**AI-BASED CROP RECOMMENDATION SYSTEM**

Submitted for the evaluation of P3

For

**Machine Learning Lab**

**(18B1WCI674)**

**Computer Science and Engineering**

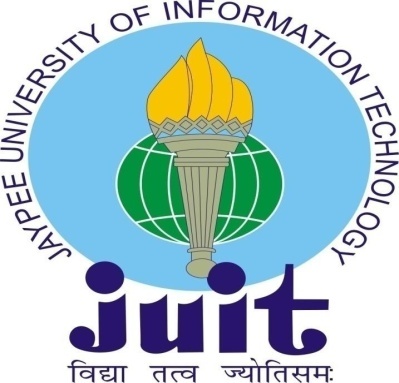
**By**

**Chandan Kumar (211301)**

**Gourav Gupta (211294)**

**Submitted To**

**Prof. Shubham Goyal**



Department of Computer Science & Engineering and Information Technology

# Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

**INTRODUCTION**

Agriculture is the backbone of many economies worldwide, providing food, employment, and raw materials for various industries. In India, agriculture is a significant sector, contributing to 17.8% of the country's Gross Value Added (GVA) in 2020, employing 41.49% of the total workforce in 2020, and sustaining about 58% of rural households.

According to the Food and Agriculture Organization (FAO), over 2.5 billion people depend on agriculture for their livelihoods. However, the agricultural sector is grappling with numerous challenges, including climate change, population growth, and resource constraints, which threaten its sustainability and productivity. These challenges are particularly acute in India, where agriculture is largely rain-fed and thus highly vulnerable to climate variability.

One of the critical decisions that farmers make is choosing the right crop to plant. This decision is influenced by a myriad of factors, including soil type, climate conditions, and market demand. Traditionally, this decision has been based on farmers' experience and local knowledge. However, this approach may not always yield optimal results, especially in the face of rapidly changing environmental and market conditions.

In recent years, there has been a growing interest in leveraging Artificial Intelligence (AI) to enhance agricultural productivity and sustainability. AI, with its ability to analyze vast amounts of data, identify patterns, and make predictions, can provide valuable insights for decision-making in agriculture. In this context, our project aims to develop an AI-based Crop Recommendation System.

This system uses a dataset available on Kaggle containing information about various crops and the conditions suitable for their growth. The dataset includes parameters such as N, P, K (primary soil nutrients), temperature, humidity, pH level, and rainfall. These parameters are crucial in determining the suitability of a crop to a particular region.

The system employs several machine-learning models, including the SdcaNonCalibrated Model, OneVersusAllLBFGS Model, and LbfgsMaximumEntropy Model, which are trained using the ModelTrainer class. These models predict the most suitable crop for given conditions.

By providing accurate and timely crop recommendations, this system can help farmers in India and other parts of the world make informed decisions, optimize resource use, increase crop yields, and ultimately enhance agricultural sustainability and food security. This project represents a significant step towards the integration of AI in agriculture, paving the way for more efficient and sustainable farming practices.

**PROBLEM STATEMENT**

Given the critical role of agriculture in supporting livelihoods and economies, particularly in countries like India, and the myriad of challenges it faces, there is an urgent need for innovative solutions that can enhance agricultural productivity and sustainability. One of the key decisions that farmers make, which significantly impacts agricultural outcomes, is the choice of crop to plant. This decision is influenced by various factors, including soil type, climate conditions, and market demand. However, traditional decision-making approaches based on farmers' experience and local knowledge may not always yield optimal results, especially in the face of rapidly changing environmental and market conditions.

Therefore, the problem this project aims to address is: **How can we leverage Artificial Intelligence to provide accurate and timely crop recommendations to farmers, helping them make informed decisions, optimize resource use, increase crop yields, and enhance agricultural sustainability and food security?**

To solve this problem, we propose to develop an AI-based Crop Recommendation System using a dataset available on Kaggle, which contains information about various crops and the conditions suitable for their growth. The system will employ several machine learning models to predict the most suitable crop for given conditions. This project represents a significant step towards the integration of AI in agriculture, paving the way for more efficient and sustainable farming practices.

**DATASET DESCRIPTION**

The dataset used in this project is the "Crop Recommendation Dataset" available on Kaggle. It contains data about various crops and the conditions suitable for their growth. The dataset is a comprehensive collection of information that can be used to understand the relationship between different environmental factors and the suitability of a particular crop.

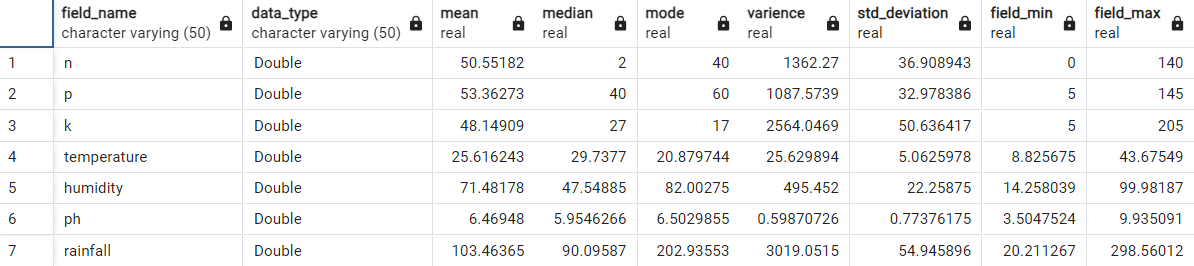
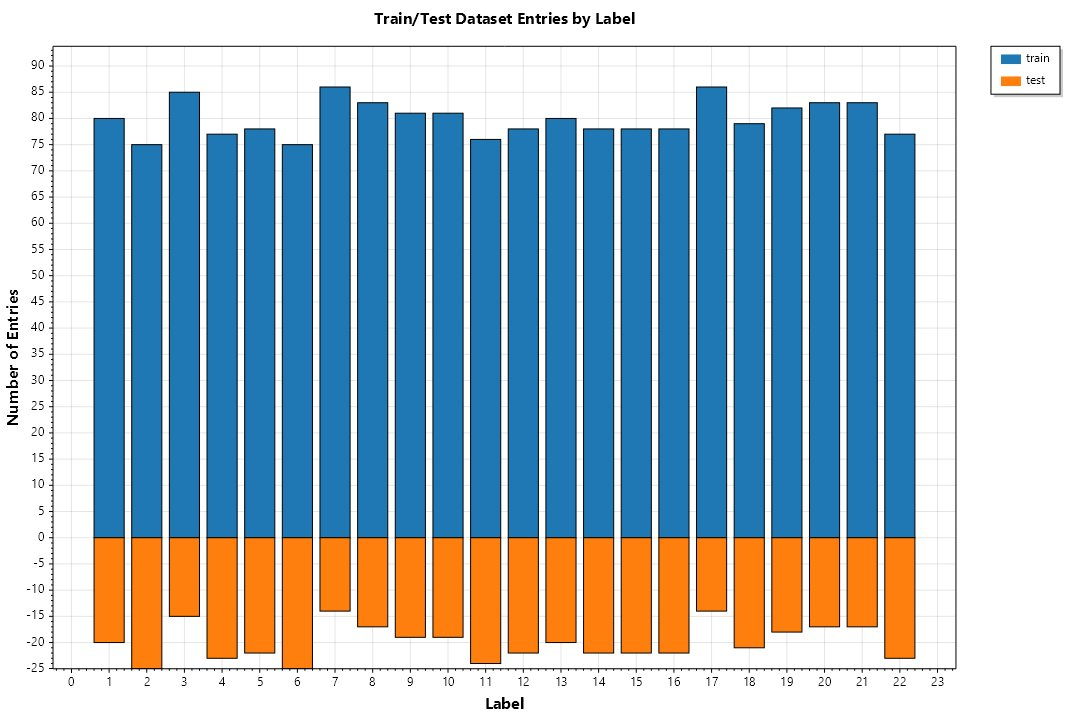


