act_report

September 16, 2020

0.1 Wrangling and analyzing WeRateDogs data

WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs is already a popular account, the median number of favourites and retweets for each tweet is about 4000 and 1500, respectively. But popularity is relative and as our high school experience and American celebrity culture confirm, popularity is a long tail state.

0.1.1 Extracting data

The data was provided in one csv and one tsv file along with which required scrapping data from twitter using tweepy.

0.1.2 Analysis

This step involved exploring, getting familiar with the data and finding all Quality and Tidiness issues.

0.1.3 Cleaning

Cleaning the data to remove all Quality and Tidiness issues found in analysis phase.

0.1.4 Visualizations

Communicating findings through visuals for better understanding and pattern displaying.

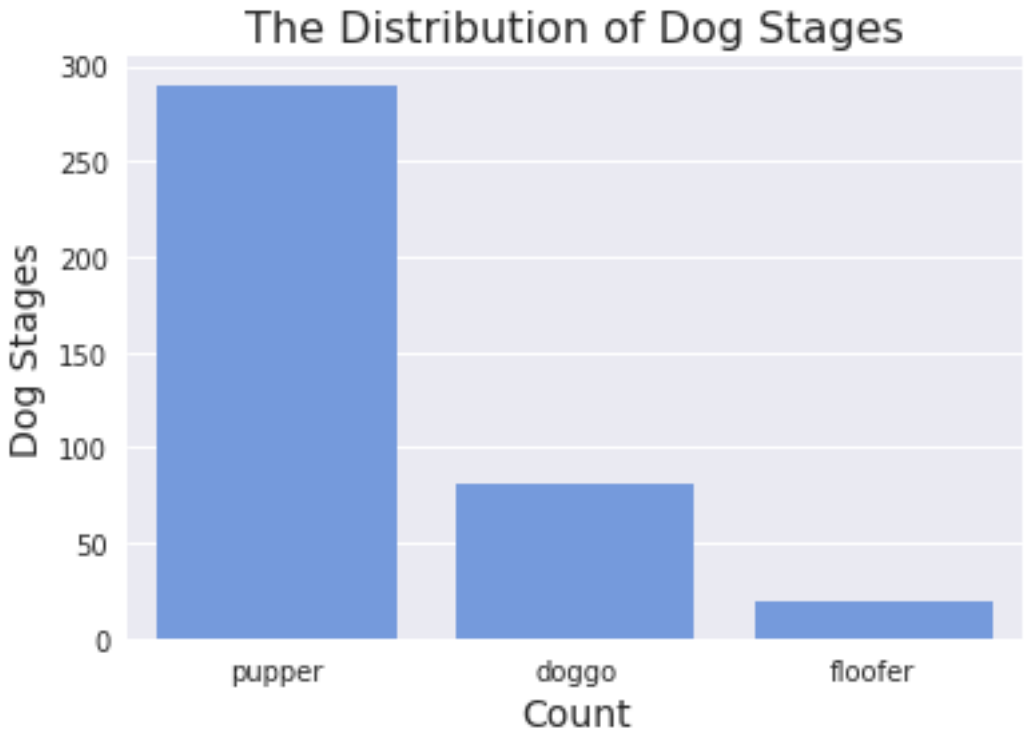
```
In [1]: import pandas as pd
        import numpy as np
        import tweepy
        import matplotlib.pyplot as plt
        import seaborn as sb
        import requests
        import datetime
        import json
        import os
        from IPython.display import Image
```

```
        from IPython.core.display import HTML
        % matplotlib inline
```

```
In [2]: #Reading csv
        image_predictions_clean = pd.read_csv('image_predictions_clean.csv')
        twitter_data_clean = pd.read_csv('twitter_archive_master.csv')
```

```
In [5]: sorted_dog_category = twitter_data_clean['dog_category'].value_counts().head(3).index
        sb.set(style="darkgrid")
        sb.countplot(data = twitter_data_clean, x = 'dog_category',color='cornflowerblue', order
        plt.xticks(rotation = 360)
        plt.xlabel('Count', fontsize=14)
        plt.ylabel('Dog Stages', fontsize=14)
        plt.title('The Distribution of Dog Stages',fontsize=16)
```

```
Out[5]: Text(0.5,1,'The Distribution of Dog Stages')
```

