ML – Capstone Project

Predicting the Results of Soccer Matches

I. Definition

Project Overview

Sports Analytics in the past 10-15 years has increasingly become a part of every sport as teams begin to make analytical, data-driven decisions rather than a conventional or instinctual feeling that coincides with traditional beliefs. As a result, statistics about games have become increasingly available to the average fan. With these statistics, we're looking to use machine learning to help predict the results of a soccer match.

Problem Statement

Predicting soccer matches is unique compared to other sports because soccer can have one out of three results, win, lose, or draw. The result of a draw happens very often in the sport where as with other sports if a draw is possible it happens very rarely. In the other top 4 sports in the US NBA and NHL games cannot end in a tie. There have only been 3 ties in the NFL since the 2008 season and in the MLB ties only occur due to weather or other extremely rare cases. Having ties as an additional result increases the complexity of creating a predictive model for soccer matches. In doing research on the topic I found this project by Felipe Hoffa and Jordan Tigani of Google during the 2014 World Cup. They looked to predict the winner of each match in the tournament and in their initial run of the data they don't train on results that end in a draw since 'they have less signal' so all of their matches end up with either of the two teams winning. Which on some levels invalidates lowers their accuracy percentage since they are assuming that the winner of the penalty kicks (deciding factor on who continues to the next match) is considered to be the winning result of the match when in actuality the match result is a draw.

Not only does soccer have an extra result that makes predicting matches difficult, it's also a difficult sport to return statistical analysis on because it lack statistical history outside of standard stats and because of it's non-stop, free flowing nature. Other sports such as baseball naturally has more stats to utilize since box scores have been published for decades now and these stats can break a game down to the pitch. Also, due to licensing terms of the data on Sportradar I was only able to pull a minimal amount of games and it's data. There may be more data that they provide for a paid version but I'

Having a minimal amount of stats and having one more outcome to predict makes predicting soccer matches more difficult than other sports. A combination of approaches might have to be taken as we explore the data and begin to break the data down to what is needed.

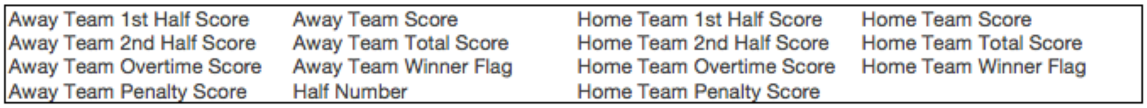
Metrics

Teams can win, draw, or lose a soccer match meaning they can earn 3, 1, or 0 points respectively. This will be the target or label of the dataset. We are going to test a variety of models but initial assumption is that accuracy will need to be determined based off of a combination of a few models and not just one. One model might not be able to clearly predict wins/losses and ties (as the project above suggested) so we'll need to identify and determine which matches might end in a draw and which matches will clearly have an outright winner.

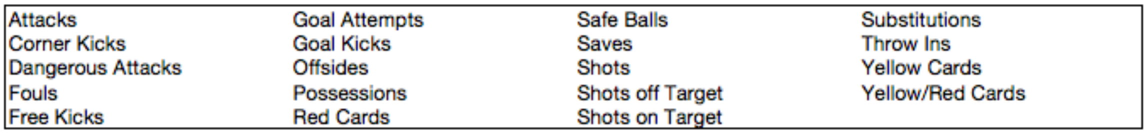
II. Analysis

Data Exploration

Using data pulled in from SportRadar's API I was able to pull Boxscore Information and Team Match Statistics in matches from MLS, Premiere League, La Liga, Ligue 1, and the Bundesliga in the current 2016 season.



Away Team 1st Half Score
Away Team 2nd Half Score
Away Team Overtime Score
Away Team Penalty Score
Away Team Score
Away Team Total Score
Away Team Winner Flag
Half Number
Home Team 1st Half Score
Home Team 2nd Half Score
Home Team Overtime Score
Home Team Penalty Score
Home Team Score
Home Team Total Score
Home Team Winner Flag



Attacks
Corner Kicks
Dangerous Attacks
Fouls
Free Kicks
Goal Attempts
Goal Kicks
Offsides
Possessions
Red Cards
Safe Balls
Saves
Shots
Shots off Target
Shots on Target
Substitutions
Throw Ins
Yellow Cards
Yellow/Red Cards

From this data, I'm able to pull in 227 matches from the different leagues with the majority being from the MLS since the European leagues just begin about a month ago. Our dataset size however will be double the number of matches since we are using a 'Current Team' vs. an 'Opponent Team' format for each match (essentially numbers mainly focusing on the Current Team's attributes). We can then flip the teams so the Opponent Team is now the Current Team and the Current Team is now the Opponent Team and have different numbers for the same match. We can use this technique to validate the predictions since the results for one team should equal out the results for the other team. If one team wins, the other team should lose, etc.

