Technical Report: Final Project DS 5110: Introduction to Data Management and Processing

Team Members: Ashwin Satish, Joseph Russell
Khoury College of Computer Sciences
Data Science Program
satish.ash@northeastern.edu, russell.jo@northeastern.edu

December 10, 2024

Contents

1 Introduction

As basketball fans, we thought it would be interesting to try to predict the outcomes of games using statistics. Using NBA data from the past 5 seasons, we gathered team and player metrics as well as historical game scores in order to determine how scores can be modeled from general statistics. Our process involved collecting multiple CSVs, compiling and preprocessing them in Python, performing exploratory data analysis and visualization, and ultimately building machine learning models to predict margins of victory for 6000 historical games.

2 Literature Review

Using basketball statistics to predict game outcomes involves a thorough understanding of how these statistics are recorded and their relevance to basketball analytics. While many of these are fairly intuitive, we researched advanced statistics that are intended to encapsulate overall player performance in different ways, such as Win Shares, PER (Player Efficiency Rating), and VORP (Value Over Replacement Player). Understanding the purpose of these metrics allows us to analyze players and teams more accurately and help build our model.

3 Methodology

The methods and techniques used in the project ranged from Data Collection, Data Proprocessing and Analysis, and machine learning modeling.

3.1 Data Collection

For collecting the appropriate team data, player data, match by match offensive and defensive data alongside the advanced player metrics which included some feature engineering, we referred to Basketball Reference (https://basketball-reference.com).

Our data collection method involved leveraging the site for all the information required and combining it into a single csv which had data for the past 5 seasons for the teams and players for each of the categories mentioned above.

3.2 Data Preprocessing

The steps we implemented for data preprocessing were
1. Handling missing values: We had many missing values present in our datasets, this is because the data given online was flawed and was not complete in nature. We had to replace numerical NaNs with appropriate means or medians where necessary.
2. Worked on correcting data inconsistencies: We had to standardize all values into appropriate formats for the model to understand. We implemented label encoding for the players, positions and teams. We also fixed data entry errors like incorrect player stats (pts scored ¿ total shots attempted)
3. Duplicate records: There were several duplicate entries which caused errors in data collection/merging.

4. Encoding categorical data: We had to work on encoding categorical data like player names, positions and teams.
5. Scaling: Implemented robust scaling after trying and understanding how each scaling method affects our data accordingly.

3.3 Analysis Techniques

The analytical techniques used in the project are:
1. 3PT shoots v Win Margins: This visualization explains how 3 pointers affect win margins and win conditions in the game.
2. Points allowed v scored: This visualization explains how many points were scored by a team and how many were scored against the team.
3. Kmeans cluster: A cluster showing the two different categories or performances of players in the dataset. Implemented k using elbow method to find optimal clusters.

3.4 Modeling

We used linear regression as a baseline model which took team data as input to predict final game scores. Improving on this model, we scraped lists of players who played in each game to remove injured or traded players from our prediction. We tried linear regression, lasso regression, ridge regression, support vector machines, and random forests.

4 Results

K-Means Clustering results:
1. Silhouette Score: 0.621
2. Optimal k: 3
3. The cluster has a proper seperation of the points, which shows that there is an appropriate basis for clustering the data

Game Outcome Modeling: 1. Linear baseline model MSE: 180.46
2. Lasso Regression MSE: 170.2

5 Discussion

Our linear regression model provided a baseline mean squared error of 180 using team data. We were able to improve model accuracy by using game-specific player data instead of team data. Our best-performing model, Lasso regression, achieved an MSE of 170. Linear regression and ridge regression had similar results, while random forests performed worse as most relationships within the data were linear.

6 Conclusion

While NBA games can be very unpredictable, our model proves that team and player statistics can be used to understand to a degree how a game will turn out. Future models could try to incorporate time series data as game outcomes are often dependent on how teams and players perform in recent games more than older ones.

