Project: NBA Line Up Problem

*ENGN 5775G:Knowledge Discovery and Data Mining

Sweta Patel
Masters in Engineering Management
100915164
sweta.patel5@ontariotechu.net

Vallika Kasibhatla
Masters in Engineering Management
100928820
vallika.kasibhatla@ontariotechu.net

***Abstract*—In today's world, filled with emerging technologies and easy access to data, machine learning has become evident even in sports. Sports winning prediction has become crucial for understanding the team and providing valuable feedback to improve the team's strategies. Therefore, effectively and accurately predicting a winning team is vital. However, this approach isn't straightforward, and for this reason, there has been an increase in the utilization of machine learning to create patterns within a dataset. In this project, various machine learning models, including Neural Networks, K-nearest Neighbours, Logistic Regression, Random Forest and Support Vector Machines, were examined and assessed through a comparative study. Ultimately, employing evaluation metrics, the model of Random Forest stood out as the optimal model for a lineup prediction.**

***Index Terms*—NBA, Machine Learning, Classification, Logistic Regression, K-nearest Neighbours, Random Forest, Neural Networks, Support Vector Machines**

I. Introduction

In sports, making accurate predictions is a powerful and valuable tool for any affected parties. Most importantly, only with information on a team's lineup, predicting if they would win or lose is crucial. Predicting a game's outcome and finding a winning lineup of players before the game is an exciting problem in Machine Learning. The main information that could be given is historical data, the players' performance, team performance, points made, and history, including past performances, current form, injury history, etc. Such insights could be valuable to sports analysts, coaches, and fans wishing to bet on the game's outcome. This problem focuses on how we rank the players and how each player impacts another to find a good team. This combination and permutation is a tricky process, and the history of the players is personalized for each game. The best approach to finding patterns between names and outcomes is through Machine Learning models; this task could be efficiently and accurately done. This paper focuses on NBA games. Here, the players' names and their team performance are given, and the aim is to predict their chances of winning. Machine learning can help create better connections in the dataset. ML is a branch of artificial intelligence focused on creating algorithms that learn and make predictions based on the various information given. The main task is to find patterns between extensive data and get statistical insights to learn from the data, consequently improving the predictive power. This paper focuses on many classification models, including neural networks, to create patterns of players in an NBA game and predict if they will win or lose. The Problem Statements addressed in this paper would be:
1) Predict the game's outcome when provided with five home team players' names and five away team players' names.
2) When four away team players and five home team players are given, predict the fifth player in each lineup to enhance the likelihood of the away team winning.

II. Literature Review

Predicting the players in a lineup is useful for sports. There are various games where understanding the importance of the teams and players is needed for lineup problems—a few examples of games that focus on predicting game outcomes based on the players' understanding and level. The League of Legends (LoL) [1] game is a multiplayer game with competitive teams, requiring players to be assigned randomly between them. This lineup is predicted based on the player's experience and must be fair [1]. Being fair in matchmaking is crucial in any game layout [1]. The features used were "players win rate, total number of games players and recent games players." The models used were Support Vector classifier, KNN, Random Forest, Gradient Boosting and Deep Neural Network. The accuracy of SVC is 74.3%, KNN has an accuracy of 72.7%, RF has an accuracy of 74.7%, and Gboost has an accuracy of 75.4%. [1]. The DNN machine learning model was a unique model used, resulting in an accuracy of 75%. The network topology for DNN is a flattened input layer, 15 layers of normalization, dropout, and dense layers; towards the end, a sigmoid activation dense layer was used. The limitation was that it only focused on the North American domain. A lineup prediction problem is also useful in football games for coaches to guarantee success [2]. A study was conducted on the "player's physiological state" so that the start team is successful. The feature selection was based on the recursive feature elimination algorithm, and the model used was logistic regression. A similar approach was used to predict the outcome of cricket games [3] using machine learning. The models used are random forests and decision trees to classify their data. Through these studies, this paper will focus on models such as Logistic Regression, K-nearest Neighbours, Random Forests, Neural Networks, and Support Vector Machines.

