MIT Academy of Engineering

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

TY MINI PROJECT REPORT

ON

CROP YIELD PREDICTION

BY

SCETTY321	Prajakta Yadav	psyadav@mitaoe.ac.in
SCETTY322	Samruddhi Yadav	ssyadav@mitaoe.ac.in
SCETTY339	Snehal Thombare	ssthombare@mitaoe.ac.in

Project Advisor

Mrs : Mayura Kulkarni

SCHOOL OF COMPUTER ENGINEERING & TECHNOLOGY
MIT ACADEMY OF ENGINEERING
ALANDI (D), PUNE

2019 – 2020

CERTIFICATE

This is to certify that the Mini Project entitled " **Crop Yield Prediction** " has been carried out by Prajakta Yadav(SCETTY321) Samruddhi Yadav(SCETTY322) Snehal Thombare(SCETTY339) in partial fulfillment of Third Year Computer Engineering as well as in the record of Mini-project work done by him/her at SCET,MIT AOE- an Autonomous institute affiliated to Savitribai Phule University, Pune under the guidance of " Mayura Kulkarni " during the academic year 2019-2020.

Sign:

Name:

Date: Prof. Ranjana Badre

Project Guide /Advisor **Sign of School Dean (SCET)**

Acknowledgement

I take this opportunity to record my profound gratitude and indebtedness to our Mini Project guide, Mrs. Mayura Kulkarni for her inspiring guidance, valuable advices, constant encouragement and untiring supervision throughout my research work.

Finally, I would like to acknowledge and express my special thanks to my family, friends and classmates for their patience, encouragement, support they have made during the period of this work.

Samruddhi Yadav
Prajakta Yadav
Snehal Thombare

Abstract

With the impact of climate change in India, majority of the agricultural crops are being badly affected in terms of their performance over a period of last two decades. Predicting the crop yield well ahead of its harvest would help the farmers for taking appropriate measures for marketing and storage. Several methods of predicting and modeling crop yields have been developed in the past with varying rate of success, as these don't take into account characteristics of the weather, and are mostly empirical. In the present study, a predictive model has been developed for predicting the crop yield (production).Multiple Linear Regression algorithm is used to find out the crop yields of selected crops. The features which we are using to predict the yield are area, soil ph, temperature, rainfall, humidity, nitrogen, and potassium and phosphorus contents in soil.

TABLE OF CONTENTS

INDEX

1. INTRODUCTION

1.1 Problem Definition/Objective:

With the increasing number of suicide rates and low productivity of crops, we want to help farmers to understand the importance of prior crop prediction, to flourish their basic knowledge about soil quality, understanding location-wise weather constraints, in order to achieve high crop yield through our technology solution. Earlier yield prediction was performed by considering the farmer's experience on a particular field and crop. However, as the conditions change day by day very rapidly, farmers are forced to cultivate more and more crops. Being this as the current situation, many of them don't have enough knowledge about the new crops and are not completely aware of the benefits they get while farming them. Also, the farm productivity can be increased by understanding and forecasting crop performance in a variety of environmental conditions. We are designing the predictive model to predict the yield of selected crop in selected states of India.

Objectives of Project:

- To design the predictive model using Machine Learning
- To predict the yield of selected crop
- To suggest the fertilizers according to the Nitrogen, Phosphorus and Potassium content in soil

1.2 Market Survey:

Crop yield prediction is an important agricultural problem. Each and Every farmer is always trying to know, how much yield will get from his expectation. In the past, yield prediction was calculated by analyzing farmer's previous experience on a particular crop. The Agricultural yield is primarily depending on weather conditions, pests and planning of harvest operation. Accurate information about history of crop yield is an important thing for making decisions related to agricultural risk management. Therefore, this paper proposes an idea to predict the yield of the crop. The farmer will check the yield of the crop as per the acre, before cultivating onto the field.

1.3 Software and Hardware Requirement Specification:

Software Requirements:

- Operating System Used: Linux
- Programming Language Used: Python
- Libraries Used: pandas,numpy,matplotlib,sklearn

Hardware Requirements:

- PC/Laptop

1.4 Target Specification

- The target of our project is to predict the yield of required crop with that of given environmental conditions
- Crop yield dataset comes with tags and therefore it comes under the supervised learning domain.
- This information can be used as training data for building a predictive model.

2. LITERATURE REVIEW

This paper proposed a ‘Crop Advisor’ that has been developed as an user friendly web page for predicting the influence of climatic parameters on the crop yields.C4.5 algorithm is used to find out the most influencing climatic parameter on the crop yields of selected crops in selected districts of Madhya Pradesh. This software provides an indication of relative influence of different climatic parameters on the crop yield, other agro-input parameters responsible for crop yield are not considered in this tool.

This paper focuses on six major crops of Bangladesh which are Aman rice, Aus rice, Boro rice, Potato, Wheat and Jute. The algorithms that were used are Multiple Linear Regression (MLR) and K-Nearest Neighbor Regression (KNNR). Multiple Linear Regression (MLR) gave the most accurate results during the analysis and was incorporated into an android application. The android application system is also able to prepare a schedule of the complete farming process for a chosen crop, e.g. the correct time to apply fertilizers, irrigation, etc.

This paper assist user the method that would help them to choose the crop. The different parameters like environmental, economic and other parameters related to the yield in nature can be analyzed for prediction of accurate resultant role. The economical parameters include demand for crop, market rate etc. whereas environmental parameters include quantity of rainfall, temperature, and type of soil. So, all these factors are considered while predicting the most efficient crop to be cultivated based on season.

This paper focus on predicting the yield of the crop based on the existing data by using Random Forest algorithm. Real data of Tamilnadu were used for building the models and the models were tested with samples. The prediction will help to the farmer to predict the yield of the crop before cultivating onto the agriculture field. To predict the crop yield in future accurately Random Forest, a most powerful and popular supervised machine learning algorithm is used.

3. SYSTEM ARCHITECTURE

3.1System Architecture

Dataset used:

We have used two datasets i.e. training and testing which contains the 10 features (columns) .The features which are included in dataset are Crop, Area (in hectares), Soil pH, Temperature (in degree Celsius), Rainfall (in mm),humidity,Nitrogen,phosphorus and potassium contents in soil(in grams) and Production (in tons).

