Analysis and Prediction of Fantasy Premier League (FPL) Performances

Project Report

Candidate no: 219

Contents

List of Figures

1. Overview

1.1 Background

The beautiful game, or football as its named is the biggest sport in the world. The importance that this seemingly simple sport has can be summed up in a quote given by the great Liverpool manager Bill Shankly. He once said *"Some people believe football is a matter of life and death. I am very disappointed with that attitude. I can assure you it is much, much more important than that."* This sentiment still exist to some degree. Fans of the sport from all around the world sit down every day to watch matches on tv. One of the biggest leagues is the english top flight division. The English Premier League, also know as EPL.

Fantasy Premier league, often shortened to FPL. Is a online football game that allows players to manage a team based on players currently playing the English Premier League. The game allows players to live out their dream of being a football manager. It heightens each game week and makes less interesting games more intense since you might have a player playing on your fantasy team. It raises the stake from just cheering on you favorite, team to analyzing matches looking to gain an edge on your friends that are in the same mini-league. There is so much data that is recorded every week on each players performance, that there are dedicated online personalties that break the data down and give you insights into the game weeks.

To better understand FPL one needs to understand the rules and how the game is played. FPL is tightly connected to the EPL, before the season starts, every FPL manager is given a pot of 100 million in game currency and is tasked with selecting 15 players. Given that players are valued at different prices, this introduces the first limitation set on the managers. Next limitation imposed on the managers are that there can only be certain numbers of players for each position and from each team you can only pick 3 players. For the different positions you need to pick:

- 2 goalkeepers
- 5 Defenders
- 5 Midfielders
- 3 Forwards

Finding the right players is crucial. When the season starts you are only given 1 transfer a week, and you can only roll the transfer one week at a time. Giving you a max free transfers of 2 at any time. You are allowed to transfer more then this however this will be deducted from you total points. Its called taking a hit.

Now that the team as been chosen, every week you field 11 players. You can set you team up as you like as long as there is at least 1 goalkeeper, 3 defenders and 1 forward. Essentially this means you can play many different formations. Each game week every player is awarded points based on their real life performance. A goal scored by a forward gives you 4 points, while a goal scored by a defender

or a goalkeeper gives you 6 points. The aim of the game is to collect as many points as possible through out the season.

There are some extra details that have been left out for brevity, however one can see that there are myriad of moving parts, for a manager to control. Every thing from rotating the team based on players opponent, selling and buying players. Most managers pick players based on the eye test. This is more or less based on how they feel about a player. There is a lot of luck in FPL, given all the factors that impact a players performance. However there are statistical data that is also collected on each player. From the number of minutes played, to number of big changes created. This data aggregated can tell us something about the game and also about the players.

In this report we will dive into the numbers behind the game and see if we cant make sense of the data.

1.2 Objectives

The objective of this report is to look at some of the data that is recored over the years. We will try to answer one central question. What is the biggest contributor to a players total points. Answering this allows us to better understand how to pick a player. Based on the answer we get, we can then hopefully make predictions on players performance and points gained in a game week.

The way we will work though the data is to first clean the data so that we are sure we have usable data. We will then start doing some exploratory data analysis (EDA) to better understand the data we have. Before trying to answer our main question. There might be some side questions that we might need to address in order to fully understand what the data is telling us.

2. Data Preparation

2.1 Data Description

The dataset used in this report is collected from a Github repository (*Vaastav/Fantasy-Premier-League*, n.d.). This repository is not affiliated with FPL or the EPL, but it is highly regarded in the FPL community as its been used in numerous credible projects. The creator is also a credible source given the credentials on his website (*Vaastav Anand*, n.d.). The open source nature of the code also lends to this credibility.

Notable uses of this database include:

- 2019-20 Winner Joshua Bull's Oxford Maths Public Lecture (Oxford Mathematics, 2020)
- How to win at Fantasy Football with Splunk and Machine Learning ("How to Win* at Fantasy Football with Splunk and Machine Learning [Part 1]," n.d.)

The datasets is deemed credible and is therefor used as the basis for the analysis in this report.

In the GitHub repository every game week since the 16/17 season has been saved in separate csv files. There is also a end of the season merged csv file created. This includes all the data collected from that season for each game week. So one row will be for a specific players game week, and that player will appear 38 times. One row for each of the 38 game weeks.

2.2 Preprocessing

In the different seasons there are differences. In the newer seasons there has been some additional information collected that is not present in previous seasons dataset. Notable is the information about position and team of player. This is first introduced in the 20/21 season and onwards. This does weaken the dataset overall however the given that positional information will not be taken into account for this analysis we can disregard this discrepancy.

Some of the formatting also changes thought the seasons. The biggest inconsistency is the naming of players. One season has the players name as "First name Last name" with space between, then an other season has "First name_Last name_player id". Making it hard to aggregate players over multiple seasons. There is also the issue of encoding of the csv files. When importing the data into pandas the standard encoding of UTF-8 is rejected and throws an error. The correct encoding is ISO-8859-1, yet there are special characters introduced in the names. This seems to however be done consistently so the name is wrong but in the same way. The number of names with these special characters are roughly 237 players across 7 seasons. There are instances where the player name is correctly formatted in one season but not in a other, giving us two names for

one player. In order to combat this. We need to first formate the names correctly. So that special characters, underscore, extra spacing and numbers are removed. We then need to look at if there are names that are similar in the dataset but slightly different. There could be two players with similar names, or it might be a spelling error of one player. In order to determine this fuzzy search is deployed to find similar names.

