### **Stats 101C Final Project: Predicting Outcomes of NBA Games**

### TEAM 15

### Jun Ze He, Eduard Bueno, Brian Mualim, Anthony Ko, Theresa Unsulangi, Chloe Guo

### 

### **TABLE OF CONTENTS**

### **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

### **Introduction 2**

### **Data Preprocessing 3**

### **Experimental Setup 4**

### **Results and Analysis 5**

### **Recommendations 10**

### **References 111. Introduction**

In this paper, we investigate the NBA 2023-2024 dataset, which contains statistics on previous matches between teams in order to predict the results of future games. To do this, we trained and tested machine learning models on features of the NBA dataset in order to determine the best model. The features in the dataset include: team name (Team), the team they matched up against (Match), date of the game (Game Date), whether they won or lost the game (W/L), minutes played (MIN), points scored (PTS), field goals made (FGM), field goals attempted (FGA), field-goal percentage (FG%), three-point goals made (3PM), three-point goals attempted (3PA), three point field goal percentage (3P%), free throws made (FMA), free throws attempted (FTA), free throw percentage (FT%), offensive rebounds (OREB), defensive rebounds (DEB), total rebounds (REB), assists (AST), steals (STL), blocks (BLK), turnovers (TOV), personal fouls (PF), and a plus/minus statistic (+/-).

The modeling process involved three algorithms: Logistic Regression, Random Forest, and XGBoost. To optimize performance, we applied feature selection through forward stepwise feature selection (FSFS), and tuned parameters using grid search and 5-fold cross validation. Additionally, Principal Component Analysis (PCA) was employed for dimensional reduction to select the optimal number of features. Among these models, logistic regression with PCA-transformed features performed the best with an accuracy of 68.55%, while XGBoost with PCA-transformed features performed the worst with an accuracy of 67.39%.

### 

### **2. Data Preprocessing**

First, we verified whether any features contained missing values (NA entries) and confirmed that none did. The *W/L* feature was converted into a binary format, where 0 represented a loss (*L*) and 1 represented a win (*W*). During our exploration of the dataset, we identified an issue with the *FT%* feature, which contained a non-numerical value, "-". This indicated that *FT%* was being treated as a categorical feature. To address this, we located the row with the "-" entry, which corresponded to a game played by the BOS team. The issue might have been caused by a data entry error or missing data for that game. To resolve this, we calculated the mean *FT%* for BOS based on games prior to the row in question. The calculated mean was 79%, and we replaced the "-" entry with this value. After this replacement, the *FT%* feature was successfully converted from categorical to numerical.

#### **Features Used to Train Models**

* **Pre\_mean\_diff\_features:** Since the original features represented the outcomes of completed games, they were unsuitable for predicting W/L in upcoming games. To predict the future outcome of a game, we calculated pre-game average features based on a team's and its opponent's prior performance in metrics. This included the following: **MIN, PTS, FGM, FGA, FG%, 3PM, 3PA, 3P%, FTM, FTA, FT%, OREB, DREB, REB, AST, STL, BLK, TOV, PF, and +/-**. To incorporate the relative performance between teams, we calculated the difference in prior performance by subtracting the opponent's averages from the team's averages. For instance, to predict whether BOS would win a game against PHX on 2023-03-10, we used the previous mean difference features to evaluate which team had better prior performance. So if the previous mean difference in PTS was 30, this indicated that BOS scored an average of 30 more points than PHX in games **before** 2023-03-10. This would provide more evidence that BOS might be more likely to win. Hence the differences of mean features could serve as valuable indicators for models learning to predict W/L outcomes. This approach could similarly be applied to the following features:

| Pre\_mean\_diff\_MIN | Pre\_mean\_diff\_PTS | Pre\_mean\_diff\_FGM | Pre\_mean\_diff\_FGA | Pre\_mean\_diff\_FG% |
| --- | --- | --- | --- | --- |
| Pre\_mean\_diff\_3PM | Pre\_mean\_diff\_3PA | Pre\_mean\_diff\_3P% | Pre\_mean\_diff\_FTM | Pre\_mean\_diff\_FTA |
| Pre\_mean\_diff\_FT% | Pre\_mean\_diff\_OREB | Pre\_mean\_diff\_DREB | Pre\_mean\_diff\_REB | Pre\_mean\_diff\_AST |
| Pre\_mean\_diff\_STL | Pre\_mean\_diff\_BLK | Pre\_mean\_diff\_TOV | Pre\_mean\_diff\_PF | Pre\_mean\_diff\_+/- |

