DMDW ASSIGNMENT 1

BACHATE SHAILESH NAVNATH
BT18CSE103

DATASET

I have selected the 'IPL 2020' dataset to work on for this assignment. This dataset is of Indian Premier League 2020, and contains information of all the matches, till the end of league stage.

I am using python libraries like **Numpy, Pandas, Matplotlib** to work on this dataset.

STRUCTURE OF DATA

The dataset has some categorical and some numerical attributes. It contains 56 entries (rows) and 20 columns. The columns are as follows...

Game: Indexes Games from 0 onwards

Stadium: Venue names where matches were played

Team_1: Home Team

Team_2: Away Team

Toss_winner: Name Of Team Who Won The Toss

Toss_decision: What Did The Winning Team Elect To Do, Bat or Field

Match_winner: Winning Team Name

Winner_num: Winning Team Number 1-Team 1 / Home Team; 2-Team 2 / Away Team.

Runs_pp1: Runs In Powerplay for Home Team

Runs_mo1: Runs In Middle Overs for Home Team

Runs_do1: Runs In Death Overs for Home Team

Wkt_pp1: Wickets Lost By Home Team In Powerplay

Wkt_mo1: Wickets Lost By Home Team In Middle Overs

Wkt_do1: Wickets Lost By Home Team In Death Overs

Runs_pp2: Runs Made By Away Team In Powerplay

Runs_mo2: Runs Made By Away Team In Middle Overs

Runs_do2: Runs Made By Away Team In Death Overs

Wkt_pp2: Wickets Lost By Away Team In Powerplay

Wkt_mo2: Wickets Lost By Away Team In Middle Overs

Wkt_do2: Wickets Lost By Away Team In Death Overs

Out of these 20 columns/attributes, 6 (stadium, team_1, team_2, toss_winner, toss_decision, match_winner) are **descriptive** attributes while the other 14 are **numeric** attributes .

LIBRARIES USED

The data is available in csv file, so the python library **'Pandas'** is used to import and read the data. This data is stored in the form of a **DataFrame**(data structure) in a variable 'df.'

To print this data, pandas library has given some functions -

To print the starting n lines, head(n) function can be used. So I have used **df.head(1)** to print the first line.

Similarly the tail(n) method is used to print n lines from the end. So I have used **df.tail(1)** to print the last line.

- Numpy and Pandas provide many features which makes it easier to access the data from the dataset.
    - For e.g. We can get the names of the teams, by getting the unique values of column 'team_1' using **df['team_1'].unique()**. This returns a **Series** data structure, which is similar to a numpy array

- Numpy indexing is pretty useful for selecting specific multiple columns.
    - If we want to check if the toss_winner also won the match, we can do that by printing these two columns in the data using **df[[‘toss_winner’, ’match_winner’]]**.
    - **df.loc[0:5][['team_1', 'team_2']]** will select the teams playing in the first six matches.
- Pandas makes it very easy to combine different data sets or different parts of the same dataset by providing functions like merge, append etc.
    - To select all the matches played by ‘MI’ we have to select all the matches where MI played as team 1 and also the matches where MI played as team 2. So we can append these two results using the append() method to get the required data.
      df[df['team_1'] == 'MI'].append(df[df['team_2'] == 'MI'])
- The selected data can be sorted using methods like **sort_index(), sort_values()**

DATA VISUALIZATION USING MATPLOTLIB

For data visualization, we can use different types of graphs provided by matplotlib.

- A graph of ‘Runs In Powerplay for Home Team’ to ‘Runs In Powerplay for AwayTeam’ will give an idea about how each team fared in the starting overs.
  df[['runs_pp1', 'runs_pp2']].plot()
- We can use other types of graphs as well like **barchart, pie chart, histogram** etc.
- Area chart given by **df[['runs_pp1', 'runs_mo1', 'runs_do1']].plot(kind='area')** will give an idea about how many runs are made during each section of the match by the home team.

- A histogram given by **df[df['team_1'] == 'MI'].append(df[df['team_2'] == 'MI'])['winner_num'].plot(kind='hist')** represent the number of times MI has won and lost the matches.

DATA STRUCTURES USED ON THE DATASET

Here, I have used the two data structures provided by pandas - **DataFrame** and **Series.**
Series is a one-dimensional labeled array and capable of holding data of any type (integer, string, float, python objects, etc.)
Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns)
Other than these, I have also used array and dictionary data structures in the assignment. A data frame can be converted to a dictionary using the **to_dict()** method on the dataframe. And a data frame can also be created from a dictionary