1. INTRODUCTION

The biggest problem with Indian farmers is that they are unaware of outcomes before selecting which crop to grow. They generally grow that product which was marketed at a great price last year.

For example, when onion prices skyrocketed in 2010-11, the land under onion cultivation doubled over the next five years, resulting in the problem of plenty. When prices crashed for three consecutive years, the government offered an MSP for onions and acquired about one million quintals. But the government did not know how to store onions, therefore entire consignment rotted away in damp warehouses. Similarly, many of the farmers decide crop to be grown without any idea about demand in the coming year.

Our project deals with this inability of farmer to decide the crop he/she needs to produce. It helps them by providing insights about the future demand based on present scenario so that they don't end up in losses.

We'll use the past trends in yield of crops and the area under cultivation to arrive at the decision of what should be the approach for the coming year. The idea requires knowledge of data mining to derive patterns and other useful information that helps us in predicting what amount of yield would produce high profits to the farmers, customers and Government.

2. PROJECT OVERVIEW

Our project main aim is to solve the surplus production problem. Many farmers go for a crop that obtained more profit in previous year, as most farmers opt this the method, production of that crop increase that particular year which results in lower selling price. This results in loss to farmer or vice versa scenario where the production is quite less which makes to import to meet the demand which results in higher selling price. This is burden for government as well as buyer.

Our project provides solution to this problem by predicting that particular year production based on previous year's production and warns the government if the production of that particular year is going to exceed based on the cultivation area information collected from farmer at the time, he buys seeds.

The project could only be successful if data is collected time to time, maintained and available to all over the country through a common database

The common challenges that one could face in this project would be collection of data from various sources, places and integrating them, maintaining them and it may also result in serious problems if data was not collected properly. Quality of seeds, seed rate at sowing and some other factors also affect the production value.

EXISTING SYSTEM:

Up to date there are only projects that forecast demand of a crop and give suggestions on which crop to grow based on some conditions like area, soil and rainfall etc. But they don't stop the excess production that may generate.

The existing system gives a brief idea to the farmer on which crop to grow for better profits or based on demand, but they are not aware of the problems they may face in future. Existing system calculates demand on quality of seeds and seed rate at sowing etc.., but they did not consider previous year's production at all.

PROPOSED SYSTEM:

Our project too opted some features from existing system like demand forecast but our demand forecast is different and based only on previous year's production not on soil or quality of seeds etc. our project deals in warning the system by which we can stop the surplus production.

Our project calculates production value of a particular year and maintains data like land area under cultivation and predict production value from that land area and if the production value is exceeding the predicted production value of that year it warns the government not to sell the seeds of that crop at it may result in surplus production.

This helps both government as well as farmer not to end up in losses. This helps a country's economy as agriculture sector plays a major role in GDP growth. Countries like India whose major GDP growth factor is agriculture, it is much important. All the developed countries now are once agricultural based countries.

3. REQUIREMENT ENGINEERING

3.1 SOFTWARE REQUIREMENTS:

Our project requires various software and its components.

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Python

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Machine Learning Packages in built in Python

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.

SciPy

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering

Pandas

Python Data Analysis Library pandas are an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

4. LITERATURE SURVEY

What is data science?

- Data science is a multidisciplinary blend of data inference, algorithm development, and technology in order to solve analytically complex problems.

- This aspect of data science is all about uncovering findings from data. Diving in at a granular level to mine and understand complex behaviours, trends, and inferences. It's about surfacing hidden insight that can help enable companies to make smarter business decisions. For example:

- Netflix data mines movie viewing patterns to understand what drives user interest and uses that to make decisions on which Netflix original series to produce.
- Target identifies what are major customer segments within its base and the unique shopping behaviours within those segments, which helps to guide messaging to different market audiences.
- Proctor & Gamble utilizes time series models to more clearly understand future demand, which help plan for production levels more optimally.

What is Machine Learning (ML)?

Machine learning (ML) is a category of algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available.

- Machine learning is of two types

    - ✓ Supervised learning

    - ✓ Unsupervised learning

Supervised Learning

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output. Supervised learning problem further categorized into regression and classification problems.

Regression

In regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. To exemplify, given data about the size of houses on the real estate market, try to predict their price. Another example would be, given a picture of a person, we must predict their age or gender.

Classification

Classification, on the other hand, is finding the category of the input variable, or in more academic terms, mapping input variables into discrete categories. Ideal sentence to find a classification problem would be, whether this or that, like, yes or no, 0 or 1, true or false. For example, from the example of house price given above, if we change the output to “Sells for more or less than asking price,” then it is a classification problem. Another example is, given a patient with tumor, we must predict whether the tumor is malignant or benign.

Unsupervised Learning

On the contrary to Supervised learning, unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.

We can derive this structure by clustering the data based on relationships among the variables in the data. With Unsupervised learning there is no feedback based on the prediction results. For example, take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on. This is a good example of clustering. Whereas, for a non-clustering problem such as "Cocktail Party Problem", it helps in identifying voices music from a mesh of sounds at a cocktail party.

5. TECHNOLOGY

PYTHON TOOL:

Python is considered as one of the widest used machine learning tools. It has many other features that attract the data science community. Being a data science tool, Python helps to explore the concepts of machine learning in the best way possible. Machine Learning is all about probability, mathematical optimization, and statistics, which are all made easy by Python.

There are lot of inbuilt machine Learning packages which will make machine learning Handy with python. What drives developers to Python is that it is easy to learn and code. It promotes an easy-to-understand syntax especially when compared to other data science languages, such as R and thereby leads to a shorter learning curve.

The reason for growing success of Python is the availability of data science libraries for aspiring candidates. These libraries have been upgraded continuously. The constraints that developers faced a year ago are now treated successfully with Python.
Many libraries are available to perform data analysis, here's an important one to start with:

1. Numpy

Numpy is important to perform scientific computing with Python. It encompasses an assortment of high-level mathematical functions to operate on multi-dimensional arrays and matrices.

2. Sklearn

Scikit-learn 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.

3. Linear regression

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:

(1) Does a set of predictor variables do a good job in predicting an outcome (dependent) variable?

(2) Which variables are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable? These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula

y = c + b*x, where

y =estimated dependent variable score,

c = constant,
b = regression coefficient,
x = score on the independent variable.