III. Dataset

The NBA Dataset used in this project consists of 197718 states of various games from years 2007 to 2012. The data is split based on game ID, the season of the game, the states in each game and the teams playing. The states are denoted with a start and end minutes. The teams are based on home vs away teams. The unique home team consists of Phoenix Suns (PHO), Los Angeles Clippers (LAC), Utah Jazz (UTA), Dallas Mavericks (DAL), Memphis Grizzlies (MEM), Philadelphia 76ers (PHI), New Orleans Pelicans (NOK), Brooklyn Nets (NJN), Houston Rockets (HOU), Milwaukee Bucks (MIL), Sacramento Kings (SAC), Golden State Warriors (GSW), Toronto Raptors (TOR), Washington Wizards (WAS), Portland Trail Blazers (POR), New York Knicks (NYK), Miami Heat (MIA), Seattle SuperSonics (SEA), Cleveland Cavaliers (CLE), Orlando Magic (ORL), Minnesota Timberwolves (MIN), San Antonio Spurs (SAS), Atlanta Hawks (ATL), Chicago Bulls (CHI), Boston Celtics (BOS), Indiana Pacers (IND), Los Angeles Lakers (LAL), Charlotte Hornets (CHA), Detroit Pistons (DET), Denver Nuggets (DEN), New Orleans Hornets (NOH), Oklahoma City Thunder (OKC). There are some NBA statistics for each state that represent the team's performance. These are numeric values. Additionally, they provided the names of all the team players for each state. There are 813 unique players in the dataset from all the years. This is the main information that will be used to solve the problems. The final column denotes the target prediction, with "1" indicating that the home team is in the lead, and "-1" indicates that the visitor team is ahead of the home team.

A. Understanding Variables

- The variables "Game", and "Season" work as IDs for the team.
- "home_team" and "away_team" are the team names that are going against each other.
- Each game is in terms of states. To indicate the duration of the states, the start and the end time were mentioned. "starting_min" and "end_min."
- There are 197718 states
- 2007-2012 seasons
- Around 813 unique players
- There are 31 unique teams. Short names: PHO, LAC, UTA, DAL, MEM, PHI, NOK, NJN, HOU, MIL, SAC, GSW, TOR, WAS, POR, NYK, MIA, SEA, CLE, ORL, MIN, SAS, ATL, CHI, BOS, IND, LAL, CHA, DET, DEN, NOH, OKC.
- Numeric Data for the team Statistics
- There are five players from the home and away team. Each name is in one cell. The 10 players were denoted with home_0, home_1, home_2, home_3, home_4, away_0, away_1, away_2, away_3, away_4.
- The rest of the variables denote numeric values representing the performance of the teams in terms of the points scored,
- Home Team= fga_home, fta_home, fgm_home, fga_2_home, fgm_2_home, fga_3_home, fgm_3_home, ast_home, blk_home, pf_home, reb_home, dreb_home, oreb_home, to_home, pts_home, pct_home, pct_2_home, pct_3_home
- Away_Team=fga_visitor,fta_visitor, fgm_visitor,fga_2_visitor, fgm_2_visitor,fga_3_visitor, fgm_3_visitor,ast_visitor,blk_visitor, pf_visitor,reb_visitor, dreb_visitor,oreb_visitor,to_visitor, pts_visitor,pct_visitor,pct_2_visitor,pct_3_visitor, home_fg_efficiency, visitor_fg_efficiency

B. Target Variable: "Outcome"

This column has binary values (-1, 1). The -1 indicates that the visitor team is ahead of the home team, "1" indicates that the home team is in the lead. Additionally, while understanding the classes (Fig. 1), the "-1," classes have 98859 entries while "0" class has 55673 entries, which is imbalanced.