Fuzzy search is a technique used to search for matches that are not 100% exact. The way this is done is by measuring how many changes needs to be done to a word or a string of characters in order for it to be the exact same. For our fuzzy search through the names we used *Levenshtein Distance* (Haldar & Mukhopadhyay, 2011) this allows us to find names that match depending on a threshold set. The threshold is expressed as percentage with 100 being an exact match. Through trial and error, 99% match was chosen as the threshold.

The fuzzy search revealed 122 pairs of similar names between players. Some of the names where just similar, but others were variations of the same name, either misspelled or containing middle names. The ones which were deemed variations on the same player, where then standardized and updated in the dataset.

The code used for both cleaning the dataset and the fuzzy search on the names can be found in Appendix A.

3. Statistical Analysis

The dataset includes many parameters. They are not equally interesting, for our goal, but to determine which parameters are valuable we need to see if there is any correlation between our main parameter total points and all the other parameters.

There are several ways to find correlation between datapoints, there can even be datapoints that don't directly correlate but they impact other datapoints that do correlate, making them correlate indirectly. We can exclude some parameters like player names, game week, etc. There might be an argument that later game weeks have a psychological effect on the players, but such effects are to far fetch for this study and as such will be excluded. The datapoints we are interested in are the values that numerically describe a players performance or directly impact his ability to gain points. Things such as minutes played, attempted shots at goal and other similar variables.

The following are a list of the parameters deemed most interesting that we will continue to work with:

- XP
- Assists
- Bonus
- Bps
- Clean Sheets
- Creativity
- Expected Assists
- Expected Goal Involvements
- Expected Goals
- Expected Goals Conceded
- Goals Conceded
- Goals Scored
- ICT Index
- Influence
- Minutes
- Own Goals
- Penalties Missed
- Penalties Saved
- Red Cards
- Saves
- Threat
- Value
- Yellow Cards
- Attempted Passes
- Big Chances Created
- Big Chances Missed
- Clearances Blocks Interceptions

- Completed Passes
- Dribbles
- Ea Index
- Errors Leading To Goal
- Errors Leading To Goal Attempt
- Fouls
- Key Passes
- Offside
- Open Play Crosses
- Penalties Conceded
- Recoveries
- Tackled
- Tackles
- Target Missed
- Winning Goals

All of these parameters are aggregated over all the seasons. Some are summed up, some are averaged out depending on what they measure. Examples like expected points (xP) of a player, here we look at the mean of this parameter since that will give us the best look at what the usual is like. Other parameters have a negative effect as for instance a yellow card will get a player deducted -1 points. We therefore sum this parameter in the negative to see total impact.

Total Points

Before we start working further with the aforementioned variables and try seeing how they are related to total points, lets take a closer look at the most important metric in FPL.

The total points at the end of a season is what matters in the game of FPL. The entire goal of the game is to consistently collect the most amount of points and in-order to achieve this the players one picks has to also achieve the most amount of points. In the following chart we can see a breakdown of how the distribution of total points across all seasons looks.

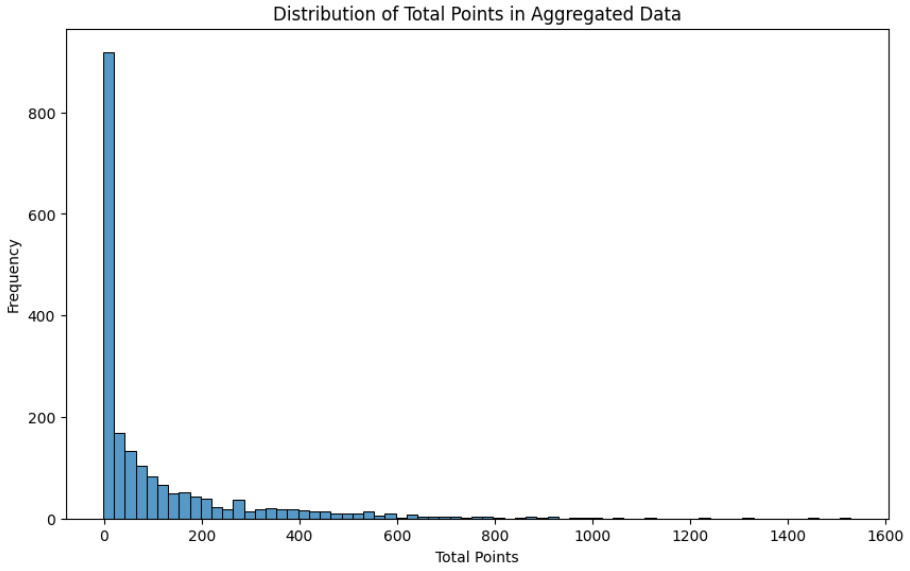

Figure 1: Distribution of Total Points

This histogram shows us that total points is a heavily skewed metric. Most players collect considerable less points then the most prolific players. The overall large span of points however shows that there is a diverse set of points collected, but that the overall total points are gathered by a minority. Further analysis of the histogram shows us that 75% of the players collected less the 140 points in total, while the highest scorer has reached 1530 points. This discrepancy can be attributed to many players entering the EPL and not either getting a lot of minutes on the field or leaving due to injury or transfers. The mean lies at 110, and there is in total 1983 players counted.

To get a better read on the we can use a box plot.

