Configuration Manual

MSc Research Project
Programme Name

Shuangyin Xie
Student ID: X18126634

School of Computing
National College of Ireland

Supervisor: Bahman Honari

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shuangyin Xie
Student ID:	X18126634
Programme:	Msc in Data Analytics
Year:	2019
Module:	Research project
Supervisor:	Bahman Honari
Submission Due Date:	12/12/2019
Project Title:	Sentiment Analysis using machine learning algorithms: online women clothing reviews
Word Count:	XXX
Page Count:	10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Sentiment Analysis using machine learning algorithms: online women clothing reviews

Shuangyin Xie
X18126634

1 Overview

In the research,Support Vector Machine, Logistic Regression, Random Forest and Naive Bayes methods are selected to solve the sentiment analysis about online clothing reviews. All the methods are conducted by python language.

The manual will be followed in terms of Overview, System introduction, Installation, Implement and Results.

2 System Introduction

2.1 Hardware

The processor specifications are 1.8GHz dual-core Intel Core i5, Turbo Boost up to 2.9GHz, with 3MB shared L3 cache, 256GB PCIe-based SSD,8GB of 1600MHz LPDDR3 onboard memory. And System is MacOS.

2.2 Software

The application requires Anaconda Navigator, jupyter notebook 6.0. Besides, Microsoft Excel from Microsoft Office is to make graphics.

3 Installation

3.1 Installing software

Before starting the solutions, some software need to be installed. Figure 1 shows how to download the Anaconda.

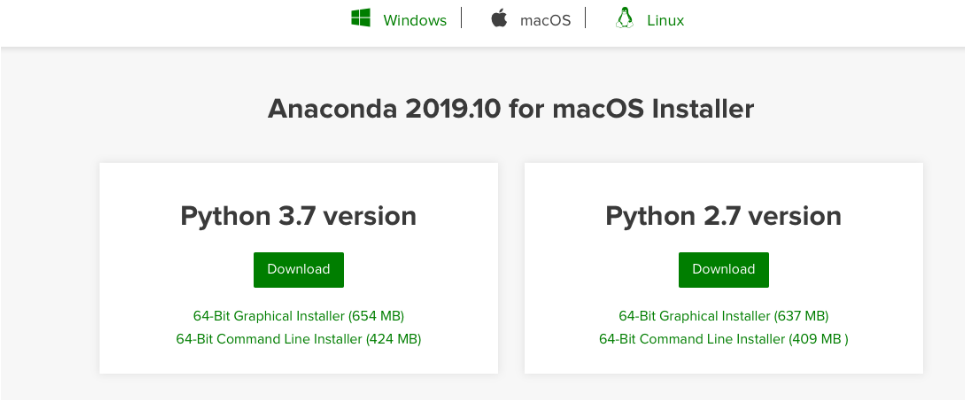
Figure 2 indicates the Matplotlib need to be installed.

Figure 3 presents the process of installing matplotlib in terminal.

3.2 Install packages

Besides Anaconda navigator, some packages also need to be installed.

Figure 5 and 6 shows the stopwords and nltk.



Anaconda 2019.10 for macOS Installer

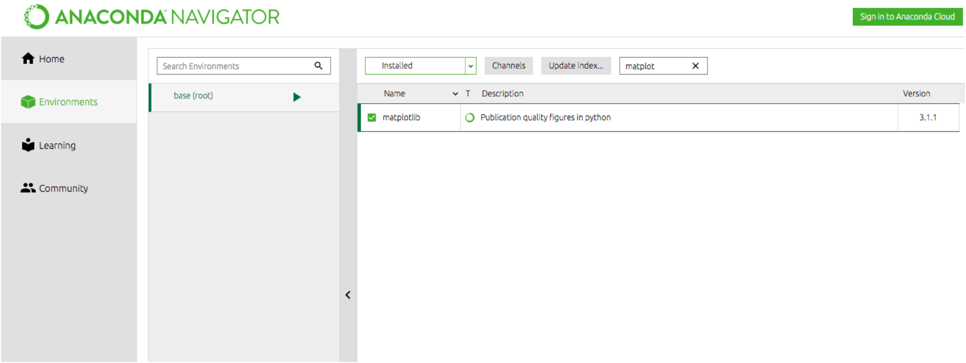
Python 3.7 version

64-Bit Graphical Installer (654 MB)
64-Bit Command Line Installer (424 MB)

Python 2.7 version

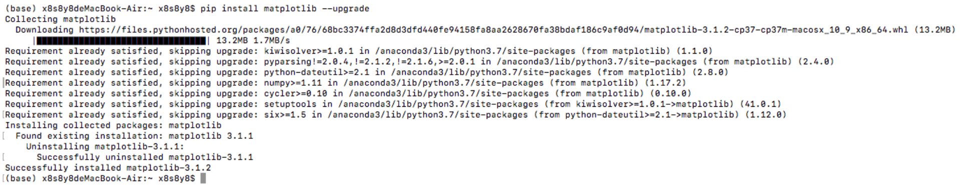
64-Bit Graphical Installer (637 MB)
64-Bit Command Line Installer (409 MB)

Figure 1: Download Anaconda



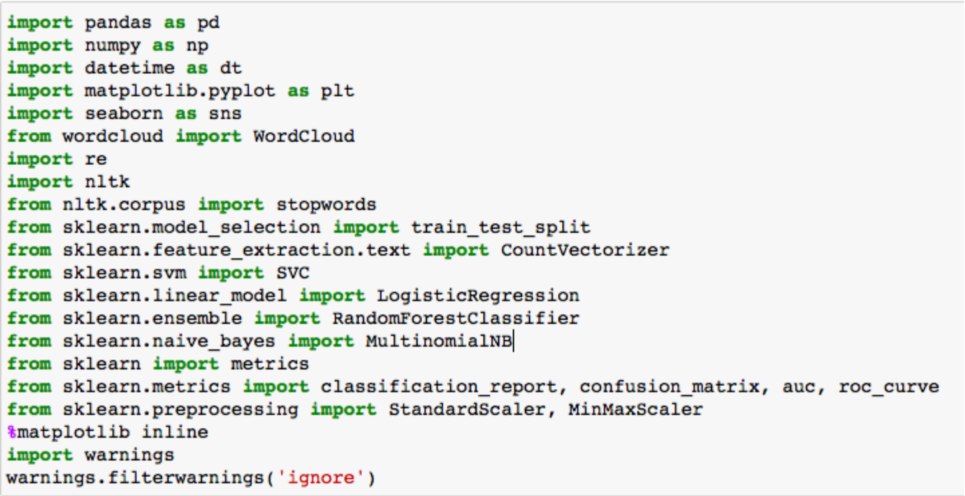
Name	T	Description	Version
☑ matplotlib		○ Publication quality figures in python	3.1.1

