

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By
Madhumitha Chandrasekar

37120043



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU

AUGUST 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of Madhumitha Chandrasekar (Reg.no.37120043) who carried out the project entitled "CROP RECOMMENDATION SYSTEM" under our supervision from November 2019 to April 2020.

Internal Guide

Dr. Mary Gladence.L M.E., Ph.D.,

Head of the Department

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I Madhumitha Chandrasekar (Reg. No. 37120043) hereby declare that the Project Report entitled "**CROP RECOMMENDATION SYSTEM**" done by me under the guidance of Dr. Mary Gladence.L M.E., Ph.D., at Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 31/08/2020

Place: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** 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. R. SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology 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. Mary Gladence. L M.E., Ph.D.**, for her 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 Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



06/27/2020

Madhumitha Chandrasekar

has successfully completed

AWS Fundamentals: Building Serverless Applications

an online non-credit course authorized by Amazon Web Services and offered through Coursera

A row of five handwritten signatures in black ink, representing the instructors of the course.

Rick Hurst, Jonathan Dion, Adam Becker, Bobbie Atristain, Wes Gruver
AWS Training and Certification

**COURSE
CERTIFICATE**



Verify at coursera.org/verify/3V8FC7AHP5PT
Coursera has confirmed the identity of this individual and
their participation in the course.



RJP INFOTEK PVT. LTD.

This is to certify that

MADHUMITHA CHANDRASEKAR

has successfully completed

VMWARE Authorized VCA - Datacenter Virtualization Training

@ SATHYABAMA INSTITUTE OF SCIENCE & TECHNOLOGY



This certificate earned on

24/10/2019

Course Co-ordinator

This certificate can be verified at RJP INFOTEK, Ph. 2473 1102

ABSTRACT

CROP RECOMMENDATION SYSTEM

Farmers hold the backbone of the agricultural system. As it is common knowledge that for a country to progress the gross domestic product should be reasonable, agriculture is one of the important parts. The agricultural system will only run if there are crops being grown and harvested, so this is where farmers come in. So, to help these farmers, I have decided to create a prototype for a real-time crop recommendation algorithm in Python using Machine Learning and Data Analytics. This work presents a system, in the form of a website. The business logic in Python uses Machine Learning techniques in order to predict the most profitable crop in the forecasted weather and soil conditions at a specified location. The proposed system will integrate the data obtained from soil, crop repository, weather department and by applying machine learning algorithm: Multiple Linear Regression, a prediction of most suitable crops according to current environmental conditions is made. This provides a farmer with a variety of options of crops that can be cultivated.

CHAPTER NO	TITLE NAME	PAGE NO
	ABSTRACT	7
	LIST OF FIGURES	9
	LIST OF ABBREVIATION	10
1	INTRODUCTION	11
	1.1 GENERAL	11
	1.2 OUTLINE OF THE PROJECT	11
2	AIM AND SCOPE	12
	2.1 AIM	12
	2.2 PROBLEM STATEMENT	12
	2.3 SCOPE	12
3	SYSTEM ANALYSIS AND DESIGN	13
	3.1 GENERAL	13
	3.2 SOFTWARES USED	14
	3.3 ALGORITHM	17
	3.4 PROJECT DESCRIPTION	18
4	RESULT AND DISCUSSION	19
	4.1 SYSTEM REQUIREMENTS	19
	4.2 RESULT	19
5	CONCLUSION AND FUTURE WORK	20
	5.1 CONCLUSION	20
	5.2 FUTURE WORK	20
	5.3 REFERENCE	20
	APPENDIX	21
	A.SCREENSHOTS	21
	B.SOURCE CODE	23

FIGURE NO	FIGURE NAME	PAGE NO
3.1	DATA FLOW	18
4.1	OUTPUT	19

LIST OF ABBREVIATIONS

ABBREVIATIONS

EXPANSION

HTML

HYPERTEXT MARK-UP LANGUAGE

CSS

CASCADING STYLE SHEETS

JS

JAVA SCRIPT

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. Crop yield prediction is one of the challenging problems in precision agriculture, and many models have been proposed and validated so far. This problem requires the use of several datasets since crop yield depends on many different factors such as climate, weather, soil, use of fertilizer, and seed variety. This indicates that crop yield prediction is not a trivial task; instead, it consists of several complicated steps.

