Sentiment Analysis Using Machine Learning

P. Naresh(Assistant Professor), A.Jahnavi Reddy(UG Scholar), S.Prem Kumar(UG Scholar), CH. Nikhil(UG Scholar), T. Chandu(UG Scholar)

Department of Information Technology, Vignan Institute of Technologyand Science, Hyderabad, Telangana.

Contributing authors: nareshintell4@gmail.com; janvireddy1601@gmail.com; siliverupremkumar03@gmail.com; nikhil.chidhirala@gmail.com; chanduthanneru8@gmail.com;

Abstract

Sentiment analysis is one of the fastest spreading research areas in computer sci- ence, making it challenging to keep track of all the activities in the area. We present a customer feedback reviews on product, where we utilize opinion mining, text mining and sentiments, which has affected the surrounded world by changing their opinion on a specific product. Data used in this study are online prod-uct reviews collected from Amazon.com. We performed a comparative sentiment analysis of retrieved reviews. This project provides you with sentimental analy- sis of various opinions on dividing them positive, negative and neutral behavior. The rating description and the rating number won't match in many cases. The ratings are done based on various components of product for example in case ofLaptop some user rate only based on speed of performance but other users ratebased on battery, display and sound. So, parsing the text description and analyzing the accurate ratings is so important than simply plotting the rating numbers. So, sentiment analysis techniques are required to find the polarity of the review. That is to find if the review is positive or negative or neutral.

1 Introduction

Analysis of sentiments is to analyze the natural language and to find the emotions, express by the human beings. The idea behind sentiment analysis is to determine polarity of textual opinion given by person. Sentiment Analysis is useful in product recommendations. Based on the reviews given by the user, the products can be recommended to another user. Major product websites are using sentiment analysis to understand the popularity and problems with the product. Sentiment analysis mainly formulated as two class classification problem, positive and negative. Sentiment analysis using ordinal classification gives more clear idea about sentiments. The proposed system determines polarity of reviews given by users, using ordinal classification. The system will give polarity using machine learning algorithms. The achieved polarity will be used to provide recommendation to users. Analysis of sentiments is to analyze the natural language and to find the emotions, express by the human beings. The idea behind sentiment analysis is to determine polarity of textual opinion given by per- son. Sentiment Analysis is useful in product recommendations. Based on the reviews given by the user, the products can be recommended to another user. Major product websites are using sentiment analysis to understand the popularity and problems with the product. Sentiment analysis mainly formulated as two class classification problem, positive and negative. Sentiment analysis using ordinal classification gives more clear idea about sentiments. The proposed system determines polarity of reviews given by users, using ordinal classification. The system will give polarity using machine learning algorithms. The achieved polarity will be used to provide recommendation to users. The primary goal of this study is to provide recommendation list based on sentiment analysis. Classification of sentiments in scale of -5 to+5 using machine learning algo- rithms and ordinal classification approach. Providing recommendation system which will offer personalized recommendation experience to users based on sentiment scale and user information.

2 Literature Survey

Conventionally, sentiment analysis has been about opinion contradiction, i.e., whethersomeone has positive, neutral, or negative opinion towards something. Data used in this paper is a combination of product reviews collected from Amazon.com, between July and September, 2018. There have been somewhat overcome in the followingtwo manners: Firstly, each product review holds inspections before it can be posted. Secondly, respective review must have a rating on it that can be used as the ground truth. The rating is based on a star scaled system, where the highest rating has 5 stars and the lowest rating has only 1 star. Here, the report tackles fundamental clauses of sentiment analysis, namely sentiment polarity distribution. This paper prospective a schema for online opinion representation. The inputs to the schema are product name, date and time and review of that product, where output contains the summary ofthe review in compact manner. The whole process includes summarization in three steps: (1) Product feature based, which is given by customer (2) In each review, Identify expected features in each opinion sentence and (3) Finding out whether the feature/opinion is positive, negative or neutral.

3 System Analysis

3.1 Existing System

The traditional techniques of machine learning which are used for sentiment analysisis 'Vader Algorithm'. But if the dataset is analyzed with Vader algorithm it is nogiving the proper result still many of the positive sentences are detected as negative and many of the negative sentences are detected as positive.

3.2 Proposed System

We have proposed an automatic system to perform aspect level sentiment analysis of customer feedbacks in ecommerce application. We have analyzed the feasibility of thesystem by working manually on third set of data and implementing some modules of the system. We are in the process of building a prototype system where system will automatically identify important features/ qualities/facts of users with their sentimentpolarity.

