Capstone 1 Final Report

Problem Statement

The aim of this project is to take a mathematical approach to betting on NFL contests. We want to predict the total number of points scored between two teams playing against each other in any particular NFL game. We will be using regression techniques for numeric predictions, and will be comparing our predictions to the game totals that professional sports books put out for the public to view and bet on. Comparing our predictions to the betting market will paint a clear picture on which games present us with an opportunity to profit. Once we have identified a game or games with glaring values in comparison to the betting market, we can recommend these plays to a potential client, friend, or even play them ourselves. Additionally, we can recommend that fantasy football players use football players in their lineups from games that we project to go “over”, and recommend to stay away from players from games that we project to go “under” the total.

Attribute Selection

The attributes that will be used for this project will NFL team matchup statistics such as the offensive team’s passing matchup, running matchup, and overall skill matchup. Additionally, we will be using team grades from profootballfocus.com, venue data, home and away data, points scored, and a defense’s points allowed. Another important piece of information is data on game totals, which will tell us what values the sports books set the betting market for each game, and can serve as a marker from which to compare our predictions and make recommendations.

Obtaining the Data

For all of the attributes described, there are many sources from which to obtain the data. One source is pro-football-reference.com, which has a database that contains nearly every relevant football statistic for every NFL game played dating back to 1920. Another source for obtaining the needed data will be profootballfocus, which has defensive and offensive grades for every team in every contest dating back to the early 2000's.

Data Collection

- Built five web data scrapers using BeautifulSoup4 to collect all data for this project
  - Pro-football-reference.com referee and game data from 2006 onward
  - Profootballfocus.com game grades for every team dating back to 2006
  - Footballoutsiders.com DVOA metrics for every team and every game since 2006
- For missing values, I explored why they were missing and found a pattern that led me to changing the code in two of my scrapers.
  - Games that went into overtime had additional columns of data that I hadn't accounted for
- Put all data into .csv files and Pandas dataframes

Data Cleaning

- Used Pandas to import data from .csv files into dataframes using .read_csv()
- Performed dataframe column operations to create new variables
- Used string methods on dataframe columns to clean string variables
- Changed date columns from strings to datetime objects
- Replaced old team names with up-to-date team names for data consistency among all tables

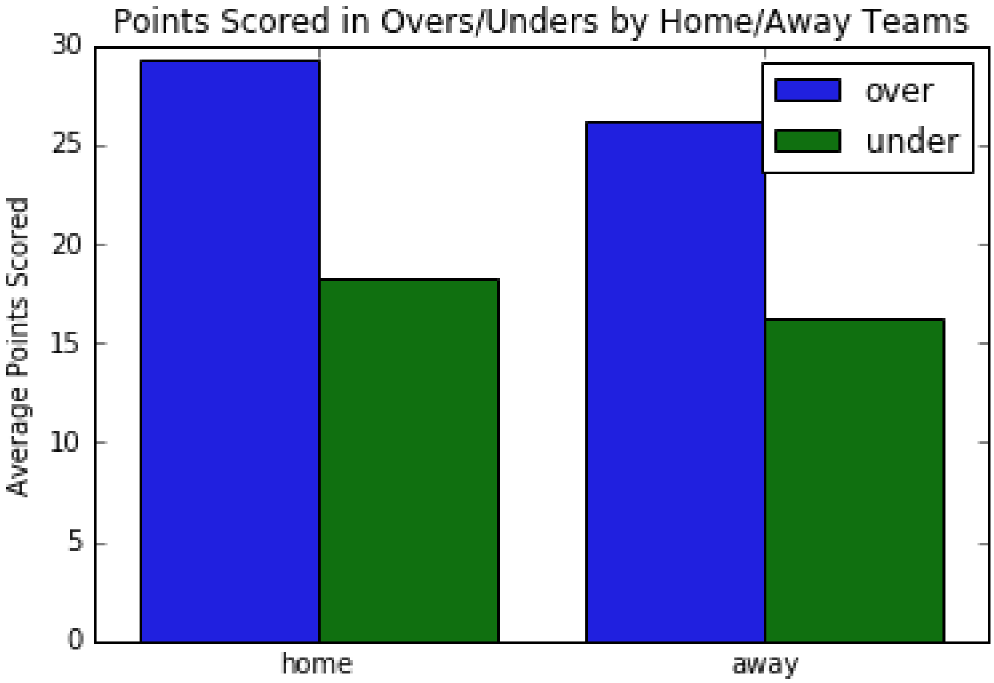
    - Used a dictionary that mapped all old team names to their current team name and applied the .map() function to the ‘team’ columns to transform the team column to a new, updated team column for every table
- Used joins and merges on tables of data to prepare data for analysis in the next part of this project

