TABLE OF CONTENTS

Sno.	*Description*	*Page No.*
1	Abstract	3
2	Introduction	4
3	Literature Review	5
4	Problem Identification and Objectives	6
5	System Methodology	7
6	Overview of Technologies	14
7	Implementation	16
	7.1 Coding	16
	7.2 Testing	26
8	Results and Discussions	29
9	Conclusions and Futurework	31
10	References	32

LIST OF FIGURES

Figure no.	*Description*	*Page No.*
1	Graph indicating months of high loss	4
2	System block diagram	7
3	Flowchart (Work flow)	8
4	Dataset	9
5	Confusion Matrix	10
6	Correlation Heatmap	14
7	Crop_Damaged Grouped Count	15
8	Estimated Insect Grouped Count	15
9	Soil Type Grouped Count	16
10	Season Grouped Count	16
11	Crop Vs Pesticide usage category	17
12	Distplot	17
13	Boxplot with outliers present	18
14	Boxplot after removing outliers	18
15	Skew Analysis	19
16	Results of Approach 1	29
17	Results of Approach 2	30
19	Results of Crop Model	30

1. ABSTRACT

The agriculture plays a dominant role in the growth of the country's economy. As we know, farmers are the backbone of our country but we genuinely don't appreciate their job as much. Pesticides are great until we use the correct dosage but if the amount of pesticides added were increased than the limit it will turn the entire harvest upside down. One financial year for a farmer is very crucial to accept the loss.
Luckily, we can determine the outcome of harvest season which means, whether the crop would be healthy, damaged by pesticides or damaged by any other reasons taking certain factors into consideration of labelled data. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. This project involves predicting outcome of the harvest from available historical available data like pesticide usage, soil parameter and historic crop yield. This project focus on predicting the outcome of the harvest and the type of the crop that has to been grown in that particular field in order to expect better yield based on the existing data by using different machine learning algorithms. The prediction will help to the farmer to predict the yield of the crop before cultivating onto the agriculture field. By machine learning techniques (including pre- processing of data, data visualisation and manipulation) we will try to build a model using different algorithm like Random forest, Decision tree, Artificial neural networks and logistic algorithms (and also some boosting algorithms if possible) and we will try to pick the best model that predicts the outcome of the harvest season and also that predicts which crop has to be cultivated by comparing the accuracy and performance of all these models of respective algorithm.

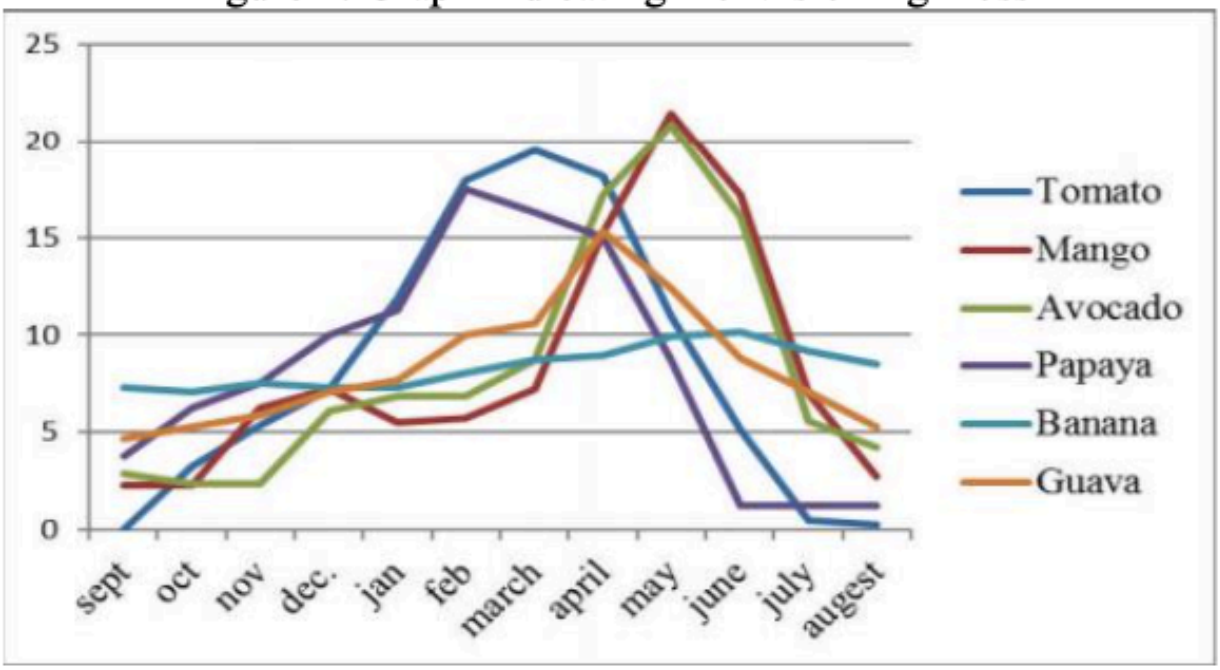
2. INTRODUCTION

Recently Machine Learning concepts evolved in every sector even in agriculture sector where they are making effort to help the farmers to make their harvest season go smooth and have a healthy plantation at the end of the season. Even it could help the farmers choose which plant to cultivate depending upon their soil fertility. Well, however they are many other certain factor where they are yet to be developed in the agriculture sector and also have to evolve technologies which might help even better than before for farmers in their irrigation.

Now, Pesticides are really good for crop while harvest period only when they are limited to certain measured amounts that required by the whole harvest season. If this is not the case then farmers would end up with the damage of whole crop in the farm and they would be falling into lots of debts and go through the tough times. But this would not be the situation if we predict the status of the crop in the early stages of harvesting so that farmer can take actions accordingly. And more over we could take the amount of chemicals in the soil into considerations and predict which plants could be cultivated.

By using machine learning techniques we can figure out the way to this. We took a few datasets that contains the details of previous harvest crop plantation like the amount of pesticides, number of insects, soil type, crop type etc., and with respective to the crop damage status of particular harvest plantation. And another dataset that consists of the amount of phosphorus, ammonia etc in the soil to predict which crop should be grown for better results.

Figure 1: Graph indicating months of high loss



3. LITERATURE REVIEW

In our study, data is extracted from the vast latitudes and delivers a resolution prophesy commensurate using a system. Crop forecasting strength before the commencement of the harvest-time and seed season. Crop Yield Prediction Using Decision Trees on a Global and Regional Scale Minnesota University, Institute for the Climate, St. Paul, MN 55108, USA. Because of its excellent accuracy and performance, the results suggest RF as a unique ML approach for predicting regional and worldwide harvest yields. The Journal is implemented using two layers one is regression and the other is a k-nearest neighbour. The machine learning technique was used to forecast agricultural yield in. The International Review of Technology Scientific Research is a publication dedicated to the advancement of technology. The goal of this research is to use the Random Egrest method to estimate crop yield based on existing data. Beulah undertook an analysis of the different data mining techniques used for crop production prediction and found that data mining approaches might help to solve the critical challenges. This Journal is now in use (SVG). There have been presented techniques for significantly excessive chemical consumption. This research have established a link between chemical use and agricultural productivity. Real-time data from Tamil Nadu (TN) was utilized to build the models, and the models are tested using samples under the survey. Under certain conditions, the Random Forest Algorithm may be used to reliably estimate crop production. This publication presents a thorough review of research on the use of machine learning methods in agricultural sectors that led to the enhancement of the technology in this domain. Machine learning was evolved to provide new chances to computation processes in agricultural functioning sectors may be explored, quantified, and evaluated, alongside digitalization, techniques, approaches, and boosted computing. Support Vector Machines (SVM) were used in the implementation of the paper. Review research on the status of nitrogen estimate using machine learning was conducted by Chlingaryan and Sukkarieh. The research concludes that rapid advances in sensing technology and machine learning approaches will result in modest agricultural solutions, supervised a review of literature on various algorithms of ML for agricultural benefits in the predictions of production using meteorological variables. According to the study, you should widen your search to find other crop yield-related factors. A review article on the use of machine learning in agriculture has just been published. This research contributes to broadening the search for other crop yield variables. The literature on agricultural, livestock, water, and soil management was utilised in the analysis. Li, Lecourt, and Bishop oversaw the research, which aimed to anticipate fruit maturity in order to determine the ideal date for harvest prediction.

4. PROBLEM IDENTIFICATION AND OBJECTIVES

4.1 Problem Statement

Pesticides are special, because while they protect the crop with the right dosage. But, if you add more than required, they may spoil the entire harvest. This data is based on crops harvested by various farmers at the end of harvest season. We also took another dataset that represents the chemicals and humidity present in the soil, in order to predict which crop to grow. To simplify the problem, we can assume that all other factors like variations in farming techniques have been controlled for.
Determining the outcome of the harvest season, i.e. whether the crop would be healthy (alive), damaged by pesticides or damaged by other reasons.
Determining the type of crop that has to be cultivated on that particular soil, in order to achieve more yield.