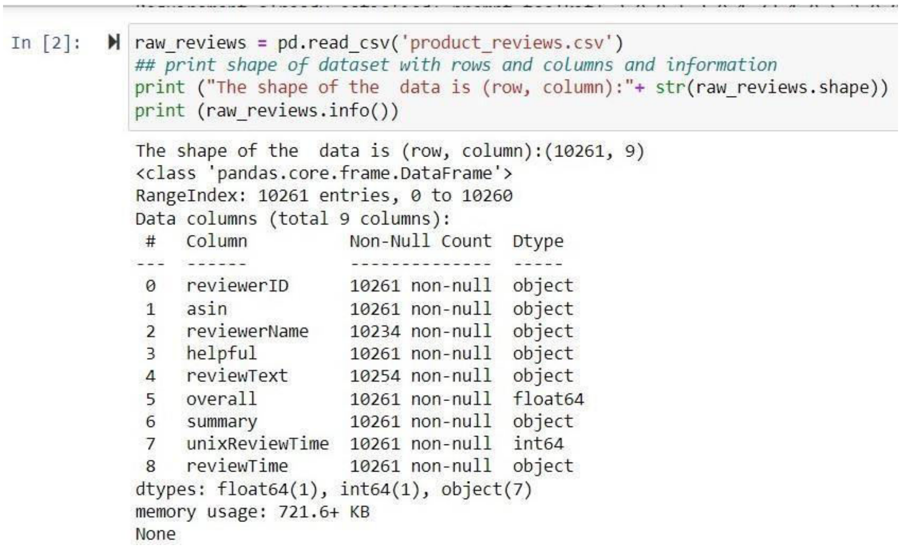
4 Methodology

Collection of Data: The appropriate amount of Data used in this paper is a arranged set of product reviews collected from amazon.com. From August to December 2018,in total, we collected over 500 sentiments of product reviews in which the products belong to 4 major categories: Mobiles, Computers, Flash drives and Electronics 3(a)). These online reviews were posted by over 3.2 million of customers (reviewers) towards 10,001 products. Each review includes the following information: 1) reviewerID; 2) product model; 3) date and time of the review; 4) review text Sentiments Sentences: Judgment and POS Tagging This process is proposed by Pang and Lee in which all objective content should be removed for analysis of sentiment. Instead of removing objective content, in our study, all subjective content was extracted for future analysis which consists of all sentiment sentences. A sentiment sentence is that which contains,at least, one positive or negative word. All of the sentences were first of all arranged into categorized English words. Every word of a sentence has its semantic role that defines how the word is used. The semantic roles are also called as the parts of speech.There are generally 8 parts of speech in English: the verb, the pronoun, the noun, someadverbs and prepositions, the interjection, and the conjunction. In natural language preference and suitability, part-of-speech (POS) taggers have been generated to classifywords based on their parts of speech. In sentiment classification, a POS tagger is most important because of the following two reasons: 1) Words like nouns and pronouns mostly do not contain any sentiment. So, it is able to filter out such words with the help of a POS tagger; 2) A POS tagger can also be useful in distinguish words that can be used in different parts of speech. For instance, as a verb, "improved" may conduct different amount of sentiment as being of an adjective. The POS tagger used for this survey is a max-entropy POS tagger developed for the Penn Treebank Project. This tagger is able to provide 46 different tags which indicate that it can identify more detailed semantic roles than only 8.

5 Implementation

The Python programming language is an Open Source, cross-platform, high level, dynamic, interpreted language. The Python ’philosophy’ emphasizes readability, clarity and simplicity, whilst maximizing the power and expressiveness available to the programmer. The ultimate compliment to a Python programmer is not that his code is clever, but that it is elegant. For these reasons Python is an excellent ’first language’, while still being a powerful tool in the hands of the seasoned and cynical programmer. NumPy is a Python package which stands for ‘Numerical Python’. Itis the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc. NumPy array can also be used as an efficient multi-dimensional container for generic data. Pandas are an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data • Pre-processing refers to the transforma- tions applied to our data before feeding it to the algorithm. • Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words,

