Model Optimization and Tuning Phase Template

Date	15 July 2024
Team ID	739791
Project Title	Flight Delay Prediction using Machine Learning.
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Random Forest Classifier	-----	

Logistic Regression	------	```python
lr = LogisticRegression()
lr.fit(x_train,y_train)

y_test_predict2 = lr.predict(x_test)
test_accuracy = accuracy_score(y_test,y_test_predict2)
test_accuracy
```
Python
c:\Users\indhu\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\utils\validation.py:1339: DataConversionWarning: A column-vector y was
y = column_or_1d(y, warn=True)
c:\Users\indhu\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\linear_model\_logistic.py:469: ConvergenceWarning: lbfgs failed to con
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(

0.9182022471910113 |
| Decision Tree Classifier | ------- | ```python
dtc = DecisionTreeClassifier()
dtc.fit(x_train,y_train)

y_test_predict3 = dtc.predict(x_test)
test_accuracy = accuracy_score(y_test,y_test_predict3)
test_accuracy
```

0.8606741573033708 |
| Extra Tree Classifier | ------ | ```python
etc = ExtraTreesClassifier()
etc.fit(x_train,y_train)

y_test_predict4 = etc.predict(x_test)
test_accuracy = accuracy_score(y_test,y_test_predict4)
test_accuracy
```
c:\Users\indhu\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:1473
  return fit_method(estimator, *args, **kwargs)

0.9087640449438202 |

**Performance Metrics Comparison Report (2 Marks):**

| Model | Optimized Metric |
|---|---|
| | |

| | |
|---|---|
| Random Forest Classifier | ```
print(classification_report(y_test,y_test_predict1))

              precision    recall  f1-score   support

         0.0       0.94      0.96      0.95      1932
         1.0       0.71      0.59      0.65       293

    accuracy                           0.92      2225
   macro avg       0.83      0.78      0.80      2225
weighted avg       0.91      0.92      0.91      2225



confusion_matrix(y_test, y_test_predict1)

array([[1863,   69],
       [ 120,  173]])
``` |
| Linear Regression | ```
y_train_predict2 = rfc.predict(x_train)
train_accuracy = accuracy_score(y_train,y_train_predict2)
train_accuracy
print(classification_report(y_test,y_test_predict2))

              precision    recall  f1-score   support

         0.0       0.95      0.96      0.95      1932
         1.0       0.70      0.65      0.68       293

    accuracy                           0.92      2225
   macro avg       0.83      0.81      0.82      2225
weighted avg       0.92      0.92      0.92      2225


confusion_matrix(y_test, y_test_predict2)

array([[1852,   80],
       [ 102,  191]])
``` |
| Decision Tree Classifier | ```
y_train_predict3 = dtc.predict(x_train)
train_accuracy = accuracy_score(y_train,y_train_predict3)
train_accuracy
print(classification_report(y_test,y_test_predict3))

              precision    recall  f1-score   support

         0.0       0.92      0.92      0.92      1932
         1.0       0.47      0.48      0.48       293

    accuracy                           0.86      2225
   macro avg       0.70      0.70      0.70      2225
weighted avg       0.86      0.86      0.86      2225



confusion_matrix(y_test, y_test_predict3)

array([[1773,  159],
       [ 151,  142]])
``` |

| Extra Tree Classifier | ```
y_train_predict4 = etc.predict(x_train)
train_accuracy = accuracy_score(y_train,y_train_predict4)
train_accuracy
print(classification_report(y_test,y_test_predict4))

              precision    recall  f1-score   support

         0.0       0.94      0.96      0.95      1932
         1.0       0.69      0.56      0.62       293

    accuracy                           0.91      2225
   macro avg       0.81      0.76      0.78      2225
weighted avg       0.90      0.91      0.90      2225


confusion_matrix(y_test, y_test_predict4)

array([[1857,   75],
       [ 128,  165]])
``` |
|---|---|

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest Classifier	The Random Forest model was selected for its superior performance, exhibiting high accuracy during train and test. Its often more accurate than decision tree it builds multiple tree and averages their predictions,reducing the risk of overfitting.It can model non-linear relationships better than Linear Regression.Effective in detecting anomalies in datasets,useful in fraud detection and network security.