Midway project report

Utsav Lal
ualal@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Swaraj Kaondal
skaonda@ncsu.edu
North Carolina State University
Raleigh, NC, USA

ABSTRACT

We suggest a method for forecasting the outcome of basketball matches between two teams composed of randomly selected players, employing machine learning algorithms. Our plan involves leveraging the CMU NBA dataset along with match-by-match NBA data obtained from Kaggle. Through analyzing this data, our objective is to uncover compelling statistics and ultimately present our discoveries in a comprehensive final project report by the semester's end.

1 INTRODUCTION AND BACKGROUND

Basketball, a team-based game typically featuring five players per team, involves competing on a rectangular court. The main aim is to shoot a basketball through the opponent's hoop, which is approximately 9.4 inches (24 cm) in diameter, mounted 10 feet (3.048 m) high on a backboard at each end of the court. The opposing team tries to prevent this while aiming to score through their own hoop. It's a sport deeply ingrained in American culture, with extensive historical data available for analysis. In the era of data analytics, there has been significant focus on uncovering hidden patterns within this data to enhance player performance. This project aims to leverage historical player statistics to forecast the outcome of basketball matches.

1.1 Problem Statement

As stated earlier, bleeding edge techniques in data analysis has brought forward new ways of finding interesting patterns within data. National Basketball Association (NBA) hosts one of the oldest basketball competitions across the USA. Due to its historic nature a huge amount of player data has been collected in the process. With this project we aim to analyze this available data of player statistics and ultimately design a machine learning algorithm that would eventually predict a winner between two teams. At the midway stage, we clean the available dataset and transform the dataset for training. We then train a Naive Bayes model which sets the baseline for future neural network implementation.

1.2 Prior Research on topic

In [1] the authors used artificial neural networks to predict the eventual champion of an NBA season. They come up with a neural network with 3 hidden layers and use batch based training with a batch size of 32 and total of 50 epochs. They came up with 2 models for their use case which performed extremely well with very small overall loss. As a final step they predicted the champion of the 2022-2023 season. Their model was able to correctly predict the winner of the season. Another similar research is that of [3] where the authors tried to predict Soccer World Cup 2018 matches using Radial-basis neural networks. They achieved a correct percentage of win and loss of 83.7% and 72.7%. The authors used a 3 layer neural network with radial basis functions as their activation functions to achieve excellent results. They were also able to find out interesting tactics which led teams to win soccer games. Another interesting research is of [6] where a decision tree classifier was trained on football matches in the CFASL league. The authors have provided an algorithm for training the decision tree and built a model using the said algorithm. They are able to achieve a 57.7% accuracy in correctly predicting a match winner.

2 METHOD

2.1 Introduction

In the project, we want to propose a artificial neural network based machine learning technique to predict the winner of a match given players on each side. To measure the success of our proposed neural network architecture we also design a baseline Naive Bayes model for match predication. Naive Bayes is a classification algorithm which uses Bayes theorem [4] to calculate the probability of a match winner. Due to its simplicity it is a great machine learning algorithm tool for comparison to our actual proposed solution which will be based on an artificial neural network.

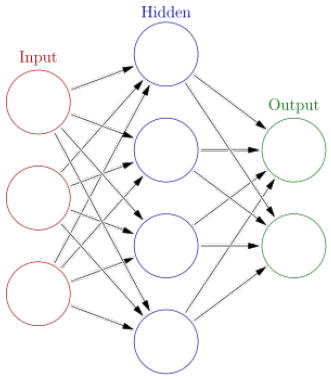

Figure 1: An artificial Neural network

2.2 Why Artificial Neural Network?

Our problem statement revolves around forecasting match winner. More specifically this translates into a classification problem. The first algorithm which we thought of was logistic regression since our problem has a binary relation. But logistic regression doesn't deal well with complicated dataset like ours. Next came binary trees, but again since we have a lot of features in our dataset it decision trees might overfit and become overly complicated to correctly make a prediction. K-Nearest neighbors doesn't work well with

large datasets. Hence in the end we decided to use Artificial neural networks (ANN) [7].

In recent years ANNs has emerged as one of the best techniques for solving classification problems. One of the main reasons for selecting neural networks for solving our problem is because of its success in solving similar problems as stated in the prior research section. As we will see later in the paper, Naive Bayes fails to capture the non linearity of the data. ANNs with its architecture of hidden layers becomes a great tool for capturing the non linearity within the data. It can also capture hidden features within the data which might not be obvious from just looking at the data. For these reasons ANNs emerge as an excellent tool for solving our problem.