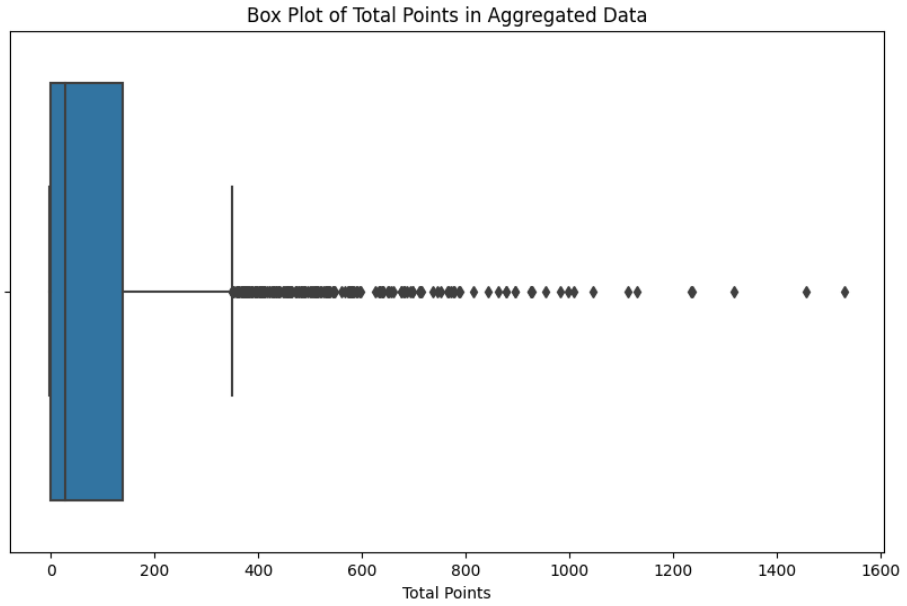

Figure 2: Box Plot of Total Points

The box plot shows us what might be intuitively understood from the histogram. The line in the the box is the median total points, this is set at 29 points. The complete box represents the interquartile range (IQR) (*InterQuartile Range (IQR)*, n.d.). Here we can see the spread of the central data. This box represents the middle 50% of the data. When calculating the IQR we can get what the 25th, 50th and 75th percentile are. We can see that around 25% (Q1) of players scored 0 or less points they are represented by the lower edge of the box. This might explain the extreme right skew the we saw in the histogram. We can also see that the outliers are extreme. The bound calculated through IQR, sets the upper bound to 350 considering everything above this an outlier and we can see that some of the biggest outliers exceeded this by a lot. The lower bound being -210 is bit of a misrepresentation given that it would be very unlikely for a player to achieve this and the lowest total points calculated is -2.

4. In-Depth Analysis

4.1 Identifying Outliers

When wanting to find what effects total points the most we need to understand how the average players collects points. We also need to look at who the outliers are. Again these skew the data so much that we need to take a closer look.

From the box plot we got that the players outside the upper bound are outliers. Looking at the number of players that are therefor classified as outliers we see this amounts to 202 players. However even the outliers are not created equal. Below we se a new box plot just containing the outliers. Here we can se that we have 10 players that are outliers. This lets us know that there is a lot of variability in the dataset. Looking at the total points accumulated at 219'184 for all players, the 202 outliers represent a major portion the points collected at 112'966. More then half the amount of total points across all the seasons we have looked at are collected by 10% of the players.

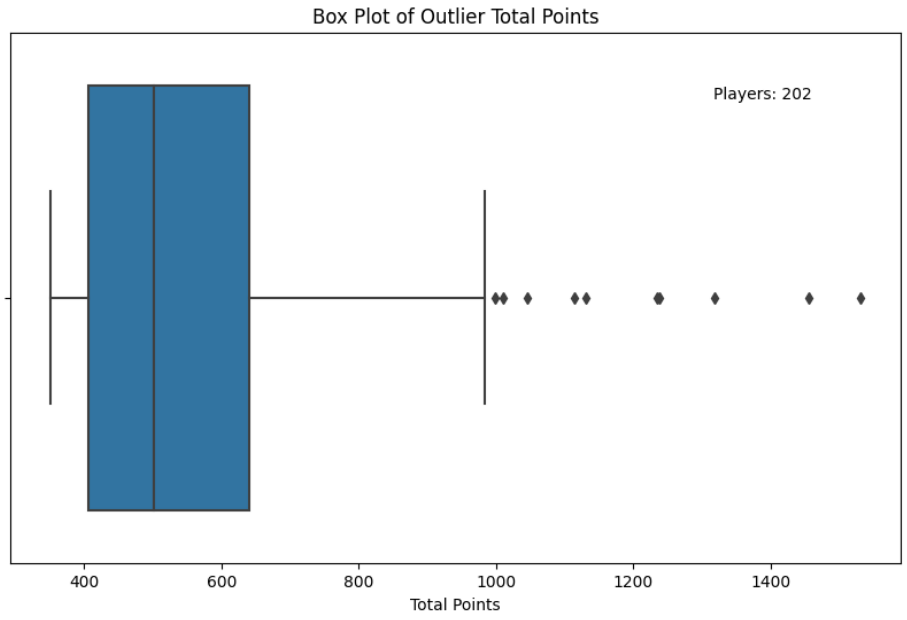

Figure 3: Box Plot of Total Points (Outliers)

When we now take a look at the non-outliers we se that the data is less spread out an has a better conciseness. We will however keep the outliers in our data since they account for such a high portion of the total points. Removing them would cause a drastic shift in the data that might not be representative of the actual data.

