MACHINE LEARNING FOR SMARTER CROP PLANNING

DEBASIS PANI, NOAH SPRENGER, SAHITHI CHALLAPALLI

Introduction: **The Global Challenge**

Feeding more people with fewer resources and greater uncertainty

- **Rising Food Demand**
  *Food demand could rise **~50% by 2050** as populations grow and diets shift.*
- **Shrinking Farmland**
  Cities and erosion remove millions of hectares of arable land each decade.
- **Declining Soil Health**
  Intensive cultivation and synthetic fertilizer/herbicide dependence degrade fertility.
- **Growing Cyclical Climate Volatility**
  Irregular weather patterns, including rainfall, global cooling, and heat waves, cut yields across regions.

Motivation: **Doing More with Less**

The challenge isn't just to grow, it's to grow smarter, more sustainably.

- **More Output, Fewer Resources**
  *Farmers must increase yields using **less land, water, and fertilizer***

- **Two Divergent paths:**
  1. Heavy synthetic inputs (fast, but destructive)
  2. **Data Guided Adaptation, working with the land**

- **Data as Leverage:**
  - Affordable soil & weather sensors now offer unprecedented data access
  - **Models must stay simple, interpretable, and accessible**
    - *i.e. not compute-intensive, cloud-reliant*

Our goal:

*Show how even a simple model like **Naïve Bayes** can turn these data into actionable, sustainable insight*

Problem Statement & Hypothesis

The Question

*Can we predict which crop will **maximize yield** under a given combination of soil nutrients and climate conditions?*

Our Hypothesis

*If we know soil chemistry and climate variables, a **Naïve Bayes** classifier can identify the crop with the **highest expected success probability**.*

Why it Matters

*This project combines **probability theory** with **data interpretation,** two cornerstones of data science, to turn environmental data into practical, yield-oriented insights.*

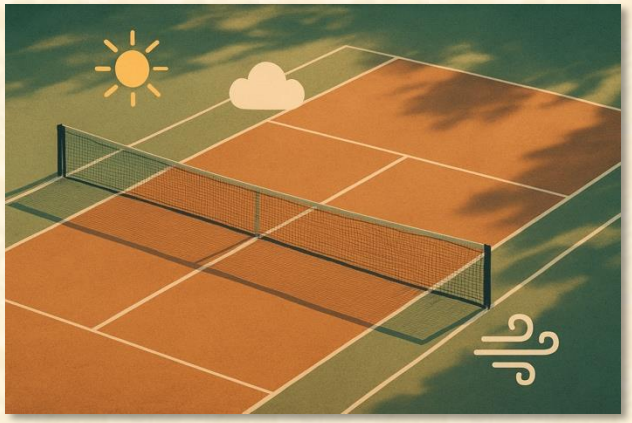
*For this proof of concept, we focused on two major crops, **rice** and **maize,** to demonstrate the method with a limited dataset before extending it to a larger, multi-crop dataset.*

Naïve Bayes Overview

From 'Play Tennis?' to 'Which Crop Yields More?'

PLAY TENNIS EXAMPLE

Weather Conditions
↓
Decision



Concept
Conditions influence the chance of playing.

Feature List

- Outlook
- Temperature
- Humidity
- Wind

Decision: Play/Don't Play

CROP PREDICTION

Soil + Climate
↓
Crop Decision

Concept
Soil and weather features influence optimal crop.

Feature List

- Soil Chemistry *(Nitrogen, Phosphorus, Potassium, pH)*
- Temperature
- Humidity
- Rainfall

Decision: Rice/Maize

Dataset Overview

- We extracted a small subset of 15 samples: **9 for rice** and **6 for maize**, from a larger agricultural dataset.
- Each sample captures **seven environmental features** matched to the **optimal crop for those conditions:** Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, and rainfall.
- These values are realistic — rice entries have higher humidity and rainfall, while maize rows represent moderate conditions.
- This dataset is small enough to compute manually but still reflects real agricultural outcomes.

