Final Project

Recommandation System:
Crop and Fertilizer Recommender

INFO7390 Advance Data Science
Fall 2023

Presented by
Nainil Maladkar | NUID 002780019
Simran Nagpurkar | NUID 002922747

OUTLINE

- DATASET
- EDA for Crop and Fertilizer
- FEATURE SCALING
- BASELINE MODEL Implementation
  - Naïve Bayes for Crop Data
  - Random Forest for Fertilizer Data
- Advance Model Implementation
  - Convolutional Neural Network
  - MLPC Classifier Neural Network
- FLASK APPLICATION
- RESULTS
- CONCLUSION
- ACKNOWLEDGEMENT

DATASET

Crops Recommendation dataset: Case Study on Kaggle Competition

Fertilizers Recommendation dataset: Github:Yash Thorbole

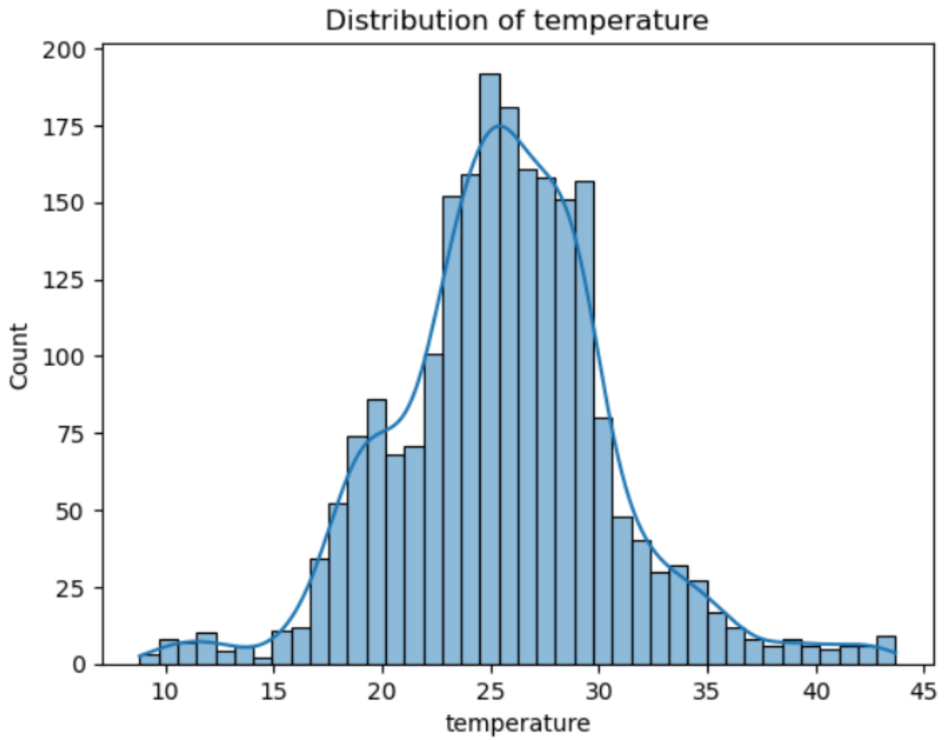
Nitrogen (N): Ranges from 0 to 140 with a mean of around 50.55.
Phosphorus (P): Ranges from 5 to 145 with a mean of approximately 53.36.
Potassium (K): Has a wide range from 5 to 205, average near 48.15.
Temperature: Varies from 8.83°C to 43.68°C, average around 25.62°C.
Humidity: Ranges widely from 14.26% to nearly 100%, with an average of 71.48%.
pH: Varies from 3.50 to 9.94, with a mean value close to 6.47, which is slightly acidic.
Rainfall: Ranges from 20.21 mm to 298.56 mm, with an average of 103.46 mm.

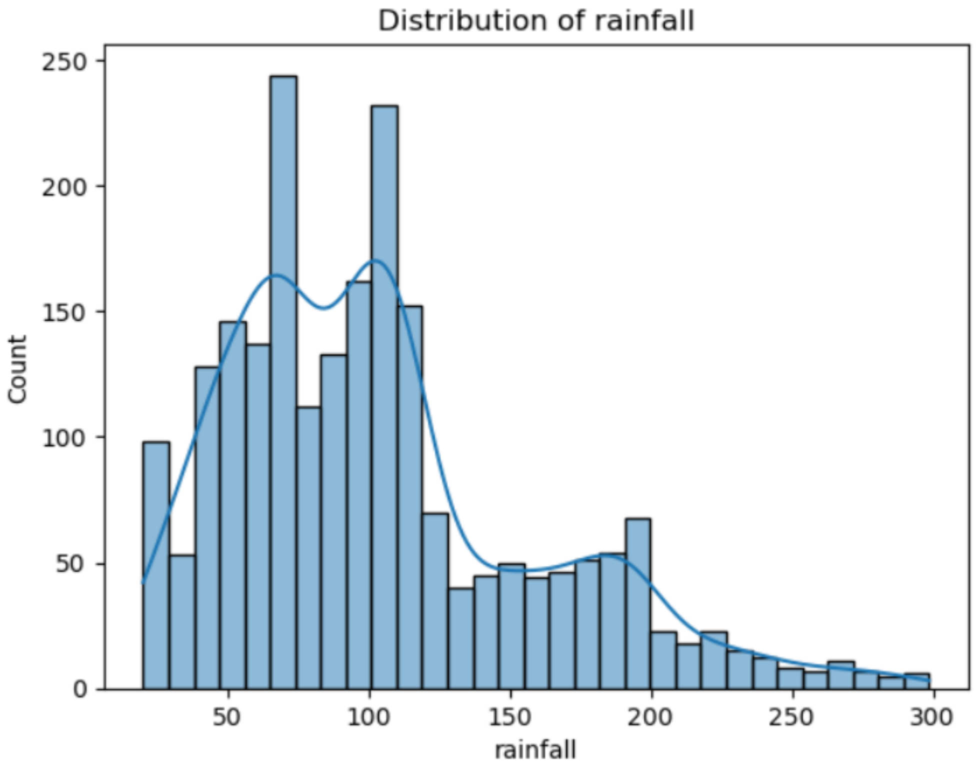
	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Fertilizer:
Urea: Contains 37% Nitrogen, 0% Potassium, and 0% Phosphorous.
DAP (Diammonium phosphate): It contains 12% Nitrogen, 0% Potassium, and 36% Phosphorous.
Fourteen-Thirty Five-Fourteen: It contains 7% Nitrogen, 9% Potassium, and 30% Phosphorous.
Twenty Eight-Twenty Eight: It contains 22% Nitrogen, 0% Potassium, and 20% Phosphorous.
Seventeen-Seventeen-Seventeen: Contains 17% Nitrogen, 17% Potassium, and 17% Phosphorous.
Ten-Twenty Six-Twenty Six: Comprises 10% Nitrogen, 26% Potassium, and 26% Phosphorous.

	Nitrogen	Potassium	Phosphorous	Fertilizer Name
0	37	0	0	Urea
1	12	0	36	DAP
2	7	9	30	Fourteen-Thirty Five-Fourteen
3	22	0	20	Twenty Eight-Twenty Eight
4	35	0	0	Urea

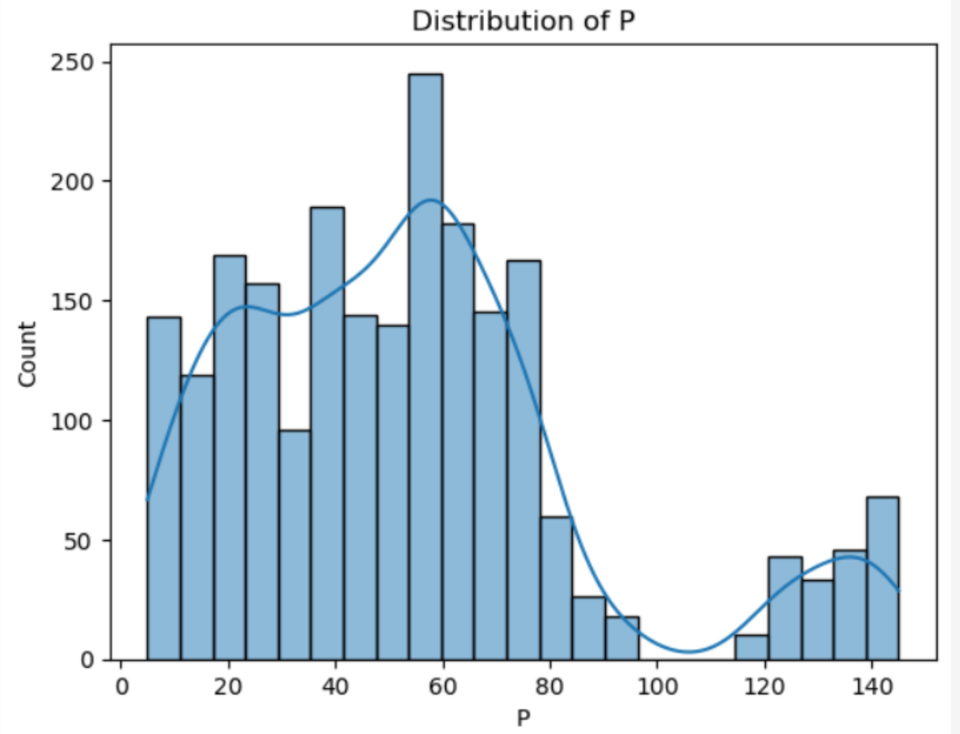
Exploratory Data Analysis for Crop Dataset

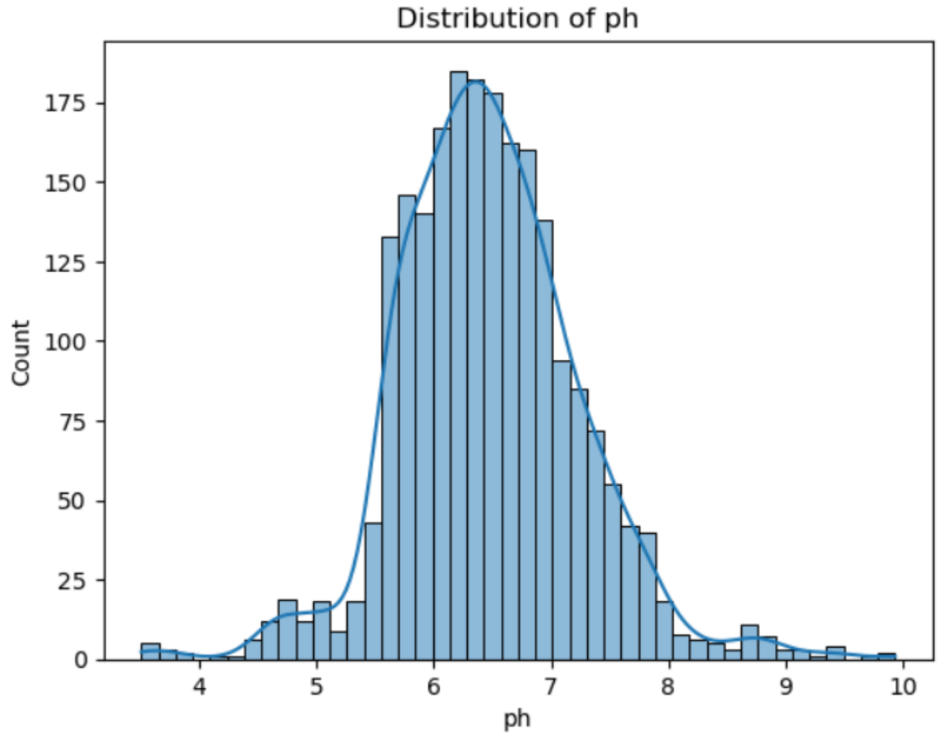




Variable	Description
N, P, K	These soil nutrients show varied distributions. Some display a bimodal nature (having two peaks), suggesting different groups in the data.
Temperature	Appears to be normally distributed.
Humidity	Shows a left-skewed distribution, with a high frequency of values towards the higher end.
pH	This is fairly normally distributed, slightly leaning towards acidic values (less than 7).
Rainfall	Displays a right-skewed distribution, indicating that higher rainfall amounts are less common.