Table 1) General Information of Dataset



Graph 1) Plot if test and train data.

The dataset comprises 2200 instances with 8 attributes, including N, P, K (primary soil nutrients), temperature, humidity, pH level, rainfall, and the label of the crop. These attributes are crucial in determining the suitability of a crop to a particular region.

Here is a brief description of each attribute:

1. **N (Nitrogen)**: Nitrogen is a vital nutrient for plant growth and plays a key role in the formation of proteins, enzymes, and chlorophyll. It is measured in kilograms per hectare.
2. **P (Phosphorus)**: Phosphorus is essential for energy transfer and storage in plants. It is also crucial for cell division and plant growth. It is measured in kilograms per hectare.
3. **K (Potassium)**: Potassium helps in the regulation of CO2 uptake and water usage in plants. It also plays a role in protein synthesis. It is measured in kilograms per hectare.
4. **Temperature**: This represents the average temperature in Celsius of the region where the crop is grown.
5. **Humidity**: This represents the relative humidity of the region where the crop is grown, measured in percentage.
6. **pH**: pH is a measure of the acidity or alkalinity of the soil. It is measured on a scale from 0 (very acidic) to 14 (very alkaline), with 7 being neutral.
7. **Rainfall**: This represents the average rainfall in millimetres received in the region where the crop is grown.
8. **Label**: This is the target variable and represents the type of crop. The dataset includes 22 different types of crops.

The dataset does not contain any missing or null values, ensuring the quality and reliability of the analyses performed on it.

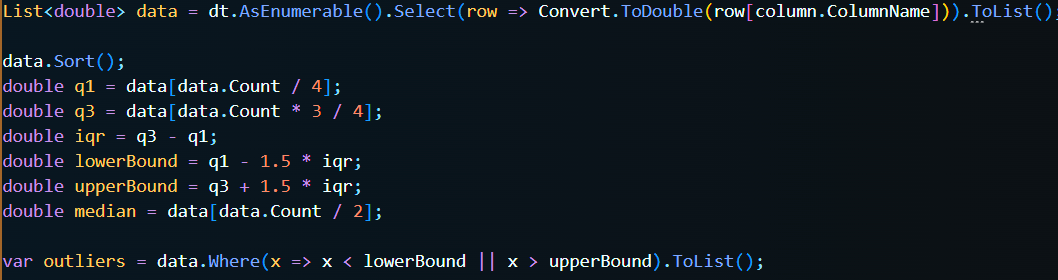
Visual inspection of the dataset reveals a diverse range of crops and a wide variation in the environmental conditions suitable for their growth. This diversity and variation present an excellent opportunity for developing robust machine-learning models for crop recommendation.

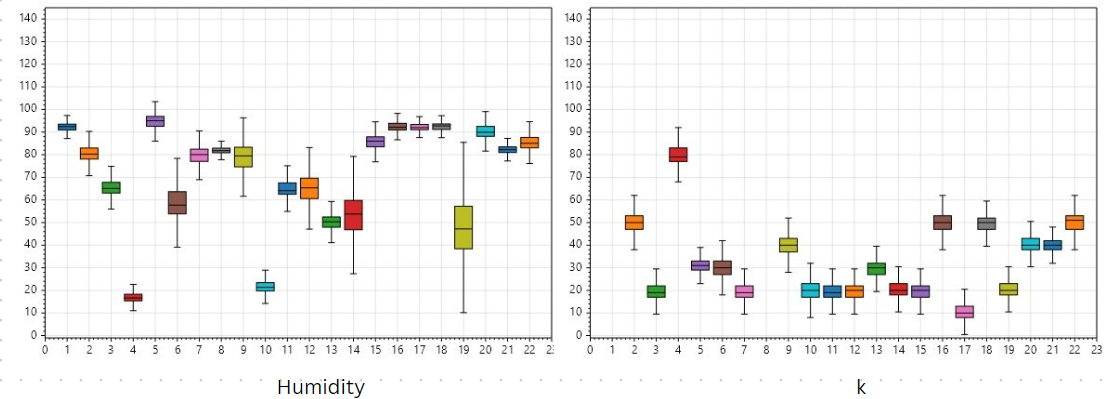
**DATA PREPROCESSING**

In our project, we have a dedicated class for data preprocessing, named DataPreprocessor.cs file. This class is responsible for preparing the raw data for further processing and analysis. The preprocessing steps include data extraction, outlier detection, normalization, and data splitting.

**Outlier Detection**

The OutlierDetector method in the DataPreprocessor class is responsible for detecting and handling outliers in the data. This method is currently commented out in the Preprocessor method, indicating that it may be a work in progress or optional depending on the data.