1.2 OUTLINE OF THE PROJECT

Machine Learning studies consist of different challenges when aiming to build a high-performance predictive model. It is crucial to select the right algorithms to solve the problem at hand, and in addition, the algorithms and the underlying platforms need to be capable of handling the volume of data. This project mainly focuses on creating a crop recommendation system using Machine learning algorithm to predict which crop can give a better yield with the conditions of a particular state or country. By applying Random forest algorithm in Machine learning model to train a dataset containing details like rainfall, pH value, humidity and temperature, we can recommend the crop suitable for a particular area

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The main objective of this project is to recommend the most suitable crop based on the given conditions of the place. This crop is recommended so as to get a better yield.

2.2 PROBLEM STATEMENT

Farmers are the backbone of our nation. But due to illiteracy they face lot of financial crisis. This project is designed so as to help the farmers to get a better idea on which crop they will have to invest to get a better yield and make profit out of it for their living.

2.3 SCOPE

The scope of the project is to determine the crop yield of an area by considering dataset with some features which are important or related to crop production such as temperature, moisture, rainfall, and production of the crop in previous years. To predict a continuous value, regression models are used. It is a supervised technique. The coefficients are preprocessed and fit into the trained data during training and construction the regression model. The main focus here is to reduce the cost function by finding the best fit-line. The output function facilitates in error measurement. During training period, error between the predicted and actual values is reduced in order to minimize error function. Python is used for this project by using Jupyter platform for simulation.

CHAPTER 3

SYSTEM ANALYSIS

3.1 GENERAL

3.1.1 MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

3.1.2 PYTHON

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

3.1.3 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

IDE stands for Integrated Development Environment. It's a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools,... And even though the IDE is a strictly defined concept, it's starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs. The IDE used in this project is Anaconda Navigator.

3.1.4 USER INTERFACE

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website

3.1.4 PYTHON WEB FRAMEWORK

A Web framework is a collection of packages or modules which allow developers to write Web applications or services without having to handle such low-level details as protocols, sockets or process/thread management. The Python web framework used in this project is Flask.

3.2 SOFTWARES USED

3.2.1 CODE EDITORS USED

VISUAL STUDIO CODE

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and MacOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is an integrated development environment developed by data scientists and is best suitable for Machine learning and Data science.

3.2.2 PROGRAMMING LANGUAGES USED

HTML5

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

CSS3

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

JAVASCRIPT

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript, often abbreviated as JS, is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

3.2.3 LIBRARIES AND FRAMEWORKS USED

FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

SCIKIT-LEARN

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

PANDAS

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

PICKLE

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.)

BOOTSTRAP

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

3.3 ALGORITHM

3.3.1 RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

3.3.2 MIN-MAX SCALER

The MinMaxScaler is the probably the most famous scaling algorithm, and follows the following formula for each feature: $\frac{x_i - \min(x)}{\max(x) - \min(x)}$ It essentially shrinks the range such that the range is now between 0 and 1 (or -1 to 1 if there are negative values). It is a preprocessing and feature scaling method inbuilt in Scikit-learn library.

3.3.3 Decision Tree

The Decision tree classifiers uses greedy approach hence an attribute chooses at first step can't be used anymore which can give better classification if used in later steps. Also it overfit the training data which can give poor results for unseen data. So, to overcome this limitation ensemble model is used. In ensemble model results from different models are combined. The result obtained from an ensemble model is usually better than the result from any one of individual models

3.4 PROJECT DESCRIPTION

This project aims at recommending the most suitable crop for the given climatic conditions using machine learning algorithms. I have created a python file which contains the machine learning algorithm. There is a user interface which I have created to display the output. The python file and the user interface is connected using a flask. I have imported python libraries such as pandas, numpy, pickle, sklearn to classify and process the data. I have used the Random classifier algorithm, The Random Forest Classifier is a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object. The MinMaxScaler transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one. The command prompt is opened and the virtual environment containing all the packages is activated. The app directory is opened in the CMD. We can open the app by running the command “python app.py”. The user interface developed using bootstrap is then opened.

