

AGRICULTURAL PRODUCT PRICE AND CROP CULTIVATION PREDICTION BASED ON SMLT

Submitted in partial fulfillment of the requirements for the

award of

Bachelor of Engineering degree in Computer Science and Engineering

by

Battula Varshitha(38110074)

Battula Varshini(38110618)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600

119

MARCH - 2022



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BATTULA VARSHINI(38110168),BATTULA VARSHITHA (38110074)** who carried out the project entitled "**AGRICULTURAL PRODUCT PRICE AND CROP CULTIVATION PREDICTION BASED ON SMLT**" under my supervision from Nov 2021 to March 2022

InternalGuide

Dr. A. VIJI AMUTHA MARY,M.Tech., Ph.D

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner External Examiner

DECLARATION

I, **BATTULA VARSHINI, BATTULA VARSHITHA** hereby declare that the project report entitled **AGRICULTURAL PRODUCT PRICE AND CROP CULTIVATION PREDICTION BASED ON SMLT** done by me under the guidance of **Dr. A. Viji Amutha Mary** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:11 Apr 2022

PLACE:Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Viji Amutha Mary,M.Tech.,Ph.D** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Agriculture is responsible for the majority of the nation's economic contribution across the world. However, due to a lack of ecosystem control methods, the majority of agricultural lands are still underdeveloped. Crop output is not improving as a result of these issues, which has an impact on the farm economy. As a result, a rise in agricultural output is aided by the forecast of plant yields. To avoid this issue, agricultural industries must use machine learning algorithms to estimate the harvest from a given dataset. The supervised machine learning method (SMLT) analyses a to capture a dataset several pieces of information, such as variable identification, Missing value treatments, univariate analysis, bivariate and multivariate analysis, and so on. A comparison of machine learning algorithms was performed to see which one was more accurate in forecasting the best crop. When entropy calculation, and are evaluated, precision, recall, F1 Score, sensitivity, specificity, & entropy come out ahead. the data show that the recommended machine learning algorithm technique has the highest accuracy.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	Abstract	i
	List of Tables	ii
	List of Figures	iii
1	Introduction	1
2	Literature Survey	2
3	Methodology	4
	3.1 Machine Learning	
	3.2 Preparing the Data set	
	3.3 List of Modules	
4	Result and Discussion	22
5	Conclusion and Future Work	23
	References	24
	Appendices	25
	A. Source Code	25-34
	B. Screen Shots	34-35
	C. Publication with Plagiarism Report	35

List of Tables

TABLE NO	TABLE NAME	PAGE NO
1	Data set Attributes	9

List of Figures

FIGURE NO	FIGURE NAME	PAGE NO
1	Process of Machine Learning	5
2	Block Diagram	7
3	Data set1	9
4	Data set2	10
5	Yield of Cost Production	11
6	Cost Production of Crop	11
7	Decision Tree	17
8	Random Forest	18
9	Naive Bayes	19
10	Support Vector	20

CHAPTER 1

INTRODUCTION

Agriculture research has boosted the world economy and offers enormous benefits to society as a whole. Crop evaluation in agriculture remains challenging, despite recent improvements that involve the use of a wide range of the availability of technical resources techniques, as well as techniques Precision farming and agri-technology sometimes referred to as virtual farming, are current research areas that boost the usage of data-intensive approaches agricultural productivity while lowering environmental impact. Accurate crop detection focuses on ecological and soil Agricultural production is dependent on a number of factors, one of which has been identified. the subject of decades of research.. The majority of existing algorithms for crop yield estimate involve machine learning (ML), but very little was done to forecast territory crops solely on soil and climate data as well as the environment Crop cultivation is influenced by a variety of factors, Soil texture, nutrients (N₂, P, and K), micronutrients (Fe, B, and Mn), temp, and rainfall are all factors to consider. Because the characteristics vary by zone, resulting in a large order to ensure the sustainability data set, it is necessary to pick key aspects that aid in the diagnosis of acceptable crops for certain land regions. Feature selection (FS) approaches are used to carry out the procedure.

Prediction relies heavily on machine learning methods. FS approaches [1]–[6] are utilized to reduce fitting problems and identify important characteristics for the prediction procedure, from the data set resulting in improved ML performance. Filters [7], wrap [8], and embed [9] are the three types of the FS approach. Filter techniques are unaffected by the classifier's performance, Wrapper approaches, on the other hand, pick features depending on the success of the classifier. Because it combines both filter and wrapper approaches, the embedded approach is analogous to the latter. This investigation focuses on wrapper FS approaches. To anticipate an appropriate crop and measure the efficacy of the FS process, the chosen attributes are fed into the k-nearest neighbor (kNN), Naïve Bayes (NB), decision tree (DT), support vector machine (SVM), and random forest (RF). The purpose of this research is just to improve the crops underlying model by selecting key data points.

CHAPTER 2

LITERATURE SURVEY

Title : Forecasting crop yield using remotely sensed vegetation indices and crop phenology metrics

Author : Douglas K. Bolton*, Mark A. Friedl

Year : 2013

We used data from NASA's Moderate Resolution Imaging Spectro radiometer (MODIS) to develop empirical models predicting maize and soybean yield in the Central United States. Results show that MODIS provides a better basis for predicting maize yields than the widely used NDVI.

Title : Crop Yield Assessment from Remote Sensing

Author : Paul C. Doraiswamy, Sophie Moulin, Paul W. Cook, and Alan Stern

Satellite remotely sensed data provide a real-time assessment of the magnitude and variation of crop condition parameters. This study investigates the use of these parameters as an input to a crop growth model. A radiative transfer model, SAIL, provided the link between the satellite data and crop model.

