In [1]:

```
#Natural Language Processing Project
#Sentiment Analysis on Google Play Store Applications user Review Data
#Dataset Collected from Kaggle
#Sentiment Analysis using NLTK
```

In [2]:

```
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#importing libraries for Data Cleaning & NLTK Processing
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
```

```
[nltk_data] Downloading package stopwords to C:\Users\Chirag
[nltk_data]     mahawar\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [3]:

```
#importing the dataset
dataset = pd.read_csv('googleplaystore_user_reviews.csv')
```

In [4]:

```
dataset.head()
```

Out[4]:

	Translated_Review	Sentiment
0	I like eat delicious food. That's I'm cooking ...	Positive
1	This help eating healthy exercise regular basis	Positive
2	NaN	NaN
3	Works great especially going grocery store	Positive
4	Best idea us	Positive

In [5]:

```
#Dropping NA
dataset=dataset.dropna(axis=0)
```

In [6]:

```
data=dataset.values
```

In [7]:

```
len(data)
```

Out[7]:

```
37427
```

In [8]:

```
#Cleaning the dataset
def clean(data):
    corpus = []
    all_stop=stopwords.words('english')
    for i in range(len(data)):
        review = re.sub('[^a-zA-Z]', ' ',data[i] )
        review = review.lower()
        review = review.split()
        ps = PorterStemmer()
        review = [ps.stem(word) for word in review if not word in all_stop]
        review=[rev for rev in review if len(rev)>1]
        review = ' '.join(review)
        corpus.append(review)
    return corpus
```

In [9]:

```
data=clean(dataset.iloc[:,0].values)
```

In [10]:

```
len(data)
```

Out[10]:

```
37427
```

In [11]:

```
data[0]
```

Out[11]:

```
'like eat delici food cook food case best food help lot also best shelf li
fe'
```

In [12]:

```
from sklearn.preprocessing import LabelEncoder
```

In [13]:

```
le=LabelEncoder()
```

In [14]:

```
labels=le.fit_transform(dataset.iloc[:,1])
```

In [15]:

```
labels.shape
```

Out[15]:

```
(37427,)
```

In [16]:

```
# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(data).toarray()
```

In [17]:

```
#splitting the dataset into training & test set
from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(X , labels , test_size = 0.20 ,
random_state = 0)
```

In [18]:

```
#using random Forest Classifier as Classification Model for NLP
#Fitting the Random Forest to the training set
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(30)
classifier.fit(X_train , y_train)
```

Out[18]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gin
i',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=30, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

In [19]:

```
#predicting the test set results
y_pred = classifier.predict(X_test)
```

In [20]:

```
#making the confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test , y_pred)
```

In [21]:

```
cm
```

Out[21]:

```
array([[1332,   56,  299],
       [  29,  915,  116],
       [ 108,   93, 4538]], dtype=int64)
```

In [22]:

```
#Calculating Accuracy of the Model
from sklearn.metrics import accuracy_score
print(round(accuracy_score(y_test,y_pred)*100,2),"%",sep=" ")
```

```
90.64 %
```

In [24]:

```
classifier.predict(cv.transform(clean(["I badly love this app"])))
```

Out[24]:

```
array([2], dtype=int64)
```

In [25]:

```
le.classes_
```

Out[25]:

```
array(['Negative', 'Neutral', 'Positive'], dtype=object)
```

In [26]:

```
classifier.predict(cv.transform(clean(["I think I am loving this app"])))
```

Out[26]:

```
array([2], dtype=int64)
```