# NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY,

BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

# DATA MINING PROJECT LA2 REPORT

on

# CROP RECOMMENDATION PREDICTION

*submitted in partial fulfilment of the requirement for the award of Degree of*

*Bachelor of Engineering*

*in*

*Computer Science and Engineering*

*Submitted by:*


| | |
|---|---|
| AABHASH MANANDHAR | 1NT19CS004 |
| CHIRAG JUNG THAPA | 1NT19CS060 |
| PRAJESH SHRESTHA | 1NT19CS139 |
| UTSAV SAPKOTA | 1NT19CS206 |


## Department of Computer Science and Engineering

2021-22

**Nitte Meenakshi Institute of Technology**
(AN AUTONOMOUS INSTITUTION AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM)
PB No. 6429, Yelahanka, Bangalore 560Telephone: 080- 22167800, 22167860-064, Karnataka
Fax: 080 - 22167805

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CERTIFICATE

This is to certify that the ***Crop recommendation prediction*** is an authentic work carried out by Aabhash Manandhar**(1NT19CS004), Chirag Jung Thapa(1NT19CS060), Prajesh Shrestha(1NT19CS139), and Utsav Sapkota(1NT19CS206)** bonafide students of **Nitte Meenakshi Institute of Technology**, Bangalore in partial fulfilment for the award of the degree of **Bachelor of Engineering** in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belgavi during the academic year **2021-2022.** It is certified that all corrections and suggestions indicated during the internal assessment have been incorporated in the report. This project has been approved as it satisfies the academic requirement in respect of project work presented for the said degree.

| **Internal Guide** | **Signature of the HOD** | **Signature of Principal** |
|---|---|---|
| Vani V<br>Professor, Dept. CSE, NMIT<br>Bangalore | Dr. Sarojadevi H.<br>Professor, Head, Dept. CSE, NMIT<br>Bangalore | Dr. H. C. Nagaraj<br>Principal,<br>NMIT,<br>Bangalore |

# DECLARATION

We hereby declare that

(i) This report does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the report and in the References sections.

(ii) All corrections and suggestions indicated during the internal presentation have been incorporated in the report.

(iii) Content of the report has been checked for the plagiarism requirement

| Name | USN | Signature |
|---|---|---|
| Aabhash Manandhar | 1NT19CS004 | |
| Chirag Jung Thapa | 1NT19CS060 | |
| Prajesh Shrestha | 1NT19CS139 | |
| Utsav Sapkota | 1NT19CS206 | |

Date:

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. I/we express my/our sincere gratitude to our Principal **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

I/we wish to thank our HoD**, Dr. Sarojadevi H.** for the support and encouragement for the project work. I/We also thank him for the invaluable guidance provided which has helped in the creation of a better technical report.

We thank **Vani V, Professor**, for the guidance and support given while doing this project work and preparing the report & presentation. I/We also thank all our friends, teaching and nonteaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the project.

Name & USN:

| Name | USN | Signature |
|---|---|---|
| Aabhash Manandhar | 1NT19CS004 | |
| Chirag Jung Thapa | 1NT19CS060 | |
| Prajesh Shrestha | 1NT19CS139 | |
| Utsav Sapkota | 1NT19CS206 | |

Date:

# ABSTRACT

There are farmers worldwide with the problem of wrong crops plantation in wrong time of the year. This can cause a lot of farmers to waste their fertilizers and not produce the right crop. To deal with this matter we have come with a model which predicts the crop to be planted on the basis of different parameters like temperature, pH value of the soil, humidity, the percentages of nitrogen and different other chemicals in the soil.

With the technique of data mining we are able to predict the suitable crop according to the parameters. We train the model with the dataset and expect a suitable answer when we ourself input the following parameters.

In this project, we help the farmers to get informed decision about the farming strategy. The algorithms used for data mining are Random Forests, Decision Tree Algorithms. We train our model using the data set provided from Kaggle which in turn helps us build predictions that are close to the accuracy that the experts make. After making a prediction we compare the predicted answer with the actual value and adjust the changes.

