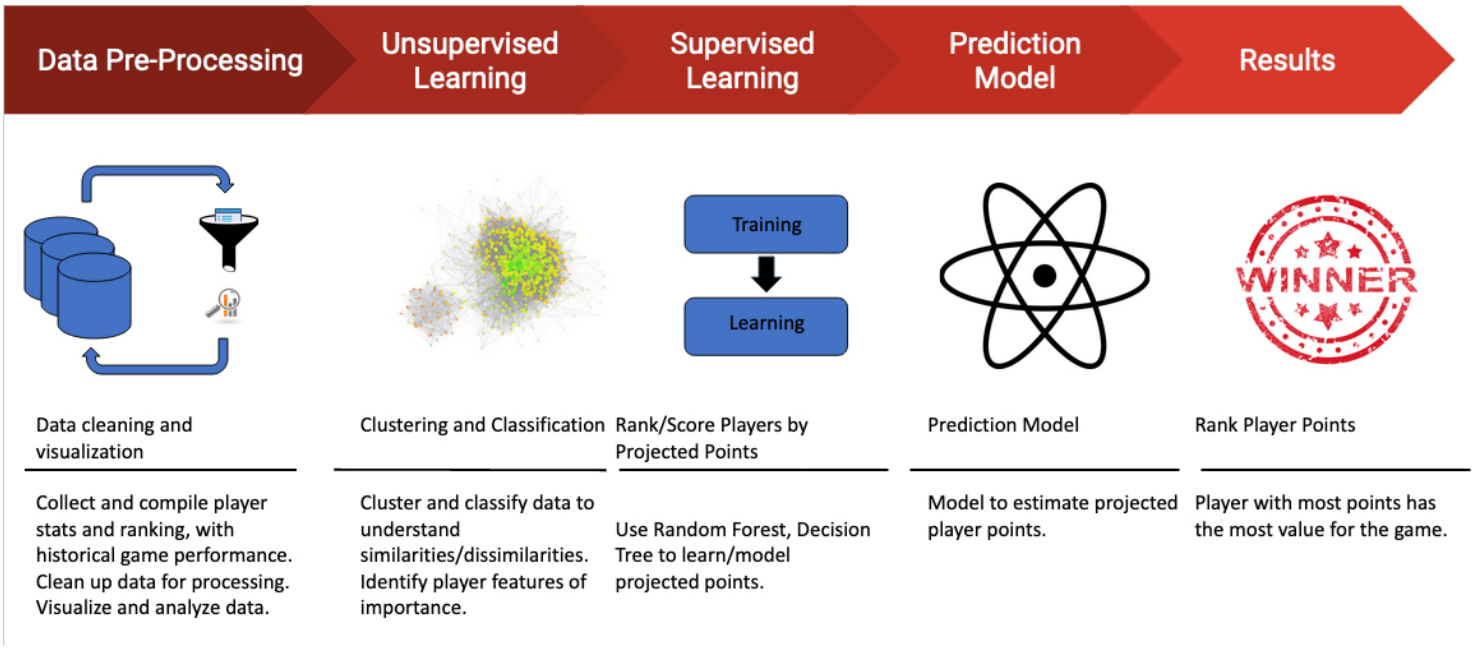
README.md

Fantasy Football Recommendations Using Machine Learning



GitHub README - https://github.com/CS4641-2021/sports-numbers-fans#readme

Presentation recording (BlueJeans)

Introduction

The goal of this project is to accurately predict average fantasy football performance for NFL wide receivers throughout a single season. Performance is calculated via points that can be earned by a player for accumulating yards gained, touchdowns scored, or passes caught. Similarly, points can be lost by a player for losing yards or fumbling the ball. At the end of every game, a player's total points are tallied, and the player with the highest number of points has the most value for that game.