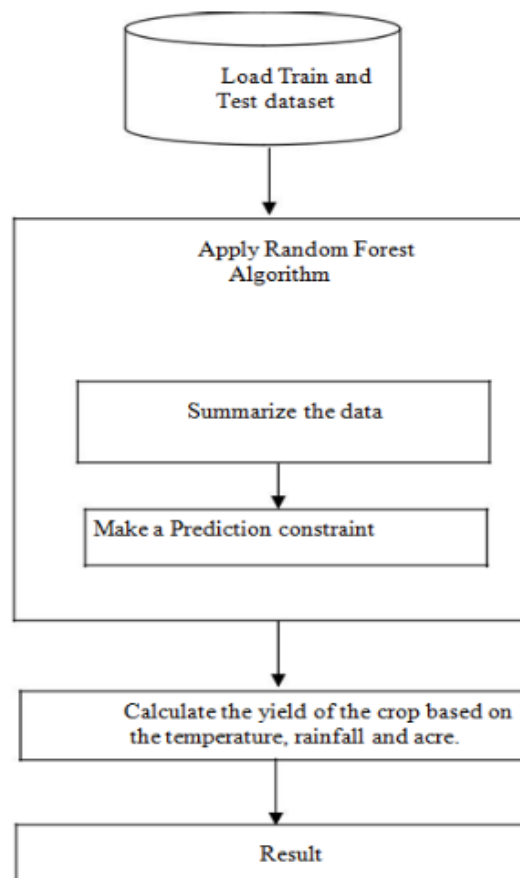


FIG 3.1 DATA FLOW

CHAPTER 4

RESULTS AND DISCUSSION

4.1 SYSTEM REQUIREMENTS

- Operating System: Windows/Mac
- RAM: 4GB
- Processor: 64x 1.0Ghz processor
- ROM: 8GB

4.2 RESULT

The Results shows that we can attain an accurate crop yield prediction using the Random Forest algorithm. Random Forest algorithm achieves a largest number of crop yield models with a lowest models. It is suitable for massive crop yield prediction in agricultural planning. This makes the farmers to take the right decision for right crop such that the agricultural sector will be developed by innovative ideas.

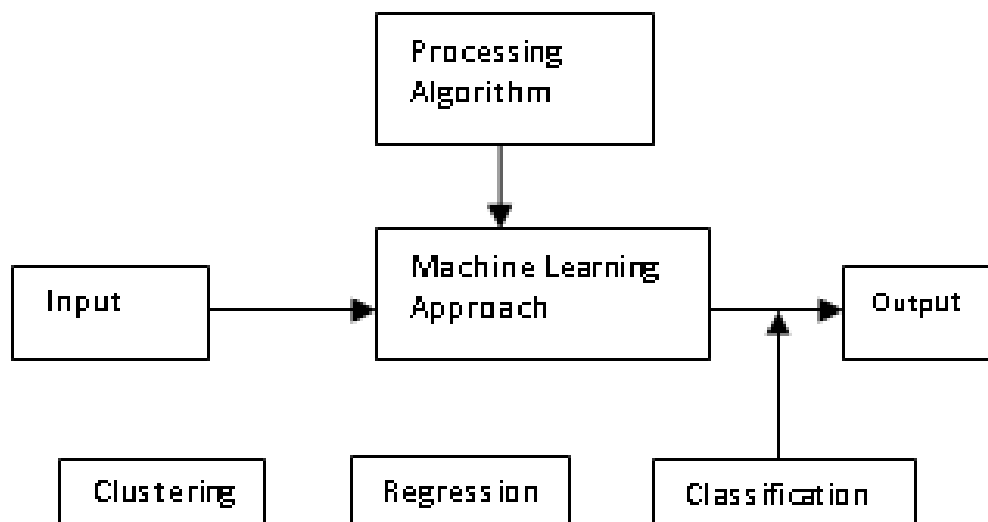


FIG 4.1 DATA FLOW

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This project is undertaken using machine learning and evaluates the performance by using Random forest, Polynomial Regression and Decision Tree algorithms. In our proposed model among all the three algorithm Random forest gives the better yield prediction as compared to other algorithms. Along with random forest, Polynomial Regression, Decision Tree model classify the output that shows improvements in dataset. So we analyzed that proposed model has got more efficiency than the existing model for finding crop yield. The implementation of above system would help in better cultivation of the agricultural practices of our country. Further it can be used to reduce the loss faced by the farmers and improve the crop yield to get better capital in agriculture. The model can be improved by integrating this with other departments like horticulture, sericulture, and others towards the agricultural development of our country.

