Overall Capstone Project

1. Proposal with Problem Statement

Can we utilize historical NFL statistics to predict future performance? This is a useful question to answer for anyone playing Fantasy Football. There are hundreds of thousands of fantasy football players across the world with an $18.6 billion market size. With these predictions players can maximize their chances of winning against the competition. It can also be used for those creating spreads and projections on the gambling side, such as DraftKings and FanDuel.

2. Data Collection and Wrangling Summary

Description of the Dataset

The dataset can be described as a culmination of NFL offensive statistics from 2014 up to the 2018-2019 season. These stats include touchdowns (passing, rushing, receiving), yards (passing, rushing, receiving), attempts (passing, rushing, receiving), yards per attempt, completions, interceptions, red zone targets, to name a few. The dataset also includes Fantasy Football scoring data, such as Fantasy Points, Points Per Reception, DraftKings and FanDuel scoring, and rankings based off of fantasy points scored per season.

The variables in the data that need to be answered for this question include the individual stats associated with positional fantasy football players. There is an abundance of statistics that make up these variables. For example, in the broad scope of the running back position we must consider rushing yards, rushing touchdowns, receptions, receiving yards, and receiving touchdowns. We can then dive deeper to include attempts per game, yards per attempt, and even yards allowed by the defence the player is facing. However to keep this project from being too cumbersome we will stick with the broader scope, only focusing on the individual players statistics.

We also must set up a basis for which we score these statistics. For example, the most common scoring gives one fantasy point per 10 rushing or receiving yards gained by a player, and one point for every 25 passing yards gained. As well, rushing and receiving touchdowns are worth six fantasy points where a passing touchdown is worth four. In most formats players earn points per reception. For this project we will be using the one point per reception format. Players also lose points for turnover, a fumble or interception result in a loss of two points. Turnovers are much too difficult to predict however, so these will not be weighted as heavily.

How I Obtained the Dataset

My dataset comes from several different sources in order to gain all of the statistical information needed. One set of data I downloaded as a CSV file, cleaned the data in Excel by removing unnecessary columns and rows before uploading it to my project notebook. This dataset came from Pro-Football-Reference.com.

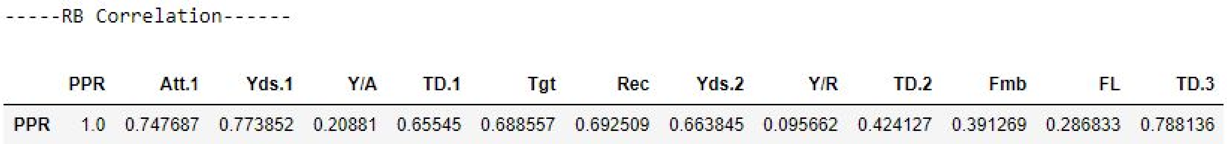
3. Exploratory Data Analysis Summary (Visualization and Inferential Statistics)

Visuals and Statistics to Support Findings

I created plots displaying the correlations of the positional stats that relate to the top performers for each position. For example, rushing yards and attempts have a high correlation to the top performing running backs. I have found several trends within the data. For example, there is a strong trend for running backs who have a high number of rushing attempts (Att.1) to produce higher fantasy points. The running back position resulted in the lowest correlation metrics compared to the other positions. Quarterbacks, wide receivers, and tight ends all displayed higher correlation metrics. This tells us that comparatively it is easier to predict the performance of those positions as a whole. Below are the results of the running back and wide receiver correlation metrics.

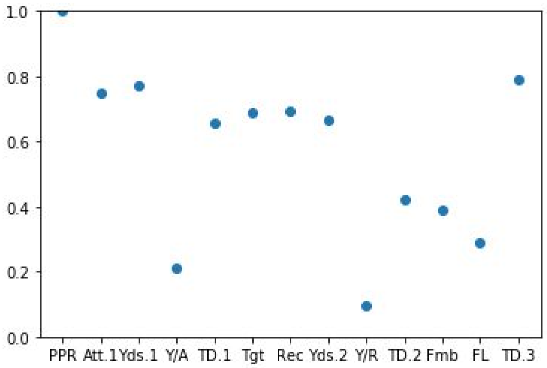
# Running Back Correlation

```
rb_corr = rb[['PPR','Att.1','Yds.1','Y/A','TD.1', 'Tgt','Rec','Yds.2','Y/R','TD.2','Fmb',
          'FL','TD.3']].corr(method='spearman',min_periods=365)
print('-----RB Correlation------')
display(rb_corr[:1])
plt.plot('PPR', data=rb_corr, linestyle='none', marker='o')
plt.ylim(0,1)
plt.show()
```



```
-----RB Correlation------
```

