International Journal of Data Analytics Research and Development (IJDARD)
Volume 1, Issue 1, January-December 2023, pp. 9–23, Article ID: IJDARD_01_01_002
Available online at https://iaeme.com/Home/issue/IJDARD?Volume=1&Issue=1
ISSN Online: 0369-2412 Journal ID: 7784-4587

DESIGN AND DEVELOPMENT OF SOCIAL MEDIA SENTIMENT ANALYSIS ON CUSTOMER REVIEWS OF AMAZON PRODUCTS FOR BUSINESS INTELLIGENCE IN PYTHON

Dr J. Komalalakshmi, MCA, B.ED, M.ED, MPHIL, PHD
Assistant Professor-Guest Faculty,
Department of Computer Applications, Bharathiar University,
Coimbatore-641041, Tamilnadu, India

ABSTRACT

Social media is used by people to communicate, share, and consume information in today's rapidly evolving technological environment. In order to give corporate intelligence a convenient and cohesive platform, social media mining combines social media, social network analysis, and data mining. The proposed social media sentiment analysis advocates Python in developing the sentiment model. In the proposed model, the CountVectorizer and TF-IDF (Term Frequency-Inverse Document Frequency) are used for feature extraction in natural language processing. The machine learning algorithms Random forest and logistic regression are used to calculate the dataset's accuracy and correctness. The suggested sentiment analysis's findings and insights provide insightful business intelligence that can help the organization grow.

Keywords: Social media Mining, Sentiment Analysis, Business intelligence, Data Analytics

Cite this Article: J. Komalalakshmi, Design and Development of Social Media Sentiment Analysis on Customer Reviews of Amazon Products for Business Intelligence in Python, International Journal of Data Analytics Research and Development (IJDARD), 1(1), 2023, pp. 9–23. https://iaeme.com/Home/issue/IJDARD?Volume=1&Issue=1

I. INTRODUCTION

Sentiment Analysis

Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral. It provides objective insights; Businesses can avoid personal bias associated with human reviewers by using artificial intelligence (AI)–based sentiment analysis tools. It builds better products and services. A sentiment analysis system helps companies improve their products and services based on genuine and specific customer feedback. Sentiment analysis are incremental because they give you an accurate picture of changing market trends and customer preferences, whatever industry you are in. Emotion mining from audience experience data from various sources such as social media platforms, review websites, news articles, and surveys, gives you critical insights for developing an impactful growth strategy that is essential for business longevity.

Intent Analysis

By examining the user's intention behind a communication and determining whether it pertains to an opinion, news, marketing, complaint, suggestion, appreciation, or query, intent analysis ups the ante. Intent-based analysis [1] distinguishes between a text's opinions and motivations. For instance, someone who is frustrated about having to change their battery online might want to contact customer care to help them with the problem. The suggested paper designs and develops a social media sentiment analysis of customer evaluations on Amazon products.

Amazon Product Reviews

Fig.1.1 Amazon Product Review

Small enterprises and companies with limited resources might expand by using the platform provided by Amazon. Additionally, because of its popularity, customers take the time to write in-depth reviews regarding the product and the brand. Thus, we may learn a lot about companies' products and how to improve their quality by evaluating that data. But a person is not able to analyse that volume of data.

II. REVIEW OF LITERATURE

Social Media Sentiment Analysis

Sentiment mining from social media listening helps you analyse audience intent and opinions expressed on various social platforms. You can get granular market analysis of customer likes and dislikes about products, brands, advertising content, and more through techniques such as TikTok social listening and Instagram social listening, for example. Similarly, you can harness market insights about a product from comments on a how-to video through YouTube video analysis.[2]

Doing so can give you much-needed information about products, your target demographic, the common themes in comments and their comparison across different social platforms, and more. Sentiment analysis,[3] also referred to as opinion mining, is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. This is a popular way for organizations to determine and categorize opinions about a product, service or idea.

Sentiment analysis involves the use of data mining, machine learning (ML), artificial intelligence and computational linguistics to mine text for sentiment and subjective information such as whether it is expressing positive, negative or neutral feelings.