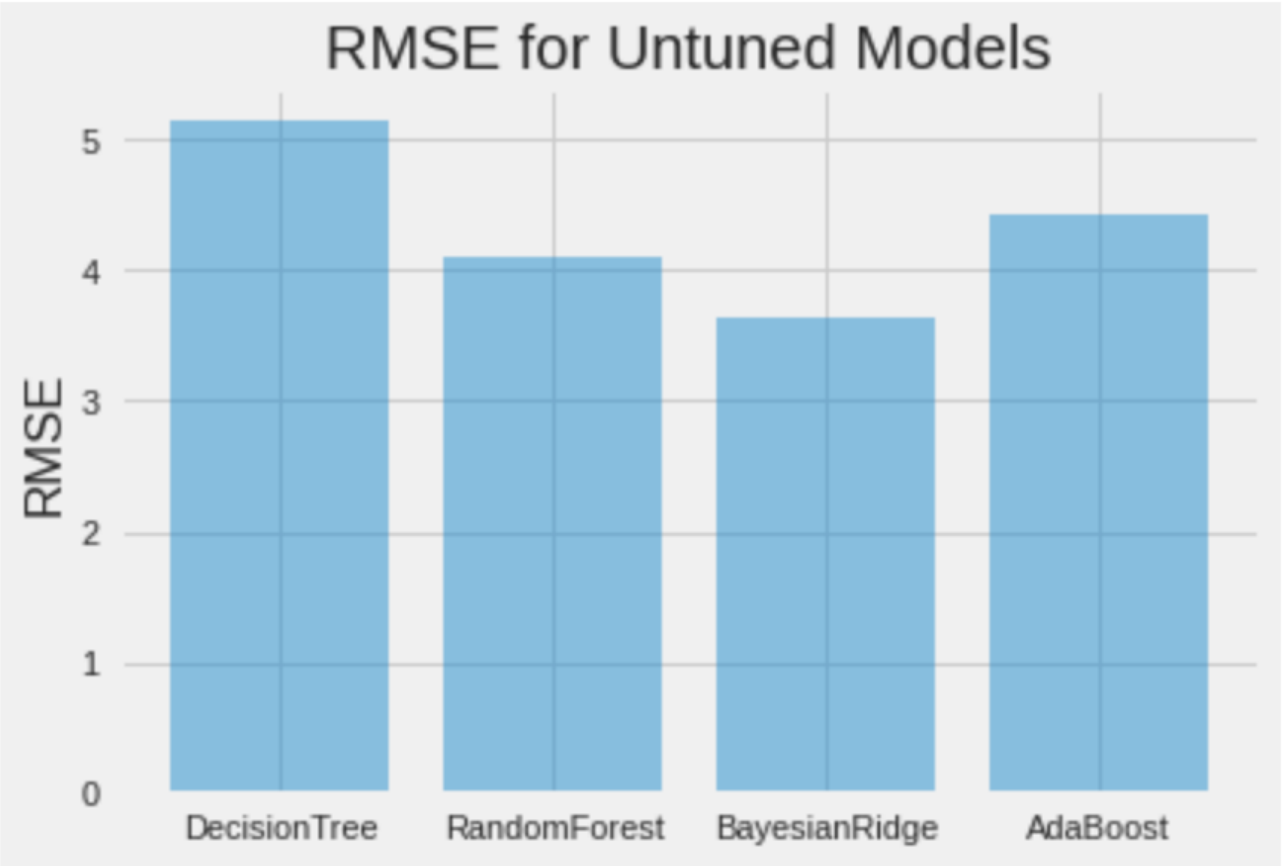
Players of *fantasy* football, users, will select a roster of players every week in an attempt to accumulate the highest sum of points via their roster. Users will use a variety of information to make an informed decision on their roster selection. Typical facets of information include previous player performance, player matchups, scouting reports, and projected points. From personal experience, most players tend to lean towards projected points as the main indicator for their analysis.

Methods

We aggregated data from fantasydata.com for the 2017-2020 season. Our data consists of approximately 35 features that reference a receivers performance in a given year. Since we are trying to predict 2020 results given 2019 data, we have three complete seasons of data to work with approximately 75 data points per season.

