NBA PREDICTIVE ANALYTICS

MACHINE LEARNING MODEL WRITTEN IN PYTHON TO ANALYZE NBA PLAYERS SHOOTING

VAMSIKRISHNA ATLURU

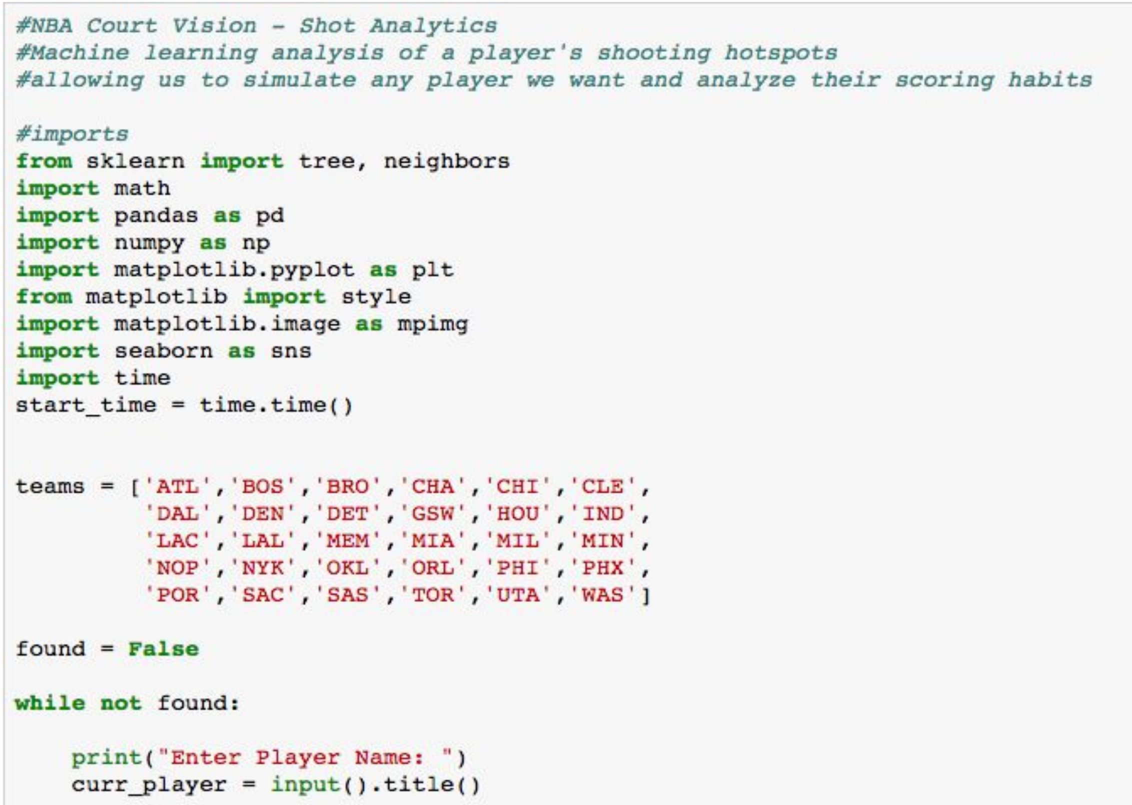
HOW IT WORKS

Using my experience in data science, I created a Python program using Jupyter Notebook which analyzes any current NBA player’s gameplay to find which areas of the court they have most shooting success and the probability of shooting from certain spots. With this analysis, it can allow coaches and players to know their opponents gameplay and can show them which areas of the court to prioritize their defense.

