Task 1

1. Approach Overview

• **Data Exploration and Preprocessing:** Initially, I delved into the dataset to understand its structure, features, and distributions. Addressing missing values was a primary concern, and I opted to drop them from the dataset. Additionally, I encoded categorical variables, specifically the crop labels, utilizing the LabelEncoder.

• **Model Training:** After careful consideration, I picked the Random Forest Classifier as the tool for learning because it can work well with both numbers and categories. Then, I split the data into groups for learning and testing.

• **Model Evaluation:** To see if the model works well, I checked how accurate it is on the testing data, using the accuracy score.

• **Joblib Model Creation and Prediction**: Next, I made a joblib model from the Random Forest Classifier I trained. This model can be used to guess what might happen in new environmental situations.

2. Challenges Encountered

• Handling missing data was a big challenge. Choosing to remove it meant possibly losing important information.

• Picking the right machine learning method needed careful thought. I had to consider things like what the data looked like, how easy it was to understand the model, and how fast it ran.

• Even though the Random Forest Classifier worked well, it wasn't as easy to understand as simpler models like Decision Trees

3. Insights from Model Evaluation

• The model showed some accuracy in guessing crop names based on the environment. This suggests it could be useful in suggesting which crops to grow.

4. Suggestions for Enhancing Model Performance

• Suggestions for Improving Model Performance

• Feature Engineering: Look into making new features or changing the ones already there to help the model make better guesses.

• Hyperparameter Tuning: Adjust the settings of the Random Forest Classifier to see if the model can do better.

• Ensemble Methods: Try other ways of combining models or even looking into deep learning to see if they can make the model work better.

• Handling Unbalanced Data: If there are more of one kind of data than another, try techniques like adding more of the smaller group or taking some away from the bigger one to help the model work better.

5. Instructions for Running the Code and Reproducing the Results:

 • Ensure the presence of the dataset "Crop_Dataset.csv" in the same directory as the Python script.

• Execute the provided Python script, which encompasses code for data preprocessing, model training, evaluation, and prediction.

• The script will output the model's accuracy on the testing dataset and save the trained model as "crop_prediction_model.joblib".

• To make predictions on new environmental conditions, uncomment the relevant code block in the script and supply the new data as instructed.

• Follow the guidance provided within the script comments for seamless execution and replication of results.