Figure 2: Install Matplotlib



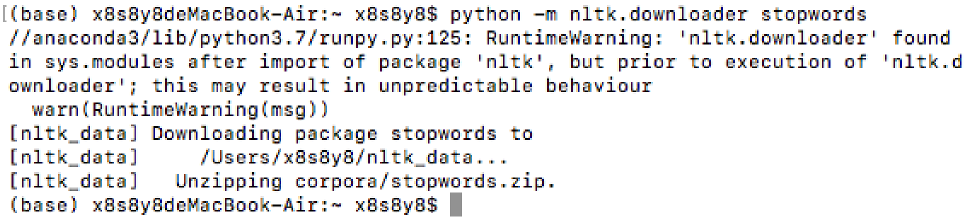
```
(base) x8s8y8deMacBook-Air:~ x8s8y8$ pip install matplotlib --upgrade
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/a0/76/68bc3374ffa2d8d3dfd440fe94158fa8aa2628670fa38bdaf186c9af0d94/matplotlib-3.1.2-cp37-cp37m-macosx_10_9_x86_64.whl (13.2MB)
     |████████████████████████████████| 13.2MB 1.7MB/s
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /anaconda3/lib/python3.7/site-packages (from matplotlib) (1.1.0)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /anaconda3/lib/python3.7/site-packages (from matplotlib) (2.4.0)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in /anaconda3/lib/python3.7/site-packages (from matplotlib) (2.8.0)
Requirement already satisfied, skipping upgrade: numpy>=1.11 in /anaconda3/lib/python3.7/site-packages (from matplotlib) (1.17.2)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /anaconda3/lib/python3.7/site-packages (from matplotlib) (0.10.0)
Requirement already satisfied, skipping upgrade: setuptools in /anaconda3/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplotlib) (41.0.1)
Requirement already satisfied, skipping upgrade: six>=1.5 in /anaconda3/lib/python3.7/site-packages (from python-dateutil>=2.1->matplotlib) (1.12.0)
Installing collected packages: matplotlib
  Found existing installation: matplotlib 3.1.1
    Uninstalling matplotlib-3.1.1:
      Successfully uninstalled matplotlib-3.1.1
Successfully installed matplotlib-3.1.2
(base) x8s8y8deMacBook-Air:~ x8s8y8$
```

Figure 3: Install Matplotlib



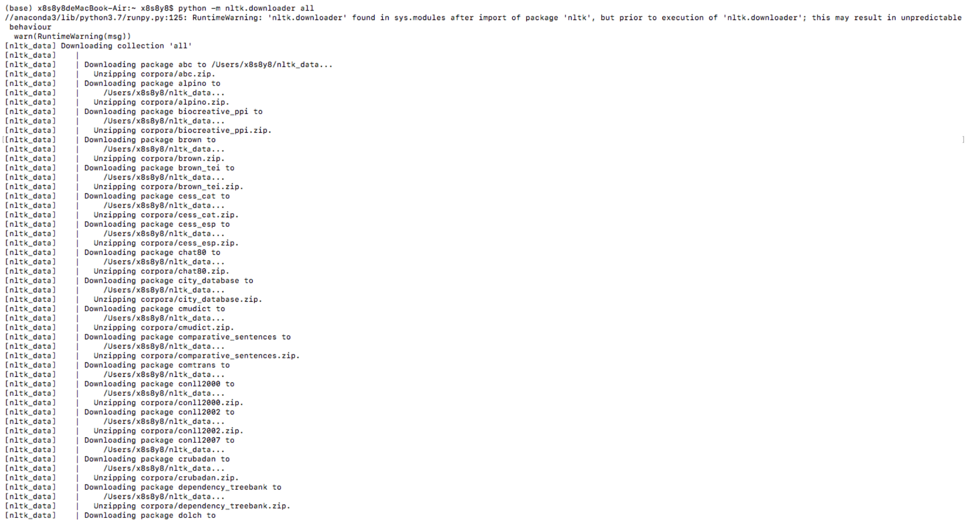
```python
import pandas as pd
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import re
import nltk
from nltk.corpus import stopwords
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix, auc, roc_curve
from sklearn.preprocessing import StandardScaler, MinMaxScaler
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Figure 4: Install packages