HOW IT WORKS

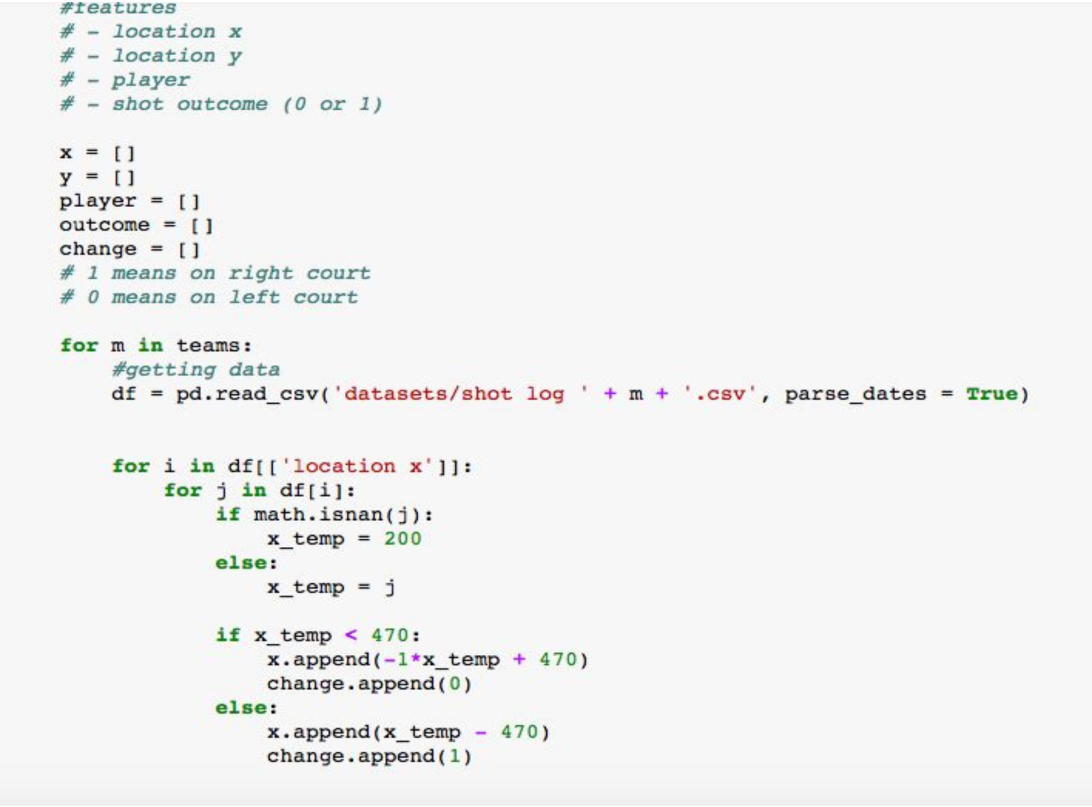
Using datasets from Kaggle (formatted with Pandas), we can use Matplotlib to illustrate our data analysis. This dataset includes every shot taken in the 2016-2017 Regular Season so any player who played a game during this season can be analyzed. By incorporating machine learning through Python's Scikit-Learn (using a K-Nearest Neighbours Classifier) we can also simulate a player's shooting from every position on the court. There are 4 different modes utilized in this model to analyze shooting data.

CODE(Repeated Code Left Out for Visual Purposes)