Data Storytelling

With the steps of data wrangling completed, it is now time to explore the data further. Our main goal is to identify trends and correlations, as well as to communicate insights found within or dataset. We want to ask questions of our data that will drive us toward our main goal of this project, which is to profit from predicting over-under outcomes of NFL games. Remember that sportsbooks set a total number of points to be scored for every NFL game, and bettors can either bet the “over” if they think the teams will combine for more points than the set total, or “under” otherwise. With this in mind, let’s proceed with exploring our data in such a way that we are pushed toward our goal of predicting NFL outcomes.

Firstly, we look at the outcomes of games within our dataset. We are using NFL games dating back to 2007 for this project. We can see within our jupyter notebook that 1,350 games went over the total set by the sportsbook, 1,341 went under, and 45 were a “push”, or scored as many points as projected. These numbers suggest that the sportsbooks are incredibly good at setting totals for games, as almost exactly 50% of games hit the over, and almost exactly 50% of the games hit the under.

A logical follow-up would be to explore what happened in games that went over/under. In games that went over the total, the home team averaged 29.3 points per game (ppg), and the away team averaged 26.2 ppg. In games that went under, the home team averaged 18.2 ppg, and the away team averaged 16.2. Therefore, in games that go over, the average number of total points scored is 55.5 ppg, and for unders it is 34.4 ppg. That is a massive difference of about 21 points, or the football equivalent of three touchdowns/extra points. Fantasy football players can see the immediate value of being able to accurately predict whether or not the game will go over or under, as they will want to choose players from games that are predicted to go over the total set by the sportsbook, since touchdowns are incredibly valuable in fantasy football.



Next, we take a look at games in which the total is set as a high score or low score. These games are interesting to investigate because in these games, the betting public usually has more of a consensus opinion on what will happen in the game. For example, if two teams with very bad offenses are playing against each other, the total for the game might be set at a number like 40, or even lower. This is considered a very low point total for a sports book to set. The public might lean toward betting the under on this game, since the two teams' offenses have been poor during the course of that season. We will use below or equal to 40 as our "low point" threshold, and above or equal to 48.5 as our “high point” threshold. Looking at games with totals below or equal to 40 points, we can see that the over hit 260 times, the under hit 207 times, and push hit 10 times. The over hits about 55.7% of the time in this case, which exceeds the 52.4% needed to beat the rake of the sportsbook. The average score for these games is 21.2 ppg for the home team, and 18.6 ppg for the away team, for a total of 39.8 points, on average. For the fantasy football player, this suggests that although we may predict one of these “low total” games to go “over”, it doesn’t mean that we should be targeting players from these games. Looking at the “high total” games with totals set at or above 48.5, we find that the under hit 254 times, the over hit 236 times, and push hit 7 times. The average score for these games is 27.4 ppg for the home team, and 24.1 ppg for the away team, for a total of 51.5 points, on average.

Lastly, we look into how one of our variables, ‘offMatchup’ is related to points scored for a team. offMatchup represents how good or bad a particular offense’s matchup is going into that game. The variable combines how good the team’s offense is with how bad that team’s opponent’s defense is. A high offMatchup score is good for a team’s offensive outlook for that game. Looking at the initial scatterplot, we can sense a positive correlation between offMatchup and points for (pf). Using the corrcoef function in Python’s numpy library, we can see that the correlation coefficient between the two (for home teams) is 0.236. Investigating further, we can see that home teams who have an offMatchup score of 10 or greater average 27 ppg, and away teams average 24.4 ppg. On the other hand, home teams who have an offMatchup score of -2 or lower average 19.9 ppg, and away teams average 17.5. Both sports bettors and fantasy players can benefit from using offMatchup scores when attempting to predict how an offense will perform.

One final piece of information to note is that when an away team is facing a team that has an average coverage score of 73 or higher, the under hits at a 63% rate. This would be an extremely profitable trend to follow as a sports bettor or daily fantasy player.

We can see that digging into the data is a worthwhile endeavor when it comes to placing placing bets, or attempting to determine how a game will play out. Someone who wishes to profit from playing fantasy sports or betting on games would be wise to take advantage of some of the information gathered from digging deeper into the data we collected and wrangled.

