learning/

Singh, N. (2020). Sport analytics:...

Solanellas, F., Muñoz, J., & Petchamé, J. (2022). An examination of ticket pricing in a multidisciplinary sports mega-event. Economies, 10(12), Article 322. https://doi.org/10.3390/economies10120322

Sörensen, K., Sevaux, M., & Glover, F. (2018). A history of metaheuristics. In R. Martí, P. M. Pardalos, & M. G. C. Resende (Eds.), Handbook of heuristics (pp. 791-808). Springer. https://doi.org/10.1007/978-3-319-07153-4_4-1

Steinhaus, H. (1956). Sur la division des corps matériels en parties [On the division of material bodies into parts]. Bulletin de l'Académie Polonaise des Sciences, Classe III, 4(12), 801-804.

Talbi, E. G. (2009). Metaheuristics: From design to implementation. John Wiley & Sons.

Thorndike, R. L. (1953). Who belongs in the family? Psychometrika, 18(4), 267-276. https://doi.org/10.1007/BF02289263

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), 411-423. https://doi.org/10.1111/1467-9868.00293

Tzanetos, A., & Dounias, G. (2021). Nature inspired optimization algorithms or simply variations of metaheuristics? Artificial Intelligence Review, 54(3), 1841-1862. https://doi.org/10.1007/s10462-020-09893-8

Valova, I., Embry, A., Trudeau, M., & Gueorguiev, G. (2014). Evolving vacation packages: Genetic algorithms for entertainment. Procedia Computer Science, 36, 312-320. https://doi.org/10.1016/j.procs.2014.09.096

Wang, X., & Song, X. (2023). Optimal path planning of logistics distribution of urban and rural agricultural products from the perspective of supply chain. Informatica, 47(5). https://doi.org/10.31449/inf.v47i5.4557

Xiong, C., & Xu, Y. (2021). Research on logistics distribution path planning based on fish swarm algorithm. Journal of Physics: Conference Series, 1883(1), 012040. https://doi.org/10.1088/1742-6596/1883/1/012040

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of Data Science, 2(2), 165-193. https://doi.org/10.1007/s40745-015-0040-1

Yang, X. S. (2020). Nature-inspired optimization algorithms (2nd ed.). Academic Press. https://doi.org/10.1016/B978-0-12-821986-7.00001-5

Zeithaml, V. A., Rust, R. T., & Lemon, K. N. (2001). The customer pyramid: Creating and serving profitable customers. California Management Review, 43(4), 118-142. https://doi.org/10.2307/41166104