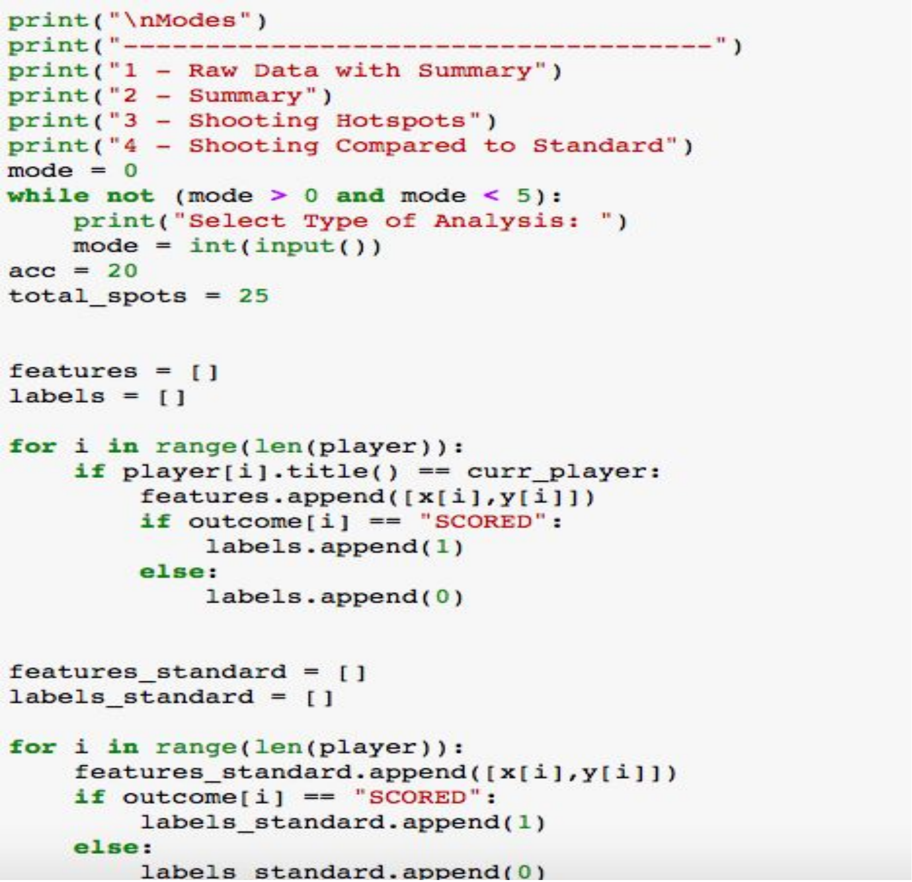
```
#NBA Court Vision - Shot Analytics
#Machine learning analysis of a player's shooting hotspots
#allowing us to simulate any player we want and analyze their scoring habits

#imports
from sklearn import tree, neighbors
import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
import matplotlib.image as mpimg
import seaborn as sns
import time
start_time = time.time()


teams = ['ATL','BOS','BRO','CHA','CHI','CLE',
         'DAL','DEN','DET','GSW','HOU','IND',
         'LAC','LAL','MEM','MIA','MIL','MIN',
         'NOP','NYK','OKL','ORL','PHI','PHX',
         'POR','SAC','SAS','TOR','UTA','WAS']

found = False

while not found:

    print("Enter Player Name: ")
    curr_player = input().title()
```

CODE



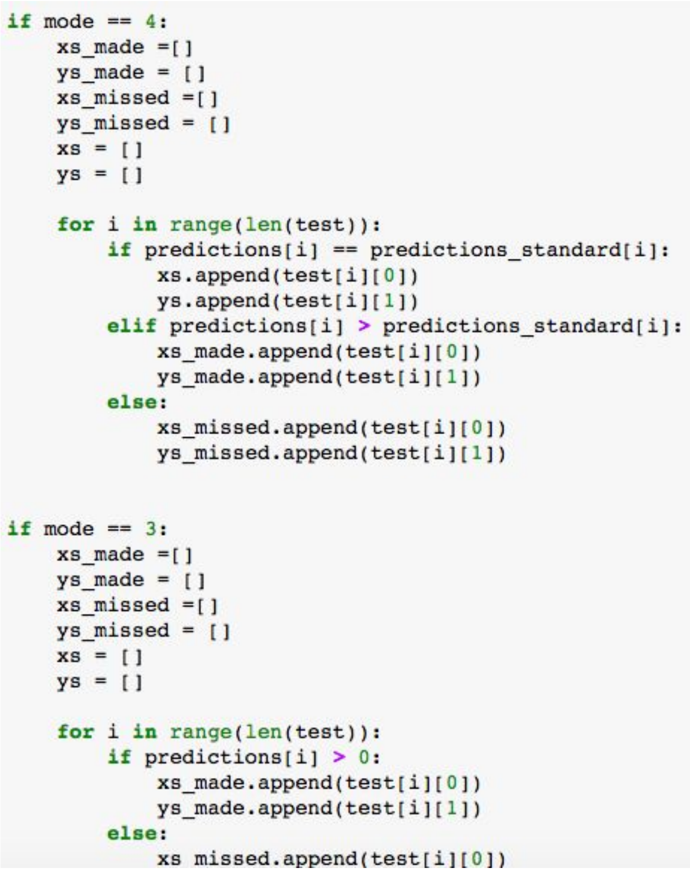
```
#features
# - location x
# - location y
# - player
# - shot outcome (0 or 1)

x = []
y = []
player = []
outcome = []
change = []
# 1 means on right court
# 0 means on left court

for m in teams:
    #getting data
    df = pd.read_csv('datasets/shot log ' + m + '.csv', parse_dates = True)


    for i in df[['location x']]:
        for j in df[i]:
            if math.isnan(j):
                x_temp = 200
            else:
                x_temp = j

            if x_temp < 470:
                x.append(-1*x_temp + 470)
                change.append(0)
            else:
                x.append(x_temp - 470)
                change.append(1)
```