```
(base) x8s8y8deMacBook-Air:~ x8s8y8$ python -m nltk.downloader stopwords
//anaconda3/lib/python3.7/runpy.py:125: RuntimeWarning: 'nltk.downloader' found
in sys.modules after import of package 'nltk', but prior to execution of 'nltk.d
ownloader'; this may result in unpredictable behaviour
  warn(RuntimeWarning(msg))
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/x8s8y8/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
(base) x8s8y8deMacBook-Air:~ x8s8y8$ 
```

Figure 5: Install stopwords



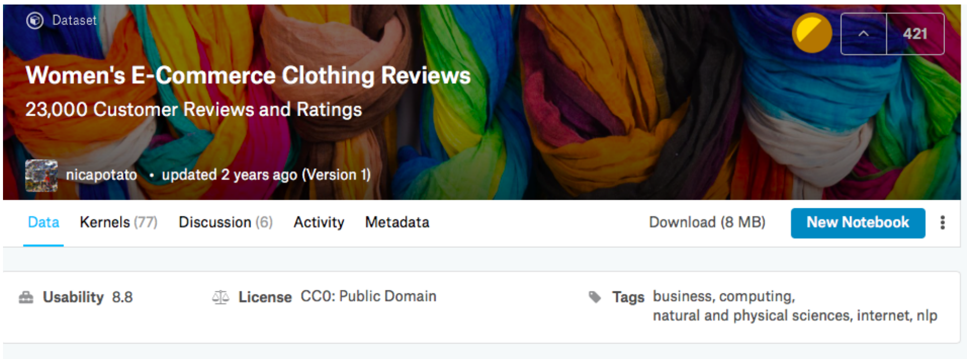
```
(base) x8s8y8deMacBook-Air:~ x8s8y8$ python -m nltk.downloader all
//anaconda3/lib/python3.7/runpy.py:125: RuntimeWarning: 'nltk.downloader' found in sys.modules after import of package 'nltk', but prior to execution of 'nltk.downloader'; this may result in unpredictable
 behaviour
  warn(RuntimeWarning(msg))
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/biocreative_ppi.zip.
[nltk_data]    | Downloading package brown to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/brown.zip.
[nltk_data]    | Downloading package brown_tei to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/brown_tei.zip.
[nltk_data]    | Downloading package cess_cat to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_cat.zip.
[nltk_data]    | Downloading package cess_esp to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_esp.zip.
[nltk_data]    | Downloading package chat80 to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/chat80.zip.
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/city_database.zip.
[nltk_data]    | Downloading package cmudict to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/cmudict.zip.
[nltk_data]    | Downloading package comparative_sentences to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/comparative_sentences.zip.
[nltk_data]    | Downloading package comtrans to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    | Downloading package conll2000 to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2000.zip.
[nltk_data]    | Downloading package conll2002 to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2002.zip.
[nltk_data]    | Downloading package conll2007 to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    | Downloading package crubadan to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/crubadan.zip.
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /Users/x8s8y8/nltk_data...
[nltk_data]    |   Unzipping corpora/dependency_treebank.zip.
[nltk_data]    | Downloading package dolch to
```

Figure 6: Install nltk

4 Implement and Results

4.1 Data Set

The data set is downloaded from Kaggle.[1]The data set is about online women clothing reviews(Agarap and Grafilon; 2018).



Dataset

Women's E-Commerce Clothing Reviews

23,000 Customer Reviews and Ratings

nicapotato • updated 2 years ago (Version 1)

Usability 8.8

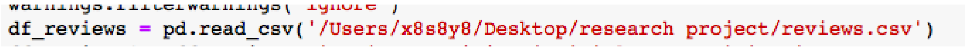
License CC0: Public Domain

Tags business, computing, natural and physical sciences, internet, nlp

Figure 7: Data set

4.2 Load Data Set

Open the Jupyter and Read the data set into the Jupyter.



```
warnings.filterwarnings('ignore')
df_reviews = pd.read_csv('/Users/x8s8y8/Desktop/research project/reviews.csv')
```

Figure 8: Read the dataset

4.3 Execute Pre-processing

After loading the dataset, pre-processing the data set in order to prepare for the model. Figure 9 shows the process of pre-processing.

4.4 Split data

Figure 10 shows the code of split data.

4.5 Model

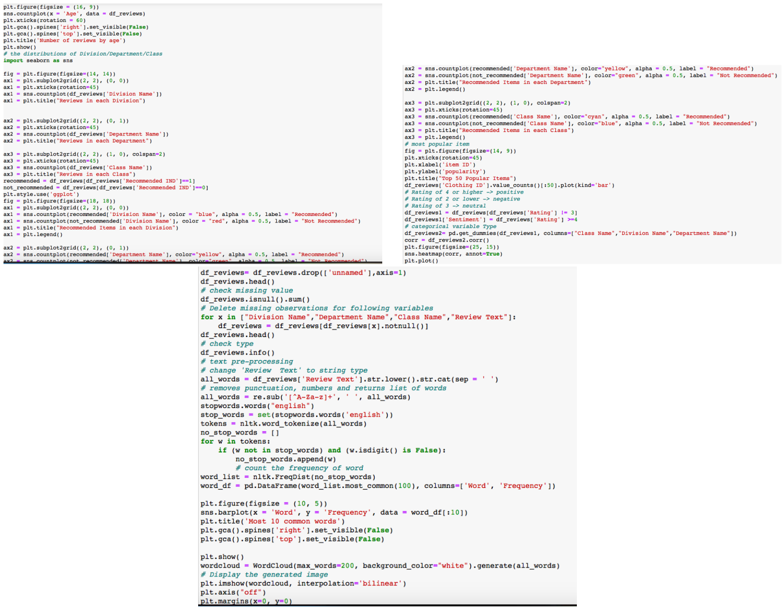
Figure 11 shows the code of models.

4.6 Results

The results are saved in review4.csv file.

Figure 12 shows the results of each model.

[1] https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews

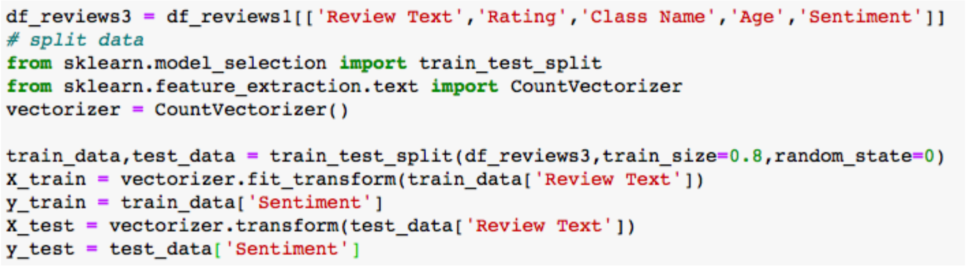