Sentiment analysis systems help organizations gather insights into real-time customer sentiment, customer experience and brand reputation. Generally, these tools use text analytics to analyse online sources such as emails, blog posts, online reviews, customer support tickets, news articles, survey responses, case studies, web chats, tweets, forums and comments. Algorithms are used to implement rule-based, automatic or hybrid methods of scoring whether the customer is expressing positive words, negative words or neutral ones[4].

In addition to identifying sentiment, sentiment analysis can extract the polarity or the amount of positivity and negativity, subject and opinion holder within the text. This approach is used to analyse various parts of text, such as a full document or a paragraph, sentence or sub sentence. Sentiment analysis uses machine learning models to perform text analysis of human language.

The metrics used are designed to detect whether the overall sentiment of a piece of text is positive, negative or neutral.

Product Development

Emotion mining from customer feedback data, surveys, news reports and articles, social media listening, and other sources can give you clever insights into how you can improve your product so that it reaches more audiences. This is also very important when launching a new product, opening a store at a new location, changing business models, and such.

Amazon Product Reviews Sentiment Analysis in Python

The machine learning component uses Natural Language Processing (NLP) to analyse and solve the issue of big datasets. Anticipating a good or negative evaluation is the aim of the proposed design. After the website is scraped, millions of reviews may be included in the actual dataset. Thus Pre-processing is finished.

III. DESIGN AND METHODOLOGY

Sentiment analysis generally follows these steps:

1. Importing Libraries and Datasets
2. Pre - processing and cleaning the reviews
3. Analysis of the Dataset
4. Converting text into Vectors
5. Model training, Evaluation, and Prediction

1. Importing Libraries and Datasets
    - **Collect data.** The text being analysed is identified and collected. This involves using a web scraping bot or a scraping application programming interface.

      The amazon product dataset is imported using the below link.

      https://www.kaggle.com/datasets/dilekbarutu/amazon-review

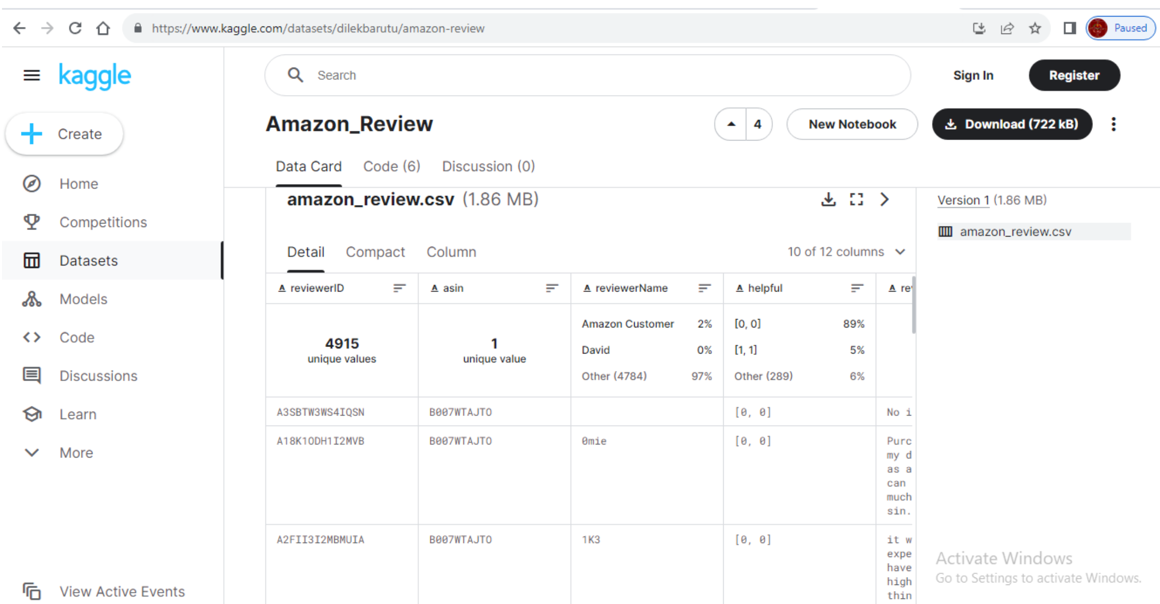


Fig 3.1 Amazon Review Products.

