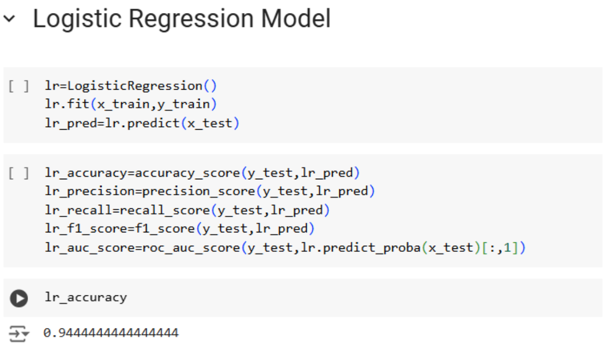
Model Development Phase Template

Date	15 July 2024
Team ID	739682
Project Title	Space X Falcon 9 First Stage Landing Success Predictor
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:



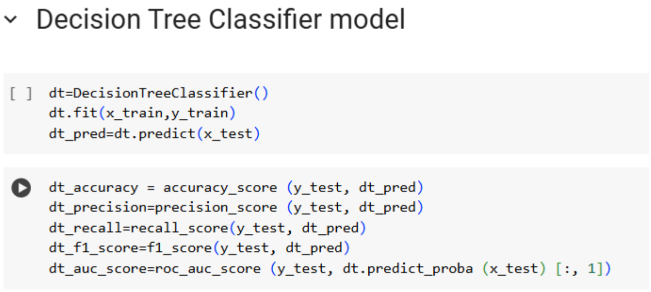
Logistic Regression Model

```
lr=LogisticRegression()
lr.fit(x_train,y_train)
lr_pred=lr.predict(x_test)
```

```
lr_accuracy=accuracy_score(y_test,lr_pred)
lr_precision=precision_score(y_test,lr_pred)
lr_recall=recall_score(y_test,lr_pred)
lr_f1_score=f1_score(y_test,lr_pred)
lr_auc_score=roc_auc_score(y_test,lr.predict_proba(x_test)[:,1])
```

```
lr_accuracy
```

```
0.9444444444444444
```

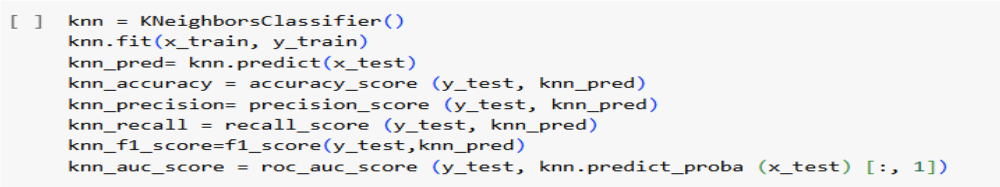


Decision Tree Classifier model

```
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
dt_pred=dt.predict(x_test)
```

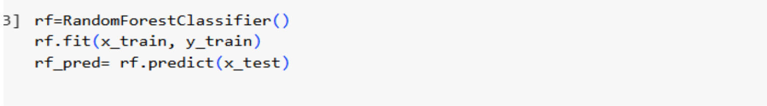
```
dt_accuracy = accuracy_score (y_test, dt_pred)
dt_precision=precision_score (y_test, dt_pred)
dt_recall=recall_score(y_test, dt_pred)
dt_f1_score=f1_score(y_test, dt_pred)
dt_auc_score=roc_auc_score (y_test, dt.predict_proba (x_test) [:, 1])
```

KNN Classifier model

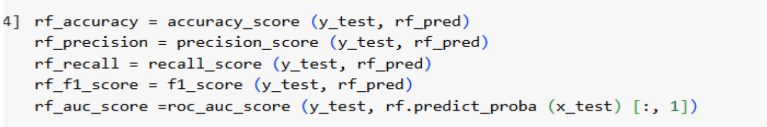


```
[ ]  knn = KNeighborsClassifier()
     knn.fit(x_train, y_train)
     knn_pred= knn.predict(x_test)
     knn_accuracy = accuracy_score (y_test, knn_pred)
     knn_precision= precision_score (y_test, knn_pred)
     knn_recall = recall_score (y_test, knn_pred)
     knn_f1_score=f1_score(y_test,knn_pred)
     knn_auc_score = roc_auc_score (y_test, knn.predict_proba (x_test) [:, 1])
```

