VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

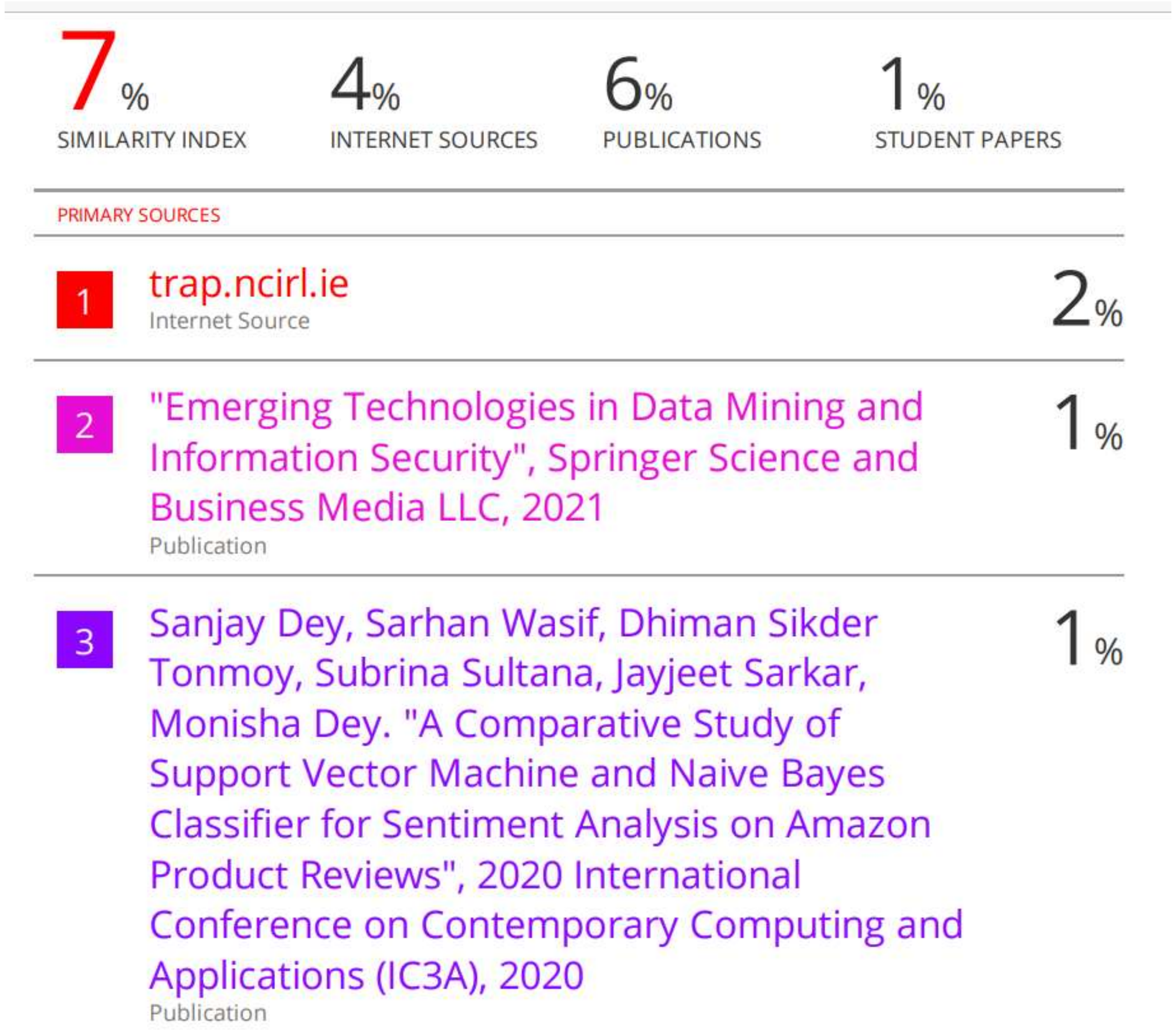
## School of Computer Science and Engineering

# A COMPARATIVE STUDY OF CLASSIFIERS ON SENTIMENT ANALYSIS FOR AMAZON PRODUCT REVIEWS

**ABIRAMI**
20MCA1024

*Research Supervisor*
**Dr. M. Sivabalakrishnan**

# Plagiarism report



| 7% | 4% | 6% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | trap.ncirl.ie<br>Internet Source | 2% |
| 2 | "Emerging Technologies in Data Mining and Information Security", Springer Science and Business Media LLC, 2021<br>Publication | 1% |
| 3 | Sanjay Dey, Sarhan Wasif, Dhiman Sikder Tonmoy, Subrina Sultana, Jayjeet Sarkar, Monisha Dey. "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews", 2020 International Conference on Contemporary Computing and Applications (IC3A), 2020<br>Publication | 1% |

# *Outline:*

- Introduction
- Objective
- Literature Survey
- Summary of Literature Survey
- Proposed Solution
- Dataset & Methodology
- Data Preprocessing
- Simple Word-Cloud
- Implementation of Models
- Results

# Introduction

- *NLP is a broader term that helps in interactions between human language and computer ,especially in processing and analyzing large amounts of natural language data.*
- **Sentiment analysis /opinion mining /emotion AI:** refers to the use of natural language processing, text analysis, computational linguistics, and to systematically identify, extract, quantify, and study affective states and subjective information.
- Sentiment analysis is applied to the customer reviews or survey responses in order to understand the targeted audience more.

# Objective

- *The objective is to make Sentiment Analysis on the customer reviews and to provide the best classification model to derive an accurate result to help the organization grow .*

# Literature survey

| S NO | TITLE | OBSERVATION | CONCLUSIONS |
|---|---|---|---|
| 1. | Sentiment Analysis Of Customer Product Reviews Using ML<br>*Zeenia Singla, Sukhchandan Randhawa, Sushma Jain*<br>2017 International Conference on Intelligent Computing and Control (I2C2)<br>DOI: 10.1109/I2C2.2017.8321910 | ✓ In this paper , e-commerce reviews from Amazon for smartphones were taken.<br>✓ In this paper various comparison of classification models like Naïve Bayesian SVM and Decision tree were made<br>✓ Predicting the Accuracy of each model using cross validation algorithm by using Syuzhet Package. | ✓ The accuracy results have been cross validated and the highest value of accuracy achieved was 81.75% for SVM.<br>✓ Among the three models while Naïve Bayes model has the least predictive accuracy. (64.57%) , in result suggesting the use of SVM model. |
| 2. | Sentiment Analysis Of Polarity In Product Reviews In Social Media<br>*Marium Nafees, Hafsa Dar, Ikram Ullah Lali, Salman Tiwana*<br>2018 14th International Conference on Emerging Technologies (ICET)<br>DOI:10.1109/ICET.2018.8603585 | ✓ Polarity of product reviews from Twitter social media.<br>✓ Labelled using hybrid approach for text and emoticons.<br>✓ Use of classifiers like SVM , NB and Logistic regression to evaluate the accuracy and to determine the best case. | ✓ After analyzing and classifying the tweets to measure the effectiveness of data ,we find the polarity analysis on the data.<br>✓ SVM is considered an efficient and best model because of its maximum accuracy outcomes. |
| 3. | Sentiment Analysis Of Restaurant Customer Reviews On Tripadvisor Using Naive Bayes<br>*Rachmawan Adi Laksono, Kelly Rossa SungkonoRiyanarto Sarno, Cahyaningtyas Sekar Wahyuni*<br>2019 12th International Conference on Information & Communication Technology and System (ICTS)<br>DOI: 10.1109/ICTS.2019.8850982 | ✓ Sentiment analysis on a restaurant's review through NB and Textblob classification process.<br>✓ Data sampling is crawled by using WebHarvy Tools to scrap data from Trip Advisor website. | ✓ Based on the results of accuracy,Naïve baye's is found to be much accurate than textblob sentiment analysis.<br>✓ Issue: with only 2.94% of accuracy difference , the efficiency is questionable |

# Literature survey(Contd)

| S NO | TITLE | OBSERVATION | CONCLUSIONS |
|---|---|---|---|
| 4. | **Sentiment analysis of smart phone product review using SVM Classification Technique**<br>*Upma Kumari,DR. Aravind K Sharma ,Dinesh Soni*<br>2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)<br>DOI:10.1109/ICECDS.2017.8389689 | ✓ Sentiment analysis on a collection of smart phones to determine their performance.<br>✓ Obtaining accuracy for individual product kind.<br>✓ Experimental work was implemented in JAVA programming language,JDK, and WAMP Server Mysql. | ✓Based on the results of accuracy,<br>✓In comparison with other methodologies used by several authors, this proposed work on SVM provides higher accuracy rate. |
| 5. | **Real-time Sentiment Analysis On E-Commerce Application**<br>*Jahanzeb Jabbar • Iqra Urooj • Wu JunSheng • Naqash Azeem -*<br>2019 IEEE 16th International Conference on Networking, Sensing and Control<br>DOI: https://doi.org/10.1109/ICNSC.2019.8743331 | ✓ A real-time sentimental analysis on the reviews of e-commerce application to enhance user experience .Implementation divided into 2 parts:-<br>✓ Sentiment analysis model containing-NLTK and SVM model , NLTK to train the model while SVM used for evaluating.<br>✓ E-commerce application development-integrating developed model into an e-commerce application for prediction of reviews. | ✓Saves the time of customers and service providers for product evaluation.<br>✓Works well on simple sentences, tough on complex structures of sentences. |
| 6. | **Comparative polarity analysis on Amazon product reviews using existing machine learning algorithms**<br>*Karthikayini T • N.K. Srinath*<br>2nd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions 2017<br>DOI: https://doi.org/10.1109/CSITSS.2017.8447660 | ✓ Polarity analysis of product reviews.<br>✓ Comparison of Classification models such as NLTK and Datumbox. And proposing a new model-Senti<br>✓ Sentence level classifiers focuses on review text alone.<br>✓ Senti focuses on overall ratings along with review text of NLTK api. | ✓Senti outperforms both the existing ML models.<br>✓Inability to calculate negated statements in NLTK can affect the new Algorithm proposed.<br>✓Limited data being polarized cannot be sufficient to make business decisions. |