```
plt.figure(figsize = (16, 9))
sns.countplot(x = 'Age', data = df_reviews)
plt.xticks(rotation = 60)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
plt.title('Number of reviews by age')
plt.show()
# the distributions of Division/Department/Class
import seaborn as sns

fig = plt.figure(figsize=(14, 14))
ax1 = plt.subplot2grid((2, 2), (0, 0))
ax1 = plt.xticks(rotation=45)
ax1 = sns.countplot(df_reviews['Division Name'])
ax1 = plt.title("Reviews in each Division")


ax2 = plt.subplot2grid((2, 2), (0, 1))
ax2 = plt.xticks(rotation=45)
ax2 = sns.countplot(df_reviews['Department Name'])
ax2 = plt.title("Reviews in each Department")

ax3 = plt.subplot2grid((2, 2), (1, 0), colspan=2)
ax3 = plt.xticks(rotation=45)
ax3 = sns.countplot(df_reviews['Class Name'])
ax3 = plt.title("Reviews in each Class")
recommended = df_reviews[df_reviews['Recommended IND']==1]
not_recommended = df_reviews[df_reviews['Recommended IND']==0]
plt.style.use('ggplot')
fig = plt.figure(figsize=(18, 18))
ax1 = plt.subplot2grid((2, 2), (0, 0))
ax1 = sns.countplot(recommended['Division Name'], color = "blue", alpha = 0.5, label = "Recommended")
ax1 = sns.countplot(not_recommended['Division Name'], color = "red", alpha = 0.5, label = "Not Recommended")
ax1 = plt.title("Recommended Items in each Division")
ax1 = plt.legend()

ax2 = plt.subplot2grid((2, 2), (0, 1))
ax2 = sns.countplot(recommended['Department Name'], color="yellow", alpha = 0.5, label = "Recommended")
ax2 = sns.countplot(not_recommended['Department Name'], color="green", alpha = 0.5, label = "Not Recommended")
```

```
ax2 = sns.countplot(recommended['Department Name'], color="yellow", alpha = 0.5, label = "Recommended")
ax2 = sns.countplot(not_recommended['Department Name'], color="green", alpha = 0.5, label = "Not Recommended")
ax2 = plt.title("Recommended Items in each Department")
ax2 = plt.legend()

ax3 = plt.subplot2grid((2, 2), (1, 0), colspan=2)
ax3 = plt.xticks(rotation=45)
ax3 = sns.countplot(recommended['Class Name'], color="cyan", alpha = 0.5, label = "Recommended")
ax3 = sns.countplot(not_recommended['Class Name'], color="blue", alpha = 0.5, label = "Not Recommended")
ax3 = plt.title("Recommended Items in each Class")
ax3 = plt.legend()
# most popular item
fig = plt.figure(figsize=(14, 9))
plt.xticks(rotation=45)
plt.xlabel('item ID')
plt.ylabel('popularity')
plt.title("Top 50 Popular Items")
df_reviews['Clothing ID'].value_counts()[:50].plot(kind='bar')
# Rating of 4 or higher -> positive
# Rating of 2 or lower -> negative
# Rating of 3 -> neutral
df_reviews1 = df_reviews[df_reviews['Rating'] != 3]
df_reviews1['Sentiment'] = df_reviews['Rating'] >=4
# categorical variable Type
df_reviews2= pd.get_dummies(df_reviews1, columns=["Class Name","Division Name","Department Name"])
corr = df_reviews2.corr()
plt.figure(figsize=(25, 15))
sns.heatmap(corr, annot=True)
plt.plot()
```

```
df_reviews= df_reviews.drop(['unnamed'],axis=1)
df_reviews.head()
# check missing value
df_reviews.isnull().sum()
# Delete missing observations for following variables
for x in ["Division Name","Department Name","Class Name","Review Text"]:
    df_reviews = df_reviews[df_reviews[x].notnull()]
df_reviews.head()
# check type
df_reviews.info()
# text pre-processing
# change 'Review  Text' to string type
all_words = df_reviews['Review Text'].str.lower().str.cat(sep = ' ')
# removes punctuation, numbers and returns list of words
all_words = re.sub('[^A-Za-z]+', ' ', all_words)
stopwords.words("english")
stop_words = set(stopwords.words('english'))
tokens = nltk.word_tokenize(all_words)
no_stop_words = []
for w in tokens:
    if (w not in stop_words) and (w.isdigit() is False):
        no_stop_words.append(w)
        # count the frequency of word
word_list = nltk.FreqDist(no_stop_words)
word_df = pd.DataFrame(word_list.most_common(100), columns=['Word', 'Frequency'])

plt.figure(figsize = (10, 5))
sns.barplot(x = 'Word', y = 'Frequency', data = word_df[:10])
plt.title('Most 10 common words')
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

plt.show()
wordcloud = WordCloud(max_words=200, background_color="white").generate(all_words)
# Display the generated image
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0, y=0)
```

Figure 9: Data pre-processing



```
df_reviews3 = df_reviews1[['Review Text','Rating','Class Name','Age','Sentiment']]
# split data
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()

