JinShuo Chen

Final Report:

NBA Players' Salary Analysis

Problem Statement

The problem I want to solve is predict an NBA player's next year salary. Knowing how much the players will cost next year will help manager to plan the team's annual budget and understand the team cost. Team manager can also the information to benchmark with competitor teams and make sure that the play is paying fairly to avoid talent churn.

The dataset I am going to use is the Basketball Dataset from stats.nba.com. It contains information for more that 4500 NBA players.

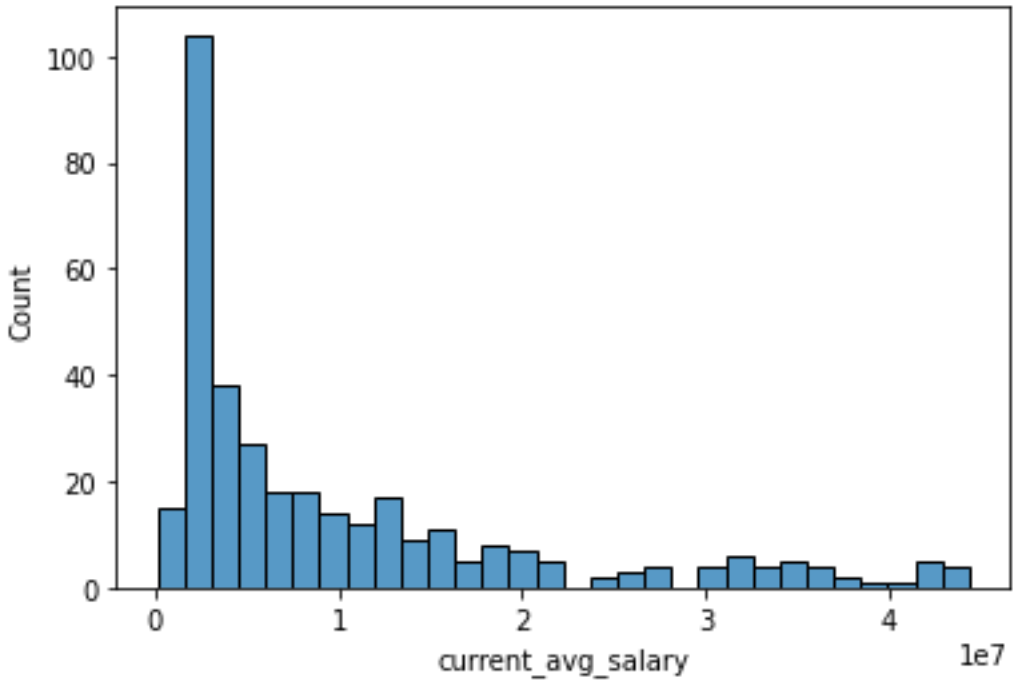
Data Wrangling

The original data set is SQLite database. It contains data on all games, all teams and all players within the NBA. To predict players', I am only interested in players' information which are the table "Player_Attributes" and table "Player_Salary". Table "Player_Attributes" has 4500 rows and 37 columns data. Each row is one player, and each column is the player's attributes such height, weight, season experience, average points, assists and rebounds. Table "Player_Salary" has 1293 rows and 12 columns data. Each row contains each player's current season salary and future seasons salary.

I started with creating SQLite engine and query data by selecting 'Player_Attributes' table and 'Player_Salary' table using inner join to combine them into one table by matching the palyer's names. The combined table has 4500 rows and 40 columns. Then, I dropped the columns that have a lot of missing columns and do not relate the salary. After that, I convert player's birth to age, then drop the birthday column. I then filled NAN using forward fill. Finally, the clean data has total 18 columns that contains all potential salary related player attributes and player salary.