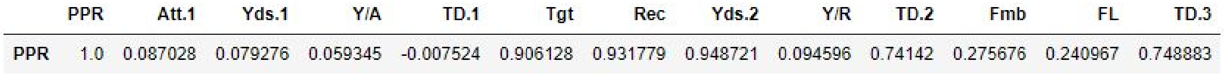
	PPR	Att.1	Yds.1	Y/A	TD.1	Tgt	Rec	Yds.2	Y/R	TD.2	Fmb	FL	TD.3
PPR	1.0	0.747687	0.773852	0.20881	0.65545	0.688557	0.692509	0.663845	0.095662	0.424127	0.391269	0.286833	0.788136



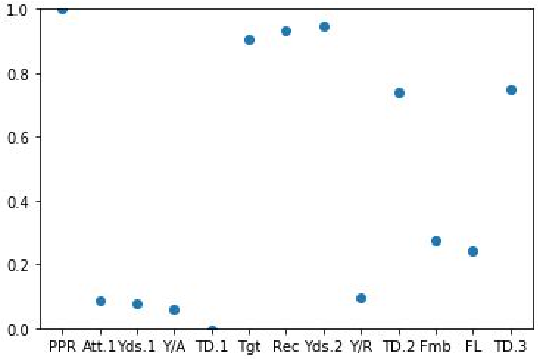
```
# Wide Receiver Correlation

wr_corr = wr[['PPR','Att.1','Yds.1','Y/A','TD.1','Tgt','Rec','Yds.2','Y/R','TD.2',
         'Fmb','FL','TD.3']].corr(method='spearman', min_periods=90)
print('-----WR Correlation------')
display(wr_corr[:1])
plt.plot('PPR', data=wr_corr, linestyle='none', marker='o')
plt.ylim(0,1)
plt.show()
```

```
-----WR Correlation------
```



	PPR	Att.1	Yds.1	Y/A	TD.1	Tgt	Rec	Yds.2	Y/R	TD.2	Fmb	FL	TD.3
PPR	1.0	0.087028	0.079276	0.059345	-0.007524	0.906128	0.931779	0.948721	0.094596	0.74142	0.275676	0.240967	0.748883

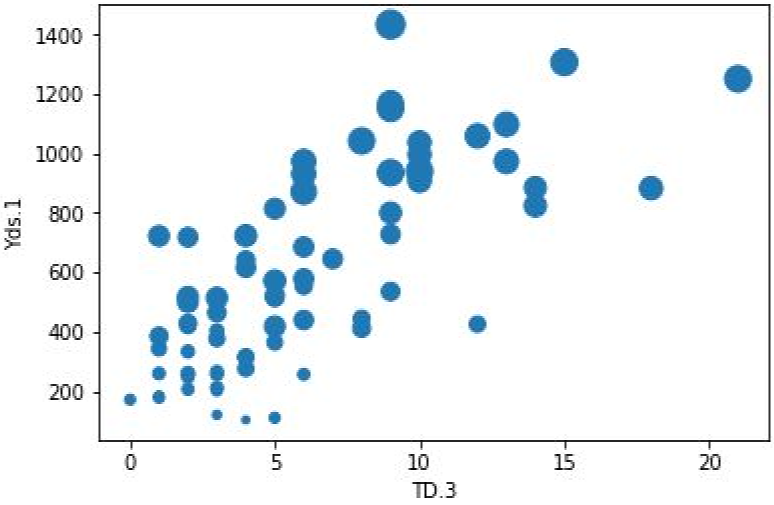


I then created scatterplots for the highest correlated variables to visually display how these stats relate to higher fantasy point production. As well, these scatter plots point out a consistent linear regression. The running back position proved to have the loosest linear fit compared to the other positions. This fluctuation can be explained when comparing the level of correlation we see in the running back position versus the other position. Seeing this tells a few things. One being that the running back position has more ways of producing fantasy points than the other positions. They are able to produce fantasy points through rushing yards, rushing touchdowns, receiving yards, receiving touchdowns, and receptions. The other being that the position is top heavy. Meaning that best players at the position are able to distance themselves from the rest of the pack. A running back that is proficient in both rushing and receiving has an advantage over those who only specialize in one of those areas. In the scope of this project, this is helpful for the end user in that it highlights the importance of having a top running back since it is much more difficult to predict a productive running back outside the top options.

Below are scatter plots displaying the highest correlators for running backs and wide receivers. From this visual display it is clear how it is easier to predict a productive middling wide receiver versus a running back.

