Taqi Hussain

USC ID: 9735993034

Final Project Report

1. **Project Name:** NBA Player Fantasy Points Performance Predictions
   - I used Draft Kings Fantasy scoring system to determine player performance.
   - I collected player statistics for every game of the 2021-2022 NBA season.
   - I converted these statistics to Draft Kings Fantasy Points
   - I split the 2021-2022 season Data into a Train and Test set.
   - I trained 5 machine learning models on the Train Dataset and tested first on the 2021-2022 Test set, and then tested the trained models on a completely new unseen Dataset, the first 2 months of the 2022-2023 NBA season.
   - **Machine Learning Models:** Linear Regression, Decision Tree, Random Forest, K-Nearest Neighbors, XGBoost
   - I compared these models using specific scoring metrics: Mean Squared Prediction Error (MSPE), Mean Absolute Error (MAE), R^2 score
   - The model with the smallest MSPE and an R^2 score closest to 1.0 is the best model for prediction.

2. **How to Run my Code:**
   - Link to GitHub Repository: taqi112/DSCI510_final_project (github.com)
   - In my README.md on GitHub I have listed the dependencies and version numbers. I also have a requirements.txt file on GitHub.
   - One unique library I used was '**fake-useragent'** with version 1.1.1
     - ♦ It can be installed like this: **pip install fake-useragent**
     - ♦ Or like this: **pip3 install fake-useragent**
     - ♦ This library is also in my **requirements.txt** and dependencies list.
   - How to re-produce my results:
     - ♦ My code is a Jupyter Notebook (.ipynb) and it could be run using Cell->Run All
     - ♦ One Potential Problem: Code may take hours to run.
     - ♦ To combat this problem, I recommend reading into pandas the (2022.csv) file right before the heading in my Jupyter Notebook "**DATA ANALYSIS AND PREPROCESSING**".
     - ♦ I also explain this in my Jupyter Notebook. Reading in that CSV file should allow you to execute my Analysis and visualization Cells for that CSV.
     - ♦ You will also need to read into pandas the ('2023.csv') file right before the heading in my Jupyter Notebook "**Read ('2023.csv') INTO PANDAS HERE"**

3. Data Collection:

- The data I collected was each players statistics for every game of the 2021-2022 NBA season. I collected the Box Score data from every game in the season and put it all into a panda DataFrame.
- I also collected each players statistics for both October and November games of the 2022-2023 NBA season. I collected the Box Score data from every game in this range and put it all into a different panda DataFrame.
- **Data Sources:**
    a. https://www.basketball-reference.com/leagues/NBA_2022_games-october.html I replaced the 'October' with every month from October – June and used this list of URL's to collect the last part of the next URL → https://www.basketball-reference.com/boxscores/202110190MIL.html In each iteration I replaced '202110190MIL' with the appropriate text to get data for that specific game.

    b. I did the same as above for the 2023 games using this URL: https://www.basketball-reference.com/leagues/NBA_2023_games-october.html I also replaced 'october' with 'november'. And I used this list of URL's to collect the last part of the next URL → https://www.basketball-reference.com/boxscores/202210180BOS.html In each iteration I replaced '202210180BOS' with the appropriate text to get data for that specific game.
- **How Did I Collect the Data:**
    a. I used both **requests** and **BeautifulSoup** Libraries to scrape the data from the website (URLs above).

    b. I specifically parsed for 'id' attribute, 'tr' tags, 'th' tags within those 'tr' tags, and 'csk' attribute within those 'th' tags.

    c. I also had to parse to get Home Team Abbreviations and Away Team Abbreviations.

    d. For Box Score data I parsed through the element //*[@id="div_box-BRK-game-basic"] updating 'BRK' every iteration to match the correct team's data.

    e. Reference my Jupyter Notebook to see exactly what I did!!
- **Data Samples:**
    a. I generated 3 CSV files from my data.

    b. "2022.csv"

    c. "2023.csv"

    d. "2023_with_predictions.csv"
- **What Changed:** I initially wanted to use an API but couldn't find a reliable one with good documentation. I also wasn't able to scrape for Defensive Ratings as I had planned; given more time I could improve my project by scraping that data and incorporating it into my models.

