STA 160 Final Report: Batting Average Prediction using Machine Learning Techniques

Audrey Heng, Brandon Lu, Hansull Joh, Louise Li, Aria Hamidi
University of California, Davis - Statistics Department
auheng@ucdavis.edu, btlu@ucdavis.edu, hsjoh@ucdavis.edu, yxyli@ucdavis.edu, hamidi@ucdavis.edu

Abstract -

The object of baseball is to score the most homeruns, and as such, it is no surprise that a key aspect is maximizing the number of home runs a team can achieve. In baseball, being at-bat is synonymous with a team being on the offense which is when a team's players go one-by-one to home plate and take their turn to bat. When a team is on offense, they have the opportunity to score runs and put their team in the lead since the three main goals of batters are to become a baserunner, to drive runners to home plate, or to advance runners along the bases for others to drive to home plate. Information about the batters is collected to calculate the batting average, which is the percentage a hitter gets a hit to get on base at-bat they take.

This research study aims to develop a predictive model that will help determine a player's added value to a major league baseball team through the prediction of their batting average. This in turn will allow teams to determine which players are better at batting and will be best in assisting the team in winning games. This information will also allow teams to better prepare the batting order for games because it is a predetermined list of players stating who bat and when. This study was based on data that was web scraped from ESPN's website regarding the batting stats 2021 for each MLB team. The results that we obtained were from the classification models including K Nearest Neighbor (KNN), XGBoost, and Random Forest from which we determined that XGBoost and Random Forest produced similar results.

I. INTRODUCTION

Major League Baseball is a professional baseball organization as well as the oldest major professional sport league in the world. As of 2022, the league consists of a total of 30 teams with 15 teams in the National League and 15 in the American League. It is a national sport that is watched by about 11.75 million people and generates approximately $10 billion in revenue with the average revenue being $318.53 million per MLB team. According to Forbes, Major League Baseball has the potential to make $11 billion in revenue in 2022.

The world series, in baseball, is the annual postseason play-off series between the champions of the two major professional baseball leagues which are the American League and the National League. The series consists of 162 games in the regular season, preceded by an eight-team divisional playoff and a championship series that is used to determine the best team for that year. With the information we collected, we are able to determine that baseball is a lucrative sport and we are given the opportunity to find ways to predict a players batting average then use that information to determine if a team is going to make it to the world series.

To predict the best performing professional baseball team, we used the data collected from the 162 games for each MLB player. The goal of the study is to build predictive models for batting averages using the data that was web scraped from ESPN by applying machine learning techniques including Random Forest, XGBoost, and K Nearest Neighbor (KNN).

II. METHOD

A. Software

The programming language Python was used for preprocessing the dataset and machine learning techniques. Also, in order to improve data modification and visualization, Jupyter Notebook platform was used. The benefits of using Python for this study is that it is a high-level programming language that can utilize many different tools and libraries to perform the necessary data analytics techniques.[1]

B. Web Scraping

We used the request library in order to pull data from the ESPN dataset on baseball players batting statistics. For web scraping, we selected the 2021 season and batting position, and we collected the data of all batters from all MLB teams. We first collected data individually from each team and then concatenated the data from all teams into one dataframe called “Batting”.

C. Dataset

The dataset used for data processing was web scraped from ESPN using information from the 30 Major League Baseball teams. It contains 19 different variables that factor into the batting average which are shown below along with its values in Figure 1. Since the goal of our project is to predict who are the

best 10 batters in the upcoming season, we decided to remove the players who played less than 50 games. Also, The dataset was cleaned after web scraping because there were no missing values.

Figure 1. Data Demography

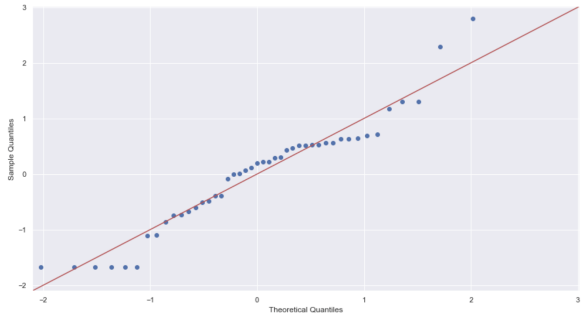
Feature	*Values*	*Features*	*Values*
Response Variable:		**3B:** Triples	0 - 8
AVG: Batting Average	0 - 1	**HR:** Home Runs	0 - 48
		RBI: Runs Batted in	0 - 121
Independent Variable:		**TB:** Total Bases	0 - 363
Name: individual player's name	Categorical	**BB:** Walks	0 - 145
Team: MLB Team Name	Categorical	**SO:** Strikeouts	0 - 202
GP: Games Played	1 - 162	**SB:** Stolen Bases	0 - 40
AB: At Bats	0 - 664	**OBP:** On Base Percentage	0 - 1
R: Runs	0 - 123	**SLG:** Slugging Percentage	0 - 1.33
H: Hits	0 - 191	**OPS:** OPB Pct + SLG Pct	0 - 2
2B: Doubles	0 - 42	**WAR:** Wins Above Replacement	-2.6 - 7.2

III. RESULTS

A. Data Visualization

For better understanding of the Batting standing of players of each team we used Python to sketch GGplot and Histograms of each team based their player's AVG:

Figure 2. Arizona DiamondBack



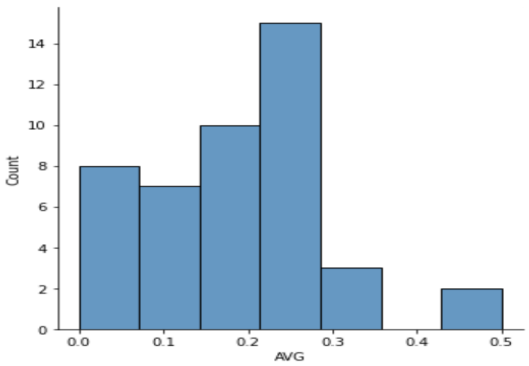


Figure 3. Atlanta Brave

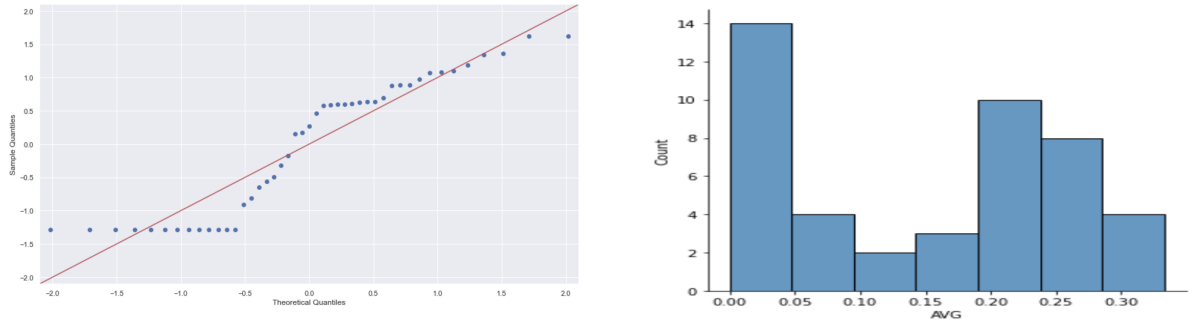


Figure 4. Baltimore Orioles

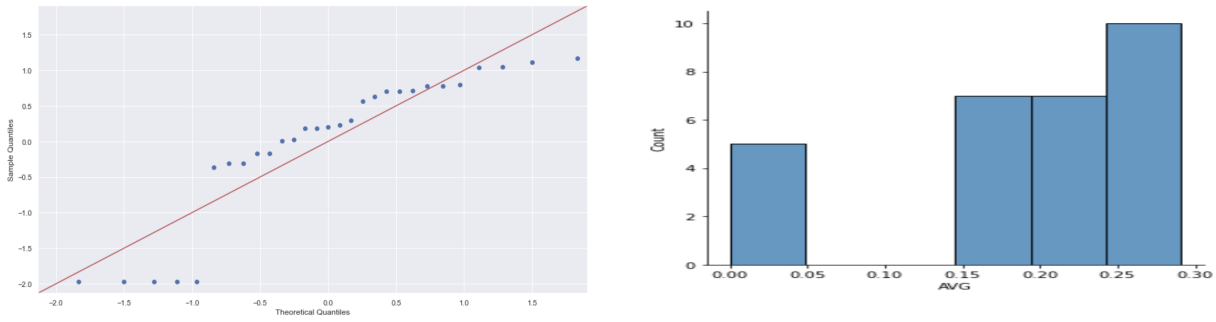


Figure 5. Boston Red Sox

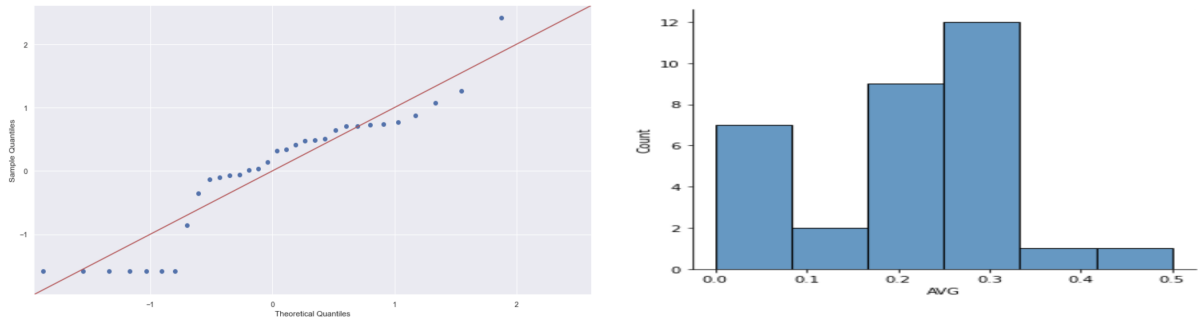


Figure 6. Chicago White Sox

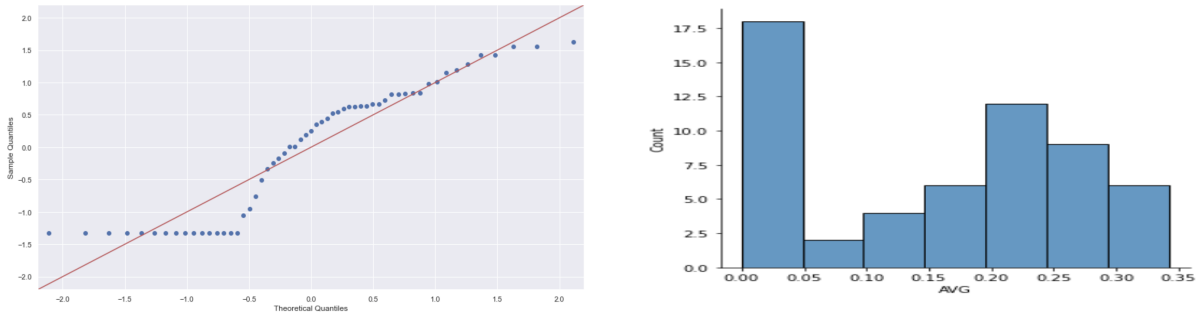


Figure 7. Cincinnati Reds

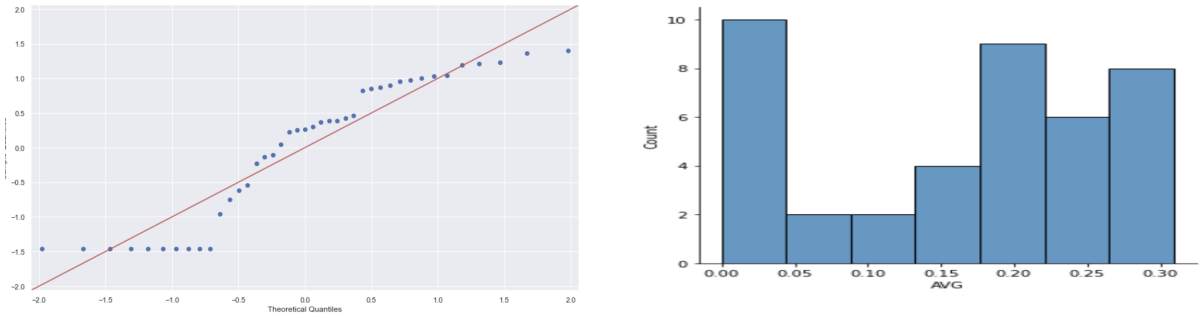


Figure 8. Cleveland Guardians

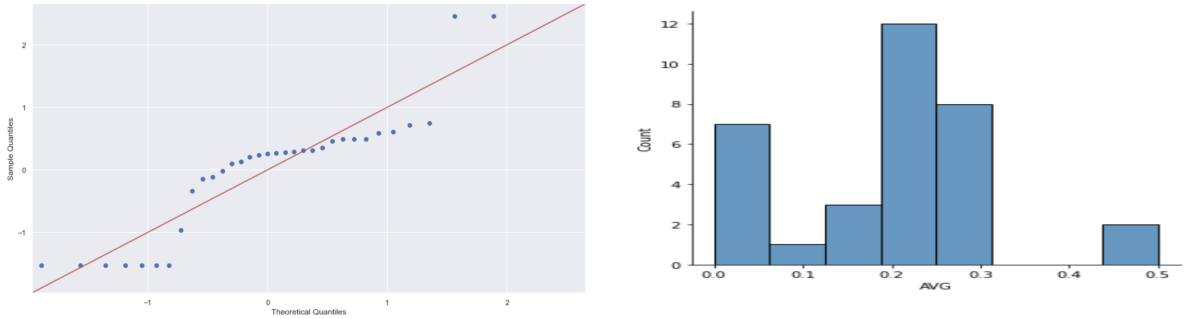


Figure 9. Colorado Rockies

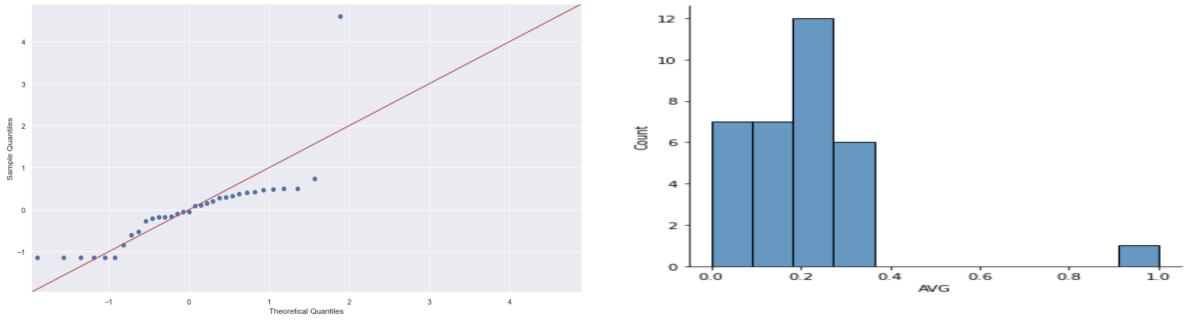


Figure 10. Detroit Tigers

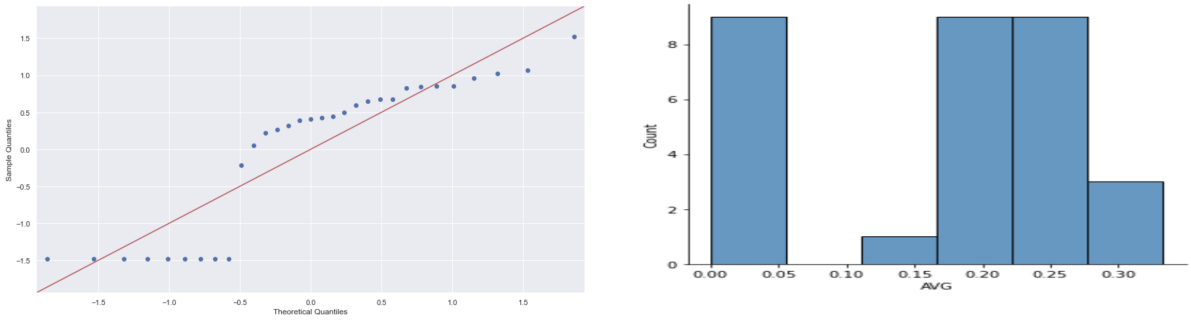


