



MKSSS's Cummins College of Engineering for Women, Pune

Academic Year : 2024 - 25

OPEN ENDED ASSIGNMENT

23PCEC403L : Machine Learning Lab

OEA Group No. : 04

UEC Number	Name
UEC2023251	Ishita Rane
UEC2023255	Swara Saoji

Build and evaluate a machine learning model to assist farmers by forecasting weather-based conditions and predicting whether a specific crop can be successfully grown.

1. Introduction :

Agriculture is the backbone of many economies, especially in regions with large rural populations. However, poor crop selection based on unsuitable soil and climatic conditions often leads to reduced productivity and economic loss. With the rise of **precision agriculture**, farmers are increasingly relying on data-driven strategies to improve their farming decisions.

This project introduces a **machine learning-based crop recommendation system** that predicts whether a specific crop can be successfully cultivated under given environmental conditions. Key input features include **nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, and rainfall**.

The primary objective is to help farmers and agricultural planners make **informed, sustainable decisions** to boost crop yield. By incorporating **visualization techniques** for feature analysis and model evaluation, the system ensures **transparency and interpretability**. Overall, the project showcases how machine learning can support smarter, climate-resilient agriculture.

2. Data Collection and Processing :

We used following open-source dataset from **Kaggle**:

<https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset>

This dataset contains **2200 records** with **7 different agricultural and environmental parameters** that are important for determining the most suitable crop to cultivate. The features are:

- **N** - Nitrogen content in soil
- **P** - Phosphorous content in soil
- **K** -Potassium content in soil
- **temperature** - temperature in degree Celsius
- **humidity** - relative humidity in percentage(%)
- **ph** - ph value of the soil
- **rainfall** - rainfall in millimeters (mm)

The target variable (label) represents the recommended crop to grow under the given conditions. There are **22 unique crops** in the dataset, including **rice, maize, chickpea, wheat, apple, banana**, among others.

To prepare the dataset for training machine learning models, several preprocessing steps were performed, including feature selection, train-test splitting, and feature scaling.

- **Train-Test Split:**

To evaluate model performance on unseen data, we split the dataset into training and testing sets using an **70-30 ratio**. A **random_state value of 27** ensured reproducibility of the results.

- **Feature Scaling:**

Feature scaling was applied using **StandardScaler** to normalize the input features. This step is crucial for algorithms like logistic regression and Naive Bayes, which are sensitive to the scale of the input data. Standardization helps in achieving better model convergence and balanced performance.

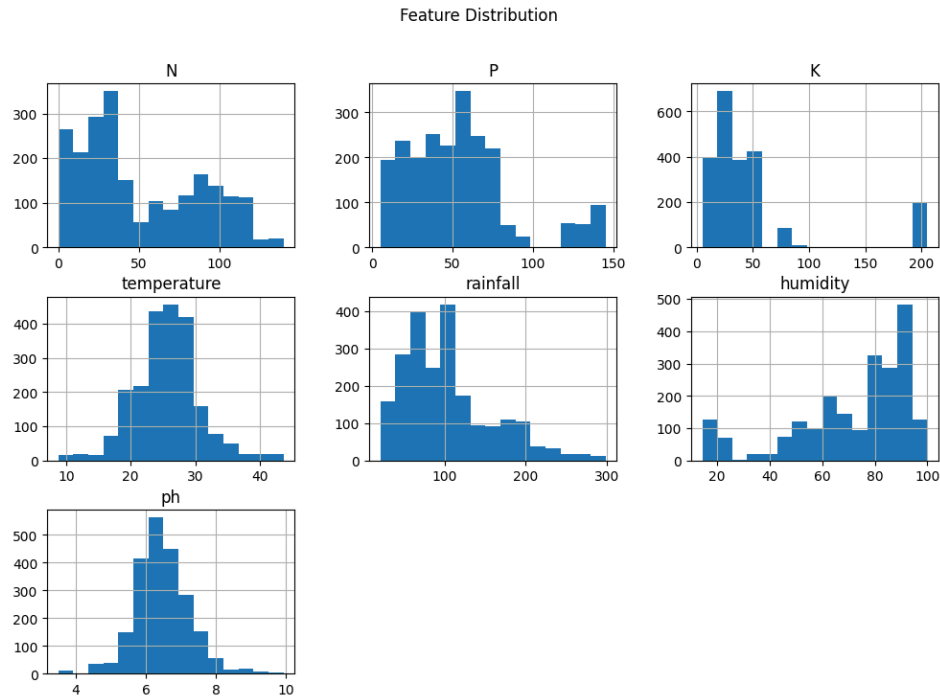
3. Visualization :