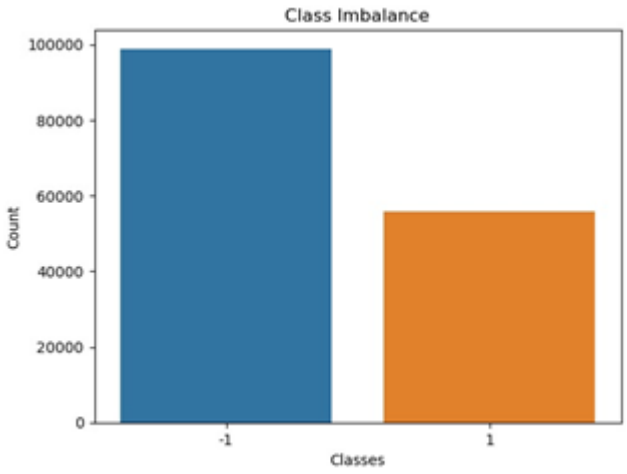

Fig. 1. Class Imbalance

IV. Data Pre-processing Methods

A. Missing value:

Data can go missing due to incomplete data entry, lost files, and many other reasons. Removing rows or columns with null values may be a step to handle missing values. Another way to handle such values is through imputation techniques, such as mode category imputation or replacing unknown values with the most frequent value in the column. The dataset contained no missing values across variables during the analysis. As a result, there was no requirement to address missing values during the analysis, as all variables had complete observations.

B. Scaling and Standardization:

Scaling, also known as data normalization, is used to normalize the independent attributes in the data to help make the attributes closer, irrespective of the units. The main purpose of standardization is to enable all the attributes to contribute equally to any analysis. Another scaling method brings attributes close to a unit standard deviation and a mean. The dataset consists of columns representing numeric scores in

fractions, so scaling is not an issue. Nevertheless, the names of the players were scaled after doing label encoding.

C. Up-Sampling

In machine learning, Up-sampling is a technique used to address the class imbalance in a dataset. Up-sampling involves adding samples from the minority class to balance the class distribution with the majority class. The reason for choosing up-sampling is so that we don't lose any information on the players' names. This is very important as our problem statement focuses on how the players and teams combination. According to the Figure, it is evident that the dataset demonstrates an imbalanced distribution. This disparity in distribution has the potential to introduce bias as it may tend to prioritize the majority class. This would lead to entries of 98859 for both classes (Fig. 2).

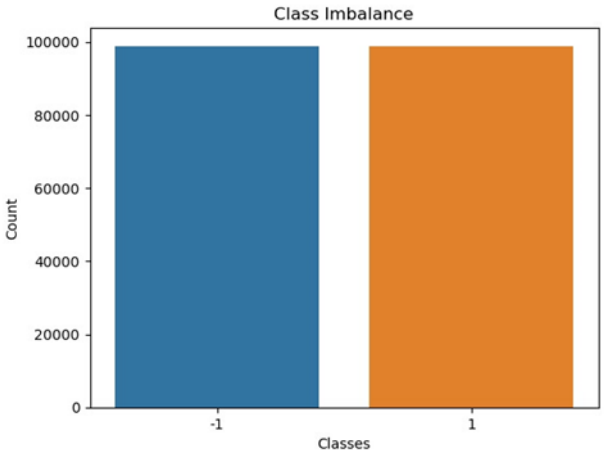

Fig. 2. Class Balanced

V. Feature Selection

The goal of feature selection is to identify the most informative and discriminative features that contribute the most to the model's predictive power while discarding irrelevant or redundant features. This process helps improve model performance, reduce overfitting, and enhance interpretability. In the Figure, the feature importance is ranked using random forest, and the few important features were home and visitors points per game, fg efficiency, and points (Fig. 3).

VI. Feature Transformation

A. Handling Text Data

Players' names are crucial in understanding the team dynamics and overall game strategies. However, since player names are categorical variables and cannot be directly inputted into most machine-learning algorithms, they need to be transformed into numerical representations. One common technique used for this purpose is label encoding. Label encoding is a method used to convert categorical data, such as player names, into a unique number format that machine learning algorithms can process. We have also tried doing One-Hot Encoding;

