ROOT, ROOT, ROOT FOR THE HOME TEAM?:

Estimating the Probability of the Home Team Winning a Regular Season MLB Game

Springboard Data Science Career Track
By Sinead O'Connor
December 2019

I. Introduction

Professional baseball is a very competitive sport, and thus predicting the outcome of a professional baseball game is challenging. In a single of Major League Baseball regular season there are 2,430 games, presenting plenty of opportunity for fans (or non-fans) to bet on which team will win.

I built classification models to estimate the probability of the home team winning a regular-season MLB game given a vector of features that characterize the home team and the visiting team. I first built a baseline logistic regression model, then extended model by selecting different features. I also built a Random Forest model, and finally used the most important features in the Random Forest model, to build a new Logistic Regression model with better results.

Building a model that predicts which team wins will be beneficial to a company who charge people a fee to get recommendations on which team to bet on.

II. Approach

A. Data Acquisition and Wrangling

I first imported Game Logs for the 2009-2018 seasons from the website Retrosheet.org using urllib request. The game log contains information on offensive, defensive, and pitching statistics on each game for both the home team and visiting team. Each log also includes a list of starting players for each team, as well as information such as date, day of the week, time of day (day/night), and attendance. https://www.retrosheet.org/gamelogs/

Since starting pitching is considered to be such an important part of a game, I used BeautifulSoup to scrape baseballmusings.com for starting pitcher logs for each team from the 2009-2018 seasons. The pitching logs have information about the starting pitchers for each game such as number of innings pitched, walks, strikeouts, and earned runs. For details, see:
https://www.baseballmusings.com/ChooseStarterTeam.html

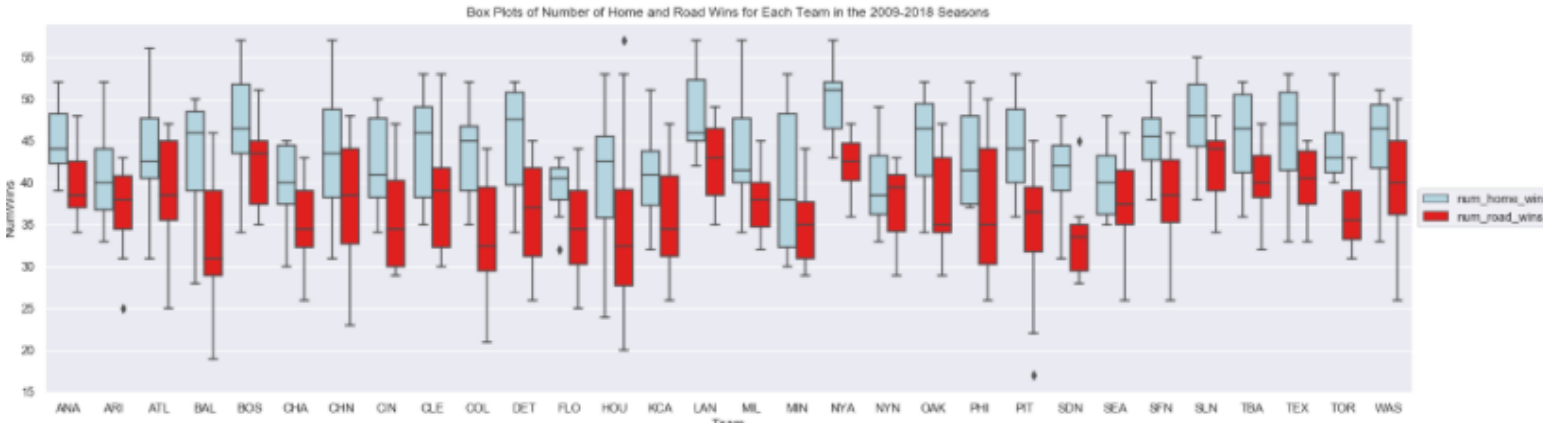
I cleaned the Retrosheet gamelogs and the Baseball Musings starting pitcher logs individually. Once they were cleaned, I merged them. The resulting dataframe has 24,298 rows, each representing a regular-season game, and 205 columns, each giving a different piece of information about the events of the game.

B. Exploratory Data Analysis – Data Storytelling

In the first part of Exploratory Data Analysis, Data Storytelling, I explored many questions having to do with the data.

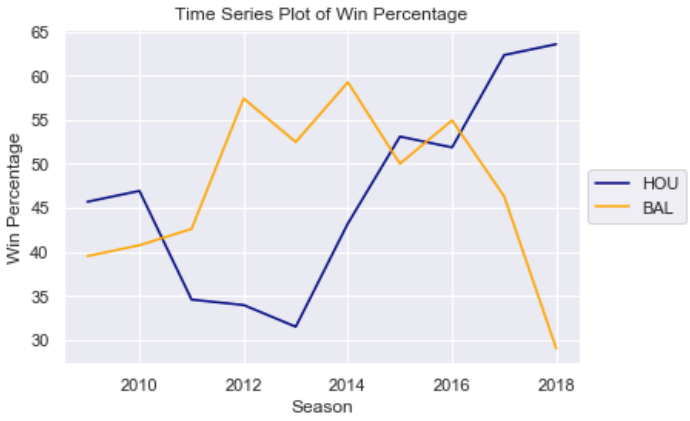
I found that in the 2009-2018 seasons, the home team won 53.7% of games. In each individual season, the home teams have won a higher percentage of games. In addition, each team higher percentage of their home games than their road games.

