



**CMPN451: Big Data**

**Credit Hours System – Spring 2024**

Team 19

|  |  |
| --- | --- |
| Ammar Nasser | 1190543 |
| Anas Sherif | 1190375 |
| Marwan Mostafa | 1190281 |
| Omar Faheem | 1190477 |

**NBA-Shot Logs**

**Dataset link:** <https://www.kaggle.com/datasets/dansbecker/nba-shot-logs>

Contents

[Problem description 3](#_Toc166511386)

[Project Pipeline 4](#_Toc166511387)

[Data loading 4](#_Toc166511388)

[Data preprocessing 4](#_Toc166511389)

[Data visualization 4](#_Toc166511390)

[Outliers detection 4](#_Toc166511391)

[Prediction models 4](#_Toc166511392)

[Map-Reduce 4](#_Toc166511393)

[Analysis and solution of the problem 5](#_Toc166511394)

[Data preprocessing 5](#_Toc166511395)

[Visualization 6](#_Toc166511396)

[Correlation heatmap: 6](#_Toc166511397)

[Interesting insight 7](#_Toc166511398)

[Correlation circle plot 8](#_Toc166511399)

[Outlier detection 9](#_Toc166511400)

[Model/Classifier training 14](#_Toc166511407)

[Logistic regression 15](#_Toc166511408)

[Random forest 16](#_Toc166511409)

[KNN 17](#_Toc166511410)

[Gradient boosting 18](#_Toc166511411)

[Map reduce 19](#_Toc166511412)

[1. the most scoring players in the last 30 seconds of the last quarter 19](#_Toc166511413)

[2. top shot players who scored 3 points with 0 dribbles 20](#_Toc166511414)

# Problem description

The problem under investigation revolves around understanding the shooting patterns and defensive performance in NBA games during the 2014-2015 season. This analysis aims to delve into the intricacies of shot logs, which contain detailed information about each shot attempted during games, including the shooter's position, shot distance, shot result (made or missed), and details about the closest defender.

Key questions we aim to address include:

1. **Shooting Efficiency**: How efficient were players in making their shots during the 2014-2015 NBA season? Were there particular trends or patterns observed in the shooting percentages based on factors such as shot distance, shot clock, or defender proximity?
2. **Defensive Impact**: What was the defensive impact on shooting percentages? Were there correlations between the proximity of the closest defender and the likelihood of a successful shot? Additionally, what insights can be gleaned from analyzing shot logs regarding defensive strategies employed by teams and individual players?
3. **Outlier Detection**: How prevalent were outliers in the dataset, and what impact did they have on the analysis? Are there specific outliers that significantly skew the data, and how should they be addressed to ensure the integrity of our findings?

By addressing these questions, we aim to gain valuable insights into the dynamics of NBA games during the specified season, shedding light on both offensive and defensive aspects of the sport. These insights will not only contribute to a deeper understanding of player performance but also have potential implications for strategic decision-making by coaches and team management.

# Project Pipeline

## Data loading

We read the data from the csv file “shot\_logs.csv” into a pandas dataframe.

## Data preprocessing

Then we do some preprocessing on the data to ensure the data has no Nulls and makes sense.

## Data visualization

We visualize the dataset with our insights and get the correlation between each column.

## Outliers detection

We detect the outliers in each column and plot them using the box plot.

## Prediction models

We train four different models (logistic regression, random forest, KNN, gradient boosting).

## Map-Reduce

We perform two map and reduce operations to get (the most scoring players in the last 30 seconds of the last quarter, top shot players who scored 3 points with 0 dribbles)