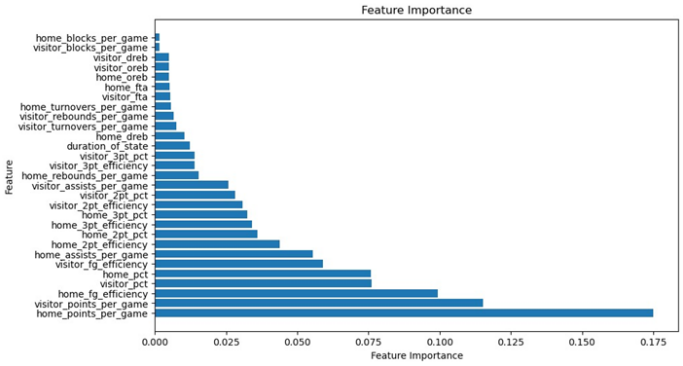

Fig. 3. Figure Importance

however, the information resulted in huge sparsity as there are 813 players. So, the project used label encoding.

home_0	home_1	home_2	home_3	home_4	away_0	away_1	away_2	away_3	away_4
41	482	511	692	715	81	474	634	710	729
41	482	511	692	715	26	494	634	710	729
511	554	673	715	785	81	346	474	494	528
482	554	692	715	785	26	494	634	710	729
41	482	511	554	715	474	494	634	710	729

Fig. 4. Label Encoding

B. Converting the target variable

The initial values of "1" and "-1" for the target variable in the original dataset gave a win-or-lose definition. The change was to "1" and "0" in order to simplify computations. This new representation makes analysis easier to understand; it is a standard way of representing a classified class and makes it easier for models in predictions. These transformations are standard preprocessing steps.

VII. Classification Models

A. Support Vector Machines

It is a popular machine learning used for classification and regression problems. SVM aims to create the best line or decision boundary to separate n-dimensional space into classes. If the data is not linearly separable, there is a function known as the Radial Basis Function (RBF) kernel, which maps the data in a higher dimension.

B. K-nearest neighbor

It is a non-parametric algorithm used to find the k-nearest data points in the training set to a specified test data point and then classify the test data point depending on the class that occurs most frequently among its K-nearest neighbours. In addition, Euclidean distance in (1) and Cosine similarity (2)

are used to calculate the distance between the test data point and each training data point.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1}$$

$$\textit{cosine\_similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|} \tag{2}$$

C. Random Forest

This ensemble learning technique builds various decision trees with the test data. It produces the mean prediction for regression or the mode of the classes for classification problems for each individual tree. Combining the strengths and essential features of several decision trees helps in generalization and improving accuracy.

D. Logistic Regression

This is a statistical technique for binary classification. Logistic regression predicts the likelihood of a binary result with regard to the features and predictor variables. It estimates coefficients to characterize the connection between the independent variables and the likelihood of the result.

E. Gradient Boosting

The effective and scalable machine learning technique known as XGBoost (Extreme Gradient Boosting) uses the gradient boosting framework. Iteratively, it creates a series of decision trees, with each tree fixing the mistakes of the preceding one. Additionally, handling large datasets, quick results, and better performance are all attributes of Gradient Boosting.

F. Neural Networks

A class of machine learning techniques called neural networks is modelled after the composition and operations of the human brain. They are made up of layers of networked nodes or neurons. Every neuron modifies its input before sending the outcome to neurons in the layer above. Neural networks are frequently employed for tasks like image recognition, regression, and classification because of their ability to discover intricate patterns in data.

VIII. Evaluation

A. Cross Validation

In this project, Cross-validation was used. It is a method in machine learning whereby the dataset is divided into a training set and a validation set. This method involves repeating the process a couple of times while taking different partitions to understand the model's overall performance on previously unseen data.

B. Metrics

These metrics can be employed to gauge the performance of the various models in our project. In the initial phase, the data is divided into training data (80%) and testing data (20%). The models are constructed using the training data and subsequently assessed using the testing data.

1) Confusion matrix: It is a tool used to evaluate the performance of a classification model with two values, 1 being positive and the other being negative. Evaluation metrics such as accuracy, recall, precision, and F1-score can be calculated through Fig 5.

