Model Optimization and Tuning Phase Template

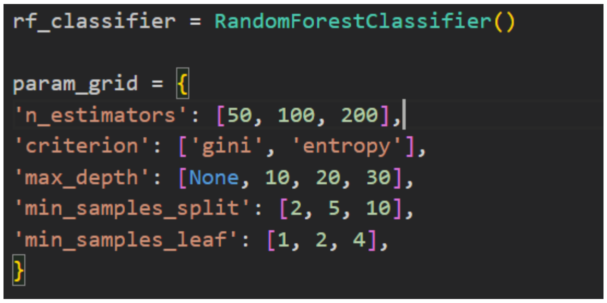
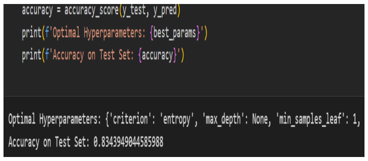
Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	10 Marks

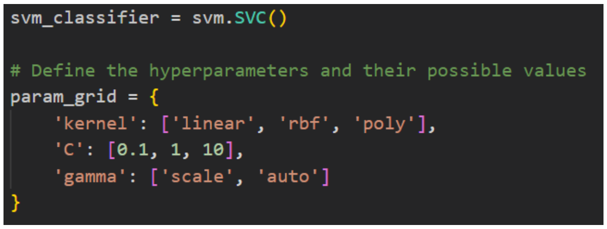
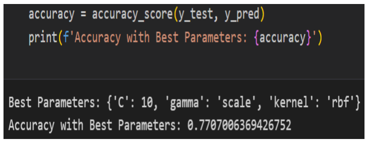
Model Optimization and Tuning Phase

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.

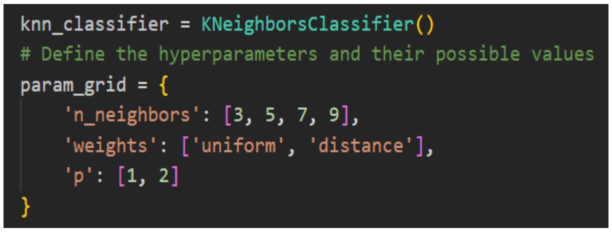
Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

Random Forest	
	`rf_classifier = RandomForestClassifier()`
	
	`param_grid = {`
	`'n_estimators': [50, 100, 200],`
	`'criterion': ['gini', 'entropy'],`
	`'max_depth': [None, 10, 20, 30],`
	`'min_samples_split': [2, 5, 10],`
	`'min_samples_leaf': [1, 2, 4],`
	`}`
		
		`accuracy = accuracy_score(y_test, y_pred)`
		`print(f'Optimal Hyperparameters: {best_params}')`
		`print(f'Accuracy on Test Set: {accuracy}')`
		
		`Optimal Hyperparameters: {'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1,`
		`Accuracy on Test Set: 0.8343949044585988`

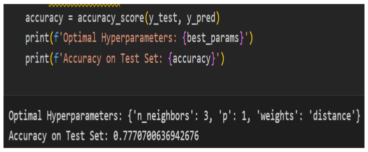
SVM	
	`svm_classifier = svm.SVC()`
	
	`# Define the hyperparameters and their possible values`
	`param_grid = {`
	`    'kernel': ['linear', 'rbf', 'poly'],`
	`    'C': [0.1, 1, 10],`
	`    'gamma': ['scale', 'auto']`
	`}`
		
		`accuracy = accuracy_score(y_test, y_pred)`
		`print(f'Accuracy with Best Parameters: {accuracy}')`
		
		`Best Parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}`
		`Accuracy with Best Parameters: 0.7707006369426752`

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



```
knn_classifier = KNeighborsClassifier()
# Define the hyperparameters and their possible values
param_grid = {
    'n_neighbors': [3, 5, 7, 9],
    'weights': ['uniform', 'distance'],
    'p': [1, 2]
}
```

</td>
<td>

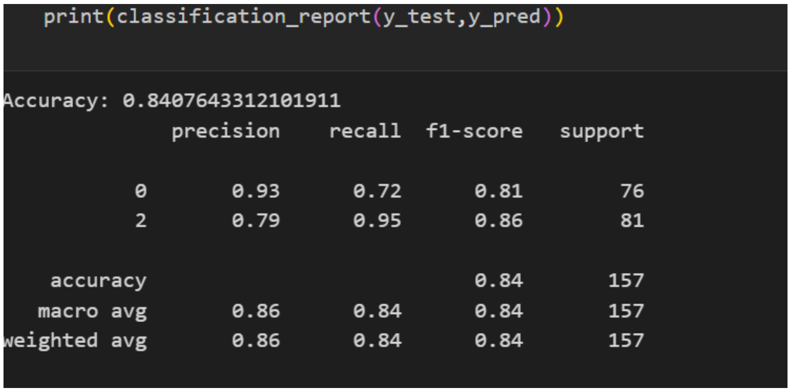


```
accuracy = accuracy_score(y_test, y_pred)
print(f'Optimal Hyperparameters: {best_params}')
print(f'Accuracy on Test Set: {accuracy}')

Optimal Hyperparameters: {'n_neighbors': 3, 'p': 1, 'weights': 'distance'}
Accuracy on Test Set: 0.7770700636942676
```