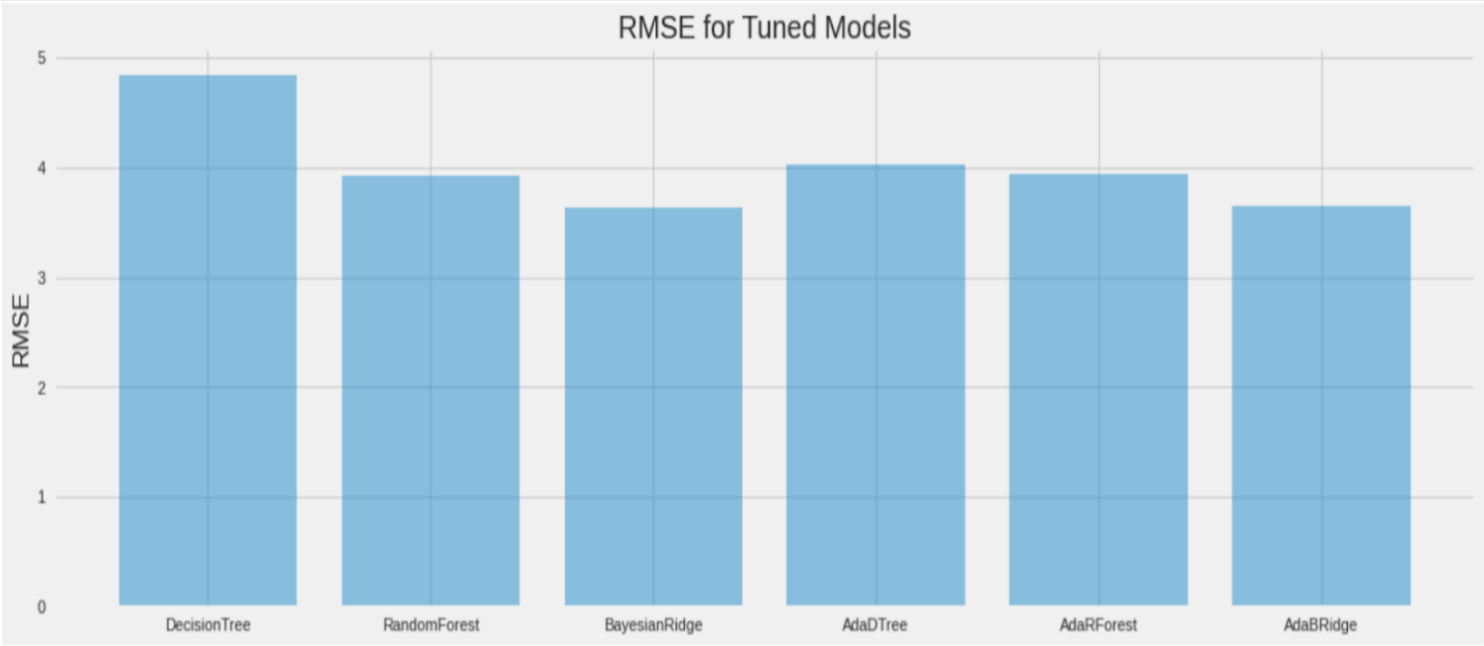
We visualized and clustered our data using unsupervised machine learning techniques in order to understand our data. We created a correlational heat map to understand relationships in features. We used K-means clustering in order to identify any potential groupings in our data points. Additionally, we used parallel coordinates to visualize a normalized version of our data, which was grouped into bins according to the target variable. All of these methods together give us an understanding of where potential sources of error can occur for a regression model.