* **home\_game**: indicate in binary if a game was a home game or not, for the team whose name came first (e.g.: GSW in “GSW vs. PHX”). This helps in seeing if a team has a “home game” advantage and thus are more likely to win.
* **Pre\_Total\_Wins**: looking at the current game’s team on the left side of “@” or “vs.”, pre\_total\_wins is how many times the team has beaten the opponent team before.
* **Weights**: for the total number of games a team has played, say *n*, the most recent game they played gets weight *n*, the second most recent *n-1*, etc. In particular, games that occurred most recently for a team were weighted more, because they were more indicative of how well the team has been playing.
* **Pre\_mean\_AST\_TOV\_Ratio**: The AST feature is for assists, meaning that a goal was made, while the TOV feature is for when a team loses the ball (e.g. the ball getting stolen by the opponent team). We included the AST TOV ratio to show if a team was more likely to make a goal (via assists) or lose the ball. If a team had a low AST TOV ratio, they would be more likely to lose, because they lose the ball more often.

To evaluate the true performance of our models, we divided the modeling dataset into a training set and a testing set based on game dates. A split date of 2024-03-01 was chosen, resulting in approximately 70% of the data being used for training and 30% for testing. Following a comprehensive feature engineering process, the raw dataset was transformed into a modeling dataset comprising 24 features. We removed Team, Match, Game Date, and Opponent from our modeling dataset because they are meaningless to a model. Above is the list of all features included in the modeling dataset.

### 

### **3. Experimental Setup**

After completing the feature engineering process, we identified a subset of features to use in building predictive models. The models employed included logistic regression with a Lasso penalty, random forest classifier, and XGBoost classifier.

We started to use all features to train and test our model to examine if our models achieve a decent performance. Since we wanted to maximize the overall classification accuracy, we had to choose the best subset of features suitable for models. The best subset of features can increase performance of each model and keep the model as less complex as possible.

To select the optimal subset of features for each model, we applied forward stepwise feature selection (FSFS), using the respective model as the estimator in the FSFS process. The selection criterion was overall classification accuracy. For instance, if a particular subset of features enabled the random forest classifier to achieve the highest accuracy compared to other subsets, FSFS identified that subset as the best for the random forest classifier model.

After identifying the optimal subset of features for each model, we used these subsets to build and improve the respective models. To fine-tune the models, we applied a **grid search** with **5 fold cross validation** to optimize hyperparameters, including **n\_estimators** and **max\_depth** for the random forest, as well as **n\_estimators**, **max\_depth**, and **learning\_rate** for XGBoost. Meanwhile, we also applied **Principal Component Analysis (PCA)** for dimensionality reduction, selecting the optimal number of principal components for each model to enhance performance. For example, suppose a random forest model achieves the highest accuracy of 65% with 5 n\_estimators, a max\_depth of 3, and 3 principal components from PCA. In that case, we consider that this is the best model, and we use it to evaluate the testing dataset.

### 

### **4. Results and Analysis**

#### **Initial Model Comparison**

We evaluated the following classification algorithms on the modeling train and modeling test datasets using all features:

1. **Logistic Regression (L1 Regularized)**: Logistic Regression is a discriminative model that estimates the conditional probability of a sample belonging to a specific class. For this experiment, we applied Logistic Regression with an L1 regularization penalty and maximum iterations set to 10000. In the original model, we achieved a highest test accuracy of 0.639
2. **Random Forest**: Random Forest is an ensemble learning algorithm that combines predictions from multiple decision trees to make a final classification. The initial number of estimators we ended up using is 5, and the maximum tree depth is 3 In the original model, we achieved a highest test accuracy of 0.643
3. **XGBoost Classifier**: XGBoost is a gradient-boosting framework known for its flexibility and efficiency. The initial number of estimators is 5, the maximum depth is 3, and the learning rate that we used was 0.15. For the initial model, we achieved a test accuracy of 0.627.

After comparing the initial models, we selected the best features from the dataset by using the **SequentialFeatureSelector** library from scikit-learn, and used a forward feature selection approach to determine the optimal number of features to select based on accuracy. The forward feature selection approach identifies the subset of features that maximizes accuracy by adding features sequentially. It starts from zero and stops when no additional features improve accuracy further