# TABLE OF CONTENTS

# INTRODUCTION

We want this project to be specially helpful for the farmers cause this is a project modelled based on their work. Our project is still in development state and is yet to be improved. Our project is suitable for farmers that are growing certain crops like maize and other staple food of south-asian countries.

Our project may not provide prediction for other types of crops but it covers most of the popular crops. This will still help most of the farmers growing common crops and it will help them not waste their fertilizers. The farmers will know which crop will grow and which will not from our software.

## 1.1 MOTIVATION

Agriculture in India plays a predominant role in economy and employment. The common problem existing among the Indian farmers are they don't choose the right crop based on their soil requirements. Due to

this they face a serious setback in productivity. This problem of the farmers has been addressed through precision agriculture. Precision agriculture is a modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection and suggests the farmers the right crop based on their site specific parameters. This reduces the wrong choice on a crop and increase in productivity. Agriculture has been practised in India for a long period of time and it has really been hard for farmers to choose the right crop. That is why we developed this project to help them yield better crops.

## 1.2 PROBLEM DOMAIN

The domain we use for this project are Data Prediction and Machine Learning Techniques. Predicting the value of a certain data on the basis of previous data is know as data prediction. This type of prediction is very useful during supervised learning. We used classification as well. This has helped us to classify between different crops. A classification model, a method of Supervised Learning, draws a conclusion from observed values as one or more outcomes in a categorical form. For example, email has filters like inbox, drafts, spam, etc. There is a number

of algorithms in the Classification model like Logistic Regression, Decision Tree, Random Forest, Multilayer Perception, etc. We used Random Forest algorithm for this project.

### 1.3 AIM AND OBJECTIVES

- To help the farmers to grow better crops.
- To develop a predictive model for the prediction of the crop
- To help the farmers not waste their time, fertilizers and money growing the crop which is not suitable for the given season.

# DATA SOURCE AND DATA QUALITY

## 2.1 DATASET USED

The data set we chose from Kaggle has the columns as:

- 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

This dataset was build by augmenting datasets of rainfall, climate and fertilizer data available for India. The dataset is suited for data mining.

```
crop=pd.read_csv('Crop_recommendation.csv')
crop.head()
```

| | 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 |

## 1.1 DATA PREPROCESSING

The data used in this project is made by augmenting and combining various publicly available datasets of India like weather, soil, etc. You can access the dataset here. This data is relatively simple with very few but useful features unlike the complicated features affecting the yield of the crop.

The data have Nitrogen, Phosphorous, Pottasium and pH values of the soil. Also, it also contains the humidity, temperature and rainfall required for a particular crop.

Hereis the data info:

```
n [5]: crop.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            2200 non-null   int64
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

For the successful application pre-processing is required. The data which is acquired from different resources are sometime in raw form. It may contain some incomplete, redundant, inconsistent data. Therefore in this step such redundant data should be filtered. Data should be normalized. But in our case here, we do not necessarily need data pre-processing as the data is clean enough.

Here is the data count:

```
In [3]: crop['label'].value_counts()
```

```
Out[3]: banana         100
        coffee         100
        coconut        100
        mango          100
        grapes         100
        lentil         100
        kidneybeans    100
        chickpea       100
        orange         100
        papaya         100
        rice           100
        jute           100
        muskmelon      100
        blackgram      100
        apple          100
        watermelon     100
        mothbeans      100
        maize          100
        pigeonpeas     100
        pomegranate    100
        mungbean       100
        cotton         100
        Name: label, dtype: int64
```

There are 2200 rows and 8 columns in our dataset.

```
crop.shape
```

```
(2200, 8)
```

# METHODS AND MODELS

## 1.1 DATA MINING QUESTIONS

- How does the data look like?

  The data looks like the following:

| | 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 |

- Are there any missing values in the dataset?

```
crop.isnull().sum()
```

```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

The data without preprocessing itself does not contain any null values. So we do not need to preprocess it.

### 3.2 DATA MINING ALGORITHMS

We could use differnet algorithms to choose the prediction of the corp to grow but for the sake of simplicity, we only chose only Descision Trees and Random Forest algorithm.