- **Histogram (Feature Distribution) :**

Histograms were used to visualize the distribution of individual features such as Nitrogen (N), Phosphorous (P), Potassium (K), temperature, and rainfall. These plots provided insights into:

- The range and spread of each feature across the dataset
- Skewness in the data distributions
- The presence of outliers or values that deviated significantly from the mean

This analysis was useful in understanding the common soil and climatic conditions that support various crops. For example, some crops were observed to grow within a narrow temperature range, while others tolerated wider nutrient variations.

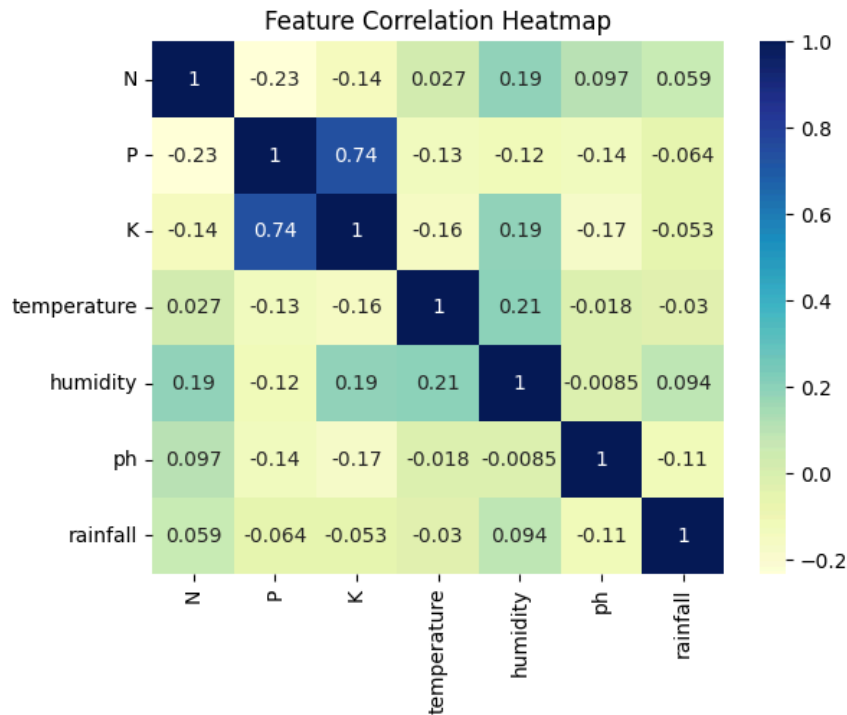


- **Heatmap (Correlation Matrix)**

A correlation heatmap was generated to examine the pairwise relationships between all numerical features. This visualization:

- Highlighted positive and negative correlations between variables such as N, P, K, temperature, humidity, and rainfall
- Helped detect multicollinearity, particularly among soil nutrients, which can influence certain machine learning algorithms (e.g., linear models)
- Provided guidance in feature selection, allowing us to prioritize features with high predictive relevance and minimal redundancy

The heatmap confirmed that while some soil nutrients are moderately correlated, most environmental features were sufficiently independent to be included together in modeling.