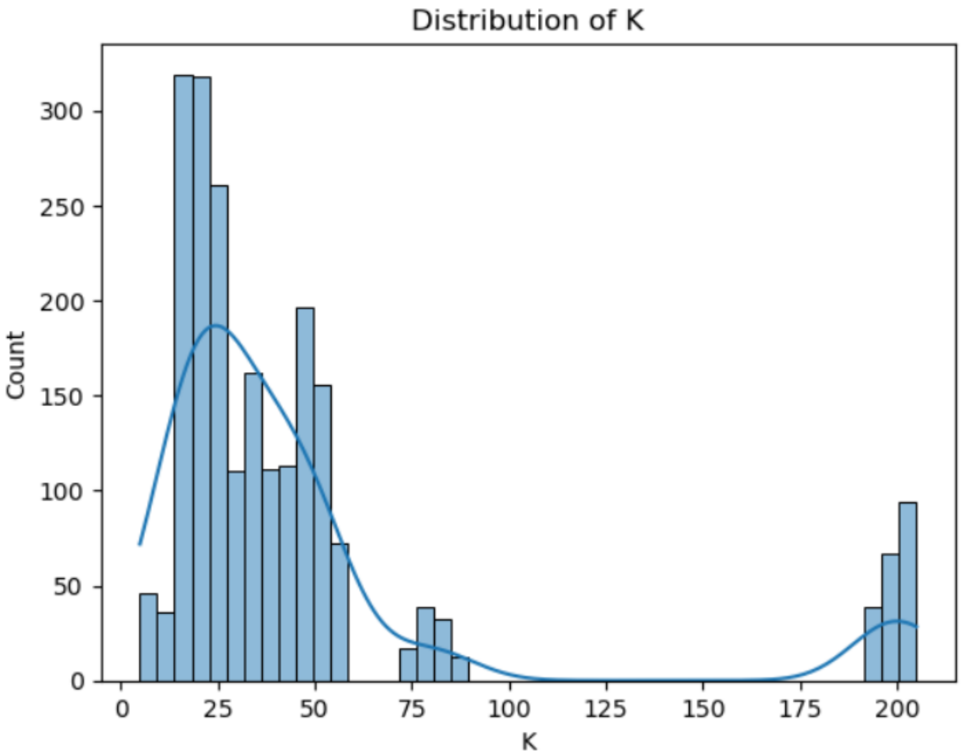
Exploratory Data Analysis for Crop Dataset

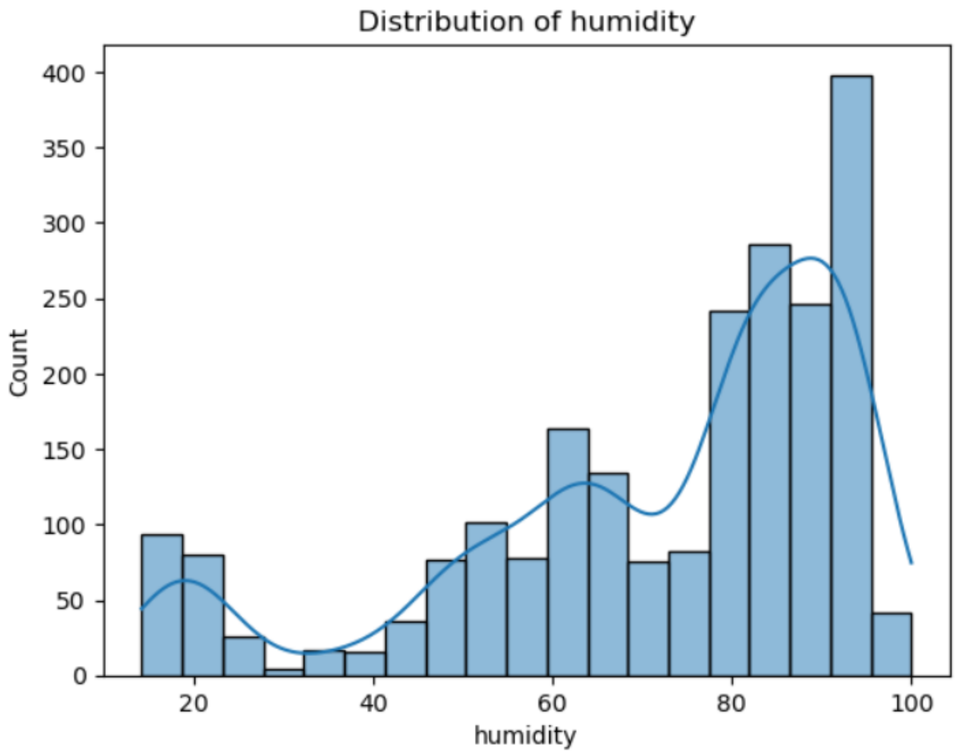




Variable	Description
N, P, K	These soil nutrients show varied distributions. Some display a bimodal nature (having two peaks), suggesting different groups in the data.
Temperature	Appears to be normally distributed.
Humidity	Shows a left-skewed distribution, with a high frequency of values towards the higher end.
pH	This is fairly normally distributed, slightly leaning towards acidic values (less than 7).
Rainfall	Displays a right-skewed distribution, indicating that higher rainfall amounts are less common.

Exploratory Data Analysis for Crop Dataset



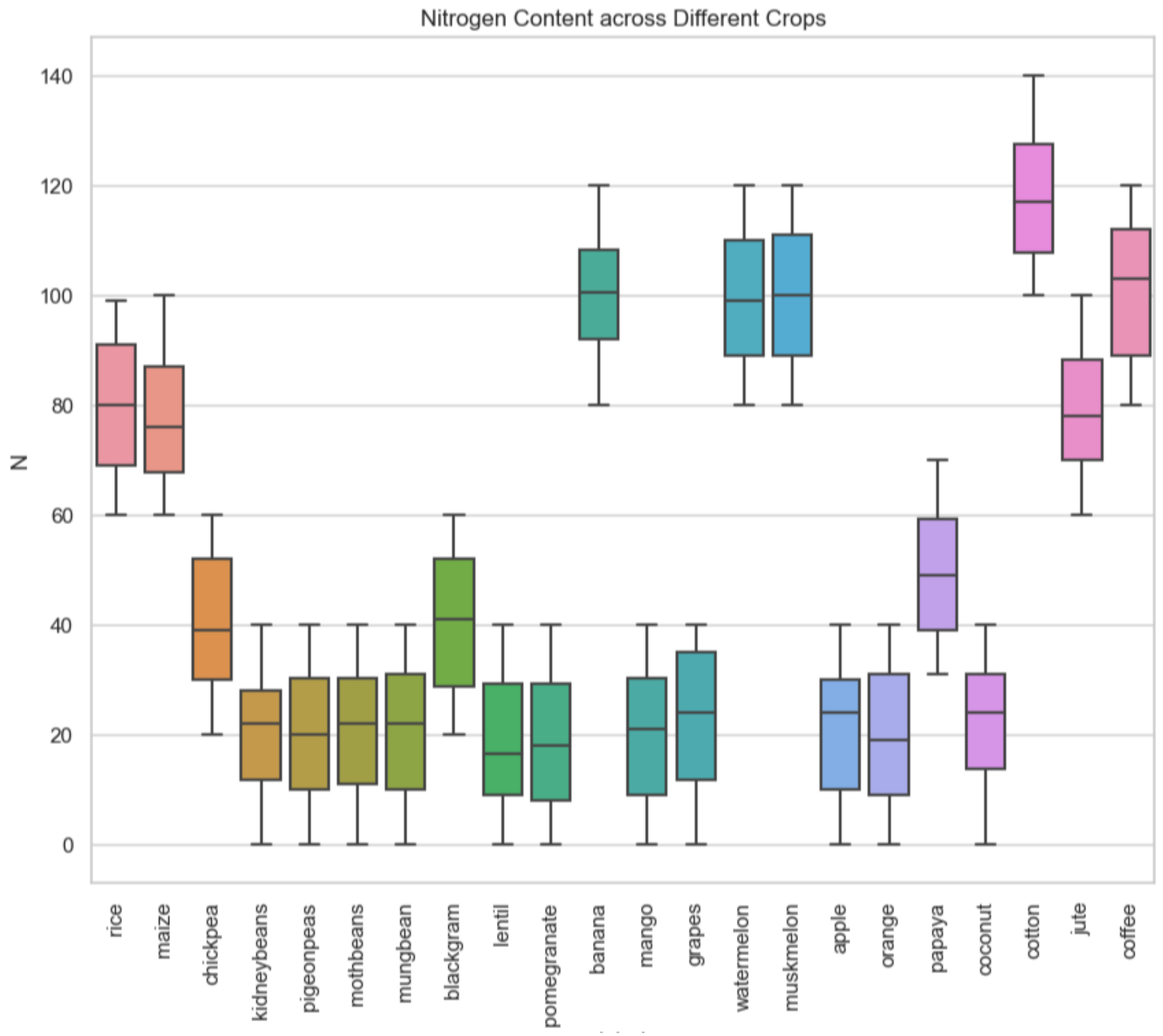


Variable	Description
N, P, K	These soil nutrients show varied distributions. Some display a bimodal nature (having two peaks), suggesting different groups in the data.
Temperature	Appears to be normally distributed.
Humidity	Shows a left-skewed distribution, with a high frequency of values towards the higher end.
pH	This is fairly normally distributed, slightly leaning towards acidic values (less than 7).
Rainfall	Displays a right-skewed distribution, indicating that higher rainfall amounts are less common.

