Introduction to the Project

We Rate Dogs, Twitter Data Analysis.

Project Outline : Access data from various sources, including Tweepy, twitter API, Assessing and Visualization of Data.

Questions Posed Prior to Data Analysis :

- In which language most of tweets are written?
- What are the most common names people give their dogs?
- Considering accumulated data, Which 'dogtype' has the highest presence in the data set? (Doggo/Floofer/Pupper/Puppo)
- Given the complete data set, What is the avg. ratio of Favorite Count to Retweet Counts?
- Visualize and assess the difference in values of these two columns.

Importing Necessary libraries:

- Importing Pandas into our Python Notebook, for reading and writing our csv,tsv or api fetched Json files.
- Importing Matplotlib for performing visualizations over our Analyzed data.
- Importing WordCloud Library for creating word cloud of certain data sets.

Gathering Data :

- Manually saving and reading given ‘twitter enhanced csv’ file using,df. read_csv() fn in dataframe,’df’.
- Mechanically fetching ‘image prediction tsv’ file using ‘requests’ library from the given URL, and saving it into a dataframe named ‘dfimage’.
- Manually reading and saving the given ‘twitter_json.txt’ file into ‘dftweet’, due unavailability of twitter developers account.
- Providing Python code for fetching data using Tweepy, for the above step.

Assessing Data :

- Now, Beginning with, twitter_enhanced_archive.csv file, we assess data using .info() fn which gives us a brief idea about the df containing the above file in tabulated form. Further, .head() and .describe() functions help us in getting a better understanding of data and its consistency & relevance of every column/row values.
- Moving to our second file stored mechanically in the same directory using ‘requests’, as ‘image_prediction.tsv’, we perform the above operations and take a glance at the dfimage dataframe.
- Last but not least, dftweet, a data frame for ‘tweet_json.txt’ file, gets operated over the above spoken functions, lastly giving us a complete idea of all three datasets provided for this project.

Cleaning Data :

- First, we save a copy of above made all three data frames , df, dfimage, dftweet in ‘[df name]_unclean’ data frames to retain the original values provided before altering data of these dataframes, for further references.
- Now, starting off with our cleaning, selecting the first dataframe, df and dropping all these irrelevant columns, 'in_reply_to_status_id','source', 'in_reply_to_user_id','retweeted_status_user_id','retweeted_status_timestamp','expanded_urls'
  to minimize data redundancy.

- Now, looking at the four columns with values related to same field, ‘dogtype’ and considering most of these values contained ‘None’ in all four columns , so we create a new column named df[dogtype] which consists of values related to every dog’s type such as : doggo/floofer/pupper or puppo.
- Next, we fix all data redundancy of newly made columns such as replacing None with ‘undefined’ so as to make better sense of undefined dog type for a dog with no particular type.

- Once. fixed all above redundancies , **we drop original given columns** named ‘doggo’ , ‘floofer’ , ‘pupper’ and ‘puppo’.
- Now, we address the issue of data irrelevance, since we deal with retweet status and counts of tweets, we engineer the df with all the values with retweet_status_id not equal to ‘None’, thereby **dropping all the irrelevant rows** as well.
- Given the id of tweets in our last acquired Json file, has a different column name **’id’** to our previous df column names **‘tweet_id’** making it tedious to operate across these dataframes. We **rename the above** given column to our desired column name.
- Now, Moving to our second dataframe with image prediction data of given tweets, inconsistent data was found across all three prediction columns named p1,p2,p3 and hence needed engineering, here we change values of all these columns to lower case alphabets there by giving a definite consistency across values in these columns.
- Finally, we move towards our last dataframe and clean it for irrelevant columns by dropping them in our cleaning process.
- Having completed all above cleaning steps we, merge all these dataframes into one dataframe with all relevant and clean data for prediction. We merge all three dataframes on tweet_id value, it being primary key in all these data frames.

Storing Data :

- We save the all three clean data frames into their respective clean .csv files, and merged clean dataframe into twitter_archive_master.csv file, giving us a clean and combined data for this analysis.
- tweet_enhanced_cleaned.csv
- tweetapi_cleaned.csv'
- image_prediction_cleaned.csv
- twitter_archive_master.csv

  Are the names of each clean and combined data frames.