

# Unmask the NBA blackbox

-Predicting NBA player salary with Machine Learning

**IEOR 4523 Data Analytics**

**Group Tofu Chill**

- Zhirui Cheng
- Ziwei Li
- Zhe Xi
- Yuchen Fei

# Agenda

1. Problem Background and Data Sources
2. Data Cleaning and Preprocessing
3. Data Descriptive Analysis
4. Assumption and Feature Models
5. Machine Learning Algorithms
6. Text Mining process
7. Conclusion and Reflections

# Agenda

1. **Problem Background and Data Sources**
2. Data Cleaning and Preprocessing
3. Data Descriptive Analysis
4. Assumption and Feature Models
5. Machine Learning Algorithms
6. Text Mining process
7. Conclusion and Reflections

# Problem Background

What are the secrets behind the high payoff of popular NBA players?

What are the possible factors that affect NBA player's salary?

- The player's basic information?
  - Age, Weight, Height
- The player's performance?
  - Wins, PTS
- Public comments and thoughts?
- US Economic factors?



# Data Sources:

Data theme	Data Source	Time Range	Approach
Player Performance	NBA official website	Season 1996 - 2018	Download
Player Salary	Hoopshype	Season 1990 - 2019	Web-Scraping
Player Bio	Kaggle Dataset	Static: Heights, weights, ...	Download
Comments on players	Twitter	Raw tweet text	Twitter API
Macroeconomics	US Bureau of Labor Statistics	Year 1996-2019	Download

# Agenda

1. Problem Background and Data Sources
2. **Data Cleaning and Preprocessing**
3. Data Descriptive Analysis
4. Assumption and Feature Models
5. Machine Learning Algorithms
6. Text Mining process
7. Conclusion and Reflections

# Data Cleaning and Preprocessing

## **Preprocess Original Data Sets:**

- Bio df: Only select 4 attributes from the biodf due to data integrity
- Salary df:
  - Use CPI to unify salary in 2019 context
  - Shift salary to match with last year performance
  - Adding feature: tenure, playing years of player in NBA
- Player df: Adjust name of TEAM to be consistent overtime

## **Merge 3 datasets by player's name and corresponding season:**

- Adjust different expressions of player's name in different datasets by looking at unmatched players after left merge and right merge

# Agenda

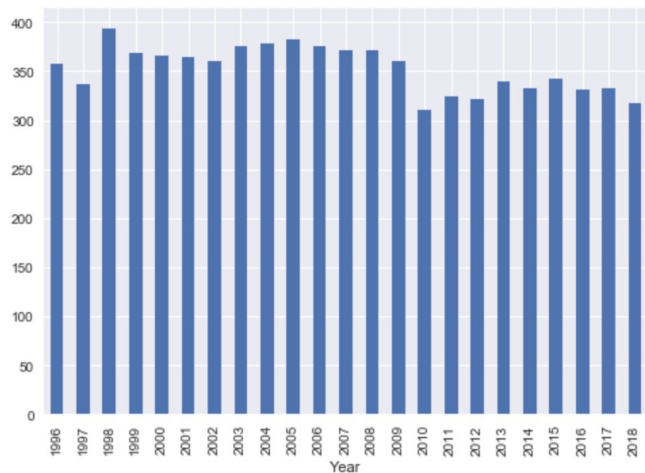
1. Problem Background and Data Sources
2. Data Cleaning and Preprocessing
3. **Data Descriptive Analysis**
4. Assumption and Feature Models
5. Machine Learning Algorithms
6. Text Mining process
7. Conclusion and Reflections