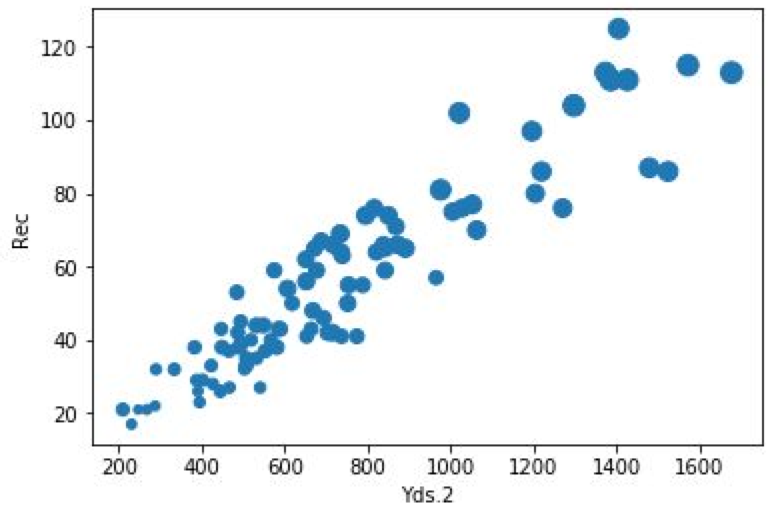
```
# Running Back Correlation Scatter Plot

        def plotyear(year):
                data = rb[rb.Year == year]
                area = data['Att.1'] * 0.67
                data.plot.scatter('TD.3', 'Yds.1', s=area)
        plotyear('2018')
```



```
# Wide Receiver Correlation Scatter Plot

        def plotyear(year):
                data = wr[wr.Year == year]
                area = data.Tgt * 0.67
                data.plot.scatter('Yds.2', 'Rec', s=area)
        plotyear('2018')
```



4. Results and In-Depth Analysis Using Machine Learning

Machine Learning: Regression Model Results

My first step was to decide what type of machine learning model to apply to my dataset. Considering that the purpose of my project is to predict fantasy points scored for players and doesn't incorporate classifying players into different categories, the most appropriate machine learning technique to apply to my dataset is a regression model.

Having decided on regression analysis being the best fit for my project I then needed to determine the most appropriate type of regression analysis to apply a; Linear, Random Forest, Ridge, or Lasso regression model.

While implementing the different models I adjusted the hyper parameters for all the models and would then run the ".score" method to test scores of both the training and testing datasets, as well as taking the Root Mean Squared Error, to see if the change I made resulted in a higher or lower score.

The Linear regression model had the fewest hyper parameters and gave us the best score using defaults. I was able to improve the score by adjusting the "train_test_split()" parameters. I found that a test_size=0.275 and a random_state=411 resulted in the highest score. I used this same "train_test_split" for all the regression models. The Linear

Regression models scores on the test dataset were (0.99886406) with a Root Mean Squared Error of (2.66563).

When tuning the parameters for the Random Forest I found that adjusting the "n_estimators" parameter to equal 2000 resulted in the highest score on the test dataset (0.9172012) with a Root Mean Squared Error of (22.75801). The Random Forest model does not have coefficients so we cannot visualize or test those parameters.

The Lasso and Ridge Regression models had more hyper parameters to tune. I chose to focus my adjustments on the alpha parameter; testing models with low alphas versus models with high alphas to see how they affect the coefficients. I found that the lower value for alpha gave us the highest score and best Root Mean Squared Error.

For the Ridge Regression models, the alpha equal to 0.01 resulted in a much better score on the test dataset (0.99886415) than setting alpha equal to 100 (0.99797109). The Root Mean Squared Error were (2.66553) and (4.17908) respectively. We received the same number of features for both values of alphas, 18.

The lower alpha value for the Lasso Regression model also resulted in the best score; alpha equal to 0.0001 (0.9988639) versus alpha equal to 0.01 (0.99884482). The Root Mean Squared Error were (2.66582) and (2.68812) respectively. However, with Lasso Regression, we saw an increase in the number of features used when decreasing the values of alpha, 18 features for the lower alpha value and 17 features for the higher alpha value.

In conclusion, after testing all the types of regression models and fine tuning their hyper parameters, I found that the Linear Regression model performed best, followed very closely by the Ridge and Lasso Regression models with the lower alpha values.

Their scores rank as follows:

Linear Regression (0.9988641)
Ridge Regression (0.9988641)
Lasso Regression (0.9988639)

Theses two models only differed in scoring by an extremely small percentage point.

With the gap being so slim between the Linear and Ridge and Lasso models I chose to display my prediction results for all three of the best performing models. This code is specifically modeled, tuned, and tested to project the fantasy points scored for the Quarterback position.

