IST 652 Final Project

**Predicting whether an NBA Player will Score More than 10 Points Per Game**

**Introduction**

The dataset that I used for this comes from Kaggle, in which there is a list of 11145 players and 22 columns about specific information about these players. This includes their name, team, college, and other statistics that they had in specific seasons like their average points per game. The overall goal in this research is to determine what factors contribute to a higher point total. In other words, what should NBA teams be looking for when drafting a player out of college. A question that I hope to answer in this research is does height and weight affect the total amount of points scored in the NBA. It is an old adage that you need height and weight to be successful in the NBA. I hope to put that theory to the test in this model. I also hope to find out which teams are the most successful in drafted players.

The columns are : Player\_ID, Player\_Name, College, Age, Height (cm), Weight (kg), Games Player,

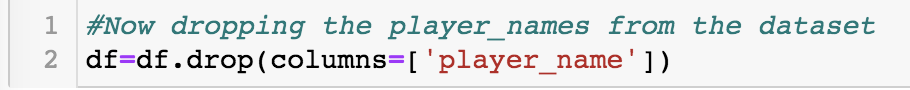
Points, Rebounds, Assists, Net Rating, Offensive Rebound Percentage, Defensive Rebound Percentage, Usage Rate, True Shooting Percentage, Assist Percentage, Draft Year, Draft Pick, Country and Season

**Problem Framing**

First, I want to clarify the difference between a supervised machine learning algorithm and unsupervised learning algorithm. A supervised machine learning algorithm has prior knowledge of what the output values of the sample should be. An unsupervised model does not have any labeled output. I wanted to perform a supervised model first to determine whether or not a player would score more than 10 points per game. I decided that the best way to do this was a logistical regression that could easily give me readable data of whether or not the player would score more than 10 points per game. This is the appropriate regression analysis to conduct in this case scenario because the dependent variable (whether or not the player scored 10 points a game) is binary. An unsupervised machine learning algorithm also needs to be used to get a new way of seeing the data. I decided to use K-Means clustering because it is a simple method of aggregated datapoints to see similarities. Therefore, if there is a certain cluster in which the players score more than 10 points per game, we can see the features of that cluster to see if there is a trend.

**Data Understanding**

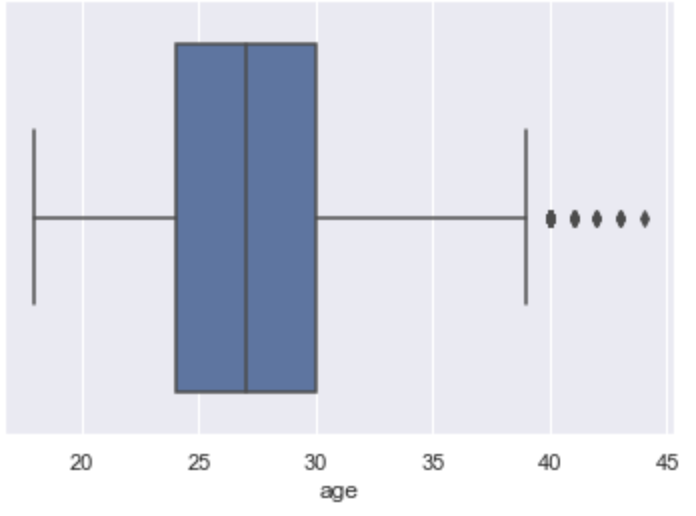
The first thing that I did for data understanding is to drop the college variable from the dataset. This is because players go to so many different colleges that it would be impossible to create dummy variables for all of the different colleges. The next thing that I did was move the Player\_ID variable and the names of the players into a separate data frame, which I plan on using later. After moving the player names into a data frame, I dropped the columns player\_name from the original data frame. This is shown below.



Next, using the describe function in Python, I got the summary statistics for each of the players in the dataset. The columns that I wanted to highlight were points, age, height, and weight. Since points is the dependent variable (or what I am trying to predict) , the mean was 8.1. This is good because I want to predict if the player scores over ten points per game which is a little above the mean. The average age for an NBA player is 27.16 years old which could imply that older players are not likely to reach 10 points per game. Finally, the average height is 200.81 cm and weight are 100 kg. These are important numbers to keep an eye out for as we move forward with this data mining research.