# Literature survey(Contd)

| S NO | TITLE | OBSERVATION | CONCLUSIONS |
|---|---|---|---|
| 7. | **Aspect-based Sentiment Analysis on mobile phone reviews with LDA-**<br>*Ye Yiran • Sangeet Srivastava*<br>ICMLT 2019:4th International Conference on Machine Learning Technologies(ICMLT)China June, 2019 –ACM<br>DOI: https://doi.org/10.1145/3340997.3341012 | ✓ Framework to perform Topic labeling and sentiment analysis on IPHONE X. . Over 4,00,000 training data and 1000. testing data was used.<br>✓ LDA Model to cluster Topic words with probability values.<br>✓ Sentiment labeling through<br>✓ ANALYSIS BY –<br>▪ Domain specific word lexicon- Stanford POS tagger<br>▪ SentiWord.Net | ✓Topic labeling – aspects – screen , camera and battery with 81 % ,77 %and 79% accuracy.<br>✓Sentiment labeling – 70% of positive feedback while negative feedback is slightly lower due to grammatical and typo errors.<br>✓Misleading data can not provide correct sentiment value. |
| 8. | **A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews-**<br>*Sanjay Dey • Sarhan Wasif • Dhiman Sikder Tonmoy • Subrina Sultana • Jayjeet Sarkar • and Monisha Dey* –<br>2020 International Conference on Contemporary Computing and Applications (IC3A)<br>DOI: https://doi.org/10.1109/IC3A48958.2020.233300 | ✓ Sentiment analysis on Books from Amazon.<br>✓ Comparison of 2 Models , NB and SVM.<br>✓ Standard statistical methods of Precision Recall and F1 score was analyzed.<br>✓ 6000 datasets were preprocessed , 2250 features were trained and almost 4000 test sets were passed through the model. | ✓Accuracy measurement of SVM and NB was made.<br>✓Predicts SVM(84%) has higher accuracy than NB.(81%)<br>✓Does not include Neutral comments. Only positive and negative opinions are mined and modelled. |
| 9. | **Opinion Mining and Sentiment Analysis on Online Customer Review-**<br>*Santhosh Kumar K L • Jayanti Desai • Jharna Majumdar* –<br>2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)<br>DOI: https://doi.org/10.1109/ICCIC.2016.7919584 | ✓Follows free review structure.<br>✓Compares 3 classifiers NB,Logistic regression and SentiWord.Net<br>✓Review data on Apple 5s ,Samsung J7 and Redmi note 3 was taken. | ✓Performance of NB is better than other classification models.<br>✓Better algorithms can also be used to improve their accuracy. |

# Literature survey(Contd)

| S NO | TITLE | OBSERVATION | CONCLUSIONS |
|---|---|---|---|
| 10. | **Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis**<br>*Yassine AL AMRANIa • Mohamed LAZAARb • Kamal Eddine EL KADIRI-*<br>Procedia computer science volume 127, 2018, pages 511-520,Elsevier<br>DOI: https://doi.org/10.1016/j.procs.2018.01.150 | ✓ A hybrid approach offered to identify product reviews.<br>✓ Comparison of RF , SVM and a new hybrid approach- RFSVM is made. | ✓Hybrid approach yields better results than other models.<br>✓Takes advantage of the individual RF and SVM characteristics.<br>✓Only a Small dataset is processed here. Results of large datasets is unknown w.r.t to the use of RFSVM. |
| 11. | **Predicting user preferences on changing trends and innovations using SVM based sentiment analysis**<br>*K. Chidambarathanu1 · K. L. Shunmuganathan2*<br>Cluster Computing volume 22, pp. 11877–11881(2019)<br>DOI: https://doi.org/10.1007/s10586-017-1505-0 | ✓ In this paper, an SVM model that combines the customer buying patterns based on the previous orders and the latest trend in the market was made.<br>✓ A filtration process of the recommendation system was the inspiration behind this idea<br>✓ A series of amazon and social network reviews were taken. | ✓ The idea is to improve the recommendation system by customer personal preferences and current trends.<br>✓ Combined method using SVM improved the quality of recommendation . |
| 12. | **Sentiment analysis using product review data**<br>*Xing Fang and Justin Zhan,*<br>Journal of Big data, June 2015, pp. 2-5.<br>DOI: https://doi.org/10.1186/s40537-015-0015-2 | ✓ This paper handles the polarization of sentiments that is one of the primary problems.<br>✓ Regular classification models were being used to represent sentence-level and review-text level categories. | ✓ SVM performs well in training of the dataset followed by NB and RF.<br>✓ F1 score on an average reaches 0.8 in sentence level categorization and 0.73 for review-level categorization. |

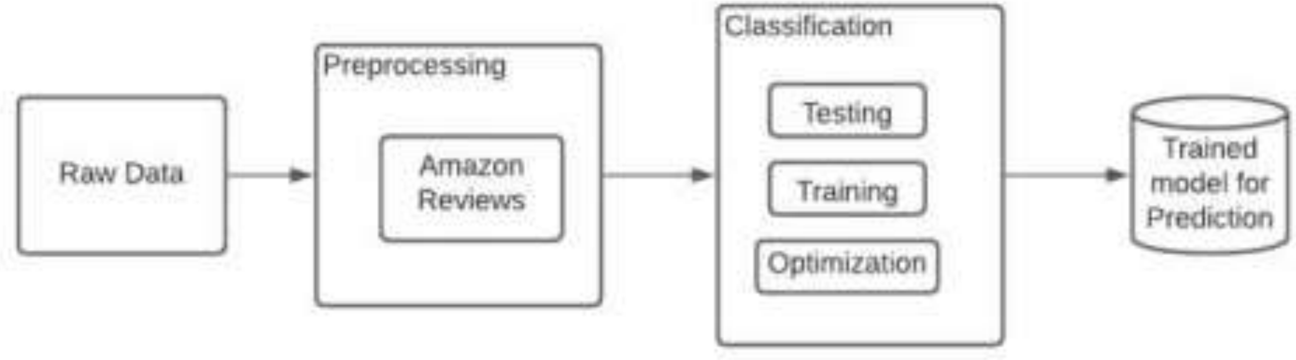
# Summary of Literature Survey:

- Suitable selection of classifier model gives better result.
- Minimum accuracy difference between classification models cannot determine the efficiency of the algorithm used or the data collected.
- From observing the literature papers , use of SVM or any other hybrid model can produce relatively a higher accuracy than the usual classification models.
- Since we consider only Supervised models – only by proper data preprocessing the efficiency of a model can be increased.
- Modelling using large datasets to ensure hybrid model's efficiency.

# Proposed work:

- Problem statement: Appropriate selection of a classification model among the observed models and proper processing of given data to show a higher accuracy than the previous papers claims.

- *My paper provides a comparative study between the classification models such as SVM, Random Forest, Logistic Regression, and Naïve Bayes model to determine the polarity of a product .*



# Dataset:

- Amazon Review Dataset of a smart phones were taken.
- Dataset contains One-plus and Redmi reviews.
- Dataset has almost 30,000 instances with 20 columns.

# Methodology

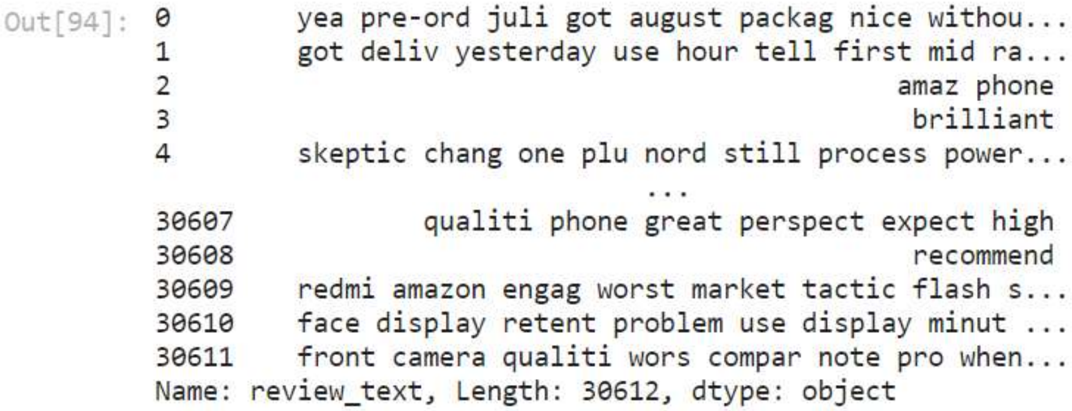
- Software: Anaconda.
- Language: Python-jupyter notebook
- *Key words*: NLTK,Textblob,sklearn-svm

# Data preprocessing

- Preprocessing involved the following processes:

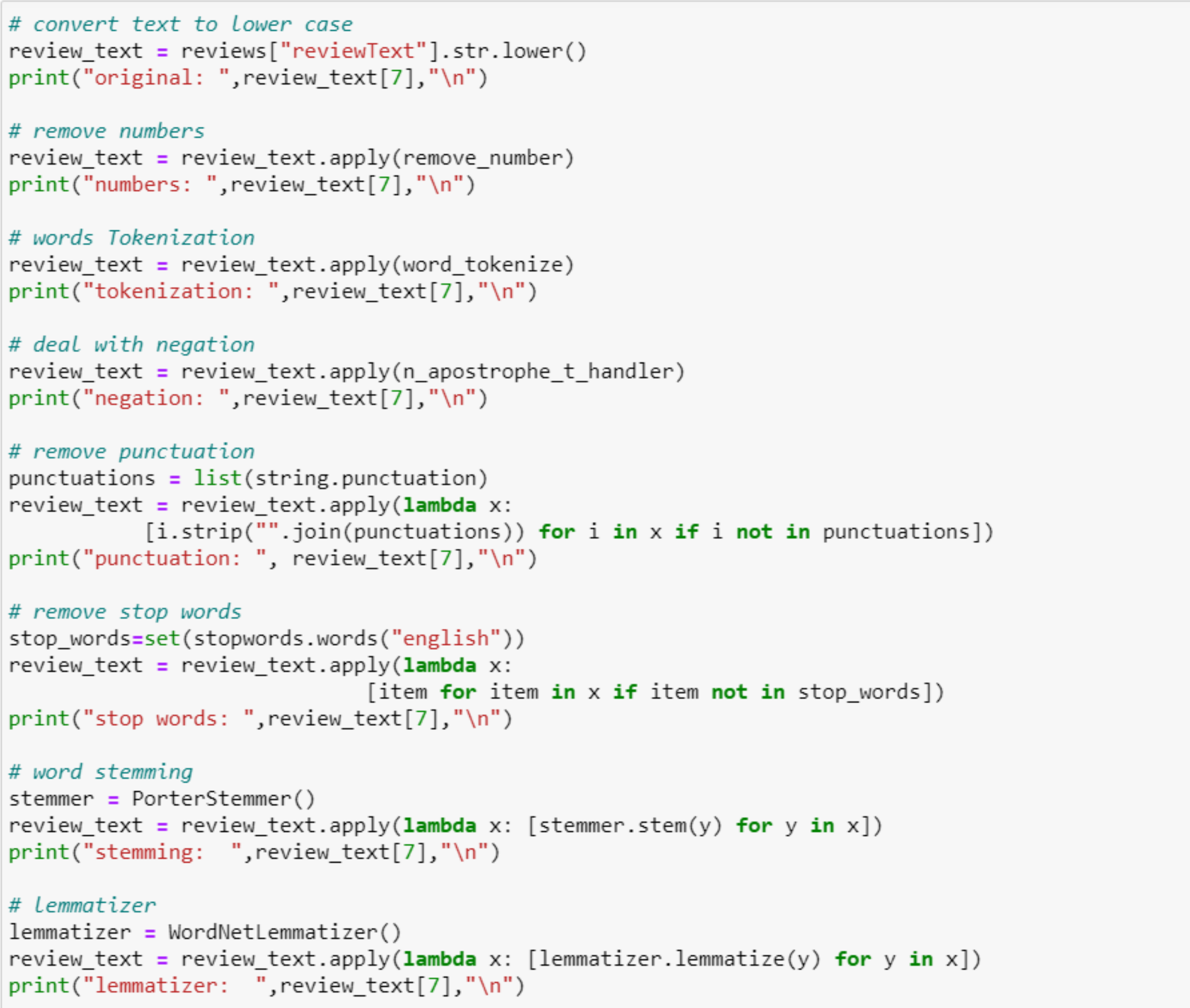
❑ removing unnecessary columns
❑ extracting rating values to numerical values
❑ filling the missing values
❑ removing numbers
❑ trimming lower case
❑ word tokenization
❑ dealing with negation
❑ removing punctuation
❑ removing stop words
❑ word stemming
❑ lemmatization

