Iteration 3

1. Final Training Results

My preliminary results gave me around 55% accuracy using the XG Boost model and every feature in my dataset except from the team name strings. I was trying to predict FTR which stands for full time result. I encountered the following problem: SVM and Logistic Regression both had the same performance.

SVC:

```
SVC accuracy: 0.69 (+/- 0.13)
              precision    recall  f1-score   support

          -1       0.67      0.80      0.73        96
           0       0.62      0.09      0.16        56
           1       0.70      0.85      0.77       137

   micro avg       0.69      0.69      0.69       289
   macro avg       0.67      0.58      0.55       289
weighted avg       0.68      0.69      0.64       289

[[ 77   2  17]
 [ 19   5  32]
 [ 19   1 117]]
```

Logistic Regression:

```
LogisticRegression accuracy: 0.69 (+/- 0.16)
              precision    recall  f1-score   support

          -1       0.70      0.79      0.74        96
           0       0.41      0.12      0.19        56
           1       0.71      0.84      0.77       137

   micro avg       0.69      0.69      0.69       289
   macro avg       0.60      0.59      0.57       289
weighted avg       0.65      0.69      0.65       289

[[ 76   3  17]
 [ 18   7  31]
 [ 15   7 115]]
```

DData preprocessing part 2

First of all, I converted every string in to numbers: 1 represents H which means home team won, 0 represents D which stands for draw and -1 represent A for away team victory. I further processed my data and combined features to give me more insightful information:

- Number of matches in total and per team
- Total away and home goals per team
- Total goal conceded at home and on away fixture
- Number of home and away games per team
- Average goal scored per game by team at home and away
- Average goal conceded per game by team at home and away
- Home win rate (shows that there is a bigger chance of winning for home team)
- Away win rate

4 distinct features were created using the following formula:

$$\frac{Single\ Team\ average\ performance}{Average\ performance\ of\ every\ team\ in\ the\ league}$$

- Attacking strength and defensive strength per team at home
- Attacking strength and defensive strength on away fixture

The attacking and the defensive strength of a team varies when it plays at home or on an away trip since we've confirmed that generally, a team performs better when it plays in his own stadium. Following theses modifications my model was able te get an accuracy of 70%.

The following confusion matrix was obtained from the optimized model:

```
LogisticRegression accuracy: 0.69 (+/- 0.13)
              precision    recall  f1-score   support

          -1       0.70      0.81      0.75        96
           0       0.42      0.14      0.21        56
           1       0.72      0.83      0.77       137

   micro avg       0.69      0.69      0.69       289
   macro avg       0.61      0.60      0.58       289
weighted avg       0.65      0.69      0.66       289

[[ 78   4  14]
 [ 18   8  30]
 [ 16   7 114]]
```