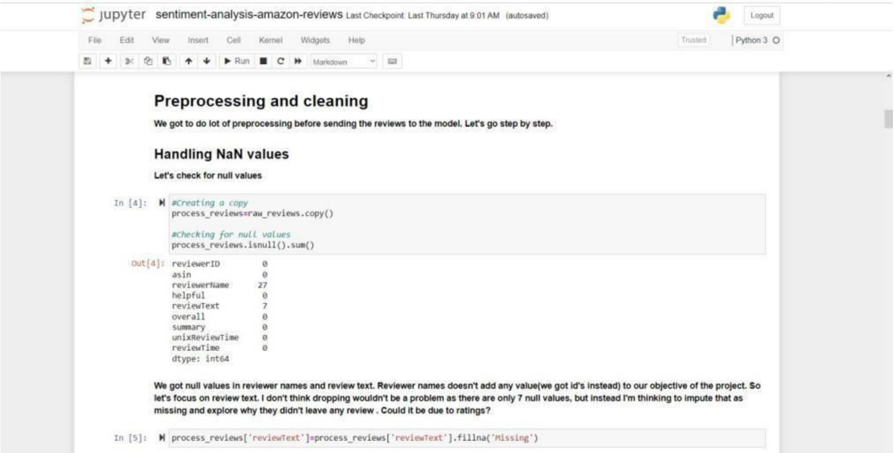
whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. Data Cleaning is oneof those things that everyone does but no one really talks about. It surely isn’t the fanciest part of machine learning and at the same time, there aren’t any hidden tricks or secrets to uncover. However, proper data cleaning can make or break your project.



```
In [2]:  raw_reviews = pd.read_csv('product_reviews.csv')
         ## print shape of dataset with rows and columns and information
         print ("The shape of the  data is (row, column):"+ str(raw_reviews.shape))
         print (raw_reviews.info())
```

```
The shape of the  data is (row, column):(10261, 9)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10261 entries, 0 to 10260
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   reviewerID      10261 non-null  object
 1   asin            10261 non-null  object
 2   reviewerName    10234 non-null  object
 3   helpful         10261 non-null  object
 4   reviewText      10254 non-null  object
 5   overall         10261 non-null  float64
 6   summary         10261 non-null  object
 7   unixReviewTime  10261 non-null  int64
 8   reviewTime      10261 non-null  object
dtypes: float64(1), int64(1), object(7)
memory usage: 721.6+ KB
None
```

Fig. 1 Importing the Dataset and finding the shape of the data


jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown

Preprocessing and cleaning

We got to do lot of preprocessing before sending the reviews to the model. Let's go step by step.

Handling NaN values

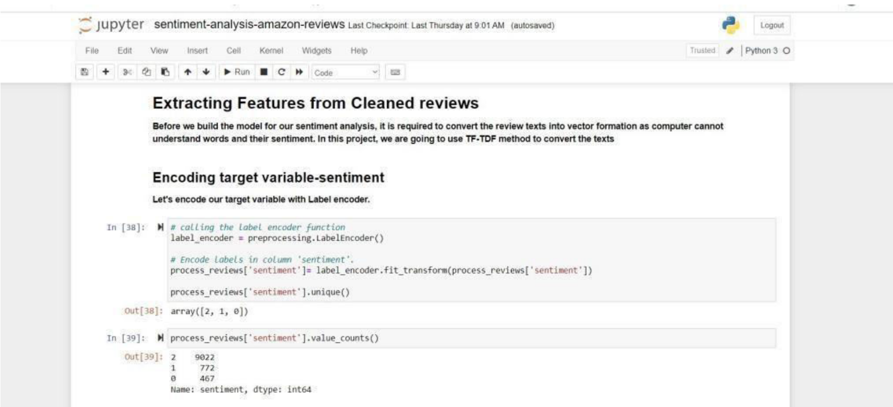
Let's check for null values

```
In [4]:  #Creating a copy
         process_reviews=raw_reviews.copy()

         #Checking for null values
         process_reviews.isnull().sum()

Out[4]:  reviewerID         0
         asin               0
         reviewerName      27
         helpful            0
         reviewText         7
         overall            0
         summary            0
         unixReviewTime     0
         reviewTime         0
         dtype: int64
```

We got null values in reviewer names and review text. Reviewer names doesn't add any value(we got id's instead) to our objective of the project. So let's focus on review text. I don't think dropping wouldn't be a problem as there are only 7 null values, but instead I'm thinking to impute that as missing and explore why they didn't leave any review . Could it be due to ratings?

```
In [5]:  process_reviews['reviewText']=process_reviews['reviewText'].fillna('Missing')
```

Fig. 2 Handling the null values before sending to the model


jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

Extracting Features from Cleaned reviews

Before we build the model for our sentiment analysis, it is required to convert the review texts into vector formation as computer cannot understand words and their sentiment. In this project, we are going to use TF-TDF method to convert the texts

