

## Model Development Phase Template

|               |  |
|---------------|--|
| Date          | 15 March 2024  |
| Team ID       | SWTID1749835773  |
| Project Title | SmartLender - Applicant Credibility Prediction for Loan Approval |
| Maximum Marks | 6 Marks  |

### Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

| Model         | Description   | Hyperparameters  | Performance Metric (e.g., Accuracy, F1 Score) |
|---------------|---|--|---|
| Random Forest | Ensemble of decision trees; robust, handles complex relationships, reduces overfitting, and provides feature importance for loan approval prediction. | <pre> grid_search.fit(x_train, y_train) print(f"Best hyperparameters found by Grid Search: {grid_search.best_params_}") best_rf_model = grid_search.best_estimator_  # Make predictions on the test set y_pred = best_rf_model.predict(x_test)  # Fitting 5 folds for each of 126 candidates, totalling 630 fits Best hyperparameters found by Grid Search: {'criterion': 'gini', 'max_depth': np.int64(1), 'n_estimators': 50 </pre>        | Accuracy score = 73.3%                        |
| Decision Tree | Simple tree structure; interpretable, captures non-linear relationships, suitable for initial insights into loan approval patterns.                   | <pre> print(f"Best hyperparameters found by Grid Search: {grid_search_dctree.best_params_}") model_opt = grid_search_dctree.best_estimator_ y_pred = model_opt.predict(x_test)  # Fitting 5 folds for each of 34 candidates, totalling 170 fits Best hyperparameters found by Grid Search: {'criterion': 'gini', 'max_depth': np.int64(15)} </pre>   | Accuracy score = 78.8%                        |
| KNN           | Classifies based on nearest neighbors; adapts well to data patterns, effective  | <pre> print(f"Best hyperparameters found by Grid Search: {grid_search_best_params}") model_knn_best = grid_search_best_estimator_ model_knn_best.fit(x_train_scaled, y_train) y_pred = model_knn_best.predict(x_test_scaled) # Use scaled data for prediction  # Fitting 5 folds for each of 40 candidates, totalling 200 fits Best hyperparameters found by Grid Search: {'n_neighbors': np.int64(10), 'p': 1, 'weights': 'uniform'} </pre> | Accuracy score = 74%                          |

|           |  |   |                        |
|-----------|--|---|------------------------|
|           | for local variations in loan approval criteria.  |   |                        |
| XGB Boost | XGB boost with trees; optimizes predictive performance, handles complex relationships, and is suitable for accurate loan approval predictions. | <pre> print("Best score:", grid_search.best_score_) print("Best parameters:", grid_search.best_params_) best_model = grid_search.best_estimator_  y_pred_xgb = best_model.predict(x_test)  Fitting 5 folds for each of 12 candidates, totalling 60 fits Best score: 0.767957834888861 Best parameters: {'learning_rate': 0.01, 'n_estimators': 50} </pre> | Accuracy score = 78.8% |