**Exploratory Data Analysis**

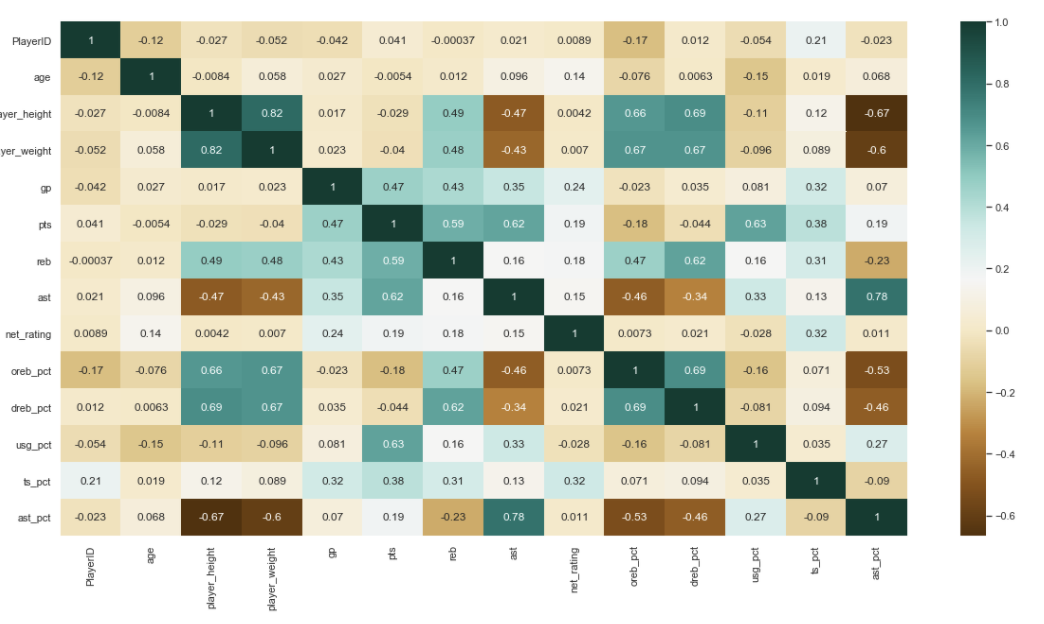
The first step in exploratory data analysis is to check for duplicates us the drop\_duplicates() function in python. In this dataset, there were no duplicated rows for each player. The next step was seeing if there were any NA’s or nulls in the dataset. To find this out, we used the isnull() function in python. There were some nulls for players who did not go to college but since we eliminated the college variable in the data framing step, there were no nulls that I could see but I used the dropna() function in the dataframe anyway. The next step in our exploratory data analysis was to remove outliers. The way that I did this was to visualize each numerical variable and made a boxplot using the seaborn package in python. For example the age boxplot is shown below.