The below boxplots show the number of home games and road games each team has won in the 2009-2018 seasons.



These boxplots suggest that each team does play better at home (blue) than on the road (red). This seems to be particularly true of teams such as the Los Angeles Angels (ANA), and the New York Yankees (NYA), and San Diego Padres (SDN), and San Francisco Giants, whose box plots for home wins and road wins have very little overlap. For teams such as the New York Mets (NYN), and the Seattle Mariners (SEA), whose boxplots do seem to have a lot of overlap, there is less discrepancy in their record at home and on the road.

Also, while some teams such as the New York Yankees consistently have winning records, other teams have not been very consistent, such as the Houston Astros and the Baltimore Ravens, whose winning percentage timelines are shown below.



None of the findings in my Data Storytelling Jupyter notebook were particularly surprising. Some other conclusions are summarized below:

- The American League won 56.8% of games at home when playing against a National League team. The National League team won 50.3% of games at home when playing American League team, so it seems the American League might be slightly better.

- As average attendance increases, so does the number of games won. It could be that the large crowds help the home team perform better, or it could be that there is higher attendance because the team is winning more, so may be more exciting to watch.

- The number of errors made per team per game is very low, which is a good sign since it is professional baseball, and there does not seem to be a big difference between the number of errors made by the home team and the visiting team.

- Starting pitchers on the winning team do seem to pitch more innings than the starting pitcher on the losing teams. This makes sense because if the starting pitcher is performing poorly, they will be taken out of the game earlier. On average the starting pitcher of the winning team pitches 0.92 innings more.

- In 52.8% of games in the 2009-2018 seasons the team with the higher on-base percentage won the game. This is not too surprising because in most games the average on-base percentage of the opponents are not very far apart, and there are many factors that go into winning an individual game.

C. Exploratory Data Analysis – Inferential Statistics:

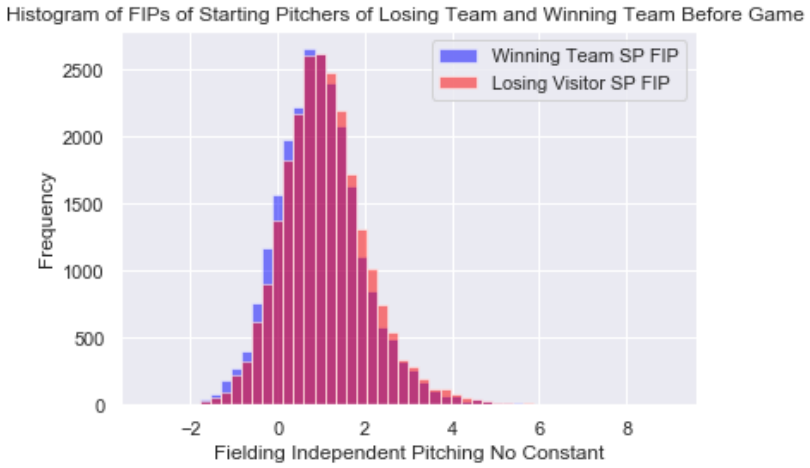
Before starting the Inferential Statistics, I created features by computing averages for starting pitchers and team averages. I also added columns for major statistics such as On-Base Percentage, Isolated Power, Earned Run Averages, Fielding Independent Pitching, Batting Average on Balls in Play, and more. I computed the averages per team per season, and used data from the end of the prior season to fill the first 5 games of each season. The averages are computed using only information from before the game takes place, so it can be used to predict the outcome of future games.

I then moved on to Inferential Statistics where I explored many hypotheses.

Here is a summary of the findings:

- The proportion of home games won by the American League and that won by the National League from the 2009-2018 seasons (53.974% and 53.488%, respectively) was not statistically significant.
- There is a statistical significance between the teams On-Base Percentage at home and on the road. It tends to be higher at home. We must keep in mind that this could be due to the very large sample size.

- There is a statistically significant difference between the Fielding Independent Pitching stat prior to the game of the starting pitcher on the team that goes on to win the game and the starting pitcher on the losing game.

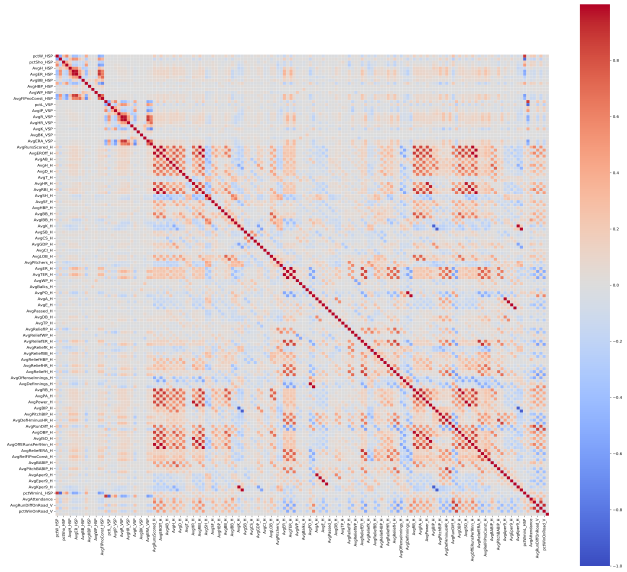


I also identified collinear variables by creating a heat map of the correlation matrix using all columns that included averages, as well as the column that indicates whether the home team won or lost.

