Saurav Thakur **1810110302**

Twitter Sentiment Analysis

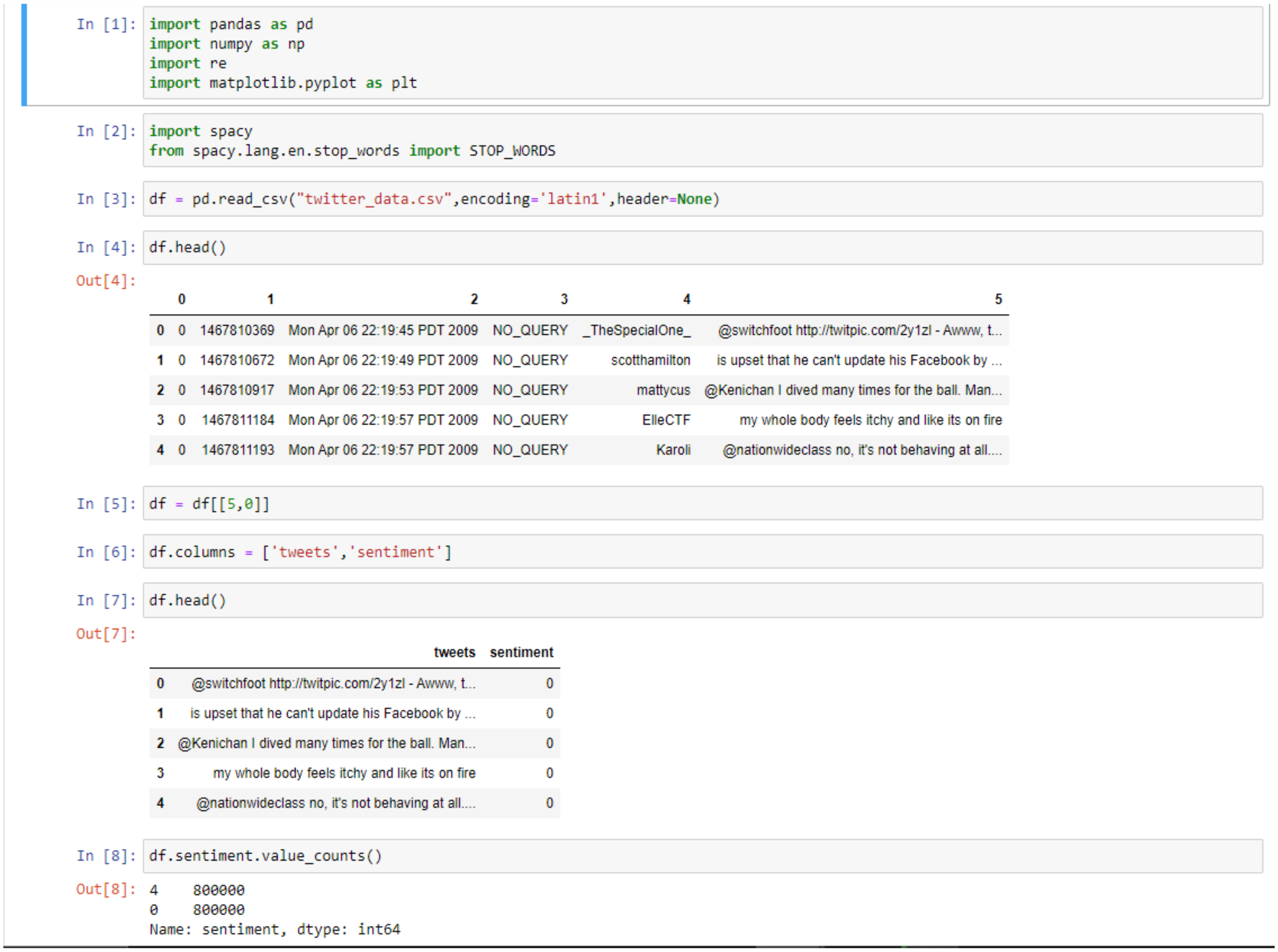
I have made a model based on twitter sentiment analysis which can determine whether the tweets are positive or negative .

Data Preprocessing:

The Data I used here is available on kaggle.

