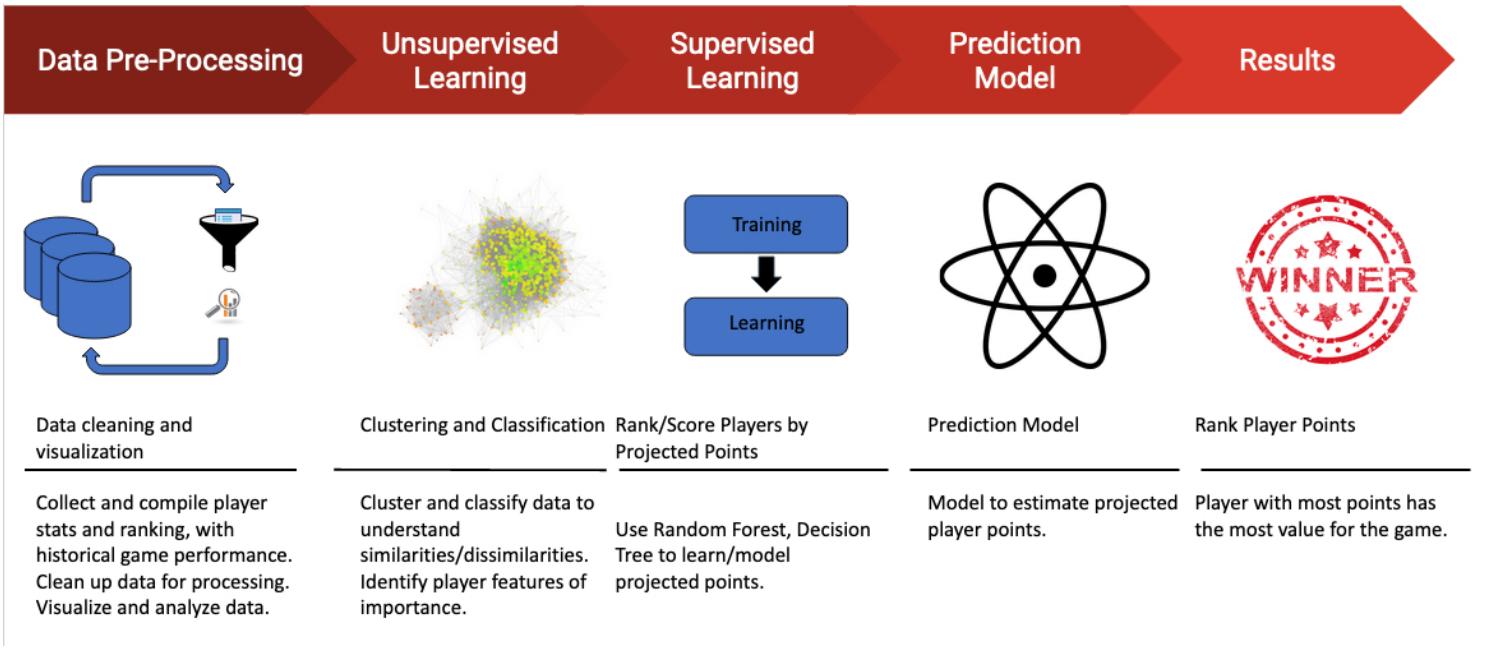
README.md

Fantasy Football Recommendations Using Machine Learning



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

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

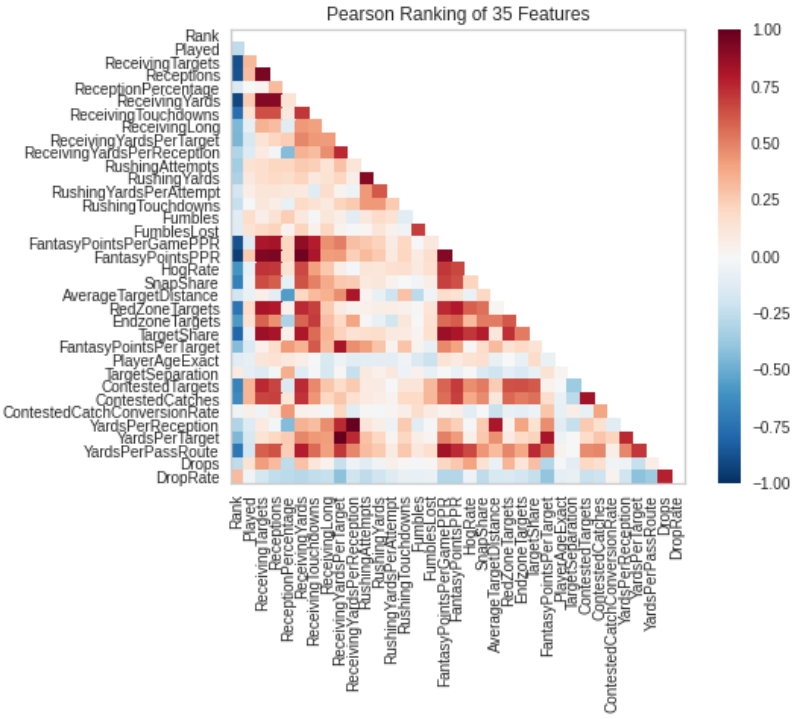
We aggregated data from fantasydata.com for the 2017-2020 season. Our data consists of approximately 35 features which reference a receivers performance in a given year. Since we are trying to predict 2020 results given 2019 data, we have three complete seasons of data to work with approximately 75 data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

