Overview:

WeRateDogs on twitter is a user that rates people's dogs with a humorous comment about the dog. WeRateDogs downloaded their twitter archive and send to us. I will need to access my twitter api to gather more data regarding retweets and favorite counts for tweet_id in this archive file. Also, I'm provided with an image prediction file which classify breeds of dogs.

I would like to first gather all three datasets from various data sources, then wrangle them with a few data wrangling techniques in Jupyter Notebook.

After that, I explore the clean dataset and conduct a few exploratory data analysis.

Data and Approach:

I import a few libraries for analysis.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

After importing libraries, I load the dataset using pandas.

```
# import twitter_archive_master dataset
df_new = pd.read_csv('twitter_archive_master.csv')
df_new.head()
```

(I) Warm-ups

Since I've gotten the needed data, I can start exploring the data by utilizing plotting techniques. To get started, I plot a histogram for the rating_numerator to check its distribution. From the histogram below (Figure 1), most rating_numerators fall into 11-13.

```
# Plot the histogram of rating_numerator with rating_denominator as 10
df_new[(df_new.rating_numerator < 50) & (df_new.rating_denominator ==
10)].rating_numerator.hist(bins=20);
plt.title('rating_numerator',size=15);
```

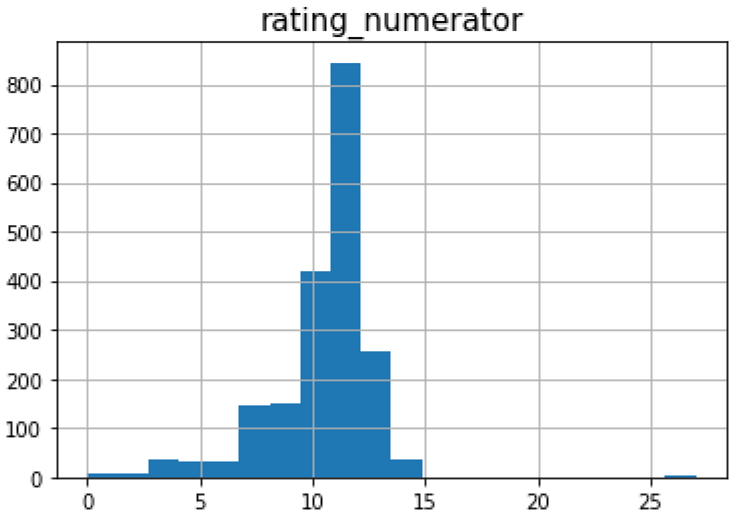

Figure 1: rating_numerator

With stage data created in data wrangling process, we would like to learn the proportions of stages for dogs. To do this, I created a bar chart to reflect this metric. We can know that most dogs with known stages are puppers, followed by doggo, puppo, and floofer.

```
# Show the frequency of different stages
df_new[df_new.stage != 'None'].stage.value_counts().plot(kind='bar',color =
'green',alpha=0.5);
plt.xlabel('stage',size=14);
plt.ylabel('count',size=14);
```

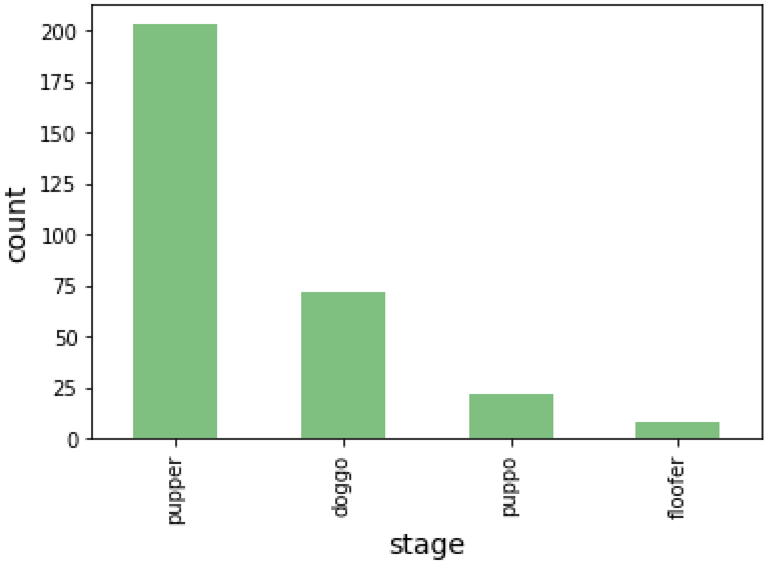

Figure 2: stage

(I I) Exploring multivariate data

To view average rating_numerator for different stages, I created a bar chart (Figure 3). For all dogs with known stages, their average rating_numerators are similar. Looks like dogs who are 'floofer' tend to have the highest average ratings.

```
# Show the average rating for different stages
df_stage_rating = df_new[(df_new.rating_numerator < 50) &
(df_new.rating_denominator == 10) & (df_new.stage != 'None')] #filter in
the non-null stage dogs
df_stage_rating.groupby('stage').rating_numerator.mean().reset_index().plot
(kind='bar',x='stage',y='rating_numerator',legend = False,alpha=0.5);
plt.ylabel('avg rating');
```

