Agriculture Crop Yield Prediction

of

Software Requirements Specification

Version: 1.0

Submitted in partial fulfillment of the requirements of

Minor Project

Instructor: Prof. Samaresh Mishra

INT ELLIGENCE

1706039	**Harsh Kr. Singh**	**Leader**
1706024	**Antara Thakuria**	**Analyst**
1706008	**Akash Pandey**	**Developer**

School of Computer Engineering

Kalinga Institute of Industrial Technology

Deemed to be University

Bhubaneswar-751024

May 2020

Revision History

Date	Description	Author	Comments
7, June 2020	1.0	Harsh Kr. Singh	

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Roll No	Member Name	Designation	Date	Signature
1706039	Harsh Kr. Singh	Leader	7, June 2020	
1706024	Antara Thakuria	Analyst	7, June 2020	
1706008	Akash Pandey	Developer	7, June 2020	
	Prof. Samaresh Mishra	Instructor	7, June 2020	

Table of Contents

Chapter 1 – Introduction

Monitoring agricultural crop conditions during the growing season and estimating the potential crop yields are both important for the assessment of seasonal production. Accurate and timely assessment of particularly decreased production caused by a natural disaster, such as drought and other climatic conditions can be critical for countries where the economy is dependent on the crop harvest. Early assessment of yield reductions could avert a disastrous situation and help in strategic planning to meet the demands.

Our project aims to convert a village into a smart village by eliminating agriculture-related issues under the Smart India scheme launched by the government.

1.1. Purpose

The purpose of this document is to present a detailed description of the Agricultural Crop Yield Prediction Software. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must be operated and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system. It will be proposed to all the agriculture-based places for their approval.

1.2. Scope of Project

This software is an Agricultural Crop yield Prediction Model for the agriculture-based places over the country. This model will be designed for the farmers whose crops are affected due to various factors like climatic conditions and soil-related issues etc. It is mostly seen due to such circumstance's farmers have to carry huge burdens of loans and ends up as suicide cases. Also, this project aims at improving the product quality and availability. Developing this project can assist the farmers and other stakeholders in better decision making in terms of agronomy and crop choice. Early detection of problems can help increase yield and subsequent profit thus fewer suicide cases. Today's world is all about technology. Agriculture is one of the precious sides which can decide the fate of lakhs and lakhs of people's life. Thus, using technology here in this field will improve the quality of life and full fill the aim of the Smart India approach.

1.3. References

1. Machine learning (E book) by AndrewNg
2. Yethiraj N G," Applying data mining techniques in the field of Agriculture and allied sciences", Vol 01, Issue 02, December 2012

3. Aditya Shastry, H. A Sanjayand E. Bhanushree, “Prediction of crop yield using Regression Technique”, International Journal of computing12 (2):96-102 2017, ISSN:1816-9503

4. For Datasets: “https://www.kaggle.com/datasets”; “https://www.data.gov/”

1.4. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter. The third chapter, Requirements Specification section, of this document, is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

Chapter 2 - Overall Description

The main objective of the Agriculture Crop Yield Production Model is that it helps in the growth of agricultural-based places. It works for the benefit of farmers who provide us the main energy source to live but they, on the other hand, are suffering. If seen from other's point of view that, it not only benefits the farmers but the whole country for the betterment of the nation. It reduces the hardships of the farmers cause by the amount of crops destroyed every year due to climate, soil and other factors by predicting the growth rate using past records and various other factors. Also, it helps the farmer with a better life and fewer debts thus reducing the number of suicide cases in most of the villages. Thus, this model results in the growth of a smart village.

2.1 Product Perspective

The existing system of crop production is very traditional and simple. Existing situation is just a process working with the past believes and chosen crops which has been cultivated in that period of time for many years without knowing the real cause about the destruction of more than 40% of the crops. So, we came up with the idea of the Crop Yield Prediction Model which is efficient and works using the past records which are accurate and provide details about the growth of the right crop in the right duration of time.

The main objectives of the proposed system are:

- Economic Feasibility

- Technical Feasibility

- User-friendly interface

2.1.1 Feasibility Study

Feasibility is a measure of how beneficial the development of the application will be to an organization. This is done by investigating the existing system in the area under investigation or general ideas about a new system. It is a test of the system proposal according to its workability, impact on the organization, ability to meet user’s needs and effective use of resources.

The main parts of feasibility for this system are:

2.1.2 Economic Feasibility

Economic analysis is a popular and frequently used method for evaluating the effectiveness of a proposed system. It is also known as cost-benefit analysis, the procedure to determine the benefits and saving expected from a system and compare them with costs.

The new system has much influence on the economical side of the place. Existing way of performing agriculture results in more loss than profit. Using this model will decrease the chances of crop destruction and help in more profit. Thus, the system is economically feasible.

2.1.3 Legal Feasibility

People are inherently resistant to change and computers have been known to facilitate change. The system is designed to facilitate each farmer with very little or no knowledge of computers. They will be granted access to any details related to crops they require from government offices.

2.1.4 Operational Feasibility

Proposed projects are considered beneficial only if they can be turned into an information system. That will meet the organization's operating needs. Some of the important issues raised are to test the operational feasibility of a project includes: Will there be any resistance from the user that will undermine the possible application benefits? Is there sufficient support for the management from the users?

This system is made such that it can positively fulfill these points. The well-planned design would ensure the optimal utilization of the computer resource and would help in the improvement of performance status.

2.1.5 Behavioral Feasibility

An estimate should be made of how strong the user is likely to move towards the development of the full technical system. These are various levels of users in order to ensure proper authentication, authorization, and security of sensitive data of the organization.

2.2 Product Functions

Agriculture Crop Yield Prediction Model is basically the methodology to predict the yield of the crops using different parameters like rainfall, soil types and past records. It provides an easy to use environment for both officials appointed for the working of the model and the farmers who would apply those in the practical world. The main objective is to provide all the features and details that the concerned people may need and must-have, with the "interface that don't scare its users!".

Agriculture Crop Yield Prediction:

1. It can be used in all the villages which mainly work for the agricultural sector.

2. Used by a government organization for predicting maximum profit from crop cultivation.

3. It is a cost-effective system as it doesn't have high-cost requirements.

4. Farmers will have no problem accessing the information as it is their right to information.

2.3 User Characteristics

Every user should have some basic requirements to gain the benefits of this system. Those are:

- The officials must be able to comfortably work with a computer.
- They must also have sufficient knowledge related to crop productions.

2.4 Constraints

Though our system gives almost perfect predictions which we have improvised by using various algorithms yet who can predict the occurrence of sudden climatic disasters? This is where our system fails as it cannot predict the sudden destruction caused due to natural disasters. Including this our project does not focus on the use of pesticides and chemicals.

2.5 Assumptions and Dependencies

First and foremost is that every record is considered accurate and well fitted for the use which is collected from tested and well-researched sites. Our total project depends on the records collected by the team.

Chapter 3 - Specific Requirements

This section specifies requirements that ensure the new product will connect properly to external components. User interfaces, hardware interfaces, software interfaces, and communication interfaces are described.

3.1 User Interfaces

The existing version of the software has a simple input-output based UI with a simple Graphical Interface. The user needs only to upload the required files (.csv files) to get the predicted result in a tabular form.