Below, we are testing various regression models on our data. These models are being used with out-of-the-box parameters and are meant to serve as a basis for future improvement. We find that Bayesian Ridge looks to be the promising regression model from the start.




```
RMSE - Decision Tree:     5.15
RMSE - Random Forest:     4.09
RMSE - Bayesian Ridge:    3.64
RMSE - AdaBoost:          4.43
```

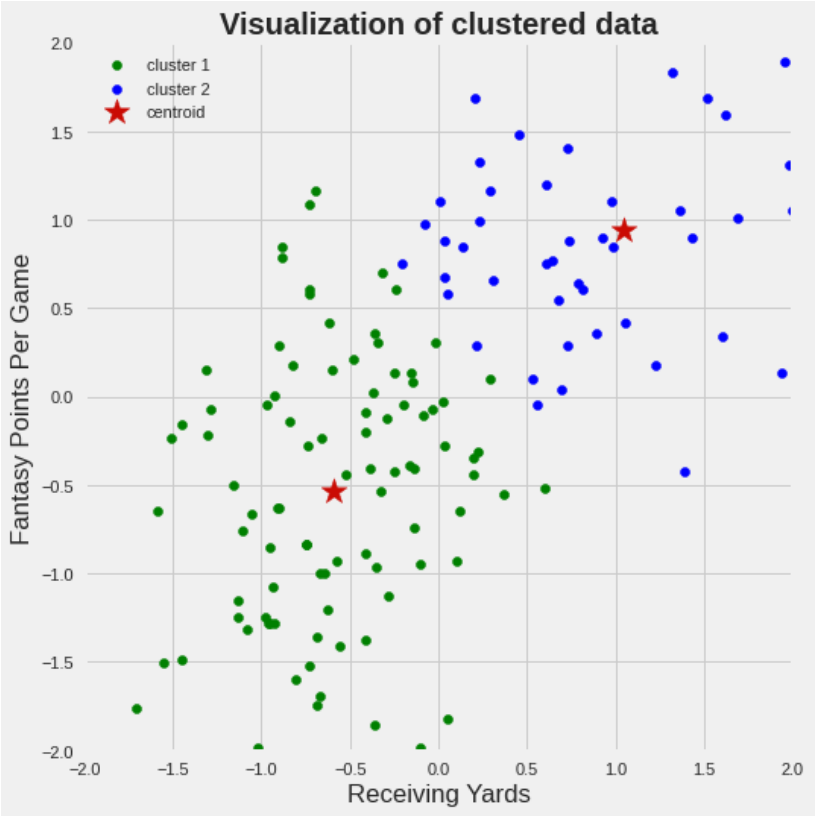
We decided to use one model going forward in order to cut back on training time and keep the project within a manageable scope. In order to do this, we tuned the hyperparameters of these models using grid search and applied AdaBoost with each optimal regressor. After tuning every model, we end up with Ridge Regression still at the number one performer with an identical error to its untuned counterpart. Moving forward, we decided to stick with the standard Ridge Regression model

