Reporting: wragle_report

Project: Wrangling of We Rate Dogs Twitter Data

Table of Contents

Introduction to Wrangling

WeRateDogs is a twitter account that posts dogs photos with a rating of the dog. The data used in this project is a download of the tweets posted by WeRateDogs account. The data contains information about the dog e.g. a photo of the dog, name, breed or 'age group' and rating of the dog. Secondary data also obtained from the data are retweet counts, favorite count. The goal of this report is to gather all the data partaining the WeRateDogs account (there are three different datasets), assesses the data, noting all the issues persent, cleaning the noted issues and perform vaious analysis.

The following are some of the research questions that are to be answered:

- What device is the most used to for tweeting the content on WeRateDogs account?
- What is the relationship between favorite counts and retweet counts?
- What is the relationship between the numerator rating and the favorite counts?

Data Gathering

Three datasets were provided for this project:

- Json file containing raw tweets data extracted using tweepy
- tweet image predictions file containing predictions of all the dog images through a machine learning model
- WeRateDogs Twitter archive file of partialy extracted tweets data

First the following packages were import to aid with the data wrangling and visualisation

In [2]:
```
import tweepy
from tweepy import OAuthHandler
import json
from timeit import default_timer as timer
import pandas as pd
import os
import requests
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import urllib.request
import re
from dateutil.parser import parse
from datetime import datetime
%matplotlib inline
```

1. JSON File

In [3]:
```
##url = 'https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-j
##tweet_json = pd.read_csv(url)
df_list = []
with open(r'C:\Users\Kakai\Dropbox (Personal)\Python\Udacity\tweet-json.txt', 'r') as fi
    for line in file:
        data = json.loads(line)
        df_list.append(data)

df = []
for dct in df_list:
    id_str = dct.get('id_str')
    retweet_count = dct.get('retweet_count')
    favorite_count = dct.get('favorite_count')
    full_text = dct.get('full_text')
    created_at = dct.get('created_at')
    source = dct.get('source')
    df.append([id_str, retweet_count, favorite_count,full_text,created_at,source])
tweets = pd.DataFrame(df,columns =['id_str','retweet_count','favorite_count','full_text'
tweets.head(3)
```

Out[3]:

	id_str	retweet_count	favorite_count	full_text	created_at	
0	892420643555336193	8853	39467	This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU	Tue Aug 01 16:23:56 +0000 2017	href="http://twitter.com rel="nofollow">Twi
1	892177421306343426	6514	33819	This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13/10 https://t.co/0Xxu71qeIV	Tue Aug 01 00:17:27 +0000 2017	href="http://twitter.com rel="nofollow">Twi
2	891815181378084864	4328	25461	This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. You never know when one may strike. 12/10 https://t.co/wUnZnhtVJB	Mon Jul 31 00:18:03 +0000 2017	href="http://twitter.com rel="nofollow">Twi

1. Tweet Image Predictions

In [4]:
```
#image_predictions to imported using requests
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictio
response = requests.get(url)
urllib.request.urlretrieve(url, 'image_predictions.tsv')
```

```
image_predictions = pd.read_csv('image_predictions.tsv',sep='\t')
image_predictions.head(5)
```

Out[4]:

	tweet_id	jpg_url	img_num	p1	p1_co
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.4650
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.5068
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.5964
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.4081
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.5603

1. WeRateDogs Twitter Archive File

In [5]:
```
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive
twitter_archive_enhanced = pd.read_csv(url)
twitter_archive_enhanced.head(2)
```

Out[5]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	sour
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a

Data Accessement

After accessing the data both visually and programatically a number of data quality and tidiness issues were noticed and recorded as follows:

Quality issues

`twitter_archive` `json` **files**

- Some numerator values were wrongly extracted as per the text field e.g tweet with id #680494726643068929 was extracted as 26/10 yet it should be 11.26/10
- Timestamp is not in the correct format, to be converted to a timedate format
- Source column is not clean, contains some html left over special charaters
- Tweet id extracted from json file has data type object and not integer
- in_reply_to_user_id/in_reply_to_status_id columns should have int as their data types and not float

- Some denominator and numerator values need to be transformed to be in the same format as other tweets e.g #677716515794329600 144/120 which should be 12/10
- rating numerator column has outliers, it has values values as big as 1776
- rating denominator column has outliers, it should a 10 across all tweets, has values both less than and more than 10