Figure 11. Houston Astros

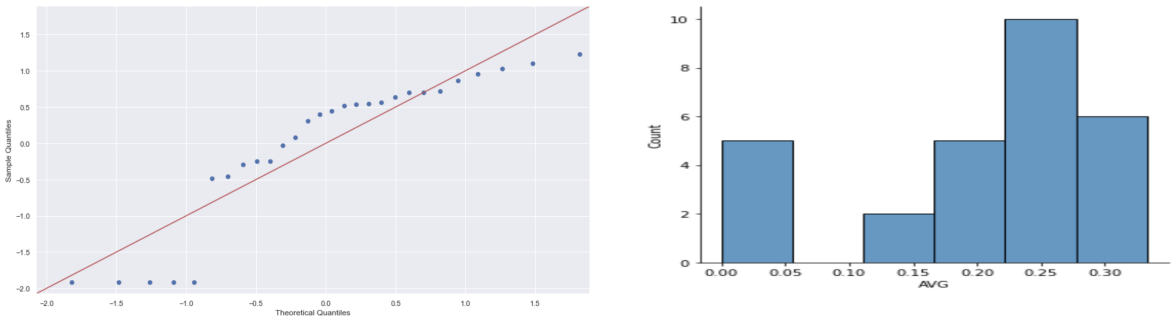


Figure 12. Kansas City Royals

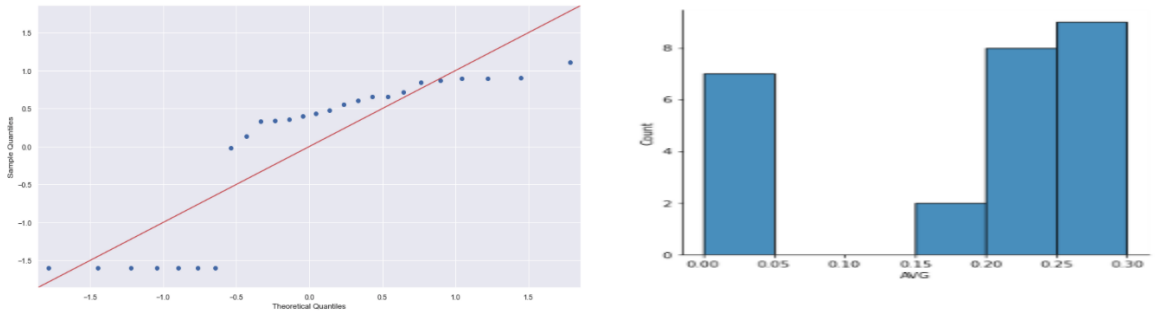


Figure 13. Los Angeles Angels

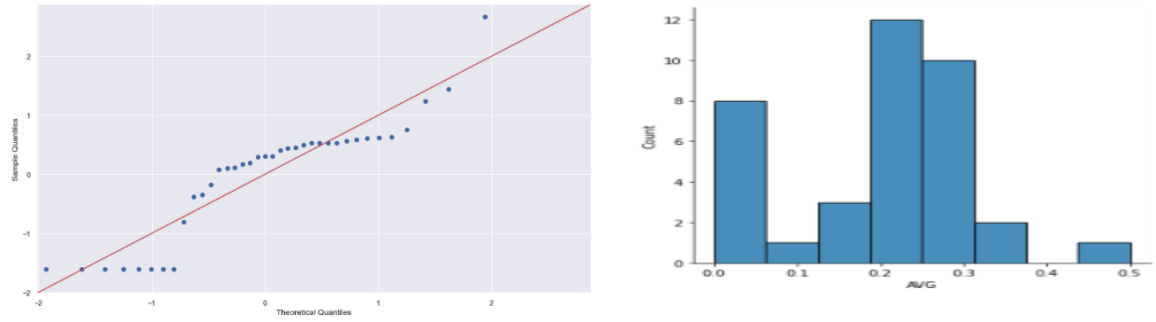


Figure 14. Los Angeles Dodgers

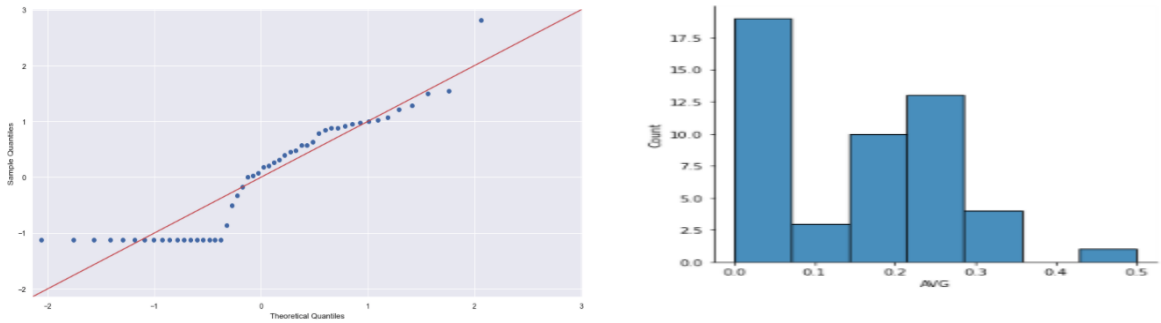


Figure 15. Miami Marlins

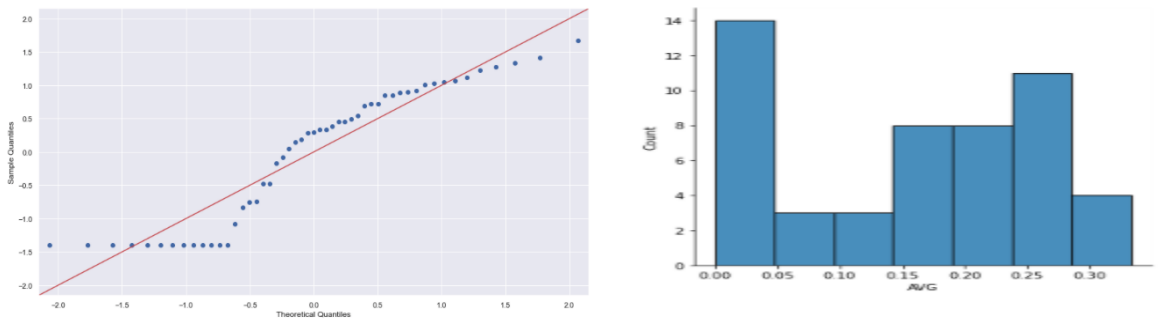


Figure 16. Milwaukee Brewers

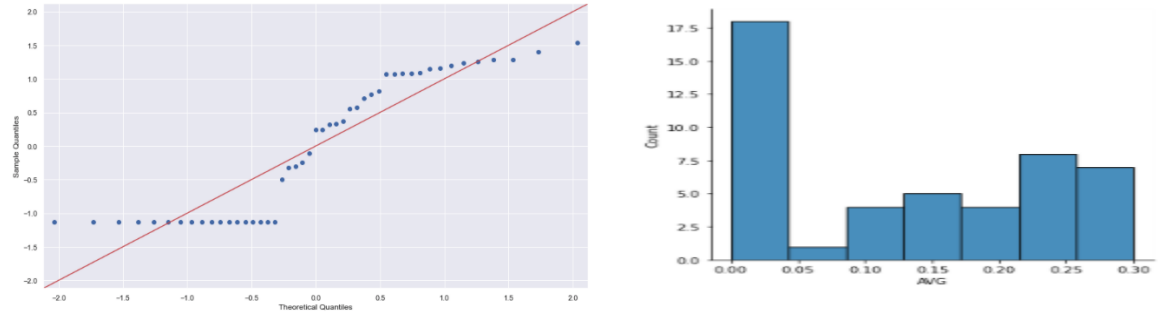


Figure 17. Minnesota Twins

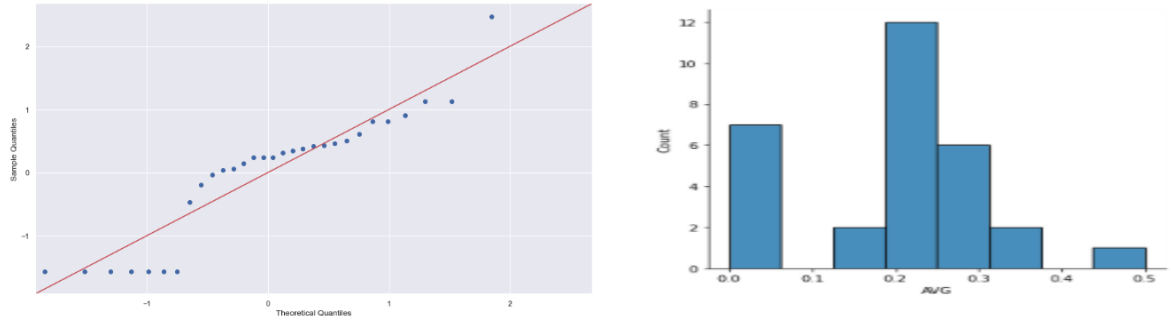


Figure 18. New York Mets

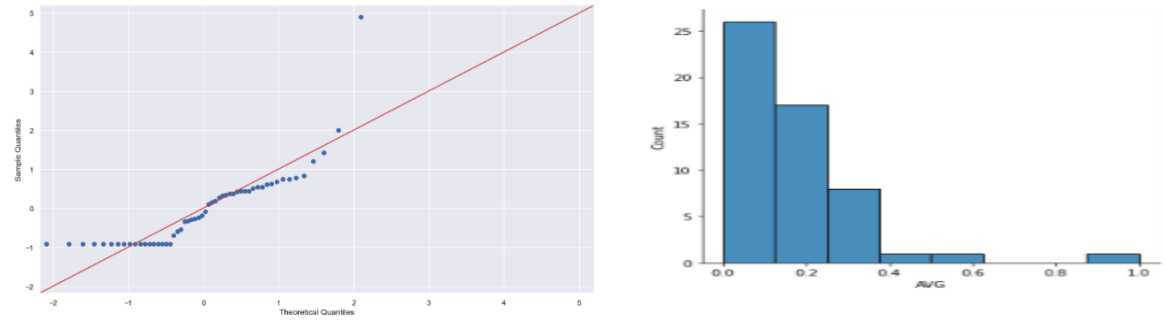


Figure 19. New York Yankees

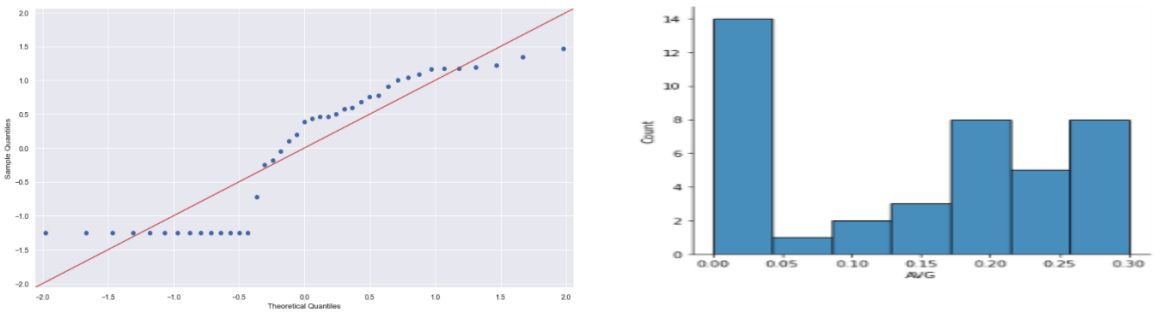