N	P	K	temperature	humidity	ph	rainfall	label
90	42	43	20.8797437	82.0027442	6.502985292000001	202.935536	rice
85	58	41	21.7704617	80.3196441	7.03809636	226.655537	rice
60	55	44	23.0044592	82.3207629	7.84020714	263.964248	rice
74	35	40	26.4910964	80.1583626	6.98040091	242.864034	rice
78	42	42	20.1301748	81.6048729	7.62847289	262.717341	rice
69	37	42	23.0580487	83.3701177	7.0734535	251.055	rice
69	55	38	22.708838	82.6394139	5.70080568	271.32486	rice
94	53	40	20.2777436	82.8940862	5.718627177999999	241.974195	rice
89	54	38	24.5158807	83.53521629999999	6.68534642	230.446236	rice
71	54	16	22.6135995	63.6907056	5.749914421000001 5	87.7595386	maize
61	44	17	26.1001842	71.5747694	6.931756557999999	102.266245	maize
80	43	16	23.5588209	71.5935137	6.65796475	66.7199547	maize
73	58	21	19.9721595	57.6827292	6.59606065	60.6517148	maize
61	38	20	18.4789126	62.6950387	5.97045843	65.4383539	maize
68	41	16	21.7768932	57.8084064	6.15883062	102.086169	maize

Data Preprocessing: **Feature Discretization**

- Key preprocessing step: ***feature discretization***

  To prepare the data for Naïve Bayes, we converted **continuous features** into **categorical ranges**.

- Method:

  We used **tercile-based thresholds**, dividing each continuous feature's distribution into three equal-frequency groups.

- Reasoning:

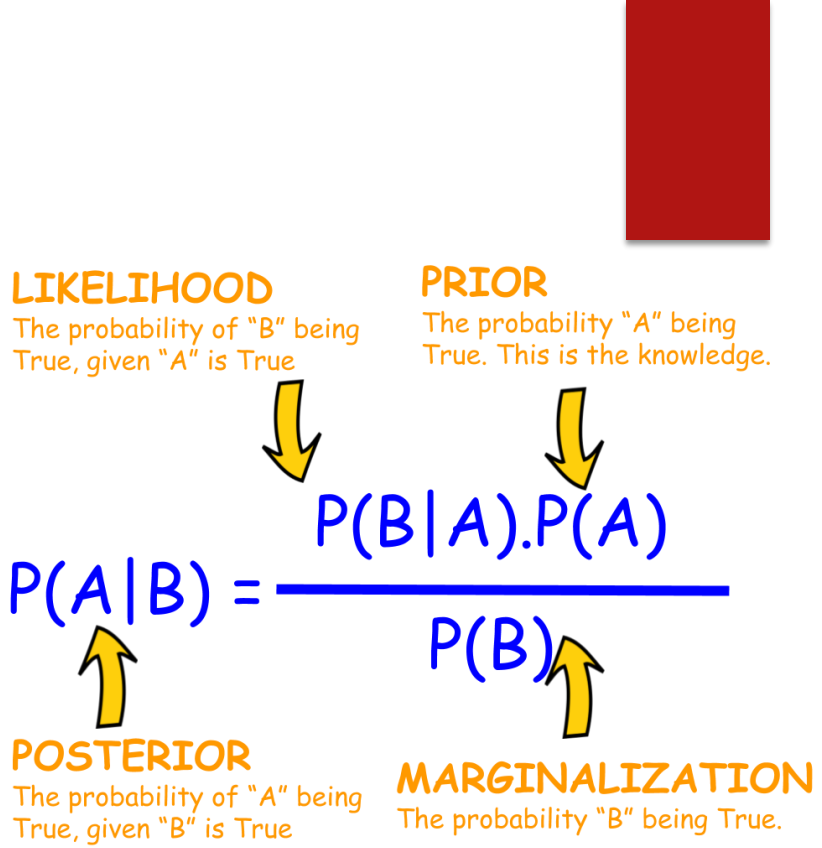
  Naïve Bayes operates on **categorical inputs**, so *discretizing* the feature variables allows us to calculate conditional probabilities directly.