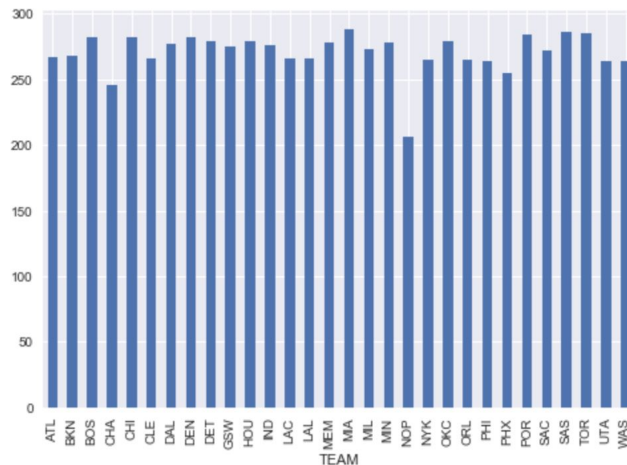
# Data Descriptive Analysis: 8117 rows, 34 columns

```
1 df.columns
```

```
Index(['Year', 'Name', 'Salary2019', 'tenure', 'TEAM', 'AGE', 'GP', 'W', 'MIN',  
      'PTS', 'FGM', 'FGA', 'FG%', '3PM', '3PA', '3P%', 'FTM', 'FTA', 'FT%',  
      'OREB', 'DREB', 'REB', 'AST', 'TOV', 'STL', 'BLK', 'PF', 'FP', 'DD2',  
      'TD3', '+/-' , 'Position', 'Height', 'Weight'],  
      dtype='object')
```



Data Size by Year



Data Size by TEAM

# Data Descriptive Analysis:

## Non-stationary issue

```
1 df.boxplot(by='Year', column=['Salary2019'], grid=False, figsize=(12,7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x120976940>
```



- **Problem:** Although preprocessed by CPI, the average NBA salary still significantly increased over the past 20 years.
- **Solution:** normalize salary data by putting it as a percentage of the league's salary cap, the total salary limit that a team can spend on its players in a given season, **which intuitively eliminate the year economic influence in salary**

# Data Descriptive Analysis:

Replace dependent variable with salary weight

```
1 df.boxplot(by='Year', column=['Salary2019'], grid=False, figsize=(12,7))
```

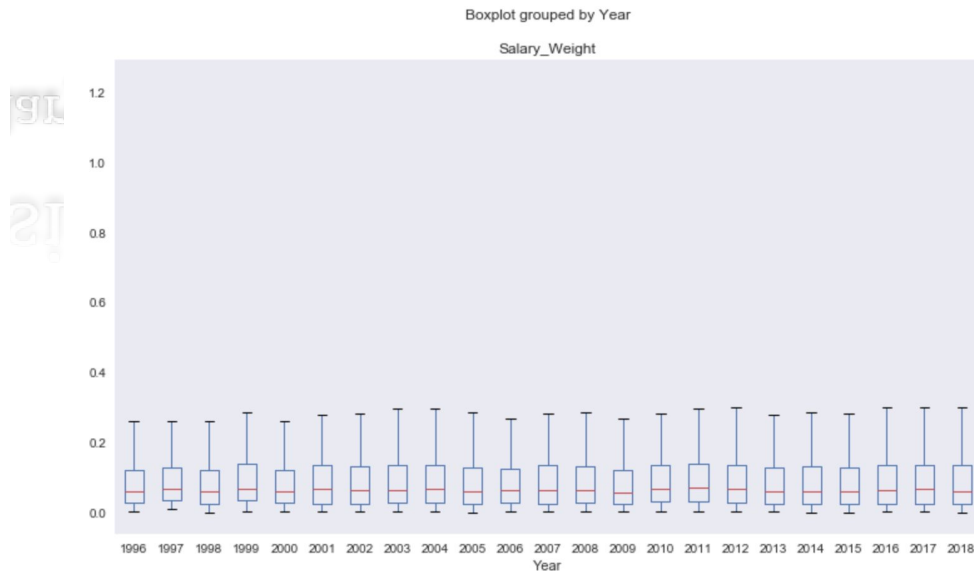
<matplotlib.axes.\_subplots.AxesSubplot at 0x120976940>



Absolute Salary

```
1 df.boxplot(by='Year', column=['Salary_Weight'], grid=False, figsize=(12,7))
```