Title : A generalized regression-based model for forecasting winter wheat yields in Kansas and Ukraine using MODIS data

Author : Becker-Reshef a, * , E. Vermote a , M. Lindeman b , C. Justice

Year : 2010

Wheat is one of the key cereal crops grown worldwide. The forecasts of production in Kansas closely matched the USDA/NASS reported numbers with a 7% error. The same regression model forecast winter wheat production in Ukraine within 10% of the official reported numbers six weeks prior to harvest.

Title : Plant Yield Prediction Model Using Firefly based Feature Selection with Modified Fuzzy Cognitive Maps

Author : 1 D. Sabareeswaran and 2 R. Gunasundari

In this paper, a development of agricultural productivity is enhanced based on the plant yield prediction. Predicted outcome is transmitted to the farmer's through smart phones which helps for identifying the growth of plant and improving the harvesting. The experimental results show that the effectiveness of the proposed technique can be compared with the other prediction techniques.

Title : Crop yield forecasting on the Canadian Prairies using MODIS NDVI data

Author : M.S. Mkhabela^{a,*}, P. Bullock^a, S. Raj^b, S. Wang^c, Y. Yang

Year : 2010

Moderate resolution imaging spectroradiometer (MODIS) could be used for crop yield prediction on the Canadian Prairies. MODIS 10-day composite NDVI data for the years 2000–2006 were obtained from the Canada Centre for Remote Sensing. New regression models were developed to forecast crops grain yield. Results showed that MODIS-NDVI data can be used effectively to predict crop yield on the Canadian Prairies. Power function models developed for each crop accounted for 48 to 90% of the grain yield variability for barley, canola, field peas and spring wheat.

CHAPTER 3

METHODOLOGY

3.1 MACHINE LEARNING

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-

class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.



Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is $y = f(X)$. The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression, multi-class classification, Decision Trees and support vector machines etc. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into Classification problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

3.2 PREPARING THE DATASET

The demo dataset is now supplied to machine learning model on the basis of this data set the model is trained. Every new detail filled at the time of application form acts as a test data set. After the operation of testing, model prediction based upon the inference it concludes on the basis of the training data sets. Satellite Imagery (Remote Sensing Data), has been widely used for predicting crop yield. This dataset is collected using the sensors mounted on satellites or planes, which

detect the energy (electromagnetic waves), reflected or diffracted from surface of the earth. Remote sensing data has a lot of energy bands to offer, but mainly only few of them have been used for crop yield prediction. Yet, there are some people who have tried generating relevant features using the bands which are typically ignored, and they have been successful with improving results with that. In case of this dataset, most people rarely explore the high-order moments of the features. Based on these datasets people have used algorithms like Regression models, Random Forest and Nearest Neighbor etc.

Table shows details of the datasets:

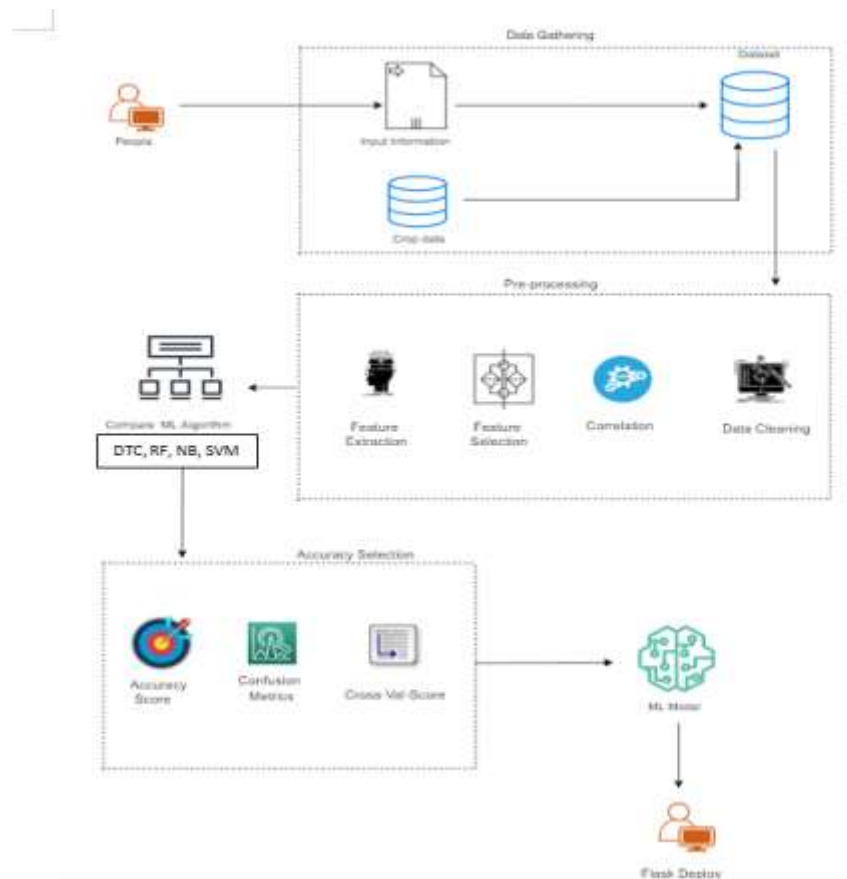
Variable	Description
Crop	Crop name
State Name	Indian state name
District Name	District name list of each state
Cost of Cultivation (`/Hectare) C2	Cultivation amount for C2 Scheme
Cost of Production (`/Quintal) C2	Production amount for A2+FL Scheme
Yield (Quintal/ Hectare)	Yield of crop
Crop year	Crop year list
District Name	District name for each state
Area	Total area of each place
Rainfall	Water availability of each crop
Average humidity	directly influences the water relations of plant and indirectly affects leaf growth
Mean Temperature	Climate of each crop
Cost Production of per yield crop	Cost of crop yield