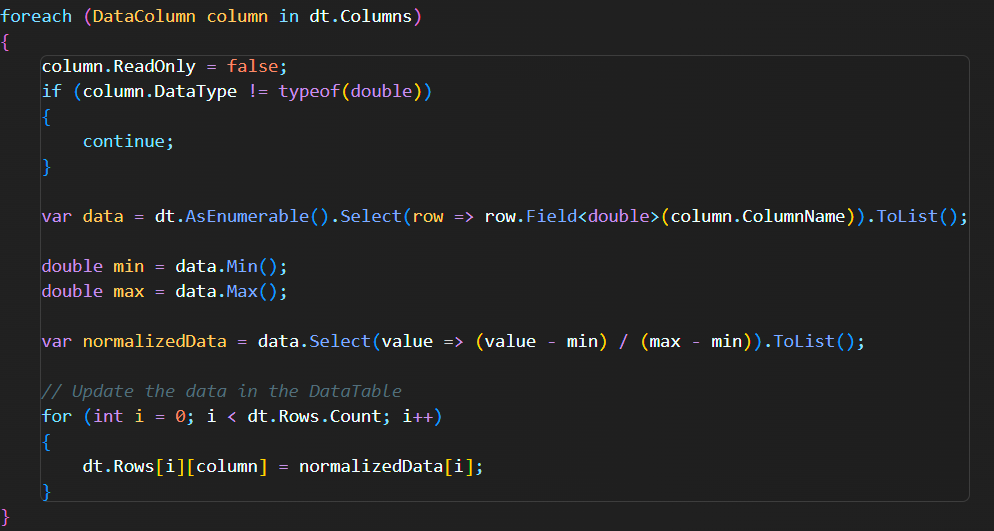




Graph 2) Plot of outlier detection result.

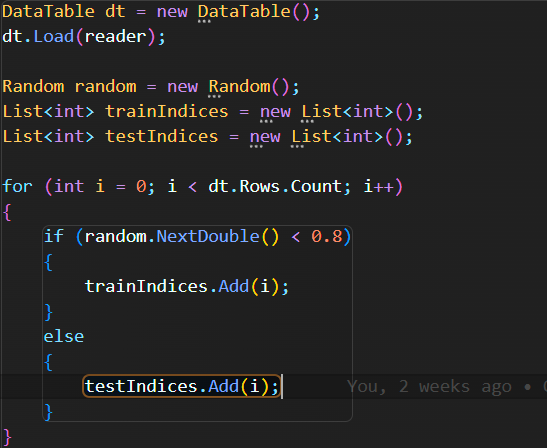
**Data Normalization**

The Normalizer method in the DataPreprocessor class is responsible for normalizing the data. Normalization is a crucial step in data preprocessing, especially for algorithms that use distance-based methods. It scales numeric data from different columns down to an equivalent scale so that no particular column influences the final result more due to its larger range.



**Data Splitting**

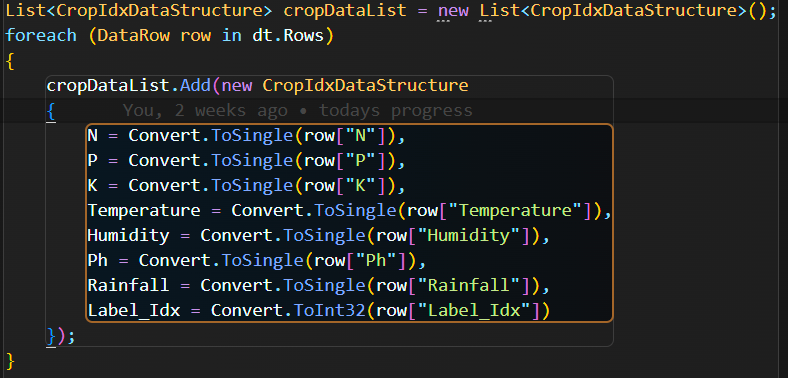
The DataSplitter method in the DataPreprocessor class is responsible for splitting the data into training and testing sets. This is a common practice in machine learning and data analysis, where a model is trained on a portion of the data (the training set) and then evaluated on a different portion (the test set) to test the model's ability to generalize to unseen data.After the data splitting process, the PlotData method is called to visualize the data, and the database connection is closed. The DataPreprocessor class and its methods provide a robust and flexible way to preprocess data, making it ready for further analysis and modeling.



**Model Building**

**Data Loading**

The first step in our model building process involves loading the data. We accomplish this using the LoadData method, which takes a SQL query as an argument and fetches the data from the database. The fetched data is then converted into a list of CropIdxDataStructure objects. This list is further converted into an IDataView object, which is a flexible, efficient way of describing tabular data (numeric and text) and is used as an input to ML.NET algorithms.

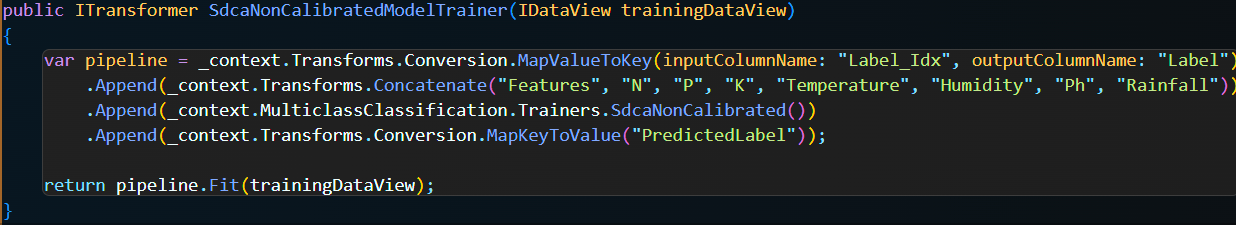


**Model Training**

We train three different models in our code:

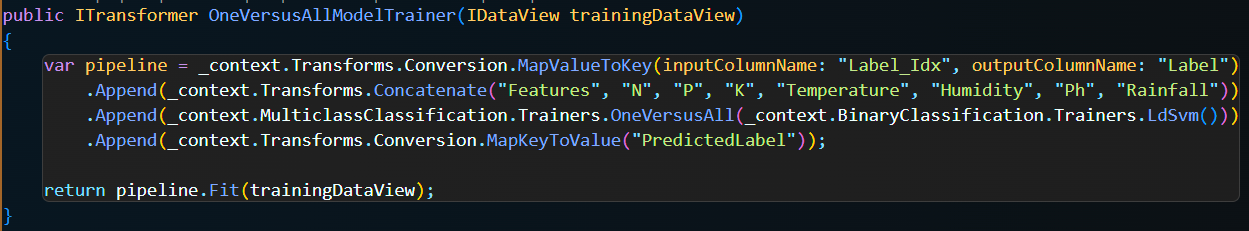
**Stochastic Dual Coordinate Ascent (SDCA)**

SDCA is an optimization algorithm used to solve certain types of optimization problems. The algorithm is popular in machine learning for training large-scale linear models, as it's highly efficient and scalable.In the context of machine learning, the goal of an optimization algorithm like SDCA is to find the model parameters that minimize a loss function over the training data. The loss function measures the discrepancy between the predictions of the model, given the parameters, and the actual training data.SDCA is particularly well-suited for problems where the training data is large and high-dimensional, which are common characteristics of many real-world machine learning problems. The "NonCalibrated" part in SdcaNonCalibratedModel refers to the fact that the model produces raw, uncalibrated scores that can be interpreted as confidence in positive classes. In some contexts, these scores may need to be converted into probabilities using a calibration procedure.