Tidiness issues

`twitter_archive` **file**

- Source column can be split into two different columns with device and url column being created
- Retweeted tweets are part of the main dataframe, these needs dropping

Data Cleaning

A copy of all the gathered datasets were created to create new datasets to aid with cleaning

In [6]:
```
twitter_archive_enhanced_cp = twitter_archive_enhanced.copy()
tweets_cp = tweets.copy()
image_predictions_cp = image_predictions.copy()
```

All the captured issues were cleaned by first redefining the issues, coding and lastly testing to check if the issues had been fixed. Below are three examples of the issues noted above having been fixed:

- Issue One: Format of the timestap column was not clean enough to be used for analysis, it had a string format instead of datetime
- Issue Three: Source column was split into device and url columns as it was not tidy enough
- Issue Seven and Eight: Here some of the values in rating_numerator column were wrongly extracted while some need transformation to fit the scale

In [7]:
```
#Code
twitter_archive_enhanced_cp['new_created_at'] = [parse(d).strftime('%Y-%m-%d::%H-%M') fo
#transforming the newly created column to datetime format
twitter_archive_enhanced_cp['new_created_at'] = pd.to_datetime(twitter_archive_enhanced_
#Test
twitter_archive_enhanced_cp['new_created_at']
```

Out[7]:
```
0       2017-08-01 16:23:00
1       2017-08-01 00:17:00
2       2017-07-31 00:18:00
3       2017-07-30 15:58:00
4       2017-07-29 16:00:00
               ...
2351    2015-11-16 00:24:00
2352    2015-11-16 00:04:00
2353    2015-11-15 23:21:00
2354    2015-11-15 23:05:00
2355    2015-11-15 22:32:00
Name: new_created_at, Length: 2356, dtype: datetime64[ns]
```

In [8]:
```
#Code
twitter_archive_enhanced_cp['device'] = twitter_archive_enhanced_cp.source.str.split('>'
```

```
#Test
twitter_archive_enhanced_cp['device']
```

Out[8]:
```
0       Twitter for iPhone
1       Twitter for iPhone
2       Twitter for iPhone
3       Twitter for iPhone
4       Twitter for iPhone
              ...
2351    Twitter for iPhone
2352    Twitter for iPhone
2353    Twitter for iPhone
2354    Twitter for iPhone
2355    Twitter for iPhone
Name: device, Length: 2356, dtype: object
```

In [9]:
```
#Code
twitter_archive_enhanced_cp['new_source'] = twitter_archive_enhanced_cp.source.str.split
#Test
twitter_archive_enhanced_cp['new_source']
```

Out[9]:
```
0       http://twitter.com/download/iphone
1       http://twitter.com/download/iphone
2       http://twitter.com/download/iphone
3       http://twitter.com/download/iphone
4       http://twitter.com/download/iphone
                       ...
2351    http://twitter.com/download/iphone
2352    http://twitter.com/download/iphone
2353    http://twitter.com/download/iphone
2354    http://twitter.com/download/iphone
2355    http://twitter.com/download/iphone
Name: new_source, Length: 2356, dtype: object
```

In [10]:
```
#Code
twitter_archive_enhanced_cp['in_reply_to_user_id'] = twitter_archive_enhanced_cp['in_rep
#Test
twitter_archive_enhanced_cp['in_reply_to_user_id']
```

Out[10]:
```
0       0
1       0
2       0
3       0
4       0
       ..
2351    0
2352    0
2353    0
2354    0
2355    0
Name: in_reply_to_user_id, Length: 2356, dtype: int32
```

In [11]:
```
#Code
print(twitter_archive_enhanced_cp.rating_denominator.unique())
twitter_archive_enhanced_cp['rating_denominator'] = twitter_archive_enhanced_cp['rating_
#Test
twitter_archive_enhanced_cp['rating_denominator'].unique()
print(twitter_archive_enhanced_cp.rating_denominator.unique())
```

```
[ 10   0  15  70   7  11 150 170  20  50  90  80  40 130 110  16 120   2]
[10]
```

In [12]:
```
#Code
tweets_cp.rename(columns = {'id_str':'tweet_id'}, inplace = True)
tweets_cp['tweet_id'] = tweets_cp.tweet_id.astype(np.int64)
#Test
tweets_cp.tweet_id
```

