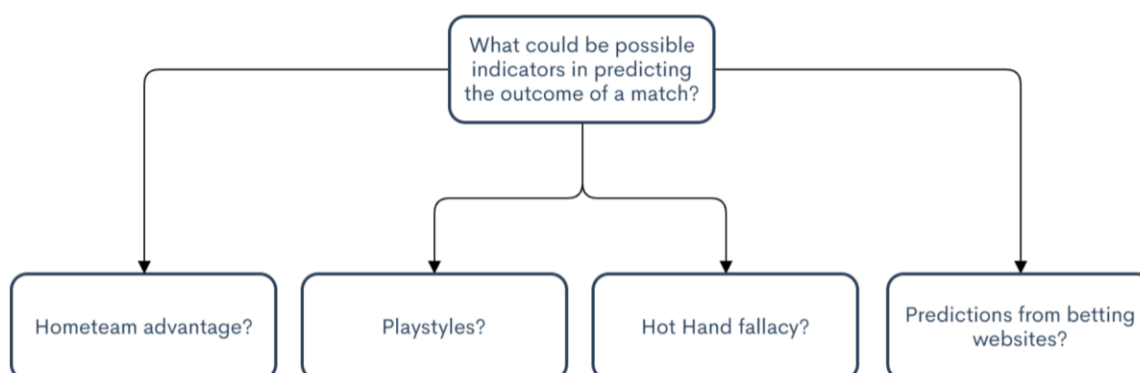


TheDataOpen

Team 2:

1. Tsun Wang Sau
2. Siddhant Pathak
3. Tsz Hin Koon
4. Derek Ng Wei Kang

The Big Question



Home Team Advantage (covered by Tsun Wang Sau)

What is the home team advantage? Put it simply, it is the idea that the team playing on their home turf has an unseen advantage in helping them win the match. Is it a myth? Well, through our study we shall visit if that is indeed the case and if it has a place in aiding us to predict the outcome of a match.

Playstyles (covered by Derek Ng Wei Kang)

In the big leagues, it is undeniable that the skill level of each team is very high, however, which of the team attributes are the game changers in terms of a team's win and losses? Finding that out will aid in our choice of feature variables to predict a matches' outcome.

Hot Hand Fallacy (covered by Siddhant Pathak)

The success of various football clubs varies over years and is dependent upon a lot of factors such as squad formation, team chemistry, and various other attributes. These are influential factors in determining the winning streak of a team, which in turn is directly proportional to the number of goals scored as well. Is there a way that, given previous seasons' data and statistics, can we base a Machine Learning model and train it in such a manner that it can predict upcoming seasons' status?

Prediction from betting websites (covered by Tsz Hin Koon)

If we believe that the sports betting market is efficient, the betting odds should somehow tell us how people think of the market, i.e., the expected win rate of each team. But how accurate are these predictions? Can these help us to build a prediction model to predict the outcome?

HOME TEAM ADVANTAGE

Key Findings

- The **home team winning ratio is 45.87%** and the **away team winning rate is 28.74%** on average considering all the matches in match.csv. The home team advantage, defined as the **average** treatment effect of being home team, is equal to **45.87% - 28.74%=20.13%**.
- The home team advantage exists for all leagues with an average of 16.77% higher winning ratio and the standard deviation is **3.55%**. **Scotland Premier League (league_id: 19694)** has the smallest home team advantage with **7.83%**, which is **farthest from the mean**.

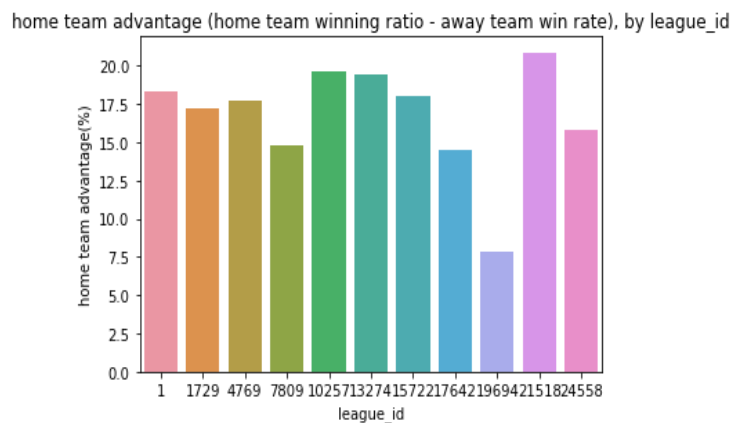


Figure 1 : Bar Chart of home team advantage for different leagues

- Home team existed for all the seasons with an **average of 17.14%** and **2.39% standard deviation**. The trend is decreasing slightly, by **19.15-13.44=5.71%**, which is not significant.