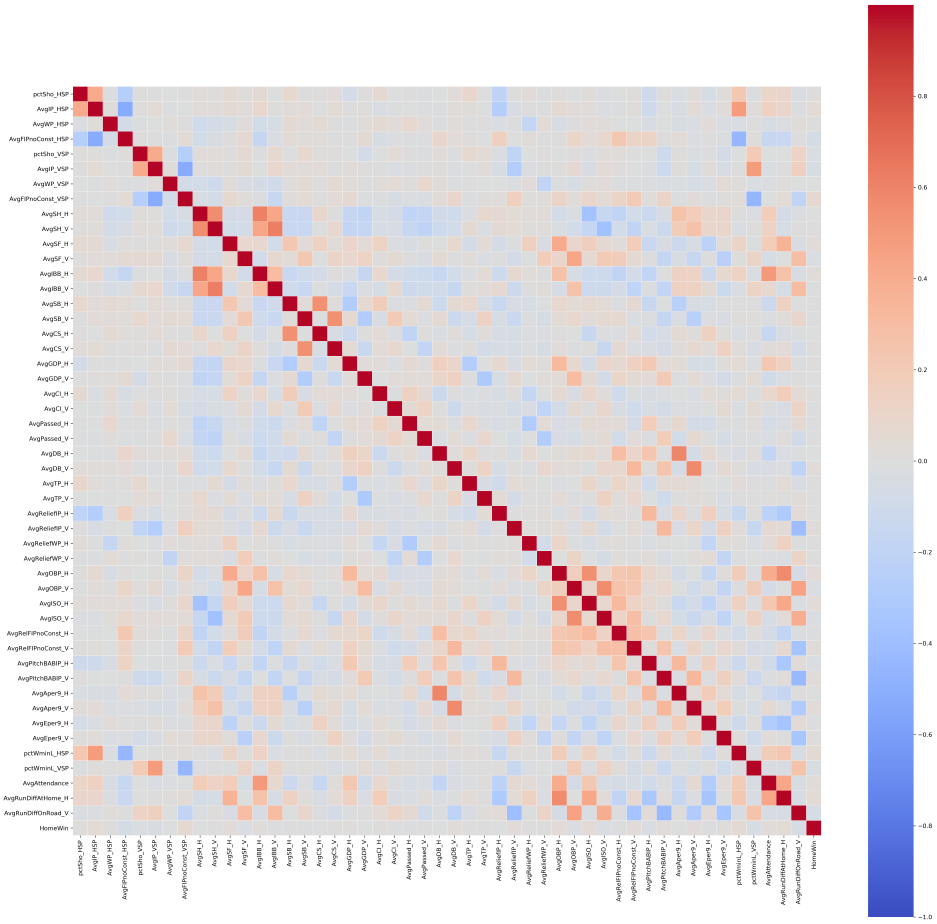
Of the variables that were highly correlated, most were trivial. For example, earned runs (AvgTER) and individual earned runs (AvgER) as well as the ones that are functions of other variables, for example BIP (balls in play) is a function of K (strikeouts).

D. Baseline Logistic Regression Model:

I built a baseline logistic regression model using expanding averages from the beginning of the 2009 season. Before building, the model I removed one of the features for each pair of collinear features. Below is a heat map of the correlation matrix:



As discussed in the previous section, there were many trivially collinear features, and we see this with the dark red squares in the heat map. Removing the correlated features resulted in a new heat map.



There were 49 features remaining. The target variable was the column "HomeWin". A "1" in this column indicates that the home team won and a "0" indicates that the visiting team won. There were many null values in April 2009 since this is the season I started data collection for, so we could not compute averages. Therefore, I used May 2009 through the 2017 season to train the model, and all games from the 2018 season to test it. Before building the model, I scaled the training data using StandardScaler(). I tested the data using the scaled test data (2018 games). In the test set, the home team wins 52.7% of the games. The training and test classification reports as well as accuracy and AUC are below.

[Training Classification Report:]

	precision	recall	f1-score	support
0	0.55	0.27	0.36	9948
1	0.56	0.81	0.66	11596
micro avg	0.56	0.56	0.56	21544
macro avg	0.55	0.54	0.51	21544
weighted avg	0.55	0.56	0.52	21544

Training Accuracy: 0.5585313776457482

[Test Classification Report:]

	precision	recall	f1-score	support
0	0.55	0.24	0.34	1148
1	0.55	0.82	0.66	1283
micro avg	0.55	0.55	0.55	2431
macro avg	0.55	0.53	0.50	2431
weighted avg	0.55	0.55	0.51	2431

Test Accuracy: 0.5475113122171946

AUC: 0.560206370630681

The test accuracy and training accuracy were close, suggesting that there was not much over fitting. Precision was 0.55 for both classes, but the recall was much better for class 1 (home team wins) than for class 0, at 0.82 and 0.24 respectively. The model was classifying most of the games as wins for the home team.

Below is the confusion matrix. The model is predicting that the home team wins about 79% of the time.

		PREDICTED RESULT	
		LOSS	WIN
TRUE RESULT	LOSS	279	869
	WIN	231	1052

E. Extended Modeling:

To extend the model, I tried several methods to improve the baseline model. I used the parameter class_weight = 'balanced' on the same set of features. This decreased test accuracy, and precision, recall, by about 0.01 each, but increased the average F1-score and the results were much less skewed towards the home team.

