Configuration Manual

MSc Research Project
MSCFTD1- Master of Science in Fintech (Financial Technology)

Olasubomi Opayemi
Student ID: 23343575

School of Computing
National College of Ireland

Supervisor: Dr Brian Byrne.

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Olasubomi Olatayo Opayemi……

Student ID: ……23343575……..……

Programme: ……MSCFTD1……**Year:** ……2024/2025..

Module: ……..……

Lecturer: ……..……

Submission Due Date: ……11TH August 2025……..……

Project Title: ……..……

Word Count: …… **Page Count:** ……..……..……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……O. OPAYEMI……

Date: ……11th August 2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

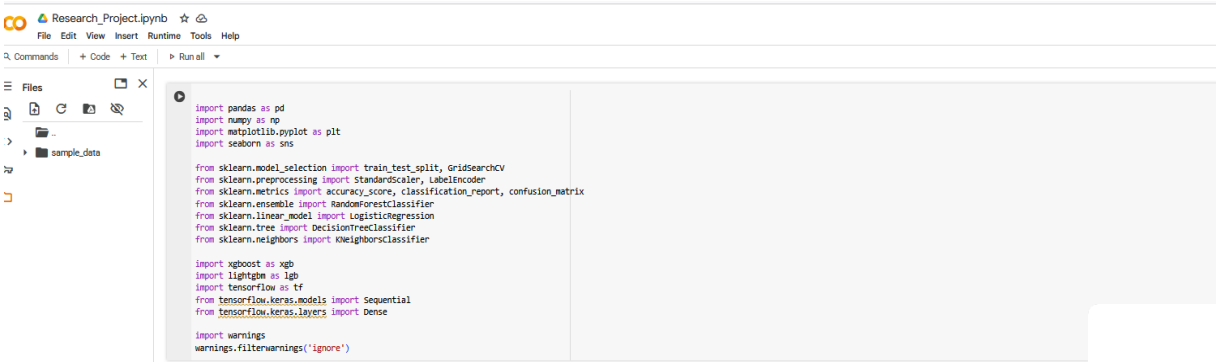
Olasubomi Olatayo Opayemi
Student ID: 23343575

This manual provides step-by-step guidance for configuring and running the Credit Risk Default Prediction project based on the Jupyter Notebook artifact attached to the project. The project uses Python programming Language for data preprocessing, feature engineering, modeling, and evaluation of machine learning techniques for loan default prediction. It is designed to run in environments like Google Colab or a local setup. Configuration focuses on environment setup, dependency installation, dataset preparation, and model engineering.

Setup

The project can run in Google Colab (cloud-based, no local install needed) or locally via Jupyter Notebook.
Google Colab Setup

1. Open a web browser and navigate to colab.research.google.com.
2. Upload the ipynb file or open it from GitHub/Drive.
3. Install dependencies in the first cell (such as the libraries needed for the code)
4. Mount Google Drive for persistent storage: Add a cell with google. colab import drive; drive. Mount('/content/drive').



Dataset Preparation

Download the dataset: collect the dataset Loan_Default_Dataset.csv (source from Kaggle). Place it in the working directory in Colab.
Load the dataset: In the notebook, update the path in df = pd.read_csv('/content/Loan_Default_Dataset.csv') if needed.
Verify: Run df.head() to inspect the first 5 rows, ensuring 34 columns (e.g., ID, year, loan_limit, ..., Status, dtir1) and no immediate errors.

Model Configuration
The notebook trains eight models namely KNN, LightGBM, DNN, ANN, Random Forest, Decision Tree, XGBoost, Logistic Regression with hyperparameter tuning via GridSearchCV. Customize as follows:

Preprocessing:

Imputation: Default is forward-fill (ffill) and mean. Change strategy in SimpleImputer(strategy='median') for median imputation if outliers are a concern.
Encoding/Scaling: No changes needed unless adding one-hot encoding for categorical features (replace LabelEncoder with OneHotEncoder from scikit-learn).

Class Balancing:
Enable SMOTE: Add from imblearn.over_sampling import SMOTE; smote = SMOTE(random_state=42); X_train, y_train = smote.fit_resample(X_train, y_train) before modeling.
For ensembles: Set class_weight='balanced' in RandomForestClassifier or scale_pos_weight in XGBoost (calculate as len(y_train[y_train==0]) / len(y_train[y_train==1])).

Hyperparameter Tuning:

GridSearchCV: Expand grids, e.g., for XGBoost: param_grid = {'learning_rate': [0.01, 0.1, 0.3], 'n_estimators': [100, 200, 300], 'max_depth': [3, 5, 7, 9]}.
Neural Network: Adjust layers in model = Sequential([Dense(64, activation='relu', input_shape=(X_train.shape[1],)), Dense(32, activation='relu'), Dense(1, activation='sigmoid')]). Change epochs or batch size in model.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.2).

Evaluation:

Add AUC-ROC: Import from sklearn.metrics import roc_auc_score and compute roc_auc_score(y_test, model.predict_proba(X_test)[:,1]).
Visualize: Extend with ROC plots via sklearn.metrics.roc_curve.

Running the Notebook

Execute cells sequentially: Start with imports, load dataset, preprocess, then model and evaluate.
Runtime: 20-45 minutes on CPU; faster with GPU for Neural Network.
Output: Trained models, performance metrics (e.g., classification reports), feature importances, and correlation heatmap.
Troubleshooting:
Errors in imports: Reinstall dependencies.
Memory issues: Reduce batch size or use Colab's high-RAM runtime.
Dataset not found: Ensure correct path.

This manual ensures smooth configuration for replicating or extending the project.