- Example discretization:
  - Nitrogen: 90 → *High Nitrogen*
  - Temperature: 23°C → *Temperate*
  - Rainfall: 250 mm → *High Rainfall*

Feature	*1st Tercile*	*2nd Tercile*	*3rd Tercile*
N (Nitrogen)	Low	Med	High
P (Phosphorus)	Low	Med	High
K (Potassium)	Low	Med	High
Temperature	Cool	Temperate	Warm
Humidity	Low	Med	High
pH	Acidic	Neutral	Alkaline
Rainfall	Low	Med	High

Mathematical Foundation: **Bayes' Theorem**

- Bayes' Theorem powers our model to predict which crop best fits given environmental conditions.
- It begins with prior knowledge — how each crop has performed historically.
- Then, it considers new evidence — rainfall, temperature, and soil data.
- The theorem updates our belief about each crop's likelihood of success.
- With every new observation, predictions become more accurate.
- Thus, probability evolves into insight — turning data into smarter farming decisions.



$$P(\text{Crop} \mid \text{Cond}) = \frac{P(\text{Cond} \mid \text{Crop}) \times P(\text{Crop})}{P(\text{Cond})}$$

Posterior Probability Calculation

- The **posterior probabilities** for both crops were computed using **Bayes' Theorem**.
- Each crop's probability was determined by multiplying its **prior** with all **conditional probabilities** for the environmental features.
- The calculations for Rice and Maize were performed separately, then **normalized** so the *total probability = 1*:
  - **P(Rice | Cond) = 0.97**
  - **P(Maize | Cond) = 0.02**
- The model predicts **Rice** as the most suitable crop, with approximately **97 % probability** under the given environmental conditions.

our test conditions:

N	P	K	temperature	humidity	ph	rainfall	label
high	low	med	cool	med	acidic	med	?

Posterior Probability Calculation

For Rice

$$P(\text{Rice} \mid \text{Cond}) \propto P(\text{Rice}) \times P(N \mid \text{Rice}) \times P(P \mid \text{Rice}) \times P(K \mid \text{Rice}) \times P(\text{Temp} \mid \text{Rice}) \times P(\text{Humidity} \mid \text{Rice}) \times P(\text{pH} \mid \text{Rice}) \times P(\text{Rainfall} \mid \text{Rice})$$
$$= 0.6 \times 0.001284$$
$$= \mathbf{0.000770}$$

For Maize

$$P(\text{Maize} \mid \text{Cond}) \propto P(\text{Maize}) \times P(N \mid \text{Maize}) \times P(P \mid \text{Maize}) \times P(K \mid \text{Maize}) \times P(\text{Temp} \mid \text{Maize}) \times P(\text{Humidity} \mid \text{Maize}) \times P(\text{pH} \mid \text{Maize}) \times P(\text{Rainfall} \mid \text{Maize})$$
$$= 0.4 \times 0.000043$$
$$= \mathbf{0.000017}$$

Normalization

To make a fair comparison between both crop classes, we normalize the posterior probabilities so that their sum equals 1.

For the given posterior values of Rice and Maize:

$$P(\text{Rice}|Cond) = \frac{0.000770}{0.000770 + 0.000017} = 0.978$$
$$P(\text{Maize}|Cond) = \frac{0.000017}{0.000770 + 0.000017} = 0.022$$

Interpretation:

After normalization, the model assigns roughly 97.8% probability to Rice and 2.2% to Maize. Under these conditions, **Rice** is clearly the predicted crop.