I also tried choosing a subset of features that I thought may accurately predict the game. I chose features that would represent each aspect of the game: pitching, defense, batting, and base running. For each feature I chose pertaining to the home team (ending in _H or _HSP), I chose the corresponding feature for the visiting team (ending in _V or _VSP).
Pitching stats chosen:

- pctWminL_HSP and pctWminL_VSP: The percent of games in which the starting pitcher records a win minus the percent of games in which he records a loss.
- AvgFIPnoConst_HSP and AvgFIPnoConst_VSP: The Fielding Independent Pitching statistics for the starting pitchers. I did not include the constant term. This measures the events pitchers have the most control over, such as walks, strikeouts, and home runs.
- AvgRelFIP_H and AvgRelFIP_HSP: The average FIP of the relief pitchers for each team.
- AvgIP_HSP and AvgIP_VSP: The average number of innings pitched per start by the starting pitchers.
- AvgPitchBABIP_H and AvgPitchBABIP_V: The average Batting Average on Balls in Play of the entire pitching staff (starters and relief included).

Defense/Fielding stats:

- AvgDB_H and AvgDB_V: Average number of double plays made by the defense per game.
- AvgAper9_H and AvgAper9_V: Average number of assists per 9 innings.
- AvgEper9_H and AvgEper9_V: Average number of errors per 9 innings.

Offense- Batting & Baserunning:

- AvgOBP_H and AvgOBP_V: On-Base Percentage of the batters.
- AVgISO_H and AvgISO_V: Isolated Power- gives more weight to extra-base hits. This may also indicate baserunning because fast players may be able to turn a single into a double.

- AvgSB_H and AvgSB_V: Average number of stolen bases per game.
- Avg CS_H and AvgSB_V: Average number of times caught stealing per game.

Other/General:

- AvgRunDiffAtHome_H: Average run differential (spread) of the home team at their home games.
- AvgRunDiffOnRoad_V: Average run differential (spread) of the visiting team at their away games.
- AvgAttendance: Average attendance at the ballpark of the home team.

Building a model on this subset of variables resulted in higher test accuracy, precision, recall, F1 score and AUC.

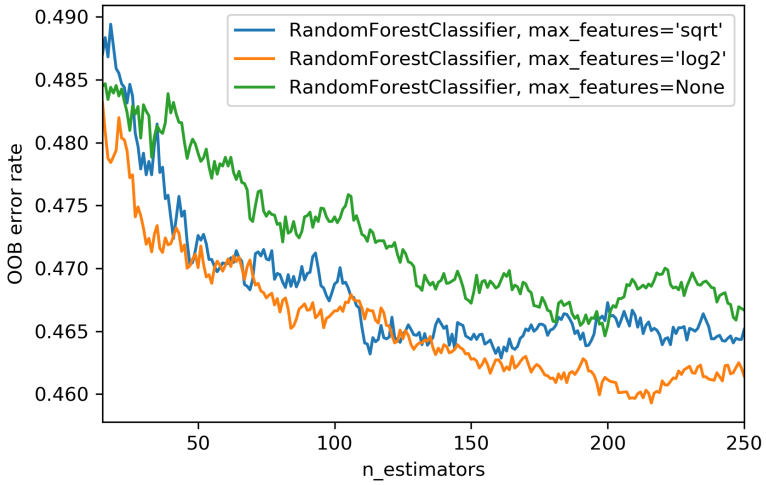
Below is a list of strategies I attempted to improve results:

- Using expanding averages of stats for the current season only. For the first 3 games of each season I used previous season data also. I chose to do this because players and coaches can change from season to season, and even if they do not- having a good season one year does not necessarily mean you will perform the same the next season. Take for example, the Boston Red Sox who set a MLB record in 2018 winning 108 games, and only 87 games in 2019 despite having mostly the same players. For starting pitchers first starts, I used the average stats of other pitchers facing the same opponent.
- Using the same subset of statistics chosen above on the current season averages.
- Using rolling averages. I chose to use the previous 162 games (a regular seasons worth of games). For the stats that applied only to the home team or visiting team (AvgAttendance, AvgRunDiffAtHome_H, and AvgRunDiffOnRoad_H), I used a rolling window of 81, and for starting pitchers, I used a rolling average of 30 starts.
- Using only the subset of stats chosen above on the rolling average data.
- At this point, the best results came from using the rolling averages, so I decided to explicitly compare the home team and visiting team by subtracting the visiting team stats from the corresponding home team stat using the rolling average data.
- Using the differences of the subset of stats that I chose above. This had the best results in terms of test accuracy, recall, precision, F1-score, and AUC.

F. Random Forest:

The best performance of the logistic regression models I built resulted from explicitly comparing the stats of the home team and visiting team and using the subset I chose. Since I chose the subset solely based on intuition, I decided to use the full set of the differences of the rolling average features to build a Random Forest model and try to find a better subset of features.

I first plotted the below out-of-bag (OOB) error rate vs. n_estimators plot to optimize the model.



From this plot, we see that the out-of-bag error rate is minimized when using the log of the number of features for max_features and for n_estimators > 250.