Encoding target variable-sentiment

Let's encode our target variable with Label encoder.

```
In [38]:  # calling the label encoder function
          label_encoder = preprocessing.LabelEncoder()

          # Encode labels in column 'sentiment'.
          process_reviews['sentiment']= label_encoder.fit_transform(process_reviews['sentiment'])

          process_reviews['sentiment'].unique()

Out[38]:  array([2, 1, 0])

In [39]:  process_reviews['sentiment'].value_counts()

Out[39]:  2    9022
          1     772
          0     467
          Name: sentiment, dtype: int64
```

Fig. 3 Extracting features from cleaned reviews

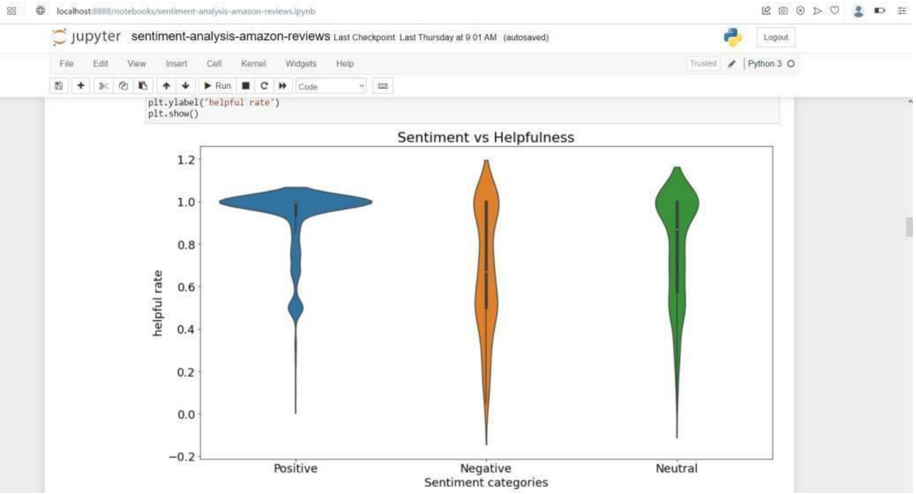
6 Results and Output

Outcomes Possible: There are three types of possible test outcomes:

•OK – This meansthat all the tests are passed.

•FAIL – This means that the test did not pass and an AssertionError exception is raised.

•ERROR – This means that the test raises an exception other than AssertionError.

