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

Soccer is becoming increasingly popular in the United States. In major cities, such as San Francisco where I live pick up soccer games happen every night of the week including players of all ages and nationalities. I myself participate on two teams in two different leagues. More games are being nationally televised in the US and Major League Soccer is adding 4 more teams (3 cities) in the next 3 years. With the increased interest and popularity, Soccer has become fun to watch and a sport myself and a lot of other people are curious to understand more about.

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 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. And due to licensing terms of the data on Sportradar I was only able to pull a

minimal amount of games and it's data. Once I pulled the data down using SportRadar's API, I saved the data into a MySQL database and then used Python to output the data as needed (which can be seen in the 'src' folder as 'raw_data.csv').

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

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. In one model we'll try to predict the result of a match for one of the two teams. A team 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. As you'll see in the Exploratory Visual section below that in our dataset Draw's happen 28.8% of the time. That means that wins and losses split the remaining difference, 71.2%, which is close enough for me to weight each label evenly. So I feel we could just use AUC (correct predictions divided by the total number of predictions made) but it's also arguable that it is far enough away from the even split so to be sure we'll use the F1 score as a metric for success.

The F1 score is calculated using the precision and the recall. The precision is considered the exactness or quality of a prediction and the recall is the measure of completeness or quality. For our 'points' model, precision is the number of correct results for a label (let's say '3' for this example) divided by the total number of predictions made for the same label, 3. Recall equals the number of correct results for label 3 divided by the total number of matches that ended with the label 3.

For our second model, we will try to predict how many goals a team scores in a match. But to make this easier, we will separate goal scoring into 2 categories (if a team scores between 0-1 or if they score 2 or more) turning the model into a binary classifier. Looking at the data in Exploratory Visual again we can see that there are more percentage of matches that end in the 0-1 scoring range than end in the 1-2. Because of this, we will also use the F1 score as a metric for this model.

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

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

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, 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. If you look at the 'raw_data.csv' you can see what statistics are exported from the Database.

Features

- match_id - unique id for a match
- team_id – unique id for a team
- team_name – name of team
- opp_id – team_id of the opponent
- opp_name – name of opponent
- scheduled – date of match scheduled
- round – round number that the league is presently playing
- games_played – number of games played in calculating those stats for that round
- is_home – whether or not the current team is home

- current_record – winning percentage of current team for past games played
- opp_record - winning percentage of opponent team for past games played
- goals_for – sum of number of goals current team has scored in games played
- opp_goals_for – sum of number of goals opponent team has scored in games played
- goals_against – sum of number of goals current team has allowed in games played
- opp_goals_against – sum of number of goals opponent team has allowed in games played
- rpi - Winning Percentage + (2 x Average of Opponents' Winning Percentages Against Other Teams) + Average of Opponents' Opponents' Winning Percentages)/4
- current_team_possession – Sum of number of minutes current team has possession of the ball per match
- current_team_yellow_cards – sum of yellow cards the current team was given yellow cards
- current_team_goal_attempts – sum of Attempts the ball is played to the goal (essentially shots_total)
- current_team_dangerous_attacks – sum of attacks played in the current teams offensive third
- current_team_sec_half_goals – sum of goals scored in the second half for a team
- current_team_saves – sum of saves made by current team
- current_team_corner_kicks – sum of corner kicks
- current_team_ball_safe - sum of safe number of balls played by a team
- current_team_first_half_goals – sum of first half goals scored
- current_team_shots_on_target – sum of shots on target by current team
- current_team_attacks – sum of attacks made by current team
- current_team_goal_attempts_allowed – sum of goal attempts allowed by current team
- current_team_goal_kicks – sum of goal kicks taken by current team
- current_team_shots_total – sum of shot total by current team
- opp_team_* - all stats taken by current opponent (team that is opp_id)

Targets
- goals – number of goals the current team scored in that match
- points – number of points current team awarded in that match

Below are some screenshots with some summary statistics on some of the features (more can be seen in the stats.ipynb).