CODE



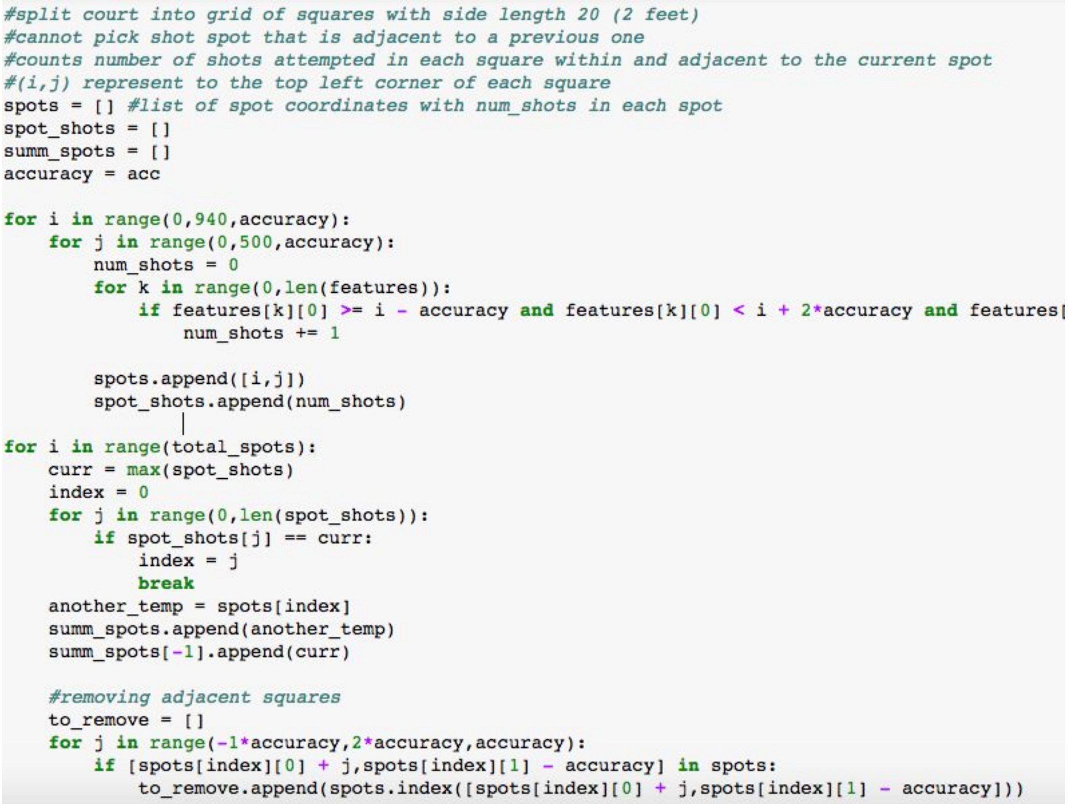
```
print("\nModes")
print("------------------------------------")
print("1 - Raw Data with Summary")
print("2 - Summary")
print("3 - Shooting Hotspots")
print("4 - Shooting Compared to Standard")
mode = 0
while not (mode > 0 and mode < 5):
    print("Select Type of Analysis: ")
    mode = int(input())
acc = 20
total_spots = 25


features = []
labels = []

for i in range(len(player)):
    if player[i].title() == curr_player:
        features.append([x[i],y[i]])
        if outcome[i] == "SCORED":
            labels.append(1)
        else:
            labels.append(0)


features_standard = []
labels_standard = []

for i in range(len(player)):
    features_standard.append([x[i],y[i]])
    if outcome[i] == "SCORED":
        labels_standard.append(1)
    else:
        labels_standard.append(0)
```

