Executive Summary: Exploratory Data Analysis

Tayo Obafaiye

BANA 780: Advanced Business Analytics

Professor Vic Perotti

February 21, 2021

Introduction

Through Exploratory Data Analysis (EDA), we predict the best strategy for a soccer club to be successful. The results, limitations and recommendations are presented in this report. We investigate the relationship between goal-related variables and points accumulated over a league season. The data used for this problem comes from the 2018/2019 season of the English Premier League (EPL). It is a soccer competition involving soccer clubs in England & Wales. The EPL is widely regarded as the most popular soccer league competition in the world and generates a sizeable TV revenue both domestically and internationally.

Description of data

The first dataset contains statistics about all the 380 soccer games played in the EPL in the 2018/2019 season. There are 62 different variables in this dataset. We would be focusing on the Team variables as well as Full Time Home Team Goals (FTHG) and Full-Time Away Team Goals (FTAG).

The second dataset is essentially the final table/log of teams at the end of the season. It is sorted according to their final league positions at the end of the season. This dataset was modified to include the Points total the teams obtained at the end of the season. The second file contains 45 variables/columns and essentially the 'Teams' and the 'Points' variables are most important for the EDA.

Code and Results

A new variable 'Goal_Difference' is created which is the difference between 'Total_Goals_Scored' and 'Total_Goals_Conceded'.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as sm
import warnings
warnings.filterwarnings('ignore')

#Read files/data
df1 = pd.read_csv("C:/Users/Tayo Obafaiye/Desktop/BANA 780/Data/season-1819_csv.csv")
df2 = pd.read_csv("C:/Users/Tayo Obafaiye/Desktop/BANA 780/Data/epl_1819.csv")

#df1.info()
#df2.info()

#Select and rename variables for analysis
df3 = df1[['HomeTeam','FTHG']]
df3 = df3.rename({'HomeTeam': 'Team', 'FTHG': 'Goals_Scored'}, axis=1)

df4 = df1[['AwayTeam','FTAG']]
df4 = df4.rename({'AwayTeam': 'Team', 'FTAG': 'Goals_Scored'}, axis=1)

df5 = df1[['HomeTeam','FTAG']]
df5 = df5.rename({'HomeTeam': 'Team', 'FTAG': 'Goals_Conceded'}, axis=1)

df6 = df1[['AwayTeam','FTHG']]
df6 = df6.rename({'AwayTeam': 'Team', 'FTHG': 'Goals_Conceded'}, axis=1)

frame1 = [df3, df4]
df7 = pd.concat(frame1, ignore_index=True)
df8 = df7.groupby('Team')[['Goals_Scored']].agg(['sum'])

frame2 = [df5, df6]
df9 = pd.concat(frame2, ignore_index=True)
df10 = df9.groupby('Team')[['Goals_Conceded']].agg(['sum'])

df11 = pd.merge(df8, df10, on='Team', how='inner')

df12 = df2[['Team','general_league_position', 'Points']]

df11 = df11.rename({'Man City': 'Manchester City', 'Man United': 'Manchester United', 'Wolves': 'Wolverhampton'}, axis=0)

finaldf = pd.merge(df12, df11, on='Team', how='inner')
finaldf.columns = ['Team', 'Final_League_Position', 'Points', 'Total_Goals_Scored', 'Total_Goals_Conceded']
finaldf['Goal_Difference'] = finaldf['Total_Goals_Scored'] - finaldf['Total_Goals_Conceded']
finaldf.head(1)

#print(finaldf['Points'].describe())
```

Out[1]:

	Team	Final_League_Position	Points	Total_Goals_Scored	Total_Goals_Conceded	Goal_D
0	Manchester City	1	98	95	23	

In [6]:
```
print(finaldf.info())
print(finaldf['Points'].describe())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20 entries, 0 to 19
Data columns (total 6 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Team                   20 non-null     object
 1   Final_League_Position  20 non-null     int64 
 2   Points                 20 non-null     int64 
 3   Total_Goals_Scored     20 non-null     int64 
 4   Total_Goals_Conceded   20 non-null     int64 
 5   Goal_Difference        20 non-null     int64 
dtypes: int64(5), object(1)
memory usage: 1.1+ KB
None
count    20.000000
mean     53.450000
std      21.007455
min      16.000000
25%      39.750000
50%      51.000000
75%      67.000000
max      98.000000
Name: Points, dtype: float64
```

In [2]:
```
#correlation
finaldf.corr()['Points']
```

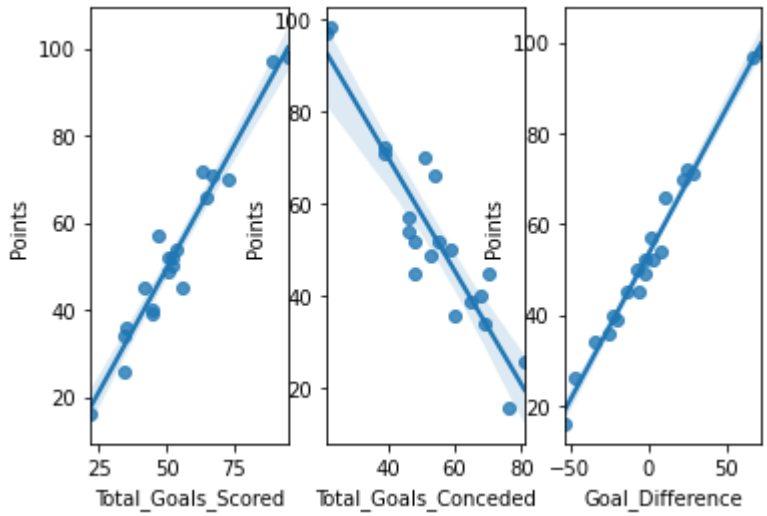
Out[2]:
```
Final_League_Position   -0.953902
Points                   1.000000
Total_Goals_Scored       0.971112
Total_Goals_Conceded    -0.922043
Goal_Difference          0.990814
Name: Points, dtype: float64
```

In [3]:
```
#Visualization
List = ['Total_Goals_Scored', 'Total_Goals_Conceded', 'Goal_Difference', 'Poin
ts']