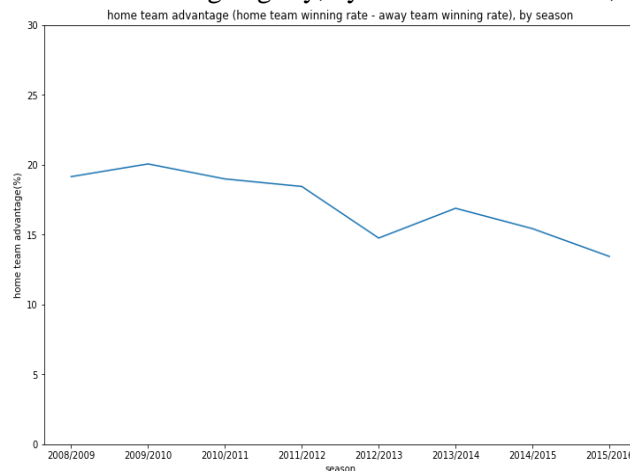


Figure 2: The Line Chart of home team advantage for different seasons

- We hypothesize that there is a linear relationship for each team that **winning rate = away team winning rate + x * D**, where x is the home team advantage and D is the condition whether the team is the home team (1 if it is the home team and 0 if away team). By **OLS (ordinary least square)**, we found that the **hypothesis is likely to be true**.
- We did not find any useful linear relationship between team attributes and home team advantage as the **R² value of the OLS model is 0.087**, which is **small**.

Data Processing

- The **selection bias is small** as the difference between the number of matches a team being a home team and away team considering all the matches in match.csv is at most 2, which implies each team plays as the home team and away team with about the same number of matches. Therefore, **the average treatment effect is the home team winning ratio - away team winning ratio with small selection bias**.
- We found the **mean of winning ratios of all matches** in match.csv by using the number of matches that the home team wins (**home_team_goal > away_team_goal**) and divides it by the total number of matches. Similarly, for the away team. We also got the home team advantage for different leagues by grouping matches with the same league id and we use the same method above to find the mean. We also do it similarly for seasons.
- For the home team advantage of each team, we used the total number of matches the team plays for all seasons, instead of analysing each season independently, as the number of plays for a team in a season is about 15 which is small as a sample size. Even though there are differences for a team in different seasons, the standard deviations are not large, given the mean of all standard deviations as shown below.

team_attributes	standard deviation
buildUpPlaySpeed	8.914053
buildUpPlayDribbling	3.398474
buildUpPlayPassing	8.242745
chanceCreationPassing	7.876966
chanceCreationCrossing	8.136518
chanceCreationShooting	8.076981
defencePressure	7.661695
defenceAggression	7.995129
defenceTeamWidth	7.297936

Figure 3: Mean of the Standard Deviation of Team Attributes of each team

- We encoded the categorical team attributes with -1,0,1 as shown in the code in the appendix.

Analysing Data

Is winning rate = away team winning rate + $x * D$?

We use **OLS** to make a **linear regression model** between a home team winning rate and away team winning rate to **see if the assumption winning rate = away team winning rate + $x * D$ is true**. From the OLS summary below, the slope of the line is 0.9095 which is close to our assumption (1) and the **standard deviation is 0.044** which is small. Also, **the constant term is 0.1795** which is quite close to the mean of the home team advantage. We may conclude that **our hypothesis is likely to be true**.

OLS Regression Results						
Dep. Variable:	home_win_rate	R-squared:	0.586			
Model:	OLS	Adj. R-squared:	0.585			
Method:	Least Squares	F-statistic:	420.7			
Date:	Fri 26 Mar 2021	Prob (F-statistic):	7.51E-59			
Time:	12:25:23	Log-Likelihood:	281.24			
No. Observations:	299	AIC:	-558.5			
Df Residuals:	297	BIC:	-551.1			
Df Model:	1					
Covariance Type: nonrobust						
	coef	std err	t	P> t	[0.025	0.975]
const	0.1795	0.012	14.647	0	0.155	0.204
away_win_rate	0.9095	0.044	20.511	0	0.822	0.997
Omnibus:	43.446	Durbin-Watson:	1.702			
Prob(Omnibus):	0	Jarque-Bera (JB)	87.167			
Skew:	-0.765	Prob(JB):	1.18E-19			
Kurtosis:	5.158	Cond. No.	8.59			

Figure 4: OLS summary for the linear relationship between a home team winning ratio and away team winning ratio

The linear relationship between team attributes and home team advantage

From the heatmap on the right, we can see that the **correlations between the numeric variables are small**, but the categorical variable is usually correlated to the numeric variable in front of it, so we drop the categorical variable and only use the numeric variable as the independent variables for the OLS model to find linear relationships between team attributes and home team advantage for each team so that we can avoid collinearity issue. From the OLS summary report below, we can see that **$R^2=0.087$** , which implies the **line does not inform us more information than just using the mean**.