## Output:



```
Out[94]: 0        yea pre-ord juli got august packag nice withou...
         1        got deliv yesterday use hour tell first mid ra...
         2                                               amaz phone
         3                                                brilliant
         4        skeptic chang one plu nord still process power...
                                        ...
         30607            qualiti phone great perspect expect high
         30608                                            recommend
         30609    redmi amazon engag worst market tactic flash s...
         30610    face display retent problem use display minut ...
         30611    front camera qualiti wors compar note pro when...
         Name: review_text, Length: 30612, dtype: object
```

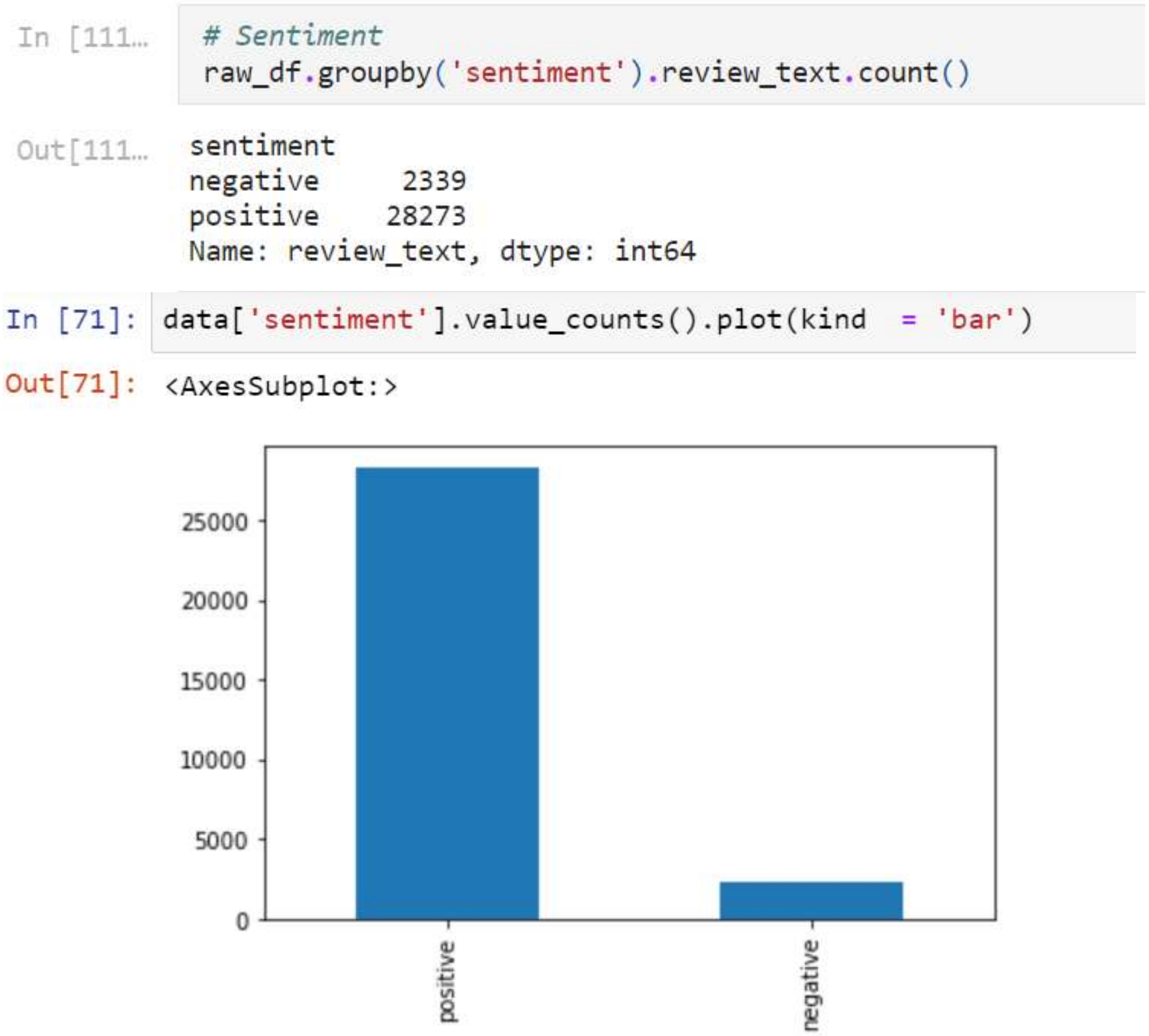
# Snippet of preprocessing



```
# convert text to lower case
review_text = reviews["reviewText"].str.lower()
print("original: ",review_text[7],"\n")

# remove numbers
review_text = review_text.apply(remove_number)
print("numbers: ",review_text[7],"\n")

# words Tokenization
review_text = review_text.apply(word_tokenize)
print("tokenization: ",review_text[7],"\n")

# deal with negation
review_text = review_text.apply(n_apostrophe_t_handler)
print("negation: ",review_text[7],"\n")

# remove punctuation
punctuations = list(string.punctuation)
review_text = review_text.apply(lambda x:
           [i.strip("".join(punctuations)) for i in x if i not in punctuations])
print("punctuation: ", review_text[7],"\n")

# remove stop words
stop_words=set(stopwords.words("english"))
review_text = review_text.apply(lambda x:
                             [item for item in x if item not in stop_words])
print("stop words: ",review_text[7],"\n")

# word stemming
stemmer = PorterStemmer()
review_text = review_text.apply(lambda x: [stemmer.stem(y) for y in x])
print("stemming:  ",review_text[7],"\n")

# lemmatizer
lemmatizer = WordNetLemmatizer()
review_text = review_text.apply(lambda x: [lemmatizer.lemmatize(y) for y in x])
print("lemmatizer:  ",review_text[7],"\n")
```

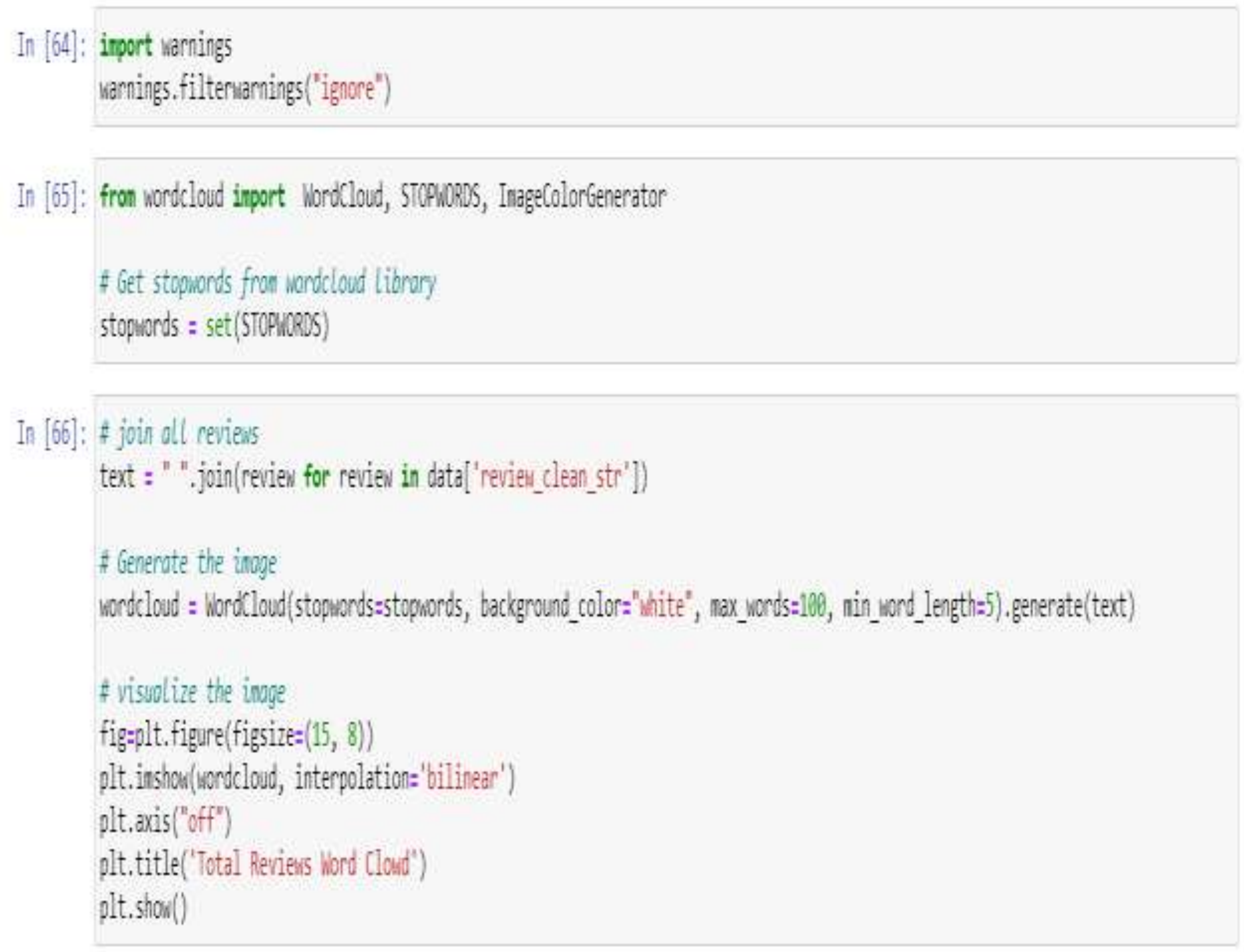
# Sentiment analysis-Textblob

- *TextBlob* is a Python library for processing textual data.
- Provides a simple API for natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
- Data is split into training and testing data.
- For instance : 30% of the total data was used as a testing data in my module.