**1. Descision Tree Classification**:

A decision tree is a flowchart-like tree structure in which the internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.

Decision Tree consists of :

- Nodes : Test for the value of a certain attribute.
- Edges/ Branch : Correspond to the outcome of a test and connect to the next node or leaf.
- Leaf nodes : Terminal nodes that predict the outcome (represent class labels or class distribution).

There are two main types of Decision Trees:

- Classification Trees.
- Regression Trees.

We use the classification decision tree in our model to predict the data

In classification decision trees, the decision variable is categorical. The output will be either yes or no. On the basis of this we will be predicting the model.

Features of the Decision Tree

- It is simple to implement, and it follows a flow chart type structure that resembles human-like decision making.
- It proves to be very useful for decision-related problems.
- It helps to find all of the possible outcomes for a given problem.

## 2. Random Forest Algorithm:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. Random Forest is basically the combination of different decision trees. In this project we use the classification version of Random Forest Algorithm.

Important Features of Random Forest:

- Diversity- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- Immune to the curse of dimensionality- Since each tree does not consider all the features, the feature space is reduced.
- Parallelization-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

Difference Between Decision Tree & Random Forest

Random forest is a collection of decision trees; still, there are a lot of differences in their behavior.

| Decision trees | Random Forest |
|---|---|

| | |
|---|---|
| 1. Decision trees normally suffer from the problem of overfitting if it's allowed to grow without any control. | 1. Random forests are created from subsets of data and the final output is based on average or majority ranking and hence the problem of overfitting is taken care of. |
| 2. A single decision tree is faster in computation. | 2. It is comparatively slower. |
| 3. When a data set with features is taken as input by a decision tree it will formulate some set of rules to do prediction. | 3. Random forest randomly selects observations, builds a decision tree and the average result is taken. It doesn't use any set of formulas. |

Thus random forests are much more successful than decision trees only if the trees are diverse and acceptable.

### 3.3 DATA MINING MODELS

We use two data mining models in this project which are:

- Descision Trees Classificaiton
- Random Forest Classification

Classification:

Classification in machine learning and statistics is a supervised learning approach in which the computer program learns from the data given to it and make new observations or classifications. The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

# MODEL EVALUATION & DISCUSSION

This is the decision tree classification:

## Descision Tree Classification

In [9]:
```
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(x_train,y_train)

predicted_values = DecisionTree.predict(x_test)
x = metrics.accuracy_score(y_test, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(y_test,predicted_values))
```

```
DecisionTrees's Accuracy is:  90.0
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.59      1.00      0.74        16
    chickpea       1.00      1.00      1.00        21
     coconut       0.91      1.00      0.95        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.74      0.93      0.83        28
 kidneybeans       0.00      0.00      0.00        14
      lentil       0.68      1.00      0.81        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       0.00      0.00      0.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      0.84      0.91        19
  pigeonpeas       0.62      1.00      0.77        18
 pomegranate       1.00      1.00      1.00        17
        rice       1.00      0.62      0.77        16
  watermelon       1.00      1.00      1.00        15

    accuracy                           0.90       440
   macro avg       0.84      0.88      0.85       440
weighted avg       0.86      0.90      0.87       440
```

In [10]:
```
score = cross_val_score(DecisionTree, features, target,cv=5)
score
```

Out[10]:
```
array([0.93636364, 0.90909091, 0.91818182, 0.87045455, 0.93636364])
```

And this is the Random forest classification:

## Random Forest

In [11]:
```
from sklearn.ensemble import RandomForestClassifier

RandomForest = RandomForestClassifier(n_estimators=20, random_state=0)
RandomForest.fit(x_train,y_train)

predicted_values = RandomForest.predict(x_test)

x = metrics.accuracy_score(y_test, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(y_test,predicted_values))
```

```
RF's Accuracy is:  0.990909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.94      1.00      0.97        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.90      1.00      0.95        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      0.95      0.97        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      1.00      1.00        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       1.00      0.81      0.90        16
  watermelon       1.00      1.00      1.00        15

    accuracy                           0.99       440
   macro avg       0.99      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440
```