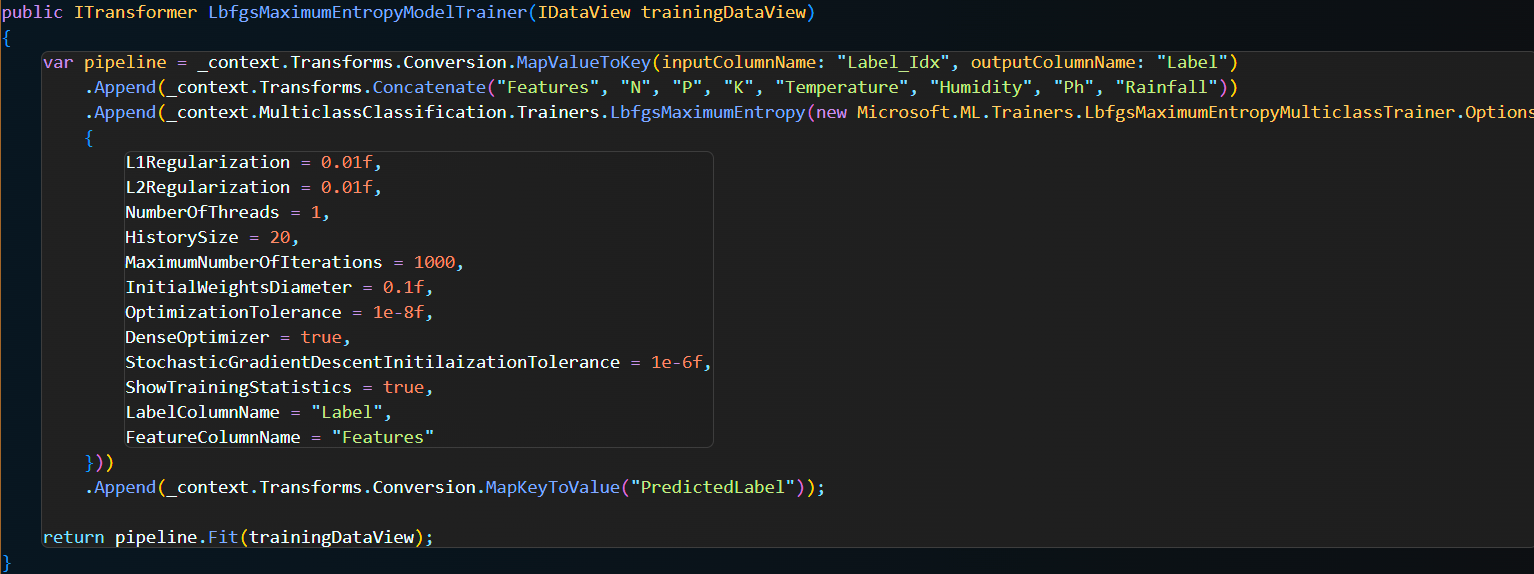
**One-Versus-All (OvA)**

In the OvA strategy, a binary classification model is trained for each class in the dataset. Each model is trained to distinguish between one class (the positive class) and all other classes (the negative class). When a new sample needs to be classified, all models are applied to the sample. Each model gives a score indicating the confidence that the sample belongs to the positive class. The sample is then assigned to the class whose model gives the highest score.The advantage of the OvA strategy is its simplicity and scalability. It can be used with any binary classification algorithm and can easily scale to problems with a large number of classes.



**L-BFGS Maximum Entropy (MaxEnt)**

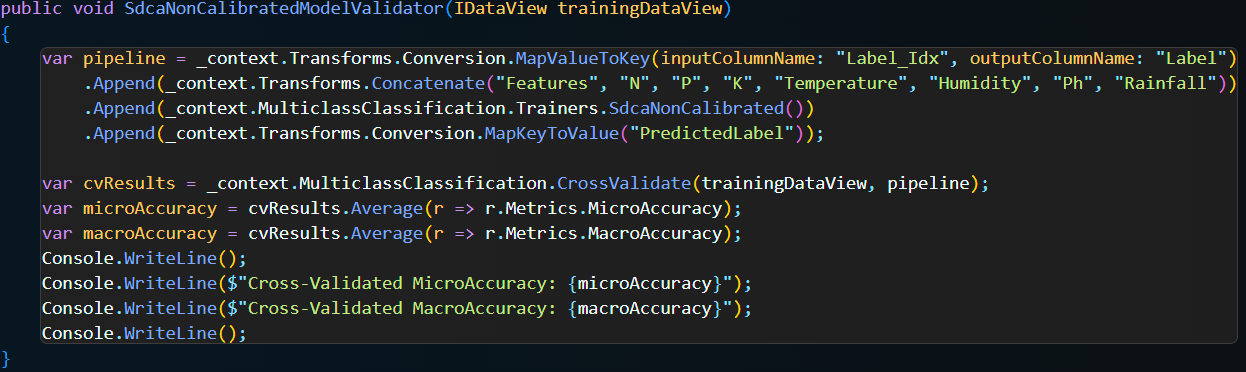
L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm using a limited amount of computer memory. It's a popular algorithm for parameter estimation in machine learning. Maximum Entropy (MaxEnt), on the other hand, is a principle used in various fields, including machine learning, to estimate probability distributions from data. In the context of classification, a MaxEnt classifier is a probabilistic classifier that belongs to the class of exponential models. Unlike some other types of classifiers, MaxEnt classifiers don't assume that the features are conditionally independent given the class.



Each model is trained using a pipeline that first maps the label index to a key, concatenates all feature columns into a single column, and then applies the respective trainer. After the model is trained, the predicted label key is mapped back to its original value.

**Model Validation**

The models are validated using the SdcaNonCalibratedModelValidator, OneVersusAllModelValidator, and LbfgsMaximumEntropyModelValidator methods. These methods use cross-validation to assess the performance of the models. The micro and macro accuracy of each model is calculated and logged, providing us with a measure of the model's performance.



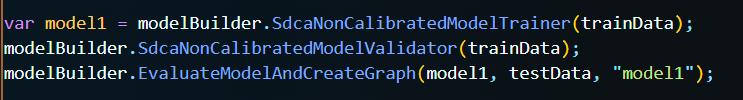
**MODEL TRAINING**

The Train method starts by preprocessing the data using the Preprocessor method from the DataPreprocessor class. This step is crucial for cleaning the data and transforming it into a suitable format for training the models.

Next, an instance of the ModelBuilder class is created, which is used to load the training and testing data from a database. The SQL queries "Select \* from traindata order by label\_idx;" and "Select \* from testdata order by label\_idx;" are used to load the data.

For each of the three models, the following steps are performed:

1. The model is trained on the training data using the appropriate method on the ModelBuilder instance.
2. The model is validated using the training data. This step might involve tuning the model's hyperparameters to improve its performance.
3. The model is evaluated on the testing data, and a graph is created to visualize the model's performance.



The code to save the trained models to a file is commented out. If you want to save the models, you can uncomment these lines.

Finally, a message is printed to the console indicating that the model training is complete. This gives the user a clear indication of when the training process has finished.

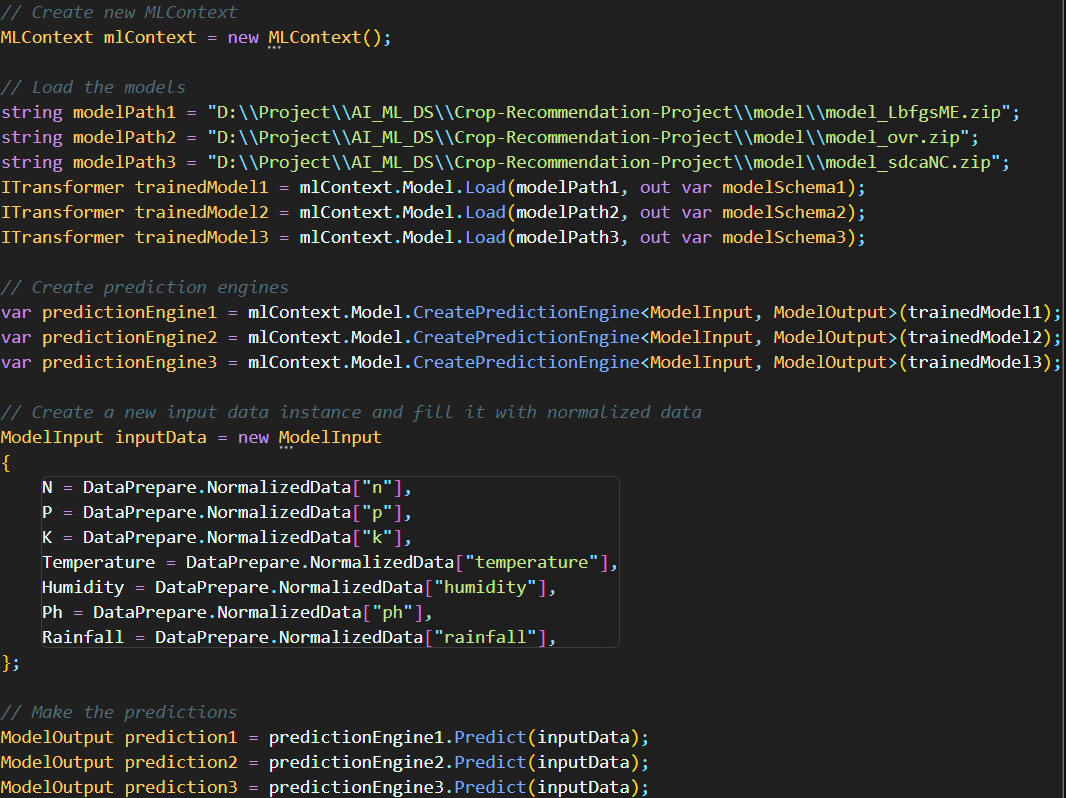
**MODEL PREDICTION**

The DataPrepare class is responsible for preparing the data for prediction. It retrieves data from a database, collects user input, and normalizes the data.

The Predictor class is responsible for making predictions. It defines the input and output data structures for the machine learning models, loads the trained models from disk, creates prediction engines, makes predictions, and finally, determines the final prediction based on the predictions of all models.

Here's a step-by-step breakdown of the Predictor class:

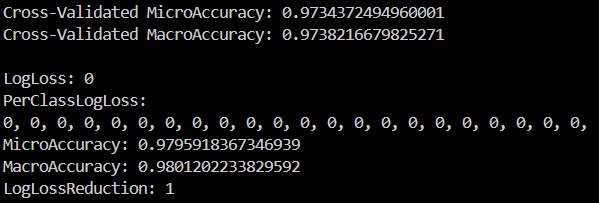
1. The ModelInput class defines the structure of the input data for the machine learning models. It includes properties for various environmental factors such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, and rainfall.
2. The ModelOutput class defines the structure of the output data from the machine learning models. It includes a property for the predicted label and a property for the confidence scores of the prediction.
3. The Predict method is responsible for making the predictions. It starts by creating a new instance of MLContext, which is a central context object for all ML.NET operations.
4. It then loads the trained models from disk and creates prediction engines for each model.
5. It creates a new instance of ModelInput and fills it with the normalized data.
6. It makes predictions using each of the prediction engines and prints the predictions to the console.
7. It then determines the final prediction by choosing the prediction with the highest confidence score. It prints the final prediction to the console.