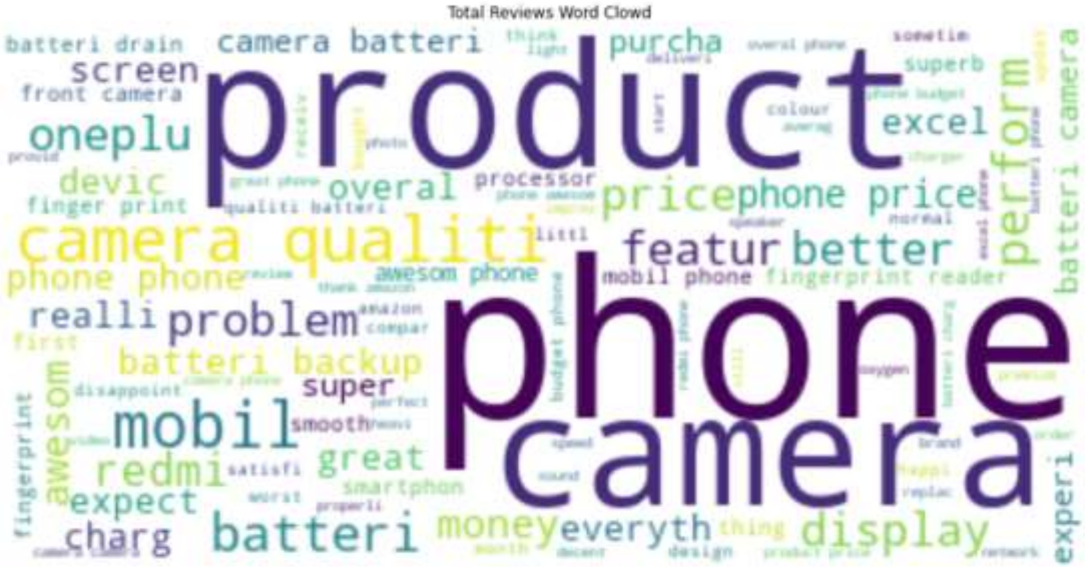


```
In [111…  # Sentiment
          raw_df.groupby('sentiment').review_text.count()

Out[111…  sentiment
          negative     2339
          positive    28273
          Name: review_text, dtype: int64

In [71]:  data['sentiment'].value_counts().plot(kind  =  'bar')

Out[71]:  <AxesSubplot:>
```

# Simple Word-Cloud for the cleaned reviews:



```
In [64]: import warnings
         warnings.filterwarnings("ignore")

In [65]: from wordcloud import  WordCloud, STOPWORDS, ImageColorGenerator

         # Get stopwords from wordcloud library
         stopwords = set(STOPWORDS)

In [66]: # join all reviews
         text = " ".join(review for review in data['review_clean_str'])

         # Generate the image
         wordcloud = WordCloud(stopwords=stopwords, background_color="white", max_words=100, min_word_length=5).generate(text)

         # visualize the image
         fig=plt.figure(figsize=(15, 8))
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis("off")
         plt.title('Total Reviews Word Cloud')
         plt.show()
```

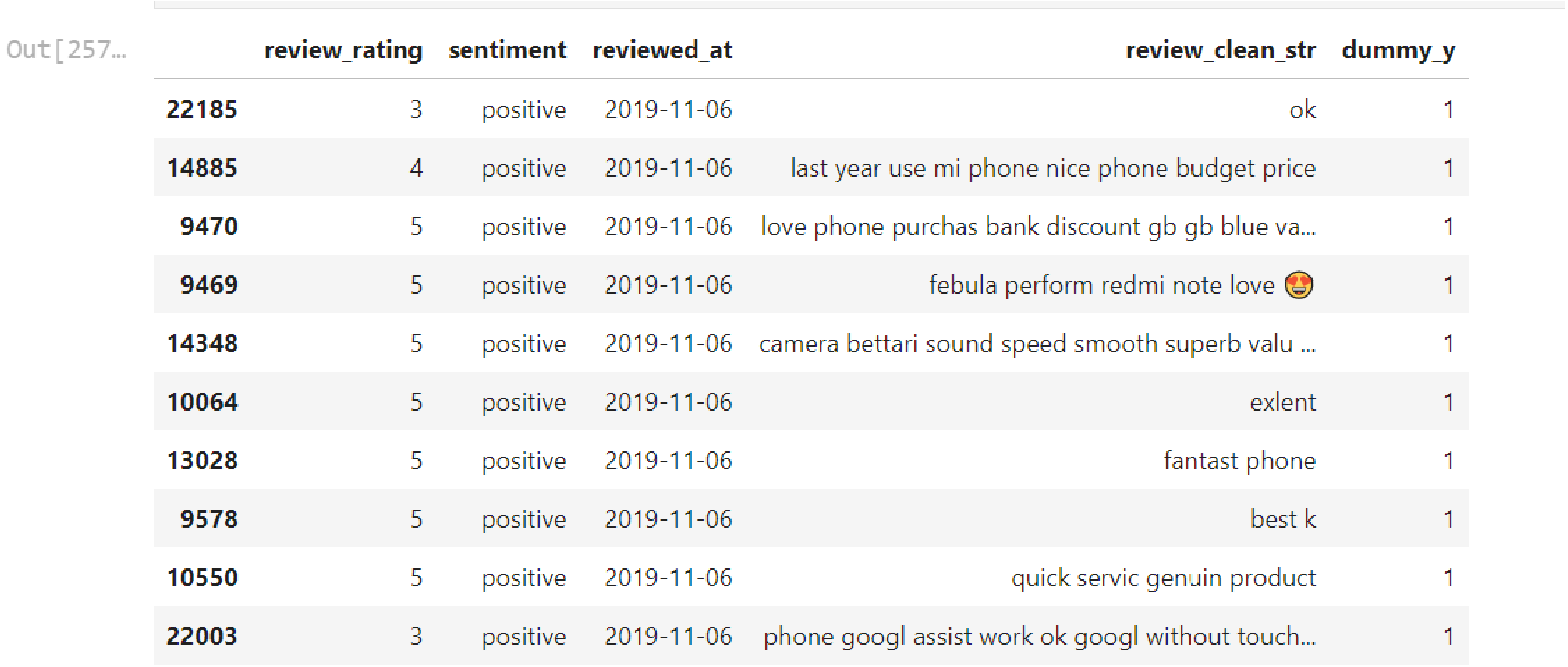


Total Reviews Word Clowd

# Implementation of SVM model:

- Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.
- The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).
- Use of sklearn library for svm classifier.

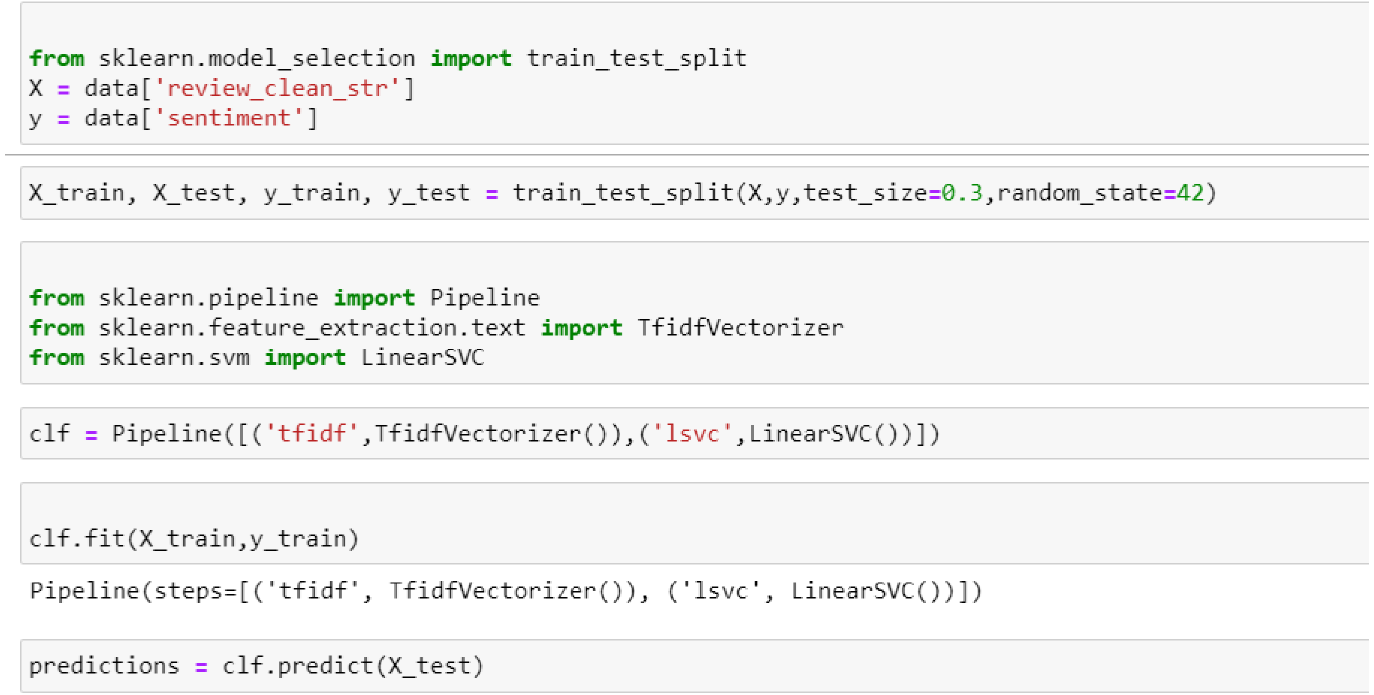
## Using subset of data



Out[257...

| | review_rating | sentiment | reviewed_at | review_clean_str | dummy_y |
|---|---|---|---|---|---|
| **22185** | 3 | positive | 2019-11-06 | ok | 1 |
| **14885** | 4 | positive | 2019-11-06 | last year use mi phone nice phone budget price | 1 |
| **9470** | 5 | positive | 2019-11-06 | love phone purchas bank discount gb gb blue va... | 1 |
| **9469** | 5 | positive | 2019-11-06 | febula perform redmi note love 😍 | 1 |
| **14348** | 5 | positive | 2019-11-06 | camera bettari sound speed smooth superb valu ... | 1 |
| **10064** | 5 | positive | 2019-11-06 | exlent | 1 |
| **13028** | 5 | positive | 2019-11-06 | fantast phone | 1 |
| **9578** | 5 | positive | 2019-11-06 | best k | 1 |
| **10550** | 5 | positive | 2019-11-06 | quick servic genuin product | 1 |
| **22003** | 3 | positive | 2019-11-06 | phone googl assist work ok googl without touch... | 1 |

# Results:



```
from sklearn.model_selection import train_test_split
X = data['review_clean_str']
y = data['sentiment']
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=42)
```