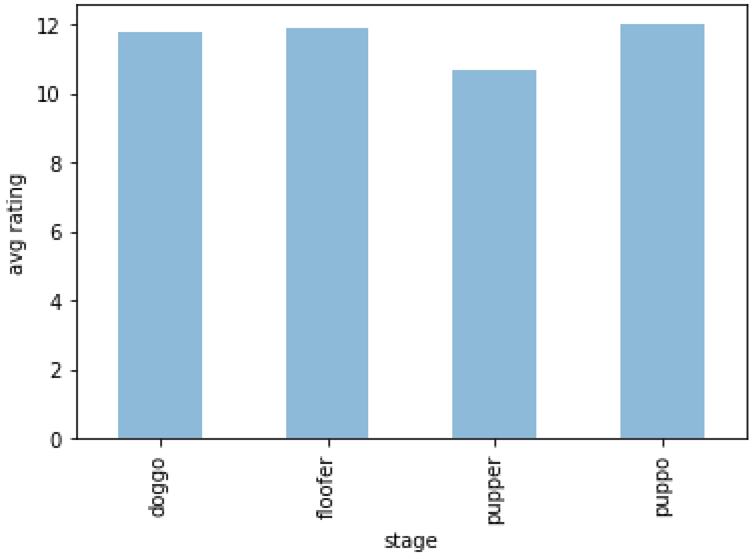

Figure 3: average rating_numerator for stages

The image prediction data has three predictions for each dog image, so we would like to compare the accuracy rate. Basically the most confident prediction is always the first one, so we would assume that the first prediction will have the most outstanding accuracy rate. This is evaluated by proportions of True in columns. I plotted a bar chart to show the results. After investigating the average accuracy rate for three predictions in Figure 4, p2(ie. the second prediction) has the highest accuracy rate compared to others.

```
# Compare the average accuracy rate for p1_dog, p2_dog, p3_dog
pd.Series([df_new.p1_dog.mean(),df_new.p2_dog.mean(),df_new.p3_dog.mean()],
index=['p1','p2','p3']).plot(kind='bar',alpha = 0.8);
plt.xlabel('prediction',size=14);
plt.ylabel('accuracy rate',size=14);
```

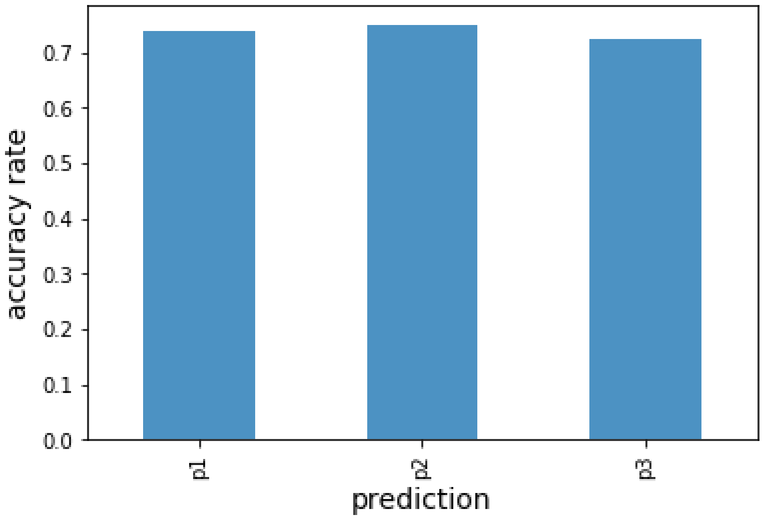

Figure 4: accuracy rate for different predictions

The next steps are to explore relationships between retweet_count, favorite_count, and rating_numerator. We would assume that the higher rating_numerator is, the more retweets or favorites. So scatterplots come into play here. First I plotted retweet_count and rating_numerator. Figure 5 shows that there's a positive relationship between rating_numerator and retweet_count. Which means that the higher rating_numerator, the more likely this tweet will be retweeted.

```
# Plot the relationship between rating_numerator and retweet_count
df_rating_retweet = df_new[(df_new.rating_denominator == 10) &
(df_new.rating_numerator < 50)]
df_rating_retweet.plot(kind='scatter',x='rating_numerator',y='retweet_count
',alpha = 0.4,color='orange',figsize = (8,6));
```