7 References

Data gathered from https://www.basketball-reference.com/

References

A Appendix A: Code

```
team_offense = pd.read_csv("drive/MyDrive/Colab Notebooks/SlamDunk/SlamDunkInsights_TeamData.csv")
team_defense = pd.read_csv("drive/MyDrive/Colab Notebooks/SlamDunk/TeamDefense.csv")
team_defense = team_defense.rename(columns={'season':'Year'})
team_defense = team_defense.drop('Rk', axis=1)
team_defense['Team'] = team_defense['Team'].str.replace('*', '', regex=False)
team_offense['Team'] = team_offense['Team'].str.replace('*', '', regex=False)
team_defense = team_defense.drop(['G', 'MP'], axis=1)
team_offense = team_offense.drop(['G', 'MP'], axis=1)
team_df = pd.merge(team_offense, team_defense, on=['Team', 'Year'])
team_df.head()
```

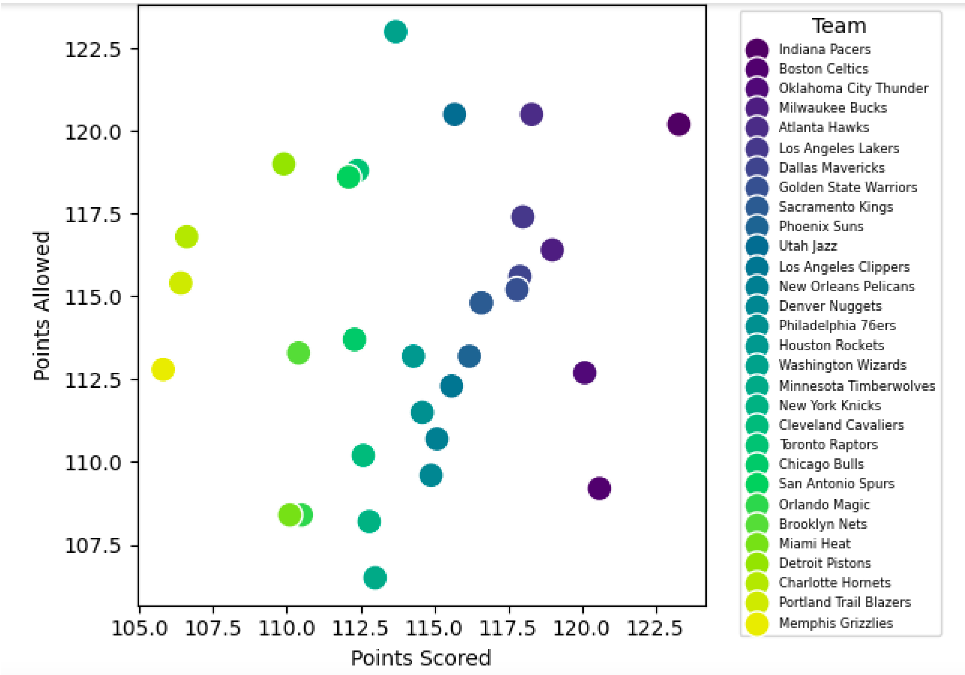
Listing 1: Preprocessing Team Data

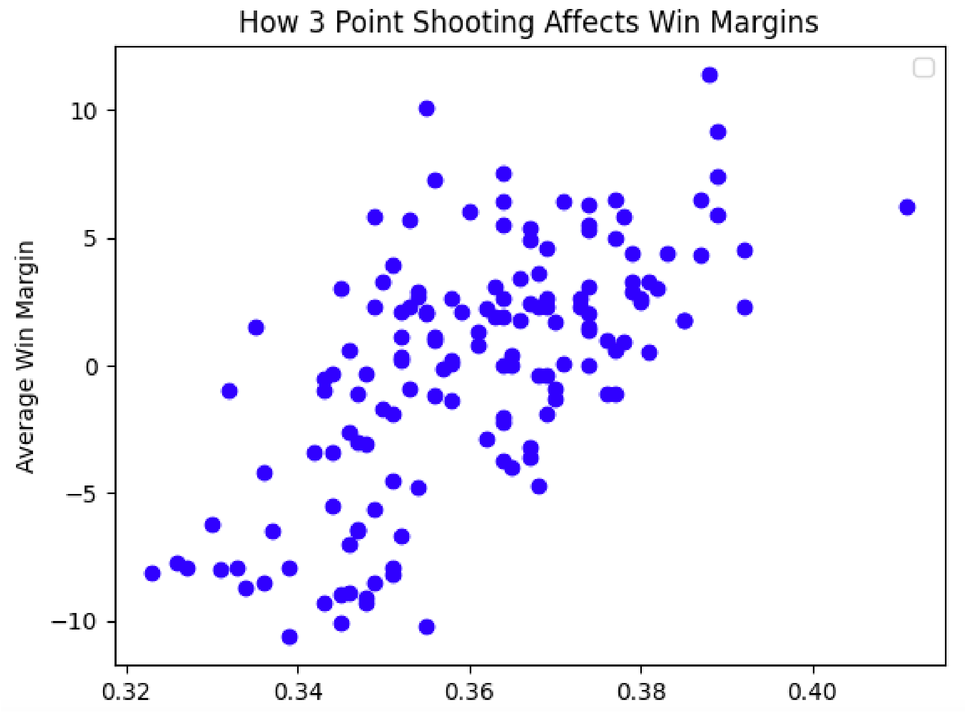
```
from sklearn.decomposition import PCA

pca = PCA(n_components=3)
pca_data = pca.fit_transform(splayer_df[cts])

explained_variance_ratio = pca.explained_variance_ratio_
variance_df = pd.DataFrame(explained_variance_ratio, columns=['Explained Variance Ratio'], index=['PC1', 'PC2','PC3'])
print(variance_df)
