A Project Report

On

Product Review Scrapping & Analyzing Tool for Flipkart

Submitted by

ABHISHEK SAXENA	**(21566001)**
ADITYA KRISHNA	**(21566002)**
JATIN BHANDARI	**(21566007)**
SUMANTH H S	**(21566017)**
VINEET GUPTA	**(21566020)**

Under the Supervision of

Dr. Gaurav Dixit

Indian Institute of Technology Roorkee

ABSTRACT

As we can see, there are numerous products available on Flipkart in the same category that is used for the same task. As a result, it is extremely difficult for a user to choose the best among them. Additionally, the manufacturer also wants to know about the product's shortcomings, so he must read through all of the user feedback, which is a time-consuming task.

The buyer wants to know what the product's advantages are, and the producer wants to know what the product's disadvantages are so that they can be reduced in the next variant.

Thus, we are developing an end-to-end model that takes the product name as input, scrapes all the reviews from the Flipkart website, performs required data cleaning, applies NLP sentiment analysis and classifies them, and then extracts the most frequently used words by consumers about the product and summarizes the product's advantages and disadvantages for the client.

INTRODUCTION

Large volumes of textual data are generated by e-commerce websites. These companies extract valuable insights that can aid in a better understanding of the end-user. Flipkart's reviews, for example, crawls through the website reading, understanding, and coming up with insights about various product categories that may be used to produce better products for its private label by analyzing product reviews.

The proportion of returns on Billion items is half that of other returns on the marketplace. Seller insights, better delivery, advice, and competition analysis could all be potential use cases for the analyzer tool.

We may use word clouds to gather the most frequently used positive and negative words by analyzing product reviews using text mining. We may conclude that text mining provides insight into client sentiment and can assist businesses in resolving issues. This method allows you to improve the entire client experience while also making a lot of money.

Billion uses artificial intelligence extensively for review analysis in order to make product suggestions at the 'aspect' level, forecast demand, and better target clients for marketing.

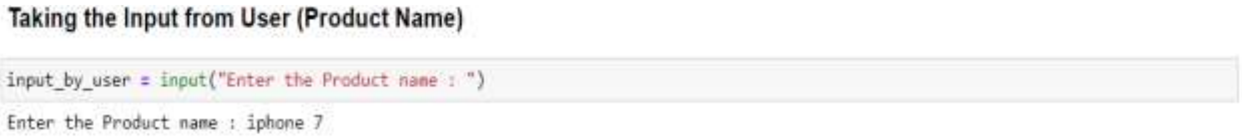
DATASET

All the Reviews for a particular product on Flipkart are scrapped using text mining. These reviews will be used as Dataset to apply NLP & Classification algorithms.

PROJECT FLOW

- Get the Product Name from user. Extract all the reviews available on flipkart
- Apply NLP tasks on the obtained dataset (Bag of Words Model).
- Apply Classification models to predict whether a review is positive or negative.
- Again apply NLP tasks to obtain the Major highlights/Pros & Major Drawbacks/Cons of the product.

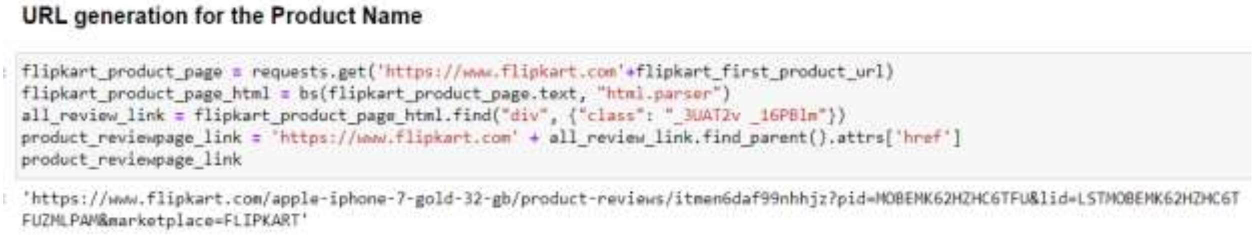
IMPLEMENTATION



Taking the Input from User (Product Name)

```
input_by_user = input("Enter the Product name : ")
```

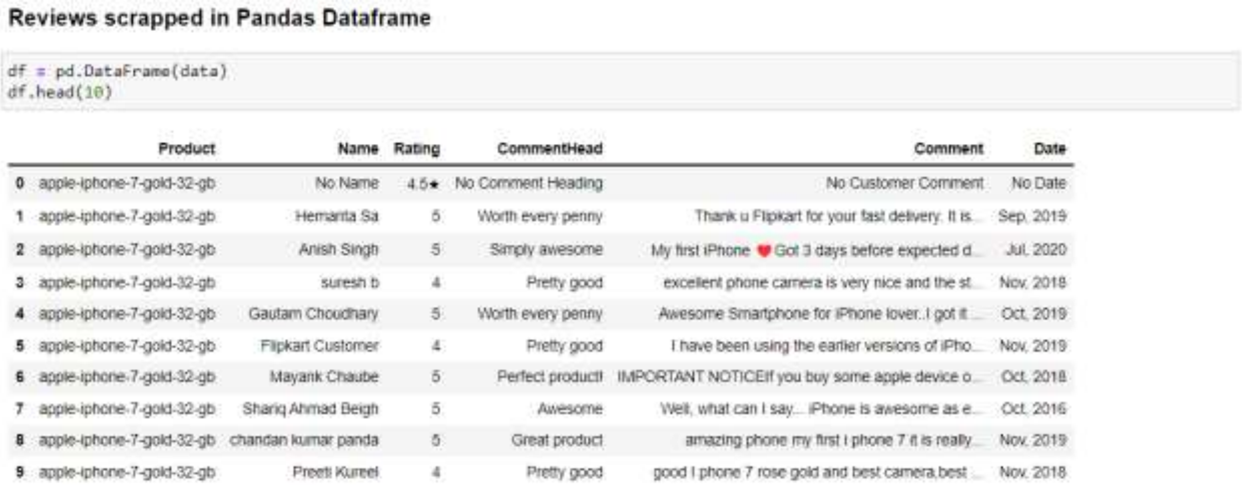
```
Enter the Product name : iphone 7
```