There are several types of linear regression

- Simple linear regression
  1 dependent variable (interval or ratio), 1 independent variable
- Multiple linear regression
  1 dependent variable (interval or ratio), 2+ independent variables

- Logistic regression

  1 dependent variable (dichotomous), 2+ independent variable(s)
- Ordinal regression

  1 dependent variable (ordinal), 1+ independent variable(s)
- Multinomial regression

  1 dependent variable (nominal), 1+ independent variable(s)

4. Lasso regression

In statistics and machine learning, **lasso** (**least absolute shrinkage and selection operator**; also Lasso or LASSO) 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 was originally introduced in geophysics literature in 1986, and later independently rediscovered and popularized in 1996 by Robert Tibshirani, who coined the term and provided further insights into the observed performance.

Lasso was originally formulated for least squares models and this simple case reveals a substantial amount about the behavior of the estimator, including its relationship to ridge regression and best subset selection and the connections between lasso coefficient estimates and so-called soft thresholding. It also reveals that (like standard linear regression) the coefficient estimates need not be unique if covariates are collinear.

Though originally defined for least squares, lasso regularization is easily extended to a wide variety of statistical models including generalized linear models, generalized estimating equations, proportional hazards models, and M-estimators, in a straightforward fashion. Lasso's ability to perform subset selection relies on the form of the constraint and has a variety of interpretations including in terms of geometry, Bayesian statistics, and convex analysis.

The LASSO is closely related to basis pursuit denoising.

5. Ridge regression

Tikhonov regularization, named for Andrey Tikhonov, is the most commonly used method of regularization of ill-posed problems. In statistics, the method is known as ridge regression, in machine learning it is known as weight decay, and with multiple independent discoveries, it is also variously known as the Tikhonov–Miller method, the Phillips–Twomey method, the constrained linear inversion method, and the method of linear regularization. It is related to the Levenberg–Marquardt algorithm for non-linear least-squares problems. Suppose that for a known matrix and vector, we wish to find a vector such that the standard approach is ordinary least squares linear regression. However, if no satisfies the equation or more than one does—that is, the solution is not unique—the problem is said to be ill posed.

In such cases, ordinary least squares estimation leads to an over determined (over-fitted), or more often an underdetermined (under-fitted) system of equations. Most real-world phenomena have the effect of low-pass filters in the forward direction where maps to .Therefore, in solving the inverse-problem, the inverse mapping operates as a high-pass filter that has the undesirable tendency of amplifying noise (eigenvalues / singular values are largest in the reverse mapping where they were smallest in the forward mapping). In addition, ordinary least squares implicitly nullifies every element of the reconstructed version of that is in the null-space of rather than allowing for a model to be used as a prior for .Ordinary least squares seeks to minimize the sum of squared residuals, which can be compactly written as where is the Euclidean norm. In order to give preference to a particular solution with desirable properties, a regularization term can be included in this minimization: for some suitably chosen Tikhonov matrix. In many cases, this matrix is chosen as a multiple of the identity matrix (), giving preference to solutions with smaller norms; this is known as L_2 regularization. In other cases, high-pass operators (e.g., a difference operator or a weighted Fourier operator) may be used to enforce smoothness if the underlying vector is believed to be mostly continuous. This regularization improves the conditioning of the problem, thus enabling a direct numerical solution. An explicit solution, denoted by, is given by the effect of regularization may be varied by the scale of matrix. For this reduces to the regularized least-squares solution, provided that $(A^TA)^{-1}$ exists.

6. DESIGN ANALYSIS

Why Use UML?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. The Unified Modelling Language (UML) was designed to respond to these needs.

6.1 UML Diagrams

UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modeling tools include. In its simplest form, a use case can be described as a specific way of using the system from a User's (actor's) perspective. A more detailed description might characterize a use case as:

- a pattern of behavior the system exhibits
- a sequence of related transactions performed by an actor and the system
- delivering something of value to the actor

Use cases provide a means to:

- capture system requirements
- communicate with the end users and domain experts
- Test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use case collection specifies all the ways of using the system.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

 User Model View

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-user's perspective.

 Structural model view

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

 Behavioral Model View

- It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

 Implementation Model View

- In this the structural and behavioral as parts of the system are represented as they are to be built.