OLS Regression Results
Dep. Variable: home_team_advantage R-squared: 0.087
Model: OLS Adj. R-squared: 0.057
Method: Least Squares F-statistic: 2.935
Date: Thu, 25 Mar 2021 Prob (F-statistic): 0.00242
Time: 16:13:54 Log-Likelihood: -1037.6
No. Observations: 288 AIC: 2095.
Df Residuals: 278 BIC: 2132.
Df Model: 9

Figure 5: OLS summary for the linear relationship between team attributes and home team advantage

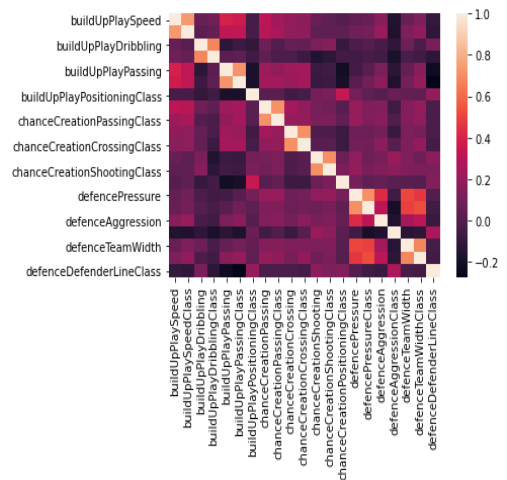


Figure 6: Heatmap of Correlation Matrix of team attributes

PLAYSTYLE

Key Findings

- We looked at the nine team attributes: **buildUpPlaySpeed**, **buildUpPlayDribbling**, **buildUpPlayPassing**, **chanceCreationPassing**, **chanceCreationCrossing**, **chanceCreationShooting**, **defencePressure**, **defenceAggression** and **defenceTeamWidth**.
- Most influential attributes: **buildUpPlayPassing** and **defencePressure** as they make the most contribution to the **win-loss ratio** of a team.

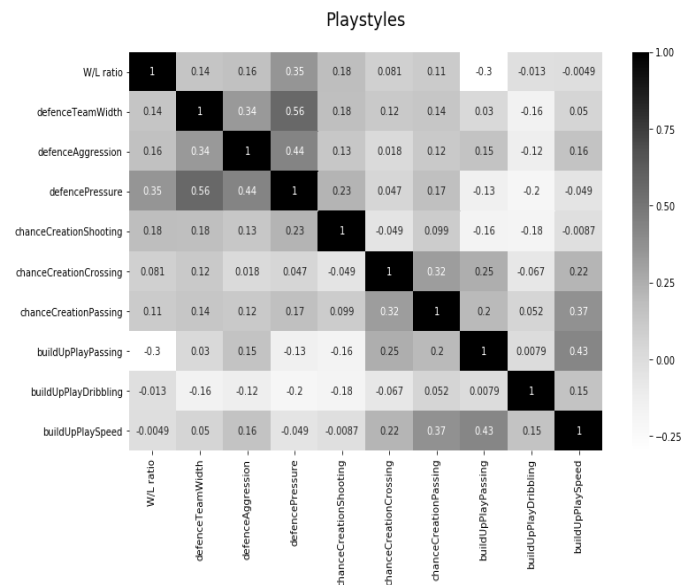


Figure 7: Correlation between the attributes and W/L ratio

- As we can see, the correlation of **buildUpPlayPassing**, **chanceCreationCrossing**, and **defencePressure** is the **highest**. Supporting the claim that the three attributes are truly the most influential.

	coef	std err	t	P> t	[0.025	0.975]
defenceTeamWidth	-0.0734	0.065	-1.134	0.258	-0.201	0.054
defenceAggression	0.0870	0.061	1.428	0.154	-0.033	0.207
defencePressure	0.2960	0.071	4.185	0.000	0.157	0.435
chanceCreationShooting	0.0756	0.056	1.358	0.176	-0.034	0.185
chanceCreationCrossing	0.1361	0.057	2.375	0.018	0.023	0.249
chanceCreationPassing	0.0371	0.060	0.617	0.538	-0.081	0.155
buildUpPlayPassing	-0.3392	0.062	-5.502	0.000	-0.461	-0.218
buildUpPlayDribbling	0.0519	0.055	0.939	0.349	-0.057	0.161
buildUpPlaySpeed	0.0932	0.063	1.487	0.138	-0.030	0.217

Figure 8: Regression models with the nine attributes

DefencePressure (DP): Compared to its other attribute counterparts, it is observed that the DP attribute has the **largest positive coefficient**. Per unit change in DP results in a **0.2960 positive change in the W/L ratio** of a team.

BuildUpPlayPassing (BUPP): It is also observed that the BUPP attribute has the **largest negative coefficient**. Per unit change in BUPP results in a **-0.3392 change in the W/L ratio** of a team.