CODE



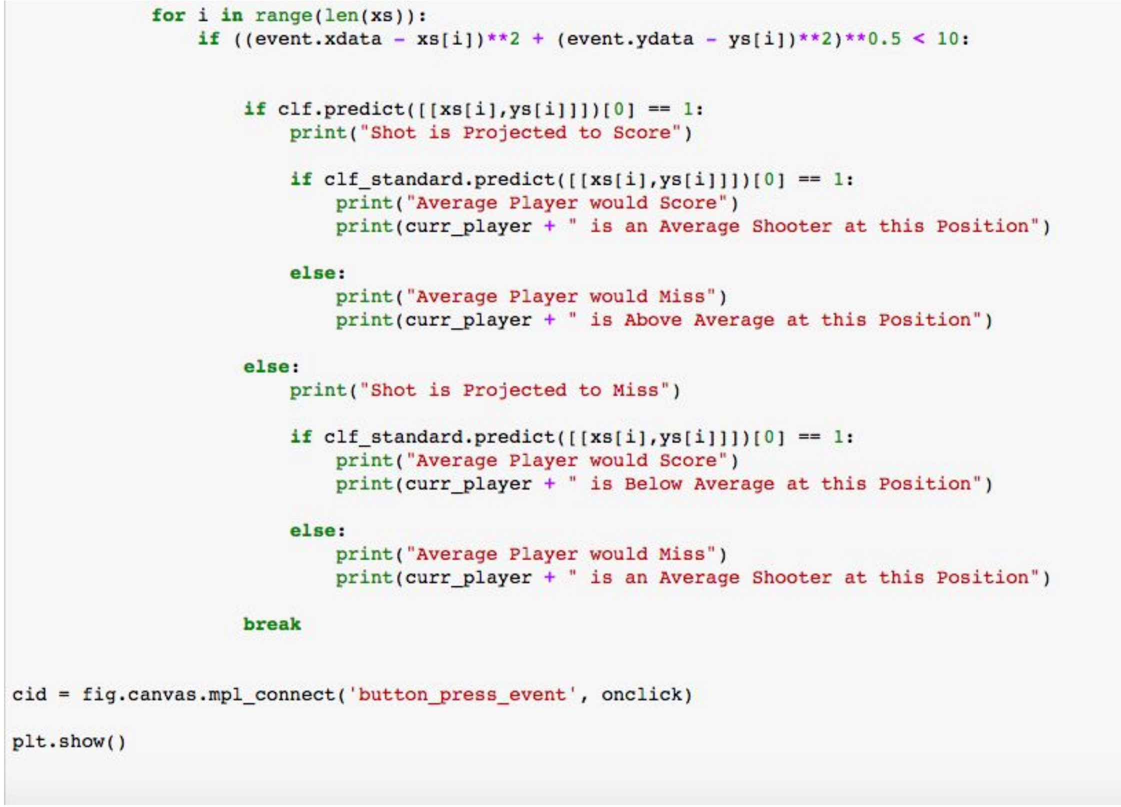
```python
if mode == 4:
    xs_made =[]
    ys_made = []
    xs_missed =[]
    ys_missed = []
    xs = []
    ys = []

    for i in range(len(test)):
        if predictions[i] == predictions_standard[i]:
            xs.append(test[i][0])
            ys.append(test[i][1])
        elif predictions[i] > predictions_standard[i]:
            xs_made.append(test[i][0])
            ys_made.append(test[i][1])
        else:
            xs_missed.append(test[i][0])
            ys_missed.append(test[i][1])


if mode == 3:
    xs_made =[]
    ys_made = []
    xs_missed =[]
    ys_missed = []
    xs = []
    ys = []

    for i in range(len(test)):
        if predictions[i] > 0:
            xs_made.append(test[i][0])
            ys_made.append(test[i][1])
        else:
            xs_missed.append(test[i][0])
```

