SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

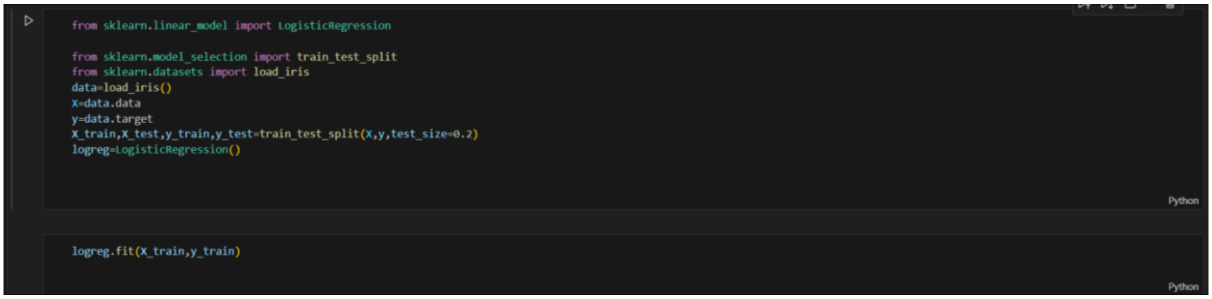
Model Development Phase Template

Date	14 July 2024
Team ID	739949
Project Title	Blood Donation Prediction
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

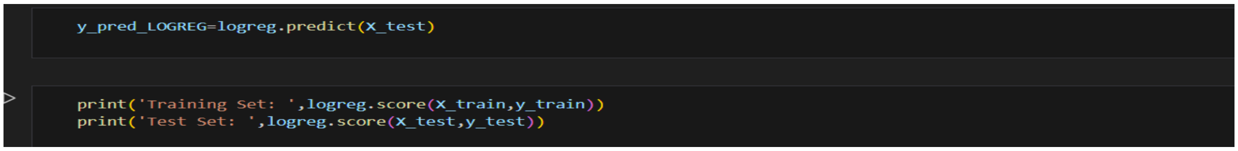
The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:



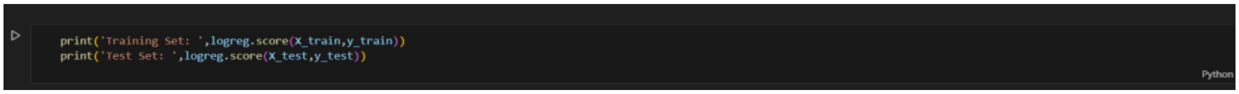
```
from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
data=load_iris()
X=data.data
y=data.target
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
logreg=LogisticRegression()
```

```
logreg.fit(X_train,y_train)
```

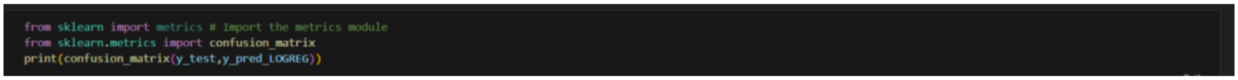


```
y_pred_LOGREG=logreg.predict(X_test)
```

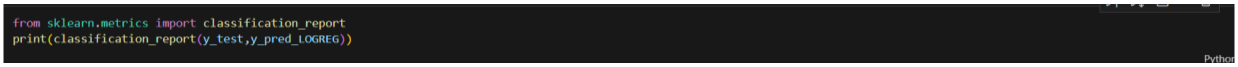
```
print('Training Set: ',logreg.score(X_train,y_train))
print('Test Set: ',logreg.score(X_test,y_test))
```



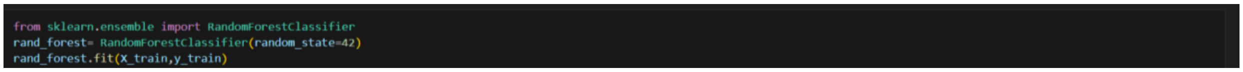
```
print('Training Set: ',logreg.score(X_train,y_train))
print('Test Set: ',logreg.score(X_test,y_test))
```