Therefore I built a Random Forest model using max_features = 'log2' and n_estimators = 1000 as parameters and got these results:

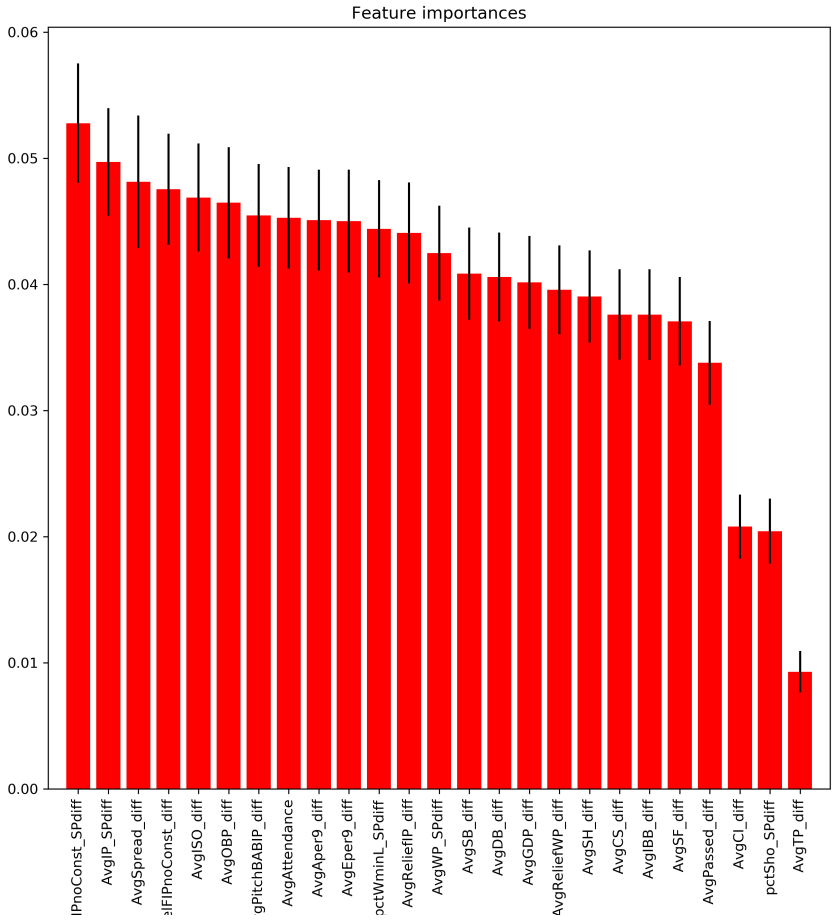
[Test Classification Report:]

	precision	recall	f1-score	support
0	0.58	0.40	0.47	1148
1	0.58	0.75	0.65	1283
micro avg	0.58	0.58	0.58	2431
macro avg	0.58	0.57	0.56	2431
weighted avg	0.58	0.58	0.57	2431

Test Accuracy: 0.5808309337721103

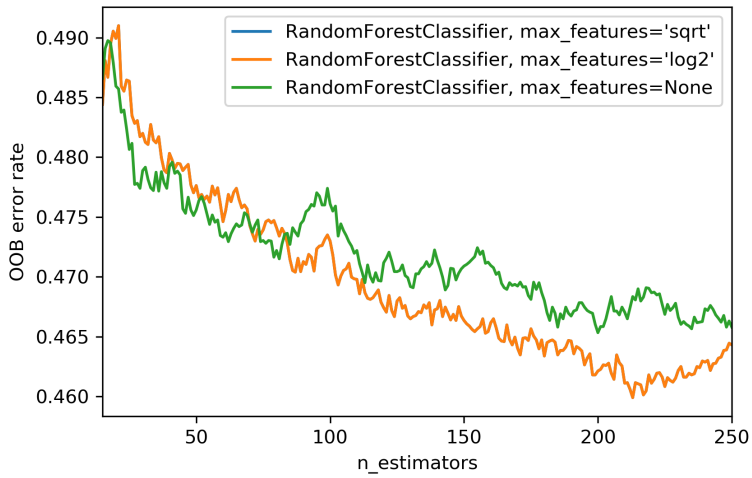
AUC: 0.5989623079617947

I also graphed the feature importances:



Based off of this bar graph, I dropped the 3 least important features and retrained the model and had better results, so I dropped the 6 least important features [AvgIBB_diff (intentional walks), AvgSF_diff (sacrifice flies), AvgPassed_diff (passed balls), AvgCI_diff (catchers interference), pctSho_SPdiff (shutouts by the starting pitcher), AvgTP_diff (triple plays). Nothing about these rankings were particularly surprising based on knowledge of the game, although I thought that AvgSpread_diff may rank the highest.

This was the resulting OOB error vs. n_estimators plot:



Again 'log2' minimized the OOB error rate, so I built the Random Forest model using 1,000 trees and max_features = 'log2' and got these results, which were better than before:

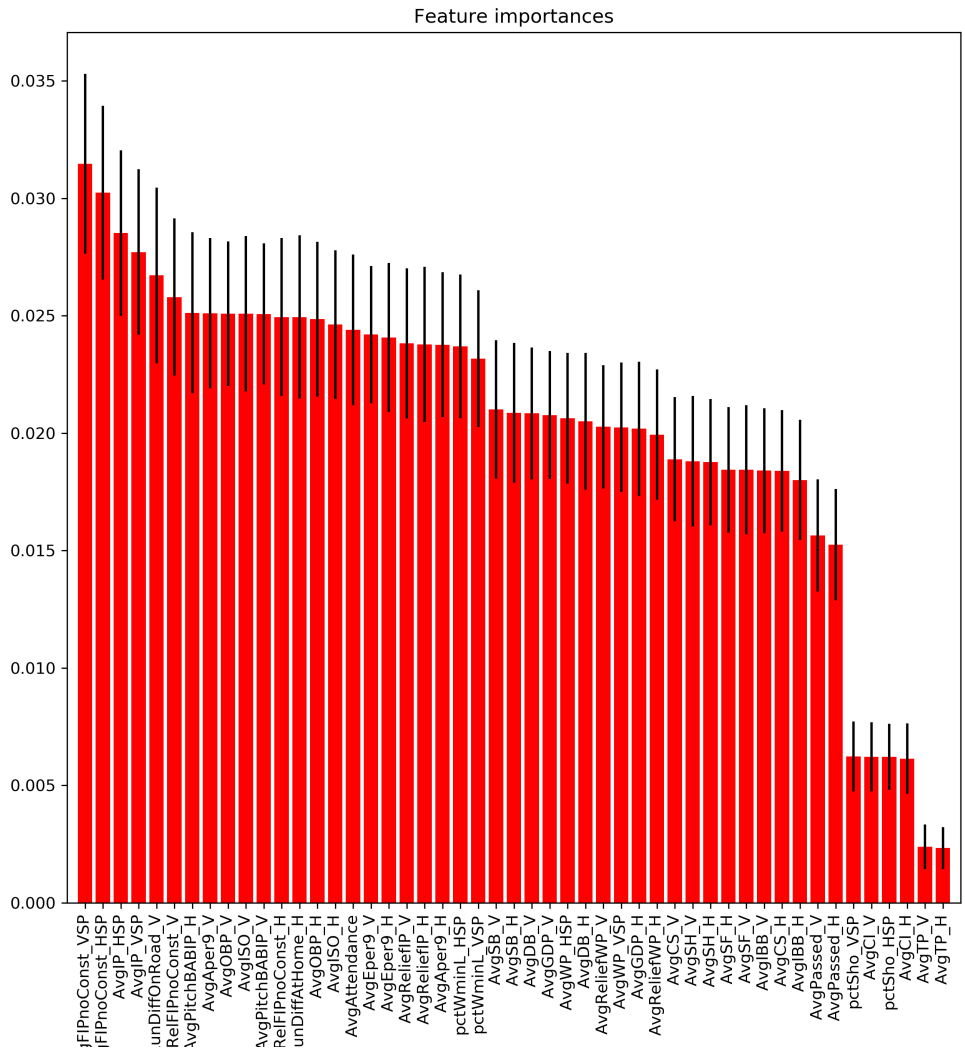
[Test Classification Report:]

	precision	recall	f1-score	support
0	0.59	0.42	0.49	1148
1	0.59	0.74	0.66	1283
micro avg	0.59	0.59	0.59	2431
macro avg	0.59	0.58	0.57	2431
weighted avg	0.59	0.59	0.58	2431

Test Accuracy: 0.5882352941176471
AUC: 0.6004736286089061

If it were not for restrictions due to time and computing power, I would have tried building the model using higher numbers of trees.

Note that out of curiosity, I also built a Random Forest model on the full set of features to see if it would perform better, but the results were better for the differences. It is interesting, however, to look at the Feature Importance rankings.



We see that in many cases the home team feature and the corresponding visiting team feature are ranked right next too each other. Comparing this graph, to the bar chart of the differences, we also see that they rank in similar positions, for example, in both cases the most important features are average FIP and the innings pitched by the starting pitcher and the least important are percent of shutouts by the starting pitcher, average number catchers interferences, and average number of triple plays.

G. Logistic Regression- Revisited:

As a final attempt (for now) to improve the logistic regression model, I used the same difference features that resulted in the best Random Forest model.

[Training Classification Report:]

	precision	recall	f1-score	support
0	0.52	0.56	0.54	9948
1	0.60	0.56	0.58	11596
micro avg	0.56	0.56	0.56	21544
macro avg	0.56	0.56	0.56	21544
weighted avg	0.56	0.56	0.56	21544

Training Accuracy: 0.5588098774600817

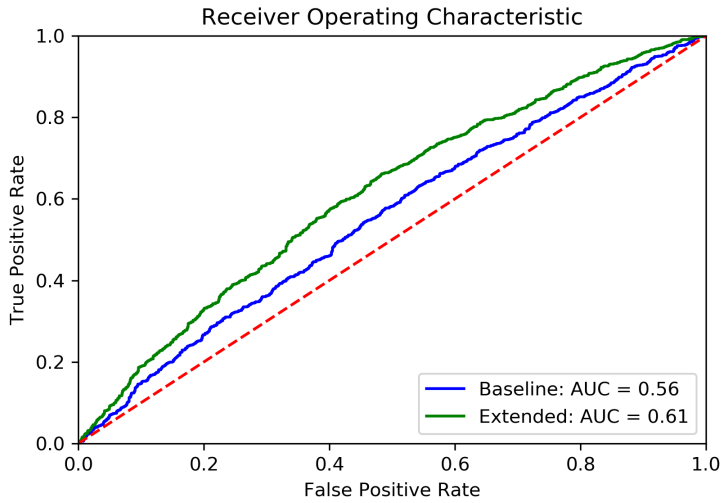
[Test Classification Report:]

	precision	recall	f1-score	support
0	0.56	0.59	0.57	1148
1	0.61	0.58	0.60	1283
micro avg	0.58	0.58	0.58	2431
macro avg	0.58	0.59	0.58	2431
weighted avg	0.59	0.58	0.59	2431

Test Accuracy: 0.5849444672974085

AUC: 0.6119891315269906

These results were an improvement over the subset of variables I had chosen. Since this was the best Logistic Regression model, I graphed an ROC Curve to visualize improvements over the baseline.



