National College of Ireland

Configuration Manual

Investment Portfolio Optimization and ESG Factors Impact on Financial Performance Using Machine Learning

Akansha Sharma
x23194961
MSCDAD
National College of Ireland

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Akansha Sharma……

Student ID: x23194961……..……

Programme: MSCDAD **Year:** 2024

Module: MSc Research Project

Lecturer: Prof. William Clifford

Submission Due Date: 12/12/2024

Project Title: Investment Portfolio Optimization and ESG Factors Impact on Financial Performance using Machine

Word Count: **1479** **Page Count:** 10

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: Akansha Sharma

Date: 12/12/2024

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

Akansha Sharma
Student ID: x23194961

1 Section 1: System Overview

This configuration manual is designed to assemble a stock data analysis system that includes Environmental, Social, and Governance (ESG) risk analysis. The configuration of the necessary hardware and software, the data acquisition processes, the preprocessing steps required for model implementation, and the implementation strategies for constructing models are instructed in the manual for user to follow. The purpose of this system was to accommodate a comprehensive financial analysis with ESG factors involved in stock performance evaluations.

1.1 Purpose of the System

This system is designed to analyse stock performance of S&P 500 companies based on their ESG risk scores. It enables users to see how ESG factors impact market behaviour and make investment decisions that fact, financial and sustainability metrics go hand in hand.

1.2 Target Audience

Written for data analysts, financial analysts, researchers, and investment professionals who want to use ESG factors in their stock analysis, this manual provides everything needed to know about ESG factors. Its important users have a basic understanding of Python programming and data analysis concepts.

2 Section 2: System Requirements

Hardware and software being used should be certain for optimal performance of the stock data analysis system.

2.1 Hardware Requirements

Processor: Minimum requirement is a multi-core processor with dual core (quad core or more recommended) to enable efficient data processing.
RAM: Minimum of 8GB (16GB recommended) is required to effectively complete data intensive workloads.
Storage: Dataset, models, and outputs will need at least 100 GB of free disk space.
Graphics Card: For deep learning tasks, it will want to speed up training times, and a dedicated GPU with CUDA support should be your recommendation.

2.2 Software Requirements

Operating System: Works with Windows 10 or newer, macOS Mojave or newer, or Linux distributions (Ubuntu recommended).
Python Version: For the sake of libraries, it needs to be running Python 3.7 (or higher).
Integrated Development Environment (IDE): Writing and executing Python scripts can be done from Jupyter Notebook, PyCharm or Visual Studio Code.

2.3 Required Libraries

The following Python libraries are necessary for the system's operation:

- yfinance: for downloading stock price data.
- pandas: For manipulation and analysis of data.
- numpy: For numerical operations.
- matplotlib and seaborn: If it need to do something with that.
- scikit-learn: For values when one is machine learning task, including model evaluation.
- keras and tensorflow: To build deep learning models.
- Users can install these libraries using pip:
    - yfinance pandas numpy matplotlib seaborn scikit-learn keras tensorflow

3 Section 3: Data Acquisition

This acquires stock price data and ESG risk scores and this section describes the process of getting stock price data and ESG risk scores for analysis.

3.1 Stock Data Acquisition

The system uses the yfinance library to download historical stock pricing data for S&P 500 companies. The following steps outline the acquisition process:

- Identify S&P 500 Companies: Get a list of S&P 500 companies from reliable financial websites or the Wikipedia.
- Download Stock Data: To do this this can use the yfinance library and download adjusted closing prices for a specified time (for instance the last ten years).

3.2 ESG Risk Scores Acquisition

Typically, available online in CSV format datasets, ESG risk scores should be sourced from reputable sources. Users should ensure that the dataset includes relevant columns such as:

- Company Name
- Total ESG Risk Score

- Sector
- Environmental Risk Score
- Social Risk Score
- Governance Risk Score

For this research, S&P 500 ESG Data was obtained from Kaggle (https://www.kaggle.com/datasets/pritish509/s-and-p-500-esg-risk-ratings).

4 Section 4: Data Preprocessing

Preparing datasets for analysis and modeling is crucial and that requires data pre-processing.

4.1 Handling Missing Values

Users should also verify that neither stock market price nor ESG data is missing. Common strategies include:

- Imputing missing values in numerical columns so that there is continuity in sequence.
- If not, then this can fill the categorical columns with default values (such as 'Unknown').
- Filling ESG scores with zeros as appropriate or median values, as appropriate.

4.2 Data Normalization

All features must contribute equally during training when using a specific normalization technique such as Min-Max scaling or Standardization for numerical features.

4.3 Feature Engineering

Creating additional features can enhance model performance:

- Rolling averages of the stock price, volatility measures.
- ESG scores can be combined into a single weight score reflecting the relative importance of ESG criteria for investment decisions, according to user defined criteria.

5 Section 5: Model Development

This discusses the steps of how to develop predictive models based on machine learning techniques suitable for financial analysis in this section.