CODE



```
#split court into grid of squares with side length 20 (2 feet)
#cannot pick shot spot that is adjacent to a previous one
#counts number of shots attempted in each square within and adjacent to the current spot
#(i,j) represent to the top left corner of each square
spots = [] #list of spot coordinates with num_shots in each spot
spot_shots = []
summ_spots = []
accuracy = acc

for i in range(0,940,accuracy):
    for j in range(0,500,accuracy):
        num_shots = 0
        for k in range(0,len(features)):
            if features[k][0] >= i - accuracy and features[k][0] < i + 2*accuracy and features[
                num_shots += 1

        spots.append([i,j])
        spot_shots.append(num_shots)

for i in range(total_spots):
    curr = max(spot_shots)
    index = 0
    for j in range(0,len(spot_shots)):
        if spot_shots[j] == curr:
            index = j
            break
    another_temp = spots[index]
    summ_spots.append(another_temp)
    summ_spots[-1].append(curr)

    #removing adjacent squares
    to_remove = []
    for j in range(-1*accuracy,2*accuracy,accuracy):
        if [spots[index][0] + j,spots[index][1] - accuracy] in spots:
            to_remove.append(spots.index([spots[index][0] + j,spots[index][1] - accuracy]))
```

CODE



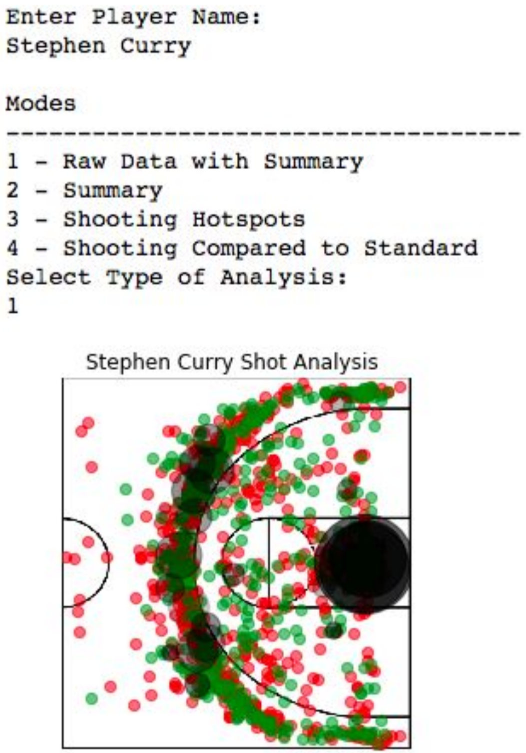
```
            for i in range(len(xs)):
                if ((event.xdata - xs[i])**2 + (event.ydata - ys[i])**2)**0.5 < 10:


                    if clf.predict([[xs[i],ys[i]]])[0] == 1:
                        print("Shot is Projected to Score")

                        if clf_standard.predict([[xs[i],ys[i]]])[0] == 1:
                            print("Average Player would Score")
                            print(curr_player + " is an Average Shooter at this Position")

                        else:
                            print("Average Player would Miss")
                            print(curr_player + " is Above Average at this Position")

                    else:
                        print("Shot is Projected to Miss")

                        if clf_standard.predict([[xs[i],ys[i]]])[0] == 1:
                            print("Average Player would Score")
                            print(curr_player + " is Below Average at this Position")

                        else:
                            print("Average Player would Miss")
                            print(curr_player + " is an Average Shooter at this Position")

                    break


cid = fig.canvas.mpl_connect('button_press_event', onclick)

plt.show()
```

How can this program be tested and used to analyze shooting data for a specific player ?

TEST EXAMPLE 1 - Stephen Curry

Lets use Stephen Curry from the Golden State Warriors as the first test example and take an in-depth view of the output generated in this program.