The Link for the dataset is:

https://www.kaggle.com/kazanova/sentiment140



```
In [1]: import pandas as pd
        import numpy as np
        import re
        import matplotlib.pyplot as plt
```

```
In [2]: import spacy
        from spacy.lang.en.stop_words import STOP_WORDS
```

```
In [3]: df = pd.read_csv("twitter_data.csv",encoding='latin1',header=None)
```

```
In [4]: df.head()
```

Out[4]:

	0	1	2	3	4	5
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

```
In [5]: df = df[[5,0]]
```

```
In [6]: df.columns = ['tweets','sentiment']
```

```
In [7]: df.head()
```

Out[7]:

	tweets	sentiment
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0
1	is upset that he can't update his Facebook by ...	0
2	@Kenichan I dived many times for the ball. Man...	0
3	my whole body feels itchy and like its on fire	0
4	@nationwideclass no, it's not behaving at all....	0

```
In [8]: df.sentiment.value_counts()
```

```
Out[8]: 4    800000
        0    800000
        Name: sentiment, dtype: int64
```

Exploratory Data Analysis:

I explored the whole data and did a lot data analysis and I got following results.

Word Count

In [10]:

```
df["word_counts"] = df['tweets'].apply(lambda x: len(str(x).split()))
```

In [11]:

```
df.head()
```

Out[11]:

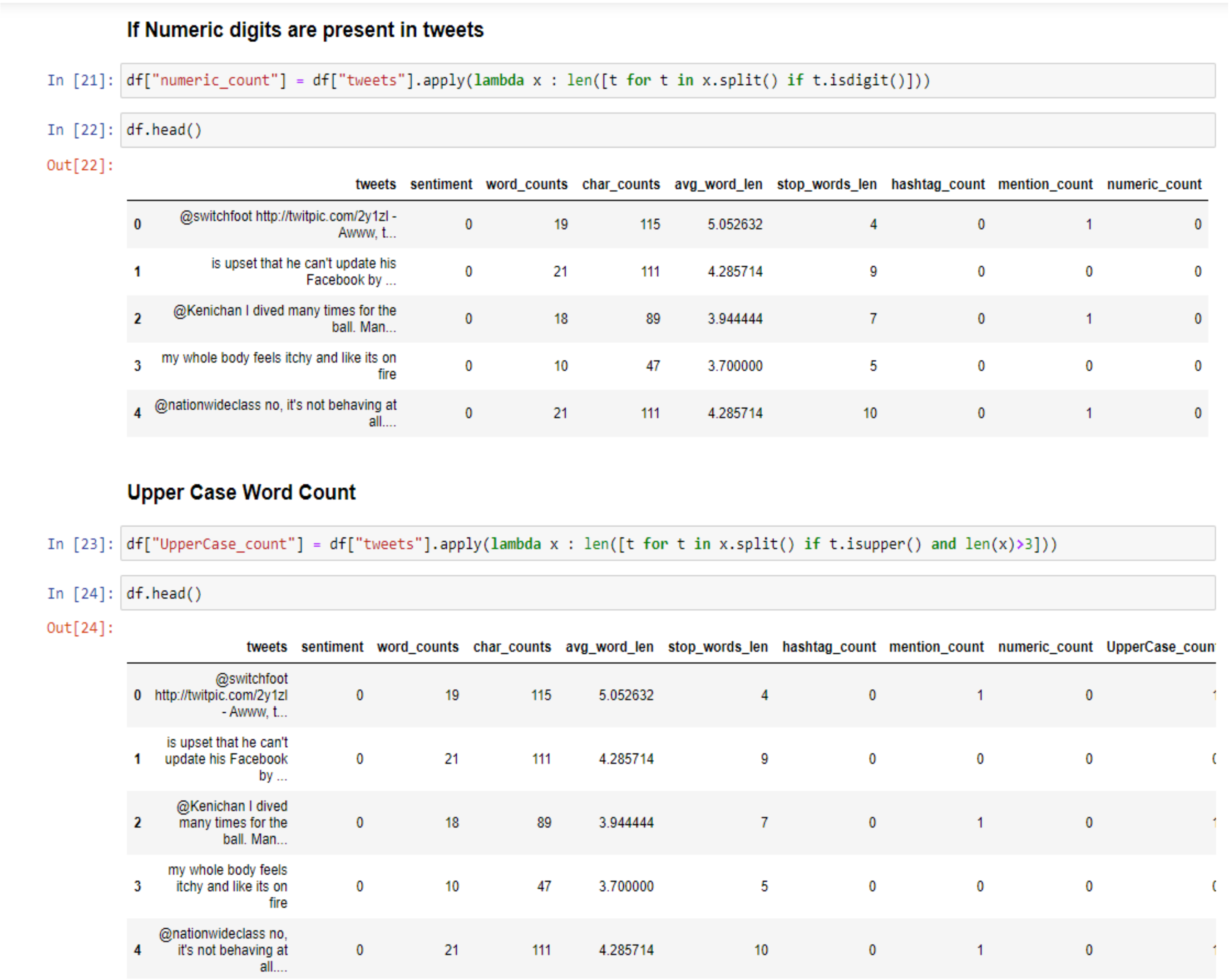
	tweets	sentiment	word_counts
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0	19
1	is upset that he can't update his Facebook by ...	0	21
2	@Kenichan I dived many times for the ball. Man...	0	18
3	my whole body feels itchy and like its on fire	0	10
4	@nationwideclass no, it's not behaving at all....	0	21

