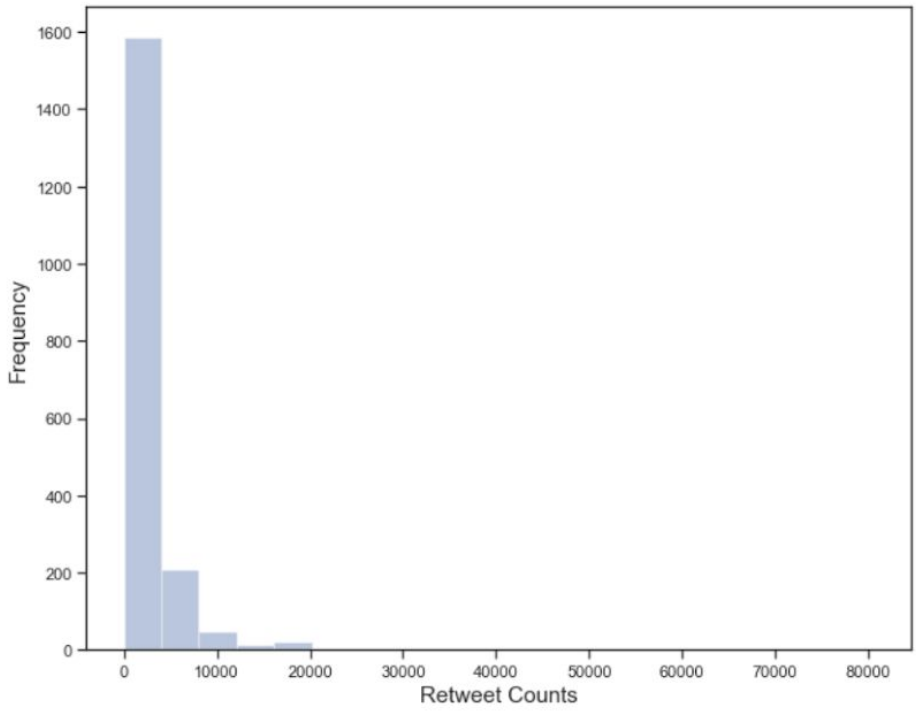
Mariam Joan
August 10, 2019
Udacity Data Analyst Nanodegree
Project: Wrangling and Analyze Data

In the final part of this project we were to store and analyze our tweet data. An in-memory SQLite database from sqlalchemy was used to store our data locally using import create_engine and to_sql to convert our pandas DataFrame to SQL which we can then query.

For analysis, pandas libraries like matplotlib and seaborn were used. In the first visualization, we are viewing retweet count and favorite count separately as a histogram or distplot in seaborn. Both variables show a similar pattern of higher frequencies in the lower values. The next visualization is a bivariate scatter plot of both our counts showing an outlier which is our max of an 80000 retweet count. This plot uses a regression line so we can see the linear relationship between our two variables.

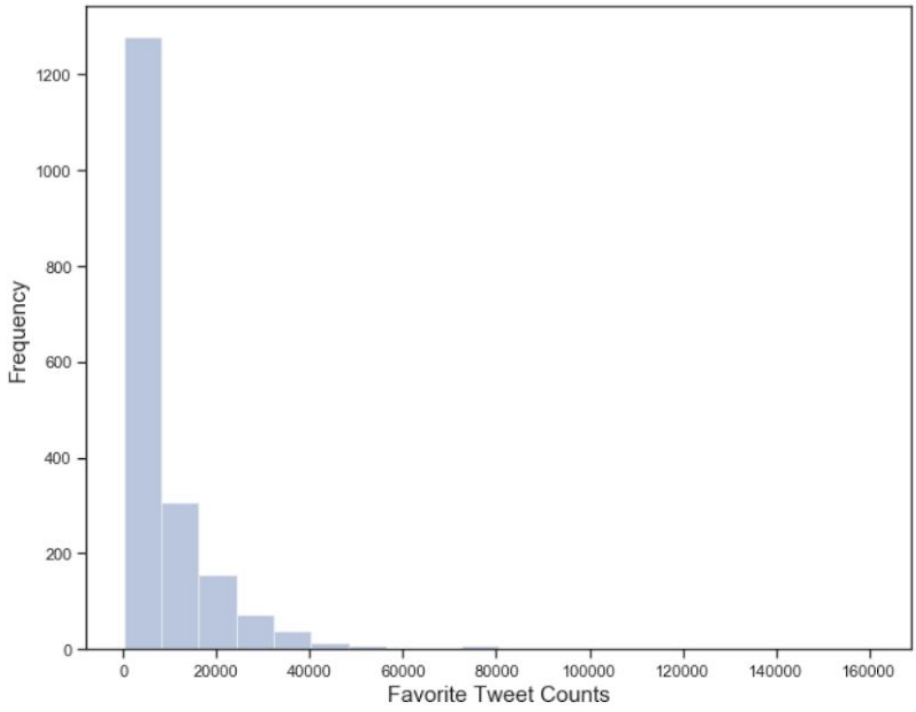
In [313]:
```
plt.figure(figsize=(10, 8))
sns.distplot(master.retweet_count, bins=20, kde=False)
plt.xlabel("Retweet Counts", fontsize=15)
plt.ylabel("Frequency",fontsize=15)
```