5.2 FUTURE WORK

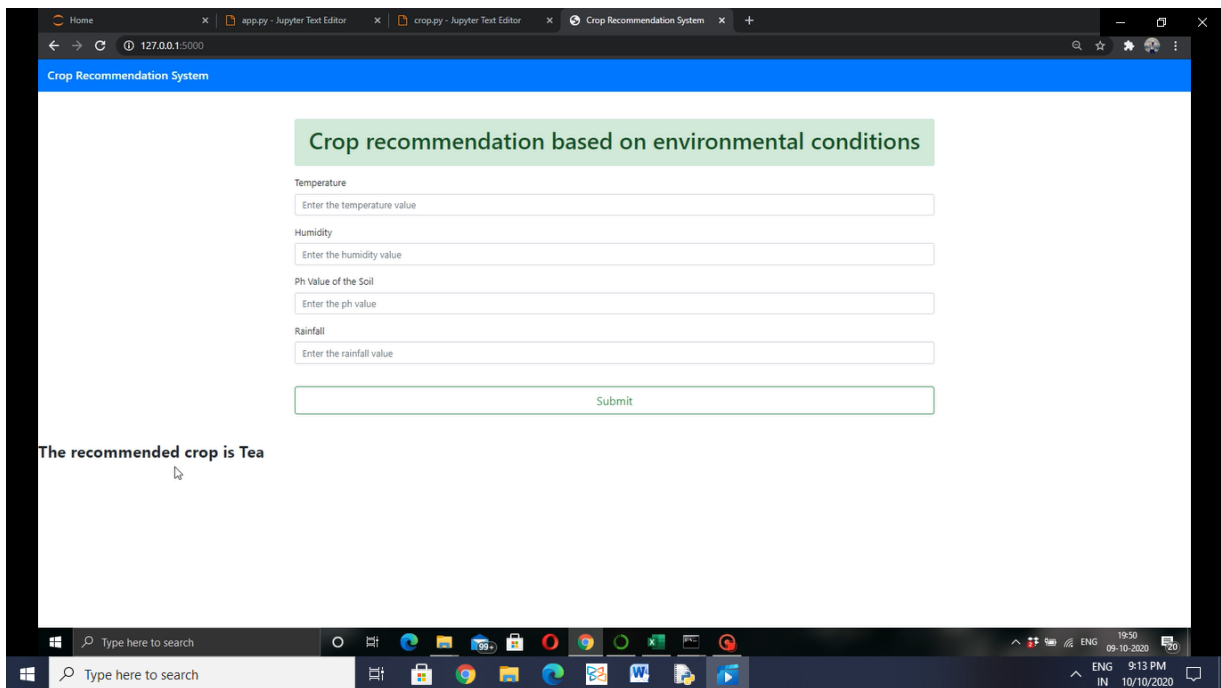
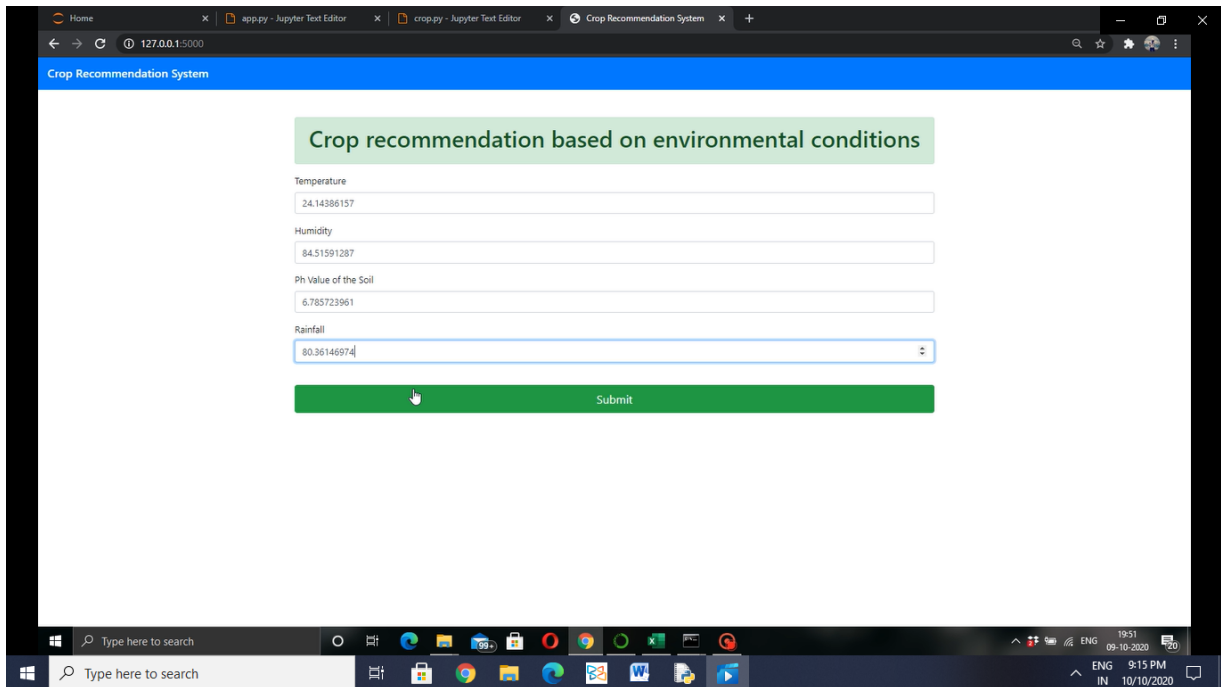
This project describes crop yield prediction ability of the algorithm. In future we can determine the efficient algorithm based on their accuracy metrics that will helps to choose an efficient algorithm for crop yield prediction