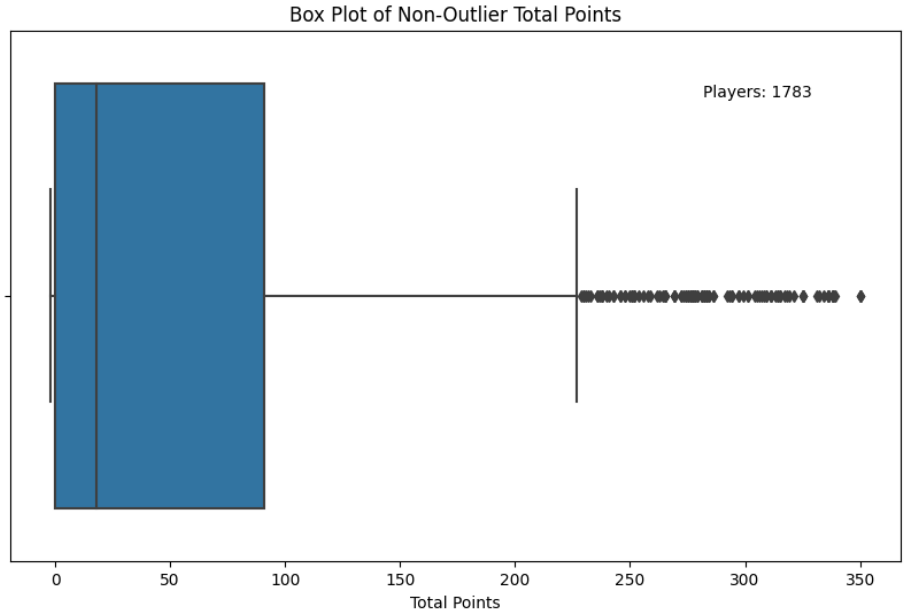

Figure 4: Box plot of Total Points (Non-outliers)

4.2 Correlation Study

When trying to find the parameters that has the most effect on a players total points we can conduct a correlation study. This is a methodology that looks a the relationship between two variables and determines the strength of said relationship. We used Pearson correlation coefficient (*Numeracy, Maths and Statistics - Academic Skills Kit*, n.d.) to determine if there was a linear correlation between any of the aforementioned variables to total points. The Pearson correlation coefficient ranks correlations between -1 and 1, -1 meaning a perfect negative correlation and 1 meaning the same just positive. Below we have a heat map showing the 5 most positive and 5 most negative correlations.

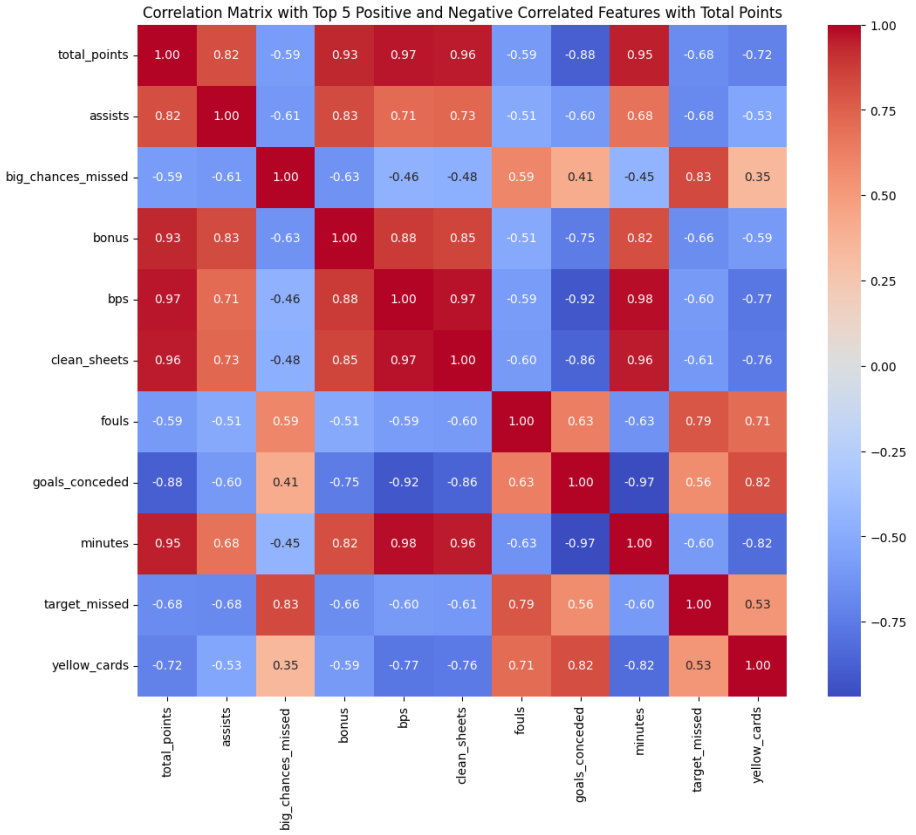

Figure 5: Heat map of Correlation Matrix

When viewing the correlation matrix we see that there are multiple variables with high correlations to total points. We have *bonus*, *minutes*, *clean sheets* and *BPS* as the closest. While variables such as *goals conceded* and *yellow cards* have the biggest negative impact.

There is an argument to say that the a linear correlation might not be the best way to look at the correlation between these variables and total points. In order to asses this better, lets take a closer look at different variables and see what they represent.

Rest of the top 5

Most of the variables are fairly self explanatory however here is a short explanation of each.

Minutes: This is the number of minutes a player plays in a game. The more he plays the better chance he has of accumulating points. 1 point is awarded for entering a game, and a second point is given for playing more then 60 min (*How the FPL Bonus Points System Works*, n.d.).