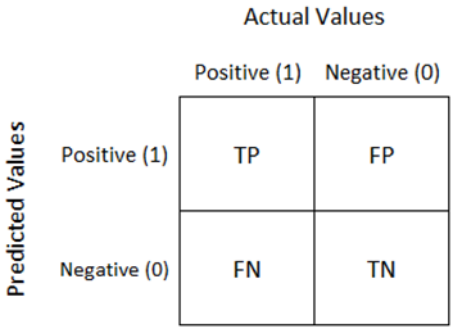

Fig. 5. Confusion Matrix

2) Accuracy: Equation (3) evaluates the number or percentage of correct predictions the built model makes.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

3) Recall: Equation (4) shows the fraction of data points correctly identified as belonging to the positive class over all those positive.

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

4) Precision: Equation (5) shows the fraction of data points that were correctly identified as belonging to the positive class over all those predicted as positive.

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

5) F1-Score: This evaluation metric considers the model's recall and precision, as shown in (6).

$$F1\text{-}Score = 2 * \frac{FPR*Recall}{Precision+Recall} \tag{6}$$

6) Receiver Operating Characteristics (ROC)/ Area under the curve (AUC): The ROC curve is a graphical representation of a binary classifier system's performance. It illustrates the trade-off between the true positive rate (TPR), which has the formula (3), and the false positive rate (FPR), which has the formula (4), at various thresholds. By calculating the area under the curve (AUC), we can gain insight into the model's performance for binary classification problems.

$$FPR = \frac{FP}{FP+TN} \tag{7}$$

IX. Results and Discussion

A. Classification Model using Label Encoding and a few features

In the model selection process, various classification methods were tested. There were four main algorithms tested. The

Model	Hyper parameters	Input	Accuracy	Issues
SVM	Kernel-linear	Label Encoding and Standardization	51%	More than simply encoding the players, talking about their performance with internal team members and how players play is necessary because there needs to be a differentiation factor. This can be the reason for low accuracy. It is difficult to define the decision boundary.
Logistic Regression	Max Iteration=1000		51%	
Neural Network	Dense Layer = Relu Sigmoid layer = last layer Learning Rate = 0.001 Loss=Binary-Cross entropy		52%	
Gradient Boosting	Learning Rate = 0.1		57%	

Fig. 6. Model Architecture for the embeddings

main input was only the label encoded values for the names (10 players) and being standardized. First, we tried the Support Vector Machine with the kernel to be linear, and it resulted in a 51% accuracy. For Logistic Regression, having the max iterations as 1000 also resulted in a classification around 51%. By trying Neural Networks, with a dense layer of relu, learning rate of 0.001, it did a bit better, giving us a 52% accuracy. However, while trying out Gradient Boosting. with a learning rate of 0.1, it got us the highest accuracy of 57% The following table represents the number of models used and the respective accuracy (Fig.6).

B. Further analysis

1) Statistics: Based on the first approach, the main issue was that we didn't have specific significance for each player. The only information used was the text data (names of the players), which were encoded and used as input. Some calculations can be made to analyze the importance of each player as an individual and not just as a team. The approach taken was feature engineering of certain statistics. Based on the research conducted on how to understand the NBA game and statistics, there were four main factors [4] that will decide if a team wins or loses. They encompass the important statistics of the team's performance. The statistics used were the following:

1. Team's Own Effective Field Goal Percentage: pct_home
2. Team's Own Turnover Percentage: to_home
3. Team's Own Offensive Rebound Percentage: oreb_home
4. Team's Own Free Throw Rate: fta_home
5. Opponent's Effective Field Goal Percentage: pct_visitor
6. Opponent's Turnover Percentage: to_visitor
7. Team's Own Defensive Rebound Percentage: dreb_home (calculated as reb_home - oreb_home)
8. Opponent's Free Throw Rate: fta_visitor

The following are the four-factor formulas which will be used for model building:

- Shooting the ball (Effective Field Goal Efficiency):

$$\text{EFG\%} = \frac{\text{Field Goals Made} + 0.5 \times \text{3P Field Goals Made}}{\text{Field Goal Attempts}}$$