train_data,test_data = train_test_split(df_reviews3,train_size=0.8,random_state=0)
X_train = vectorizer.fit_transform(train_data['Review Text'])
y_train = train_data['Sentiment']
X_test = vectorizer.transform(test_data['Review Text'])
y_test = test_data['Sentiment']
```

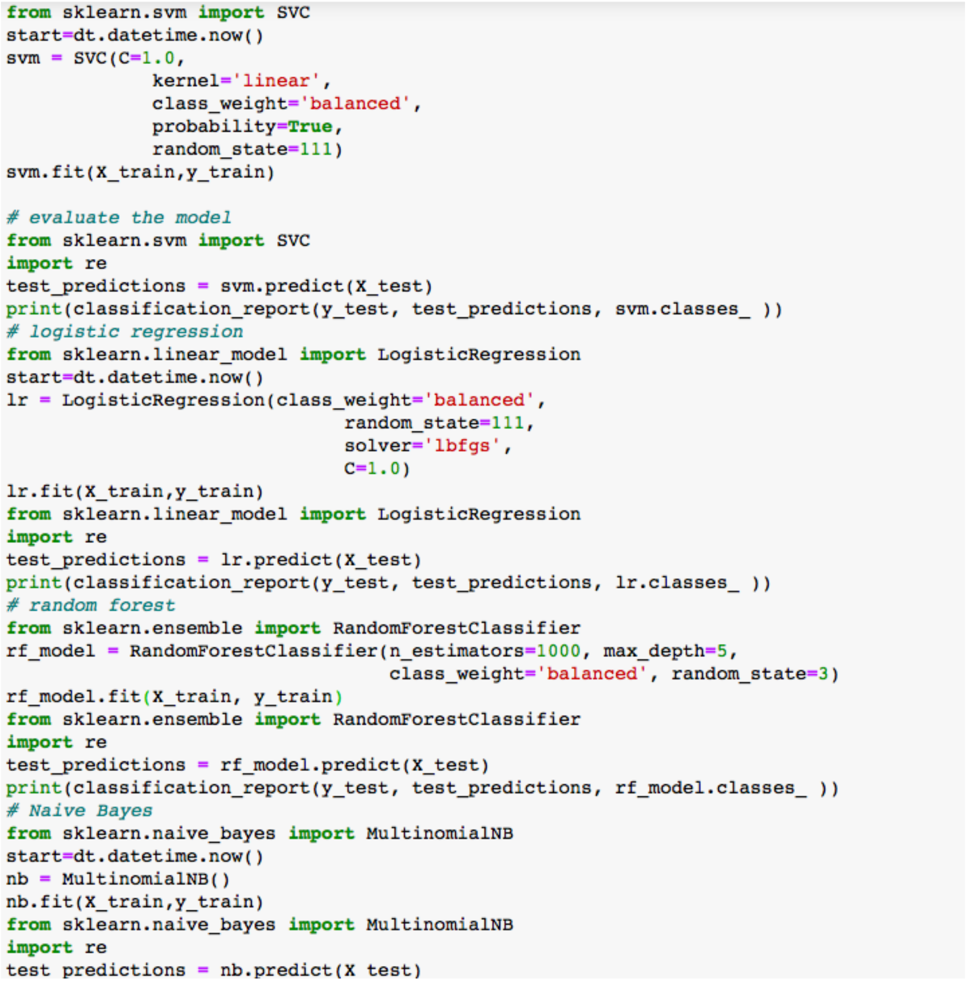
Figure 10: Split the data

4.7 Evaluation

Figure 13 shows the AUC and ROC.

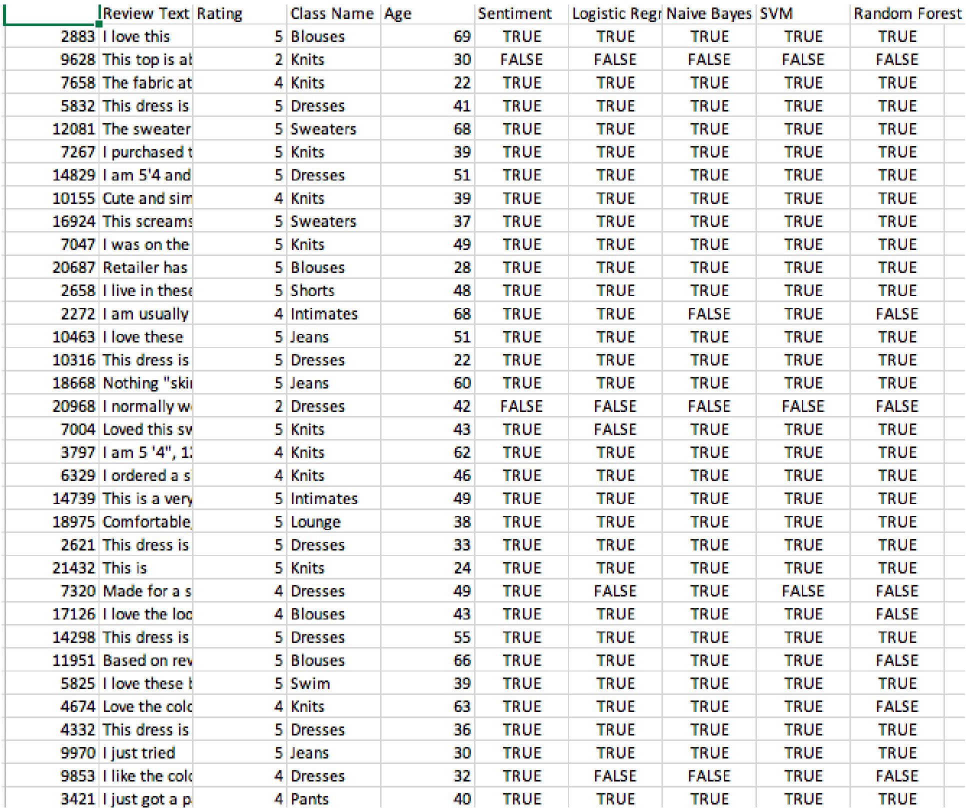
References

Agarap, A. F. M. and Grafilon, P. M. (2018). Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network, *https://www.researchgate.net/publication/323545316* .



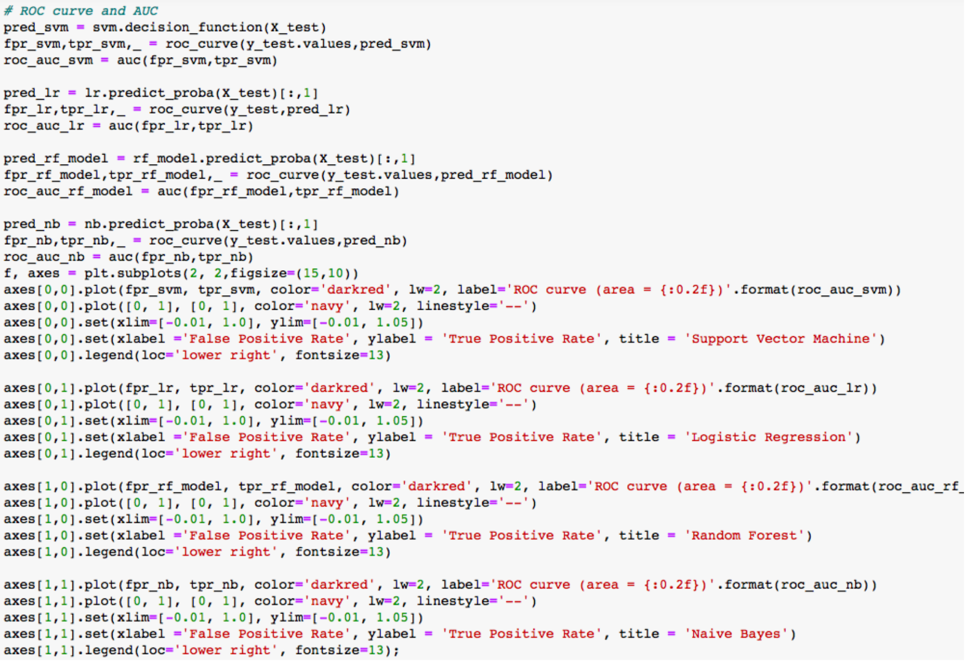
```
from sklearn.svm import SVC
start=dt.datetime.now()
svm = SVC(C=1.0,
             kernel='linear',
             class_weight='balanced',
             probability=True,
             random_state=111)
svm.fit(X_train,y_train)