Dataset Attributes

3.3 List of Modules:

- Data Pre-processing
- Data Analysis of Visualization
- Comparing Algorithm with prediction in the form of best accuracy result
- Deployment Using GUI

Architecture Diagram



Block Diagram

3.3.1 Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers

use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset

- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

3.3.2 Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

	State_Name	District_Name	Crop_Year	Season	Crop	Area	rainfall	Average Humidity	Mean Temp	Cost of Cultivation (Hectare) C2	Cost of Production (Quintal) C2	Yield (Quintal/ Hectare)	cost of production per yield
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	0.012360	57	62	23076.74	1941.56	9.83	19085.4365
1	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Arecanut	1254.0	0.084119	56	58	12610.85	1691.66	6.83	11554.0378
2	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Arecanut	1258.0	0.080064	58	53	32683.46	3207.35	9.33	29924.5755
3	Andaman and Nicobar Islands	NICOBARS	2003	Whole Year	Arecanut	1261.0	0.181051	57	58	13209.32	2228.97	5.90	13150.9230
4	Andaman and Nicobar Islands	NICOBARS	2004	Whole Year	Arecanut	1264.7	0.035446	63	67	22560.30	1595.56	13.57	21651.7492

Data set1

	Crop_Year	Area	rainfall	Average Humidity	Mean Temp	Cost of Cultivation (/Hectare) C2	Cost of Production (/Quintal) C2	Yield (Quintal/ Hectare)	cost of production per yield
Crop_Year	1.000000	-0.033998	0.001955	-0.012323	0.053708	-0.008896	0.004705	0.017974	0.006590
Area	-0.033998	1.000000	-0.020967	0.007052	0.002213	0.001313	0.004999	-0.004052	-0.001414
rainfall	0.001955	-0.020967	1.000000	-0.009805	-0.311162	-0.030486	-0.014171	0.069267	-0.046942
Average Humidity	-0.012323	0.007052	-0.009805	1.000000	-0.252588	-0.009753	0.030869	-0.131063	-0.090237
Mean Temp	0.053708	0.002213	-0.311162	-0.252588	1.000000	0.098394	-0.041967	0.057664	0.015582
Cost of Cultivation (/Hectare) C2	-0.008896	0.001313	-0.030486	-0.009753	0.098394	1.000000	0.025679	0.028143	-0.055532
Cost of Production (/Quintal) C2	0.004705	0.004999	-0.014171	0.030869	-0.041967	0.025679	1.000000	-0.308414	-0.056326
Yield (Quintal/ Hectare)	0.017974	-0.004052	0.069267	-0.131063	0.057664	0.028143	-0.308414	1.000000	0.743026
cost of production per yield	0.006590	-0.001414	-0.046942	-0.090237	0.015582	-0.055532	-0.056326	0.743026	1.000000

Data set2

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : removing noisy data

Exploration data analysis of visualization

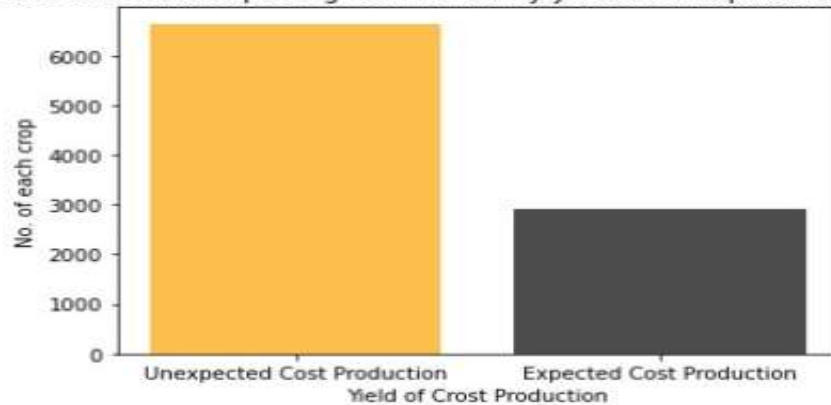
Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields

themselves and it will recommend a deeper dive into some of the books mentioned at the end.

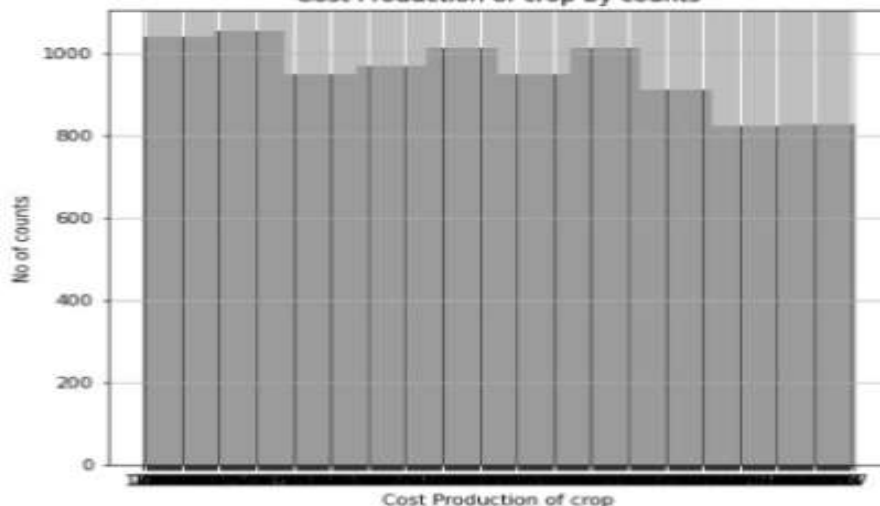
Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.