•

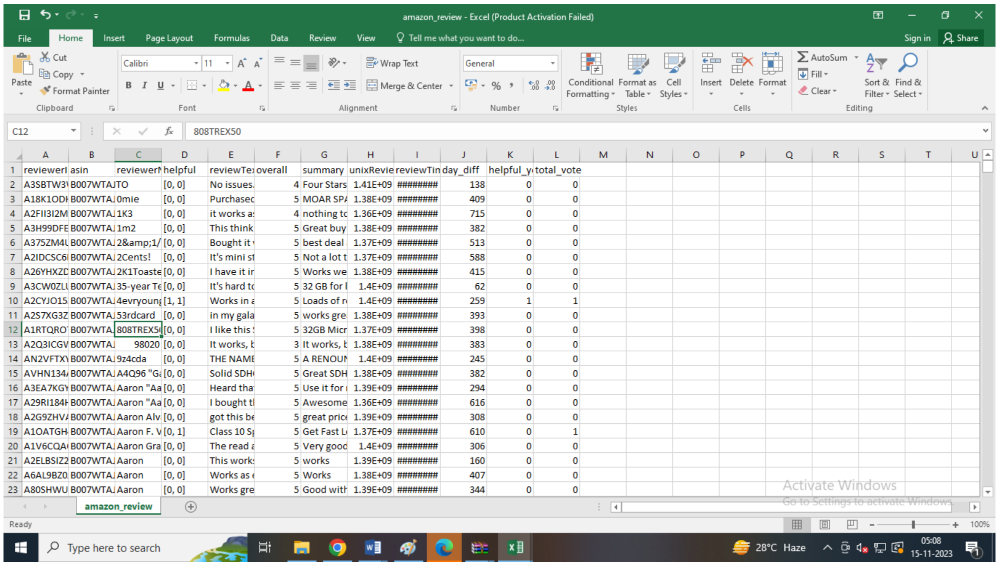


Fig 3.2 csv file-1

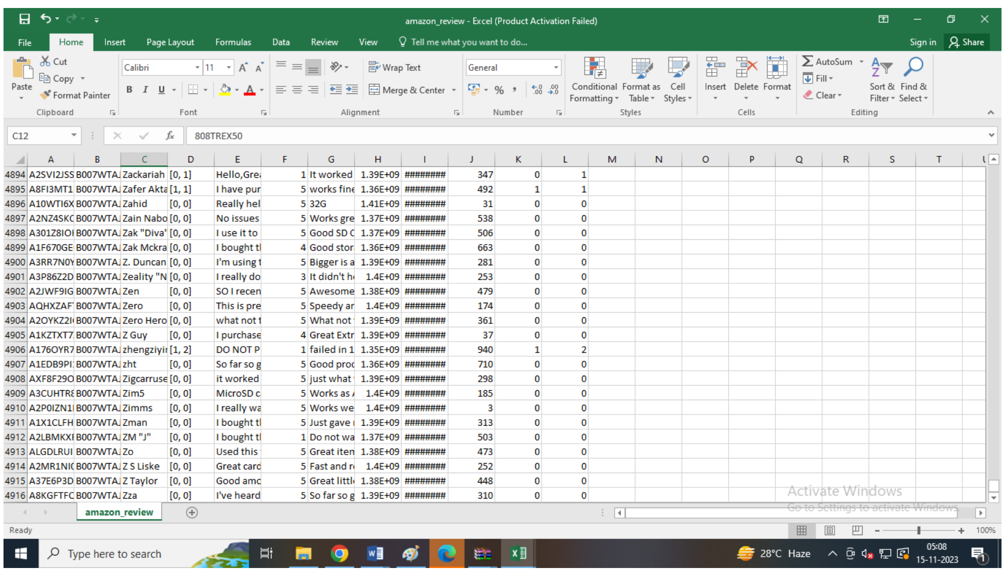


Fig 3.3 csv file-2

2. Pre - processing and cleaning the reviews

- **Clean the data.** The data is cleaned and processed to eliminate background noise and speech that doesn't contribute to the overall sentiment of the text.