# evaluate the model
from sklearn.svm import SVC
import re
test_predictions = svm.predict(X_test)
print(classification_report(y_test, test_predictions, svm.classes_ ))
# logistic regression
from sklearn.linear_model import LogisticRegression
start=dt.datetime.now()
lr = LogisticRegression(class_weight='balanced',
                              random_state=111,
                              solver='lbfgs',
                              C=1.0)
lr.fit(X_train,y_train)
from sklearn.linear_model import LogisticRegression
import re
test_predictions = lr.predict(X_test)
print(classification_report(y_test, test_predictions, lr.classes_ ))
# random forest
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(n_estimators=1000, max_depth=5,
                                  class_weight='balanced', random_state=3)
rf_model.fit(X_train, y_train)
from sklearn.ensemble import RandomForestClassifier
import re
test_predictions = rf_model.predict(X_test)
print(classification_report(y_test, test_predictions, rf_model.classes_ ))
# Naive Bayes
from sklearn.naive_bayes import MultinomialNB
start=dt.datetime.now()
nb = MultinomialNB()
nb.fit(X_train,y_train)
from sklearn.naive_bayes import MultinomialNB
import re
test_predictions = nb.predict(X_test)
```

Figure 11: Split the data



	Review Text	Rating	Class Name	Age	Sentiment	Logistic Regr	Naive Bayes	SVM	Random Forest
2883	I love this	5	Blouses	69	TRUE	TRUE	TRUE	TRUE	TRUE
9628	This top is ab	2	Knits	30	FALSE	FALSE	FALSE	FALSE	FALSE
7658	The fabric at	4	Knits	22	TRUE	TRUE	TRUE	TRUE	TRUE
5832	This dress is	5	Dresses	41	TRUE	TRUE	TRUE	TRUE	TRUE
12081	The sweater	5	Sweaters	68	TRUE	TRUE	TRUE	TRUE	TRUE
7267	I purchased t	5	Knits	39	TRUE	TRUE	TRUE	TRUE	TRUE
14829	I am 5'4 and	5	Dresses	51	TRUE	TRUE	TRUE	TRUE	TRUE
10155	Cute and sim	4	Knits	39	TRUE	TRUE	TRUE	TRUE	TRUE
16924	This screams	5	Sweaters	37	TRUE	TRUE	TRUE	TRUE	TRUE
7047	I was on the	5	Knits	49	TRUE	TRUE	TRUE	TRUE	TRUE
20687	Retailer has	5	Blouses	28	TRUE	TRUE	TRUE	TRUE	TRUE
2658	I live in these	5	Shorts	48	TRUE	TRUE	TRUE	TRUE	TRUE
2272	I am usually	4	Intimates	68	TRUE	TRUE	FALSE	TRUE	FALSE
10463	I love these	5	Jeans	51	TRUE	TRUE	TRUE	TRUE	TRUE
10316	This dress is	5	Dresses	22	TRUE	TRUE	TRUE	TRUE	TRUE
18668	Nothing "skir	5	Jeans	60	TRUE	TRUE	TRUE	TRUE	TRUE
20968	I normally w	2	Dresses	42	FALSE	FALSE	FALSE	FALSE	FALSE
7004	Loved this sw	5	Knits	43	TRUE	FALSE	TRUE	TRUE	TRUE
3797	I am 5 '4", 1	4	Knits	62	TRUE	TRUE	TRUE	TRUE	TRUE
6329	I ordered a s	4	Knits	46	TRUE	TRUE	TRUE	TRUE	TRUE
14739	This is a very	5	Intimates	49	TRUE	TRUE	TRUE	TRUE	TRUE
18975	Comfortable,	5	Lounge	38	TRUE	TRUE	TRUE	TRUE	TRUE
2621	This dress is	5	Dresses	33	TRUE	TRUE	TRUE	TRUE	TRUE
21432	This is	5	Knits	24	TRUE	TRUE	TRUE	TRUE	TRUE
7320	Made for a s	4	Dresses	49	TRUE	FALSE	TRUE	FALSE	FALSE
17126	I love the loo	4	Blouses	43	TRUE	TRUE	TRUE	TRUE	FALSE
14298	This dress is	5	Dresses	55	TRUE	TRUE	TRUE	TRUE	TRUE
11951	Based on rev	5	Blouses	66	TRUE	TRUE	TRUE	TRUE	FALSE
5825	I love these b	5	Swim	39	TRUE	TRUE	TRUE	TRUE	TRUE
4674	Love the colo	4	Knits	63	TRUE	TRUE	TRUE	TRUE	FALSE
4332	This dress is	5	Dresses	36	TRUE	TRUE	TRUE	TRUE	TRUE
9970	I just tried	5	Jeans	30	TRUE	TRUE	TRUE	TRUE	TRUE
9853	I like the colo	4	Dresses	32	TRUE	FALSE	FALSE	TRUE	FALSE
3421	I just got a p	4	Pants	40	TRUE	TRUE	TRUE	TRUE	TRUE

Figure 12: Results