Prediction results expecting from farmer by yield of crop production amount



Cost Production of crop by counts



MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : visualized data

3.3.3 Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

- Random Forest
- Decision Tree Classifier
- Naive Bayes
- Support Vector Machine

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value

indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

$$\text{True Positive Rate(TPR)} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{False Positive Rate(FPR)} = \text{FP} / (\text{FP} + \text{TN})$$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

$$F\text{- Measure} = 2TP / (2TP + FP + FN)$$

F1-Score Formula:

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

ALGORITHM AND TECHNIQUES

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.

- Here, we are using some of its modules like `train_test_split`, `DecisionTreeClassifier` or `Logistic Regression` and `accuracy_score`.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

Decision Tree Classifier:

One of the most effective and well-known algorithms is the decision-tree algorithm. It may be used with both continuous and categorical output variables.

Decision tree

We consider the full training set to be the root at first. Attributes are designed to be categorical for information gathering, yet they are regarded continuous. Recursively, records are dispersed based on attribute values. For arranging characteristics as root or internal node, we apply statistical approaches.

In the shape of a tree structure, a decision tree creates classification or regression models. While developing a decision tree, it separates a set of data into shorter and shorter subgroups in the background. The greatest predictor is the root node, which is technically the topmost decision node in a tree.

Classification report of Decision Tree Classifier Results:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1776
1	1.00	1.00	1.00	1087
accuracy			1.00	2863
macro avg	1.00	1.00	1.00	2863
weighted avg	1.00	1.00	1.00	2863

Accuracy result of Decision Tree Classifier is 100.0

Confusion Matrix result of Decision Tree Classifier is:

```
[[1776  0]
 [  0 1087]]
```

Sensitivity : 1.0

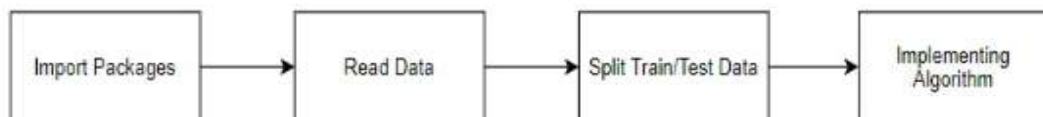
Specificity : 1.0

Cross validation test results of accuracy:

```
[1. 1. 1. 1. 1.]
```

Decision Tree

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

Random Forest Classifier

Random forests, also known as random choice forests, are a type of ensemble learning approach used to solve problems like classification and regression. Ensemble methods is a type of machine learning in which many different versions of same algorithms are combined to create a more accurate model model. The random forest approach may be used in both classification and regression applications.

The following is how the random forest approach is implemented:

At random, choose N records from of the data collection.

Make a decision tree using these N records.

Rep steps 1 & 2 as needed for the number of trees in your method.

A forest's trees each forecast an X (input) or Y (output) value (value). The ultimate result is obtained by adding all of the predicted from all the trees. Every tree in the forests predicts which category that additional data will fall into when it comes to categorisation challenges.

```
Classification report of Random Forest Results:

              precision    recall  f1-score   support

     0       1.00      1.00      1.00     1776
     1       1.00      1.00      1.00     1087

 accuracy          1.00
 macro avg          1.00
weighted avg          1.00

Accuracy result of Random Forest is: 100.0

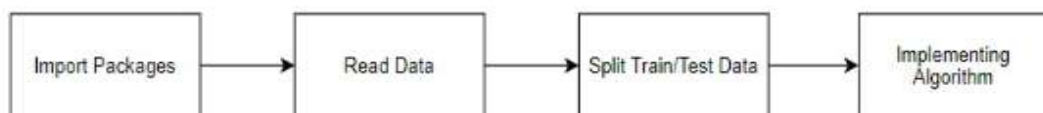
Confusion Matrix result of Random Forest is:
[[1776   0]
 [   0 1087]]

Sensitivity : 1.0
Specificity : 1.0

Cross validation test results of accuracy:
[1. 1. 1. 1. 1.]
```

Random Forest

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

Naive Bayes algorithm

Modeling complicated data with the Naive Bayes method is a quick and efficient process. It produces a prediction based on the likelihood that each characteristic belongs to each class.

The Bayes Theorem is used to build the Naive Bayes statistical classification technique. It's one of the most fundamental supervised learning algorithms, as well as one of the fastest, most accurate, and trustworthy. To create a forecast, we may compute the probabilities of each class's occurrence and choose the class value with the highest likelihood.

This is due to the assumption that each attribute in a class is independent of other characteristics such as wealth, age, and location.

```

Classification report of Naive Bayes Results:

              precision    recall  f1-score   support

     0       0.95         1.00         0.97         1776
     1       1.00         0.91         0.95         1087

 accuracy          0.97         0.95         0.96         2863
 macro avg          0.97         0.95         0.96         2863
 weighted avg          0.97         0.97         0.96         2863

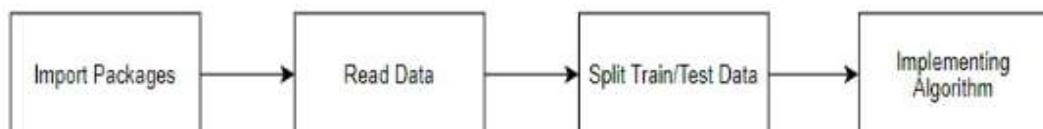
Accuracy result of Naive Bayes is: 96.50716032134125

Confusion Matrix result of Naive Bayes is:
[[1776   0]
 [ 100 987]]

Sensitivity : 1.0
Specificity : 0.9080036798528058
  
```

Naive Bayes

MODULEDIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

Support Vector Machines

Vector Machines one among the most popular well-known and discussed machine learning techniques. They were immensely popular in the 1990s, when they were first introduced. A classifier that categories a data collection by determining the

best hyperplane between the data points. With a little tweaking, this model can produce a high prediction rate.