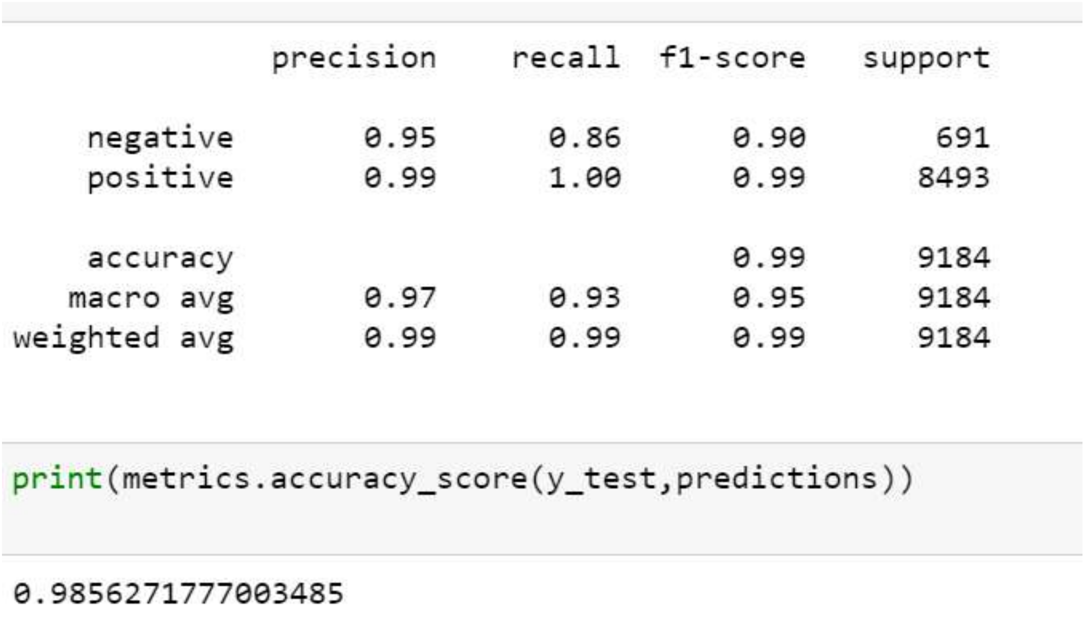
```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
```

```
clf = Pipeline([('tfidf',TfidfVectorizer()),('lsvc',LinearSVC())])
```

```
clf.fit(X_train,y_train)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('lsvc', LinearSVC())])
```

```
predictions = clf.predict(X_test)
```



```
              precision    recall  f1-score   support

    negative       0.95      0.86      0.90       691
    positive       0.99      1.00      0.99      8493

    accuracy                           0.99      9184
   macro avg       0.97      0.93      0.95      9184
weighted avg       0.99      0.99      0.99      9184
```

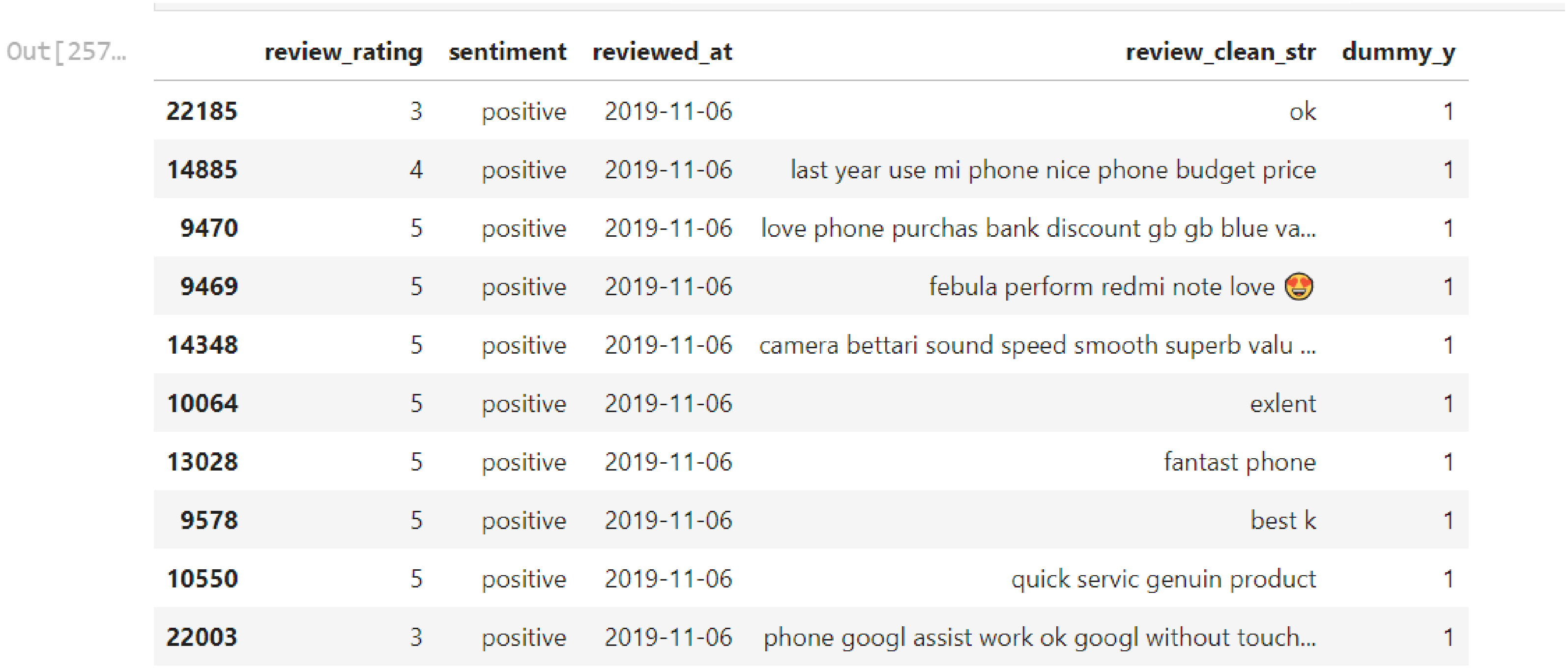
```
print(metrics.accuracy_score(y_test,predictions))
```

```
0.9856271777003485
```

# Implementation of Random Forest model:

- Random forest algorithm(SML) creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.
- Ensemble method which is better than a single decision tree as-
  - it reduces the over-fitting by averaging the result.

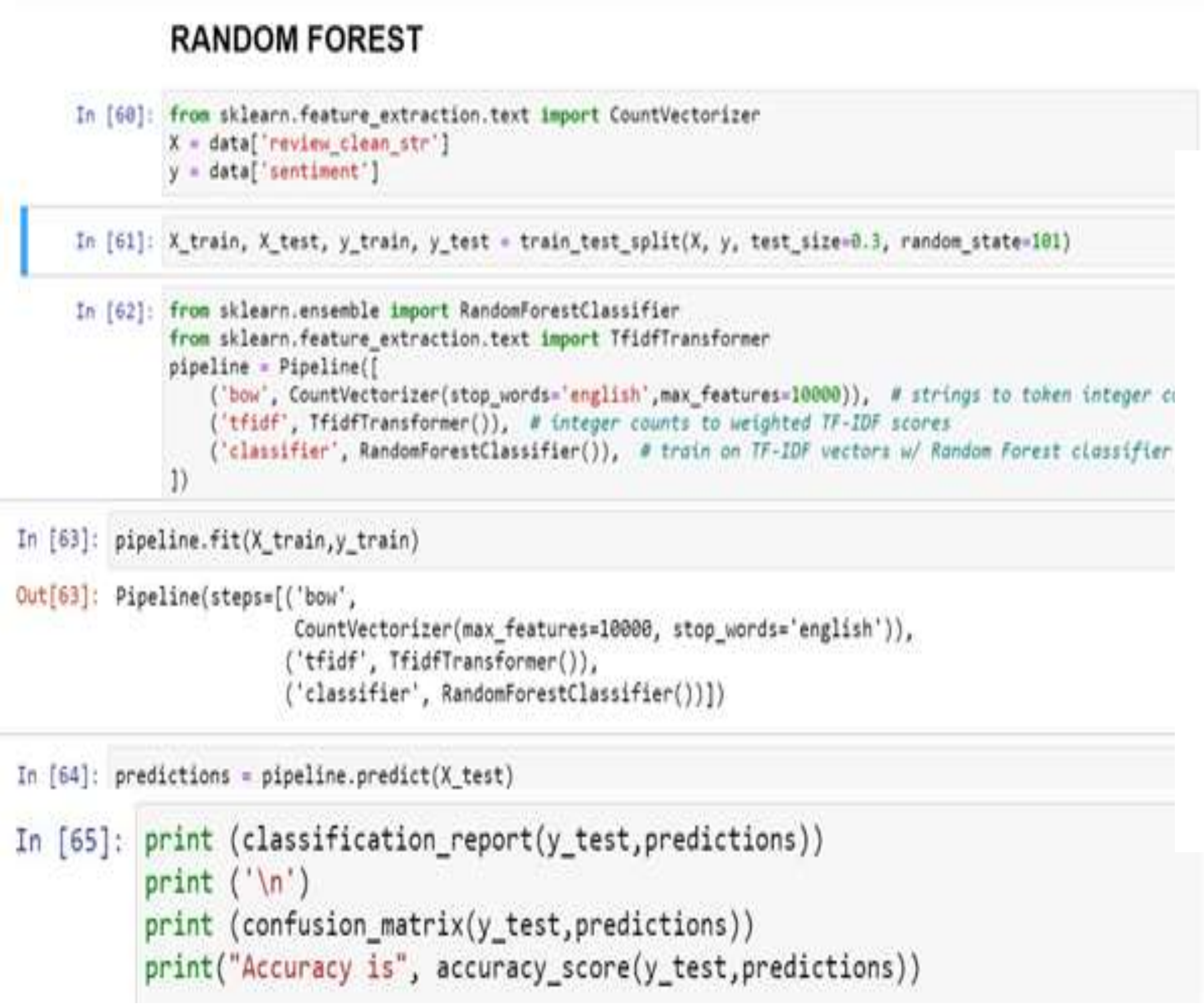
## Using subset of data



Out[257...

