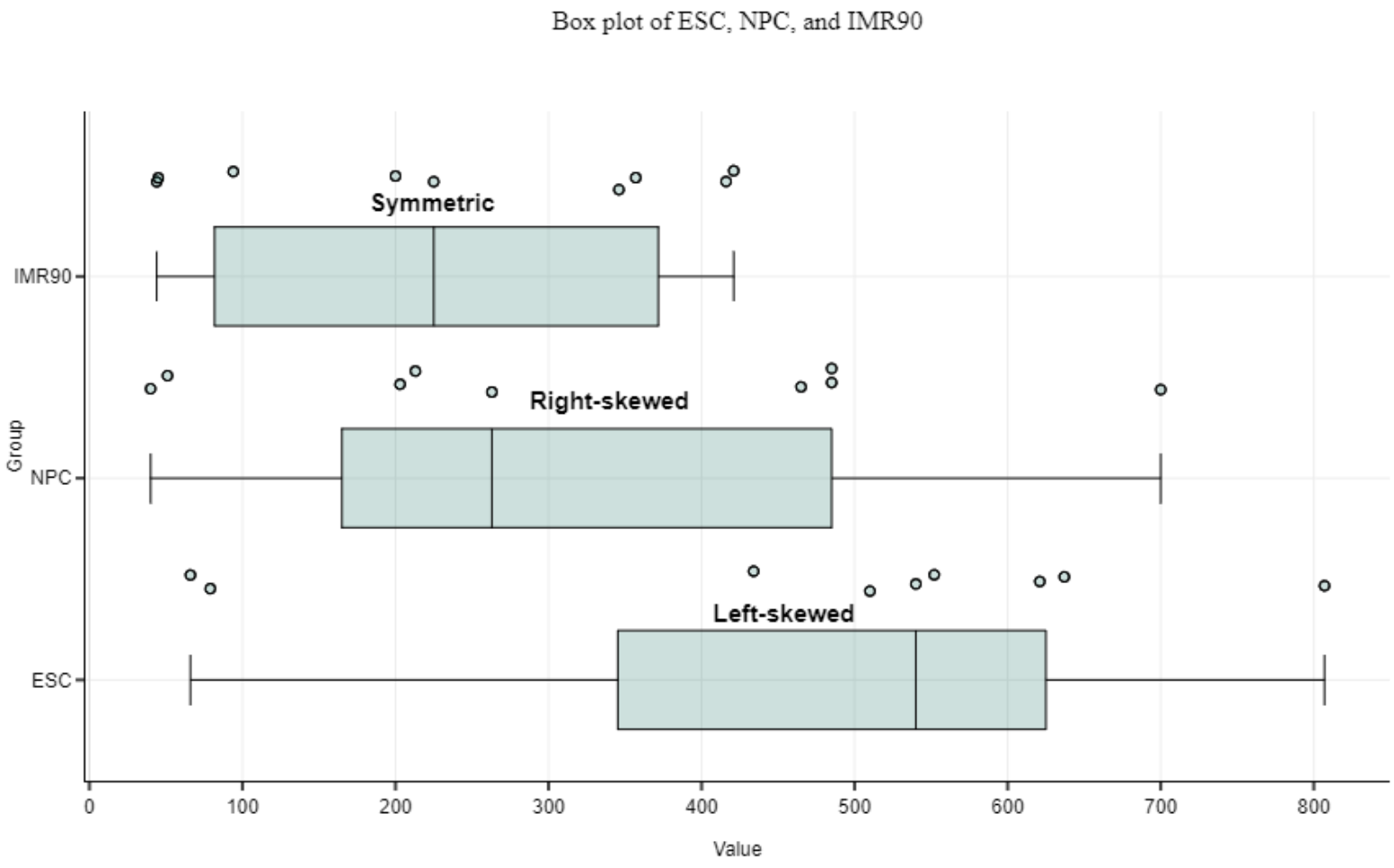
Yoga Sri Varshan V

CED18I058

May, 2022

ASBD Lab End Sem Report



A comprehensive report on the Brazilian Football League dataset from Kaggle!

