# Week 9 Deliverables

## Group Details

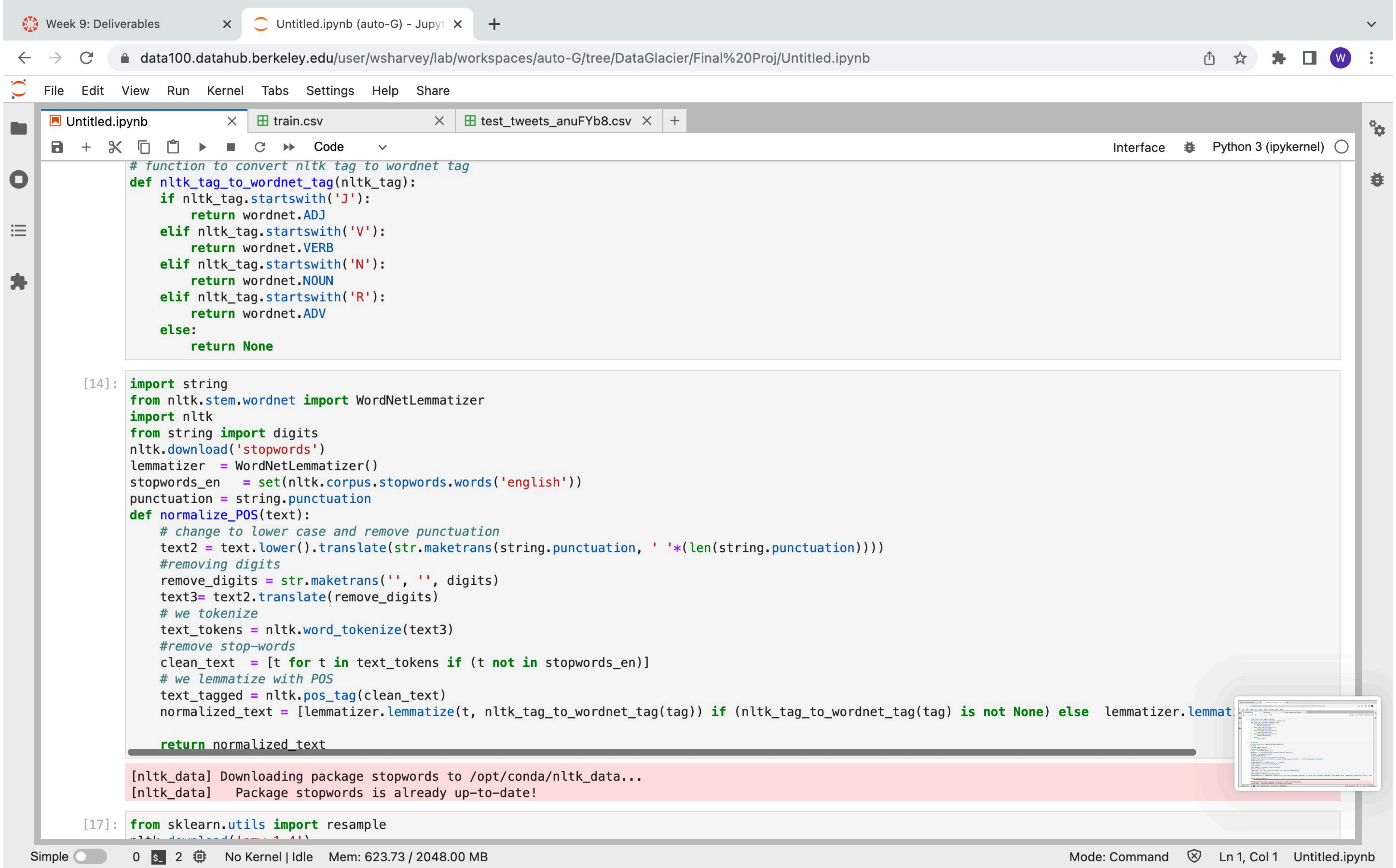
- Name: William Harvey
- Email: will.harvey@att.net
- Country: United States of America
- College: University of California, Berkeley
- Specialization: NLP Analyst

## Problem Description

- Advanced NLP working with detecting hate speech.