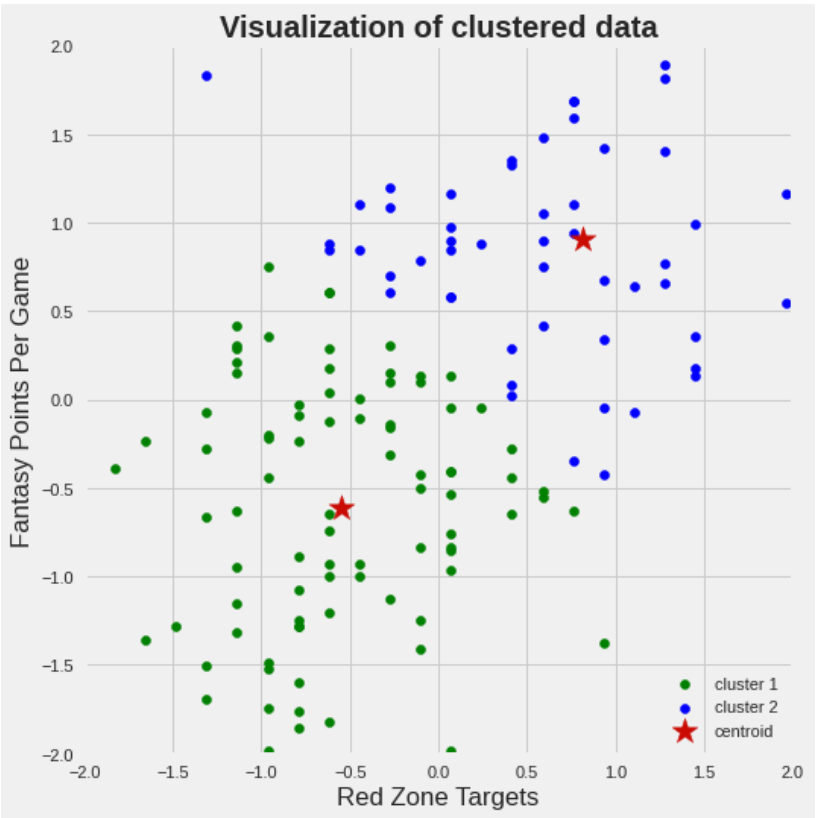



```
RMSE - Tuned Decision Tree:          4.85   Reduction in RMSE from tuning:   0.43
RMSE - Tuned Random Forest:          3.92   Reduction in RMSE from tuning:   0.19
RMSE - Tuned Bayesian Ridge:         3.64   Reduction in RMSE from tuning:   -0.00
RMSE - Tuned AdaBoost / DTree:       4.03   Reduction in RMSE from tuning:   1.24
RMSE - Tuned AdaBoost / RForest:     3.94   Reduction in RMSE from tuning:   0.17
RMSE - Tuned AdaBoost / BRidge:      3.65   Reduction in RMSE from tuning:   -0.01
```

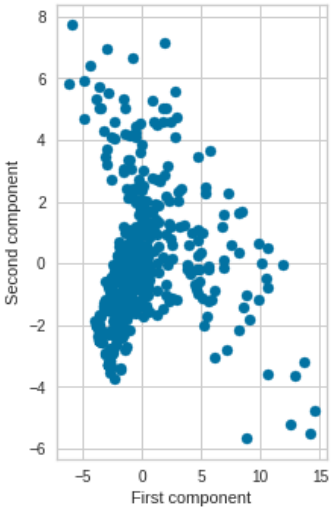
Given the limited set of data points in our model, we decided to use K-Nearest neighbors with 25 nearest neighbors to impute missing values instead of dropping the data point altogether. This allowed us to increase our number of data points from 146 to 462 total data points from the 2017-2018 season. We focused on feature engineering to improve the accuracy of our model, and used both unsupervised and supervised learning techniques like forward feature selection, K-means clustering, and principal component analysis (PCA).

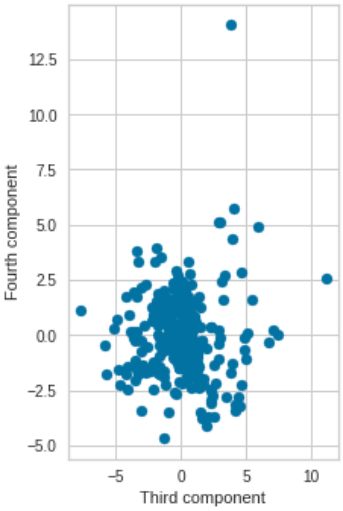
With K-means clustering on select features, we found that our optimal number of clusters based on rushing yards and red zone targets was 2 clusters, as shown below.

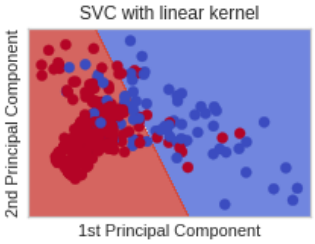


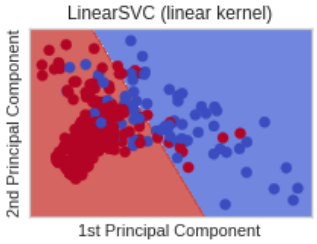


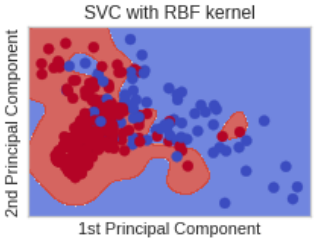
We applied PCA and selected 2 components for our Support Vector Machine (SVM) classifiers to categorize our data into two tiers - tier1 for players with greater than 10 fantasy points, and tier2 for the rest.