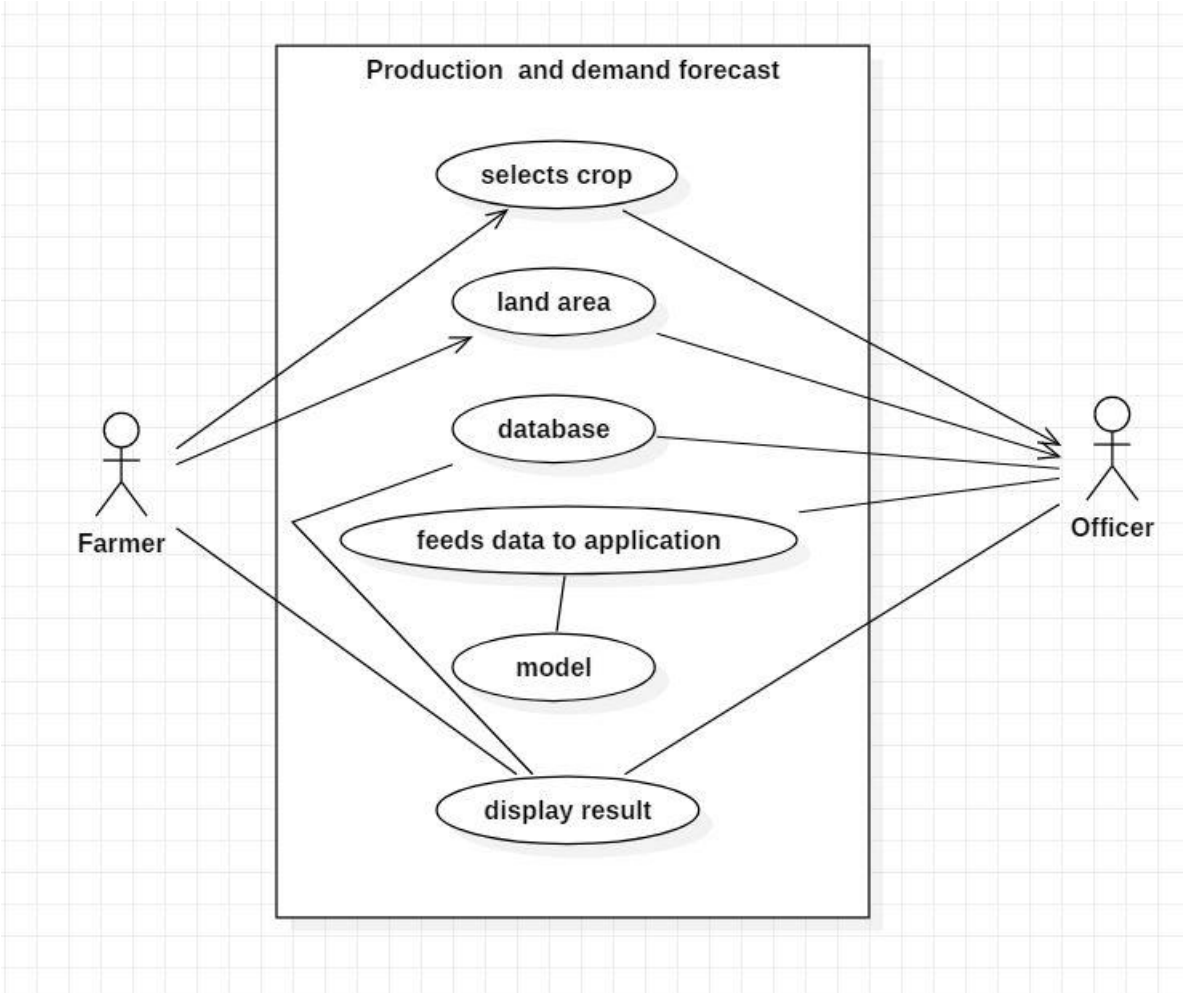
 Environmental Model View

- In this the structural and behavioral aspect of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral

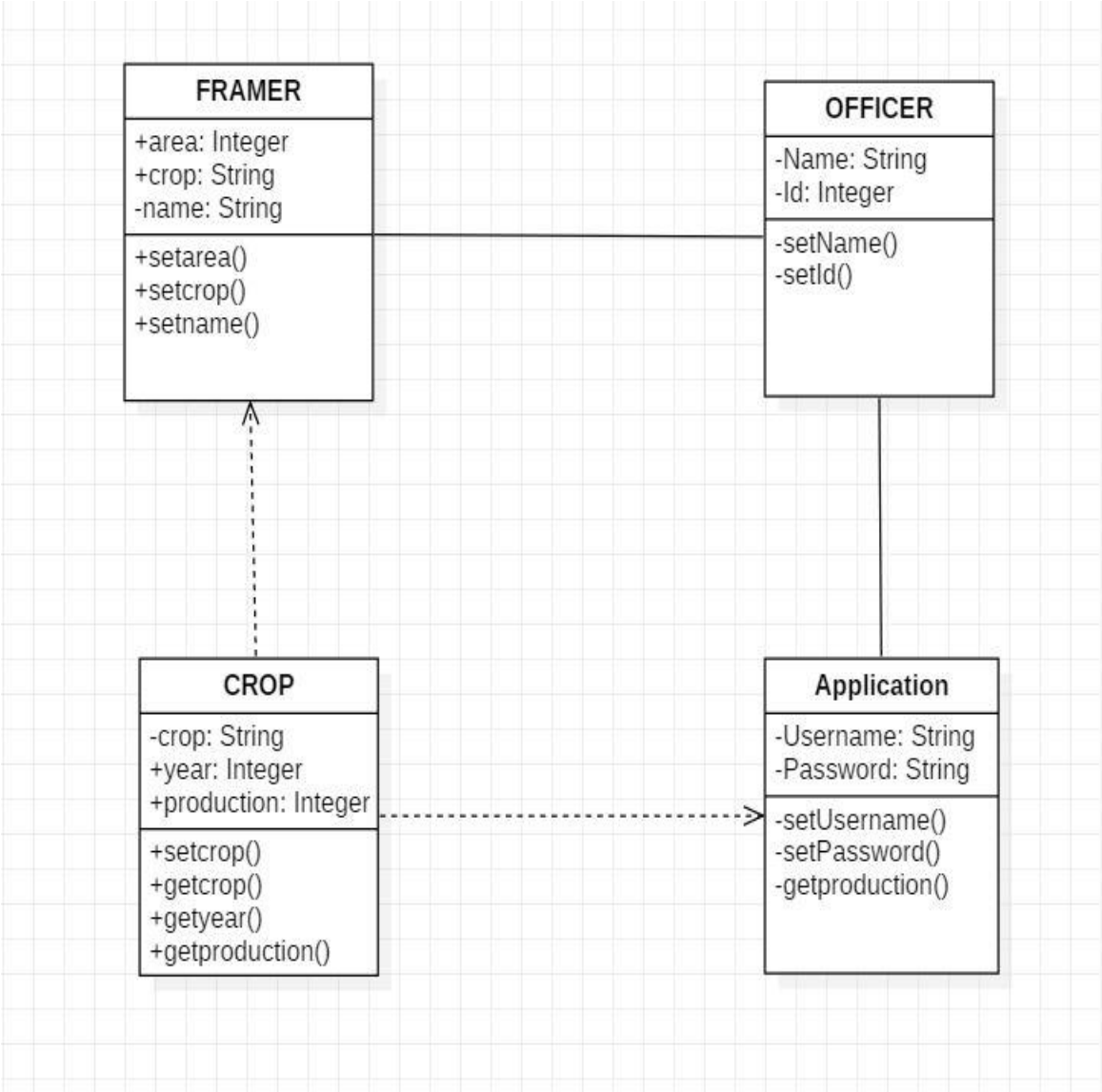
6.1.1 USECASE DIAGRAM



- The above use case diagram has two actors and 6 uses cases.
  Actors: - Farmer and Officer
  Use cases: - selects crop, land area, database, feeds data to application, Model, display result

- We used directed association and simple association relationships.