fig, ax = plt.subplots(1, 3)

for i, ax in enumerate(fig.axes):
    if i < len(List) - 1:
        sns.regplot(x = List[i], y = 'Points', data=finaldf, ax=ax)

plt.show()
```



In [4]:
```
#regression models
#result1 = sm.ols(formula="Points ~ Total_Goals_Scored", data=finaldf).fit() | R-squared: 94.3%
#result2 = sm.ols(formula="Points ~ Total_Goals_Conceded", data=finaldf).fit() | R-squared: 85%
result = sm.ols(formula="Points ~ Goal_Difference", data=finaldf).fit()
result.summary()
```

Out[4]:

OLS Regression Results

Dep. Variable:	Points	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.981
Method:	Least Squares	**F-statistic:**	966.2
Date:	Sun, 21 Feb 2021	**Prob (F-statistic):**	4.28e-17
Time:	19:32:23	**Log-Likelihood:**	-48.749
No. Observations:	20	**AIC:**	101.5
Df Residuals:	18	**BIC:**	103.5
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	53.4500	0.653	81.896	0.000	52.079	54.821
Goal_Difference	0.6430	0.021	31.084	0.000	0.600	0.686

Omnibus:	0.858	**Durbin-Watson:**	1.482
Prob(Omnibus):	0.651	**Jarque-Bera (JB):**	0.708
Skew:	-0.023	**Prob(JB):**	0.702
Kurtosis:	2.080	**Cond. No.**	31.6

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Discussion of Results & Recommendations

The three variables under consideration, 'Total_Goals_Scored', 'Goal_Difference' and 'Total_Goals_Conceded', are all highly correlated to 'Points'. From the regression plots, 'Points' has a positive linear relationship with both 'Total_Goals_Scored' and 'Goal_Difference' and a negative relationship with 'Total_Goals_Conceded'. The Regression model was run using statsmodel. The highest correlation and the best linear relationship with 'Points' is obtained using 'Goal_Difference' as the predictor/regressor variable. 'Points' vs 'Goal_Difference' has an R-squared value of 98.6%. The next best relationship is obtained with 'Total_Goals_Scored' with a coefficient of determination of 94.3%. 'Total_Goals_Conceded' explains 85% of the variability in points. 'Total_Goals_Scored', 'Goal_Difference' and 'Total_Goals_Conceded' can somewhat be interpreted as attacking, balanced and defensive strategies. Based on the results obtained, I would recommend a team prioritizes a moderately attacking strategy.

Limitations in problem

This EDA only considers one season of results. We could get different results for other seasons. Also teams have different objectives, financial backing, talent, etc. The alternative would be to use a dataset with more than one season of data for example, 20 years. However the pitfall here is that not all teams are constant over that period i.e. the bottom three teams drop down to the lower division. Hence after EDA on this dataset, we would end up with a small sample of teams to base the training set on.

Conclusion

This is a practical situation and a sporting team can make future plans including coaching choice, talent recruitment, and organizational philosophy based on this analysis. We can conclude that for a team to stay in the competition and avoid being relegated, it should opt for a moderately attacking strategy. Sport analysts likely work on more sophisticated and specialiazed stats than the ones covered in this problem but the EDA performed in this problem provides a broad and perhaps quick solution that can be further looked into.

Glossary

Div = League Division; Date = Match Date (dd/mm/yy); Time = Time of match kick-off; HomeTeam = Home Team; Away team = Away Team; FTHG and HG = Full Time Home Team Goals; FTAG and AG = Full-Time Away Team Goals; FTR and Res = Full-Time Result (H=Home Win, D=Draw, A=Away Win); HTHG = Half Time Home Team Goals; HTAG = Half Time Away Team Goals; HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win);

Match Statistics (where available) Attendance = Crowd Attendance; Referee = Match Referee; HS = Home Team Shots; AS = Away Team Shots; HST = Home Team Shots on Target; AST = Away Team Shots on Target; HHW = Home Team Hit Woodwork; AHW = Away Team Hit Woodwork; HC = Home Team Corners; AC = Away Team Corners; HF = Home Team Fouls Committed; AF = Away Team Fouls Committed; HFKC = Home Team Free Kicks Conceded; AFKC = Away Team Free Kicks Conceded; HO = Home Team Offsides; AO = Away Team Offsides; HY = Home Team Yellow Cards; AY = Away Team Yellow Cards; HR = Home Team Red Cards; AR = Away Team Red Cards; HBP = Home Team Bookings Points (10 = yellow, 25 = red); ABP = Away Team Bookings Points (10 = yellow, 25 = red).

Data sources

https://datahub.io/sports-data/english-premier-league (https://datahub.io/sports-data/english-premier-league)
https://www.kaggle.com/thesiff/premierleague1819 (https://www.kaggle.com/thesiff/premierleague1819)