The following text pre-processing is done to clean the data.

Normalizing, case folding, removing punctuation, removing numbers, removing stop words, removing rare words, lemmatize

3. **Converting text into Vectors**

- **Extract features.**

A variety of tools are available in this module to convert unprocessed data into formats that may be used with machine learning models. It is frequently utilized in jobs where the input data must be transformed into a numerical representation that machine learning algorithms can comprehend, such as text categorization, clustering, and regression.

A module called **sklearn. feature extraction** provided by Scikit-learn that includes various tools for feature extraction from raw data is used for feature extraction. Some of the key features include:

Text Feature Extraction

CountVectorizer: Converts a collection of text documents to a matrix of token counts

TfidfVectorizer: Converts a collection of raw documents to a matrix of TF-IDF features.

4. **Analysis of the Dataset**

- **ML model.**

Logistic Regression and Random Forest represent distinct machine learning approaches applied to sentiment analysis, a task within natural language processing (NLP) aimed at discerning the sentiment conveyed in a given text. In this context, sentiments are typically classified into categories such as positive, negative, or neutral. These algorithms serve as tools to automatically analyse and categorize textual content based on the expressed emotions or opinions. Logistic Regression employs a linear model suitable for binary classification, predicting whether the sentiment is positive or negative. On the other hand, Random Forest is an ensemble learning method that constructs multiple decision trees to collectively determine the sentiment by considering the majority class across the individual trees. Both algorithms offer valuable approaches for sentiment modelling, each with its strengths and applications in NLP tasks.

- **Logistic regression model**

The logistic regression hypothesis can be expressed as follows

$$h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+...+\theta_n x_n)}}$$

- $h_0(x)$ is the predicted probability that the dependent variable (output) is 1 given in the input features $x_1, x_2, \dots x_n$

- $\Theta_0, \Theta_1, \dots, \Theta_n$ are the parameters (weights) that the algorithms learns during training.

- e is the base of the natural algorithm

- **Random forest model**

Random Forest is an ensemble learning algorithm that constructs a multitude of decision trees during training. Each decision tree in the forest independently predicts the class, and the final prediction is determined by a majority vote or averaging (for regression problems) across all trees. The general formula for the prediction in a Random Forest is:

$$\text{Prediction} = \text{Majority Vote}\,(\text{Tree}_1(x), \text{Tree}_2(x)\ldots., \text{Tree}_k(x))$$

Where

- $\text{Tree}_1(x), \text{Tree}_2(x)\ldots., \text{Tree}_k(x)$ are the individual decision tree predictions
- The final prediction is the majority vote or average, depending on the task

5. Model training, Evaluation, and Prediction

- **Sentiment classification.**

Once a model is picked and used to analyse a piece of text, it assigns a sentiment score to the text including positive, negative or neutral. Organizations can also decide to view the results of their analysis at different levels, including document level, which pertains mostly to professional reviews and coverage; sentence level for comments and customer reviews; and sub-sentence level, which identifies phrases or clauses within sentences.

The accuracy and prediction is done for both logistic regression and random forest using the **Count Vectorizer and TF-IDF.**

IV. DEVELOPMENT OF PROPOSED WORK

The following procedure explains development of the Social media sentiment analysis on customer reviews of amazon products using Python.

STEP1: Choose a programming language as Python.

STEP 2: Import necessary library packages to perform Social Sentiment Analysis.

STEP 3: Load the dataset from Kaggle on amazon products: https://www.kaggle.com/datasets/dilekbarutu/amazon-review, and perform textpreprocessing.

STEP 4: By selecting particular column, the text visualization has been done using bar plot and word cloud to find the term frequencies.

STEP 5: Perform feature engineering to create labels and split the dataset into trainand test.

STEP 6: To create the sentiment modelling, the logistic regression and randomforest have been used to find the accuracy of the dataset.

STEP 7: The CountVectorizer and TF-IDF are used in logistic regression andrandom forest for feature extraction.

STEP 8: Prepare the report and print the result.