Data Exploratory Analysis

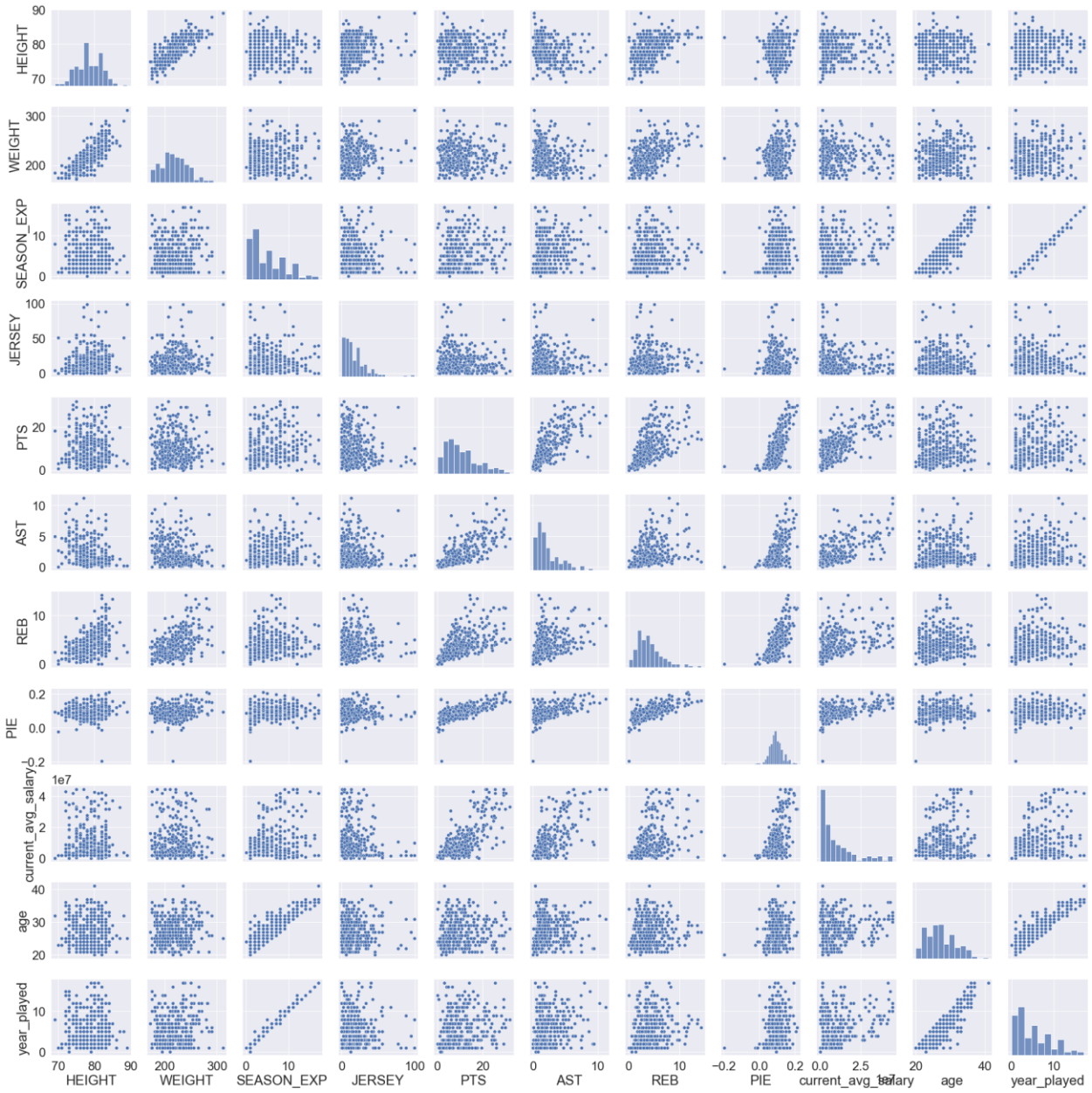
First, I plotted histogram the salary to see the distribution