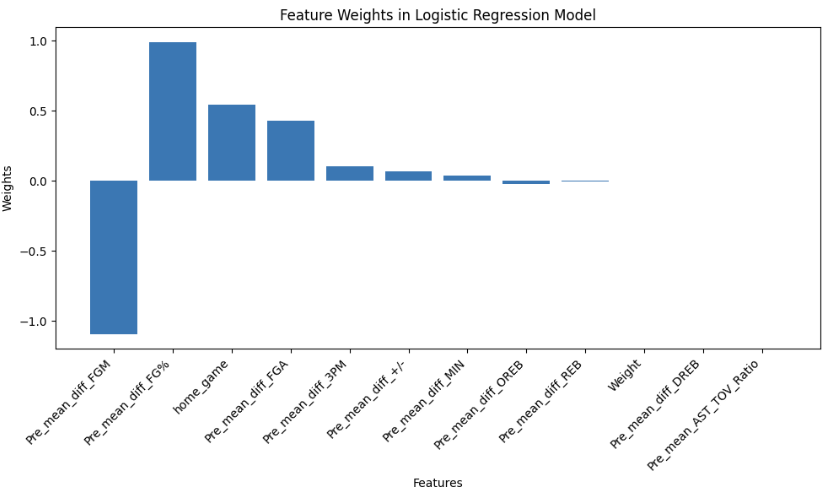
#### **Selected Features for Models**

**Logistic Regression:**

Variables Used:

| home\_game | Pre\_mean\_diff\_MIN | Pre\_mean\_diff\_FGM |
| --- | --- | --- |
| Pre\_mean\_diff\_FGA | Pre\_mean\_diff\_FG% | Pre\_mean\_diff\_3PM |
| Pre\_mean\_diff\_OREB | Pre\_mean\_diff\_DREB | Pre\_mean\_diff\_REB |
| Pre\_mean\_diff\_+/- | Weight | Pre\_mean\_AST\_TOV\_Ratio |

The model now yields an accuracy rate 0.640, a 0.05 increase than previously tested.

****

From the graph shown above, we can also see that previous mean different FGM have the greatest impact on the W/L of any given team with a weight of -1.34, which also suggests that it’s negatively correlated to the W/L outcome of the team.

**Random Forests**

Variables Used:

| home\_game | Pre\_mean\_diff\_MIN | Pre\_mean\_diff\_PTS |
| --- | --- | --- |
| Pre\_mean\_diff\_FGA | Pre\_mean\_diff\_FG% | Pre\_mean\_diff\_3PM |
| Pre\_mean\_diff\_3PA | Pre\_mean\_diff\_3P% | Pre\_mean\_diff\_TVO |
| Pre\_mean\_diff\_+/- | Pre\_Total\_Wins | Weight |

The model now yields an accuracy rate 0.646, a 0.13 increase than previously tested.

**XGBoost**

Variables Used:

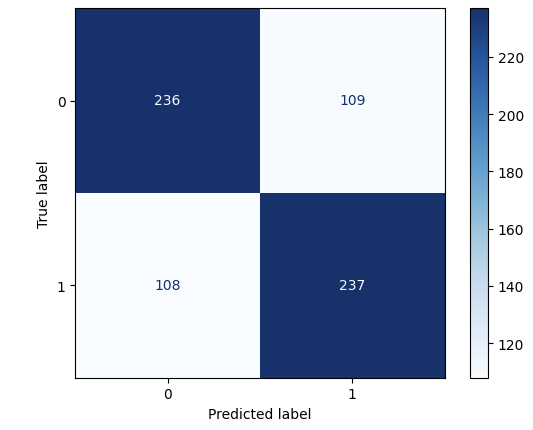
| Pre\_mean\_diff\_MIN | Pre\_mean\_diff\_PTS | Pre\_mean\_diff\_FGM |
| --- | --- | --- |
| Pre\_mean\_diff\_FG% | Pre\_mean\_diff\_3P% | Pre\_mean\_diff\_FTM |
| Pre\_mean\_diff\_FTA | Pre\_mean\_diff\_BLK | Pre\_mean\_diff\_TVO |
| Pre\_mean\_diff\_+/- | Pre\_Total\_Wins | Weight |

The model now yields an accuracy rate 0.649, a 0.022 increase than previously tested.

**Fine Tuning through PCA:**As shown by the logistic regression weight graph, we still have some variables which contribute very little to our results, yet are included as part of our predictor variables. We fine-tuned our results through the use of PCA to reduce the dimensionality of our datasets while maintaining the variance within our data.

**Logistic Regression**

A grid search is conducted over the hyper parameters of the logistic regression to automatically select the best regularization strength between 2, 5, 7, and 10, using a 5-fold cross-validation. The model then undergoes a PCA transformation, and we find out the best model includes **4 components** with **5 regularization strength**, and has an accuracy of 0.6855