URL generation for the Product Name

```
flipkart_product_page = requests.get('https://www.flipkart.com'+flipkart_first_product_url)
flipkart_product_page_html = bs(flipkart_product_page.text, "html.parser")
all_review_link = flipkart_product_page_html.find("div", {"class": "_3UAT2v _16PBlm"})
product_reviewpage_link = 'https://www.flipkart.com' + all_review_link.find_parent().attrs['href']
product_reviewpage_link
```

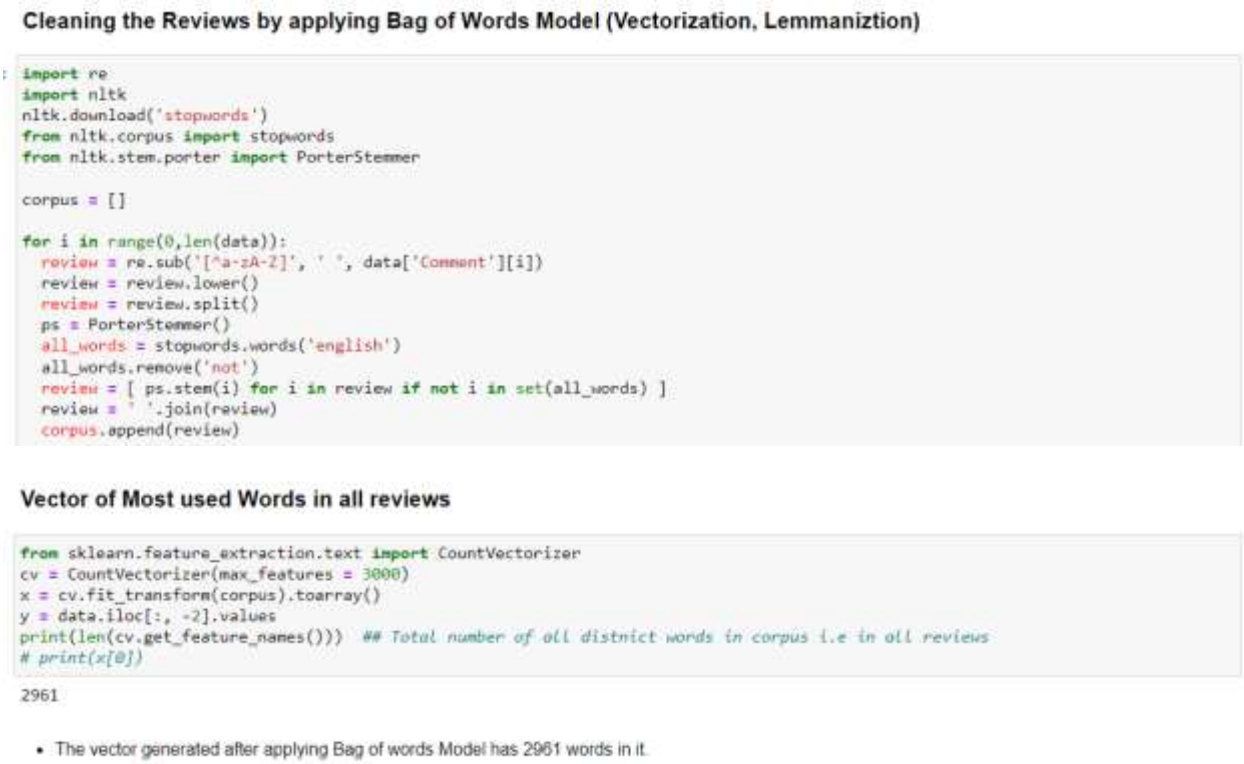
```
'https://www.flipkart.com/apple-iphone-7-gold-32-gb/product-reviews/itmen6daf99nhhjz?pid=MOBEMK62HZHC6TFU&lid=LSTMOBEMK62HZHC6T
FUZMLPAM&marketplace=FLIPKART'
```



Reviews scrapped in Pandas Dataframe

```
df = pd.DataFrame(data)
df.head(10)
```

	Product	Name	Rating	CommentHead	Comment	Date
0	apple-iphone-7-gold-32-gb	No Name	4.5★	No Comment Heading	No Customer Comment	No Date
1	apple-iphone-7-gold-32-gb	Hemanta Sa	5	Worth every penny	Thank u Flipkart for your fast delivery. It is...	Sep, 2019
2	apple-iphone-7-gold-32-gb	Anish Singh	5	Simply awesome	My first iPhone ❤️ Got 3 days before expected d...	Jul, 2020
3	apple-iphone-7-gold-32-gb	suresh b	4	Pretty good	excellent phone camera is very nice and the st...	Nov, 2018
4	apple-iphone-7-gold-32-gb	Gautam Choudhary	5	Worth every penny	Awesome Smartphone for iPhone lover..I got it ...	Oct, 2019
5	apple-iphone-7-gold-32-gb	Flipkart Customer	4	Pretty good	I have been using the earlier versions of iPho...	Nov, 2019
6	apple-iphone-7-gold-32-gb	Mayank Chaube	5	Perfect product!	IMPORTANT NOTICEIf you buy some apple device o...	Oct, 2018
7	apple-iphone-7-gold-32-gb	Shariq Ahmad Beigh	5	Awesome	Well, what can I say... iPhone is awesome as e...	Oct, 2016
8	apple-iphone-7-gold-32-gb	chandan kumar panda	5	Great product	amazing phone my first I phone 7 it is really...	Nov, 2019
9	apple-iphone-7-gold-32-gb	Preeti Kureel	4	Pretty good	good I phone 7 rose gold and best camera,best ...	Nov, 2018



Cleaning the Reviews by applying Bag of Words Model (Vectorization, Lemmaniztion)

```
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

corpus = []

for i in range(0,len(data)):
  review = re.sub('[^a-zA-Z]', ' ', data['Comment'][i])
  review = review.lower()
  review = review.split()
  ps = PorterStemmer()
  all_words = stopwords.words('english')
  all_words.remove('not')
  review = [ ps.stem(i) for i in review if not i in set(all_words) ]
  review = ' '.join(review)
  corpus.append(review)
```