Clean Sheet: One of the key objectives in football is not letting the other team score. Points are awarded and deducted depending on of you keep a clean sheet, meaning no goals are scored, or if you let to many goals in. The points awarded depend on the position a player plays. A clean sheet for a goalkeeper or a defender is 4 points. For a midfielder they get 3 points. Strikers are not awarded any points for this. 1 point is deducted for a goalkeeper or a defender for every 2 goals they let in. Midfielders are not effected negatively by this metric (*How the FPL Bonus Points System Works*, n.d.).

Bonus: This is directly tied to BPS and is given out based on how high you bps is for that game. You can get 3, 2 or 1 points (*How the FPL Bonus Points System Works*, n.d.).

BPS

In FPL there is a lot off different variables that are used to track a players performance that are not presented to the public. Some of the more obvious ones like expected goals pr 90 min played are posted on the FPL website but, in game a football there are so many things to keep track of. Companies such as OPTA ("Opta Data from Stats Perform," n.d.) calculate and track most these statistics and these are then used to calculate a score called Bonus Point System (BPS) (*How the FPL Bonus Points System Works*, n.d.). This score is done using a proprietary formula, that ends up awarding 3 points to the player with the highest score that game, 2 to the second highest and 1 to the third. Essentially this can be seen as a best man award where the best player gets bonus points.

Below we have a graph showing the linear relationship between BPS and total points. Here we use linear regression to fit a function that best describes the all the datapoints. This is done by utilizing a process called *the least squares fitting*, it tries to draw a line such that distance away from each datapoint is as small as possible ("An Introduction to Statistical Learning," n.d.). We further calculate the "correctness" of our line. R^2 is a value between 0 and 1, it sees how much of deviation in total points can be explained there by BPS. The closer to 1 the

more variation can be explained by bps. Meaning that a $R^2 = 0.94$ indicates that around 6% of variation is due to other factors.

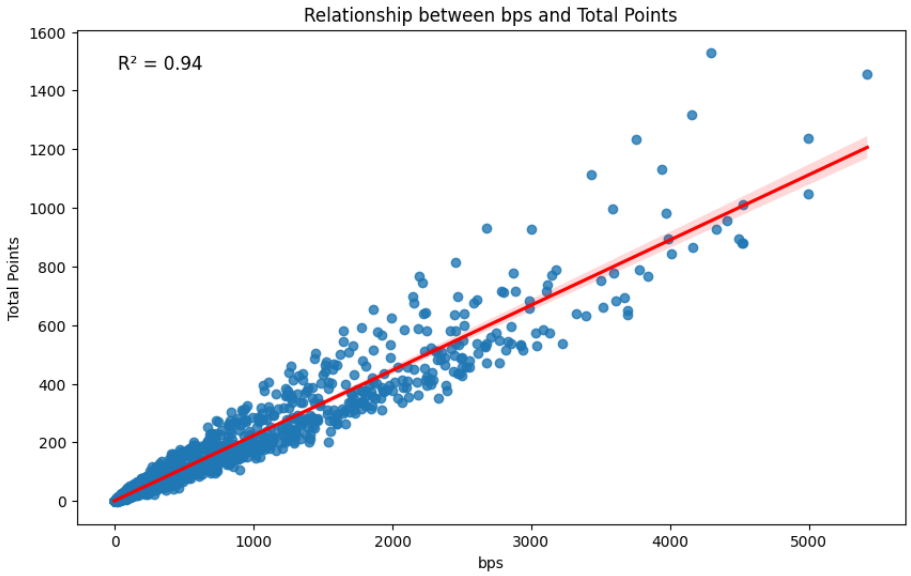

Figure 6: Relationship between BPS and Total points

Finding a perfect linear relationship between this variables and total points might be impossible due to the many many factors that come into a players achievement on the field. Everything from injury, team and even just a bad day has an effect making a linear relationship misguided. However for the sake of this study we find it worthwhile to simplify and see things as more minutes as more points. This simplification of the variables allows us to take a novel look and maybe if there is a cursory correlation a casual player might be able to use in their player selection.

5. Regression Modeling

5.1 Model Construction

Now that we have found the most impactful variables correlated to total points we can try to see how well we can predict the total points a player will make. For this we create a model using *Multiple Linear Regression*. We create formula using the different positive variables we earlier found. We give each parameter a weight that indicates its importance. This is calculated by training the model on a portion of the dataset. We allocate 80& of the dataset for training. The training goes on until it finds the best balance. These coefficients are then added to the formula that predicts the total points, this is then projected on the remaining dataset [("An Introduction to Statistical Learning," n.d.)].

5.2 Evaluation

Following are the different metrics used to plot the predict line:

Metric	Value
Mean Squared Error	362.53
R-squared	0.99

Features included in the prediction model

- BPS
- Clean Sheets
- Minutes
- Bonus
- Assists

The *Mean Squared Error* tells us average squared difference between the actual total points and predicted. We can see from the table that this is at 362.53 points. Comparing this to the mean of all total points (29) we can se that our model is quite of from this. This might be due to how the training data is split. If more of the outliers are included in the training data we might see it more skewed towards a higher mean. Looking at figure 2 we can see that the upper bound is 350 and given that more then half the total points accumulated come from players above this line we can say that the MSE is more accurate then it might seem.

R^2 is almost at 1, indicating an almost perfect prediction. This might indicate that there might be an issue with overfitting. Given that we don't have any

other independent data its hard to completely rule this out. The more likely scenario is that our simplification of using linear correlations has had an adverse effect on the model.