- Taking care of the ball (Turnover Rate:

$$\text{TR\%} = \frac{\text{Turnovers}}{\text{Field Goal Attempts} + 0.44 \times \text{Free Throw Attempts} + \text{Turnovers}}$$

- Offensive Rebounding:

$$\text{OR\%} = \frac{\text{Offensive Rebounds}}{\text{Offensive Rebounds} + \text{Opponent's Defensive Rebounds}}$$

- Getting to the foul line:

$$\text{Free Throw Rate} = \frac{\text{Free Throws Made}}{\text{Field Goals Attempted}}$$

or

$$\text{Free Throw Rate} = \frac{\text{Free Throws Attempted}}{\text{Field Goals Attempted}}$$

2) Embeddings: Embeddings [5] , a common technique in data analysis, can add significant meaning to text data. By identifying patterns between names and statistics, embeddings transform high-dimensional vectors into a more manageable, low-dimensional space. This not only aids in dimension reduction but also captures the semantics of the input data through similarities. In our dataset, for instance, similar combinations of players with statistics will be closer in the embedding space, while the rest are further apart, providing a practical application of embeddings in data analysis. This could uncover some important relationships between the players. This approach has also been observed in this paper [**?**].

The embedding process included using a supervised learning approach as a target variable exists (labelled data). A neural network was constructed; there are three inputs in total. The home and away players' names are the first two layers. The third layer has all the statistics formed by the four factors. The embeddings are then flattened and concatenated with the statistics again, which is then passed into a dense layer. The following is the model architecture. The model gives a 32-dimensional feature vector for each player, which will later serve as an input in our classification model (Fig.8).

C. Classification model with Embedded weights

Previously, the classification model inputs were just the player's Label Encoded values, which were standardized, resulting in the highest accuracy of 57% with Gradient Boosting. However, the main drawback was that the inputs didn't provide

Model	Hyperparameters	Input	Accuracy
SVM	Kernel=" Linear"	Embeddings	59%
Gradient Boosting	Learning Rate=0.001		57%
Neural Network	Dense Layer = Relu Sigmoid layer = last layer Learning Rate = 0.001 Loss = Binary-Cross Entropy		68%
KNN	Euclidean Distance and Cos-Sin K=2 and 3		68%
Random Forest			79%

Fig. 7. Model Architecture for the embeddings

Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 5)	0	-
input_layer_1 (InputLayer)	(None, 5)	0	-
home_embedding (Embedding)	(None, 5, 32)	25,696	input_layer[0][0]
away_embedding (Embedding)	(None, 5, 32)	25,568	input_layer_1[0]…
flatten (Flatten)	(None, 160)	0	home_embedding[0…
flatten_1 (Flatten)	(None, 160)	0	away_embedding[0…
input_layer_2 (InputLayer)	(None, 8)	0	-
concatenate (Concatenate)	(None, 328)	0	flatten[0][0], flatten_1[0][0], input_layer_2[0]…
dense (Dense)	(None, 32)	10,528	concatenate[0][0]
dense_1 (Dense)	(None, 1)	33	dense[0][0]

Fig. 8. Model Architecture for the embeddings

valuable information on each player's performance. Embeddings were created to combat this issue. The result was a 32-dimensional feature vector that shows the significance of the player based on the performance across all seasons and years while considering its relationship with the internal and away team members. The new input for a classification model would be the embeddings. We have used multiple models to test the classification. The SVM model with a simple linear kernel resulted in an accuracy of 59%. With Gradient Boosting, the accuracy is 57%. Neural Network leads to an accuracy of 68% while using Dense Layer as Relu, a sigmoid layer as the last layer, Learning Rate as 0.001 and Loss with Binary-Cross Entropy. While using KNN, with Euclidean distance and k-value as 2 and 3, we got an average accuracy of 68%. The best model is through random forest, with an accuracy of 79%.

X. Discussion and Results

The Random Forest performs best, with a 79% accuracy rate. An evaluation study has been conducted to analyze these results further. The confusion matrix and classification report are as follows: The true positive has 15068 entries, the false positive has 4627 entries, the false negative has 3650 entries, and the true negative has 16199 entries.