H. Simulation:

Once I had the best result Random Forest model and Logistic Regression model, I made a simple simulation to see how much targeted clients would make if they used the models' predictions to bet on games.

In this very simplified model, the client bets $100 on each game. If they are wrong, they lose the money and if they are right they win $50 per game.

I chose a random subset of 30 games in the 2018 season (the test set), and under the simulation a client who placed all bets using the logistic regression model would have lost $600 with a Return on Investment (ROI) of -20%, while if they bet using the Random Forest model, they have win $150, with an ROI of 5%.

Date	Visiting Team	Home Team	Prob of Home Win LR	LR winnings	Prob of Home Win RF	RF winnings	Winner
2018-04-12	CHA	MIN	0.569380	50	0.629	50	Home
2018-04-13	MIL	NYN	0.457341	-100	0.554	50	Home
2018-04-13	SFN	SDN	0.501470	50	0.553	50	Home
2018-04-19	TOR	NYA	0.568426	50	0.699	50	Home
2018-04-30	NYA	HOU	0.486090	-100	0.561	50	Home
2018-05-01	NYA	HOU	0.515514	-100	0.561	-100	Visitor
2018-05-04	CHN	SLN	0.481290	-100	0.414	-100	Home
2018-05-15	CLE	DET	0.399107	-100	0.494	-100	Home
2018-05-20	LAN	WAS	0.491341	50	0.492	50	Visitor
2018-05-21	FLO	NYN	0.504208	50	0.512	50	Home
2018-05-21	KCA	SLN	0.578380	50	0.633	50	Home
2018-05-29	CHA	CLE	0.660552	50	0.682	50	Home
2018-06-02	TOR	DET	0.438834	-100	0.503	50	Home
2018-06-18	FLO	SFN	0.536889	-100	0.573	-100	Visitor
2018-06-19	TEX	KCA	0.466499	50	0.554	-100	Visitor
2018-06-29	KCA	SEA	0.584006	50	0.528	50	Home
2018-07-05	ATL	MIL	0.535343	50	0.535	50	Home
2018-07-12	NYA	CLE	0.457258	50	0.454	50	Visitor
2018-07-13	ARI	ATL	0.457604	50	0.498	50	Visitor
2018-07-22	LAN	MIL	0.481166	50	0.540	-100	Visitor
2018-07-26	CHA	ANA	0.593188	50	0.531	50	Home
2018-07-31	CHN	PIT	0.481106	-100	0.518	50	Home
2018-08-05	NYA	BOS	0.484164	-100	0.593	50	Home
2018-08-08	LAN	OAK	0.420042	-100	0.555	50	Home
2018-08-11	ARI	CIN	0.421537	-100	0.575	50	Home
2018-08-26	SEA	ARI	0.521747	50	0.634	50	Home
2018-08-29	TBA	ATL	0.509076	-100	0.586	-100	Visitor
2018-09-05	MIN	HOU	0.609870	50	0.548	50	Home
2018-09-12	SDN	SEA	0.511387	-100	0.543	-100	Visitor
2018-09-30	CHA	MIN	0.474632	-100	0.434	-100	Home

If someone bet on all 2,431 games in the 2018 season risking $243,100 (which I would advise against) the Logistic Regression predictions would result in a loss of $29,800 (ROI of -12.25%) while the Random Forest would result in a loss of $28,600 (ROI of -11.76%). So losses are minimized with the Random Forest model.

Another approach would be to consider ensembling both models. One simple method of doing so, would be to only bet on games in which both models have the

same predicted outcome (either both predict the home team win or both predict the home team loses). If a client bet on all such games in the 2018 season, they would have bet on 1,917 games with 60.98% accuracy and would lose $16,350 and the ROI would increase to -8.53%.

I. Discussion of Results

Now that we have a Logistic Regression model and a Random Forest model, we can discuss the results. Below are confusion matrices for each model.

Confusion Matrix for the LogReg Model		PREDICTED RESULT	
		LOSS	WIN
TRUE RESULT	LOSS	677	471
	WIN	538	745

Confusion Matrix for the RF Model		PREDICTED RESULT	
		LOSS	WIN
TRUE RESULT	LOSS	479	669
	WIN	332	951

Here is a table describing which model is the better choice based on different metrics.

Test Metric	Chosen Model	Value of Metric	Business Interpretation
Accuracy	RF	58.82%	Accuracy is the number of correct predictions over the number of total predictions made. The more accurate, the better, and the more money you will make using the model to bet. However, accuracy could be misleading, for example if the model predicted that the home team won every game it would still have an accuracy of about 53%, but would not be a good model.
Precision	Either	0.59	Both models had average precision of 0.59. The RF model had 0.59 precision for both classes while the LR had 0.56 precision for class 0 and 0.61 precision for class 1. This means when the model predicted the visiting team won it was correct 56% of the time and when it predicted the home team won it was right 61% of the time.
Recall	RF	0.59	Even though the RF model has a higher average recall, the LR model's recall is very close at 0.58, and may make more sense. Choosing based on the model with higher avg recall may be misleading because the Random Forest's recall for class 0 (home loss) is 0.42 , while its recall for class 1 is 0.74, which means when the home team actually wins the model is correct 74% of the time, but when the visiting team actually wins it's only correct 42% . The LR model has recalls

			of 0.59 and 0.58 for class 0 and 1. In the case of baseball games, the recall should be similar for both classes because both are equally important (unlike when you are trying to detect cancer, for example).
F1 Score	LR	0.59	This is a good balance between precision and recall.
AUC	LR	0.6119	The higher the AUC, the better. A high AUC indicates that the ratio of true positive rate (recall) and false positive rate (predicting that the home team won when they actually lost) is higher.
Log Loss	LR	0.674	We want to minimize log loss. Log loss penalizes highly confident predictions that are wrong. Even if a client loses money they maybe more upset if a model predicts that the home team will win with 90% probability and they lose than if the model had predicted 52% for example, and the client knew it was a riskier bet. However, upsets do happen frequently in baseball.
Winnings/ ROI	RF	-$28,600/ -11.76%	People using services want to maximize winning or at least minimize losses.