5.3 Interpretation

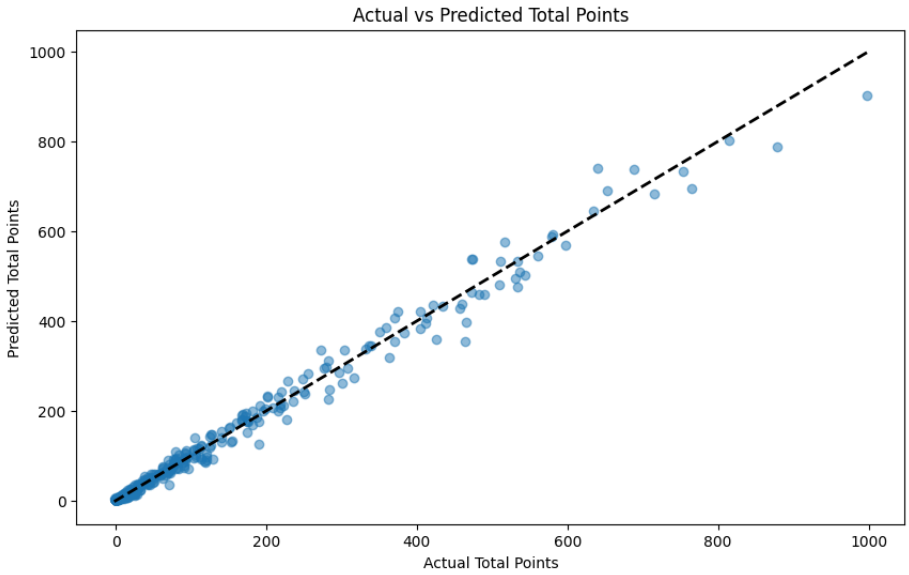

Figure 7: Actual vs Predicted Total Points

When looking at the predicted line in the scatterplot we can see that we have a very tight line across the actual datapoints. This indicates that our predictions is very accurate. We can see that the as we move along the dotted line the accuracy starts falling and the distance between the line and the datapoints start increasing. This tells us that our model is getting less accurate as it progresses. This might indicate overfitting, however the majority of the datapoints are very close to the line. Giving reason to think that it stills holds some what of a valuable insight. The factors not included and the choice of treating the parameters as linear might be causing this.

6. Findings and Insights

6.1 Main Findings

In reviewing both the correlation study and the predictive model generated, we can asses that even though there is a more complex and intertwined relationship between the different features and in regards to their impact to total points. We can say that the features found in the correlation matrix most likely have a strong correlation with total points. The predictive model might be somewhat idealistic in its prediction, but it still gives a good indication that that those features in the model are the features have the most impact.

6.2 FPL Strategy

With the knowledge gained from this study one can start looking at how this could be applied. There insights gained from this study both strengthen some of the perceived features that where most important. There where new insights that where interesting to note. The fact that neither *value* or xP points where of not much consequence relative to total points seems counter intuitive. Taking the results at face value one can extrapolate that lagging variables based on a single metric such as *value* and xP are poor indicators of performance. More complex and feature engineered parameters consisting of multiple other features might be better. Even though *minutes* which is a single variable feature is among the top features. This might be because it directly impacts performance. A player needs minutes to gain points. These more direct variables are more correlated to points then the secondary variables.

6.3 Limitations

The novel nature of this study limits the scoop and complexity used to analyze this dataset. One limitation this study has is its assumption of a linear relationship between the different parameters in regards to their effect on total points accrued by a player. The game of football and by extensions FPL is a complex in nature. The many factors from 3 or 4 degrees separated from total points are vast. Reducing the complexity gave us a simpler and more clear insight, however this insight has inherent flaws that acts as a limitation.

7. Conclusions

This study tried to take a closer look at the relationship between total points and other variables that is collected to describe a players performance, to see how if we could predict a players total points for a season. This was done using a dataset that contains many different variables. We conducted a linear correlation study to see how the different features correlated to the variable total points, assuming they had a linear relationship. This correlation study indicated that the following features had the most positive bearing on the total points accumulated by a player:

- BPS
- Clean Sheets
- Minutes
- Bonus
- Assists

The result from the correlation study was then used in a multivariable linear regression in order to predict the total points accumulated. We found that the predictive model was quite accurate in its prediction, giving a R^2 of 0.99, however the *MSE* was at 362.52, while the mean for total points where at 29. Indicating there might be a non-linear relationship between the different parameters and total points.

8. References

An Introduction to Statistical Learning. (n.d.). In *An Introduction to Statistical Learning*. https://www.statlearning.com.

Haldar, R., & Mukhopadhyay, D. (2011). Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. *Computing Research Repository - CORR*.

How the FPL Bonus Points System works. (n.d.). https://www.premierleague.com/news/106533.

How to win* at Fantasy Football with Splunk and Machine Learning [Part 1]. (n.d.). In *Splunk-Blogs*. https://www.splunk.com/en_us/blog/tips-and-tricks/how-to-win-at-fantasy-football-with-splunk-and-machine-learning-part-1.html.

InterQuartile Range (IQR). (n.d.). https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_summarizingdata/bs704_summarizingdata7.html.

Numeracy, Maths and Statistics - Academic Skills Kit. (n.d.). https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/strength-of-correlation.html.

Opta data from Stats Perform. (n.d.). In *Stats Perform*. https://www.statsperform.com/opta/.

Oxford Mathematics. (2020). *Can maths tell us how to win at Fantasy Football? - Joshua Bull*.

Vaastav Anand. (n.d.). https://vaastavanand.com/.

Vaastav/Fantasy-Premier-League: Creates a .csv file of all players in the English Player League with their respective team and total fantasy points. (n.d.). https://github.com/vaastav/Fantasy-Premier-League.