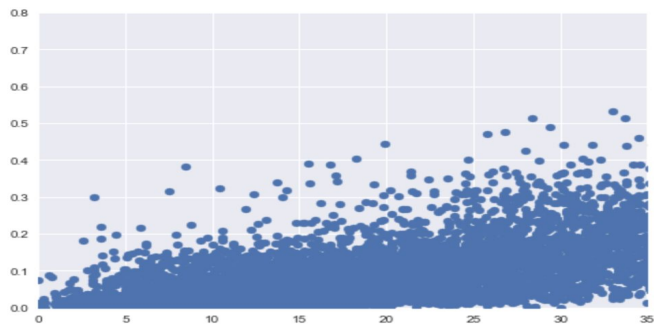
<matplotlib.axes.\_subplots.AxesSubplot at 0x121b78cc0>



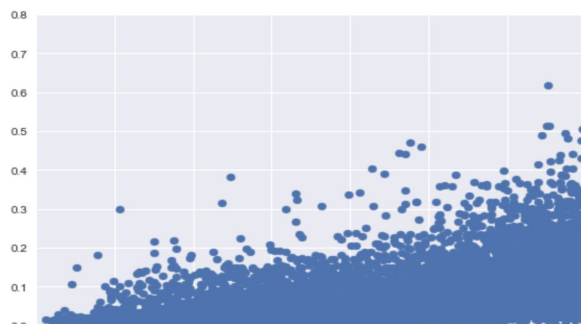
Salary Weight

# Data Descriptive Analysis

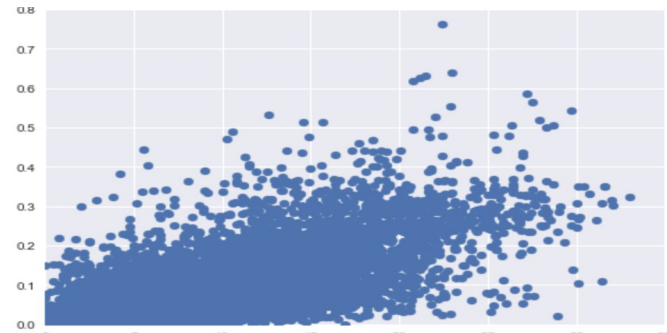
FP v.s. Salary\_Weight



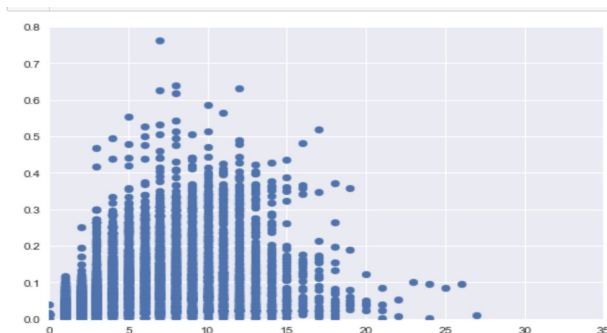
MIN v.s. Salary\_Weight



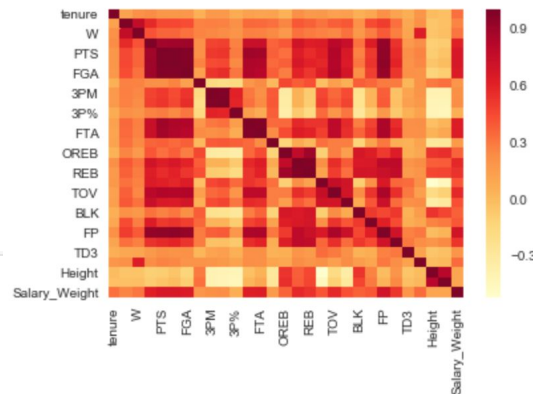
PTS v.s. Salary\_Weight



tenure v.s. Salary\_Weight



Correlation Plot



# Agenda

1. Problem Background and Data Sources
2. Data Cleaning and Preprocessing
3. Data Descriptive Analysis
4. **Assumption and Feature Models**
5. Machine Learning Algorithms
6. Text Mining process
7. Conclusion and Reflections