Out[12]:
```
0       892420643555336193
1       892177421306343426
2       891815181378084864
3       891689557279858688
4       891327558926688256
               ...
2349    666049248165822465
2350    666044226329800704
2351    666033412701032449
2352    666029285002620928
2353    666020888022790149
Name: tweet_id, Length: 2354, dtype: int64
```

In [13]:
```
#Code
#tweet_ids with rating numerator column that needs transforming/rectifying
tweet_id = [677716515794329600,675853064436391936,682808988178739200,713900603437621249,
674637031882764288,704054845121142784,709198395643068416,680494726643068929,786709082849
#transformed rating columns for the tweets
rating = ['12/10','11/10','12.5/10','11/10','12/10','11/10','12/10','11/10','10/10','12.

df = pd.DataFrame(tweet_id,rating).reset_index().rename(columns = {0:'tweet_id','index':
df['numerator'] = df['rating'].str.split('/').str[0]
df['numerator'] = pd.to_numeric(df['numerator'], downcast = 'float')
df.drop(columns = ['rating'], axis = 1, inplace = True)
#merging the new dataframe with twitter_archive_enhanced_cp
twitter_archive_enhanced_cp = twitter_archive_enhanced_cp.merge(df, how='left', on = 'tw
#Test
twitter_archive_enhanced_cp['rating_numerator'] = twitter_archive_enhanced_cp['rating_nu
#replacing original values with dataset with transformed rating values
twitter_archive_enhanced_cp.loc[twitter_archive_enhanced_cp['numerator'].notnull(), 'rat
twitter_archive_enhanced_cp.drop(columns = ['numerator'], axis = 1, inplace = True)
```

In [14]:
```
#Code
twitter_archive_enhanced_cp_iqr = twitter_archive_enhanced_cp[['rating_numerator']]
Q1 = twitter_archive_enhanced_cp_iqr.quantile(0.25)
Q3 = twitter_archive_enhanced_cp_iqr.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
twitter_archive_enhanced_cp_iqr_clean = twitter_archive_enhanced_cp_iqr[~((twitter_archi
twitter_archive_enhanced_cp_iqr_clean
#Test
twitter_archive_enhanced_cp['rating_numerator'] = twitter_archive_enhanced_cp_iqr_clean
twitter_archive_enhanced_cp['rating_numerator'].unique()
```

Out[14]:
```
rating_numerator    2.0
dtype: float64
array([13.        , 12.        , 14.        ,         nan, 11.        ,
       10.        , 15.        ,  9.75      ,  7.        ,  9.        ,
        8.        , 11.27000046, 12.5       , 11.26000023])
```

In [15]:
```
#Code
print(twitter_archive_enhanced_cp.loc[twitter_archive_enhanced_cp.text.str[:2] == 'RT'].
retweets = twitter_archive_enhanced_cp.loc[twitter_archive_enhanced_cp.text.str[:2] == '
print(twitter_archive_enhanced_cp.shape)
for index in retweets.index:
    twitter_archive_enhanced_cp.drop(index,axis = 0, inplace = True)
#Test
twitter_archive_enhanced_cp.shape
```

Out[15]:
```
(183, 20)
(2357, 20)
(2174, 20)
```

Analyzing & Visualizing Data

Storing Data

Save gathered, assessed, and cleaned master dataset to a CSV file named "twitter_archive_master.csv".

In [16]:
```
twitter_archive_master = twitter_archive_enhanced_cp.merge( tweets_cp, on = 'tweet_id').
twitter_archive_master.head(3)
```

Out[16]:

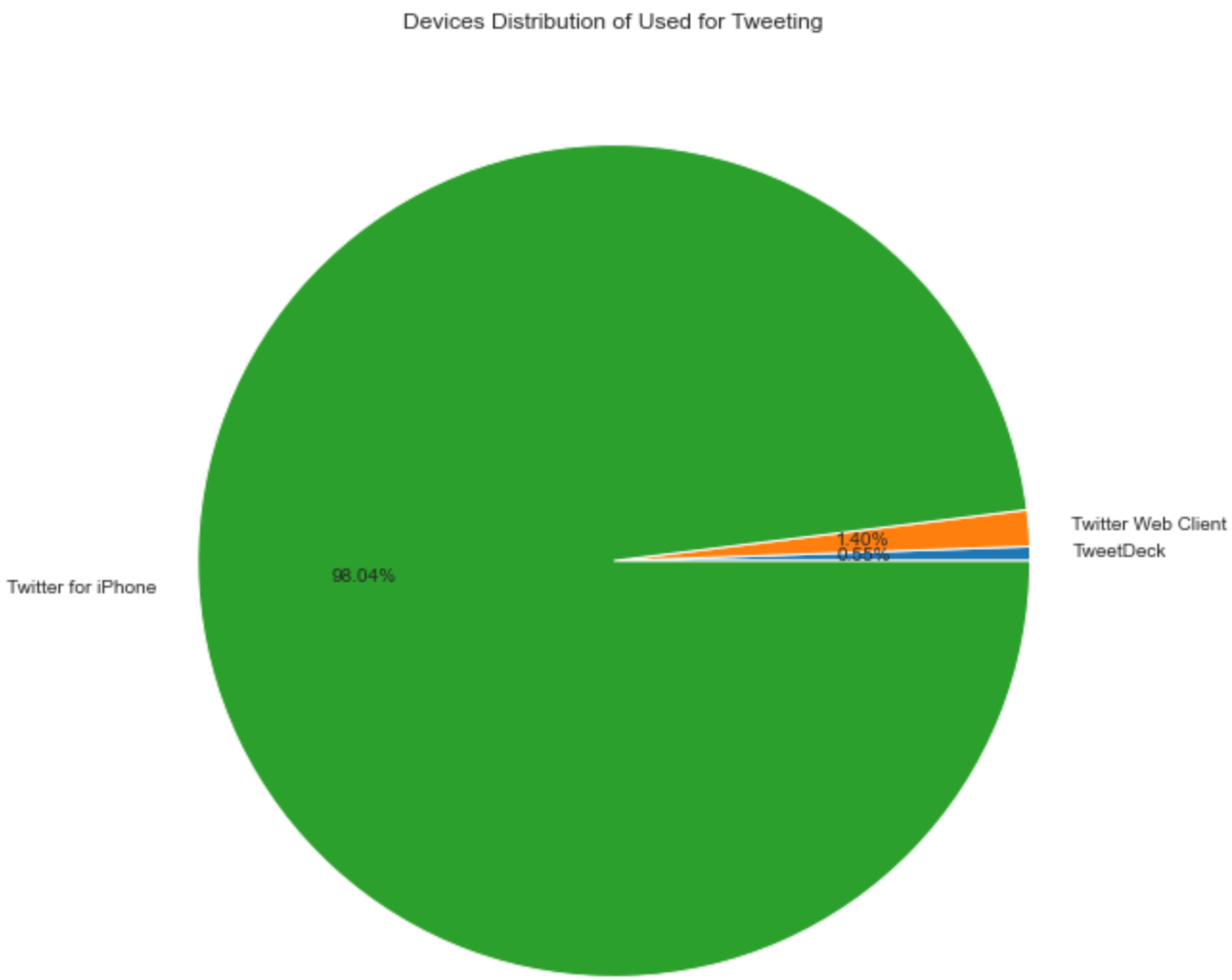
	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source
0	892420643555336193	NaN	0	2017-08-01 16:23:56 +0000	href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
1	892177421306343426	NaN	0	2017-08-01 00:17:27 +0000	href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
2	891815181378084864	NaN	0	2017-07-31 00:18:03 +0000	href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a

Research Questions

- What device is the most used to for tweeting the content on WeRateDogs account?