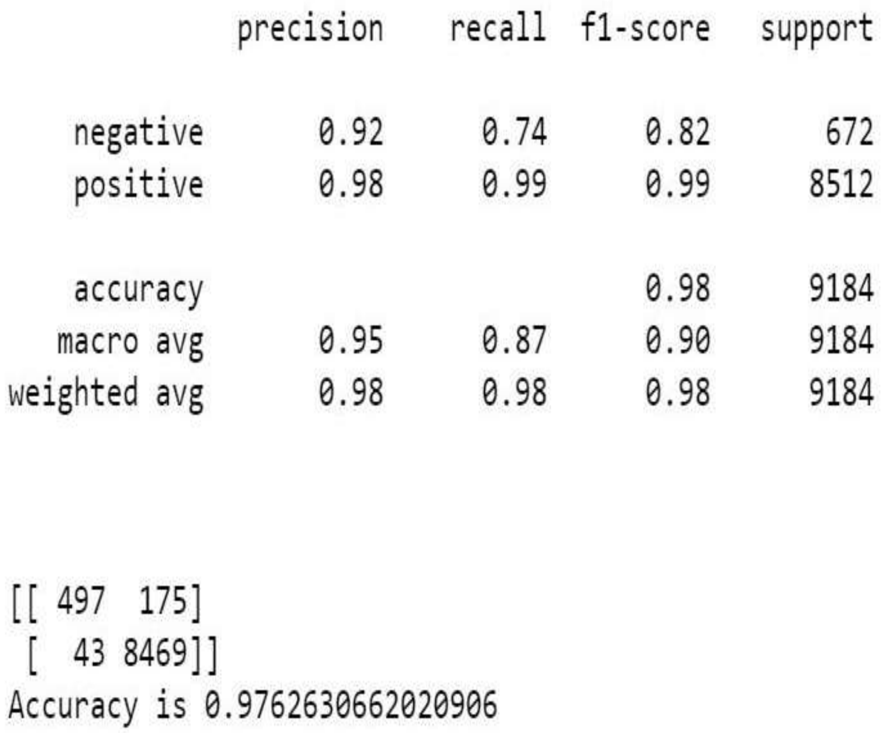
| | review_rating | sentiment | reviewed_at | review_clean_str | dummy_y |
|---|---|---|---|---|---|
| **22185** | 3 | positive | 2019-11-06 | ok | 1 |
| **14885** | 4 | positive | 2019-11-06 | last year use mi phone nice phone budget price | 1 |
| **9470** | 5 | positive | 2019-11-06 | love phone purchas bank discount gb gb blue va... | 1 |
| **9469** | 5 | positive | 2019-11-06 | febula perform redmi note love 😍 | 1 |
| **14348** | 5 | positive | 2019-11-06 | camera bettari sound speed smooth superb valu ... | 1 |
| **10064** | 5 | positive | 2019-11-06 | exlent | 1 |
| **13028** | 5 | positive | 2019-11-06 | fantast phone | 1 |
| **9578** | 5 | positive | 2019-11-06 | best k | 1 |
| **10550** | 5 | positive | 2019-11-06 | quick servic genuin product | 1 |
| **22003** | 3 | positive | 2019-11-06 | phone googl assist work ok googl without touch... | 1 |

# Results:



### RANDOM FOREST

```
In [60]: from sklearn.feature_extraction.text import CountVectorizer
         X = data['review_clean_str']
         y = data['sentiment']

In [61]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

In [62]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.feature_extraction.text import TfidfTransformer
         pipeline = Pipeline([
             ('bow', CountVectorizer(stop_words='english',max_features=10000)),  # strings to token integer c
             ('tfidf', TfidfTransformer()),  # integer counts to weighted TF-IDF scores
             ('classifier', RandomForestClassifier()),  # train on TF-IDF vectors w/ Random Forest classifier
         ])

In [63]: pipeline.fit(X_train,y_train)

Out[63]: Pipeline(steps=[('bow',
                          CountVectorizer(max_features=10000, stop_words='english')),
                         ('tfidf', TfidfTransformer()),
                         ('classifier', RandomForestClassifier())])

In [64]: predictions = pipeline.predict(X_test)

In [65]: print (classification_report(y_test,predictions))
         print ('\n')
         print (confusion_matrix(y_test,predictions))
         print("Accuracy is", accuracy_score(y_test,predictions))
```

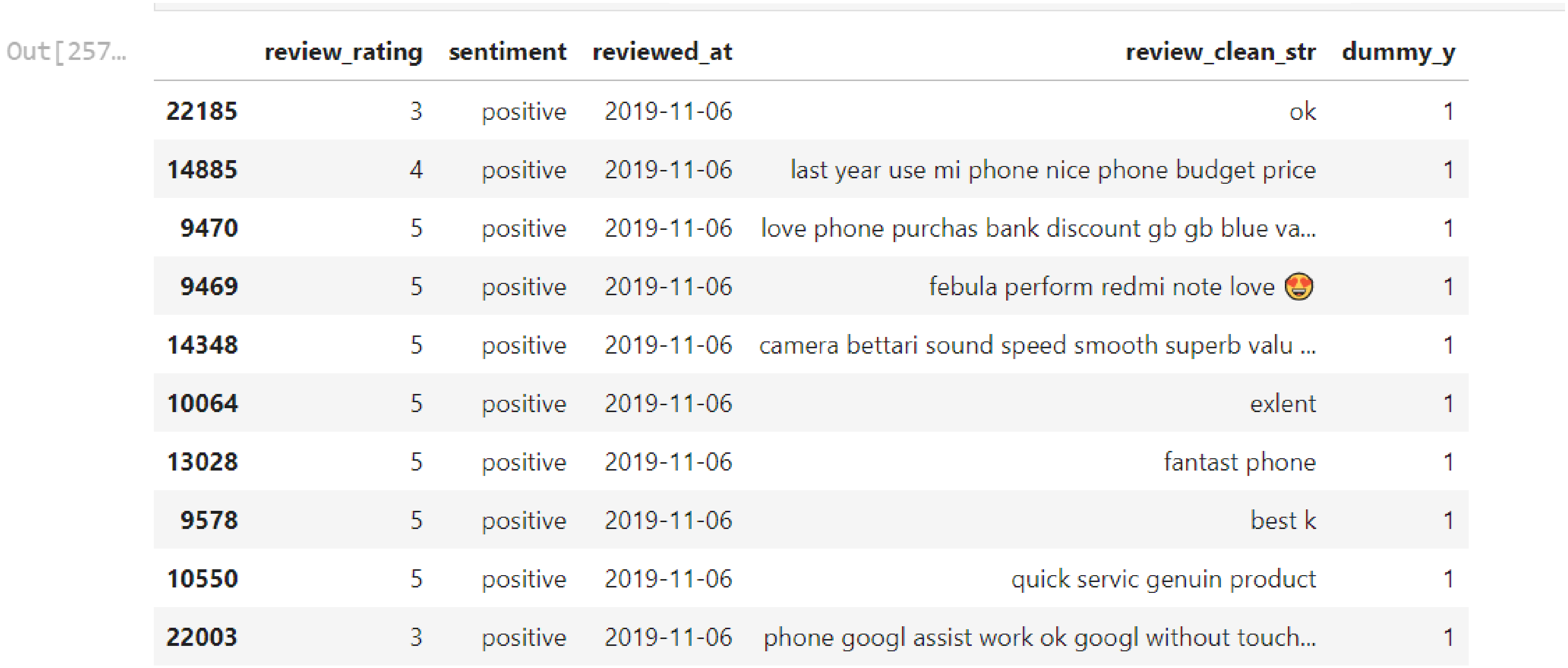


```
              precision    recall  f1-score   support

    negative       0.92      0.74      0.82       672
    positive       0.98      0.99      0.99      8512

    accuracy                           0.98      9184
   macro avg       0.95      0.87      0.90      9184
weighted avg       0.98      0.98      0.98      9184



[[ 497  175]
 [  43 8469]]
Accuracy is 0.9762630662020906
```

# Implementation of Logistic Regression model:

- It's a classification algorithm(SML), that is used where the response variable is categorical. The idea of **Logistic Regression** is to find a relationship between features and probability of particular outcome.
- To classify the observations using different types of data and can easily determine the most effective variables used for the classification.

## Using subset of data

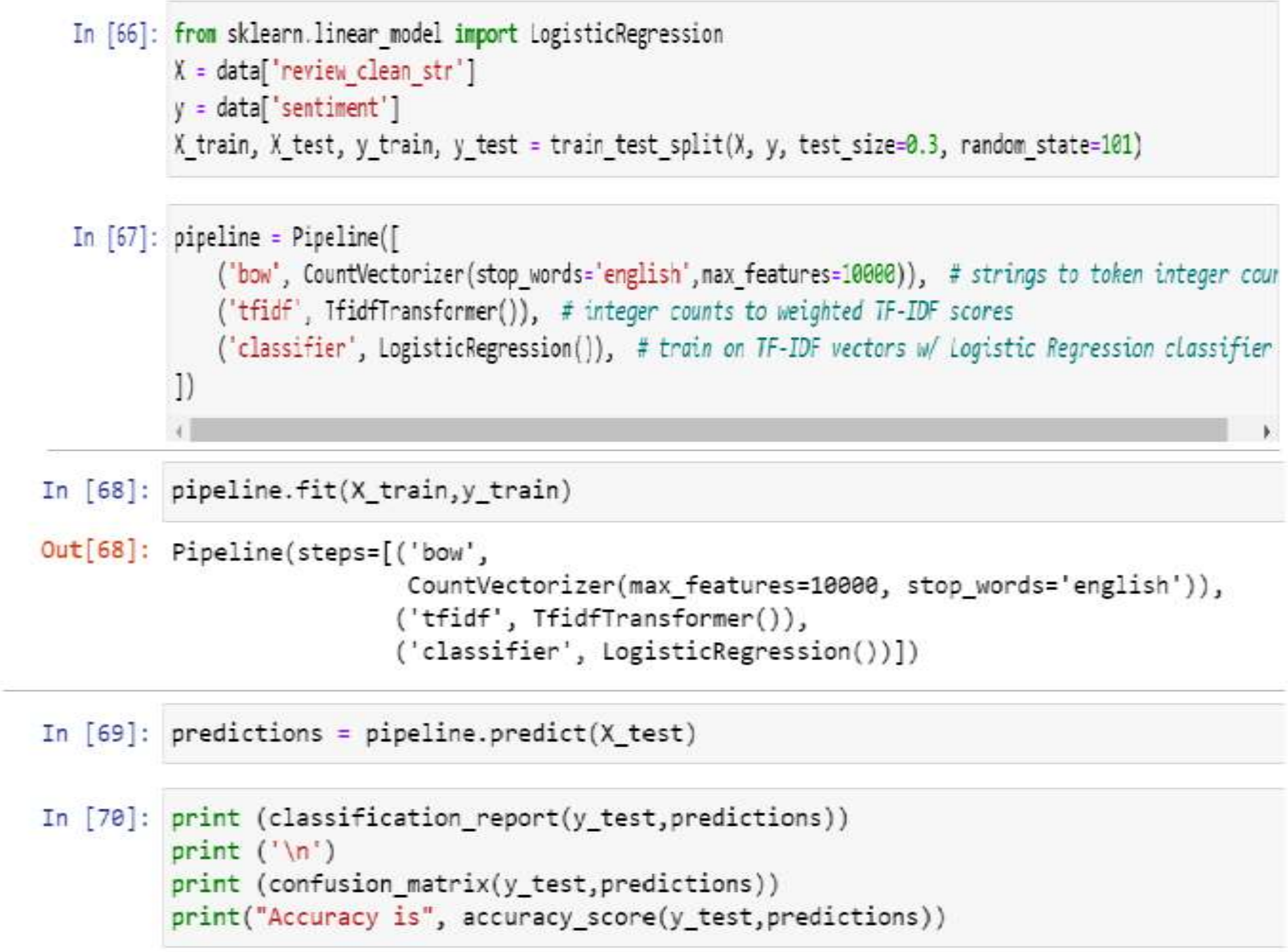


Out[257...