Figure 20. Oakland Athletics

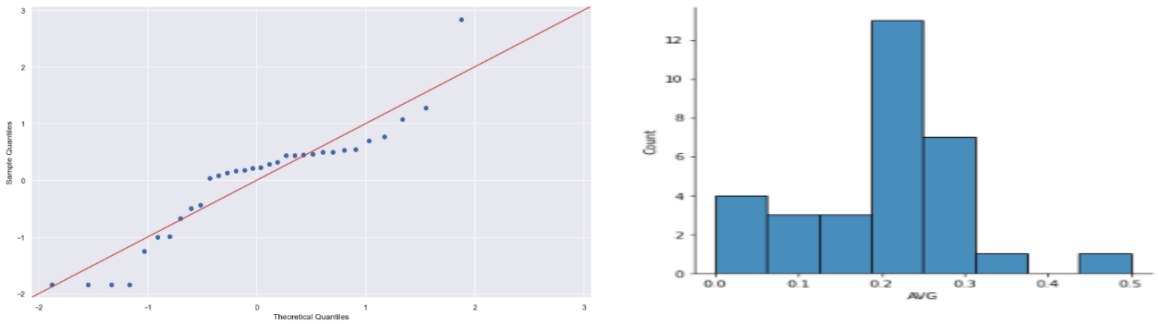


Figure 21. Philadelphia Phillies

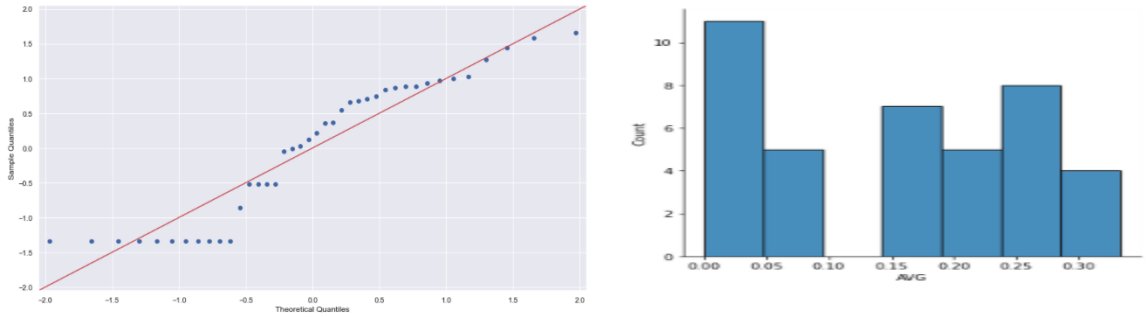


Figure 22. Pittsburgh Pirates

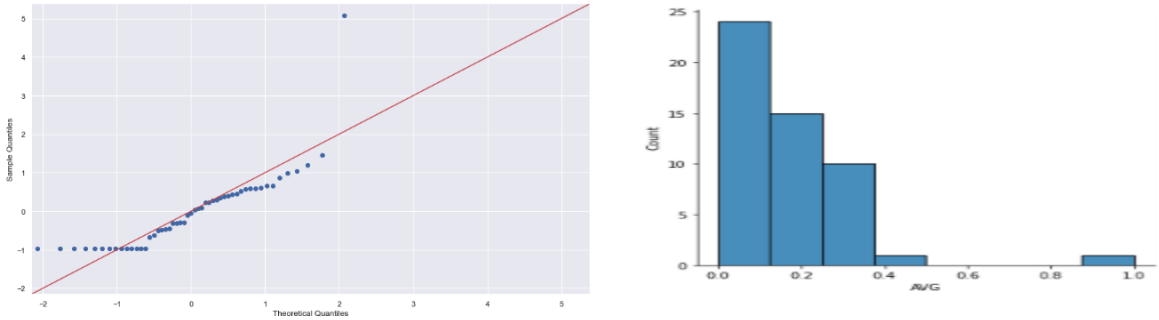
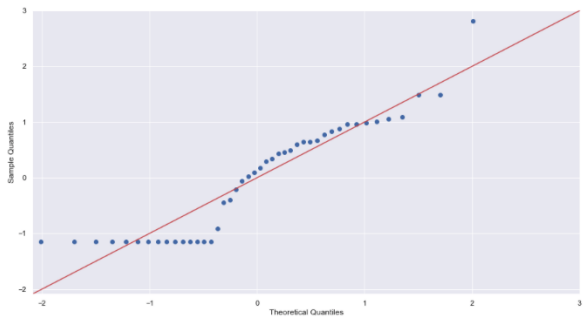


Figure 23. San Diego Padres



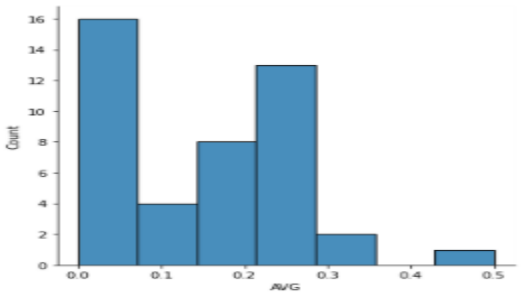
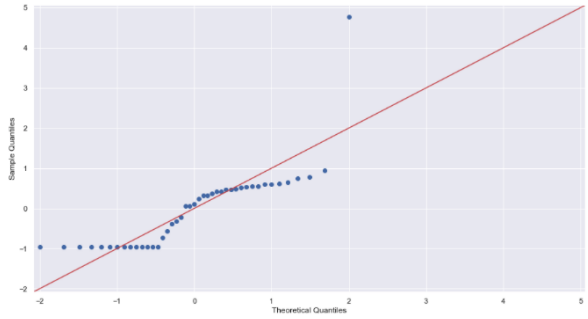


Figure 24. San Francisco Giants



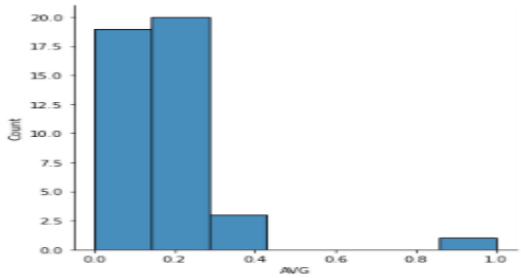
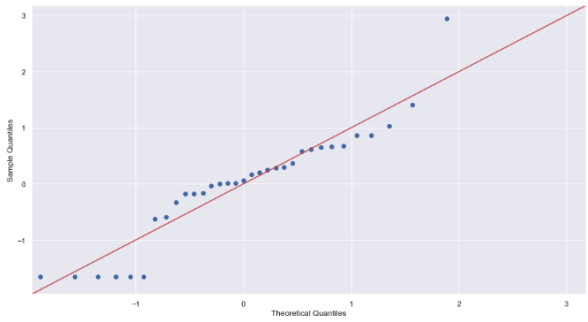


Figure 25. Seattle Mariners



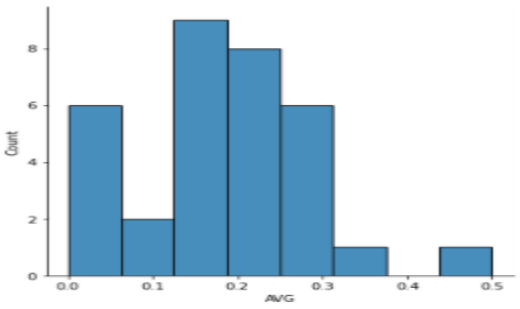
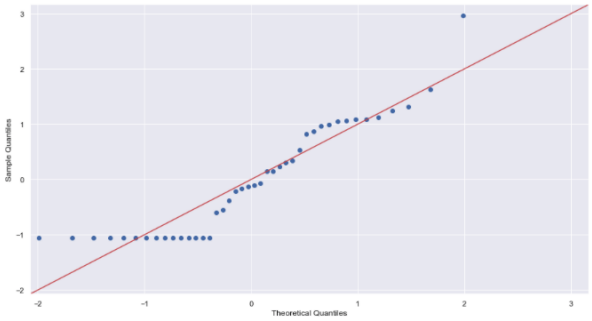