Question:

For the Given dataset, apply apt data pre-processing techniques to clean the data for further processing. Exploit the concepts discussed in Descriptive Statistics that relate to the data set to gain key insights from the data. Adopt a thorough exploratory data analytics approach, relating the various concepts and plots discussed in the course / tested in the lab assignments to gain key insights from the given data set. On the Pre-processing and EDA front adopt an exhaustive approach relating the maximum no of techniques / features under each set. Over the cleaned data set, apply the following algorithms.

Algorithm 1: FP-growth

Algorithm 2: Naive Bayes Classification or Regression

Algorithm 3: k-medoids Clustering

Dataset Name: (1) Brasil Soccer League Dataset
https://www.kaggle.com/macedojleo/campeonato-brasileiro-2003-a-2019

General Instruction: You shall apply necessary pre-processing techniques like discretization, binning etc to make the dataset suitable for applying FIM algorithm. You may also make any valid assumptions required for the entire exercise and state them explicitly in your documents submitted. Submit a complete report describing the techniques employed, code snippets and corresponding output as done for your lab submissions or share the corresponding notebook link with all data present in the file and mention the dataset name in your answer script. : Based on the type of the assigned dataset, you shall either consider the entire set of features (or) subset of features to generate frequent patterns and apply predictive analytics.

Solution Link -
https://colab.research.google.com/drive/120g1IEww04BME0wQySD5rP_t-tPVhzaU?usp=sharing

Brasil Football League Dataset -

https://www.kaggle.com/datasets/macedojleo/campeonato-brasileiro-2003-a-2019

Contents of the Report:

Preprocessing:

1. Many columns which were unnecessary at first glance were removed.

```
[223]  1  #Removed ID and OBS since they are of no use
       2  df.drop(['Data','ID', 'OBS'], axis=1, inplace=True)
```

```
[224]  1  nRow, nCol = df.shape
       2  print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 6886 rows and 10 columns
```

2. All the columns were renamed from Spanish to English for better understanding.

```
[226]  1  #Renaming columns from Spanish to English
       2  df.columns = ['Year','Round', 'Team 1', 'Team 2','Home Team Goals','Away Team Goals','Home Team State','Away Team State','Winning Team','Arena
```

3. Null rows and duplicated rows were dropped for redundancy.

```
1  #Remove Duplicates
2  df.duplicated()
3  df.drop_duplicates()
```

4. New columns were added, two of which are - Total Goals scored in a match and Home/Away Win(Which team out of home/away won the match?)

```
1  #Creating new columns - Total Goals scored in a match and Home/Away Win(Which team out of home/away w
2  df['Total Goals'] = df['Home Team Goals'] + df['Away Team Goals']
3  df['Total Goals'] = df['Total Goals'].astype(str)
4  df['Home/Away Win'] = df['Winning Team']
```

5. The values of the winning team were changed to the respective names of team 1/team 2/draw from Spanish to English.

```
[253]  1  #Changing the values of the winning team to the respective names of team 1/team 2/draw from Spnish to English
       2  df.loc[df["Home/Away Win"] == "Mandante", "Home/Away Win"] = "Home"
       3  df.loc[df["Home/Away Win"] == "Visitante", "Home/Away Win"] = "Away"
       4  df.loc[df["Home/Away Win"] == "Empate", "Home/Away Win"] = "Draw"
       5  df.loc[df["Winning Team"] == "Mandante", "Winning Team"] = df["Team 1"]
       6  df.loc[df["Winning Team"] == "Visitante", "Winning Team"] = df["Team 2"]
       7  df.loc[df["Winning Team"] == "Empate", "Winning Team"] = "Draw"
```

