

Project Report

Title of Project = *Sentiment Analysis*

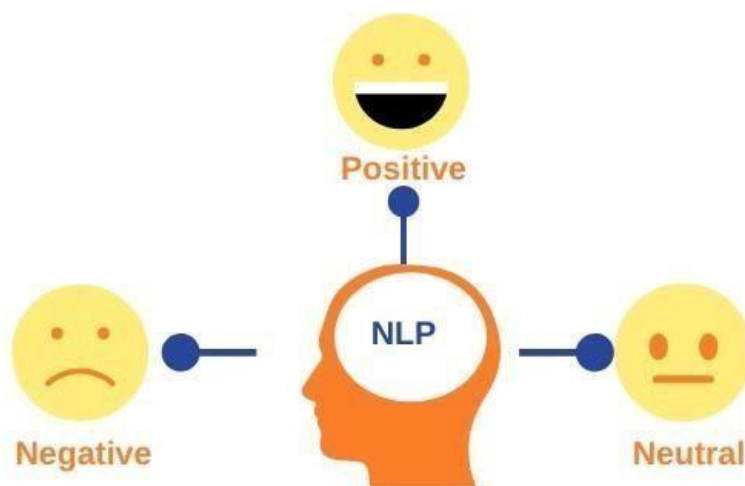
Submitted By:

Gaurav Deore

Introduction:

NLP Natural Language Processing is a branch of artificial intelligence (AI) that helps the computers to understand the way that human speak and write. NLP combines the studies of data science, computer science and linguistics. It gives a accurate analysis, improve customer satisfaction, etc.,

Sentiment Analysis is a technique of NLP to extract data from text. It is the dissection of data (text, voice, etc.,) in order to determine whether it's positive, neutral or negative.



Sentiment Analysis

Sentiment analysis can transform large archives of customer reviews into actionable, quantified results.

Dataset Used:

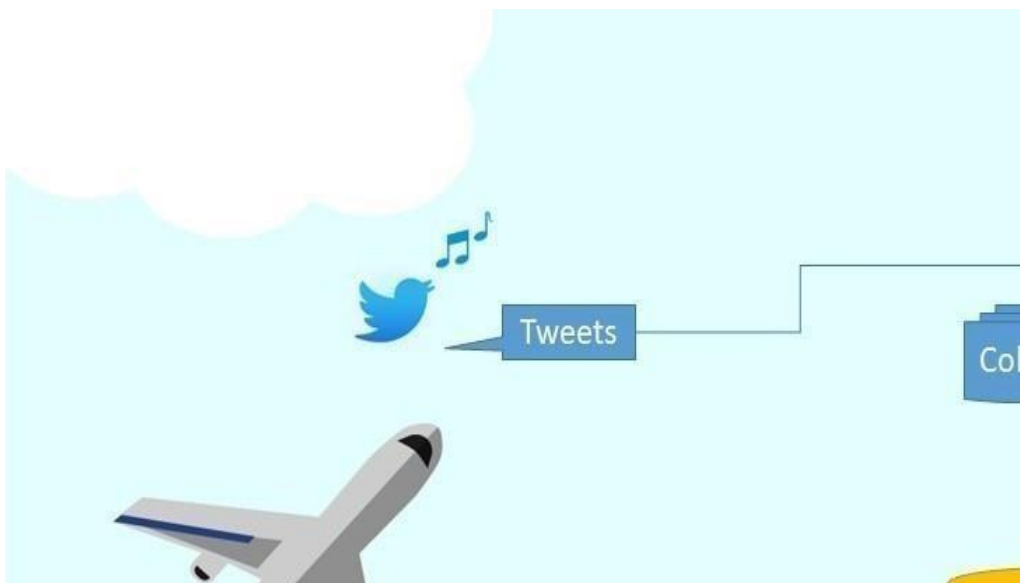
Used 2 sets of data. One is about airline reviews and another one is Disneyland amusement park reviews.