III. Conclusions

A. Summary

In conclusion, I have gathered data from retrosheet.org and BaseballMusings.com and cleaned and explored these data. I have built two classification models, a cross-validated Logistic Regression model with class_weight = 'balanced' and a Random Forest model with hyper-parameter tuning, to estimate the probability of the home team winning a regular season game in the MLB. To build the models, I used May 1st , 2009 – the end of the 2017 regular season as the training set, and the 2018 season as the test set. I have also built a simple simulation to estimate how much money someone using the model-driven prediction service would win or lose if placing bets based on each of the models' predictions.

Both models are built using the same set of features. For the home team and the visiting team, I calculated the average of the each teams' previous 162 games. For features pertaining to the starting pitcher, I calculated the averages of the previous 30 starts for each pitcher. For AvgAttendance, AvgRunDiffAtHome_H, and I used the averages of the previous 81 game home games, and for AvgRunDiffOnRoad_V, I used the averages of the previous 81 away games of the visiting team. I then subtracted each visiting team stat from the corresponding home team stat.

Both models are built using the same set of 19 features, and this is the order in which the Random Forest model ranked in terms of feature importance:

1. AvgFIPnoConst_SPdiff

2. AvgIP_SPdiff
3. AvgRelFIPnoConst_diff
4. AvgSpread_diff = AvgRunDiffAtHome_H - AvgRunDiffOnRoad_V
5. AvgOBP_diff
6. AvgISO_diff
7. AvgAttendance
8. AvgPitchBABIP_diff
9. AvgEper9_diff
10. AvgAper9_diff
11. AvgReliefIP_diff
12. pctWminL_SPdiff
13. AvgWP_SPdiff - Diff in number of wild pitches thrown by the home teams starting pitcher for the game and that of the visiting team
14. AvgSB_diff
15. AvgDB_diff
16. AvgGDP_diff - Diff of avg number of times batters grounded into double play of home and away team
17. AvgReliefWP_diff – Diff in number of wild pitches thrown by the home teams relief pitchers and those of the visiting team
18. AvgSH_diff
19. AvgCS_diff

B. Future Work

Much work can be done in the future to improve the model, which I would have tried given more time and computing power.

Here is a list of some of the things that I would like to do in the future:

- One thing that I will try is selecting different features. I will try different values for the window sizes for rolling averages. For example, it may make sense to see how many innings relief pitchers have pitched over the past 10 games rather than 162 because if they have pitched a lot they may be more tired. I would also consider factors such as the distance the team had to travel to the game, the number of days since their last game, and the number of days since the starting pitchers previous start. It may also make sense to consider a teams past performance at a particular ballpark or against a particular team. It also may make sense to explicitly compare certain stats for the home and away team, while for others it may not make sense.

- I would also gather more data on previous seasons.

- I would also like to build different classification models, such as K-Nearest neighbors and Support Vector Machines to see if they perform better. I would

also like to ensemble more models, taking averages or majority votes of several models to make a prediction.

- One thing I noticed about the Logistic Regression model was that it was performing better on the test set than the training set. Test precision, recall, and F1 score were 0.03, 0.02, and 0.03 higher than their training counterparts, respectively, and test accuracy was about 2.5% higher than test accuracy. This could be because the 2018 season happened to be a particularly "linear" season with more predictable results. To test this hypothesis I changed the test set to July 2018-the end of the season, and used the rest of the data and all test metrics improved (precision was 0.6, recall 0.59, F1 0.59, test accuracy 0.594, and AUC 0.6213). This supports my hypothesis and suggests that the more games in the 2018 season used to train the model, the better. However, to further explore this I would like to make a plot where:
  Train_0 = all games before 2018 season
  Test_0 = 2018 season
  Train_i = all games before 2018 season + first i% of 2018 season
  Test_i = the whole 2018 season – first i% of the season

  At the time of this writing, the game logs for the 2019 season are not yet available on Retrosheet.org, but it would be interesting to train the model on 2009-2008 seasons and test on the 2019 season to see if the trend continues.

- In the future, I would also like to build a more complex simulation to determine how much a prospective client should charge for the offered services. I would also consider the predicted probabilities of the home team winning to determine clients' returns, rather than just whether or not they correctly predicted which team would win.

IV. Recommendations to the Client

If the 2020 season were to start next week with no time for further analysis or texting, my recommendation to the client would be the following:

- If the person is using the client's service is solely winning money based whether they correctly predicting the winner of the game, I would recommend that bet based on the predictions of the Random Forest model, as it has the highest accuracy and the losses are minimized. I would also suggest that they bet on games in which both the Logistic Regression and the Random Forest model predict the same outcome, because in the 1,917 games in which this was the case, it was 60.9% accurate.

- If the person using the client's service plans to bet more or less money based on the predicted probabilities, I would suggest that they use the results of the Logistic Regression model, because it has a slightly lower log loss.