6. Info about the dataset that is ready for EDA:

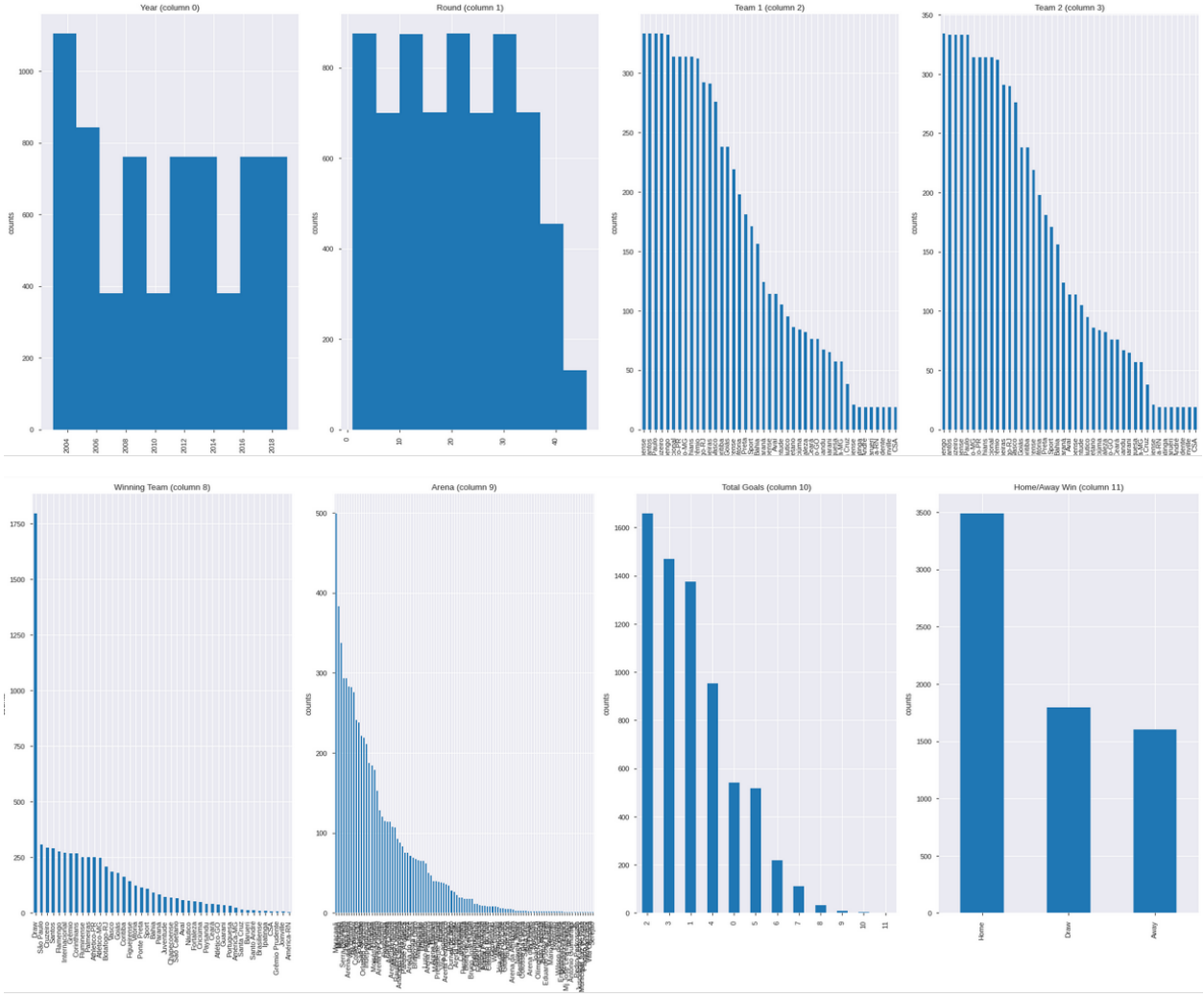
```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Year             6886 non-null   int64
 1   Round            6886 non-null   int64
 2   Team 1           6886 non-null   object
 3   Team 2           6886 non-null   object
 4   Home Team Goals  6886 non-null   int64
 5   Away Team Goals  6886 non-null   int64
 6   Home Team State  6886 non-null   object
 7   Away Team State  6886 non-null   object
 8   Winning Team     6886 non-null   object
 9   Arena            6886 non-null   object
 10  Total Goals      6886 non-null   object
 11  Home/Away Win    6886 non-null   object
dtypes: int64(4), object(8)
memory usage: 699.4+ KB
17
46
43
43
8
8
16
15
44
101
12
3
```