6.1.2 CLASS DIAGRAM

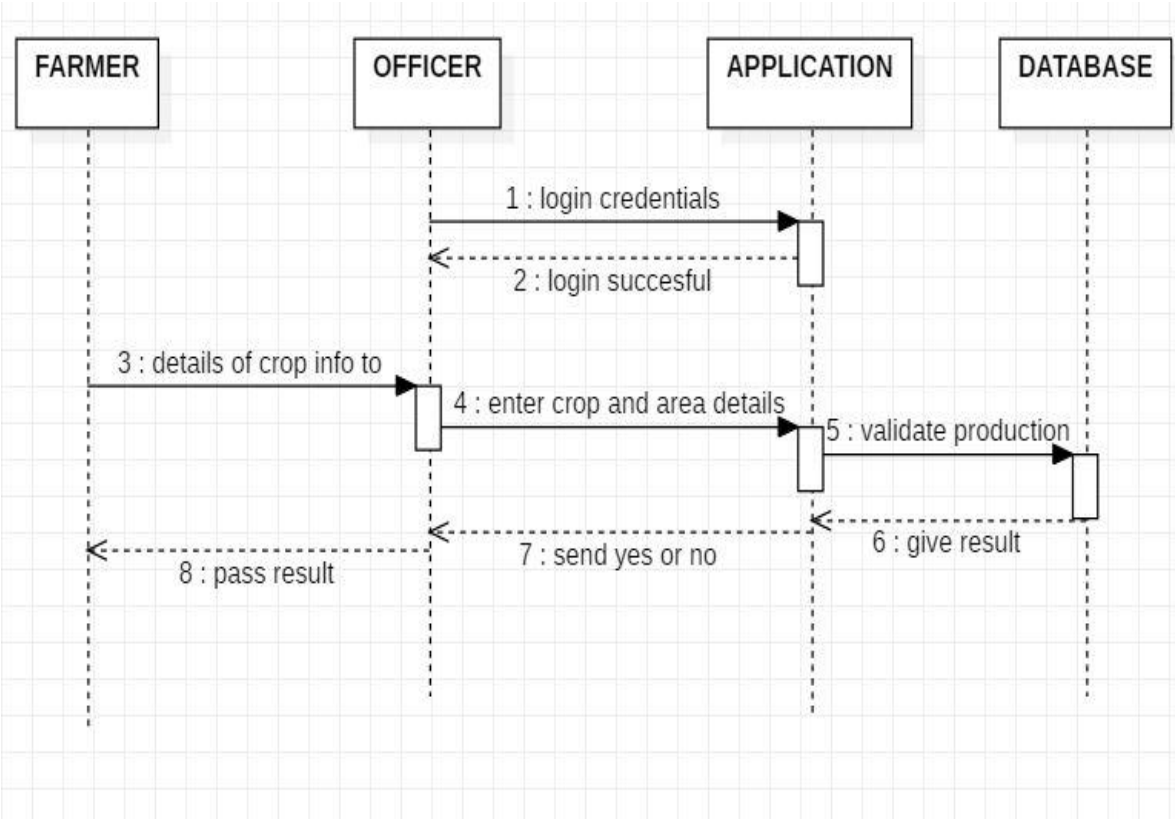


- The class diagram contains four classes.
  Class: -Farmer, Officer, Application and Crop

- We used association relationship between (1) Officer and farmer and (2) Application and officer.

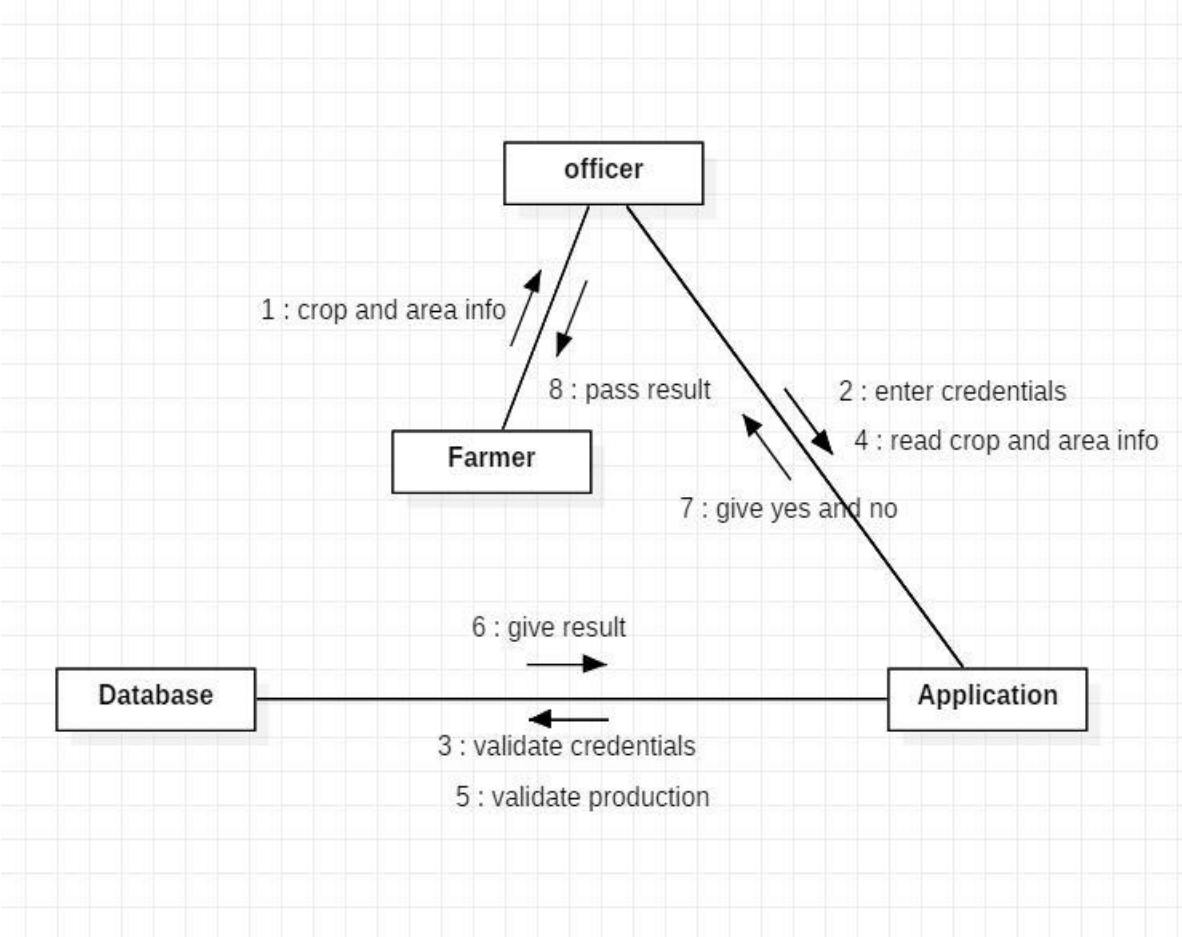
- We used dependency relationship between (1) Farmer and crop and (2) Application and crop

