

Model Development Phase Template

Date	03 June 2024
Team ID	739981
Project Title	Harvesting Brilliance : A Taxanomic Tale of Pumpkin Seeds Varieties
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.

Model	Classification Report & Accuracy	Accuracy
-------	----------------------------------	----------

Initial Model Training Code:

```
#splitting into training and testing dataset
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=30)
```

Model Validation and Evaluation Report:

Random forest classifier

0.875

```
random_forest=RandomForestClassifier()
random_forest.fit(X_train,Y_train)
y_pred=random_forest.predict(X_test)
```

```
acc_rf=accuracy_score(y_test,y_pred)
c_rf=classification_report(y_test,y_pred)
```

```
print('Accuracy Score:',acc_rf)
print(c_rf)
```

```
Accuracy Score: 0.875115890543259
              precision    recall  f1-score   support

Coronavirus    0.86      0.91      0.88         257
Not a Coronavirus    0.90      0.84      0.87         248

accuracy              0.88
macro avg              0.88      0.87      0.87         407
weighted avg           0.88      0.88      0.87         407
```

Logistic regression

```
logistic_regression=LogisticRegression()
logistic_regression.fit(X_train,Y_train)
y_pred=logistic_regression.predict(X_test)
```

```
acc_lr=accuracy_score(y_test,y_pred)
c_lr=classification_report(y_test,y_pred)
```

```
print("Accuracy Score:",acc_lr)
print(c_lr)
```

```
Accuracy Score: 0.8611818181818182
```

	precision	recall	f1-score	support
Coronavirus	0.85	0.91	0.88	257
Not a Coronavirus	0.89	0.81	0.85	248
accuracy			0.87	407
macro avg	0.87	0.87	0.87	407
weighted avg	0.87	0.87	0.87	407

0.86

Decision Tree classifier

```
decision_tree_model=DecisionTreeClassifier()
decision_tree_model.fit(X_train,Y_train)
Y_pred=decision_tree_model.predict(X_test)

acc=accuracy_score(Y_test,Y_pred)
c_dt=classification_report(Y_test,Y_pred)

print("Accuracy Score:",acc_dt)
print(c_dt)
```

Accuracy Score:	0.82897384386587581			
	precision	recall	f1-score	support
Cerebellum	0.81	0.87	0.84	257
Urgip Sivarisi	0.85	0.78	0.82	240
accuracy			0.83	497
macro avg	0.83	0.83	0.83	497
weighted avg	0.83	0.83	0.83	497

0.82

Naïve Bayes

```
nb=MultinomialNB()
nb.fit(X_train,Y_train)
Y_prednb=nb.predict(X_test)

acc_nb=accuracy_score(Y_test,Y_prednb)
c_nb=classification_report(Y_test,Y_prednb)

print("Accuracy Score:",acc_nb)
print(c_nb)
```

Accuracy Score:	0.8148893368108966			
	precision	recall	f1-score	support
Cerebellum	0.75	0.95	0.84	257
Urgip Sivarisi	0.93	0.67	0.78	240
accuracy			0.81	497
macro avg	0.84	0.81	0.81	497
weighted avg	0.84	0.81	0.81	497

0.81

Gradient Boosting Classifier

0.66

Support vector classifier

```
support_vector=svm()
support_vector.fit(X_train,Y_train)
Y_pred=support_vector.predict(X_test)

acc_svm=accuracy_score(Y_test,Y_pred)
c_svm=classification_report(Y_test,Y_pred)

print("Accuracy Score:",acc_svm)
print(c_svm)
```

Accuracy Score:	0.668888882897385			
	precision	recall	f1-score	support
Cerebellum	0.64	0.88	0.71	257
Urgip Sivarisi	0.71	0.53	0.60	240
accuracy			0.67	497
macro avg	0.68	0.68	0.66	497
weighted avg	0.68	0.67	0.66	497

0.88

```
lbc=LinearSVC()
lbc.fit(X_train,Y_train)
Y_predlbc=lbc.predict(X_test)

acc_gbc=accuracy_score(Y_test,Y_pred)
c_gbc=classification_report(Y_test,Y_pred)

print("Accuracy Score:",acc_gbc)
print(c_gbc)
```

Accuracy Score:	0.881999786707566			
	precision	recall	f1-score	support
Cerebellum	0.86	0.92	0.89	257
Urgip Sivarisi	0.91	0.95	0.93	240
accuracy			0.88	497
macro avg	0.89	0.89	0.89	497
weighted avg	0.89	0.89	0.89	497

