Using Machine Learning to Predict the NBA MVP

Team K

Robert Kapsch, Michael Rylance, John Baird

I. Introduction

A. At the end of each season, the most prestigious award is the Michael Jordan Trophy or the NBA Most Valuable Player Trophy. The award is traditionally given to the player deemed to have the greatest impact on their team's success. Our goal in this project is to unveil the statistical features that most significantly impact the likelihood of a player winning MVP. We plan to build a model to intake the regular season stats and output the player most likely to win MVP. The data we worked with came from Kaggle and included every player since 1980 who received MVP votes, along with relevant season statistics for each year. Using this data, we can identify which areas of the game help or hurt a player's MVP odds. Outside of this assignment, our model will have real-world applications that extend across many industries. NBA teams can gain insight into the types of players they want to add that will best benefit their teams. Sports media can now use data-driven analytics to more accurately update the general population on the current MVP race. This paper will navigate the previous NBA MVP seasons and pinpoint the most relevant statistics that helped them win MVP.

II. Problem/Question

A. The main questions we are trying to answer are

1. What statistical factors most influence player likelihood on winning MVP?

2. Does the MVP usually have numerous high performance games, or possess a more consistent playstlye over the course of the season?
3. How does team success correlate with player MVP odds?
4. Can we use this model to predict future MVP winners?

III. Survey

A. Currently, the MVP is presented to fans without giving clear insight into how the rankings are established. A singular high-performance game can jump players into the MVP race even if they have not had an outstanding season. Additionally, some narratives may impact the MVP rankings that extend past what happens on the basketball court. Last year, for example, many people had Joel Embiid winning the MVP even though Nikola Jokic was having a statistically more impressive season and would go on to help his team win the NBA championship. Our model focuses on only on-court performance while avoiding bias towards high-performance games. We will only be focusing on regular season statistics, as the trophy also accounts for it. We will not include players who played less than 40 games or who did not receive MVP votes.

IV. Proposed Method

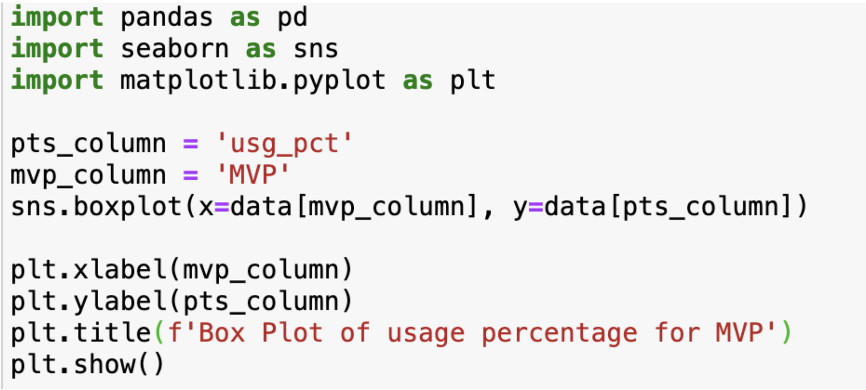
A. Data Cleaning

1. While our dataset did include all players who received MVP votes, it did not designate who actually won the award. We manually went in and added a column called ‘MVP’ and added a ‘1’ if the player did win and a ‘0’ if the player did not win for a given season. Furthermore, we dropped columns with ‘NaN’ values (only about 7 in our 500) to allow the machine to most effectively learn from the data. The only column we removed from our dataset was the player name column.

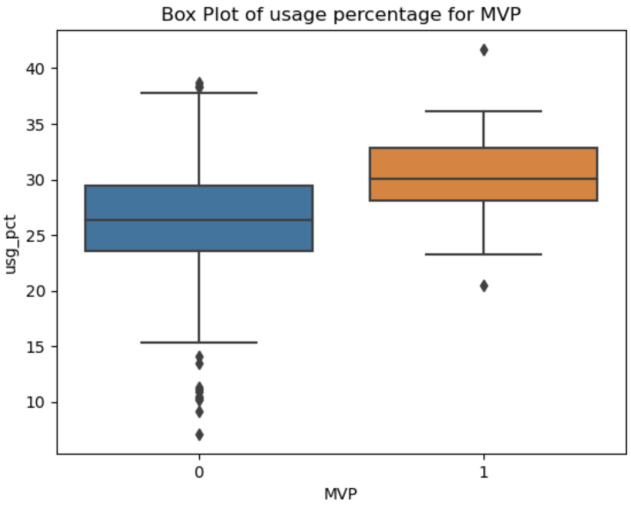
B. Initial Visualization

1. We wanted to see the difference between a player who won MVP and one who did not during a season. To do this, we created a boxplot for a side by side (winner vs non-winner) comparison of overall averages of each statistical category. We