This would result in the creation of a classification report with two classes of "0" and "1". The precision for class "0" is 81%, while for class "1" it is 78%. The recall for class "0" is 77%, while for class "1" it is 82%. The f1-score for class "0" is 78%, while for class "1" it is 80%. The accuracy is 79%.(Fig. 10) Based on the result, the model is able to predict the values well enough where there was no bias towards any class.

Additionally, the ROC curve was drawn (Fig. 11). The optimum value of AUC would need to be 1, but based on our ROC curve, we were able to get a value between 0.7 and 0.8, which is a fair classifier.

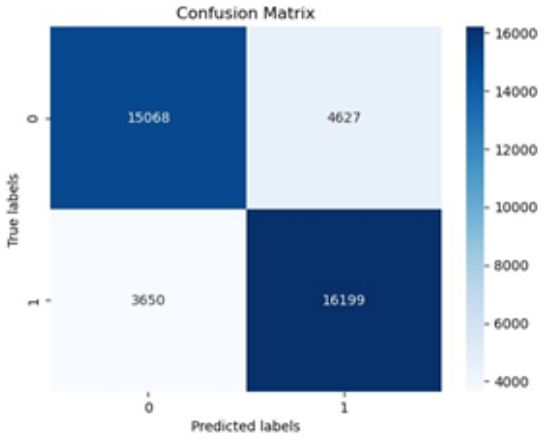

Fig. 9. Confusion Matrix

```
Classification Report:
              precision    recall  f1-score

         0.0       0.81      0.77      0.78
         1.0       0.78      0.82      0.80

    accuracy                           0.79
```

Fig. 10. Classification Report

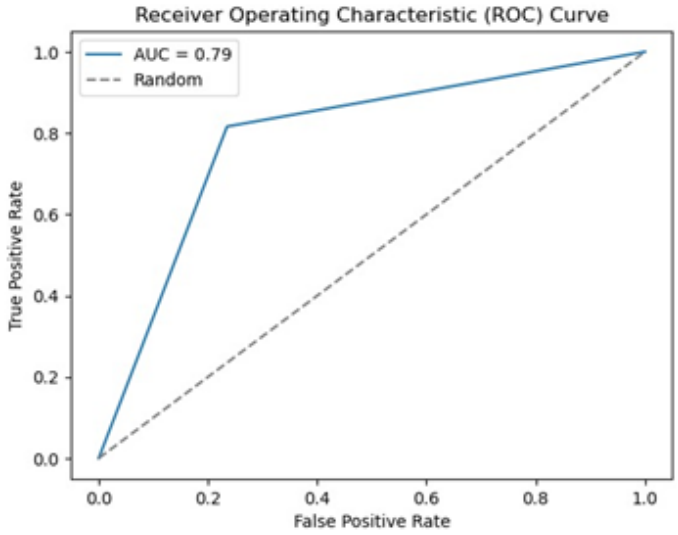

Fig. 11. ROC Curve

A. Test Cases

Problem 1: To predict if the lineup of five home players and five away players would lead to a win or loss for the home team. The input of the classification model will be the 10 players in terms of the embedded values after finding the encoded values. The output will be targeting the outcome of "1," meaning the home team won, or "0," that the away team won.

Problem 2: To predict one player for the away team that would result in a win "1." The input for this problem would be four players from the away team and five players from the home team, along with the year and the home and away team names. Once we can find where the players' names have been taken from, we can find the remaining players from the away team. Find the encoded values and embedding of those players, which can be used in the model. The new array would be 9 given players plus filling the position of the 10th player. These 10 players would be the input of the classification model. The output would be the target outcome ("1" or "0"). The lineup with the highest confidence for predicting a "1" will be the suggested lineup. The outcome of the model is in terms of an array, which is in the format of "0" or "1" (fig. 12). Additionally, we have another array representing the probabilities with which it is classifying the outcome as "0" or "1" (fig. 13). The first column is for class label "0," and the

```
array([1., 0., 1., 1., 1., 1., 0., 0., 1., 1., 1., 0.])
```

Fig. 12. Outcome Array

second column is for class label "1." Since we are focusing on the winnings of the away team, we will be focusing on the probability value of the first column.

```
array([[0.48083333, 0.51916667],
       [0.55916667, 0.44083333],
       [0.4715    , 0.5285    ],
       [0.44233333, 0.55766667],
       [0.49566667, 0.50433333],
       [0.38983333, 0.61016667],
       [0.54233333, 0.45766667],
       [0.5145    , 0.4855    ],
       [0.436     , 0.564     ],
       [0.45866667, 0.54133333],
       [0.46016667, 0.53983333],
       [0.53216667, 0.46783333]])
```

Fig. 13. Probability Matrix

For example, Test case 1 (fig. 14): if home team names are 'Joe Johnson', 'Josh Childress', 'Marvin Williams', 'Salim Stoudamire', 'Tyronn Lue' and the away team trial names are 'Grant Hill', 'Leandro Barbosa', 'Raja Bell', 'Sean Marks'. The Away team was 'PHO,' the home team was 'ATL', and the season was 2008. The following would print stating the probability of the suggested 10th player with their probability of winning.

```
Player :  521  Marcus Banks Probability : 0.636
Player :  699  Shawn Marion Probability : 0.83
Player :  96  Brian Skinner Probability : 0.616
Player :  718  Steve Nash Probability : 0.726
Player :  80  Boris Diaw Probability : 0.7
```

Fig. 14. Test Case 1 Output

For Test case 2 (fig. 15): if home team names are 'Acie Law','Al Horford','Joe Johnson','Josh Smith','Marvin Williams' and the away team trial names are 'Boris Diaw','Grant Hill','Raja Bell','Shawn Marion'.The Away team was 'PHO,' the home team was 'ATL', and the season was 2008.

```
Player :  Eric Piatkowski Probability : 0.5591666666666666
Player :  Linton Johnson Probability : 0.5423333333333333
Player :  Shaquille O'Neal Probability : 0.5145000000000001
Player :  Steve Nash Probability : 0.5321666666666667
```

Fig. 15. Test Case 2 Output

XI. Conclusion

The NBA Dataset consists of information on various games from 2007 to 2012. Each game was divided into states; each state had five home and away players' names, season, game ID, team names and statistics. There are two main problems. The objective of the classification task was to predict if the home team would win. Additionally, there is a need to predict the 10th player in the away team that would lead to a win against the home team. That is the problem we are solving. The main information used is the players' names and a few of the statistics that describe the team's performance, along with the outcome of the game. However, an up-sampling technique was employed due to the imbalance in the data. Furthermore, a feature transformation method of Label Encoding was utilized. Along with feature engineering, four factors (shooting the ball, taking care of the ball, offensive rebounding, and getting to the foul line) were also calculated. Various classification algorithms, namely SVM, KNN, Neural Networks, Random Forest, and Gradient Boost were applied to the dataset. In conclusion, using Embeddings as inputs and Random Forest as the classification model was proven to be the best model with an accuracy of 79%

References

[1] tiffany Do, "Using machine learning to predict game outcomes based on player-champion experience in League of Legends," The 16th International Conference on the Foundations of Digital Games (FDG) 2021.

[2] football match line-up prediction based on physiological variables: A machine learning approach, https:www.researchgate.netpublication359160060_Football_Match_Line-Up_Prediction_Based_on_Physiological_Variables_A_Machine_Learning_Approach

[3] N. M. Patil, B. H. Sequeira, N. N. Gonsalves, and A. A. Singh, "Cricket team prediction using Machine Learning Techniques," SSRN, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3572740.

[4] Four factors — Basketball-Reference.com. (n.d.). Basketball-Reference.com.

[5] "Embeddings in machine learning: Everything you need to know," FeatureForm, https://www.featureform.com/post/the-definitive-guide-to-embeddings.

[6] "NBA player Embiidings," NBA Player Embiidings, https://www.connorlandy.com/projects/nba-player-embiidings.