Northeastern University

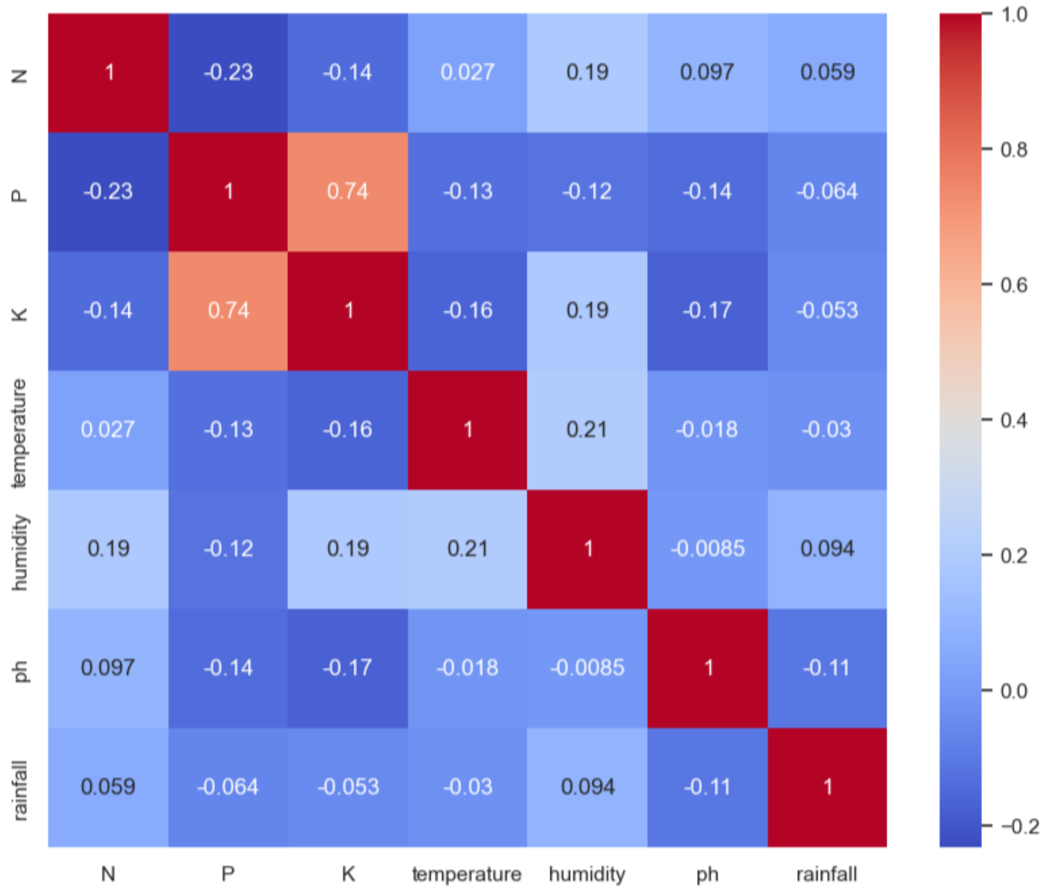
Exploratory Data Analysis for Crop Dataset

Bivariate Analysis with Data Visualization

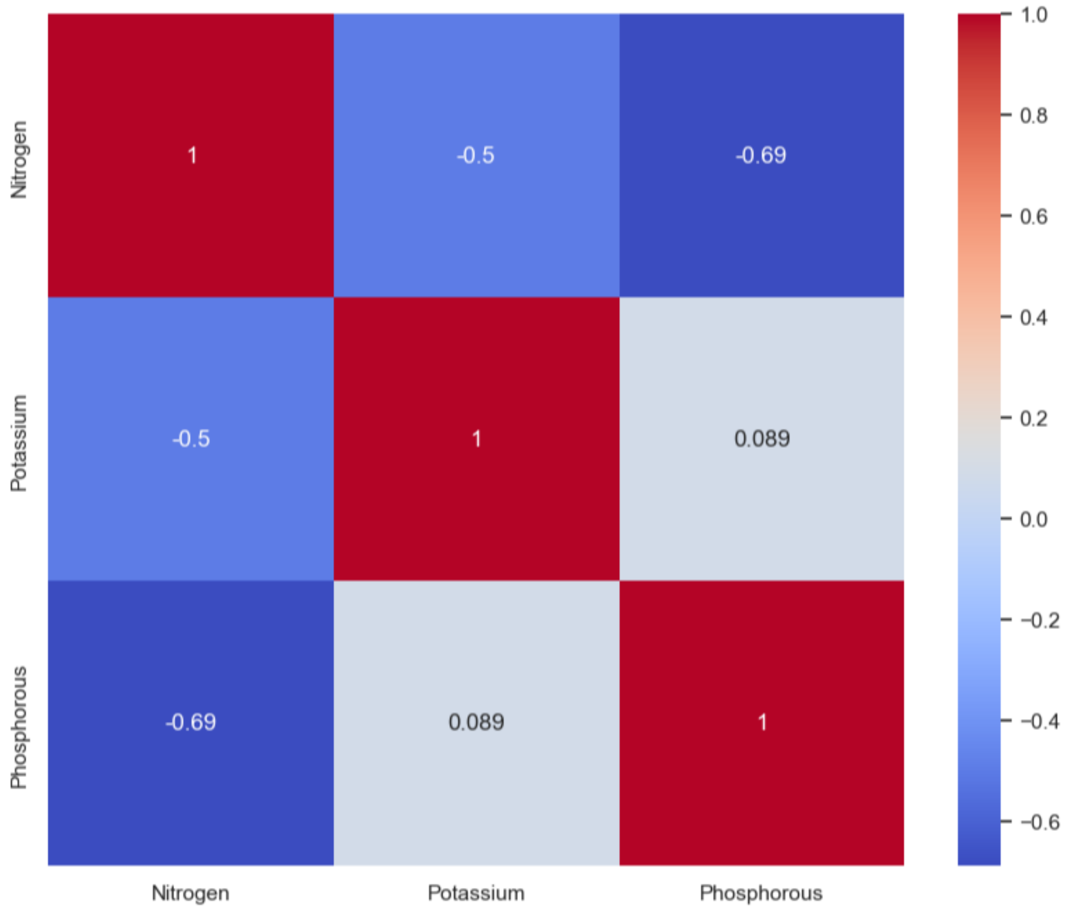


Exploratory Data Analysis

Correlation Matrix for Crop Data

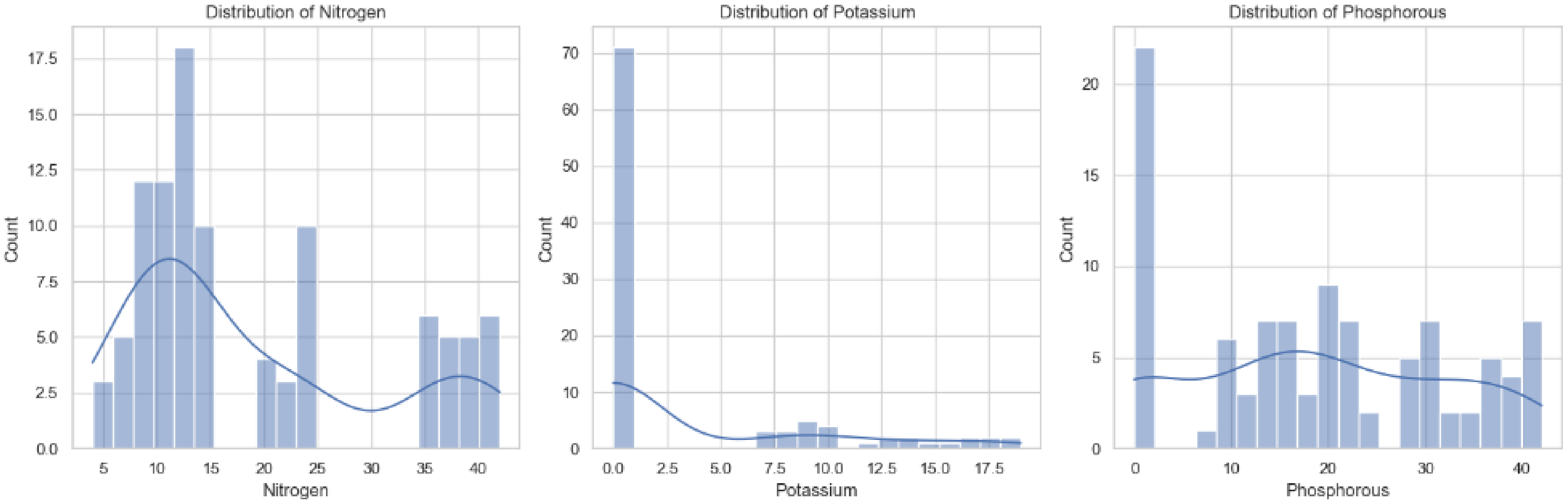


Correlation Matrix for Fertilizer Data



Northeastern University

Exploratory Data Analysis for Fertilizer Dataset



Nitrogen:

- The histogram exhibits a multimodal distribution with several peaks, suggesting multiple common values of Nitrogen.
- The line shows these modes as peaks in the probability density, indicating clusters of data points.

Potassium:

- The distribution of Potassium is highly skewed towards the lower end, with a sharp peak at the lowest bin.
- This skewness is evident in the curve, which has a steep drop-off as the values increase.

Phosphorous:

- Phosphorous levels are more evenly spread across the range, with a slight concentration at the lower end.
- The curve for Phosphorous is flatter than that of Potassium, suggesting less skewness in the data.

FEATURE SCALING

Label Encoding

Converting Categorical varibales to a integer format

```
crop_dict = {
    'rice': 1,
    'maize': 2,
    'jute': 3,
    'cotton': 4,
    'coconut': 5,
    'papaya': 6,
    'orange': 7,
    'apple': 8,
    'muskmelon': 9,
    'watermelon': 10,
    'grapes': 11,
    'mango': 12,
    'banana': 13,
    'pomegranate': 14,
    'lentil': 15,
    'blackgram': 16,
    'mungbean': 17,
    'mothbeans': 18,
    'pigeonpeas': 19,
    'kidneybeans': 20,
    'chickpea': 21,
    'coffee': 22
}
crop['crop_num']=crop['label'].map(crop_dict)
```

	Encoded
original	
DAP	0
Fourteen-Thirty Five-Fourteen	1
Seventeen-Seventeen-Seventeen	2
Ten-Twenty Six-Twenty Six	3
Twenty Eight-Twenty Eight	4
Twenty-Twenty	5
Urea	6

Importance of Feature Scaling in Recommendation

Standardizing features ensures they contribute equally to the model's predictions, crucial for algorithms that assume normally distributed features or are sensitive to the scale of input data

Benefits of Standardization:
Handling Normally Distributed Features:

1. Models like Gaussian Naive Bayes assume features to be normally distributed. Standardization makes this assumption more valid by centering the data around the mean with a unit standard deviation.

Why Use MinMaxScaler?

• Normalizing Measurement Scales:
In crop recommendation datasets, features like temperature, humidity, and soil pH can have different scales and units. MinMaxScaler ensures that these features with varying ranges don't disproportionately influence the model.