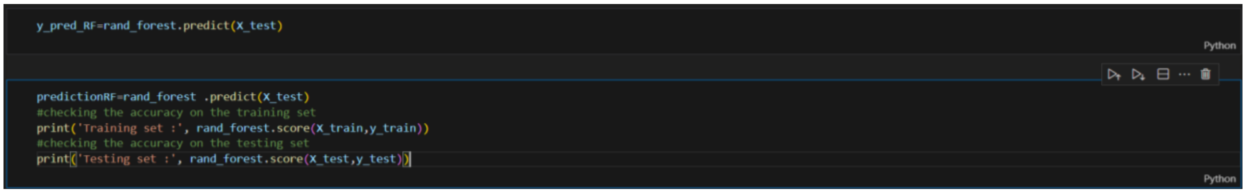
```
from sklearn import metrics # Import the metrics module
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred_LOGREG))
```



```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred_LOGREG))
```

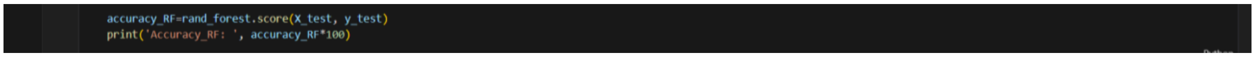


```
from sklearn.ensemble import RandomForestClassifier
rand_forest= RandomForestClassifier(random_state=42)
rand_forest.fit(X_train,y_train)
```

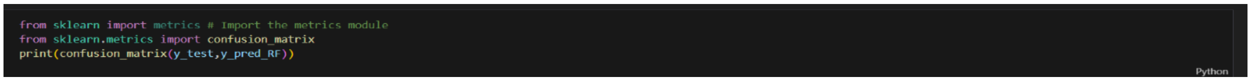


```
y_pred_RF=rand_forest.predict(X_test)
```

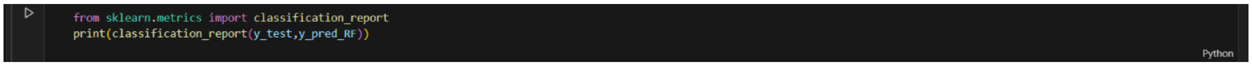
```
predictionRF=rand_forest .predict(X_test)
#checking the accuracy on the training set
print('Training set :', rand_forest.score(X_train,y_train))
#checking the accuracy on the testing set
print('Testing set :', rand_forest.score(X_test,y_test))
```



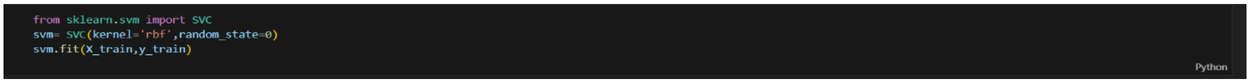
```
accuracy_RF=rand_forest.score(X_test, y_test)
print('Accuracy_RF: ', accuracy_RF*100)
```



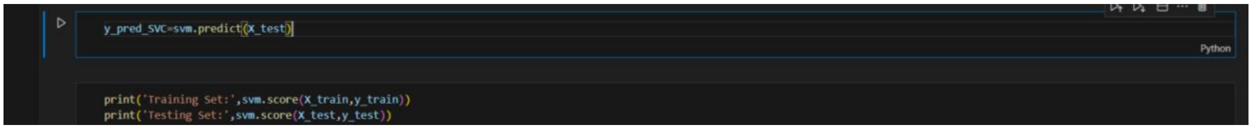
```
from sklearn import metrics # Import the metrics module
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred_RF))
```



```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred_RF))
```

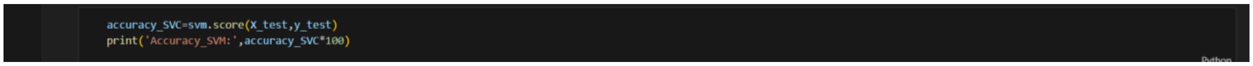


```
from sklearn.svm import SVC
svm= SVC(kernel='rbf',random_state=0)
svm.fit(X_train,y_train)
```