4.2 Objectives

The following are the objectives of this project:

- To provide an efficient model to farmers for their best practices.
- To Enhance the Crop status after the whole harvest season
- To predict whether the crop is alive and or damaged by pesticides or damaged by some other reason.
- To recommend which crop should be cultivated in their lands according to their soil fertility.
- To provide the farmers a user friendly front end to use this model and use them for their harvest prediction and crop recommendation.

5. SYSTEM METHODOLOGY

This Chapter describes the proposed system, working methodology, performance metrics, and software and hardware details.

5.1 Proposed System

In this study, for the harvest model two approaches are explored with the data points to bring the better true positives out from the new prediction points. Firstly, a whole inspection of the data where there are continuous data points in all the columns is performed. There are 8 independent variables and a dependent variable which is a predictor with 3 unique data points, (0,1,2) which is [Crop is alive, damaged by pesticides, damaged by some reason].For the crop prediction model, we have 7 independent variables and 1 dependent variable with 22 different types of crops that can be classified into. After meticulous visualization of the data, the missing values and outliers are corrected with the mean of the columns and once the data is performed with data pre-processing it is injected with several machine learning algorithms, Random Forest, KNN, Decision Trees, AdaBoost, XGBoost, Gaussian Naive Bayes and LightGBM. Then observed the better algorithm that fits the model but with a minimal amount of accuracy, precision, recall and F1-Score. Now another approach to increase efficiency is introduced. Here extension of columns with all possible ways of grouping columns and then injecting all the algorithms again with the extended dataset yielded high accuracy and better results.

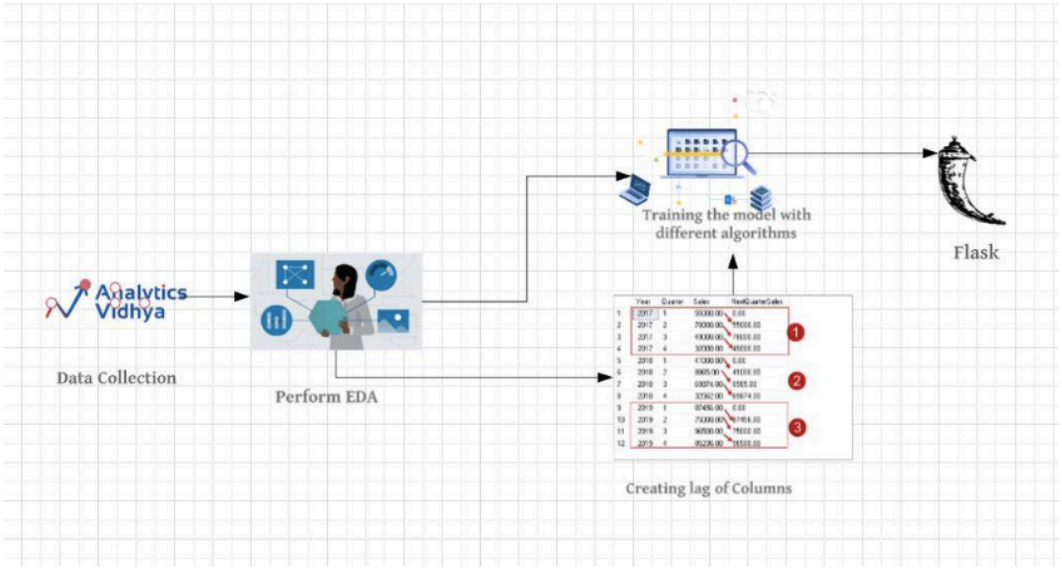


Figure 2 : System Block Diagram

5.2 Working Methodology

We took the data from a hack-a-thon conducted by Analytical vidya. There are two approaches that I did with the data points to bring out the better true positives out from the new data prediction points.

Firstly, we did the whole inspection on the data where there are continues data points in the all columns there are 8 independent variables and a dependent variable (predictor) with 3 unique data points [0,1,2] i.e.,[Crop is alive, damaged by pesticides, damaged by some reason] and there are some missing values in the data.
Now, we did the data visualization to get better understanding of the data and then we found that there are some outliers present in the data. So, we replaced them with the mean of those particular columns and then we certainly put our data into algorithms and observed the results and found out the better algorithm that fits to my model but with minimal amount of accuracy, precision, recall and F1-Score.
Hence we did approach another way to increase our model accuracy. Here, we extended the columns with all possible ways of grouping the columns together and again we put this into algorithms where we observed that our accuracy increased compared with the last approach.
For the crop model, the dataset is clean so without any major preprocessing, we have moved forward to model training and obtained the best suitable model for the work.

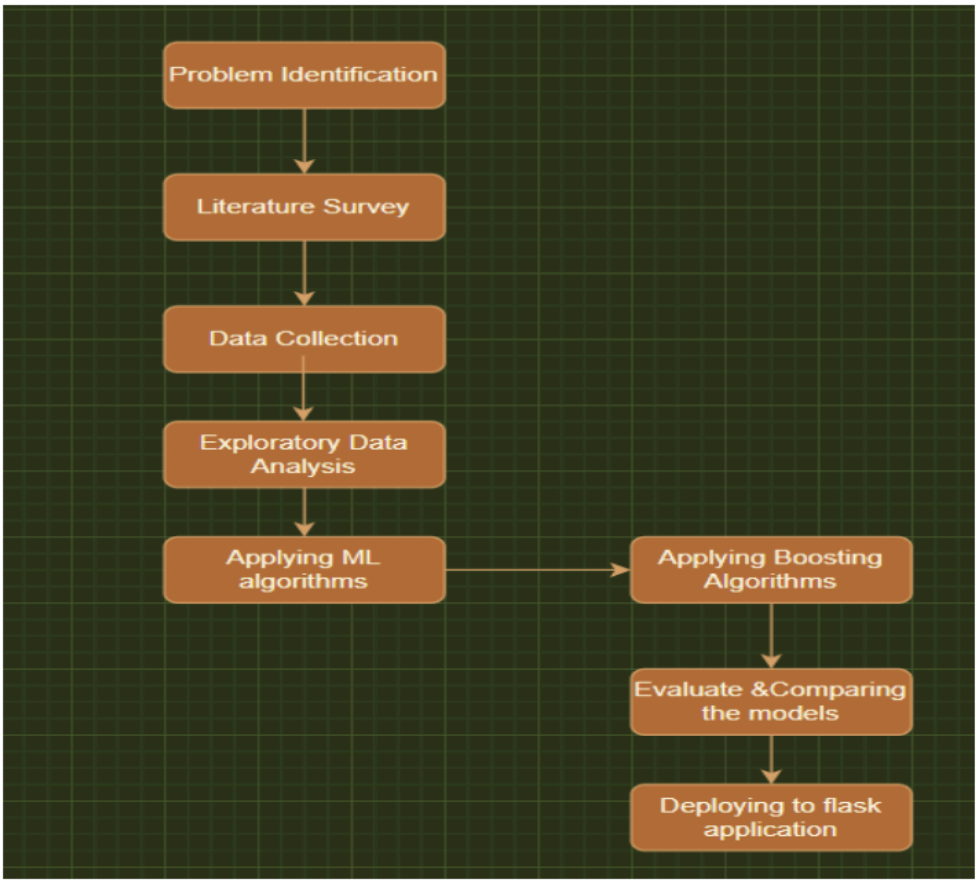


Figure 3: Flow Chart

5.2.1 Data Set

	A	B	C	D	E	F	G	H	I	J
1	ID	Estimated_Insects_Count	Crop_Type	Soil_Type	Pesticide_Use_Category	Number_Doses_Week	Number_Weeks_Used	Number_Weeks_Quit	Season	Crop_Damage
2	F00000001	188	1	0	1	0	0	0	1	0
3	F00000003	209	1	0	1	0	0	0	2	1
4	F00000004	257	1	0	1	0	0	0	2	1
5	F00000005	257	1	1	1	0	0	0	2	1
6	F00000006	342	1	0	1	0	0	0	2	1
7	F00000008	448	0	1	1	0		0	2	1
8	F00000009	448	0	1	1	0		0	2	1
9	F00000010	577	1	0	1	0	0	0	1	2
10	F00000012	731	0	0	1	0	0	0	2	0
11	F00000020	1132	1	0	1	0	0	0	1	2
12	F00000021	1212	1	0	1	0		0	3	0
13	F00000023	1575	0	0	1	0	0	0	1	1
14	F00000024	1575	0	1	1	0	0	0	2	1
15	F00000028	1575	1	1	1	0	0	0	2	1
16	F00000029	1575	1	1	1	0	0	0	2	2
17	F00000030	1785	1	1	1	0	0	0	2	1
18	F00000035	2138	0	1	1	0	0	0	1	1
19	F00000037	2401	0	1	1	0		0	1	1
20	F00000038	2401	1	1	1	0	0	0	2	1
21	F00000039	2401	1	1	1	0	0	0	2	1
22	F00000045	2999	0	1	1	0	0	0	3	1

Figure 4 : Dataset

- ID – unique harvest ID
- Estimated Insect Count – Number of insects estimated in particular Harvest Season
- Crop Type – Type of the crop in the harvest
- Soil Type – Type of the Soil in the Harvest
- Pesticide Use Category – Category of the pesticides used in harvest season
- Number Doses Week – No: of Doses with respective to pesticides
- Number Weeks Used – No: of weeks that pesticides are used
- Number week Quit – Number of Weeks that the plantation quits
- Season – Type of season
- Crop Damage – (Dependent variable) shows the status of crop