This article discusses support vector machines (SVM) and how to properly prepare your data for the SVM method. We also look at how a trained SVM model representation may be used to predict what would happen in real life.

```
Classification report of Support Vector Machines Results:

              precision    recall  f1-score   support

     0       0.88        0.90        0.89        1776
     1       0.82        0.80        0.81        1087

 accuracy          0.86        2863
 macro avg         0.85        2863
 weighted avg      0.86        2863

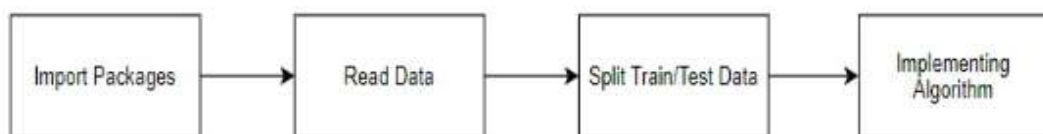
Accuracy result of Support Vector Machines is: 85.78414250785889

Confusion Matrix result of Support Vector Machines is:
[[1591  185]
 [ 222  865]]

Sensitivity :  0.8958333333333334
Specificity :  0.795768169273229
```

Support Vector

MODULEDIAGRAM



3.3.4 Deployment GUI

Tkinter tutorial provides basic and advanced concepts of Python Tkinter. Our Tkinter tutorial is designed for beginners and professionals. Python provides the standard library Tkinter for creating the graphical user interface for desktop based applications.

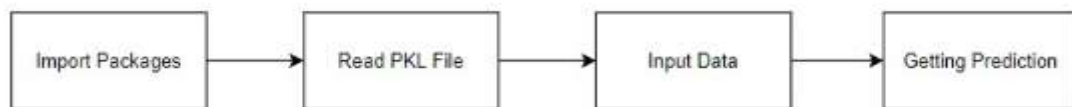
Developing desktop based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps. Tkinter is a python library for developing GUI (Graphical User Interfaces).

We use the tkinter interface and Tkinter will come with Python as a standard package, it can be used for security purpose of each users or accountants. There will be two kinds of pages like registration user purpose and login entry purpose of

library for creating an application of UI (User Interface), to create windows and all other graphical user users.

1. import the Tkinter module.
2. Create the main application window.
3. Add the widgets like labels, buttons, frames, etc. to the window.
4. Call the main event loop so that the actions can take place on the user's computer screen.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data values

output : predicting output

CHAPTER 4

RESULT AND DISCUSSION

When it comes to choosing, critical variables for the purpose of agricultural forecasting, this section evaluates the outcomes of the developed approach to several wrapper FS techniques. These researches studied the effectiveness of the approach by using supervised learning methods like the k - means, Naïve bayes, DT, SVM, and RF to select the optimal crop for cultivation.

The approach is then shown to be as applicable to agricultural sets of data as it would be to others. When compared to other strategies, finds the best relevant features with enhanced prediction Accuracy in the shortest amount of time due to the use of permutations and ranking.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Conclusion

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. Finally we predict the crop using machine learning algorithm with different results. This brings some of the following insights about crop prediction. As maximum types of crops will be covered under this system, farmer may get to know about the crop which may never have been cultivated and lists out all possible crops, it helps the farmer in decision making of which crop to cultivate. Also, this system takes into consideration the past production of data which will help the farmer get insight into the demand and the cost of various crops in market.

Future Work

- Remaining SMLT algorithms will be involve to finding the best accuracy with applying to predict the crop yield and cost.
- Agricultural department wants to automate the detecting the yield crops from eligibility process (real time).
- To automate this process by show the prediction result in web application or desktop application.
- To optimize the work to implement in Artificial Intelligence environment.

REFERENCES

- [1] A. Mark Hall, "Feature selection for discrete and numeric class machine learning," *Comput. Sci., Univ. Waikato*, pp. 359–366, Dec. 1999.
- [2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
- [3] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection—theory and algorithms," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 43.
- [4] P. S. Maya Gopal and R. Bhargavi, "Feature selection for yield prediction in boruta algorithm," *Int. J. Pure Appl. Math.*, vol. 118, no. 22, pp. 139–144, 2018.
- [5] S. Ji, S. Pan, X. Li, E. Cambria, G. Long, and Z. Huang, "Suicidal ideation detection: A review of machine learning methods and applications," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 1, pp. 214–226, Feb. 2021.
- [6] K. Ranjini, A. Suruliandi, and S. P. Raja, "An ensemble of heterogeneous incremental classifiers for assisted reproductive technology outcome prediction," *IEEE Trans. Comput. Social Syst.* early access, Nov. 3, 2020, doi: 10.1109/TCSS.2020.3032640.
- [7] H. Liu and R. Setiono, "A probabilistic approach to feature selection—a filter solution," in *Proc. 13th Int. Conf. Int. Conf. Mach. Learn.*, vol. 96, 1996, pp. 319–327.
- [8] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.
- [9] H. Wang, M. Taghi Khoshgoftaar, and K. Gao, "Ensemble feature selection technique for software quality classification," in *Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng.*, 2010, pp. 215–220.
- [10] D. H. Zala and M. B. Chaudhri, "Review on use of BAGGING technique in agriculture crop yield prediction," *Int. J. Sci. Res. Develop.*, vol. 6, no. 8, pp. 675–677, 2018.

