**Building a Model in Python to Predict Crop Damage during Harvest Season**

Alisha Fal Desai

# Contents

# Problem Definition

# Data Analysis

# EDA Concluding Remarks

# Pre-processing Pipeline

# Building Machine Learning Model

# Concluding Remarks

# Problem Definition

Agriculture plays a critical role in the global economy. With the continuing expansion of the human population understanding worldwide crop yield and the quality of the crop is central to addressing food security challenges and reducing the impacts of climate change.

A farmer's job is a test of endurance and determination. Once the seeds are sown, he works days and nights to make sure that he cultivates a good harvest at the end of season. A good harvest is ensured by several factors such as availability of water, soil fertility, protecting crops from rodents, timely use of pesticides & other useful chemicals and nature. While a lot of these factors are difficult to control for, the amount and frequency of pesticides is something the farmer can control.

Tremendous benefits have been derived from the use of pesticides. They have been an integral part of the process by reducing losses from the weeds, diseases and insect pests that can markedly reduce the amount of harvestable produce. Warren drew attention to the spectacular increases in crop yields in the United States in the twentieth century. Webster et al.stated that “considerable economic losses” would be suffered without pesticide use and quantified the significant increases in yield and economic margin that result from pesticide use. But, while they protect the crop with the right dosage, if you add more than required, they may spoil the entire harvest. A high level of pesticide can deem the crop dead / unsuitable for consumption among many outcomes.

Understanding the amount of damage to the harvest and the factors responsible for it can help the farmer work towards better conditions for reducing crop damage and reaping the best possible yield to minimize damages and maximize profit. We will be using a method that could be used for a variety of Machine Learning problems, following a step-by-step systematic approach. This project would fall under the category **Crop Yield Analytics.**

# In this project, we will attempt to solve the following problems:

* What is the amount of damage a harvest is likely to sustain?
* What are the key indicators that cause damage to crops?

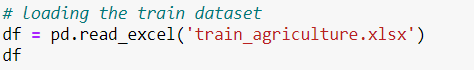
Given that we have data on variety of crop yields, this is a**standard supervised classification problem** where the label is '0'(Alive), '1'(damage due to other causes) and ‘2’ (damage due to pesticides). Here the target variable is 'Crop\_Damage', the amount of damage a crop sustains and whether it is feasible for further processing or consumption or not.

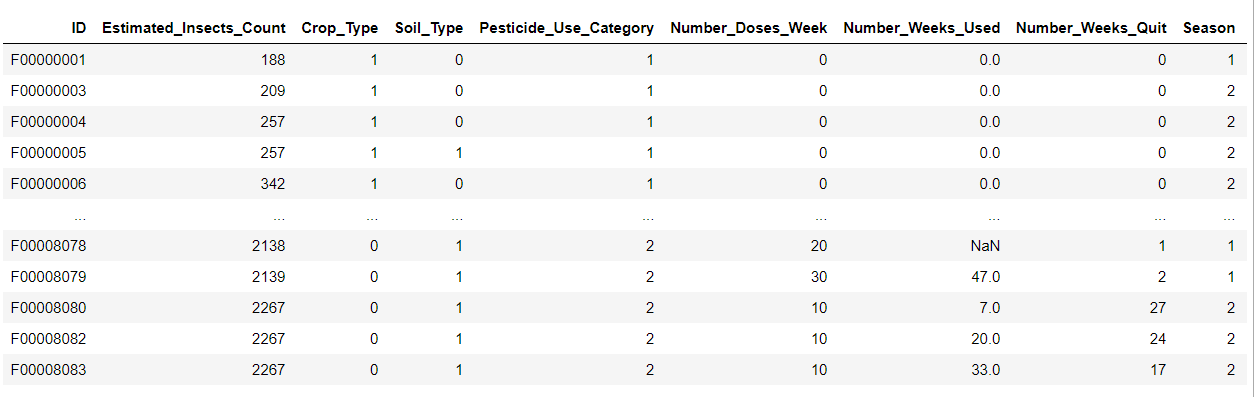
# Data Analysis

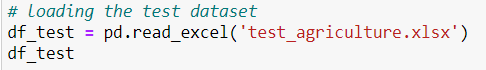
For this project, we have used the test and train dataset named agriculture from GitHub, which contains data for 1199 and 4599 respectively. The features in the dataset include ‘Crop\_Type’, ‘Soil\_Type’, ‘Season’ and ‘Crop\_Damage’ among others. We will study the features of this dataset to predict whether specific conditions will damage the crop yield or not.