5.2.1.2 Crop Dataset

192	79	59	17	20.37999665	63.73849998	6.644205485	108.5054416	maize
193	91	55	15	18.09300227	72.61024172	6.376651091	78.96159541	maize
194	76	51	18	26.16985907	71.96246617	6.247040422	79.84925393	maize
195	87	48	25	18.65396672	61.37879671	6.656730007999999	93.62039175	maize
196	71	60	22	26.07470121	59.37147589	6.2048017	85.75692395	maize
197	90	57	24	18.92851916	72.80086137	6.158860284	82.34162918	maize
198	67	35	22	23.30546753	63.24648023	6.3856842139999985	108.7603001	maize
199	60	54	19	18.74826712	62.49878458	6.417820493	70.23401597	maize
200	83	58	23	19.74213321	59.66263104	6.381201909	65.50861389	maize
201	83	57	19	25.73044432	70.74739256	6.877869005	98.73771338	maize
202	40	72	77	17.02498456	16.98861173	7.485996067	88.55123143	chickpea
203	23	72	84	19.02061277	17.13159126	6.920251378	79.92698081	chickpea
204	39	58	85	17.88776475	15.40589717	5.9969320370000005	68.54932919	chickpea
205	22	72	85	18.86805647	15.65809214	6.391173589	88.51048983	chickpea
206	36	67	77	18.36952567	19.56381041	7.152811172000001	79.26357665	chickpea
207	32	73	81	20.45078582	15.40312102	5.988992796000002	92.68373702	chickpea
208	58	70	84	20.6543203	16.60820843	6.231049027999999	74.6631118	chickpea
209	59	70	84	17.3348681	18.74926979	7.550808267000001	82.61734721	chickpea
210	42	62	75	18.17912258	18.90426935	7.010570541	81.84997529	chickpea
211	28	74	81	18.01272266	18.30968112	8.753795334	81.98568791	chickpea
212	58	66	79	20.99373558	19.33470387	8.718192847000001	93.55280105	chickpea
213	43	66	79	19.46233971	15.22538951	7.976607593	74.58565097	chickpea

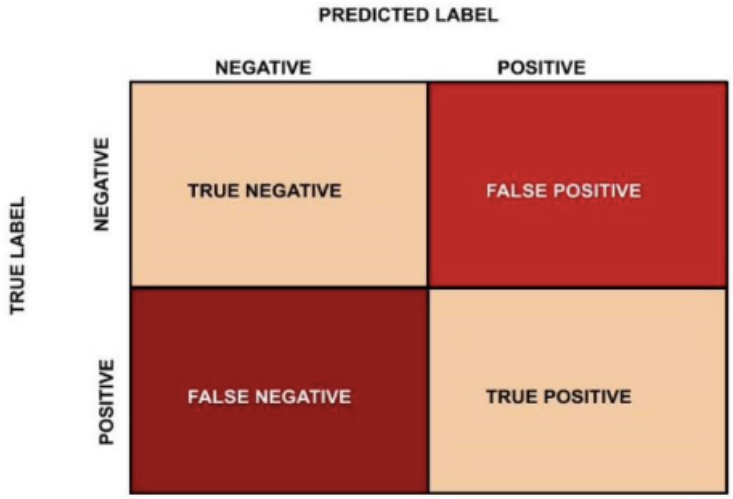
Figure 4 : Dataset

- N - Nitrogen value in the soil.
- P - Phosphorous value in the soil.
- K - Potassium value in the soil.
- Temperature - The temperature of the soil.
- Humidity - Humidity levels in soil.
- Ph - ph value of the soil
- Rainfall - The amount of rainfall the soil receives every year.
- Label - The type of the crop that can be grown in that soil.

5.2.2 Performance Metrics

i. Accuracy

The accuracy of a model is the number of new data points that the algorithm correctly classified. For instance, if the algorithm was tested on 100 new data points, and the algorithm correctly classified 97 of them — then we know that the accuracy is 97%.
A confusion matrix is a method for summing up the presentation of an order calculation. Arrangement precision alone can be misdirecting assuming we have an inconsistent number of perceptions in each class or on the other hand on the off chance that we have multiple classes in your dataset. Computing a disarray network can provide us with a superior thought of what the arrangement model is getting right and what kinds of mistakes it is making.



$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Figure 5 : Confusion Matrix

ii. Precision

The accuracy is the proportion tp/(tp+fp) where tp is the quantity of genuine up-sides and fp the quantity of misleading up-sides. The accuracy is naturally the capacity of the classifier not to mark as certain an example that is negative. The best worth is 1 and the most awful worth is 0.

$$Precision = \frac{Correct\ Positive\ Predictions}{All\ Positive\ Prediction} = \frac{TP}{TP + FP}$$

iii. Recall

The recall is the proportion tp/(tp+fn) where tp is the quantity of genuine up-sides and fn the quantity of bogus negatives. The review is instinctively the capacity of the classifier to track down every one of the positive examples. The best worth is 1 and the most obviously awful worth is 0.

$$Recall = \frac{Correct\ Positive\ Predictions}{All\ Positives} = \frac{TP}{TP + FN}$$

iv. F1-Score

F1-score is otherwise called adjusted F-score or F-measure. It very well may be deciphered as a weighted normal of the accuracy and review, where a F1 score arrives at its best worth at 1 and most terrible score at 0. The overall commitment of accuracy and review to the F1 score are equivalent.

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} \right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

v. Loss

The lower the misfortune, the better a model will be (except if the model has over-fitted to the preparation information). The misfortune is determined on preparing and approval and its translation is the way well the model is accomplishing for these two sets. Dissimilar to precision, misfortune isn't a rate. It is a summation of the blunders made for every model in preparing or approval sets.

Misfortune esteem infers how well or inadequately a specific model acts after each emphasis of improvement. Preferably, one would anticipate the decrease of misfortune after each, or a few, iteration(s)

6. OVERVIEW OF TECHNOLOGIES

6.1. Numpy

NumPy is python package used of scientific computing. Numpy is a python library that includes a multidimensional arrays and its derived objects such as matrices and masked arrays. It also contains variety of routines to perform fast array operations such as mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more.

6.2. Pandas

Pandas is an open source Python bundle that is generally broadly utilized for information science/information investigation and AI errands. Pandas is palced above the Numpy bundle, which offers help for multi-faceted clusters. As quite possibly the most famous datum fighting bundles, Pandas functions admirably with numerous different information science modules inside the Python environment, and is commonly remembered for each Python dissemination, from those that accompany your working framework to business seller circulations like ActiveState's ActivePython.

6.3. Matplotlib

Matplotlib is a cross-stage, information representation and graphical plotting library for Python and its mathematical augmentation NumPy. All things considered, it offers a suitable open source option in contrast to MATLAB. Engineers can likewise utilize matplotlib's APIs (Application Programming Interfaces) to install plots in GUI applications.

6.4. Seaborn

Seaborn is an open-source Python library which is based on top of matplotlib. It is utilized for information perception and exploratory information investigation. Seaborn works effectively with dataframes and the Pandas library. The diagrams made can likewise be modified without any problem. The following are a couple of advantages of Data Visualization.

6.5. Scipy

SciPy is a logical calculation library that utilizes NumPy under. SciPy represents Scientific Python. It gives greater utility capacities to improvement, details and sign handling. SciPy is also open source like Numpy, so we can utilize it unreservedly. SciPy was made by NumPy's maker Travis Olliphant.

6.6. Scikit learn

Scikit-learn is a library in Python that gives numerous unaided and regulated learning calculations. It's based upon a portion of the innovation you could currently be acquainted with, as NumPy, pandas, and Matplotlib!
The usefulness that scikit-learn gives include:

- Relapse, including Linear and Logistic Regression
- Arrangement, including K-Nearest Neighbors
- Grouping, including K-Means and K-Means++
- Model choice
- Preprocessing including Min-Max Normalization

6.7. NLTK

The Natural Language Toolkit (NLTK) is a stage utilized for building Python programs that work with human language information for applying in factual regular language handling (NLP).
It contains text handling libraries for tokenization, parsing, grouping, stemming, labeling and semantic thinking. It likewise incorporates graphical shows and test informational collections as well as joined by a cook book and a book which makes sense of the standards behind the basic language handling errands that NLTK upholds.

6.8. LightGBM

LightGBM is an angle supporting structure that utilizations tree based learning calculations. It is intended to be circulated and effective with the accompanying benefits:

- Quicker preparing speed and higher effectiveness.
- Lower memory use.
- Better precision.
- Backing of equal, appropriated, and GPU learning.
- Fit for dealing with huge scope information.

7. IMPLEMENTATION

7.1 Coding

7.1.1 Data Visualisation

Various visualisations used in this project are :

I. Correlation between the attributes

Correlation states the strong and weak bond between the columns respectively. So that it help us while we are filling the missing values, to do feature engineering etc.