6.1.3 SEQUENCE DIAGRAM



- The sequence diagram has four objects and various interactions (messages) between them.
- We place the important objects from left to right.
- Activation boxes represents the time an object needs to complete a task

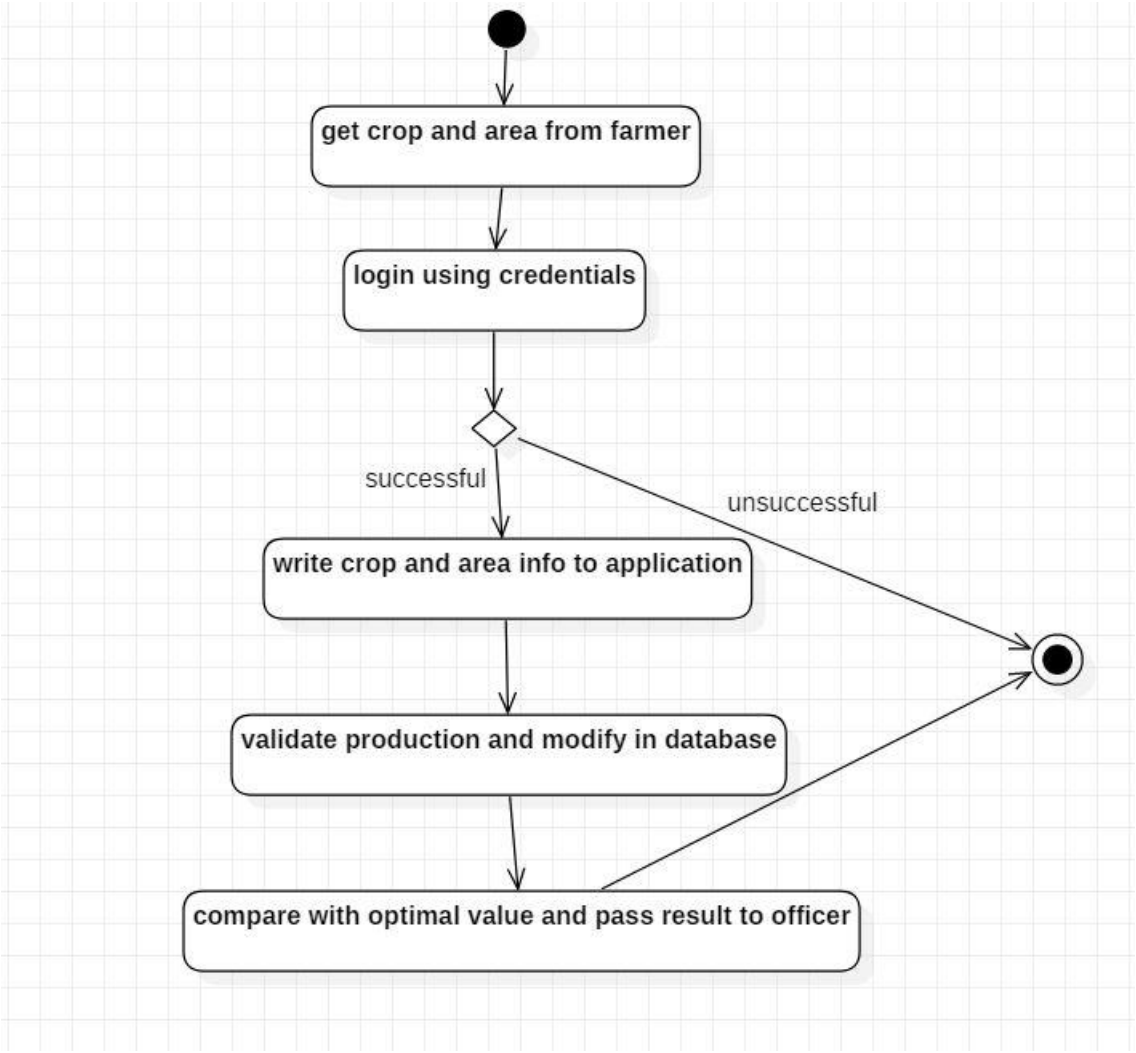
6.1.4 COLLABORATION DIAGRAM



- Collaboration diagram is also called as communication diagram or interaction diagram.

- It is an illustration of the relationships and interactions among software objects.

6.1.5 ACTIVITY DIAGRAM



- An activity diagram consists of activity states and action states, transactions, objects.

- It starts with initial state and ends with final state symbols.

7. IMPLEMENTATION

7.1 MODULES:

MODULE 1: COLLECTION OF DATA

In the first and starting module we collected various kinds of data like name of the state, cultivated agricultural land area (in million hectares), production (in million quintals) from different sources. We collected land area and production values of different states i.e. Chhattisgarh, Andhra Pradesh, Karnataka, Kerala, and Maharashtra of various years from 1997 to 2014. We got too many divided values for each particular year, but we integrated them and organized it year wise and state wise manner.

For example,

	Year	Rice	Area	Yield (Kg/Hect)
2	2017-18	112.91	43.79	2578
3	2016-17	109.698	43.99	2494
4	2015-16	104.41	43.49	2400
5	2014-15	105.48	43.86	2390
6	2013-14	106.65	43.95	2424
7	2012-13	105.24	42.75	2461
8	2011-12	105.3	44.01	2393
9	2010-11	95.98	42.86	2239
10	2009-10	89.09	41.92	2125
11	2008-09	99.18	45.54	2178
12	2007-08	96.69	43.91	2202
13	2006-07	93.36	43.81	2131
14	2005-06	91.79	43.66	2102
15	2004-05	83.13	41.91	1984
16	2003-04	88.53	42.59	2077
17	2002-03	71.82	41.18	1744
18	2001-02	93.34	44.9	2079
19	2000-01	84.98	44.71	1901
20	1999-00	89.68	45.16	1986
21	1998-99	86.08	44.8	1921
22	1997-98	82.54	43.45	1900
23	1996-97	81.73	43.43	1882
24	1995-96	76.98	42.84	1797
25	1994-95	81.81	42.81	1911
26	1993-94	80.3	42.54	1888
27	1992-93	72.86	41.78	1744

Rices

We observed variations in land area and production values of every year and for every crop and observed that some of the farmers shifted to other crop that got high selling price in previous year. We analyzed how they shifted and on what basis and considered factors whatever we want.

