Analysis of Upsets

Team Number 14

Members:

Siddharth Meenachi Sundaram

Pratyush Kumar Pandey

Rajarshi Basu

Archit Rungta

Problem statement: Exploring and analyzing upsets in football; Upsets are matches where underdog teams win the matches against well know and renowned teams

Probabilistic upsets:

Since the match schema contained data on betting odds, we decided to calculate implied probability using the formula:

$$prob_{win} = \frac{\frac{1}{odds_{win}}}{\frac{1}{odds_{win}} + \frac{1}{odds_{loss}} + \frac{1}{odd_{draw}}}$$

Please note we are considering the results with respective to the home team only. After calculating implied probabilities, we decided to retrieve match details for matches where the home team's probability of winning was less than a certain ***criterion***, but the team still emerged victorious.

We decided to do some exploratory analysis on 3 different values for the ***criterion: 0.05, 0.10, 0.20***

Initially, we decided just to do some analysis on the data of the most famous betting site: Bet365. After that we found the mean betting odds for the home team, away team and for a draw across all the betting site and analyzed that data as well.

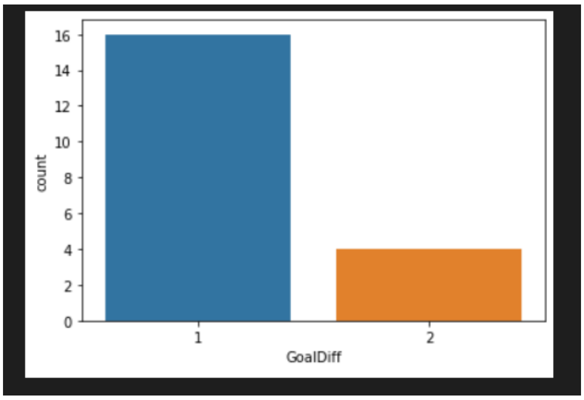
Bet365 EDA:

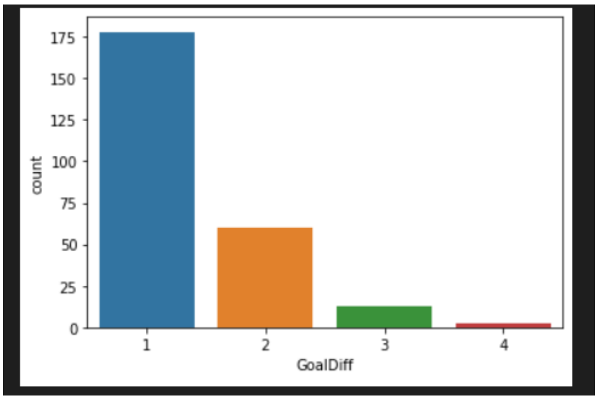
We first found the number of upsets according to the 3 values for the criterion:

```
[146]
print("15 percent:")
print(upsets_15['league_id'].count())
print("10 percent:")
print(upsets_10['league_id'].count())
print("20 percent:")
print(upsets_20['league_id'].count())

15 percent:
110
10 percent:
20
20 percent:
253
```

The results were what we expected as the betting sites allot betting odds after careful analysis and such matches are rare occasions. We further visualized plots of goal difference in such matches:

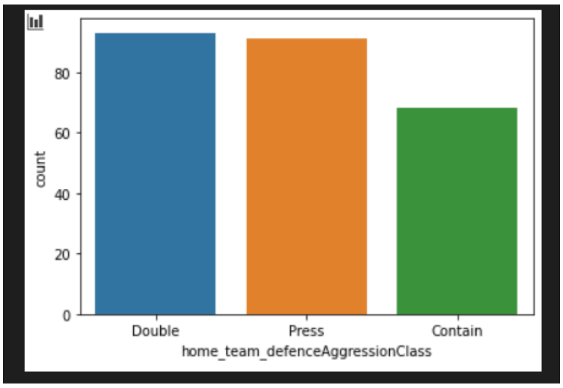


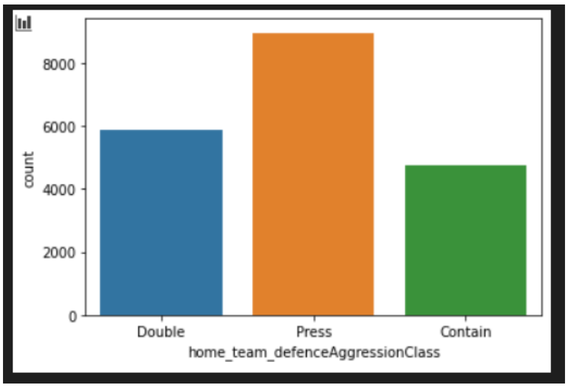


Count plot when criterion is 0.10 Count plot when criterion is 0.20

These results were also expected as such upsets are usually close matches with 1 as the goal margin. Again, when the criterion is increased to 20%, we witness 3 goal difference as well as 4 goal difference. This happens because when the criterion increases, we consider stronger underdog teams with higher probability to win.

Hypothesis: The winning teams in upsets or underdogs usually have very defensive strategy and tend to park the bus. This means most of the possession lies with the stronger teams and the weaker teams believe in counter attacking strategy. Hence, our next task was to explore any similarities in strategies/attributes amongst the winning teams in upset matches. We fix the criterion as 0.20 to consider more upsets





1. Upset dataset (b) All the matches

The aforementioned count plots demonstrate the number of winning teams adopting each defenceAggression strategies for the upsets and all the matches. In upsets, the winning teams tend to have a 'double' defence strategy. For all other team attributes or classes like defencePressure, the count plot for upsets resembles the count plots of all the matches. Therefore, we have ***no evidence to back our hypothesis.***

After this we decided to find if there are specific underdog teams which have been the reason for a lot of upsets:

Stoke city: 1019, Real Sociedad: 9560 as shown below

```
[78]
# finding the team with highest upsets

upsets_20['home_team_id'].value_counts()
```

```
10194    7
9761     6
8560     6
9904     5
10217    5
        ..
8466     1
10000    1
8462     1
9997     1
8406     1
Name: home_team_id, Length: 140, dtype: int64
```

Probable Reasons for upsets for stoke city: increase in buildupplay speed from 50 to 75 and increased chance creation passing from 30 to 59

	team_id	date	buildUpPlaySpeed	buildUpPlaySpeedClass
1242	10194	2/22/10 0:00	65	Balanced
1243	10194	2/22/11 0:00	50	Balanced
1244	10194	2/22/12 0:00	75	Fast
1245	10194	9/20/13 0:00	75	Fast
1246	10194	9/19/14 0:00	48	Balanced
1247	10194	9/10/2015 0:00	48	Balanced

Hypothesis: Real Sociedad is not an underdog team even then Bet365 allots very high odds for the team for matches against FC Barcelona though it emerged victorious 5 times in 5 consecutive seasons on their home ground against FC Barcelona. We hypothesize that Bet365 has a bias for teams which have a good brand name and reputation and also heavily considers popular opinion to the extent that affects their analysis.

In order to test out this hypothesis, we use a neural network to predict winners in all the matches. After this, our goal is to compare the probability of victory for the team predicted by the neural network against the implied probability calculated from Bet365 data.

Dataset Preparation:

For this task, we prepared the dataset by once again scanning through all the match records, and including all of the numerical and categorical attributes (listed below) for the home and away teams for every match record into the match database.

Attributes used:

- buildUpPlaySpeed
- buildUpPlaySpeedClass
- buildUpPlayDribblingClass
- buildUpPlayPassing
- buildUpPlayPassingClass
- buildUpPlayPositioningClass
- chanceCreationPassing
- chanceCreationPassingClass
- chanceCreationCrossing
- chanceCreationCrossingClass
- chanceCreationShooting
- chanceCreationShootingClass
- chanceCreationPositioningClass
- defencePressure
- defencePressureClass
- defenceAggression
- defenceAggressionClass
- defenceTeamWidth
- defenceTeamWidthClass
- defenceDefenderLineClass

Now, all the categorical variables were encoded by first casting them to by of type category and then using the in-built pandas categorical encoding.

Decision Tree Classifier/Random Forest Classifier:

The models were trained by passing in the predictor variables as all of the above-specified attributes, and the target to predict was the result of the match. The optimum depth was allowed to be chosen by the algorithm itself on observing the data, and the following results were obtained:

	Decision Tree	Random Forest
Train Accuracy	0.480995528359614	0.671334431630972
Test Accuracy	0.4832941176470588	0.444

From the results summarized above, it can be seen that once again, the match cannot simply be predicted as a combination of factors, even when treated as a multivariate model. Although a relatively higher train accuracy of 67% is obtained with the random forest, it is most likely overfitting as suggested by the very low test accuracy.

Neural Network:

Our rationale behind moving further with the same hypothesis was because of the fact that sometimes a neural network is capable of capturing very high level features not easily detected by relatively simpler classification models like decision tree and random forest. To train a neural network, we first carefully chose the following architecture:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 20)                820       
_________________________________________________________________
dense_2 (Dense)              (None, 10)                210       
_________________________________________________________________
dense_3 (Dense)              (None, 6)                 66        
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 28        
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 10        
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 9         
=================================================================
Total params: 1,143
Trainable params: 1,143
Non-trainable params: 0
_________________________________________________________________
```

The reasoning behind the choice of this architecture was due to the fact that neural networks are well-known to be effective at extracting high-level features from low-level ones. Our hope was to extract a refined high-level numeric representation of the home and away teams' skill level at the layer with 2 neurons. So, essentially, we were hoping that the neural network could behave as an autoencoder.

Unfortunately, the results obtained did not line up with our hopes as the training and testing accuracies were both very low, thus suggesting that the neural network could not extract any interesting or insightful high-level features. The results are summarized below:

```
Epoch 32/40
15934/15934 [==============================] - 4s 220us/step - loss: 1.0337 - accuracy: 0.4827 - val_loss: 1.0413 - val_accurac
y: 0.4778
Epoch 33/40
15934/15934 [==============================] - 4s 233us/step - loss: 1.0333 - accuracy: 0.4869 - val_loss: 1.0382 - val_accurac
y: 0.4785
Epoch 34/40
15934/15934 [==============================] - 3s 172us/step - loss: 1.0324 - accuracy: 0.4870 - val_loss: 1.0377 - val_accurac
y: 0.4789
Epoch 35/40
15934/15934 [==============================] - 2s 144us/step - loss: 1.0335 - accuracy: 0.4836 - val_loss: 1.0374 - val_accurac
y: 0.4791
Epoch 36/40
15934/15934 [==============================] - 2s 144us/step - loss: 1.0313 - accuracy: 0.4849 - val_loss: 1.0375 - val_accurac
y: 0.4804
Epoch 37/40
15934/15934 [==============================] - 2s 100us/step - loss: 1.0310 - accuracy: 0.4870 - val_loss: 1.0433 - val_accurac
y: 0.4817
Epoch 38/40
15934/15934 [==============================] - 2s 99us/step - loss: 1.0307 - accuracy: 0.4894 - val_loss: 1.0361 - val_accurac
y: 0.4795
Epoch 39/40
15934/15934 [==============================] - 2s 102us/step - loss: 1.0305 - accuracy: 0.4877 - val_loss: 1.0382 - val_accurac
y: 0.4793
Epoch 40/40
15934/15934 [==============================] - 2s 104us/step - loss: 1.0293 - accuracy: 0.4883 - val_loss: 1.0441 - val_accurac
y: 0.4816
```

Subsequent Hypotheses:

Hypothesis: The average rating of all the players in a team strongly correlates with their ability to win a match.

Dataset preparation:

Firstly, as we needed the ratings of every player in the home and away side for every match, all records which had not have the player ID of any one of the 22 players on the field were removed. As a result, the size of the match database was reduced from 25979 records to 21374 records, which is still reasonably enough data to test our hypothesis on. We prepared our dataset by scanning through all the records in the match database, linking each of the 22 home and away players with the player database, and retrieving each player's rating at the rough time period when the match took place. This was done by using extensively vectorized operations in pandas. Then, using these ratings, the average home team score and away team score was calculated for each match record in the match database. Here is a snippet of the database after these operations:

home_team_id	away_team_id	home_team_goal	away_team_goal	...	VCD	VCA	GBH	GBD	GBA	BSH	BSD	BSA	home_team_score	away_team_score
8203	9987	2	1	...	3.25	2.35	2.90	3.25	2.30	2.80	3.2	2.25	59.727273	64.090909
9984	8342	1	3	...	3.20	2.35	2.90	3.20	2.30	2.62	3.2	2.38	65.818182	68.545455
8635	10000	2	0	...	4.35	8.00	1.35	4.33	8.50	1.36	4.2	7.00	70.454545	61.000000
8203	8635	2	1	...	3.30	1.75	4.50	3.40	1.75	4.20	3.3	1.75	59.818182	70.545455
10000	9999	0	0	...	3.50	4.50	1.65	3.50	5.00	1.70	3.4	4.33	62.090909	60.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Exploring the Correlation between Average Scores and Match Result:

After this, the correlation between the difference between the scores (ratings)of the home and away teams, and the result of the match was obtained using different thresholds of score difference between the teams. In the screenshot, a threshold of 2 indicates that the correlation was found between the cases when the difference between the teams’ scores was greater than or equal to 2 and the cases when the matches were drawn.