2.3 Approach

(1) Naive Bayes: For Naive Bayes, the approach will be simple. For Naive Bayes we assume that each of the feature within the dataset is conditionally independent of each other given we have an outcome of the match. It might not be true for our dataset but it helps us create a baseline model. In our case since the player attributes are continues we will use the Guassian Naive Bayes algorithm. Probability of a feature given y will be defined as:

$$P(x_i/y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} * e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}} \quad (1)$$

And the probability of an outcome is given by

$$P(y/x_1x_2...x_n) = \arg\max P(Y=y) \prod P(x_i/Y=y]) \quad (2)$$

With this we can predict the winner of a match with players of given statistics. Each stat in this case will be a feature.

(2) Artificial Neural Network: An artificial neural network at its core has two modes of operations forward propagation and backward propagation.
   - During forward propagation, input data is passed through the network, layer by layer, from the input layer to the output layer. Each neuron receives inputs from the previous layer, applies a weighted sum operation, adds a bias term, and passes the result through an activation function to produce an output. This process continues until the output layer produces the final prediction or output.
   - Backpropagation is the process by which the network learns from the errors in its predictions. It involves calculating the gradients of the loss function with respect to the network's weights and biases. These gradients indicate how much each weight and bias contributed to the error, allowing the network to adjust its parameters to minimize the error. This process is typically performed using optimization algorithms such as gradient descent.

   A major aspect of neural network training is adjusting hyper parameters like number of hidden layers, number of neurons in the layer, choice of activation function, learning rate, etc. While we have not yet decided the intricate details of our neural network the basic structure will be as discussed above.

3 PLAN AND EXPERIMENT

3.1 Datasets

We have access to data spanning from 1946 to 2004 for both the NBA and ABA basketball leagues. This dataset encompasses various attributes of players across three different types of basketball matches within these leagues, which are as follows:

- Regular Season: The regular season in basketball consists of a series of scheduled games where teams compete to accumulate wins and secure a spot in the playoffs.
- Playoffs: The playoffs are a series of elimination games following the regular season, where the top teams from each conference compete in a tournament-style format to determine the league champion.
- All-Star Game: The All-Star game is an exhibition match held midway through the season, featuring the league's top players as selected by fan, player, and media voting, providing a showcase of talent and entertainment for fans.

3.2 Data Source and Description

3.2.1 NBA Statistics data, Carnegie Mellon University. [2]

(1) Players table - The master list of all players.
(2) Player regular season table: The stats of each player for each season.
(3) Player regular season career table: Cumulative stats of each player during their career.
(4) Player playoffs table: The stats of players who have played in playoffs
(5) Player playoffs career table: The cumulative stats of players who have played in playoffs.
(6) Player allstar table: The stats of players for all star matches
(7) Teams table: Master list of all teams in NBA and ABA
(8) Team season table: The stats of whole team over a season for each season
(9) Draft table: The list of players drafted
(10) Coaches season table: The stats of coaches for a season for all season
(11) Coaches career table: The stats of coaches aggregated for all season in their career.

3.2.2 NBA Dataset, Kaggle. [5] The dataset includes information on individual games spanning from 1946 to 2023. The data is organized at the team level and includes attributes such as attempted and made goals, assists, steals, and blocks. Each player, team, and coach is assigned a unique ID that is shared across all entities.

4 HYPOTHESIS

A team with higher sum of normalized statistics should always win the match regardless of other external factors. We want to find out with our proposed solution if this hypothesis always holds true or not. To verify this hypothesis we will train an artificial neural network on the dataset and ask the model to predict the winner for matches which it had not seen during training.

5 EXPERIMENTS

5.1 Data Pre-processing

Due to the lack of match-specific details in the NBA Statistics data, we aim to merge the Kaggle dataset, which includes match outcomes, with CMU's dataset, containing player statistics for those matches. The objective is to develop a model capable of establishing a relationship between a set of player statistics and the likelihood of winning by training on historical match data. Initially, the primary challenge lies in preprocessing the data to ensure its compatibility with the model. This includes aligning the Kaggle dataset with CMU's dataset, as they have varying naming conventions and attributes. Ensuring data authenticity involves verifying values from both datasets through cross-referencing for validation purposes.

Our objective is to predict the winner of a match given a list of players and their attributes. To train a model for such predictions, we require the attributes of each player who played a match and the outcome of that match. Unfortunately, we couldn't find a data source that could provide such data directly.