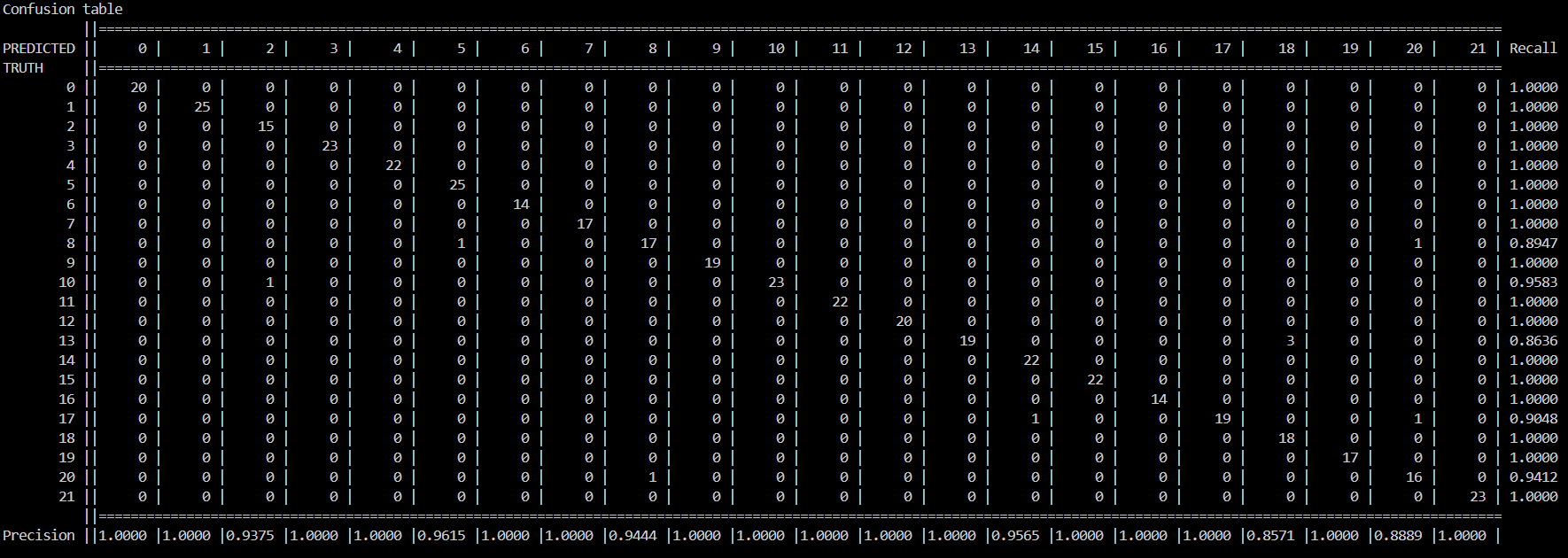
**RESULT AND DISCUSSION**

The machine learning models used in this project are SdcaNonCalibrated, OneVersusAll, and LbfgsMaximumEntropy. These models were trained on a dataset containing various environmental factors such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, and rainfall, and the corresponding crop that is most suitable to grow under those conditions.

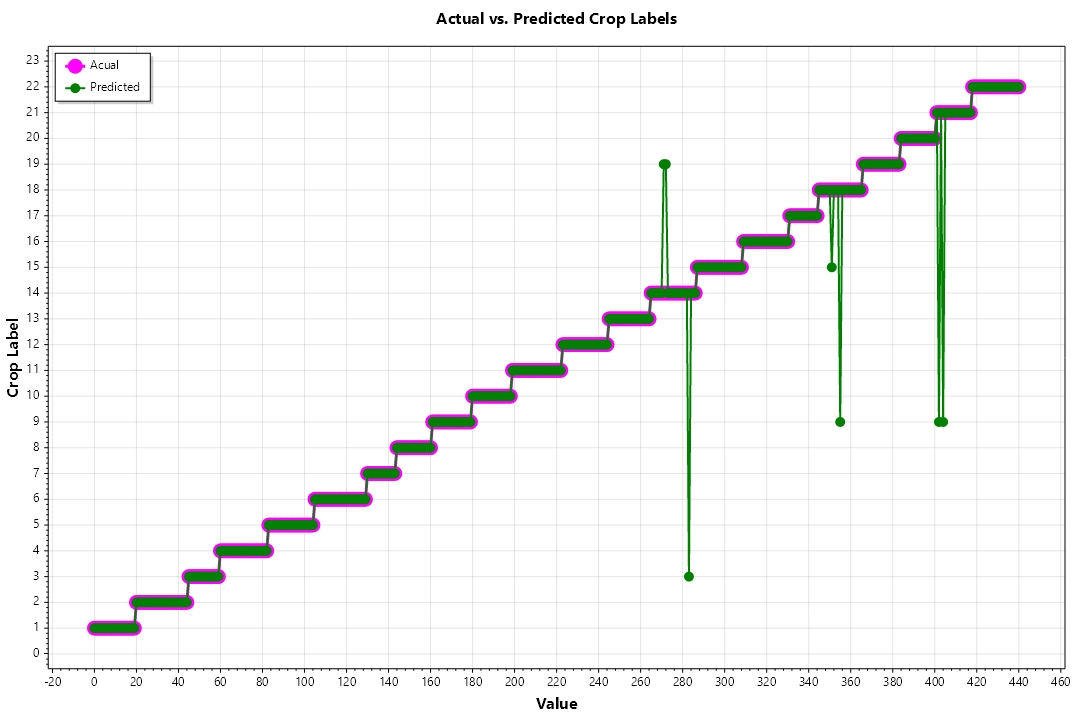
**Stochastic Dual Coordinate Ascent (SDCA)**