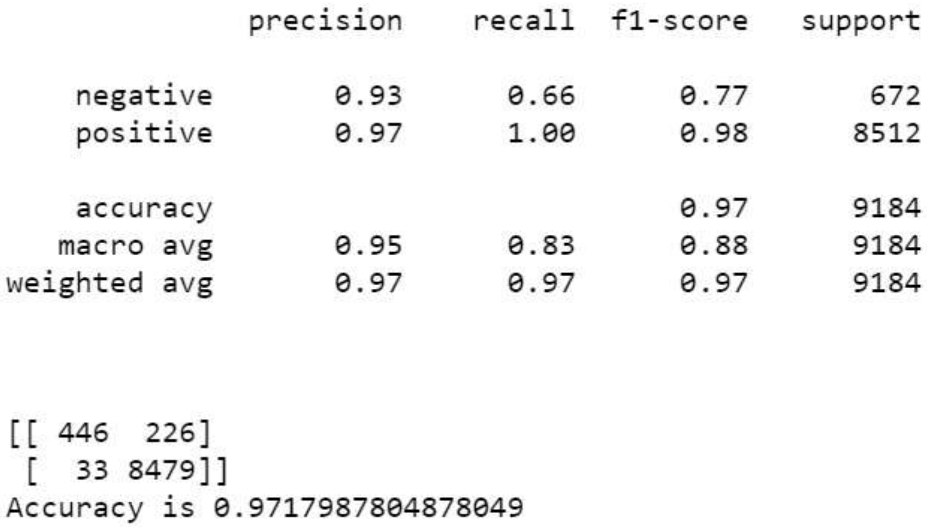
| | review_rating | sentiment | reviewed_at | review_clean_str | dummy_y |
|---|---|---|---|---|---|
| **22185** | 3 | positive | 2019-11-06 | ok | 1 |
| **14885** | 4 | positive | 2019-11-06 | last year use mi phone nice phone budget price | 1 |
| **9470** | 5 | positive | 2019-11-06 | love phone purchas bank discount gb gb blue va... | 1 |
| **9469** | 5 | positive | 2019-11-06 | febula perform redmi note love 😍 | 1 |
| **14348** | 5 | positive | 2019-11-06 | camera bettari sound speed smooth superb valu ... | 1 |
| **10064** | 5 | positive | 2019-11-06 | exlent | 1 |
| **13028** | 5 | positive | 2019-11-06 | fantast phone | 1 |
| **9578** | 5 | positive | 2019-11-06 | best k | 1 |
| **10550** | 5 | positive | 2019-11-06 | quick servic genuin product | 1 |
| **22003** | 3 | positive | 2019-11-06 | phone googl assist work ok googl without touch... | 1 |

# Results:

## LOGISTIC REGRESSION



```
In [66]: from sklearn.linear_model import LogisticRegression
         X = data['review_clean_str']
         y = data['sentiment']
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

In [67]: pipeline = Pipeline([
             ('bow', CountVectorizer(stop_words='english',max_features=10000)),  # strings to token integer cour
             ('tfidf', TfidfTransformer()),  # integer counts to weighted TF-IDF scores
             ('classifier', LogisticRegression()),  # train on TF-IDF vectors w/ Logistic Regression classifier
         ])

In [68]: pipeline.fit(X_train,y_train)

Out[68]: Pipeline(steps=[('bow',
                          CountVectorizer(max_features=10000, stop_words='english')),
                         ('tfidf', TfidfTransformer()),
                         ('classifier', LogisticRegression())])

In [69]: predictions = pipeline.predict(X_test)

In [70]: print (classification_report(y_test,predictions))
         print ('\n')
         print (confusion_matrix(y_test,predictions))
         print("Accuracy is", accuracy_score(y_test,predictions))
```

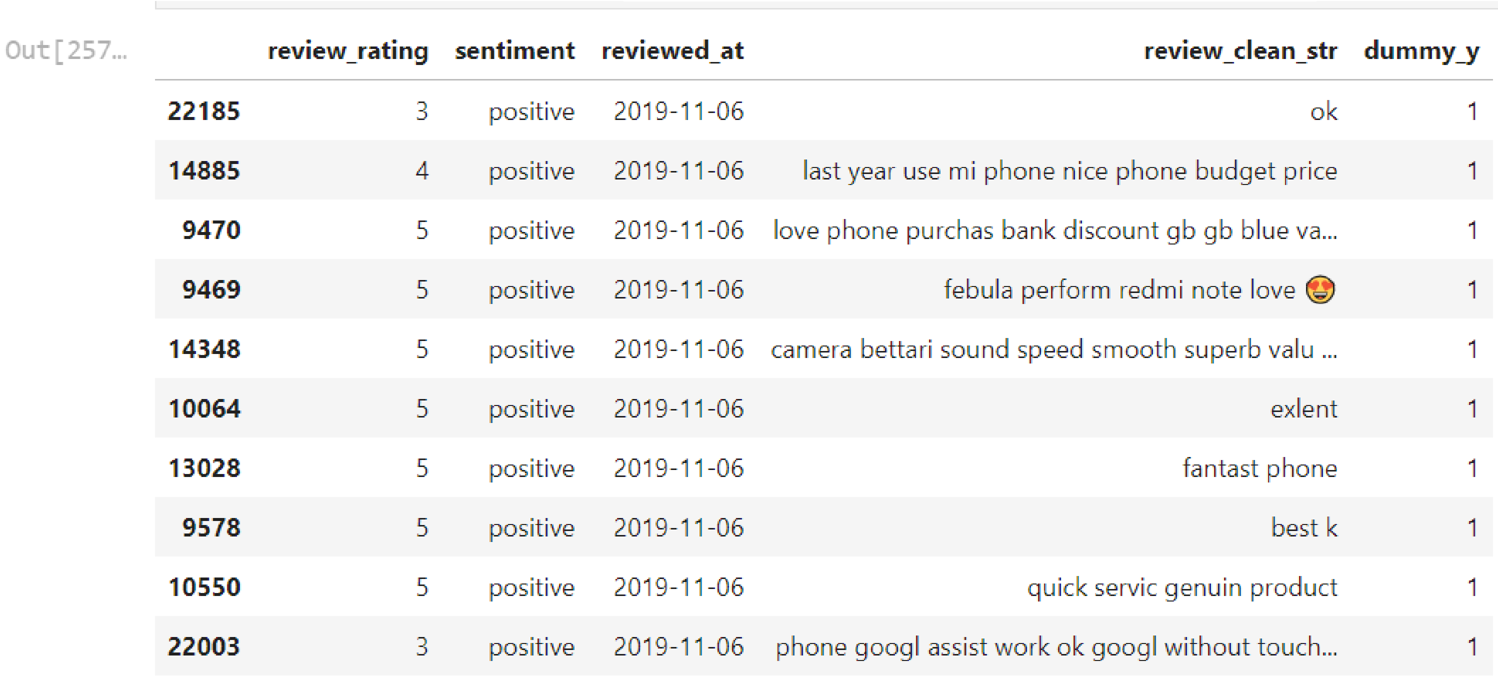


```
              precision    recall  f1-score   support

    negative       0.93      0.66      0.77       672
    positive       0.97      1.00      0.98      8512

    accuracy                           0.97      9184
   macro avg       0.95      0.83      0.88      9184
weighted avg       0.97      0.97      0.97      9184



[[ 446  226]
 [  33 8479]]
Accuracy is 0.9717987804878049
```

# Implementation of Naïve Bayes' model:

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.

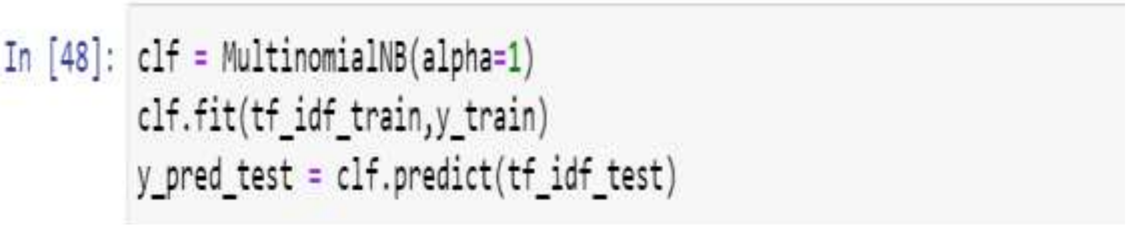
## Using subset of data



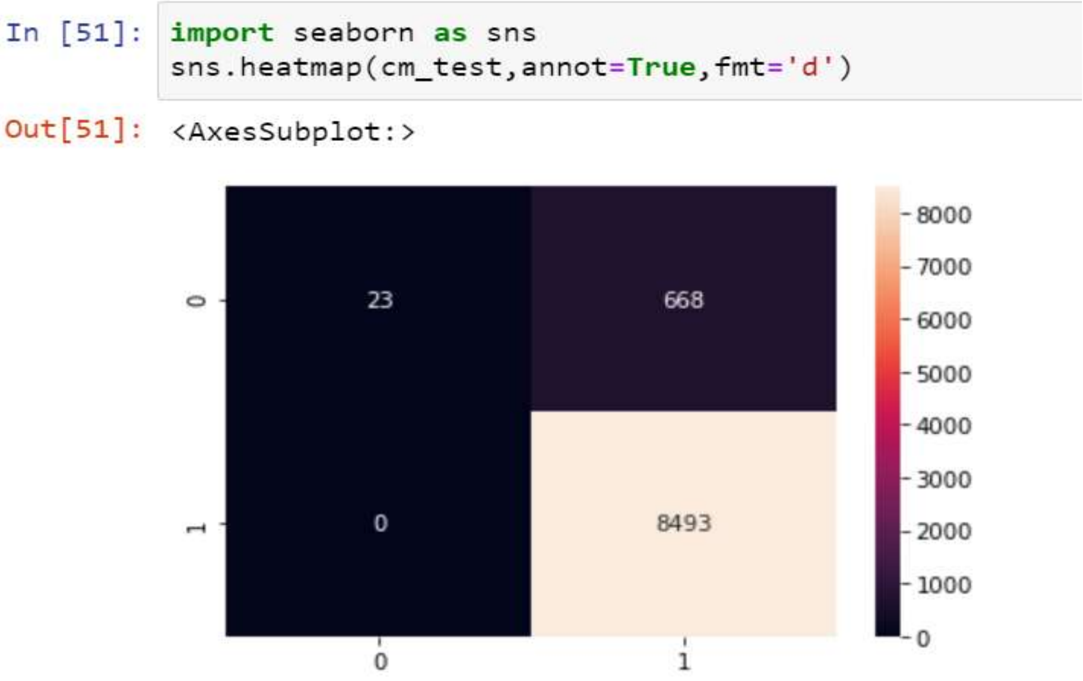
Out[257...