	A	B	C	D	E	F	G	H	I	J
1	Crop	Area	Soil_pH	Humidity	Temperature	Rainfall	Nitrogen	Phosphorous	Potassium	Production
2	Maize	1	6.3	0.5	26	80	437	51	280	1113
3	Arhar/Tur	17600	4.8	0.6	28	40	515	38	241	6300
4	Bajra	274100	5.9	0.45	25	100	330	51	277	152800
5	Gram	40800	4.9	0.62	32	77	343	48	265	18600
6	Jowar	900	6.8	0.54	27	42	517	43	251	1100
7	Maize	4400	6.1	0.56	17	60	433	47	260	4700
8	Moong(Green Gram)	10200	4.3	0.64	16	60	298	45	272	900
9	Pulses total	451	5	0.62	30	40	453	48	258	130
10	Ragi	2600	6.5	0.7	25	95	295	43	280	2100
11	Rice	5900	5.3	0.58	28	121	469	48	260	7200
12	Sugarcane	45900	6.3	0.63	28	130	361	48	276	38940
13	Total foodgrain	3384	6.4	0.61	29	25	355	43	241	1836
14	Urad	1600	4.1	0.53	31	36	371	50	260	800
15	Jowar	598400	6.9	0.51	25	44	508	43	290	217000
16	Maize	6200	6.5	0.52	24	78	324	48	245	9100
17	Rabi pulses	60	4.8	0.67	22	44	453	39	289	3456
18	Wheat	79700	6.4	0.59	24	82	320	43	266	87100
19	Maize	1100	4.6	0.68	29	42	391	39	276	1900
20	Cotton(lint)	98	5.6	0.55	27	87	341	44	286	17900
21	Arhar/Tur	81200	4.9	0.63	22	77	376	51	257	64400
22	Bajra	5600	6.8	0.56	27	41	452	48	278	2400
23	Gram	37500	6.1	0.7	26	62	414	45	234	14800
24	Jowar	193600	5.3	0.64	35	60	428	46	265	290100
25	Maize	700	5.2	0.62	30	40	400	47	269	800
26	Moong(Green Gram)	119400	6.3	0.54	25	95	304	44	254	43100
27	Pulses total	2853	4.3	0.56	28	121	476	46	239	1497
28	Rice	2900	6.7	0.64	25	76	433	42	248	1600
29	Total foodgrain	4890	6.1	0.62	27	30	549	51	275	4451
30	Urad	81400	7.4	0.7	29	36	471	44	253	41000
31	Jowar	3400	5.3	0.58	22	54	321	46	283	900
32	Rabi pulses	10	9.4	0.63	21	77	350	41	280	7891
33	Wheat	20900	4.4	0.53	15	81	324	48	257	21800
34	Cotton(lint)	3582	7.2	0.51	29	42	511	47	271	183400
35	Arhar/Tur	83400	5.4	0.52	31	84	452	48	249	61300
36	Bajra	1700	6.3	0.67	22	72	328	39	272	700
37	Gram	48300	6.4	0.59	27	44	556	49	258	19400
38	Jowar	119500	6.3	0.68	33	62	377	44	282	174200
39	Maize	700	4.8	0.6	36	60	509	49	263	800
40	Moong(Green Gram)	53400	5.9	0.45	30	40	510	47	261	22300
41	Pulses total	1468	4.9	0.62	25	95	535	44	261	877
42	Rice	9500	6.8	0.54	28	120	358	39	241	5800
43	Total foodgrain	2799	6.1	0.56	25	76	392	42	236	2703
44	Urad	8800	4.3	0.64	27	30	486	40	251	3600
45	Jowar	200	5	0.53	16	36	493	50	261	100
46	Rabi pulses	10	6.5	0.51	19	54	473	44	251	8001
47	Wheat	16300	5.3	0.52	24	57	387	46	281	12800
48	Maize	100	6.3	0.67	35	62	398	40	232	100

Fig.3.1.1 Dataset

Libraries Used:

To implement this algorithm, we have used scikit-learn library. Scikit-learn is a free software machine learning library for the Python programming language. Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent

interface in python. It works with the numerical values. The library is focused on modelling the data.

Pandas is an open source Python library providing high performance and data manipulation and analysis tool. We are using it for loading the dataset into in-memory data objects from different file formats. NumPy is a general-purpose array-processing package. Matplotlib is visualization library in Python for 2d plots of arrays. We can draw any plots like line, scatter, bar, histogram. We have used this library to visualize the data and to analyze how one variable affects the other.

Algorithm Used:

Multiple linear regression (MLR) is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and response (dependent) variable.

Multiple regression is an extension of simple linear regression. It is used when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable). The variables we are using to predict the value of the dependent variable are called the independent variables (or sometimes, the predictor, explanatory or regressor variables).

Let y denotes the dependent (or study) variable that is linearly related to k independent (or explanatory) variables X1, X2,…., Xk through the parameters β1, β2,…, βk and we write

$$y=X1\beta1+X2\beta2+ \ldots.+Xk\beta k+\varepsilon$$

This is called as the multiple linear regression model. The parameters β1, β2,…, βk are the regression coefficients associated with X1, X2,…, Xk respectively and ε is the random error component reflecting the difference between the observed and fitted linear relationship

Tkinter:

It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter:

- Importing the module – tkinter
- Create the main window (container)
- Add any number of widgets to the main window
- Apply the event Trigger on the widgets.