3.2 Software Interfaces

The following project is made using Python Programming Language in Google CoLabs. It has a simple interface with the basic input-output functions which generates the output in a tabular form.

3.3 Memory Constraints

Our project is based on mining and analyzing a large amount of data based like that in Big Data. So, for compiling a huge amount of data to compile the program you need a system with at least 4GB RAM and 20GB memory.

3.4 Operations

This project is a simple python program. So, the steps required to generate the output for the program is:

1) The user uploads the respected files or dataset int the program when run.
2) The machine learning algorithm compiles and prints the average root mean square error for different algorithms to compare them.
3) The most accurate algorithm is then used to predict the future production value in 1 Lakh KG/ yr. terms.
4) The predicted values are then shown in a tabular form.

3.5 Site Adaptation Requirements

The user needs only to have a Computer System with Python installed in it or with an uninterrupted internet connection to run the program. The user also needs a stable power supply to start the computer.

Chapter 4 - System Features

4.1 Input Data

We have used 5 different databases containing the required information for prediction both at the state level as well as the national level. The database we used have information about the Rice production in different states of Haryana for the past years, average rainfall in the state of Haryana for the last decade and last to last decade, Phosphorous Content of the soil in different states of Haryana, and the average rice production in India. Given Below are the description and content of some of the datasets we used.

```
[ ] rice_haryana = rice[rice["State_Name"]=="Haryana"]
    rice_haryana.head()
```

	Unnamed: 0	State_Name	ind_district	Crop_Year	Season	Crop	Area	Production	Value
1113	4210	Haryana	ambala	1997	kharif	Rice	65000.0	182000.0	2.800000
1114	4211	Haryana	ambala	1998	kharif	Rice	71365.0	186000.0	2.606320
1115	4212	Haryana	ambala	1999	kharif	Rice	72185.0	206000.0	2.853778
1116	4213	Haryana	ambala	2000	kharif	Rice	71840.0	217000.0	3.020601
1117	4214	Haryana	ambala	2001	kharif	Rice	74881.0	233000.0	3.111604

Fig 1: - Haryana State Rice Production Database

```
Xn.describe()
```

	Crop_Year	Production	X1	X2	X3	X4
count	399.000000	3.990000e+02	3.990000e+02	3.990000e+02	399.000000	399.000000
mean	2007.240602	1.992532e+05	2.003415e+05	2.007776e+05	736.783208	710.573684
std	1.813477	2.820193e+05	2.915813e+05	2.931155e+05	395.713328	373.922819
min	2005.000000	2.000000e+00	2.000000e+00	2.000000e+00	126.300000	108.800000
25%	2006.000000	5.812500e+03	4.876000e+03	4.671500e+03	476.500000	438.600000
50%	2007.000000	7.686800e+04	7.204800e+04	6.830000e+04	651.200000	638.600000
75%	2009.000000	2.820520e+05	2.791305e+05	2.914400e+05	879.150000	879.350000
max	2010.000000	1.637000e+06	1.710000e+06	1.710000e+06	2940.500000	2940.500000

X1 - Last Year Crop Production
X2 - Last to Last Year Crop Production
X3 - Present Year Rainfall
X4 - Last Year Rainfall

*All data are for the state of haryana

Fig 2: - Rice Dataset Description (Haryana-calculated from graph)

[29]

```
print(rice.describe())
print(rain1.describe())
print(rain2.describe())
```

```
Crop_Year          Area    Production        Value
count  13169.000000   13169.000000  1.314700e+04  13169.000000
mean    2005.762397   50640.056200  1.057203e+05      1.942265
std        5.063566   71019.932027  1.759126e+05      2.144512
min     1997.000000       1.000000  0.000000e+00      0.000000
25%     2001.000000    3200.000000  4.966500e+03      1.258584
50%     2006.000000   19000.000000  2.944800e+04      1.894216
75%     2010.000000   73275.890000  1.319115e+05      2.494297
max     2015.000000  687000.000000  1.710000e+06    223.727273
              Year        Value
count  4878.000000  4878.000000
mean   1998.000000  1157.397511
std       2.582254   717.308841
min    1994.000000    55.502000
25%    1996.000000   743.227500
50%    1998.000000   971.803500
75%    2000.000000  1301.227500
max    2002.000000  9357.259000
              Year        Value
count  3006.000000  3006.000000
mean   2007.007651  1242.812309
std       2.031272   872.137946
min    2004.000000     7.500000
25%    2005.000000   698.025000
50%    2007.000000   994.700000
```

Fig 3: - Rice and rain data description

```
rice.describe()
```

	Crop_Year	Area	Production	Value	X1	X2	X3	X4
count	5463.000000	5463.000000	5.449000e+03	5463.000000	5.090000e+03	4.809000e+03	2783.000000	2968.000000
mean	2005.394106	64076.532900	1.375553e+05	1.946222	1.047449e+05	1.309790e+05	1030.554382	1054.704046
std	4.949146	76112.336892	1.905742e+05	3.164509	1.821700e+05	1.918992e+05	623.505868	622.106761
min	1997.000000	1.000000	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	76.944000	118.200000
25%	2001.000000	7982.000000	1.006300e+04	1.212146	3.197750e+03	5.691000e+03	653.662000	677.295250
50%	2005.000000	32583.000000	5.785400e+04	1.852538	2.525150e+04	4.530000e+04	843.453000	883.950000
75%	2010.000000	101065.500000	2.059410e+05	2.483993	1.268458e+05	1.957000e+05	1190.236000	1197.745250
max	2015.000000	545965.000000	1.710000e+06	223.727273	1.710000e+06	1.710000e+06	4999.200000	5243.000000

Fig 4: - Rice and Rain data

[29]

```
print(rice.describe())
print(rain1.describe())
print(rain2.describe())
```

```
Crop_Year          Area    Production        Value
count  13169.000000   13169.000000  1.314700e+04  13169.000000
mean    2005.762397   50640.056200  1.057203e+05      1.942265
std        5.063566   71019.932027  1.759126e+05      2.144512
min     1997.000000       1.000000  0.000000e+00      0.000000
25%     2001.000000    3200.000000  4.966500e+03      1.258584
50%     2006.000000   19000.000000  2.944800e+04      1.894216
75%     2010.000000   73275.890000  1.319115e+05      2.494297
max     2015.000000  687000.000000  1.710000e+06    223.727273
              Year        Value
count  4878.000000  4878.000000
mean   1998.000000  1157.397511
std       2.582254   717.308841
min    1994.000000    55.502000
25%    1996.000000   743.227500
50%    1998.000000   971.803500
75%    2000.000000  1301.227500
max    2002.000000  9357.259000
              Year        Value
count  3006.000000  3006.000000
mean   2007.007651  1242.812309
std       2.031272   872.137946
min    2004.000000     7.500000
25%    2005.000000   698.025000
50%    2007.000000   994.700000
```