Visualization: **Seeing the Data**

- After completing the calculations, we visualized the data to verify if it aligned with our probability results.
- The rainfall distribution showed **Rice thriving in higher rainfall zones**, while **Maize dominated moderate regions**.
- Likewise, **Rice correlated with higher humidity**, whereas **Maize preferred slightly acidic pH conditions**.
- These visual patterns validated that our **dataset and model were in strong agreement**.

Implementation in Python

- Implemented a **Naïve Bayes classifier** using GaussianNB from *scikit-learn*
- Followed a standard **data-science workflow:**
  data preparation → encoding → training → prediction
- Used the **same 15-sample dataset** as in the manual example (9 rice, 6 maize)
- 7 Input features:
  N, P, K, temperature, humidity, pH, and rainfall
- Crop labels were numerically encoded for model compatibility

NB_crop_predict_model.py

```python
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
import pandas as pd

# 15-sample subset (9 rows rice, 6 maize)
X = data[["N", "P", "K", "temperature", "humidity", "ph", "rainfall"]]
y = LabelEncoder().fit_transform(data["label"])

# train and predict
model = GaussianNB()
model.fit(X, y)
y_pred = model.predict(X)

# compare predictions
accuracy = (y == y_pred).mean() * 100
```

The model produced **identical predictions** to manual calculation, confirming the logic translated correctly into code

Sample Output

```
Model Accuracy: 100.00%

Actual vs Predicted:
Actual Predicted
0 rice rice
1 rice rice
2 rice rice

Predicted crop for new conditions → rice
```

Implementation in Python

Scaling and Testing the Model

We scaled from a 15-row subset to a full **1,400-sample dataset** to see how a *programmatic version* of our model performed under broader, more realistic conditions.

- **Goal:** explore how model accuracy *changed (or stubbornly refused to change)* as we scaled data and shifted its composition.

Experimental Variables:

- Rows per crop: 5-100
- Number of crops: 2-14
- Train/test splits: 10%-90%

Purpose:
To understand whether scaling the dataset revealed **genuine learning behavior** or if accuracy stayed **unrealistically high due to clean data.**

*Python: using **pandas** and GaussianNB to extend the 15-row prototype into a fully scalable model.*

programmatic_GaussianNB_crop_predict_v2.py

```
# define per-crop sampling
+ CROP_ROWS = {"rice": 50, "maize": 40, "chickpea": 35}
+ TRAIN_SPLIT = 0.1
+ RANDOM_SEED = 42

# load and split dataset
+ df = pd.read_csv("Crop dataset updated.csv")
+ for crop, n in CROP_ROWS.items():
+     crop_data = df[df["label"] == crop].sample(n)
+     train, test = train_test_split(crop_data, train_size=0.1)

# cross-validation and accuracy checks
+ scores = cross_val_score(GaussianNB(), X_train, y_train, cv=5)
+ if accuracy > 0.95:
+     check_feature_separation()
```

Example Case:

- ***3 crops***
- ***12 training samples***
- ***113 test samples***

Even with minimal training data and added crops, accuracy barely dropped.

Programmatic Implementation Sample Output

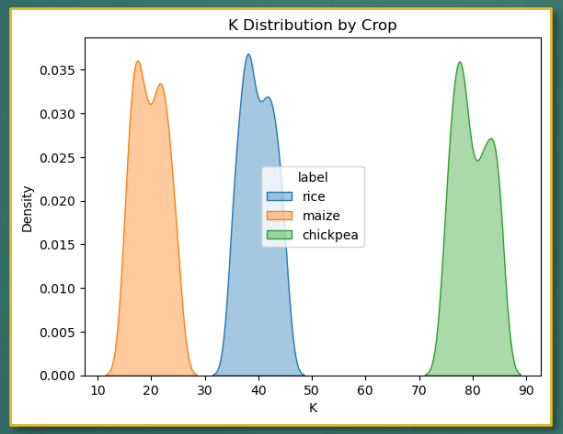
```
Samples: 1,400 | Features: 7 | Crops: 3
Train/Test Split: 12 / 113

Overall Accuracy: 98.2%  (111 / 113 correct)

Per-Crop Results:
• Rice      → 45 / 45  (100%)
• Maize     → 34 / 36  (94%)
• Chickpea  → 32 / 32  (100%)
```

Implementation in Python

Data Visualization

To understand why accuracy remained so high, we visualized feature distributions across crops.