Inferential Statistics

The next step in our process of predicting over/unders for NFL games is to use inferential techniques to explore the data further. Using an iPython notebook, we break our data down into two tables: homeMatchup, and awayMatchup. The homeMatchup table contains instances of games for the home team. Each instance has variables that contain information about a particular home team's matchup for that game. One example includes the ptsMatchup variable, which is calculated by taking the home team's average points scored over the past seven games, and adding that number to the home team's opponent's average points allowed over the past seven games. There are ten independent variables for each table, and the dependent variable for each table is points scored (pf).

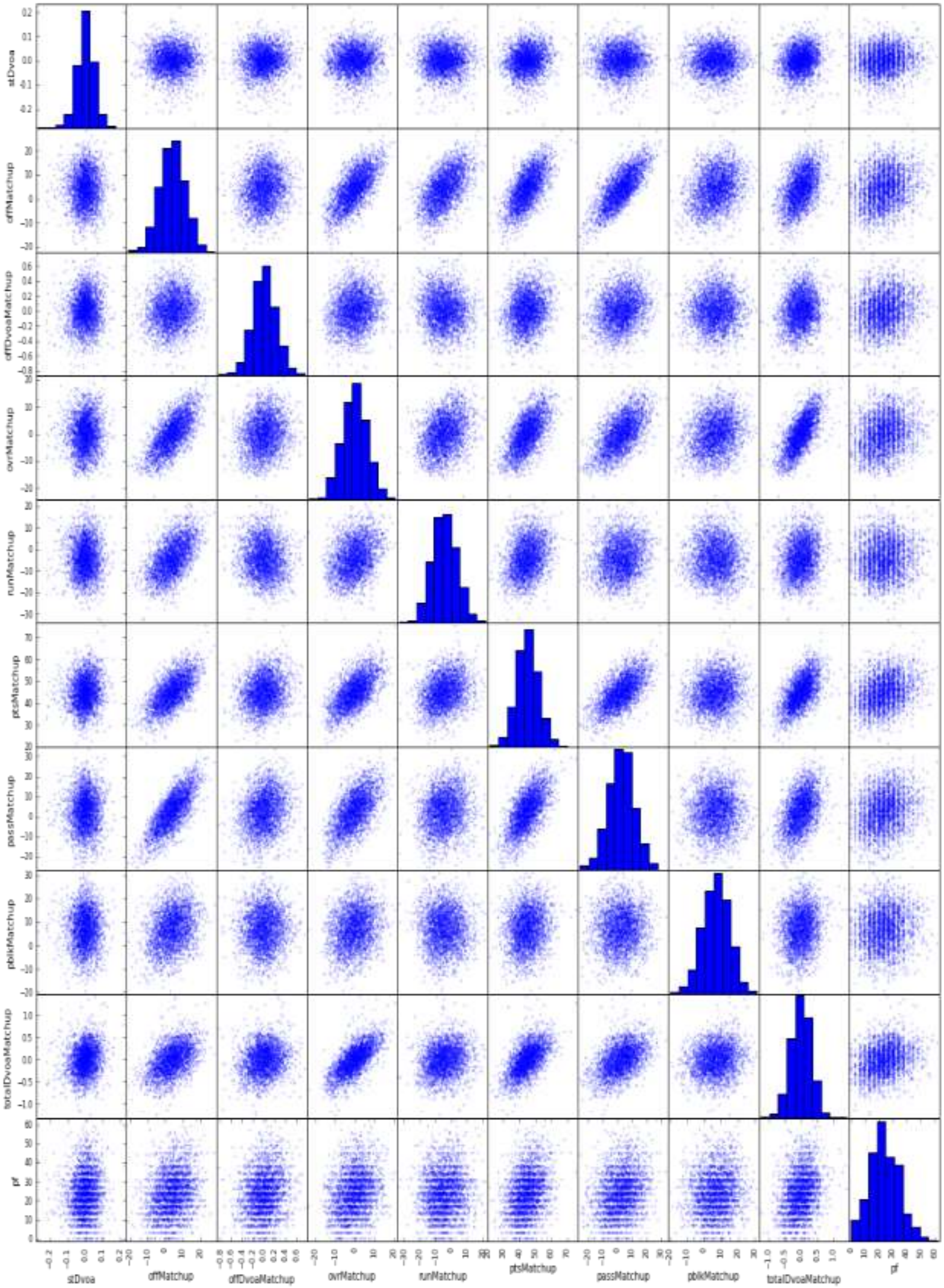
Looking at the correlation heatmap matrix for both tables, we can see some positive correlations between the independent variables and pf variable for both tables. Additionally, there are some notably strong correlations between independent variables in both tables as well (passMatchup and offMatchup have a pearson correlation of about .77 in both tables). We should make sure to keep this in mind when we begin the model-building process, as the issue of multicollinearity may arise.