Figure 26. St. Louis Cardinals



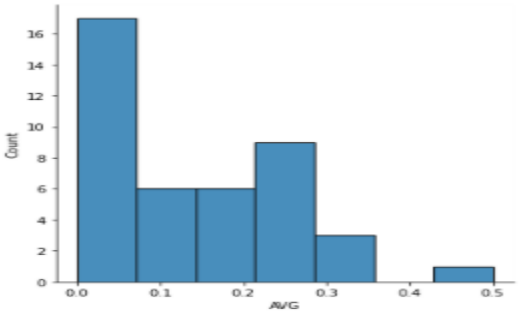


Figure 27. Tampa Bay Rays

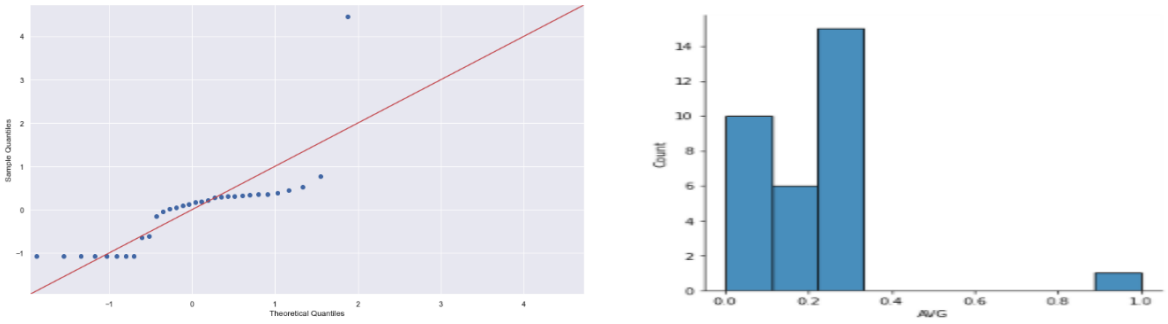


Figure 28. Texas Rangers

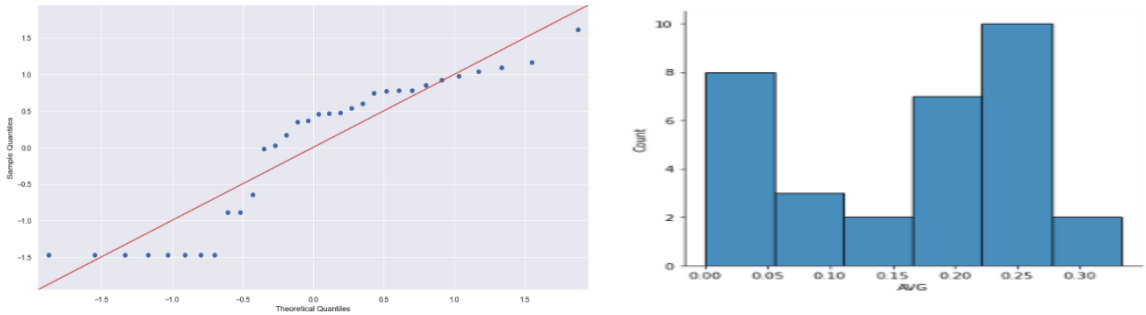


Figure 29. Toronto Blue Jays

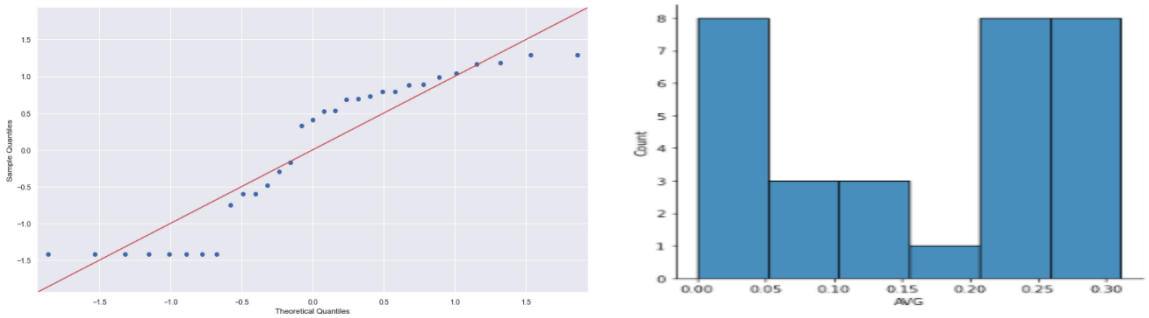


Figure 30. Washington Nationals

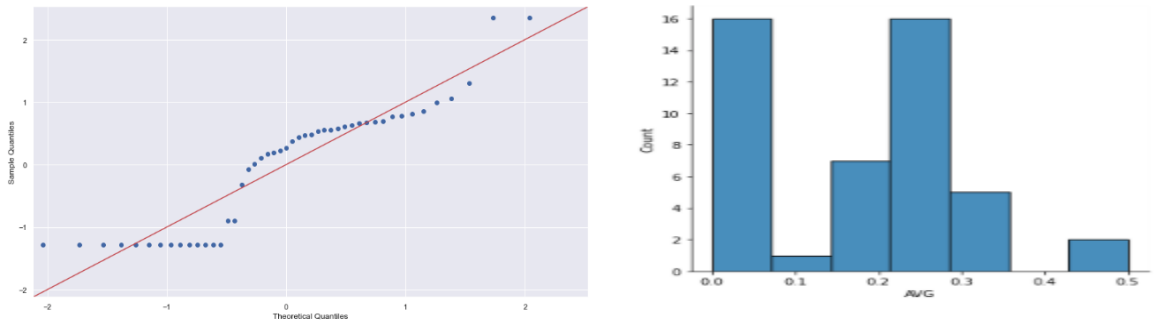
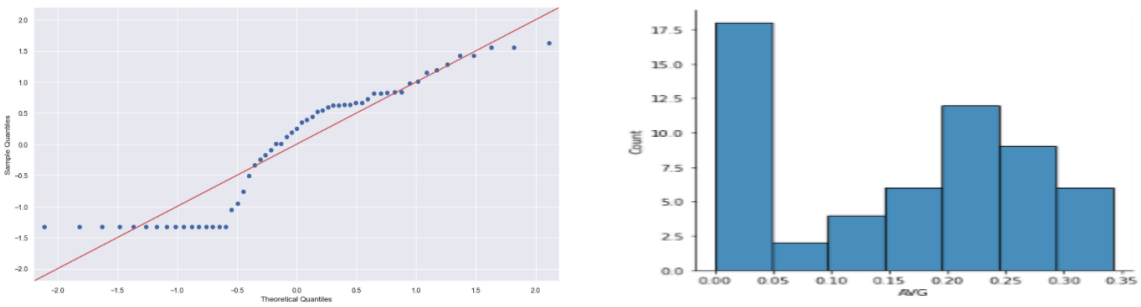


Figure 31. Chicago Cubs



Furthermore, we also decided to combine all the data in the plots above and illustrate them as one plot to show the distribution of the team's AVG and compare them easier before doing the feature selection. Here's the plot:

Figure 32-1. Distribution of Team's AVG without dropping samples

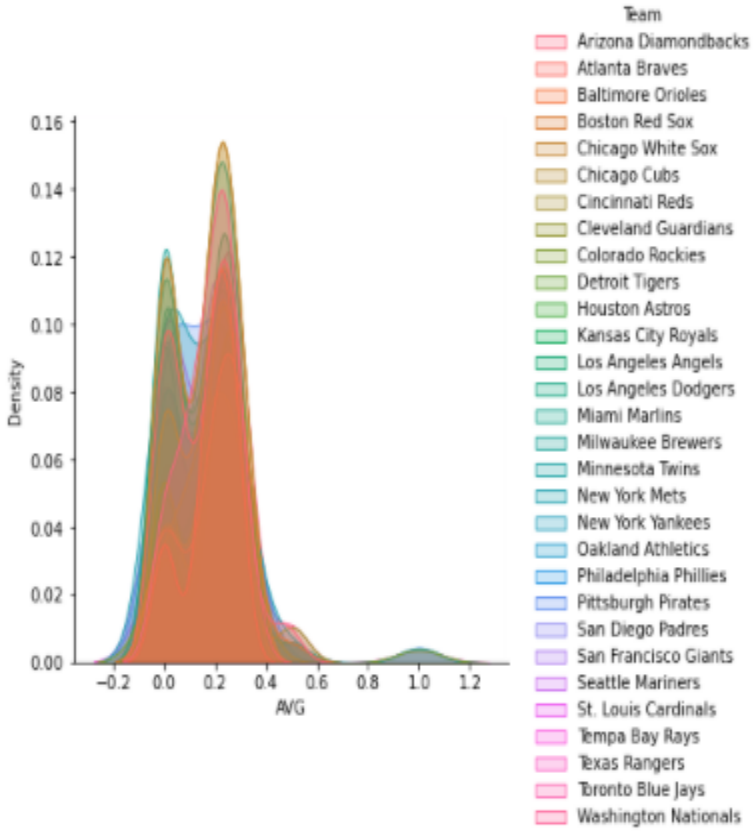
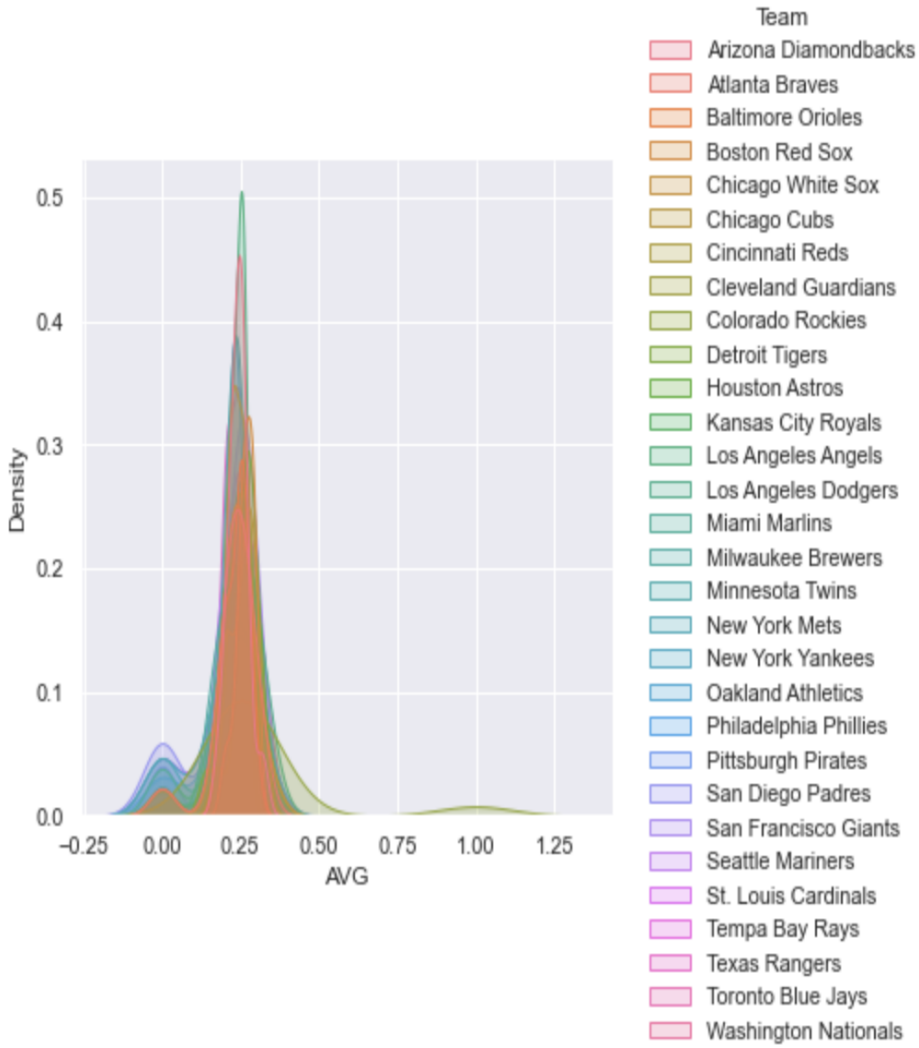
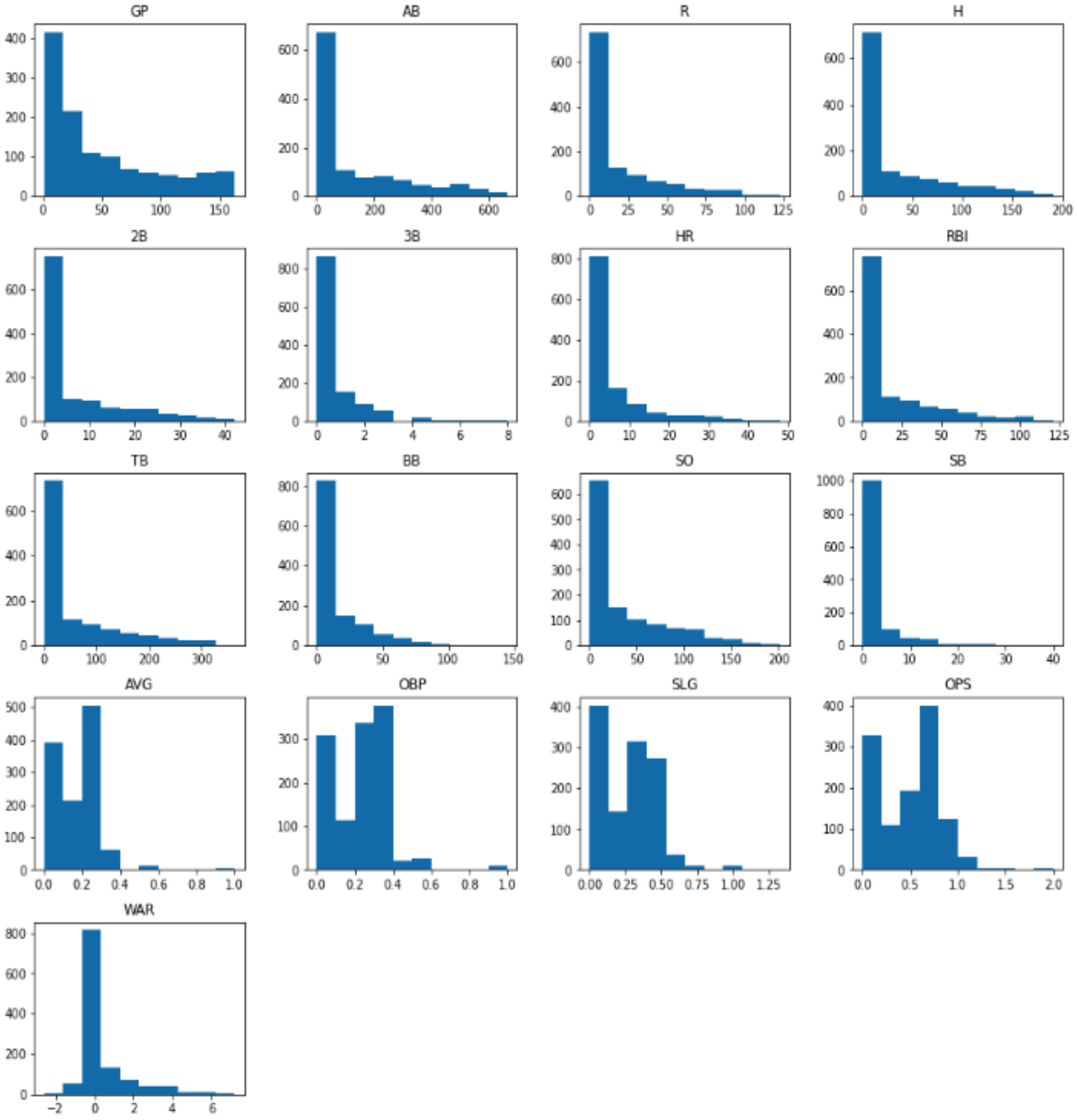


Figure 32-2. Distribution of Team's AVG after choosing player GP>=50



We can see that the initial AVG scores including for different teams have two peaks (though not symmetric), after dropping players with GP lower than 50, the distribution looks nicer and more stable. There are still some players with 0 batting score but most of the AVG scores are steady around 0.25. It is also a little bit right skewed with some outliers.

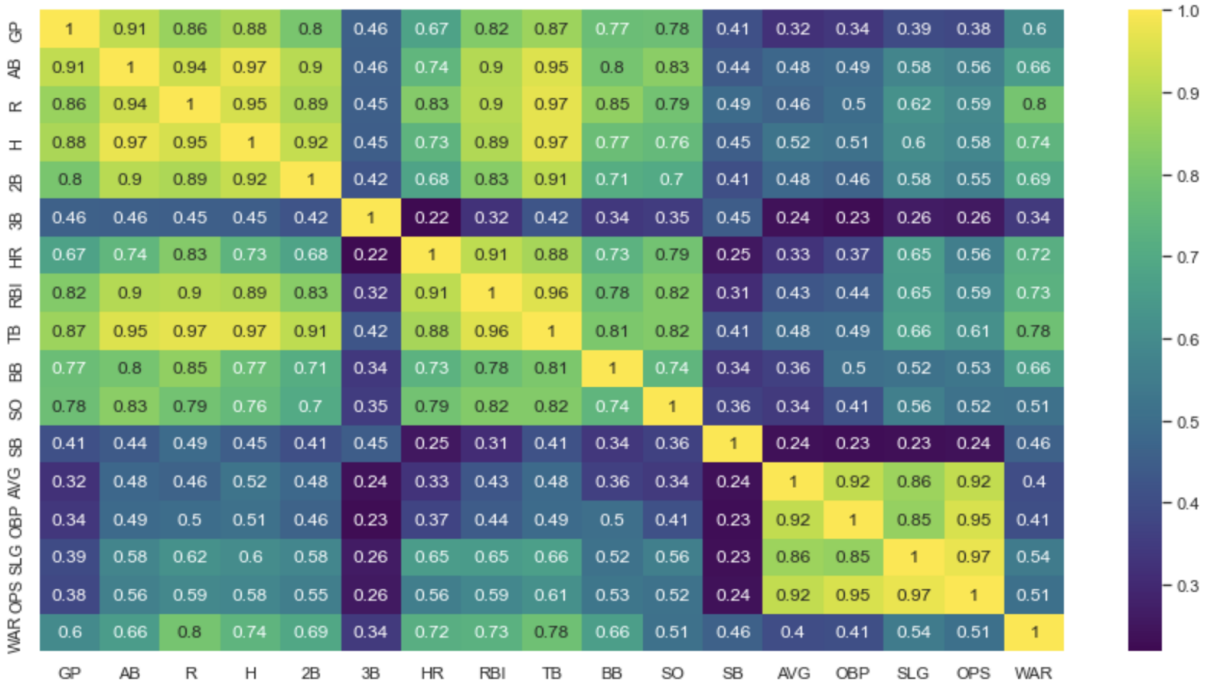
Figure 33. Histograms of Every Variable



B. Feature Selection

For this step, we generated a correlation matrix. We found the correlations between all variables, and we excluded all variables that held a correlation below a certain threshold. We used the heatmap in order to make the correlations more readable and observable by using different colors.