In [12]:
```
score = cross_val_score(RandomForest,features,target,cv=5)
score
```

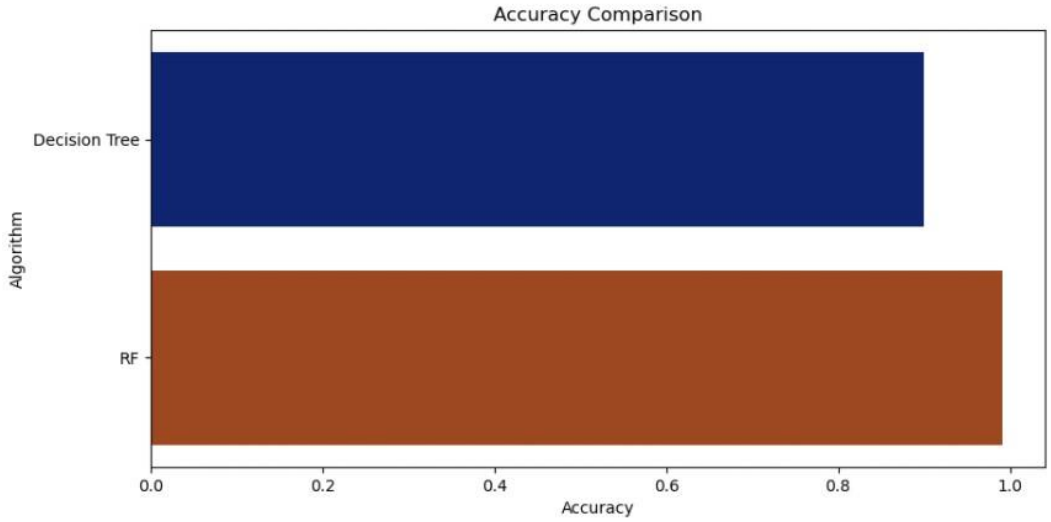
Out[12]:
```
array([0.99772727, 0.99545455, 0.99772727, 0.99318182, 0.98863636])
```

# CONCLUSION & FUTURE DIRECTION

## Accuracy test

In [13]:
```
plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
```

Out[13]: <AxesSubplot:title={'center':'Accuracy Comparison'}, xlabel='Accuracy', ylabel='Algorithm'>




In [14]:
```
accuracy_models = dict(zip(model, acc))
for k, v in accuracy_models.items():
    print (k, '-->', v)
```

```
Decision Tree --> 0.9
RF --> 0.990909090909091
```

Here we can see that the accuracy of Random Forest is greater than that of Decision tree.

## Making a Prediciton

```
In [15]: data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
         prediction = RandomForest.predict(data)
         print(prediction)

         ['coffee']
```

Here we make a simple prediction by inputing the value of the different prarameters. By doing so we get the answer ‘coffee’. This is the final conclusion of the project after we made the predition.

This project can be introduced to farmers when more prediction models are added to it. There is a future scope to this project in the sector of agriculture. The farmer can enter the values in the parameters and the required prediction can be made.

# REFLECTION PORTFOLIO

This project has helped us learn different concepts of data mining on our own. We went from absolutely zero knowledge to somewhat implementing the task we chose. This project has made us realize our drawbacks and incompetence in the sector of Data Mining. That has made use prepare for these tasks beforehand and made us self study more.

We learnt about teamwork and the importance of it. That is why we have the same team for other projects as well. This project has somewhat gave us confidence on what we could accomplish in a given time and we are really thankful for this opportunity.

# REFERENCES

https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-decision-tree-classificationusing-python/ https://www.edureka.co/blog/classification-in-machine-learning/

https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

And different research paper

# APPENDICES

Link to the dataset used:

https://www.kaggle.com/atharvaingle/crop-recommendation-dataset python

code implemented:

https://github.com/prajesh-sth/DataMining-LA2

Setup to execute the code:

Python 3.7.1 and up
Pandas
Scikit learn
Matplotlib
Seaborn