Model Development Phase Template

Date	15 July 2024
Team ID	740685
Project Title	**SDSS galaxy classification using Machine Learning**
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

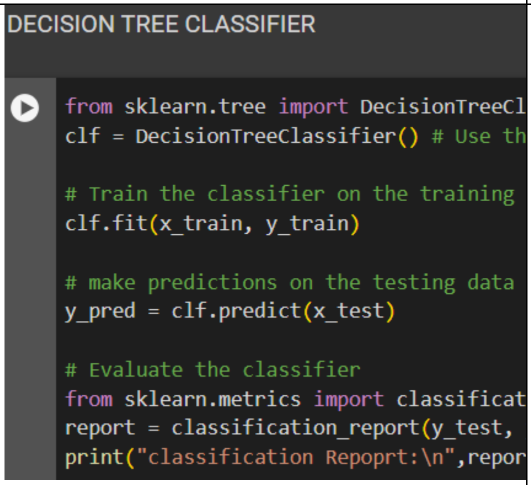
The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

Paste the screenshot of the model training code

Model Validation and Evaluation Report:

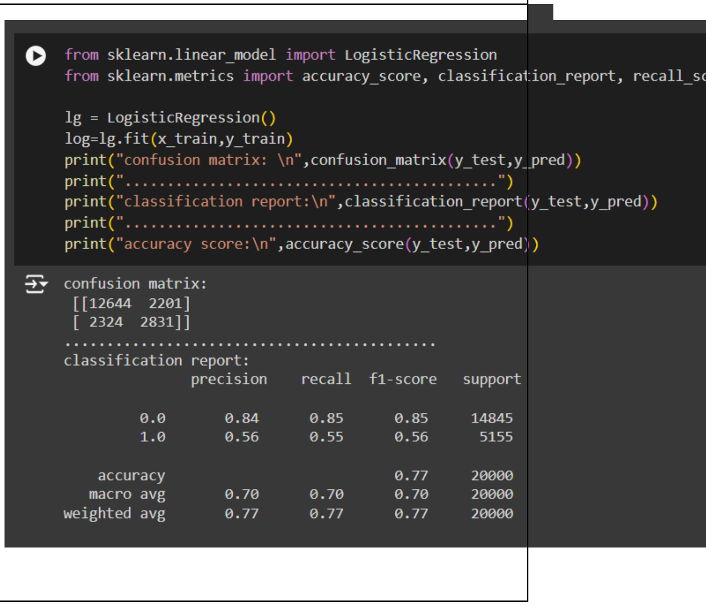
<table>
<tr><th>Model</th><th>Classification Report</th><th>Accuracy</th><th>Confusion Matrix</th></tr>
<tr>
<td>Decision Tree</td>
<td>



DECISION TREE CLASSIFIER

```
from sklearn.tree import DecisionTreeCl
clf = DecisionTreeClassifier() # Use th

# Train the classifier on the training
clf.fit(x_train, y_train)

# make predictions on the testing data
y_pred = clf.predict(x_test)

# Evaluate the classifier
from sklearn.metrics import classificat
report = classification_report(y_test,
print("classification Repoprt:\n",repor
```

