PROJECT I

TITLE:- CROP RECOMMENDATION SYSTEM USING MACHINE LEARNING

SEMESTER:- V CLASS:-TY BRANCH:-CSE(AIML)

PROJECT TEAM

SR.NO	NAME	ROLL NO
1	Gourav Jagannath Kumbhar	53
2	Aniket Machindra Yadav	54
3	Aditya Dhondiram Patil	57
4	Omkar Pratap Khapare	76

INDEX

PROBLEM STATEMENT

- Agriculture is important for the Global Economy and feeding the growing population.
- The problem is that Farmers often struggle to choose the right crops due to unpredictable weather and soil conditions..
- Choosing the right crops based on environmental conditions can greatly improve Agricultural productivity and profitability.
- Developing a supervised machine learning-based System for Recommending Best Crop for farming.
- Enhancing farmers decision-making for better harvests.

OBJECTIVES

- **The Objectives of the proposed work are as follows:**

1. Collect appropriate data and examine it.
2. Choosing a crop and maximize the yield.
3. Data cleaning and Exploratory Data Analysis (EDA) on the datasets.
4. To Make Crop selection process easier for farmers.
5. To Minimize difficulties faced by farmers
6. To Reduce financial risks of farmers.

TOOLS & METHODS

- **Tools :**
  - **Programming Languages:**
    - Python
  - **Machine Learning Libraries:**
    - Scikit-Learn ,Pandas,Numpy etc.
  - **Data Visualization:**
    - Matplotlib
    - Seaborn

TOOLS & METHODS

- **METHODS :**

1. Data Preprocessing

- Data Cleaning
- Data Wrangling
- Data Modeling
- Model Training
- Model Testing

2.Feature Selection

3.Machine Learning Algorithms:

- Logistic Regression
- Decision Tree
- Random Forests
- Support Vector Machines (SVM)
- K-Nearest Neighbors (KNN)

4.Model Evaluation

BASIC SYSTEM FLOW CHART

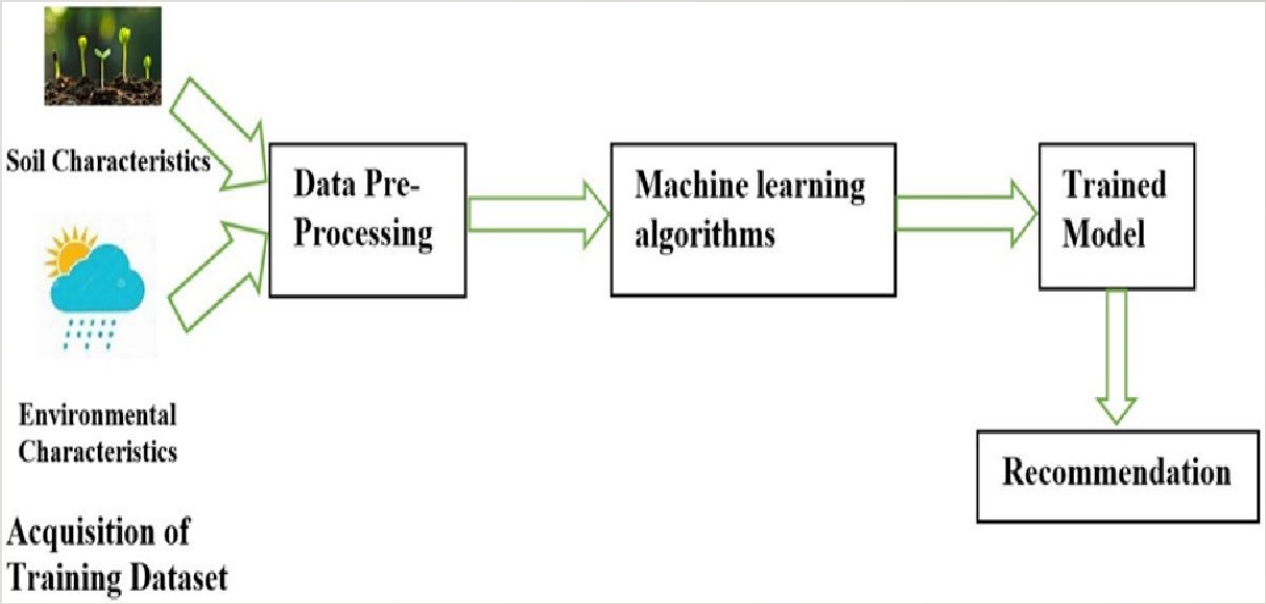


Fig. Basic System Flowchart

BASIC SYSTEM ARCHIT-ECHT URE

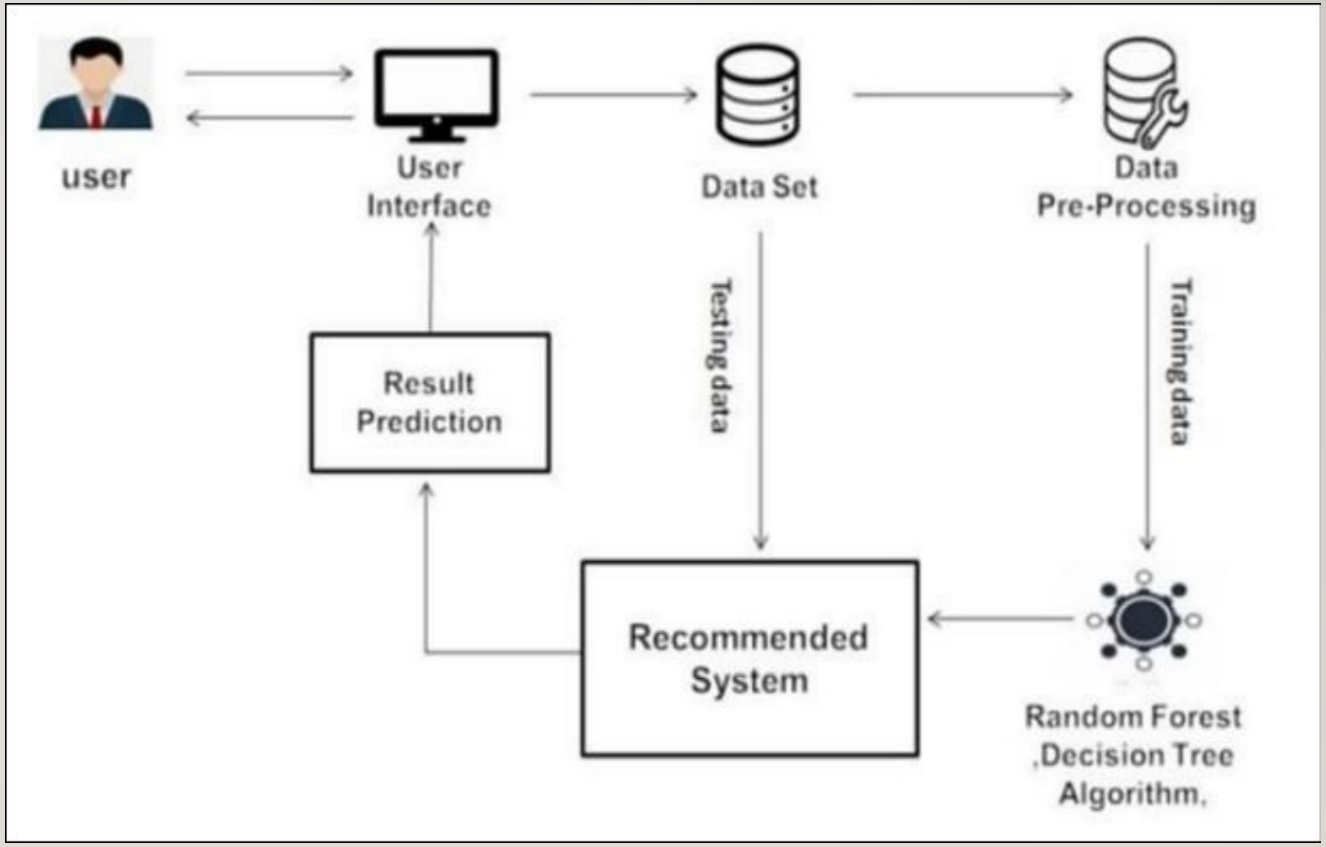


Fig. Basic System Architecture

MODULES

1. **Data Collection and Preparation:**
   - Gathered placement data from various sources such as GitHub, Kaggle.
   - Cleaned the data by handling missing values: impute missing values for numerical features (e.g., using mean or median) and decide on a strategy for categorical features.

2. **Model Selection:**
   - Researched and selected appropriate regression algorithms (Linear Regression, Decision Trees, Random Forest, Gradient Boosting, etc.).
   - Considered the trade-offs between interpretability and predictive performance of different algorithms.

MODULES

4. **Model Training:**
   - Splitted the dataset into training and testing sets (e.g., 70% for training and 30% for testing).
   - Trained selected regression model using the training data and chosen algorithms.
5. **Hyperparameter Tuning:**
   - Identified hyperparameters specific to each algorithm (e.g., learning rate, max depth, number of estimators).
   - Used techniques like grid search or random search to find the combination of hyperparameters that optimize model performance.
6. **Model Evaluated:**
   - Evaluated model performance on the testing set using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) score.
   - Visualized model predictions against actual prices to understand where the model performs well and where it struggles.

MODULES

7. **Cross-Validated:**
   - Implemented k-fold cross-validation to assess how well the model generalizes to new, unseen data.
   - Calculated average performance metrics across different folds to get a more reliable estimate of model performance.

8. **Documentation and Reporting:**
   - Created comprehensive documentation detailing the steps taken in each module.
   - Included explanations of data preprocessing, feature engineering, model selection, training, evaluation, and deployment.

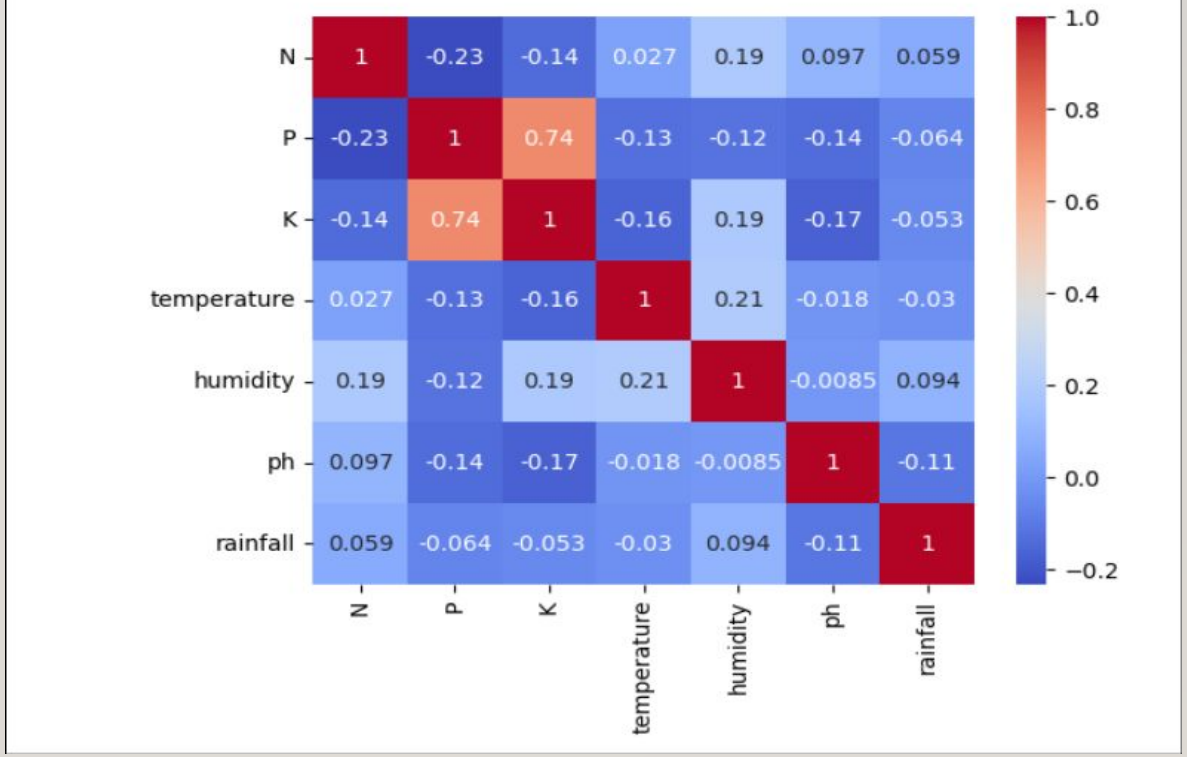
SYSTEM REQUIREMENTS

1. **Hardware Requirements:**
   - Processor: Intel Core i3
   - RAM: 8 GB
   - Hard Disc: Minimum 64GB
2. **Software Requirements:**
   - Operating System: Windows OS
   - Programming languages: Python
   - Database: Kaggle
   - Internet and Browsing Facilities
   - Jupyter notebook

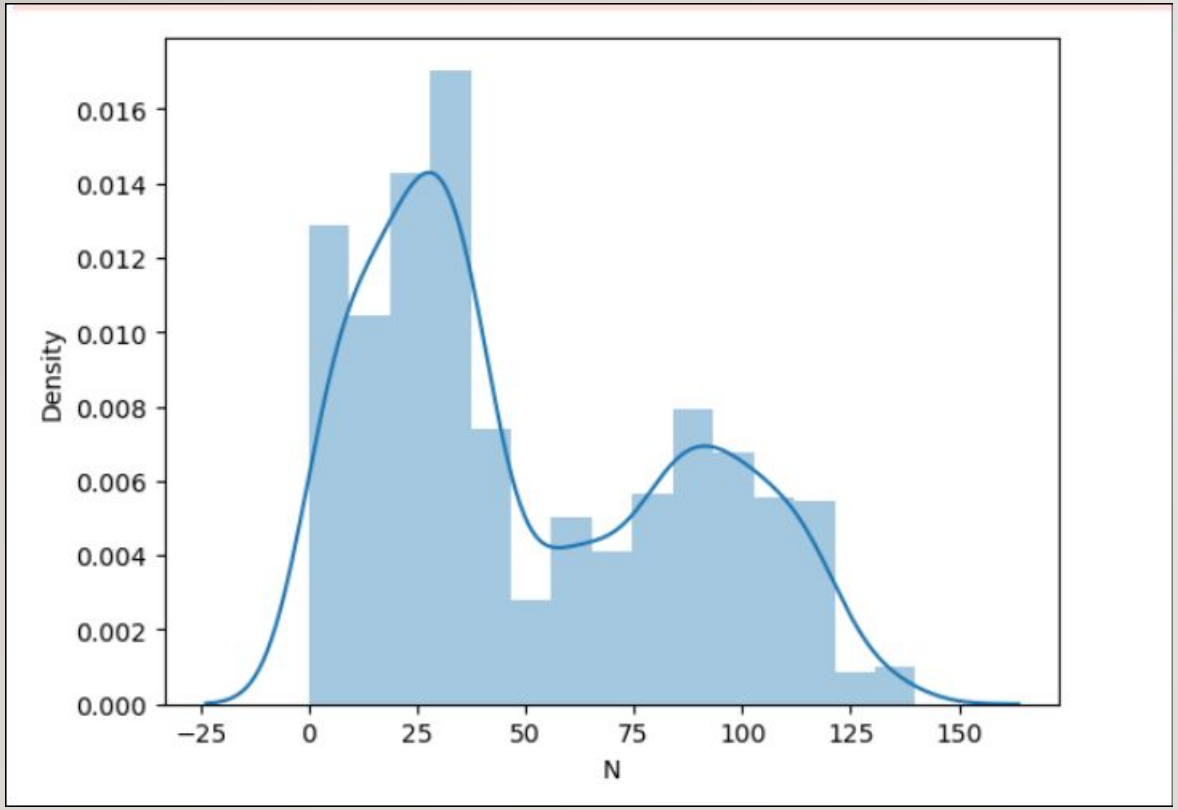
RESULT ANALYSIS

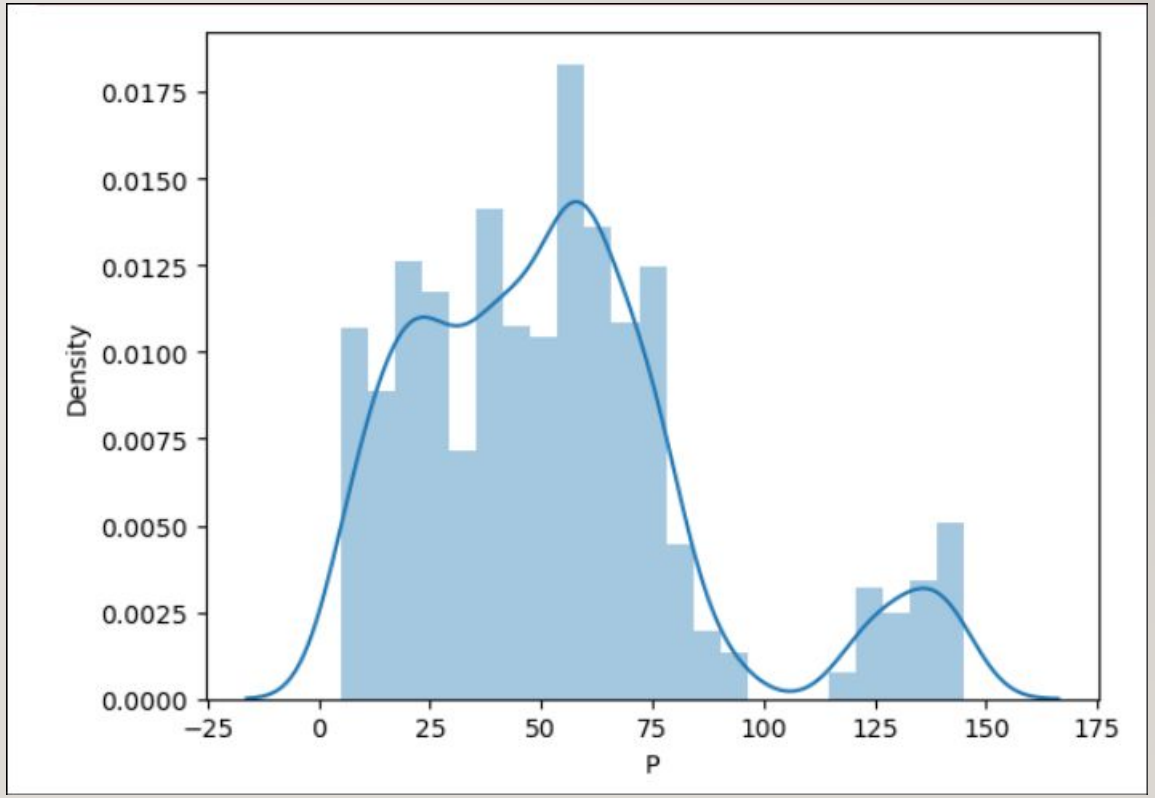
Data Visualization

```
In [10]: sns.heatmap(corr,annot=True,cbar=True,cmap='coolwarm')
Out[10]: <Axes: >
```



Distribution Plot :





Dataset Training ,Testing and Splitting :

Training , Testing and Splitting Dataset¶

In [16]:
```
X=dataset.drop('crop_num',axis=1)
Y=dataset['crop_num']
```

In [17]:
```
X.head()
```

Out[17]:

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

In [18]:
```
Y.head()
```

Out[18]:
```
0    1
1    1
2    1
3    1
4    1
Name: crop_num, dtype: int64
```

In [19]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42)
```

In [20]:
```
X_train.shape
```

Out[20]:
```
(1760, 7)
```

In [21]:
```
X_test.shape
```

Out[21]:
```
(440, 7)
```

In [22]:
```
X_train
```

Out[22]:

	N	P	K	temperature	humidity	ph	rainfall
1656	17	16	14	16.396243	92.181519	6.625539	102.944161
752	37	79	19	27.543848	69.347863	7.143943	69.408782
892	7	73	25	27.521856	63.132153	7.288057	45.208411
1041	101	70	48	25.360592	75.031933	6.012697	116.553145
1179	0	17	30	35.474783	47.972305	6.279134	97.790725
...	...	...	...	...	...	...	...
1638	10	5	5	21.213070	91.353492	7.817846	112.983436
1095	108	94	47	27.359116	84.546250	6.387431	90.812505
1130	11	36	31	27.920633	51.779659	6.475449	100.258567
1294	11	124	204	13.429886	80.086340	6.361141	71.400430
860	32	78	22	23.970814	62.355576	7.007038	53.409060

1760 rows × 7 columns

MODEL TRAINING:

```
In [ ]: from sklearn.ensemble import RandomForestClassifier

In [24]: classifier=RandomForestClassifier()
         classifier.fit(X_train,Y_train)
         ypred=classifier.predict(X_test)

In [25]: accuracy_score(Y_test,ypred)*100

Out[25]: 99.31818181818181
```

MODEL TESTING:

In [32]:
```
N = 30
P = 110
k = 20
temperature = 59.0
humidity = 56
ph = 80
rainfall = 100

predict = recommendation(N,P,k,temperature,humidity,ph,rainfall)

crop_dict = {1: "Rice", 2: "Maize", 3: "Jute", 4: "Cotton", 5: "Coconut", 6: "Papaya", 7: "Orange",
                8: "Apple", 9: "Muskmelon", 10: "Watermelon", 11: "Grapes", 12: "Mango", 13: "Banana",
                14: "Pomegranate", 15: "Lentil", 16: "Blackgram", 17: "Mungbean", 18: "Mothbeans",
                19: "Pigeonpeas", 20: "Kidneybeans", 21: "Chickpea", 22: "Coffee"}

if predict[0] in crop_dict:
    crop = crop_dict[predict[0]]
    print("{} is a best crop to be cultivated ".format(crop))
else:
    print("Sorry are not able to recommend a proper crop for this environment")
```

```
Pigeonpeas is a best crop to be cultivated
```

CONCLUSION

The project aimed to predict for Choosing the right crops based on local conditions can greatly improve agricultural productivity and profitability.

Developing a supervised machine learning-based system for Recommending best Crop for farming.

User-friendly system that recommends suitable crops based on environment conditions enhancing farmers' decision-making for better harvests.

REFERENCES

1. [1] Jiangshan M. Jehovah, Nikhil Gondaliya, Vinita Shah A Review on Data Mining Techniques for Fertilizer Recommendation 2018.

2. [2] M.C.S. Geetha A Survey on Data Mining Techniques in Agriculture, 2015.\ a-survey-on-data-mining-techniques-inagriculture.pdf (rroij.com) .

3. [3] S. Srija, R. Geetha Chanda, S .Lavanya, Dr. M. Kalpana Ph.D. Agro Nutri AndroidApplication,2016. Crop Yield Prediction Using Machine Learning Algorithm.pdf (ijaem.net)

4. [4] Karan deep Machine Learning: Applications in Indian Agriculture, 2016. Crop Yield Prediction Based On Indian Agriculture Using Machine Learning | Request PDF (researchgate.net) .

5. [5] Geeks for Geeks (How it works) - Simply Explained. [Online] Explain the working of HTTPS – Geeks for Geeks

THANK YOU!