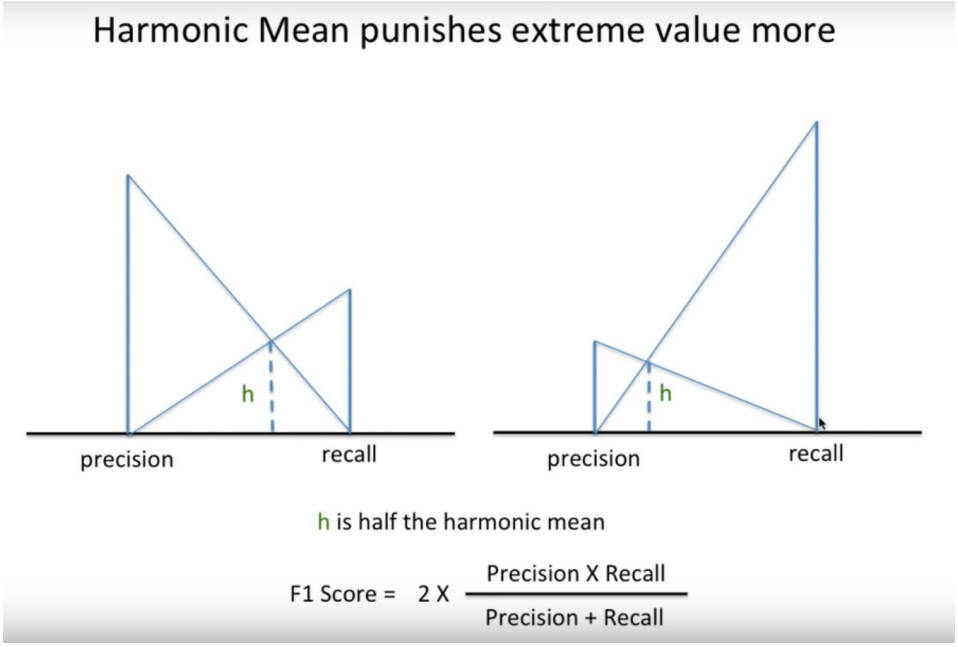
```
LogisticClassifier = LogisticRegression(solver = 'newton-cg', multi_class= 'multinomial', dual = False, C = 0.75, max_iter =3
```

```
SVC accuracy: 0.69 (+/- 0.14)
              precision    recall  f1-score   support

          -1       0.68      0.77      0.72        96
           0       0.42      0.09      0.15        56
           1       0.72      0.88      0.79       137

   micro avg       0.69      0.69      0.69       289
   macro avg       0.61      0.58      0.55       289
weighted avg       0.65      0.69      0.64       289

[[ 74   4  18]
 [ 22   5  29]
 [ 13   3 121]]
```

```
svm = SVC(gamma=0.01,C=10, decision_function_shape='ovo')
```

The accuracy score obtained by cross validation is 69% which means that 69% of the predictions from this model where right (total number of correct predictions / total number of data). F1 score is a good metric when data is imbalanced. The precision columns indicate how likely a prediction is correct. I decided to take as the logistic regression classifier, because it performs slightly better.

F1 Score is the harmonic mean of recall and precision:



Therefore, the F1 score can find an equal balance between the weights of the samples from each class on the accuracy. When the samples of each class are imbalanced i.e. in this case A=96, D=56, H=137, the F1 score is a better indicator of the model's performance since it takes in consideration this imbalance. **The definition of recall is the number of true positives divided by the number of true positives plus the number of false negatives. True positives are data point classified as positive by the model that actually are positive (meaning they are correct), and false negatives are data points the model identifies as negative that actually are positive (incorrect).** The false negatives are respectively 18 for class "-1", 48 for class "0" and 23 for class "1". An observation that explains the low F1-score at class "0" is that the number of "1" and "-1" that are actually 0 is fairly very high (false negative). On the other side, the model is performing relatively well for predicting away wins ("A") and home wins ("H") with a respective f1-score of 75% and 77% respectively. For hyper-parameters tuning, the Grid Search method was used which is implemented using sk-learn. Since 'multinomial' from the multi_class parameter has to be set, it was mandatory to set 'dual' to 'False' and use one of the following solvers that incorporate multinomial': 'newton-cg', 'sag','saga' and 'lbfgs'.

The cost function we are trying to minimize is:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\}\log\frac{\exp\left(\theta_j^T x^{(i)}\right)}{\sum_{1\le l\le k}\exp\left(\theta_l^T x^{(i)}\right)}\right] \quad [3]$$

We want to find the parameter theta that would give us the lowest gradient of this cost function. The following parameter were found using the GridSearch method:

```
Best: 0.692042 using {'C': 0.75, 'dual': False, 'max_iter': 3}
```

Theses change had minimal effect on the accuracy of my model which stayed at 69% with 4 additional correct predictions.

I have accomplished my initial goal which is to predict soccer matches with more than 2 correct out of 3 matches (>67%), but my model can still undergo optimization in a lot of areas. I still don't think that my model would make a better prediction than an expert analyst of the game which is my ultimate goal. In fact, I would need a lot more data to get a better profile of each team. In addition, it should incorporate sentiment analysis for example for data on twitter, previous statistics from all the previous years on every team and add the current form each team which could be done by getting the last five results of each team against another team.

Final demonstration proposal

I would like to make a web app. However, I don't have any experience in web development but I am planning to take classes on code academy before doing my first website.