There are also some positive correlations between the independent variables, and the dependent variable for both home teams and away teams. The strongest correlations for home teams with respect to points scored are ptsMatchup, totalDvoaMatchup, offMatchup, passMatchup, and ovrMatchup, with r = .2853, .2693, .2360, .2341, and .2334, respectively. The strongest correlations for away teams with respect to points scored are ptsMatchup, totalDvoaMatchup, offMatchup, passMatchup, and ovrMatchup, with r = .2925, .2804, .2490, .2567, and .2385, respectively.

Using critical values for Pearson's Correlation Coefficient with alpha = 0.05, we have enough samples, and thus degrees of freedom, from both the home and away tables to put a baseline correlation of 0.06 as our mark of whether or not a variable's correlation is significant. Looking at the correlation heatmap, we see that for the homeMatchup table, all dependent variables have significant correlations to the target variable (pf). In the awayMatchup table, all independent variables except for stDvoa and pblkMatchup have significant correlations.

For the home table, we can say that there was a significant positive relationship between all individual independent variables, and the dependent variable ($p < 0.05$). For the away table, we can say that there was a significant positive relationship between all individual independent variables ($p < 0.05$) except for stDvoa and pblkMatchup ($p > 0.05$), and the dependent variable.

Pictured on the next page is the scatterplot matrix, which shows how all of the variables for home teams correlate with each other, as well as a histogram for each variable. It is a reasonable assumption to deem each histogram as “normal” for each variable, based on the visualizations. The dependent variable, pf, is the very last row and column.


stDvoa
offMatchup
offDvoaMatchup
ovrMatchup
runMatchup
ptsMatchup
passMatchup
pblkMatchup
totalDvoaMatchup
pf
−0.2 −0.1 0.0 0.1 0.2
−20 −10 0 10 20
−0.8 −0.6 −0.4 −0.2 0.0 0.2 0.4 0.6
−20 −10 0 10 20
−30 −20 −10 0 10 20
20 30 40 50 60 70
−20 −10 0 10 20 30
−20 −10 0 10 20 30
−1.0 −0.5 0.0 0.5 1.0
0 10 20 30 40 50 60

Machine Learning

Now that we've used inferential techniques to explore our data further, we can proceed with the model building process. Due to the nature of our problem, we will be using supervised learning for numeric predictions. More specifically, we will be building regression models to predict points scored for away teams and home teams. After we have predicted the number of points scored for away teams and home teams, we will combine the predicted points scored for both teams to get a prediction for the total number of points scored in every game. This total can be compared to the total that was put out by the sports book, and we can make a decision to take the over or under based on the comparison.

We start with our dataset that was prepared in the last few sections, which we split into a dataset for away teams and home teams. In both tables, the variables that will be considered in our models are stDvoa, runMatchup, ptsMatchup, offDvoaMatchup, offMatchup, ovrMatchup, passMatchup, surface, pblkMatchup, roof, and totalDvoaMatchup. For the roof and surface variables, we transform them into binary variables, since each of them only has two possible classes (roof: dome or outdoors, surface: grass or turf). Now that we have all variables into number form, we proceed by splitting the datasets into train/test data. We use 80% of each dataset as training data, and 20% for testing.

For our first model, we use linear regression from Python's sklearn library. We fit the training data to the training target variable, points scored, for both away and home teams. We then cross validate our training data to ensure that we are not overfitting. For linear regression, we use R-squared score as our metric for cross-validation. The average score across our ten folds of data is roughly 0.11, and the R-squared range for the ten folds is from 0.05-0.18.

