Model Optimization and Tuning Phase Template

Date	15 July 2024
Team ID	739886
Project Title	Telecom Customer Churn Prediction
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
-	-	-

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric

SVM

```
[54]: print(classification_report(svm_pred,y_test))
              precision    recall  f1-score   support

           0       1.00      0.80      0.89      2000
           1       0.00      0.00      0.00         0

    accuracy                           0.80      2000
   macro avg       0.50      0.40      0.44      2000
weighted avg       1.00      0.80      0.89      2000

[55]: confusion_matrix(svm_pred,y_test)

[55]: array([[1595,  405],
             [   0,    0]], dtype=int64)
```

Logistic Regression

```
[57]:
print(classification_report(model.predict(x_test),y_test))
              precision    recall  f1-score   support

           0       0.97      0.82      0.89      1875
           1       0.18      0.58      0.27       125

    accuracy                           0.81      2000
   macro avg       0.57      0.70      0.58      2000
weighted avg       0.92      0.81      0.85      2000

[58]:
confusion_matrix(model.predict(x_test),y_test)
[58]:
array([[1542,  333],
       [  53,   72]], dtype=int64)
[59]:
```

Decision Tree

```
[61]:
print(classification_report(pred,y_test))
              precision    recall  f1-score   support

           0       0.85      0.87      0.86      1562
           1       0.51      0.47      0.49       438

    accuracy                           0.78      2000
   macro avg       0.68      0.67      0.67      2000
weighted avg       0.78      0.78      0.78      2000

[62]:
confusion_matrix(pred,y_test)
[62]:
array([[1362,  200],
       [ 233,  205]], dtype=int64)
```

Random Forest

```
[66]:
rfc_con=confusion_matrix(pred,y_test)


rfc_con
[66]:
array([[1528,  205],
       [  67,  200]], dtype=int64)
[67]:
print(classification_report(pred,y_test))
              precision    recall  f1-score   support

           0       0.96      0.88      0.92      1733
           1       0.49      0.75      0.60       267

    accuracy                           0.86      2000
   macro avg       0.73      0.82      0.76      2000
weighted avg       0.90      0.86      0.88      2000
```

KNeighbors Classifier

```
[76]:
print(classification_report(knn.predict(x_test),y_test))
              precision    recall  f1-score   support

           0       0.94      0.87      0.90      1728
           1       0.43      0.64      0.51       272

    accuracy                           0.83      2000
   macro avg       0.68      0.75      0.71      2000
weighted avg       0.87      0.83      0.85      2000

[77]:
knn_con=confusion_matrix(knn.predict(x_test),y_test)
knn_con
[77]:
array([[1496,  232],
       [  99,  173]], dtype=int64)
```

Naïve bayes

```
[79]:
print(classification_report(gnb.predict(x_test),y_test))
              precision    recall  f1-score   support

           0       0.97      0.84      0.90      1846
           1       0.26      0.69      0.38       154

    accuracy                           0.83      2000
   macro avg       0.62      0.77      0.64      2000
weighted avg       0.92      0.83      0.86      2000

[80]:
nb_con=confusion_matrix(gnb.predict(x_test),y_test)
nb_con
[80]:
array([[1548,  298],
       [  47,  107]], dtype=int64)
```

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest Classifier	Random Forest is favored for telecom churn prediction due to its high accuracy with complex, feature-rich datasets. It excels in capturing non-linear relationships and interactions while mitigating overfitting through ensemble learning. Feature importance ranking aids in identifying key predictors, and its robustness against data imbalance makes it ideal for detecting churn patterns in telecom customer data.