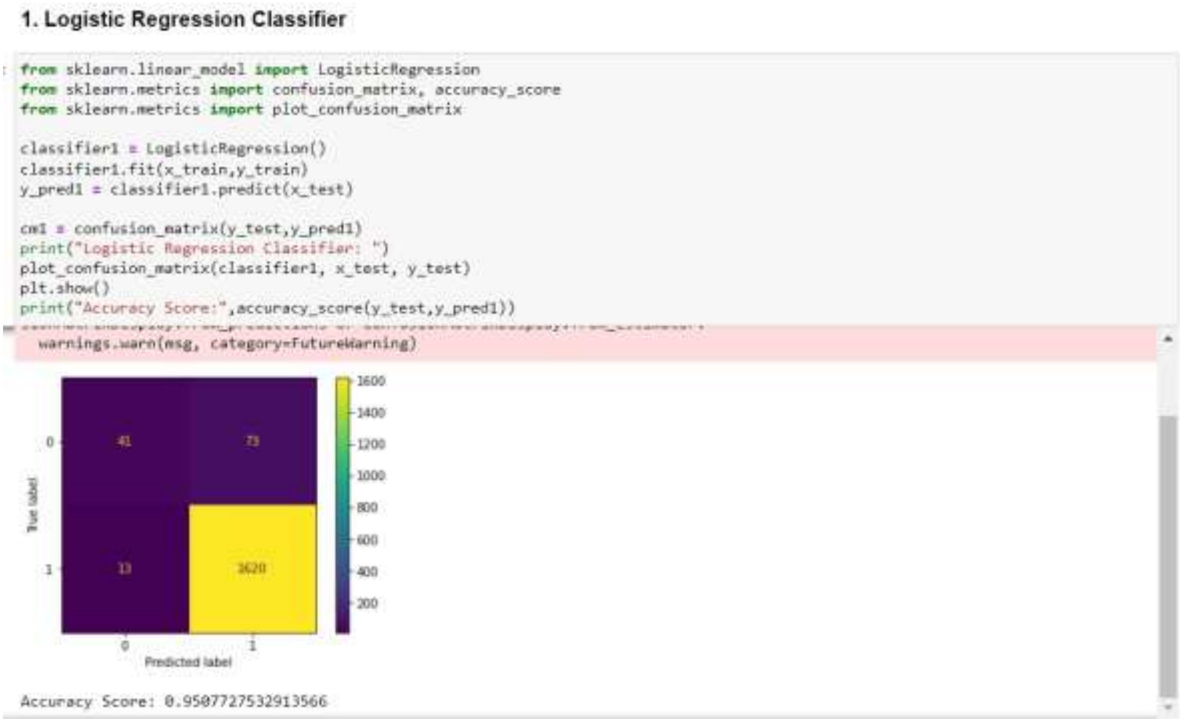
Vector of Most used Words in all reviews

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 3000)
x = cv.fit_transform(corpus).toarray()
y = data.iloc[:, -2].values
print(len(cv.get_feature_names()))  ## Total number of all distnict words in corpus i.e in all reviews
# print(x[0])
```

```
2961
```

- The vector generated after applying Bag of words Model has 2961 words in it.

- After getting the vector from Bag of words model, several classification algorithms are applied to predict whether a new review is positive or negative.

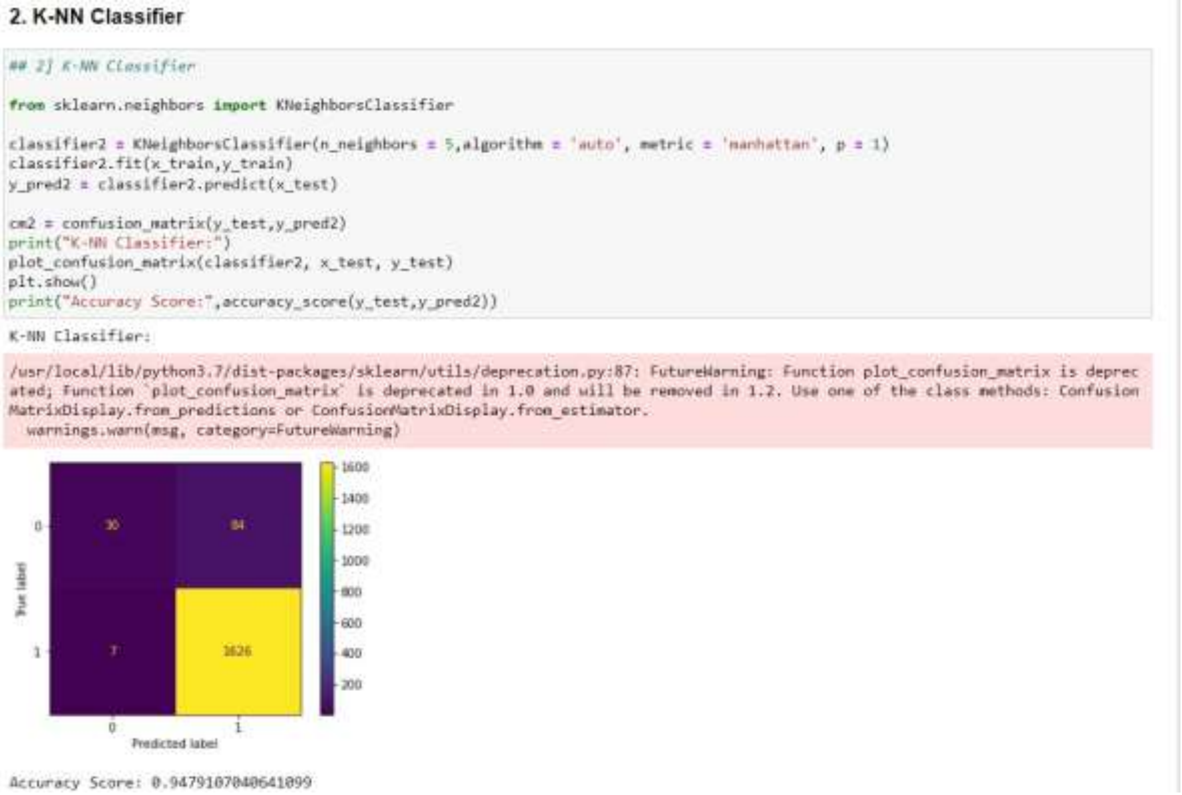


1. Logistic Regression Classifier

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.metrics import plot_confusion_matrix

classifier1 = LogisticRegression()
classifier1.fit(x_train,y_train)
y_pred1 = classifier1.predict(x_test)

cm1 = confusion_matrix(y_test,y_pred1)
print("Logistic Regression Classifier: ")
plot_confusion_matrix(classifier1, x_test, y_test)
plt.show()
print("Accuracy Score:",accuracy_score(y_test,y_pred1))
```

```
  warnings.warn(msg, category=FutureWarning)
```

```
Accuracy Score: 0.9507727532913566
```