•Improving Model Performance:
Many machine learning algorithms perform better when data is on a similar scale. MinMaxScaler can help in faster convergence and improved performance

BASELINE MODEL IMPLEMENTATION

FOR CROP DATASET

NAÏVE BAYES

Handling Continuous Data: GaussianNB is particularly effective when dealing with continuous data. It assumes that the continuous values associated with each feature are distributed according to a Gaussian distribution (normal distribution). This is relevant in agricultural datasets where many features such as temperature, rainfall, and pH levels are continuous and can be assumed to follow a Gaussian distribution.

Good Performance with Small Datasets: Even with a smaller amount of data, Naive Bayes can perform quite well, making it a good choice for projects where the amount of data may be limited.

- **Strengths**: Naive Bayes is simple, fast, and performs exceptionally well when the assumption of feature independence holds. It's particularly effective in high-dimensional spaces, which might be the case with our crop dataset.
- **Weaknesses**: The assumption of feature independence rarely holds true in real-world data, which can limit its performance in some scenarios. Naive Bayes also struggles with zero-frequency problems where it assigns zero probability to unseen features/labels combinations.
- **Improvements**: Applying smoothing techniques like Laplace estimation can help with zero-frequency problems. Feature engineering to reduce dependency among variables can also improve performance.

```
Precision: 0.9958181818181817
Recall: 0.9954545454545455
F1-score: 0.9954229797979798
Accuracy: 0.9954545454545455
```

BASELINE MODEL IMPLEMENTATION

FOR CROP DATASET

NAÏVE BAYES

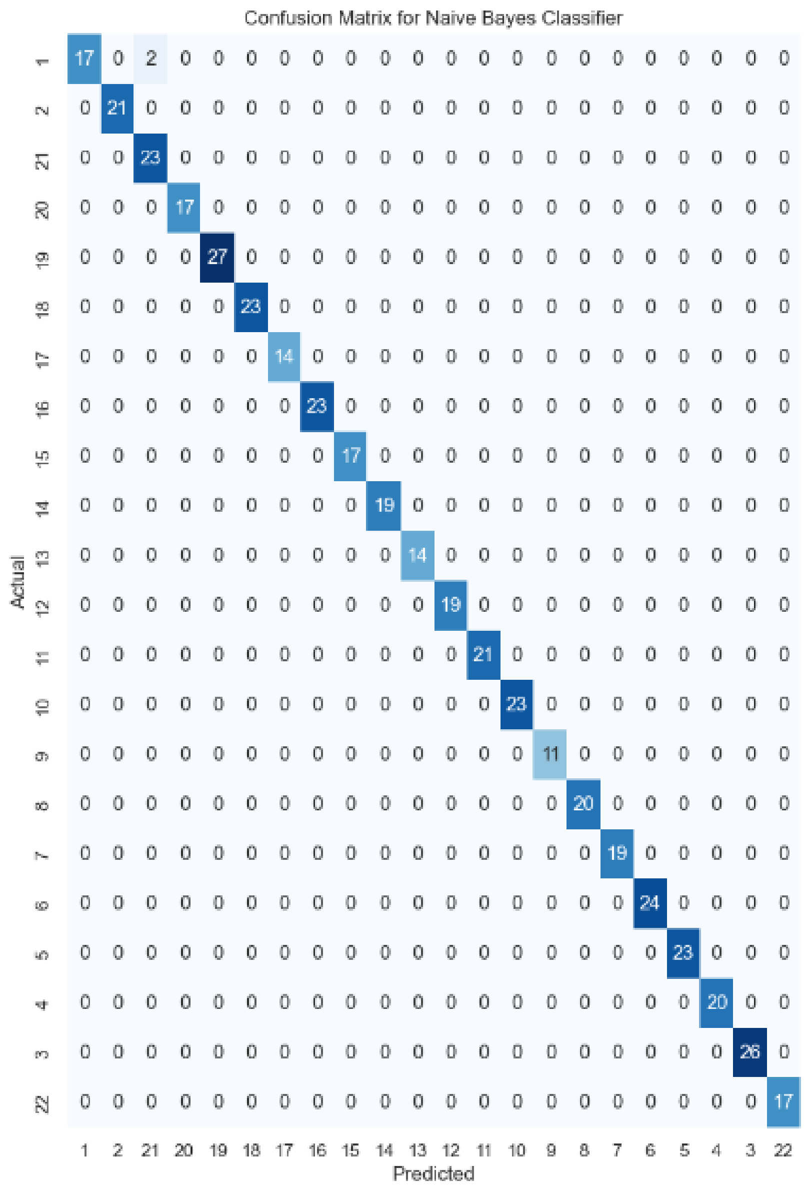
Analysis

- High Values on the Diagonal: Indicate good performance for specific crop types.
- High Values Off-Diagonal: Suggest confusion between certain crops. For example, if a high number appears in the row for crop A and the column for crop B, it means crop A is often misclassified as crop B.

Model Improvement

- If certain crops are consistently misclassified, you might need to investigate the features associated with those crops.
- Misclassifications can guide you in refining the features used for training or in tweaking the model parameters.

In crop recommendation project, the confusion matrix helps identify which crops are being accurately predicted and which ones are prone to misclassification. This insight is crucial for improving the model's performance, perhaps by collecting more data for underrepresented crops, feature engineering, or trying different algorithms



BASELINE MODEL IMPLEMENTATION

FOR FERTILIZER DATASET

RANDOM FOREST CLASSIFIER

- **Strengths:** Excellent for handling varied data types and complex relationships. Robust against overfitting due to ensemble nature.
- **Weaknesses:** Can be computationally intensive. Interpretability is less straightforward than simpler models.
- **Improvements:** Feature selection and hyperparameter tuning can enhance performance. Simplifying the model could improve interpretability and reduce computational load.

```
rand = RandomForestClassifier(random_state = 42)
rand.fit(x_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
pred_rand = rand.predict(x_test)
```

```
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report

params = {
    'n_estimators':[300,400,500],
    'max_depth':[5,6,7],
    'min_samples_split':[2,5,8]
}
grid_rand = GridSearchCV(rand,params,cv=3,verbose=3,n_jobs=-1)

grid_rand.fit(x_train,y_train)

pred_rand = grid_rand.predict(x_test)

print(classification_report(y_test,pred_rand))

print('Best score : ',grid_rand.best_score_)
print('Best params : ',grid_rand.best_params_)
```

```
Fitting 3 folds for each of 27 candidates, totalling 81 fits
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00         2
           3       1.00      1.00      1.00         3
           4       1.00      1.00      1.00         2
           5       1.00      1.00      1.00         2
           6       1.00      1.00      1.00         6

    accuracy                           1.00        20
   macro avg       1.00      1.00      1.00        20
weighted avg       1.00      1.00      1.00        20

Best score :  0.9876543209876543
Best params :  {'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 300}
```

Northeastern University

BASELINE MODEL IMPLEMENTATION

FOR FERTILIZER DATASET

RANDOM FOREST CLASSIFIER

Analysis

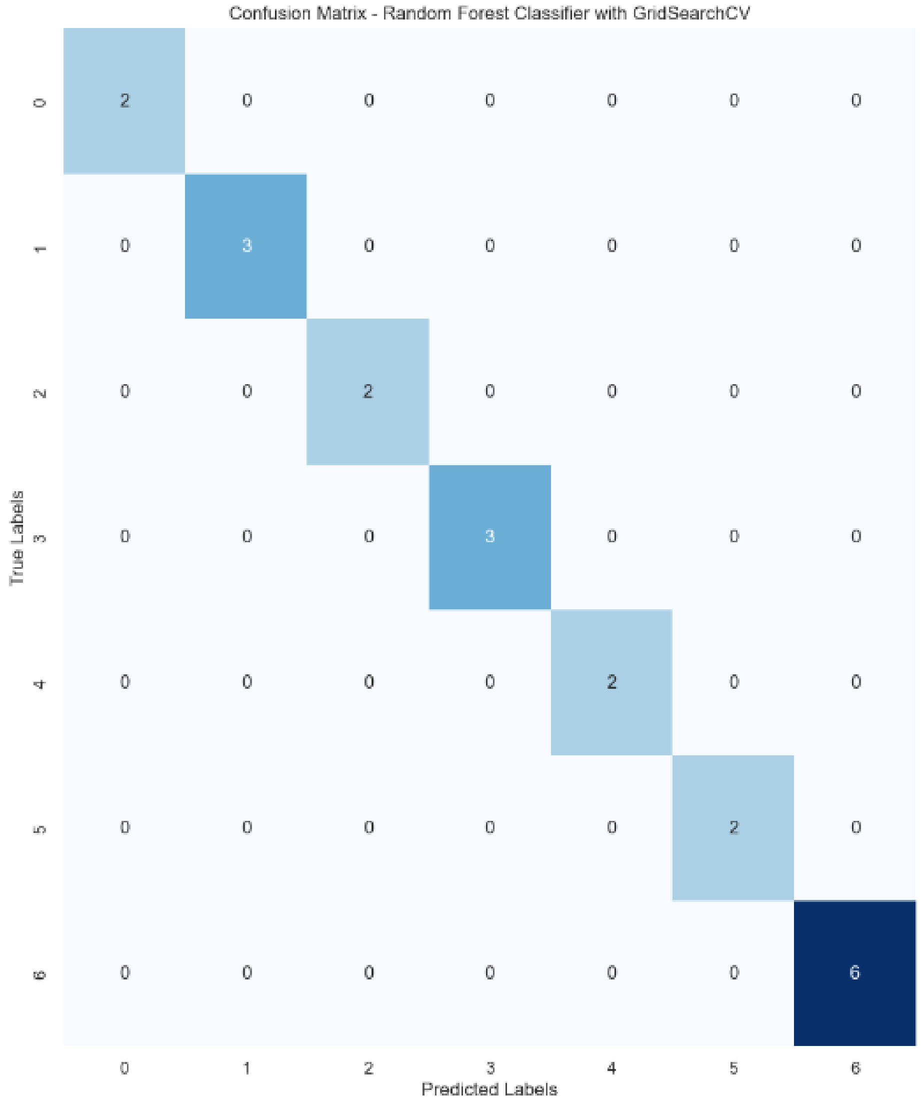
•Predominance of Zero Off-Diagonal Values: Many off-diagonal cells have zero values, which suggests that there are no misclassifications between many pairs of classes, indicating a well-performing model for those class distinctions.

•Concentration of Errors: The errors are not spread out but concentrated between specific classes, which can indicate similar feature patterns for these classes causing confusion for the model.

Model Improvement

•To further improve the model, we should examine the feature importance given by the Random Forest and consider collecting more data for the misclassified classes or reevaluating the features that lead to confusion.

The accuracy metrics (95% accuracy, 100% precision, 95% recall, and 96.6% F1-score) suggest that the Random Forest Classifier performs exceptionally well for this task. The high precision indicates that the model has a very low false positive rate, and the F1-score shows a strong balance between precision and recall. However, the points of confusion highlighted by the confusion matrix offer pathways for potential model refinement and improved accuracy.



