National College of Ireland

Configuration Manual

MSc Research Project
MSc in FinTech Information

Laura Gomez
Student ID: x23369795

School of Computing
National College of Ireland

Supervisor: Dr. Brian Byrne

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Laura Gomez
Student ID:	x23369795
Programme:	MSc in FinTech Information
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr. Brian Byrne
Submission Due Date:	15/09/2025
Project Title:	Configuration Manual
Word Count:	672
Page Count:	4

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:	*Laura Gomez*
Date:	11th September 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

Laura Gomez
x23369795

1 Introduction

This configuration manual provides a detailed description of the software tools, libraries, and settings needed to reproduce the experimental setup displayed in this research. Implied volatility (IV) modeling using out-of-the-money (OTM) options through machine learning and deep learning approaches is executed on a given computing environment to ensure consistency, reproducibility, and accuracy. The experiments were conducted using Python in a Google Colaboratory environment, leveraging both traditional and modern libraries for data processing, modeling, optimization, and explainability. The guide includes all the dependencies, tools, and recommended runtimes necessary to reproduce the models and analyses, making it possible to reproduce them in future research or practical applications.

2 System Specification

Model: IdeaPad 5 2-in-1 14AHP9
Processor: AMD Ryzen 5 8645HS w/ Radeon 760M Graphics (4.30 GHz)
Installed RAM: 16.0 GB (13.8 GB usable)
System type: 64-bit operating system, x64-based processor
Edition: Windows 11 Home
Version: 24H2
OS build: 26100.4770

3 Execution Environment

The research was conducted using a Google Colaboratory (Colab)[1] environment, a hosted Jupyter Notebook service that offers free access to computer resources, including GPUs and TPUs, with no setup required. Colab is particularly well-suited to data science, teaching, and machine learning.

- Platform: Google Colab
- Runtime type: 3.10+
- Hardware accelerator: T4 GPU

[1] https://colab.google/

To replicate the research, it is recommended to use Google Colab. Alternatively, the setup can be replicated in a local or cloud-based JupyterLab environment using pip or conda to install the necessary packages.

4 Required Packages

Table 1 contains a list of Python libraries used in the code artefact workflow.

Package	**Purpose**
`numpy`	Numerical operations
`pandas`	Data handling and preprocessing
`scipy`	Statistical analysis
`matplotlib`	Plotting and visualization
`seaborn`	Enhanced data visualization
`plotly`	Plotting and visualization
`scikit-learn`	Machine learning models and metrics
`xgboost`	Gradient boosting regression
`lightgbm`	Alternative boosting model (optional)
`tensorflow`	Neural networks (CNN, LSTM, GRU, MLP)
`keras`	High-level deep learning API
`optuna`	Optuna for hyperparameter tuning
`shap`	Explainable AI (SHAP values)
`lime`	Local interpretability of models
`joblib`	Model serialization

Table 1: Recommended Python packages

5 Data Requirements

Data source:
Primary Source: Yahoo Finance API [2]
Stock: Apple Inc. (AAPL)
Data Type: Options chain data
File Format: CSV
Data Structure: The columns in the input CSV file include:

- contractSymbol: Option contract identifier
- lastTradeDate: Timestamp of last trade
- strike: Strike price
- lastPrice: Last traded price
- bid: Bid price
- ask: Ask price

[2]https://finance.yahoo.com/quote/AAPL/options/

- change: Price change
- percentChange: Percentage change
- volume: Trading volume
- openInterest: Open interest
- impliedVolatility: Implied volatility
- inTheMoney: Boolean flag
- contractSize: Contract size
- currency: Currency denomination
- Expiry: Expiration date
- Type: Option type (Call/Put)
- Date: Trading date
- Days to Expiration: Days until expiration
- Stock Price: Underlying stock price
- Dividend Yield: Dividend yield
- Treasury Rate: Risk-free rate

6 Execution Steps

To replicate the artefact:

1. Start the notebook: Open the provided link in the research report in Google Colab environment.
2. Upload Dataset: Load the dataset 'AAPL_Options_extended.csv' file by clicking the folder icon in the left-side controls and then selecting the upload icon, as shown in Figure 1. Make sure you are connected to Python in the upper right corner.

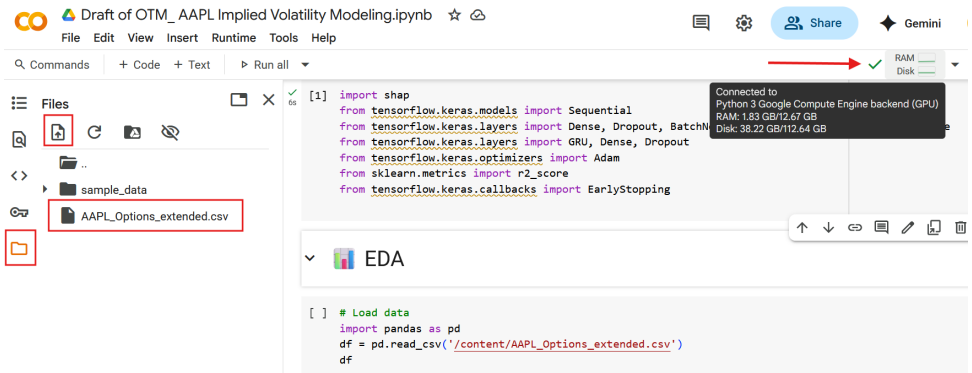

Figure 1: Load the dataset

3. Run cells sequentially, the notebook is structured into the following sections:

    - Data Loading and Cleaning
    - Feature Engineering (Moneyness, OTM Distance, Bid-Ask Spread)
    - Model Definition and Training (Linear, Tree-based, Neural Networks)
    - Hyperparameter Optimization with Optuna
    - Evaluation using MAE, MSE, RMSE, R^2
    - SHAP and LIME for Interpretability
    - Residual Analysis for the model prediction evaluation.

Note: Each cell should be run top-down without skipping.

7 Final Checklist

Before running the full experiment, ensure:

- ❑ All required packages installed and importable
- ❑ The dataset file is accessible and uploaded
- ❑ The GPU is configured
- ❑ Sufficient computational resources are available

Estimated Setup Time: 15-30 minutes for Google Colab, 30-60 minutes for local environment.

Total Experiment Runtime: 2-4 hours, depending on hardware and hyperparameter optimization settings.