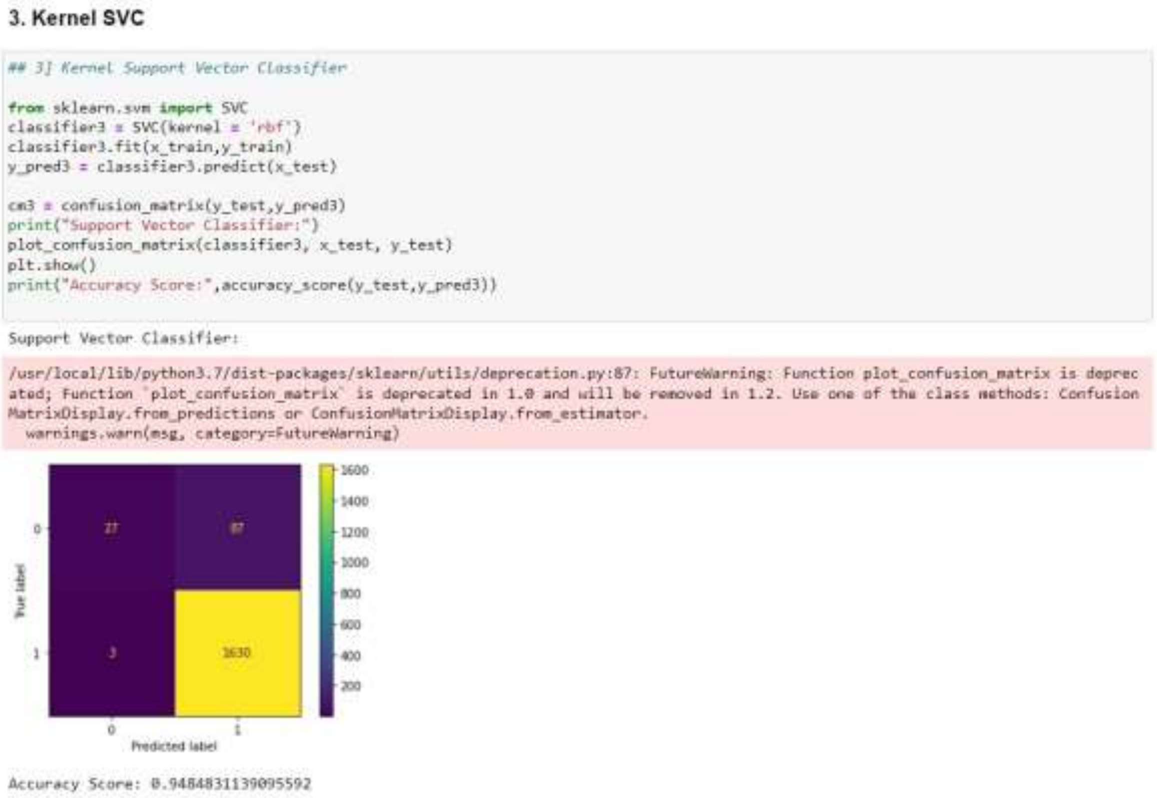
2. K-NN Classifier

```
## 2] K-NN Classifier

from sklearn.neighbors import KNeighborsClassifier

classifier2 = KNeighborsClassifier(n_neighbors = 5,algorithm = 'auto', metric = 'manhattan', p = 1)
classifier2.fit(x_train,y_train)
y_pred2 = classifier2.predict(x_test)

cm2 = confusion_matrix(y_test,y_pred2)
print("K-NN Classifier:")
plot_confusion_matrix(classifier2, x_test, y_test)
plt.show()
print("Accuracy Score:",accuracy_score(y_test,y_pred2))
```

```
K-NN Classifier:
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprec
ated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: Confusion
MatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```

```
Accuracy Score: 0.9479107040641099
```



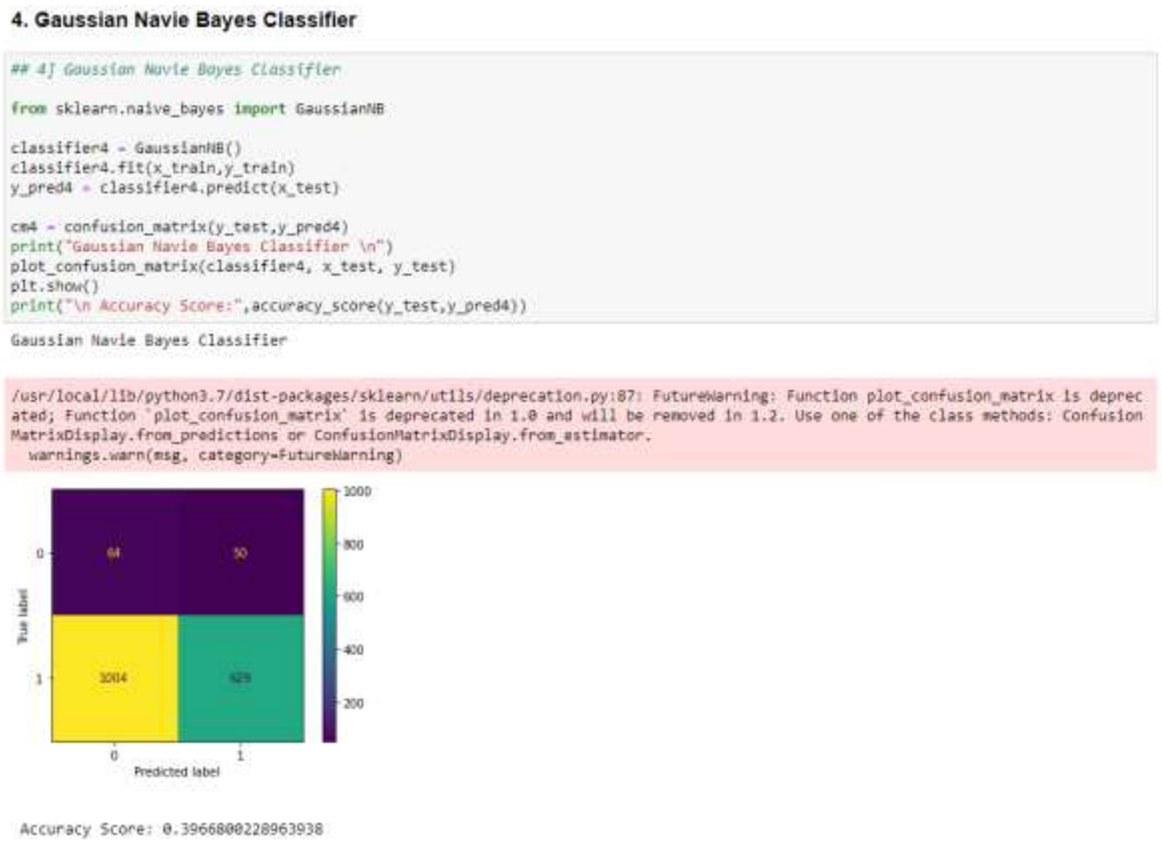
3. Kernel SVC