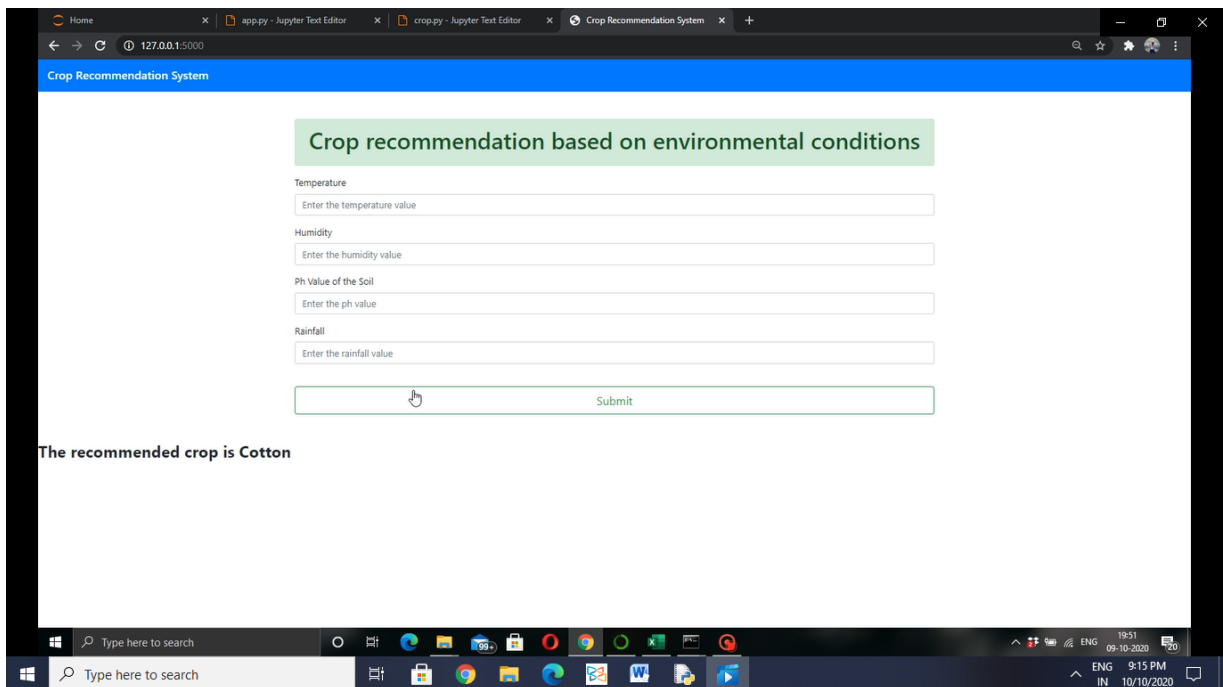
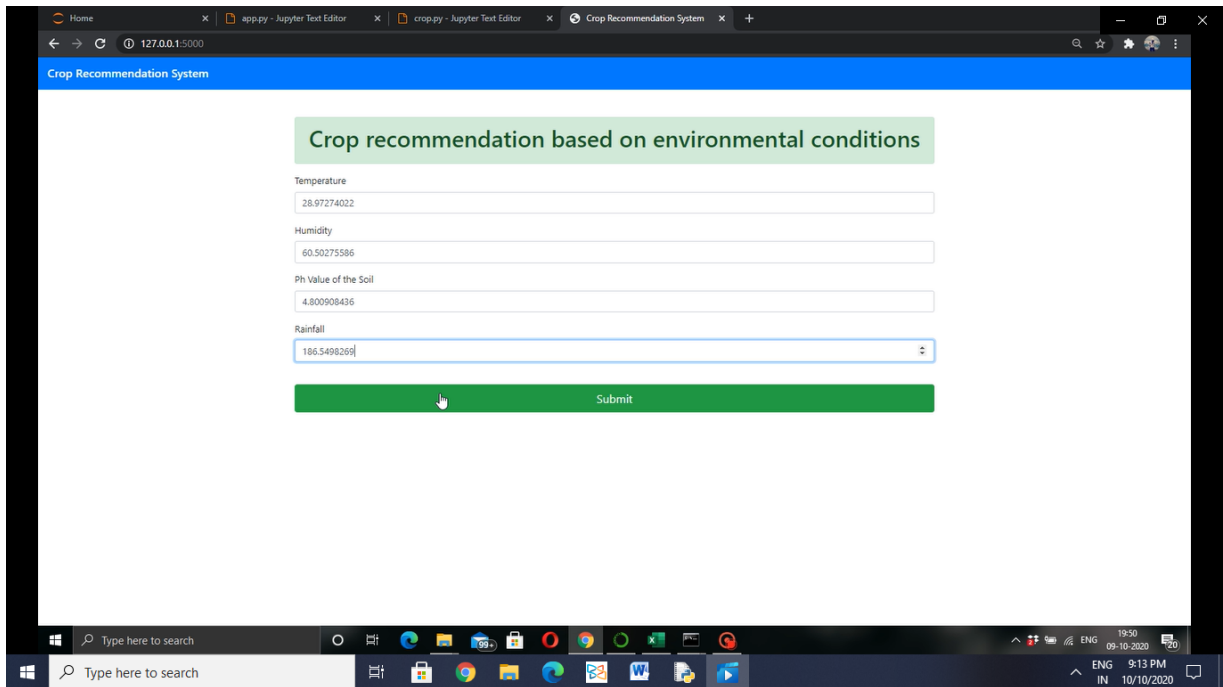
5.3 REFERENCES