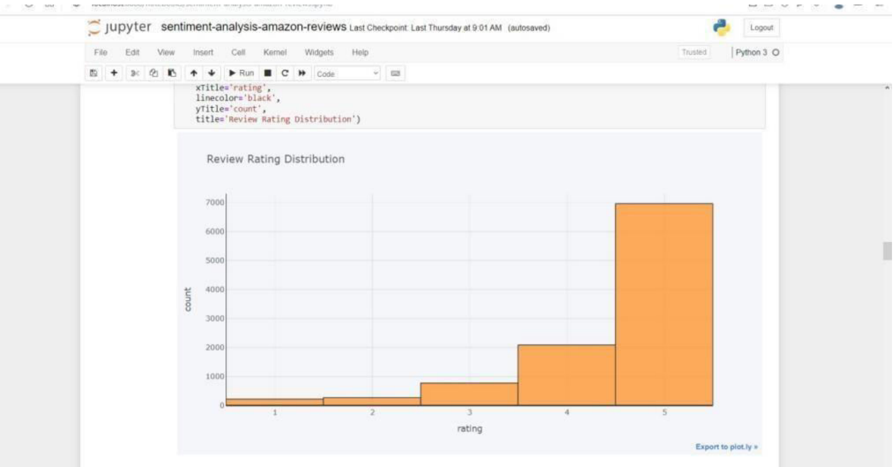


localhost:8888/notebooks/sentiment-analysis-amazon-reviews.ipynb
jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
plt.ylabel('helpful rate')
plt.show()
```

Sentiment vs Helpfulness
helpful rate
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Positive
Negative
Neutral
Sentiment categories

Fig. 4 Output screens (sentiment vs helpfulness graph)



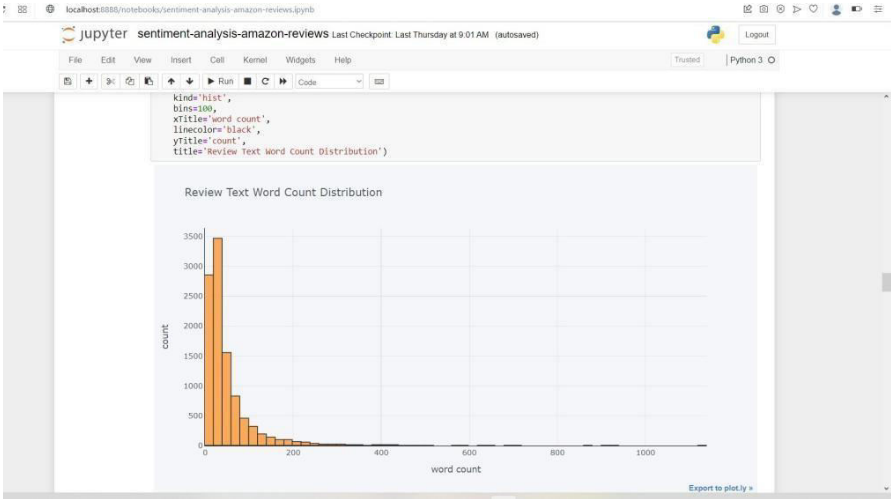
jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
xTitle='rating',
linecolor='black',
yTitle='count',
title='Review Rating Distribution')
```

Review Rating Distribution
count
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
rating
Export to plot.ly »

Fig. 5 Review rating distribution

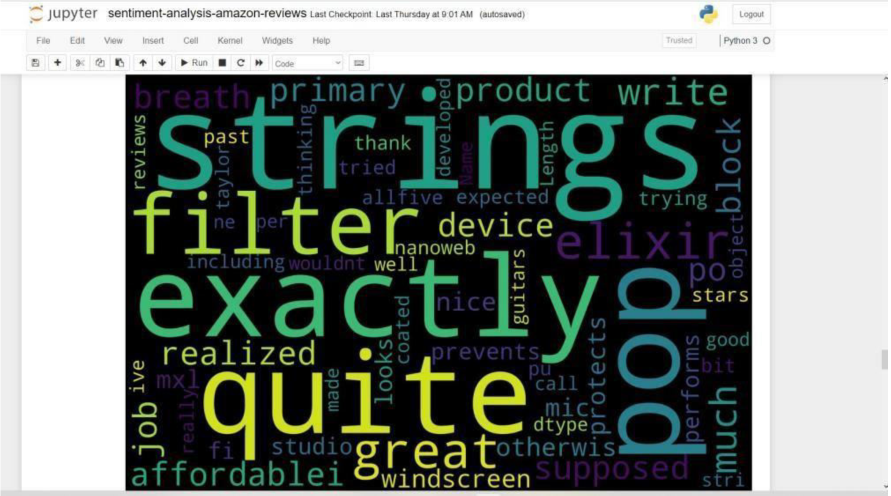
6



localhost:8888/notebooks/sentiment-analysis-amazon-reviews.ipynb
jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 O
Run
Code
Review Text Word Count Distribution
3500
3000
2500
2000
1500
1000
500
0
count
0
200
400
600
800
1000
word count
Export to plot.ly »

```
kind='hist',
bins=100,
xTitle='word count',
linecolor='black',
yTitle='count',
title='Review Text Word Count Distribution')
```

Fig. 6 text word count distribution



jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 O
Run
Code
breath primary product write
strings
past thank tried developed Name length block reviews taylor thinking
allfive expected trying
filter ne per device elixir object
nanoweb including wouldnt well po
exactly nice guitars stars
pop
realized coated looks prevents good bit
quite ive mxl made pu call protects performs
mic really job fi studio dtype much
great otherwis
affordablei windscreen supposed stri

