Twitter Airline sentiment Analysis

[1]Anurag Lahon
[1]Student
[1]KIIT University

***Abstract* - Social Media is used by various companies to gather information from their customers. Customers share their opinion and feedback about the products and services. Social Media has become beneficial for companies where they analyse the feedbacks to improve their services and products. In this paper we will classify twitter data into feedbacks using machine learning and natural language processing (NLP) with the help of programming language Python. The tweets are categorized into neutral, positive, negative sentiments. We will use different machine learning algorithms.**

***keywords* - analytics,nlp**

I. INTRODUCTION

Twitter has become popular over the years and it has various types of information. Huge number of daily post on twitter where public expressed their views. In airline industry, large amount of customers post their views regarding services of the airlines like bag lost, good food, flight delay and many others .This helps the airline companies to improve their services and attract more customers.
In this paper, we are using a dataset where there are tweets from various airlines and classified them into three categories: - positive, negative and neutral. There are around 11,000 tweets which are trained with the help of machine learning techniques and natural language processing.
Each tweet is only 280 characters which allow people to use many shortcuts and are not in proper languge processing.

II. DESIGN AND IMPLEMENTATION

After obtaining the twitter dataset, we divide the dataset into train and test dataset. We download the libraries which is essential for natural language processing.

```
from nltk.tokenize import sent_tokenize,word_tokenize
```

```
from nltk.corpus import stopwords
```

```
from nltk.stem import PorterStemmer
```

```
from nltk import pos_tag
from nltk.corpus import state_union
```

```
import nltk
```

```
from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\anura_000\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

(i)Lemmatization: Lemmatization is the process of grouping together the inflected forms of the word so they can be analysed.
(ii) Part of Speech: Part of Speech is the category of words which have similar grammatical properties.
(iii) Stemming: Processing of reducing inflected words to their word stem ,base or root form.
(iv)Tokenisation: Tokenisation is the process by which big quantity of text is divided into smaller parts called tokens. These tokens are useful and are considered as a base step for stemming and lemmatization.
We imported all the basic libraries and load the train and test data using pandas.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
tweets=pd.read_csv("C:/Users/anura_000/Downloads/twitter_airline.csv")
tweets_test=pd.read_csv("C:/Users/anura_000/Downloads/twitter_test.csv")
```

There are three categories in the train data set and we displayed their count and visualize it.

```
mood_count = tweets['airline_sentiment'].value_counts()
mood_count
```

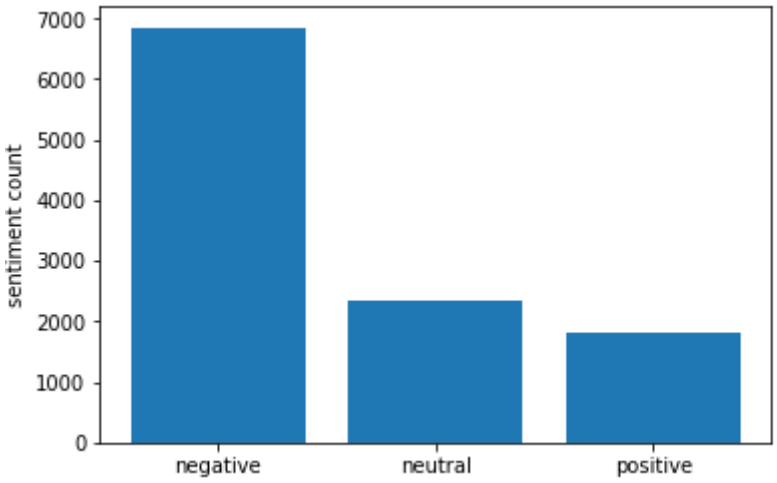
```
negative    6851
neutral     2327
positive    1802
Name: airline_sentiment, dtype: int64
```

```
tweets['airline'].value_counts()
```

```
United            2928
US Airways        2152
American          2078
Southwest         1817
Delta             1639
Virgin America     366
Name: airline, dtype: int64
```

```
Index = [1, 2, 3]
plt.bar(Index, mood_count)
plt.xticks(Index, ['negative', 'neutral', 'positive'])
plt.ylabel('sentiment count')
```

```
Text(0,0.5,'sentiment count')
```



Stop words: Stop words are generally the most common words used by all natural language processing tools.

```
import re
import nltk
from nltk.corpus import stopwords
```

```
tweets['tokenized_sents'] = tweets.apply(lambda row: nltk.word_tokenize(row['text']), axis=1)
tweets_test['tokenized_sents'] = tweets_test.apply(lambda row: nltk.word_tokenize(row['text']), axis=1)
```

We use stop words to remove the redundant words on both train and test set.

```
documents=list(tweets[['tokenized_sents','airline_sentiment']].itertuples(index=False, name=None))
```

```
document_test=list(tweets_test.tokenized_sents)
```

Make the tweets converted into lists.

WorldNetLemmatizer is used to Lemmatized the tweets and we will do a reshuffle of the tweets.

```
from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()
```

```
from nltk.corpus import wordnet
```

```
from nltk import pos_tag
import random
random.shuffle(documents)
documents[0:5]
```

```
def get_simple_pos(tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN
```

Stop words are used in the English version.We also add punctuation to our stopword list.

```
stops=set(stopwords.words('english'))
```

```
punctuations=list(string.punctuation)
```

```
stops.update(punctuations)
```

```
stops,string.punctuation
```

After getting the stop words we will create a function to remove the stop words from our tweets for both of our training and test data set. We will apply our function to remove the stop words from our tweets. Then we create a dictionary of the most common words.

```
def clean_reviews(words):
    output_words=[]
    for w in words:
        if not w.lower() in stops:
            pos=pos_tag([w])
            clean_word=lemmatizer.lemmatize(w,get_simple_pos(pos[0][1]))
            output_words.append(clean_word.lower())
    return output_words
```

```
documents=[(clean_reviews(document),airline_sentiment) for document,airline_sentiment in documents]
documents
```

```
document_test=[(clean_reviews(document)) for document in document_test]
document_test
```

```
all_words=[]
```

```
for doc in training_documents:
    all_words+=doc[0]
```

```
import nltk
freq=nltk.FreqDist(all_words)
```

```
common=freq.most_common(3000)
```

```
features=[i[0] for i in common]
```

```
def get_feature_dict(words):
    current_features={}
    words_set=set(words)
    for w in features:
        current_features[w]=w in words_set
    return current_features
```

```
training_data=[(get_feature_dict(doc),category) for doc,category in training_documents]
training_data[0]
```

```
({'flight': False,
  'united': False,
  'usairways': False,
  'americanair': False,
  'southwestair': True,
  'jetblue': False,
  'get': False,
  "n't": False,
  "'s": False,
  'http': False,
  'hour': False,
  'thanks': False,
  'cancelled': False,
  '...': False,
  'service': False,
  'help': False,
  'customer': False,
```

Then we apply Naive Bayes Classifier which gives the best result on our text dataset as we have tested. We have also tested on other classifier like SVC which gives less accuracy than Naive Bayes Classifier.

```
from nltk import NaiveBayesClassifier
Classifier=NaiveBayesClassifier.train(training_data)
length=len(testing_data)
for i in range(0,length):
    print(Classifier.classify(testing_data[i]))
```

III. CONCLUSION

In this paper we displayed how we use machine learning methods to obtain feedback from customers .We found that Naive Bayes Classifier works well in our dataset.

IV. REFERENCE

[1] https://www.wikipedia.org/
[2] Sentiment Analysis of twitter data- Hamid Bagheri, Md Johirne Islam
[3] https://www.nltk.org
[4] My github: anuraglahon16