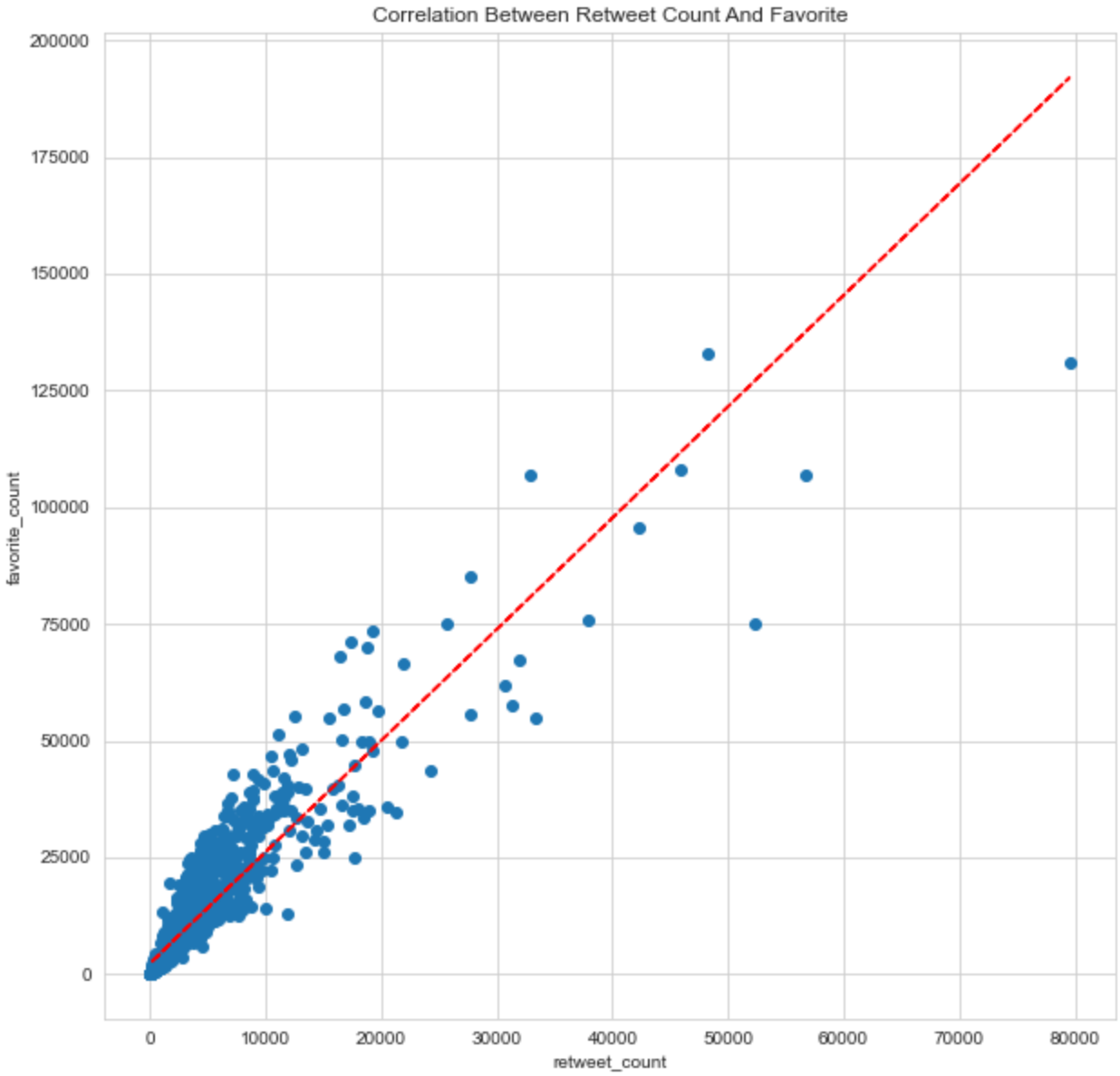
In [17]:
```
devicedistribution = twitter_archive_master.groupby('device')['tweet_id'].count().reset_
print(devicedistribution)
#Visualisation
sns.set_style("whitegrid")
fig = plt.figure(figsize=(10,10))
plt.pie(data = devicedistribution, x = devicedistribution['Count'], labels = devicedistr
plt.title("Devices Distribution of Used for Tweeting")
plt.show();
```

```
              device  Count
0           TweetDeck     11
1  Twitter Web Client     28
2  Twitter for iPhone   1955
```