With this data, we're trying to predict upcoming matches based on data from the 3 previous matches that the two teams have played. One limitation from the dataset we have is that we technically don't have all the recent games played by the team. The data we have for teams is limited to league games. So any tournament that a team may play in during the season will not be counted in this dataset. This occurred due to the limitation that SportsRadar enforced on the trial version of their API. Also, some features inputted into the database weren't filled 100% and had some missing data. But since averages are taken on the previous games of a team, we use numpy to ignore that 'null' space and not factor in that game for that feature. Also note that because our model uses 3 previous game stats to determine the outcome of the current game, we need 3 previous weeks data so therefore the matches start at week 4.

There are some adjustments that could be made in the future to the data that could help balance out 'blowout' games. For instance, you could lessen the weight of 'Goals Scored' if the margin is greater than 2 goals. Since this means the other team likely isn't trying as hard the goal isn't as significant as a goal to put a team ahead.

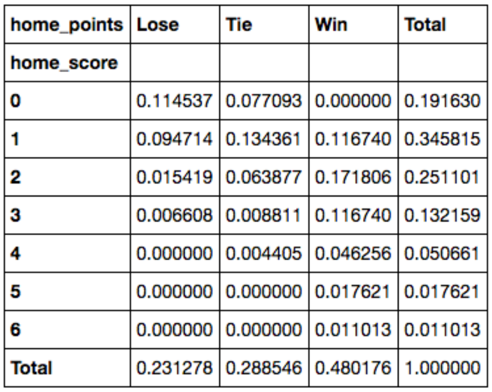
With this data from the API, I looked to modify and enhance the given statistics into relevant features that can help predict the result of an upcoming soccer match. Out of all the data given, I tried to focus on the features that have the most visible impact on the game or at least with other features. The list of features imported from SportRadar's API: is_home, current_formation, avg_points, avg_goals_for, avg_goals_against, margin, goal_diff, goal_effeciency, win_percentage, sos, rpi, opp_avg_points, opp_avg_goals, opp_margin, opp_goal_efficiency, opp_win_percentage, opp_sos, opp_rpi. Some of the other main features that helped to specifically describe the teams in the match are possession, attacks, dangerous, attacks, yellow_cards, corner_kicks, shots_on_target, shots_total, ball_safe, goal_attempts, saves, first_half_goals, sec_half_goals, and goal_kicks. We also have the same features that are the 'opponents' but applied to the previous opponents of the current team. We also have some calculated features using the data. We have goal_efficiency, which is the ratio of shots_on_target compared to the goals scored. We also have other ratios in 'goals_op_ratio', 'ball_safe_op_ratio', and 'goal_attempts_op_ratio', which compare the current teams stats to their previous opponents stats. We also have 'sos' and 'rpi' for both the current and the opponents.

These stats should help in determining where the current team stands in compared to the opponent that is playing them.

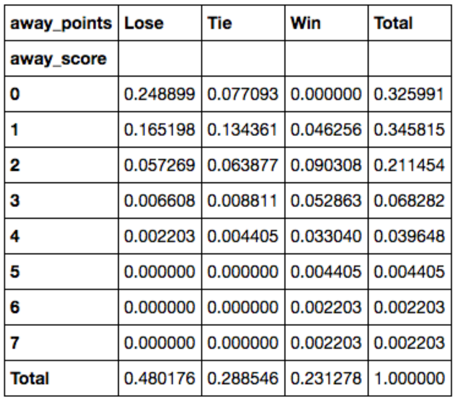
$$\text{SOS} = \frac{2(\text{OR}) + (\text{OOR})}{3}$$

RPI = (Winning Percentage + (2 x Average of Opponents' Winning Percentages Against Other Teams) + Average of Opponents' Opponents' Winning Percentages)/4

Upon first thinking about exploring the data (all stats in Stats.pynb) and how to pick the results of the match we should first see what the current proportions are between teams who win, lose, or draw. The below images show a breakdown of the score of both Home/Away teams and the result of the match. Some interesting points to note is that away teams who do not score any points have lost 25% of the games. Where on the other hand home teams who have not scored have lost only 11%. Also, 67% of the games the away team has not scored or has only scored one goal. Only 5% of the total games have the away team won when scoring 1 goal in a match. For the home team, 80% of their games they have scored at least 1 goal and when scoring at least 1 goal the home team has only lost 9.5% of the time.