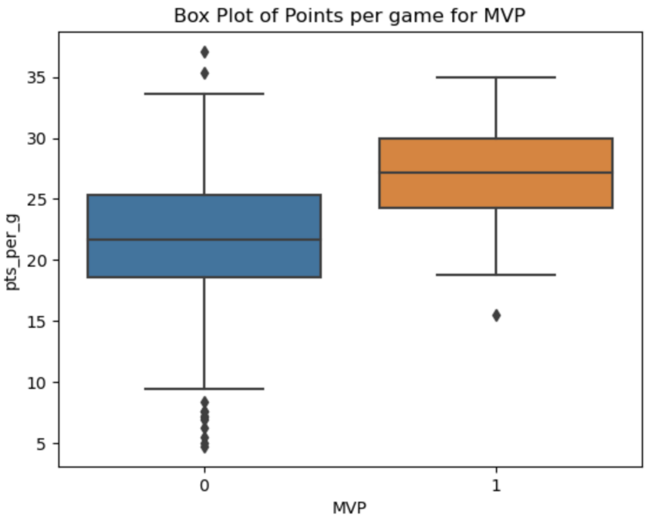
will show two examples, but this code will work for every category simply by switching the column we wish to compare.

2. 

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

pts_column = 'usg_pct'
mvp_column = 'MVP'
sns.boxplot(x=data[mvp_column], y=data[pts_column])

plt.xlabel(mvp_column)
plt.ylabel(pts_column)
plt.title(f'Box Plot of usage percentage for MVP')
plt.show()
```

3. 

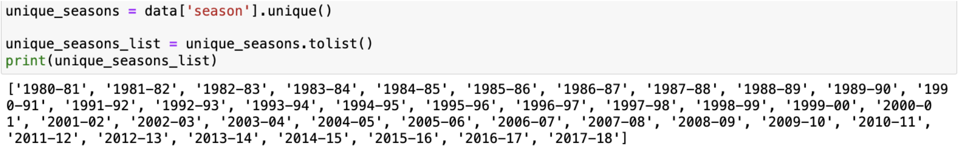
a) In this example, we compared usage percentage, the percentage of scoring plays either scored by or assisted by a player for a team. The data shows that MVP caliber players on average account for around 4% more scoring plays than their peers. It is also evident that the 25% quartile for MVP players is higher than theoverall average for non-MVP players.



4.

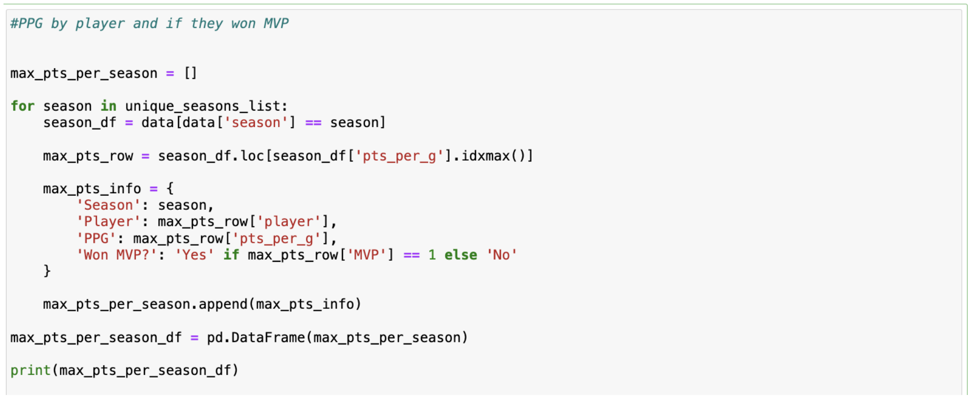
a) This box plot shows that MVPs on average scored around five more points per game than their peers. The MVP data is significantly more centralized while the non-MVP data has many outliers.

C. Next, we wanted to see how often the MVP also was the scoring leader for the league. We were interested in this because the NBA shows bias in the number of highlights they show and the attention a player gets. To do this, we picked the top scorer from each season then used the data to see if they won MVP

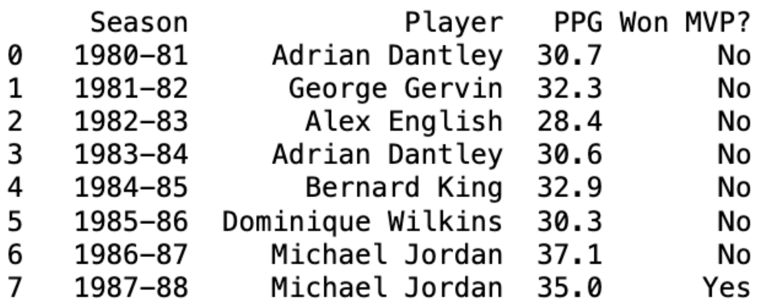