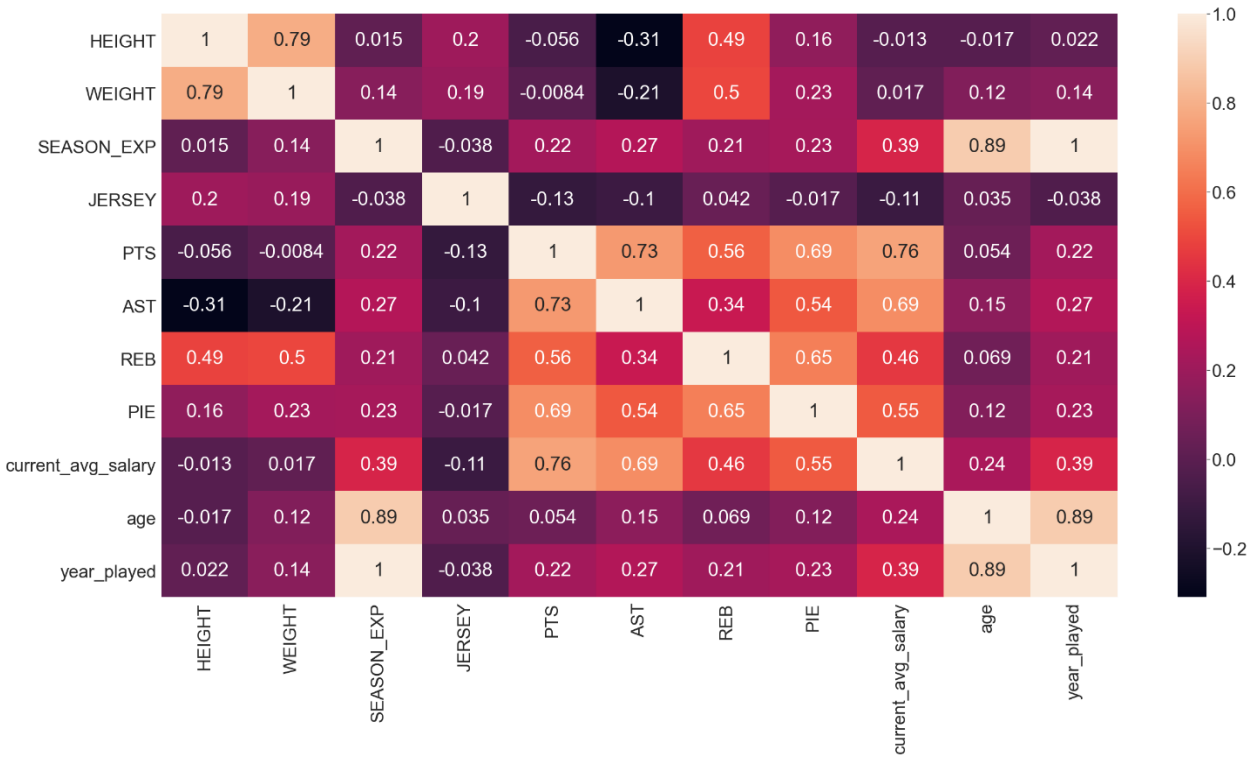
As we can see the salary distribution is non-normal distributed which make sensor by knowing the start players have more high salary then non-start player. The average salary is about 10 million, but the median salary is only about 5.6 million. The standard deviation is about 10.8 million

```
count    3.530000e+02
mean     1.009223e+07
std      1.084653e+07
min      9.902000e+04
25%      1.842959e+06
50%      5.655148e+06
75%      1.366667e+07
max      4.439366e+07
```

Next, I make a pair plot of the data the to see the correlation between the columns.