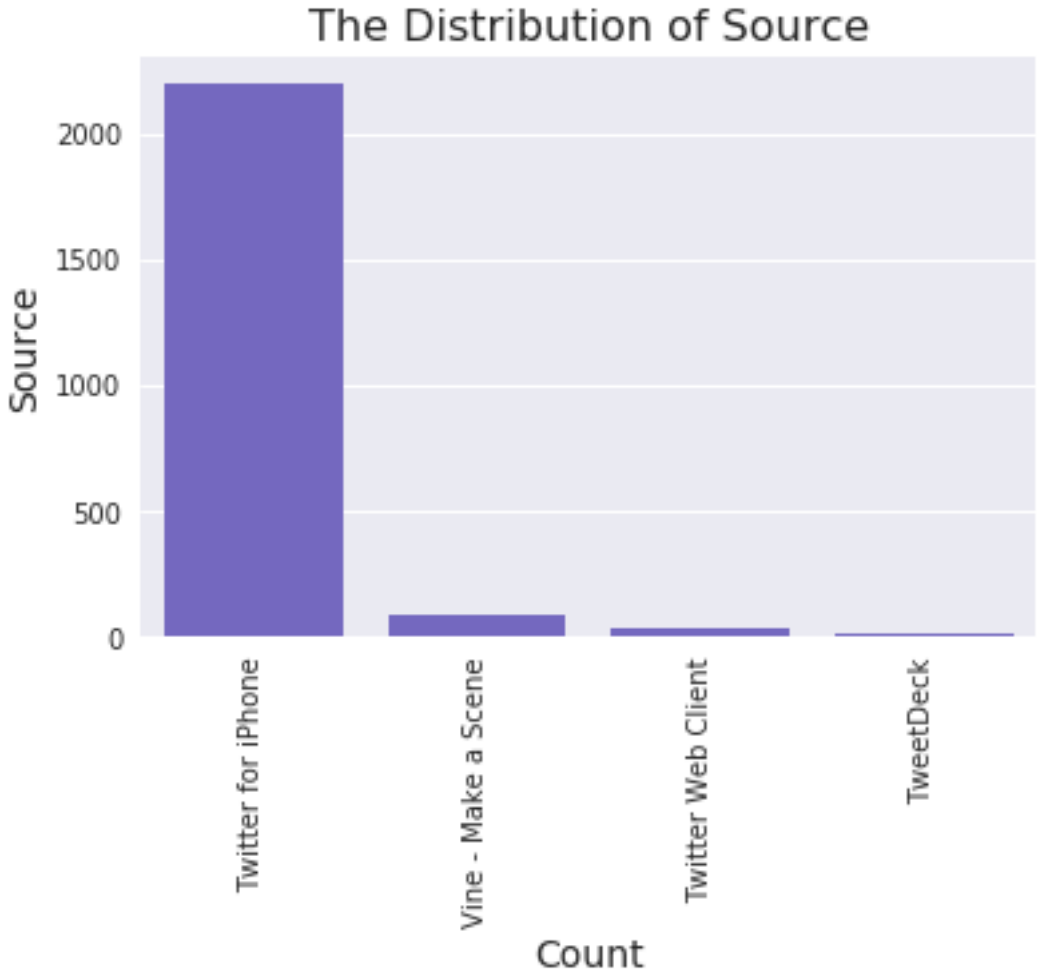


Most of the data has a pupper type of dog_category.

```
In [6]: sorted_source = twitter_data_clean['source'].value_counts().index
        print(twitter_data_clean['source'].value_counts())
        sb.set(style="darkgrid")
        sb.countplot(data = twitter_data_clean, x = 'source',color='slateblue', order = sorted_s
        plt.xticks(rotation = 90)
        plt.xlabel('Count', fontsize=14)
```

```
        plt.ylabel('Source', fontsize=14)
        plt.title('The Distribution of Source',fontsize=16);
```

```
Twitter for iPhone     2197
Vine - Make a Scene      91
Twitter Web Client       33
TweetDeck                10
Name: source, dtype: int64
```



Source for most of the data on twitter is from Twitter for iPhone, and least from TweetDeck

```
In [8]: image_predictions_clean['first_prediction'].value_counts().head(10)
```