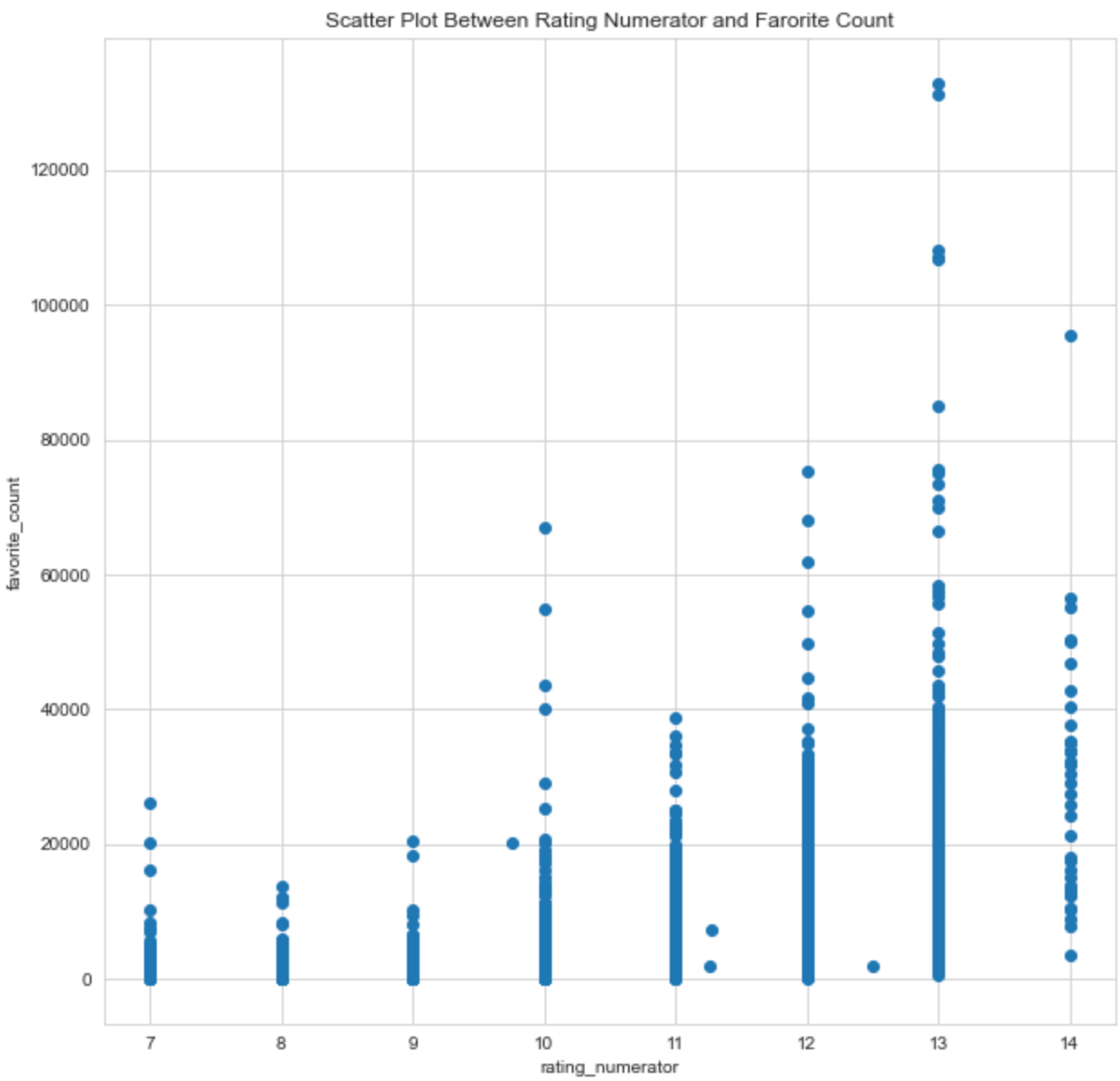
- What is the relationship between favorite counts and retweet counts?

In [18]:
```
fig = plt.figure(figsize=(10,10))
plt.scatter('retweet_count', 'favorite_count', data=twitter_archive_master)
#Adding the aesthetics
plt.title('Correlation Between Retweet Count And Favorite')
plt.xlabel('retweet_count')
plt.ylabel('favorite_count')
#Show the plot
z = np.polyfit(twitter_archive_master['retweet_count'], twitter_archive_master['favorite
p = np.poly1d(z)
plt.plot(twitter_archive_master['retweet_count'],p(twitter_archive_master['retweet_count
plt.show()
```



- What is the relationship between the numerator rating and the favorite counts?

In [19]:
```
fig = plt.figure(figsize=(10,10))
plt.scatter('rating_numerator', 'favorite_count', data=twitter_archive_master)
#Adding the aesthetics
plt.title('Scatter Plot Between Rating Numerator and Farorite Count')
plt.xlabel('rating_numerator')
plt.ylabel('favorite_count')
#Show the plot
plt.show()
```



Conclusions

> Majority of the tweets from WeRateDogs accounts were posted using an Iphone. 98.04% of the tweets have Twitter for Iphone as the device used with the remaining less than two percentage having been posted by TweetDeck (1.40%) and Twitter Web Client (0.56%)
>
> There is a positive correlation between retweet count and favorite count, an increase in retweet count would mean also there would an increase in the favourites counts
>
> Best rated dogs are between ratings 13 and 14. Likewise these dogs received more favorite counts as compared to least rate dogs at 7 which similarly received low counts of retweets and favorite