Fig 5: - Rice and rain data description

```
ricep.head()
```

	State_Name	ind_district	Crop_Year	Season	Crop	Area	Production	phosphorus	X1	X2	X3	X4
15	Andhra Pradesh	anantapur	1999	kharif	Rice	37991.0	105082.0	-1.0	96800.0	75400.0	643.720	881.473
16	Andhra Pradesh	anantapur	2000	kharif	Rice	39905.0	117680.0	-1.0	105082.0	96800.0	767.351	643.720
17	Andhra Pradesh	anantapur	2001	kharif	Rice	32878.0	95609.0	-1.0	117680.0	105082.0	579.338	767.351
18	Andhra Pradesh	anantapur	2002	kharif	Rice	29066.0	66329.0	-1.0	95609.0	117680.0	540.070	579.338
21	Andhra Pradesh	anantapur	2005	kharif	Rice	25008.0	69972.0	-1.0	85051.0	44891.0	819.700	564.500

Fig 6: - Haryana State Soil Phosphorous Content

```
ricep.head()
```

	Unnamed: 0	State_Name	ind_district	Crop_Year	Season	Crop	Area	Production	phosphorus	X1	X2	X3	X4
0	15	Andhra Pradesh	anantapur	1999	kharif	Rice	37991.0	105082.0	0.0	96800.0	75400.0	643.720	881.473
1	16	Andhra Pradesh	anantapur	2000	kharif	Rice	39905.0	117680.0	0.0	105082.0	96800.0	767.351	643.720
2	17	Andhra Pradesh	anantapur	2001	kharif	Rice	32878.0	95609.0	0.0	117680.0	105082.0	579.338	767.351
3	18	Andhra Pradesh	anantapur	2002	kharif	Rice	29066.0	66329.0	0.0	95609.0	117680.0	540.070	579.338
4	21	Andhra Pradesh	anantapur	2005	kharif	Rice	25008.0	69972.0	0.0	85051.0	44891.0	819.700	564.500

Fig 7: - Rice production dataset for all rice growing Indian States

4.2 Technologies and Algorithms

a) **Machine Learning:** It is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. We used Regression techniques to solve the problem and predict the Rice production.

b) **Linear Regression:** It is a linear approach for modelling the relationship between the scalar response and the one or more explanatory variables.

c) **Ridge Regression:** It is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. It adds a ridge parameter (k), of the identity matrix to the cross-product matrix, forming a new matrix (X`X + kI).

d) **Lasso Regression:** It is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. It is almost similar to ridge regression but the value of parameter (k) is different.

e) **Support Vector Regression (RBF kernel):** Kernel functions are used in support vector machines (SVM) to compute inner product in a higher dimensional feature space. The radial basis function (RBF) kernel is a distance-based kernel that has been successfully applied in many tasks. It is an example of classification.

f) **Support Vector Regression (Degree 3 and Degree 4 polynomial):** In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

4.3 Output Data

The program prints the results i.e. the predicted values in a tabular form with average RSME values for all the algorithms used in the project. Given below are actual and predicted results of the test cases from the dataset for India and the state of Haryana.

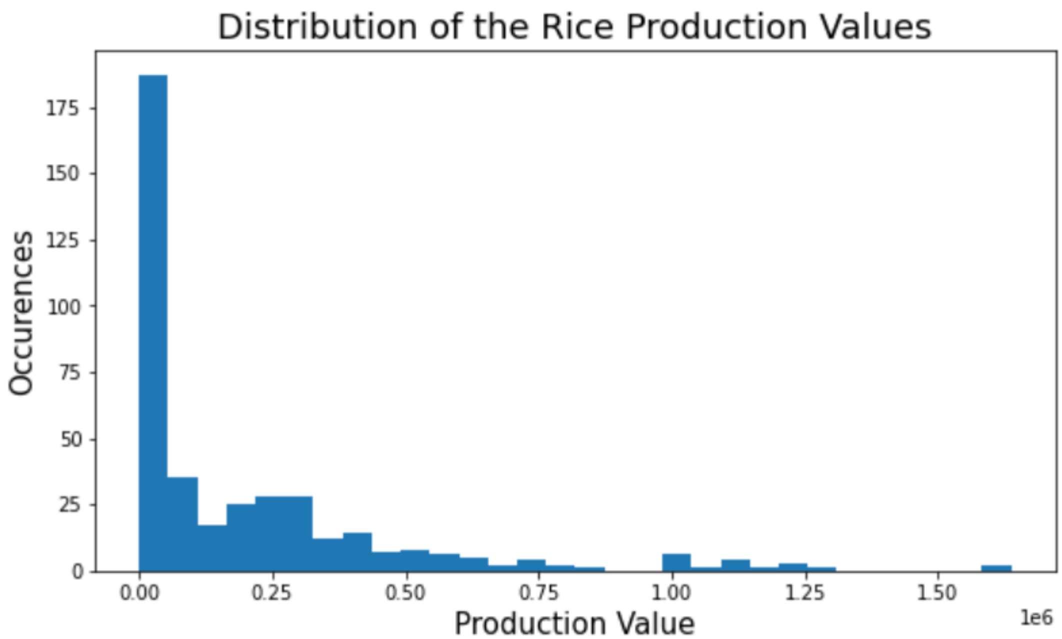
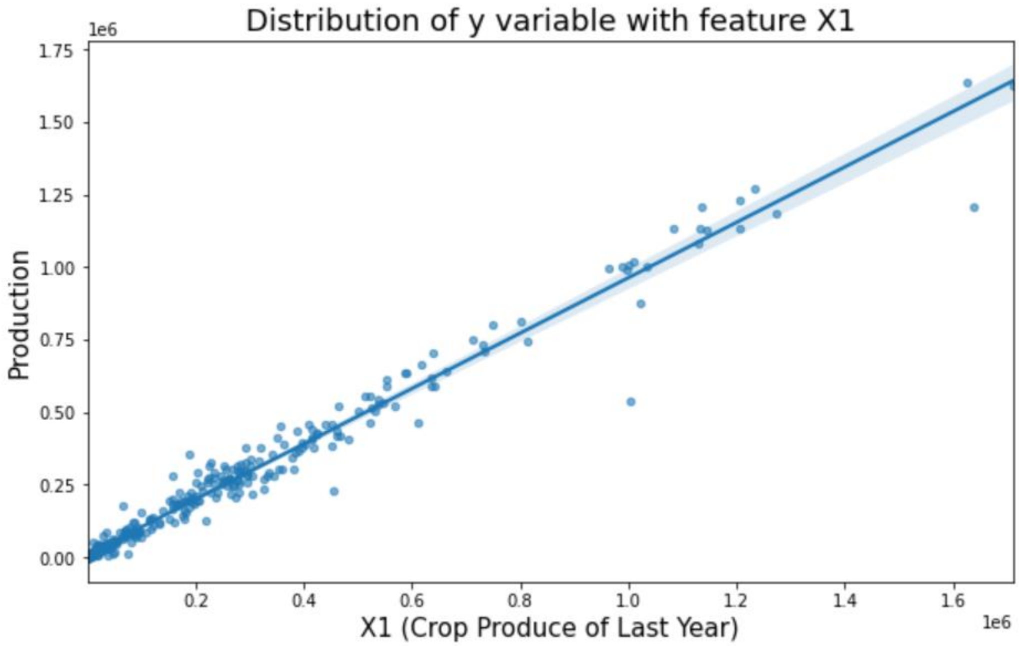
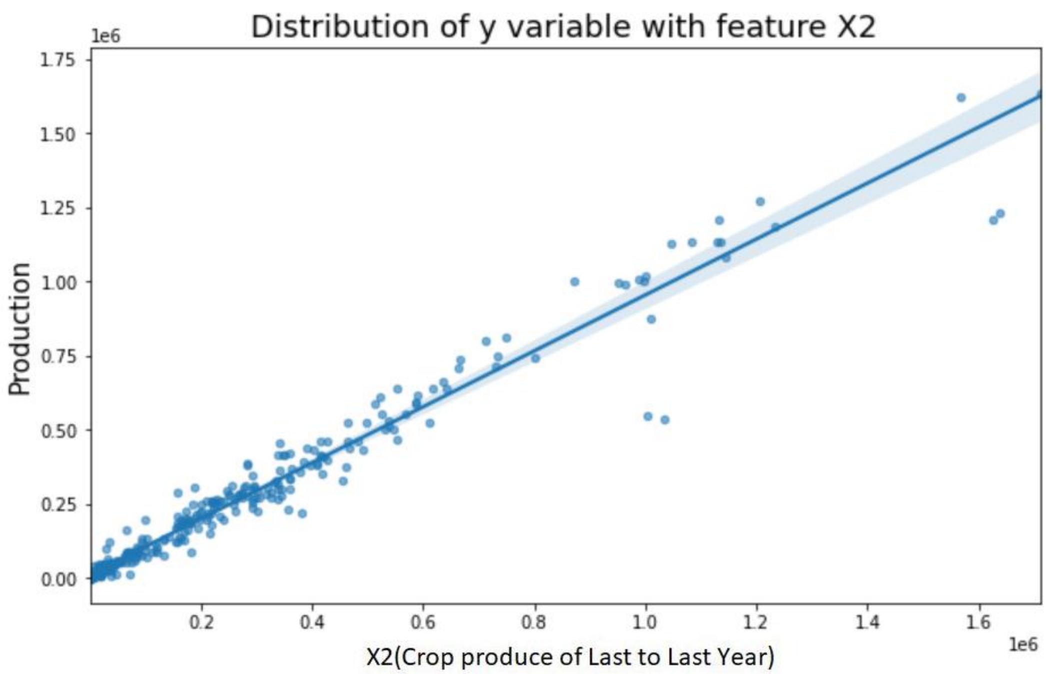