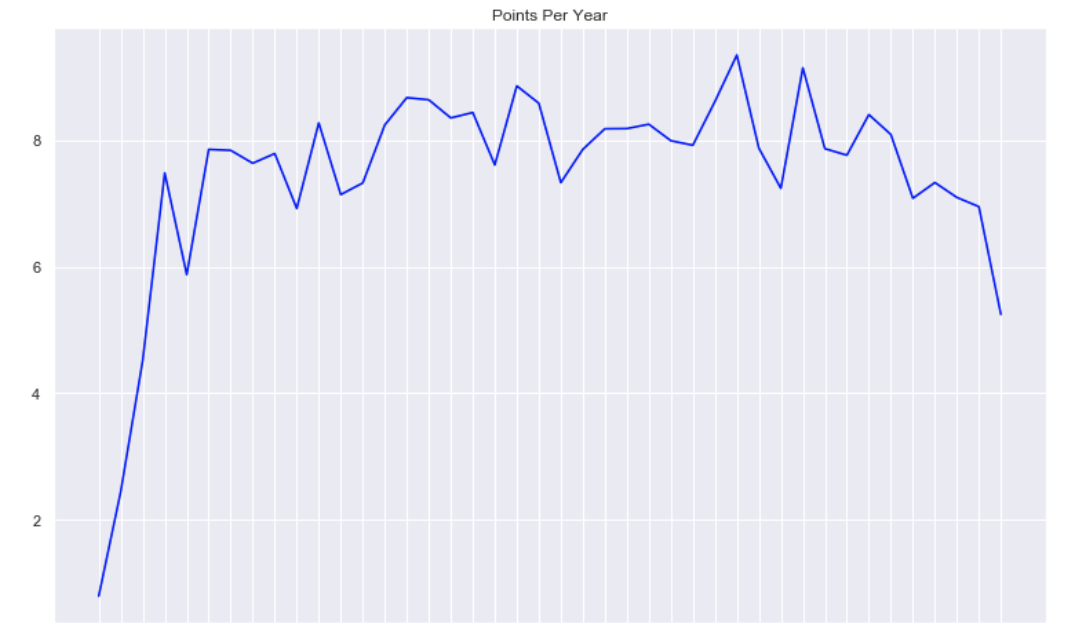
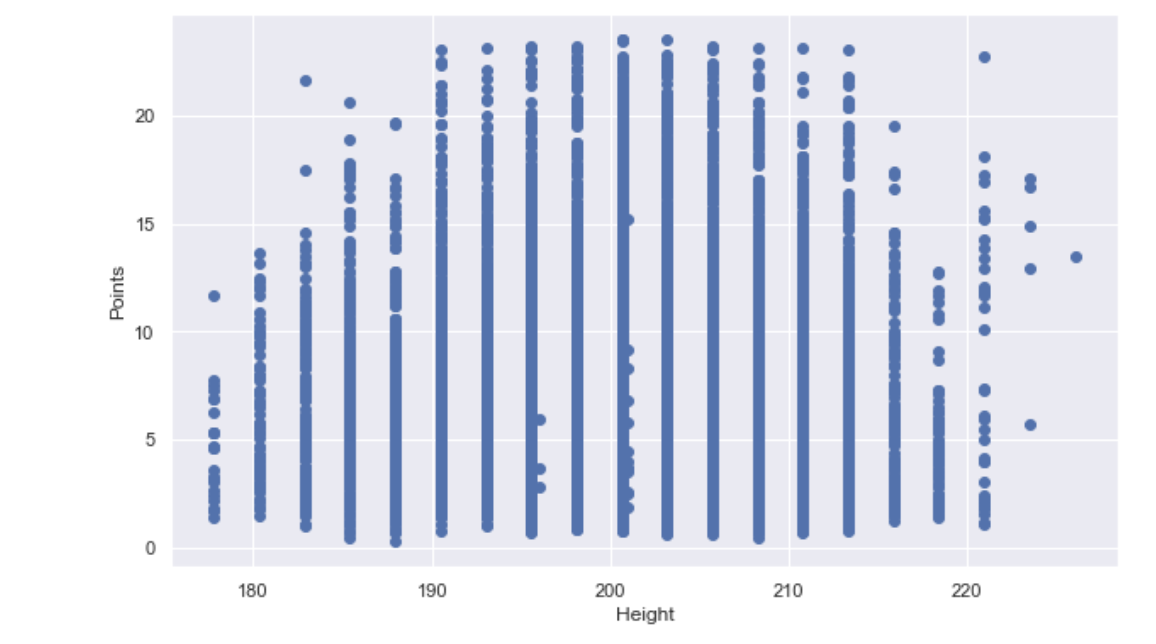
This boxplot shows that there are a few outliers, especially after the age of 40 years old. For the height variable, basketball players are usually tall, so it is no surprise that this is skewed towards the higher side. Other important variables that I want to highlight is the points boxplot, which has some outliers above 30 points per game. This need to be eliminated for a few reasons. First, they are outliers need to be eliminating to correct the model. Second, most of the players who average more than 30 points per game played in the 1960s ,where the game of basketball was so different. There were a few outliers in the other variables, but I will not go into that. However, the goal of this was to eliminate outliers. Therefore, used the interquartile range of each of the columns and removed values that were not either Q1-1.5 \* IQR or Q3 + 1.5 \* IQR.

**Visualization**

For visualization techniques, I wanted to do a few things. The first goes along with our exploratory data analysis and is a heatmap, which is shown below. As expected, columns like height and weight are highly correlated. Points is more correlated with rebounds than any other variable which could imply that more athletic players who get more rebounds.

