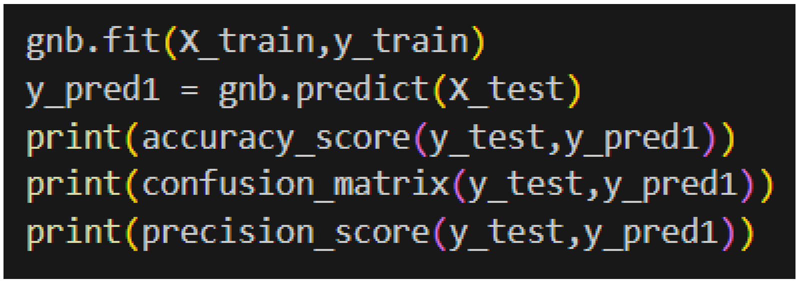
Model Development Phase Template

Date	15 March 2024
Team ID	SWTID1727180793
Project Title	SMS- Spam Detection Using NLP
Maximum Marks	10 Marks

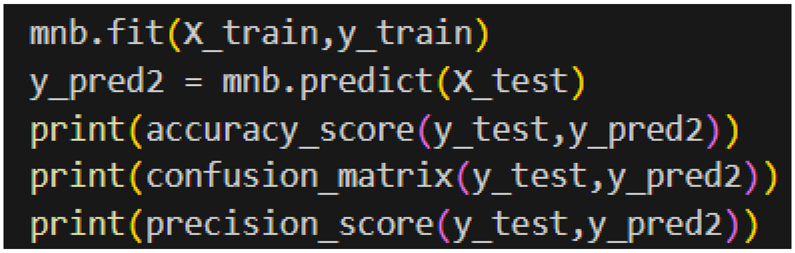
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

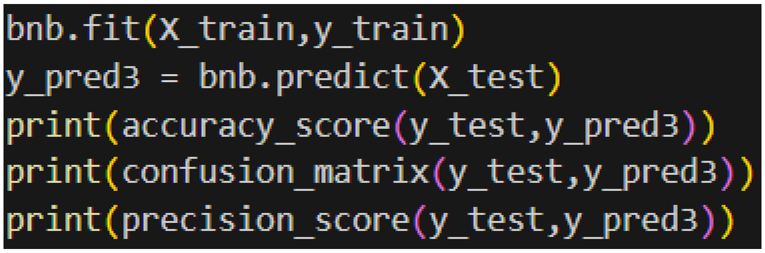
Initial Model Training Code (5 marks):



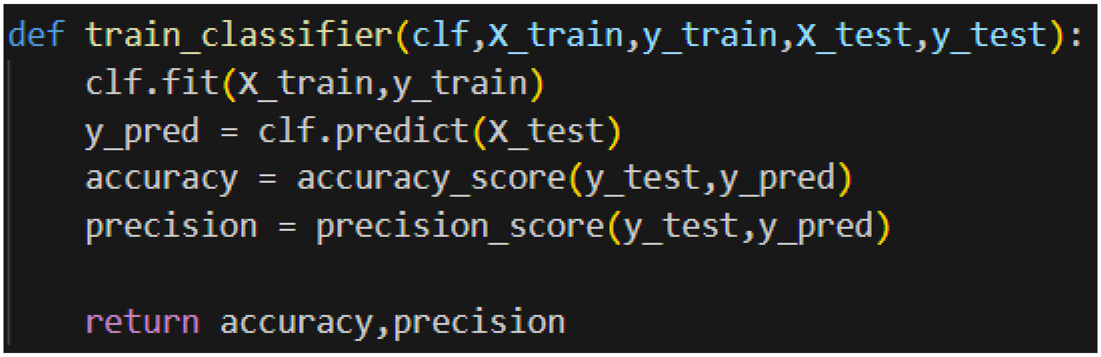
```
gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)
print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
```



```
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

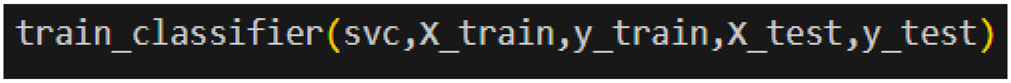


```
bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

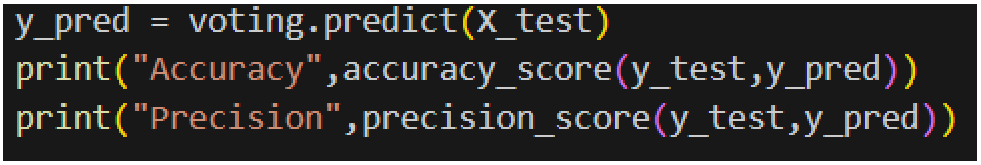


```
def train_classifier(clf,X_train,y_train,X_test,y_test):
    clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
    precision = precision_score(y_test,y_pred)

    return accuracy,precision
```



```
train_classifier(svc,X_train,y_train,X_test,y_test)
```

