

**“CROP RECOMMENDATION**

**PREDICTIVE ANALYSIS**

**MACHINE LEARNING”**

**PROJECT REPORT**

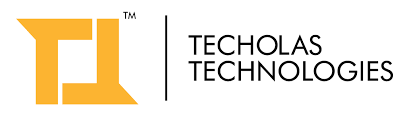
By

**ANGEL JACOB**

**DATA ANALYST**

UNDER THE GUIDANCE AND SUPERVISION OF

**Miss ARUNA**

****

Kochi, Kerala, 682025

**CONTENTS**

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE NO:** |
| ABSTRACT | 1 |
| INTRODUCTION | 2 |
| OBJECTIVES | 3 |
| DATA ANALYSIS | 4 |
| DATA INTERPRETATION | 6 |
| SUGGESTIONS | 7 |
| CONCLUSION | 8 |
| APPENDIX I | 9 |
| APPENDIX II | 10 |
| REFERENCE | 13 |

**ABSTRACT**

Crop Recommendation System for agriculture is based on various input parameters. This proposes a hybrid model for recommending crops to south Indian states by considering various attributes such as soil type, Rainfall, Groundwater level, Temperature, Fertilizers, Pesticides and season.

The recommender model is built as a hybrid model using the classifier machine learning algorithm. Based on the appropriate parameters, the system will recommend the crop. Technology based crop recommendation system for agriculture helps the farmers to increase the crop yield by recommending a suitable crop for their land with the help of geographic and the climatic parameters.

The proposed hybrid recommender model is found to be effective in recommending a suitable crop. Crop yield production value updation has a positive practical significance for guiding agricultural production and for notifying the change in market rate of crop to the farmer.

The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So the quality of the crops are identified using ranking process. By this process the rate of the low quality and high quality crop is also notified.

The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of crop which is yielded for further

**INTRODUCTION**

In recent years, the use of machine learning techniques in agriculture has gained significant attention. These techniques have proven to be beneficial in various agricultural applications such as crop recommendation systems, disease detection, yield prediction, and many more. The aim of this project is to develop a crop recommendation system using machine learning techniques.

The crop recommendation system developed in this project aims to provide farmers with recommendations on which crop to plant based on several factors such as soil type, weather conditions, and the farmer's location. The system uses several machine learning algorithms such as Logistic Regression, Random Forest, and Decision Tree to predict the most suitable crop for a given set of conditions.

The project uses various Python libraries such as NumPy, Pandas, Matplotlib, Scikit-learn, Seaborn, and Plot for data visualization, and machine learning model development. These libraries provide a wide range of functions that make it easy to develop and implement machine learning algorithms.

**OBJECTIVES**

* **To predict suitable crops based on soil and environmental conditions**

This means using machine learning to analyse data like soil type, temperature, rainfall, humidity, and other environmental factors to suggest which crop will grow best in a particular area. This helps in making informed decisions rather than guessing or using traditional methods.

* **To analyse various machine learning models for improving prediction accuracy.**

This involves comparing different machine learning algorithms (like Random Forest, Logistic Regression, Decision Tree, etc.) to find out which one gives the best results in predicting suitable crops. The goal is to select the most accurate model for reliable recommendations.

* **To enhance crop yield, minimize resource wastage and increase profitability for farmers.**

By recommending the most suitable crop, the system helps in increasing production (crop yield), using fewer resources (like water, fertilizer, and labour), and ultimately helping farmers earn more profit. It makes farming more efficient and sustainable.

Machine learning model analyse data such as soil, weather condition and other factors to provide accurate prediction. This helps in selecting right crop, applying correct amount of fertilizers and water. As a result improve productivity, inputs are used efficiently and farming become more profitable.

**DATA ANALYSIS**

**PROCEDURE**

1. **Import Essential Libraries**

* Import pandas, numpy for data handling.
* Use matplotlib, seaborn for data visualization.
* Load scikit-learn models for machine learning tasks.

1. **Load the Data**

* Read the dataset using pandas.read\_csv().
* Display dataset structure using .info(), .head(), and .describe().

1. **Check for Missing Values and Clean Data**

* Identify missing values using .isnull().sum().
* Handle missing values through imputation (mean/median/mode) or removal.

1. **Explore and Visualize Data**

* Use .value\_counts() to check categorical variable distributions.
* Plot correlation heatmaps, and barchart to understand data trends.

1. **Assign Dependent and Independent Variables**

* Dependent Variable: status (crop type).
* Independent Variables: nitrogen, phosphorus, potassium, temperature, humidity, pH and rainfall.