# Data Cleansing And Transformation

- I utilized various libraries like regex, ntlk, and other python libraries to clean the text of each tweet in both the training and test dataframes. Once the tweets were clean and tokenized, I used the sklearn "resample" method in order to account for the data imbalance of the given dataframes.


Week 9: Deliverables
Untitled.ipynb (auto-G) - Jupyt
data100.datahub.berkeley.edu/user/wsharvey/lab/workspaces/auto-G/tree/DataGlacier/Final%20Proj/Untitled.ipynb
File Edit View Run Kernel Tabs Settings Help Share
Untitled.ipynb
train.csv
test_tweets_anuFYb8.csv
Code
Interface
Python 3 (ipykernel)
Simple 0 2 No Kernel | Idle Mem: 623.73 / 2048.00 MB
Mode: Command Ln 1, Col 1 Untitled.ipynb

```python
# function to convert nltk tag to wordnet tag
def nltk_tag_to_wordnet_tag(nltk_tag):
    if nltk_tag.startswith('J'):
        return wordnet.ADJ
    elif nltk_tag.startswith('V'):
        return wordnet.VERB
    elif nltk_tag.startswith('N'):
        return wordnet.NOUN
    elif nltk_tag.startswith('R'):
        return wordnet.ADV
    else:
        return None
```

[14]:

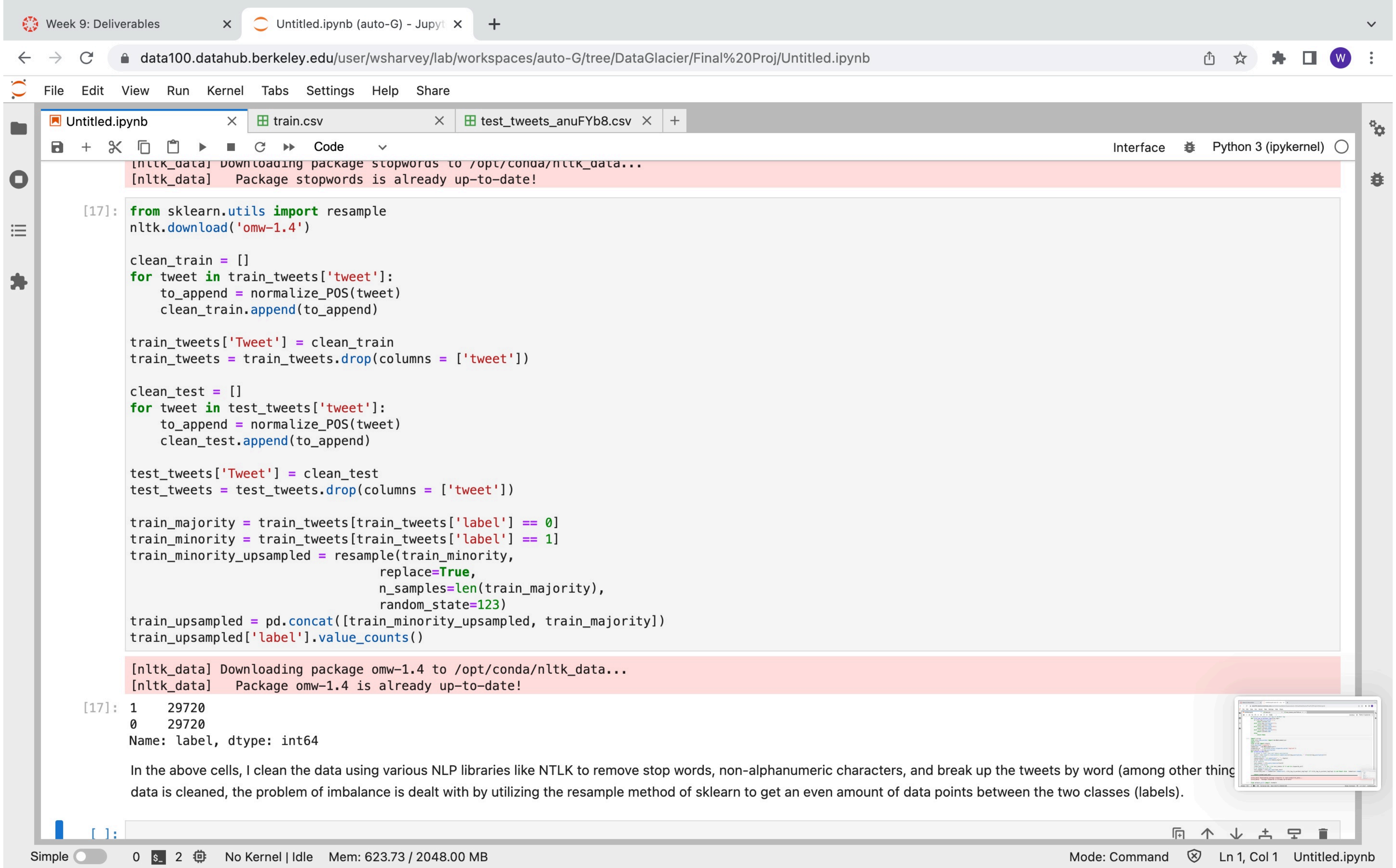
```python
import string
from nltk.stem.wordnet import WordNetLemmatizer
import nltk
from string import digits
nltk.download('stopwords')
lemmatizer  = WordNetLemmatizer()
stopwords_en   = set(nltk.corpus.stopwords.words('english'))
punctuation = string.punctuation
def normalize_POS(text):
    # change to lower case and remove punctuation
    text2 = text.lower().translate(str.maketrans(string.punctuation, ' '*(len(string.punctuation))))
    #removing digits
    remove_digits = str.maketrans('', '', digits)
    text3= text2.translate(remove_digits)
    # we tokenize
    text_tokens = nltk.word_tokenize(text3)
    #remove stop-words
    clean_text  = [t for t in text_tokens if (t not in stopwords_en)]
    # we lemmatize with POS
    text_tagged = nltk.pos_tag(clean_text)
    normalized_text = [lemmatizer.lemmatize(t, nltk_tag_to_wordnet_tag(tag)) if (nltk_tag_to_wordnet_tag(tag) is not None) else  lemmatizer.lemmat

    return normalized_text
```

```
[nltk_data] Downloading package stopwords to /opt/conda/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

[17]:

```python
from sklearn.utils import resample
```


Week 9: Deliverables
Untitled.ipynb (auto-G) - Jupyt
data100.datahub.berkeley.edu/user/wsharvey/lab/workspaces/auto-G/tree/DataGlacier/Final%20Proj/Untitled.ipynb
File Edit View Run Kernel Tabs Settings Help Share
Untitled.ipynb
train.csv
test_tweets_anuFYb8.csv
Code
Interface
Python 3 (ipykernel)

```
[nltk_data] Downloading package stopwords to /opt/conda/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

[17]:

```python
from sklearn.utils import resample
nltk.download('omw-1.4')

clean_train = []
for tweet in train_tweets['tweet']:
    to_append = normalize_POS(tweet)
    clean_train.append(to_append)

train_tweets['Tweet'] = clean_train
train_tweets = train_tweets.drop(columns = ['tweet'])

clean_test = []
for tweet in test_tweets['tweet']:
    to_append = normalize_POS(tweet)
    clean_test.append(to_append)

test_tweets['Tweet'] = clean_test
test_tweets = test_tweets.drop(columns = ['tweet'])

train_majority = train_tweets[train_tweets['label'] == 0]
train_minority = train_tweets[train_tweets['label'] == 1]
train_minority_upsampled = resample(train_minority,
                                 replace=True,
                                 n_samples=len(train_majority),
                                 random_state=123)
train_upsampled = pd.concat([train_minority_upsampled, train_majority])
train_upsampled['label'].value_counts()
```

```
[nltk_data] Downloading package omw-1.4 to /opt/conda/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

[17]:

```
1    29720
0    29720
Name: label, dtype: int64
```

In the above cells, I clean the data using various NLP libraries like NTLK to remove stop words, non-alphanumeric characters, and break up the tweets by word (among other thing
data is cleaned, the problem of imbalance is dealt with by utilizing the resample method of sklearn to get an even amount of data points between the two classes (labels).

[ ]: