National College of Ireland

Configuration Manual

MSc Research Project
MSc Cloud Computing

Deva Loknathan Marri
Student ID: x21231346

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Deva Loknathan Marri
Student ID:	x21231346
Programme:	MSc Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Ahmed Makki
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	708
Page Count:	5

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:	
Date:	14th December 2023

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

Deva Loknathan Marri
x21231346

1 Introduction

This guidebook tells you in great detail how to set up all the tools and software you'll need to build a whole system from scratch. With the help of the setup instructions, the study can be repeated in a more useful way. We are going to look at how the system works as a whole along with its user interface. In other words, how a person or a user will use the system user interface to connect with our system. This paper lists the steps that can be taken to replicate the implementation of the "ML model and Partial Homomorphic Encryption".
The Configuration Manual will be segmented into the following parts.

- Environmental Setup
- Code Implementation

2 Environmental Setup

2.1 Hardware Requirements:

- 8GB RAM.
- 250 GB HDD.
- 2.6 GHz Intel. Core i5

2.2 Software Requirements:

- Windows 10
- Python 3.11.7

2.3 Programming Requirements:

- Python(Version 3.11.7)
- Jupyter Notebook
- Visual Studio Code

Library	Usage
pandas	Data manipulation and analysis. Used for reading and processing CSV files, as well as handling dataframes.
numpy	Numerical operations. Used for numerical computations and operations on arrays, such as converting data to arrays for machine learning models.
sklearn	Machine learning library. Used for linear regression modeling, data preprocessing (LabelEncoder), and splitting data into training and testing sets.
phe	Paillier homomorphic encryption library. Used for generating Paillier public and private keys, encrypting and decrypting data, and performing homomorphic computations.
json	JSON encoding and decoding. Used for serializing and deserializing data to and from JSON format, particularly for storing and loading keys and encrypted data.
matplotlib.pyplot	Plotting library for creating visualizations.
seaborn	Statistical data visualization library based on Matplotlib.
ipywidget	Interactive HTML widgets for Jupyter notebooks.

Table 1: Libraries and Their Usage

3 Libraries required:

The necessary libraries for constructing this research project are listed in table 1, along with their respective applications.

4 Coding Implementation:

This section presents the implementation of a Partial homomorphic encryption system. The whole system has been divided into fundamental functions, including Machine learning predictionModel, key generation, encryption, decryption, and evaluation. To execute the program, we must access the folder that contains the code in Visual Studio Code.

"Partial Homomorphic Encryption" Data encryption in classical cryptography relies on a public key. The two parties then trade private keys in order to decode it. For data processing to take place in the cloud, the secret key must be accessible to the cloud server. With homomorphic encryption, the cloud may securely conduct calculations on encrypted data or ciphertext, simplifying the process. Afterwards, provide the data owner with the encrypted findings. Therefore, data remains completely private regardless of its storage location as it is never decrypted.

Partially Homomorphic Encryption is a kind of encryption that requires two conditions

to be satisfied: 1) The product of D and E, where D is the Decrypt function and E is the Encrypt function, and A and B are the ciphertexts, is equal to A+B. 2) D(E(A)*scaler) = A*scaler, where scaler is a constant number (the value's exponent according to this project).

1)I have included Pascal Paillier Homomorphic Encryption in the given code. This section of the code is structured as follows: first, linmodel.py: Using characteristics like N, P, K, temperature, humidity, ph, and rainfall, this website uses linear regression on the crop recommendation csv dataset to forecast results for certain conditions. Second, log.py: Using characteristics like N, P, K, temperature, humidity, ph, and rainfall, this website uses logistic regression on the crop recommendation csv dataset to forecast results for certain conditions.

2) cust.py: On the client side, this page uses Paillier Homomorphic encryption to generate public and private keys. It then uses the encrypt() function to serialize the data, producing an encrypted number that can be saved on the cloud. Finally, it creates a json file to be sent to the server or the cloud. Additionally, the answer.json file that the client gets from the server is loaded using the loadAnswer() method.

3) The third page, servercalc.py, is responsible for decrypting the data given by the customer. It does this by using the EncryptedNumber() function, which generates the client-side EncryptedNumber using the public key and ciphertext. The server-side serialization process continues by multiplying the encrypted number by the coefficients produced by linear regression on the data. This produces the expected outcomes for the related data. Last but not least, updating the answer.json file with the serialized forecast data.