Index	Year	Round	Team 1	Team 2	Home Team Goals	Away Team Goals	Home Team State	Away Team State	Winning Team	Arena	Total Goals	Home/Away Win
0	2003	1	Guarani	Vasco	4	2	SP	RJ	Guarani	Brinco de Ouro	6	Home
1	2003	1	Athletico-PR	Grêmio	2	0	PR	RS	Athletico-PR	Arena da Baixada	2	Home
2	2003	1	Flamengo	Coritiba	1	1	RJ	PR	Draw	Maracanã	2	Draw
3	2003	1	Goiás	Paysandu	2	2	GO	PA	Draw	Serra Dourada	4	Draw
4	2003	1	Internacional	Ponte Preta	1	1	RS	SP	Draw	Beira-Rio	2	Draw
5	2003	1	Criciúma	Fluminense	2	0	SC	RJ	Criciúma	Heriberto Hulse	2	Home
6	2003	1	Juventude	São Paulo	2	2	RS	SP	Draw	Alfredo Jaconi	4	Draw
7	2003	1	Fortaleza	Bahia	0	0	CE	BH	Draw	Castelão	0	Draw
8	2003	1	Cruzeiro	São Caetano	2	2	MG	SP	Draw	Mineirão	4	Draw
9	2003	1	Vitória	Figueirense	1	1	ES	SC	Draw	Barradão	2	Draw
10	2003	1	Santos	Paraná	2	2	SP	PR	Draw	Vila Belmiro	4	Draw
11	2003	1	Corinthians	Atlético-MG	0	3	SP	MG	Atlético-MG	Pacaembu	3	Away
12	2003	2	Fluminense	Fortaleza	1	1	RJ	CE	Draw	Maracanã	2	Draw
13	2003	2	Atlético-MG	Santos	0	0	MG	SP	Draw	Mineirão	0	Draw
14	2003	2	Coritiba	Internacional	0	1	PR	RS	Internacional	Couto Pereira	1	Away
15	2003	2	Grêmio	Guarani	3	1	RS	SP	Grêmio	Olímpico	4	Home
16	2003	2	Bahia	Flamengo	1	2	BH	RJ	Flamengo	Fonte Nova	3	Away
17	2003	2	Figueirense	Corinthians	3	3	SC	SP	Draw	Orlando Scarpelli	6	Draw

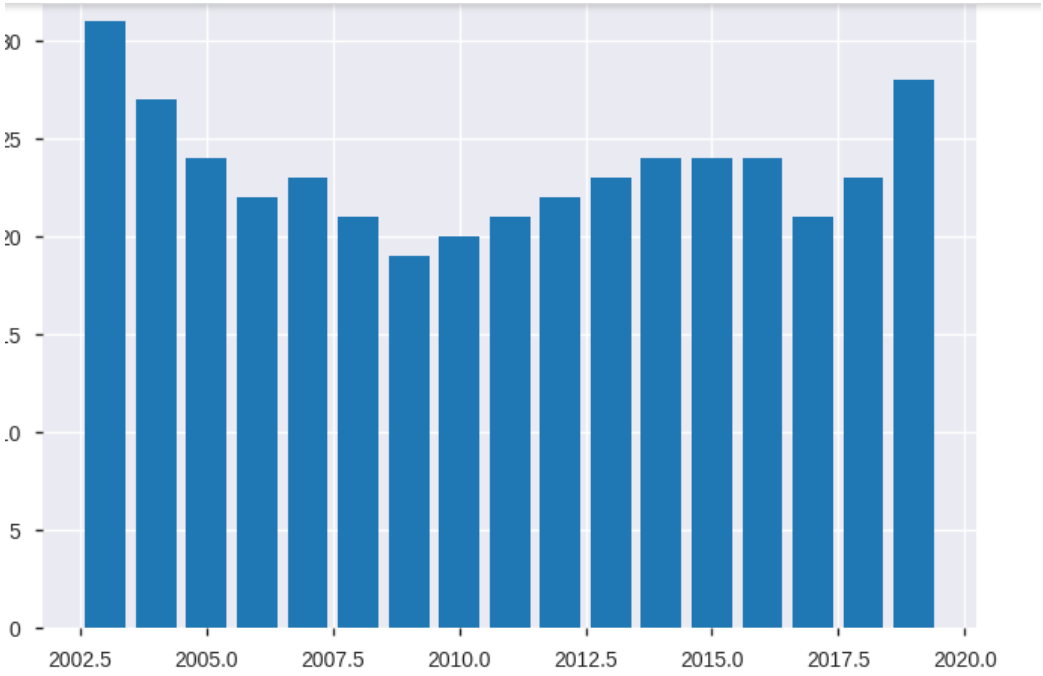
Exploratory Data Analysis

1. An extensive profiling act was taken as a part of the pandas profiling.Detailed report is found at - https://github.com/YogaVicky/Brasil-Football-League-Data-Analysis/blob/main/BrasilFootballLeagueProfiling.pdf
2. Plot per column distribution was done.