Fig 8: - Rice Production distribution graph



<u>Fig 9: - Last Year Production Graph</u>



<u>Fig 10: - Last to Last Year Production Graph</u>

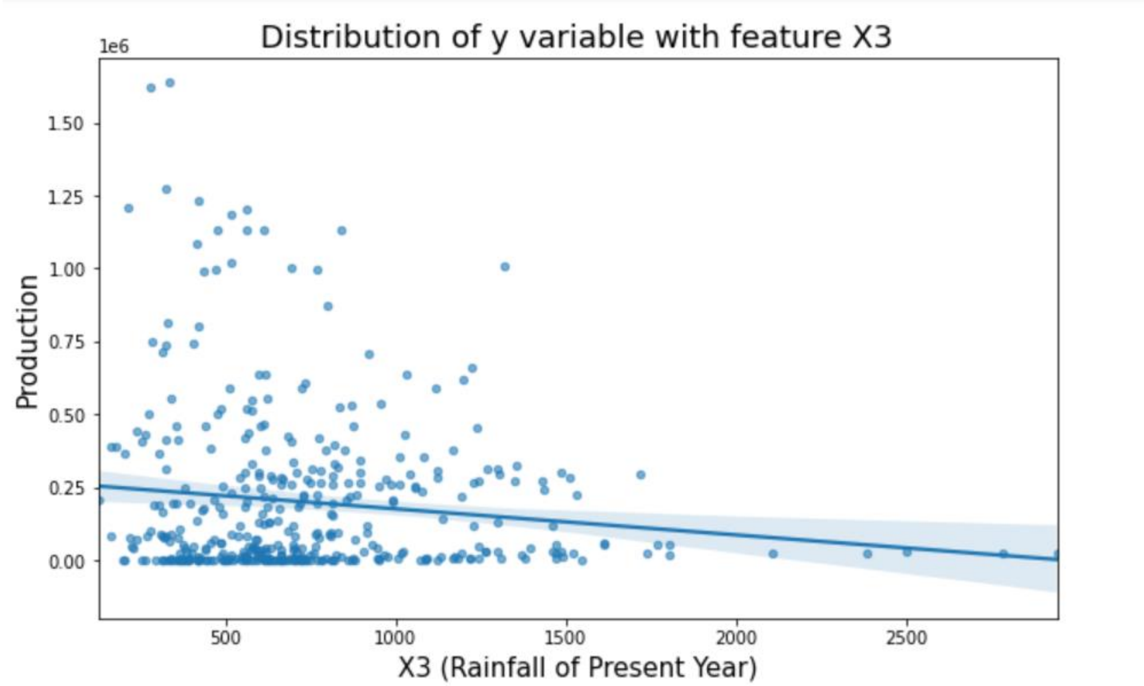


Fig 11: - Present Year Rainfall Graph

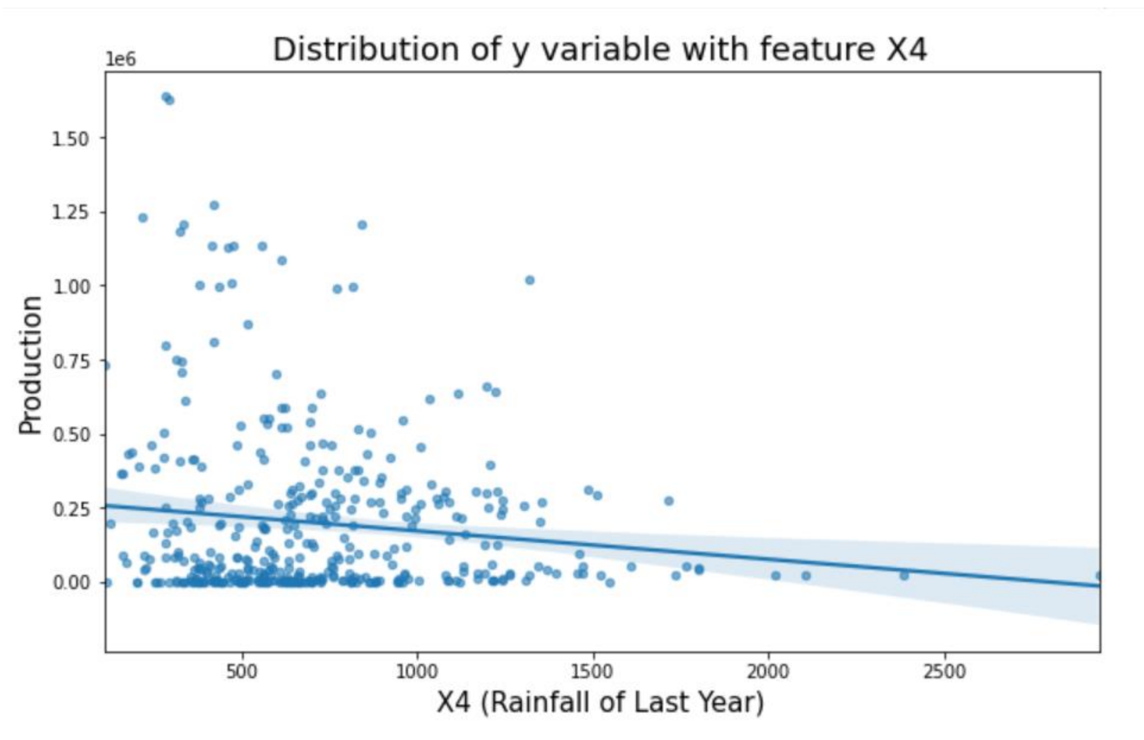


Fig 12: - Previous Year Rainfall Graph

Using Linear Regression to find RMSE

```
[ ] alg = LinearRegression()
    alg.fit(X_train, y_train)

    coef = alg.coef_
    coef = coef.round(decimals=2)
    np.set_printoptions(suppress=True) #gem
    print("The coefficients for the linear regression model learnt are\n")
    print(coef)
    print()

    y_predict = alg.predict(X_test)
    rmse = sqrt(mean_squared_error(y_predict, y_test))
    print("RMSE(LR):")
    print(rmse)
```

```
The coefficients for the linear regression model learnt are

[199526.7   87885.81   4104.33  -5125.82]

RMSE(LR):
71366.57352459784
```

Fig 13: - RMSE (Root mean squared error) value using Linear Regression

Haryana State Results.

```
[ ] print(p)
```

```
    y_predicted    y_test
0           7.5     7.351
1          56.7    57.854
2          -1.3     0.005
3         297.3   328.560
4         271.8   302.339
..          ...       ...
75       1612.8  1207.000
76        270.0   296.072
77          8.8     1.135
78          7.5     0.479
79        235.6   248.000

[80 rows x 2 columns]
```

```
[ ] #root mean square error.
    rmse/1000
```

```
71.36657352459784
```

Fig 14: - Haryana State Prediction Results

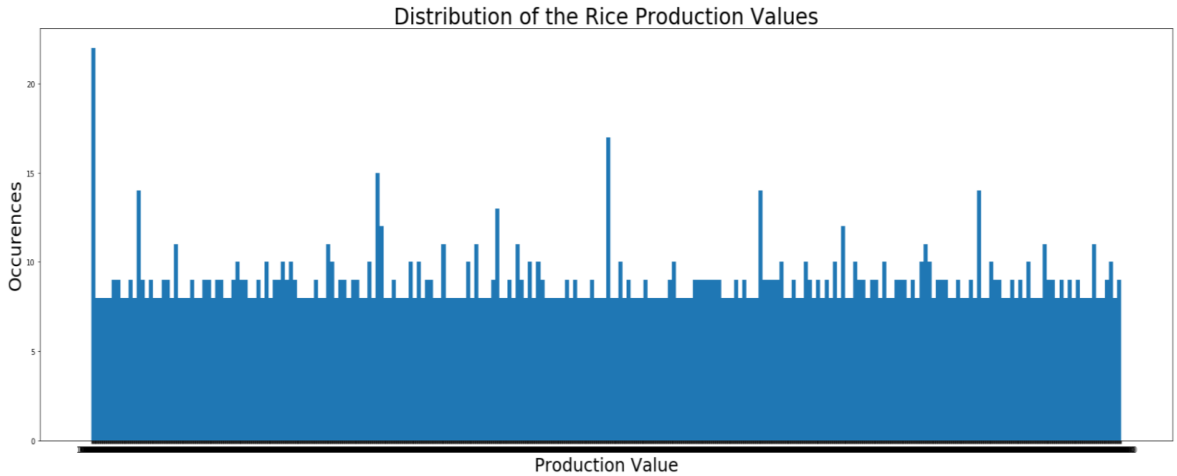


Fig 15: - Graph of (production of rice value) vs (no of districts that have the same value)