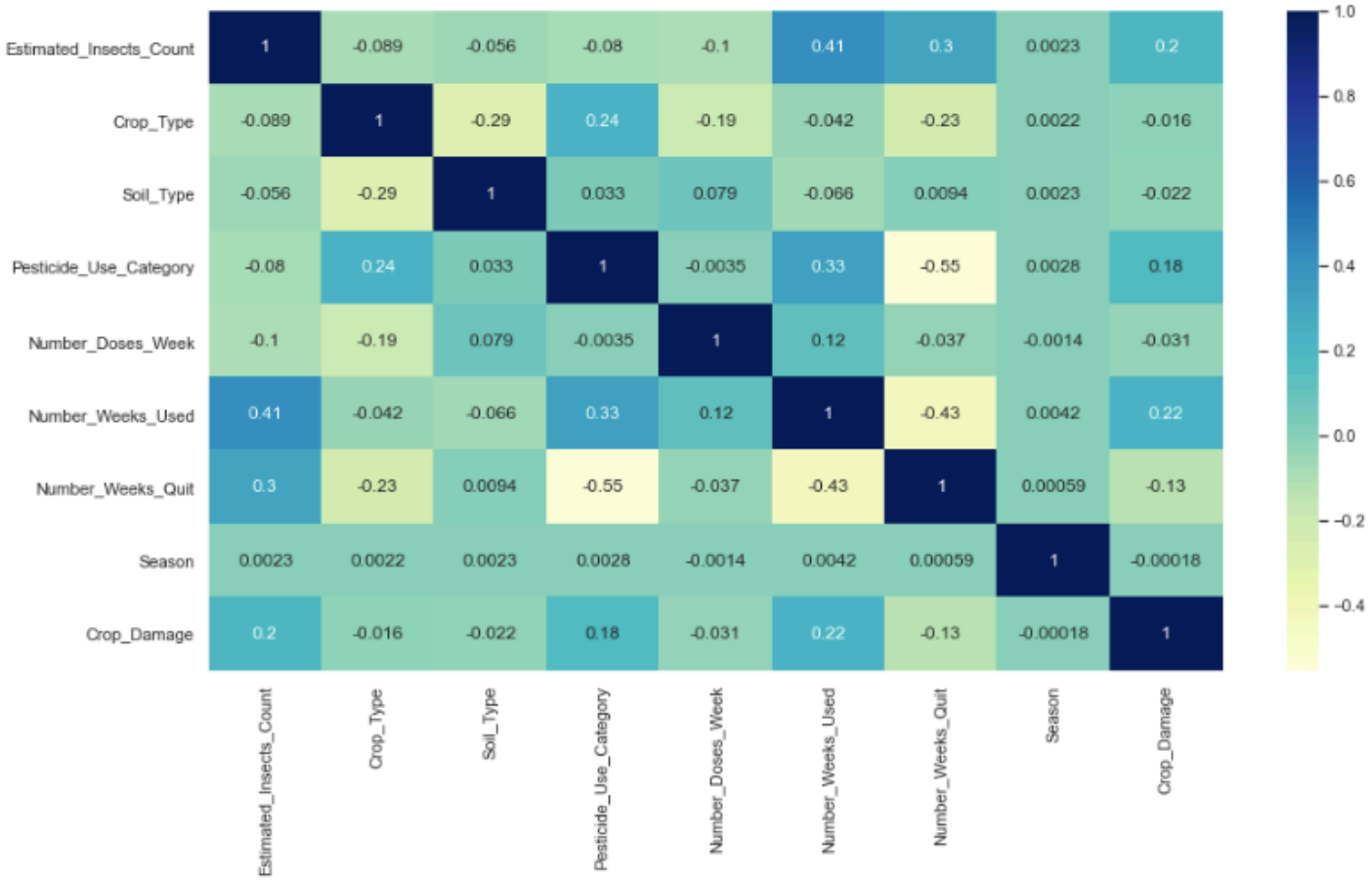


Figure 6 : Correlation Heatmap

II. Univariate Analysis

A. Crop Damage Grouped Count

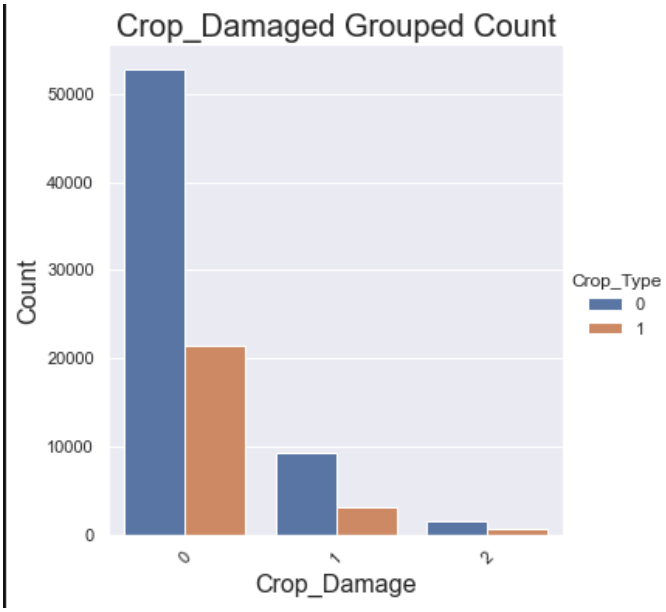


Figure 7 : Crop_Damaged Grouped Count

Here is the graph of crop_type Grouped count which is observed between the crop_type with respective to the crop damage where we got some insights like crop_type 0 is damaged more compared with crop_type 1.

B. Estimated Insects Grouped Count

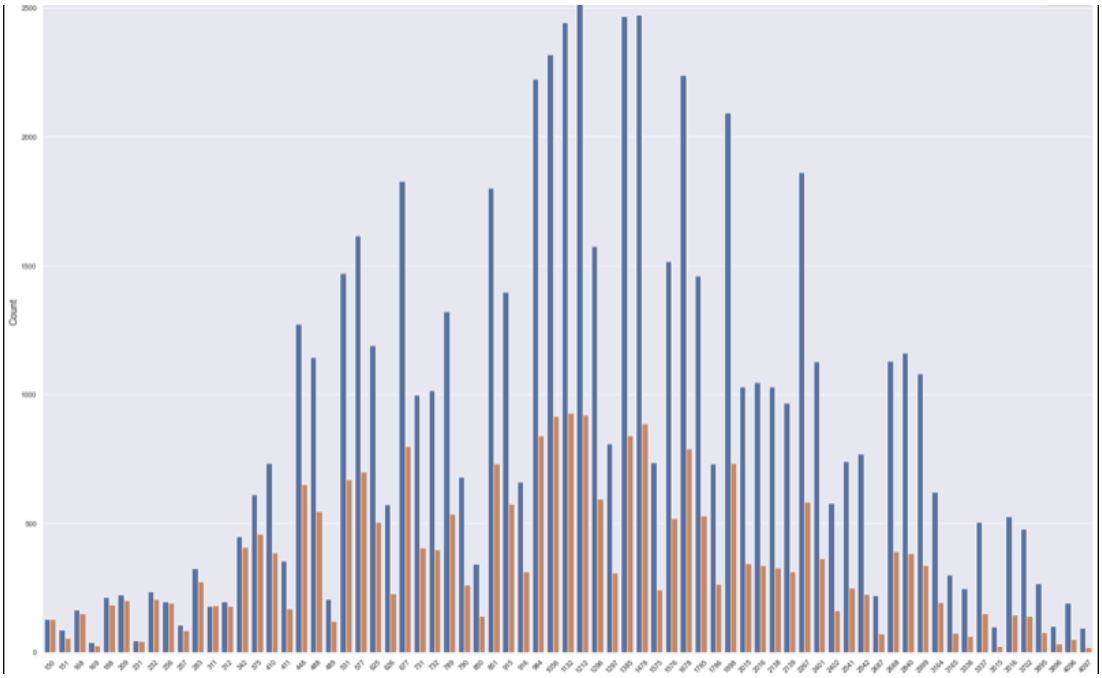


Figure 8 : Estimated Insects Grouped Count

Here is the graph of Estimated Insects Grouped Count which is observed between the estimated insect count with respective to the crop_type where we got some insights like crop_type 0 contains more insects than crop_type 1.