The dataset required for the project can be downloaded from the link as mentioned below:

1. For Airline review : [Dataset Link](#)
2. For Disneyland Amusement Park review : [Dataset](#)

Steps Followed:

1. For Airline review dataset



Loaded and explored the data by mounting the drive, read the csv file. Did the text cleaning by removing hashtags, links, extra spaces, stopwords and lemmatize the text. The data preparation is done by label encoding using LabelEncoder then splitted the data into train and test and used TF-IDF

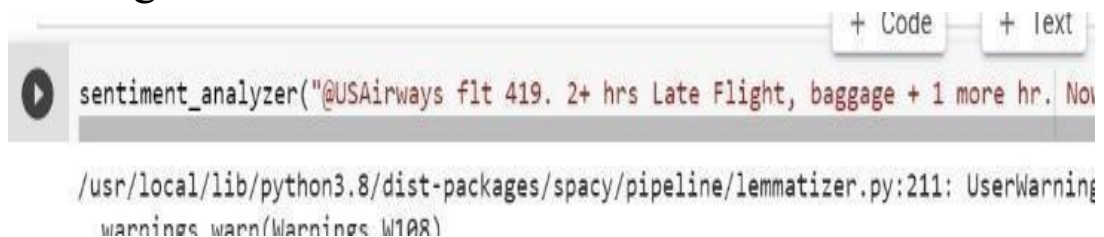
feature by importing Tf-idfVectorizer. Build a model using Naïve Bayes, Logistic Regression and LSTM.

Used libraries:

numpy, pandas, sklearn (MultinomialNB, f1_score, LogisticRegression), tensorflow.keras (Sequential, LSTM, Dropout)

Outputs

1. LogisticRegression



```
sentiment_analyzer("@US Airways flt 419. 2+ hrs Late Flight, baggage + 1 more hr. No")
```

/usr/local/lib/python3.8/dist-packages/spacy/pipeline/lemmatizer.py:211: UserWarning: warnings.warn(Warnings.W108)

2. Comparison of Naive Bayes and LogisticRegression

Model Building Summary

Model	Train Set	Validation Set
Naive Bayes	0.7303	0.6884
Logistic Regression	0.8074	0.7530

3. LSTM

```

test_sentence1 = "I loved my journey on this flight."
predict_sentiment(test_sentence1)

test_sentence2 = "This is the worst flight experience of my life!"
predict_sentiment(test_sentence2)

```

```

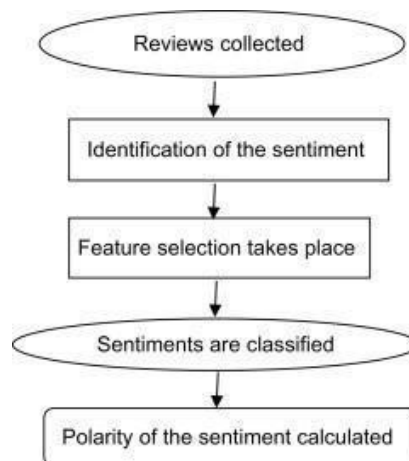
➔ 1/1 [=====] - 0s 297ms/step
Predicted label: positive
1/1 [=====] - 0s 39ms/step
Predicted label: negative

```

Conclusion:

Based on the comparison of Naive Bayes and Logistic Regression, the Logistic Regression performs better than Naive Bayes on this dataset.

2. For Disneyland Amusement Park review



Used VADER and TextBlob separately to make the code work more efficiently. VADER and TextBlob use a lexiconbased method instead of machine learning approaches. The lexicon sentiment analysis outputs a polarity score of -1 to 1, where

-1 represents highly negative sentiment

1 represents highly positive sentiment

0 represents neutral sentiment

Used wordcloud too, to show the differences of the words according to the ratings

Used libraries :

numpy, pandas, nltk, vader_lexicon, textblob, seaborn, matplotlib.pyplot, wordcloud.

Output1:

```
Sentence: disneyland is a fantastic place for to go with f
VADER sentiment score: 0.7579
```

Output2:

```
Sentence: disneyland is a fantastic place for to go with family and kids, it is a must to good place...!
VADER sentiment score: 0.7955
TextBlob sentiment score: 0.7
```

Conclusion:

Based on the comparison above, we can see that VADER provides more accurate sentiment score than TextBlob.

VADER takes capitalization, repeated words and emoji into consideration while evaluating the sentiment of the text.

Note : The source code is on the [github](#) repository .