

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELGAUM-590014**



**DATAMINING AND DATAWAREHOUSING ACTIVITY
REPORT ON**

**“CROP PREDICTION USING CLASSIFICATION
ALGORITHMS”**

For the Academic Year 2021-2022

**Submitted By
Bindu SN (1JS19IS023)
Nishrutha CG (1JS19IS056)**

**Under the guidance of
Dr. Malini M Patil
Associate Professor, ISE**



**DEPARTMENT OF INFORMATION SCIENCE OF ENGINEERING
JSS ACADEMY OF TECHNICAL
EDUCATION**

**JSSATEB Campus, Dr Vishnuvardhan Road, Uttrahalli Kengeri Main
Road.
Bangalore-560060**

PAGE INDEX

Sl.	Description	Page
1	I: INTRODUCTION	1
2	II: METHODOLOGY	2
3	III: WORKING OF SYSTEM	5
4	IV: EXPERIMENTAL ANALYSIS	16
5	V: SUMMARY	24
6	VI: CONCLUSION AND FUTURE WORK	25
7	VII: REFERENCES	26

ABSTRACT

Agriculture and its allied sectors are undoubtedly the largest providers of livelihoods in rural India. The agriculture sector is also a significant contributor factor to the country's Gross Domestic Product (GDP). Blessing to the country is the overwhelming size of the agricultural sector.

This project proposes a viable and user-friendly yield prediction system for the farmers. The user provides the soil type, temperature, humidity and rainfall as input. Machine learning algorithms allow choosing the most profitable crop list or predicting the crop yield for a user-selected crop. To predict the crop yield, selected Machine Learning algorithms such as Support Vector Machine (SVM), Gaussian Naïve Bayes (GNB), Logistic Regression, and K-Nearest Neighbour (KNN) are used. Among them, the Gaussian Naïve Bayes (GNB) showed the best results with 95% accuracy. The model with highest accuracy is deployed using streamlit.

I: INTRODUCTION

1.1 Introduction

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data. This project aims to predict the crop to be harvested by analyzing data of soil, temperature and other environmental factors which classifies what crop to be harvested using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. there is a common set of core factors that influence what crop to be grown. By collecting the data from various sources, classifying them under suitable headings and finally analyzing to extract the desired data, we can say that this technique can be very well adapted for the prediction of crop harvest.

1.2 Objective of the Project

The objective of the project is to automate crop prediction using machine learning classifier algorithms.

The main objective is to obtain a better variety of crops that can be grown over the season. The proposed system would help to minimize the difficulties faced by farmers in choosing a crop and maximize the yield.

1.2.1 Problem statement

To design a crop prediction web app using machine learning classification algorithms.

1.2.2 Problem description

The proposed model provides crop selection based on environmental conditions, . The proposed model predicts the crop yield by studying factors such as rainfall, temperature, area, season, soil type etc.

The user provides soil type and various environmental conditions as inputs. As per the input, the app recommends a suitable crop to harvest.

1.2.3 Applications of the proposed system

1) Can be employed in crop recommendation platforms.

II: METHODOLOGY

2.1 PROPOSED SYSTEM

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is pre-processed into the required format. The data is then divided into two parts : training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. This system is implemented using the following modules.

- 1.) Collection of Dataset
- 2.) Selection of attributes
- 3.) Data Pre-Processing
- 4.) Balancing of Data
- 5.) Disease Prediction

2.1.1 Collection of datasets

Initially, we collect a dataset for our crop recommender system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model.

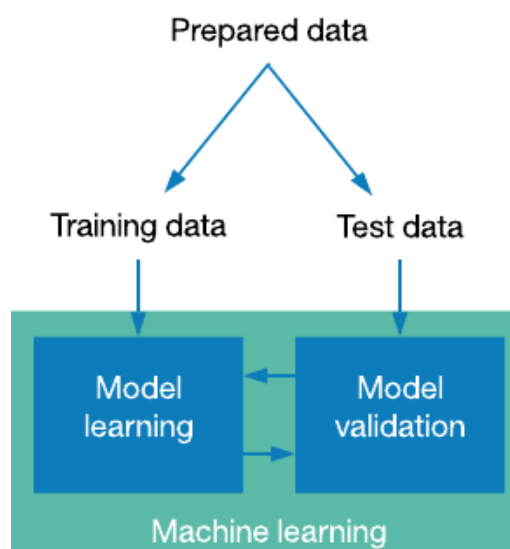


Figure 2.1: Collection of Data

Data Preparation

Source : Crop Recommendation dataset

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

The dataset is publicly available on the Kaggle website. This dataset was built by augmenting datasets of rainfall, climate and fertilizer data available for India. The classification goal is to predict which crop to harvest based on the input data: type of soil (N, P, K, pH), temperature, humidity, and rainfall. It includes over 2,000 records and 7 attributes.

2.1.2 Selection of attributes

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system.

Each attribute is a potential factor.

- N - ratio of Nitrogen content in soil
- P - ratio of Phosphorous content in soil
- K - ratio of Potassium content in soil
- temperature - temperature in degree Celsius
- humidity - relative humidity in %
- ph - ph value of the soil
- rainfall - rainfall in mm

Predict variable (desired target)

- label - name of the crop predicted

2.1.3 Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Pre-processing of data is required for improving the accuracy of the model.



Figure 2.2: Data Pre-processing

2.1.4 Balancing of Data

Imbalanced datasets can be balanced in two ways.

- (a) Under Sampling: In Under Sampling, dataset balance is done by the reduction of the size of the sample class. This process is considered when the amount of data is adequate.
- (b) Over Sampling: In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

2.1.5 Prediction of output

Various machine learning algorithms like SVM, Gaussian Naive Bayes, Decision Tree, Logistic Regression, K- Nearest Neighbours classifier are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for crop recommendation.

2.1.6 Deployment using streamlit

- Streamlit lets us to create apps for Machine Learning project using simple code.
- It also supports hot-reloading that lets the app update live as we edit and save your file.
- Using streamlit, creating an app is very easy, adding a widget is as simple as declaring a variable.

III: WORKING OF SYSTEM

3.1 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system. The working of this system is described as follows:

Dataset collection is collecting data which contains soil type, temperature, humidity and rainfall.

Attributes selection process selects the useful attributes for the prediction of type of crop.

After identifying the available data resources, they are further selected, cleaned, made into the desired form.

Different classification techniques as stated will be applied on preprocessed data to predict the type of crop.



Figure 3.1: System architecture

3.2 MACHINE LEARNING

In machine learning, classification refers to a predictive modelling problem where a class label is predicted for a given example of input data.

- **Supervised Learning**

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

● **Unsupervised learning**

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

● **Reinforcement learning**

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is well trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

3.2.1 Learners in Classification Problems

In classification problems, there are two types of learners:

1. **Lazy Learners:** Lazy Learner firstly stores the training dataset and wait until it receives the test dataset. In Lazy learner case, classification is done on the basis of the most related data stored in the training dataset. It takes less time in training but more time for predictions.

Example: K-NN algorithm, Case-based reasoning.

2. **Eager Learners:** Eager Learners develop a classification model based on a training dataset before receiving a test dataset. Opposite to Lazy learners, Eager Learner takes more time in learning, and less time in prediction. **Example:** Decision Trees, Naïve Bayes, ANN.

3.2.3 Types of ML Classification Algorithms

Classification Algorithms can be further divided into the Mainly two category:

- **Linear Models**
 - Logistic Regression
 - Support Vector Machines
- **Non-linear Models**
 - K-Nearest Neighbours
 - Kernel SVM
 - Naïve Bayes
 - Decision Tree Classification
 - Random Forest Classification

3.2.3.1 SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/ vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in

classification problems. They are extremely popular because of their ability to handle multiple continuous and categorical variables.

The followings are important concepts in SVM

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the 12 support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Types of SVM

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

The **advantages** of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The **disadvantages** of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

3.2.3.2 DECISION TREE ALGORITHM

Decision Tree is a Supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision Tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure. In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection.

Working:

In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the Decision Tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step - 3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

3.2.3.3 LOGISTIC REGRESSION ALGORITHM

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Advantages:

- Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.
- The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e., positive or negative is also given. So we can use Logistic Regression to find out the relationship between the features.
- This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.
- Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

Disadvantages:

- Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid overfitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.
- Nonlinear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world scenarios. So, the transformation of nonlinear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.
- Non-Linearly Separable Data: It is difficult to capture complex relationships using logistic regression. More powerful and complex algorithms such as Neural Networks can easily outperform this algorithm.

3.2.3.3 NAIVE BAYES ALGORITHM

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes' theorem

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where,

$P(A|B)$ is Posterior probability: Probability of hypothesis A on the observed event B.

$P(B|A)$ is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

$P(A)$ is Prior Probability: Probability of hypothesis before observing the evidence.

$P(B)$ is Marginal Probability: Probability of Evidence.

Types of Naive Bayes model

- **Gaussian:** The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

- **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as 15 Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

3.2.3.4 K NEAREST NEIGHBOURS ALGORITHM

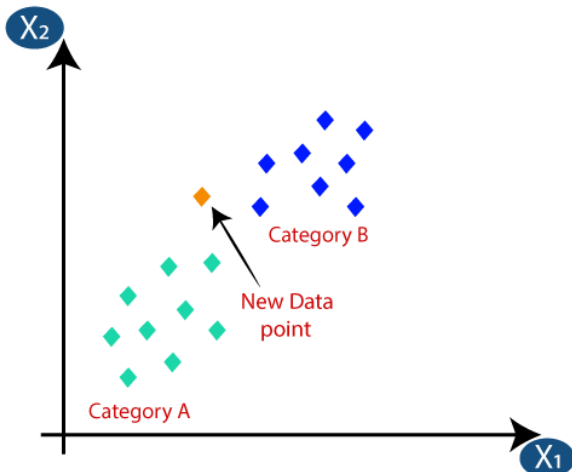
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The K-NN working can be explained on the basis of the below algorithm:

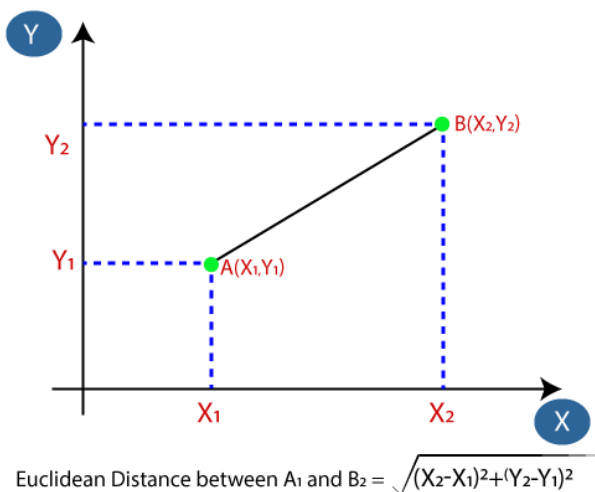
- **Step-1:** Select the number K of the neighbours
- **Step-2:** Calculate the Euclidean distance of **K number of neighbours**
- **Step-3:** Take the K nearest neighbours as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbours, count the number of the data points in each category.

- **Step-5:** Assign the new data points to that category for which the number of the neighbour is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



- Firstly, we will choose the number of neighbours, so we will choose the $k=5$.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



- By calculating the Euclidean distance, we got the nearest neighbours, as three nearest neighbours in category A and two nearest neighbours in category B. Consider the below image:



- As we can see the 3 nearest neighbours are from category A, hence this new data point must belong to category A.

IV: EXPERIMENTAL ANALYSIS

4.1 SYSTEM CONFIGURATION

4.1.1 Hardware requirements

Processor : Any Update Processor

Ram : Min 4GB

Hard Disk : Min 100GB

4.1.2 Software requirements

Operating System : Windows family

Technology : Python3.7

IDE : Visual Studio Code

4.2 DATASET DETAILS

- All the 7 attributes are considered for the prediction of the output.
- Crop-recommendation dataset: <https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset/Crop-recommendation.csv>

	A	B	C	D	E	F	G	H
1	N	P	K	temperatu	humidity	ph	rainfall	label
2	90	42	43	20.87974	82.00274	6.502985	202.9355	rice
3	85	58	41	21.77046	80.31964	7.038096	226.6555	rice
4	60	55	44	23.00446	82.32076	7.840207	263.9642	rice
5	74	35	40	26.4911	80.15836	6.980401	242.864	rice
6	78	42	42	20.13017	81.60487	7.628473	262.7173	rice
7	69	37	42	23.05805	83.37012	7.073454	251.055	rice
8	69	55	38	22.70884	82.63941	5.700806	271.3249	rice
9	94	53	40	20.27774	82.89409	5.718627	241.9742	rice
10	89	54	38	24.51588	83.53522	6.685346	230.4462	rice
11	68	58	38	23.22397	83.03323	6.336254	221.2092	rice
12	91	53	40	26.52724	81.41754	5.386168	264.6149	rice
13	90	46	42	23.97898	81.45062	7.502834	250.0832	rice
14	78	58	44	26.8008	80.88685	5.108682	284.4365	rice
15	93	56	36	24.01498	82.05687	6.984354	185.2773	rice
16	94	50	37	25.66585	80.66385	6.94802	209.587	rice
17	60	48	39	24.28209	80.30026	7.042299	231.0863	rice
18	85	38	41	21.58712	82.78837	6.249051	276.6552	rice
19	91	35	39	23.79392	80.41818	6.97086	206.2612	rice
20	77	38	36	21.86525	80.1923	5.953933	224.555	rice
21	88	35	40	23.57944	83.5876	5.853932	291.2987	rice
22	89	45	36	21.32504	80.47476	6.442475	185.4975	rice
23	76	40	43	25.15746	83.11713	5.070176	231.3843	rice
24	67	59	41	21.94767	80.97384	6.012633	213.3561	rice
25	83	41	43	21.05254	82.6784	6.254028	233.1076	rice
26	98	47	37	23.48381	81.33265	7.375483	224.0581	rice
27	66	53	41	25.07564	80.52389	7.778915	257.0039	rice
28	97	59	43	26.35927	84.04404	6.2865	271.3586	rice
29	97	50	41	24.52923	80.54499	7.07096	260.2634	rice
<div> <div> <div></div> <div></div> </div> <div>Crop_recommendation</div> <div>+</div> </div>								

Figure 4.1: Dataset Attributes

Dataset attributes

Sl.	Attribute	Description	Datatype
1	N	Ratio of nitrogen content in the soil	Int
2	P	Ratio of phosphorous content in the soil	Int
3	K	Ratio of potassium content in the soil	Int
4	Temperature	Temperature in Celsius	Float
5	Humidity	Humidity in %	Float
6	pH	pH value of soil	Float
7	Rainfall	Rainfall in mm	Float
8	Label	Name of the crop predicted	object

4.3 PERFORMANCE ANALYSIS

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, KNN, Logistic Regression are used for the prediction. All the 7 attributes are considered for the prediction of the output. The accuracy for individual algorithms is measured and whichever algorithm is giving the best accuracy, that is considered for the crop prediction. For evaluating the experiment, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered.

TP: True positive FP: False Positive FN: False Negative TN: True Negative

4.3.1 Accuracy

Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset.

It is expressed as: $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN})$

4.3.2 Confusion matrix

It gives us a matrix as output and gives the total performance of the system.

```
=== Confusion Matrix ===
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v  <-- classified as
92  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8  0 | a = rice
 0 99  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 | b = maize
 0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | c = chickpea
 0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | d = kidneybeans
 0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | e = pigeonpeas
 0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | f = mothbeans
 0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | g = mungbean
 0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | h = blackgram
 0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0  0 | i = lentil
 0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0  0 | j = pomegranate
 0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0  0 | k = banana
 0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0  0 | l = mango
 0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0  0 | m = grapes
 0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0  0 | n = watermelon
 0  0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0  0 | o = muskmelon
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0  0 | p = apple
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0  0 | q = orange
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0  0 | r = papaya
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0  0 | s = coconut
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 100  0  0 | t = cotton
 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 98  0 | u = jute
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 100 | v = coffee
```

Figure 4.2: Confusion matrix

4.3.3 Precision

It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as: $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$

4.3.4 Recall

It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as: $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$

4.3.5 F1 Score

It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1.

It can be expressed as: $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$

Training Accuracy Score: 99.5%				
Validation Accuracy Score: 99.3%				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	18
banana	1.00	1.00	1.00	18
blackgram	1.00	1.00	1.00	22
chickpea	1.00	1.00	1.00	23
coconut	1.00	1.00	1.00	15
coffee	1.00	1.00	1.00	17
cotton	1.00	1.00	1.00	16
grapes	1.00	1.00	1.00	18
jute	0.88	1.00	0.93	21
kidneybeans	1.00	1.00	1.00	20
lentil	1.00	1.00	1.00	17
maize	1.00	1.00	1.00	18
mango	1.00	1.00	1.00	21
mothbeans	1.00	1.00	1.00	25
mungbean	1.00	1.00	1.00	17
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	23
papaya	1.00	1.00	1.00	21
pigeonpeas	1.00	1.00	1.00	22
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.88	0.94	25
watermelon	1.00	1.00	1.00	17
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 4.3: Evaluation metrics

4.4 PERFORMANCE MEASURES

4.4.1 Logistic Regression

```

▶ pipeline = make_pipeline(StandardScaler(), LogisticRegression())
model = pipeline.fit(X_train, y_train)
y_pred = model.predict(X_test)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_metrics(pipeline, conf_matrix)

```

Training Accuracy Score: 97.5%
Validation Accuracy Score: 96.4%

4.4.2 K Nearest Neighbours

```
▶ pipeline = make_pipeline(StandardScaler(), KNeighborsClassifier())  
model = pipeline.fit(X_train, y_train)  
y_pred = model.predict(X_test)  
conf_matrix = confusion_matrix(y_test, y_pred)  
classification_metrics(pipeline, conf_matrix)
```

↳ Training Accuracy Score: 98.4%
Validation Accuracy Score: 97.7%

4.4.3 Decision Tree

```
▶ pipeline = make_pipeline(StandardScaler(), DecisionTreeClassifier())  
model = pipeline.fit(X_train, y_train)  
y_pred = model.predict(X_test)  
conf_matrix = confusion_matrix(y_test, y_pred)  
classification_metrics(pipeline, conf_matrix)
```

Training Accuracy Score: 100.0%
Validation Accuracy Score: 98.9%

4.4.4 Gaussian NB

```
▶ pipeline = make_pipeline(StandardScaler(), GaussianNB())  
model = pipeline.fit(X_train, y_train)  
y_pred = model.predict(X_test)  
conf_matrix = confusion_matrix(y_test, y_pred)  
classification_metrics(pipeline, conf_matrix)
```

↳ Training Accuracy Score: 99.5%
Validation Accuracy Score: 99.3%

4.4.5 Support Vector Classifier

```
▶ pipeline = make_pipeline(StandardScaler(), SVC())  
model = pipeline.fit(X_train, y_train)  
y_pred = model.predict(X_test)  
conf_matrix = confusion_matrix(y_test, y_pred)  
classification_metrics(pipeline, conf_matrix)
```

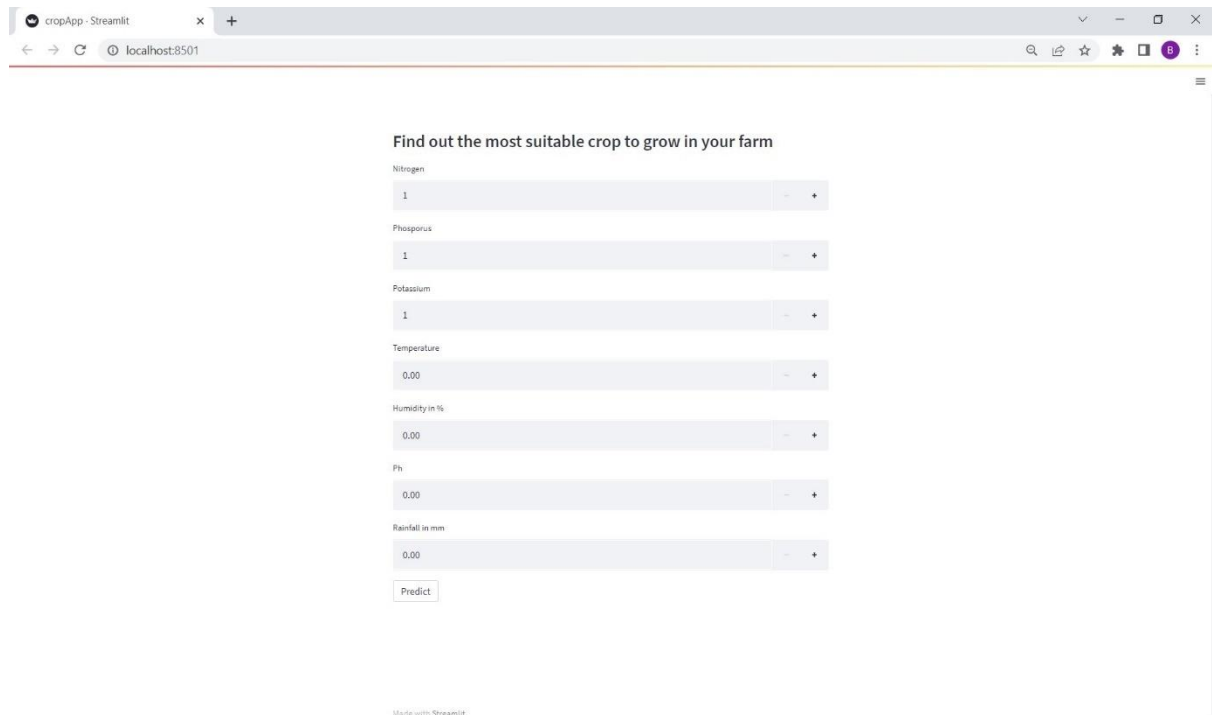
Training Accuracy Score: 98.8%
Validation Accuracy Score: 97.7%

After performing the machine learning approach for training and testing we find that accuracy of the Gaussian Naïve Bayes is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the number count of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that GaussianNB is the best with 99.3% accuracy.

Training Accuracy Score: 99.5%				
Validation Accuracy Score: 99.3%				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	18
banana	1.00	1.00	1.00	18
blackgram	1.00	1.00	1.00	22
chickpea	1.00	1.00	1.00	23
coconut	1.00	1.00	1.00	15
coffee	1.00	1.00	1.00	17
cotton	1.00	1.00	1.00	16
grapes	1.00	1.00	1.00	18
jute	0.88	1.00	0.93	21
kidneybeans	1.00	1.00	1.00	20
lentil	1.00	1.00	1.00	17
maize	1.00	1.00	1.00	18
mango	1.00	1.00	1.00	21
mothbeans	1.00	1.00	1.00	25
mungbean	1.00	1.00	1.00	17
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	23
papaya	1.00	1.00	1.00	21
pigeonpeas	1.00	1.00	1.00	22
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.88	0.94	25
watermelon	1.00	1.00	1.00	17
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 4.4: Evaluation metrics of GaussianNB

4.5 RESULTS



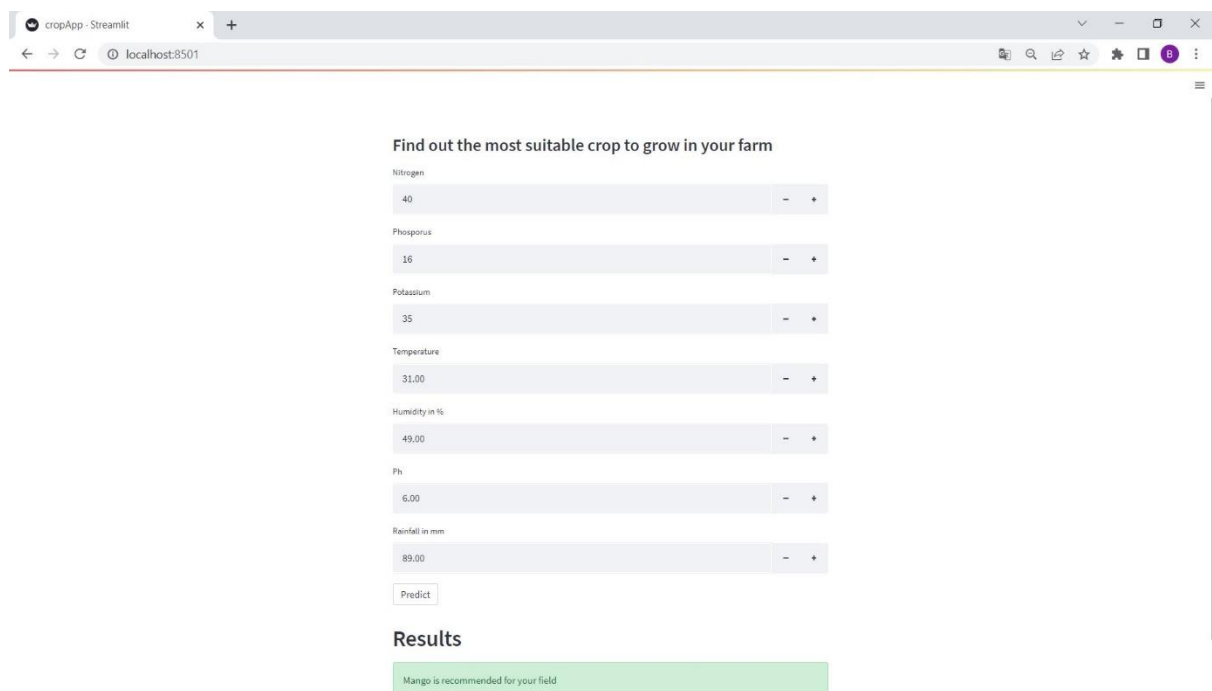
The screenshot shows a web browser window with the address bar displaying 'localhost:8501'. The page title is 'cropApp - Streamlit'. The main heading is 'Find out the most suitable crop to grow in your farm'. Below this, there are eight input fields, each with a label, a value, and minus/plus buttons. The inputs are: Nitrogen (1), Phosphorus (1), Potassium (1), Temperature (0.00), Humidity in % (0.00), Ph (0.00), Rainfall in mm (0.00), and a 'Predict' button. At the bottom, it says 'Made with Streamlit'.

Factor	Value
Nitrogen	1
Phosphorus	1
Potassium	1
Temperature	0.00
Humidity in %	0.00
Ph	0.00
Rainfall in mm	0.00

Predict

Made with Streamlit

Figure 4.5: App view



The screenshot shows the same web browser window as Figure 4.5, but with different input values. The inputs are: Nitrogen (40), Phosphorus (16), Potassium (35), Temperature (31.00), Humidity in % (49.00), Ph (6.00), and Rainfall in mm (89.00). The 'Predict' button is still present. Below the inputs, there is a 'Results' section with a green box containing the text 'Mango is recommended for your field'.

Factor	Value
Nitrogen	40
Phosphorus	16
Potassium	35
Temperature	31.00
Humidity in %	49.00
Ph	6.00
Rainfall in mm	89.00

Predict

Results

Mango is recommended for your field

Figure 4.6: Predicting the result

V: SUMMARY

The self-learning activity helped us in better understanding of the ETL process.

This activity helped us to investigate crop prediction with machine learning. The prediction accuracy varies for different scale, environmental conditions and crop.

The choice of features is dependent on the availability of the dataset and the aim of the research.

To find the best performing model, models with more and fewer features should be tested. Many algorithms have been used for the purpose. The most used models are the decision tree, logistic regression and SVM.

The results show that Gaussian Naïve Bayes model shows greater efficiency.

VI: CONCLUSION AND FUTURE WORK

This project proposes a viable and user-friendly yield prediction system for the farmers. The user provides the soil type, temperature, humidity and rainfall as input. Machine learning algorithms allow choosing the most profitable crop list or predicting the crop yield for a user-selected crop.

To predict the crop yield, selected Machine Learning algorithms such as Support Vector Machine (SVM), Guassian Naïve Bayes (GNB), Logistic Regression, and K-Nearest Neighbour (KNN) are used. Among them, the Guassian Naïve Bayes (GNB) showed the best results with 95% accuracy.

This project can be further designed such that the crop disease can be detected along with the possible prediction of the yield.

VII: REFERENCES

- [1] https://www.researchgate.net/publication/343730263_Crop_yield_prediction_using_machine_learning_A_systematic_literature_review
- [2] <http://www.kaggle.com/>
- [3] <https://www.javatpoint.com/>
- [4] <https://towardsdatascience.com/>
- [5] <https://streamlit.io/>
- [6] <https://www.datacamp.com/tutorial/streamlit>
- [7] <https://stackoverflow.com/>