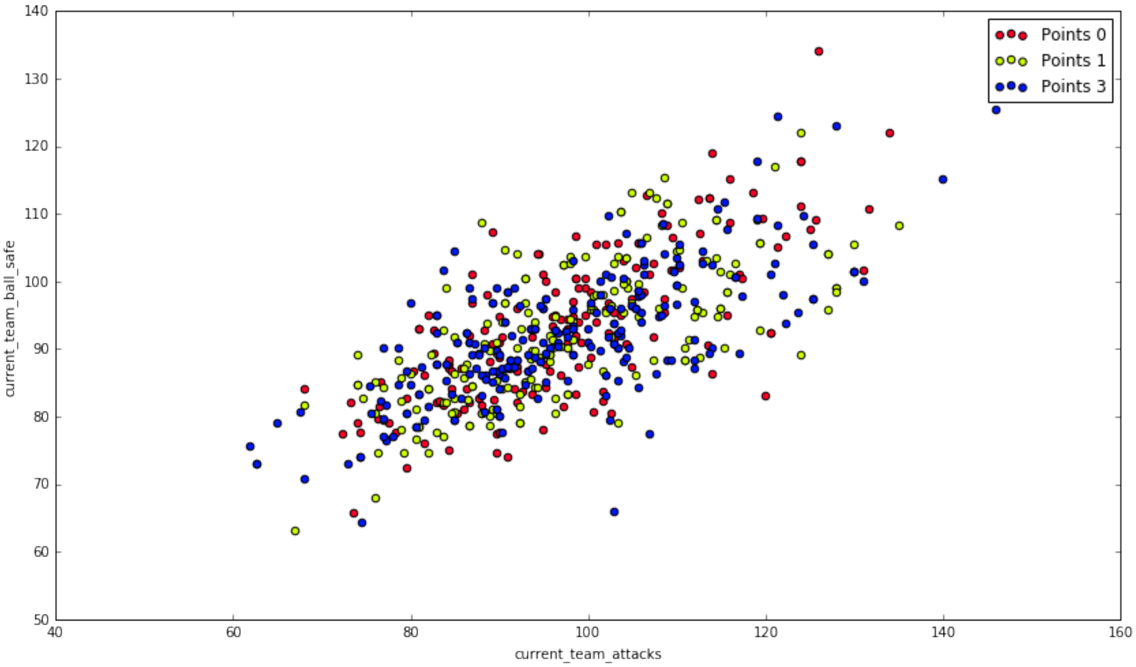
home_points	Lose	Tie	Win	Total
home_score				
0	0.114537	0.077093	0.000000	0.191630
1	0.094714	0.134361	0.116740	0.345815
2	0.015419	0.063877	0.171806	0.251101
3	0.006608	0.008811	0.116740	0.132159
4	0.000000	0.004405	0.046256	0.050661
5	0.000000	0.000000	0.017621	0.017621
6	0.000000	0.000000	0.011013	0.011013
Total	0.231278	0.288546	0.480176	1.000000



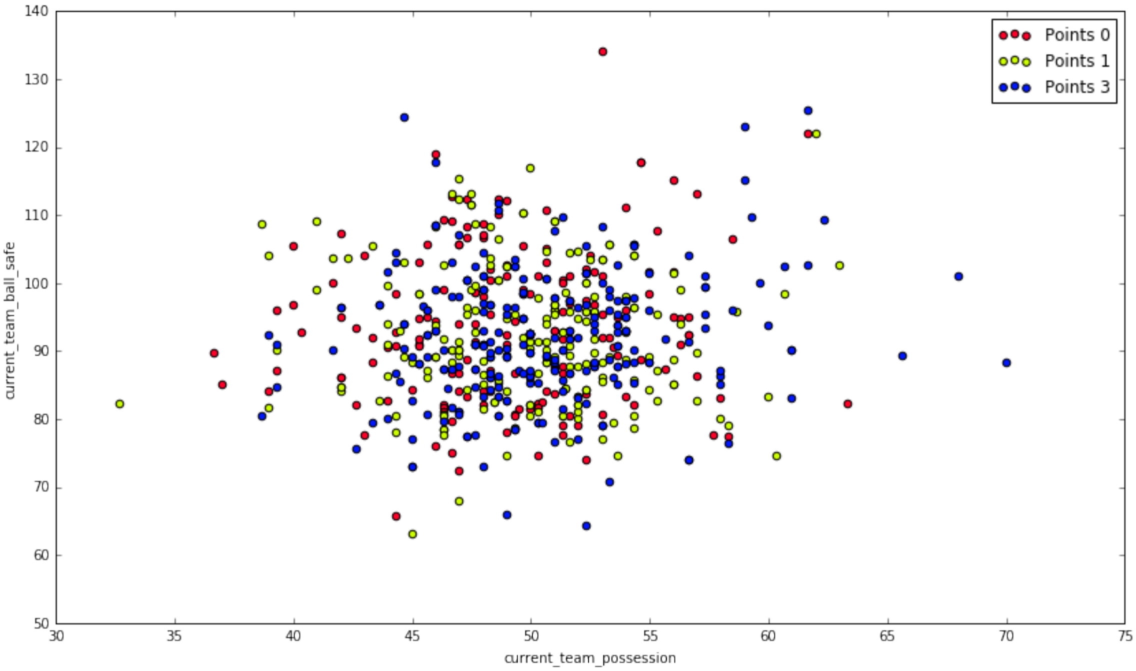
away_points	Lose	Tie	Win	Total
away_score				
0	0.248899	0.077093	0.000000	0.325991
1	0.165198	0.134361	0.046256	0.345815
2	0.057269	0.063877	0.090308	0.211454
3	0.006608	0.008811	0.052863	0.068282
4	0.002203	0.004405	0.033040	0.039648
5	0.000000	0.000000	0.004405	0.004405
6	0.000000	0.000000	0.002203	0.002203
7	0.000000	0.000000	0.002203	0.002203
Total	0.480176	0.288546	0.231278	1.000000