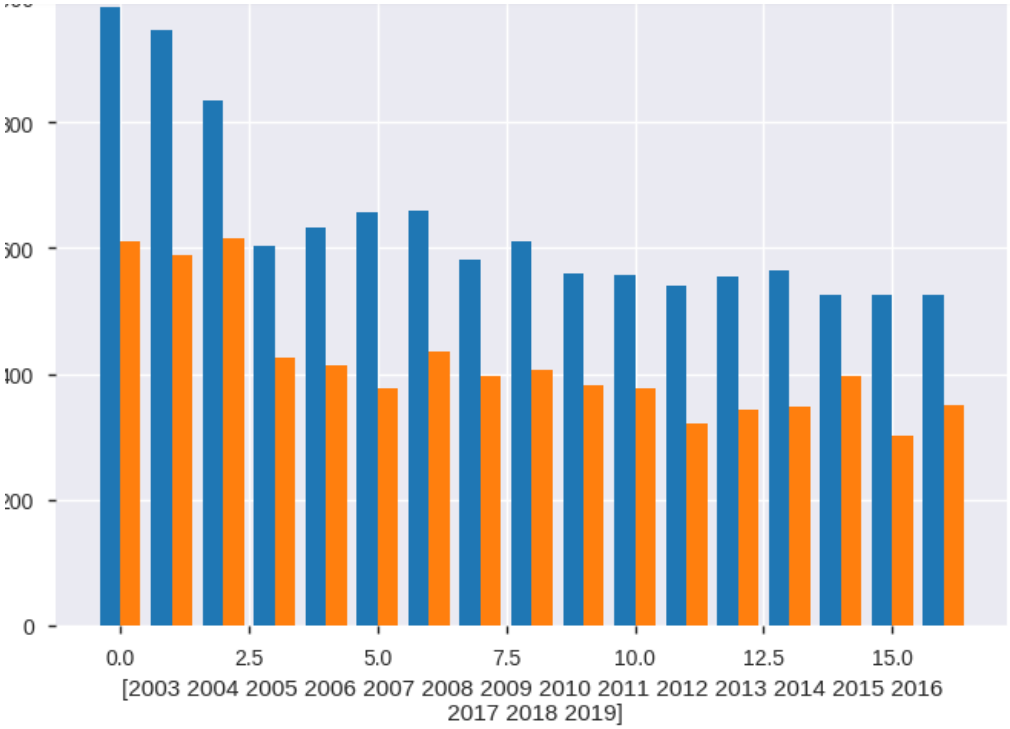
```
#Plot per column distribution
plotPerColumnDistribution(df,14,4)
```