APPENDICES

A. Source Code

```
import numpy as np

import pandas as pd

In [ ]:df = pd.read_csv("re.csv")

In [ ]:df

In [ ]:df.shape

In [ ]:df.columns

In [ ]:l1=['Above_1_ton_per_hect', 'Below_1_ton_per_hect', 'Above_3_tons_per_hect',
'Above_100_tons_per_hect', 'Above_70_tons_per_hect', 'Above_40_tons_per_hect',
'Above_6_tons_per_hect', 'Above_50_tons_per_hect', 'Above_10_tons_per_hect']

In [ ]:l2=['Below_20000_hect', 'Above_20000_hect', 'Above_40000_hect',
'Above_80000_hect']

In [ ]:l3=['Andaman and Nicobar Islands', 'Andhra Pradesh', 'Arunachal
Pradesh', 'Assam', 'Bihar', 'Chandigarh', 'Chhattisgarh', 'Dadra and Nagar Haveli',
'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir', 'Jharkhand',
'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Puducherry', 'Punjab',
'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar
Pradesh', 'Uttarakhand']

In [ ]:l4=['Kharif', 'Whole year', 'Rabi', 'Autumn', 'Summer', 'Winter']

In [ ]:l5=['TA50', 'TA60', 'TA80']

In [ ]:l6=['HB25', 'HA25', 'HA60', 'HA80']

In [ ]:df['Crop'].unique()

In [ ]:croptypes=['Sugercane', 'Rice', 'Tobacco', 'Wheat', 'Coconut',
'Yam', 'Sweetpotato', 'Turmeric', 'WaterMelon', 'Tapioca', 'Urad', 'Varagu']

In [ ]:l7=['Above_1_ton_per_hect', 'Below_1_ton_per_hect', 'Above_3_tons_per_hect',
'Above_100_tons_per_hect', 'Above_70_tons_per_hect', 'Above_40_tons_per_hect',
'Above_6_tons_per_hect', 'Above_50_tons_per_hect', 'Above_10_tons_per_hect',
'Below_20000_hect', 'Above_20000_hect', 'Above_40000_hect',
'Above_80000_hect', 'Andaman and Nicobar Islands', 'Andhra Pradesh', 'Arunachal
Pradesh', 'Assam', 'Bihar', 'Chandigarh', 'Chhattisgarh', 'Dadra and Nagar Haveli',
'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir', 'Jharkhand',
'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Puducherry', 'Punjab',
'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar
```

```
Pradesh','Uttarakhand','Kharif','Whole year','Rabi','Autumn','Summer','Winter',  
'TA50','TA60','TA80','HB25','HA25','HA60','HA80']
```

```
In [ ]:l8=[]
```

```
for x in range(0,len(l7)):
```

```
l8.append(0)
```

```
In [ ]:df.replace({'Crop':{'Sugercane':0, 'Rice':1, 'Tobacco':2, 'Wheat':3,  
'Coconut':4, 'Yam':5, 'Sweetpotato':6, 'Turmeric':7, 'WaterMelon':8, 'Tapioca':9,  
'Urad':10,'Varagu':11}},inplace=True)
```

```
In [ ]:X_test= tr[l7]
```

```
y_test = tr[["Crop"]]
```

```
np.ravel(y_test)
```

```
In [ ]:def randomforest():
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
clf4 = RandomForestClassifier()
```

```
clf4 = clf4.fit(X,np.ravel(y))
```

```
# calculating accuracy-----
```

```
from sklearn.metrics import accuracy_score
```

```
y_pred=clf4.predict(X_test)
```

```
print(accuracy_score(y_test, y_pred))
```

```
print(accuracy_score(y_test, y_pred,normalize=False))
```

```
# -----
```

```
terms = [State.get(),Season.get(),Tl.get(),Hl.get(),Yl.get()]
```

```
for k in range(0,len(l7)):
```

```
for z in terms:
```

```
if(z==l7[k]):
```

```
l8[k]=1
```

```
inputtest = [l8]
```

```
predict = clf4.predict(inputtest)
```

```
predicted=predict[0]
```

```
h='no'
```

```
for a in range(0,len(croptc)):
```

```

if(predicted == a):
    h='yes'
    break
if (h=='yes'):
    t2.delete("1.0", END)
    t2.insert(END, cropec[a])
else:
    t2.delete("1.0", END)
    t2.insert(END, "Not Found")
In [ ]:root = Tk()
root.title("Indian Crop Prediction")
#root.configure(background='black')
In [ ]:canvas = Canvas(root,width=1880,height=1253)
canvas.pack()
photo = PhotoImage(file ='im3.png')
canvas.create_image(0,0,image=photo,anchor=NW)
In [ ]:# gui_stuff-----
# entry variables
State = StringVar()
State.set(None)
Season = StringVar()
Season.set(None)
Tl = StringVar()
Tl.set(None)
fn= StringVar()
vn= StringVar()
dn=StringVar()
rl= IntVar()
In [ ]:# Heading