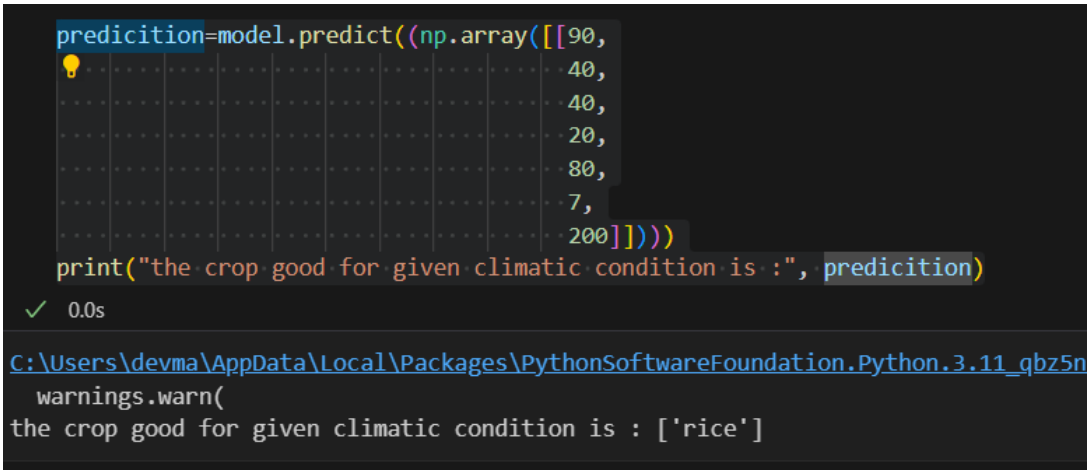

Figure 1: ML output

References

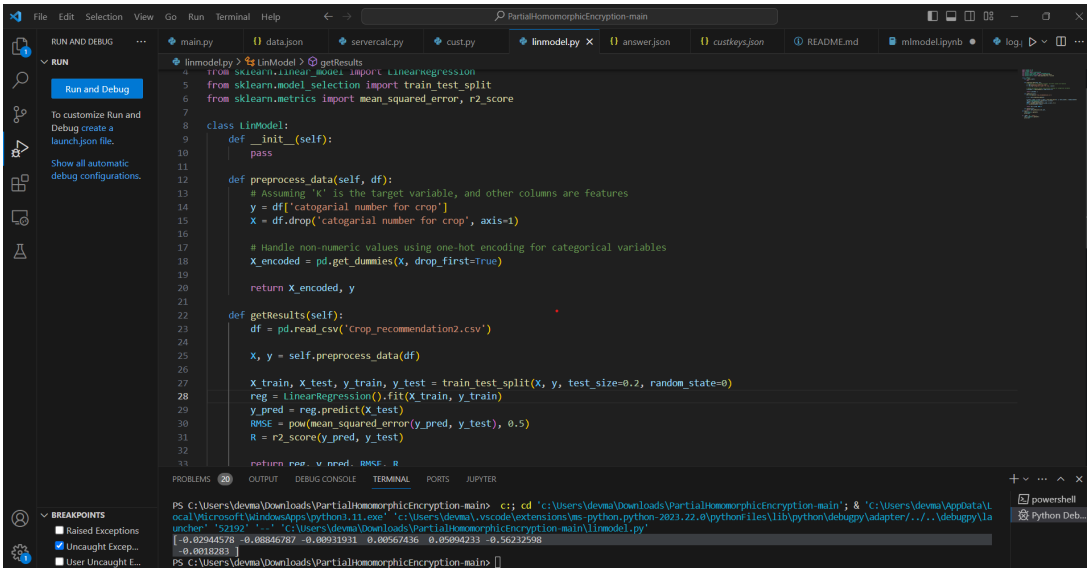

Figure 2: linear regression