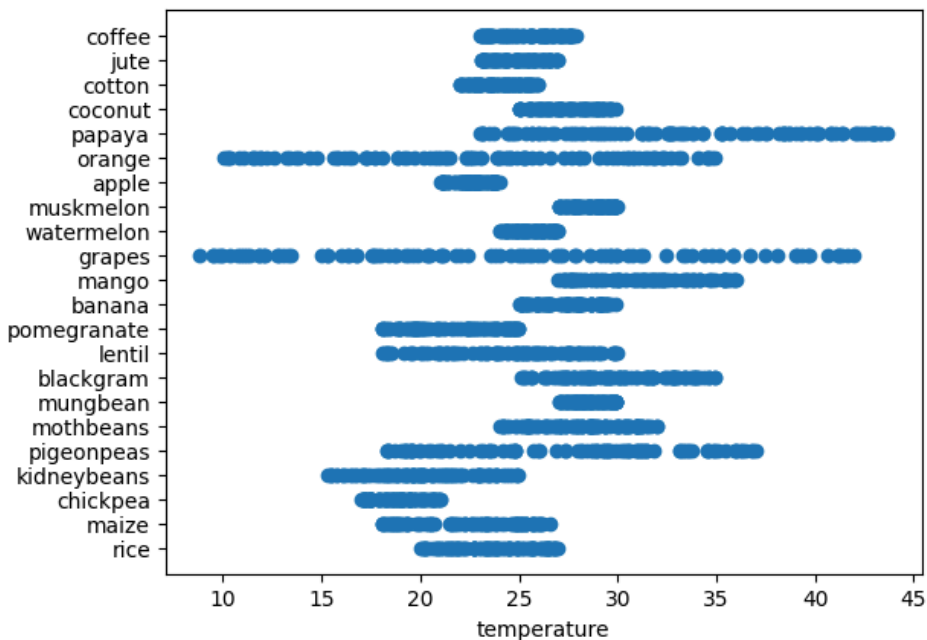


- **Scatter Plot**

To explore how individual environmental features relate to crop types, we used a scatter plot to visualize the distribution of **rainfall,temperature** across different **crop labels**. This visualization allowed us to:

- Observe how **rainfall levels,temperature levels** vary for different crops
- Identify **overlapping and distinct rainfall ranges** across crop categories
- Detect possible **clusters or outliers**, which may influence model performance

This plot provided early insight into how **rainfall,temperature** might contribute to crop classification. For more complex interactions, pairwise feature relationships were further explored using Seaborn's pairplot.



4. Feature Selection :

Feature selection is a critical step in improving model performance, reducing complexity, and avoiding overfitting. During the preprocessing phase, we carefully analyzed the dataset to determine which features contributed most effectively to crop classification. This process involved:

- Generating a correlation heatmap to assess linear relationships between input features and the target variable.
- Conducting visual inspections (e.g., histograms and scatter plots) to evaluate the distribution and variation of features across different crop labels.

Dropped Features

Based on our analysis, the following features were excluded from model training due to limited predictive value:

- **Rainfall**

- Demonstrated low correlation with the crop label.
- Exhibited uniform or inconsistent values across many records, reducing its effectiveness in learning useful patterns.
- Excluding this feature simplified the model without negatively affecting accuracy.
- **pH**

While pH is important in certain agricultural applications, in this dataset it showed minimal variation across crop types.

- Visualizations confirmed its limited role, making it a non-essential feature for our classification task.

Final Selected Features

The following features were retained for model training, as they showed strong relevance to crop classification based on both statistical and visual analysis:

- N (Nitrogen)
 - P (Phosphorous)
 - K (Potassium)
 - Temperature
 - Humidity
-

5. Model Selection :

Final Choice: Naive Bayes

After evaluating multiple machine learning models, we selected **Naive Bayes** as the final model for crop prediction based on its superior performance and suitability for the task. The reasons for choosing Naive Bayes are as follows:

- **Highest Accuracy:** Naive Bayes achieved the highest accuracy (**95%**) compared to all other tested models, demonstrating its ability to correctly predict crop labels from

environmental features.

- **Balanced Performance:** It performed well across multiple evaluation metrics, including **precision, recall, and F1-score**, indicating a well-rounded classification model with minimal bias.
- **Resilience to Overfitting:** Unlike the **Decision Tree** model, Naive Bayes is **less prone to overfitting**, especially in cases where there are many features or limited training data.
- **Simplicity and Interpretability:** Naive Bayes is a **simple and interpretable** model, making it ideal for stakeholders such as farmers who require clear, understandable recommendations for crop planting.
- **Efficiency:** The model is **fast** to train and deploy, making it a practical choice for large-scale agricultural datasets, where computational efficiency is crucial.
- **Suitability for Multi-Class Classification:** Naive Bayes works well with **categorical outputs**, which is particularly suited for our multi-class crop prediction task, where each record corresponds to one of 22 unique crop labels.