```

```
w2 = Label(canvas, justify=LEFT, text="Crop Prediction using Machine Learning",  
fg="white", bg="red")
```

```
w2.config(font=("Elephant", 20))
```

```
w2.grid(row=1, column=0, columnspan=2, padx=100)
```

```
w2 = Label(canvas, justify=LEFT, text="Indian Agricultural Department",  
fg="green")
```

```
w2.config(font=("Times Roman", 15))
```

```
w2.grid(row=2, column=0, columnspan=2, padx=100)
```

```
In [ ]:# labels
```

```
FNLb = Label(canvas, text="Farmer Name",fg="yellow", bg="black")
```

```
FNLb.grid(row=5, column=0, pady=15, sticky=W)
```

```
VNLb = Label(canvas, text="Village Name",fg="yellow", bg="black")
```

```
VNLb.grid(row=6, column=0, pady=15, sticky=W)
```

```
DNLb = Label(canvas, text="District Name",fg="yellow", bg="black")
```

```
DNLb.grid(row=7, column=0, pady=15, sticky=W)
```

```
In [ ]:# labels
```

```
stLb = Label(canvas, text="State Name",fg="yellow", bg="black")
```

```
stLb.grid(row=8, column=0, pady=15, sticky=W)
```

```
seLb = Label(canvas, text="Season Details",fg="yellow", bg="black")
```

```
seLb.grid(row=9, column=0, pady=15, sticky=W)
```

```
tlLb = Label(canvas, text="Temperature Level",fg="yellow", bg="black")
```

```
tlLb.grid(row=5, column=1, pady=15, sticky=W)
```

```
hlLb = Label(canvas, text="Humidity Level",fg="yellow", bg="black")
```

```
hlLb.grid(row=6, column=1, pady=15, sticky=W)
```

```
ylLb = Label(canvas, text="Yield level (per Hectare)",fg="yellow", bg="black")
```

```
ylLb.grid(row=7, column=1, pady=15, sticky=W)
```

```
ycLb = Label(canvas, text="Yield cost amount (per Hectare)",fg="yellow",  
bg="black")
```

```
ycLb.grid(row=8, column=1, pady=15, sticky=W)
```

```
In [ ]:ranfLb = Label(canvas, text="RF", fg="white", bg="red")
```

```

ranfLb.grid(row=15, column=0, pady=10, sticky=W)

In [ ]:# entries
OPTIONS1 = sorted(l1)
OPTIONS2 = sorted(l2)
OPTIONS3 = sorted(l3)
OPTIONS4 = sorted(l4)
OPTIONS5 = sorted(l5)
OPTIONS6 = sorted(l6)

In [ ]:FNEn = Entry(canvas, textvariable=fn)
FNEn.grid(row=5, column=0)
VNEEn = Entry(canvas, textvariable=vn)
VNEEn.grid(row=6, column=0)
DNEEn = Entry(canvas, textvariable=dn)
DNEEn.grid(row=7, column=0)
stEn = OptionMenu(canvas, State,*OPTIONS3)
stEn.grid(row=8, column=0)
seEn = OptionMenu(canvas, Season,*OPTIONS4)
seEn.grid(row=9, column=0)
tlEn = OptionMenu(canvas, Tl,*OPTIONS5)
tlEn.grid(row=5, column=1)
hlEn = OptionMenu(canvas, Hl,*OPTIONS6)
hlEn.grid(row=6, column=1)
ylEn = OptionMenu(canvas, Yl,*OPTIONS1)
ylEn.grid(row=7, column=1)
ycEn = OptionMenu(canvas, Yc,*OPTIONS2)
ycEn.grid(row=8, column=1)

In [ ]:rnf = Button(canvas, text="Check the result",
command=randomforest,bg="cyan",fg="green")
rnf.grid(row=15, column=1,padx=10)

```

```

In [ ]:#textfileds

t2 = Text(canvas, height=1, width=40,bg="orange",fg="black")

t2.grid(row=15, column=0 , padx=10)

import numpy as np

import pandas as pd

In [ ]:df = pd.read_csv("re1.csv")

In [ ]:df

In [ ]:df.shape

In [ ]:df.columns

In [ ]:l1=['Sugercane', 'Rice', 'Tobacco', 'Wheat', 'Coconut', 'Yam', 'Sweetpotato',
'Tapioca', 'Turmeric', 'WaterMelon', 'Urad', 'Varagu', 'Banana']

In [ ]: l4=['Summer', 'Winter', 'Kharif', 'Rabi',
l5=['TA50', 'TA60', 'TA80']

In [ ]:l6=['HB25', 'HA25', 'HA60', 'HA80']

In [ ]:df['Class'].unique()

In [ ]:cropryc=['Cost Yield is ten tons and Cost Production is ten lakshs',
'Cost Yield is one ton and Cost Production is twenty thousand',
'Cost Yield is twenty tons and Cost Production is forty lakshs',
'Cost Yield is two tons and Cost Production is forty thousand',
'Cost Yield is three tons and Cost Production is sixty thousand',
'Cost Yield is six tons and Cost Production is six lakshs ',
'Cost Yield is two tons and Cost Production is forty thousand',
'Cost Yield is five tons and Cost Production is one lakshs',
'Cost Yield is ten tons and Cost Production is two lakshs']

In [ ]:l7=['Sugercane', 'Rice', 'Tobacco', 'Wheat', 'Coconut', 'Yam', 'Sweetpotato',
'Tapioca', 'Turmeric', 'WaterMelon', 'Urad', 'Varagu', 'Banana', 'THIRUVALLUR',
'TUTICORIN', 'Summer', 'Winter', 'Kharif', 'Rabi', 'Whole Year', 'Autumn', 'TA50',
'TA60', 'TA80', 'HB25', 'HA25', 'HA60', 'HA80']

In [ ]:l8=[]

for x in range(0,len(l7)):

l8.append(0)