These observations show a huge importance on a couple of the major features when determining the outcome of a match. Home field advantage plays a major part in the results and obviously the amount of goals scored. From this we can start to break down what features have an influence or correlation on the amount of goals scored for a team in a match and even what features have influence on the amount of goals scored against a team.

The most prominent relationship when viewing the features is the relationship between ‘Ball_Safe’ and ‘Attacks’. Obviously there should be some direct relationships between some of the features (‘possession’, ‘ball_safe’, ‘attacks’, ‘dangerous_attacks’, ‘goal_attempts’, ‘shots_on_target’) such as ‘attacks’ and ‘dangerous attacks’ and also ‘goal_attempts and ‘shots_on_target’. But ‘Ball Safe’, which SportRadar defines as ‘a ball controlled by a team on their end of the field’, influences ‘Attacks’ which consists of a team playing the ball in the offensive third (opposite side) of the field. I’m assuming the reasoning behind this is in order to start an attack a team must first safely have possession of the ball and transition the ball over into the offensive third of the field. Essentially this could be a ‘conversion’ stat expressing when a team moves from it’s half of the field to the opponents half.

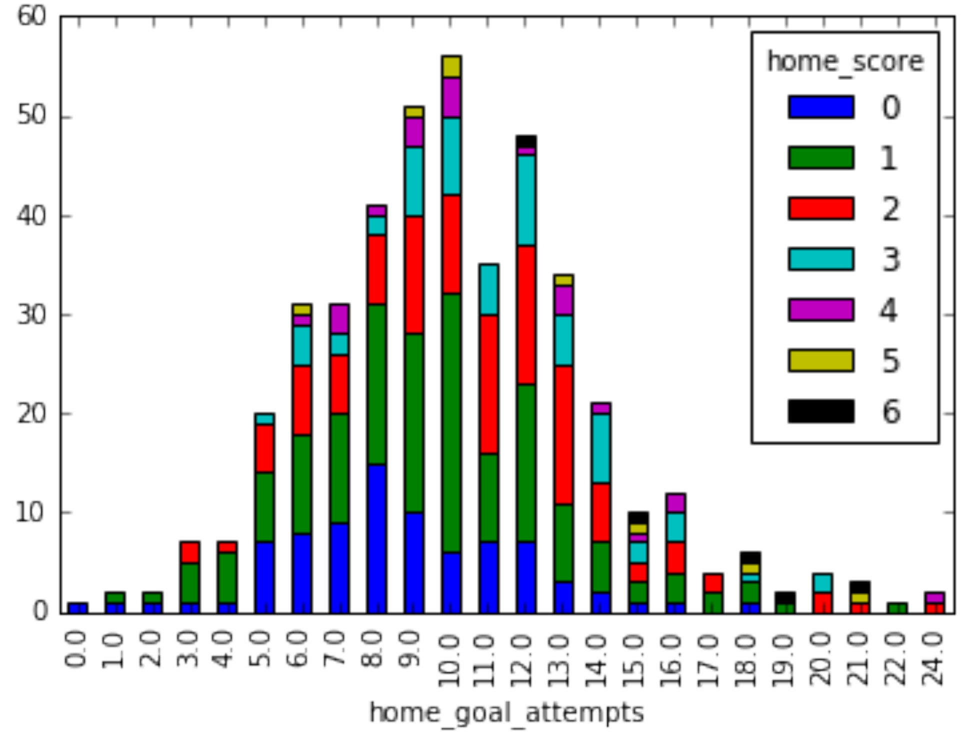


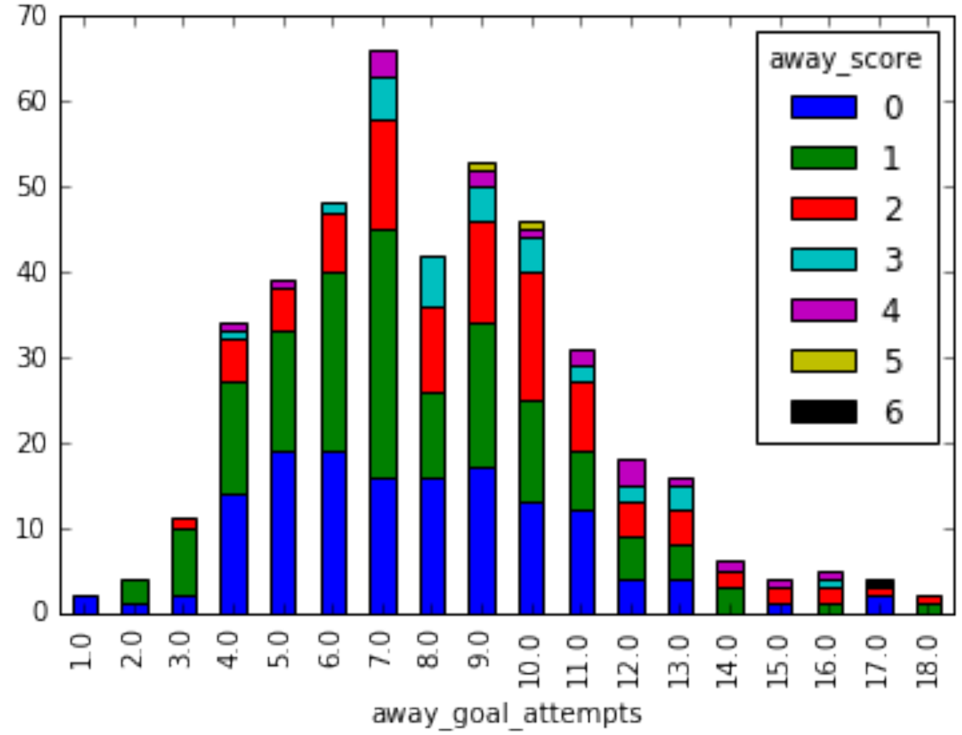