Thus, out of all the attributes given, we could potentially use these two attributes to shape our model to predict the outcome of the matches.

	team_id	team_long_name	team_short_name	Hometeam goal ratio	Awayteam goal ratio	W/L ratio
184	1601	Ruch Chorzów	CHO	1.203252	0.781065	0.992159
21	1773	Oud-Heverlee Leuven	O-H	1.013699	0.584270	0.798984
194	1957	Jagiellonia Białystok	BIA	1.430894	0.578125	1.004510
222	2033	S.C. Olhanense	OLH	0.760870	0.581197	0.671033
191	2182	Lech Poznań	POZ	2.362637	1.319328	1.840983

Attaining the W/L ratio

- As there are **anomalies** in some wins and losses such that there were cases where **the team got dominated or the team dominated the opponent**.
- To get a better representation of the team's performance, I attained the **ratio of goals** scored as a **home team** against their opponents and the ratio of goals scored as an **away team** against home team opponents.
- The **W/L ratio** is the **average of the home team and away team goal ratio**.

	team_id	team_long_name	team_short_name	Hometeam goal ratio	Awayteam goal ratio	W/L ratio	defenceTeamWidth	defenceAggression	defencePressure	chanceCreationStr
184	1601	Ruch Chorzów	CHO	1.203252	0.781065	0.992159	49.333333	47.333333	47.166667	
21	1773	Oud-Heverlee Leuven	O-H	1.013699	0.584270	0.798984	50.000000	44.000000	43.000000	
194	1957	Jagiellonia Białystok	BIA	1.430894	0.578125	1.004510	53.666667	56.333333	49.333333	
222	2033	S.C. Olhanense	OLH	0.760870	0.581197	0.671033	45.400000	33.800000	42.200000	
191	2182	Lech Poznań	POZ	2.362637	1.319328	1.840983	54.500000	48.500000	51.666667	

- After sorting the team_attribute data, it was found that there was only data for 288 teams while the team data has 299 teams, meaning there was a need to **filter out the 11 rows**.
- To go about this, we sorted both data frames according to the team_id, followed by iteration and comparison of the team_attribute and team data frames.
- To later remove the rows in the data frame of a team without data, **rows with column values NaN were dropped** as they would remain unchanged after manual insertion of data.

HOT HAND FALLACY

Key Findings:

- The **summary statistics** are as follows:

goals scored	
count	46.000000
mean	75.260870
std	19.597545
min	40.000000
25%	62.250000
50%	74.500000
75%	87.000000
max	115.000000

Figure 9: Summary Statistics of Goals Scored by FC Barcelona from 1970-2015

- The **time-based plot** of the data provided (from **1970-2015**) is as follows:

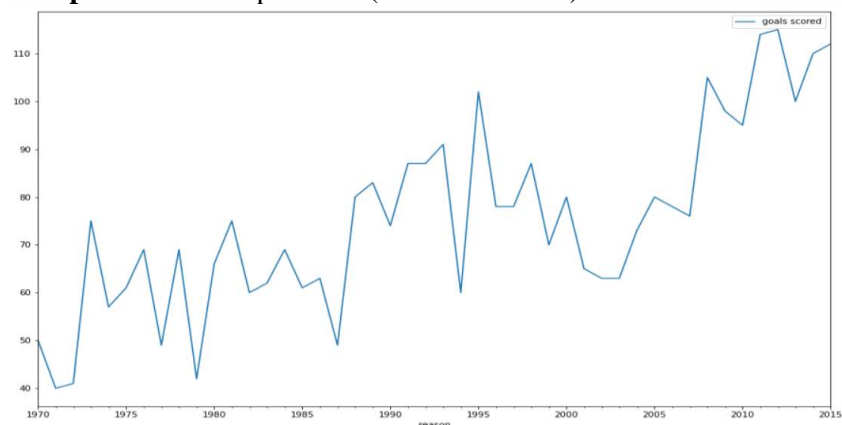


Figure 10: Time-Series on Number of Goals Scored over the years

- To highlight the skewness of the data, we used the **Normal Q-Q graph** (Sample Quantities vs Theoretical Quantities):

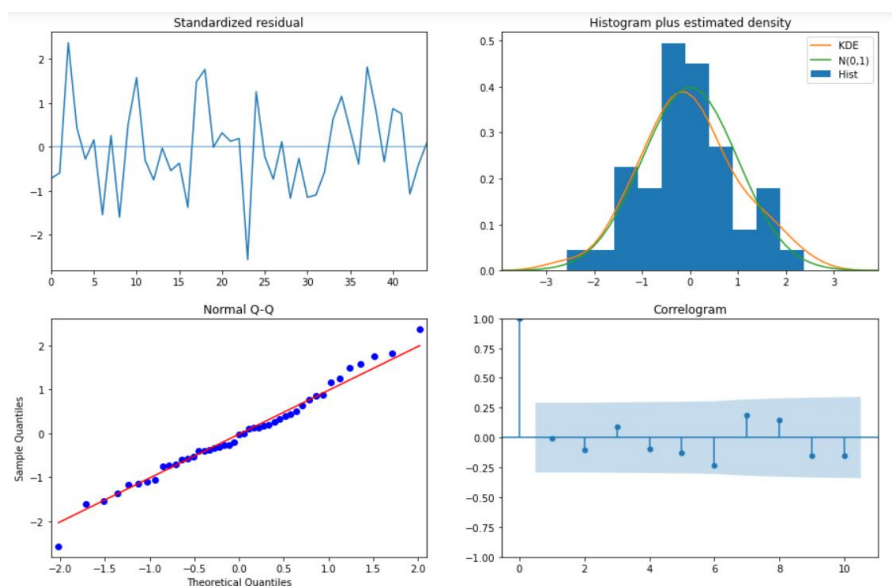


Figure 11: Basic Diagnostic Plots to check distribution of data

- Top Left:** The residual errors seem to fluctuate around a mean of zero and have a non-uniform variance.
- Top Right:** The density plot suggested normal distribution with mean zero.