Out[313]: Text(0, 0.5, 'Frequency')



In [314]:
```
plt.figure(figsize=(10, 8))
sns.distplot(master.favorite_count, bins=20, kde=False)
plt.xlabel("Favorite Tweet Counts", fontsize=15)
plt.ylabel("Frequency",fontsize=15)
```

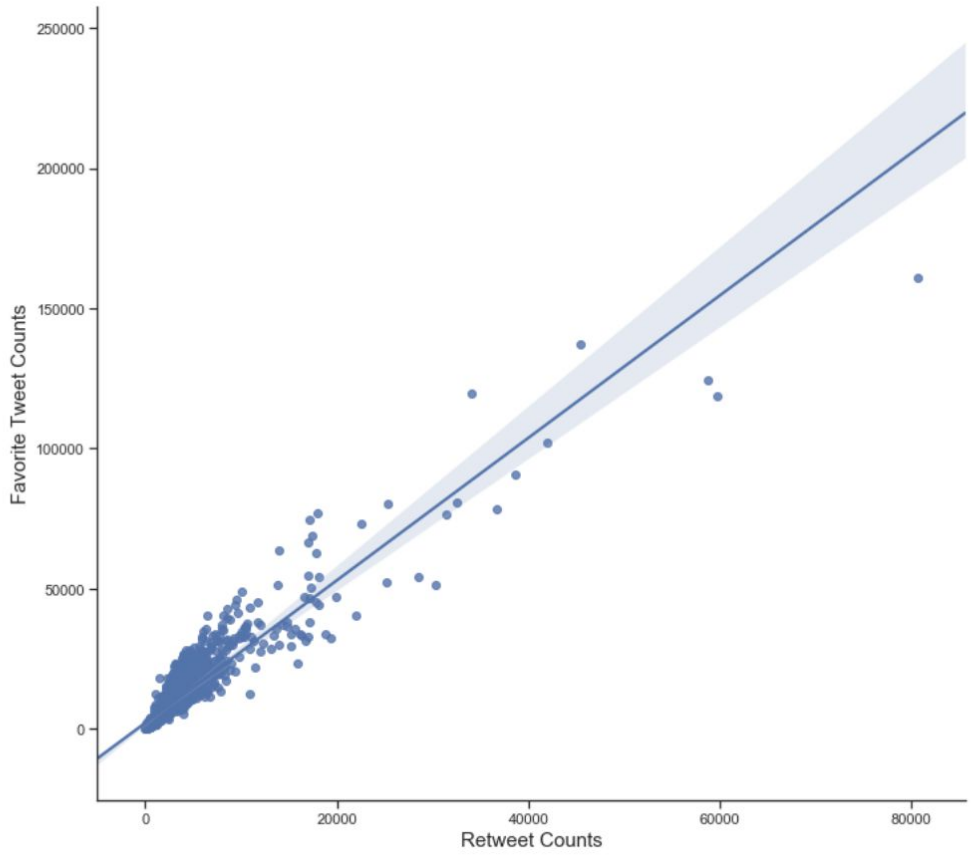
Out[314]: Text(0, 0.5, 'Frequency')