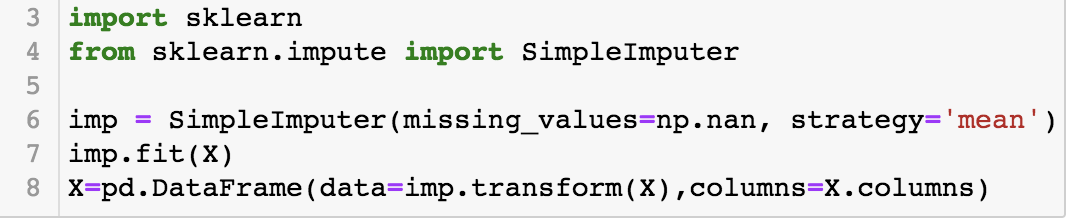


The second graph that I wanted to see if height contributed to the most points scored. There is a positive trend, as in the taller you are, the more points that you would score. This could be an important feature that in the model. I also graphed age and points per game but there was no correlation. The third graph shows the change in points per game over time. As you can see, there is no clear pattern. This shows the years 1979 through present. In the 1990s, it did spike but not a significant amount.



**Data Preprocessing**

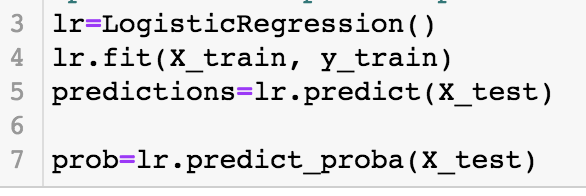
For the first step in the data preprocessing, I first made a new column called point which gave everyone who average 10 points per game a 1 and everyone else a 0. Then I created a X dataframe in which I dropped everyone who average 10 points per game. This dropped the number of rows to 8735. Then I create a Y dataframe of just the point column that I previously created. The next step was to encode the categorical variables such as season, team, and draft year. In order to do this, we used a variation of one hot encoding using the pd.get\_dummies because all of these columns have no ordinal relationship. However, the country variable had 71 categories. Therefore, I marked players not born in USA with an International so that there were only two categories. Then, I used the get\_dummies on all of the categorical columns. The next step that I did was using the simple imputer method in sklearn to replace missing values that may have come about during other preprocessing steps or other methods. Therefore, I replaced the missing values with the mean of the columns as shown below.



The next step in the data preprocessing step was to separate the data into training and test sets for the model using X and Y. I made the test size .3, therefore the training size would be .7. This was done using the train\_test\_split from the model selection. The next thing that I did was using the standard scaler function. The standardscaler transforms the data such that the distribution will have a mean value of 0 and standard deviation of 1. Because this has multiple columns, each column is done independently. In other words, the StandardScaler() function normalizes the features.