4. **Analysis and Visualizations:**

   I. **Analysis:**

      i. My primary analysis was to build models that make predictions of the Fantasy Points and compare these models using specific scoring metrics I explained above to determine which model is the best for predictions.

      ii. I found that the Linear Regression model is a perfect fit for the data, making 100% accurate predictions. This makes perfect sense of course because I used as predictors the statistics of the players and the response (Fantasy Points) are a clear linear combination of the player statistics.

      iii. My initial Regression Assumption was: "**Expected values of Fantasy Points follow a regression function**". This was the basis to using Regression Models.

      iv. The Best Model: Closest Model to the Regression Function

      v. Therefore, **Linear Regression was the best model** it's the closest to the regression function in fact it is the regression function. It also had the lowest MSPE and highest R^2 Score.

      vi. When I made predictions on unseen new data (2022-2023 NBA Season), the results were very similar nearly identical.

      vii. **The Worst Model: KNN**

      viii. KNN had the highest MSPE and the lowest R^2 score. The main difference between Linear Regression and KNN is that Linear Regression is a parametric model and KNN is a non-parametric model.

      ix. **Future Work:** If given more time I would try making the relationship between predictors and response less linear by adding more predictors maybe even categorical predictors such as opponent defense rating etc.

   II. **Visualizations:**

      i. I made quite a few different types of visualizations.

      ii. I plotted the distributions of the features and response (Fantasy PTS).

      iii. I also have visualizations of the predicted fantasy points vs actual fantasy points as scatter plots with an OLS trendline for each model.

      iv. I have visualizations of the Top Features for some of my models and how important they were in making the predictions.

      v. I have visualizations of 3 players Actual Fantasy Points vs Predicted Models for the 2022-2023 NBA Season. Players: **Nikola Jokic, Stephen Curry, Luka Doncic**

5. **Future Work:**

   - If given more time I would certainly incorporate more predictors in training my models. More categorical predictors like I said above Defense, Defense by position, home court advantage, etc.

6. **<u>Some Results:</u>**

- Below I will report some of my results not all visualizations but some.
- All of my results will be in the **Results Folder**