Figure 34. Correlation Matrix for all players from all teams before dropping



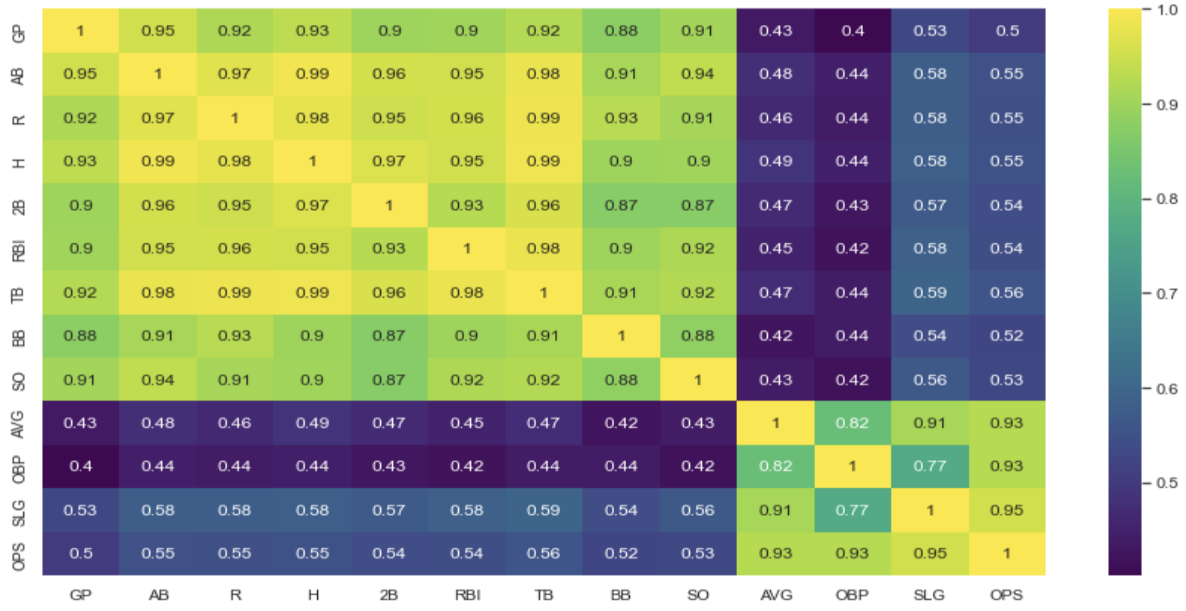
We find out the correlation matrix follows an interesting pattern: there are separate blocks that make up the whole matrix, and the division lines are also obvious. In the first block, variables Game Played, At Bats, Runs, Hits, and Doubles share high correlation values while Triples does not show a correlation with all of the other variables. In the following blocks, Home Runs, Runs Batted in, Total Bases, Walks, and Strikeouts also show high correlation to the previous variables. We can also see that the Stolen Bases make another division within variables. However, the following variables Batting Average, On Base Percentage, Slugging Percentage, and OBP+SLG Percentage have a high correlation within the group but it seems like they do not have a high correlation to the other variables. As our inputs are numerical and our predicted value AVG is also numerical, we decide to drop Triples and Stolen Bases and not use them to predict the AVG scores. As for WAR value, its correlation to other variables especially to AVG is relatively low so we also drop it.

Pearson's Correlation Test:

Since the input and output of the whole feature selection process is numerical, we decided to use Spearman's correlation test which measures the direction and monotonic association between two variables. For the correlation test, we selected 0.4 to be our threshold limitation coefficient and we used that threshold to keep features which have higher than 0.4 correlation coefficient. Ultimately, we came up with the relevant features which are listed in the table below:

Figure 35. Correlation Table	
Variables	**Correlation Coefficients**
OPS	0.919761
OBP	0.919470
SLG	0.862728
H	0.523903
TB	0.485000
2B	0.480243
AB	0.477137
R	0.464911
RBI	0.429397

Figure 36. Correlation Matrix for all players from all teams after dropping



After conducting the feature selection, we plotted the correlation heatmap again and you can observe that above and see the difference between the ones before the feature selection. The correlation table above looks nicer and most of the variables have high correlation values to more than one variable.

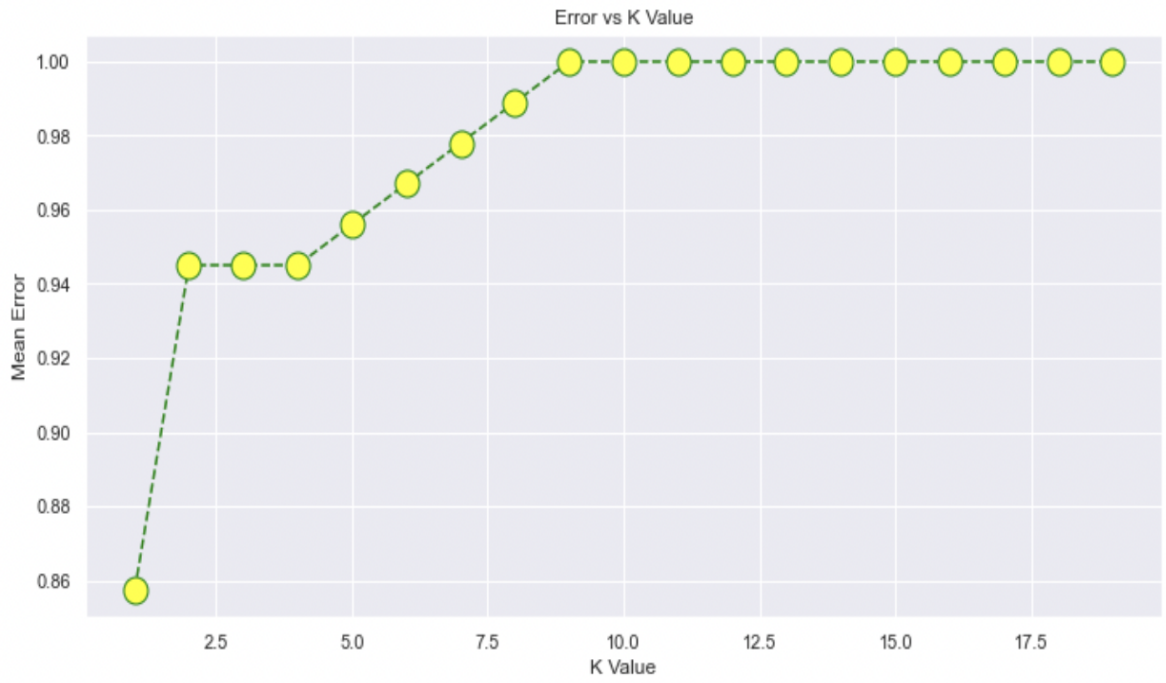
We keep GP to SO because they share high correlation among themselves and their relative correlation value to AVG score is also close to 0.4.