We explored the CMU dataset, which has all the attributes of players for each season but lacks match-specific data. On the other hand, the Kaggle dataset includes the list of players who played a match but lacks player attribute data. Thus, our first task was to map the player IDs from the Kaggle dataset to the CMU dataset.

5.1.1 Player Roster Extraction For Each Match. Although the Kaggle dataset contains match-specific data, it is not directly available. Kaggle's play-by-play data indicates each point scored, pass, block, or steal and the player who made the play. We parsed this data to identify the list of players for each team in a specific game and consolidated it into one table. However, even after this consolidation, one piece of information was missing: the winner of the match. To determine the winner, we joined the new table with another table from the Kaggle data that provided this information.

5.1.2 Player ID Mapping: Linking CMU and Kaggle Datasets. Our next task was to map the player IDs in the CMU data to the Kaggle data. This mapping was necessary because Kaggle had game-specific data, including the list of players present in the game but not their attributes, while the CMU data had player attributes but not the list of players present in a specific game. To perform this mapping, we joined the two-player datasets based on their first name, last name, and birth year. Initially, there were 3572 unique players in the CMU data and 4171 unique players in the Kaggle dataset. After the join, we were left with 2717 unique players, indicating that only these players had matching first names, last names, and birth years in both datasets. This suggests some discrepancies in the names of players or the absence of player data in either of the datasets.

5.1.3 Merging Game Data with Player Information. Using the game data created in step 1 and the player mapping created in step 2, we were able to map the game data with the players present in each game using the Kaggle player IDs. Before this mapping, we had data for around 20,000 games. After the mapping, the number reduced to around 8,000 games because we were not able to map the IDs of many players.

5.1.4 Player Statistic Normalization and Averaging by Season. The player statistics data provides the overall performance of the player for each attribute. For example, if a player has played 80 games in a season, then the dataset has the sum of all the points scored by that player in those matches. Using this raw data for predictions could disadvantage players who have played fewer games, as their overall scores for points scored, blocks, and steals would be lower compared to those who have played more games. To address this, we normalized the data by dividing each attribute by the number of games played. For predicting the winner of a match given the player attributes, we needed to determine which season's attributes to consider. We had two options: consider the attributes of a player from the last season or take the average of the attributes from all the previous seasons. We chose to take the average of attributes from all the previous seasons and calculated this for every season.

5.1.5 Calculating team statistics. With the normalized and averaged attributes of all previous seasons for all players and the game data containing the list of players in each game and the game-winner, we merged this data and took the average of each player's attribute to get the attributes of the team. This data now included the attributes of two teams and the winner of each game, making it ready for training and predictions.

5.2 Feature Selection

As each table contains a wide range of attributes related to teams and players, a crucial step is to choose appropriate features for training the model. To accomplish this, we have employed Pearson Correlation. The correlation of each feature with every other feature and with the target variable is depicted in Figure 2. The heat map reveals that there is no linear correlation between the target variable "winner" and the features.

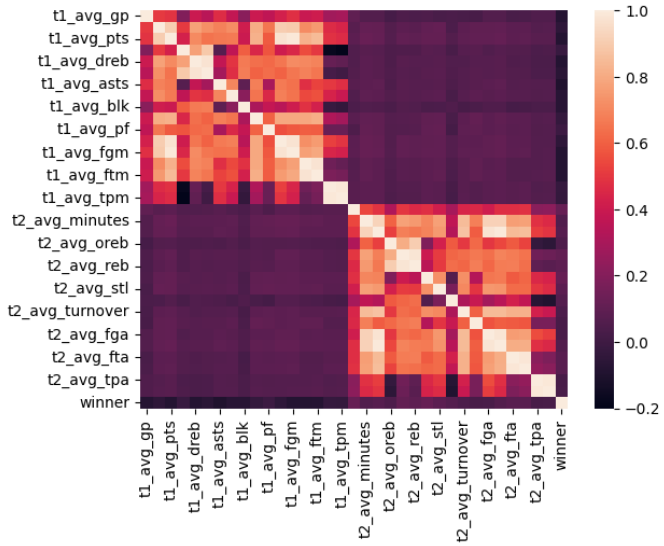

Figure 2: Feature Correlation

The correlation matrix suggests that the previously stated hypothesis may not hold true, as there appears to be minimal linear

correlation between player statistics and match outcomes. However, artificial neural networks offer the potential to uncover nonlinear relationships between features. By employing ANNs in our analysis, we aim to gain a clearer understanding of the hypothesis and its validity.

5.3 Modelling

As per our proposal, we initially adopted Naive Bayes as the baseline model. However, the analysis of feature selection indicates no linear correlation between the features and the target variable. Consequently, we anticipate that the performance of the Naive Bayes model may be sub-optimal. To address this, we intend to utilize a neural network to capture any complex relationships that exist between the features.