2. PYTHON PROGRAMMING

a) IMPORTING NECESSARY LIBRARIES

```
!pip install nltk
!pip install textblob
!pip install wordcloud
from warnings import filterwarnings
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from PIL import Image
from nltk.corpus import stopwords
from nltk.sentiment import SentimentIntensityAnalyzer
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score, GridSearchCV, cross_validate, train_test_split
from sklearn.preprocessing import LabelEncoder
from textblob import Word, TextBlob
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
```

```
import nltk
nltk.download("stopwords")
nltk.download("wordnet")
nltk.download("vader_lexicon")

filterwarnings("ignore")
pd.set_option("display.max_columns", None)
pd.set_option("display.width", 500)
pd.set_option("display.float_format", lambda x: '%.2f' % x)
```

b) IMPORTING DATASET

```
df = pd.read_csv("/content/amazon_review.csv", sep=",")
df.head()
```

c) TEXT PREPROCESSING

```
def text_preprocessing(dataframe, dependent_var):
  # Normalizing Case Folding - Uppercase to Lowercase
  dataframe[dependent_var] = dataframe[dependent_var].apply(lambda x: " ".join(x.lower() for x in str(x).split()))
   # Removing Punctuation
  dataframe[dependent_var] = dataframe[dependent_var].str.replace('[^\w\s]','')
   # Removing Numbers
  dataframe[dependent_var] = dataframe[dependent_var].str.replace('\d','')
   # StopWords
  sw = stopwords.words('english')
  dataframe[dependent_var] = dataframe[dependent_var].apply(lambda x: " ".join(x for x in x.split() if x not in sw))
   # Remove Rare Words
  temp_df = pd.Series(' '.join(dataframe[dependent_var]).split()).value_counts()
  drops = temp_df[temp_df <= 1]
  dataframe[dependent_var] = dataframe[dependent_var].apply(lambda x: " ".join(x for x in str(x).split() if x not in drops))
    # Lemmatize
  dataframe[dependent_var] = dataframe[dependent_var].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))

  return dataframe
```

```
df = text_preprocessing(df, "reviewText")
```

```
df["reviewText"].head()
```

d) TEXT VISUALIZATION

```
def text_visulaization(dataframe, dependent_var, barplot=True, wordcloud=True):
    # Calculation of Term Frequencies
    tf = dataframe[dependent_var].apply(lambda x: pd.value_counts(x.split(" "))).sum(axis=0).reset_index()
    tf.columns = ["words", "tf"]
    if barplot:
        # Bar Plot
        tf[tf["tf"]>1000].plot.barh(x="words", y="tf")
        plt.title("Calculation of Term Frequencies : barplot")
        plt.show()
    if wordcloud:
        # WordCloud
        text = " ".join(i for i in dataframe[dependent_var])
        wordcloud = WordCloud(max_font_size=100, max_words=1000, background_color="white").generate(text)
        plt.figure(figsize=[10, 10])
        plt.imshow(wordcloud, interpolation="bilinear")
        plt.axis("off")
        plt.title("Calculation of Term Frequencies : wordcloud")
        plt.show()
        wordcloud.to_file("wordcloud.png")
```

```
text_visulaization(df, "reviewText")
```

e) SENTIMENT ANALYSIS

```
def create_polarity_scores(dataframe, dependent_var):
    sia = SentimentIntensityAnalyzer()
    dataframe["polarity_score"] = dataframe[dependent_var].apply(lambda x: sia.polarity_scores(x)["compound"])
```

```
create_polarity_scores(df, "reviewText")
df.head()
```