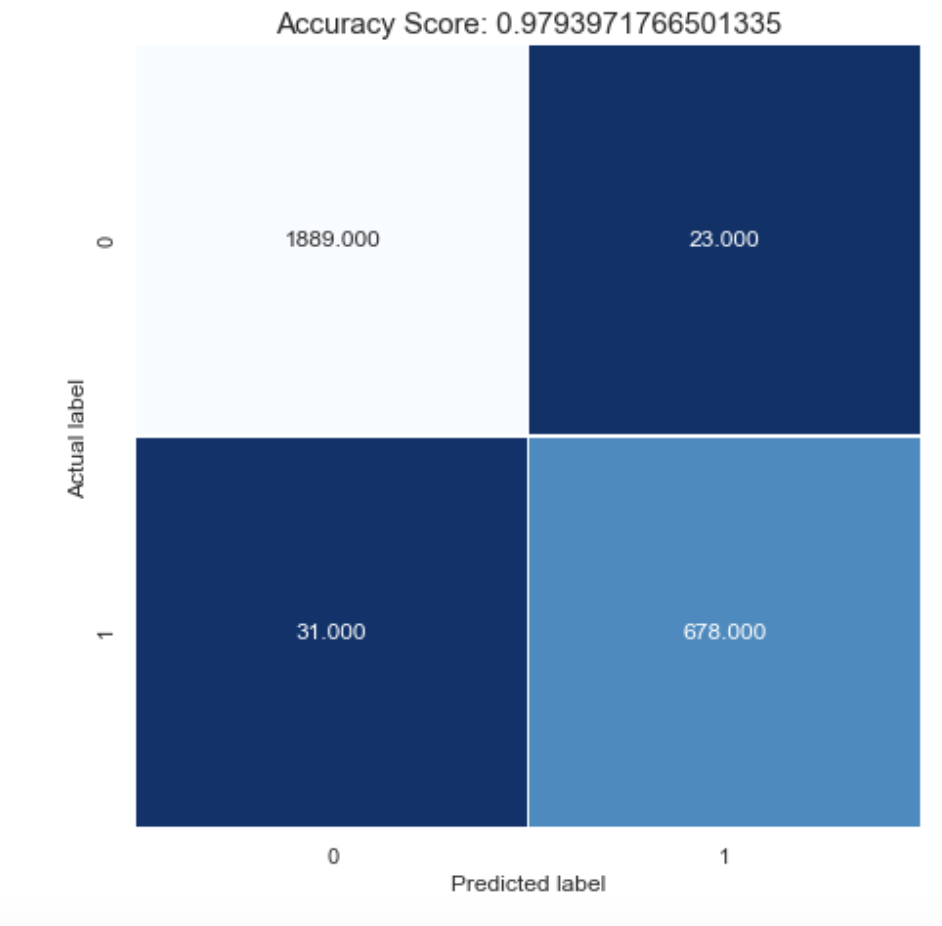
**Logistical Regression Model**

Since the points variable is binary, I wanted to use a logistical regression. I used the logistic regression and fit the model to the X\_train and Y\_train data. Then I made predictions and got the probability that each prediction would be correct.



**Logistical Regression Model Performance Evaluation**

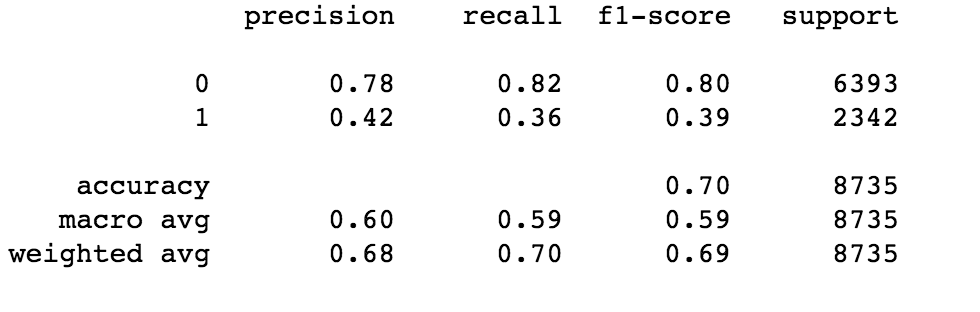
For the model performance evaluation, I first went the simple route and got the score of the logistical regression which came out as a .97939, which says to me that the model was extremely good at predicting whether or not the player would score more than 10 points per game. The next piece of model evaluation that I performed was a confusion matrix, which shows the number of true positives, false positives, true negatives, and false negatives. The heatmap of the confusion matrix is shown below.



As you can see in the visualization above, there were 1889 true negatives, 678 true positives, 23 false positives and 31 false negatives. This again shows the accuracy of the model in predicting whether or not the player will average more than 10 points per game. The false positives were players who played to an old age. These players mostly average more than 10 points per game in during their earlier ages but declined as they age. The false negatives were low draft picks who came out of nowhere to average 10 points per game. These were players who were young who never averaged more than 10 points per game before the current season. The next method of model evaluation that I used to test the validity of the model is R^2 and the MSE (mean square error) of the model. The R2 was .900, which illustrates that there is 90% of the variation that is explained by the model. The MSE was 0.2. Since the mean square error is small, that means that we are very close to finding the line of best fit.

Another method to determine the ROC\_auc\_score. A ROC curve is a graph showing the performance of the classification model at all classification thresholds. It shows the true positive rate and false positive rate as a graph. The AUC is the area underneath the ROC curve. The ROC curve for this model was .99584, which again proves that this is a very good model in predicting whether or not a player will score more than 10 points per game.