	match_id	team_id	opp_id	round	games_played	is_home	current_record	opp_record	goals_for	opp_goals_for
count	634.000000	634.000000	634.000000	634.000000	634.0	634.000000	634.000000	634.000000	634.000000	634.000000
mean	328.523659	42.198738	42.198738	13.599369	3.0	0.500000	0.500743	0.500743	4.116719	4.116719
std	205.832299	26.165018	26.165018	7.795190	0.0	0.500395	0.227672	0.227672	2.128332	2.128332
min	28.000000	21.000000	21.000000	4.000000	3.0	0.000000	0.000000	0.000000	0.000000	0.000000
25%	258.000000	27.000000	27.000000	6.000000	3.0	0.000000	0.339286	0.339286	3.000000	3.000000
50%	337.000000	33.000000	33.000000	12.000000	3.0	0.500000	0.500000	0.500000	4.000000	4.000000
75%	434.000000	40.000000	40.000000	21.000000	3.0	1.000000	0.642857	0.642857	5.000000	5.000000
max	893.000000	120.000000	120.000000	27.000000	3.0	1.000000	1.000000	1.000000	12.000000	12.000000

goals_against	opp_goals_against	rpi	goals	points	current_team_possession	current_team_yellow_cards
634.000000	634.000000	634.000000	634.000000	634.000000	634.000000	634.000000
4.105678	4.105678	0.500409	1.400631	1.340694	149.989011	5.037736
1.919353	1.919353	0.102546	1.205726	1.260914	12.937566	2.189004
0.000000	0.000000	0.212798	0.000000	0.000000	98.000000	0.000000
3.000000	3.000000	0.431136	1.000000	0.000000	143.000000	4.000000
4.000000	4.000000	0.504501	1.000000	1.000000	149.989011	5.000000
5.000000	5.000000	0.565580	2.000000	3.000000	157.000000	6.000000
12.000000	12.000000	0.753571	7.000000	3.000000	210.000000	13.000000

Exploratory Visualization

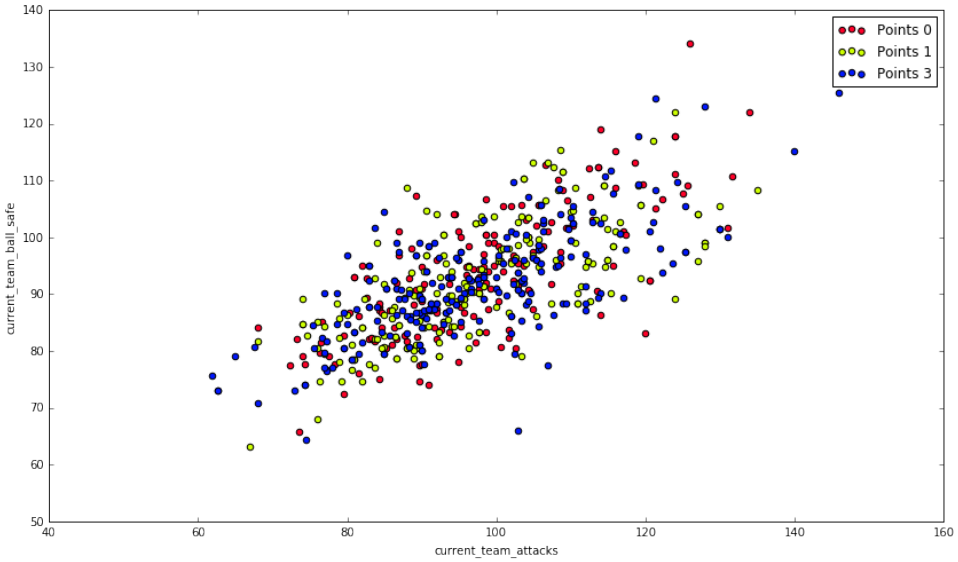
Upon first thinking about exploring the data (all stats in Stats.pynb) and how to pick the results of the match we should first see what the current proportions are between teams who win, lose, or draw. The below images show a breakdown of the score of both Home/Away teams and the result of the match. 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