```
unique_seasons = data['season'].unique()

unique_seasons_list = unique_seasons.tolist()
print(unique_seasons_list)
```

```
['1980-81', '1981-82', '1982-83', '1983-84', '1984-85', '1985-86', '1986-87', '1987-88', '1988-89', '1989-90', '199
0-91', '1991-92', '1992-93', '1993-94', '1994-95', '1995-96', '1996-97', '1997-98', '1998-99', '1999-00', '2000-0
1', '2001-02', '2002-03', '2003-04', '2004-05', '2005-06', '2006-07', '2007-08', '2008-09', '2009-10', '2010-11',
'2011-12', '2012-13', '2013-14', '2014-15', '2015-16', '2016-17', '2017-18']
```

1. 

```
#PPG by player and if they won MVP


max_pts_per_season = []

for season in unique_seasons_list:
    season_df = data[data['season'] == season]

    max_pts_row = season_df.loc[season_df['pts_per_g'].idxmax()]

    max_pts_info = {
        'Season': season,
        'Player': max_pts_row['player'],
        'PPG': max_pts_row['pts_per_g'],
        'Won MVP?': 'Yes' if max_pts_row['MVP'] == 1 else 'No'
    }

    max_pts_per_season.append(max_pts_info)

max_pts_per_season_df = pd.DataFrame(max_pts_per_season)

print(max_pts_per_season_df)
```

2. 