# Assumption:

Model	Assumption	Sample Feature
Baseline Model	The rationale of salary decision does not change over time	PTS, 3PM, W, ...
Auto-regression Model	Previous year salary might also serve as a reference of decision	Salary_I
Time-Lagged feature Model	Previous year performance and salary both contribute to the new decision	PTS_I1, 3PM_I1, W_I1, ...
Text-addition Model	Upon the best of previous models, public comments might capture some neglected information in salary decision	Pos, Neg, Joy, Fear, ...

# Agenda

1. Problem Background and Data Sources
2. Data Cleaning and Preprocessing
3. Data Descriptive Analysis
4. Assumption and Feature Models
- 5. Machine Learning Algorithms**
6. Text Mining process
7. Conclusion and Reflections

# Baseline Model: Preparations

## Preparation steps:

- Split Data into Training and Testing set: 80% training + 20% Testing
  - **80% data:** 10-fold CV to to compare models
- Create dummy variables for “Team”, “Position” and “Year” variables

## Trial on Linear Regression:

**Salary\_weight ~ PTS + W + MIN + ... + height + ... + team\_NYK + ... + Year\_2018 + ...**

- Adjusted R<sup>2</sup>: 0.58

Performance

Bio

Team

Year

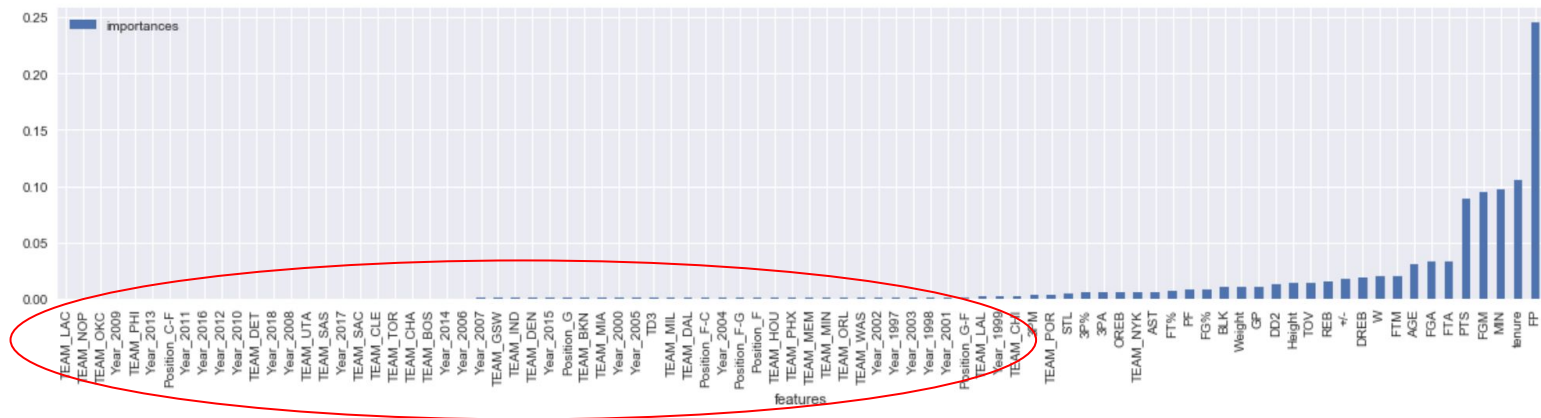
## Algorithms employed:

- Decision Tree & Random Forest & Bagging
- Gradient Boosting Machine & XGBoost
- Support Vector Machine