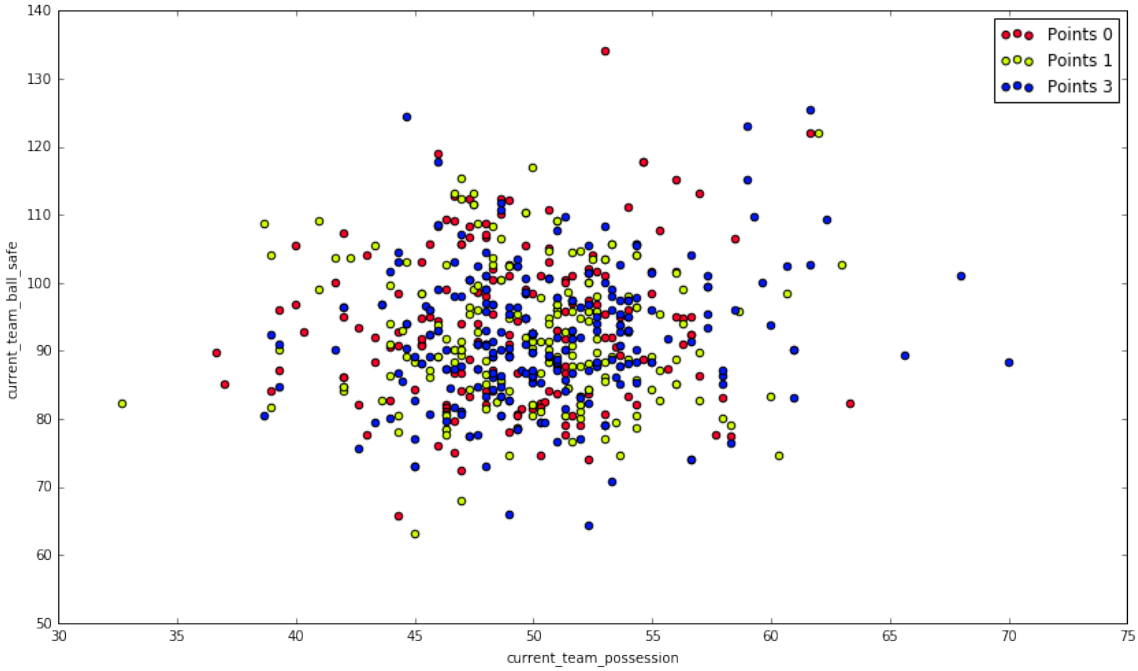
away_points	Lose	Tie	Win	Total
away_score				
0	0.248899	0.077093	0.000000	0.325991
1	0.165198	0.134361	0.046256	0.345815
2	0.057269	0.063877	0.090308	0.211454
3	0.006608	0.008811	0.052863	0.068282
4	0.002203	0.004405	0.033040	0.039648
5	0.000000	0.000000	0.004405	0.004405
6	0.000000	0.000000	0.002203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