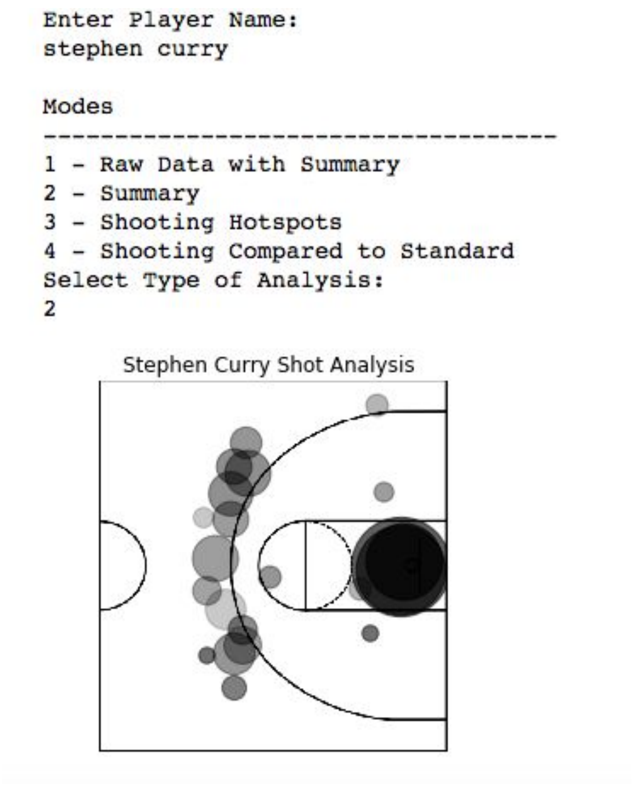
Stephen Curry Analysis



```
Enter Player Name:
Stephen Curry

Modes
------------------------------------
1 - Raw Data with Summary
2 - Summary
3 - Shooting Hotspots
4 - Shooting Compared to Standard
Select Type of Analysis:
1
```

In this plot, the green dots represent scoring shots while the red dots represent missed shots. The black dots represent spots that Curry is most likely to shoot from. The size of the dots represent the relative probability of Curry shooting from that position and the darkness of the dot illustrates his shot accuracy from that spot (with darker shades representing a higher accuracy). This is the first mode in this program returning the raw data with summary. As the raw data can be hard to visualize we will investigate the other modes in this program to attain a better visual representation.

Stephen Curry Analysis



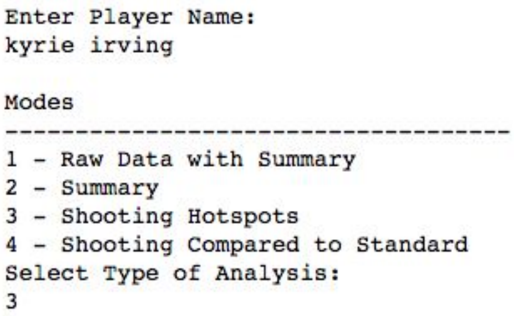
```
Enter Player Name:
stephen curry

Modes
------------------------------------
1 - Raw Data with Summary
2 - Summary
3 - Shooting Hotspots
4 - Shooting Compared to Standard
Select Type of Analysis:
2
```

This plot shows a clear visualization of what types of shots Stephen Curry is most likely to take. Again, the black dots represent spots that Curry is most likely to shoot from.The size of the dots represent the relative probability of Curry shooting from that position and the darkness of the dot illustrates his shot accuracy from that spot (with darker shades representing a higher accuracy). The plot says that Curry is most likely to shoot from very close range (at layup distance) or from the 3-point range. As confirmed by the knowledge of Curry's 3-shooting and ability to pass defenders for layups, we can assess this model successful for this test example.

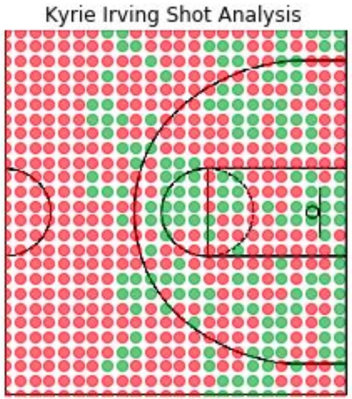
TEST EXAMPLE 2- Kyrie Irving

In the previous test example, we have used Stephen Curry to assess the model to generate the 1st and 2nd mode of the model including “Raw Data with Summary” and “Shooting Summary”. Let’s use Kyrie Irving from the Boston Celtics as the second test example to look at the 3rd and 4th modes of this machine learning model including “Shooting Hotspots” and “Shooting Compared to the Standard NBA Player”.