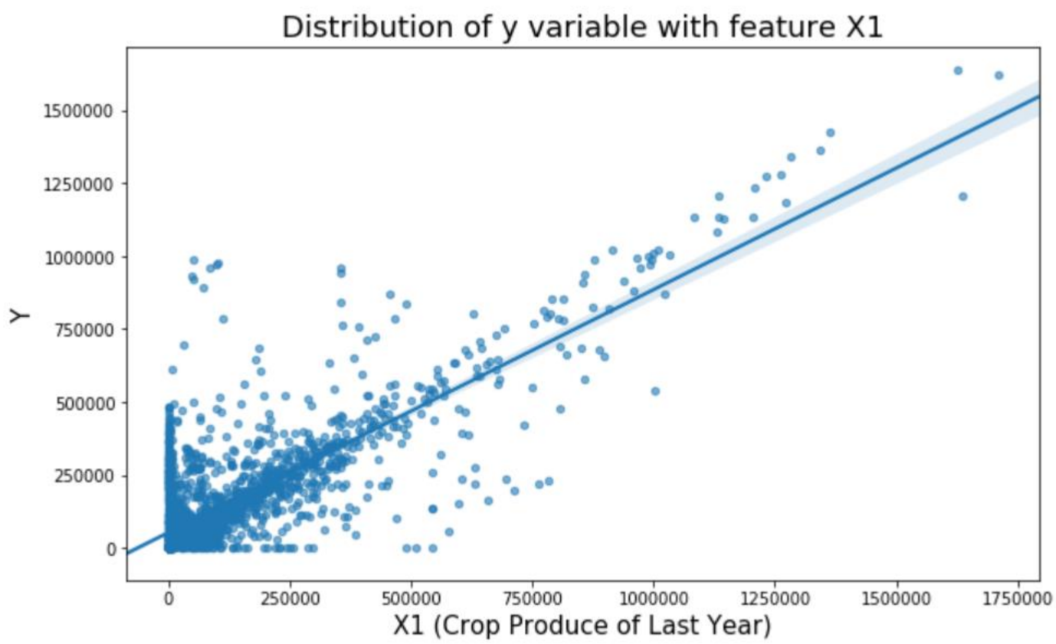


Fig 16: - Graph of production previous vs past year

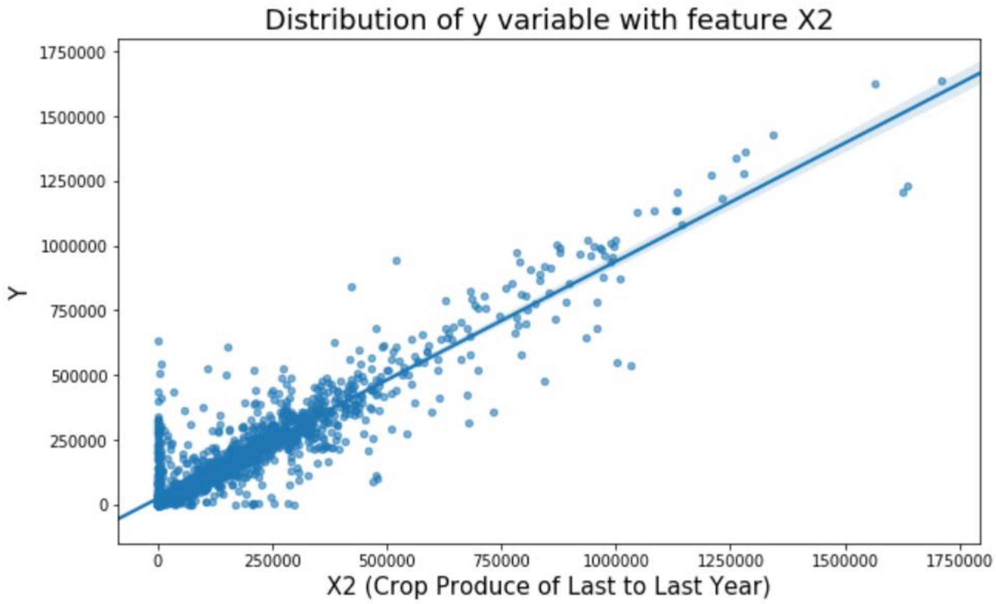


Fig 17: - Graph of production vs rainfall

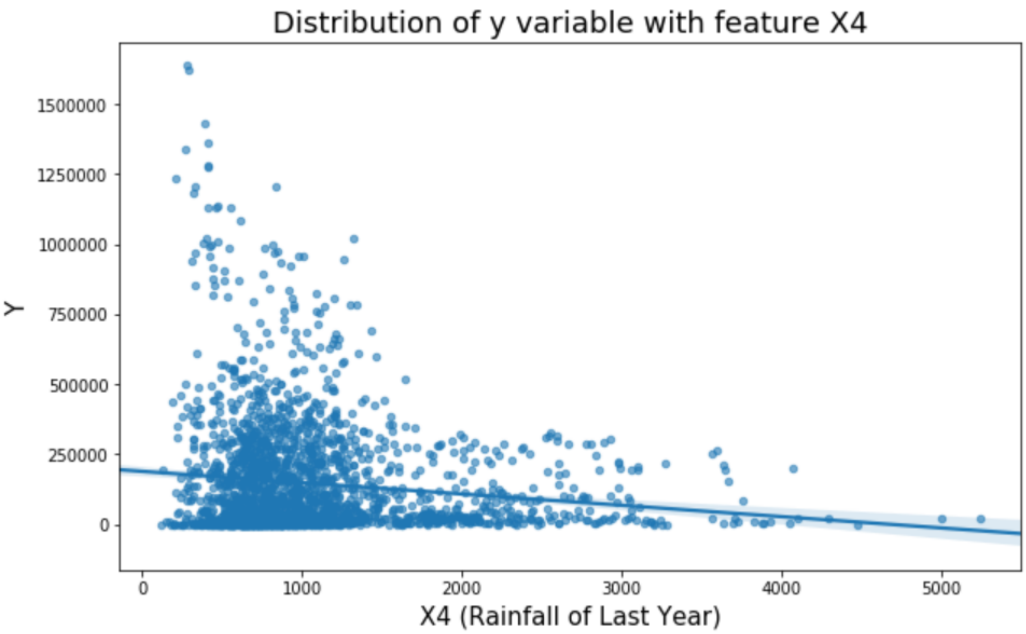


Fig 18: - Graph of production vs past rainfall

Lets calculate the average RMSE (Cross Validation, 5-Fold)

```
[61]
clf = LinearRegression()
scores = cross_val_score(clf, X, y, cv=5, scoring='neg_mean_squared_error')
```

```
[62]
for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])
```

```
[63]
print(scores)
avg_rmse = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[ 90696.68166565  61035.55459524  51367.37486568  95344.81806082
  74890.37880977]


Avg RMSE is  74666.9615994
```

Fig 19: - RMSE value using Linear Regression

```
[66]
p = pd.DataFrame(data=yo,columns=['Predicted','Actual'])
p.describe()
```

	Predicted	Actual
count	442.000000	442.000000
mean	138.243213	143.045928
std	191.785703	213.756353
min	2.600000	0.000000
25%	27.325000	6.200000
50%	62.950000	59.000000
75%	174.400000	191.175000
max	1255.300000	1362.000000

Fig 20: - Haryana state prediction table (Description)

Result