3.2Data Flow Diagram

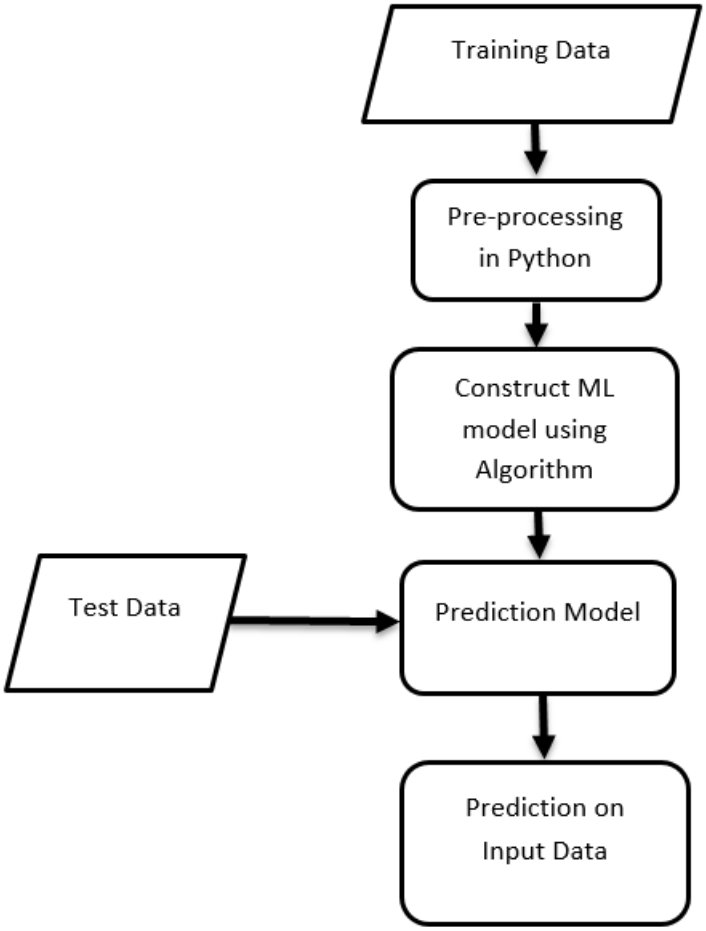

Fig3.2.1 Data Flow Diagram

3.3UML Diagram

Use Case Diagram

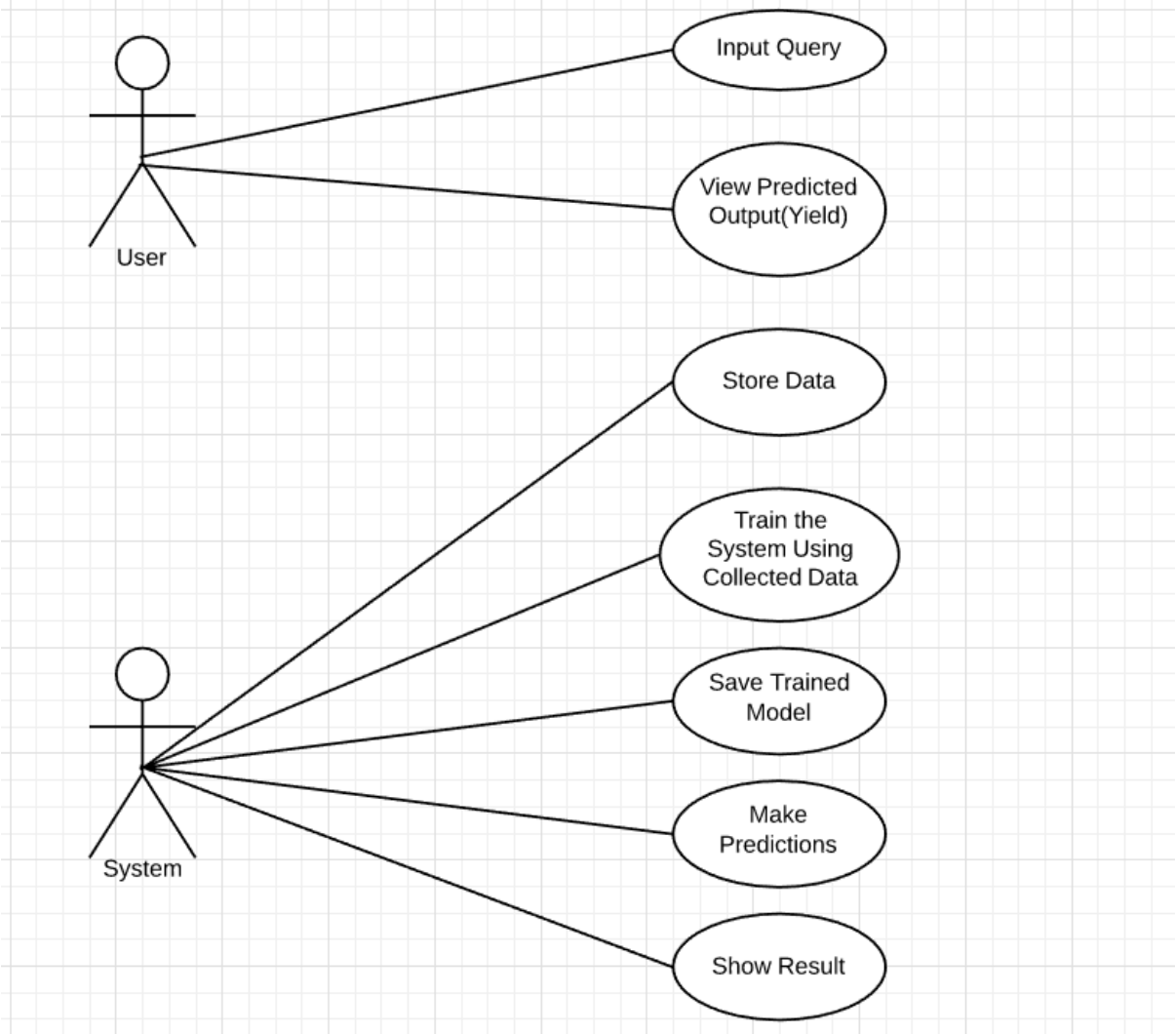

Fig3.2.2 Use Case Diagram

4. SIMULATION/SOFTWARE DEVELOPMENT FABRICATION OF PROJECT

4.1 Process of manufacturing or inventing something

We have collected the data from various sources and the dataset which we are using is in .csv format. After that, the required libraries i.e. sklearn, pandas, NumPy and matplotlib are installed

Before implementing the algorithm, data pre-processing has done in which the missing values are found out and impute for the missing values the median for each corresponding variable. In our dataset, Production feature has the missing values and to fill that missing values, we have taken the median of values. For that we have concatenated the dataset to find out the missing values in overall dataset.

We have used two datasets, i.e. Training and testing. First of all we have read the two datasets. After that, from testing dataset, we have dropped the value which we are going to predict. Modified dataset is converted into .csv format.X_train is the array which contains all the values of all features and it should contain only the input values. So, for that we have dropped Production column.Y_train contains values of production.X_test contains the whole modified testing dataset. So, now our data has splitted.

Then LinearRegression module has imported from the sklearn library and we have passed normalize=True that will normalize our dataset i.e. scaling between 0 and 1.After that we have trained our model using fit function. After training the model, we have predicted the results of X_test data. User input is taken in another numpy array called x_new and predicted the yield of user input.

Various graphs are plotted using matplotlib.One joint bar graph is plotted to show the comparison between actual and predicted values. And the impact of some features on production, how they are affecting, is plotted using bar graphs and scatter plot. One extra feature we have added i.e. user can view the list of all available nitrogenous, potassic and phosphatic fertilizers and can buy according to that. For that, we have used the textbox widget in Tkinter library.