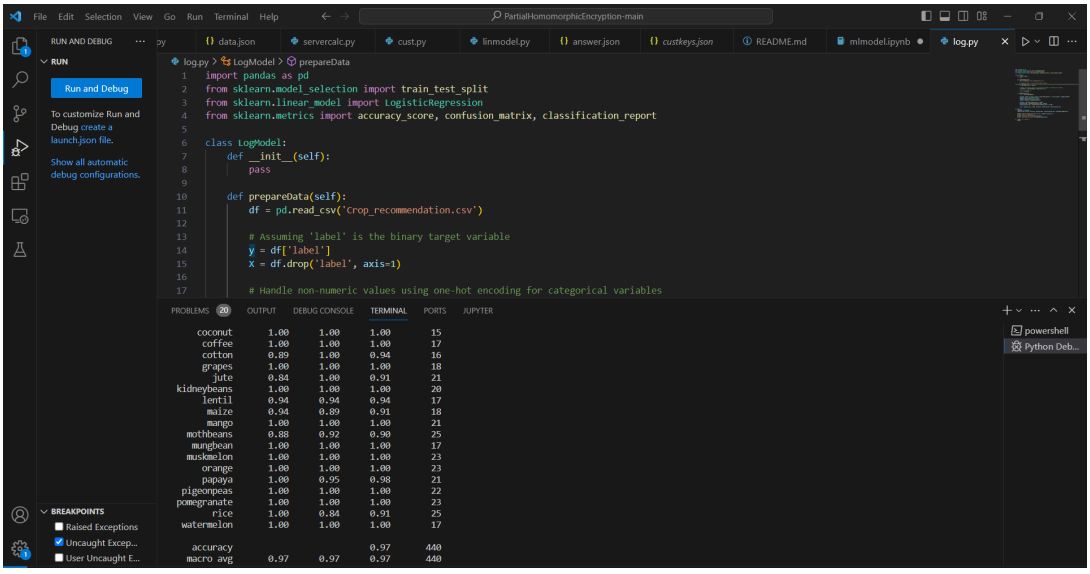

Figure 3: logistic regression

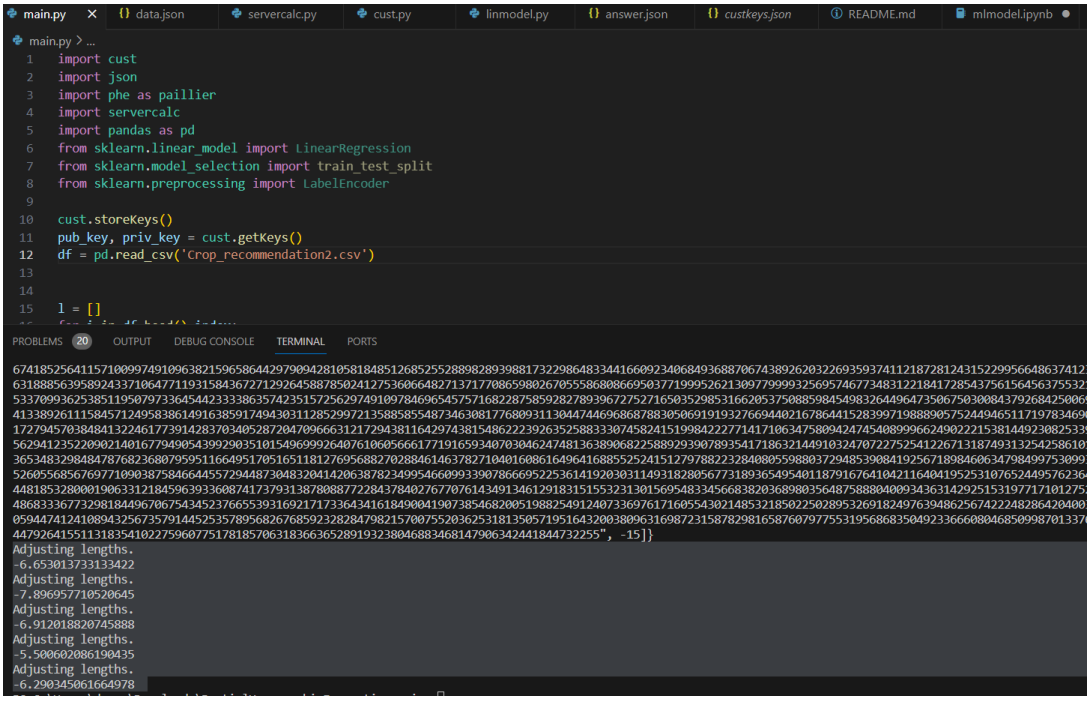

Figure 4: main.py