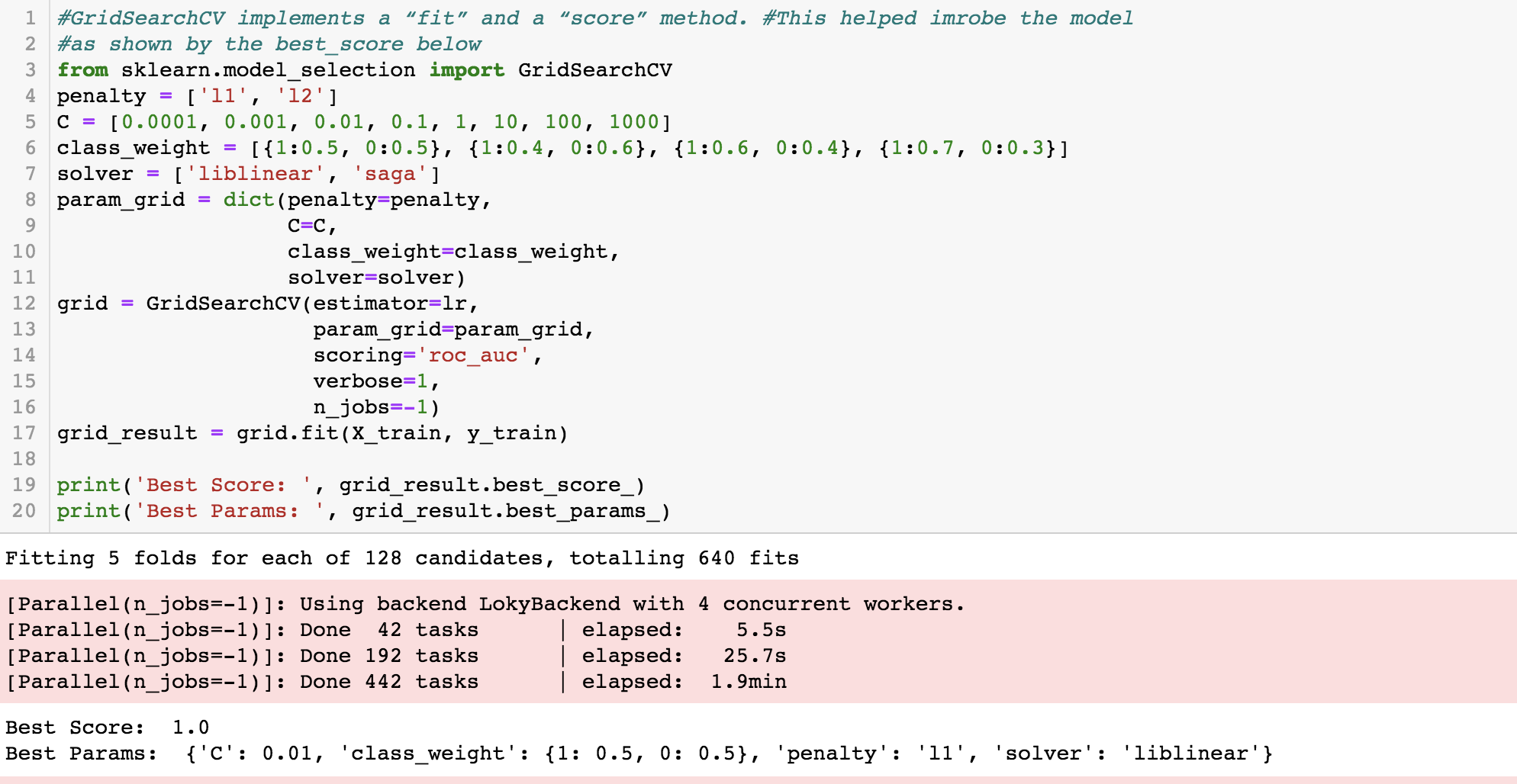
I got the accuracy of the model which was .98, which is just the amount of correct predictions over the amount of total predictions. I also performed a classification report that showed the precision (percentage of relevant results), recall (True Positives/Predicted Results) , f1-score(2\* (precision\*recall)/(precision+recall) for the model. They are all above average, which is consistent with the confusion matrix which showed a limited number of false positives and negatives.



The last step that I used to measure the success of the model is measuring the how effective the preprocessing is. I just dropped na and made the same model. The AUC model was .965, which is lower than the .995 auc score when I implemented preprocessing steps such as getting rid of outliers.