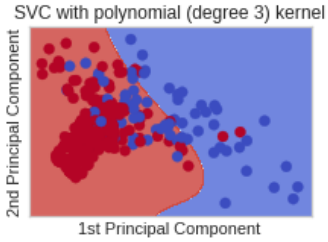












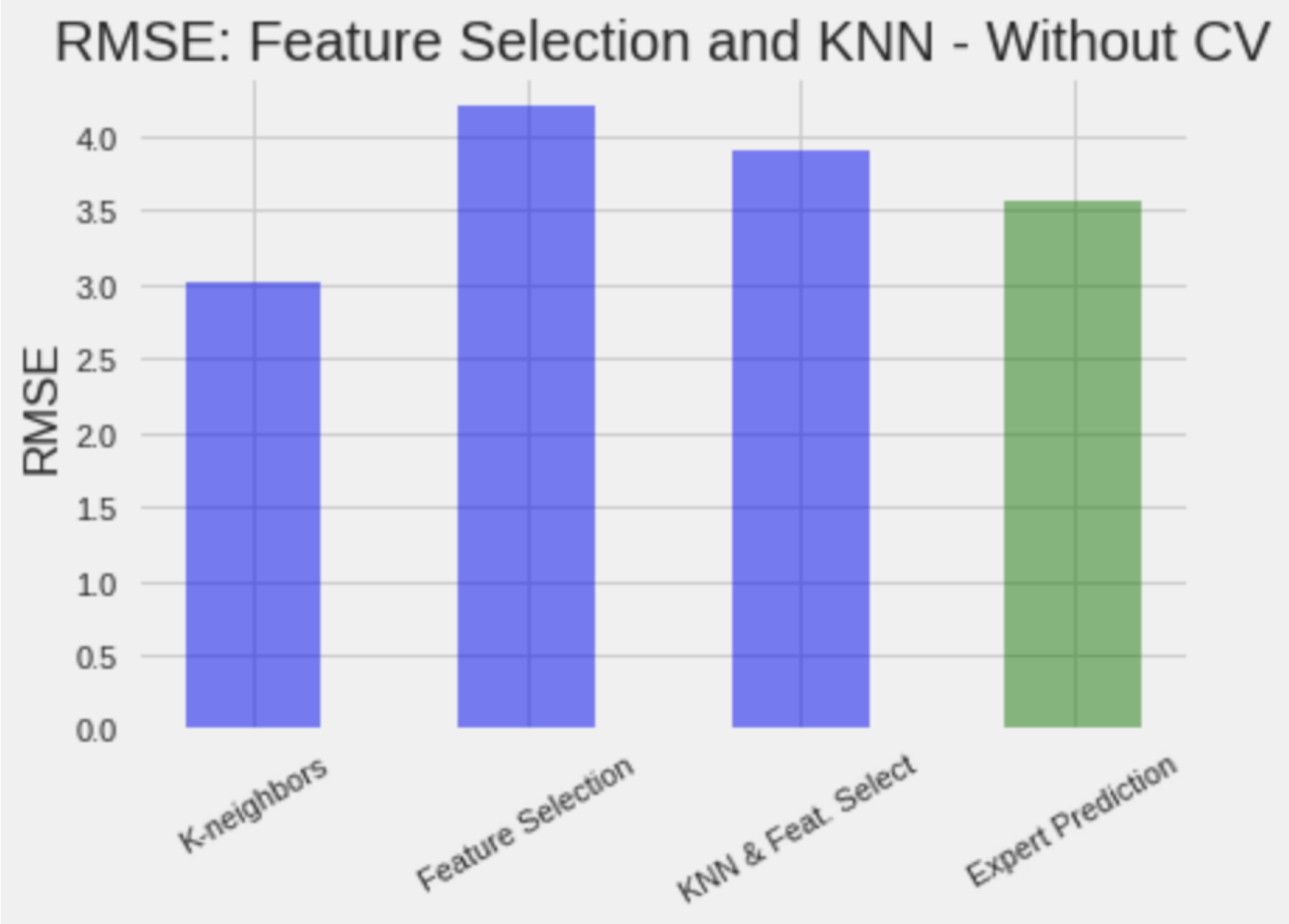
From our SVM runs, we found that the radial basis function (RBF) SVM model gave us the best fit for our 2D decision boundary.