C. Algorithms (Random Forest, XGBoost,KNN)

KNN Regression:

We decided to implement the K nearest neighbors model for our regression analysis, due to its ease of use allowed by the fact that there are no statistical assumptions tied to this model. To calculate the kth nearest neighbor, we used the euclidean distance. To measure our model's accuracy, we used MSE as it's easy to interpret, since its scale is the square of the response variable. We chose the closest 3 neighbors for our model. While finding the closest 1 neighbor minimizes our MSE, it is likely that our model is overfitting, and so we decided to choose the closest 3 neighbors. Our MSE was 0.00078

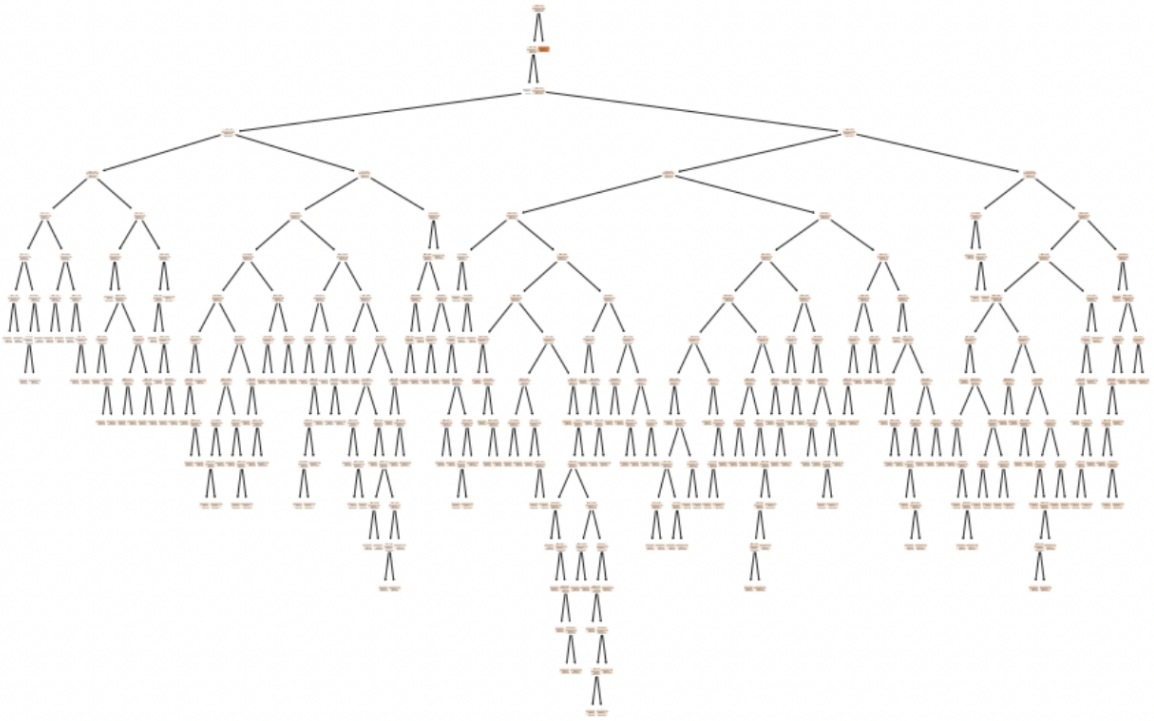
Figure 37. Error Rate for different K-values



Random Forest

Another method of predicting average batting is through the use of the Random Forest model. The random forest model is in essence a democratic model as it generates decision trees in parallel, the majority of the trees' decisions is what the model decides. It also requires little to no statistical assumptions to be met in order to use it. We found that 20 trees were the most optimal to predict average batting without overfitting the data. Our MSE was 0.00021

Figure 38. Decision Tree of the Random Forest Model.



XGBoost

XGBoost is a popular competition algorithm due to its ease of use, and its rigorous accuracy. Its main flaw is its prone-ness to overfit the data. Its counterpart is the Random forest model, in that the XGBoost model is also a tree based model. Although instead of parallel trees, it's an additive tree model, where the trees get added to correct the error in each model. The parameters we chose were a max tree depth of 3, a learning rate of 1, and n_estimators as 100. These parameters gave as an MSE of 0.00015

Top 10 Batters in the next season

With utilizing the Machine Learning Algorithms we were able to calculate the predicting AVG value of the players, and then combined them with X test values and make a prediction dataframe. We sorted the new data based on the players with highest AVG score which are listed below:

Figure 39. The Prediction of Top 10 batters in 2021 season of MLB by Decision Tree

Rank	Player Name	Team	Predicted AVG (Average Batting)	True AVG (Average Batting)
1	Daniel Bard	Colorado Rockies	0.307930	1.0
2	Bryce Harper	Philadelphia Phillies	0.275997	0.309
3	Juan Soto	Washington Nationals	0.275790	0.313
4	Ronald Acuna Jr.	Atlanta Braves	0.275763	0.283
5	Vladimir Guerrero	Toronto Blue Jays	0.275680	0.311
6	Frank Schwindel	Chicago Cubs	0.275520	0.342
7	Frank Schwindel	Chicago White Sox	0.275520	0.342
8	Jesse Winker	Cincinnati Reds	0.275510	0.305
9	Brandon Belt	San Francisco Giants	0.275437	0.274
10	Trea Turner	Los Angeles Dodgers	0.275420	0.338

Figure 40. The Prediction of Top 10 batters in 2021 season of MLB by XGboost				
Rank	Player Name	Team	Predicted AVG (Average Batting)	True AVG (Average Batting)
1	Frank Schwindel	Chicago White Sox	0.374855	0.342
2	Frank Schwindel	Chicago Cubs	0.374855	0.342
3	Adam Frazier	Pittsburgh Pirates	0.325386	0.324
4	Starling Marte	Oakland Athletics	0.323557	0.312
5	Starling Marte	Miami Marlins,True	0.320062	0.305
6	Thairo Estrada	San Francisco Giants	0.319079	0.273
7	Juan Soto	Washington Nationals	0.315312	0.313
8	Yuli Gurriel	Houston Astros	0.314484	0.319
9	Trea Turner	Washington Nationals	0.313053	0.322
10	Jesse Winker	Cincinnati Reds	0.310711	0.305

Figure 41. The Prediction of Top 10 batters in 2021 season of MLB by KNN n=3				
Rank	Player Name	Team	Predicted AVG (Average Batting)	True AVG (Average Batting)
1	Manny Machado 3B	San Diego Padres	0.283105	0.278
2	Jose Ramirez 3B	Cleveland Guardians	0.281947	0.266
3	Matt Olson 1B	Oakland Athletics	0.281421	0.271
4	Freddie Freeman 1B	Atlanta Braves	0.281105	0.3
5	Yuli Gurriel 1B	Houston Astros	0.280053	0.319
6	Vladimir Guerrero Jr. 1B	Toronto Blue Jays	0.279684	0.311
7	Jose Altuve 2B	Houston Astros	0.279632	0.277
8	Bryan Reynolds LF	Pittsburgh Pirates	0.278947	0.302
9	Bryce Harper DH	Philadelphia Phillies	0.278842	0.309
10	Cedric Mullins CF	Baltimore Orioles	0.277316	0.291

Figure 42. The Actual Top 10 batters in 2021 season of MLB			
Rank	Player Name	Team	True AVG (Average Batting)
1	Daniel Bard	Colorado Rockies	1
2	Frank Schwindel	Chicago White Sox	0.342
3	Frank Schwindel	Chicago Cubs	0.342
4	Trea Turner SS	Los Angeles Dodgers	0.338
5	Phil Bickford	Los Angeles Dodgers	0.333
6	Adam Frazier	Pittsburgh Pirates	0.324
7	Trea Turner	Washington Nationals	0.322
8	Yuli Gurriel	Houston Astros	0.319
9	Ketel Marte	Arizona Diamondbacks	0.318
10	Juan Soto	Washington Nationals	0.313

By applying the three models we have generated, we make predictions to the top 10 players to each of the models. The prediction results are not that accurate but they also show some trends. For instance, our Random Forest model has the highest accuracy among the tree, and our predicted 10 top players correspond to the true top 10 players most, despite there are some low predictions which are lower than 0.3 even 0.285.

The leading player Daniel Bard had been only predicted in the Random Forest Model, while the next good player Frank Schiwinde has been predicted by both the Random Forest model and the XGboost model.

We can also make predictions for the 2022 season, by using the same models, but there are two conditions we need to use as preconditions: time series including the previous seasons' samples instead of samples only from 2021, or the input sample features from the 2022 season.

IV. DISCUSSION

Through our study, we leaned towards the random forest model as it had the best model performance or 95% accuracy. We have conceded in our study that our sample size has been rather small, and thus may have led us to inaccurate results due to the nature of small sample sizes. While accuracy tended to be rather good, we are limited by our data's brevity.

The most surprising result we got from the experiment was when to select samples by Game Play in total. In the beginning, we chose to select players who have played over 50 games because many statistics only count for players who have played over 50 games. However, we later discussed that the median number may be a better choice because the dataset could be larger because the whole dataset was over 1000 and using GP > 50 sharply decreased it to 450. Therefore, we used the median GP is 31 to build models. In most of the parts the new dataframe worked well but in KNN model, the accuracy dropped from 0.74 to 0.62 which is a great change. In order to prevent such low accuracy which is less than 0.7, we changed the selection back to >=50 again, the accuracy surprisingly raised to 0.9 which is much greater than 0.74.

Therefore, from such experiments, we know KNN is a relatively sensitive method influenced by inputs compared to Random Forest and XGboost especially when the neighbors are few. In predicting a more precise and complicated dataset, we may use the Decision Tree or XGboost method which will take more time but the result should be better.

V. CONCLUSION

This project showed the usefulness of machine learning algorithms to predict the potential performance value a given player adds to a team, through the prediction of the player's batting average.

VI. REFERENCES

[1] "Batting (Baseball)." *Wikipedia*, Wikimedia Foundation, 24 Mar. 2022, https://en.wikipedia.org/wiki/Batting_(baseball) .

[2] "Baseball Batting." *Rookieroad.com*, https://www.rookieroad.com/baseball/101/batting/.

[3]Boone, Brian. "How the World Series Works." *HowStuffWorks*, HowStuffWorks, 31 Aug. 2012, https://entertainment.howstuffworks.com/world-series.htm.