

## Model Development Phase Template

Date	5th July 2024
Team ID	739687
Project Title	SMS SPAM DETECTION
Maximum Marks	4 Marks

### Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot.  
The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

### Initial Model Training Code:

## Creating a model using Multinomial NaiveBayes

```
from sklearn.naive_bayes import MultinomialNB  
model = MultinomialNB()
```

```
#Fitting the model to the training sets  
model.fit(X_train, y_train)
```

```
MultinomialNB()
```

## Prediction

```
y_pred=model.predict(X_test)  
y_pred  
array([0, 0, 0, ..., 0, 1, 0], dtype=uint8)
```

## Model Validation and Evaluation Report:

# Evaluating Model

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
score = accuracy_score(y_test, y_pred)
print(cm)
print('Accuracy Score Is:- ', score*100)
```

```
[[716  16]
 [ 17 286]]
```

```
Accuracy Score Is:- 96.81159420289856
```

```
from sklearn.svm import SVC
svm1=SVC(kernel='rbf')
svm1.fit(X_train,y_train)
```

SVC()

```
y_pred4=svm1.predict(X_test)
from sklearn.metrics import accuracy_score
svm_rbf=accuracy_score(y_test,y_pred4)
svm_rbf
```

0.9623188405797102

```
svm2=SVC(kernel='sigmoid')
svm2.fit(X_train,y_train)
```

SVC(kernel='sigmoid')

```
y_pred5=svm2.predict(X_test)
from sklearn.metrics import accuracy_score
svm_sig=accuracy_score(y_test,y_pred5)
svm_sig
```

0.9652173913043478

```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(X_train,y_train)
```

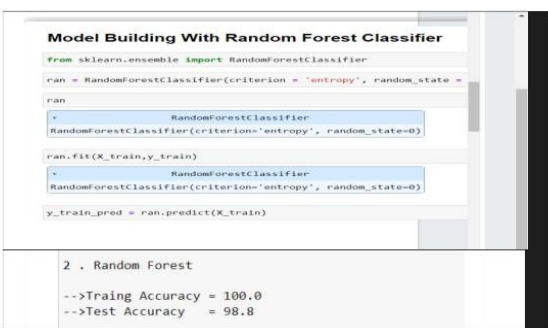
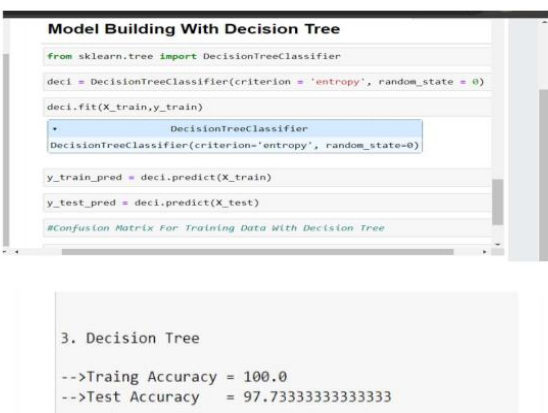
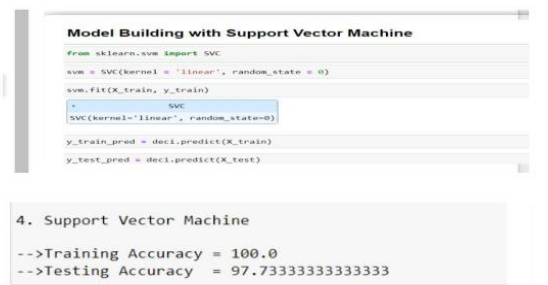
DecisionTreeClassifier()

```
y_pred6=dt.predict(X_test)
from sklearn.metrics import accuracy_score
dec_tree=accuracy_score(y_test,y_pred6)
dec_tree
```

0.9468599033816425

```
models = pd.DataFrame({
    'Model': [ 'MultinomialNB', 'SVM-rbf', 'SVM-sigmoid', 'Decision Tree'],
    'Test Score': [ score, svm_rbf, svm_sig, dec_tree,]})
models.sort_values(by='Test Score', ascending=False)
```

	Model	Test Score
0	MultinomialNB	0.968118
2	SVM-sigmoid	0.965217
1	SVM-rbf	0.962319
3	Decision Tree	0.946880

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest Regressor	 <pre> Model Building With Random Forest Classifier  from sklearn.ensemble import RandomForestClassifier  ran = RandomForestClassifier(criterion = 'entropy', random_state = 0)  ran = RandomForestClassifier(criterion='entropy', random_state=0)  ran.fit(X_train,y_train)  y_train_pred = ran.predict(X_train)  2 . Random Forest  --&gt;Traing Accuracy = 100.0 --&gt;Test Accuracy  = 98.8 </pre>	98%	
Decision Tree Regressor	 <pre> Model Building With Decision Tree  from sklearn.tree import DecisionTreeClassifier  deci = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)  deci.fit(X_train,y_train)  y_train_pred = deci.predict(X_train) y_test_pred = deci.predict(X_test)  #Confusion Matrix For Training Data With Decision Tree  3. Decision Tree  --&gt;Traing Accuracy = 100.0 --&gt;Test Accuracy  = 97.73333333333333 </pre>	97%	
Gradient Boosting Regressor	 <pre> Model Building with Support Vector Machine  from sklearn.svm import SVC  svm = SVC(kernel = 'linear', random_state = 0)  svm.fit(X_train, y_train)  y_train_pred = svm.predict(X_train) y_test_pred = svm.predict(X_test)  4. Support Vector Machine  --&gt;Training Accuracy = 100.0 --&gt;Testing Accuracy  = 97.73333333333333 </pre>	97%	