Finally, in order to train our model with more data points, we used cross-validation to leverage all of the data rather than setting aside a portion for training and testing.

Results

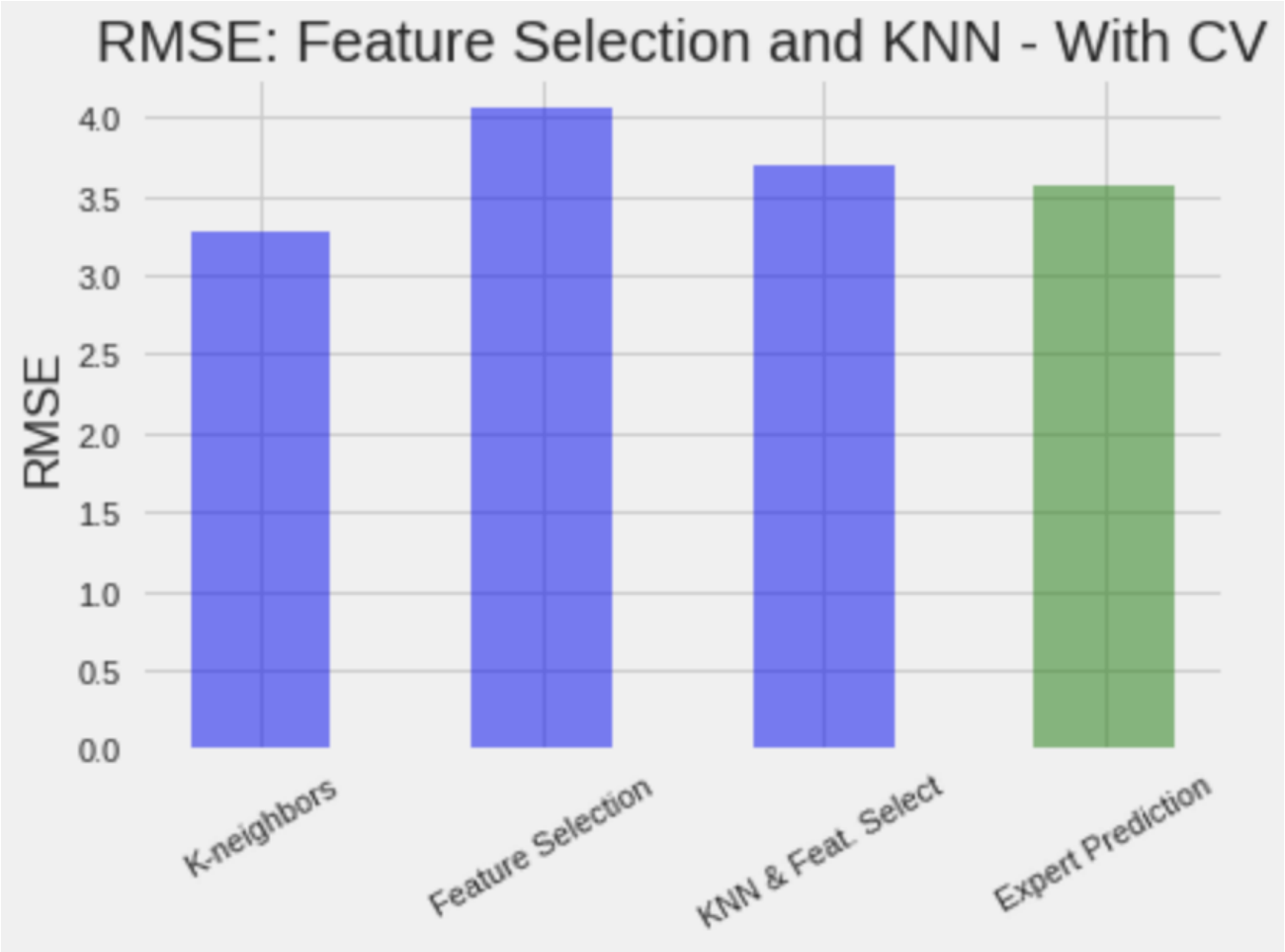
We were able to achieve an RMSE which was significantly less than the RMSE for fantasy football experts' predictions. Our best-performing model used Ridge Regression on a normalized set of inputs of real and artificial values generated by the K-nearest-neighbors algorithm. We saw an error reduction of ~17% just by applying the KNN algorithm to our dataset. This algorithm benefited our model significantly because it added roughly three times as much training data to our dataset. This benefit was amplified by the fact that our dataset was already very small to begin with.