3. **Bottom Left:** All the dots should fall perfectly in line with the red line. Any significant deviations would imply the distribution is skewed.
 4. **Bottom Right:** The Correlogram, aka, ACF plot shows the residual errors are not autocorrelated. Any autocorrelation would imply that there is some pattern in the residual errors which are not explained in the model.
- To check for the accuracy metrics, we used the model to predict the number of goals scored by the club in the season “2016-17”.
 - The model predicts that approximately there will be 110 goals scored by the club in the 2016/17 season. We checked these with the actual metric, from the official website of the club and Google:

116

The 116 goals scored by FC Barcelona in the league season 2016/17 has set a new record at the Club. 22 May 2017

[www.fcbarcelona.com > news > fc-barcelona-end-the-seas...](http://www.fcbarcelona.com/news/fc-barcelona-end-the-seas...)

[FC Barcelona end the season with 116 goals, a new record in ...](#)

Figure 12: Referenced from Google.com

- I observed that even though there was a variable trend in the number of goals scored in the previous seasons, there is almost a linear increase in the upcoming seasons.

Technical Exposition:

I made use of certain Python Libraries (externally installed ones too) such as **Pandas**, **Numpy**, **Matplotlib**, **StatsModels**, and **PyramidARIMA**.

PyramidARIMA is a statistical library designed to fill the void in Python's time series analysis capabilities which include a lot of useful functions including the creation of the **AUTO-ARIMA** model by fine-tuning the hyperparameters, by itself, to optimize the AIC.

Data Acquisition and Cleaning

For more values, especially about years before '2008', I have procured an additional dataset named "FMEL Dataset.csv". This dataset contains information on the **famous Spanish clubs from 1970-2017/18**. I will be using the data from **1970-2007/08** for our research and development of the Time Series Model. It was acquired from **Kaggle**.

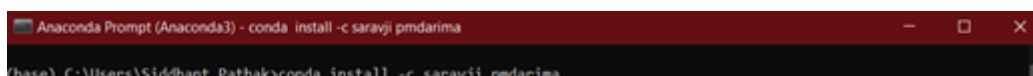
The unnecessary columns were dropped, and the useful columns were combined to a cleaner and simpler **Pandas DataFrame**. The cleaned dataset and indexed datasets looked like this:

```
In [23]: timeseries2.head()
Out[23]:
```

season	goals scored
1970-01-01	50
1971-01-01	40
1972-01-01	41
1973-01-01	75
1974-01-01	57

Data Modelling

The **pmdarima** package requires **external installation**, for that you need to go to the Anaconda Prompt/any IDE prompt and use this statement:



```
Anaconda Prompt (Anaconda3) - conda install -c saravji pmdarima
(base) C:\Users\Siddhant Pathak>conda install -c saravji pmdarima
```


We used the inbuilt function: **auto_arima()**, which uses a stepwise approach to search multiple combinations of hyperparameters and chooses the best model. The results of the same are as follows:

```

Performing stepwise search to minimize aic
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=362.314, Time=0.06 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=380.316, Time=0.01 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=366.065, Time=0.02 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=361.887, Time=0.02 sec
ARIMA(0,1,0)(0,0,0)[0] : AIC=378.656, Time=0.01 sec
ARIMA(0,1,2)(0,0,0)[0] intercept : AIC=361.092, Time=0.04 sec
ARIMA(1,1,2)(0,0,0)[0] intercept : AIC=363.090, Time=0.03 sec
ARIMA(0,1,3)(0,0,0)[0] intercept : AIC=inf, Time=0.13 sec
ARIMA(1,1,3)(0,0,0)[0] intercept : AIC=inf, Time=0.17 sec
ARIMA(0,1,2)(0,0,0)[0] : AIC=inf, Time=0.06 sec

Best model: ARIMA(0,1,2)(0,0,0)[0] intercept
Total fit time: 0.565 seconds
SARIMAX Results
=====
Dep. Variable: y No. Observations: 46
Model: SARIMAX(0, 1, 2) Log Likelihood: -176.546
Date: Fri, 26 Mar 2021 AIC: 361.092
Time: 22:45:54 BIC: 368.319
Sample: 0 HQIC: 363.786
Covariance Type: opg
=====
coef std err z P>|z| [0.025 0.975]
-----
intercept 1.5114 1.106 1.366 0.172 -0.657 3.680
ma.L1 -0.7818 0.181 -4.314 0.000 -1.137 -0.427
ma.L2 0.3520 0.149 2.360 0.018 0.060 0.644
sigma2 147.4846 31.001 4.757 0.000 86.724 208.245
=====
Ljung-Box (L1) (Q): 0.00 Jarque-Bera (JB): 0.12
Prob(Q): 0.95 Prob(JB): 0.94
Heteroskedasticity (H): 0.75 Skew: 0.12
Prob(H) (two-sided): 0.59 Kurtosis: 3.05
=====