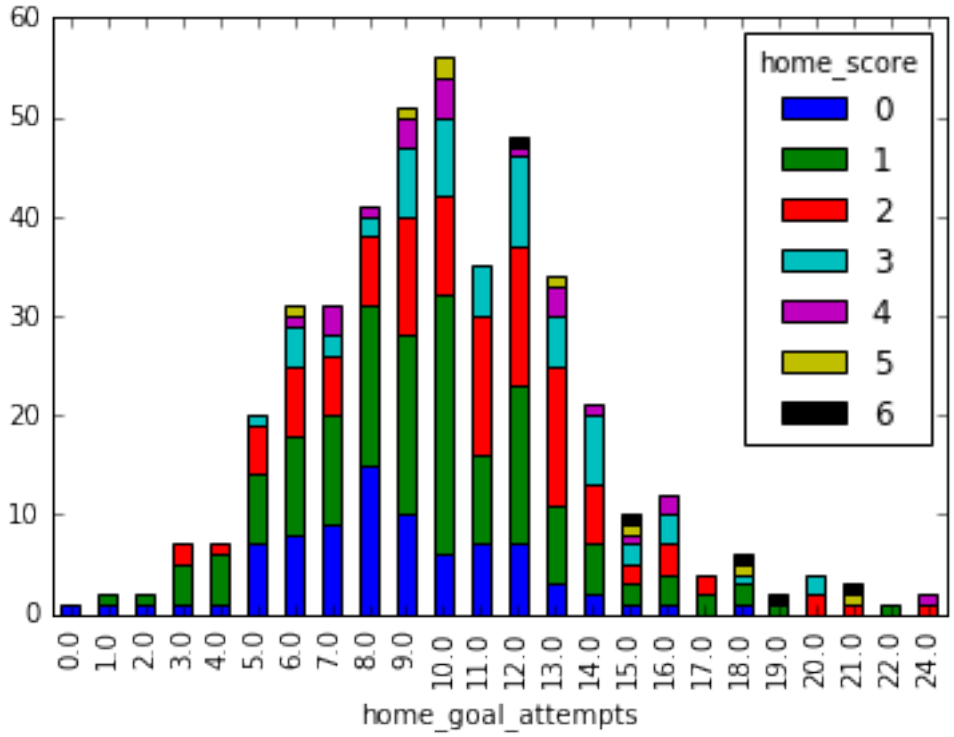


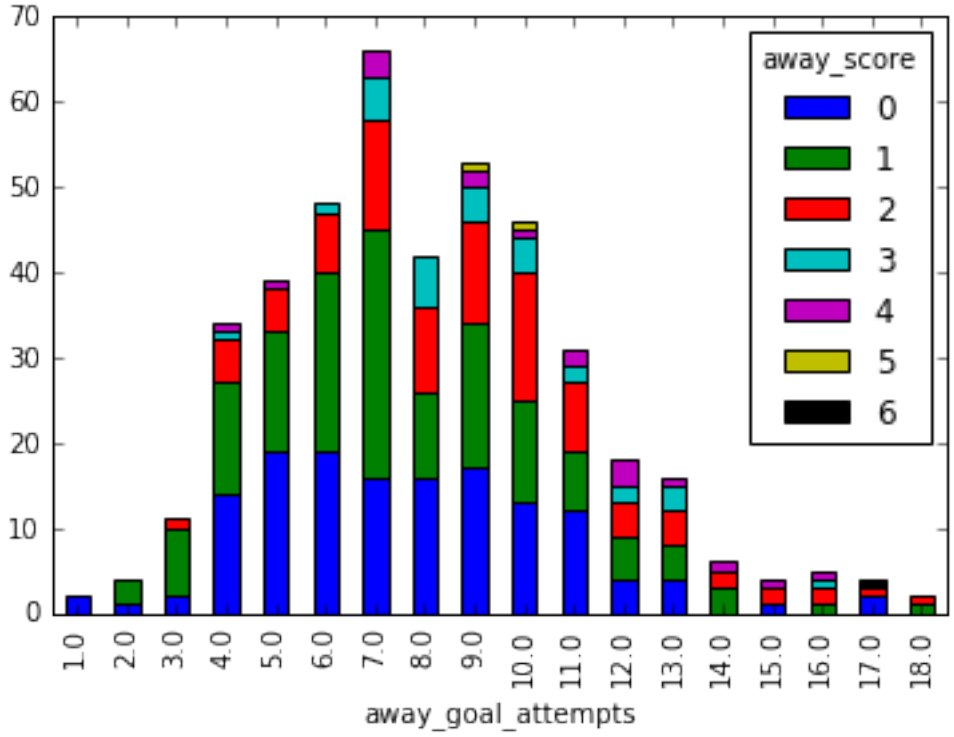
If this theory were to hold true you would think that there would be a strong relationship between the Ball_Safe and Possession features but there is very little if any relationship at all. This also could have to do with how they are calculating possession. There are different methods used in calculating 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.

