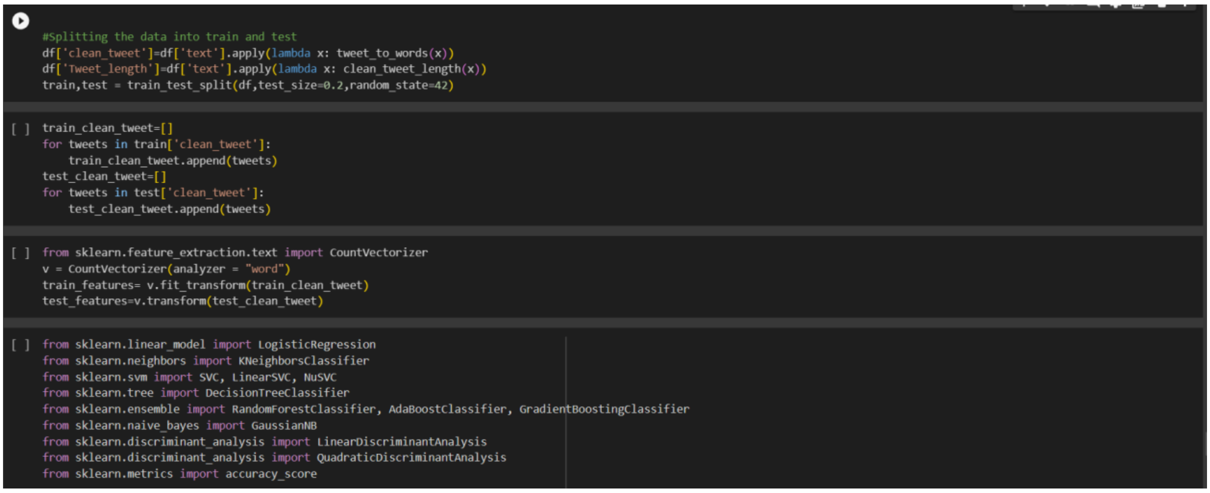
SENTIMENT ANALYSIS FOR MARKETING

PROJECT TITLE	SENTIMENT ANALYSIS FOR MAKETING
SKILLS TAKEN AWAY	❖ Python script ❖ EDA ❖ UI deployment
DOMAIN	FMCG[FAST MOVING CONSUMER GOODS]

TRAINING THE MODEL

1.SPLITING THE DATA TO TRAIN AND TEST :

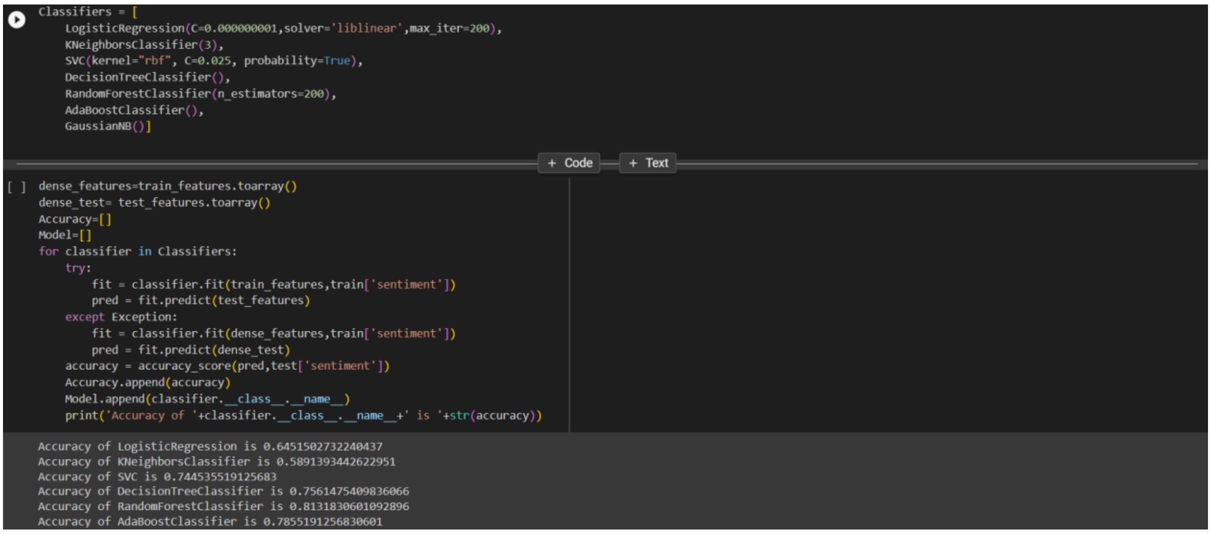


```
#Splitting the data into train and test
df['clean_tweet']=df['text'].apply(lambda x: tweet_to_words(x))
df['Tweet_length']=df['text'].apply(lambda x: clean_tweet_length(x))
train,test = train_test_split(df,test_size=0.2,random_state=42)
```

```
train_clean_tweet=[]
for tweets in train['clean_tweet']:
    train_clean_tweet.append(tweets)
test_clean_tweet=[]
for tweets in test['clean_tweet']:
    test_clean_tweet.append(tweets)
```

```
from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer(analyzer = "word")
train_features= v.fit_transform(train_clean_tweet)
test_features=v.transform(test_clean_tweet)
```