```
# ROC curve and AUC
pred_svm = svm.decision_function(X_test)
fpr_svm,tpr_svm,_ = roc_curve(y_test.values,pred_svm)
roc_auc_svm = auc(fpr_svm,tpr_svm)

pred_lr = lr.predict_proba(X_test)[:,1]
fpr_lr,tpr_lr,_ = roc_curve(y_test,pred_lr)
roc_auc_lr = auc(fpr_lr,tpr_lr)

pred_rf_model = rf_model.predict_proba(X_test)[:,1]
fpr_rf_model,tpr_rf_model,_ = roc_curve(y_test.values,pred_rf_model)
roc_auc_rf_model = auc(fpr_rf_model,tpr_rf_model)

pred_nb = nb.predict_proba(X_test)[:,1]
fpr_nb,tpr_nb,_ = roc_curve(y_test.values,pred_nb)
roc_auc_nb = auc(fpr_nb,tpr_nb)
f, axes = plt.subplots(2, 2,figsize=(15,10))
axes[0,0].plot(fpr_svm, tpr_svm, color='darkred', lw=2, label='ROC curve (area = {:0.2f})'.format(roc_auc_svm))
axes[0,0].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[0,0].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[0,0].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title = 'Support Vector Machine')
axes[0,0].legend(loc='lower right', fontsize=13)

axes[0,1].plot(fpr_lr, tpr_lr, color='darkred', lw=2, label='ROC curve (area = {:0.2f})'.format(roc_auc_lr))
axes[0,1].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[0,1].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[0,1].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title = 'Logistic Regression')
axes[0,1].legend(loc='lower right', fontsize=13)

axes[1,0].plot(fpr_rf_model, tpr_rf_model, color='darkred', lw=2, label='ROC curve (area = {:0.2f})'.format(roc_auc_rf_
axes[1,0].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[1,0].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[1,0].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title = 'Random Forest')
axes[1,0].legend(loc='lower right', fontsize=13)