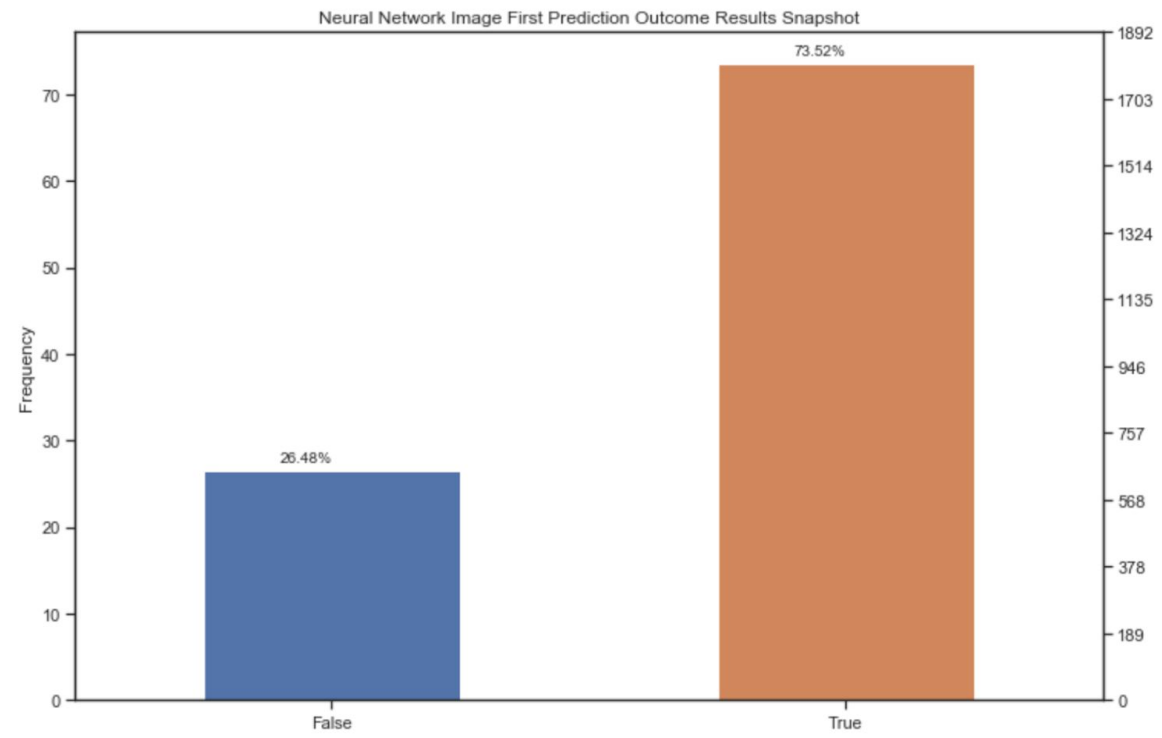
bivariate scatter plot of both counts above to show correlation and outliers

```
In [318]: g=sns.lmplot('retweet_count', 'favorite_count', data=master, fit_reg=True)
          g.fig.set_size_inches(12, 10)
          plt.xlabel("Retweet Counts", fontsize=15)
          plt.ylabel("Favorite Tweet Counts",fontsize=15)

Out[318]: Text(-14.450000000000003, 0.5, 'Favorite Tweet Counts')
```



The next set of visualizations reported in the notebook were focusing on how many times the neural network prediction outcome was true versus false. As you can see with each bar chart image the prediction outcomes were mostly true for first, second and third prediction outcomes. The very first bar chart in the notebook and provided below uses extra logic to attach percentages with each bar so we can see in the first prediction outcome, 73.5% was true versus 26% were false.



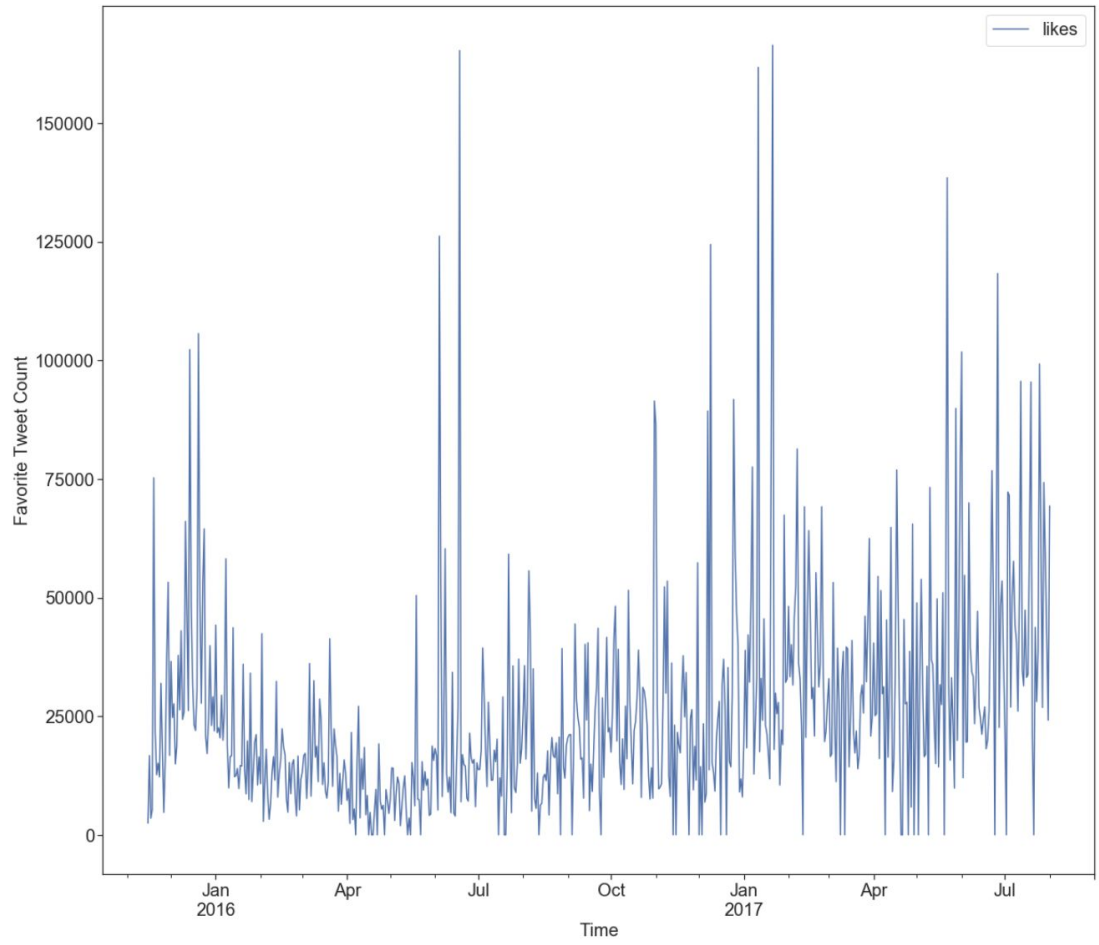
**breakdown of first prediction outcome*