f) FEATURE ENGINEERING

```
# Create Lables
def create_label(dataframe, dependent_var, independent_var):
    sia = SentimentIntensityAnalyzer()
    dataframe[independent_var] = dataframe[dependent_var].apply(lambda x: "pos" if sia.polarity_scores(x)["compound"] > 0 else "neg")
    dataframe[independent_var] = LabelEncoder().fit_transform(dataframe[independent_var])

    X = dataframe[dependent_var]
    y = dataframe[independent_var]

    return X, y
```

```
X, y = create_label(df, "reviewText", "sentiment_label")
```

```
# Split Dataset
def split_dataset(dataframe, X, y):
    train_x, test_x, train_y, test_y = train_test_split(X, y, random_state=1)
    return train_x, test_x, train_y, test_y
```

```
train_x, test_x, train_y, test_y = split_dataset(df, X, y)
```

```
def create_features_count(train_x, test_x):
  # Count Vectors
  vectorizer = CountVectorizer()
  x_train_count_vectorizer = vectorizer.fit_transform(train_x)
  x_test_count_vectorizer = vectorizer.fit_transform(test_x)

  return x_train_count_vectorizer, x_test_count_vectorizer
```

```
x_train_count_vectorizer, x_test_count_vectorizer = create_features_count(train_x, test_x)
```

```
def create_features_TFIDF_word(train_x, test_x):
  # TF-IDF word
  tf_idf_word_vectorizer = TfidfVectorizer()
  x_train_tf_idf_word = tf_idf_word_vectorizer.fit_transform(train_x)
  x_test_tf_idf_word = tf_idf_word_vectorizer.fit_transform(test_x)

  return x_train_tf_idf_word, x_test_tf_idf_word
```

```
x_train_tf_idf_word, x_test_tf_idf_word = create_features_TFIDF_word(train_x, test_x)
```

g) SENTIMENT MODELING - CREATE MODEL

The feature extraction CountVectorizer and TF-IDF are used as follows.

```
#Logistic Regression
def crate_model_logistic(train_x, test_x):
  # Count
  x_train_count_vectorizer, x_test_count_vectorizer = create_features_count(train_x, test_x)
  log_count = LogisticRegression(solver='lbfgs', max_iter=1000)
  log_model_count = log_count.fit(x_train_count_vectorizer, train_y)
  accuracy_count = cross_val_score(log_model_count, x_test_count_vectorizer, test_y, cv=10).mean()
  print("Accuracy - Count Vectors: %.3f" % accuracy_count)

  # TF-IDF Word
  x_train_tf_idf_word, x_test_tf_idf_word = create_features_TFIDF_word(train_x, test_x)
  log_word = LogisticRegression(solver='lbfgs', max_iter=1000)
  log_model_word = log_word.fit(x_train_tf_idf_word, train_y)
  accuracy_word = cross_val_score(log_model_word, x_test_tf_idf_word, test_y, cv=10).mean()
  print("Accuracy - TF-IDF Word: %.3f" % accuracy_word)

  return log_model_count, log_model_word
```

```
log_model_count, log_model_word = crate_model_logistic(train_x, test_x)
```

```python
# Random Forest
def crate_model_randomforest(train_x, test_x):
    # Count
    x_train_count_vectorizer, x_test_count_vectorizer = create_features_count(train_x, test_x)
    rf_count = RandomForestClassifier()
    rf_model_count = rf_count.fit(x_train_count_vectorizer, train_y)
    accuracy_count = cross_val_score(rf_model_count, x_test_count_vectorizer, test_y, cv=10).mean()
    print("Accuracy - Count Vectors: %.3f" % accuracy_count)

    # TF-IDF Word
    x_train_tf_idf_word, x_test_tf_idf_word = create_features_TFIDF_word(train_x, test_x)
    rf_word = RandomForestClassifier()
    rf_model_word = rf_word.fit(x_train_tf_idf_word, train_y)
    accuracy_word = cross_val_score(rf_model_word, x_test_tf_idf_word, test_y, cv=10).mean()
    print("Accuracy - TF-IDF Word: %.3f" % accuracy_word)

    return rf_model_count, rf_model_word
```

```python
rf_model_count, rf_model_word = crate_model_randomforest(train_x, test_x)
```

h) PREDICTION

```python
def predict_count(train_x, model, new_comment):
    new_comment= pd.Series(new_comment)
    new_comment = CountVectorizer().fit(train_x).transform(new_comment)
    result = model.predict(new_comment)
    if result==1:
        print("Comment is Positive")
    else:
        print("Comment is Negative")
```

```python
# Logistic Regression
predict_count(train_x, model=log_model_count, new_comment="this product is very good :)")
```

```python
# Random Forest
predict_count(train_x, model=rf_model_count, new_comment="this product is very bad :)")
```

```python
# Sample Review
new_comment=pd.Series(df["reviewText"].sample(1).values)
new_comment
```

```python
# Sample Review - Random Forest
predict_count(train_x, model=rf_model_count, new_comment=new_comment)
```