ADVANCE MODEL IMPLEMENTATION

FOR CROP DATASET

NEURAL NETWORK with MLPC CLASSIFIER

```
Accuracy: 0.9681818181818181
Precision: 0.9715087526852233
Recall: 0.9681818181818181
F1-score: 0.9687032498174405
```

•**Strengths**: Ability to learn non-linear relationships; suitable for large datasets with many features; highly flexible.

•**Weaknesses**: Require more data to train; more computationally intensive; less interpretable than simpler models.

•**Improvements**: Tweaking the architecture (like number of layers, neurons); using regularization techniques to prevent overfitting; exploring different activation functions.

Accuracy (0.9681):
Indicates that the model correctly classified about 96.81% of the crop types. High accuracy is crucial for ensuring reliable crop recommendations.

Precision:
Reflects the model's ability to correctly identify a specific crop type without falsely classifying other crops as that type.

Recall: Measures the model's ability to find all instances of a particular crop type.

F1 Score: Harmonic mean of precision and recall, providing a balance between them. Useful when dealing with imbalanced datasets.

ADVANCE MODEL IMPLEMENTATION

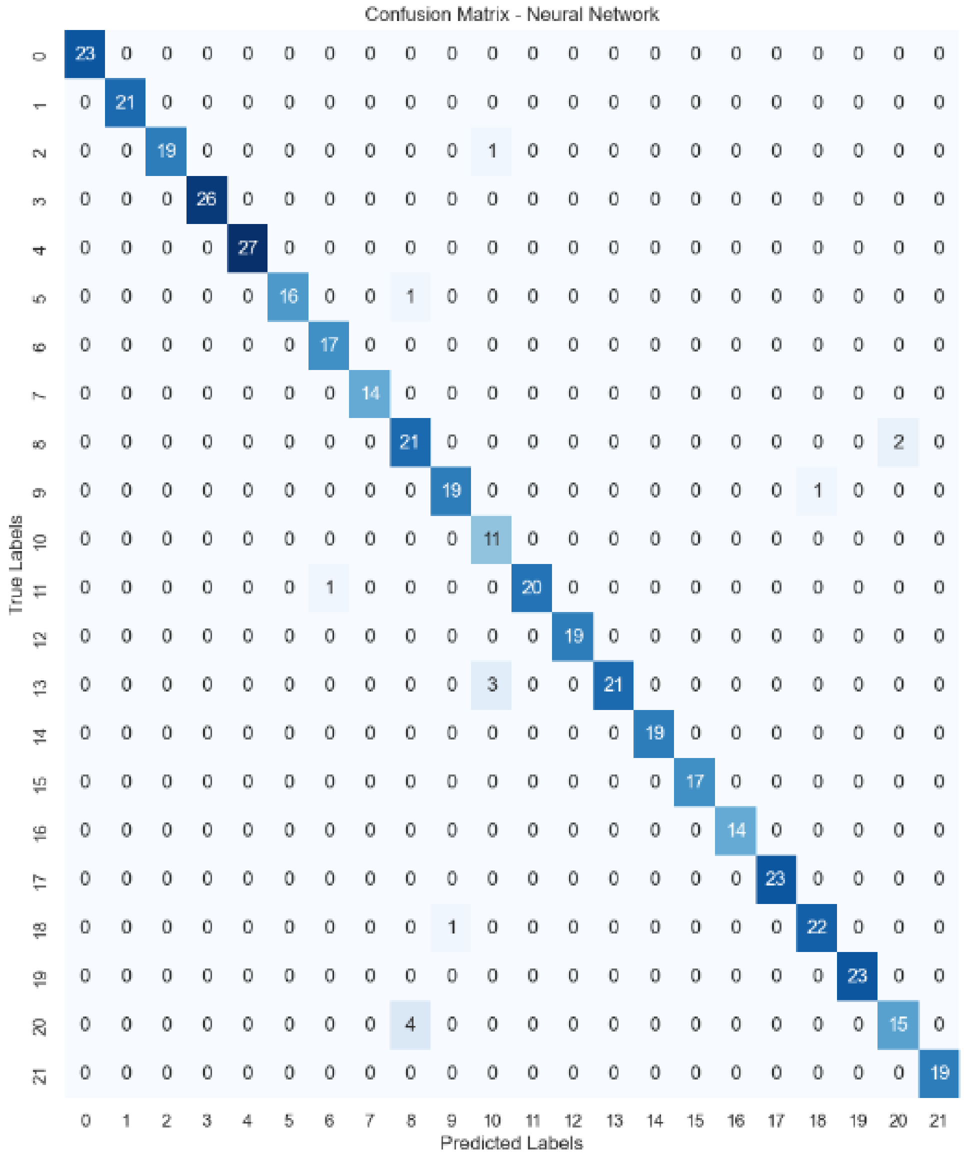
FOR CROP DATASET

NEURAL NETWORK with MLPC CLASSIFIER

Analysis
High Values on the Diagonal: Such values illustrate strong predictive performance for specific crop types by the Neural Network model.
High Values Off-Diagonal: Notably, there are some off-diagonal values, such as a count of 4 for the predicted label 21 when the actual label was 27, indicating a case of misclassification.

Model Improvement
The presence of misclassifications, although relatively low, indicates potential areas for improvement. It may be necessary to delve deeper into the features correlated with those specific crops to understand why misclassifications occurred. Enhancing the feature set, performing further feature engineering, or adjusting the model's hyperparameters could reduce these errors.

Overall, the confusion matrix provides critical insights into which crops are being precisely predicted and which are more challenging for the model. This understanding is invaluable for refining the Neural Network model, which might include gathering more data on less-represented crops or experimenting with alternative modeling techniques to boost predictive accuracy.



ADVANCE MODEL IMPLEMENTATION

FOR FERTILIZER DATASET

NEURAL NETWORK with MLPC CLASSIFIER

```
Neural Network Model Metrics:
Accuracy: 0.95
Precision: 1.0
Recall: 0.95
F1-score: 0.9666666666666666
```

•**Strengths**: Highly adaptable to complex, non-linear relationships. Exceptional in large datasets and diverse feature sets.

•**Weaknesses**: Prone to overfitting. Requires substantial data for training. Less interpretable.

•**Improvements** : Regularization techniques and proper validation can reduce overfitting. More data and improved architecture could enhance performance.

- The accuracy of 0.95 indicates that the model makes the correct predictions 95% of the time.
- The precision of 1.0 suggests that there are no false positives; every fertilizer type predicted by the model is correct.
- A recall of 0.95 means that the model captures 95% of the actual fertilizer types correctly.
- An F1 score of 0.966 confirms that the model maintains an excellent balance between precision and recall, which is essential for a recommendation system where both false positives and false negatives have significant implications.

Northeastern University

ADVANCE MODEL IMPLEMENTATION

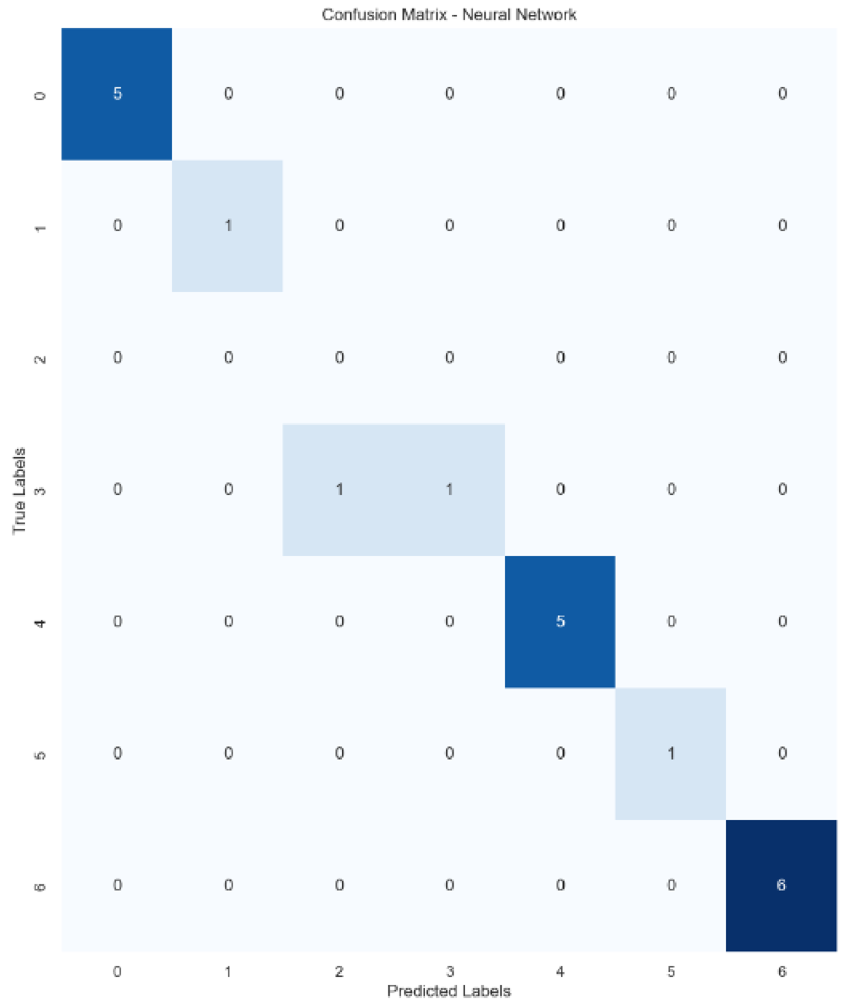
FOR FERTILIZER DATASET

NEURAL NETWORK with MLPC CLASSIFIER

Analysis
•**Predominant Diagonal Values**: The high values on the diagonal for certain classes suggest that the model is particularly effective at correctly classifying these classes.
•**Sparse Off-Diagonal Values**: The presence of few off-diagonal values indicates that there are relatively few misclassifications overall, which is a positive indicator of model performance.

Model Improvement
•The misclassifications that do occur provide opportunities for model improvement. Investigating the features that lead to confusion between classes '1' and '0', as well as '4', '3', and '5', could offer insights into how to refine the model.
•Enhancements might include feature engineering to better distinguish between these classes, collecting more representative data for the underperforming classes, or adjusting the model architecture and hyperparameters.

Given the model's performance metrics, such as an accuracy of 0.95, precision of 1.0, recall of 0.95, and F1 score of 0.966, it is evident that the Neural Network is performing well. These high scores reflect a model that is not only accurate but also balanced in terms of precision and recall, making it reliable for making fertilizer recommendations.