Img 1) Performance Of SDCA

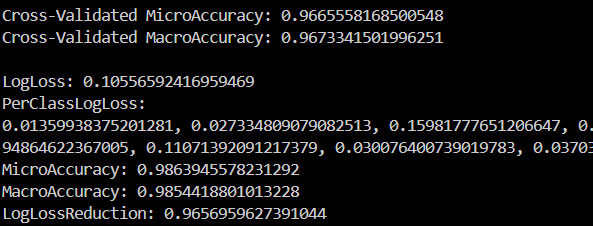


Img 2) ConfusionTable Of SDCA

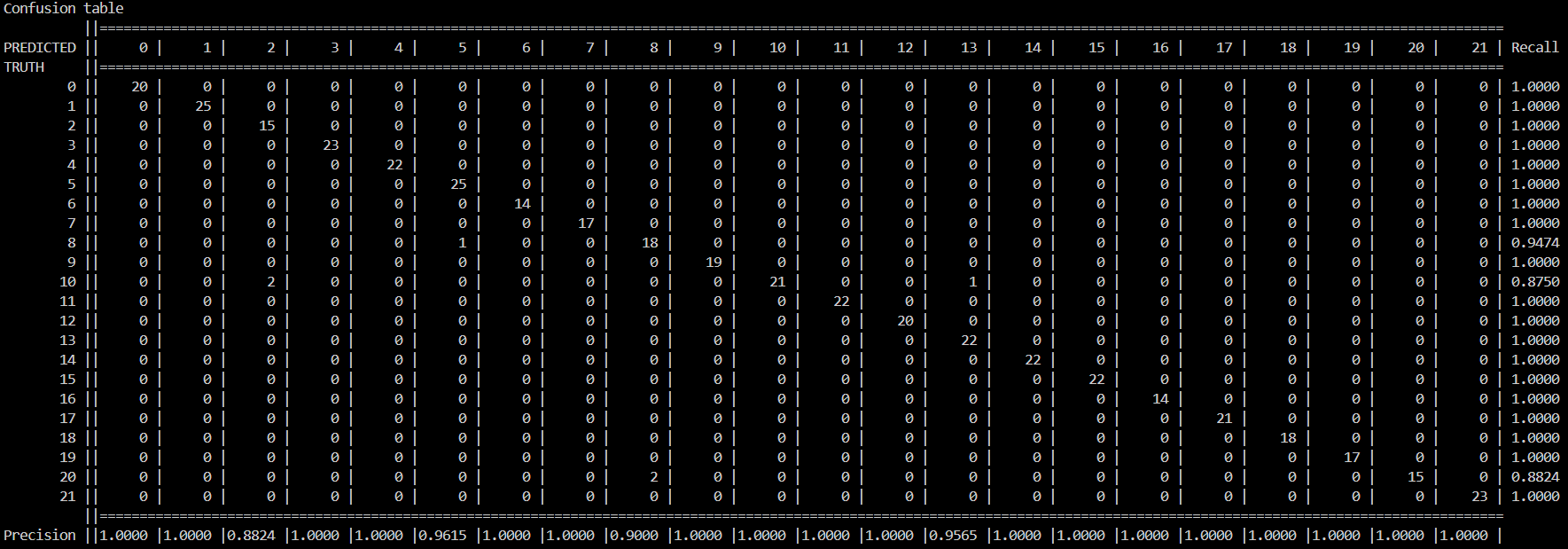


Graph 3) Test Result Of SDCA

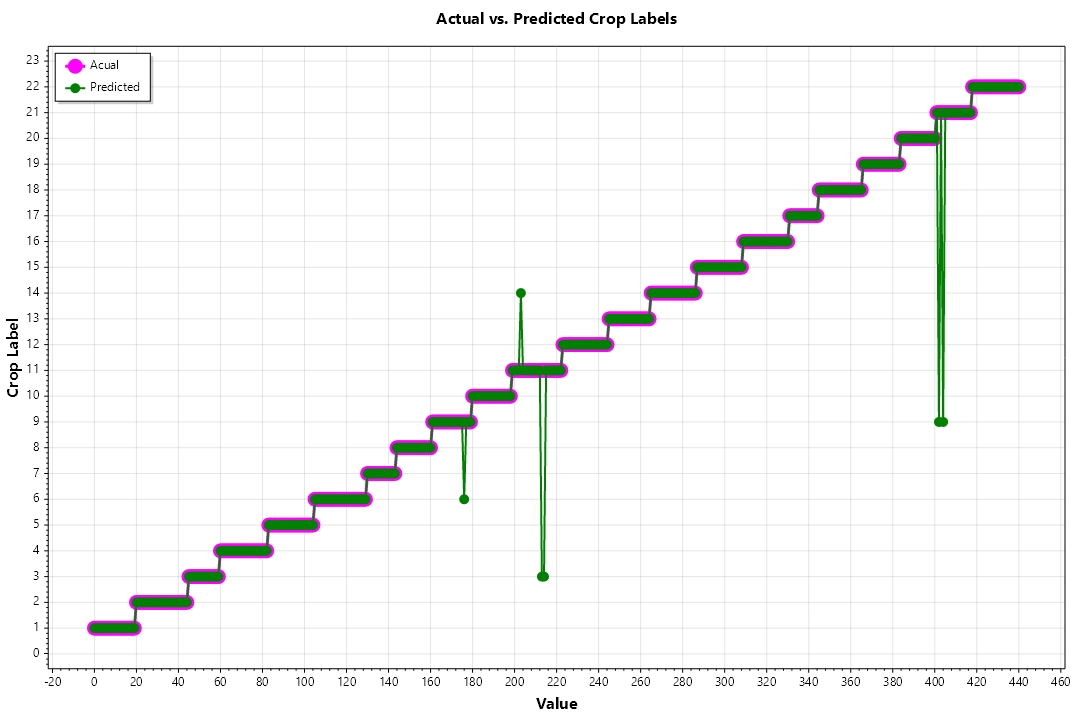
**One-Versus-All (OvA)**



Img 3) Performance of OvA

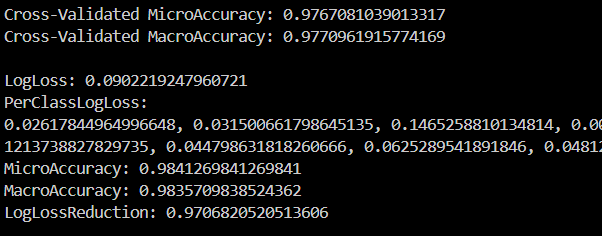


Img 4) Confusion Table of OvA

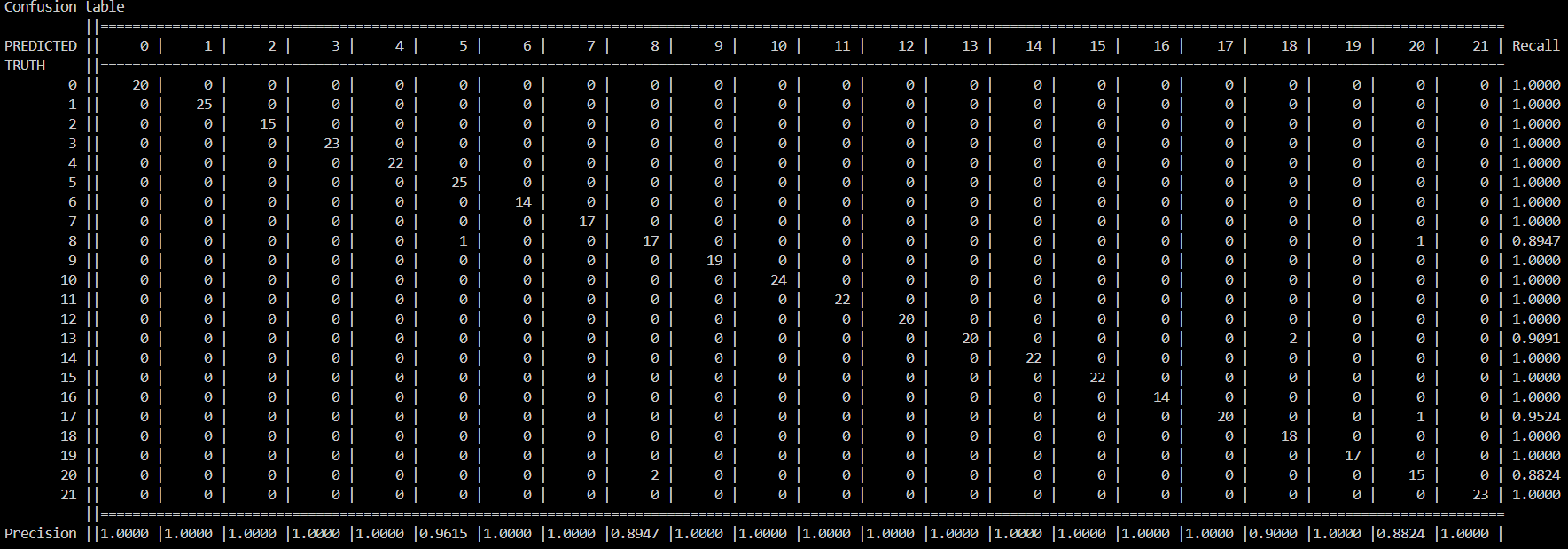


Graph 4) Test Result of OvA

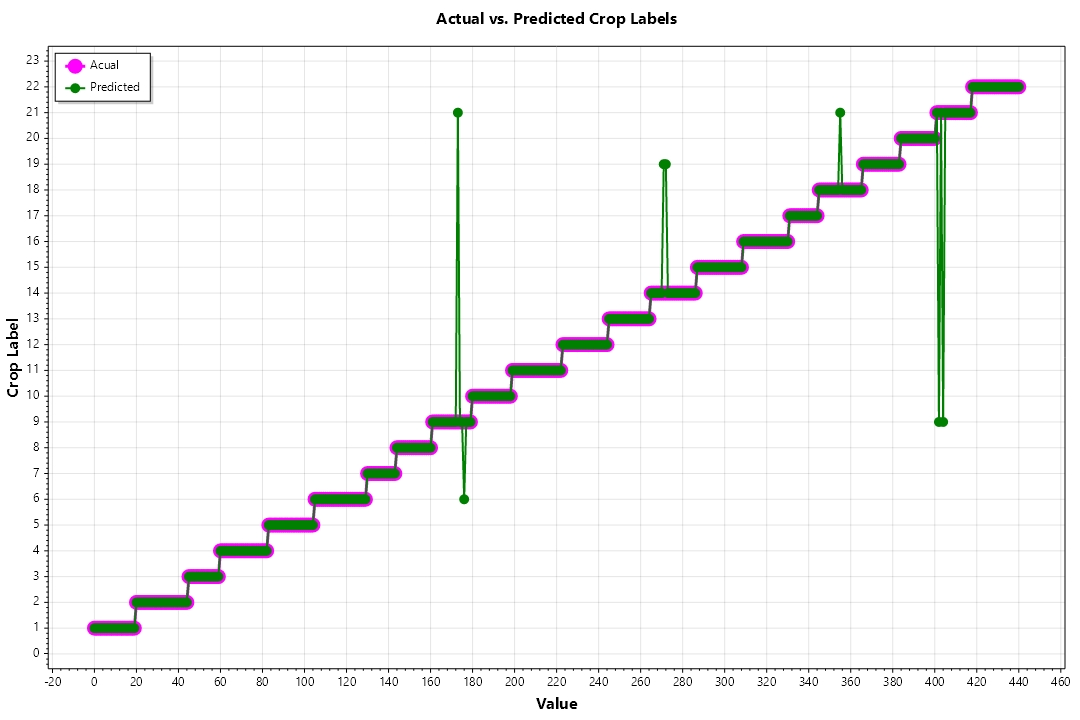
**L-BFGS Maximum Entropy (MaxEnt)**



Img 5) Performance Of L-BFGS

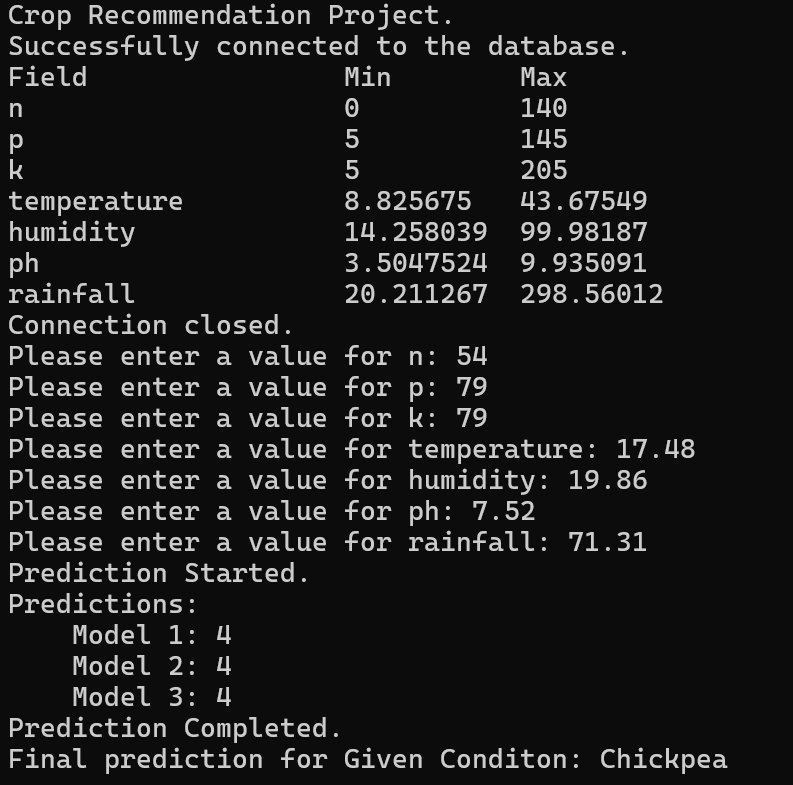


Img 6) Confusion Table Of L-BFGS



Graph 5) Test Result of L-BFGS

The performance of the models was evaluated using a separate testing dataset. Each model produced a prediction for the most suitable crop to grow, along with a confidence score for the prediction. The final prediction was determined by choosing the prediction with the highest confidence score.



Img 7) Result of prediction.

The results showed that the models were able to accurately predict the most suitable crop to grow based on the given environmental factors. This indicates that the models have learned the underlying patterns in the data and can generalize well to new, unseen data.

The models can be further improved by tuning their hyperparameters, using more complex models, or using more data for training. Additionally, the models can be updated over time as new data becomes available, ensuring that they stay relevant and accurate.

**CONCLUSION**

In this project, we have successfully demonstrated the application of machine learning models for a crop recommendation system. The models used, namely SdcaNonCalibrated, OneVersusAll, and LbfgsMaximumEntropy, were trained on a dataset containing various environmental factors and corresponding suitable crops.