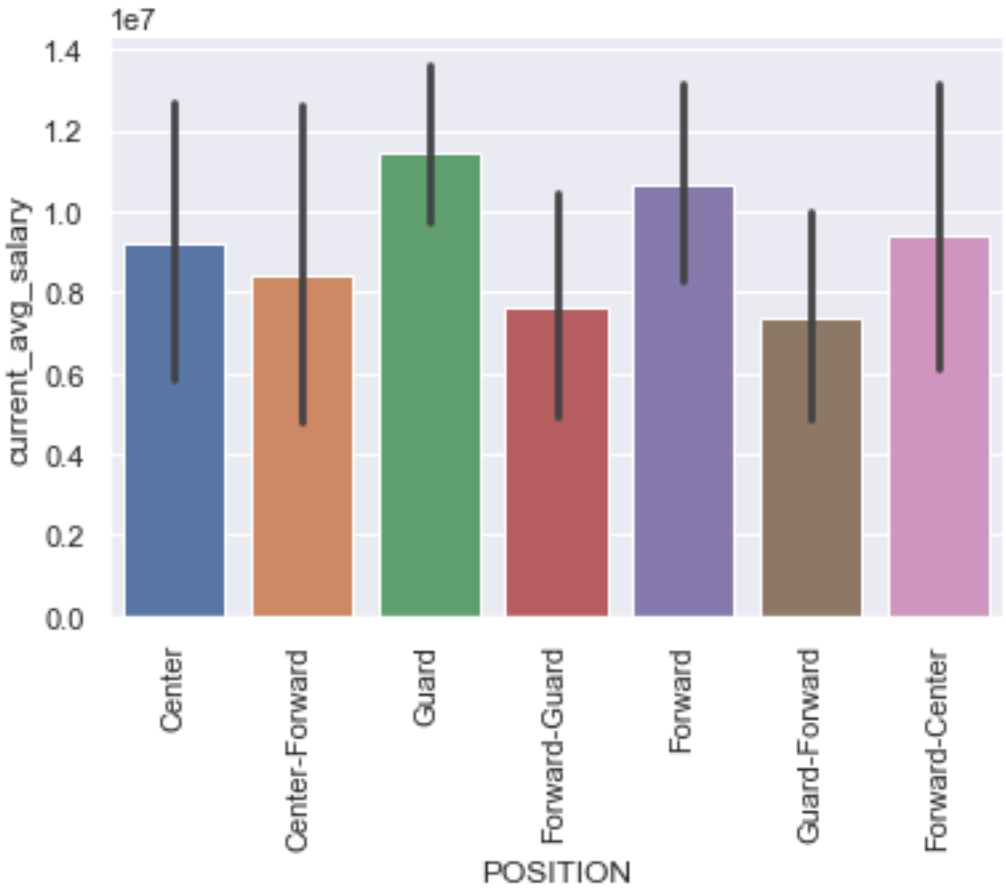


I found that PTS (points), AST (assist), REB (rebound) and PIE (player impact estimate) have strong correlation with salary. And it is interesting that height and weight seem do not have correlation with salary. One reason could be NBA players have similar height and weight at each position.

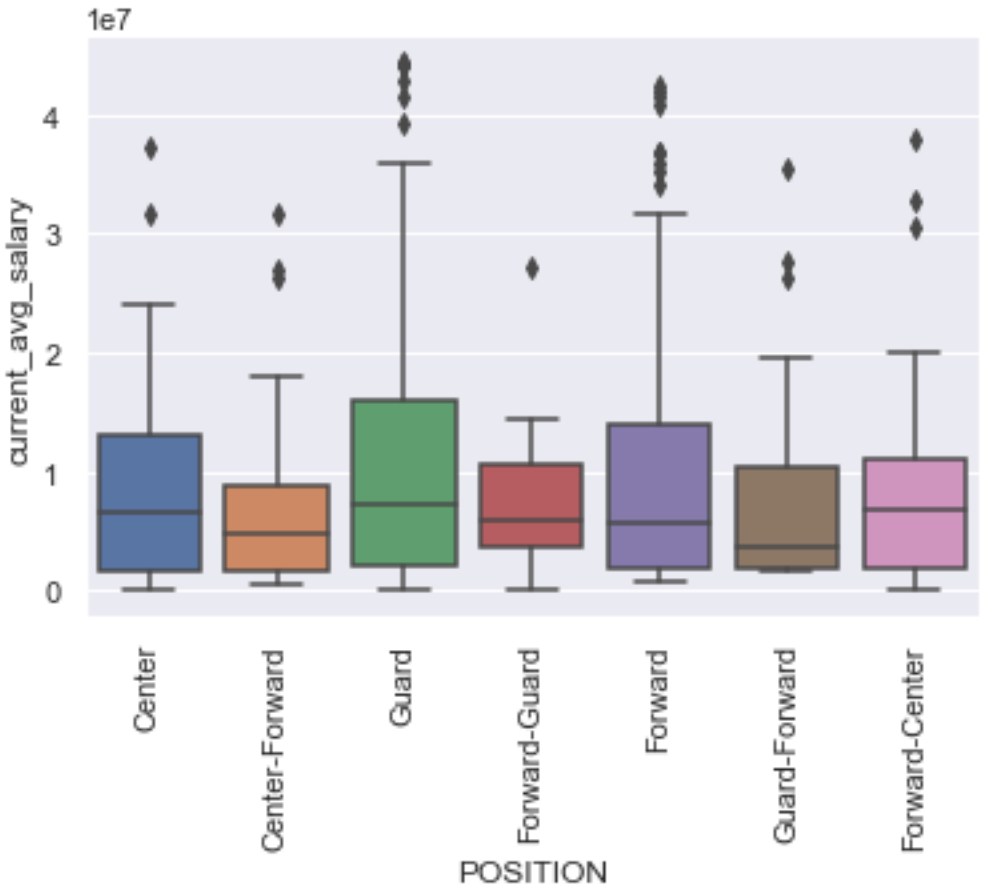


From heat map, I see PTS has the highest correlation with salary.