Next, we set up our model so that we can begin to eliminate variables that aren't significant in predicting our target. We use the backwards stepwise regression strategy to eliminate variables. We proceed by fitting the model with all variables, then eliminating the variable with the highest p-value, as long as the p-value is over a certain threshold. In this case, we use alpha = 0.05 as the threshold. For the away table, the variables that made it to the final model are ptsMatchup, offDvoaMatchup, passMatchup, and totalDvoaMatchup. The adjusted R-squared for this model is roughly 0.114, and the equation for the model is as follows:

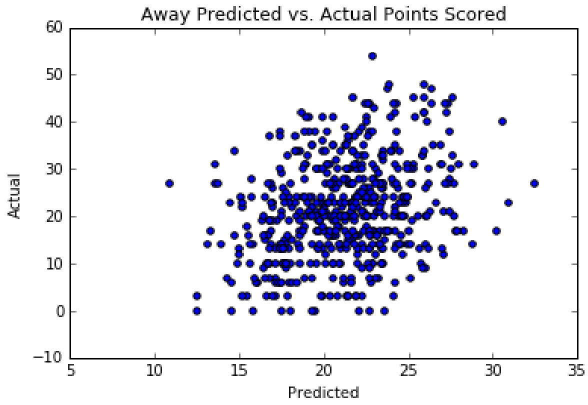
$$y_{away} = 0.195x_{ptsMatchup} + 2.566x_{offDvoaMatchup} + 0.099x_{passMatchup} + 4.957x_{totalDvoaMatchup} + 12.042$$

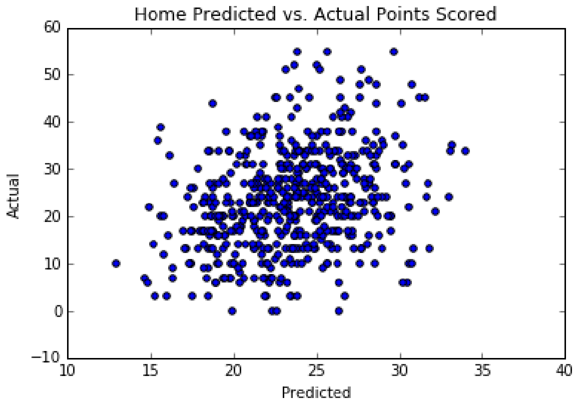
For the home table, the variables that made it to the final model are ptsMatchup, offDvoaMatchup, pblkMatchup, roof, and totalDvoaMatchup. The adjusted R-squared for this model is roughly 0.121, and the equation for the model is as follows:

$$y_{home} = 0.279x_{ptsMatchup} + 5.944x_{offDvoaMatchup} + 0.071x_{pblkMatchup} - 1.61x_{roof} + 4.805x_{totalDvoaMatchup} + 11.919$$

Now that we have our equation for both away and home teams, we can make predictions on our test data, check our model assumptions, and evaluate our models.

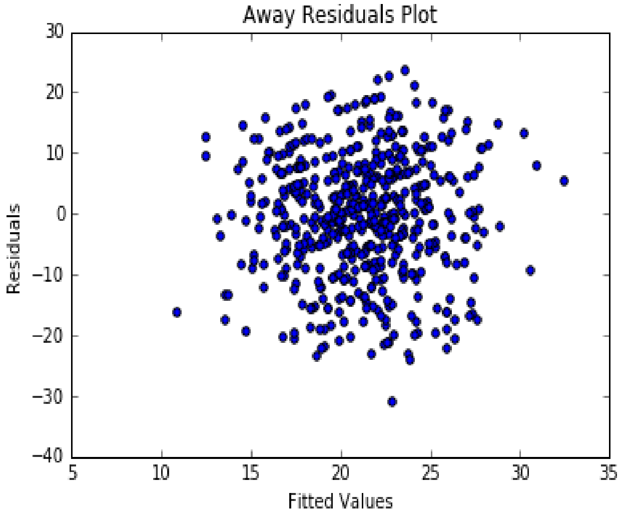
First, we plot our predicted vs actual for both models. We hope to observe a positive correlation so that we know our models are predicting with some degree of accuracy.