```
## 3] Kernel Support Vector Classifier

from sklearn.svm import SVC
classifier3 = SVC(kernel = 'rbf')
classifier3.fit(x_train,y_train)
y_pred3 = classifier3.predict(x_test)

cm3 = confusion_matrix(y_test,y_pred3)
print("Support Vector Classifier:")
plot_confusion_matrix(classifier3, x_test, y_test)
plt.show()
print("Accuracy Score:",accuracy_score(y_test,y_pred3))
```

```
Support Vector Classifier:
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprec
ated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: Confusion
MatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```

```
Accuracy Score: 0.9484831139095592
```



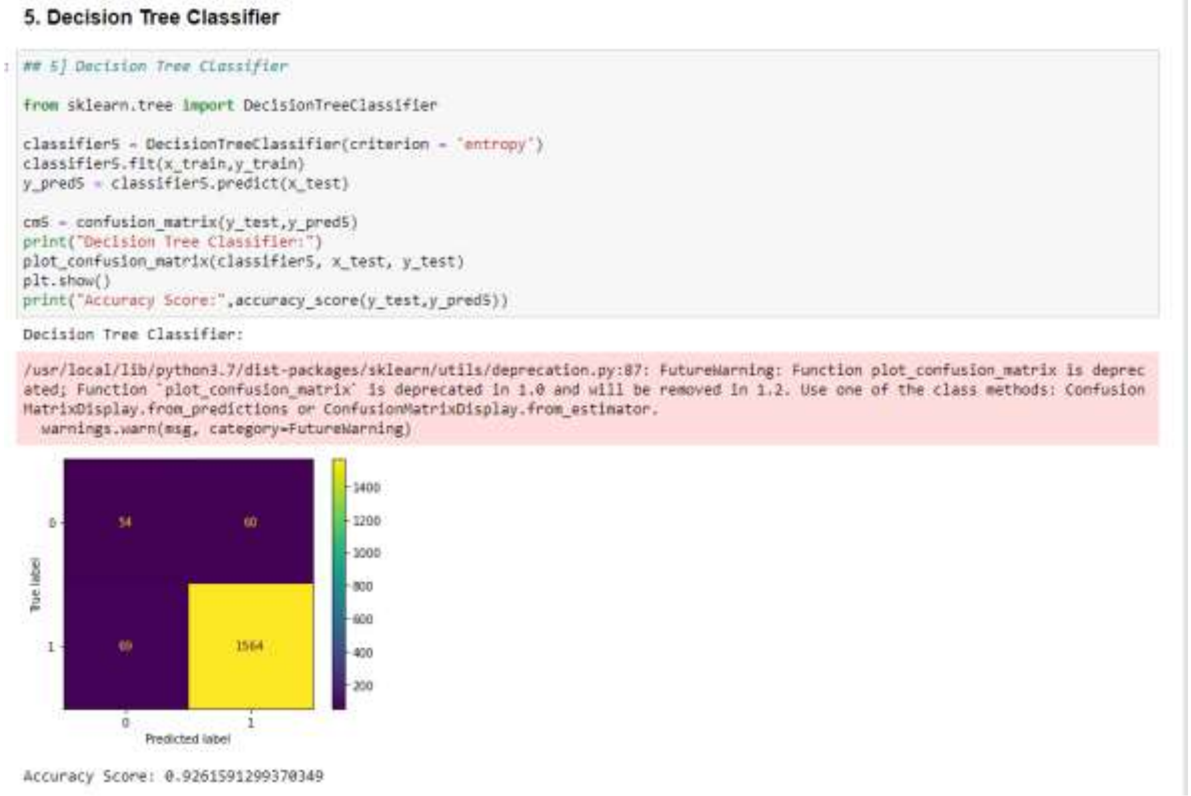
4. Gaussian Navie Bayes Classifier

```
## 4] Gaussian Navie Bayes Classifier

from sklearn.naive_bayes import GaussianNB

classifier4 = GaussianNB()
classifier4.fit(x_train,y_train)
y_pred4 = classifier4.predict(x_test)

cm4 = confusion_matrix(y_test,y_pred4)
print("Gaussian Navie Bayes Classifier \n")
plot_confusion_matrix(classifier4, x_test, y_test)
plt.show()
print("\n Accuracy Score:",accuracy_score(y_test,y_pred4))
```

```
Gaussian Navie Bayes Classifier
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprec
ated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: Confusion
MatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```