1. <https://getbootstrap.com/>
2. <https://www.w3schools.com/html/default.asp>
3. https://www.w3schools.com/html/html_css.asp
4. <https://jupyter.org/try>

APPENDIX

A.SCREENSHOTS





B.SOURCE CODE

1. CROP.PY

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
df=pd.read_csv('cpdata.csv')
features=['temperature','humidity','ph','rainfall']
X=df[features]
df.label=LabelEncoder().fit_transform(df.label)
print(df.label.unique)
Y=df.label

from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(X,Y)
preds=rf.predict(X)
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(Y,preds))
print(confusion_matrix(Y,preds))
import pickle
with open('crop.pth','wb') as f:
    pickle.dump(rf,f)
```

2. APP.PY

```
from flask import Flask, render_template, request
#import jsonify
import requests
import pickle
import numpy as np
import sklearn
```

```

import pandas as pd
import joblib
import pickle
import pickle

#from sklearn.preprocessing import MinMaxScaler
app = Flask(__name__,template_folder='templates')
with open('crop.pkl','rb') as f:
    model=pickle.load(f)

#standard_to = MinMaxScaler()cd
@app.route("/", methods=['GET','POST'])
def main():
    #Fuel_Type_Diesel=0
    if request.method == 'POST':
        temperature=request.form.get('temperature')
        humidity=request.form.get('humidity')
        ph=request.form.get('ph')
        rainfall=request.form.get('rainfall')
        crop=['Adzuki Beans', 'Black gram', 'Chickpea', 'Coconut', 'Coffee', 'Cotton',
'Ground Nut', 'Jute', 'Kidney Beans', 'Lentil', 'Moth Beans', 'Mung Bean', 'Peas', 'Pigeon
Peas', 'Rubber', 'Sugarcane', 'Tea', 'Tobacco', 'apple', 'banana', 'grapes', 'maize', 'mango',
'millet', 'muskmelon', 'orange', 'papaya', 'pomegranate', 'rice', 'watermelon', 'wheat']

        prediction=model.predict([[temperature,humidity,ph,rainfall]])
        print(prediction)
        #output=round(prediction[0],2)
        #rec_crop=crop[prediction]
        if prediction==0:
            return render_template('main.html',result='The recommended crop is Adzuki Beans')
        if prediction==1:
            return render_template('main.html',result='The recommended crop is Black gram')

```