C. Soil Type Grouped Count

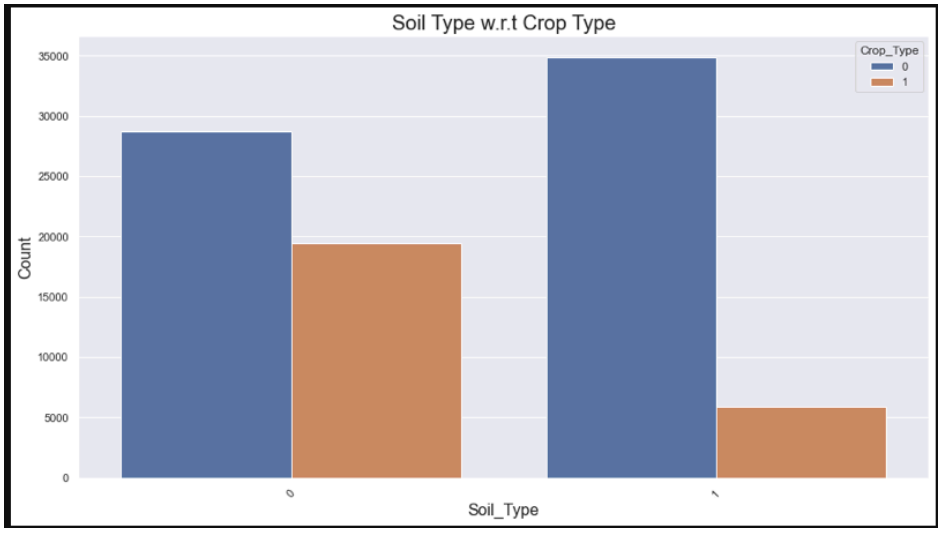


Figure 9 : Soil Type Grouped Count

Here is the graph of soil_type grouped count which is observed between the crop_type with respective to the soil_type where we got some insights like crop_type 0 is showing good results in both soil types compared with crop_type 1.

D. Season Grouped Count

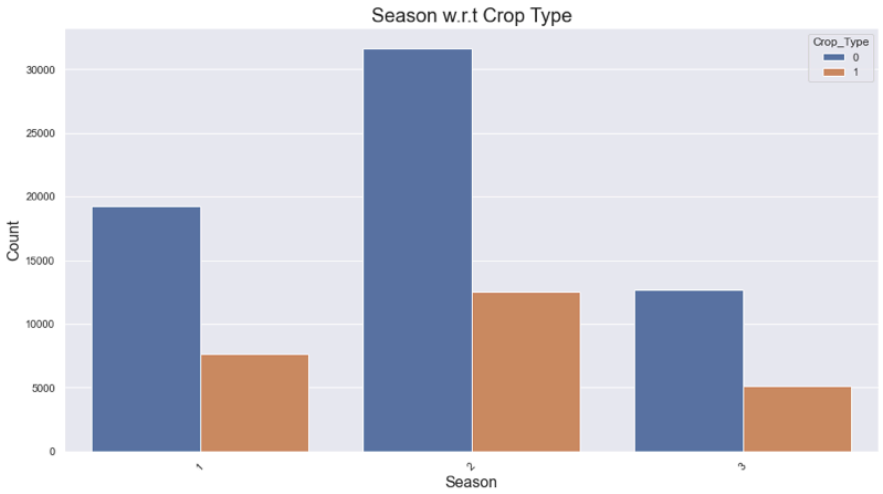


Figure 10 : Season Grouped Count

Here is the graph of season grouped count which is observed between the crop_type with respective to the season where we got some insights like type 2 season shows the highest production of both the type of crops

III. Bivariate Analysis

A. Crop Damage Vs Pesticide Use Category

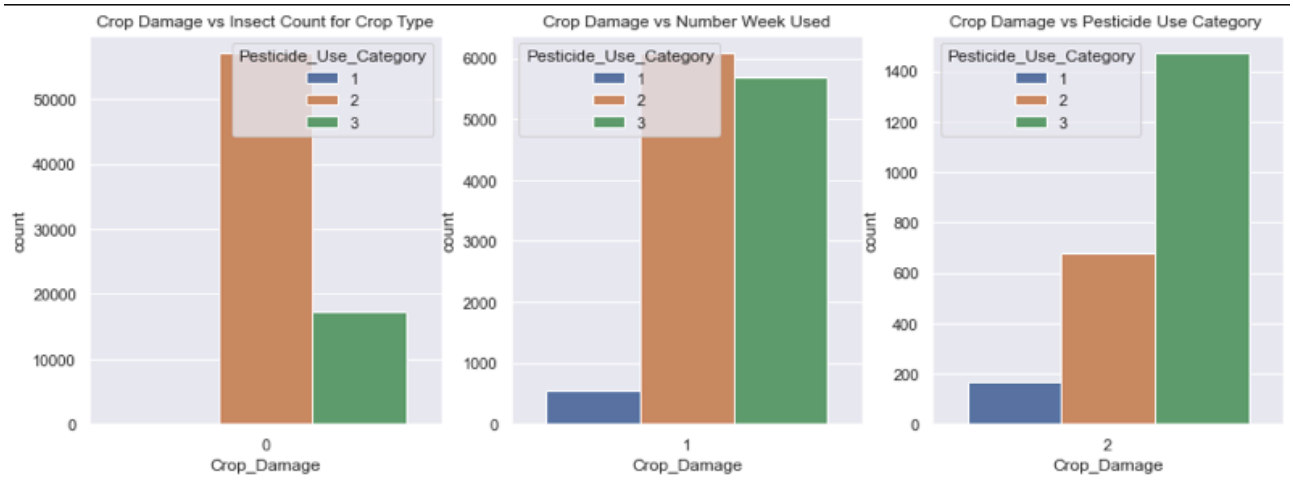


Figure 11 : Crop Damage Vs Pesticide Use Category

The above bar graph shows us the crop damage vs Pesticide use category.

B. Distplot

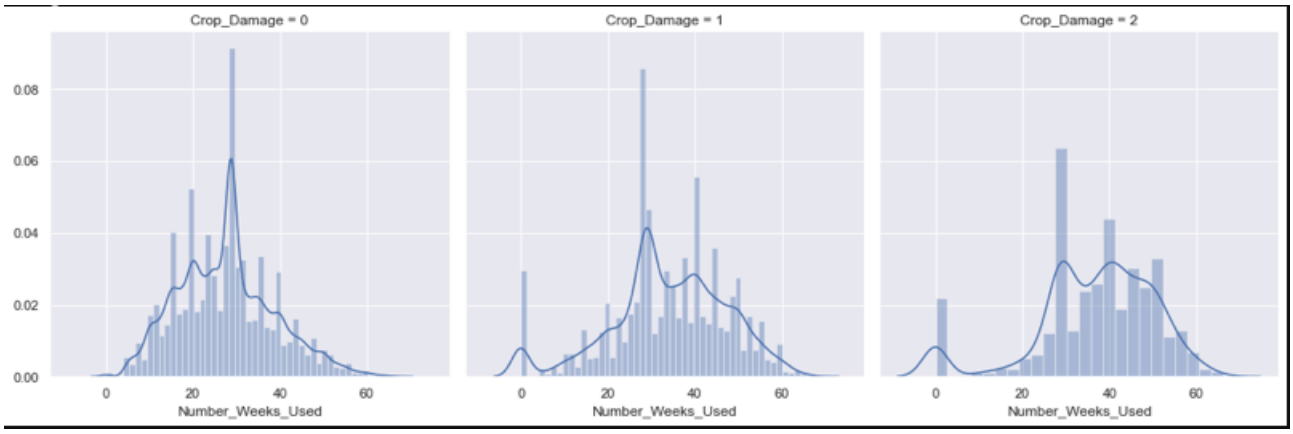


Figure 12 : Distplot

The above graph shows the distribution of Number weeks used with respective to the dependent class variable.

C. Boxplot with outliers present

```
Estimated_Insects_Count      AxesSubplot(0.125,0.749828;0.133621x0.130172)
Crop_Type                  AxesSubplot(0.285345,0.749828;0.133621x0.130172)
Soil_Type                   AxesSubplot(0.44569,0.749828;0.133621x0.130172)
Pesticide_Use_Category     AxesSubplot(0.606034,0.749828;0.133621x0.130172)
Number_Doses_Week          AxesSubplot(0.766379,0.749828;0.133621x0.130172)
Number_Weeks_Used             AxesSubplot(0.125,0.593621;0.133621x0.130172)
Number_Weeks_Quit          AxesSubplot(0.285345,0.593621;0.133621x0.130172)
Season                      AxesSubplot(0.44569,0.593621;0.133621x0.130172)
Crop_Damage                AxesSubplot(0.606034,0.593621;0.133621x0.130172)
dtype: object
```