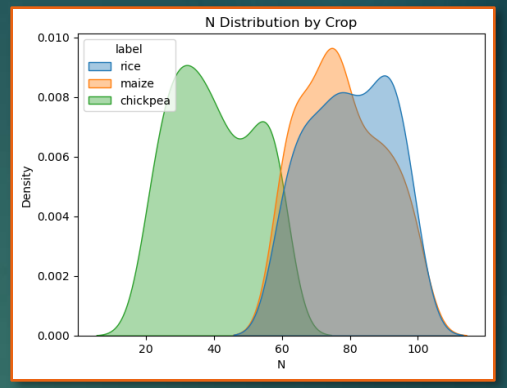
These plots show that crop classes are almost perfectly separated, especially by **Potassium (K):**

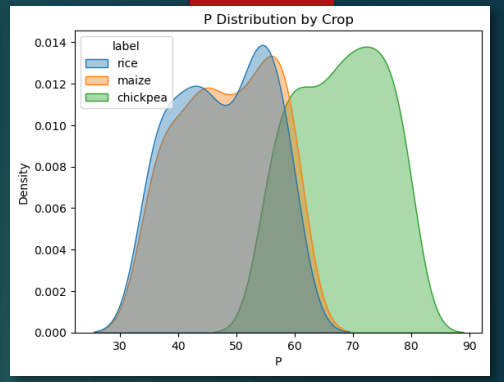


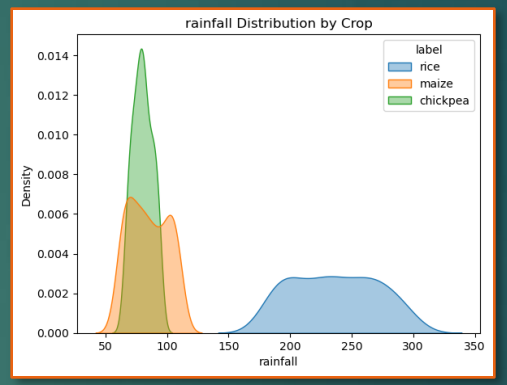
The K (Potassium) feature alone nearly predicts the crop identity.

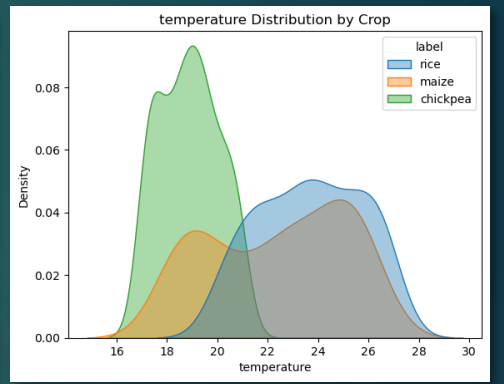
- In many cases, each crop occupies a distinct region of feature space
- This explains the model's near-perfect accuracy: it didn't need to "learn," just recognize clean boundaries.