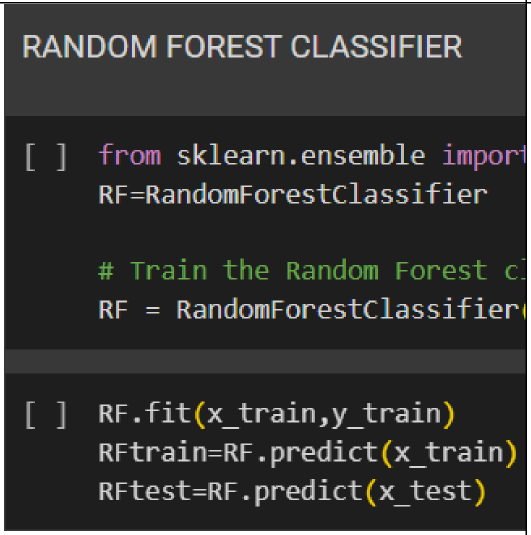
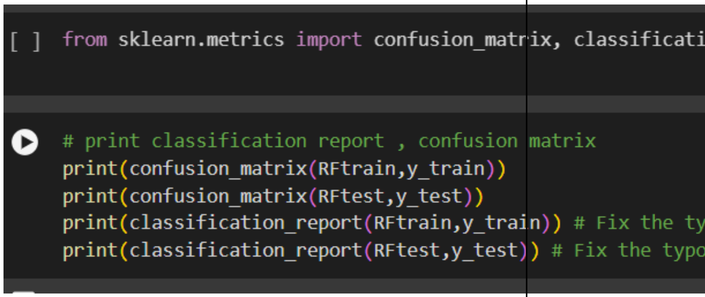
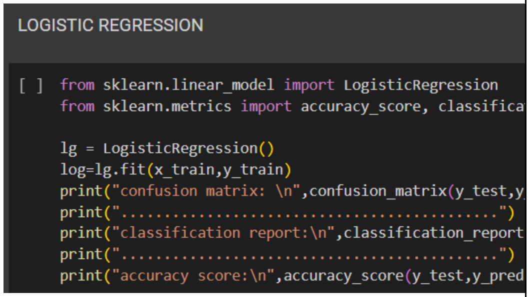
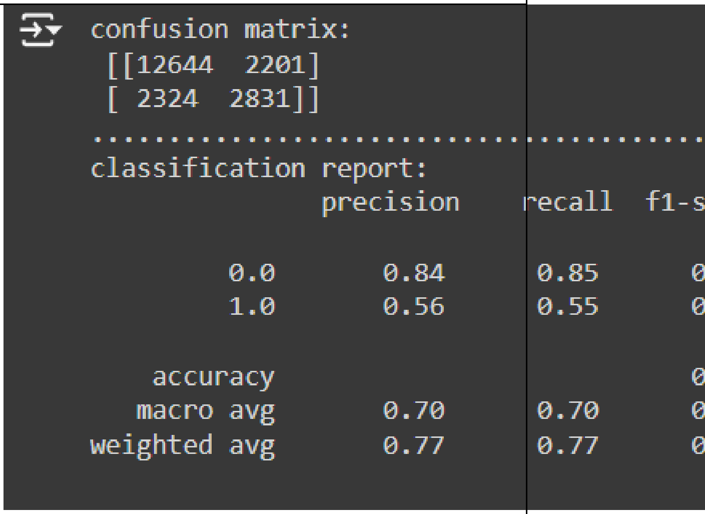
</td>
<td>0.77</td>
<td>



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, recall_sc

lg = LogisticRegression()
log=lg.fit(x_train,y_train)
print("confusion matrix: \n",confusion_matrix(y_test,y_pred))
print("..........................................")
print("classification report:\n",classification_report(y_test,y_pred))
print("..........................................")
print("accuracy score:\n",accuracy_score(y_test,y_pred))
```

```
confusion matrix:
 [[12644  2201]
 [ 2324  2831]]
............................................
classification report:
               precision    recall  f1-score   support

         0.0       0.84      0.85      0.85     14845
         1.0       0.56      0.55      0.56      5155

    accuracy                           0.77     20000
   macro avg       0.70      0.70      0.70     20000
weighted avg       0.77      0.77      0.77     20000
```

</td>
</tr>
</table>

RandomForest		1.00	
Logistic Regression		0.77	

RANDOM FOREST CLASSIFIER

```
[ ]  from sklearn.ensemble import
     RF=RandomForestClassifier

     # Train the Random Forest cl
     RF = RandomForestClassifier(
```

```
[ ]  RF.fit(x_train,y_train)
     RFtrain=RF.predict(x_train)
     RFtest=RF.predict(x_test)
```

```
[ ]  from sklearn.metrics import confusion_matrix, classificati
```

```
     # print classification report , confusion matrix
     print(confusion_matrix(RFtrain,y_train))
     print(confusion_matrix(RFtest,y_test))
     print(classification_report(RFtrain,y_train)) # Fix the ty
     print(classification_report(RFtest,y_test)) # Fix the typo
```

LOGISTIC REGRESSION

```
[ ]  from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import accuracy_score, classifica

     lg = LogisticRegression()
     log=lg.fit(x_train,y_train)
     print("confusion matrix: \n",confusion_matrix(y_test,y
     print(".......................................")
     print("classification report:\n",classification_report
     print(".......................................")
     print("accuracy score:\n",accuracy_score(y_test,y_pred
```

```
confusion matrix:
 [[12644  2201]
 [ 2324  2831]]
.................................
classification report:
              precision    recall  f1-s

         0.0       0.84      0.85      0
         1.0       0.56      0.55      0

    accuracy                           0
   macro avg       0.70      0.70      0
weighted avg       0.77      0.77      0
```