Sentiment Analyzer

Bhavuk Sagar

Introduction

Sentiment mainly refers to feelings, emotions, opinion or attitude1 . With the rapid increase of World Wide Web, people often express their sentiments over internet through social media, blogs, rating and reviews. Due to this increase in the textual data, there is a need to analyze the concept of expressing sentiments and calculate the insights for exploring business. Business owners and advertising companies often employ sentiment analysis to discover new business strategies and advertising campaign.

Problem

On a daily basis many posts, reviews on products and comments are posted on different platforms. Reading those reviews or comments are not possible by human.

Solution

I'm creating a machine learning model which is able to predict sentiment of any written text. After predicting the sentiment its become easy to visualize numbers of positive reviews and negative reviews.

Data acquisition and cleaning

Data Source

I'm taking the users review dataset from the kaggel which contains 11,000 reviews.

Data Cleaning

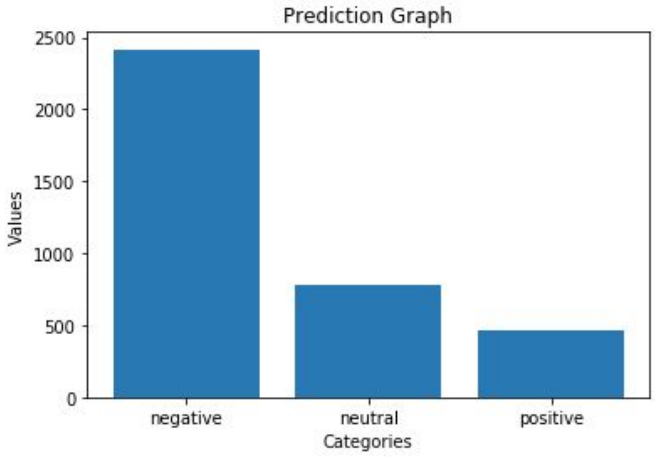
- Data cleaning is the most important task of data science. First of all i'm checking for NaN values in my dataset but there's no NaN value present in the data.
- After that I'm converting my data into a dataframe with the help of pandas library.
- In text data the main task is to take out the most clean words which means removing all the stop words(is,a,an,the,that,when am, he ,she, where etc) and punctuations. And for the cleaning process I'm using sklearn library for removing all the stop words and punctuations.
- Remaining words is used for train the model.

Data Conversion

We have to convert the english words into the numerical form because machine learning algorithm only understand and take input in numerical form. In conversion process words are replaced with their frequency of appearing in the dataset.

After that high frequency words are used for training and I'm using 5000 words.

Data Visualization



Predictive Modeling

Our problem is classification problem. I'm using all classification problem which their best parameters for selecting the best algorithm for prediction. Different algorithm give different result like:-

SVM- 82.2% accuracy.

Logistic Regression- 85.22%

KNN- 70%

Decision Tree- 90.1%

Decision tree gives me the best result. That's why I trained my model on decision tree.

Model Testing Outputs

Testing model

```
In [70]: s=[input("Enter your Review:---")]
         xx=cv.transform(s)
         print(clg.predict(xx))

         Enter your Review:---i love it
         ['positive']

In [ ]:  s=[input("Enter your Review:---")]
         xx=cv.transform(s)
         print(clg.predict(xx))

In [72]: s=[input("Enter your Review:---")]
         xx=cv.transform(s)
         print(clg.predict(xx))

         Enter your Review:---it was a good flight and i love to use there service again
         ['positive']

In [ ]:
```