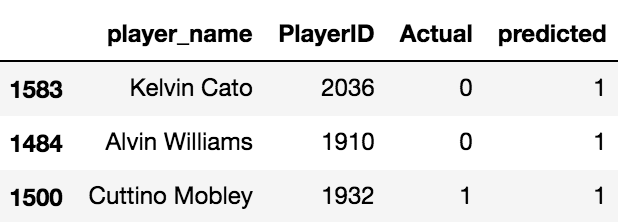
**Logistical Regression Model Algorithm Fine Tuning**

The method that I used to fine tune the model was the GridSearchCSV method which is from the sklearn.model\_selection. The GridSearch implements a fit and score method which is used to implement the scikit-learn estimator interface. The gives a list of params in order to fine tune the model. In otherwords, it finne-tunes the model. The GridSearchCV considers all parameter combinations. It also gives the best score and the pest params. The best score predicted an almost perfect model, as shown below.



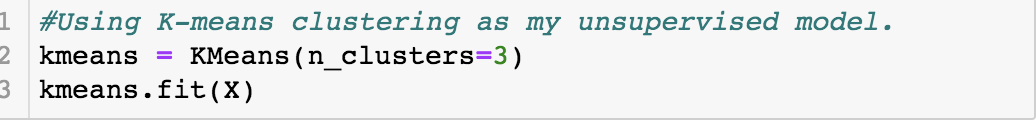
**Logistical Regression Model Interpretation**

In lieu on getting the coefficient, I got the most important features of the model using the feature selection tool in sklearn. The most important features were games played, drafted early in the draft and have a high net rating. Therefore, it shows that teams are usually good at selecting players in the draft. An interesting feature is that height, weight or age were not important features. Therefore, the old adage that taller basketball players are better does not apply in this model. I also created another dataframe using the player names (which I took out earlier and said it would come in handy later), and the actual and predicted variables. It looked like this. This once again told me that players who were drafted earlier are more likely to reach 10 points per game.