3. A Year wise highest number of wins histogram was plotted(Years from 2003 to 2019).

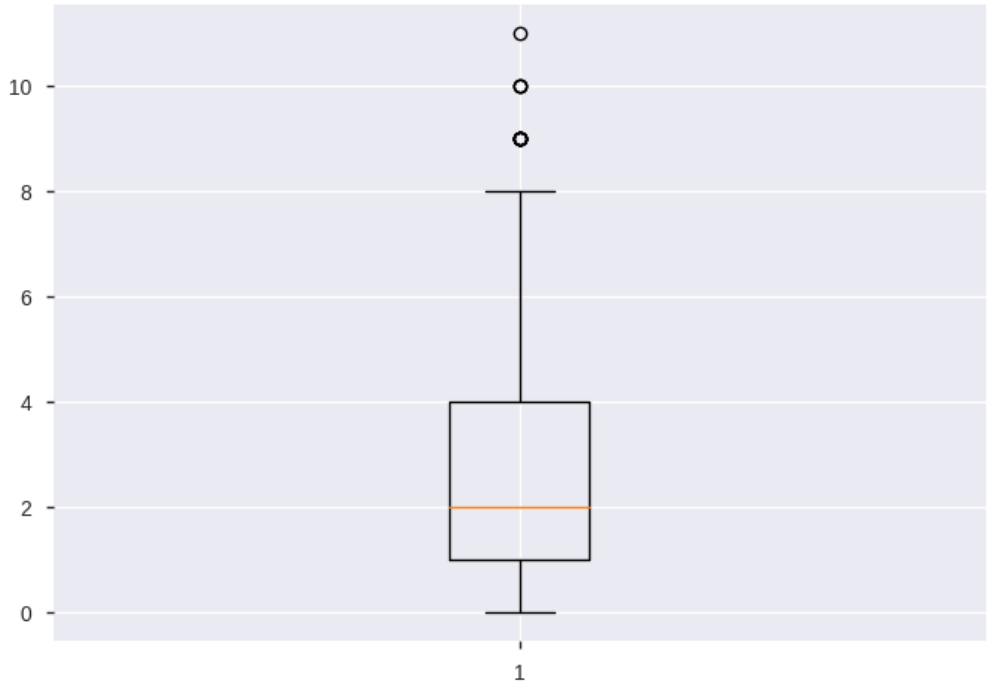


4. Number of Goals - home and away every year was also plotted.



5. Number of wins of every team every Year from 2003 to 2019 to better understand the frequent patterns.

6. Box plots to find outliers in matches regards to Total Goals scored in a match were plotted.



The exploratory data analysis gave a very good feel and touch to the dataset in hand and prompted for further pattern mining and classification tasks.

FP Growth

1. The attributes used are Team 1,Team 2,Winning Team,Arena and Total Goals.
2. The frequent patterns occurring in the dataset containing these attributes specifically are found using FP Growth Algorithm.

```
3  df_fp = df.filter(['Team 1','Team 2','Winning Team','Arena','Total Goals'], axis=1)
4  df_fp
```

	Team 1	Team 2	Winning Team	Arena	Total Goals
0	Guarani	Vasco	Guarani	Brinco de Ouro	6
1	Athletico-PR	Grêmio	Athletico-PR	Arena da Baixada	2
2	Flamengo	Coritiba	Draw	Maracanã	2
3	Goiás	Paysandu	Draw	Serra Dourada	4
4	Internacional	Ponte Preta	Draw	Beira-Rio	2
...	...	...	...	...	...
6881	Vasco	Chapecoense	Draw	Maracanã	2
6882	Botafogo-RJ	Ceará	Draw	Engenhão	2
6883	Avaí	Athletico-PR	Draw	Ressacada	0
6884	Goiás	Grêmio	Goiás	Serra Dourada	5
6885	CSA	São Paulo	São Paulo	Rei Pelé	3

6886 rows × 5 columns

1. The data frame was converted into a list.
2. The list is then converted into encoded transactions.
3. FP growth() was used to calculate the frequent itemset for varied itemsets.

```
[264] 1  #For every min support count,the frequent item sets are generated till we reach empty set by increasing the Support cour
      2  min_support=0.0025
      3  while(1):
      4      min_support*=2
      5      freq_items=fpgrowth(trans_df, min_support=min_support, use_colnames=True)
      6
      7      if(min_support>1.30 or len(freq_items)==0):
      8          if(len(freq_items)==0):
      9              print("\n Current Min_Support = "+str(min_support)+" generates no frequent itemset,Null Set is Reached")
     10              break
     11      print("\n Frequent Itemset with Min_Support of "+str(min_support)+"\n",freq_items)
```

```
 Frequent Itemset with Min_Support of 0.005
      support                            itemsets
0    0.080163                             (Vasco)
1    0.031804                                 (6)
2    0.018879                           (Guarani)
3    0.009730                 (Brinco de Ouro)
4    0.240778                                 (2)
..        ...                                 ...
508  0.008278                 (Chapecoense, 1)
509  0.008423                 (Chapecoense, 2)
510  0.016555       (Arena Condá, Chapecoense)
511  0.015684  (Arena Corinthians, Corinthians)
512  0.007261      (Arena Palmeiras, Palmeiras)

[513 rows x 2 columns]

 Frequent Itemset with Min_Support of 0.01
      support                            itemsets
0    0.080163                             (Vasco)
1    0.031804                                 (6)
2    0.018879                           (Guarani)
3    0.240778                                 (2)
4    0.091200                   (Athletico-PR)
..        ...                                 ...
230  0.010892               (Náutico, Aflitos)
231  0.023236          (Botafogo-RJ, Engenhão)
```

For every min support count,the frequent item sets are generated till we reach the empty set by increasing the Support count.