```

if prediction==2:
    return render_template('main.html',result='The recommended crop is Chickpea')
if prediction==3:
    return render_template('main.html',result='The recommended crop is Coconut')
if prediction==4:
    return render_template('main.html',result='The recommended crop is Coffee')
if prediction==5:
    return render_template('main.html',result='The recommended crop is Cotton')
if prediction==6:
    return render_template('main.html',result='The recommended crop is Ground Nut')
if prediction==7:
    return render_template('main.html',result='The recommended crop is Jute')
if prediction==8:
    return render_template('main.html',result='The recommended crop is Kidney Beans')
if prediction==9:
    return render_template('main.html',result='The recommended crop is Lentil')
if prediction==10:
    return render_template('main.html',result='The recommended crop is Moth Beans')
if prediction==11:
    return render_template('main.html',result='The recommended crop is Mung Bean')
if prediction==12:
    return render_template('main.html',result='The recommended crop is Peas')
if prediction==13:
    return render_template('main.html',result='The recommended crop is Pigeon Peas')
if prediction==14:
    return render_template('main.html',result='The recommended crop is Rubber')
if prediction==15:
    return render_template('main.html',result='The recommended crop is Sugarcane')
if prediction==16:
    return render_template('main.html',result='The recommended crop is Tea')
if prediction==17:
    return render_template('main.html',result='The recommended crop is Tobacco')
if prediction==18:
    return render_template('main.html',result='The recommended crop is apple')

```

```

if prediction==19:
    return render_template('main.html',result='The recommended crop is banana')
if prediction==20:
    return render_template('main.html',result='The recommended crop is grapes')
if prediction==21:
    return render_template('main.html',result='The recommended crop is maize')
if prediction==22:
    return render_template('main.html',result='The recommended crop is mango')
if prediction==23:
    return render_template('main.html',result='The recommended crop is millet')
if prediction==24:
    return render_template('main.html',result='The recommended crop is muskmelon')
if prediction==25:
    return render_template('main.html',result='The recommended crop is orange')
if prediction==26:
    return render_template('main.html',result='The recommended crop is papaya')
if prediction==27:
    return render_template('main.html',result='The recommended crop is pomegranate')
if prediction==28:
    return render_template('main.html',result='The recommended crop is rice')
if prediction==29:
    return render_template('main.html',result='The recommended crop is watermelon')
if prediction==30:
    return render_template('main.html',result='The recommended crop is wheat')

else:
    return render_template('main.html')

if __name__=="__main__":
    app.run(debug=True)

```

3. MAIN.HTML

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGM
N5t9UJ0Z" crossorigin="anonymous">

    <title>Crop Recommendation System</title>
  </head>
  <body>
    <nav class="navbar navbar-dark bg-primary">
      <span class="navbar-brand mb-0 h1 ">Crop Recommendation System </span>
    </nav>
    <br>
    <br>
    <div class="container">
      <div class="alert alert-success" role="alert">
        <h1 class="text-center">Crop recommendation based on environmental conditions</h1>
      </div>

      <form action="{{ url_for('main')}}" method="post">
        <div class="form-group">
          <label for="formGroupExampleInput">Temperature</label>
          <input type="number" step="any" class="form-control" name="temperature"
placeholder="Enter the temperature value">
        </div>
        <div class="form-group">
```

```

        <label for="formGroupExampleInput2">Humidity</label>
        <input type="number" step="any" class="form-control" name="humidity"
placeholder="Enter the humidity value">
    </div>
    <div class="form-group">
        <label for="formGroupExampleInput">Ph Value of the Soil</label>
        <input type="number" step="any" class="form-control" name="ph"
placeholder="Enter the ph value">
    </div>
    <div class="form-group">
        <label for="formGroupExampleInput2">Rainfall</label>
        <input type="number" step="any" class="form-control" name="rainfall"
placeholder="Enter the rainfall value">
    </div>
    <br>
    <button type="submit" class="btn btn-outline-success btn-lg btn-
block">Submit</button>
    </form>
</div>
<br><br><h3><b>{{ result }}</b></h3>
</div>
</div>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VVKalzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>

```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPIYxofvL8/KUEfYiJOMMV
+rV" crossorigin="anonymous"></script>
</body>
</html>
```