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

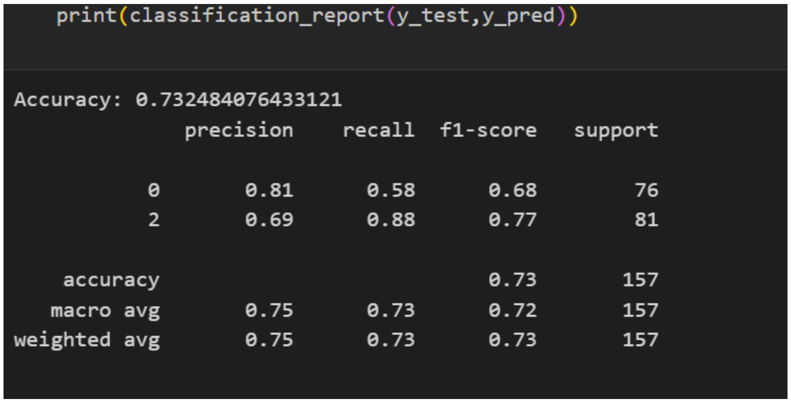
Performance Metrics Comparison Report (2 Marks):

<table>
<tr>
<th>Model</th>
<th>Baseline Metric, optimal metrics</th>
</tr>
<tr>
<td>Random Forest</td>
<td>



```
   print(classification_report(y_test,y_pred))

Accuracy: 0.8407643312101911
              precision    recall  f1-score   support

           0       0.93      0.72      0.81        76
           2       0.79      0.95      0.86        81

    accuracy                           0.84       157
   macro avg       0.86      0.84      0.84       157
weighted avg       0.86      0.84      0.84       157
```

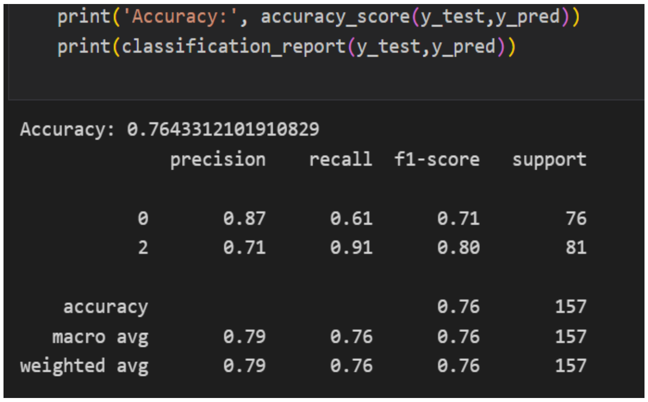
</td>
</tr>
<tr>
<td>SVM</td>
<td>



```
    print(classification_report(y_test,y_pred))

Accuracy: 0.732484076433121
              precision    recall  f1-score   support

           0       0.81      0.58      0.68        76
           2       0.69      0.88      0.77        81

    accuracy                           0.73       157
   macro avg       0.75      0.73      0.72       157
weighted avg       0.75      0.73      0.73       157
```

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

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

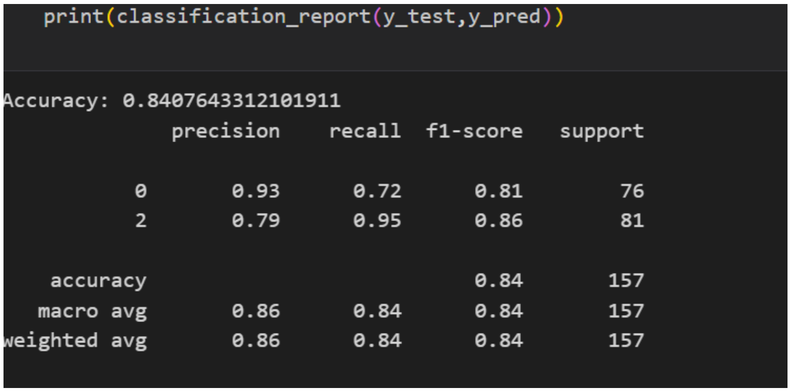


```
print('Accuracy:', accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))

Accuracy: 0.7643312101910829
              precision    recall  f1-score   support

           0       0.87      0.61      0.71        76
           2       0.71      0.91      0.80        81

    accuracy                           0.76       157
   macro avg       0.79      0.76      0.76       157
weighted avg       0.79      0.76      0.76       157
```

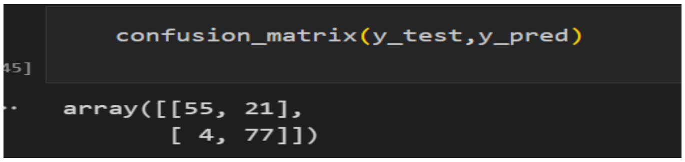
</td>
</tr>
</table>

Final Model Selection Justification (2 Marks):

<table>
<tr>
<th>Final Model</th>
<th>Reasoning</th>
</tr>
<tr>
<td>Random Forest</td>
<td>



```
print(classification_report(y_test,y_pred))

Accuracy: 0.8407643312101911
              precision    recall  f1-score   support

           0       0.93      0.72      0.81        76
           2       0.79      0.95      0.86        81

    accuracy                           0.84       157
   macro avg       0.86      0.84      0.84       157
weighted avg       0.86      0.84      0.84       157
```



```
        confusion_matrix(y_test,y_pred)
45]

..  array([[55, 21],
           [ 4, 77]])
```

- This model has been selected because it has the high accuracy and f1-score compared to the other model mentioned above.

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

NOTE: I have done other models like Gradient Boosting Classifier, AdaBoost Classifier these model will be available in the lliver.ipynb file.