```

Figure 13: SARIMAX Test Results using AUTO-ARIMA models

The **final forecast modeling step** made use of the inbuilt functions as well:

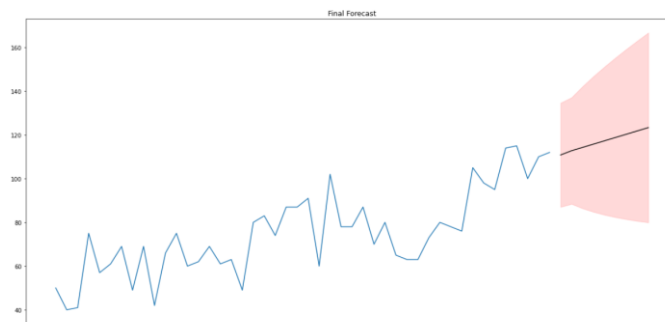


Figure 14: Forecast predicted by the Auto-ARIMA Model

```
In [36]:
```

	time	
42	2012-01-01	115.0
43	2013-01-01	100.0
44	2014-01-01	110.0
45	2015-01-01	112.0
46	NaT	110.0
47	NaT	112.0
48	NaT	114.0

Figure 15: The values with “NaT” (Not a Time) are the ones that are predicted by the model.

The predictions are close to the actual value which implies that our model is very good, with some errors that are bound to happen and are uncontrollable.

We can make predictions for longer periods as well by changing the values of *n_periods* in the code above while training the model. Similar procedures can be followed for other football clubs as well to work out their numbers as well.

PREDICTIONS OF BETTING WEBSITES

Key Findings

- The **implied win rates** show that people are usually more confident in betting when the home team has an obvious edge, but less confident when the away team has the edge.
- The implied draw rate shows a **threshold of around 30%**, suggesting that people are usually less confident and bad at placing draws bets and this affect the accuracy of our models.
- We successfully built two models based on logistic regression and decision trees that attain an **accuracy score of over 53%**, compared to around **38% when directly applying** the implied probability model and **27% when wild guessing**.

Implied Win Rate

We compute the implied probability with the following formula:

$$P(x) = \frac{1}{\frac{1}{n} \sum_{n=1}^n x_n}$$

Figure 16: Mathematical Formula for Implied Probability where n is number of agencies and x_n is the odd (Home/Away/Draw)

Noted that $P(\text{Home}) + P(\text{Away}) + P(\text{Draw}) \neq 1$, this is because of the betting agencies' margin, to adjust, we use the following formula:

$$P(x)_{\text{adjusted}} = \frac{P(x)}{P(\text{Home}) + P(\text{Away}) + P(\text{Draw})}$$

Figure 17: Mathematical Formula for Implied Probability (considering error margins)

Some interesting findings when looking at the win rates

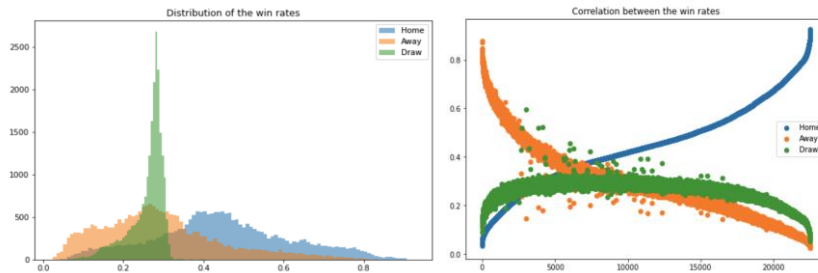


Figure 18: Distribution of the Win Rates Figure 19: Correlation between the Win Rates

- The implied probability shows that the market is often more **confident** in the **home** team.
- People are more certain in placing bets when they are **more confident in the home team**, given that people rarely bet for draws when the home team is expected to win, but **more draw bets when they are not confident in the home team**.
- There is a **certain threshold for draw rates**, suggesting that people **are not confident/ bad at betting for draws**.

How accurate is the implied probability model?

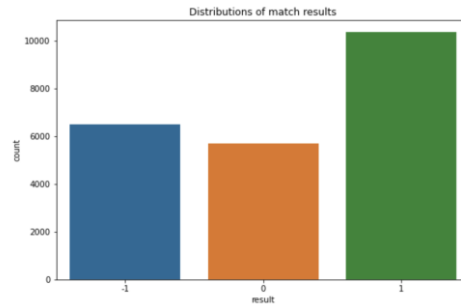


Figure 20: Distribution of Match Results

- Here we try to use the **Monte Carlo method** to find out how accurate the agencies' implied rates is when comparing to wild guessing.

Random {-1, 0, 1}

```
rand_ls = []
for i in range(repeat):
    rand_result['rand'] = [np.random.randint(low = -1, high = 1) for k in rand_result.index]
    rand_ls.append(metrics.accuracy_score(rand_result['result'], rand_result['rand']))
print("Accuracy Score: " + str(np.array(rand_ls).mean()))
```

Accuracy Score: 0.2702513430715269

Implied Probability

```
implied_ls = []
for i in range(repeat):
    rand_result['prob'] = rand_result[['mean|adjusted_implied_home_win_rate', 'mean|adjusted_implied_away_win_rate']]
    rand_ls.append(metrics.accuracy_score(rand_result['result'], rand_result['prob']))
print("Accuracy Score: " + str(np.array(rand_ls).mean()))
```

Accuracy Score: 0.38199017300833216

- The implied probability model is just slightly predicting more accurate results (38.2%) than **wild guessing (27.0%)**.

How can we make a better model?

The correlation map of the implied rates shows a strong correlation between home win rates, away win rates, and the results, not too strong for draw rate, same as what we expected based on the previous observations.

The **negative correlation** of **draw rate** to the **results** suggesting that when the draw rate is high, the home team is **less likely to win**, which is the same as our previous observations.

Based on the correlation that we noticed here, we believed a **logistic regression model**, or a decision tree model will be suitable to tackle it, we will see both in the following.

We split **30% of our data as test data**.

Logistic Regression Model

- Here we fit our **train data** with the results and built a **logistic regression model** and feed it with the test data and see how it performs.

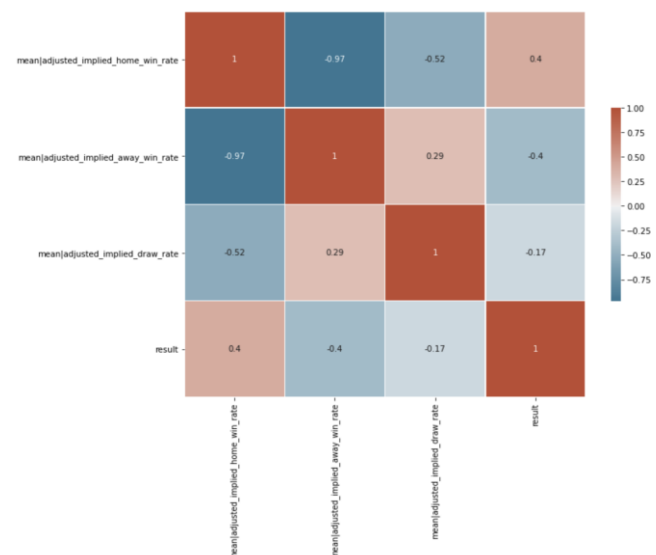
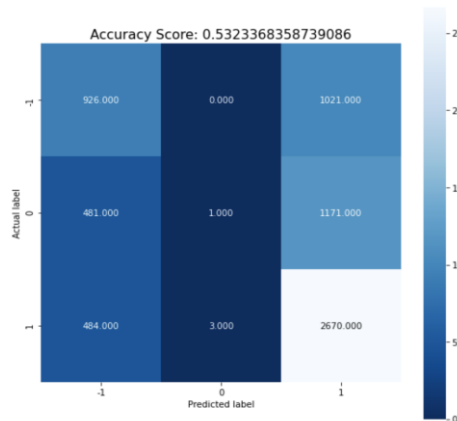


Figure 21: Correlation Heatmap of the implied rates



Our model shows an accuracy score of 0.532, meaning that it is successful in predicting **53.2%** of the match results correctly, which is a significant increase from what we had before.

One thing that worth noticing there is that our model **did not make too many bets on draws** - which is again suggesting that the **betting odds cannot affect or be useful in predicting draws**.

Figure 22: Confusion Matrix (Actual v/s Predicted)

Decision Tree Model

- Here we try to build a decision tree model to see what will happen, an accuracy score computed by the **Monte Carlo method** shows 0.53 (**53% accurate predictions**) for our model, like our logistic regression model above, which is **encouraging**,