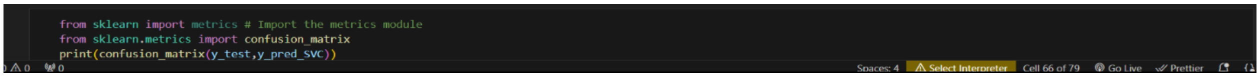


```
y_pred_SVC=svm.predict(X_test)
```

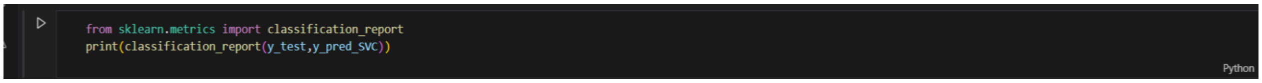
```
print('Training Set:',svm.score(X_train,y_train))
print('Testing Set:',svm.score(X_test,y_test))
```



```
accuracy_SVC=svm.score(X_test,y_test)
print('Accuracy_SVM:',accuracy_SVC*100)
```

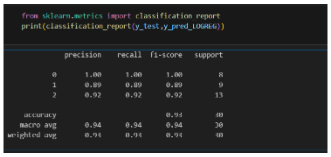
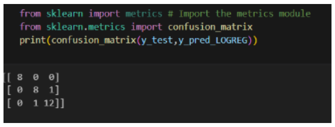
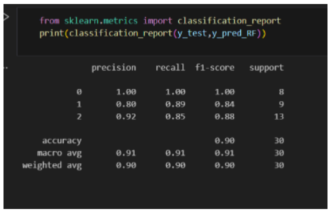
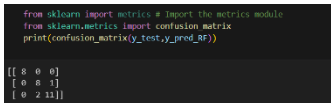


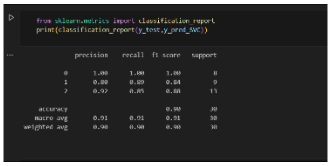
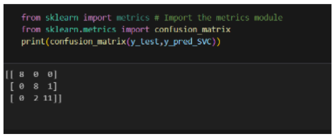
```
from sklearn import metrics # Import the metrics module
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred_SVC))
```



```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred_SVC))
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Logistic Regression	 from sklearn.metrics import classification_report print(classification_report(y_test,y_pred_LOGREG)) precision recall f1-score support 0 1.00 1.00 1.00 8 1 0.89 0.89 0.89 9 2 0.92 0.92 0.92 13 accuracy 0.93 30 macro avg 0.94 0.94 0.94 30 weighted avg 0.93 0.93 0.93 30	93.3	 from sklearn import metrics # Import the metrics module from sklearn.metrics import confusion_matrix print(confusion_matrix(y_test,y_pred_LOGREG)) [[8 0 0] [0 8 1] [0 1 12]]
Random Forest	 from sklearn.metrics import classification_report print(classification_report(y_test,y_pred_RF)) precision recall f1-score support 0 1.00 1.00 1.00 8 1 0.80 0.89 0.84 9 2 0.92 0.85 0.88 13 accuracy 0.90 30 macro avg 0.91 0.91 0.91 30 weighted avg 0.90 0.90 0.90 30	90.0	 from sklearn import metrics # Import the metrics module from sklearn.metrics import confusion_matrix print(confusion_matrix(y_test,y_pred_RF)) [[8 0 0] [0 8 1] [0 2 11]]

Support Vector Machine	 from sklearn.metrics import classification_report print(classification_report(y_test,y_pred_SVC)) precision recall f1-score support 0 1.00 1.00 1.00 8 1 0.88 0.89 0.84 9 2 0.92 0.85 0.88 13 accuracy 0.90 30 macro avg 0.91 0.91 0.91 30 weighted avg 0.90 0.90 0.90 30	**90.0**	 from sklearn import metrics # Import the metrics module from sklearn.metrics import confusion_matrix print(confusion_matrix(y_test,y_pred_SVC)) [[8 0 0] [0 8 1] [0 2 11]]