The archive data did show that the neural network sometimes didn't actually predict a dogs name. Sometimes you would see object names because a dog was in a car and so the prediction name was then a car. I wanted to gauge how many times the neural network appropriately guessed an actual dog breed. I found and cited a Github repo that had a basic text file of dog breed names which I converted to a list. From there, I combined all predicted names from our archive data and merged them to a unique set. Then we compared this unique set with actual dog breed names which produced 78 correct matches.

I also wanted to see tweets likes over time, and so I made sure to convert the date column into a datetime data type, and then create a new DataFrame with dates, and tweet likes. With our seaborn visual below we can see that the highest favorited count occurred during mid June 2016.

here we are finally doing a "time-series" on how many tweets occured per day! It seems we had a lot of action the second to third week in June 2016!

In [296]:
```
date_likes.resample('D').sum().plot(figsize=(20, 18), fontsize=20 )
plt.xlabel('Time', fontsize=20);
plt.ylabel('Favorite Tweet Count', fontsize=20);
plt.legend(loc='best', fontsize=20)
plt.show()
```



Lastly, being that we had the tweet text and counts available, I was able to use Python's collections library and the Counter class with most_common function to show from tweet text what the most common words used within our dataset.

```
[('is', 1286), ('This', 1087), ('a', 960), ('to', 607), ('the', 558), ("He's", 446), ('12/10', 434), ('He', 423), ('w
ould', 382), ('11/10', 382), ('10/10', 379), ('for', 321), ('of', 296), ('in', 284), ('13/10', 254), ('and', 223),
('his', 214), ('Meet', 201), ('just', 164), ('this', 164), ('on', 163), ('be', 163), ('with', 158), ('pet', 150), ('y
ou', 148), ('pupper', 146), ('She', 145), ("She's", 143), ('I', 142), ('af', 135), ('af.', 132), ('but', 130), ('9/1
0', 125), ('dog', 117), ('he', 113), ('that', 112), ('at', 109), ('an', 108), ('your', 104), ('her', 103), ('good', 9
7), ('it', 97), ('all', 97), ('very', 94), ('like', 93), ('not', 92), ('hello', 92), ('as', 91), ('was', 90), ('onl
y', 89), ('Say', 89), ('pup', 88), ('8/10', 82), ('has', 81), ('h*ckin', 76), ('RT', 72), ('have', 71), ('here.', 7
1), ('him', 70), ('so', 70), ('dogs.', 70), ('Very', 69), ('by', 68), ('rate', 67), ('are', 67), ('still', 67), ('on
e', 66), ('out', 63), ('We', 59), ('from', 58), ("he's", 58), ('dog.', 57), ('we', 56), ('@dog_rates:', 56), ('up', 5
5), ('about', 54), ("don't", 51), ('pupper.', 51), ("Here's", 49), ('get', 49), ('tongue', 48), ('tho.', 48), ('Not',
48), ('Both', 48), ('doggo', 47), ('no', 47), ('Please', 47), ('&amp;', 47), ('7/10', 47), ('wants', 45), ('can', 4
5), ("doesn't", 44), ('help', 44), ("you're", 43), ('do', 43), ('send', 43), ('him.', 42), ('pup.', 41), ("can't", 4
1), ('Here', 40)]
```

**top 100 words from our tweets using most_common from Counter class*