Statistics Collection :

The scripts to get all matches' stats for the current Premier League season is inside 'historicalStats' folder. Open the folder and run the following command:
> scrapy crawl team-stats -o stats.json -t json
A file containing all the stats will be created in the same folder.

Data Cleaning-Data Preprocessing:
Source (raw) data should be placed in folder "Premier_Data". This includes Google Trends Index data about teams (20 files, one for each team), searched by official name as well as scraped match statistics data from www.sportinglife.com . Exported data files are created in "cleaned_premier_data" folder. To execute a script just run the command:

>python <script_name.py>

after accessing the folder containing the script.

- googleIndex_cleaning_merging.py: This script performs multiple jobs on 20 files including Google Trends data. Firstly, it only keeps the relevant data that is considered the Google Trends Index for every week of the last 12 months. Data about location of origin of search, weekly percent change in index etc is excluded from our research. In addition, start and end date of the week are partitioned and a teamNo attribute is added that is used later as a key to correspond team names typed in different ways. A new file is created in "cleaned_premier_data" directory with name "cleaned_gtrends.csv".
- stats_cleaning.py: While preprocessing the data, a functionality issue of scraper was spotted as it did not return values for zero values in statistics. This script finds any missing keys in statistics data and in a new file "cleaned_stats.json" it returns the cleaned data along with the corresponding "teamNo". Another file named "badformat_stats.json" was used to output the matches which did not have the appropriate number of keys/attributes.
- data_merging.py: This script merges the two datasets mentioned above by teamNo and date, matching matchDate in statistics with the corresponding week range containing that date and Google Trends Index for a specific team. The output file is named "merged_data.csv" and is handled in MATLAB to be analyzed, applying Linear regression to decide whether interest Index can be predicted from statistics data.
- classify_stats.py: This final script produces classified features of play as described in the model defining the presentation of the project, according to the mean and standard deviation of the statistics. The outputs of this script are "classified_data.json" , "detailed_classified_data.json" and "classified_data.csv". The last file is used for analysis while the first two were used for visualization.

Data analysis was implemented in file "analysis.m". It includes calculating means and standard deviation for every statistical measure, applying Linear Regression to define a model that could predict Google Trends Index values from match statistics values and test the reliability of the model on a test set. In addition, using

Tweets Collection:

'Tweets' folder contains the python script 'oldMatchesAnalyzer.py' that collects the tweets and classifies them.
The script needs to find an input file 'matches.json' in the same path, an example of the json file could be found in the same folder. It also needs 21 training sets that should be copied from the 'trainingSets' folder to the 'Tweets' folder.

To run the script, python should be installed along with NLTK libraries. Run the script using:
> python oldMatchesAnalyzer.py

The output of the script is a file for each match analyzed. The files should be moved to the 'Webapp' folder for visualization.

Web Application:

The Web application is composed of a backend implemented with python's library flask and a front end with 3 pages coded with html, css and D3.js.
The application code is inside 'WebApp' folder, the backend script is the 'app.py' file and the code for the pages is inside the 'Templates folder.
Once the tweets files, the cleaned stats and the features' file are copied inside the 'WebApp' folder, the application can be run using:
> python3 app.py

To access the pages use the following URLs:

http://localhost:5000/
http://localhost:5000/2/
http://localhost:5000/3/