The use of feature selection hindered our models overall performance when compared to our initial results. Reducing our feature space from forty features to four components cut available information for the regression model by ninety percent. Despite the high explained variance for our principal components, there did not seem to be enough truly informative data available for PCA to serve as a useful tool. This suggests that we need to spend more time gathering and processing our data.




```
RMSE - K-neighbors:              3.01
RMSE - Feature Selection:        4.20
RMSE - KNN & Feat. Select:       3.91
RMSE - Expert Prediction:        3.56
```

Unfortunately, we were unable to achieve any better results using cross-validation throughout the entirety of this project. This suggests the presence of bias in our model, which was expected to a certain extent due to the limited amount of complete, uncorrelated data.




```
RMSE - K-neighbors:                3.28
RMSE - Feature Selection:          4.05
RMSE - KNN & Feat. Select:         3.69
RMSE - Expert Prediction:          3.56
```

Discussion and Next Steps

Our models predicted the average number of points per game that wide receivers would score on a seasonal basis. We measured the performance of our models against expert predictions using root mean square error, or RMSE. After using various models and applying different machine learning methods to our dataset, we were able to achieve results that outperformed the expert predictions. These methods can be used in a Fantasy Football league to select wide receivers that will score the most points throughout a season.

There are many methods that can be used to benefit the performance of our models. We were unable to acquire more data, as every year of data requires a payment to fantasydata.com. The datasets from fantasydata.com were also missing some data points. We used the K-nearest-neighbors method to fill in any missing data, however, the real values for the missing data would further improve our models. Creating a program that would scrape through the web pages to acquire the desired information would both be cheaper and boost the performance of our models. Furthermore, the addition of a biasing component to our algorithms would fix the underfitting issues that our models exhibited.

Further fine-tuning of the models will lead to better results, thus more accurately predicting which wide receivers will perform the best. If these methods are extended to other positions in the NFL, or used on a game-by-game basis, a roster selected purely from machine learning predictions would be feasible. Thus, increasing potential earnings made from scoring the most points in a Fantasy Football league.

References

[1]Greenberg, Neil "How The Post's fantasy football projections work". The Washington Post https://www.washingtonpost.com/sports/2019/08/14/how-posts-fantasy-football-projections-work/

[2]Fantasy Data, https://fantasydata.com/nfl/fantasy-football-leaders?position=4&season=2020&seasontype=1&scope=2&subscope=1&startweek=1&endweek=1&aggregatescope=1&range=3

[3]Sports Data, https://sportsdata.io/developers/data-dictionary/nfl

[4]NFL Statistics, https://www.kaggle.com/kendallgillies/nflstatistics?select=Career_Stats_Receiving.csv

[5]Scikit-Learn User Guide, https://scikit-learn.org/stable/user_guide.html