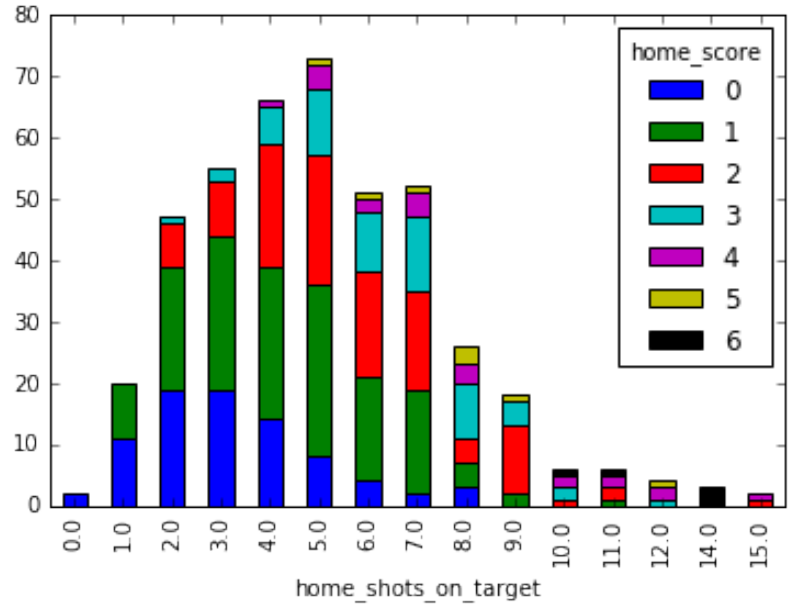
Also to note is that within these ranges, the home team 19.4% (45/231) of the time never score when they take these amount of shots where the away teams never score 31.7% (81/255) of the time. Which confirms the slight shift between the home and away data and also suggests a relationship (though weak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

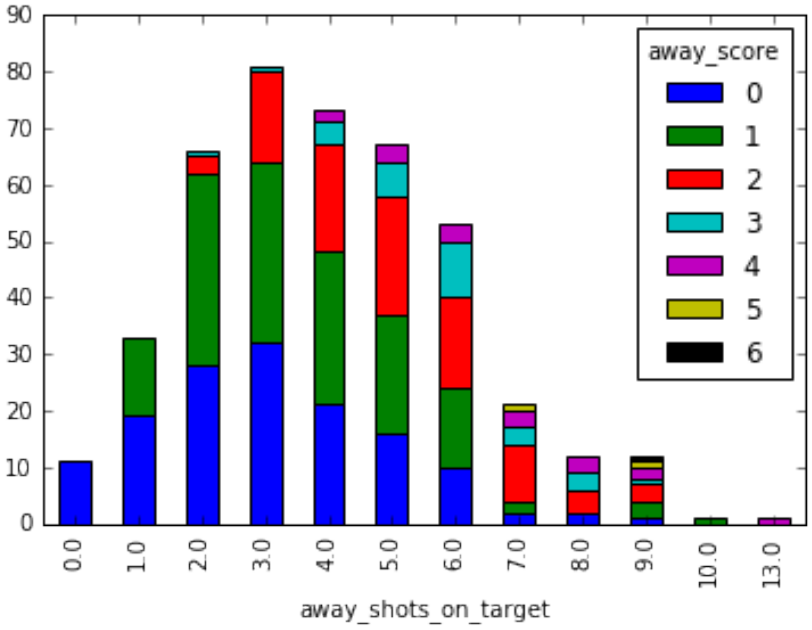




Though overall Goal Attempts are slightly shifted, drilling down even more to Shots on Target show that ~80% (+/- 3%) of both Home and Away Teams get between 2-7 Shots on Target. There is no shift in the data as previously seen with Goal Attempts.

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.





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

Algorithms and Techniques

There seems to be a strong correlation between the features 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 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.

As for the algorithms that we'll be testing with I'm trying a variety of classifiers that should help to describe the data. We're running initial tests using Logistic Regression, SVM, GNB, KNN, and Random Forest classifiers. As stated previously about our data some of the features can be dependent on each other therefore could help determine the result of the match or the number of goals scored by a team. Logistic Regression will help us determine how much this is true. If Logistic Regression has a successful time at predicting the outcomes then we can assume that our features are related and do depend on one another. Another advantage of Logistic Regression is that the result also gives us probabilities that we can use to rank each classifier. This helps in sports to give us an idea of the chances a team has to win, draw or lose. Also, SVM is similar to a LR in the fact that it tries to separate the data on a straight line using a linear kernel. Because of the noise in our data a non-linear kernel will need to be used to see if our labels are easily separated. We also have a bit more flexibility with SVMs in terms tuning the regularization parameters to help fit the model. But a disadvantage to using SVM is that it won't provide probabilities for each classifier in the output. A downside to both of these classifiers is they are expensive to train. Cross validation on the SVM took a while on our dataset. If we were to continue to expand our dataset then SVM and LR could become costly. I also wanted to try the Gaussian Naïve Bayes classifier as it assumes the features are independent and gives probabilistic value for each of the