In [12]:

```
df["char_counts"] = df['tweets'].apply(lambda x: len(x))
df.head()
```

Out[12]:

	tweets	sentiment	word_counts	char_counts
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0	19	115
1	is upset that he can't update his Facebook by ...	0	21	111
2	@Kenichan I dived many times for the ball. Man...	0	18	89
3	my whole body feels itchy and like its on fire	0	10	47
4	@nationwideclass no, it's not behaving at all....	0	21	111



If Numeric digits are present in tweets

```
In [21]: df["numeric_count"] = df["tweets"].apply(lambda x : len([t for t in x.split() if t.isdigit()]))
```

```
In [22]: df.head()
```

Out[22]:

	tweets	sentiment	word_counts	char_counts	avg_word_len	stop_words_len	hashtag_count	mention_count	numeric_count
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0	19	115	5.052632	4	0	1	0
1	is upset that he can't update his Facebook by ...	0	21	111	4.285714	9	0	0	0
2	@Kenichan I dived many times for the ball. Man...	0	18	89	3.944444	7	0	1	0
3	my whole body feels itchy and like its on fire	0	10	47	3.700000	5	0	0	0
4	@nationwideclass no, it's not behaving at all....	0	21	111	4.285714	10	0	1	0

Upper Case Word Count

```
In [23]: df["UpperCase_count"] = df["tweets"].apply(lambda x : len([t for t in x.split() if t.isupper() and len(x)>3]))
```

```
In [24]: df.head()
```

Out[24]:

	tweets	sentiment	word_counts	char_counts	avg_word_len	stop_words_len	hashtag_count	mention_count	numeric_count	UpperCase_coun
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0	19	115	5.052632	4	0	1	0	1
1	is upset that he can't update his Facebook by ...	0	21	111	4.285714	9	0	0	0	0
2	@Kenichan I dived many times for the ball. Man...	0	18	89	3.944444	7	0	1	0	1
3	my whole body feels itchy and like its on fire	0	10	47	3.700000	5	0	0	0	0
4	@nationwideclass no, it's not behaving at all....	0	21	111	4.285714	10	0	1	0	1

Data Cleaning:

For data cleaning I removed Urls , removed accented characters, punctuation and special characters and more.