After designing the predictive model, we have worked for GUI (Graphical User Interface) and it is done in Tkinter-a python library. We have created one window named crop yield prediction in which for crops there is option menu and for other features there are entry boxes in which user can enter any value. We have given one background image using Pillow library. There are 4 buttons like SUBMIT, SHOW RESULTS, SHOW GRAPHS and FERTILIZERS. In SUBMIT button, the details entered by the user will be submitted and displayed on the terminal. In SHOW RESULTS, the joint bar graph of actual and predicted yield values will be displayed. In SHOW GRAPHS, 3 graphs will be displayed which are impact of temperature, rainfall, humidity on production. In FERTILIZERS button, the details of nitrogenous, potassic and phosphatic fertilizers will be displayed to the user.

4.2 Process of Assembly, Testing, Troubleshooting

For execution of code, we have installed some libraries which are required for project such as pandas, numpy, matplotlib and scikit-learn.
The commands for installing the above libraries are-

- sudo yum install pandas
- sudo yum install matplotlib
- sudo yum install python scikit-learn

Sometimes we got the error of python version because we have installed the libraries for python 2.7 and not for python 3.So, we have to mention the version of python which are using to execute the code.

At once, we have defined one function and pass the parameters and when we call that function in another function, we have not passed the sufficient parameters. At that time we got the error.

For testing purpose, initially we have done data pre-processing.

```
snehal@snehal-Inspiron-5567:~$ python crop_final2.py
('Missing values of each feature:', Crop           0.000000
Area           0.000000
Soil_pH        0.000000
Humidity       0.000000
Temperature    0.000000
Rainfall       0.000000
Nitrogen       0.000000
Phosphorous    0.000000
Potassium      0.000000
Production     0.271739
source         0.000000
dtype: float64)
```

Fig 4.2.1Data Pre-processing Output

Before designing the GUI, we have implemented it on terminal.

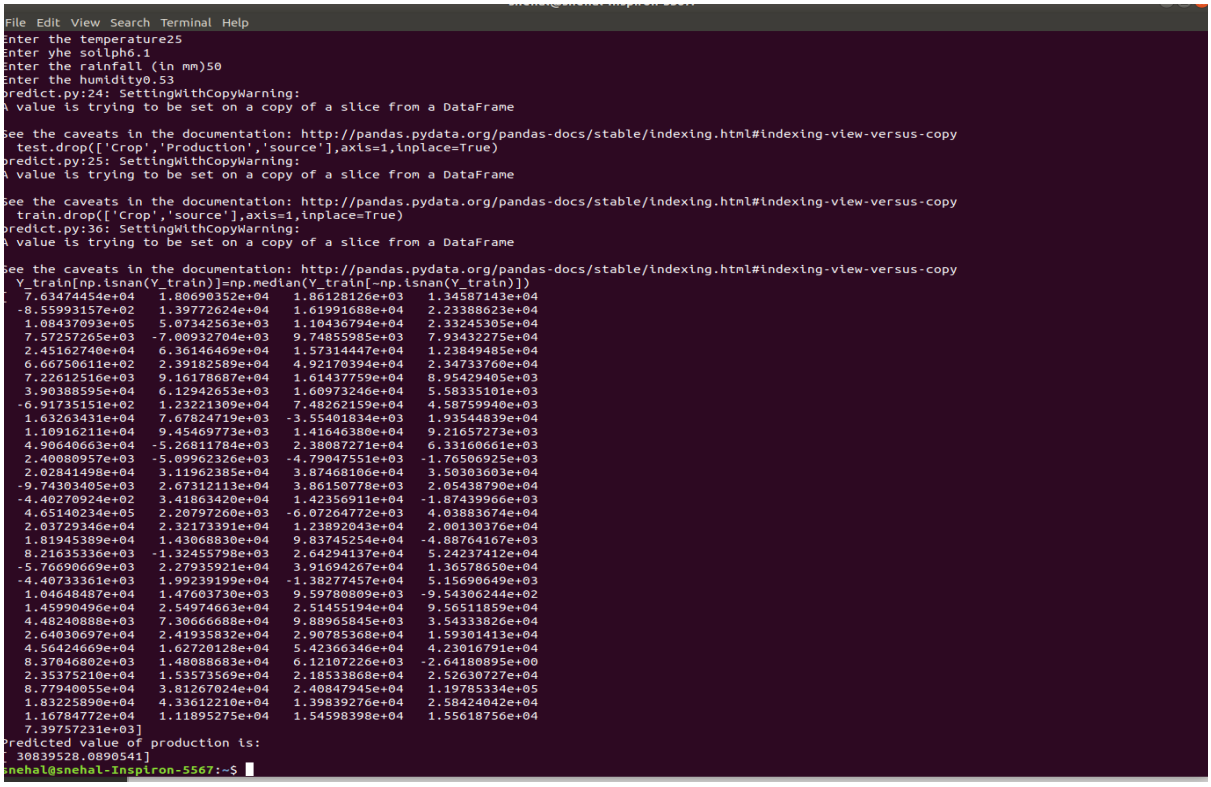

Fig 4.2.2 User Input on Terminal

Y_test Values:

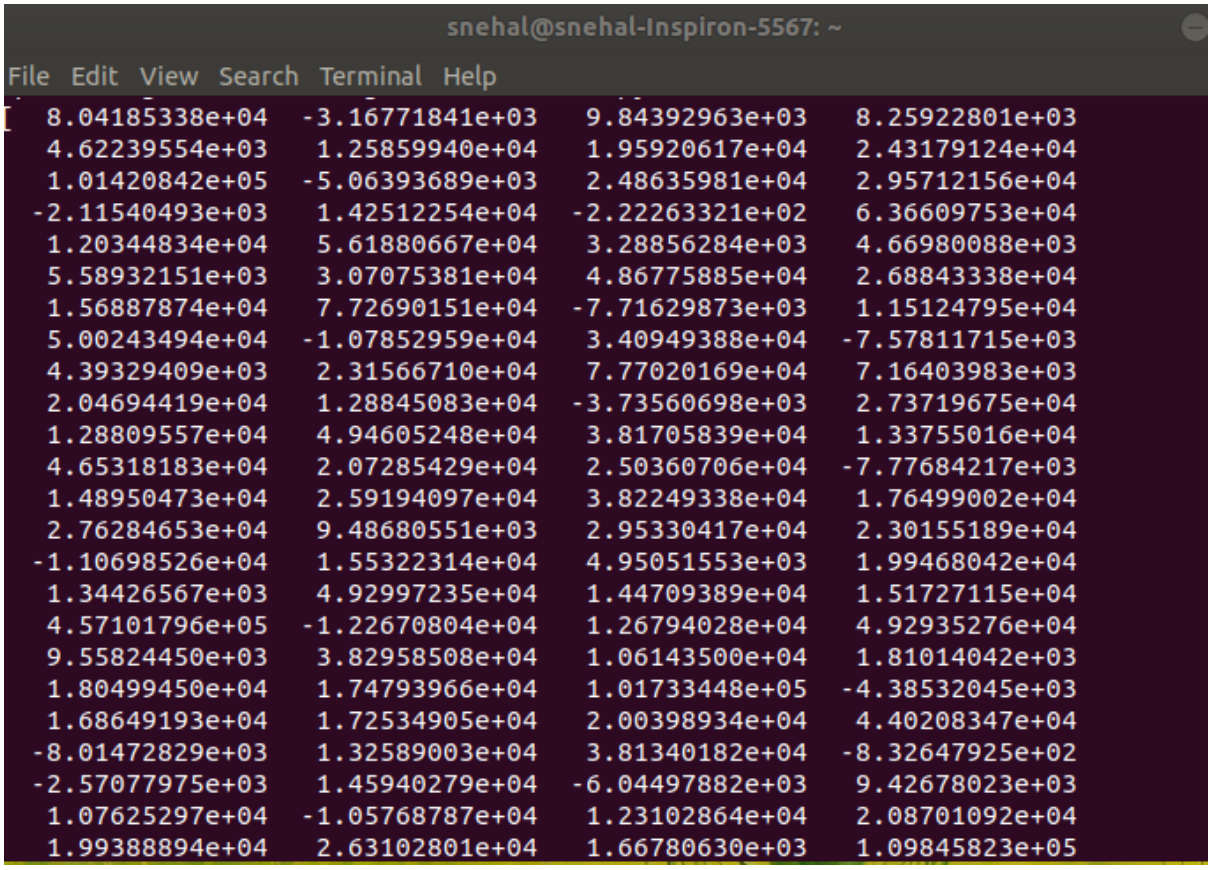

Fig 4.2.3 Predicted Y_test Values

4.3 Code:

```
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
#from sklearn.preprocessing import LabelEncoder, OneHotEncoder
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
train=pd.read_csv("data2/train2.csv")
test=pd.read_csv("data2/test2.csv")
crop={'Maize':0,'Arhar/Tur':1, 'Bajra':2, 'Gram':3, 'Jowar':4, 'Moong(Green Gram)':5,
      'Pulses total':6, 'Ragi':7, 'Rice':8, 'Sugarcane':9, 'Total foodgrain':10,
      'Urad':11, 'Rabi pulses':12, 'Wheat':13, 'Cotton(lint)':14, 'Groundnut':15,
      'Niger seed':16, 'Other Kharif pulses':17, 'Sesamum':18, 'Soyabean':19,
      'Sunflower':20, 'Safflower':21, 'Small millets':22,
```

```
    'Other Cereals & Millets':23, 'Rapeseed &Mustard':24}
train.Crop=[crop[item] for item in train.Crop]
test.Crop=[crop[item] for item in test.Crop]

train['source']='train'
test['source']='test'
data=pd.concat([train,test],ignore_index=True)
'''
le=LabelEncoder()
var_mod=['Crop']
for i in var_mod:
    data[i]=le.fit_transform(data[i])
data=pd.get_dummies(data,columns=['Crop'])
'''
miss=data.isnull().sum()/data.shape[0]*100
print("Missing values of each feature:",miss)
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
'''labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categorical_features = [3])
X = onehotencoder.fit_transform(X).toarray()
'''
train=data.loc[data['source']=="train"]
test=data.loc[data['source']=="test"]

Y_test=test['Production']
test.drop(['Production','source'],axis=1,inplace=True)
train.drop(['source'],axis=1,inplace=True)
train.to_csv("data/train_modified.csv",index=False)
test.to_csv("data/test_modified.csv",index=False)
train_df=pd.read_csv('data/train_modified.csv')
test_df=pd.read_csv('data/test_modified.csv')
X_train=train_df.drop(['Production'],axis=1)
Y_train=train_df['Production']
```

```
X_test=test_df.copy()
X_train[np.isnan(X_train)]=np.median(X_train[~np.isnan(X_train)])
X_test[np.isnan(X_test)]=np.median(X_test[~np.isnan(X_test)])
Y_train[np.isnan(Y_train)]=np.median(Y_train[~np.isnan(Y_train)])
Y_test[np.isnan(Y_test)]=np.median(Y_test[~np.isnan(Y_test)])
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

lr = LinearRegression(normalize=True)
lr.fit(X_train, Y_train)
lr_pred = lr.predict(X_test)
#print "Accuracy: ", lr.score(X_train, Y_train) * 100
lr_accuracy = round(lr.score(X_train,Y_train) * 100,2)
print('Accuracy of Model: %.4g' %lr_accuracy)
from Tkinter import *
application=Tk()
application.title("Crop Yield Prediction")
from PIL import ImageTk,Image
img = ImageTk.PhotoImage(Image.open("farm3.png"))
#Displaying it
imglabel = Label(application, image=img).grid(row=1, column=1)
lbl1=Label(application,text="  CROP  YIELD  PREDICTION
",fg="white",bg="black",font="Times 24 bold").place(x=720,y=100)
cropList = ['Maize', 'Arhar/Tur', 'Bajra', 'Gram', 'Jowar', 'Moong(Green Gram)',
      'Pulses total', 'Ragi', 'Rice', 'Sugarcane', 'Total foodgrain',
      'Urad', 'Rabi pulses', 'Wheat', 'Cotton(lint)', 'Groundnut',
      'Niger seed', 'Other Kharif pulses', 'Sesamum', 'Soyabean',
      'Sunflower', 'Safflower', 'Small millets',
      'Other Cereals & Millets', 'Rapeseed &Mustard']
crop = StringVar()
crop.set(cropList[0])
```

```
label2 = Label(application, text="Select Crop ", font=("arial",15, "bold"),fg = "black").place(x=700, y=260)
cropMenu=OptionMenu(application,crop,*cropList).place(x=1000, y=260)
label3 = Label(application,text="Area (in Hectare)",font=('Times New Roman',14,'bold')).place(x=700,y=320)
area_1= IntVar()
cultarea = Entry(application,bd=2,width=20,textvariable=area_1,bg='White',font=('Times New Roman',12)).place(x=1000,y=320)
label4 = Label(application,text="Soil pH",font=('Times New Roman',14,'bold')).place(x=700,y=380)
ph = DoubleVar()
soph= Entry(application,bd=2,width=20,textvariable=ph,bg='White',font=('Times New Roman',12)).place(x=1000,y=380)
label6 = Label(application,text="Humidity (in Percentage)",font=('Times New Roman',14,'bold')).place(x=700,y=440)
humid_1= DoubleVar()
humidity=
Entry(application,bd=2,width=20,textvariable=humid_1,bg='White',font=('Times New Roman',12)).place(x=1000,y=440)
label7 = Label(application,text="Temperature (in C)",font=('Times New Roman',14,'bold')).place(x=700,y=500)
temp_1= IntVar()
temper= Entry(application,bd=2,width=20,textvariable=temp_1,bg='White',font=('Times New Roman',12)).place(x=1000,y=500)
label8 = Label(application,text="Rainfall (in mm)",font=('Times New Roman',14,'bold')).place(x=700,y=560)
rainfall_1 = IntVar()
rain= Entry(application,bd=2,width=20,textvariable=rainfall_1,bg='White',font=('Times New Roman',12)).place(x=1000,y=560)
label9 = Label(application,text="Nitrogen Content (in mg)",font=('Times New Roman',14,'bold')).place(x=700,y=620)
nitro = DoubleVar()
nitrogen= Entry(application,bd=2,width=20,textvariable=nitro,bg='White',font=('Times New Roman',12)).place(x=1000,y=620)
```

```
label10 = Label(application,text="Phosphorus Content (in mg)",font=('Times New Roman',14,'bold')).place(x=700,y=680)
phos= DoubleVar()
phosphorus= Entry(application,bd=2,width=20,textvariable=phos,bg='White',font=('Times New Roman',12)).place(x=1000,y=680)