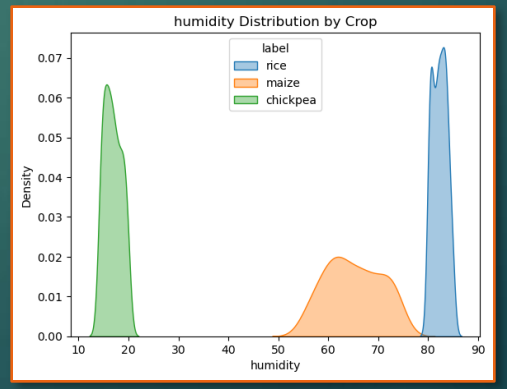
Real agricultural data would have overlapping nutrient ranges and noise, which would lower accuracy to realistic levels

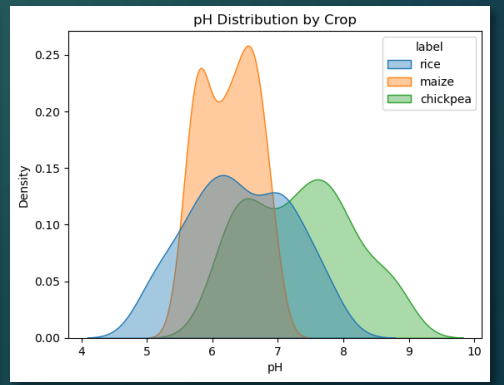












Related Work

- ***"Development of a Naive Bayes-based Framework for Optimizing Crop Recommendation System and Enhancing Agricultural Yield Prediction"***
- Naïve Bayes, Logistic Regression, Decision Tree, Random Forest
- Used similar dataset with soil nutrients and environmental factors
- Calculated priors, likelihoods and posterior probability
- *"most effective for accurately predicting suitable crop options based on a range of key parameters"*
- *"most effective due to its simplicity, efficiency, and robust performance with large datasets."*

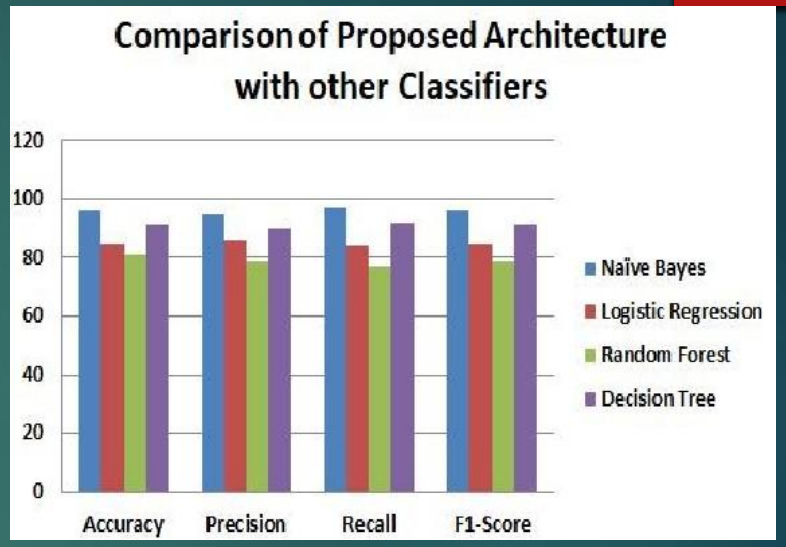


TABLE I

Performance Comparison of Proposed Architecture with other Classifiers

Algorithm	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	**96**	**95**	**97**	**96**
Logistic Regression	85	86	84	85
Random Forest	81	79	77	79
Decision Tree	91	90	92	91

Conclusion

- Hypothesis: The Naïve Bayes model can accurately predict the crop using the soil nutrient data and environmental conditions.
- It is simple, fast, and interpretable
- Next steps:
  - Larger dataset with live data
  - Additional features and data from other regions
- *"Future work could further augment the model's accuracy by incorporating real-time sensor data and considering economic factors, providing farmers with even more precise and actionable insights to improve productivity and reduce resource waste."*

Citations

P. Swetha and J. Senthilkumar, "Development of a Naive Bayes-based Framework for Optimizing Crop Recommendation System and Enhancing Agricultural Yield Prediction," *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, Bhimdatta, Nepal, 2025, pp. 1262-1267, doi: 10.1109/ICSADL65848.2025.10933086.