axes[1,1].plot(fpr_nb, tpr_nb, color='darkred', lw=2, label='ROC curve (area = {:0.2f})'.format(roc_auc_nb))
axes[1,1].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[1,1].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[1,1].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title = 'Naive Bayes')
axes[1,1].legend(loc='lower right', fontsize=13);
```

Figure 13: AUC and ROC

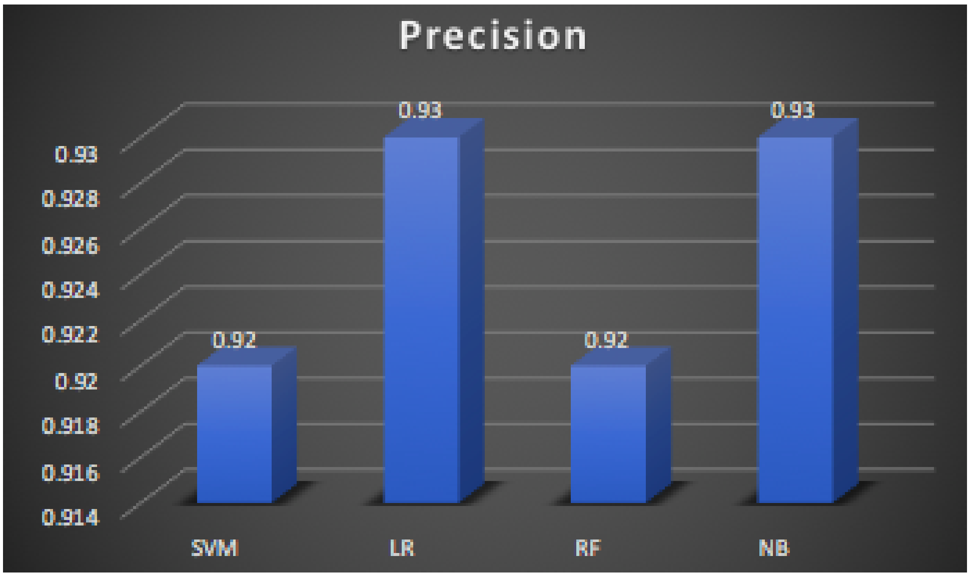


Figure 14: Precision

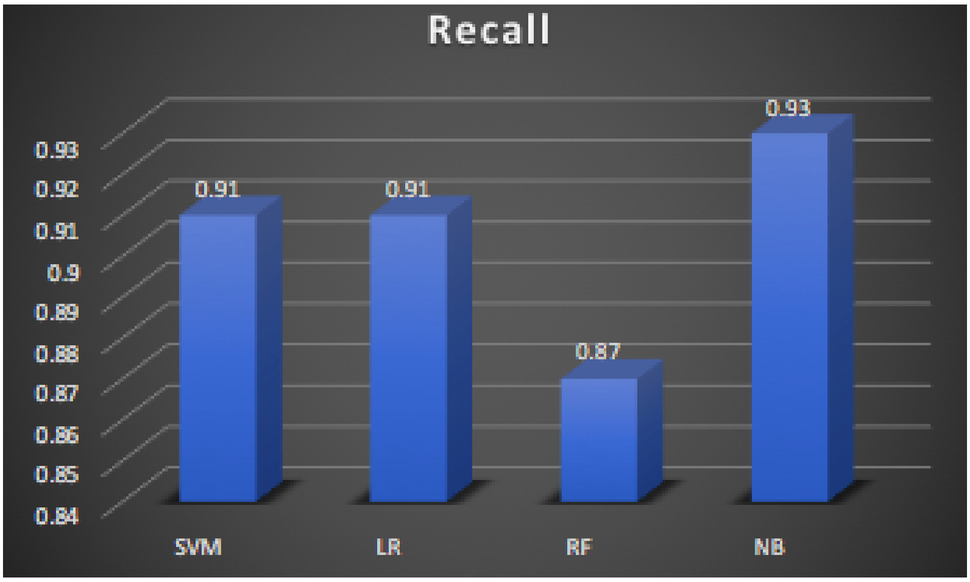


Figure 15: Recall

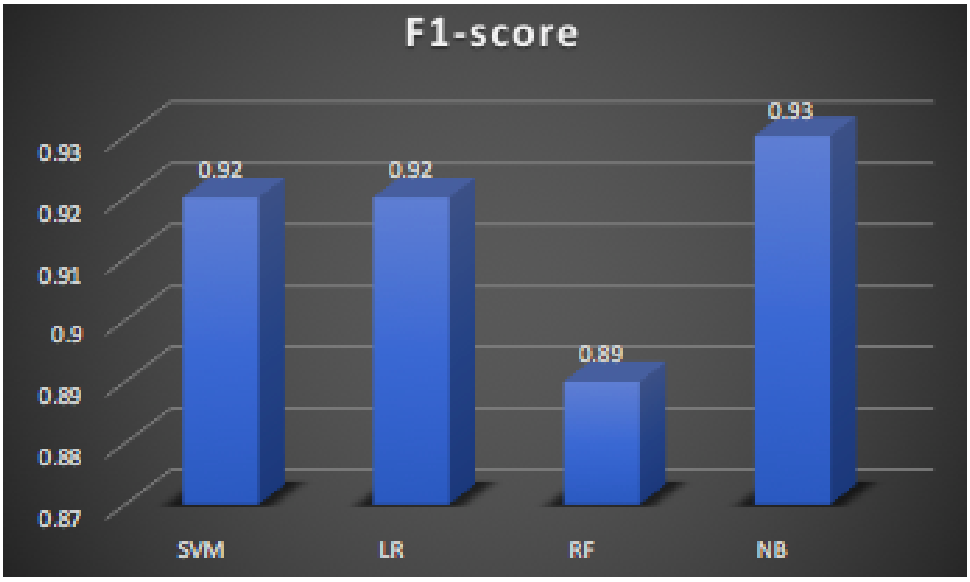


Figure 16: F1-score

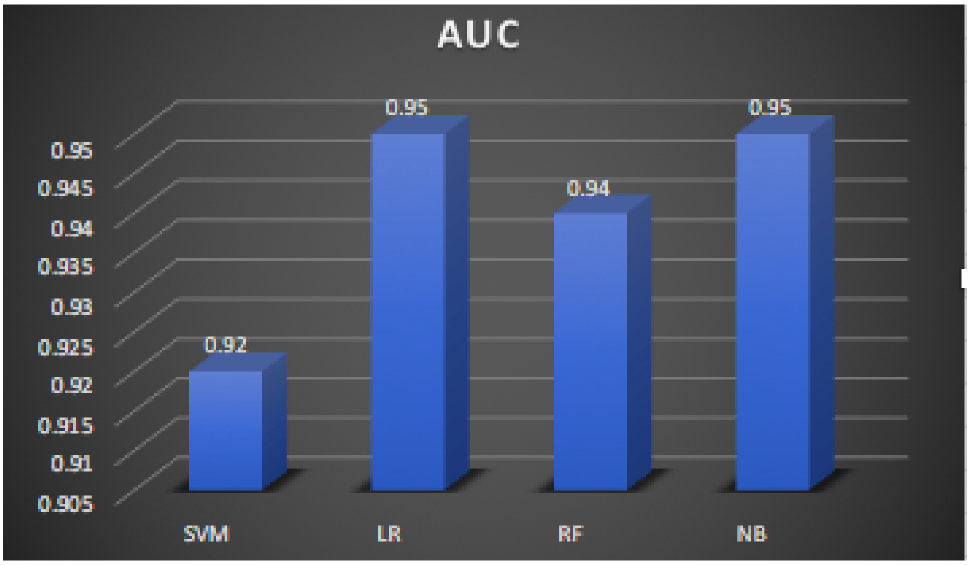


Figure 17: AUC

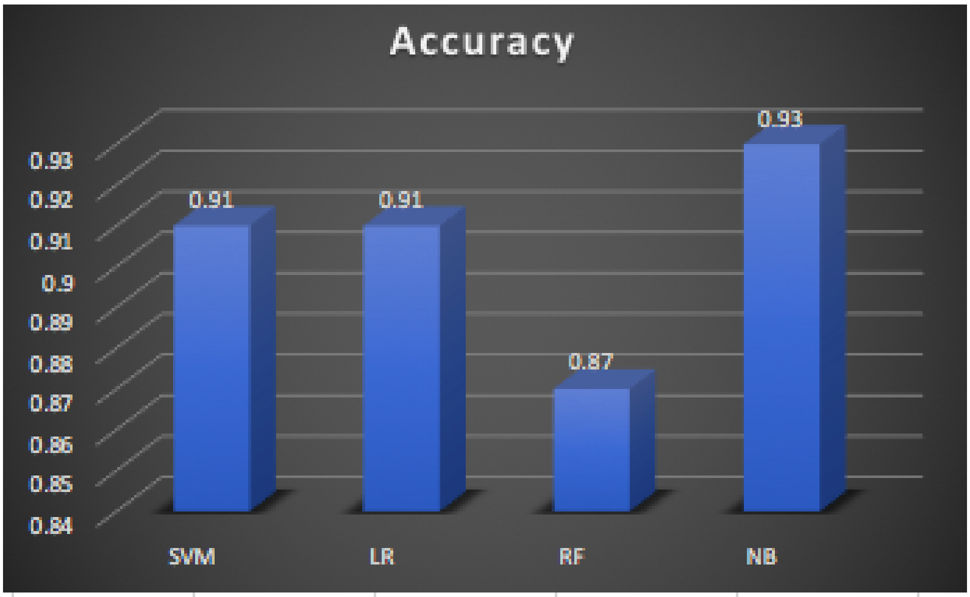


Figure 18: Accuracy