Ridge Regression Model code.

```
# Creating Ridge Regression Model - Low Alpha
        rr = Ridge(alpha=0.01)
        rr.fit(X_train, y_train)
```

```
# Scoring the Ridge Regression Model
        Ridge_train_score = rr.score(X_train,y_train)
        Ridge_test_score = rr.score(X_test, y_test)
        coeff_used01 = np.sum(rr.coef_!=0)
        print("Ridge Regression Train Score Low Alpha:", Ridge_train_score.round(7))
        print("Ridge Regression Test Score Low Alpha:", Ridge_test_score.round(7))
        print("Number of Features Used for Alpha=100:", coeff_used01)
```

```
# Output
        Ridge Regression Train Score Low Alpha: 0.998476
        Ridge Regression Test Score Low Alpha: 0.9988641
        Number of Features Used for Alpha=100: 18
        Root Mean Squared Error: 2.66553
```

```
# Plot for Ridge Regression Coefficient - Low Alpha
        plt.plot(rr.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,
                        color='red',label=r'Ridge; $\alpha = 0.01$',zorder=7)
```

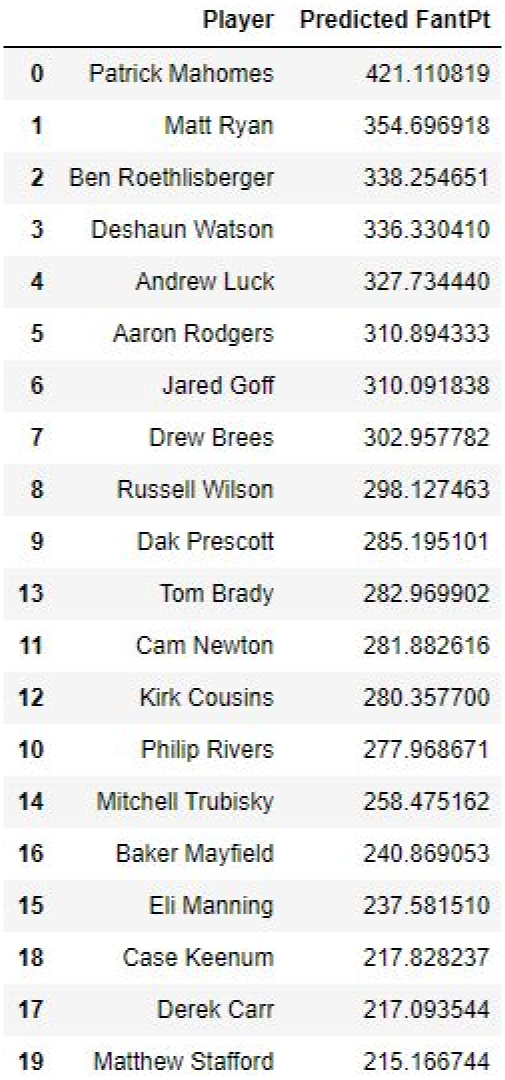
```
# Plot for Ridge Regression Coefficient - High Alpha
        plt.plot(rr100.coef_,alpha=0.5,linestyle='none',marker='d',markersize=6,
                        color='blue',label=r'Ridge; $\alpha = 100$')
```

```
# Plot for Linear Regression Coefficient
```

```
    plt.plot(lr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,
            color='green',label='Linear Regression')
    plt.xlabel('Coefficient Index',fontsize=16)
    plt.ylabel('Coefficient Magnitude',fontsize=16)
    plt.legend(bbox_to_anchor=(1, 0.5))
    plt.show()

# Predicting QBs with Ridge Regression
    top20_qbs = X_qb[:20]
    y_top20_qbs_rr = rr.predict(top20_qbs)

# Creating DataFrame to Visualize Prediction Results
    y_top20_qbs_rr = pd.DataFrame(y_top20_qbs_rr)
    qb_names = qb['Player'][:20]
    frames = [qb_names, y_top20_qbs_rr]
    qb_names_pred = pd.concat(frames, axis=1)
    qb_names_pred.columns = ['Player', 'Predicted FantPt']
    qb_names_pred.sort_values(['Predicted FantPt'], ascending=False)
```



	Player	Predicted FantPt
0	Patrick Mahomes	421.110819
1	Matt Ryan	354.696918
2	Ben Roethlisberger	338.254651
3	Deshaun Watson	336.330410
4	Andrew Luck	327.734440
5	Aaron Rodgers	310.894333
6	Jared Goff	310.091838
7	Drew Brees	302.957782
8	Russell Wilson	298.127463
9	Dak Prescott	285.195101
13	Tom Brady	282.969902
11	Cam Newton	281.882616
12	Kirk Cousins	280.357700
10	Philip Rivers	277.968671
14	Mitchell Trubisky	258.475162
16	Baker Mayfield	240.869053
15	Eli Manning	237.581510
18	Case Keenum	217.828237
17	Derek Carr	217.093544
19	Matthew Stafford	215.166744

Machine Learning: In-Depth Analysis

Hypothesis Development

My original hypothesis; The higher a players usage rate is (determined by attempts, percentage of snaps, percentage of team targets, or percentage of team attempts for the associated position) results in higher fantasy point production.