Random Forest Model

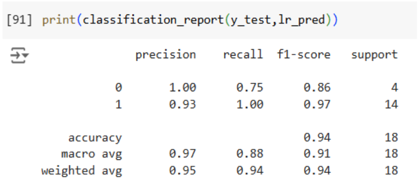
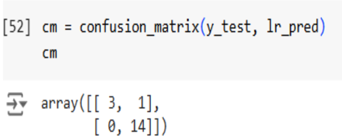
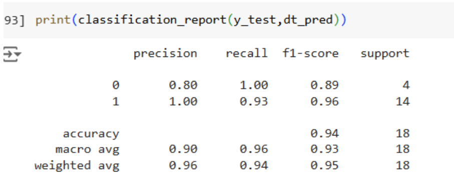
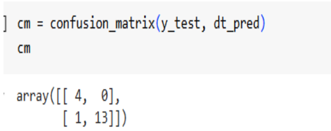


```
3]  rf=RandomForestClassifier()
    rf.fit(x_train, y_train)
    rf_pred= rf.predict(x_test)
```

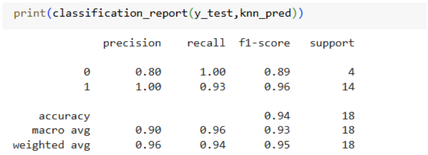


```
4]  rf_accuracy = accuracy_score (y_test, rf_pred)
    rf_precision = precision_score (y_test, rf_pred)
    rf_recall = recall_score (y_test, rf_pred)
    rf_f1_score = f1_score (y_test, rf_pred)
    rf_auc_score =roc_auc_score (y_test, rf.predict_proba (x_test) [:, 1])
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Logistic Regression	 [91] print(classification_report(y_test,lr_pred)) precision recall f1-score support 0 1.00 0.75 0.86 4 1 0.93 1.00 0.97 14 accuracy 0.94 18 macro avg 0.97 0.88 0.91 18 weighted avg 0.95 0.94 0.94 18	96%	 [52] cm = confusion_matrix(y_test, lr_pred) cm array([[3, 1], [0, 14]])
Decision Tree	 93] print(classification_report(y_test,dt_pred)) precision recall f1-score support 0 0.80 1.00 0.89 4 1 1.00 0.93 0.96 14 accuracy 0.94 18 macro avg 0.90 0.96 0.93 18 weighted avg 0.96 0.94 0.95 18	96%	] cm = confusion_matrix(y_test, dt_pred) cm array([[4, 0], [1, 13]])

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



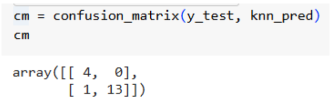
```
print(classification_report(y_test,knn_pred))

              precision    recall  f1-score   support

           0       0.80      1.00      0.89         4
           1       1.00      0.93      0.96        14

    accuracy                           0.94        18
   macro avg       0.90      0.96      0.93        18
weighted avg       0.96      0.94      0.95        18
```

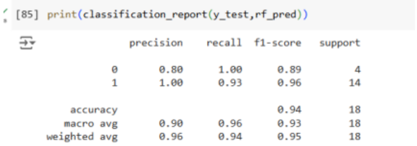
</td>
<td>96%</td>
<td>



```
cm = confusion_matrix(y_test, knn_pred)
cm

array([[ 4,  0],
       [ 1, 13]])
```

</td>
</tr>
<tr>
<td>Random
Forest</td>
<td>



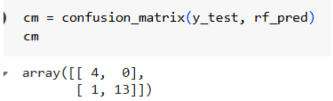
```
[85] print(classification_report(y_test,rf_pred))

              precision    recall  f1-score   support

           0       0.80      1.00      0.89         4
           1       1.00      0.93      0.96        14

    accuracy                           0.94        18
   macro avg       0.90      0.96      0.93        18
weighted avg       0.96      0.94      0.95        18
```

</td>
<td>96%</td>
<td>



```
cm = confusion_matrix(y_test, rf_pred)
cm

array([[ 4,  0],
       [ 1, 13]])
```

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