# Baseline Model: Feature selection and GridSearch

**Feature Selection:** inspect feature importance in **Random Forest** to drop some of the unimportant features



# Results across Models and Algorithms

-Unexpected after scaling

**Criteria for comparison:** CV Root Mean Squared Error (RMSE)

Algorithm	Baseline	Auto-Regression	TimeLag1	TimeLag2
Linear Regression	0.05822	0.04309	0.04303	0.04356
Decision Tree	0.06238	0.04633	0.04647	0.04775
Random Forest	0.05382	0.04165	0.04163	0.04312
Bagging	0.05451	0.04278	0.04816	0.04339
Gradient Boost Machine	0.05388	0.04201	0.04281	0.04337
XGBoost	0.05391	0.04222	0.04245	0.04319

**Final Model:** Random Forest with Auto-Regression Assumption

- Testing Data Performance: RMSE = 0.04221

# Agenda

1. Problem Background and Data Sources
2. Data Cleaning and Preprocessing
3. Data Descriptive Analysis
4. Assumption and Feature Models
5. Machine Learning Algorithms
- 6. Text Mining process**
7. Conclusion and Reflections






# Sentiment Analysis and feature generating

## Preparation Steps:

- Twitter api: Select netizens' comments on NBA active players and scrape them as texts
  - ~ 80000 tweets, total 540 players
- Data Cleaning:
  - Eliminate the players with scarce text (only a few comments in Twitter)
  - Eliminate duplicate text in each player's corpus

## Sentimental Analysis:

- Analysis using NRC data: divide words into two major categories (positive & negative) and 8 different sentiment types and add these features to our Machine Learning models

 James Harden  
 Anthony Davis  
 LeBron James  
 Damian Lillard  
 Giannis Antetokounmpo

```
import nltk
from nltk.corpus import PlaintextCorpusReader
tweets_root = excel_code
player_lst=[]
for i,player in enumerate(players):
    player_file = "{}.*".format(player)
    player_data = PlaintextCorpusReader(tweets_root,player_file)
    player_lst.append([player,player_data.raw()])
dff=comparative_emotion_analyzer(player_lst,object_name='player',print_output=False)
```

# Addition models with features from text

- Instead of using our original best model: Random Forest under Auto-Regression Assumption, we decided to apply **bagging method** under Auto-Regression Assumption to the the **new data set with sentiment features** due to a limit in the data size.
- We compared Root Mean Squared Error (RMSE) of original and text model, and the results are as follows:

Model	Original Model	Text model
Bagging	0.066073565	0.065481707

# Conclusions: Unexpected —————> Expected

## Conclusions: Break the stereotype, high performance doesn't mean bigger paychecks

- Indeed, there are a lot of real scenarios, for instance:
  - **Stephen Curry won the MVP** in 2014-2015 season, but he earned **only 10 million dollars**, at that time, the top salary was **20 million**.
  - Players may get injured resulting in a bad performance during the long-term contract. The Rookie Contracts' salaries are relatively low, but they can have awesome performance.
  - Teams sometimes sign a player by judging his potential, but the potential may fail to achieve.

## Limitations:

- Text Mining:
  - Limited time-span of text available
  - Limited Player mentioned available
  - fan's opinion might matter less than the professional sports critics
- Peak Year effect and free agent:
  - Free player compromise on a small package to wait for a peak year contract
- Business endorsement

# Appendix: Performance Stats Glossary

**GP** Games Played

**3P%** 3 Point Field Goals Percentage

**STL** Steals

**W** Wins

**FTM** Free Throws Made

**BLK** Blocks

**L** Losses

**FTA** Free Throws Attempted

**PF** Personal Fouls

**MIN** Minutes Played

**FT%** Free Throw Percentage

**FP** Fantasy Points

**FGM** Field Goals Made

**OREB** Offensive Rebounds

**DD2** Double doubles

**FGA** Field Goals Attempted

**DREB** Defensive Rebounds

**TD3** Triple doubles

**FG%** Field Goal Percentage

**REB** Rebounds

**PTS** Points

**3PM** 3 Point Field Goals Made

**AST** Assists

**+/-** Plus Minus

**3PA** 3 Point Field Goals Attempted

**TOV** Turnovers