After exploring the numerical values, I then explore the categorical values. The first categorical value I explored is position.

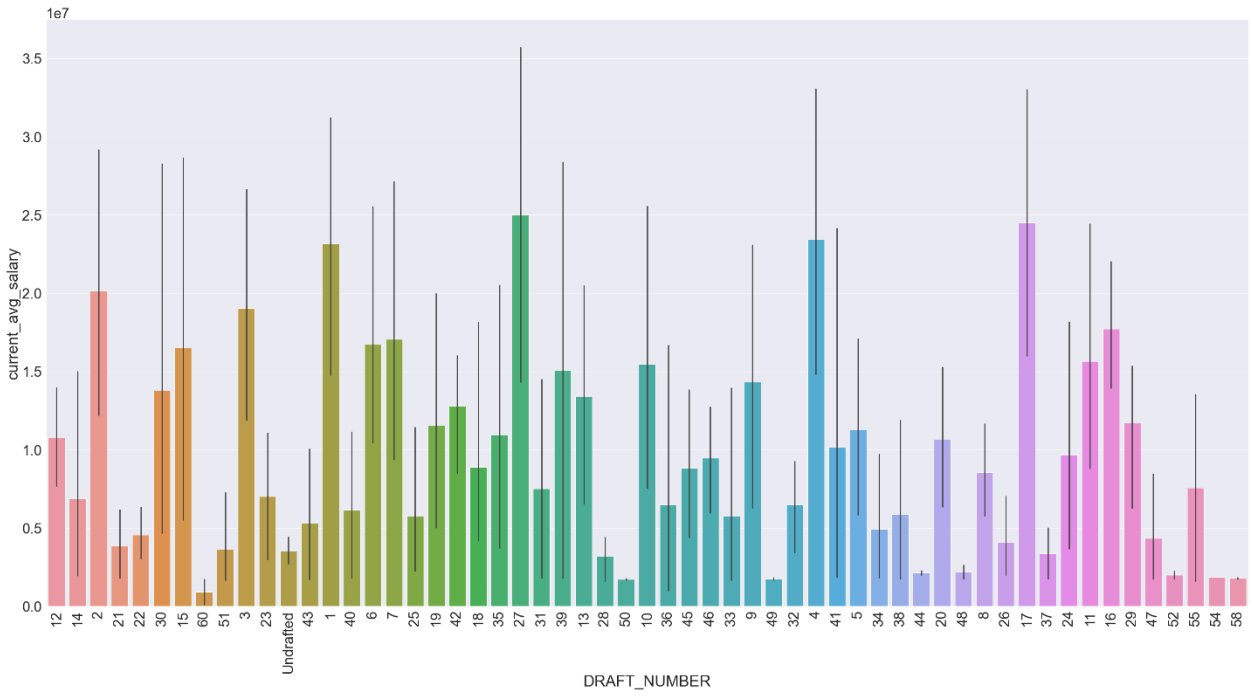


I found guard has the highest average salary. Then I created box plot to see if there are any outliner.



OK, guard have the biggest outliners.