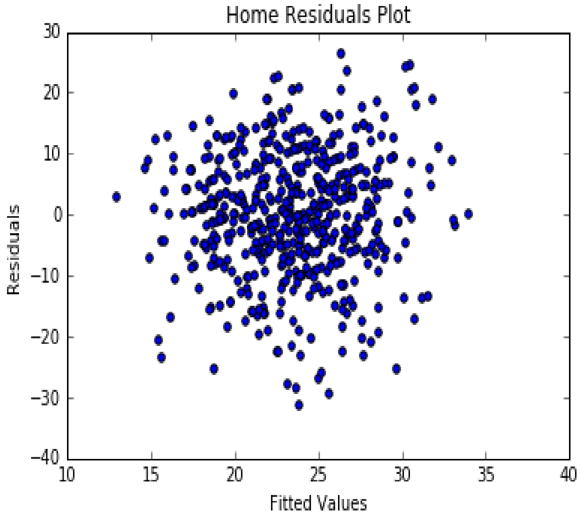




We can see a positive correlation for both the away and home models. For the away model, the pearson correlation is roughly 0.307, and for the home model, it is approximately 0.306.

Next, we plot the fitted vs residuals for both models to ensure there are no patterns with the residuals, heteroskedasticity, etc.

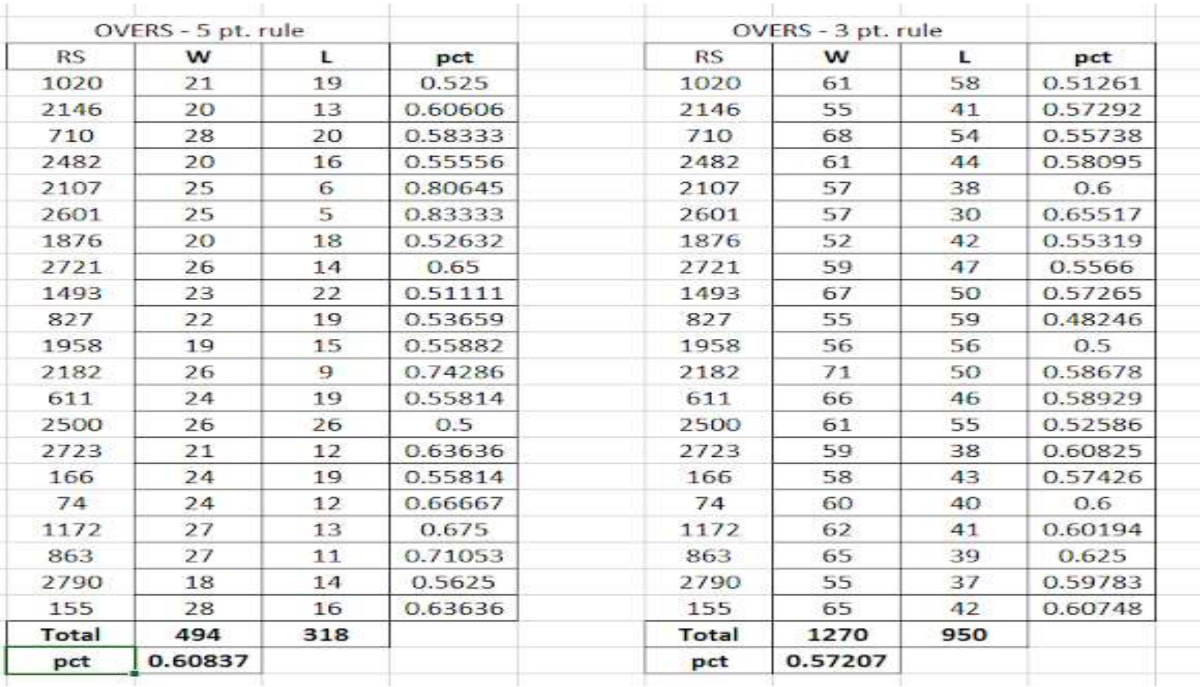




As we can see, there are no violations with our residuals plots. Observing the histograms of the residuals in our jupyter notebook, we conclude that both models have approximately normal residuals as well. From observing the scatterplot matrices in the last section, we also know that we have no multicollinearity issues for our model. Thus, all model assumptions are valid, and we can feel confident about deploying these models.

We next merge our predictions for away and home teams together based on gameId, so that we may add the two scores together to get a total to compare to the total put out by the sports book. After comparing the predictions to the totals put out by the sports book, a pattern was identified that helps making predictions on the over/under more accurate. Instead of betting the over if our prediction is greater than the sports book's total, or under if our prediction is less than the number, we notice that there is a huge edge when only betting games in which the prediction exceeds the sports book's total by five or more points. We call this the "5-point Overs Rule". We also test for a 3 and 1-point Overs rule, and do the same for unders. To gather results for our system, we train 21 random seeds of our dataset, and test on each seed's respective test set using the exact same variables for both the away and home teams. It was found that using the 5-point overs rule, our average win percent was 60.8%. This exceeds the required 52.4% needed to be long-term profitable when betting on NFL games. This would be good for a 8.4% return on investment if one were to follow this system's recommended plays. The results for the 3-point rule also proved to be profitable, but to a lesser extent. An average win rate of 57.2% was found using this rule, although there were nearly triple the amount of plays recommended using this system.