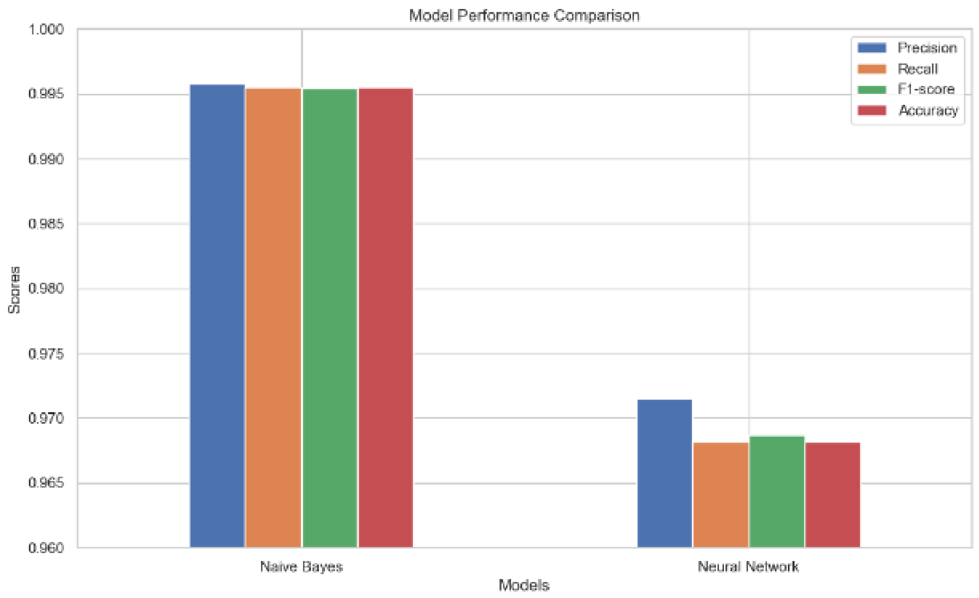
RESULTS for Crop Dataset

Why Might Naive Bayes Perform Better?
The high performance of Naive Bayes suggests that the dataset likely has features that are relatively independent, a condition where Naive Bayes thrives. Its simplicity also helps to avoid overfitting, a problem that more complex models can sometimes face.
Additionally, if the data has many categorical features or features following a probability distribution that Naive Bayes assumes, it can outperform other models.

Analysis of Lower Scores in Other Models
• For Neural Networks, the lower score might be due to overfitting, insufficient training, or a need for more data.

• Decision Trees and XGBoost may have been too complex for the dataset, leading to overfitting despite their capability to handle complex patterns.

• Random Forest's lower performance compared to Naive Bayes might be due to its ensemble nature, where the randomness didn't align well with the dataset patterns as much as the probabilistic approach of Naive Bayes.



RESULTS for Crop Dataset

Training Time Comparison

The time taken to train a model is crucial, especially in large datasets or when frequent retraining is necessary.

•Observations: Naive Bayes, typically, requires significantly less training time compared to Neural Networks, making it efficient for scenarios demanding quick model updates or limited computational resources.

Memory Usage Evaluation

We evaluated the memory consumption during the training phase of each model, a critical aspect in resource-constrained environments.

•Observations: Neural Networks often consume more memory due to their complex architecture, while Naive Bayes, with its simpler structure, has a lower memory footprint, suitable for deployment with restricted memory.

Log Loss Analysis

Log loss measures the confidence of the predictions, penalizing false classifications more heavily if the model is very confident in its incorrect predictions.

•Observations: A lower log loss is indicative of a model with reliable and confident predictions. This aspect is vital in applications like crop recommendations, where uncertain predictions can lead to significant consequences.

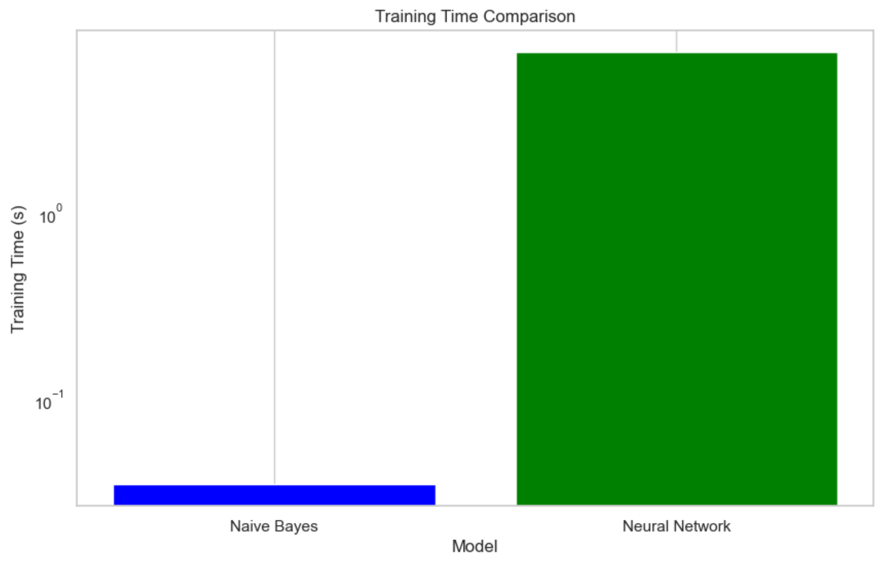
Model Complexity and Size

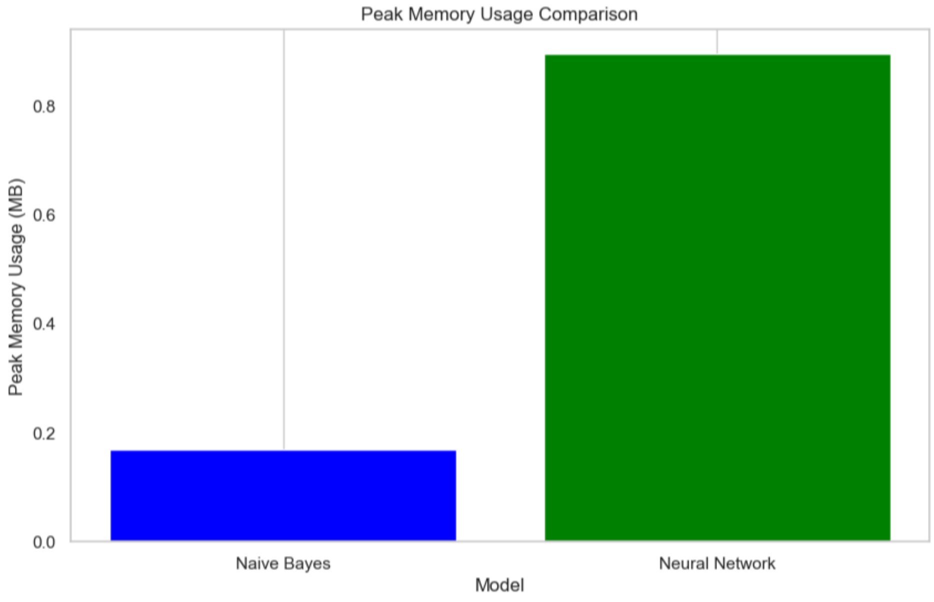
We also considered the complexity and size of each model, as a simpler model with fewer parameters is easier to deploy, especially in environments with limited computational resources.

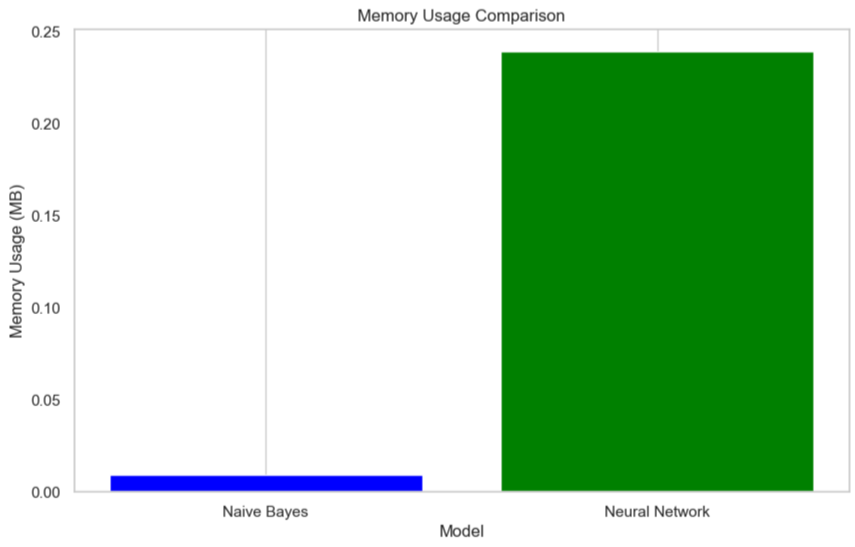
•Observations: Naive Bayes, being fundamentally simpler, has fewer parameters and thus lower complexity.

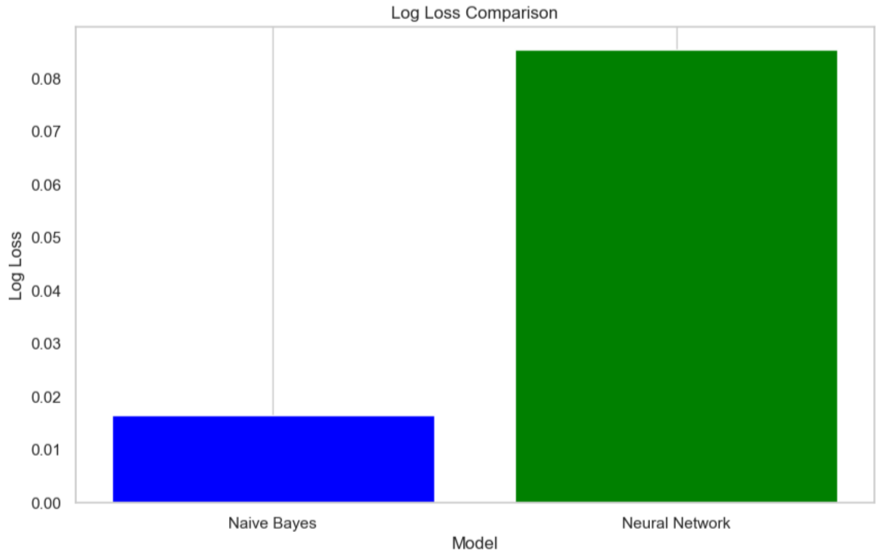
```
Naive Bayes training time: 0.0348 seconds
Naive Bayes memory usage: 0.0092 MB; Peak: 0.1701 MB
Naive Bayes Log Loss: 0.0165

              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        23
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        26
     coconut       1.00      1.00      1.00        27
      coffee       1.00      1.00      1.00        17
      cotton       1.00      1.00      1.00        17
      grapes       1.00      1.00      1.00        14
        jute       0.92      1.00      0.96        23
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        11
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        19
   mothbeans       1.00      1.00      1.00        24
    mungbean       1.00      1.00      1.00        19
   muskmelon       1.00      1.00      1.00        17
      orange       1.00      1.00      1.00        14
      papaya       1.00      1.00      1.00        23
  pigeonpeas       1.00      1.00      1.00        23
 pomegranate       1.00      1.00      1.00        23
        rice       1.00      0.89      0.94        19
  watermelon       1.00      1.00      1.00        19

    accuracy                           1.00       440
   macro avg       1.00      1.00      1.00       440
weighted avg       1.00      1.00      1.00       440

Neural Network training time: 7.2994 seconds
Neural Network memory usage: 0.2391 MB; Peak: 0.8956 MB
Neural Network Log Loss: 0.0854

              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        23
      banana       1.00      1.00      1.00        21
   blackgram       1.00      0.95      0.97        20
    chickpea       1.00      1.00      1.00        26
     coconut       1.00      1.00      1.00        27
      coffee       1.00      1.00      1.00        17
      cotton       0.94      1.00      0.97        17
      grapes       1.00      1.00      1.00        14
        jute       0.88      0.96      0.92        23
 kidneybeans       0.95      0.95      0.95        20
      lentil       0.73      1.00      0.85        11
       maize       1.00      0.95      0.98        21
       mango       1.00      1.00      1.00        19
   mothbeans       1.00      0.88      0.93        24
    mungbean       1.00      1.00      1.00        19
   muskmelon       1.00      1.00      1.00        17
      orange       1.00      1.00      1.00        14
      papaya       1.00      1.00      1.00        23
  pigeonpeas       0.96      0.96      0.96        23
 pomegranate       1.00      1.00      1.00        23
        rice       0.94      0.84      0.89        19
  watermelon       1.00      1.00      1.00        19

    accuracy                           0.97       440
   macro avg       0.97      0.98      0.97       440
weighted avg       0.98      0.97      0.98       440
```