Fig. 7 word-cloud positive reviews

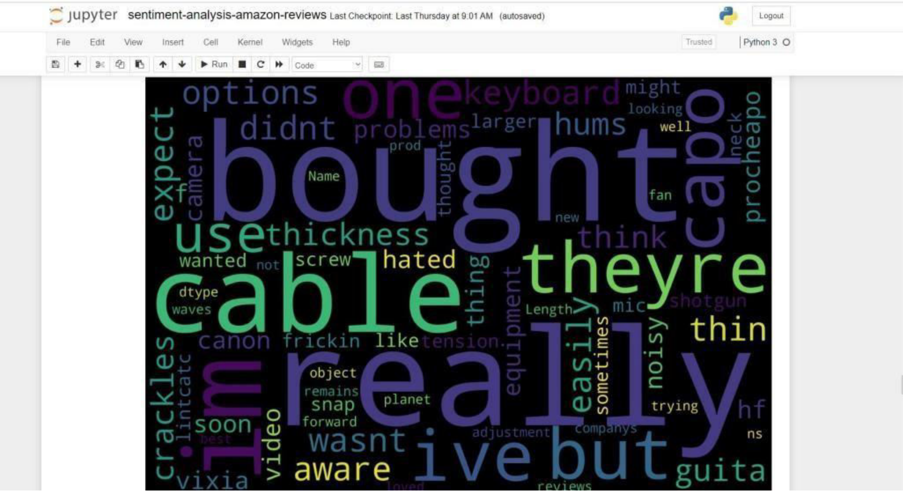
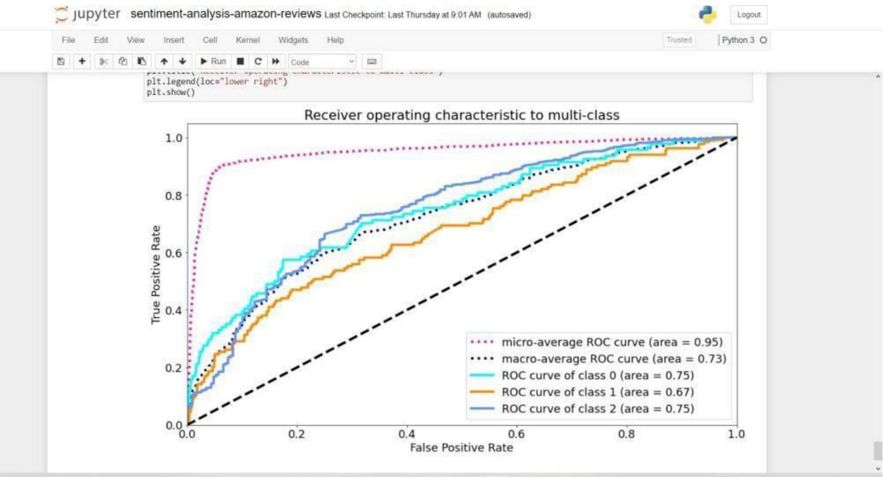


Fig. 8 word-cloud negative reviews


jupyter sentiment-analysis-amazon-reviews Last Checkpoint: Last Thursday at 9:01 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Receiver operating characteristic to multi-class
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
micro-average ROC curve (area = 0.95)
macro-average ROC curve (area = 0.73)
ROC curve of class 0 (area = 0.75)
ROC curve of class 1 (area = 0.67)
ROC curve of class 2 (area = 0.75)

```
plt.legend(loc="lower right")
plt.show()
```

Fig. 9 ROC-AUC curve

6 Conclusion

Sentiment Analysis or opinion mining is a case study which analyses people’s senti- ments, attitudes, entropy or emotions towards certain entities. This project tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. The data for this research is collected from online product reviews from Amazon.com. A process known as sentiment polarity categorization and POS has been proposed along with detailed descriptions of each step. These steps consist of pre-processing, pre- filtering, biasing, data accuracy etc. features which require the knowledge of machine learning. A lot of work in opinion mining and sentiments of customer reviews hasbeen conducted to mine opinions in form of document, sentence and feature level sentiment analysis. For future preferences, Opinion Mining can be carried out on set of discovered feature expressions extracted from reviews become a most interesting research area.More innovative and effective techniques have to be invented which should overcome the current challenges faced by Opinion Mining and SentimentAnalysis.

Acknowledgement

We would like to convey our heartfelt gratitude towards the Department of Information Technology at Vignan Institute of Technology and Science,Hyderabad,for providingour team with every tool and resource possible,assistance,and guidance necessary to complete this research.

References

[1] Apoorv Agarwal, BoyiXie Ilia Vovsha, Owen Rambow, RebeccaPassonneau, Sentiment Analysis of Twitter Data.

[2] Apoor v Agarwal, Jasneet Singh Sabharwal, End-to-End Sentiment Analysis of Twitter Data, Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data, pages 39–44,COLING 2012, Mumbai, December 2012.

[3]V.K. Singh, R. Piryani, A. Uddin,P. Waila, Sentiment analysis of movie reviews:A new feature-based heuristic for aspect-level sentiment classification, Conference Paper March 2013, DOI: 10.1109/iMac4s.2013.6526500

[4] Alexander Pak, Patrick Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining.

[5] Ajinkya Ingle, Anjali Kante, ShriyaSamak, Anita Kumari, Sentiment Analy-sis of Twitter Data Using Hadoop, International Journal of Engineering Research and General Science Volume 3.