Remove URLs

```
In [32]: import re
```

```
In [33]: df['urls_flag'] = df['tweets'].apply(lambda x: len(re.findall(r'(http|ftp|https)://([\w_-]+(?:(?:\.[\w_-]+)+))([\w.,@?^=%&:/~+#-]
```

```
In [34]: df['tweets'] = df['tweets'].apply(lambda x: re.sub(r'(http|ftp|https)://([\w_-]+(?:(?:\.[\w_-]+)+))([\w.,@?^=%&:/~+#-]*[\w@?^=%&/
```

```
In [35]: df.head()
```

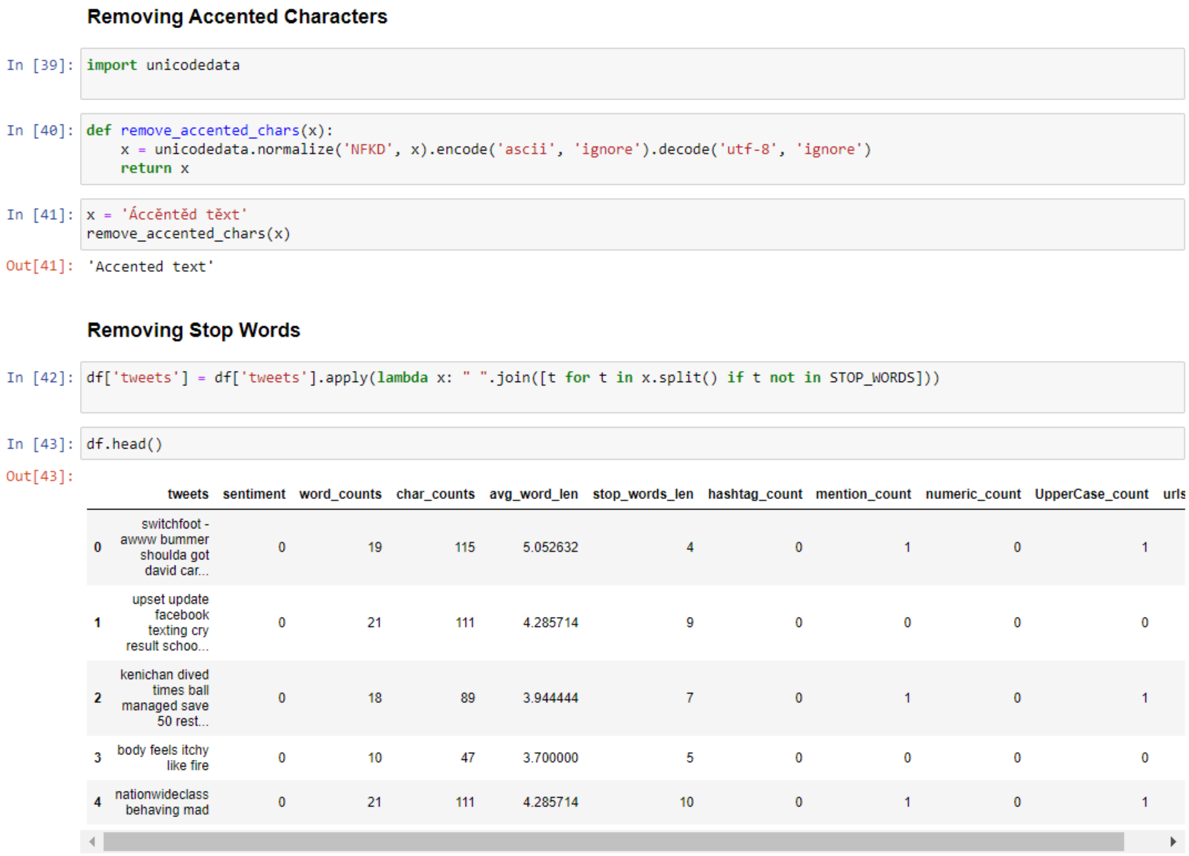
Out[35]:

	tweets	sentiment	word_counts	char_counts	avg_word_len	stop_words_len	hashtag_count	mention_count	numeric_count	UpperCase_count	u
0	@switchfoot - awww, that is a bummer. you sh...	0	19	115	5.052632	4	0	1	0	1	
1	is upset that he cannot update his facebook by...	0	21	111	4.285714	9	0	0	0	0	
2	@kenichan i dived many times for the ball. man...	0	18	89	3.944444	7	0	1	0	1	
3	my whole body feels itchy and like its on fire	0	10	47	3.700000	5	0	0	0	0	
4	@nationwideclass no, it is not behaving at all...	0	21	111	4.285714	10	0	1	0	1	