Inference:

1. Some interesting frequent patterns were found.
2. Patterns like Home team name,Winning team being the home and arena being home arena were common.
3. This denotes an excellent win ratio for home teams in their home arena.

```
510  0.016555       (Arena Condá, Chapecoense)
511  0.015684  (Arena Corinthians, Corinthians)
512  0.007261      (Arena Palmeiras, Palmeiras)
```

Naive Bayes

1. The attributes used are Team 1,Team 2,Home Team Goals,Home/Away Win,Away Team Goals,Arena.
2. Given Team 1,Team 2,Home Team goals,Away Team Goals and Arena,whether a home team will win or an away team will win or will the match result in a draw will be classified by the Naive Bayes Classifier.
3. Here the classified result is the Home/Away Win(Which team out of home and away teams win the match or whether it results in a draw).

	Team 1	Team 2	Home Team Goals	Home/Away Win	Away Team Goals	Arena
0	Guarani	Vasco	4	Home	2	Brinco de Ouro
1	Athletico-PR	Grêmio	2	Home	0	Arena da Baixada
2	Flamengo	Coritiba	1	Draw	1	Maracanã
3	Goiás	Paysandu	2	Draw	2	Serra Dourada
4	Internacional	Ponte Preta	1	Draw	1	Beira-Rio
...	...	...	...	...	...	...
6881	Vasco	Chapecoense	1	Draw	1	Maracanã
6882	Botafogo-RJ	Ceará	1	Draw	1	Engenhão
6883	Avaí	Athletico-PR	0	Draw	0	Ressacada
6884	Goiás	Grêmio	3	Home	2	Serra Dourada
6885	CSA	São Paulo	1	Away	2	Rei Pelé

6886 rows × 6 columns

4. A 80-20 split was followed for preprocessing.

	Team 1	Team 2	Home Team Goals	Home/Away Win	Away Team Goals	Arena
0	Guarani	Vasco	4	Home	2	Brinco de Ouro
1	Athletico-PR	Grêmio	2	Home	0	Arena da Baixada
2	Flamengo	Coritiba	1	Draw	1	Maracanã
3	Goiás	Paysandu	2	Draw	2	Serra Dourada
4	Internacional	Ponte Preta	1	Draw	1	Beira-Rio
...	...	...	...	...	...	...
6881	Vasco	Chapecoense	1	Draw	1	Maracanã
6882	Botafogo-RJ	Ceará	1	Draw	1	Engenhão
6883	Avaí	Athletico-PR	0	Draw	0	Ressacada
6884	Goiás	Grêmio	3	Home	2	Serra Dourada
6885	CSA	São Paulo	1	Away	2	Rei Pelé

6886 rows × 6 columns

5. The Gaussian Naive Bayes Classification Algorithm available in SkLearn Library was used.

```
[276]  1  test_pred=(gnb.predict(test_x))
```

```
[277]  1  # Accuracy of the classifier
       2  NB_Result=accuracy_score(test_y,test_pred)
       3  print(NB_Result)
```

```
0.9013062409288825
```

```
[279]  1  # Recall of the classifier
       2  from sklearn.metrics import recall_score
       3  recall = recall_score(test_y,test_pred, average=None)
       4  print(recall)
```

```
[0.66066066 0.93274854 1.        ]
```

```
[280]  1  # Precision of the classifier
       2  from sklearn.metrics import precision_score
       3  precision = precision_score(test_y,test_pred, average=None)
       4  print(precision)
```

```
[0.99547511 0.73842593 0.96965517]
```

Inference:

1. Given Team 1,Team 2,Home Team goals,Away Team Goals and Arena,whether the home team will win or the away team will win or whether it resulted in a draw was classified as the result.
2. The accuracy of the classification was pretty good.
3. The recall values were decent,but for all the three classes,the precision values were very good.