```

```

Production is fourty thousand':3,
'Cost Yield is three tons and Cost Production is sixty thousand':4,
'Cost Yield is six tons and Cost Production is six lakshs ':5,
'Cost Yield is two tons and Cost Production is fourty thousand':6,
'Cost Yield is five tons and Cost Production is one lakshs':7,
'Cost Yield is ten tons and Cost Production is two lakshs':8}},inplace=True)

In [ ]:X= df[l7]
y = df[["Class"]]
np.ravel(y)

# calculating accuracy-----

from sklearn.metrics import accuracy_score
y_pred=clf4.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred,normalize=False))

# -----

terms = [Dist.get(),Season.get(),Tl.get(),Hl.get(),Crop.get()]
for k in range(0,len(l7)):
    for z
        h='no'
        for a in range(0,len(croptc)):
            if(predicted == a):
                h='yes'
                break
            if (h=='yes'):
                t2.delete("1.0", END)
                t2.insert(END, croptc[a])
            else:
                t2.delete("1.0", END)
                t2.insert(END, "Not Found")

```

```

In [ ]:root = Tk()
root.configure(background='black')

In [ ]:# gui_stuff-----

# entry variables
Dist = StringVar()
Dist.set(None)
Season = StringVar()
Season.set(None)
Crop =StringVar()
Crop.set(None)
fn= StringVar()
vn= StringVar()
rl= IntVar()

In [ ]:# Heading
w2 = Label(root, justify=LEFT, text="Prediction of Yield and Yield cost by Crop
using Machine Learning", fg="white", bg="red")
w2.config(font=("Elephant", 20))
w2.grid(row=1, column=0, columnspan=2, padx=100)

w2 = Label(root, justify=LEFT, text="Tamilnadu Agricultural Department",
fg="green")
w2.config(font=("Times Roman", 15))
w2.grid(row=2, column=0, columnspan=2, padx=100)

In [ ]:# labels
FNLb = Label(root, text="Farmer Name",fg="yellow", bg="black")
FNLb.grid(row=5, column=0, pady=15, sticky=W)
VNLb = Label(root, text="Village Name",fg="yellow", bg="black")
VNLb.grid(row=6, column=0, pady=15, sticky=W)
RFLb = Label(root, text="Rainfall Value",fg="yellow", bg="black")
RFLb.grid(row=8, column=1, pady=15, sticky=W)

In [ ]:# labels

```

```

dtLb = Label(root, text="District Name",fg="yellow", bg="black")
dtLb.grid(row=8, column=0, pady=15, sticky=W)
seLb = Label(root, text="Season Details",fg="yellow", bg="black")
seLb.grid(row=9, column=0, pady=15, sticky=W)
tlLb = Label(root, text="Temperature Level",fg="yellow", bg="black")
tlLb.grid(row=5, column=1, pady=15, sticky=W)
hlLb = Label(root, text="Humidity Level",fg="yellow", bg="black")
hlLb.grid(row=6, column=1, pady=15, sticky=W)
crLb = Label(root, text="Crop Details",fg="yellow", bg="black")
crLb.grid(row=7, column=1, pady=15, sticky=W)
In [ ]:ranfLb = Label(root, text="RandomForest", fg="white", bg="red")
ranfLb.grid(row=15, column=0, pady=10, sticky=W)
In [ ]:# entries
OPTIONS1 = sorted(l1)
OPTIONS2 = sorted(l2)
OPTIONS4 = sorted(l4)
OPTIONS5 = sorted(l5)
OPTIONS6 = sorted(l6)
In [ ]:FNEn = Entry(root, textvariable=fn)
FNEn.grid(row=5, column=0)
VNEn = Entry(root, textvariable=vn)
VNEn.grid(row=6, column=0)
RFEn = Entry(root, textvariable=rl)
RFEn.grid(row=8, column=1)
crEn = OptionMenu(root, Crop,*OPTIONS1)
crEn.grid(row=7, column=1)
dtEn = OptionMenu(root, Dist,*OPTIONS2)
dtEn.grid(row=8, column=0)
seEn = OptionMenu(root, Season,*OPTIONS4)

```

```

seEn.grid(row=9, column=0)
tlEn = OptionMenu(root, TI,*OPTIONS5)
tlEn.grid(row=5, column=1)
hlEn = OptionMenu(root, HI,*OPTIONS6)
hlEn.grid(row=6, column=1)
In [ ]:rnf=Button(root,text="Check the result",command=randomforest,bg="cyan",
fg"green")
rnf.grid(row=15, column=1,padx=10)
In [ ]:#textfileds
t2 = Text(root, height=1, width=70,bg="orange",fg="black")
t2.grid(row=15, column=0 , padx=10)

```

B. SCREEN SHOTS

Crop Prediction using Machine Learning

Indian Agricultural Department

Farmer Name		Temperature Level	None ▾
Village Name		Humidity Level	None ▾
District Name		Yield level (per Hect)	None ▾
State Name	None ▾	Yield cost amount (	None ▾
Season Details	None ▾		

RF Check the result

Crop Prediction using Machine Learning

Indian Agricultural Department

Farmer Name	veera	Temperature Level	TA50
Village Name	veerapanenigudem	Humidity Level	HA60
District Name	Krishna	Yield level	Above_50_tons_perhec
State Name	Andhra Pradesh	Yield cost and	Above_80000hec
Season Details	Winter		

RF
Rice
Check the result

Prediction of Yield and Yield cost by Crop using Machine Learning

Tamilnadu Agricultural Department

Farmer Name		Temperature Level	None
Village Name		Humidity Level	None
District Name	None	Crop Details	None
Season Details	None	Rainfall Value	0

Prediction of Yield and Yield cost by Crop using Machine Learning

Tamilnadu Agricultural Department

Farmer Name	Veera	Temperature Level	TA60
Village Name	narayapuram	Humidity Level	HA60
District Name	KANICHIPURAM	Crop Details	Banana
Season Details	Whole Year	Rainfall Value	15

C. PUBLICATION WITH PLAGIARISM REPORT

[Report.pdf](#)