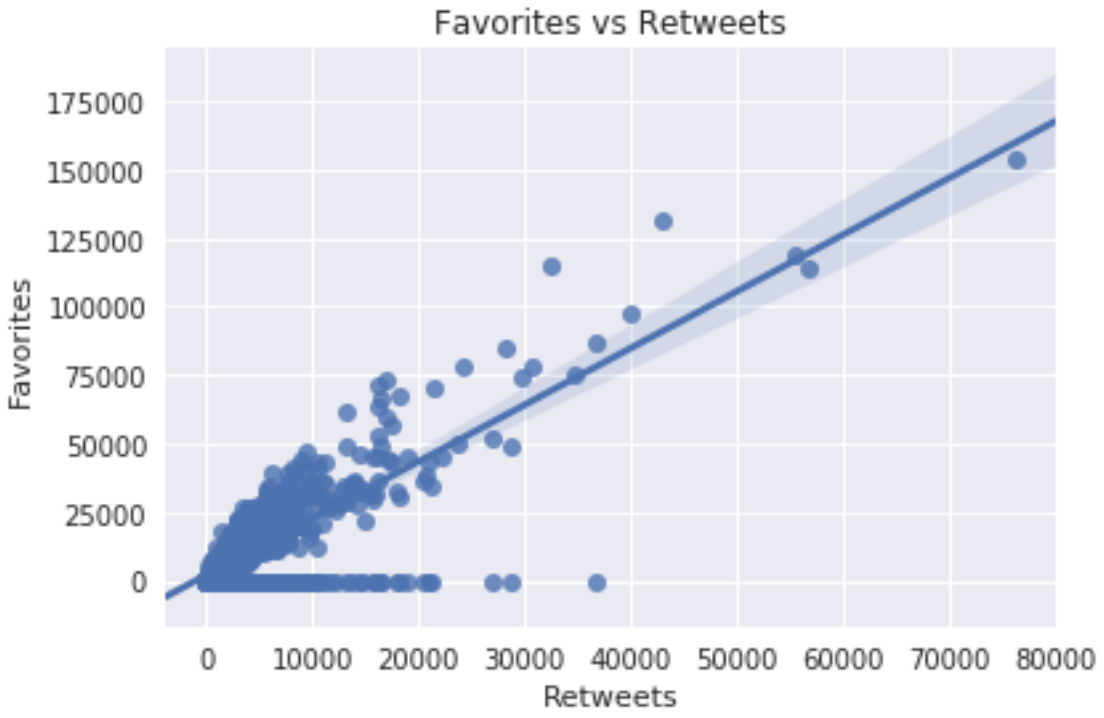
```
Out[8]: golden retriever      150
        Labrador retriever    100
        Pembroke               89
        Chihuahua              83
```

```
pug                     57
chow                    44
Samoyed                 43
toy poodle              39
Pomeranian              38
malamute                30
Name: first_prediction, dtype: int64
```

- The prediction for golden retrievers is the most accurate of other
- Prediction for Chihuahuas looks to be least accurate. The least predictions are recorded for Malamutes and most for Golden Retrievers

Conclusion, this model works in favor of the Golden Retrievers in this dataset

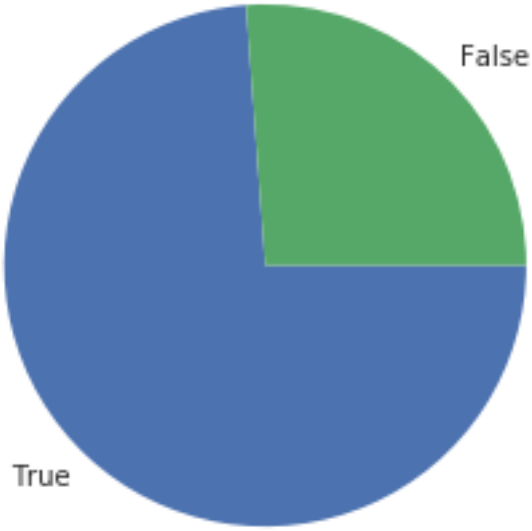
```
In [9]: sb.regplot(x=twitter_data_clean.retweets, y=twitter_data_clean.favorites)
        plt.title("Favorites vs Retweets")
        plt.xlabel('Retweets')
        plt.ylabel('Favorites');
```



Observations:

I tested out the correlation between 'retweets' and 'favorites' and it is clear that it is a positive correlation between them, according to pearson's definition.

```
In [11]: sorted_p1 = image_predictions_clean['first_dog_prediction'].value_counts()
         plt.pie(sorted_p1, labels = sorted_p1.index, counterclock = False,)
         plt.axis('square');
```



In [ ]: The Prediction model's accuracy for dog in first predictions is near to 3/4 or can be sa