This time the predictions are not as good as we got for just one state.

```
[67]
p
```

	Predicted	Actual
0	28.1	107.0
1	100.0	90.0
2	127.2	123.3
3	83.7	112.5
4	46.6	23.2
5	17.9	0.8
6	12.8	0.3
7	64.3	59.0
8	66.3	3.2
9	157.0	214.1
10	33.6	76.2
11	281.2	413.1
12	166.1	215.6

Fig 21: - Haryana State Prediction Table

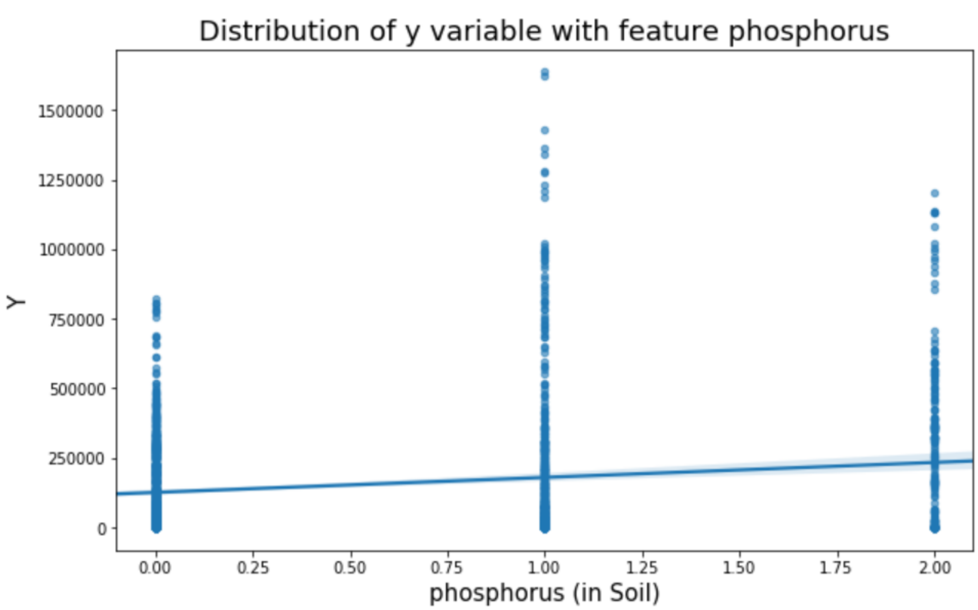


Fig 22: - Graph production vs Phosphorous in soil

```
[93]
p = pd.DataFrame(data=yo,columns=['Predicted','Actual'])
p.describe()
```

	Predicted	Actual
count	387.000000	387.000000
mean	149.640052	142.889406
std	201.643917	189.788422
min	-5.600000	0.000000
25%	14.600000	9.650000
50%	72.600000	71.300000
75%	227.450000	221.750000
max	1630.500000	1637.000000

Fig 23: - Haryana State Prediction with phosphorous data

Avg RMSE

```
[117]
clf = linear_model.Ridge(alpha = 7.1)
scores = cross_val_score(clf, X_old, y, cv=5, scoring='neg_mean_squared_error')

for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])

print(scores)
avg_rmse_ridge = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[ 76134.52790453  27507.68332601  61457.10987059  54957.99018935
  39147.52019336]


Avg RMSE is  51840.9662968
```