V.RESULTS

The output of the model is explained in screenshot as follows.

b) LOADING THE DATASET

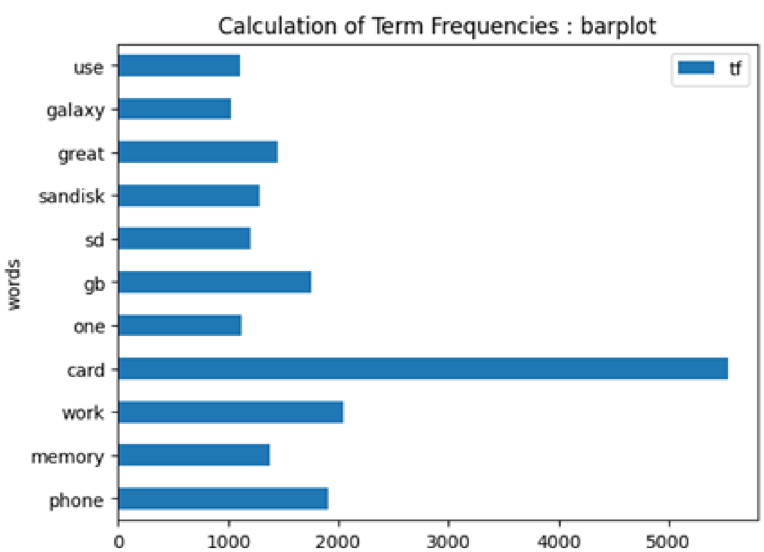
	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime	day_diff	helpful_yes	total_vote
0	A3SBTW3WS4IQSN	B007WTAJTO	NaN	[0, 0]	No issues.	4.00	Four Stars	1406073600	2014-07-23	138	0	0
1	A18K1ODH1I2MVB	B007WTAJTO	0mie	[0, 0]	Purchased this for my device, it worked as adv...	5.00	MOAR SPACE!!!	1382659200	2013-10-25	409	0	0
2	A2FII3I2MBMUIA	B007WTAJTO	1K3	[0, 0]	it works as expected. I should have sprung for...	4.00	nothing to really say....	1356220800	2012-12-23	715	0	0
3	A3H99DFEG68SR	B007WTAJTO	1m2	[0, 0]	This think has worked out great.Had a diff. br...	5.00	Great buy at this price!!! *** UPDATE	1384992000	2013-11-21	382	0	0
4	A375ZM4U047O79	B007WTAJTO	2&1/2Men	[0, 0]	Bought it with Retail Packaging, arrived legit...	5.00	best deal around	1373673600	2013-07-13	513	0	0

c) TEXT PREPROCESSING

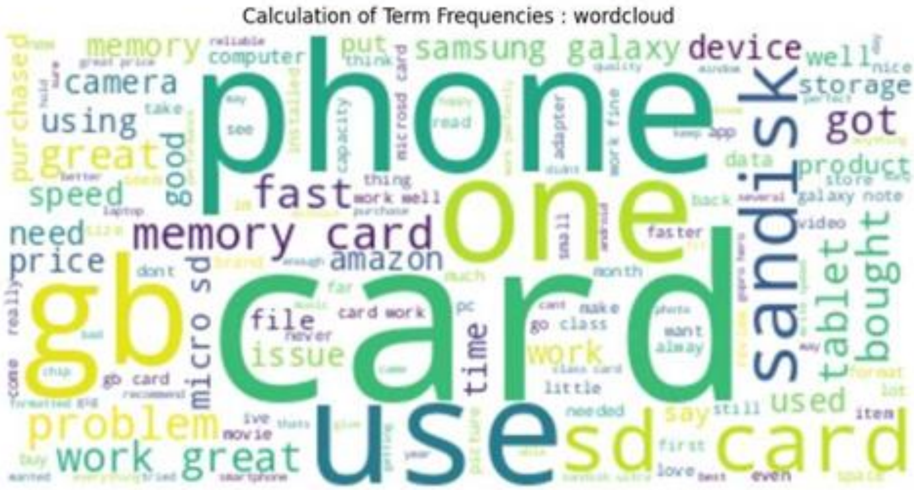
```
0                                                 issue
1     purchased device worked advertised never much ...
2     work expected higher capacity think made bit e...
3     think worked gb card went south one held prett...
4     bought retail packaging arrived legit envelope...
Name: reviewText, dtype: object
```