MODULE 2:

PREDICTING PRODUCTION VALUE ON PREVOUS YEAR'S CULTIVATED AREA AND ITS RESPECTIVE PRODUCTION

In the second module, we used Jupyter notebook software and python language for prediction. We used various machine learning algorithms for prediction and in this module we did all our coding part. We trained and validated the data and then tested data in this module using various algorithms like lasso regression, linear regression and ridge regression. We got fruitful results and much accuracy.

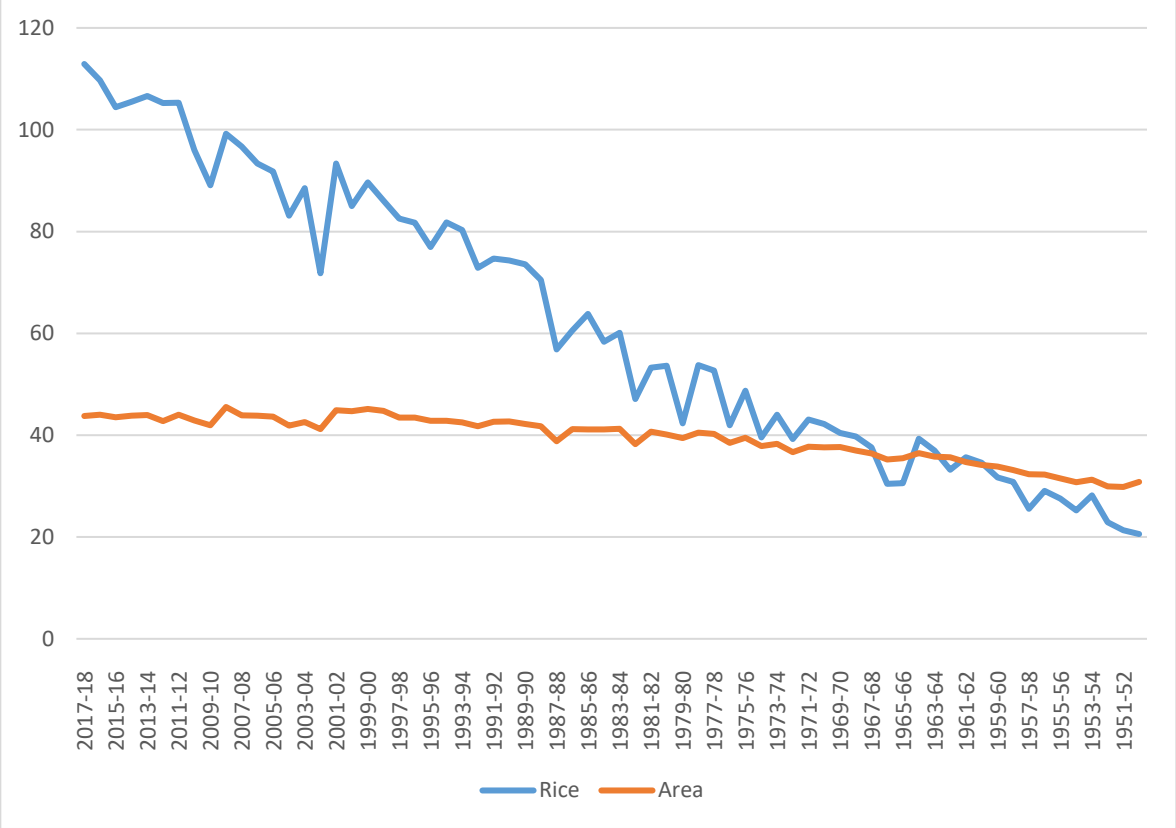
7.2 SOURCE CODE

```
import pandas as pd
from sklearn import preprocessing
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso, Ridge
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score, confusion_matrix

df=pd.read_csv('C:\\Users\\vinay\\OneDrive\\Desktop\\Groundnut.csv')

le = preprocessing.LabelEncoder()
le.fit(df['Year'])
k=le.transform(df['Year'])
df['Year']=k
#del df['Year']

X = np.array(df.drop(['Groundnut'], 1))
y = np.array(df['Groundnut'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

lreg = Lasso(max_iter=10000)
rreg = Ridge()
parameters = {'alpha':[-15, -10, -8, -4, -3, -2, 1, 5, 10, 20]}
```

```
for i in range(5):
    lclf = Lasso(alpha = i)
    rclf = Ridge(alpha = i)

    print(i)

    lclf.fit(X_train, y_train)
    y_pred = lclf.predict(X_test)

    #print('Coefficients: \n', lclf.coef_)
    print("Mean squared error: %.2f"% mean_squared_error(y_test, y_pred))
    print('Variance score: %.2f' % r2_score(y_test, y_pred))

    #print(df.std())

    ytrain_pred = lclf.predict(X_train)

    print("Mean squared error: %.2f"% mean_squared_error(y_train, ytrain_pred))
    print(lclf.score(X_test,y_test))

    print("\n")
    rclf.fit(X_train, y_train)
    y_pred = rclf.predict(X_test)

    #print('Coefficients: \n', rclf.coef_)
    print("Mean squared error: %.2f"% mean_squared_error(y_test, y_pred))
    print('Variance score: %.2f' % r2_score(y_test, y_pred))

    #print(df.std())

    ytrain_pred = rclf.predict(X_train)

    print("Mean squared error: %.2f"% mean_squared_error(y_train, ytrain_pred))
    print(rclf.score(X_test,y_test))
    print('')
```

8. RESULTS

RICE:



The results include the accuracy score, mean squared error and variance of the data. The following are the metrics or measures obtained from our algorithm