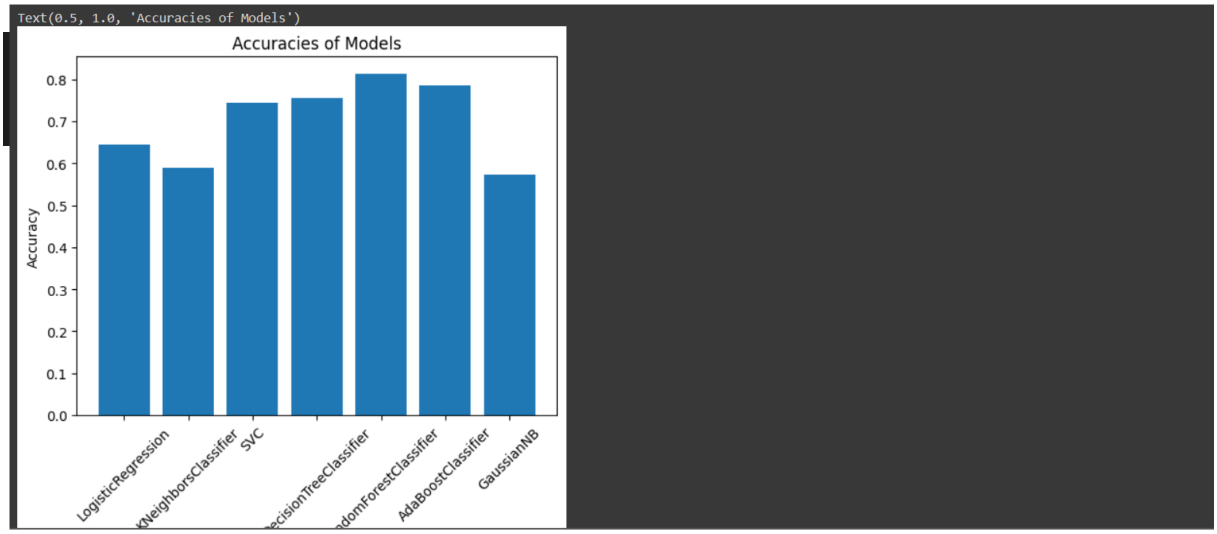
```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC, LinearSVC, NuSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.metrics import accuracy_score
```



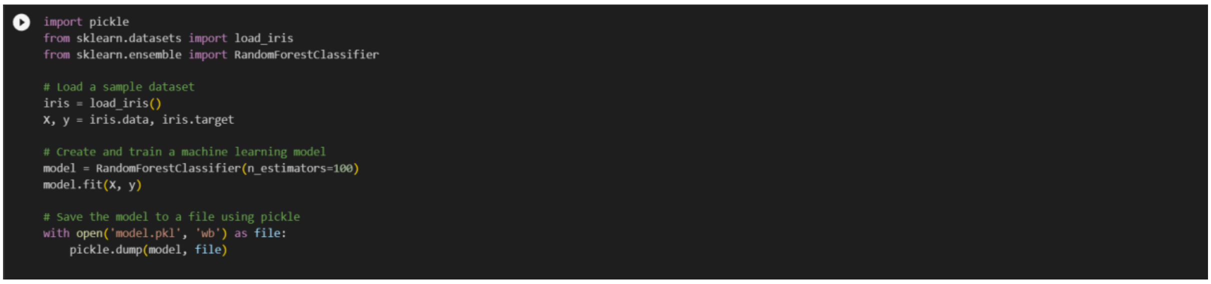
```
Classifiers = [
    LogisticRegression(C=0.000000001,solver='liblinear',max_iter=200),
    KNeighborsClassifier(3),
    SVC(kernel="rbf", C=0.025, probability=True),
    DecisionTreeClassifier(),
    RandomForestClassifier(n_estimators=200),
    AdaBoostClassifier(),
    GaussianNB()]
```

```
dense_features=train_features.toarray()
dense_test= test_features.toarray()
Accuracy=[]
Model=[]
for classifier in Classifiers:
    try:
        fit = classifier.fit(train_features,train['sentiment'])
        pred = fit.predict(test_features)
    except Exception:
        fit = classifier.fit(dense_features,train['sentiment'])
        pred = fit.predict(dense_test)
    accuracy = accuracy_score(pred,test['sentiment'])
    Accuracy.append(accuracy)
    Model.append(classifier.__class__.__name__)
    print('Accuracy of '+classifier.__class__.__name__+' is '+str(accuracy))
```

```
Accuracy of LogisticRegression is 0.6451502732240437
Accuracy of KNeighborsClassifier is 0.5891393442622951
Accuracy of SVC is 0.744535519125683
Accuracy of DecisionTreeClassifier is 0.7561475409836066
Accuracy of RandomForestClassifier is 0.8131830601092896
Accuracy of AdaBoostClassifier is 0.7855191256830601
```



```
Text(0.5, 1.0, 'Accuracies of Models')
```



```
import pickle
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier

# Load a sample dataset
iris = load_iris()
X, y = iris.data, iris.target

# Create and train a machine learning model
model = RandomForestClassifier(n_estimators=100)
model.fit(X, y)

# Save the model to a file using pickle
with open('model.pkl', 'wb') as file:
    pickle.dump(model, file)
```