We visualized and clustered our data using a couple unsupervised machine learning techniques in order to understand our data. We created a coorelational 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 gives us an understanding of where potential sources of error can occur for a regression model.

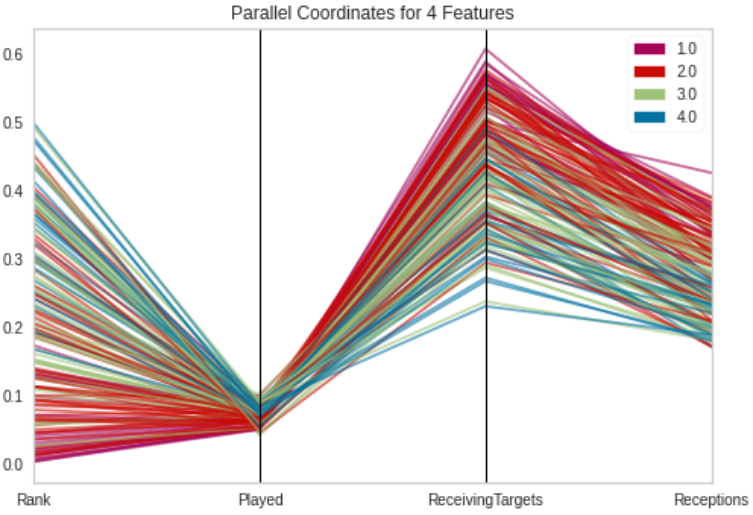
We tested multiple regression models, and given our relatively low number of data points and high features, we decided to use Bayesian ridge regression with a normalized feature set.