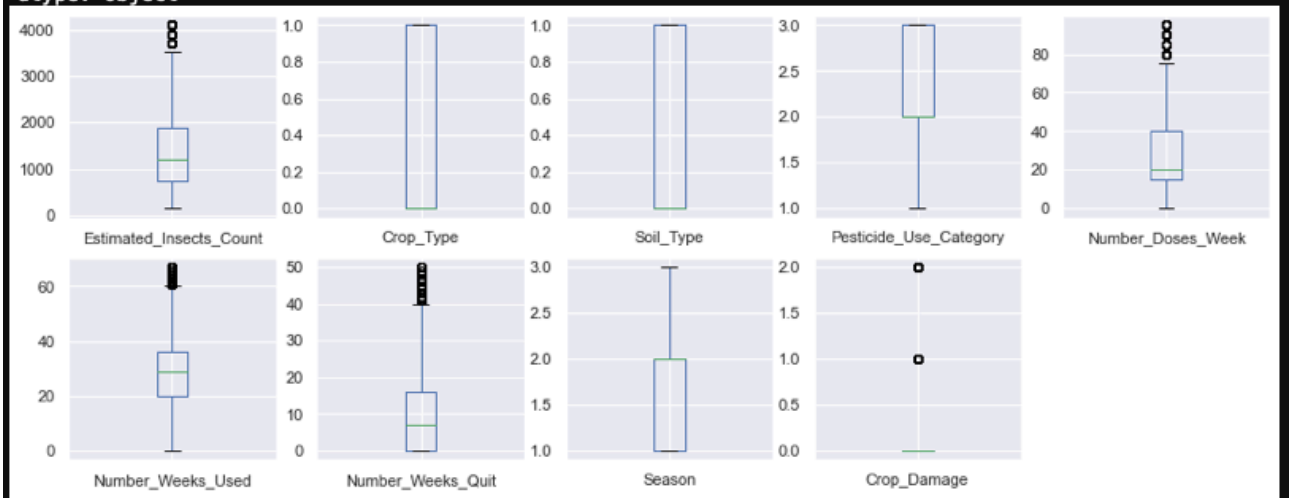


Figure 13 : Boxplot with outliers present

D. Boxplot after removing the outliers

```
Estimated_Insects_Count      AxesSubplot(0.125,0.749828;0.133621x0.130172)
Crop_Type                  AxesSubplot(0.285345,0.749828;0.133621x0.130172)
Soil_Type                   AxesSubplot(0.44569,0.749828;0.133621x0.130172)
Pesticide_Use_Category     AxesSubplot(0.606034,0.749828;0.133621x0.130172)
Number_Doses_Week          AxesSubplot(0.766379,0.749828;0.133621x0.130172)
Number_Weeks_Used             AxesSubplot(0.125,0.593621;0.133621x0.130172)
Number_Weeks_Quit          AxesSubplot(0.285345,0.593621;0.133621x0.130172)
Season                      AxesSubplot(0.44569,0.593621;0.133621x0.130172)
Crop_Damage                AxesSubplot(0.606034,0.593621;0.133621x0.130172)
dtype: object
```

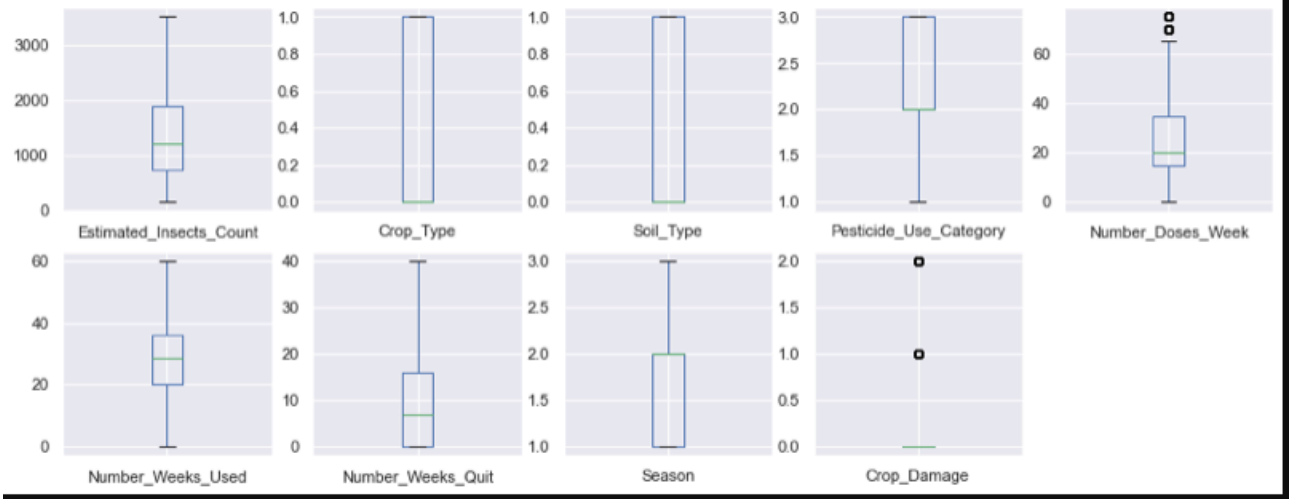


Figure 14 : Boxplot after removing the outliers

7.1.2 Skew Analysis

Skewness alludes to a bending or unevenness that goes astray from the balanced chime bend, or typical dissemination, in a bunch of information. Assuming that the bend is moved to the left or to the right, it is supposed to be slanted. Skewness can be measured as a portrayal of the degree to which a given dissemination fluctuates from a typical appropriation. An ordinary dissemination has a slant of nothing, while a lognormal conveyance, for instance, would display some level of right-slant.

```
array([[<AxesSubplot:title={'center':'Estimated_Insects_Count'}>,
        <AxesSubplot:title={'center':'Crop_Type'}>,
        <AxesSubplot:title={'center':'Soil_Type'}>,
        <AxesSubplot:title={'center':'Pesticide_Use_Category'}>],
       [<AxesSubplot:title={'center':'Number_Doses_Week'}>,
        <AxesSubplot:title={'center':'Number_Weeks_Used'}>,
        <AxesSubplot:title={'center':'Number_Weeks_Quit'}>,
        <AxesSubplot:title={'center':'Season'}>],
       [<AxesSubplot:title={'center':'Crop_Damage'}>, <AxesSubplot:>,
        <AxesSubplot:>, <AxesSubplot:>],
       [<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]],
      dtype=object)
```

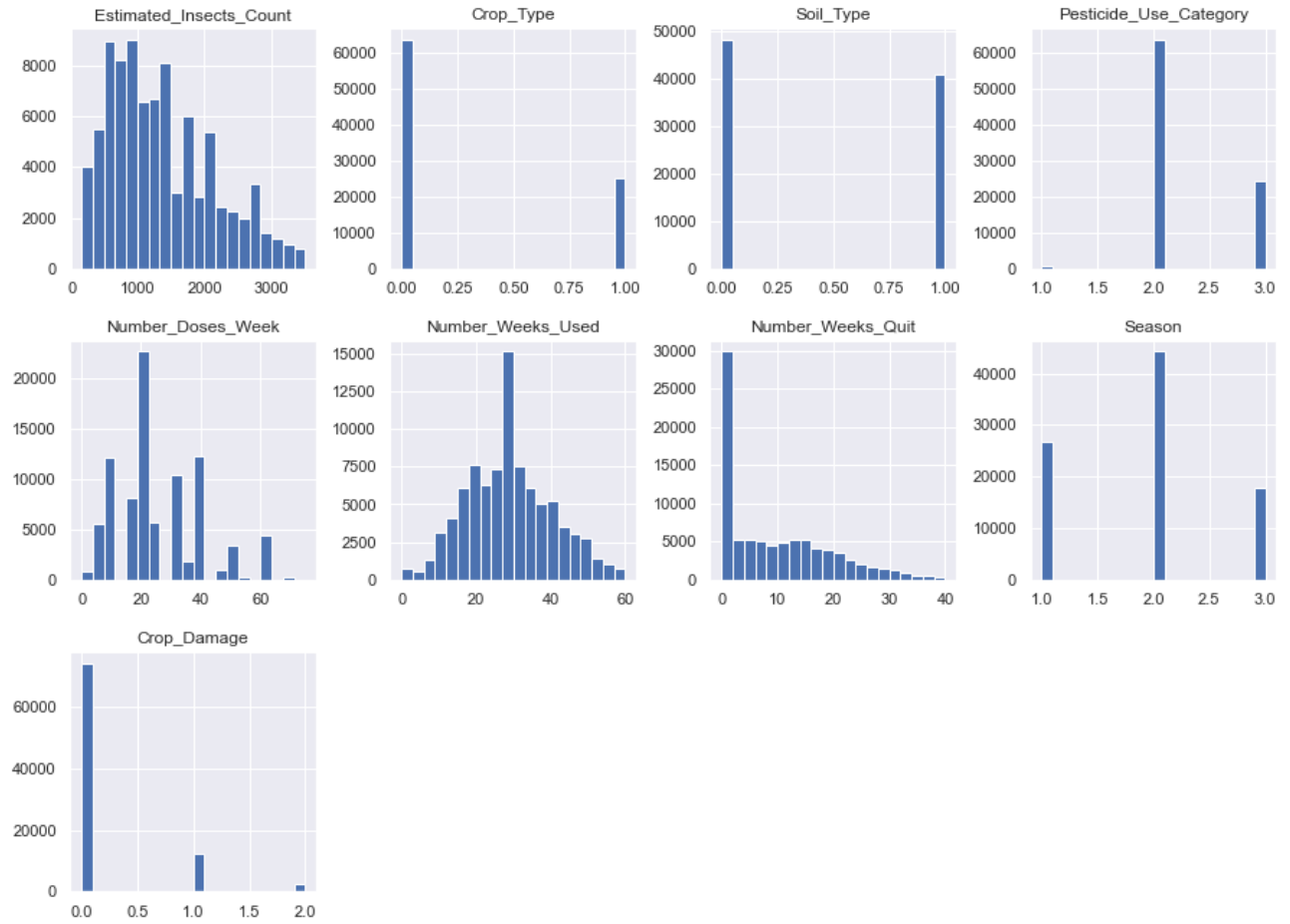


Figure 15: Skew Analysis

After preprocessing the data and removing the outliers it is clear that the data is normally distributed. We can now move further to train the model.