If this theory were to hold true you would think that there would be a strong relationship between the Ball_Safe and Possession features but there is very little if any relationship at all. This also could have to do with how they are calculating possession. There are different methods used in calculating as describe in this article. If SportsRadar were using the timing method then it would make sense as to why it has very little relationship with Possession. Possession has been 'the' stat for soccer. Essentially the thought is the longer a team holds possession of the ball the more they dominate a game and the higher chance they have to win but based on this subset of data Possession has a very loose relationship with Attacks, Dangerous Attacks, and Shots Total. It's not to say that these determine the outcome of the game but it's noteworthy in that Possession might not control all aspects of the game as previously thought.



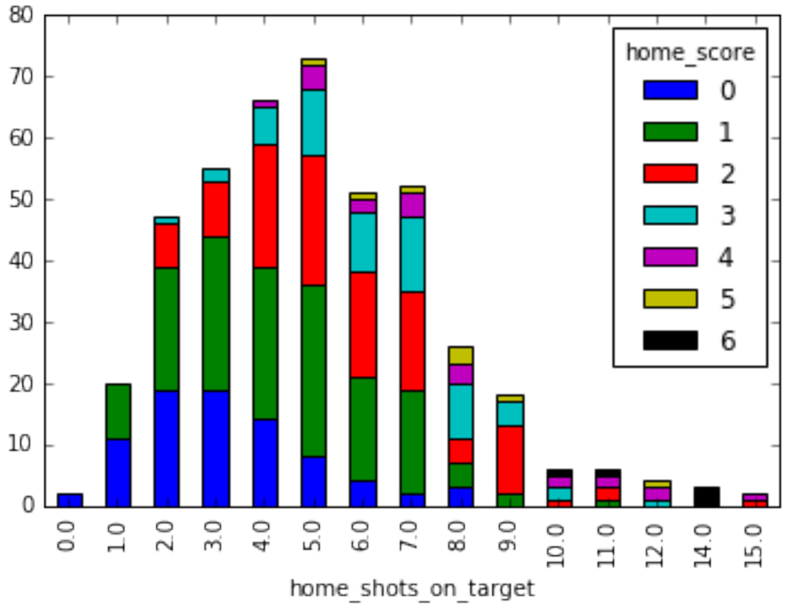
Comparing Goal Attempts and the Scores values we can see that in the majority of the games, home teams generally get more attempts on goal 8-12 (231/431 = 53.5%) where as the away teams shots are lower 6-10 (225/431 = 52.2%). Home games are more evenly distributed amongst the data compared to the away games where it's apparent there is a skew in the data to the left. This essentially supports the theory of home field advantage and the teams are 'weaker' when they play away.