```
Threshold: 0.1
0.24931126008780025
Threshold: 0.2
0.24717801075256898
Threshold: 0.3
0.24820088087977962
Threshold: 0.4
0.24716014069398975
Threshold: 0.5
0.24707519422108998
Threshold: 0.6
0.24658705744381318
Threshold: 0.7
0.246705165257946
Threshold: 0.8
0.24679759416721125
Threshold: 0.9
0.24548576141305425
Threshold: 1.0
0.24344558912617797
Threshold: 2.0
0.23168987048599335
Threshold: 5.0
0.1872360398538149
Threshold: 5.5
0.17662717570924888
```

It can clearly be inferred from these values that the correlation is strictly less than 25% for all the different tested values of the threshold. This is a strong indicator that our hypothesis may be wrong and a team’s performance in a specific match cannot indeed be confidently decided by the strength of the individual players on its side.

Final Testing of Hypothesis Using Decision Tree and Random Forest Classifier:

It is known that using advanced models such as decision trees and ensemble methods such as random forests can sometimes lead to the detection of patterns not clearly detectable through basic statistical data analysis. To confirm that our hypothesis truly does not hold weight, we trained decision tree and random forest models. The results of our models for various depths are summarized below:

	Decision Tree	Random Forest
	Depth 1	
Train Accuracy	0.4603193169191181	0.4951751564418972
Test Accuracy	0.4535672514619883	0.4844444444444444
	Depth 2	
Train Accuracy	0.4980408211006492	0.4966372302473829
Test Accuracy	0.4856140350877193	0.4849122807017544
	Depth 3	
Train Accuracy	0.4980408211006492	0.5103222410667291
Test Accuracy	0.4856140350877193	0.4963742690058479
	Depth 4	

Train Accuracy	0.5189777179952044	0.5180419907596936
Test Accuracy	0.5071345029239767	0.5069005847953216
	Depth 5	
Train Accuracy	0.5199719281829347	0.523714837124978
Test Accuracy	0.5080701754385964	0.5111111111111111
	Depth 6	
Train Accuracy	0.5277501608281187	0.5282765073980935
Test Accuracy	0.5155555555555555	0.5134502923976608
	Depth 7	
Train Accuracy	0.5317854845312592	0.5312591379612843
Test Accuracy	0.5090058479532164	0.5134502923976608
	Depth 8	
Train Accuracy	0.5383940581320545	0.5402655126030762
Test Accuracy	0.5057309941520468	0.5181286549707602
	Depth 9	
Train Accuracy	0.5500321656237207	0.555412597227908
Test Accuracy	0.4998830409356725	0.5108771929824562
	Depth 10	
Train Accuracy	0.5646529036785777	0.5750043862214165
Test Accuracy	0.4935672514619883	0.512046783625731
	Depth 20	
Train Accuracy	0.8351950406456518	0.8887654248786478
Test Accuracy	0.4140350877192982	0.4381286549707602
	Depth 50	
Train Accuracy	0.8908708111585473	0.8907538452541084
Test Accuracy	0.4056140350877193	0.44
	Depth 100	
Train Accuracy	0.8908708111585473	0.8908123282063278