RESULTS for Crop Dataset









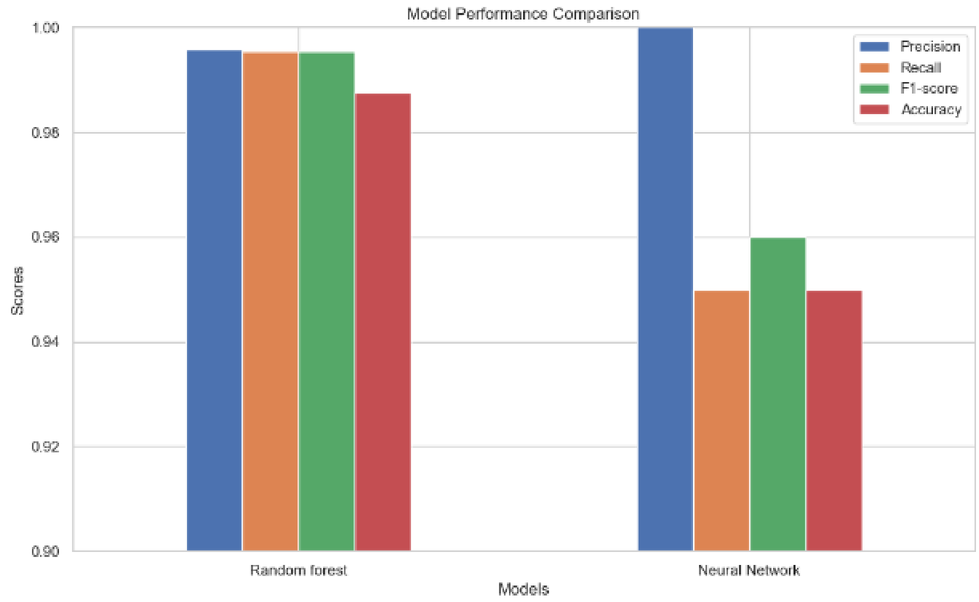
RESULTS for Fertilizer Dataset

Analysis of Model Performance
The superior performance of Random Forest in this context can be attributed to severa factors:

1.Data Characteristics: The dataset might have features and relationships well-captured by the decision trees in Random Forest.

2.Overfitting Avoidance: Random Forest naturally avoids overfitting better than Neural Networks, especially if the dataset isn't massive.

3.Complexity Balance: Random Forest strikes a balance between handling complex relationships and not becoming too complex itself, unlike Neural Networks which can become overly complex.



- In the fertilizer recommendation project, the choice between Random Forest and Neural Network should consider the dataset's nature and the project's specific needs. Random Forest emerges as a more balanced choice, offering robust performance with less risk of overfitting and a good handle on complex data relationships. Its ability to provide high accuracy with less computational complexity makes it suitable for a variety of scenarios.

- Neural Networks, while powerful, may require more data and careful tuning to achieve their full potential. They are more suited to scenarios where the complexity and size of the dataset justify their use.

- Ultimately, the model selection should align with the dataset's characteristics, computational resources, and the required balance between accuracy and interpretability. For many agricultural datasets, Random Forest offers a compelling mix of high performance and practical usability.

RESULTS for Fertilizer Dataset

Training Time: The model took approximately 0.891 seconds to train.
Memory Usage: The training process consumed 0.2875 MB of memory, with a peak memory usage of 3.0337 MB.

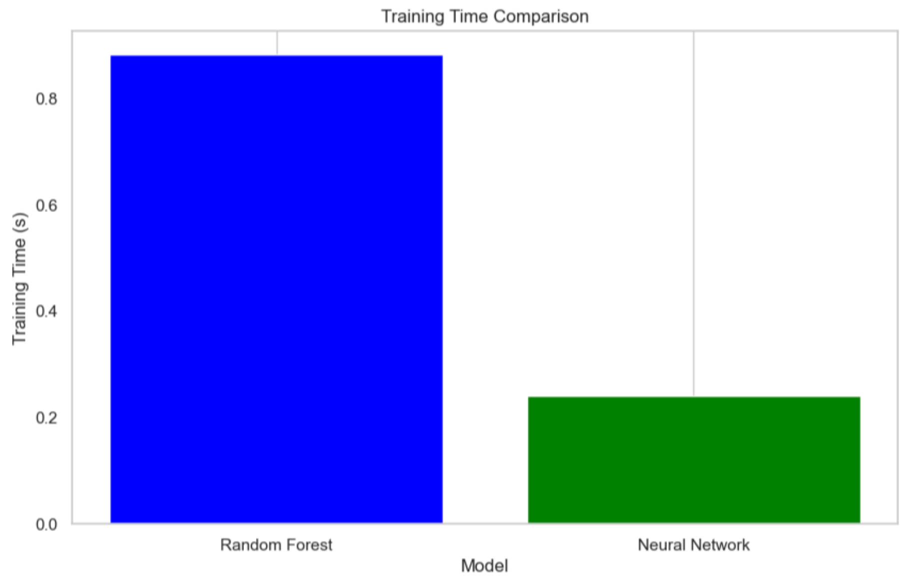
Performance Metrics:
Precision, Recall, and F1-Score: These metrics are perfect (1.00) across all classes, which include various categories like 'Fourteen-Thirty Five-Fourteen' and 'Twenty-Eight-Twenty-Eight'. This suggests that the model has predicted every class with 100% accuracy without any false positives or false negatives.
Support: This column indicates the number of true occurrences of each class in the dataset. The classes have varying support, with some having only 1 instance and others having more, up to 5.
Macro Average: Averages the performance metrics for each class, and these are also perfect (1.00), indicating uniform excellence across all classes despite the imbalance in their representation.

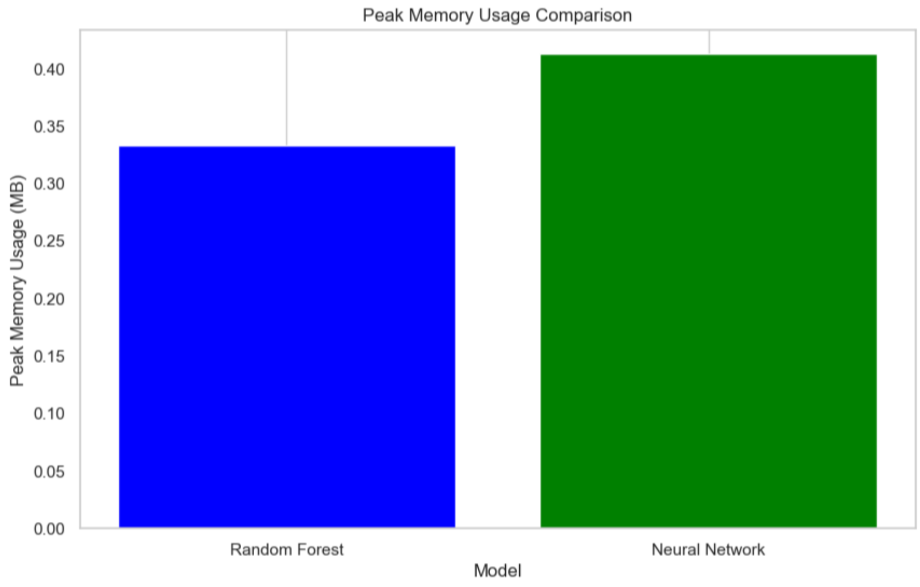
NEURAL NETWORK:
Training Time: The Neural Network model took significantly longer to train, with 2.8411 seconds.
Memory Usage: It required more memory, with 0.1895 MB used and a peak of 4.4129 MB.

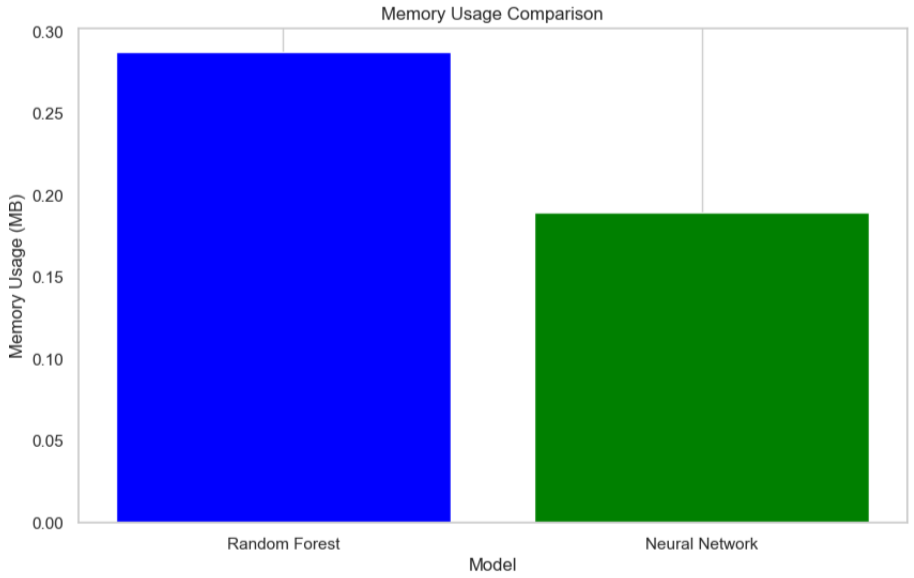
Performance Metrics:
Precision, Recall, and F1-Score: Like the Random Forest Classifier, the Neural Network also achieved perfect scores across all classes.
Support: The distribution of true occurrences is the same as for the Random Forest.
Macro and Weighted Averages: Both are perfect at 1.00.