1. **Encode Categorical Variables**

* Convert categorical data (e.g., label) using Label Encoding

1. **Split the Data (Train-Test Split)**

* Use train\_test\_split() to divide data into 80% training and 20% testing.

1. **Train Models Using Classification Algorithms**

* Apply multiple classification models to predict crop: Logistic Regression Random Forest Classifier and Decision Tree Classifier

1. **Evaluate Model Performance**

* Use accuracy, precision, recall, and F1-score to compare model effectiveness.

**10. Make Predictions and Generate Insights**

* Input test data into the best-performing model.
* Predict suitable crop.

1. **ABOUT DATASET**

The dataset comprising the soil specific attributes which are collected from Kaggle. In addition, similar online sources of general crop data were also used. The crops considered in our model include rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mungbean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, coffee gives an analysis of the dataset. The number of instances of each crop available in the training dataset is depicted. The attributes considered where Nitrogen(N), Potassium(K), Phosphorus(P), Temperature, Humidity, Ph and Rainfall.

**Data fields**

* **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

**II. METHODOLOGY**

1. **Jupyter Notebook**

* Jupyter Notebook is used for programming in Python for machine learning tasks, including data preprocessing, analysis, and model building.

1. **Libraries**

* Pandas – Used for data cleaning, transformation, analysis, visualization, and input/output operations.
* NumPy – Supports numerical operations, including array manipulations, mathematical functions, and statistical calculations.
* Matplotlib & Seaborn – Used for data visualization, trend analysis, and graphical representation of stress factors.

1. **Machine Learning Models for Classification**

* Logistic Regression – Used for binary classification tasks such as predicting crop types
* Decision Tree Classifier
* Random Forest Classifier –multiple decision trees to classify soil condition

1. **Model Evaluation & Preprocessing**

* Accuracy Score – Evaluates the classification models’ performance in predicting suitable crop
* Confusion Matrix
* Classification Report
* Train-Test Split – Splits the dataset into training and testing sets for model evaluation, reducing overfitting.
* Label Encoding – Converts categorical data (e.g., gender, department) into numerical values for machine learning models.

**III. DATA INTERPRETATION**

The analysis aims to identify the relationship between the dependent variable and several independent variables using machine learning models. The goal is to find best crop type based on soil condition and environmental condition

1. **Model Training and Evaluation**

• Dependent Variable: Crop type

• Independent Variables:

* 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

**SUGGESTIONS**

* **User feedback mechanism**

We will add feedback mechanism where farmers can provide input on the recommended crops and their performance, enabling continuous learning of the system

* **Mobile application development**

We will develop a user friendly mobile application that allows farmers to access crop recommendations, receive notification or alerts, and interact with the system on the go

* **Customized recommendations**

Provide options for farmers to specify their preferences and constraints (eg: organic farming, budget constraints) for recommendations to their specify needs

* **Crop rotation planning**

We will expand the system to include recommendations for crop rotation schedules to improve soil fertility, pest management, and overall farm sustainability

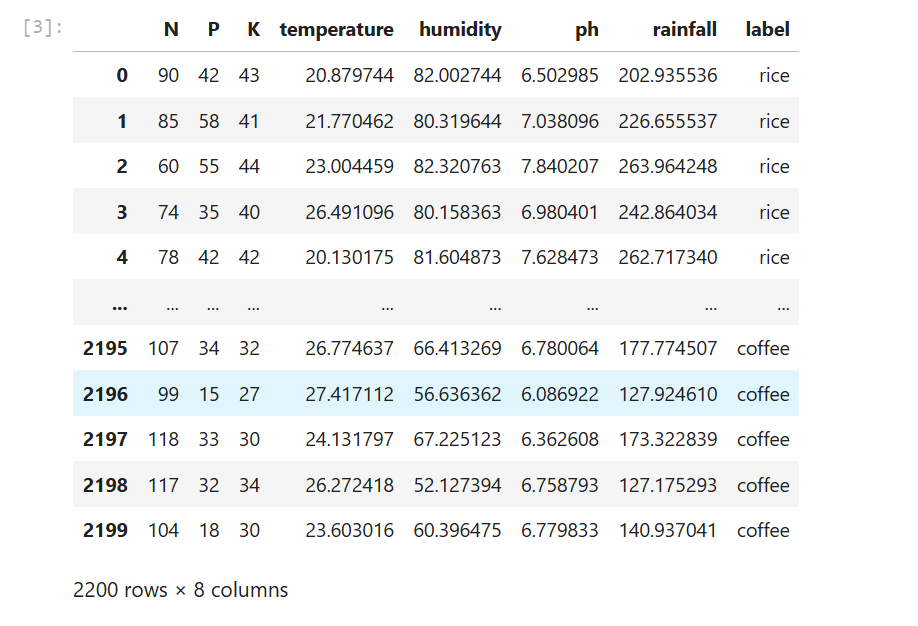
**CONCLUSION**

The Crop Recommendation System is a machine learning-based tool designed to recommend the most suitable crop to cultivate based on various soil and environmental factors. By leveraging Random Forest Classification, this system achieves an impressive accuracy of **99.31%**. It uses key parameters such as nitrogen (N), phosphorus (P), potassium (K), temperature (°C), humidity (%), pH, and rainfall (mm) to predict the ideal crop for a given set of conditions.