Test Accuracy	0.4028070175438596	0.4360233918128655

From the results collated above, we can first observe that both the train and test accuracies for both models are first low and increase. While the train accuracy strictly increases as the depth is increased, for both models, the test accuracy peaks somewhere between depth 10 and 20. Since the test accuracy is only the range of 50%, we can conclude that the model is not capable of predicting the result of the match using only the average ratings of the home and away teams.

Hypothesis: The upsets would generally relate to the anomalies in the datasets in terms of the team's skill level.

Dataset Preparation:

We prepared our dataset by scanning through the match record database and linked up the home team ID and away team ID with the teams database in order to retrieve the following attributes:

- buildUpPlaySpeed
- buildUpPlayPassing
- chanceCreationPassing
- chanceCreationCrossing
- chanceCreationShooting

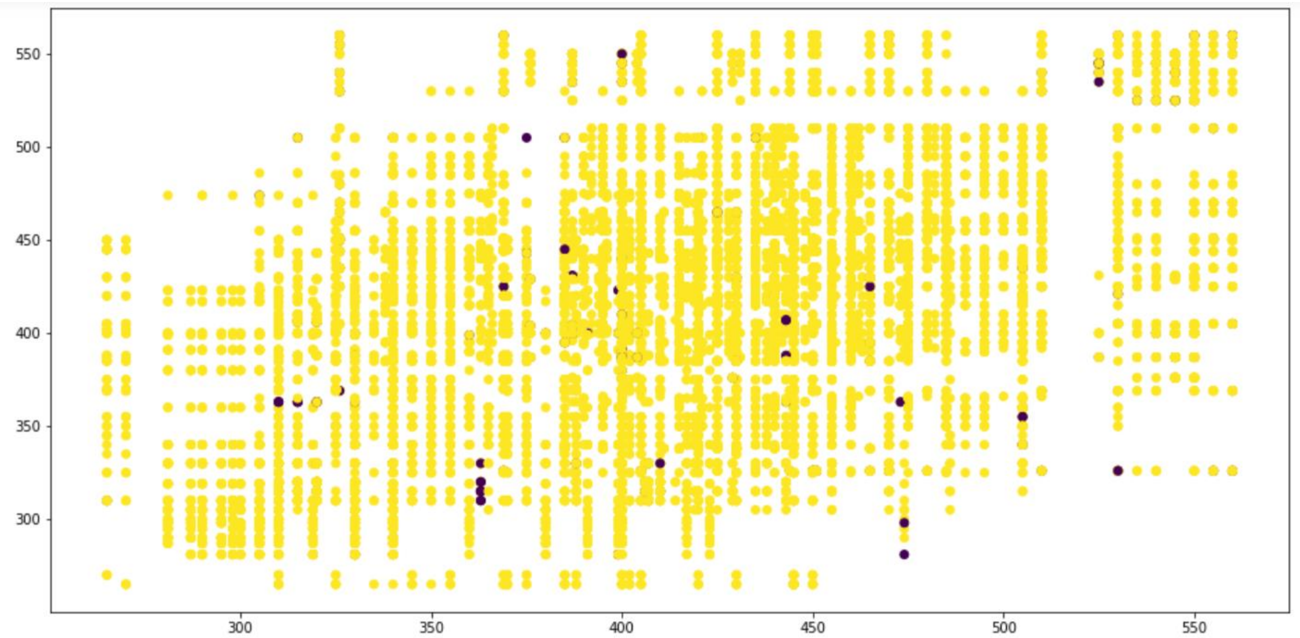
- defencePressure
- defenceAggression
- defenceTeamWidth

Now, the values of all these numeric attributes are summed up in order to give a numeric representation of the team's capability.

Further, we converted the result of the match to categorical values 1, 0 and -1, where 1 represents a victory for the home side, 0 represents a draw and -1 represents a loss for the home side.

Anomaly Detection Model:

We hypothesized that the upsets detected using the bet365 data would be related to the anomalies found with the team's numeric skill level as calculated above. In order to test this, we ran the Local Outlier Factor algorithm on the home and away team's aggregate scores, with number of neighbors set to 5 and fraction of anomalies set to 0.123 and obtained the following:



The ideal parameters for the number of neighbours and the contamination fraction was determined by trial and error. In this process, we can examine from the plot that as expected, the algorithm has found multiple points where the difference between the home team and away team aggregate score are large. This is to be expected due to the nature of the local outlier factor algorithm, and these records can be expected to give biased outputs. However, as we can see, the algorithm has also detected anomalies in the central region. When the list of matches detected by the algorithm were compared against the list of matches extracted as upsets using the bet365 data, 6 records were found to be common. Although this number seems quite small, it gives a good pointer to the fact that we are proceeding in the correct direction to detecting the upsets without using a criterion-based algorithm.