cumulative_variance_ratio = pca.explained_variance_ratio_.cumsum()
print("Cumulative Explained Variance Ratio:")
print(cumulative_variance_ratio)
loadings = pd.DataFrame(pca.components_.T, columns=['PC1', 'PC2','PC3'], index=cts)
print(loadings)
```

Listing 2: Dimensionality Reduction

B Appendix B: Additional Figures

Include any additional figures or tables that support the analysis.


Points Allowed
122.5
120.0
117.5
115.0
112.5
110.0
107.5
105.0
107.5
110.0
112.5
115.0
117.5
120.0
122.5
Points Scored
Team
Indiana Pacers
Boston Celtics
Oklahoma City Thunder
Milwaukee Bucks
Atlanta Hawks
Los Angeles Lakers
Dallas Mavericks
Golden State Warriors
Sacramento Kings
Phoenix Suns
Utah Jazz
Los Angeles Clippers
New Orleans Pelicans
Denver Nuggets
Philadelphia 76ers
Houston Rockets
Washington Wizards
Minnesota Timberwolves
New York Knicks
Cleveland Cavaliers
Toronto Raptors
Chicago Bulls
San Antonio Spurs
Orlando Magic
Brooklyn Nets
Miami Heat
Detroit Pistons
Charlotte Hornets
Portland Trail Blazers
Memphis Grizzlies


How 3 Point Shooting Affects Win Margins
Average Win Margin
10
5
0
−5
−10
0.32
0.34
0.36
0.38
0.40

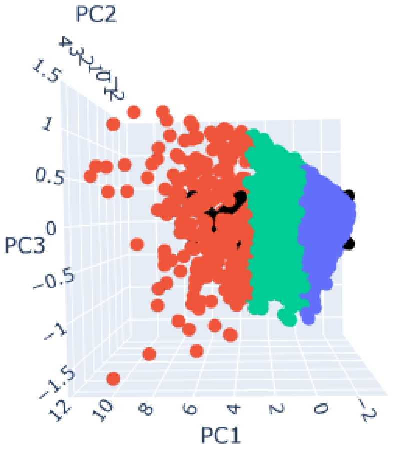


Figure 1: K-Means Cluster