5.1 Model Selection

Three primary modeling approaches are recommended:

- Portfolio Implementation Model: An ESG optimized asset allocation model based on historical performance data.

- LSTM (Long Short-Term Memory): It is suited to time series prediction tasks because it captures temporal dependencies in stock prices.
- SVM (Support Vector Machine): For classification tasks based on developer's feature sets built from financial and ESG data, the model is effective.
- Each model contributes to solving a different part of the overall analysis framework.

6 Section 6: Portfolio Implementation configuration

To implement a portfolio optimization model effectively, specific configurations must be established:

6.1 Portfolio Optimisation Data Preparation

- Downloaded stock price data, aggregate historical returns.
- This calculates the expected returns based on historical performance metrics.
- It can compute covariance matrices amongst selected stocks to show risk exposure.

6.2 Selection of Optimization Algorithm

Choose an optimization algorithm suitable for portfolio construction:
Mean-Variance Optimization: According to this classical approach, historical return distributions are maximized with minimum risk.
Heuristic Model: An advanced method using SLSQP method with accomodating additional contraints.

6.3 Implementation Steps

- Similarly, it defines maximum investment limits per asset and the sum of total portfolio weights; these must be one.
- While it is executing the optimization algorithm, use optimization libraries such as SciPy or specialized financial libraries.
- Once optimized portfolio is back tested against historical data to see how it has performed over time.

7 Section 7: LSTM Model Configuration

The LSTM model requires specific configurations tailored for time series forecasting tasks:

7.1 Data Preparation for LSTM

Create sequences of input data suitable to train the LSTM model (structure input data by defining a time window (previous 5 days prices).

Min-Max scaling of input features to make all values of input features fall in the same range.

7.2 Model Architecture Design

- Design an LSTM architecture that includes:
- Input Layer: It accepts sequences of historical prices.
- Hidden Layers: A series of LSTM layers and Dense layers taking outputs of LSTM layers.
- Output Layer: Given input sequences, it produces predictions of prices or returns.

7.3 Training Configuration

- Configure training parameters such as:
- Number of epochs (e.g., 50).
- Batch size (e.g., 32).
- Monitored validation loss based learning rate adjustments.

8 Section 8: SVM Model Configuration

The SVM model configuration focuses on classification tasks related to stock performance predictions based on ESG factors:

8.1 Data Preparation for SVM

- In selecting relevant features, financial metric and ESG score are selected as the input of SVM model.
- StandardScaler these features because they will use consistent scaling across features.

8.2 Model Training Configuration

Set up training parameters:

- Kernel Type: Give a choice of a linear, polynomial or radial basis function (RBF) kernel depending on dataset characteristics.
- Hyperparameter Tuning: It can use scikit-learn's GridSearchCV or RandomizedSearchCV to find the best hyperparameters: C (regularization parameter) and gamma (kernel coefficient) for example.

9 Section 9: Evaluation Metrics

Model evaluation is essential to assess predictive accuracy across all implemented models:

9.1 LSTM and SVM Performance Metrics

Utilize metrics such as:

- Regression tasks – predicting stock prices with LSTM: Mean Squared Error (MSE) and Mean Absolute Error (MAE).
- This compute accuracy score, precision, recall, F1-score and confusion matrix for classification tasks run by the SVM models.

10 Section 10: Conclusion

The purpose of this configuration manual is to be a complete guide on how to construct an ESG risk assessment integrated stock data analysis system. Adhering to these recommendations will help the users to identify stock performance while keeping in view key ESG factors and make more educated investments currently of present market's scenario.
If the users follow this manual's recommendations on how the hardware setup, software configuration, data acquisition process, preprocessing steps, model development strategies and evaluation techniques are handled, they will get a combination of financial metrics and sustainability considerations with no extra hassle.

References

Wikipedia Contributors. (2024). List of S&P 500 companies. Available at: https://en.wikipedia.org/wiki/List_of_S%26P_500_companies (Accessed: 9 December 2024).
Yahoo Finance. (2024). Historical stock data for S&P 500 companies. Available at: https://finance.yahoo.com/ (Accessed: 9 December 2024).
Pandas Development Team. (2023). pandas: powerful Python data analysis toolkit. Available at: https://pandas.pydata.org/ (Accessed: 9 December 2024).
NumPy Developers. (2023). NumPy: the fundamental package for scientific computing with Python. Available at: https://numpy.org/ (Accessed: 9 December 2024).
Seaborn Contributors. (2023). Seaborn: statistical data visualization. Available at: https://seaborn.pydata.org/ (Accessed: 9 December 2024).
Scikit-learn Developers. (2023). scikit-learn: machine learning in Python. Available at: https://scikit-learn.org/stable/ (Accessed: 9 December 2024).
Keras Contributors. (2023). Keras: the Python deep learning library. Available at: https://keras.io/ (Accessed: 9 December 2024).
TensorFlow Developers. (2023). TensorFlow: an end-to-end open-source platform for machine learning. Available at: https://www.tensorflow.org/ (Accessed: 9 December 2024).