Kyrie Irving Analysis



```
Enter Player Name:
kyrie irving

Modes
------------------------------------
1 - Raw Data with Summary
2 - Summary
3 - Shooting Hotspots
4 - Shooting Compared to Standard
Select Type of Analysis:
3
```

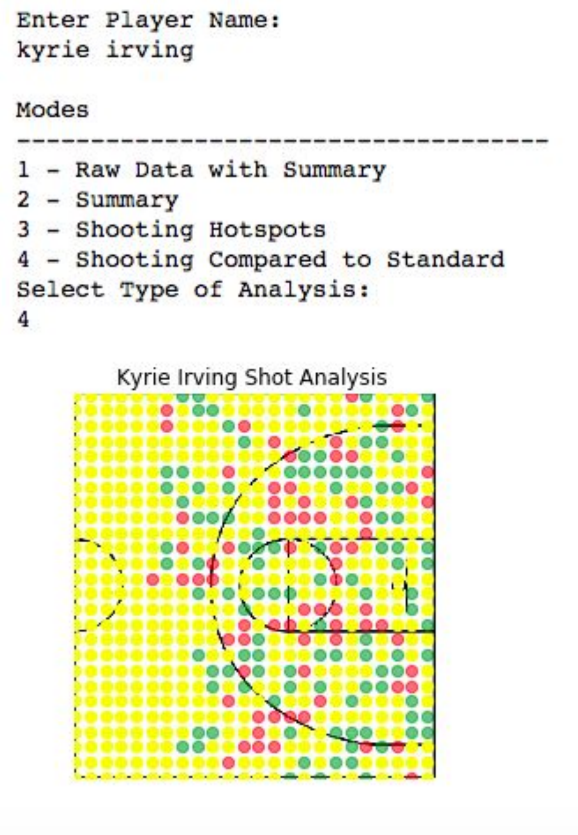


In this analysis we are incorporating machine learning by using a Decision Tree to simulate Kyrie Irving's shooting throughout the court. With this analysis we have a more uniform distribution for our analysis so we can predict how Irving will shoot based on his past shooting habits. In this plot, the green dots represent scoring shots and the red dots represent missed shots.

From this analysis we can determine that Irving has a uniform shooting distribution. He also is predicted to score with high consistency throughout the key and around the 3-point perimeter - another prediction which is confirmed by his known playing style.

But we can take this one step further, let's compare Kyrie Irving's shooting to the average NBA player.

Kyrie Irving Analysis



```
Enter Player Name:
kyrie irving

Modes
------------------------------------
1 - Raw Data with Summary
2 - Summary
3 - Shooting Hotspots
4 - Shooting Compared to Standard
Select Type of Analysis:
4
```

Similar to how we simulated Kyrie Irving's shooting with machine learning, we can use the data for all NBA players in the 2016-2017 Regular Season to find the shooting habits of the average NBA player. In this plot, green dots represent shots that Irving made but the NBA average player missed (meaning Irving is above average shooting at this positions). In contrast, the red dots represent shots that Irving missed but the average player scored (meaning Irving is below average at these spots). Additionally, the yellow dots represent neutral areas where both Irvingand the average player scored or missed.

From the plot, we can see that Irving is above average in the majority of the court. He dominates shooting throughout the key and around the 3-point range - once again aligning with his known game style.

CONCLUSION

Overall, this predictive model can predict NBA players' shooting habits with a high degree of effectiveness as seen from the analysis on Curry and Irving. We can determine exactly which areas a player is above average at shooting and which they are below. We can also predict where they are most likely to shoot from and their accuracy from these positions. With this data, NBA coaches can plan defense strategies with much higher precision by focusing on these areas that players shoot with greater accuracy and creating strategies to force players into locations on the floor that they shoot with less accuracy. If you are further interested in this data science model, please contact me or try the model yourself with your favorite NBA player on my github repository.