Results

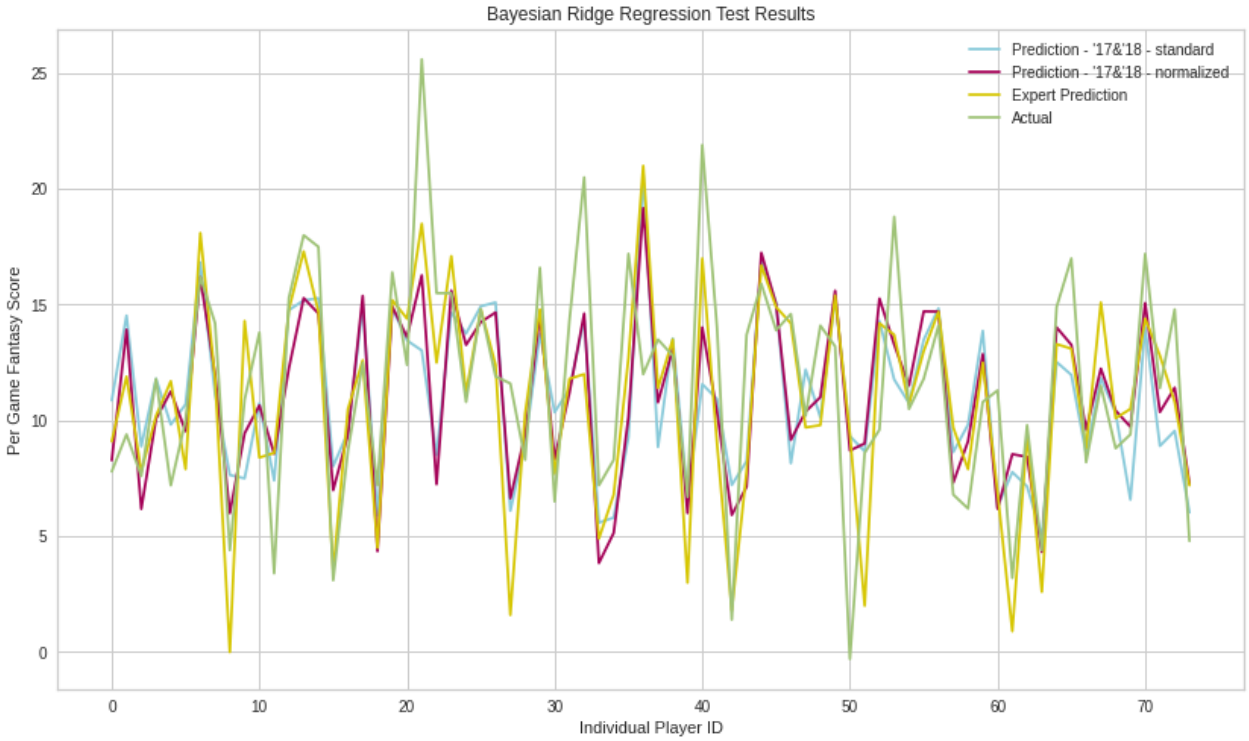
In our correlation matrix, the darker points represent stronger correlation. In particular, we are concerned about the correlation between rank and relevant features. Candidates for features that have a strong effect on the rank of the player are ReceivingTargets, Receptions, ReceivingYards, ReceivingTouchdowns, SnapShare, RedZone Targets, TargetShare, and YardsPerPassRoute.

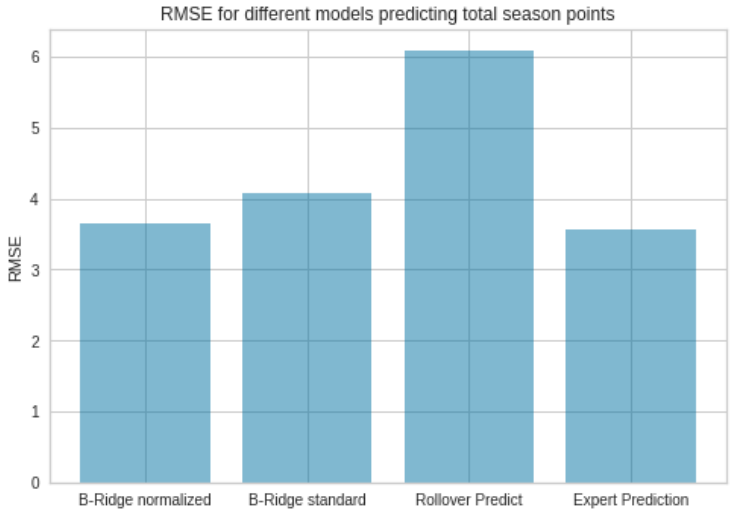


In our parallel coordinates matrix, we are able to determine which features performed normally or abnormally when considering the performance of the feature for the testing data. Features that perform abnormally are RushingYards and RedZoneTargets.