```
    Season             Player   PPG Won MVP?
0  1980-81     Adrian Dantley  30.7       No
1  1981-82      George Gervin  32.3       No
2  1982-83       Alex English  28.4       No
3  1983-84     Adrian Dantley  30.6       No
4  1984-85       Bernard King  32.9       No
5  1985-86  Dominique Wilkins  30.3       No
6  1986-87     Michael Jordan  37.1       No
7  1987-88     Michael Jordan  35.0      Yes
```

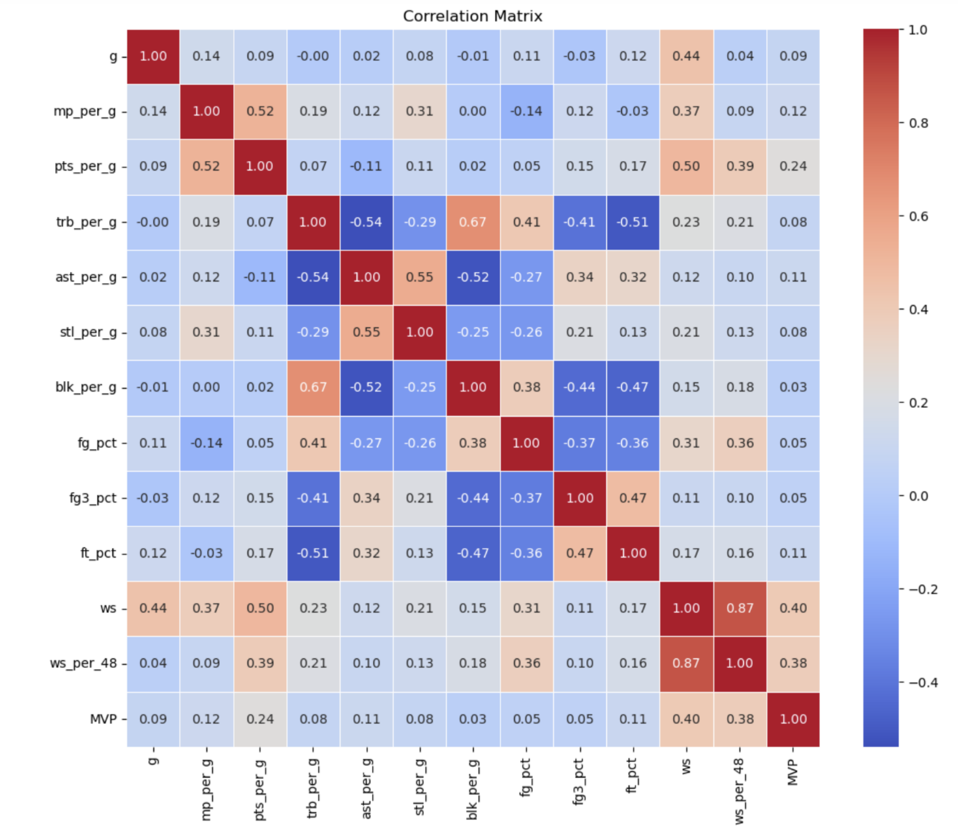
3. We found that a player who won MVP was also the NBA scoring leader only 31.5% of the time.

D. Machine learning

1. Our end goal was to create a model that can intake season statistics and give an accurate prediction of who will win MVP. Our method will be shown in the next section.

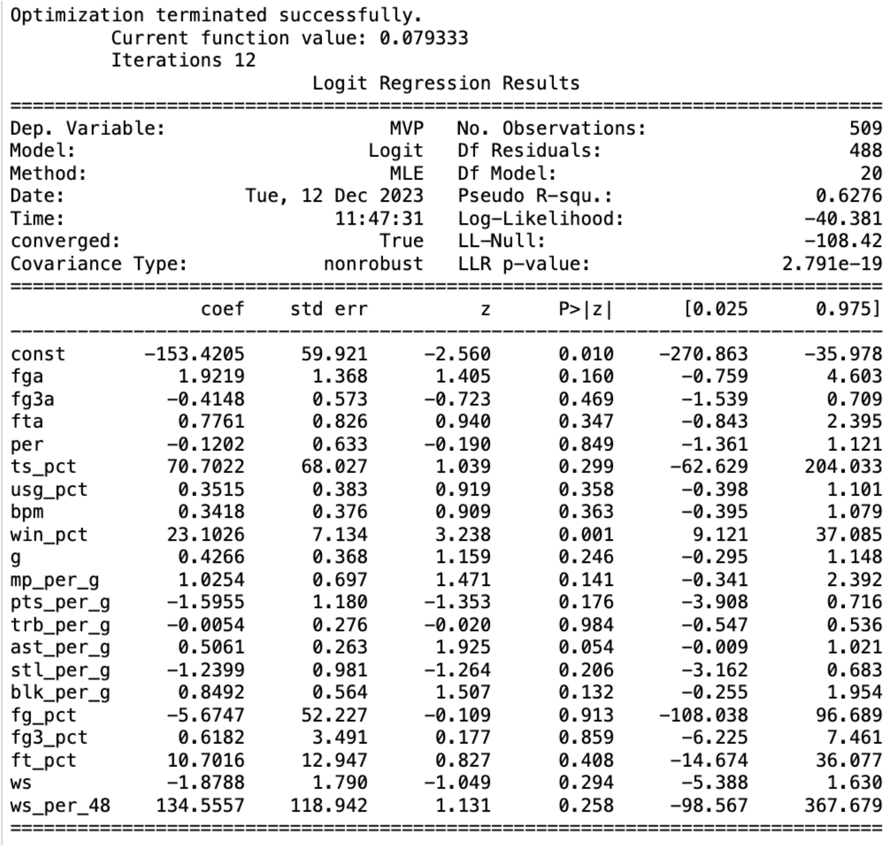
V. Experiment/Evaluation

A. Right out of the gate, we wanted to see how each statistical category related with eachother. To do this, we created a correlation matrix.

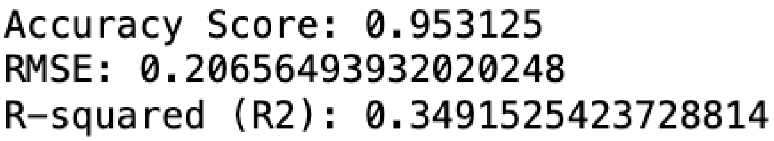
1. 

a) The highest correlation with MVP was the winshare (ws) statistic. Win share is, “a player statistic which attempts to divvy up credit for team success to the individuals on the team” (BasketaballReference.com). This demonstrates that a player who contributes most to his teams success should win the MVP, the definition of the award. By doing this, we were able to see not only how each statistic correlated with MVP, but also how the statistics correlated with each other. The correlation matrix shows us a player who has a lot of steals in a season is also likely to have a lot of rebounds in a season. This can go further to show the playstyle of certain players and like if they are an offensive player or defensive player.

B. We next moved to deciding which model (Regression, KNN, or Random Forest) would be the most accurate. After making each model, we found that Logisitic regression gave us the most precise model

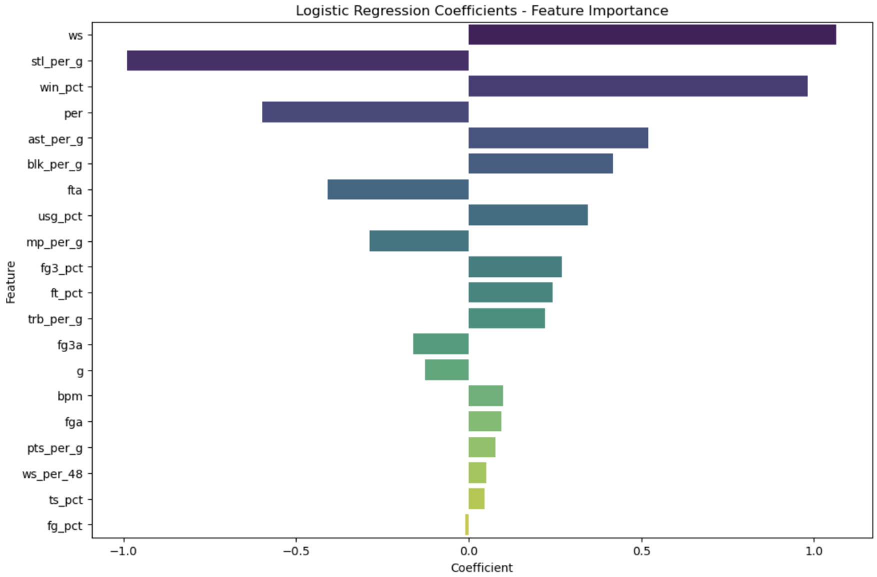
1. 

