Model Development Phase Template

Date	23 September 2024
Team ID	LTVIP2024TMID24998
Project Title	Flight Delays Prediction 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 final submission through screenshots. The model validation and evaluation report will summarize the performance of the flight delay using metrics such as **accuracy**, **precision**, **recall**, **F1-score** through respective screenshots.

Initial Model Training Code:

1. **Linear Regression:**

```python
#Linear Regression
from sklearn.linear_model import LinearRegression
LinR = LinearRegression()
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

fitResult = LinR.fit(X_train_sc,y_train)
y_pred = fitResult.predict(X_test_sc)
print ('MAE:' ,  mean_absolute_error(y_test, y_pred))
print ('MSE:' , mean_squared_error(y_test, y_pred))
print('RMSE:' , np.sqrt(mean_squared_error(y_test, y_pred)))
print ('R2:' , r2_score(y_test, y_pred))
```

Python

2. **RandomForest Regression:**

```python
# RandomForest Regression
from sklearn.ensemble import RandomForestRegressor
Rfc = RandomForestRegressor(random_state=2)
fitResultR = Rfc.fit(X_train_sc,y_train)
predictedValues = fitResultR.predict(X_test_sc)
print ('MAE:' ,  mean_absolute_error(y_test, predictedValues))
print ('MSE:' , mean_squared_error(y_test, predictedValues))
print('RMSE:' , np.sqrt(mean_squared_error(y_test, predictedValues)))
print ('R2:' , r2_score(y_test, predictedValues))
```

3. Decision Tree Regressor:

```
# Decision Tree
from sklearn.tree import DecisionTreeRegressor
Dtc = DecisionTreeRegressor(random_state = 2)

fitResultdtc = Dtc.fit(X_train_sc,y_train)
predictedValues = fitResultdtc.predict(X_test_sc)
print ('MAE:' ,  mean_absolute_error(y_test, predictedValues))
print ('MSE:' , mean_squared_error(y_test, predictedValues))
print('RMSE:' , np.sqrt(mean_squared_error(y_test, predictedValues)))
print ('R2:' , r2_score(y_test, predictedValues))
```

4. K nearest neighbour

```
# K nearest neighbours
y_pred=objClassifier.predict(X_test_sc)

#Making the confussion matarix

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)

score=objClassifier.score(X_test,y_test)
```

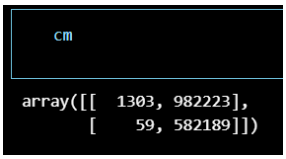
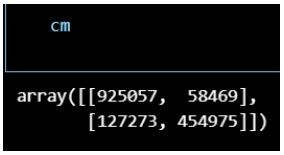
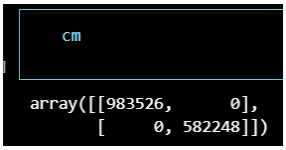
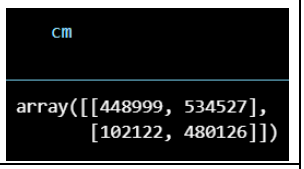
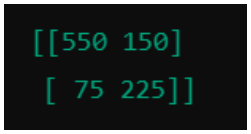
5. Logistic Regression:

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train_sc, y_train)
# Predicting the Test set results
y_pred = classifier.predict(X_test_sc)

# Making the Confusion Matrix
score = classifier.score(X_test_sc,y_test)
cm = confusion_matrix(y_test, y_pred)
```

6. Naïve Bayes :

```
# Naive Bayes
# Predicting the Test set results
y_pred = objclassifierGNB.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
score = objclassifierGNB.score(X_test_sc,y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Decision Tree Regressor	Score:0.27 Precision Score : 0.66 Recall Score : 0.50	F1 score : 0.98	
K nearest neighbour	Score:0.37 Precision Score : 0.88 Recall Score : 0.86	F1 score : 0.86	
Logistic Regression	Score:1.0 Precision Score : 1.0 Recall Score : 1.0	F1 score:1	
Naïve Bayes	Score: 0.83 Precision Score : 0.64 Recall Score : 0.64	F1 score : 0.59	
Linear Regression	Precision Score: 0.83 Recall Score: 1.0 Score: 0.91	F1 Score: 0.91	
Random Forest Regression	Precision Score: 0.56 Recall Score: 0.63 Score: 0.59	F1 Score: 0.59