label11 = Label(application,text="Potassium Content (in mg)",font=('Times New Roman',14,'bold')).place(x=700,y=740)
pota= IntVar()
potassium= Entry(application,bd=2,width=20,textvariable=pota,bg='White',font=('Times New Roman',12)).place(x=1000,y=740)

def do_it():
    print("Crop: "+str(crop.get()))
    print("Area: ",area_1.get())
    '''print("Soil pH: ",soph.get())
    print("Humidity: ",humid.get())
    print("Tempearture: ",temper.get())
    print("Rainfall: ",rainfall_1.get())'''

button=Button(application,text ='SUBMIT',font=('Times New Roman',18,'bold'),command=do_it,bg='black',fg='white',width=18,height=1).place(x=660,y=830)
def show_result():
    df = pd.DataFrame({'Actual': Y_test, 'Predicted': lr_pred})
    df1 = df.head(20)
    df1.plot(kind='bar',figsize=(16,10))
    plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
    plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
    plt.show()

button1=Button(application,text ='SHOW RESULT',font=('Times New Roman',18,'bold'),command=show_result,bg='black',fg='white',width=18).place(x=940,y=830)
```

```
def suggest():
        root = Tk()
        S = Scrollbar(root)
        T = Text(root, height=10, width=100)
        S.pack(side=RIGHT, fill=Y)
        T.pack(side=LEFT, fill=Y)
        S.config(command=T.yview)
        T.config(yscrollcommand=S.set)
        quote = """
                NITROGENOUS FERTILIZERS:
            1.Ammonium Sulphate
            2.Ammonium chloride
            3.Anhydrous ammonia
                4.Urea
                5.Ammonium Nitrate
                6.Calcium Cynamide

                POTASSIC FERTILIZERS:
                1.Potassium Chloride (Muriate of Potash)
                2.Potassium Sulphate
                3.Potassium Schoenite
                4.Potassium Chloride (Muriate of Potash) (Granular)

                PHOSPHATIC FERTILIZERS:
                1.Single Superphosphate (16% P2O 5  Powdered)
                2.Triple Superphosphate
                3.Bone meal, Raw
                4.Bone meal, Steamed
                5.Rockphosphate
                6.Single Superphosphate (16% P2O 5   Granulated)
                7.Superphosphoric Acid (70% P2O 5  ) (Liquid) """
        T.insert(END, quote)
        mainloop()
        #print("Urea")
```

```
button2=Button(application,text ='FERTILIZERS',font=('Times New Roman',18,'bold'),command=suggest,bg='black',fg='white',width=18).place(x=940,y=880)
def show_graph():
    plt.figure(figsize=(10,9))
    plt.xlabel("Temperature(in degree Celcius)")
    plt.ylabel("Production(in tons)")
    plt.title("Impact of Temperature on Production")
    plt.plot(train.Temperature,train["Production"],'.',alpha=0.5)
    plt.show()
    '''
    Humidity_pivot=\
    train.pivot_table(index='Humidity',values="Production",aggfunc=np.median)
    Humidity_pivot.plot(kind='bar',color='blue',figsize=(10,7))
    plt.xlabel("Humidity")
    plt.ylabel("Production(in tons)")
    plt.title("Impact of humidity on Production")
    plt.xticks(rotation=0)
    plt.show()
    Rainfall_pivot=\
    train.pivot_table(index='Rainfall',values="Production",aggfunc=np.median)
    Rainfall_pivot.plot(kind='bar',color='blue',figsize=(20,7))
    plt.xlabel("Rainfall")
    plt.ylabel("Production(in tons)")
    plt.title("Impact of Rainfall on Production")
    plt.xticks(rotation=0)
    plt.show()
    '''
button2=Button(application,text ='SHOW GRAPH',font=('Times New Roman',18,'bold'),command=show_graph,bg='black',fg='white',width=18).place(x=660,y=880)

application.mainloop()
crop=crop.get()
```

```
cultarea=area_1.get()
soph=ph.get()
humidity=humid_1.get()
temper=temp_1.get()
rain=rainfall_1.get()
nitrogen=nitro.get()
phosphorus=phos.get()
potassium=pota.get()


if crop=="Maize":
    crop=0
elif crop=="Arhar/Tur":
    crop=1
elif crop=="Bajra":
    crop=2
elif crop=="Gram":
    crop=3
elif crop=="Jowar":
    crop=4
elif crop=="Moong(Grren Gram)":
    crop=5
elif crop=="Pulses total":
    crop=6
elif crop=="Ragi":
    crop=7
elif crop=="Rice":
    crop=8
elif crop=="Sugarcane":
    crop=9
elif crop=="Total foodgrain":
    crop=10
elif crop=="Urad":
    crop=11
```

```
elif crop=="Rabi pulses":
    crop=12
elif crop=="Wheat":
    crop=13
elif crop=="Cotton(lint)":
    crop=14
elif crop=="Groundnut":
    crop=15
elif crop=="Niger Seed":
    crop=16
elif crop=="Other Kharif Pulses":
    crop=17
elif crop=="Seasamum":
    crop=18
elif crop=="Soyabean":
    crop=19
elif crop=="Sunflower":
    crop=20
elif crop=="Safflower":
    crop=21
elif crop=="Small Millets":
    crop=22
elif crop=="Other Cereals & Millets":
    crop=23
else:
    crop=24


x_new=np.array([[crop,cultarea,soph,humidity,temper,rain,nitrogen,phosphorus,potassiu
m]])
y_new = lr.predict(x_new)
Window2 =Tk()
Window2.title("MINI Project")
Window2.resizable(width=FALSE, height=FALSE)
```

```
Window2.geometry('730x500')

#Setting it up
img = ImageTk.PhotoImage(Image.open("farm3.png"))

#Displaying it
imglabel = Label(Window2, image=img).grid(row=1, column=1)

label = Label(Window2, text= "Predicted Yield", font=("arial",20, "bold"),fg = "white",bg="black").place(x=100, y=250)
ans = y_new[0]
label = Label(Window2, text= ans, font=("arial",20, "bold"),fg = "white",bg="black").place(x=350, y=250)
Window2.mainloop()
```