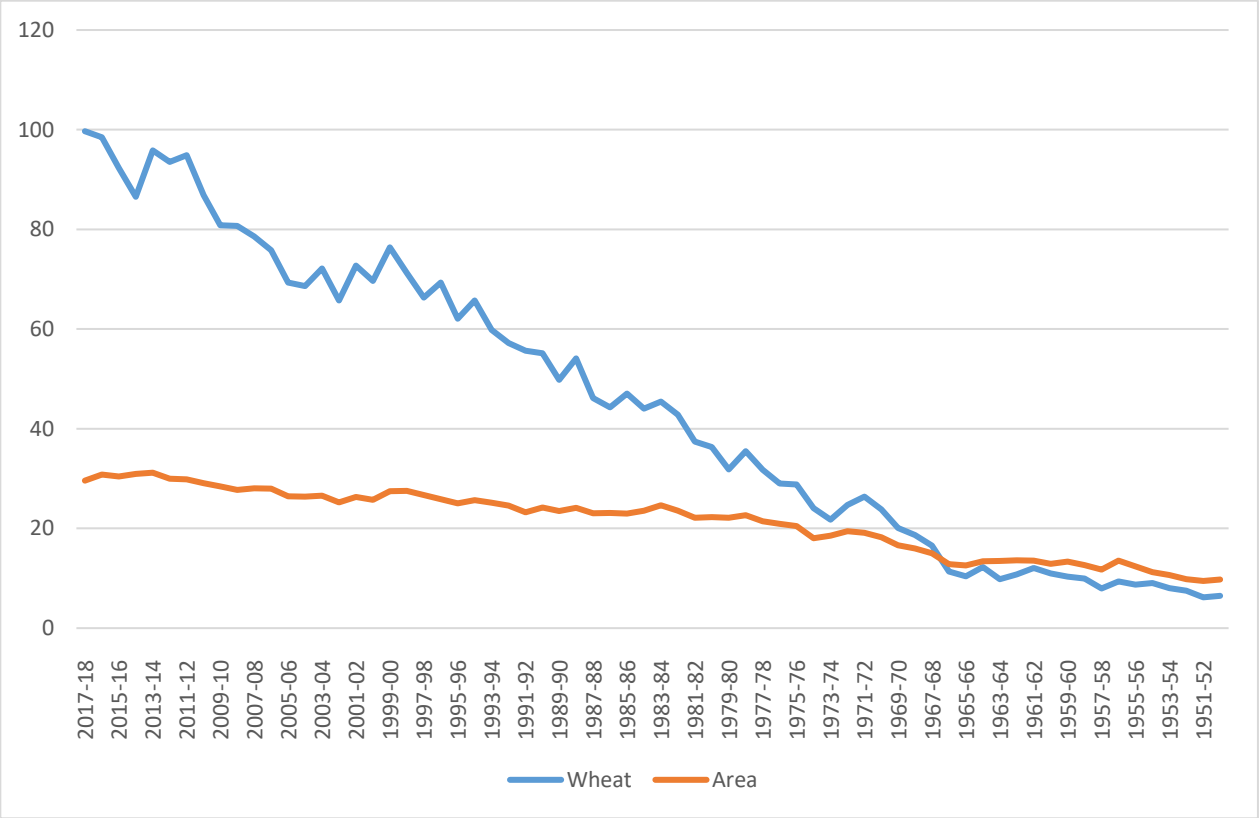
```
0
Mean squared error: 2.03
Variance score: 1.00
Mean squared error: 0.89
0.9977661171110016
```

```
1
Mean squared error: 1.79
Variance score: 1.00
Mean squared error: 1.22
0.9980291983377203
```

```
Mean squared error: 2.03
Variance score: 1.00
Mean squared error: 0.89
0.9977661171110016
```

```
Mean squared error: 2.02
Variance score: 1.00
Mean squared error: 0.89
0.9977790733428882
```

WHEAT:



The results include the accuracy score, mean squared error and variance of the data. The following are the metrics or measures obtained from our algorithm

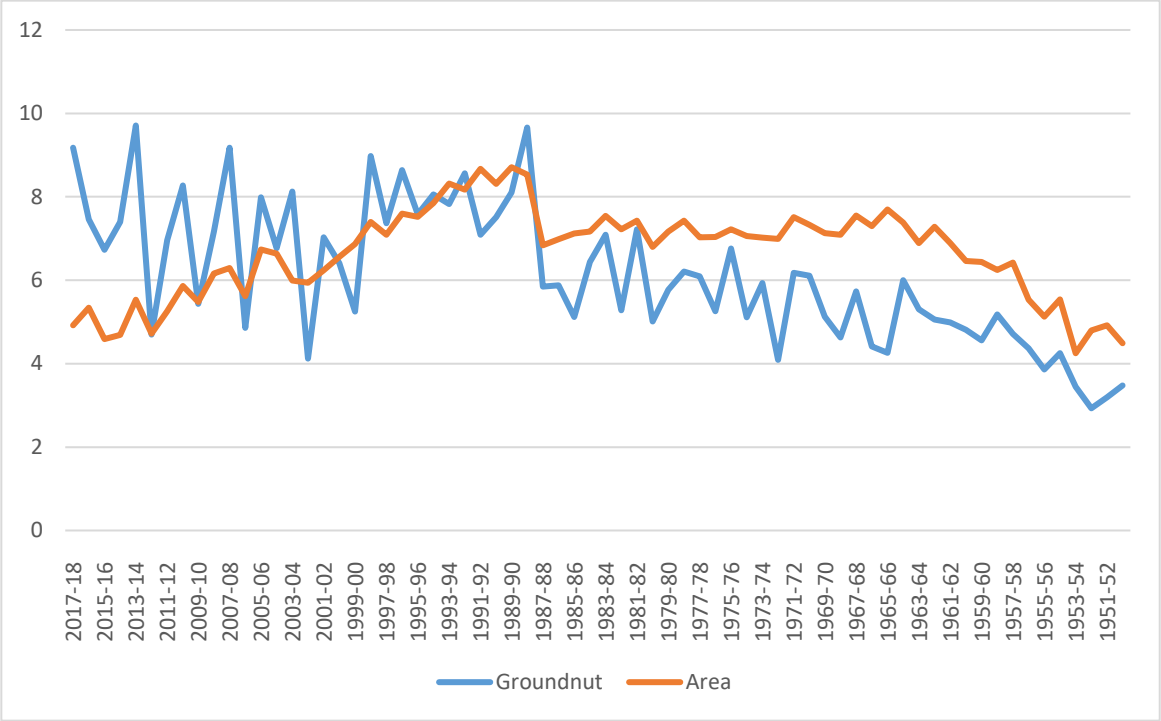
```
0
Mean squared error: 2.03
Variance score: 1.00
Mean squared error: 0.89
0.9977661171110016
```

```
1
Mean squared error: 8.75
Variance score: 0.99
Mean squared error: 9.91
0.9902112486704387
```

```
Mean squared error: 2.03
Variance score: 1.00
Mean squared error: 0.89
0.9977661171110016
```

```
Mean squared error: 8.85
Variance score: 0.99
Mean squared error: 9.71
0.9900981647058618
```