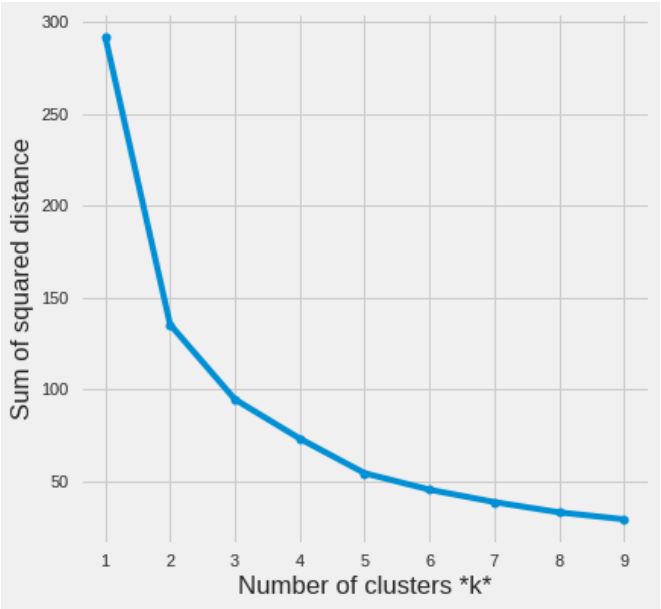


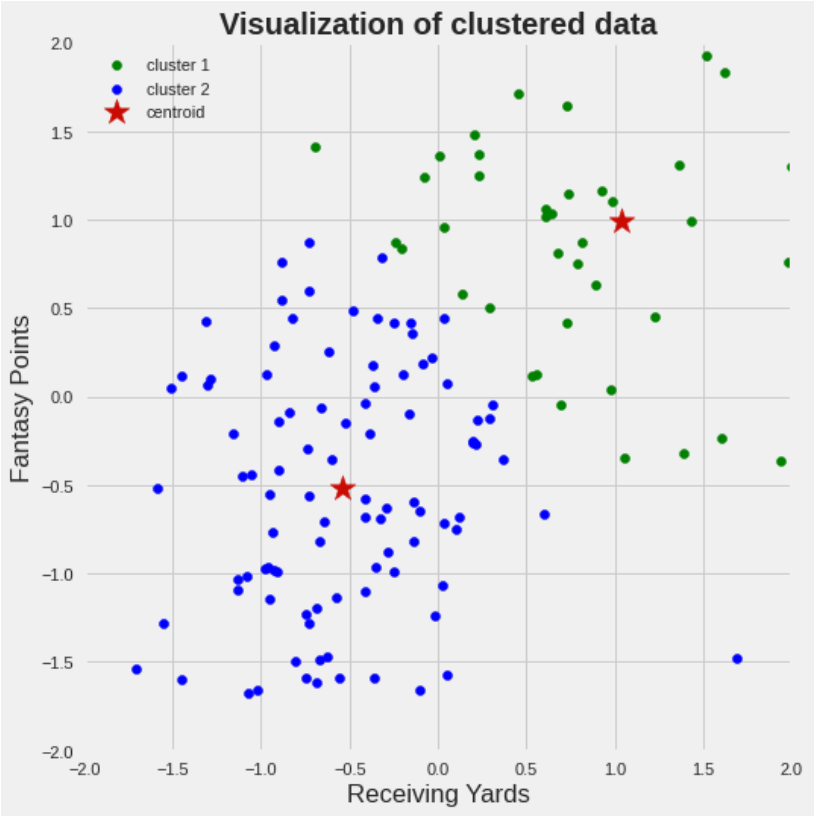
In the Bayesian Ridge plot as well as the RMSE chart, it is clear that the predictions for the normalized data performs better than the predictions for the standard data. In the RMSE chart, both sets of data outperform the rollover predictions. The predictions for the normalized data only performs slightly worse than the expert predictions, having an RMSE value of 3.6375 and 3.557 respectively.