Using the 5-point overs rule, clients can expect to receive approximately 19 recommendations per season, or around 1 recommendation per week. Using the 3-point overs rule, clients can expect to receive approximately 52 recommendations per season, or around 3 recommendations per week. The results for these two rules are as follows:



OVERS - 5 pt. rule

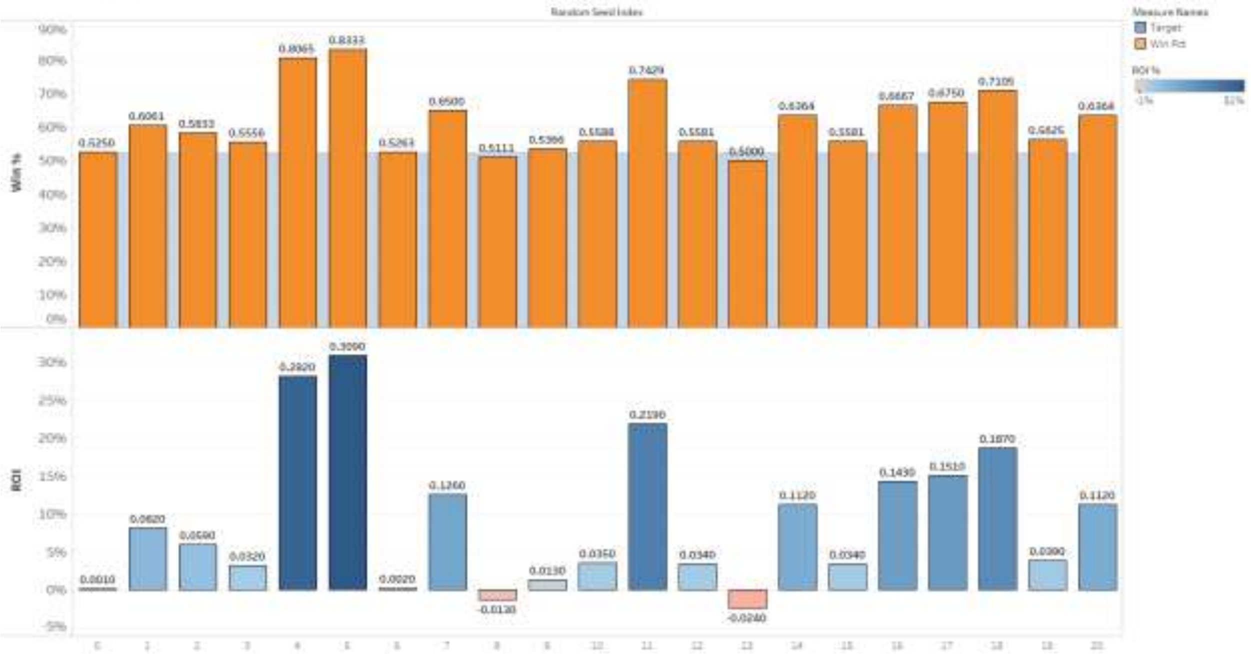
RS	W	L	pct
1020	21	19	0.525
2146	20	13	0.60606
710	28	20	0.58333
2482	20	16	0.55556
2107	25	6	0.80645
2601	25	5	0.83333
1876	20	18	0.52632
2721	26	14	0.65
1493	23	22	0.51111
827	22	19	0.53659
1958	19	15	0.55882
2182	26	9	0.74286
611	24	19	0.55814
2500	26	26	0.5
2723	21	12	0.63636
166	24	19	0.55814
74	24	12	0.66667
1172	27	13	0.675
863	27	11	0.71053
2790	18	14	0.5625
155	28	16	0.63636
Total	**494**	**318**	
pct	**0.60837**		