7.1.3 Training the Harvest Model

The harvest model is trained using two approaches:

Approach 1:

The model is trained using KNN, lightgbm and adaboost.

KNN: K-nearest neighbours is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data.

```
Results for model :  KNeighbors Classifier
max accuracy score is 0.8442913654344564
Mean accuracy score is :  0.8420400913398536
Std deviation score is :  0.0015047495269905472
Cross validation scores are :   [0.84396804 0.84256133 0.84115463 0.8428901  0.83962636]
```

Light GBM: It is a fast, distributed, high-performance gradient boosting framework based on a decision tree algorithm, used for ranking, classification and many other machine learning tasks.

```
0.8471190636957011
[[14611   237     0]
 [ 2018   443     0]
 [  328   134     1]]
              precision    recall  f1-score   support

           0       0.86      0.98      0.92     14848
           1       0.54      0.18      0.27      2461
           2       1.00      0.00      0.00       463

    accuracy                           0.85     17772
   macro avg       0.80      0.39      0.40     17772
weighted avg       0.82      0.85      0.81     17772
```

AdaBoost: It is an ensemble learning method (also known as “meta-learning”) which was initially created to increase the efficiency of binary classifiers. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers, and turn them into strong ones.

```
0.8407607472428539
[[14827    21     0]
 [ 2346   115     0]
 [  427    36     0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91     14848
           1       0.67      0.05      0.09      2461
           2       0.00      0.00      0.00       463

    accuracy                           0.84     17772
   macro avg       0.50      0.35      0.33     17772
weighted avg       0.80      0.84      0.78     17772
```

Approach 2:

The data is manipulated using lags and trained using lightgbm because it ensures the accuracy to be more specific than the previous approach.

Lags: The pandas library includes built-in functionalities that allow you to perform different tasks with only a few lines of code. One of these functionalities is the creation of lags and leads of a column. lag shifts a column down by a certain number. lead shifts a column up by a certain number.

```
df_data['Crop_Damage_lag1'] = df_data['Crop_Damage'].shift(fill_value=-999)
df_data['Estimated_Insects_Count_lag1'] = df_data['Estimated_Insects_Count'].shift(fill_value=-999)
df_data['Crop_Type_lag1'] = df_data['Crop_Type'].shift(fill_value=-999)
df_data['Soil_Type_lag1'] = df_data['Soil_Type'].shift(fill_value=-999)
df_data['Pesticide_Use_Category_lag1'] = df_data['Pesticide_Use_Category'].shift(fill_value=-999)
df_data['Number_Doses_Week_lag1'] = df_data['Number_Doses_Week'].shift(fill_value=-999)
df_data['Number_Weeks_Used_lag1'] = df_data['Number_Weeks_Used'].shift(fill_value=-999)
df_data['Number_Weeks_Quit_lag1'] = df_data['Number_Weeks_Quit'].shift(fill_value=-999)
df_data['Season_lag1'] = df_data['Season'].shift(fill_value=-999)

df_data['Crop_Damage_lag2'] = df_data['Crop_Damage'].shift(periods=2,fill_value=-999)
df_data['Estimated_Insects_Count_lag2'] = df_data['Estimated_Insects_Count'].shift(periods=2,fill_value=-999)
df_data['Crop_Type_lag2'] = df_data['Crop_Type'].shift(fill_value=-999)
df_data['Soil_Type_lag2'] = df_data['Soil_Type'].shift(fill_value=-999)
df_data['Pesticide_Use_Category_lag2'] = df_data['Pesticide_Use_Category'].shift(periods=2,fill_value=-999)
df_data['Number_Doses_Week_lag2'] = df_data['Number_Doses_Week'].shift(periods=2,fill_value=-999)
df_data['Number_Weeks_Used_lag2'] = df_data['Number_Weeks_Used'].shift(periods=2,fill_value=-999)
df_data['Number_Weeks_Quit_lag2'] = df_data['Number_Weeks_Quit'].shift(periods=2,fill_value=-999)
df_data['Season_lag2'] = df_data['Season'].shift(periods=2,fill_value=-999)
```

```
clf = lgb.LGBMClassifier(**params)

clf.fit(df_train[feature_cols], df_train[label_col], eval_metric='multi_error', verbose=False, categorical_feature=cat_

# eval_score_auc = roc_auc_score(df_train[label_col], clf.predict(df_train[feature_cols]))
eval_score_acc = accuracy_score(df_train[label_col], clf.predict(df_train[feature_cols]))

print('ACC: {}'.format(eval_score_acc))
```

```
ACC: 0.9911881878952936
```

7.1.4 Training the Crop Prediction Model

Different models used for training the data

Model 1:

Decision Tree: Decision Tree is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

```
5]:
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(Ytest,predicted_values))
```

```
DecisionTrees's Accuracy is:  90.0
```

Model 2:

Gaussian Naive Bayes: Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. We have explored the idea behind Gaussian Naive Bayes along with an example. Before going into it, we shall go through a brief overview of Naive Bayes.

```
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
Naive Bayes's Accuracy is:  0.990909090909091
```

Model 3:

Random forest: It is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
RF's Accuracy is:  0.990909090909091
```

7.2 Results

Our Services

CROP PREDICTION

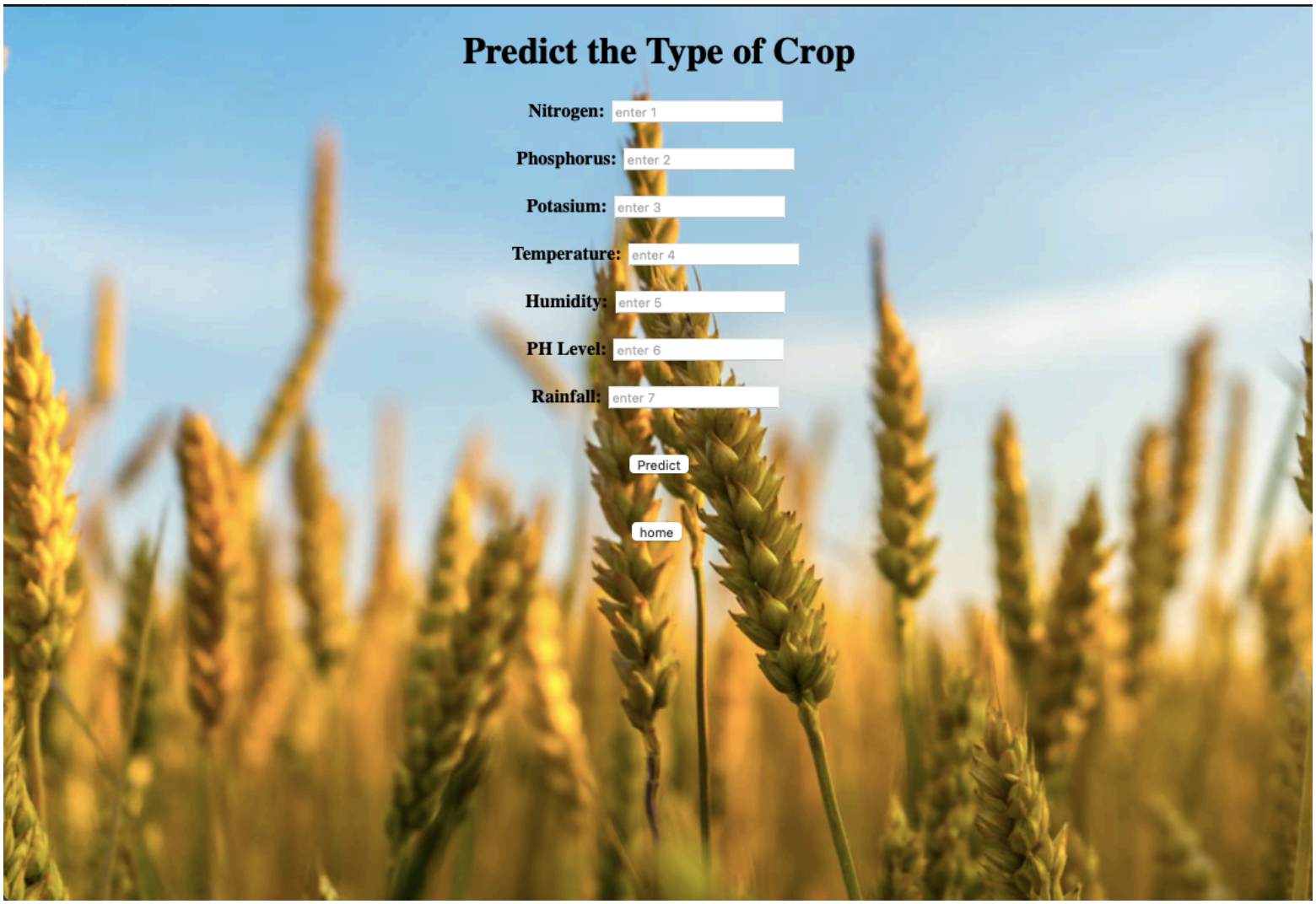
Recommendation about the type of crops to be cultivated which is best suited for the respective conditions

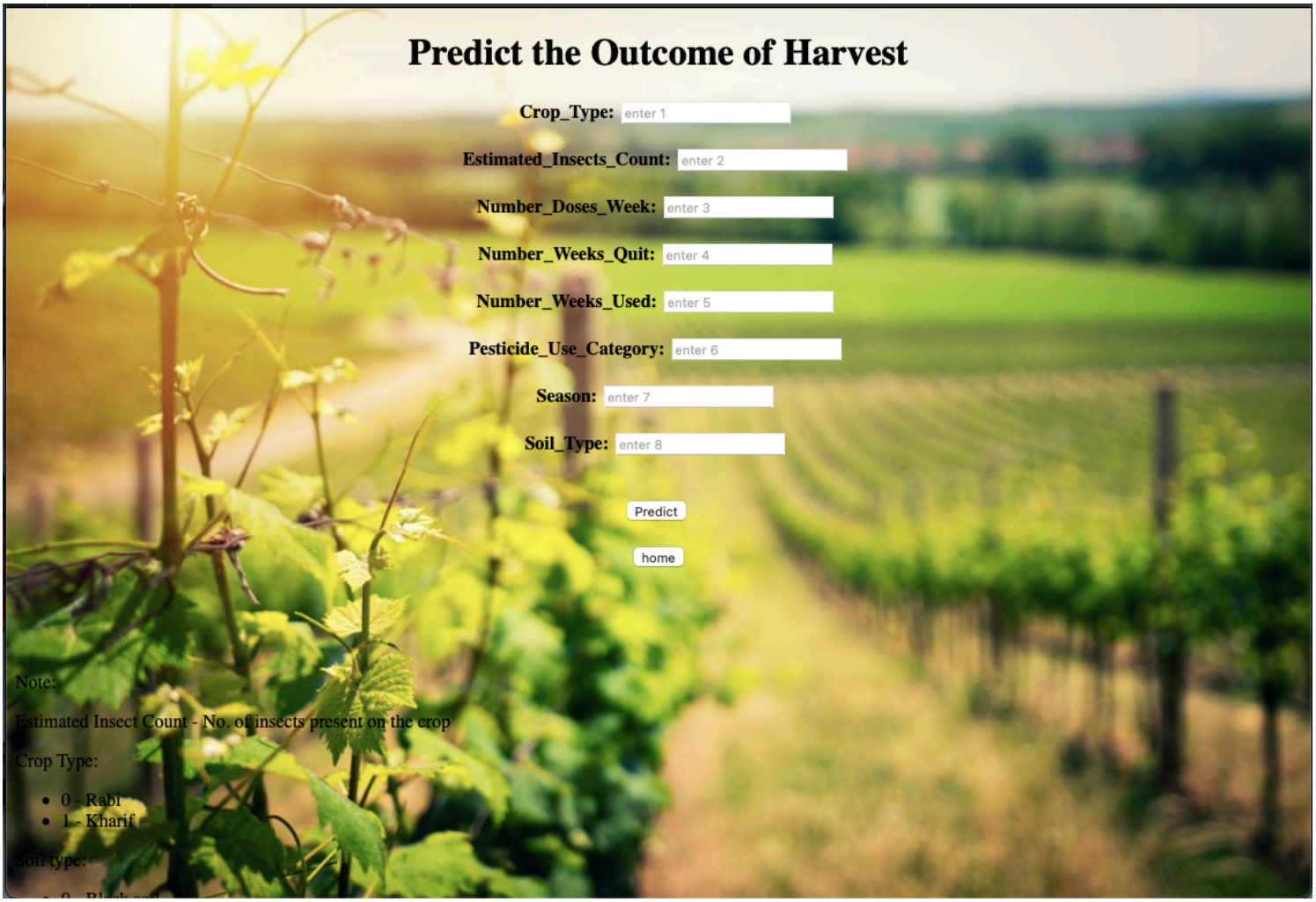
Start Crop Prediction

HARVEST DETERMINER

Determine whether the crop is healthy or damaged

Start Harvest Prediction


Predict the Type of Crop
Nitrogen:
enter 1
Phosphorus:
enter 2
Potasium:
enter 3
Temperature:
enter 4
Humidity:
enter 5
PH Level:
enter 6
Rainfall:
enter 7
Predict
home


Predict the Outcome of Harvest
Crop_Type: enter 1
Estimated_Insects_Count: enter 2
Number_Doses_Week: enter 3
Number_Weeks_Quit: enter 4
Number_Weeks_Used: enter 5
Pesticide_Use_Category: enter 6
Season: enter 7
Soil_Type: enter 8
Predict
home
Note:
Estimated Insect Count - No. of insects present on the crop
Crop Type:
0 - Rabi
1 - Kharif
Soil type:

Results and Discussions

Harvest Model:

Outcome of harvest season is all about predicting the outcome or status of the crop plantation at the end of harvest season. Hence, we solved this problem using machine learning where we did a comparative study by following some algorithms and tried to fit our model in them they are., Random Forest Classifier, K-Nearest Neighbor, Decision Tree Classifier, Gaussian NB, Ada-Boost, LightGBM, XgBoost and later we observed the performance metrics of each algorithm by that we concluded LightGBM has high performance when compared with remaining algorithms.

Also, this is the first approach where we performed exploratory data analysis, and trained our model but then we only got 84.6% as the highest accuracy so, we did approach a second method.

S.NO	ALGORITHM	ACCURACY	Precision	Recall	F1-Score
1	Random Forest Classifier	82.2%	0.74	0.81	0.73
2	K-Nearest Neighbours	84.3%	0.79	0.84	0.80
3	Decision Tree Classifier	75.3%	0.66	0.71	0.78
4	Gaussian NB	82.4%	0.75	0.80	0.74
5	Adaboost	84%	0.79	0.84	0.77
6	**Lightgbm**	**84.6%**	**0.82**	**0.85**	**0.81**
7	Xgboost	80%	0.72	0.70	0.65

Figure 16: Results of Approach 1

In the second approach we appended some new columns to the existing dataset where we created lags of the columns to create trend in the data also added some inconsistent data and then we repeated to apply exploratory data analysis, feature engineering and again we tried the algorithms to my model. We observed a hike in performance wise and again the boosting algorithm LightGBM has high accuracy than remaining algorithms i.e., 97.2%.

S.NO	ALGORITHM	ACCURACY
1	Lightgbm	97.2%
2	Random Forest Classifier	92.6%
3	K-Nearest Neighbours	93.2%
4	Gaussian NB	77.4%
5	Decision Tree	90.3%
6	Ada Boost	95.4%

Figure 17: Results of Approach 2

Crop prediction is all about predicting which crop should be planted or cultivated by considering the pH values, chemicals present in the soil, humidity of the soil etc. This would give an idea for the farmers which crop should be cultivated at that particular time in that soil. We have the dataset with 7-independent and 1- dependent variable.

By doing data inspection, exploratory data analysis and model building, we got a final accuracy of 99% with average weights of precision, recall and F1-Score as 0.99.

S.NO	ALGORITHM	ACCURACY
1	**Random Forest Classifier**	**99.1%**
2	Logistic Regression	95%
3	Naive Bayes	99%
4	Decision Tree	90%
5	Support Vector Machine	10.6%
6	Xgboost	98%

Figure 18: Results of Crop Model

Conclusions and Future Scope

Outcome of harvest season is all about predicting the outcome or status of the crop plantation at the end of harvest season. This would give an idea for the farmers that how much amount of pesticides he has to use and can also take care of the insects in his field so that he can increase efficiency of the harvest crop to be healthy. I have the dataset with 8-independent and 1- dependent variable. Here, in the dependent variable I have 3 class labels where 0 - Crop is alive, 1 – Damaged by pesticides. 2 – Damaged by some other reason.

By doing data inspection, exploratory data analysis and model building, we got a final accuracy of 97% with average weights of precision, recall and F1-Score as 0.97.

In the future, we will try to research on some more ways to protect crop like controlling the over flow of water in field by measuring the certain amount that required by the fields. We will also do the research on chance to decrease the floods in the crop field by this the farmer ending up on debts will decreases and could live his life smoothly only then we could live our lives smoothly because farmers are the backbone of our country.

8. REFERENCES

1. Sk Al Zaminur Rahman, S.M. Mohidul Islam, Kaushik Chandra Mitra,‖ Soil Classification using Machine Learning Methods and Crop Suggestion Based on Soil Series‖,2018 21st International Conference of Computer and Information Technology (ICCIT), 21-23 December, 2018.
2. S. Panchamurthi. M.E.,M.D. Perarulalan,A. Syed Hameeduddin,P. Yuvaraj,‖Soil Analysis and Prediction of Suitable Crop for Agriculture using Machine Learning‖, International Journal for Research in Applied Science & Engineering Technology(IJRASET),ISSN:2321-9653;IC Value:45.98;SJ Impact Factor:6.887,Volume 7 Issue III,Mar 2019.
3. D Ramesh,B Vishnu Vardhan,—Data Mining Techniques and Applications to Agricultural Yield Data,‖ International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 9, September 2013.
4. https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html
5. https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/
6. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
7. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
8. https://xgboost.readthedocs.io/en/latest/