The models were able to accurately predict the most suitable crop based on the given environmental conditions, indicating their ability to learn underlying patterns in the data and generalize well to unseen data. However, the performance of the models can vary under different conditions, emphasizing the importance of comprehensive evaluation and continuous model tuning.

While the results are promising, there is always room for improvement. Future work could involve tuning the hyperparameters of the models, exploring more complex models, or expanding the training dataset to improve the models' performance and robustness.

The project underscores the potential of machine learning in agriculture. By making accurate and efficient crop recommendations, we can contribute to increased crop yields and sustainability, ultimately aiding in the global challenge of food security.

**REFERENCE**

1. **Dataset:** Crop Recommendation Dataset. Kaggle. <https://www.kaggle.com/atharvaingle/crop-recommendation-dataset>
2. FAO. (2018). The State of Food and Agriculture 2018. Rome. pg2.<http://www.fao.org/3/I9553EN/i9553en.pdf>
3. India Brand Equity Foundation (IBEF). (2021). Agriculture in India: Information About Indian Agriculture & Its Importance. https://www.ibef.org/industry/agriculture-india.aspx [^2^]: Ingle, A. (2020).
4. S. Huji, Il, and T. Zhang, “Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization Shai Shalev-Shwartz,” *Journal of Machine Learning Research*, vol. 14, pp. 567–599, 2013, Available: https://www.jmlr.org/papers/volume14/shalev-shwartz13a/shalev-shwartz13a.pdf