```
Optimization terminated successfully.
         Current function value: 0.079333
         Iterations 12
                           Logit Regression Results
==============================================================================
Dep. Variable:                    MVP   No. Observations:                  509
Model:                          Logit   Df Residuals:                      488
Method:                           MLE   Df Model:                           20
Date:                Tue, 12 Dec 2023   Pseudo R-squ.:                  0.6276
Time:                        11:47:31   Log-Likelihood:                -40.381
converged:                       True   LL-Null:                       -108.42
Covariance Type:            nonrobust   LLR p-value:                 2.791e-19
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       -153.4205     59.921     -2.560      0.010    -270.863     -35.978
fga            1.9219      1.368      1.405      0.160      -0.759       4.603
fg3a          -0.4148      0.573     -0.723      0.469      -1.539       0.709
fta            0.7761      0.826      0.940      0.347      -0.843       2.395
per           -0.1202      0.633     -0.190      0.849      -1.361       1.121
ts_pct        70.7022     68.027      1.039      0.299     -62.629     204.033
usg_pct        0.3515      0.383      0.919      0.358      -0.398       1.101
bpm            0.3418      0.376      0.909      0.363      -0.395       1.079
win_pct       23.1026      7.134      3.238      0.001       9.121      37.085
g              0.4266      0.368      1.159      0.246      -0.295       1.148
mp_per_g       1.0254      0.697      1.471      0.141      -0.341       2.392
pts_per_g     -1.5955      1.180     -1.353      0.176      -3.908       0.716
trb_per_g     -0.0054      0.276     -0.020      0.984      -0.547       0.536
ast_per_g      0.5061      0.263      1.925      0.054      -0.009       1.021
stl_per_g     -1.2399      0.981     -1.264      0.206      -3.162       0.683
blk_per_g      0.8492      0.564      1.507      0.132      -0.255       1.954
fg_pct        -5.6747     52.227     -0.109      0.913    -108.038      96.689
fg3_pct        0.6182      3.491      0.177      0.859      -6.225       7.461
ft_pct        10.7016     12.947      0.827      0.408     -14.674      36.077
ws            -1.8788      1.790     -1.049      0.294      -5.388       1.630
ws_per_48    134.5557    118.942      1.131      0.258     -98.567     367.679
==============================================================================
```

2. 