labels. Also, GNB requires a huge data set as it uses probabilities based on existing samples. If a sample is not represented it could affect the posterior probability estimate. I'm not sure we have a big enough data set to confidently use GNB. Also not expecting much out of this classifier because I already believe that the some of the features are dependent on each other. The KNN classifier was used with the thought that the statistic summary of a match and similar matches around it should be a good indicator of the result of upcoming matches even if the labels of the matches aren't separable by a line. KNN does this by using the local data points surrounding the upcoming data point to obtain a prediction. The last classifier that I decided to try was using a Random Forest algorithm. Similar to KNN, the Random Forest algorithms takes a combination of individual tree predictors where each tree is close in distribution to the other trees and then outputs a class that is the average of all the classes outputted by all the trees. This helps overcome the problems of over fitting by the individual trees. This algorithm could be useful to our data as it can go through each feature decision systematically and compare upcoming matches to those same features and will give it a similar prediction.

Benchmark

Our goal is for the models to perform better than assigning random results to the upcoming matches. For the binary classifier where we are looking for the 'goals scored' by a team, if we are to assign a random result to each match up and then apply the F1 score function to this random series we get a score of 41.7%. If we assign a random series of results to our other 'points' model we get a score of 38.5%. These are our benchmarks for how well our predictive models perform.

III. Methodology

Data Preprocessing

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.

Implementation

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.

A persistent challenge through this entire process was querying Pandas Dataframes for conditional data. Being new to Python let alone Dataframes learning the concept of Dataframes and the power to query the DF's so quickly and easily it took awhile to get used to. Especially considering the amount of data needed to calculate every match for every league. Learning how to trim that down and make it as simple as possible was a challenge that took many iterations to improve upon.

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.

Refinement

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.

IV. 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).

Model Evaluation and Validation

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 Logistic Regression 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.

Justification

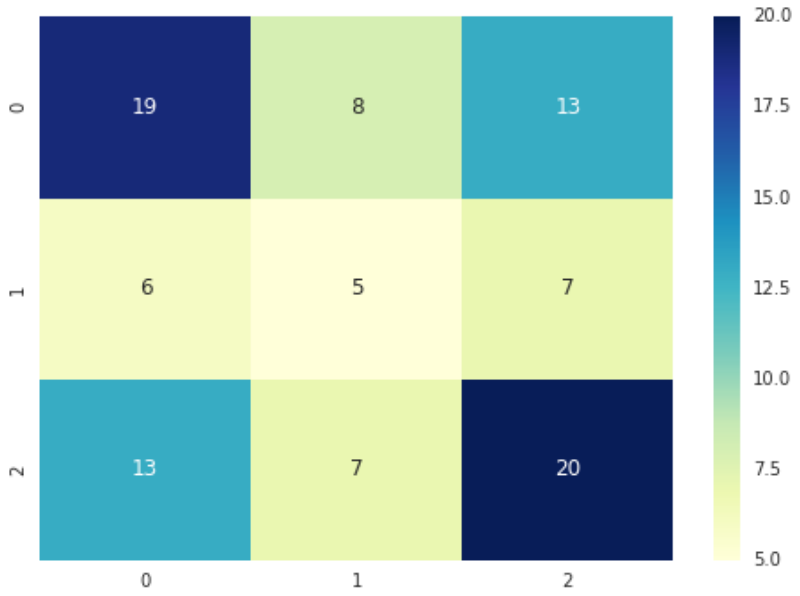
For the 'points' model, checking the models prediction with the F1 score method that we discussed in the notes the Logistic Regression classifier received a 54.4%. This is an improvement on the 38.5% benchmark we set against the Random numbers we used and though it's a significant increase it's still is not enough to predict a game with confidence. As for the 'goal' model, we again have both the Logistic Regression model and the GNB model with the highest F1 score of 51.9%. This is a much greater improvement on the 41.7% that we benchmarked earlier.