```
 Accuracy Score: 0.3966800228963938
```



5. Decision Tree Classifier

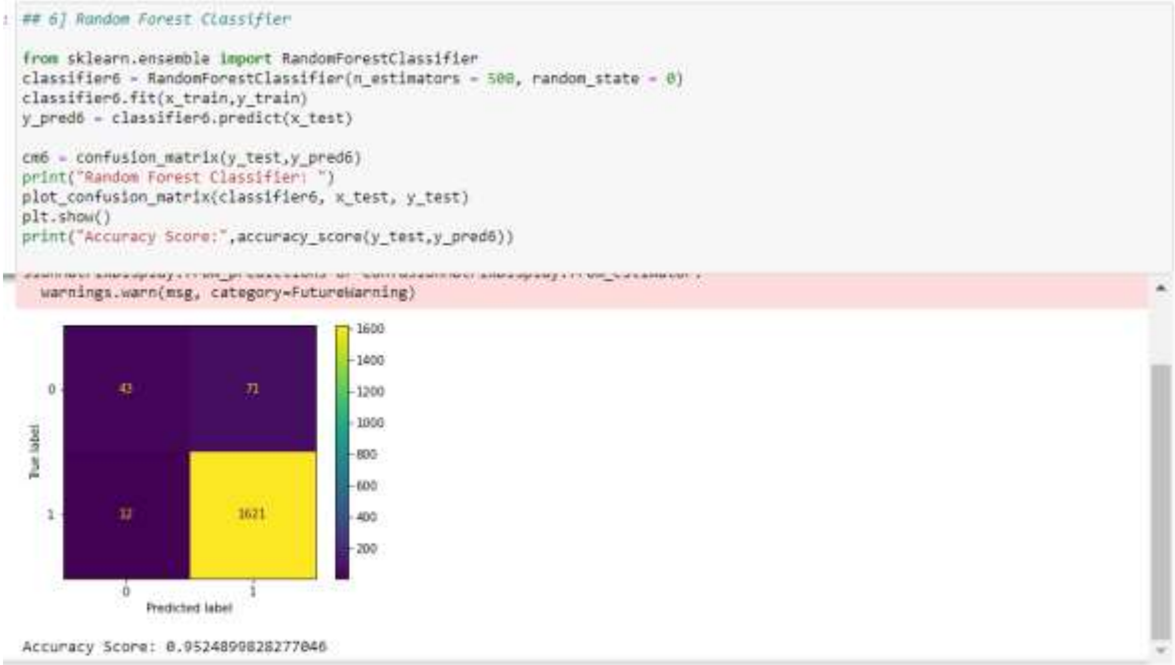
```
## 5] Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier

classifier5 = DecisionTreeClassifier(criterion = 'entropy')
classifier5.fit(x_train,y_train)
y_pred5 = classifier5.predict(x_test)

cm5 = confusion_matrix(y_test,y_pred5)
print("Decision Tree Classifier:")
plot_confusion_matrix(classifier5, x_test, y_test)
plt.show()
print("Accuracy Score:",accuracy_score(y_test,y_pred5))
```

```
Decision Tree Classifier:
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprec
ated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: Confusion
MatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```

```
Accuracy Score: 0.9261591299370349
```

6. Random Forest Classifier

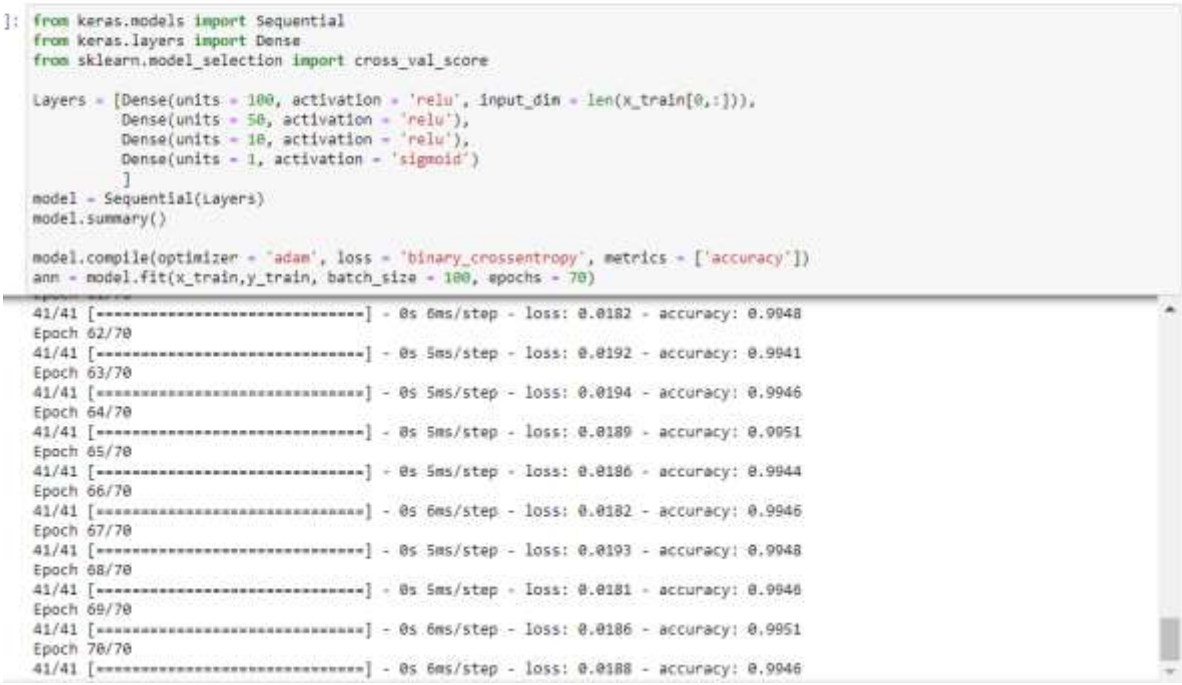


```
## 6] Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier
classifier6 = RandomForestClassifier(n_estimators = 500, random_state = 0)
classifier6.fit(x_train,y_train)
y_pred6 = classifier6.predict(x_test)

cm6 = confusion_matrix(y_test,y_pred6)
print("Random Forest Classifier: ")
plot_confusion_matrix(classifier6, x_test, y_test)
plt.show()
print("Accuracy Score:",accuracy_score(y_test,y_pred6))
```

```
  warnings.warn(msg, category=FutureWarning)
```

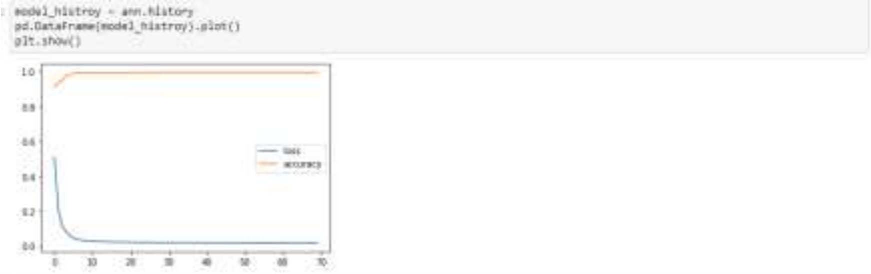
```
Accuracy Score: 0.9524899828277046
```