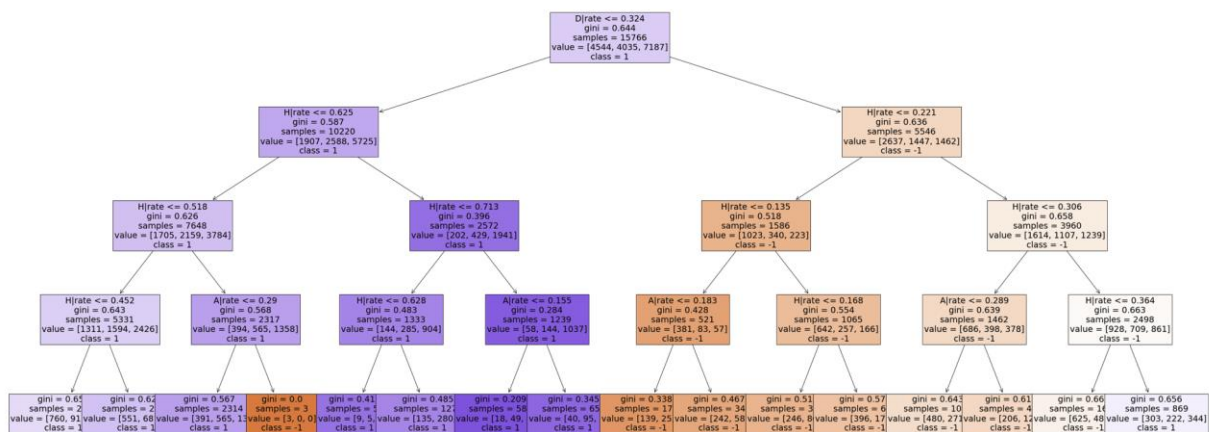


Figure 23: Decision Tree fit on our Data

- The above is one of our trees, same as what happened in our logistic regression model, however same as what we discussed before, the market does not seem to have a good prediction in draws, this greatly affects our predictions.

Conclusion

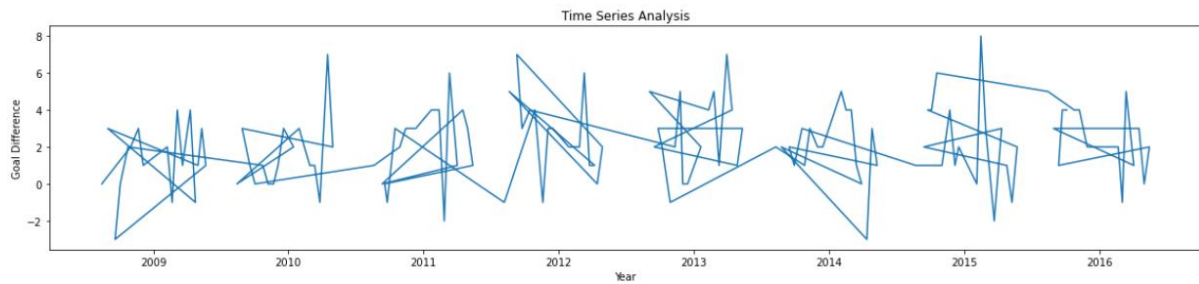
- In short, our logistic regression model and decision tree model here **greatly increase the accuracy of predicting match results**.
- However, it is **not perfect**, betting odds along do **not work well when predicting draws**, feeding our models with more relevant data can probably increase the accuracy of the predictions.

APPENDIX AND REFERENCES

Encoding Categorical Data

```
team_attributes['buildUpPlaySpeedClass'] = team_attributes['buildUpPlaySpeedClass'].map({'Balanced': 0, 'Fast':1, 'Slow':-1})
team_attributes['buildUpPlayDribblingClass'] = team_attributes['buildUpPlayDribblingClass'].map({'Little': -1, 'Normal':0, 'Lots':1})
team_attributes['buildUpPlayPassingClass'] = team_attributes['buildUpPlayPassingClass'].map({'Short': -1, 'Mixed':0, 'Long':1})
team_attributes['buildUpPlayPositioningClass'] = team_attributes['buildUpPlayPositioningClass'].map({'Organised':0, 'Free Form':1})
team_attributes['chanceCreationPassingClass'] = team_attributes['chanceCreationPassingClass'].map({'Safe': -1, 'Normal':0, 'Risky':1})
team_attributes['chanceCreationCrossingClass'] = team_attributes['chanceCreationCrossingClass'].map({'Little': -1, 'Normal':0, 'Lots':1})
team_attributes['chanceCreationShootingClass'] = team_attributes['chanceCreationShootingClass'].map({'Little': -1, 'Normal':0, 'Lots':1})
team_attributes['chanceCreationPositioningClass'] = team_attributes['chanceCreationPositioningClass'].map({'Organised':0, 'Free Form':1})
team_attributes['defencePressureClass'] = team_attributes['defencePressureClass'].map({'Deep': -1, 'Medium':0, 'High':1})
team_attributes['defenceAggressionClass'] = team_attributes['defenceAggressionClass'].map({'Press': -1, 'Double':0, 'Contain':1})
team_attributes['defenceTeamWidthClass'] = team_attributes['defenceTeamWidthClass'].map({'Narrow': -1, 'Normal':0, 'Wide':1})
team_attributes['defenceDefenderLineClass'] = team_attributes['defenceDefenderLineClass'].map({'Cover':0, 'Offside Trap':1})
```

Complex Time Series



Additional Data Set, compiled by Ricardo Moya, from Kaggle:

<https://www.kaggle.com/ricardomoya/football-matches-of-spanish-league>