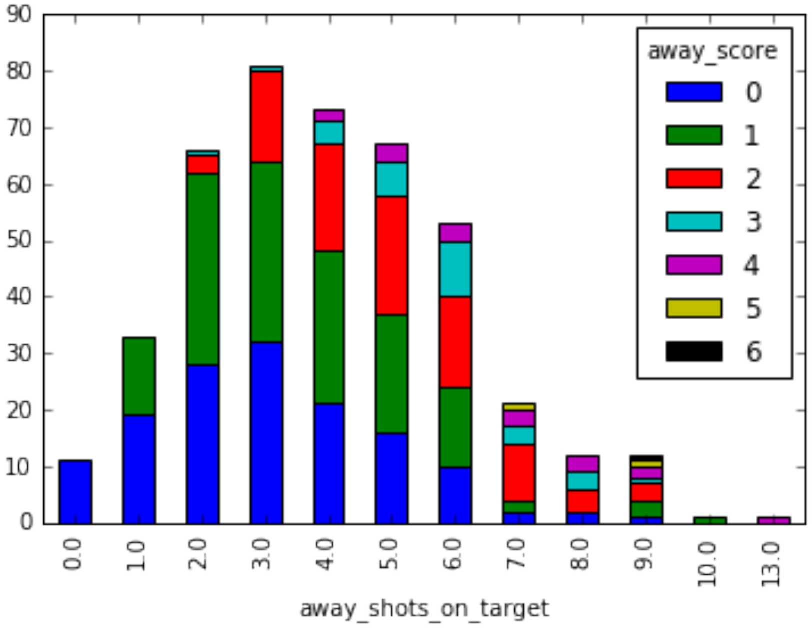
Also to note is that within these ranges, the home team 19.4% (45/231) of the time never score when they take these amount of shots where the away teams never score 31.7% (81/255) of the time. Which confirms the slight shift between the home and away data and also suggests a relationship (though weak it may be) between scoring and goal attempts.





Though overall Goal Attempts are slightly shifted, drilling down even more to Shots on Target show that ~80% (+/- 3%) of both Home and Away Teams get between 2-7 Shots on Target. There is no shift in the data as previously seen with Goal Attempts. At first thought one may assume this would weaken the relationship between Goal Attempts and Shots on Target but though it does show vulnerability to the relationship it doesn't account for bad shots. A player could have a good opportunity at a goal attempt and completely waste the opportunity with a poor shot. And though Shots on Target remain consistent between Home and Away teams, within the 2-7 Shots on Target range don't score any goals 20% of the time while Away teams don't score any goals 30% of the time. These numbers remain consistent with the Goal Attempts data we saw above.





III. Initial Hypothesis

After analyzing the data one can begin to understand the relationship between the features and how we can begin to determine the results of a game. There is a strong correlation between the features ‘Ball Safe’ and ‘Attacks’. The more ‘Attacks’ there are the more ‘Dangerous Attacks’ there will be. The more ‘Dangerous Attacks’ there are increases the likelihood of ‘Goal Attempts’. The more ‘Goal Attempts’ there are increases the chances of there being a high number of ‘Shots on Target’. And the more ‘Shots on Target’ there is lead to more ‘opportunities’ to score goals for a team.

Ball Safe → Attacks → Dangerous Attacks → Goal Attempts → Shots on Target → Goals

With the features that we have there seems to be a strong correlation between them and the numbers of goals a team scores in a match. And since the number of goals scored determines the outcome of a match we should first try to predict the numbers of goals scored by a team instead of first attempting to predict the results of a match.