| | review_rating | sentiment | reviewed_at | review_clean_str | dummy_y |
|---|---|---|---|---|---|
| **22185** | 3 | positive | 2019-11-06 | ok | 1 |
| **14885** | 4 | positive | 2019-11-06 | last year use mi phone nice phone budget price | 1 |
| **9470** | 5 | positive | 2019-11-06 | love phone purchas bank discount gb gb blue va... | 1 |
| **9469** | 5 | positive | 2019-11-06 | febula perform redmi note love 😍 | 1 |
| **14348** | 5 | positive | 2019-11-06 | camera bettari sound speed smooth superb valu ... | 1 |
| **10064** | 5 | positive | 2019-11-06 | exlent | 1 |
| **13028** | 5 | positive | 2019-11-06 | fantast phone | 1 |
| **9578** | 5 | positive | 2019-11-06 | best k | 1 |
| **10550** | 5 | positive | 2019-11-06 | quick servic genuin product | 1 |
| **22003** | 3 | positive | 2019-11-06 | phone googl assist work ok googl without touch... | 1 |

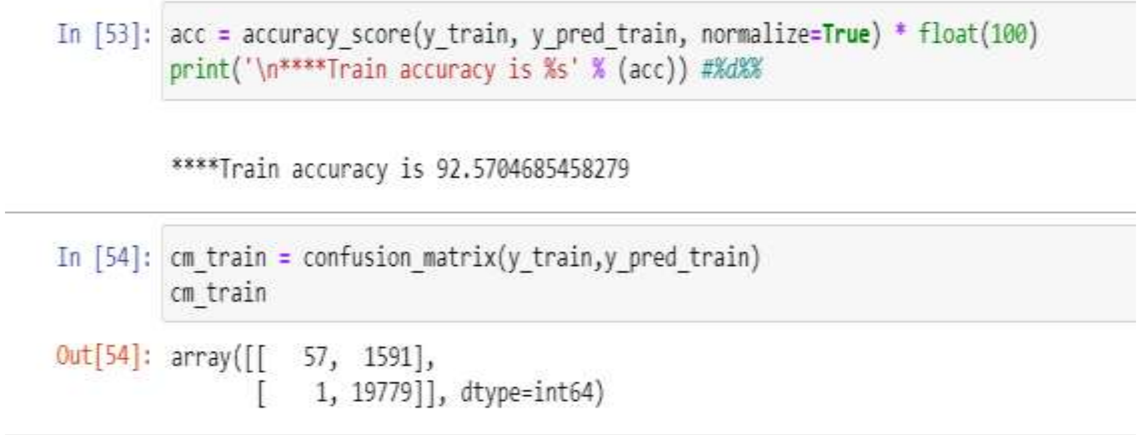
# Results:



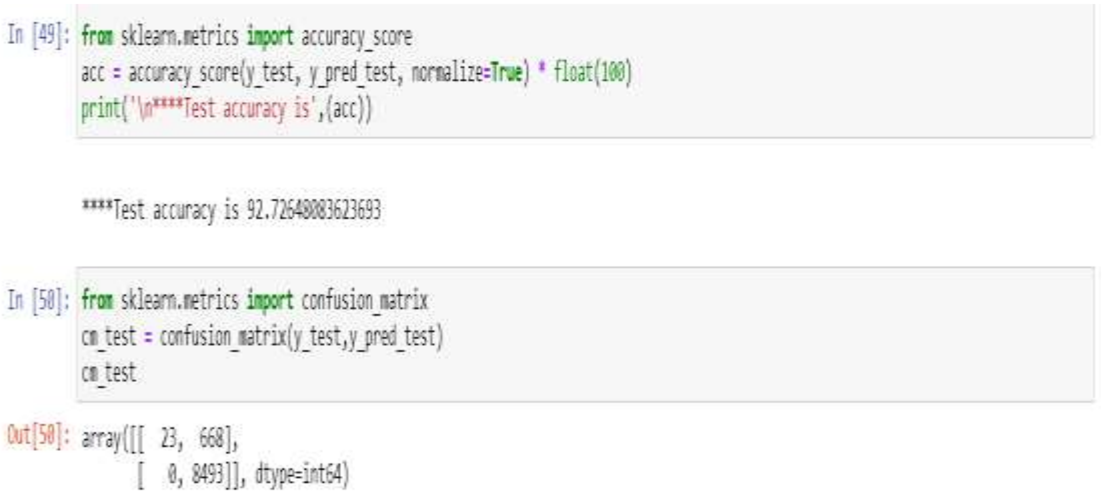
```
In [48]: clf = MultinomialNB(alpha=1)
         clf.fit(tf_idf_train,y_train)
         y_pred_test = clf.predict(tf_idf_test)
```



```
In [51]: import seaborn as sns
         sns.heatmap(cm_test,annot=True,fmt='d')

Out[51]: <AxesSubplot:>
```



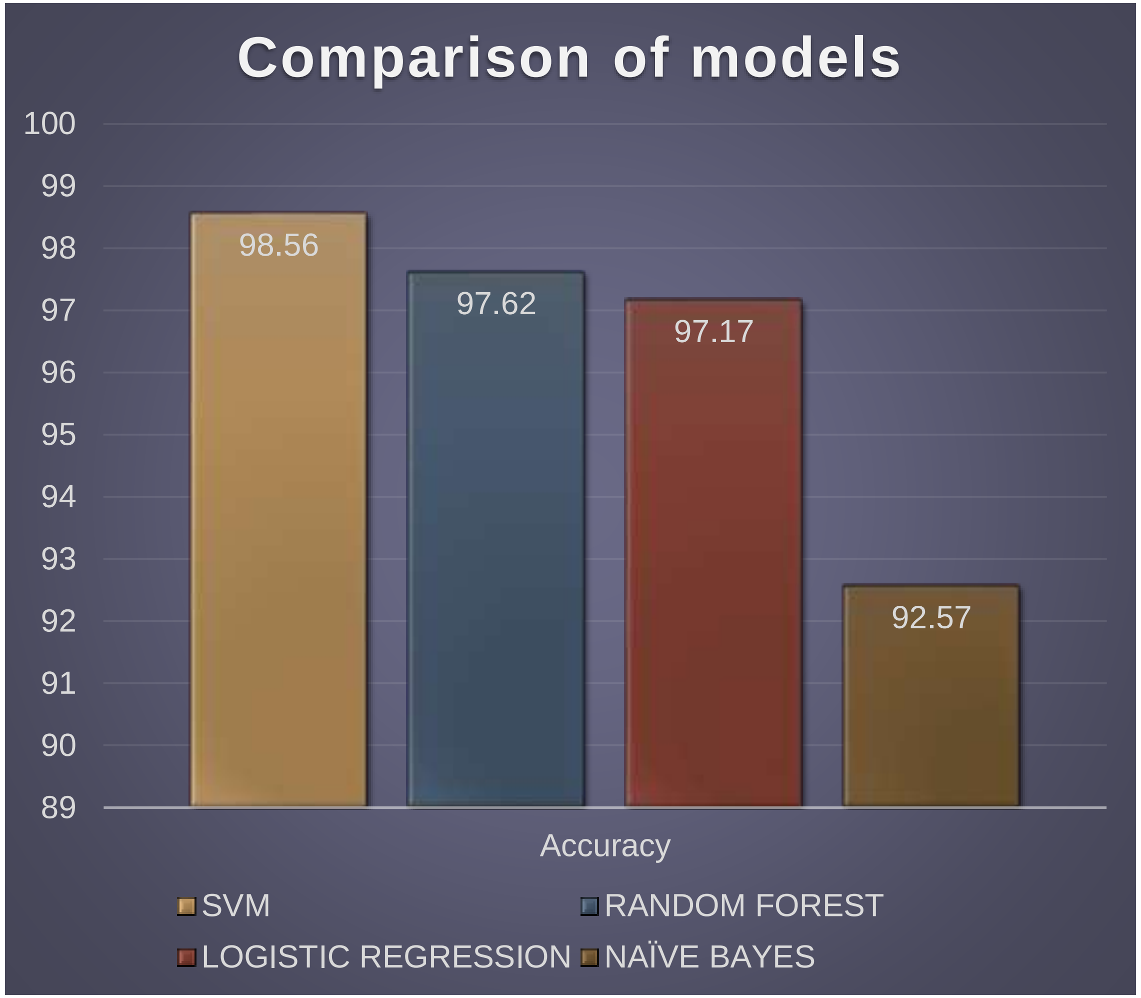
```
In [53]: acc = accuracy_score(y_train, y_pred_train, normalize=True) * float(100)
         print('\n****Train accuracy is %s' % (acc)) #%d%%

         ****Train accuracy is 92.5704685458279

In [54]: cm_train = confusion_matrix(y_train,y_pred_train)
         cm_train

Out[54]: array([[   57,  1591],
                [    1, 19779]], dtype=int64)
```



```
In [49]: from sklearn.metrics import accuracy_score
         acc = accuracy_score(y_test, y_pred_test, normalize=True) * float(100)
         print('\n****Test accuracy is',(acc))

         ****Test accuracy is 92.72648083623693

In [50]: from sklearn.metrics import confusion_matrix
         cm_test = confusion_matrix(y_test,y_pred_test)
         cm_test

Out[50]: array([[  23,  668],
                [   0, 8493]], dtype=int64)
```

# Comparison of Implemented models:

| S.NO | SVM | RANDOM FOREST TREE | LR | NB |
|---|---|---|---|---|
| Accuracy | 98.56% | 97.62% | 97.17% | 92.57% |



# Conclusion:

- From the above implementation we are able to identify that SVM provides the highest accuracy among the given text classifiers.
- And, SVM by far
  - Works really well with a clear margin of separation
  - Effective in high dimensional spaces.

- Further Research Area:
  - Deep Learning models have made advances in NLP in terms of Sentiment Analysis.
  - Such as LSTM,CNN models can be deployed in the future to predict the Sentiment analysis.

# References

- https://ieeexplore.ieee.org/document/8321910
- https://ieeexplore.ieee.org/document/8603585
- https://ieeexplore.ieee.org/document/8850982
- https://ieeexplore.ieee.org/document/8389689
- https://doi.org/10.1109/ICNSC.2019.8743331
- https://doi.org/10.1109/CSITSS.2017.8447660
- https://doi.org/10.1145/3340997.3341012
- https://doi.org/10.1109/IC3A48958.2020.233300
- https://doi.org/10.1109/ICCIC.2016.7919584
- https://doi.org/10.1016/j.procs.2018.01.150
- Other references:
- Business reviews classification using sentiment analysis-https://doi.org/10.1109/SYNASC.2015.46
- Sentiment Analysis in TripAdvisor-https://doi.org/10.1109/MIS.2017.3121555
- Using Objective Words in SentiWordNet to Improve Word-ofMouth Sentiment Classification-https://doi.org/10.1109/MIS.2013.1

# Thank You