```
Accuracy Score: 0.953125
RMSE: 0.20656493932020248
R-squared (R2): 0.3491525423728814
```

3. Using our model we were able to make a plot showing the feature importance to MVP.

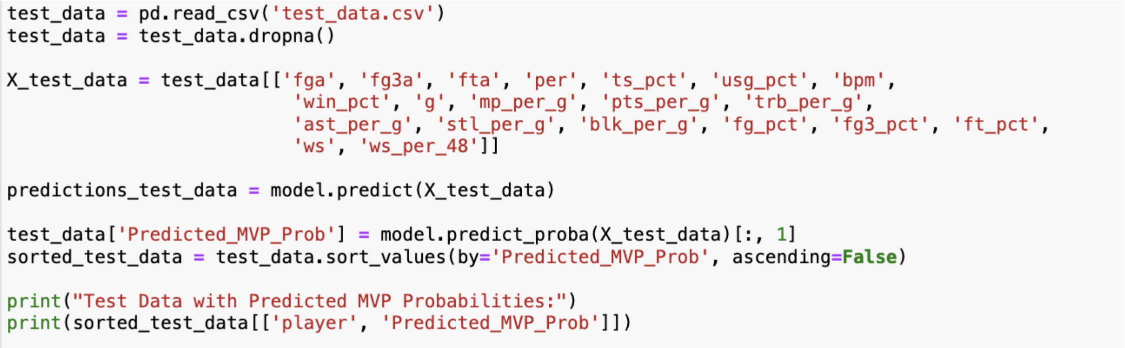


4.
    a) Unsurprisingly, our model showed that winshare was the most relevant statistic in deciding MVP. Steals per game was the statistic most detrimental to winning MVP. This makes sense because steals is typically a statistic that defensive players are more likely to accumulate while offensive minded players are almost always MVP winners. Team win percentage also proved important furthering our assumption that a players contribution to a teams success improves there MVP status
    b) We were surprised that relevant statistics the league is keen to update fans on such as average points per game did not have more impact on the MVP race. We believe this is because a player must contribute to all aspects of the game to benefit his team's success. Assists per game was a high impact statistic, likely because it means the

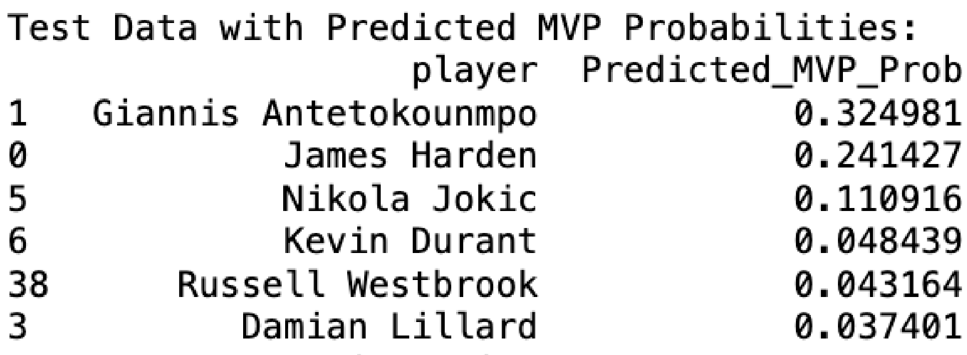
player was creating success for his teammates and adding value to his team that way.

C. Using our model to make predictions

1. To test our model, we used the most recent season data from our data set (which was set aside at the beginning for test data) to gauge the efficiency of our model.

2. 