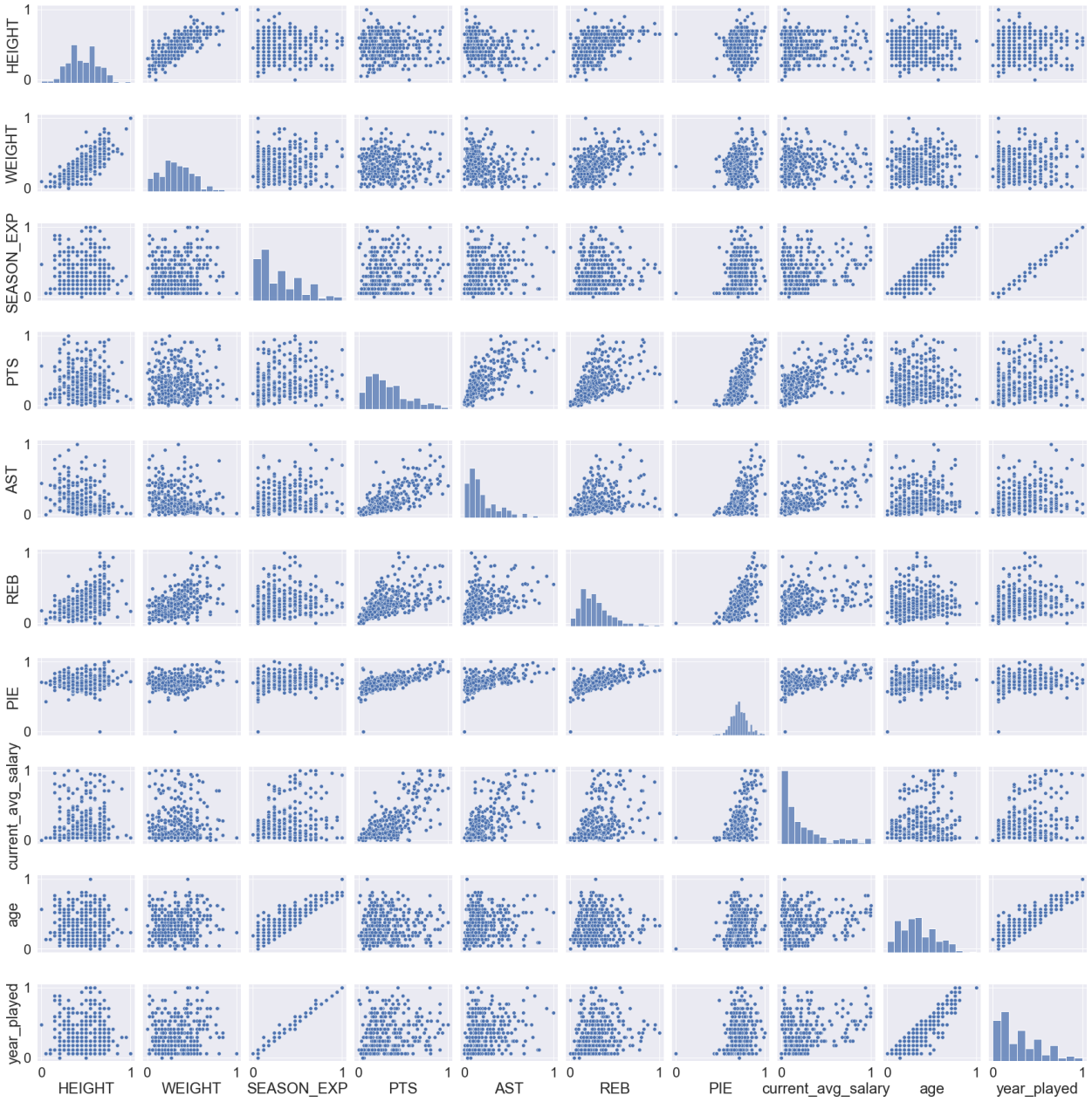
The second categorical value I explored is draft number. Typically, the player has draft number 1 is the best player.



I found the first four draft number 1-4 have very high salary, but 5 is relatively low. And I found it is very interesting that draft number 17 and 27 are the highest.

Pre-processing and Training Data Development

First step, since I am going to use regression models to analyze the data, I created dummy features for all the categorical values using 'get_dummies'. There are too many schools. I only created dummy features for schools that have more than 6 players, and use 'Other' columns to represent other. The total columns became 190. Second step, since the scales for numerical values are different and the values are all positive. I used 'MinMaxScaler' to standardize the magnitude of numeric features. After scaling the data, I made a pair plot of the data to check if there are any unwanted changes



