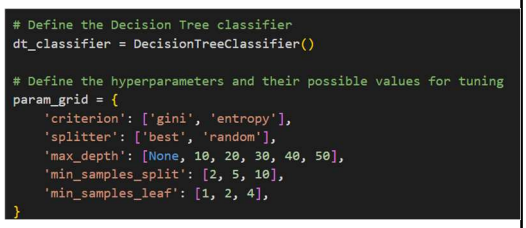
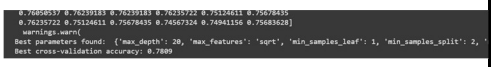
Model Optimization and Tuning Phase Report

Date	20 June 2025
Team ID	SWTID1749791625
Project Title	Smart Lender- Applicant Credibility Prediction for Loan Approval
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
Decision Tree		
Random Forest		

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

```python
# Define the KNN classifier
knn = KNeighborsClassifier()

# Define the hyperparameters and their possible values for tuning
param_grid = {
    'n_neighbors': [3, 5, 7, 9],
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan']
}
# Define the KNN classifier
knn = KNeighborsClassifier()

# Define the hyperparameters and their possible values for tuning
param_grid = {
    'n_neighbors': [3, 5, 7, 9],
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan']
}
```

</td>
<td>

```python
# Evaluate the performance of the tuned model
accuracy = accuracy_score(y_test, y_pred)
print("Optimal Hyperparameters:", best_params)
print("Accuracy on Test Set:", accuracy)
```

```
Optimal Hyperparameters: {'metric': 'euclidean', 'n_neighbors': 9, 'weights': 'uniform'}
Accuracy on Test Set: 0.8536585365853658
```

</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>

```python
from sklearn.model_selection import GridSearchCV
from xgboost import XGBClassifier

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'subsample': [0.8, 1.0],
    'colsample_bytree': [0.8, 1.0]
}

xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42
grid_search = GridSearchCV(estimator=xgb, param_grid=param_grid,
                           scoring='accuracy', cv=5, n_jobs=-1, verbose=2)
grid_search.fit(X_train_scaled, y_train)
```

</td>
<td>

```python
print("Best Loading... s found: ", grid_search.best_params_)
print("Best cross-validation accuracy: {:.4f}".format(grid_search.best_score_))
```

```
Fitting 5 folds for each of 108 candidates, totalling 540 fits
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [10:05:36] WARNING: /workspace/src/lear
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters found:  {'colsample_bytree': 1.0, 'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 200, 'su
Best cross-validation accuracy: 0.7984
```

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

Performance Metrics Comparison Report (2 Marks):

<table>
<tr><th>Model</th><th>Optimized Metric</th></tr>
<tr>
<td>Decision Tree</td>
<td>

```
Test Set Metrics for Decision Tree:
              precision    recall  f1-score   support

           0       0.46      0.68      0.55        38
           1       0.82      0.65      0.72        85

    accuracy                           0.66       123
   macro avg       0.64      0.67      0.64       123
weighted avg       0.71      0.66      0.67       123
```

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

Random Forest	🧪 Test Set Metrics for Random Forest: precision recall f1-score support 0 0.52 0.66 0.58 38 1 0.83 0.73 0.78 85 accuracy 0.71 123 macro avg 0.67 0.69 0.68 123 weighted avg 0.73 0.71 0.72 123
KNN	🧪 Test Set Metrics for KNN: precision recall f1-score support 0 0.54 0.66 0.60 38 1 0.83 0.75 0.79 85 accuracy 0.72 123 macro avg 0.69 0.71 0.69 123 weighted avg 0.74 0.72 0.73 123
Gradient Boosting	🧪 Test Set Metrics for XGBoost: precision recall f1-score support 0 0.33 0.79 0.46 38 1 0.74 0.27 0.40 85 accuracy 0.43 123 macro avg 0.53 0.53 0.43 123 weighted avg 0.61 0.43 0.42 123

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Ensemble Voting Classifier	After training and evaluating multiple machine learning models — including Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and XGBoost — I selected the Ensemble Voting Classifier as the final model.This decision was based on its superior performance across metrics such as accuracy, precision, recall, and f1-score.