Working through this project I hypothesized that touchdowns and yards (passing, rushing, or receiving) would be the best predictor for future performance. While doing my correlation analysis my results supported my hypothesis. The two highest correlated statistics to fantasy points scored where touchdowns and yards, respectively.Touchdowns with a 0.928808 correlation and yards with a 0.900263 correlation. The next highest correlator was completions with a 0.834737 correlation. These data apply to the Quarterback position.

The data supports this throughout the other positions (Running Back, Wide Reciever, and Tight Ends). However, the correlation is much lower. This is due to their ability and frequency to score points in a variety of ways. For example, the highest correlator for the Running Back position is TD.3, or total touchdowns. This stems from the running back having a higher frequency in rushing and receiving touchdowns, as well as scoring points for receptions. So the highest correlator for running backs is at 0.788136 with rushing yards not far behind with a correlation of 0.747687.

Where my hypothesis failed to find support was in the Wide Receiver and Tight End positions. Both positions highest correlator was the Yds.2 stat, or receiving yards. The next highest correlators were receptions and targets, respectively. An in depth analysis of these results proved that receiving yards and receptions makes more sense to be the best predictor of future fantasy performance. First, scoring touchdowns is much more volatile for the receiving positions compared to the Quarterback and Running Back positions. Second, with this being true, the more reliable source of points for the receiving positions comes from their receiving yards and receptions. Considering that the data is based on the points per reception scoring system, the receptions are much more valuable for the receiving positions than touchdowns.

Conclusion of Machine Learning Analysis

After an in-depth analysis of the machine learning techniques I was able to find the most appropriate, best fitting regression models for each specific position, as well as further develop my hypothesis. Having analyzed the results from the process of creating machine learning models I found that my original hypothesis held true for the Quarterbacks and Running Backs position, however, was not the case for the Wide Receiver and Tight Ends position. Even though this goes against my original hypothesis it makes logical sense considering the scope of the scoring system and the high variability in receiving touchdowns for a player from year to year. It would be appropriate that the highest correlators to scoring for a receiving position comes in the form of receiving yards and receptions.

Recommendations and Next Steps

Having completed my machine learning regression analysis with the dataset I have chosen to base my project on, there are a few more advanced statistics I wish I would've been able to incorporate. The reason I was unable to incorporate these more advanced analytics into my project stem from issues associated with wrangling the data from different sources then integrating the data with my dataset. For example, most advanced analytics require some sort of payment/subscription in order to view or download. Then we face the issue of cleaning these data in order to apply, let's say Todd Gurley's statistics for rushing yards after first contact, to the players current set of statistics in our dataset. These advanced analytics are very helpful in determining future player performance. So, my next steps in creating a more accurate prediction model would be to incorporate these advanced analytics.

Another issue I came across when using this dataset was that it didn't take into account players changing teams and how a player moving to a new team affects their future performance as well the players on the team they moved to. For example, Odell Beckham Jr. a top wide receiver moved to the Cleveland Browns and my dataset did not reflect this move. His change in team not only affects his stats since he has a new quarterback throwing him passes, but also affects the stats of his new quarterback, Baker Mayfield. This then begets another question, which is how his new team will use the player or that position in their gameplan. Meaning that, is he moving to a team that have a higher or lower tendency to pass the ball. The advanced analytics would give us insight into this by showing us a team's offensive run versus pass play percentage. However, we run into more trouble here since this is a product of the team's coaching staff which also change year to year. So we would need the analytics on the tendencies of specific coaches, then apply those to the team they currently coach, and pass these data along to the players. For the most part these tendencies will persist, but they can also be a factor of the teams makeup and might change to better suit what players the coach has at their disposal.

With all of this in mind, I chose to keep a smaller scope for my dataset. Only incorporating easy to acquire statistics and not to use team/coach tendencies. Knowing how much the team/coach that a player plays under can affect their performance, and the predictive power advanced analytics provides, these are the statistics I most want to incorporate into my machine learning models in the future.