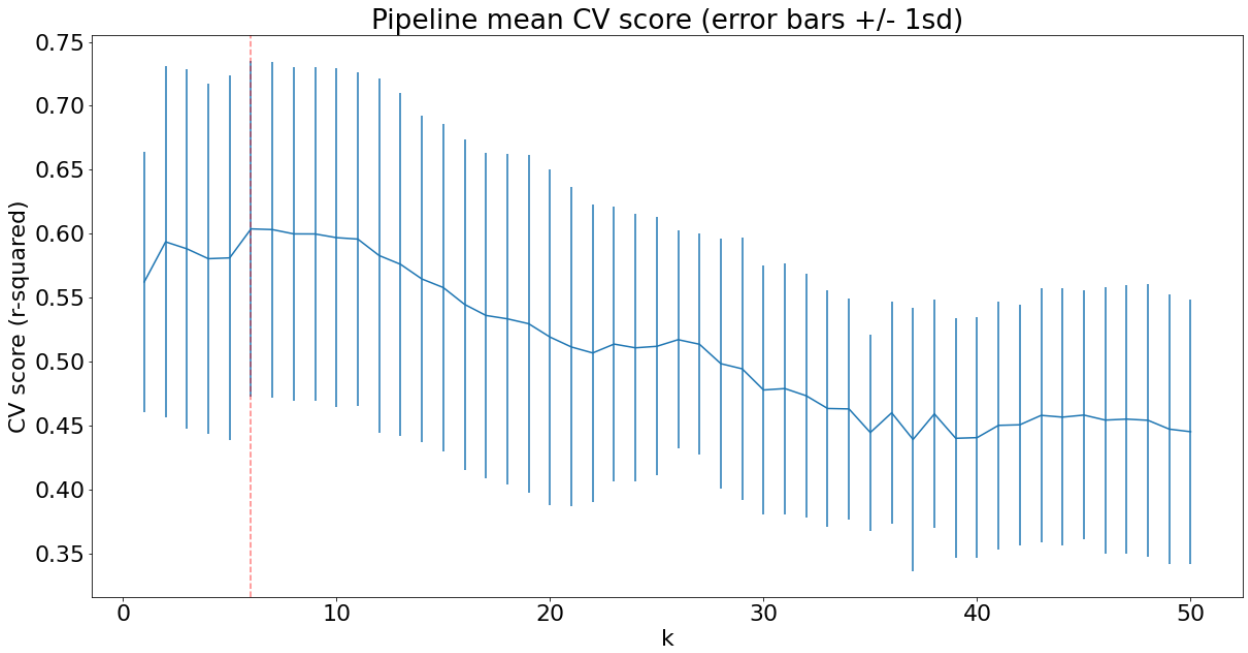
The plots look the same as before scaled. And then, I spared the data into training set, and test set. I used 20% of the data for test.

Model Selection

I tested 3 different machine learning regression models: Linear Regression, Random Forest Regressor and KNN regression. The metric I am going to evaluate the models was mean absolute error.

For the linear Regression, I first fit it with all the features. But it turned out overfitting. The mean absoluter error for train set is 0.0077, but for the test set is 4.7e+19. The evaluation metric shows the model doing good to predict train data, but very poor on the test data. It is clearly overfitting. The model needs to reduce features. To do that, I created a pipeline including 'Seleckbest' as a step. Then, I used

grid search cross validation to get the best K is 6 which means use the top 6 features to fit model will be the best.



