WRANGLE REPORT

The Twitter Archive Master wrangled dataset was derived from the three (3) Datasets, The Twitter Archive Enhanced TSV file, The Image Prediction CSV file and The Twitter Json Text file all containing over 5000 Tweets Data.

The Wrangling was necessary before going to Exploratory Analysis because 12 issues (9 Quality and 3 Tidiness) was found in the three (3) Datasets after The Datasets were physically and programmatically assessed.

These include:

Quality issues

1. The retweets are not part of the analysis as they do not have original ratings and dog images.
2. Timestamp is in wrong datatype.
3. Suggested breed needed on data.
4. Rating numerator is inconsistent and not uniform.
5. Not all algorithm prediction are dogs.
6. 'Tweet id' and 'in reply status id' is in wrong datatype.
7. The source needs to be unambiguous.
8. Not all values in Name Column are Dog's.
9. retweet count, favorite count, confidence level, rating numerator and rating denominator in wrong data

Tidiness issues

10. Having three separate Dataset seems not tidy, We'll merge all three Datasets into one and call it df
11. Dog stages repetition as Doggo, Floofer, Pupper, Puppo.
12. retweeted status Id, retweeted status user Id and retweeted timestamp are null

The first issue is there are three different datasets which needs to be combined into a single dataset data frame

```
#merge Image Prediction df_2 and twitter archive dataset df_1
#this merger is called df_4
df_4=df_1.merge(df_2, on ='tweet_id', how='left')
#merge the twitter json data df_3
```

#and the df_4(image prediction + twitter archive)
#to get a combined dataset named df
#merge on the tweet id
df= df_4.merge(df_3, on = 'tweet_id', how='left)

The second issue was, that of Dog Tweets, Dog ratings, Dog Images etc do not need anything like retweets, as retweets do not contain some salient data like original ratings and dog Images. The rows containing the null values for the retweets i.e The Retweeted status ID, Retweeted Status User ID and the Retweeted status Timestamp were retained using the
df_1=df_1[df_1['retweeted_status_id'].isna()].

The third issue was that, not all the data contained in the dataset were dogs as shown from the three (3) algorithms used.

For this, the XOR Operator ,
"(df_2['p1_confirmation'] ^ (df_2['p2_confirmation'] ^df_2
['p3_confirmation']))"
Was used to combine the three algorithms to return only those algorithms that has at least a True as its confirmation.

The fourth issue was that the Timestamp Datatype was in object which is not a standard for Time. The pandas to date-time function
"pd.to_datetime(df_1['timestamp'])"
Was used to convert the datatype to date time.

The Fifth issue was the a breed needs to be derived from the three (3) algorithms and aggregate confidence level, so a suggested breed was derived with a function definition,

"suggested_breed = []

confidence_lvl = []

def image(df_clean):

```
    if df_clean['p3_dog'] == True:
        suggested_breed.append(df_clean['p1'])
        confidence_lvl.append(df_clean['p1_conf'])
    elif df_clean['p2_dog'] == True:
        suggested_breed.append(df_clean['p2'])
        confidence_lvl.append(df_clean['p2_conf'])
    elif df_clean['p3_dog'] == True:
        suggested_breed.append(df_clean['p3'])
        confidence_lvl.append(df_clean['p3_conf'])
    else:
        suggested_breed.append('Nan')
        confidence_lvl.append('Nan')

#series objects having index the image_prediction_clean column.
df_clean.apply(image, axis=1)
#create new columns
df_clean['suggested_breed'] = suggested_breed
df_clean['confidence_lvl'] = confidence_lvl"
```

The sixth issue was that the Source Platform from which these tweets emanate from needs to be clear. For this,

" df_1.source.str.replace('href tag', 'body')"

Was used to extract the source devices from the ambiguous source.

The seventh issue was, Inconsistency in the rating numerators was observed,

#We can pick the decimals in the ratings using a regex

"rating = df_clean.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)

rating.columns = ['rating_numerator', 'rating_denominator']"

The Eighth issue was, retweeted status Id, retweeted status user Id and retweeted timestamp are null and as such they need to be dropped.
"df_clean=df_clean.drop(columns=['retweeted_status_id', 'retweeted_status_user_id','retweeted_status_timestamp'], axis=1)"

Was used to drop the three columns.

The Nineth issue was Doggo, Floofer, Pupper and Puppo are dog stages which need to be combined into one column. So for this,
The none value was first replaced with NaN

"df_clean_clean['doggo']=df_clean_clean['doggo'].replace(to_replace='None', value=np.nan)

df_clean_clean['floofer']=df_clean_clean['floofer'].replace(to_replace='None', value=np.nan)

df_clean_clean['pupper']=df_clean_clean['pupper'].replace(to_replace='None', value=np.nan)

df_clean_clean['puppo']=df_clean_clean['puppo'].replace(to_replace='None', value=np.nan)"

Then the melt function was used to combine all dog stages into one column

“df_clean_clean=pd.melt(df_clean_clean, id_vars=["tweet_id"], value_name='dog_stage', value_vars=['doggo','floofer','pupper','puppo'])”

The Tenth issue was, retweet count, favorite count, confidence level, rating numerator and rating denominator in wrong data. So the astype() function was used to rectify these.
“df_clean.tweet_id=df_clean.tweet_id.astype(str)
df_clean. favorite count =df_clean. favorite count.astype(int)
df_clean.tweet_count=df_clean.tweet_count.astype(int)
df_clean.confidence_lvl=df_clean.confidence_lvl.astype(float)
”

The Eleventh issue was that not all values in Name Column are Dog's. Several values that are not dog names. All of these observations have lowercase characters, an important pattern that could be used to clean up this field.

First, to check the improper names in the df, the regex keyword was used

“mask = df_clean.name.str.contains('^[a-z]', regex = True)

df_clean[mask].name.value_counts().sort_index()”

then, to remove the dog names lowercase values from the dataset, index of rows where the dog names were lower case.

“lower_dog_name_index = df_clean[df_clean.name.str.islower()].index”

The Twelfth issue was 'Tweet id' and 'in reply status id' was in wrong datatype. Tweet id, in reply to status id etc. should be objects, because they are not intended for any calculations.

#to convert to object datatype

df_clean.tweet_id=df_clean.tweet_id.astype(str)

#to convert to object datatype

df_clean.in_reply_to_status_id=df_clean.in_reply_to_status_id.astype(str)

#to convert to object datatype

df_clean.in_reply_to_user_id=df_clean.in_reply_to_user_id.astype(str

Conclusion:
After wrangling dataset, the dataset seems tidy and of good quality to analyse and study Visuals for insights.