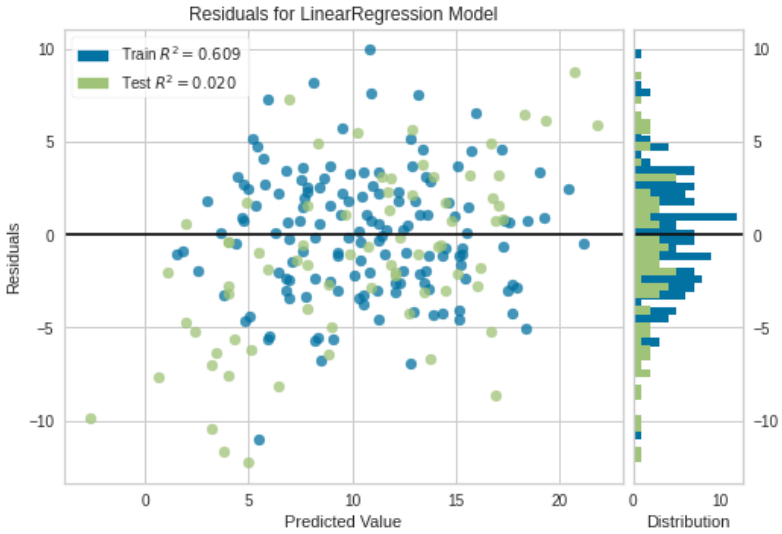


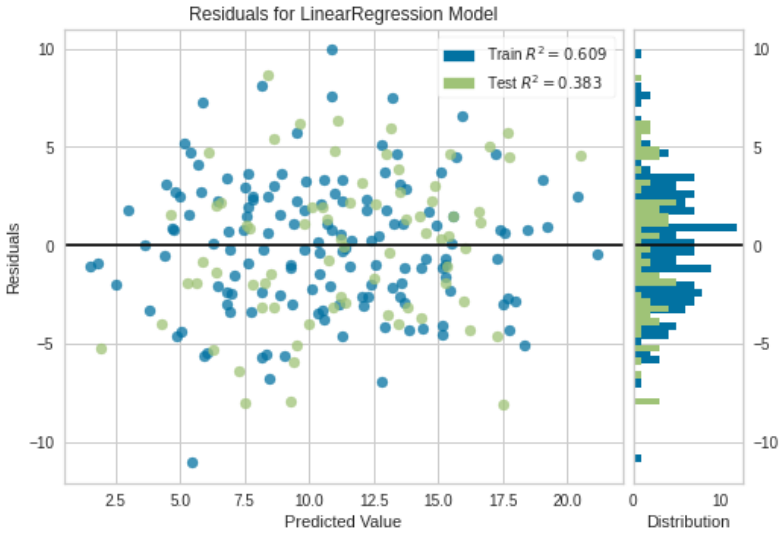
Using the elbow method for the K-Means algorithm, the optimal value for k was determined to be 2. The data appears to be clustered relative to performance. The green data points correspond with players who perform better and the blue data points correspond with players who perform worse.





Using the residuals from a linear regression model, we see that the values for training R^2 values and testing R^2 values for testing are different. Ideally, these values are closer together. The differences in these values may suggest that our model is overfitting our data.





Discussion and Next Steps

Our Bayesian Ridge Regression algorithm perform well, however our main goal is for an algorithm to perform better than the expert predictions. Due to potential overfitting for our model, we will use principal component analysis for determination of important features and implementation of dimension reduction. Additionally, we will use a random forest algorithm for our prediction and determine which algorithm performs the best using cross validation.

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