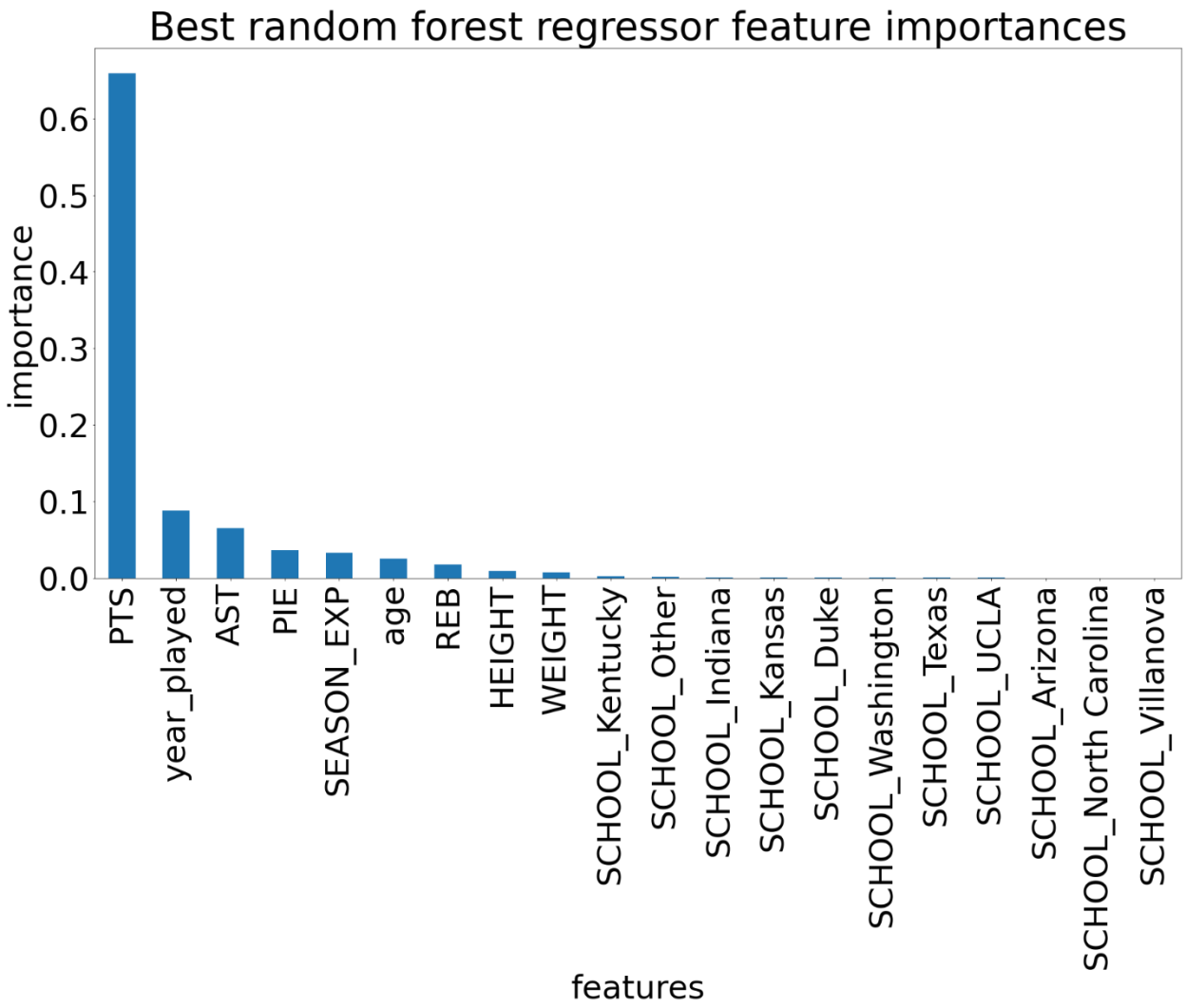
The above suggests a good value for k is 6, but there was decline with k, follow by a rapid increase with k. After the best k, there is a very slow decline. It is interesting that the variance of the results does not change much as k increased.

Which features were most useful? Step into best model. I got the following:

```
PTS              0.579280
AST              0.364178
year_played      0.204195
REB              0.058857
DRAFT_ROUND_1    0.024058
PIE             -0.108458
```

The results suggest that PTS (total number of points) is the biggest positive feature. Thus, makes intuitive sense. Player who has higher total number of points get pays more.

The second model I tested was Random Forest regression. The best params I found the model is max depth is 5 and n_estimators is 112.



Total number of points is the most importance feature same as what linear model suggest. However, the second most importance feature is year played instead of Assist to turnover ratio.

The last model I tested was the KNN regression model. The best parameter I found for the model is 13 neighbors and uniform weights.

Compared those three models, Random Forest regression model has the best performance. It has mean absolute error 0.0794 which scaled it back to salary is about %3.5 million. Given that the average salary is about 10 million, the error is about 35%. It is not the best model, but it is in the reasonable range. It would be still useful to predict salary player that are more than 30 million.

Future Research

The regression models seem did not fit the data very well. In the future, I would like tested categorical model which I will separate the salary into difference categorical such lower, below average, average, above average, high and super high. In addition, I will like to search for more data such as the last ten years salary of each player along with they are performance like PTS, AST, and REB for each year

For the dataset it only contains the player’ current salary and overall performance data which I think not enough.