OVERS - 3 pt. rule

RS	W	L	pct
1020	61	58	0.51261
2146	55	41	0.57292
710	68	54	0.55738
2482	61	44	0.58095
2107	57	38	0.6
2601	57	30	0.65517
1876	52	42	0.55319
2721	59	47	0.5566
1493	67	50	0.57265
827	55	59	0.48246
1958	56	56	0.5
2182	71	50	0.58678
611	66	46	0.58929
2500	61	55	0.52586
2723	59	38	0.60825
166	58	43	0.57426
74	60	40	0.6
1172	62	41	0.60194
863	65	39	0.625
2790	55	37	0.59783
155	65	42	0.60748
Total	**1270**	**950**	
pct	**0.57207**		

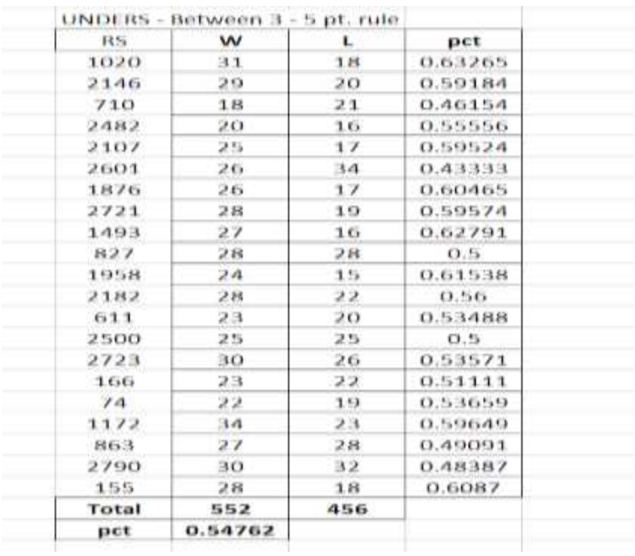
In the tables above, RS is the random seed of data we are testing on, and W and L are the number of wins and losses that the random seed of data observed using this system. Each seed's win % is shown on the right, and the total win % is shown at the bottom of each table.

Win % of Each Random Seed



In the above chart, we visualize our results from the 5-point overs rule. The win % and profit from each random seed is shown, as well as the 52.4% target shaded in blue in the background of the upper visualization. We can see that using this rule, we get only two of twenty-one random seeds of data that do not exceed the 52.4% win percentage mark needed to be profitable. Using the 3-point rule, we get three of twenty-one random seeds of data that aren't profitable. One could be confident that using our system will give them a very good chance of seeing a positive return on their investment.

We repeat this entire process of training our dataset and recording our results for both Random Forest and XGBoost regressors. Neither of these two regressors were able to exceed the ROI provided by using the multiple linear regression (MLR) model, but XGBoost was able to outperform the MLR model's unders predictions using a "Between-3-and-5-point Rule". This means that when the XGBoost model predicts a total that is between three and five points less than the total put out by the sports book, we should use this model instead of our MLR model. Roughly a 2.362% ROI was discovered using this XGBoost model rule. The results table for this rule using XGBoost are below:



UNDERS - Between 3 - 5 pt. rule

RS	W	L	pct
1020	31	18	0.63265
2146	29	20	0.59184
710	18	21	0.46154
2482	20	16	0.55556
2107	25	17	0.59524
2601	26	34	0.43333
1876	26	17	0.60465
2721	28	19	0.59574
1493	27	16	0.62791
827	28	28	0.5
1958	24	15	0.61538
2182	28	22	0.56
611	23	20	0.53488
2500	25	25	0.5
2723	30	26	0.53571
166	23	22	0.51111
74	22	19	0.53659
1172	34	23	0.59649
863	27	28	0.49091
2790	30	32	0.48387
155	28	18	0.6087
Total	**552**	**456**	
pct	**0.54762**		

Next Steps/Future Work

Now that we have finished building, testing, and evaluating our models, we can begin to think about what steps to take next. The model building process resulted in, on average, at least one recommendation per week for potential clients, friends, or ourselves. This would serve as the starting point as a pitch to our audience, but it would be nice to be able to recommend more games with high confidence. One area we could explore is collecting additional data that deals more with in-game statistics, such as passing yards per game, opponents passing yards allowed per game, etc. We could see how these new variables add to our models, and tinker with which variables are included in our models. Additionally, we could try out additional algorithms, and see if any of them could outperform the algorithms we used.

We could also take our findings to the sportsbooks, and pitch this as a way to improve their totals that they release to the public. Knowing information that we relay to the sportsbooks could help them tinker with their process in a way that allows them to see increased long-term ROI.