ANN Classifier

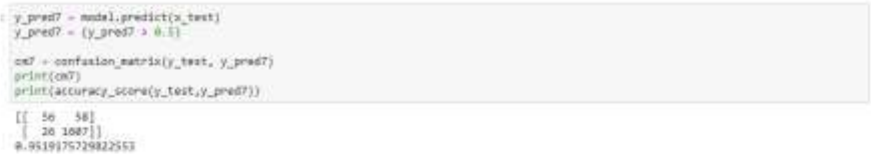


```
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import cross_val_score

Layers = [Dense(units = 100, activation = 'relu', input_dim = len(x_train[0,:])),
          Dense(units = 50, activation = 'relu'),
          Dense(units = 10, activation = 'relu'),
          Dense(units = 1, activation = 'sigmoid')
          ]
model = Sequential(Layers)
model.summary()

model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
ann = model.fit(x_train,y_train, batch_size = 100, epochs = 70)
```

```
41/41 [==============================] - 0s 6ms/step - loss: 0.0182 - accuracy: 0.9948
Epoch 62/70
41/41 [==============================] - 0s 5ms/step - loss: 0.0192 - accuracy: 0.9941
Epoch 63/70
41/41 [==============================] - 0s 5ms/step - loss: 0.0194 - accuracy: 0.9946
Epoch 64/70
41/41 [==============================] - 0s 5ms/step - loss: 0.0189 - accuracy: 0.9951
Epoch 65/70
41/41 [==============================] - 0s 5ms/step - loss: 0.0186 - accuracy: 0.9944
Epoch 66/70
41/41 [==============================] - 0s 6ms/step - loss: 0.0182 - accuracy: 0.9946
Epoch 67/70
41/41 [==============================] - 0s 5ms/step - loss: 0.0193 - accuracy: 0.9948
Epoch 68/70
41/41 [==============================] - 0s 5ms/step - loss: 0.0181 - accuracy: 0.9946
Epoch 69/70
41/41 [==============================] - 0s 6ms/step - loss: 0.0186 - accuracy: 0.9951
Epoch 70/70
41/41 [==============================] - 0s 6ms/step - loss: 0.0188 - accuracy: 0.9946
```

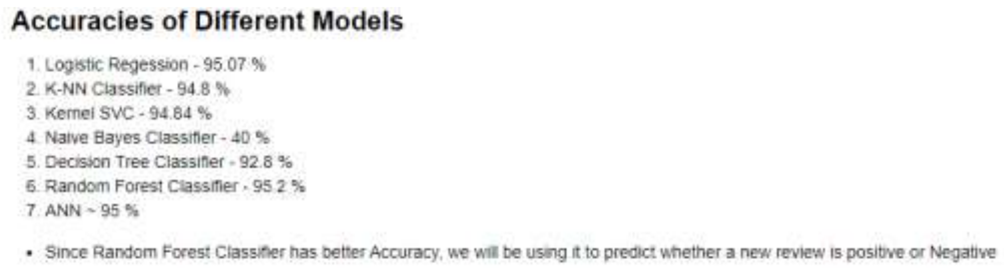


```
model_histroy = ann.history
pd.DataFrame(model_histroy).plot()
plt.show()
```



```
y_pred7 = model.predict(x_test)
y_pred7 = (y_pred7 > 0.5)

cm7 = confusion_matrix(y_test, y_pred7)
print(cm7)
print(accuracy_score(y_test,y_pred7))
```

```
[[  56   58]
 [  26 1607]]
0.9519175729822553
```



Accuracies of Different Models

1. Logistic Regession - 95.07 %
2. K-NN Classifier - 94.8 %
3. Kernel SVC - 94.84 %
4. Naive Bayes Classifier - 40 %
5. Decision Tree Classifier - 92.8 %
6. Random Forest Classifier - 95.2 %
7. ANN ~ 95 %

- Since Random Forest Classifier has better Accuracy, we will be using it to predict whether a new review is positive or Negative

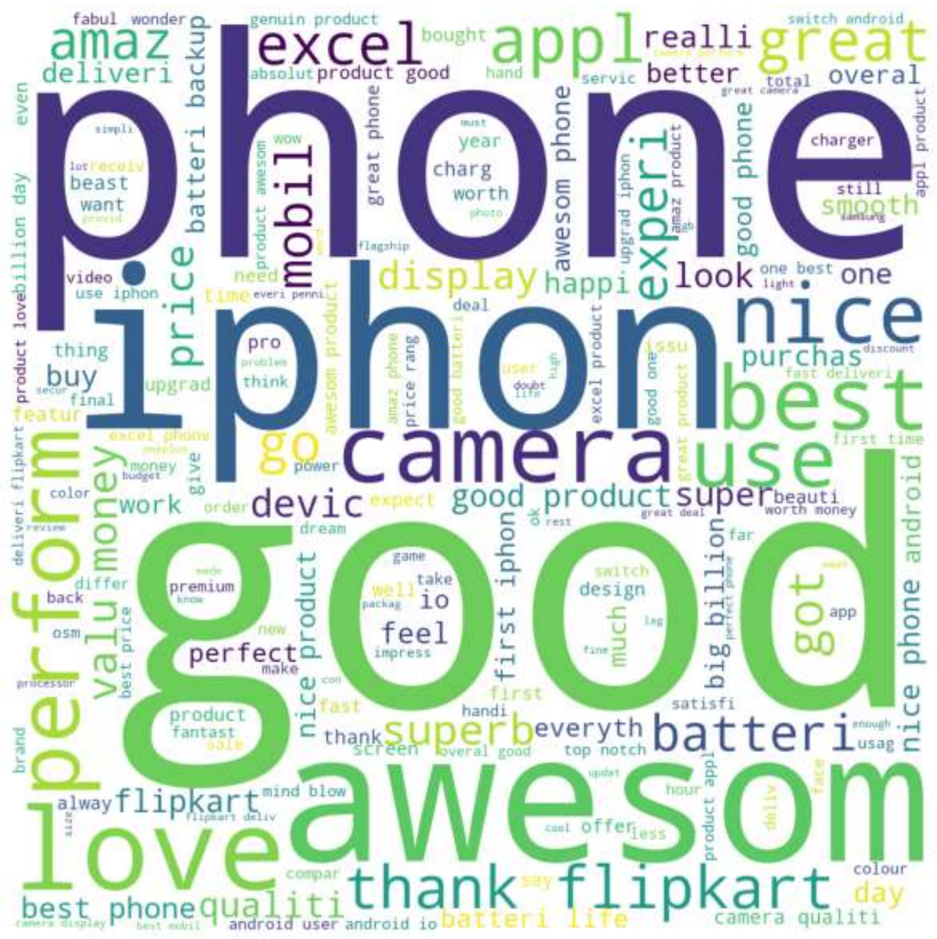
- The above are the accuracies of different models, since Random Forest model has he best accuracy, we will be moving ahead with it.