K Medoids Clustering:

1. The attributes used are Year,Team 1,Team 2,Home Team Goals.

	Year	Team 1	Team 2	Home Team Goals
0	2003	Guarani	Vasco	4
1	2003	Athletico-PR	Grêmio	2
2	2003	Flamengo	Coritiba	1
3	2003	Goiás	Paysandu	2
4	2003	Internacional	Ponte Preta	1
...	...	...	...	...
6881	2019	Vasco	Chapecoense	1
6882	2019	Botafogo-RJ	Ceará	1
6883	2019	Avaí	Athletico-PR	0
6884	2019	Goiás	Grêmio	3
6885	2019	CSA	São Paulo	1

6886 rows × 4 columns

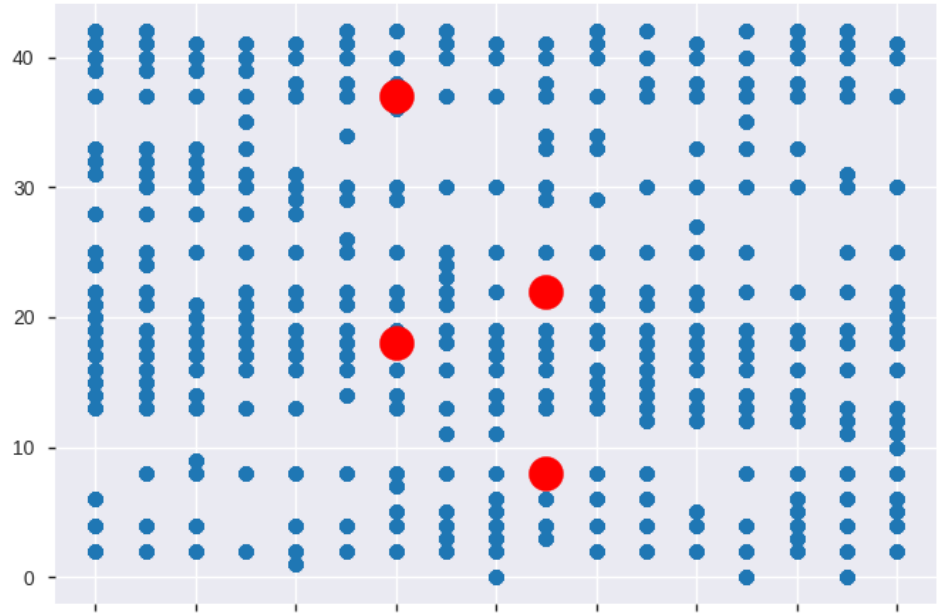
2. The data was categorized and was clustered using the KMedoids clustering module from SkLearn Library - 4 clusters were chosen.

```
# K-Mediods constant of 4 clusters initialised
kmedoids = KMedoids(n_clusters=4, random_state=42).fit(X)
```

```
# Kmed_pred is the result of our clustering
Kmed_pred=kmedoids.fit_predict(X)
```

3. A wonderful cluster plot was made using scatter plots to get a real feel for the clustered data.

```
plt.scatter(X[:,0], X[:,1])
plt.scatter(kmedoids.cluster_centers_[:, 0], kmedoids.cluster_centers_[:, 1], s=300, c='red')
plt.show()
```



Inference:

1. The clustering was decent enough but not robust due to the lack of natural boundaries between the data points.
2. The red ones are the medoids and the circular blues are the data points.

Conclusion:

The Dataset was subjected to the following:

1. Removing the outliers
2. Removing duplicates and Null values
3. Adding of New columns to better make sense of the data - Columns added - Home/Away Win,Total Goals
4. Changing all Spanish to English!!
5. Carrying extensive EDA on the dataset to make great sense and feel of the data in hand.
6. FP Growth to understand the frequently occuring patterns.
7. Naive Bayes Classifier to predict whether the home team will win or the away team will win.
8. K Medoids Clustering on the dataset and effective graph plots.

Overall it was a wonderful experience to work on this amazing dataset :)

< Thank You >