9. Appendices

Appendix A

Code

```
"""
Resources used to create code:
https://chat.openai.com/
https://bard.google.com/
https://stackoverflow.com/
https://www.statlearning.com/
https://www.youtube.com/@NeuralNine
"""

# Import libraries
import os
import pandas as pd
import re
import unicodedata
import seaborn as sns
import matplotlib.pyplot as plt
from fuzzywuzzy import process, fuzz
from sklearn.metrics import r2_score
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score

# Function to clean and format names
def clean_name(name):
    normalized_name = unicodedata.normalize('NFKC', name)
    name = normalized_name.replace('_', ' ')
    filtered_name = re.sub(r'[^a-zA-ZÀ-ÖØ-öø-ÿ \-]', '', name)
    joined_name = ' '.join(filtered_name.split())
    titled_name = joined_name.title()
    return titled_name

# Function to find similar names using fuzzy search
def find_similar_names(df, column_name='player_name', threshold=99):
    similar_names_dict = {}
    names = df[column_name].tolist()
    exclude_names = ['Kyle Walker', 'Kyle Walker-Peters']
    for name in names:
```

```
        if name in exclude_names:
            continue
        matches = process.extractBests(name, names,
        scorer=fuzz.token_set_ratio, score_cutoff=threshold)
        filtered_matches = [match for match, score in matches
        if match != name and match not in exclude_names and match
        not in similar_names_dict]
        if filtered_matches:
            similar_names_dict[name] = filtered_matches
    return similar_names_dict


def tally_similar_names(similar_names_dict):
    return sum(len(matches) for matches in similar_names_dict.values())


# Load different csv files, combine them into one dataframe
dataset_dir = "/content/Dataset"
combined_df = pd.DataFrame()
for file_name in os.listdir(dataset_dir):
    if file_name.endswith('.csv'):
        file_path = os.path.join(dataset_dir, file_name)
        df = pd.read_csv(file_path, encoding='ISO-8859-1')
        df['season'] = file_name.replace('.csv', '')
        combined_df = pd.concat([combined_df, df], ignore_index=True)



# Clean player names in the dataset
combined_df['clean_name'] = combined_df['name'].apply(clean_name)
player_names_df = pd.DataFrame(combined_df['clean_name'].unique(),
columns=['player_name'])

# Apply fuzzy search to find and handle similar names
similar_names = find_similar_names(player_names_df)
similar_names = {key: value for key, value in similar_names.items()
if 'Kyle Walker-Peters' not in value}

# Standardize names in the DataFrame
for original_name, similar_names_list in similar_names.items():
    for similar_name in similar_names_list:
        combined_df.loc[combined_df['clean_name'] == similar_name,
        'clean_name'] = original_name
```

```
combined_df.loc[combined_df['clean_name'].isin(['Bobby De Cordova-Reid',
'Bobby Reid']), 'clean_name'] = 'Bobby De Cordova-Reid'
player_names_df = pd.DataFrame(combined_df['clean_name'].unique(),
columns=['player_name'])

total_similar_pairs = tally_similar_names(similar_names)
print(f"Total similar name pairs found: {total_similar_pairs}")

# Save the cleaned data to a CSV file
combined_df.to_csv('/content/updated_combined_data.csv', index=False)


# Aggregate different features
aggregations = {
    'xP': 'mean',
    'assists': 'sum',
    'bonus': 'sum',
    'bps': 'sum',
    'clean_sheets': 'sum',
    'creativity': 'mean',
    'expected_assists': 'mean',
    'expected_goal_involvements': 'mean',
    'expected_goals': 'mean',
    'expected_goals_conceded': lambda x: -x.mean(),
    'goals_conceded': lambda x: -x.sum(),
    'goals_scored': 'sum',
    'ict_index': 'mean',
    'influence': 'mean',
    'minutes': 'sum',
    'own_goals': 'sum',
    'penalties_missed':lambda x: -x.sum(),
    'penalties_saved': 'sum',
    'red_cards': lambda x: -x.sum(),
    'saves': 'sum',
    'threat': 'mean',
    'total_points': 'sum',
    'value': 'mean',
    'yellow_cards': lambda x: -x.sum(),
    'attempted_passes': 'sum',
    'big_chances_created': 'sum',
    'big_chances_missed': lambda x: -x.sum(),
    'clearances_blocks_interceptions': 'sum',
    'completed_passes': 'sum',
    'dribbles': 'sum',
```

```
    'ea_index': 'first',
    'errors_leading_to_goal':lambda x: -x.sum(),
    'errors_leading_to_goal_attempt': lambda x: -x.sum(),
    'fouls': lambda x: -x.sum(),
    'key_passes': 'sum',
    'offside': lambda x: -x.sum(),
    'open_play_crosses': 'sum',
    'penalties_conceded': lambda x: -x.sum(),
    'recoveries': 'sum',
    'tackled': 'sum',
    'tackles': 'sum',
    'target_missed': lambda x: -x.sum(),
    'winning_goals': 'sum'
}


# Aggregate player data by season and overall
df_grouped_season = combined_df.groupby(['clean_name', 'season']).
agg(aggregations).reset_index()
df_grouped_overall = combined_df.groupby('clean_name').agg(aggregations)
.reset_index()

df_descriptive = df_grouped_overall.copy()

# Calculate and print descriptive statistics for 'total_points'
descriptive_stats = df_descriptive['total_points'].describe()
print(descriptive_stats)

# Print graphs
plt.figure(figsize=(10, 6))
sns.histplot(df_descriptive['total_points'], kde=False)