Throughout this project, we have developed crop recommendation system that provide farmers with personalized recommendations for their specific agricultural conditions. Our system not only addresses the challenges faced by the traditional methods, but also holds importance for the future of agriculture. Together we are revolutionizing agriculture for a more resilient and prosperous future.

The crop recommendation system developed in this project is a promising tool for improving agricultural practices. Further research can be done to improve the accuracy of the system by incorporating more data sources and using more advanced machine learning algorithms. The ultimate goal is to develop a system that can accurately predict the yield of different crops and provide farmers with recommendations on how to maximize their yield while minimizing their environmental impact.

**APPENDIX I**

****

**APPENDIX II**

import pandas as pd

data=pd.read\_csv(" Crop\_recommendation.csv")

data.head()

data.shape

data.tail()

data.info()

data.isnull().sum()

data.describe()

data.duplicated().sum()

**Visualization**

#**create Heatmap**

plt.figure(figsize=(10,7))

corr=data[['N','P','K','temperature','humidity','ph','rainfall']].corr()

sns.heatmap(corr,annot=True,linewidths=0.5,cmap='coolwarm')

plt.title('Correlation Heatmap',fontsize=20)

plt.show()

**#Histplot**

plt.hist(data['temperature'],color="blue",edgecolor="black")

plt.xlabel("Temperature in Celsius")

plt.ylabel("Frequency")

plt.title("Temperature Distribution")

plt.show()

#**Histplot**

plt.hist(data['rainfall'],color="green",edgecolor="black")

plt.xlabel("Rainfall in mm")

plt.ylabel("Frequency")

plt.title("Rainfall Distribution")

plt.show()

**Label Encoding**

from sklearn.preprocessing import LabelEncoder

obj=LabelEncoder()

data['label']=obj.fit\_transform(data['label'])

data

**Finding value counts in a column**

data['label'].value\_counts()

**Separate dependent and independent variable**

x=data.drop('label',axis=1)

x.head()

y=data.label

y.head()

**Model training and evaluating:**

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=42)

**Import the models**

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

**Scores**

from sklearn.metrics import accuracy\_score,confusion\_matrix,classification\_report

#**Accuracy Score**

ac\_score=accuracy\_score(y\_test,y\_pred)

ac\_score

**#Confusion Matrix**

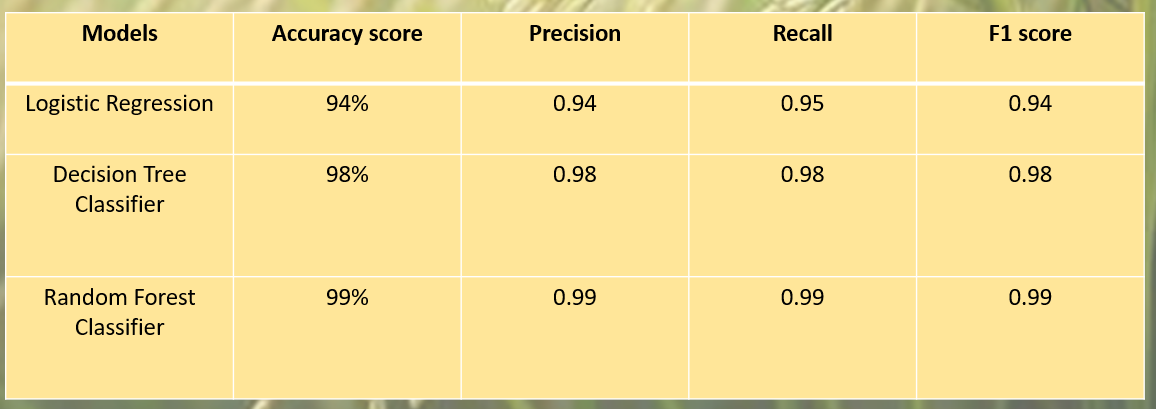
cm=confusion\_matrix(y\_test,y\_pred)

print(cm)

**#Classification Report**

cr=classification\_report(y\_test,y\_pred)

print(cr)



**References**

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