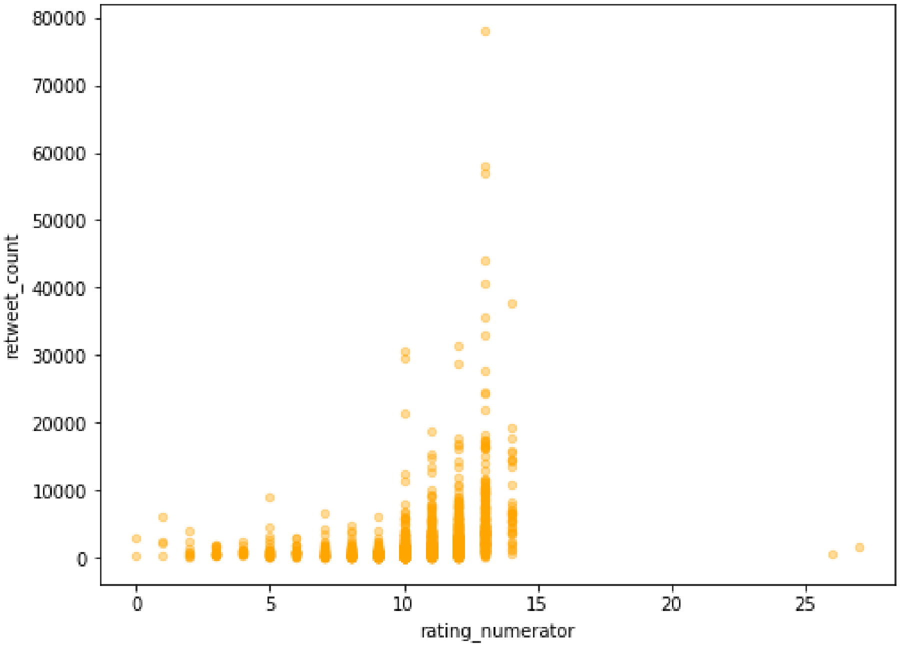

Figure 5: retweet_count vs. rating_numerator

The same for favorite_count and rating_numerator. Like the relationship above, there's a positive relationship between rating_numerator and favorite_count.

```
# Plot the relationship between rating_numerator and favorite_count
df_rating_retweet.plot(kind='scatter',x='rating_numerator',y='favorite_count',a
lpha = 0.4,color='orange',figsize = (8,6));
```

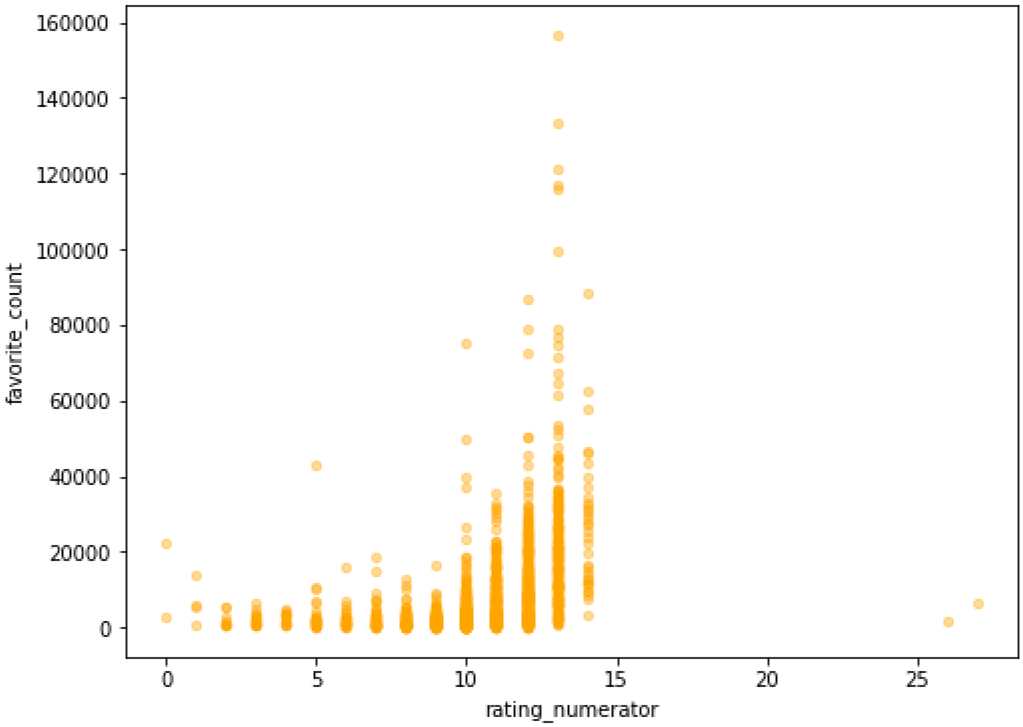

Figure 5: favorite_count vs. rating_numerator

Result

We’ve conducted a few exploratory analysis towards the cleaned dataset. From the analysis above, we can reach a few conclusions.

- Most rating_numerators fall into 11-13.
- Counts by dog names above shows us that the most frequent name used is Oliver.
- Most dogs with known stages are puppers, followed by doggo, puppo, and floofer.
- For all dogs with known stages, their average rating_numerators are similar. Looks like dogs who are 'floofer' tend to have the highest average ratings.
- After investigating average accurate rate for three predictions, p2(ie. the second prediction) has the highest accuracy rate compared to others.
- The scatterplot shows that there's a positive relationship between rating_numerator and retweet_count. Which means that the higher rating_numerator, the more likely this tweet will be retweeted.
- There's a positive relationship between rating_numerator and favorite_count.