Model Optimization and Tuning Phase Report

Date	21 June 2024
Team ID	740029
Project Title	Startup Prophet
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.

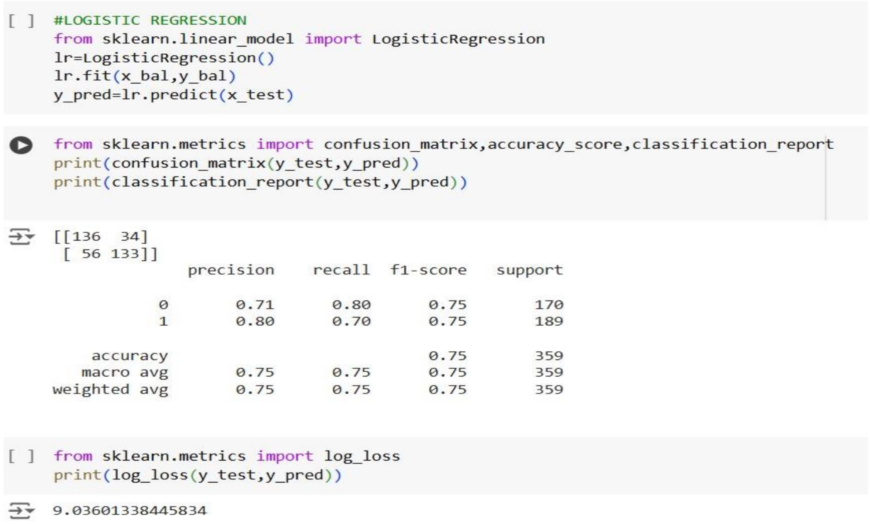
Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Logistic Regression Model	-	-

Support Vector Machine	-	-

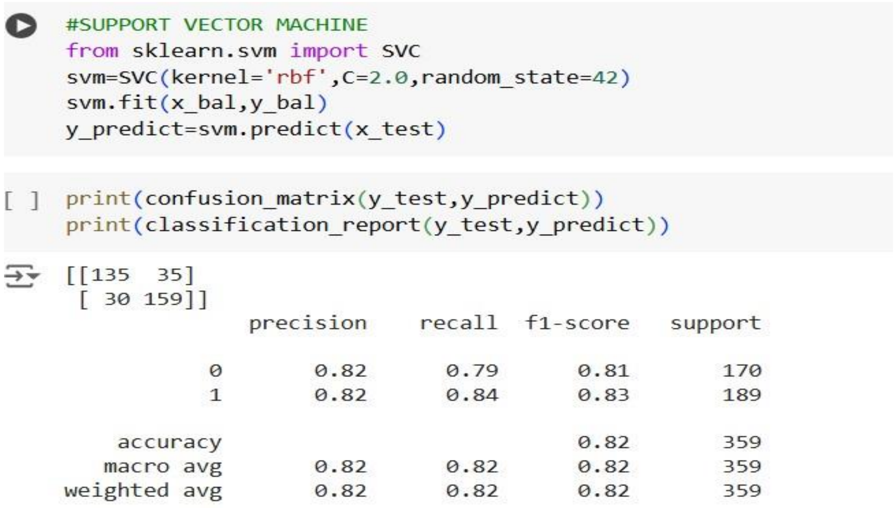
Random Forest Model	-	-

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Logistic Regression Model	

```
[ ] #LOGISTIC REGRESSION
    from sklearn.linear_model import LogisticRegression
    lr=LogisticRegression()
    lr.fit(x_bal,y_bal)
    y_pred=lr.predict(x_test)

    from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
    print(confusion_matrix(y_test,y_pred))
    print(classification_report(y_test,y_pred))

    [[136  34]
     [ 56 133]]
                  precision    recall  f1-score   support

               0       0.71      0.80      0.75       170
               1       0.80      0.70      0.75       189

        accuracy                           0.75       359
       macro avg       0.75      0.75      0.75       359
    weighted avg       0.75      0.75      0.75       359

[ ] from sklearn.metrics import log_loss
    print(log_loss(y_test,y_pred))

    9.03601338445834
```

Support Vector Machine

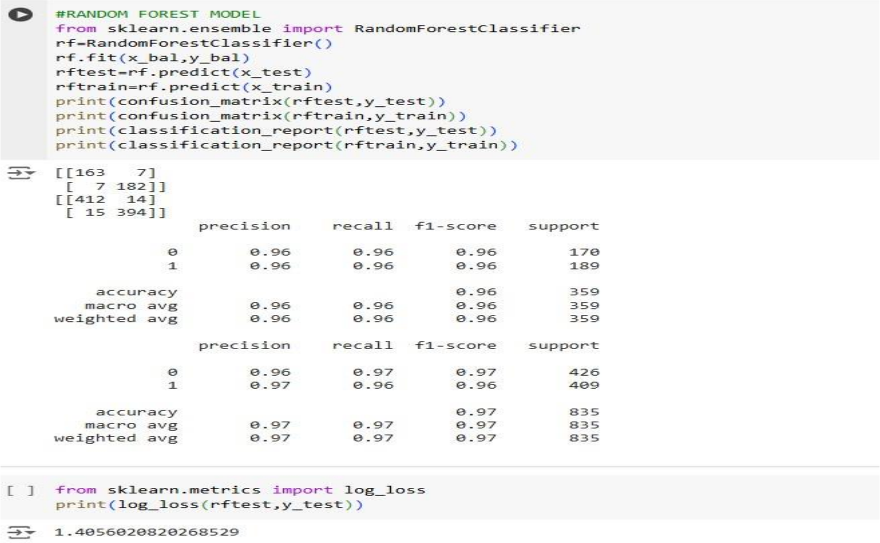


```
#SUPPORT VECTOR MACHINE
from sklearn.svm import SVC
svm=SVC(kernel='rbf',C=2.0,random_state=42)
svm.fit(x_bal,y_bal)
y_predict=svm.predict(x_test)
```

```
[ ]  print(confusion_matrix(y_test,y_predict))
     print(classification_report(y_test,y_predict))
```

```
[[135  35]
 [ 30 159]]
              precision    recall  f1-score   support

           0       0.82      0.79      0.81       170
           1       0.82      0.84      0.83       189

    accuracy                           0.82       359
   macro avg       0.82      0.82      0.82       359
weighted avg       0.82      0.82      0.82       359
```

Random Forest Model



```
#RANDOM FOREST MODEL
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(x_bal,y_bal)
rftest=rf.predict(x_test)
rftrain=rf.predict(x_train)
print(confusion_matrix(rftest,y_test))
print(confusion_matrix(rftrain,y_train))
print(classification_report(rftest,y_test))
print(classification_report(rftrain,y_train))
```

```
[[163   7]
 [  7 182]]
[[412  14]
 [ 15 394]]
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       170
           1       0.96      0.96      0.96       189

    accuracy                           0.96       359
   macro avg       0.96      0.96      0.96       359
weighted avg       0.96      0.96      0.96       359

              precision    recall  f1-score   support

           0       0.96      0.97      0.97       426
           1       0.97      0.96      0.96       409

    accuracy                           0.97       835
   macro avg       0.97      0.97      0.97       835
weighted avg       0.97      0.97      0.97       835
```

```
[ ]  from sklearn.metrics import log_loss
     print(log_loss(rftest,y_test))
```

```
1.4056020820268529
```

Final Model Selection Justification (2 Marks):

Final Model	Reasoning

Random Forest Model	The Random Forest Model was selected for its superior performance, exhibiting high accuracy. Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model.