GROUNDNUT:



The results include the accuracy score, mean squared error and variance of the data. The following are the metrics or measures obtained from our algorithm

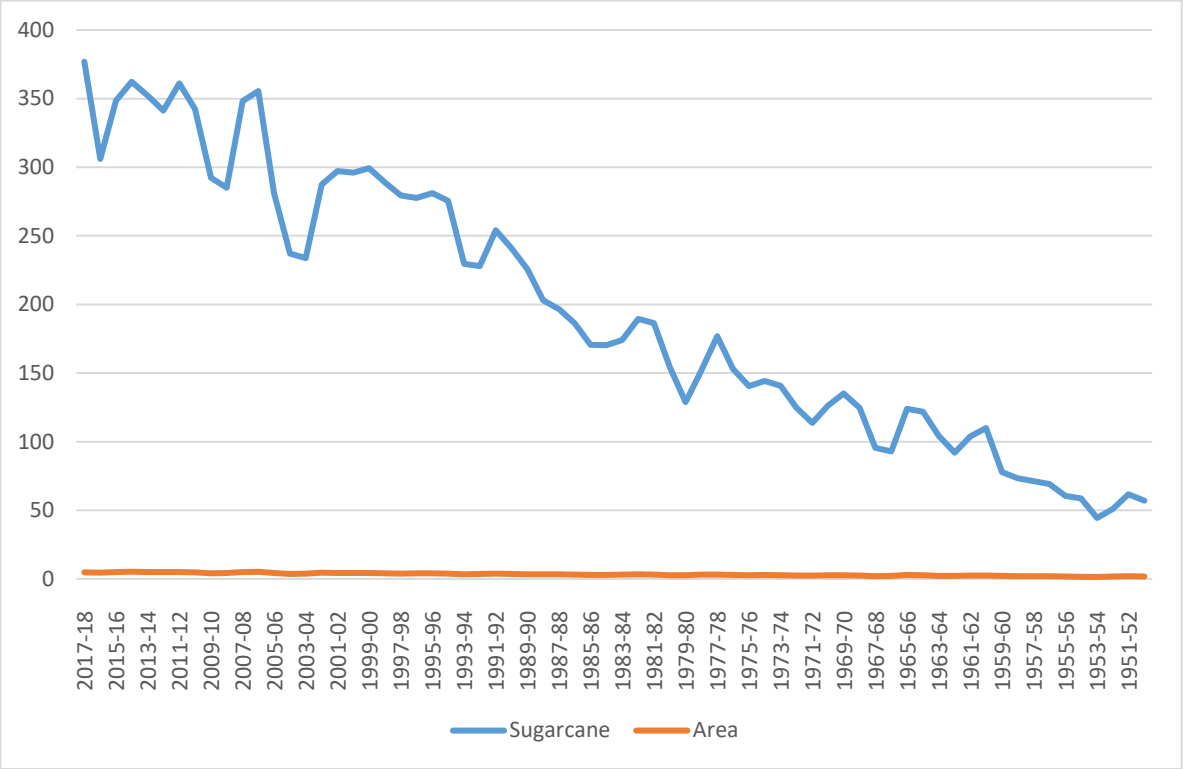
```
0
Mean squared error: 0.04
Variance score: 0.99
Mean squared error: 0.08
0.9876658112213991
```

```
1
Mean squared error: 1.29
Variance score: 0.57
Mean squared error: 0.80
0.5702969383851659
```

```
Mean squared error: 0.04
Variance score: 0.99
Mean squared error: 0.08
0.9876658112213988
```

```
Mean squared error: 0.04
Variance score: 0.99
Mean squared error: 0.08
0.9863970068137351
```

SUGARCANE:



The results include the accuracy score, mean squared error and variance of the data. The following are the metrics or measures obtained from our algorithm

```
0
Mean squared error: 158.22
Variance score: 0.98
Mean squared error: 66.91
0.9813751983920359
```

```
1
Mean squared error: 157.26
Variance score: 0.98
Mean squared error: 77.15
0.9814879236303958
```

```
Mean squared error: 158.22
Variance score: 0.98
Mean squared error: 66.91
0.9813751983920359
```

```
Mean squared error: 160.64
Variance score: 0.98
Mean squared error: 82.58
0.9810895578645042
```

9. CONCLUSION

Our project deals with this inability of farmer to decide the crop he/she needs to produce/shift. It helps them by providing insights about the future demand based on present scenario so that they don't end up in losses. Our project calculates production value of a particular year and when a production value calculated using real time data exceeds production value of that year it warns the government not to sell that particular crop seeds. It not only stops the surplus production but also helps government and farmer from not ending up in losses. Our project used knowledge of data mining to derive patterns and other useful information that helped us in predicting what amount of yield would produce high profits to the farmers, customers and Government.

As agricultural sector plays an important role in GDP which is used to measure the economic development of a country, our project would help in GDP growth to some extent and stops inflation which causes loss to value of money (price increases purchasing power decreases.

10. FUTURE ENHANCEMENTS

Our project predicts only previous year's production and area values but not on any other else. We can enhance this project and can increase its scope. we can add some features like soil, rainfall, quality of seeds, seed rate at sowing and some other features for better yield and we can also predict the crop he should if his selected crop's production is exceeded based on his soil, rainfall and climatic conditions. We can also create an algorithm to generate a price for a crop using above features.

11. BIBLIOGRAPHY

- ANALYSIS OF TRENDS IN INDIA'S AGRICULTURAL GROWTH
  http://www.environmentportal.in/files/file/Analysis%20of%20Trends%20in%20India%E2%80%99s%20Agricultural%20Growth.pdf

- AGRICULTURAL RESEARCH IN INDIA: AN EXPLORATORY STUDY
  http://www.indiaenvironmentportal.org.in/files/file/agricultural%20research.pdf

- LINEAR, LASSO AND RIDGE REGRESSION – MATH AND PYTHON IMPLEMENTATION
  - ✓ Math - https://medium.com/datadriveninvestor/the-math-behind-ridge-regularization-and-lasso-regularization-c4473332dbda
  - ✓ Python - https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b