Once we have determined our model and are satisfied with the predictions, we can then either input the goal predictions (expected score) in the final classification model to help predict the result of the match or we can somehow combine both predictions to help narrow down the results of the match. Which method we choose depends on the accuracy of the goal predictions model. If we can get a fairly high accuracy in the goal predictions model (70-75%) inputting the goal scored as a feature may be an option but anything less and the model only helps us to describe the matches.

With Soccer scoring does not happen very often if at all. Instead of using each Goal as a classifier (0-6), I decided to break down the match to a team having either a low scoring game or a high scoring game, basically if a team scores 0-1 goals or 2 or more goals. Home teams score 0-1 times in 54% of their games whereas Away teams score 0-1 goals 67% of the time. As seen from the stats previously discussed, determining just this information can help us predict the outcome of the match. For instance if an Away Team scores in the 0-1 goal category they only have a 5% chance of winning and a 42% chance of losing.

IV. Methodology

Admittedly, the majority of time of this project has been the repetitive process of formulating the features based on the limited, relevant statistics (compared to other paid API’s) that are given and trying to optimize their input into the models predictions. However, after numerous iterations it does seem that there is a maximum amount that can be captured with the given data and because of this the results might not be as high as desired.

First we started with essentially all the data possible. I used all the data from all the previous games of a current team in a match and used not only the current team stats and the opponent stats; I also combined all the stats from the opponents of the current team and all the stats of the opponents of the opponent team. This combined for over 40+ features in my initial model and understandably became over fit on the upcoming matches.

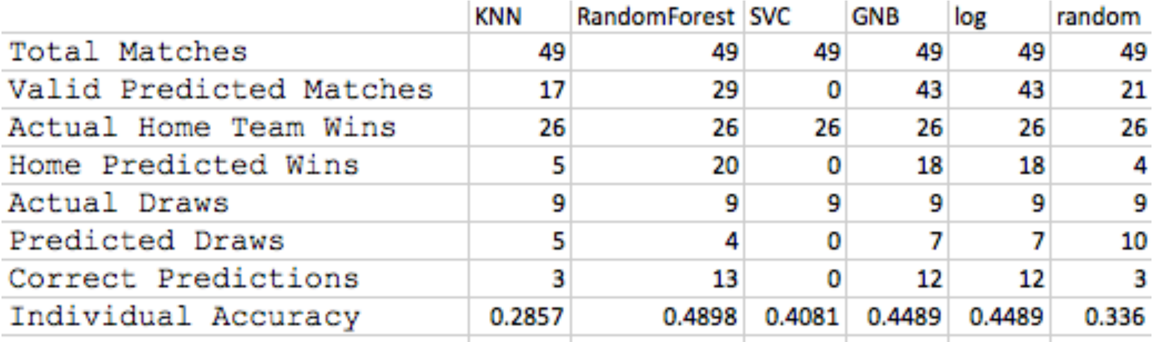
Slowly but surely I began to eliminate data group-by-group and trying to capture the data the best way possible (can be verified by all the logs in Git). I eliminated opponents of the opponent's stats and previous opponents of the current team's stats and replace simply with the RPI stat. Next instead of using both a current teams feature and the opponent's feature, I looked to combine them into one feature. I tried both using ratio and the difference of the stats squared. The latter seemed to work a bit better I'm assuming because of the distance in the numbers helps to clear the noise. To also help with compare the two teams in competition especially for the goals model, I created an offensive rank and a defensive ranking system that attempts to capture the goal efficiency (goal attempts over goals scored) of a team on both sides of the ball. For all the rankings, RPI, Offensive, and Defensive I ended up assigning each team to its appropriate quartile (0, .333, .666, 1) to help reduce the noise as there is not much distance between the top and bottom teams. Also, as previously mentioned some of the data pulled from SportsRadar was missing so in order to fill in the gaps for those games we essentially used the average from the other games in the season. By the final implementation, instead of using all the previous games from the current match, I'm only using the previous 3 games from the current match. This helps track how a team is currently doing rather than equally weighting the first game of the season and the their last game.

V. Results

To test our models I used 'upcoming matches' for the following week that hasn't been used by the models at all. This will give us a realistic result on how the model performs on unseen data. In the upcoming week of Soccer we have 49 matches coming up but with our structure it's really 98 different samples (remember we have Current Team vs. Opponent and then we reverse that to get the second teams perspective).