```
test_data = pd.read_csv('test_data.csv')
test_data = test_data.dropna()

X_test_data = test_data[['fga', 'fg3a', 'fta', 'per', 'ts_pct', 'usg_pct', 'bpm',
                          'win_pct', 'g', 'mp_per_g', 'pts_per_g', 'trb_per_g',
                          'ast_per_g', 'stl_per_g', 'blk_per_g', 'fg_pct', 'fg3_pct', 'ft_pct',
                          'ws', 'ws_per_48']]

predictions_test_data = model.predict(X_test_data)

test_data['Predicted_MVP_Prob'] = model.predict_proba(X_test_data)[:, 1]
sorted_test_data = test_data.sort_values(by='Predicted_MVP_Prob', ascending=False)

print("Test Data with Predicted MVP Probabilities:")
print(sorted_test_data[['player', 'Predicted_MVP_Prob']])
```

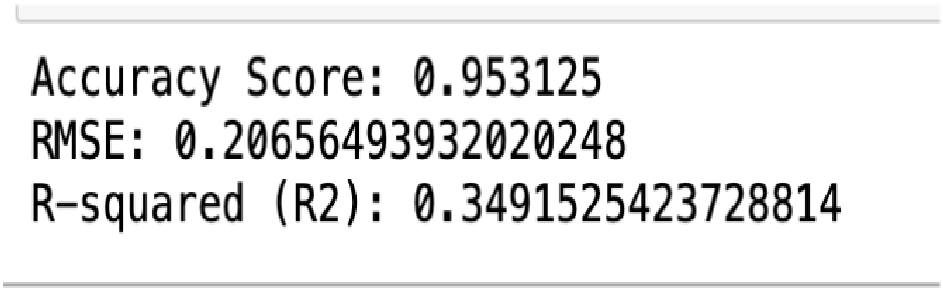
3. 

```
Test Data with Predicted MVP Probabilities:
                   player  Predicted_MVP_Prob
1   Giannis Antetokounmpo            0.324981
0            James Harden            0.241427
5            Nikola Jokic            0.110916
6            Kevin Durant            0.048439
38      Russell Westbrook            0.043164
3          Damian Lillard            0.037401
```

a) Not only did our model correctly predict the 2019 MVP, it also included other players who finished in the top 5 in voting (not in order).

VI. Conclusion & Discussion

A. Our purpose for this report was to successfully and accurately depict previous MVP winners with our regression model as well as implement statistical numbers into the result. The picture below shows our accuracy rate of our overall correctness in prediction demonstrating a high success rate.

1. 

```
Accuracy Score: 0.953125
RMSE: 0.2065649393202O248
R-squared (R2): 0.3491525423728814
```

B. The next steps in our testing and editing process would be to change our regression model into a predictive model. Still including previous stats to accompany for individual players winning the chance of the MVP title. Taking in the factor of newly drafted players as well as surges in player’s play style’s changing, continuous testing of most achieving candidates statistics would play a huge role in determining the next year’s MVP as well as our success rate in a predictive model.

C. We enjoyed this project because we all like basketball and enjoy having corversations about who we think will win MVP.

VII. Distribution and Team Effort

A. Each team member contributed a strong amount to the overall project as we got together and worked on it as a team.

VIII. References

1) “Basketball Statistics & History of Every Team & NBA and WNBA Players.” *Basketball*, www.basketball-reference.com/. Accessed 12 Dec. 2023.

https://www.basketball-reference.com/

2) *The Official Site of the NBA for the Latest NBA Scores, Stats & News. | NBA.Com*, www.nba.com/. Accessed 12 Dec. 2023.

https://www.nba.com/

3) Danchyy. “NBA MVP Votings through History.” *Kaggle*, 14 May 2019, www.kaggle.com/datasets/danchyy/nba-mvp-votings-through-history.