d) TEXT VISUALIZATION

CALCULATION OF TERM FREQUENCIES : BARPLOT



CALCULATION OF TERM FREQUENCIES : WORD CLOUD



e) SENTIMENT ANALYSIS

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime	day_diff	helpful_yes	total_vote	polarity_score
0	A3SBTW3WS4IQSN	B007WTAJTO	NaN	[0, 0]	issue	4.00	Four Stars	1406073600	2014-07-23	138	0	0	0.00
1	A18K1ODH1I2MVB	B007WTAJTO	0mie	[0, 0]	purchased device worked advertised never much ...	5.00	MOAR SPACE!!!	1382659200	2013-10-25	409	0	0	0.00
2	A2FII3I2MBMUIA	B007WTAJTO	1K3	[0, 0]	work expected higher capacity think made bit e...	4.00	nothing to really say....	1356220800	2012-12-23	715	0	0	0.40
3	A3H99DFEG68SR	B007WTAJTO	1m2	[0, 0]	think worked gb card went south one held prett...	5.00	Great buy at this price!!! *** UPDATE	1384992000	2013-11-21	382	0	0	0.65
4	A375ZM4U047O79	B007WTAJTO	2&1/2Men	[0, 0]	bought retail packaging arrived legit envelope...	5.00	best deal around	1373673600	2013-07-13	513	0	0	0.86

f) FEATURE ENGINEERING

The CountVectorizer and TF-IDF are applied on the trained dataset for testing.The obtained test data is the review report.

g) SENTIMENT MODELING - CREATE MODEL

LOGISTIC REGRESSION

```
Accuracy - Count Vectors: 0.832
Accuracy - TF-IDF Word: 0.801
```

RANDOM FOREST

```
Accuracy - Count Vectors: 0.810
Accuracy - TF-IDF Word: 0.801
```

h) PREDICTION

```
Comment is Negative
```

```
Comment is Positive
```

```
0    fast give alot space love free app come contro...
dtype: object
```

```
Comment is Positive
```

VI. DISCUSSION

Insight Summary on Social Media Sentiment Analysis on Customer Reviews of Amazon Products:

Thus the social media sentiment model predicts the review of the customer and their sentiment is analysed as follows.

- The sentiment falls on both positive and negative sentiments.
- The reviews reflect highly positive comment over amazon is predicted for the given dataset.
- The sentiment analysis on customer reviews on products in social media mining has been designed and developed

VII. CONCLUSION

The author assures that the proposed model can be useful for academicians and students to learn and progress in their career as data analyst. The model can be extended to any particular domain like health care, cosmetics, science and technology, e commerce so on and so forth.

VIII. ACKNOWLEDGEMENT

The author expressed her sincere thanks to Dr.M. Punithavalli, Head of The Department, Department of Computer Application, Bharathiar University, for being mentor and inspiring all the way of her career. The author submits her due credit to the online dataset available for academic research purposes to enhance the progress.

REFERNCES

[1] A. Abbasi. 2010. Intelligent feature selection for opinion classification. IEEE Intell. Syst. 25, 4 (2010), 75--79.

[2] https://www.repustate.com/blog/sentiment-analysis-benefits/

[3] https://www.techtarget.com/searchbusinessanalytics/definition/opinion-mining-sentiment-mining

[4] https://link.springer.com/article/10.1007/s13278-021-00776-6

Citation: Komalalakshmi, Design and Development of Social Media Sentiment Analysis on Customer Reviews of Amazon Products for Business Intelligence in Python, International Journal of Data Analytics Research and Development (IJDARD), 1(1), 2023, pp. 9–23

Abstract Link:
https://iaeme.com/Home/article_id/IJDARD_01_01_002

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJDARD/VOLUME_1_ISSUE_1/IJDARD_01_01_002.pdf

editor@iaeme.com