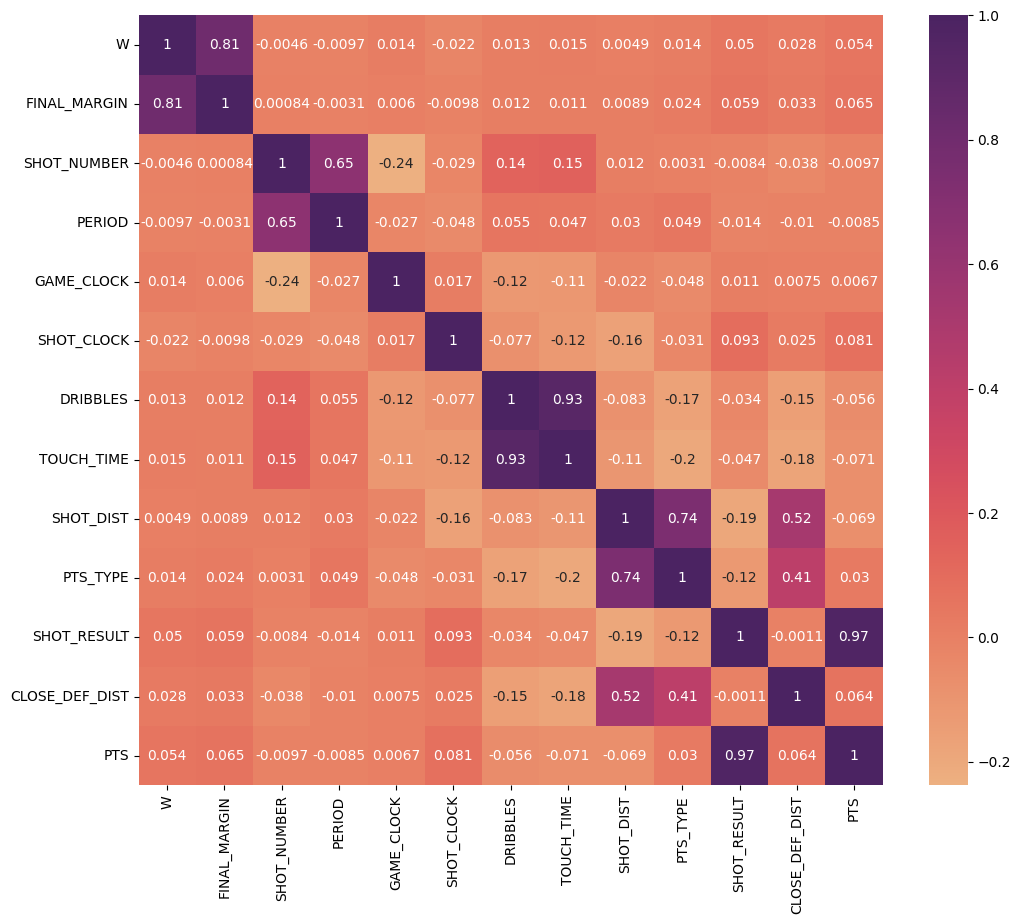
# Analysis and solution of the problem

## Data preprocessing

* **Handling Null values:** we first extract the columns that contain null values, then we fill them with forward filling.
* **Handling negative touch time:** we replace any negative value in the TOUCH\_TIME column with the mean of all the positive values.
* **Removing unnecessary columns:** we remove the FGM column as it corresponds exactly to the column SHOT\_RESULT, as the data set is per shot, so the FGM is either 0 if missed and 1 if made.
* **Label encoding:** we change the values from the W column to be 1 or 0 instead of W or L
* **Time conversion:** in the GAME\_CLOCK, we calculate the time of the game in seconds instead of the format mm:ss.
* **Handling outliers (optional):** we calculate the lower and upper bound for the outliers in each column then remove that column and return the cleaned data.

## Visualization

### Correlation heatmap:



The heatmap uses a color scheme to show the strength of the correlation between two features, where a darker color indicates a weaker correlation.

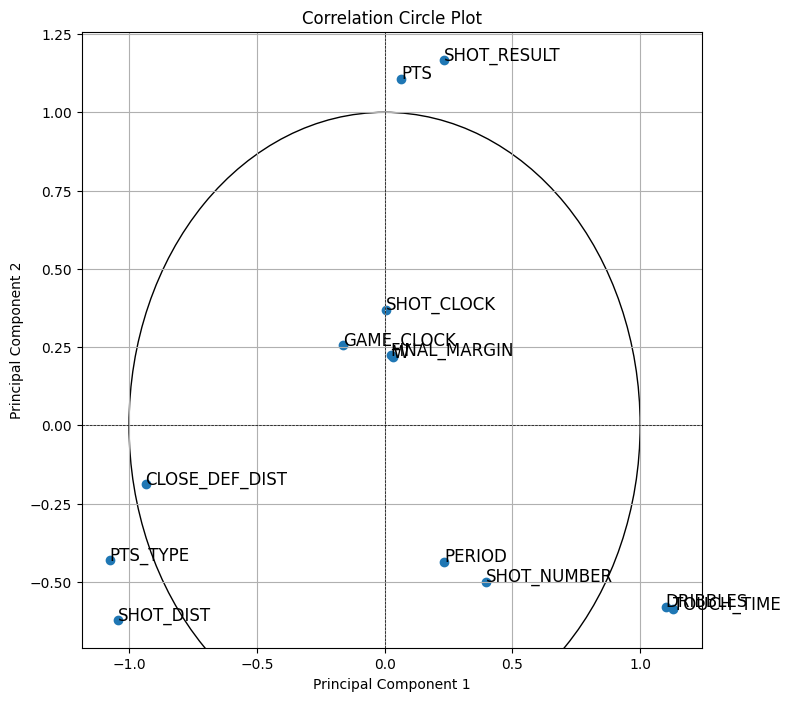
### Interesting insight

A pie chart with numbers and a percentage

Description automatically generated

In basketball, shots made from over 23.9 yards count as 3 points. However, if a player is fouled attempting a shot and scores, they get an extra free throw. So, if a player is fouled near the 23.9-yard mark and scores, they earn 3 points (2 points for the basket plus 1 point for the free throw). This can explain why about 8% of shots made from less than 23.9 yards are recorded as 3 points.

### Correlation circle plot



The correlation circle plot visually represents the relationships between multiple features in a dataset. Each feature is represented by a point on the plot, and the proximity of these points to each other indicates the strength and direction of their correlation. The plot uses vectors to illustrate the correlation structure, with the angle between vectors indicating the degree of correlation. A shorter distance between points or vectors implies a stronger correlation, while a longer distance suggests a weaker correlation. Therefore, the plot serves as a visual aid to understand the complex interplay between different features in the dataset.

## Outlier detection

## 

## 

## 

## 

## 

## 

## Model/Classifier training

We split the data randomly to 75% for the training and 25% for the evaluation.

We used 4 different prediction models.

### Logistic regression

A screenshot of a computer

Description automatically generated

### Random forest

A screenshot of a computer

Description automatically generated

### KNN

A screenshot of a computer

Description automatically generated

### Gradient boosting

A screenshot of a computer

Description automatically generated

## Map reduce

### the most scoring players in the last 30 seconds of the last quarter

We first made the map reduce function using pySpark.

A screenshot of a computer program

Description automatically generated

Then we implemented the mapper function and reducer function from scratch.

A screenshot of a computer screen

Description automatically generated

### top shot players who scored 3 points with 0 dribbles

We first made the map reduce function using pySpark.

A screenshot of a computer

Description automatically generated

Then we implemented the mapper function and reducer function from scratch.

A screenshot of a computer

Description automatically generated