6. Model Description / Algorithm :

Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem, which calculates the posterior probability of each class given the input features. The model relies on the "naive" assumption that all features are independent of each other, which simplifies computation. Despite this assumption, Naive Bayes often performs surprisingly well in practice, especially for classification tasks where features are not highly correlated.

It is particularly useful for multi-class classification problems and has been shown to deliver strong results in many real-world applications, including text classification, medical diagnoses, and in this case, agricultural crop prediction.

Algorithm Steps:

1. **Input Data:**
The cleaned and preprocessed dataset, consisting of the selected features X and the corresponding target labels y , is used for training the model.

2. Initialize Model:

We initialize the Naive Bayes classifier. In this case, we use the Gaussian Naive Bayes variant, which assumes that the features follow a Gaussian (normal) distribution.

```
from sklearn.naive_bayes import GaussianNB

classifier_nb = GaussianNB()
```

3. Model Training:

The model is trained using the training dataset (xtrain, ytrain).

```
classifier_nb.fit(xtrain, ytrain)
```

4. Prediction:

After training, we use the model to predict the target labels for the test data (xtest).

```
y_pred_nb = classifier_nb.predict(xtest)
```

Evaluation Metrics:

The model's performance is evaluated using the following metrics:

- **Accuracy:** Overall proportion of correctly classified instances.
- **Precision (weighted):** The ability of the model to correctly identify crops, averaged across all classes.
- **Recall (weighted):** The ability of the model to find all relevant instances, averaged across all classes.
- **F1-score (weighted):** The harmonic mean of precision and recall, providing a balanced measure of the model's performance across all classes.

Classification Report (Naive Bayes):				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	28
banana	1.00	1.00	1.00	22
blackgram	0.85	0.87	0.86	39
chickpea	1.00	1.00	1.00	35
coconut	1.00	0.97	0.98	29
coffee	1.00	1.00	1.00	30
cotton	1.00	1.00	1.00	33
grapes	1.00	1.00	1.00	30
jute	0.69	0.72	0.71	25
kidneybeans	1.00	1.00	1.00	29
lentil	0.77	0.89	0.83	27
maize	1.00	1.00	1.00	34
mango	1.00	1.00	1.00	31
mothbeans	0.95	0.95	0.95	38
mungbean	1.00	1.00	1.00	29
muskmelon	1.00	1.00	1.00	29
orange	1.00	1.00	1.00	27
papaya	1.00	1.00	1.00	19
pigeonpeas	0.91	0.79	0.85	39
pomegranate	0.97	1.00	0.98	31
rice	0.80	0.78	0.79	36
watermelon	1.00	1.00	1.00	20

Why Not the Other Models?

- **Logistic Regression**

- Primarily designed for **binary classification** tasks, making it less suitable for our multi-class crop prediction problem.
- Achieved slightly lower accuracy (**91%**) than Naive Bayes.
- Assumes **linear decision boundaries**, which may not be ideal for the crop prediction task, as the relationships between environmental factors and crop types could be more complex.

- **Decision Tree**

- Achieved lower accuracy (**94%**) with a **maximum depth of 6**.
- **Prone to overfitting**, especially with noisy or limited data, which could negatively impact generalization to unseen data.
- Showed **less consistency** across cross-validation folds, suggesting that it may not be as robust or reliable as Naive Bayes for this task.

7. Testing and Evaluation of Model :

To evaluate the performance of our Naive Bayes model, we employed several key metrics to assess its predictive accuracy and robustness across the multi-class crop prediction task.

1. Confusion Matrix :

- The confusion matrix provides a detailed view of model performance by displaying the number of correct and incorrect predictions for each crop class.
- It helped us identify which crops were frequently misclassified and how well the model differentiated between similar crop types.
- By visualizing the matrix, we gained a clear understanding of the model's ability to classify various crops and detected any imbalances or confusion between certain crop classes.