V. Conclusion

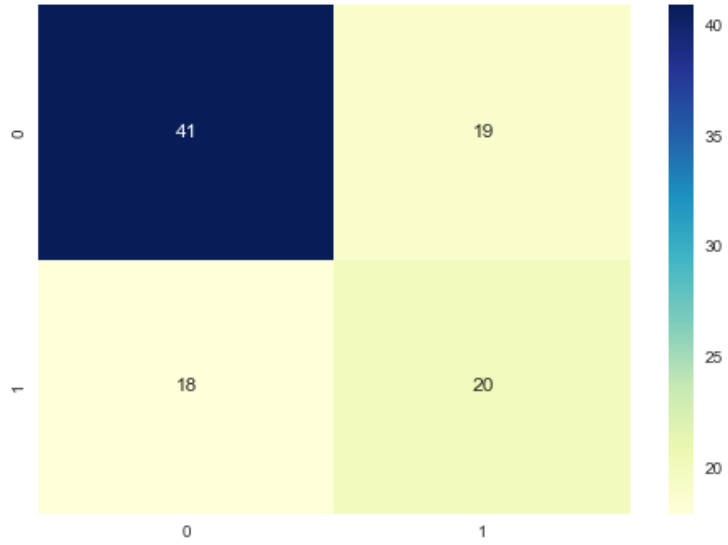
Reflection and Free-Form Visualization

In the beginning we started with an idea that we were going to be able to predict the final result of a soccer match. Who wins, loses, or draws and what the score of that match will be. Right now we have 2 models that attempt to predict those 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.

Below is a Confusion Matrix with the results of the 'Points' model. This graph is breaking out the Actual values from the Predicted values. The labels on the left are Actual Results and the labels on the bottom are Predicted. Out of the '0' labels, we predicted 19 teams losing correctly and assigned '1' to 8 of the predictions and 13 to the '3' class. Those numbers on that single row equals 40 teams losing total. Which easily flips to the bottom row since the numbers should match up. If 40 teams lose a match then 40 teams need to win a match. The middle row is number of ties, which is a total of 9 matches, 18 teams. We predicted 5 ties right but predicted falsely assigned 8 ties as losses and 8 teams as wins.



Similarly, below we have the Confusion Matrix for the 'Goals' method. We have 41 teams score predicted correctly scoring 0-1 goals but also predicted an extra 19 teams scoring 0-1 when they actually scored 2 or more. Also, we correctly predicted 20 teams scoring 2 or more goals but missed an extra 18 teams that scored in that same category.



Both models succeed at predicting their respective results better than picking the results at random. Because of this, we know that our models can start to form a description of our matches on very limited data. But we can also see how our models begin to lack information and are not able to fully summarize matches. Looking at the

predictions we can start to see how both the models make predictions in comparison to the home team advantage that we initially mentioned. For the 'points' model, it predicted the home team to win 8 more games than they actually did and predicted the away team to lose 11 more games. As for the 'goals' model, it actually predicted the away team would score more (2 or more) in more games but actually ended up scoring less. It also predicted the home team would score less (0-1) in more games but actually ended up having more teams with 2 or more goals. The 'goals' model actually seems to go against the home field advantage and generally makes them the lesser scoring team.

Points Results

log	0.0	1.0	3.0	Total
is_home				
0.0	37	3	9	49
1.0	4	6	39	49
Total	41	9	48	98

actual	0	1	3	Total
is_home				
0.0	26	9	14	49
1.0	14	9	26	49
Total	40	18	40	98

Goals Results

GNB	0.0	1.0	Total
is_home			
0.0	31	18	49
1.0	28	21	49
Total	59	39	98

actual_converted_goals	0	1	Total
is_home			
0.0	37	12	49
1.0	23	26	49
Total	60	38	98

Improvement

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.