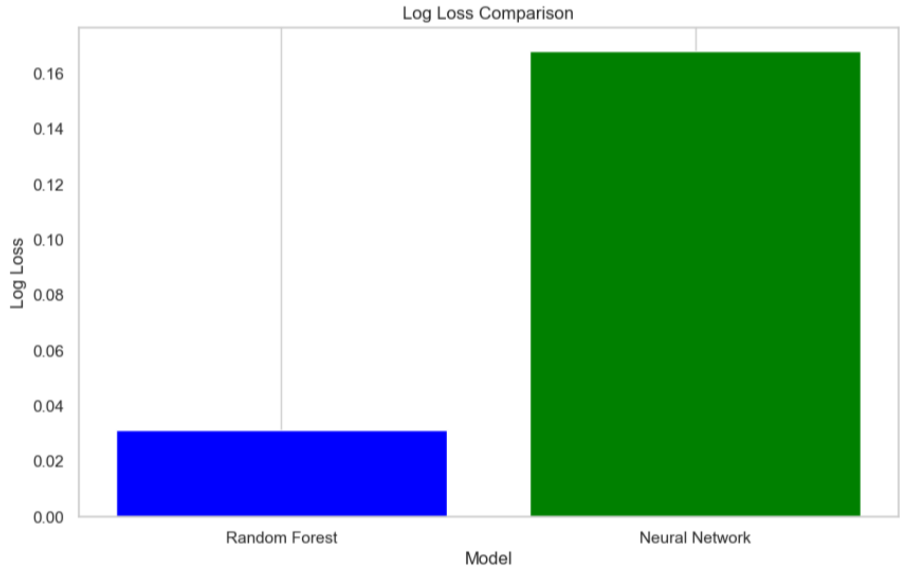
```
Random Forest Classifier:
Random Forest Classifier Training Time: 0.8821 seconds
Random Forest Classifier Memory Usage: 0.2875 MB; Peak: 0.3337 MB
Random Forest Classifier Log Loss: 0.03315874044298863

                               precision    recall  f1-score   support

                          DAP       1.00      1.00      1.00         4
Fourteen-Thirty Five-Fourteen       1.00      1.00      1.00         3
Seventeen-Seventeen-Seventeen       1.00      1.00      1.00         1
    Ten-Twenty Six-Twenty Six       1.00      1.00      1.00         1
    Twenty Eight-Twenty Eight       1.00      1.00      1.00         3
                Twenty-Twenty       1.00      1.00      1.00         3
                         Urea       1.00      1.00      1.00         5

                     accuracy                           1.00        20
                    macro avg       1.00      1.00      1.00        20
                 weighted avg       1.00      1.00      1.00        20


Neural Network:
Neural Network Training Time: 0.2411 seconds
Neural Network Memory Usage: 0.1895 MB; Peak: 0.4129 MB
Neural Network Log Loss: 0.16810064239464428

                               precision    recall  f1-score   support

                          DAP       1.00      1.00      1.00         4
Fourteen-Thirty Five-Fourteen       1.00      1.00      1.00         3
Seventeen-Seventeen-Seventeen       1.00      1.00      1.00         1
    Ten-Twenty Six-Twenty Six       1.00      1.00      1.00         1
    Twenty Eight-Twenty Eight       1.00      1.00      1.00         3
                Twenty-Twenty       1.00      1.00      1.00         3
                         Urea       1.00      1.00      1.00         5

                     accuracy                           1.00        20
                    macro avg       1.00      1.00      1.00        20
                 weighted avg       1.00      1.00      1.00        20
```

RESULTS for Fertilizer Dataset



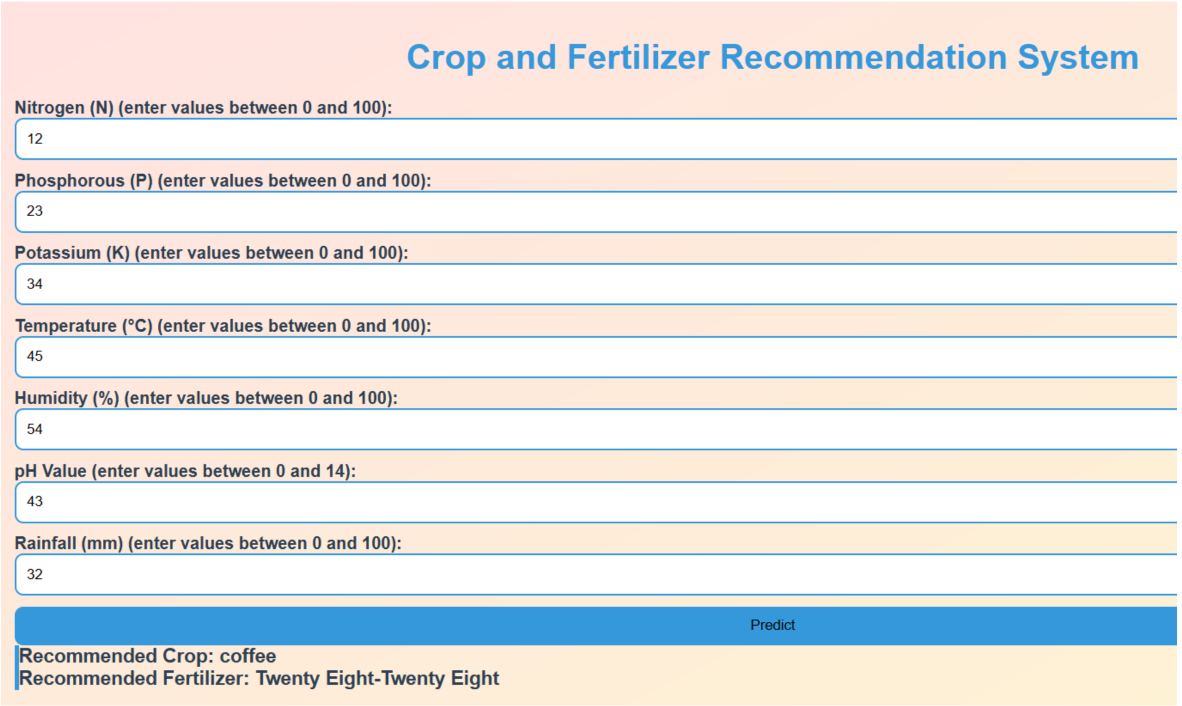


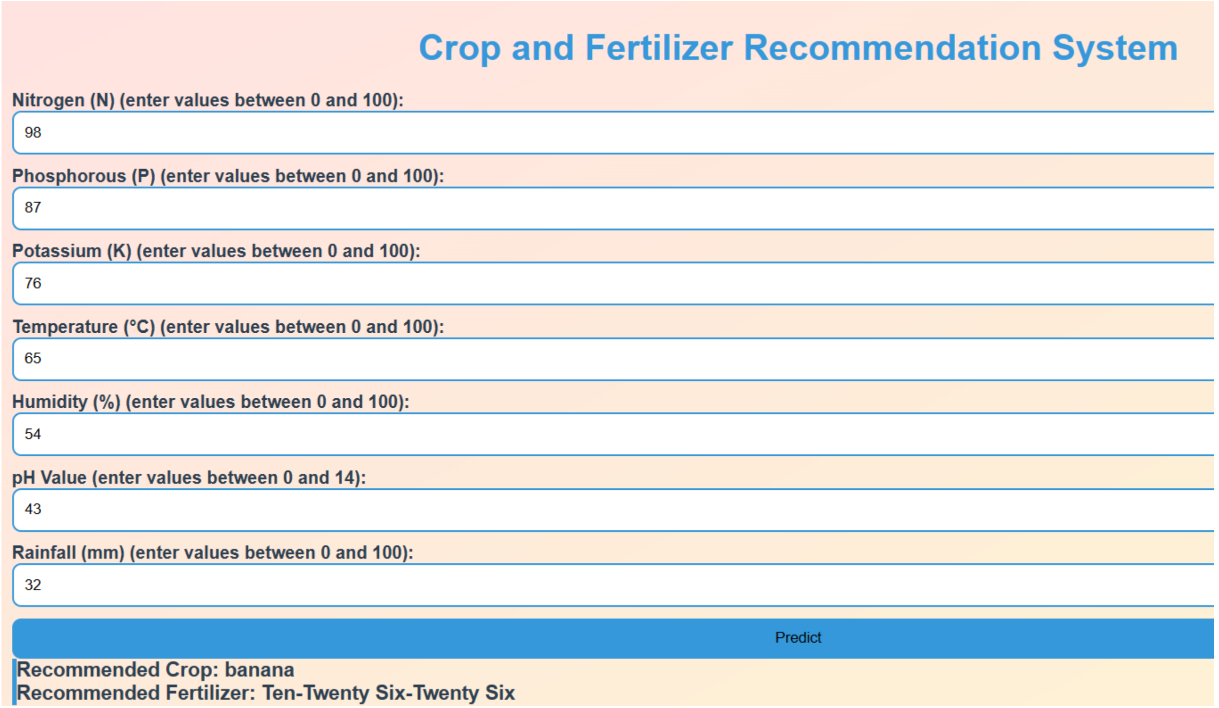




FLACK APPLICATION

App.py file with GET and POST requests based on User input received for all parameters based on 'Index.html' file
The result is recommended by using pre trained models called in flask framework uby picle files for specif datasets





CONCLUSION

Comprehensive Conclusion on Model Selection for Crop and Fertilizer Recommendation

Model Suitability: Model choice should be tailored to the dataset's unique traits and the project's objectives. Random Forest offers a balanced approach with high accuracy and low risk of overfitting, suitable for varied scenarios.

Random Forest Advantages: It handles complex data relationships effectively, with less computational demand, making it ideal for agricultural datasets where performance and usability are key.

Neural Networks Considerations: While Neural Networks are adept at managing complex datasets, they require more data and meticulous hyperparameter tuning to fully leverage their capabilities.

Naive Bayes Appropriateness: For datasets with independent features and possibly small or imbalanced data, Naive Bayes is efficient and simple, though caution is advised for more complex datasets where feature independence is not assured.

Cross-Validation Importance: Ensuring that the model performs well on unseen data through cross-validation is crucial, affecting the choice between simpler models like Naive Bayes and more complex ones like Neural Networks.

Final Decision Factors: The decision on the optimal model, be it Naive Bayes or Neural Networks, hinges on a balance between predictive performance, model transparency, and the dataset’s complexity.

Northeastern University

ACKNOWLEDGEMENT

- Majority of work has been divided among team members
- Nainil worked on the data preprocessing and EDA along with feature engineering.
- Additionally, Model implementation including Naïve Bayes and NN was carried out by Nainil
- Nainil built a flask framework to use model for training of web model.

- Simran worked on some feature engineering, data visualization and carried out model implementation using Random Forest and Neural Networks
- Simran made html interface with input from user for interactive application for crop and fertilizer recommendation

References

1. Recommender System lecture python notebook by Prof. Junwei Huang
2. [Crop Recommendation using ML] (https://ieeexplore.ieee.org/document/9734173)
3. Kaggle : (https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset)
4. Fertilizer Kaggle: (Plant Disease Classification - ResNet- 99.2% | Kaggle)
5. Microssoft Bing AI chat