Confusion Matrix (Naive Bayes):

```
[[28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 34  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  1  0  0]
 [ 0  0  0 35  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 18  0  0  0  0  0  0  0  0  0  0  0  7]
 [ 0  0  0  0  0  0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  0  0  0  0  0  0  0  0 24  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 34  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 31  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 36  0  0  0  2  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 19  0  0]
 [ 0  0  3  0  0  0  0  0  0  0  0  3  0  0  2  0  0  0  0 31  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 31  0  0]
 [ 0  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0 28  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 20]]
```

2. Accuracy Score :

- The accuracy score measures the percentage of correct predictions over the total number of predictions made.
- Our Naive Bayes model achieved a high accuracy of **95%**, demonstrating its strong and reliable performance on unseen data. This indicates that the model is highly effective in predicting the correct crop label for the majority of cases.

3. Precision, Recall, and F1-Score :

- Precision: This metric measures the proportion of correctly predicted crops out of all crops predicted by the model. It answers the question, "Of the crops predicted, how many were actually correct?"
- Recall: Recall measures the proportion of correctly predicted crops out of all the actual crops in the dataset. It addresses the question, "Of the crops that were actually present, how many did the model identify?"
- F1-Score: The F1-score is the harmonic mean of precision and recall, offering a balanced metric that considers both false positives and false negatives, which is crucial in a multi-class setting like ours.



These metrics provided deeper insight into the model's performance, especially in a multi-class classification task where crop types are diverse and may be unevenly distributed. They allowed us to understand not only how accurate the model is but also how well it handles class

imbalances and the trade-off between false positives and false negatives.

8. Outcome/Results:

To make the model actionable for users such as farmers or agricultural advisors, we implemented an interactive input system where users can enter environmental parameters (e.g., N, P, K, temperature, rainfall, etc.) specific to their farmland. If any value is not available, the system defaults to the average value for that feature, ensuring flexibility and ease of use.

Prediction Functionality:

- Users are prompted to enter values for each selected environmental feature (or use dataset averages by pressing Enter).
- The model then predicts the most suitable crop for the provided conditions.
- To validate the recommendation, users can also enter the crop they plan to grow.
- The system compares the **predicted crop** with the **intended crop**, and returns:
 -  **Yes**, if the user's planned crop matches the model's recommendation.
 -  **No**, if the model does not recommend the intended crop based on the input conditions

```
print('\nEnter the following values (or press enter to use average):')
```

```
user_input = []
```

```
for feature in features:
```

```
    value = input(f"{feature}: ")
```

```
    if value:
```

```
        user_input.append(float(value))
```

```
    else:
```

```
        user_input.append(feature_means[feature])
```

Enter the following values (or press enter to use average):

N: 90

P: 42

K: 43

temperature: 20

humidity: 82

```
r_input_np = np.array(user_input).reshape(1, -1)
predicted_crop = classifier_tree.predict(user_input_np)
```

```
crop_input = input("\nEnter the crop you plan to grow: ")
if predicted_crop[0].lower() == crop_input.lower():
    print("✅ Yes, this crop is suitable for your conditions.")
else:
    print("❌ No, this crop is not recommended based on your input.")
```

Enter the crop you plan to grow: rice

✅ Yes, this crop is suitable for your conditions.

Significance:

- This functionality demonstrates the **practical application** of the model in **real-world decision-making**.
- It supports users in making **data-driven crop choices**, potentially improving yield, sustainability, and profitability.
- The interactive nature also helps users understand how different environmental factors influence crop suitability.

9. Conclusion:

The Crop Recommendation System effectively predicts the most suitable crop based on key soil and environmental parameters using machine learning. After testing Logistic Regression, Decision Tree, and Naive Bayes, Naive Bayes achieved the highest accuracy of 95% and was chosen as the best-performing model.

Visual tools like heatmaps and scatter plots helped in selecting relevant features and improving model interpretability. Additionally, an interactive prediction system was built using the Decision Tree model to assist users in real-time crop validation.

Overall, the system is fast, reliable, and practical for supporting sustainable and data-driven farming decisions.