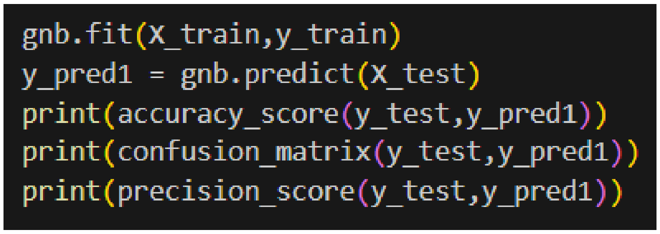
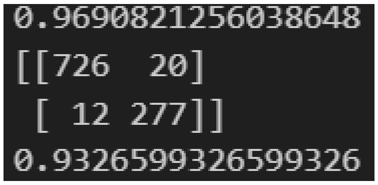


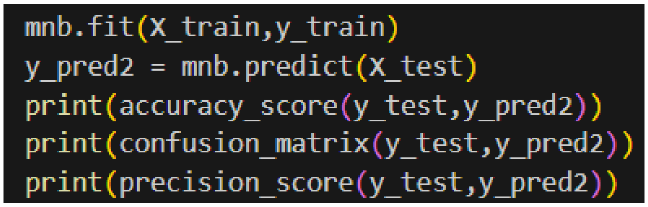
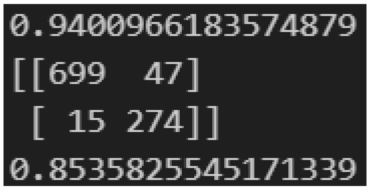
```
y_pred = voting.predict(X_test)
print("Accuracy",accuracy_score(y_test,y_pred))
print("Precision",precision_score(y_test,y_pred))
```

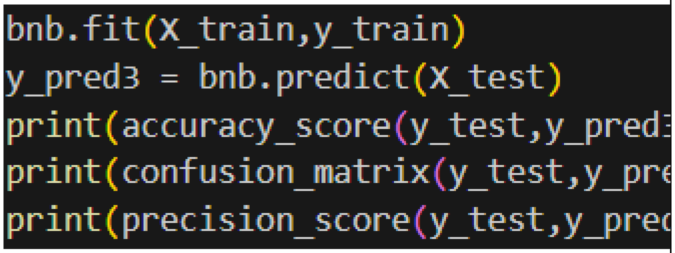
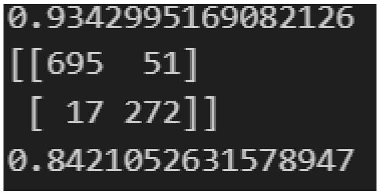
Paste the screenshot of the model training code

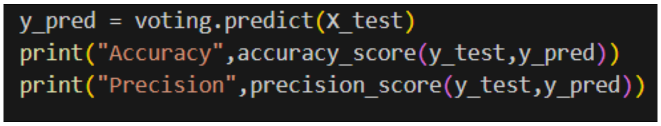
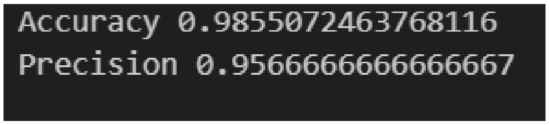
Model Validation and Evaluation Report (5 marks):

Model	**Summary**	**F1 SCORE**	**Training and Validation Performa**

Model 1	
	gnb.fit(X_train,y_train)
	y_pred1 = gnb.predict(X_test)
	print(accuracy_score(y_test,y_pred1))
	print(confusion_matrix(y_test,y_pred1))
	print(precision_score(y_test,y_pred1))
		0.93%	
			0.9690821256038648
			[[726 20]
			 [12 277]]
			0.9326599326599326

Model 2	
	mnb.fit(X_train,y_train)
	y_pred2 = mnb.predict(X_test)
	print(accuracy_score(y_test,y_pred2))
	print(confusion_matrix(y_test,y_pred2))
	print(precision_score(y_test,y_pred2))
		0.85%	
			0.9400966183574879
			[[699 47]
			 [15 274]]
			0.8535825545171339

Model 3	 bnb.fit(X_train,y_train) y_pred3 = bnb.predict(X_test) print(accuracy_score(y_test,y_pred print(confusion_matrix(y_test,y_pre print(precision_score(y_test,y_pred	0.84%	 0.9342995169082126 [[695 51] [17 272]] 0.8421052631578947

Model 4	
	y_pred = voting.predict(X_test)
	print("Accuracy",accuracy_score(y_test,y_pred))
	print("Precision",precision_score(y_test,y_pred))
		0.98%	
			Accuracy 0.9855072463768116
			Precision 0.9566666666666667