Removing Retweets

```
In [36]: df['tweets'] = df['tweets'].apply(lambda x: re.sub('RT', "", x))
```

Removing Special Characters and Punctuations



Removing Accented Characters

```
In [39]: import unicodedata
```

```
In [40]: def remove_accented_chars(x):
             x = unicodedata.normalize('NFKD', x).encode('ascii', 'ignore').decode('utf-8', 'ignore')
             return x
```

```
In [41]: x = 'Áccĕntĕd tĕxt'
         remove_accented_chars(x)

Out[41]: 'Accented text'
```

Removing Stop Words

```
In [42]: df['tweets'] = df['tweets'].apply(lambda x: " ".join([t for t in x.split() if t not in STOP_WORDS]))
```

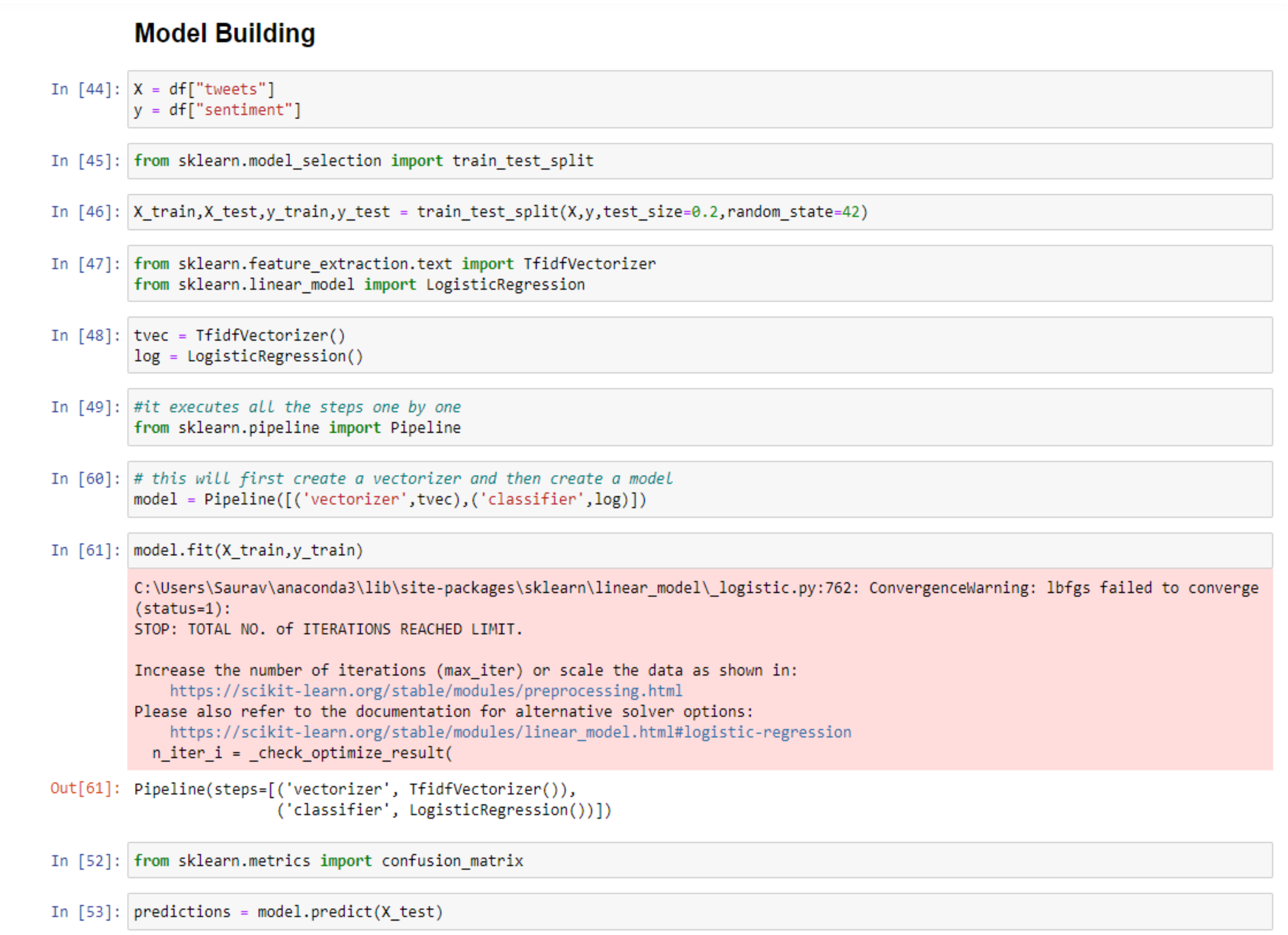
```
In [43]: df.head()

Out[43]:
```

	tweets	sentiment	word_counts	char_counts	avg_word_len	stop_words_len	hashtag_count	mention_count	numeric_count	UpperCase_count	urls
0	switchfoot - awww bummer shoulda got david car...	0	19	115	5.052632	4	0	1	0	1	
1	upset update facebook texting cry result schoo...	0	21	111	4.285714	9	0	0	0	0	
2	kenichan dived times ball managed save 50 rest...	0	18	89	3.944444	7	0	1	0	1	
3	body feels itchy like fire	0	10	47	3.700000	5	0	0	0	0	
4	nationwideclass behaving mad	0	21	111	4.285714	10	0	1	0	1	

Model Building:

For Model building I used Tfidf, Logistic regression and I created pipeline which will execute tfidf and logistic regression sequentially.



Model Building

```
In [44]: X = df["tweets"]
         y = df["sentiment"]

In [45]: from sklearn.model_selection import train_test_split

In [46]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)

In [47]: from sklearn.feature_extraction.text import TfidfVectorizer
         from sklearn.linear_model import LogisticRegression

In [48]: tvec = TfidfVectorizer()
         log = LogisticRegression()

In [49]: #it executes all the steps one by one
         from sklearn.pipeline import Pipeline

In [60]: # this will first create a vectorizer and then create a model
         model = Pipeline([('vectorizer',tvec),('classifier',log)])

In [61]: model.fit(X_train,y_train)

         C:\Users\Saurav\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge
         (status=1):
         STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