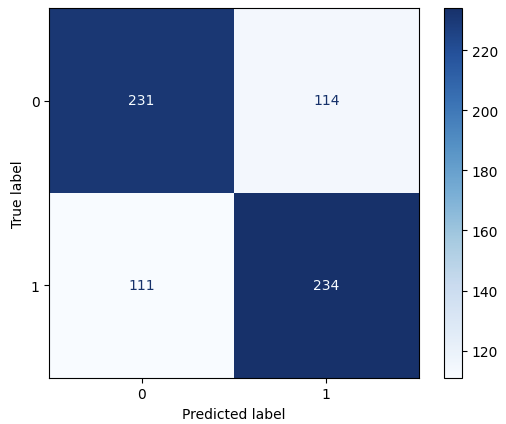
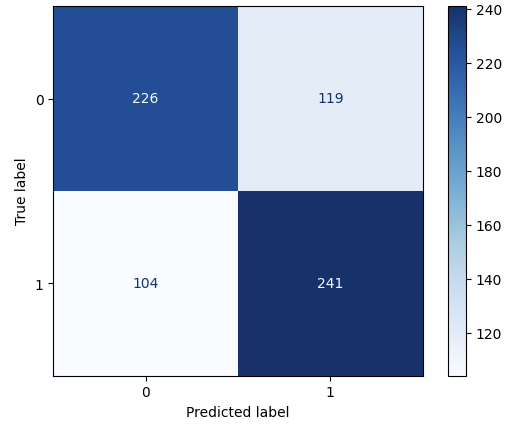


**Random Forest**

To optimize its performance, we conducted a grid search with 5 fold cross validation over hyperparameters such as the number of estimators, maximum tree depth, and maximum features.This model performed best on the reduced dataset of **3 components**, achieving an accuracy of 0.6768. The optimal configuration involved setting the **maximum depth to 3** and using **50 estimators**.

**XGB Boosting**

In this study, we tuned parameters such as tree depth, learning rate, and the number of estimators through a grid search with 5 fold cross validation. Using PCA-transformed features of **6 components**, the model achieved a competitive test accuracy of 0.6739. The best results were obtained with a **maximum tree depth of 3**, **10 estimators**, and a **learning rate of 0.2**.



Random Forest XGBoosting

**Conclusion:**

For this dataset, logistic regression with PCA-transformed features performed the best with a 0.6826 accuracy rate, while XGB Boosting with PCA-transformed features performed the worst with a 0.626 accuracy rate.

### 

### **5 Recommendations**

A key consideration in this project is the choice between logistic regression and tree-based methods. Logistic regression was chosen because it outperforms tree methods in this specific case. The primary factor influencing this decision is the size of the dataset. Logistic regression is particularly well-suited for smaller datasets due to its robustness and ability to generalize without overfitting [1]. On the other hand, tree-based methods, while highly effective for larger datasets, tend to struggle with small datasets [2]. They are prone to overfitting and may fail to capture meaningful patterns when there is limited data available.

While the final accuracy between logistic regression and tree methods is close, the interpretability and stability of logistic regression make it the more appropriate choice. Logistic regression provides straightforward coefficients that indicate the direction and magnitude of variable relationships, which is especially valuable for explaining results and drawing actionable conclusions. To improve the performance of both methods and obtain more reliable results, it would be beneficial to increase the number of observations in the dataset. Tree methods, in particular, are known to excel with larger datasets, where their ability to model complex interactions and non-linear relationships can be fully utilized. Expanding the dataset could help bridge the performance gap and provide a clearer understanding of the underlying patterns [2]. For now, however, logistic regression remains the most practical and effective approach given the constraints of the data size.

Logistic regression and tree-based methods each have distinct advantages and limitations. Logistic regression, a widely used statistical modeling technique for binary classification, is particularly effective for small to moderately sized datasets. It assumes a linear relationship between predictors and the log odds of the outcome, offering clear interpretability through its coefficients and computational efficiency. Its simplicity and resistance to overfitting make it a strong choice for smaller datasets. In contrast, tree-based methods, such as decision trees, random forests, and gradient boosting, excel at capturing complex interactions and non-linear relationships [2]. These flexible models, however, typically require larger datasets to perform optimally and are more prone to overfitting when applied to smaller datasets. Additionally, tree-based methods often lack the straightforward interpretability of logistic regression, which can make explaining model results more challenging.

### 

### 

### **6 References**

[1] Raphael Cournonne, Philipp Probst, and Anne-Laure Boulesteix, “Random forest versus logistic regression: a large-scale benchmark experiment”, *BMC*, 2018, <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5>

[2] Sharmeen Binti Syazwan Lai, Nur Huda Nabihan Binti Md Shahri, Mazni Binti Mohamad, Hezlin Aryani Binti Abdul Rahman, and Adzhar Bin Rambli, “Comparing the Performance of AdaBoost, XGBoost, and Logistic Regression for Imbalanced Data”, *Horizon Research Publishing*, November 17, 2020, <https://www.hrpub.org/download/20210530/MS20-13491670.pdf>