Avg RMSE with Linear Regression : 51874.9

Avg RMSE with Ridge Regression : 51840.9

So Ridge is slightly better than Linear

Fig 24: - Avg RSME value using Ridge Regression and comparison with LR

Avg RMSE

[121]

```
clf = linear_model.Lasso(alpha = 0.01)
scores = cross_val_score(clf, X_old, y, cv=5, scoring='neg_mean_squared_error')

for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])

print(scores)
avg_rmse_las = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[ 76204.74665396  27581.32715608  61253.46437277  54898.96269897
  39436.18029115]


Avg RMSE is  51874.9362346
```

Avg RMSE with Linear Regression : 51874.9

Avg RMSE with Ridge Regression : 51840.9

Avg RMSE with Lasso Regression : 51874.9

Fig 25: - Avg RSME value using Lasso Regression and comparison with LR and RR

First checking the avg RMSE for Linear Regression

[9]

```
clf = LinearRegression()
scores = cross_val_score(clf, X_, y, cv=5, scoring='neg_mean_squared_error')
for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])

print(scores)
avg_rmse = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[ 1030.92314374  1109.37929379   972.36266895  1487.52744177   491.48595541]


Avg RMSE is  1018.33570073
```

Fig 26: - Finding average RSME value for India using Linear Regression

```
[13] ▷ M↓
# 5 Fold CV, to calculate avg RMSE
clf = SVR(C=1000.0, epsilon=0.1, kernel='rbf', gamma=0.027)
scores = cross_val_score(clf, X_old, y.values.ravel(), cv=5, scoring='neg_mean_squared_error')
for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])

print(scores)
avg_rmse = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[  903.93008696   753.88394413   765.69751566  1574.251674     636.95214188]


Avg RMSE is  926.943072526
```

SVR : 927

LR : 1018

SVR (RBF kernel) works better than Linear Regression.

<u>Fig 27: - Finding average RSME value for India using Support Vector Regression (RBF Kernel)</u>

Degree 3 Polynomial

```
[16] ▷ M↓
# 5 Fold CV, to calculate avg RMSE
clf = SVR(kernel='poly', gamma='auto', degree=3, coef0=2)
scores = cross_val_score(clf, X_old, y.values.ravel(), cv=5, scoring='neg_mean_squared_error')
for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])

print(scores)
avg_rmse = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[  906.20976415   837.77643762  1049.76326739  1568.88777167   504.49443066]


Avg RMSE is  973.426334297
```

Polynomial Kernel also does better than Linear Regression

<u>Fig 28: - Finding average RSME value for India using Support Vector Regression (Degree 3 polynomial)</u>

```
Degree 4 Polynomial
```

```
[17]
# 5 Fold CV, to calculate avg RMSE
clf = SVR(kernel='poly', gamma='auto', degree=4, coef0=2)
scores = cross_val_score(clf, X_old, y.values.ravel(), cv=5, scoring='neg_mean_squared_error')
for i in range(0,5):
    scores[i] = sqrt(-1*scores[i])

print(scores)
avg_rmse = scores.mean()
print("\n\nAvg RMSE is ",scores.mean())
```

```
[  907.10874357   787.20784909   848.64917648  1570.06140194   557.83575489]


Avg RMSE is  934.172585194
```

Fig 29: - Finding average RSME value for India using Support Vector Regression (Degree 4 polynomial)

4.4 Use Case – Crop Yield Prediction System

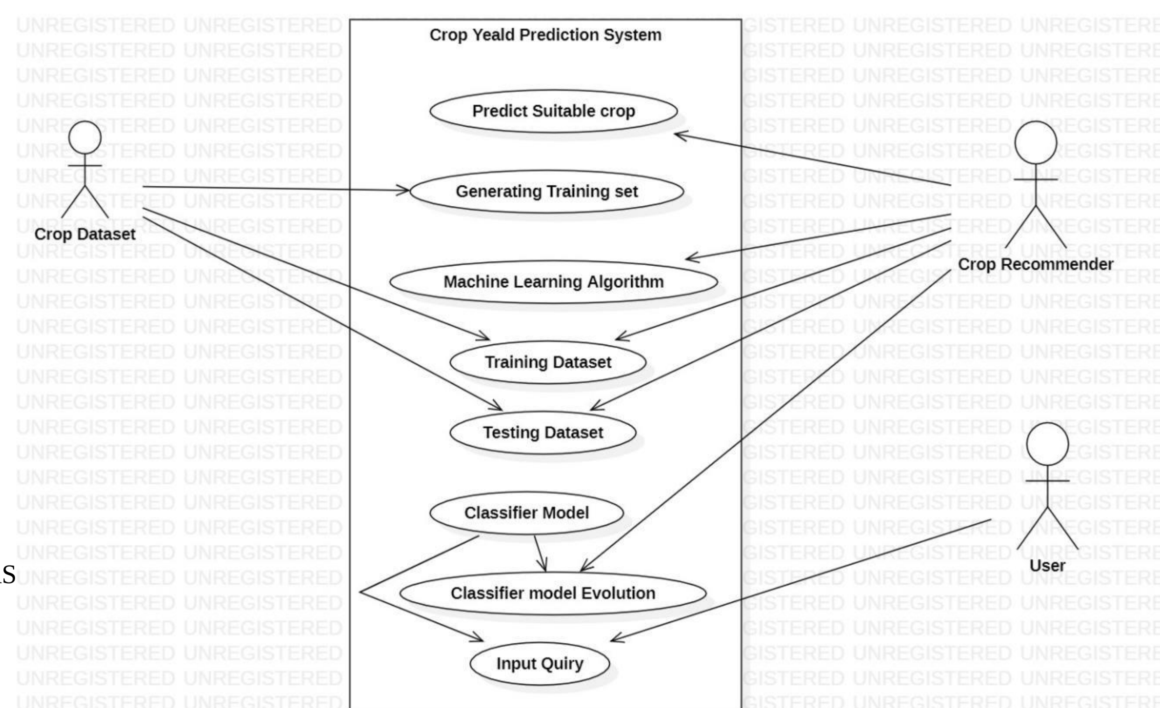


Fig 30: - Use Case Diagram

Chapter 5 - Non-Functional Requirements

This software provides various other requirements that define the productivity and utility of it. Some of which are mentioned below for detailed understanding.

5.1 Performance

This software works using the databases provided. Thus, the performance of the model depends on the size of the database provided. It also depends on the system it is being run on. A system with better hardware can process the data faster than the one with outdated hardware.

5.2 Reliability

As to start with a fully functional software one important feature of it is its error-free property. We begin by comparing the accuracy of the various approaches in predicting yields in years using different algorithms. It can be desirable to determine which variables and group of variables contribute most to predictive skills.

5.3 Availability

This software will be helpful for the economy of an agricultural-based place. It will help farmers from facing loss due to damaged crops or lower productivity. Thus, this software is meant for the farmers and government. They can grant access to the particular information they need for deciding the right choice of crop for the right time.

5.4 Security

This project does not have any credentials related to either public or government information. It predicts the result based on the dataset which is publicly available by the government.

5.5 Design Constraints

Specify design constraints imposed by other standards, company policies, hardware limitations, etc. that will impact this software project.