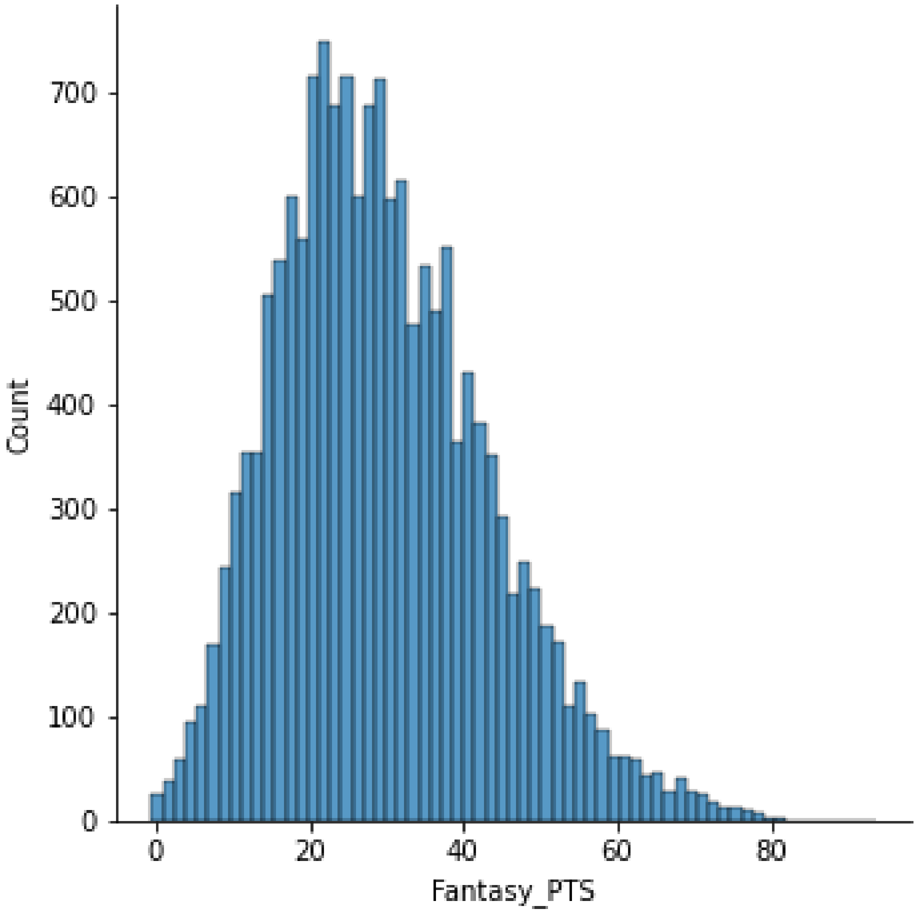
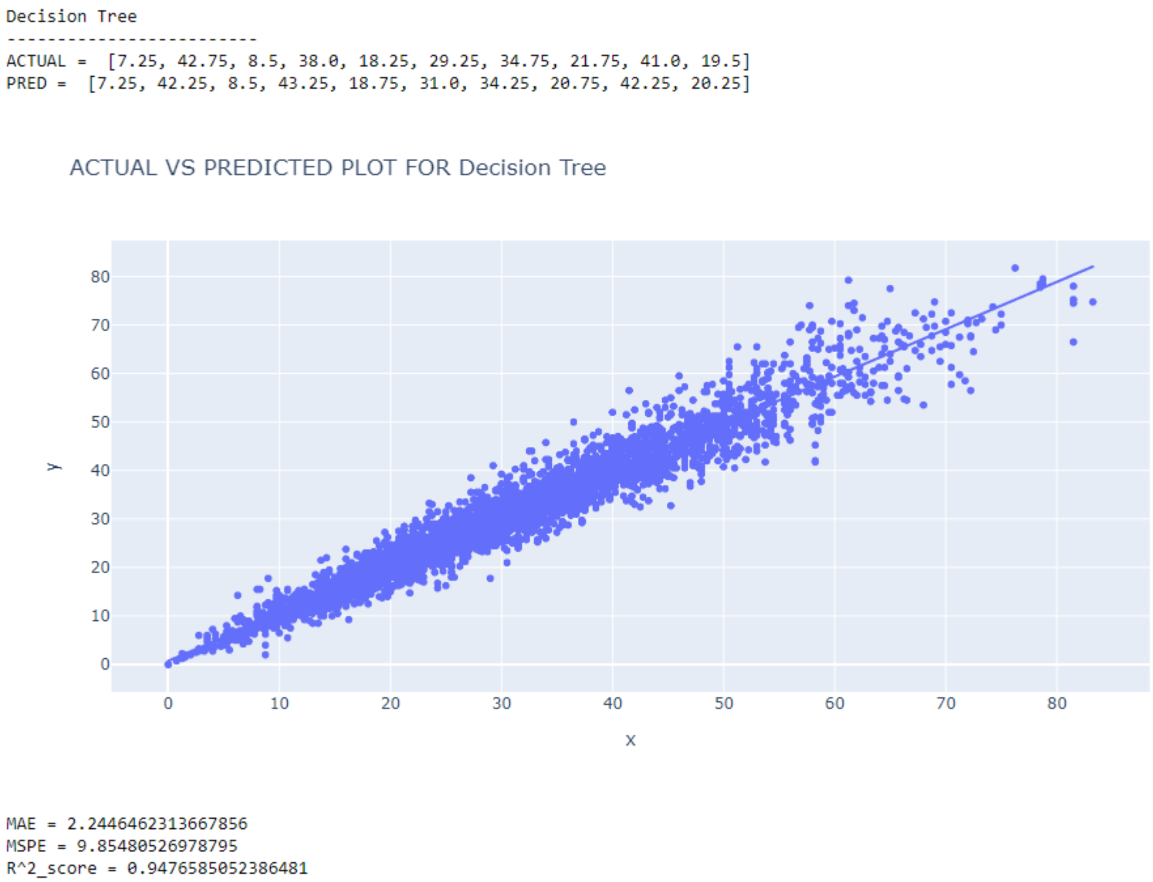


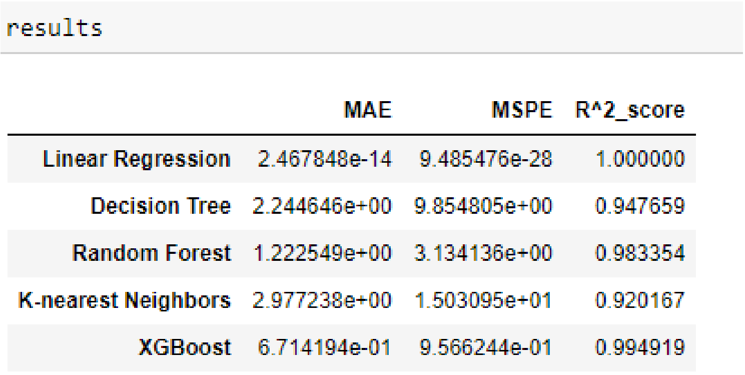
Figure 1: Distribution of Fantasy_PTS



```
Decision Tree
-------------------------
ACTUAL =  [7.25, 42.75, 8.5, 38.0, 18.25, 29.25, 34.75, 21.75, 41.0, 19.5]
PRED =  [7.25, 42.25, 8.5, 43.25, 18.75, 31.0, 34.25, 20.75, 42.25, 20.25]
```

```
MAE = 2.2446462313667856
MSPE = 9.85480526978795
R^2_score = 0.9476585052386481
```

Figure 2: Decision Tree model



```
results
```

	MAE	MSPE	R^2_score
Linear Regression	2.467848e-14	9.485476e-28	1.000000
Decision Tree	2.244646e+00	9.854805e+00	0.947659
Random Forest	1.222549e+00	3.134136e+00	0.983354
K-nearest Neighbors	2.977238e+00	1.503095e+01	0.920167
XGBoost	6.714194e-01	9.566244e-01	0.994919

Figure 3: Model Comparison by scoring metric

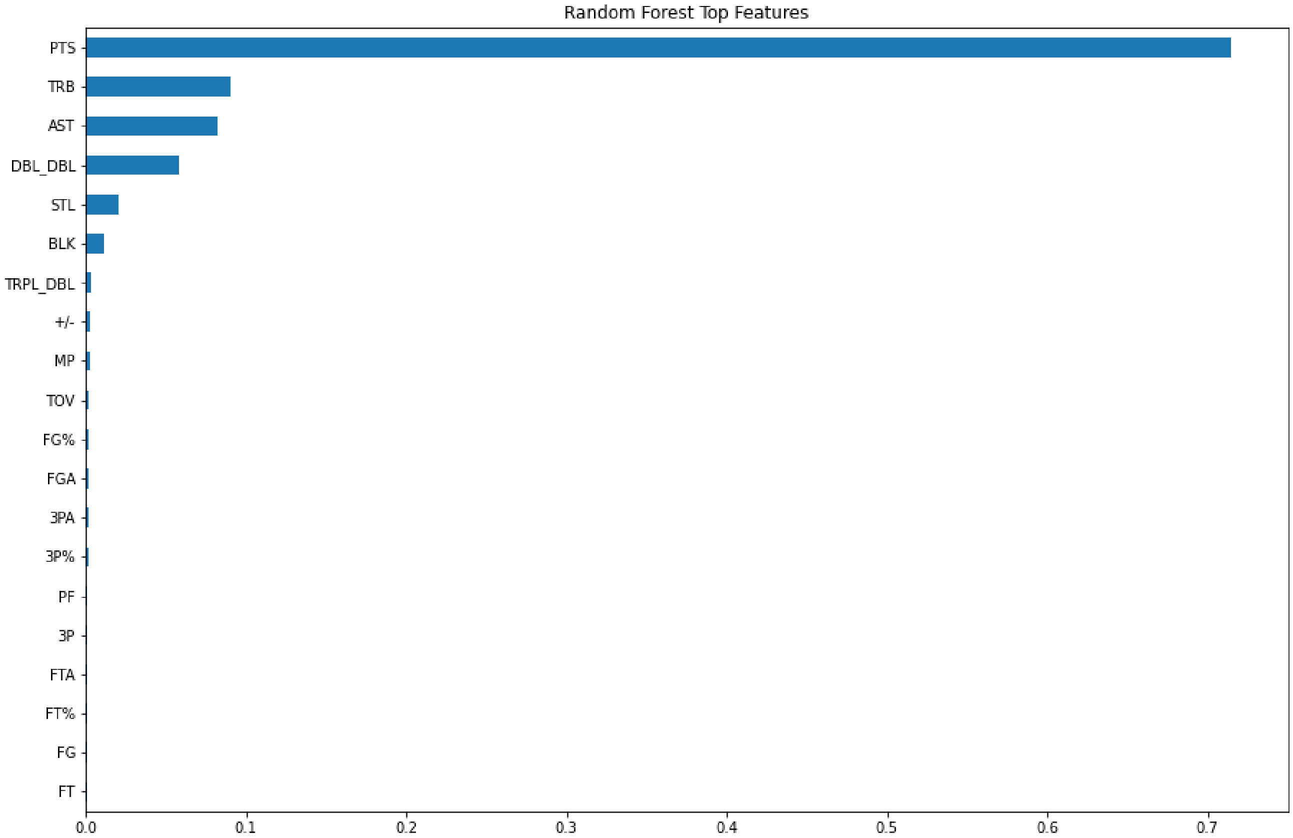


Figure 4: Random Forest Top Features

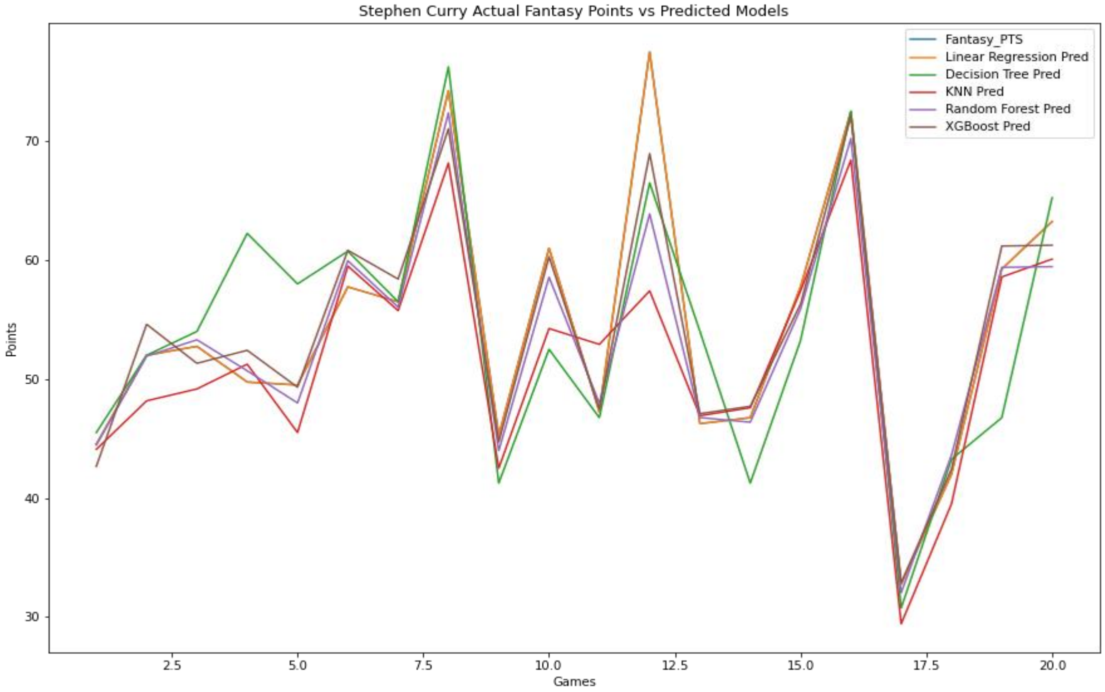


Figure 5: Stephen Curry Fantasy Points vs Models