Creating Word Cloud for Positive and Negative Reviews

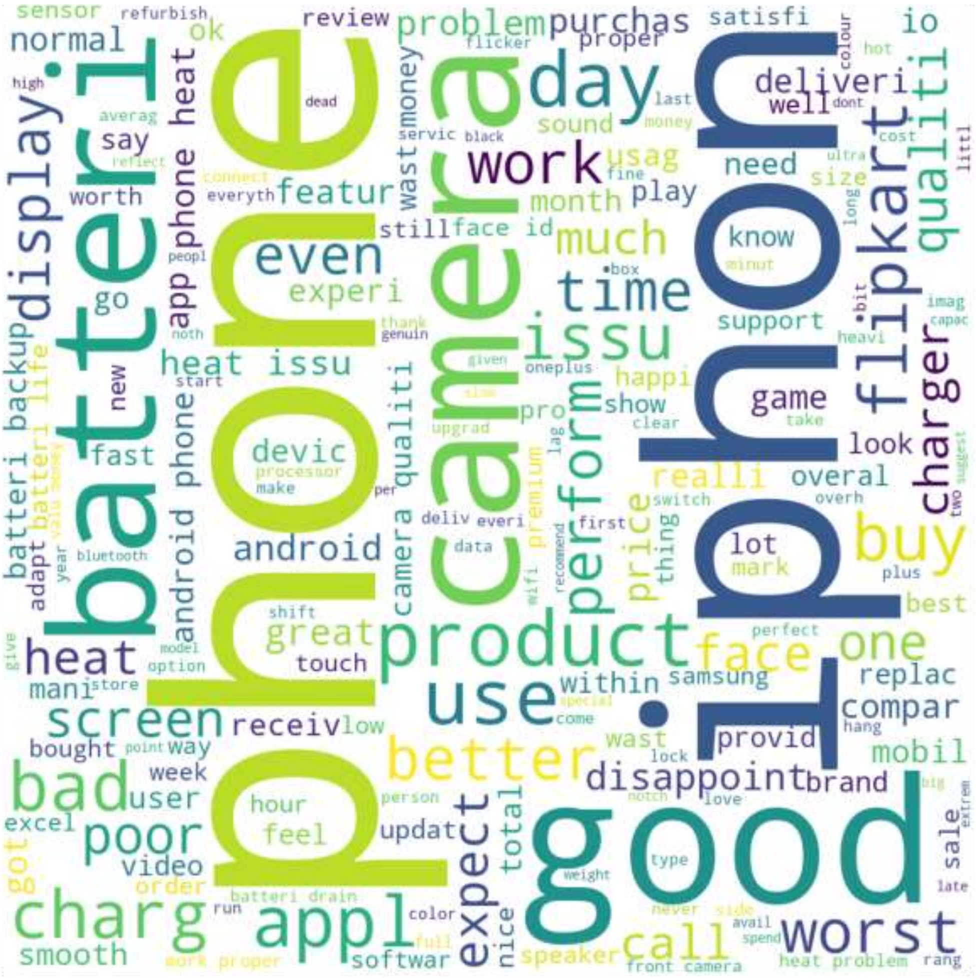
- We will split all the reviews into 2 categories (Positive & Negative)
- For each category we will create a Word Cloud to know the most frequently occurred words in the reviews.

Below is the Word Cloud for Positive Reviews

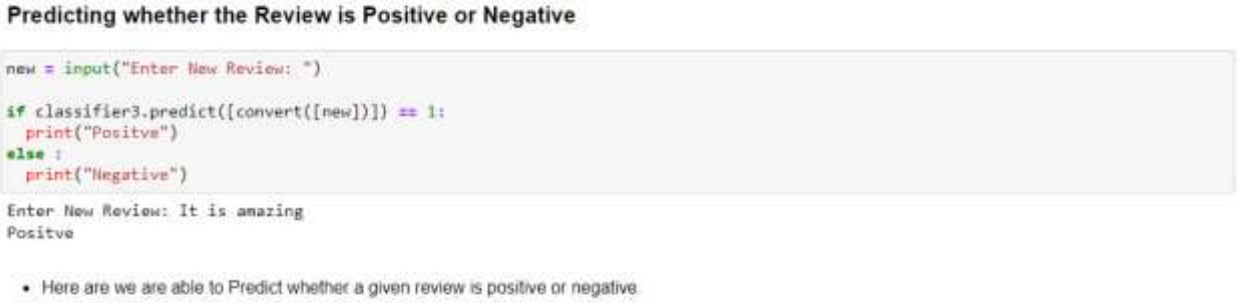
World Cloud for Positive Reviews :



Below is the Word Cloud for Negative Reviews



- From the above 2 Images we can also get to know the sentiment of the buyers.
- Using the classification model we can also predict whether a new review is positive or negative as below.



Predicting whether the Review is Positive or Negative

```
new = input("Enter New Review: ")

if classifier3.predict([convert([new])]) == 1:
  print("Positve")
else :
  print("Negative")
```

```
Enter New Review: It is amazing
Positve
```

- Here are we are able to Predict whether a given review is positive or negative

Conclusion

- Just by entering the name of product, we are able to know its Major pros & cons without actually going through all the reviews on Flipkart

- For any new review, we are also able to predict whether it’s Positive or Negative

- Both Customers & Company personnel’s can get to know the product’s sentiment without much hassle.