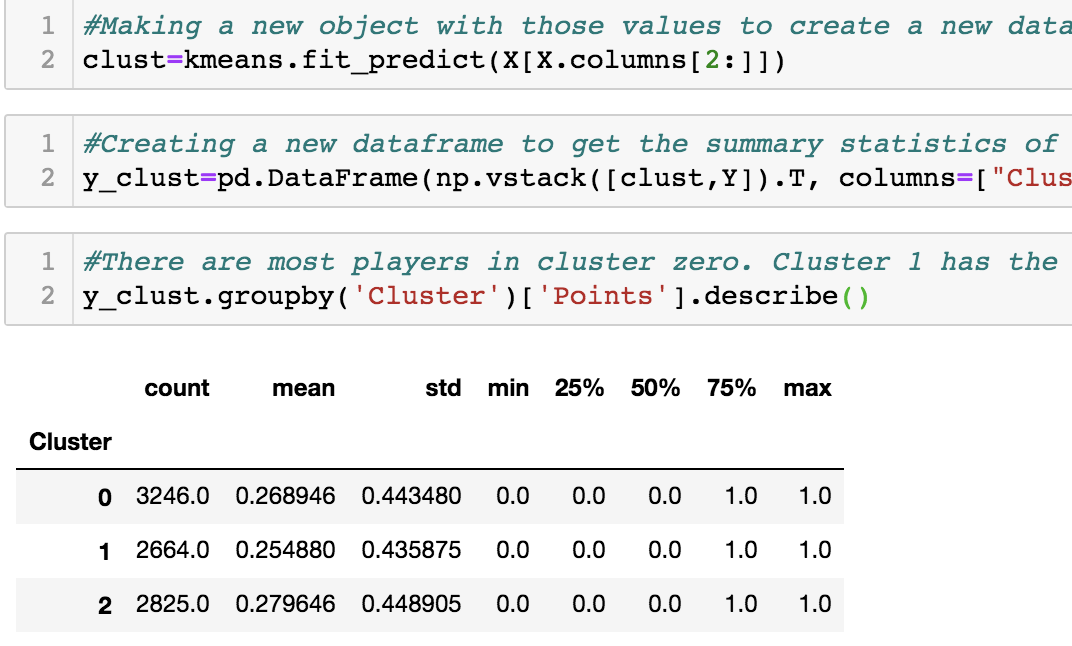


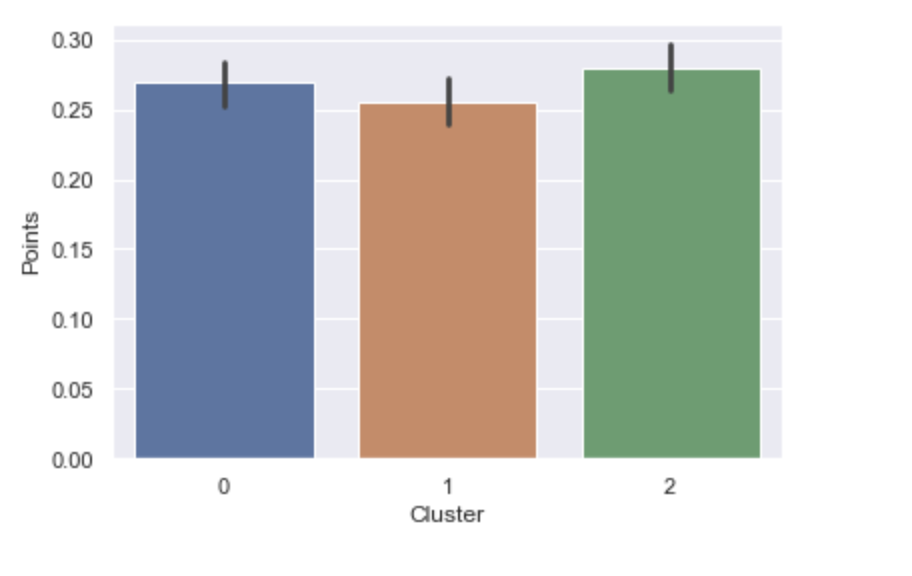
**K-Mean Clusters**

K-Means Clustering allows us to segment each of the players into different clusters which gives us different datapoints aggregates together due to some similarities. To figure out the number of clusters needed to be used, we found out the sum of squared errors. As I graphed the SSE, I had to account for some degree of error and showed me that three was the suitable amount of clusters.



I then made the cluster that each player was segmented in a separate column in the dataframe. I then made a dataframe of each of the clusters to get som summary statistics as shown below.



****

As you can see above, cluster 1 had the lowest prediction of .25., while 0 and 2 had a higher percentage of picks. Below I will list some characteristics of each cluster which illustrates why 0 and 2 had a higher mean value than cluster 1

Cluster 0: These are mostly players who shoot a lot and do not rebound or have many assists. These players only shoot and do not pass the ball that much. These players are good at shooting three pointers. Players in this cluster include Stephen Curry and Ray Allen (two of the best shooters of all time)

Cluster 1: Cluster 1 is mostly very tall players who rebound the ball a lot. This makes sense because taller players in basketball are not tasked with scoring the ball that much. Player in this cluster include Dennis Rodman (great rebounder, does not score a lot)

Cluster 2: Cluster 2 are big men who call shoot the ball. These players are tall but do not specialize in rebounding or defense. This cluster includes players such as Karl Malone (second all-time in scoring) and Chris Bosh (future hall of fame player).

**Conclusion**

The overall goal of this research was to determine which players are more likely to score more than 10 points per game. After conducting two machine learning models, I have come to the conclusion that height and weight do not matter when predicting if a player will score at least 10 points per game in a season. However, it seems like most of the players who scored at least 10 points per game were at least 6 foot 3. However, the NBA does do a good job of scouting because it seems as the earlier a player is selected in a draft, the more likely they will score at least 10 points in a game. This model was also correct in that it accounted for the fact that if a player scored at least 10 points per game in a previous season, they are more likely to do it again.

**Video Link**

<https://youtu.be/epEhvFC0NRE>

**Sources**

Helped With K-Means Clustering.

<https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

Helped Determine SSE to get the number of clusters.

<https://www.researchgate.net/figure/Plot-of-SSE-versus-the-number-of-clusters-K-for-the-minimum-temperature_fig1_267862793>

Helped with Fine Tuning the logistical regression

<https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a>

Helped with various preprocessing techniques

<https://www.youtube.com/watch?v=V0u6bxQOUJ8>

Slides from IST 652 Class