5.6 Logical Database Requirements

The following project is based on Supervised Learning. So, Database holds a very important use. All the testing and training data are derived from the database provided. The following are the uses of the database in our project:

-It stores all the information like the production per year values of both India as well as its states, Rainfall per year values for all the states and soil phosphorous contents.

-We need this information almost in every stage of the process of the data for prediction.

-All the data are can be downloaded for the government sites and various Analysis Models.

Chapter 6 – Analysis Models

6.1 Activity Diagram

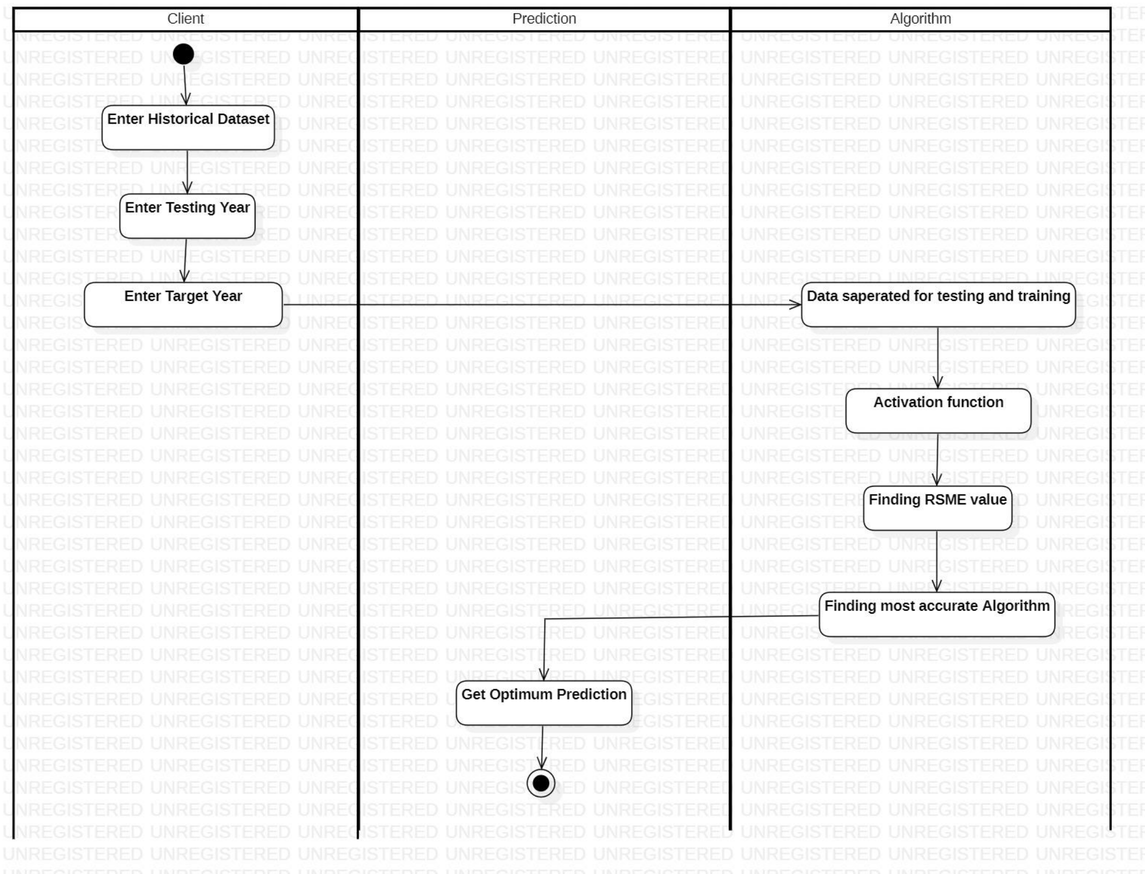


Fig 31: - Activity Diagram

6.2 Data Flow Diagram

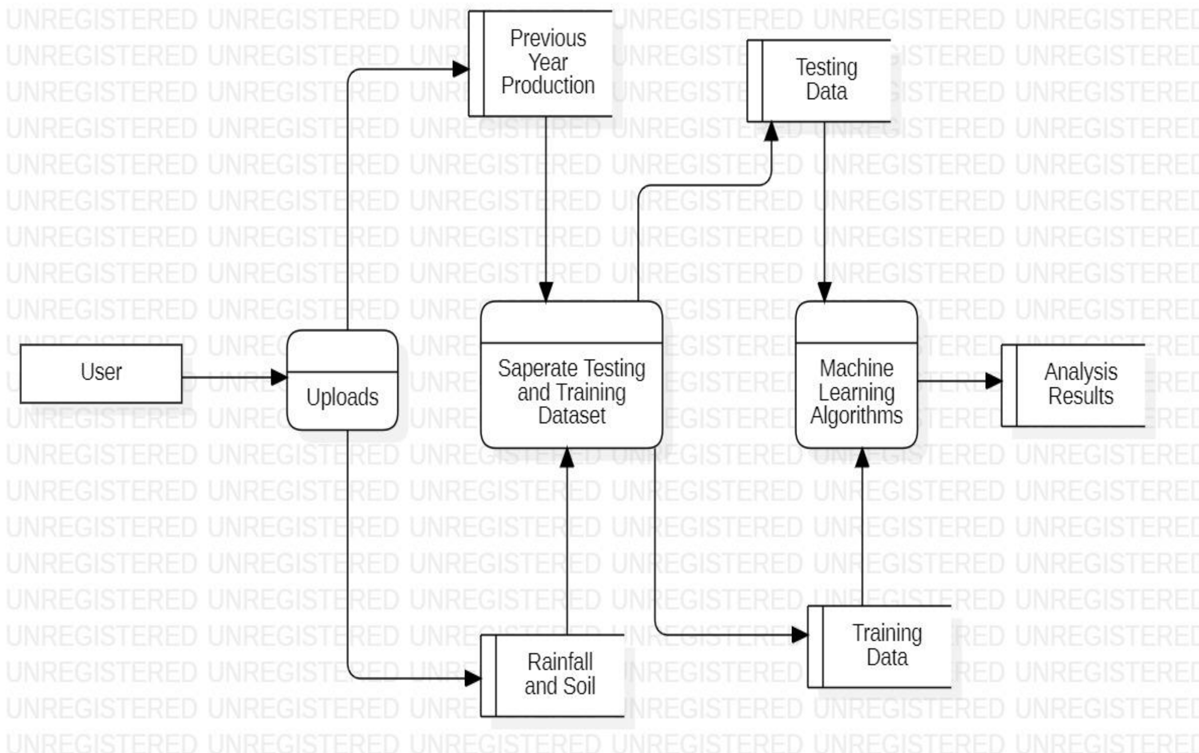


Fig 32: - Data Flow Diagram

Chapter 7 - Change Management Process

Change Management is the creation and implementation of the roles, processes, and tools that each of these groups uses to effectively and efficiently manage the people side of change. Project team, Sr. leader, managers, supervisors and employees are involved in managing change. There are usually three phases of change in change management. Namely Prepare, Implement and Reinforce.

All the changes would be made by the developer team if needed as per the request of the customer. The requests can be made through either via email or through other officials and the changes or features are made or added as per the requirement.

Chapter 8 - Appendices

8.1 Appendix 1 – Textual Index

8.2 Appendix 2 – Graphical Index