4.4 Results:

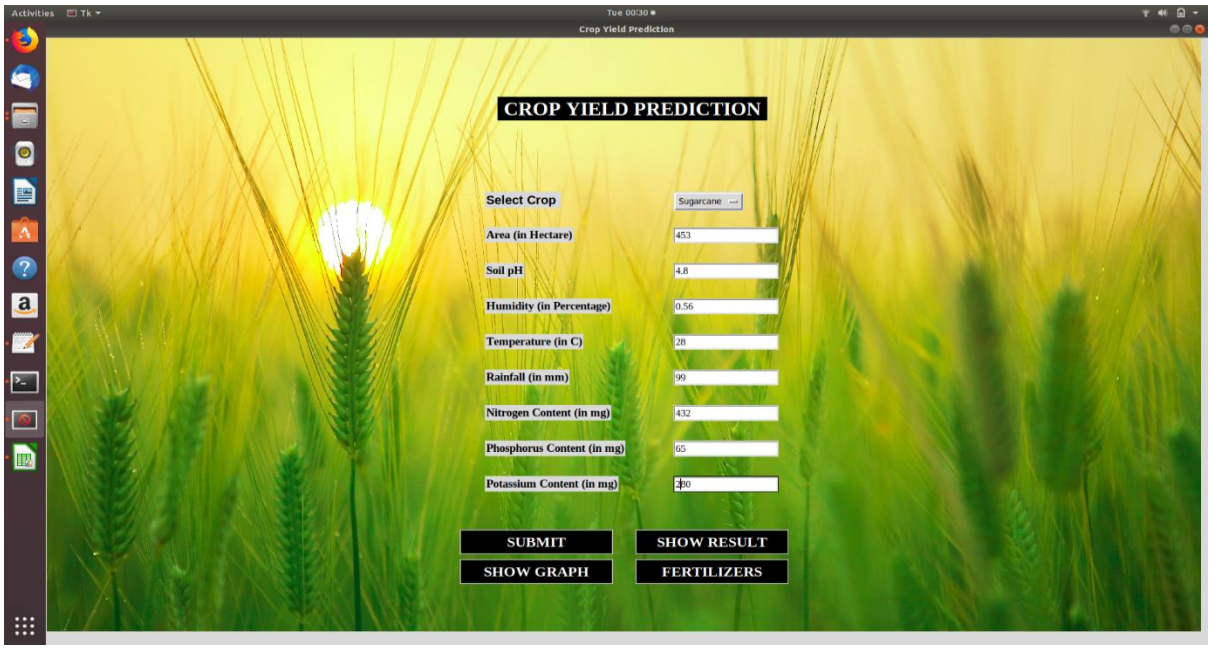

Fig 4.4.1 GUI

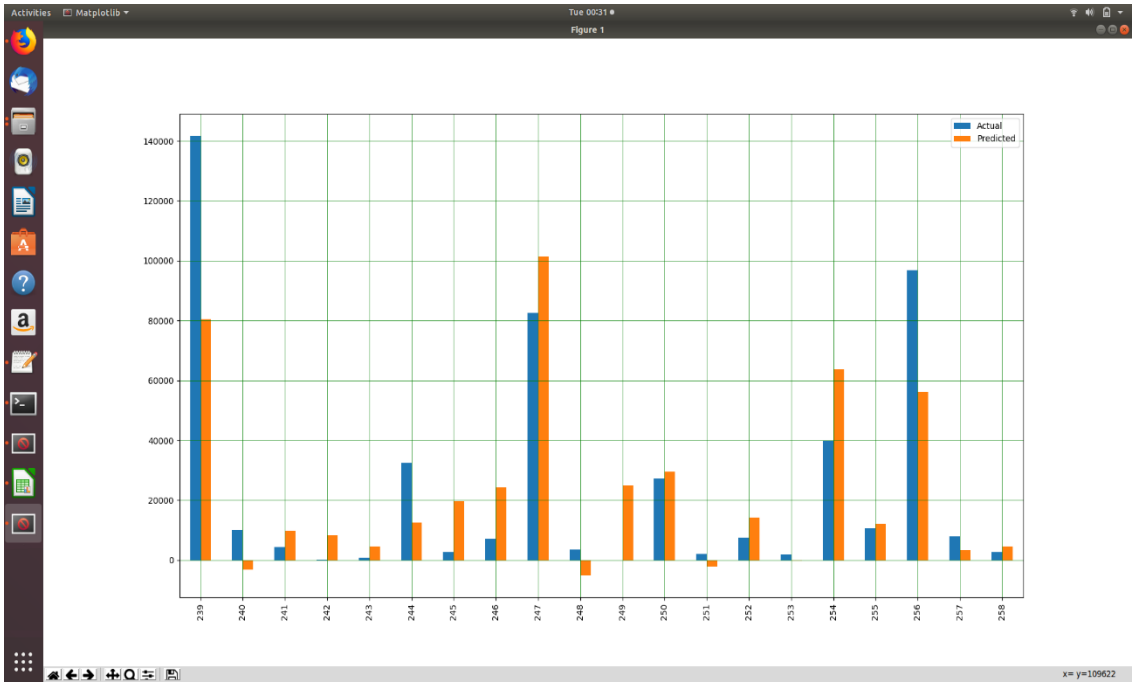

Fig 4.4.2 Comparison of Actual and Predicted Yield

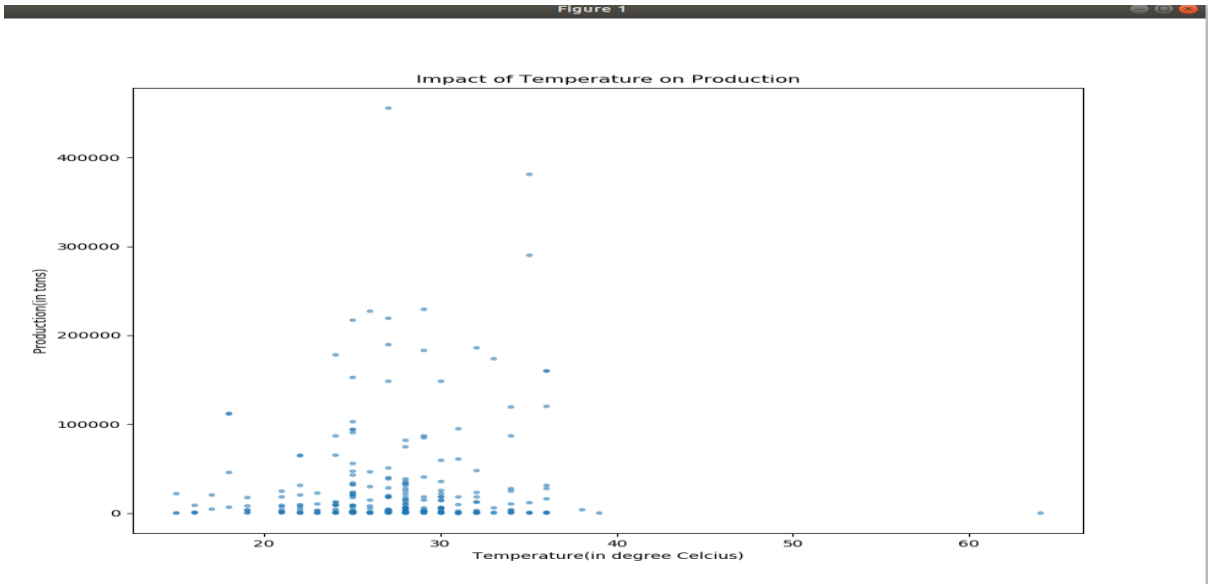

Fig 4.4.3 Scatter plot(Impact of Temperature on Production)

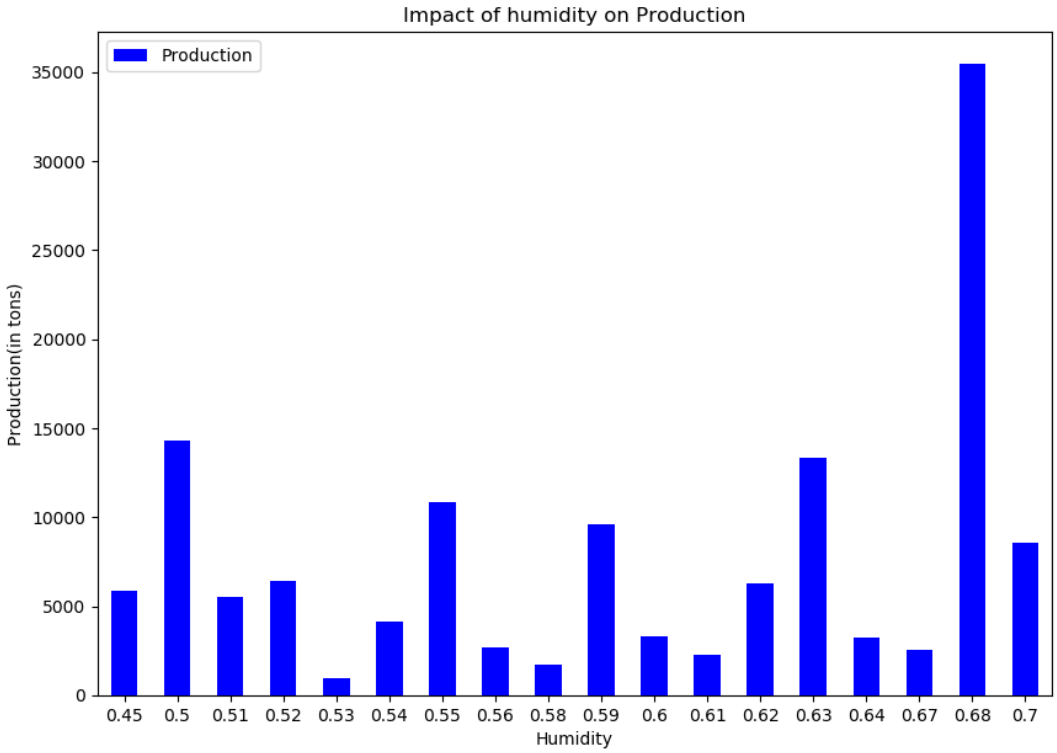

Fig 4.4.4 Bar Graph(Impact of Humidity on Production)

5. OTHER SPECIFICATIONS

5.1 Advantages

- It will be helpful in maximize the crop yield
- It will be helpful in predicting the yield of crop in the given environmental conditions
- The farmers can make decisions based on the predicted outcome that the given crop should be taken or not

5.2 Limitations

- It will not predict the yield much accurately. The accuracy is about
- All environmental parameters which are affecting the crop yield are not considered.
- All crops are not present in the dataset. So, farmers can see the predicted yield of only selected crops.

5.3 Applications

- It will be very useful to the farmers to achieve high crop yield through technological solution
- The farm productivity can be increased by understanding and forecasting crop performance in a variety of environmental conditions.
- It will help them to make decision about the cultivation of any crop

6. Conclusion and Future Work

The model which we have built predicts the production of the particular crop according to the different features like area, humidity, soil pH, temperature, rainfall, phosphorus, nitrogen, potassium.

In the future, all farming devices can be connected over the internet using IOT. The sensors i.e. temperature, humidity sensor etc. can be employed in farm which will collect the information about the current farm conditions and devices can increase the moisture, acidity, etc. accordingly.

REFERENCES

[1] Nishit Jain, Amit Kumar, Sahil Garud, Vishal Pradhan, Prajakta Kulkarni," Crop Selection Method Based on Various Environmental Factors Using Machine Learning"
[2] Rushika Ghadge, Juilee Kulkarni, Pooja More, Sachee Nene, Priya R L," Prediction of crop yield using machine learning"
[3] P.Priya, U.Muthaiah & M.Balamurugan," Predicting yield of the crop using machine learning algorithm"
[4] Prof.K.D.Yesugade,Aditi Kharde, Ketki Mirashi, Kajal Muley,Hetanshi Chudasama,"Machine Learning approach based on Agro-climatic conditions"
[5] Prof. D.S. Zingade,Omkar Buchade,Nilesh Mehta,Shubham Ghodekar,Chandan Mehta,"Crop Prediction System using Machine Learning"