For training the Naive Bayes model, we have data from 8777 games, with each game having 34 continuous features (17 for each team) and a discrete target variable that indicates the winning team with a binary value.

5.4 Results

We trained a Gaussian Naive Bayes model using 80% of matches as training data. We used the remaining 20% data for our analysis of the classifier. As anticipated, the model's performance was poor, achieving an accuracy of approximately 56%, which is comparable to random guessing.

Table 1: Confusion Matrix

	Predicted	
Actual	**Team 1 Win**	**Team 2 Win**
Team 1 Win	549	401
Team 2 Win	380	426

Table 2: Classification Report

	Precision	Recall	F1-Score	Support
Team 1 Win	0.59	0.58	0.58	950
Team 2 Win	0.52	0.53	0.52	806
Accuracy			0.56	1756
Macro Avg	0.55	0.55	0.55	1756
Weighted Avg	0.56	0.56	0.56	1756

Due to the many number of features in our dataset, Naive Bayes as predicted earlier performs pretty poorly for our data. But this gives us a good baseline model to built upon. Further work which is required going ahead in the project is described in the following section.

6 PROPOSED WORK

We are going to perform the following tasks in the future to create a model with performance metrics.

6.1 Design of future experiments

- Review the process of merging player IDs from the two datasets to minimize data loss. During the merging of player data with game data, approximately 12,000 game records had to be omitted due to an inability to map the players involved in those games.
- Reconsider the approach for predicting match outcomes. Instead of averaging a player's performance across all previous seasons, it may be more appropriate to focus solely on their performance in the last season.
- In NBA games, only 5 players are on the court at a time, with around 10 additional players on the bench who can substitute. We need to determine whether to consider the attributes of all 15 players to calculate team performance or only focus on the top 5 players or those who played the match for the longest duration. Experimenting with different player selection methodologies and evaluating their impact on model training performance is necessary.
- The feature correlation heat-map analysis indicates no linear correlation between the target variable and the features. Therefore, implementing a Neural Network is proposed to capture the complex relationships between the features and the target variables.
- Using the trained model to verify the hypothesis stated above. We will use data previously unseen by the model and compare actual results with the predicted results. With this we can prove or disprove the hypothesis.

6.2 Plan of activities

The project workload will be distributed as follows.

6.2.1 *Tasks for Swaraj.*

- Try out different strategies for calculating the player attributes for a match by looking up the player performance in the past seasons.
- Try out different strategies to aggregate the player performances to calculate the performance of the entire team.

6.2.2 *Tasks for Utsav.*

- Validate the data pre-processing code to check if all the joins are correct and is there any data loss due to missing entries or duplicate rows due to the absence of one to one mapping.
- Design and train neural network after validating and implying different techniques of aggregating the data.

6.2.3 *Future Meeting schedules.*

(1) 04-03-2024, Wednesday, 1:15pm to 2:00pm
(2) 04-05-2024, Friday, 1:15pm to 2:00pm
(3) 04-10-2024, Wednesday, 1:15pm to 2:00pm
(4) 04-12-2024, Friday, 1:15pm to 2:00pm

REFERENCES

[1] Bruin Sports Analytics. [n. d.]. Decoding the Game: Forecasting NBA Champions with Neural Network Algorithms — bruinsportsanalytics.com. https://www.bruinsportsanalytics.com/post/nba-champs-neural-network. [Accessed 03-03-2024].
[2] School of Computer Science Carnegie Mellon University. [n. d.]. *NBA statistics data.* http://www.cs.cmu.edu/~awm/10701/project/databasebasketball2.0.zip

[3] Amr Hassan, Abdel-Rahman Akl, Ibrahim Hassan, and Caroline Sunderland. 2020. Predicting wins, losses and attributes' sensitivities in the soccer world cup 2018 using neural network analysis. *Sensors* 20, 11 (2020), 3213.
[4] James Joyce. 2003. Bayes' theorem. (2003).
[5] Kaggle. [n. d.]. *NBA dataset.* https://www.kaggle.com/datasets/nathanlauga/nba-games
[6] Xiaohu Tang, Zhifeng Liu, Taizhao Li, Wenbin Wu, and Zhenhua Wei. 2018. The application of decision tree in the prediction of winning team. In *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*. IEEE, 239–242.
[7] Bayya Yegnanarayana. 2009. *Artificial neural networks.* PHI Learning Pvt. Ltd.