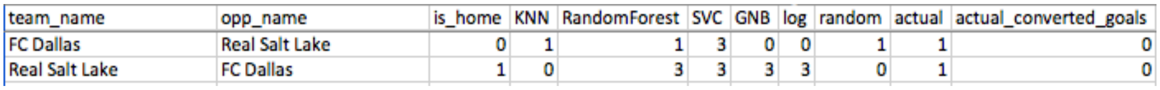
The best performing classifier out of the many that I tried was using the Logistic Regression model, which performed very similar to the other models initially (~60%) but when adjusted the parameter for the weight class to be 'balanced' it rose to ~65%. 65% accuracy I would label as an 'average' score for this prediction model. The baseline for this binary classifier would be 50% which is what we would expect to get if we picked each match randomly. The 15% difference is essentially picking 15 extra matches correctly. It's a significant improvement that can possibly help us describe a match but not enough to input into the points model to help predict the results of the match.

For the Points Prediction Model, interpretation of the results are much more difficult because of our setup. I ran a few different models and also used a random Series to help set the baseline and the results are interesting to say the least.



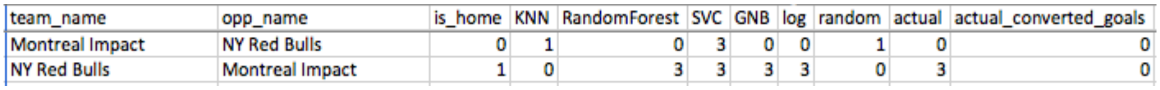
	KNN	RandomForest	SVC	GNB	log	random
Total Matches	49	49	49	49	49	49
Valid Predicted Matches	17	29	0	43	43	21
Actual Home Team Wins	26	26	26	26	26	26
Home Predicted Wins	5	20	0	18	18	4
Actual Draws	9	9	9	9	9	9
Predicted Draws	5	4	0	7	7	10
Correct Predictions	3	13	0	12	12	3
Individual Accuracy	0.2857	0.4898	0.4081	0.4489	0.4489	0.336

Individual Accuracy cannot alone be used to determine the effectiveness of the model but it does help. Because the matches are paired the model could get one of the two correct but the second of the pair could be wrong, misleading the prediction. However, RandomForest, GNB, and log all performed better than if we were to pick the results at random.



team_name	opp_name	is_home	KNN	RandomForest	SVC	GNB	log	random	actual	actual_converted_goals
FC Dallas	Real Salt Lake	0	1	1	3	0	0	1	1	0
Real Salt Lake	FC Dallas	1	0	3	3	3	3	0	1	0

Or the model can have both pairs correct leading the user to believe in a high confidence that the model picked a correct outcome for both independently.



team_name	opp_name	is_home	KNN	RandomForest	SVC	GNB	log	random	actual	actual_converted_goals
Montreal Impact	NY Red Bulls	0	1	0	3	0	0	1	0	0
NY Red Bulls	Montreal Impact	1	0	3	3	3	3	0	3	0

Also to note is to see how the models performed in making valid predictions. In an ideal world a model should make all valid predictions for every match. GNB and log have more valid matches than RandomForest but still have less correct predictions. Though valid predictions may lead you to believe that the model is very confident in it's pick, it still only gets the correct prediction at most 50% of the time depending on the model.

Hard to interpret but information I still find insightful and I believe the model benefits from the twice as much data since the pair of samples in a match give different numbers, perspectives. Instead of reformulating all the data and combining everything into a one sample, one match set I believe we can simply remove one of the samples from the match in our current data and check the accuracy of those results for a reasonable baseline on what a one sample, one match dataset would produce. Many of the features wouldn't change so the results should be close, at worst the lowest that a new dataset could reach.

Removing one set of sample for the sample pairs led to 44.8% accuracy in guessing the correct outcome of a match. Shifting by one and removing the other samples let to 53% correctness in accuracy. If we look at our random samples for those sets they come out to 38.7%and 32.6% respectively.

VI. Conclusion

We have 2 models that attempt to predict different results of a soccer match (both predictions can be seen in their respective 'prediction.csv'). One attempts to predict the amount of goals scored and the other attempts to predict the end result of the game. Both succeed at predicting their respective results better than picking the results at random by at least 10-15%. That improvement is significant enough but I feel there is room for improvement on both models. First step would be improved data as we can only do so much with what we are given. Our data gives a decent description of a game but there is so much more out there that can be filtered and used to help describe the games and even break down how teams play. Another easy step would be to remove Draws from of our model. Before I started on this project I always looked at ties as two teams being even in skill and score. But now, I see Draws as really two teams playing a game that runs out of time. If they were allowed to continue to play, eventually one team would break that Draw. And that could be a method to determine a draw. Find the predicted winner of each match and from that determine which games might be draws instead of trying to find all results at once. Other improvements/additions include gradually weighting the games so that the most recent games have more weight. Adding all games instead of just leagues to the models. Even adding players to the matches would help describe the game even more detailed.