         Increase the number of iterations (max_iter) or scale the data as shown in:
             https://scikit-learn.org/stable/modules/preprocessing.html
         Please also refer to the documentation for alternative solver options:
             https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
           n_iter_i = _check_optimize_result(

Out[61]: Pipeline(steps=[('vectorizer', TfidfVectorizer()),
                         ('classifier', LogisticRegression())])

In [52]: from sklearn.metrics import confusion_matrix

In [53]: predictions = model.predict(X_test)
```

Predictions:

After Building the model I got the predictions 0 and 4 where 0 means negative and 4 means positive. I used data from my twitter account tweets for predicting the model. The model has an accuracy of 77.84%.

Model Predictions

```
In [55]: from sklearn.metrics import accuracy_score,precision_score,recall_score
```

```
In [56]: print("Accuracy : ",accuracy_score(predictions,y_test))
         print("Precision : ",precision_score(predictions,y_test,average='weighted'))
         print("Recall : ",recall_score(predictions,y_test,average='weighted'))
```

```
Accuracy :  0.77846875
Precision :  0.7793796503192831
Recall :  0.77846875
```

Predicting

0 - Negative

4 - Positive

```
In [57]: example = ["I hate you"]
         model.predict(example)
```

```
Out[57]: array([0], dtype=int64)
```

```
In [66]: model.predict(["So happy the Greatest Of All Time will meet again tonight It's gonna be a showdown Watch out  Ronaldo"])
```

```
Out[66]: array([4], dtype=int64)
```

```
In [62]: model.predict(["I need to say this so people know how big of a mistake this was, I was traumatized by Human Centipede back in 200
```

```
Out[62]: array([0], dtype=int64)
```

```
In [64]: model.predict(["As cases of Covid-19 continue to rise across the country, a poll of firefighters in the Fire Department of New Yo
```

```
Out[64]: array([4], dtype=int64)
```

```
In [65]: model.predict(["way too much money invested by these pharmaceuticals than to create a faulty fatal vaccine that would be financia
```

```
Out[65]: array([4], dtype=int64)
```