plt.title('Distribution of Total Points in Aggregated Data')
plt.xlabel('Total Points')
plt.ylabel('Frequency')
plt.show()


plt.figure(figsize=(10, 6))
sns.boxplot(x=df_descriptive['total_points'])
plt.title('Box Plot of Total Points in Aggregated Data')
plt.xlabel('Total Points')
plt.show()

# Calculate and print key statistics
```

```
descriptive_stats = df_grouped_overall['total_points'].describe()

median = descriptive_stats['50%']
q1 = descriptive_stats['25%']
q3 = descriptive_stats['75%']
iqr = q3 - q1
min_value = descriptive_stats['min']
max_value = descriptive_stats['max']

lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

print(f"Median: {median}")
print(f"First Quartile (Q1): {q1}")
print(f"Third Quartile (Q3): {q3}")
print(f"Interquartile Range (IQR): {iqr}")
print(f"Minimum Value: {min_value}")
print(f"Maximum Value: {max_value}")
print(f"Lower Bound for Outliers: {lower_bound}")
print(f"Upper Bound for Outliers: {upper_bound}")


# Identify and analyze outliers in total points
q1 = df_descriptive['total_points'].quantile(0.25)
q3 = df_descriptive['total_points'].quantile(0.75)

iqr = q3 - q1
outlier_condition = (df_descriptive['total_points'] < (q1 - 1.5 * iqr)) |
(df_descriptive['total_points'] > (q3 + 1.5 * iqr))
outliers = df_descriptive[outlier_condition]
non_outliers = df_descriptive[~outlier_condition]

total_points_sum = df_descriptive['total_points'].sum()

non_outlier_total = non_outliers['total_points'].sum()
outlier_total = outliers['total_points'].sum()

outlier_q1 = outliers['total_points'].quantile(0.25)
outlier_q3 = outliers['total_points'].quantile(0.75)

outlier_iqr = outlier_q3 - outlier_q1

outlier_outlier_condition = (outliers['total_points'] < (outlier_q1 -
1.5 * outlier_iqr)) | (outliers['total_points'] > (outlier_q3 + 1.5 * outlier_iqr))
```

```
outlier_outliers = outliers[outlier_outlier_condition]


num_outliers = outliers.shape[0]
num_non_outliers = non_outliers.shape[0]
num_outlier_outliers = outlier_outliers.shape[0]


print(f"Total outliers: {num_outliers}")
print(f"Outliers within outliers: {num_outlier_outliers}")
print(f"Total points of all players: {total_points_sum}")
print(f"Total points of non-outliers: {non_outlier_total}")
print(f"Total points of outliers: {outlier_total}")

# Print graphs
sns.boxplot(x=non_outliers['total_points'])
plt.title(f"Box Plot of Non-Outlier Total Points - Players: {num_non_outliers}")
plt.xlabel('Total Points')
plt.annotate(f"Players: {num_non_outliers}\nTotal Points: {non_outlier_total}"
, xy=(0.9, 0.9), xycoords='axes fraction', fontsize=10, ha='right', va='top')

plt.figure(figsize=(10, 6))
sns.boxplot(x=outliers['total_points'])
plt.title(f"Box Plot of Outlier Total Points - Players: {num_outliers}")
plt.xlabel('Total Points')
plt.annotate(f"Players: {num_outliers}\nTotal Points: {outlier_total}",
xy=(0.9, 0.9), xycoords='axes fraction', fontsize=10, ha='right', va='top')

plt.show()


# Correlation analysis
df_eda = df_grouped_overall.copy()
correlation_with_points = df_eda.corr()['total_points']
top_positive_features = correlation_with_points.sort_values(ascending=False)[1:6]
top_negative_features = correlation_with_points.sort_values(ascending=True)[:5]
combined_features = top_positive_features.index.union(top_negative_features.index)
selected_features = combined_features.insert(0, 'total_points')
combined_correlation_matrix = df_eda[selected_features].corr()

# Print graphs
plt.figure(figsize=(12, 10))
sns.heatmap(combined_correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm")
plt.title('Correlation Matrix with Top 5 Positive and
```

```
Negative Correlated Features with Total Points')
plt.show()



# Analyze relationship between total points and other features
features = ['bps', 'goals_conceded']
for feature in features:

    if feature in df_eda.columns:
        plt.figure(figsize=(10, 6))
        sns.regplot(x=feature, y='total_points', data=df_eda,
        line_kws={"color": "red"})
        plt.title(f'Relationship between {feature} and Total Points')
        plt.xlabel(feature)
        plt.ylabel('Total Points')
        x = df_eda[feature].dropna()
        y = df_eda['total_points'][x.index]
        r_squared = r2_score(y, np.poly1d(np.polyfit(x, y, 1))(x))
        plt.text(0.05, 0.95, f'R² = {r_squared:.2f}', transform=plt.gca().transAxes,
        fontsize=12, verticalalignment='top')
        plt.show()

# Train multivariable linear regression model
correlation_with_points = df_eda.corr(numeric_only=True)['total_points']
.sort_values(ascending=False)
top_features = correlation_with_points.index[1:6].tolist()
X = df_eda[top_features]
y = df_eda['total_points']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("MSE:", mse)
print("R-squared:", r2)


coefficients_table = pd.DataFrame({'Feature': top_features, 'Coefficient': model.coef_})
print(coefficients_table)
```

```
# Print graphs
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)


plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)


plt.xlabel('Actual Total Points')
plt.ylabel('Predicted Total Points')
plt.title('Actual vs Predicted Total Points')
plt.show()
```