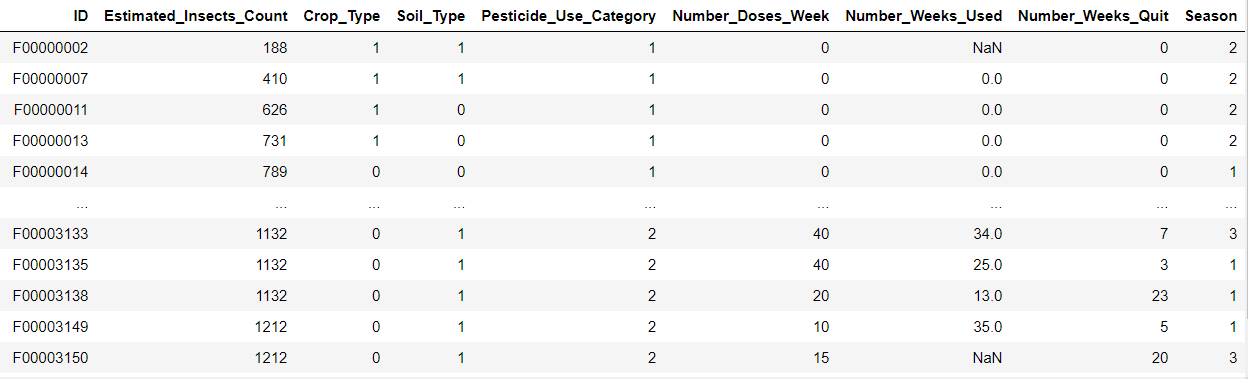
**2.1) Data Description**

First we import the xlsx files for both the test and train datasets. The train dataset has 4599 rows and 10 columns and the test dataset has 1199 rows and 9 columns.

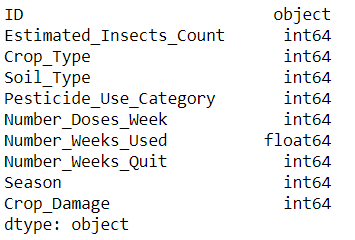








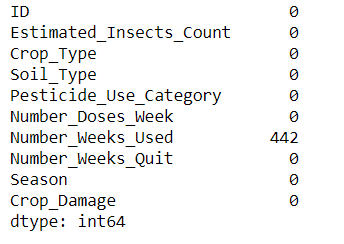
The datasets contain one categorical and several numerical features that give the details of pesticide usage leading to harvesting.



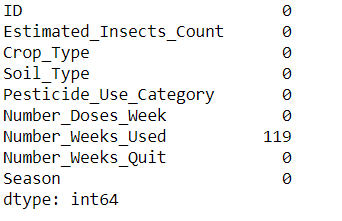
**2.2) Missing Values**

After importing the dataset, we need to check if it has any missing values.









‘Number\_Weeks\_Used’ column seems to have missing values in both the train and test dataset. Before performing any further analysis or visualizations on the dataset we need to fill these missing values.

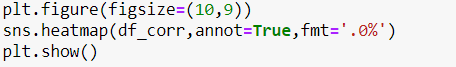
Since ‘Number\_Weeks\_Used’ column is of float64 datatype, we will impute the missing values with the mean of the entire row in the respective datasets.

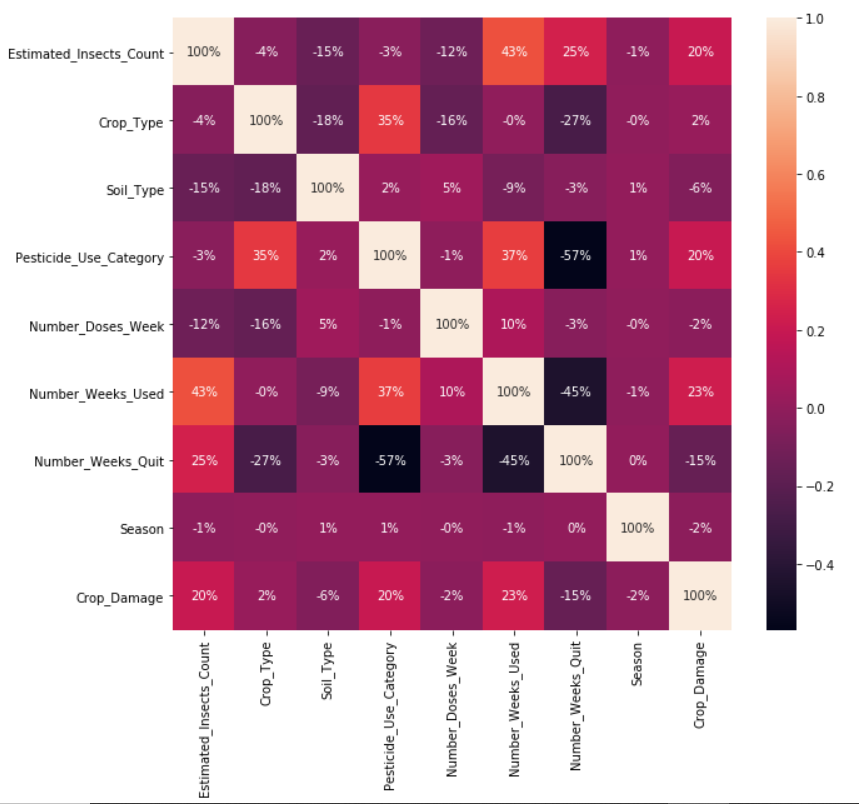




**2.3) Correlation**

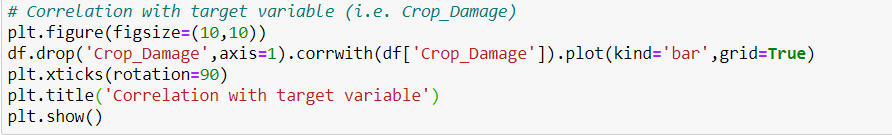
In this project, we are hoping to predict the crop damage in a yield, and in order to do that we need to know how the features are correlated to each other. We are plotting a seaborn heatmap of the correlation of the numeric features of that dataset to study the correlation between them.

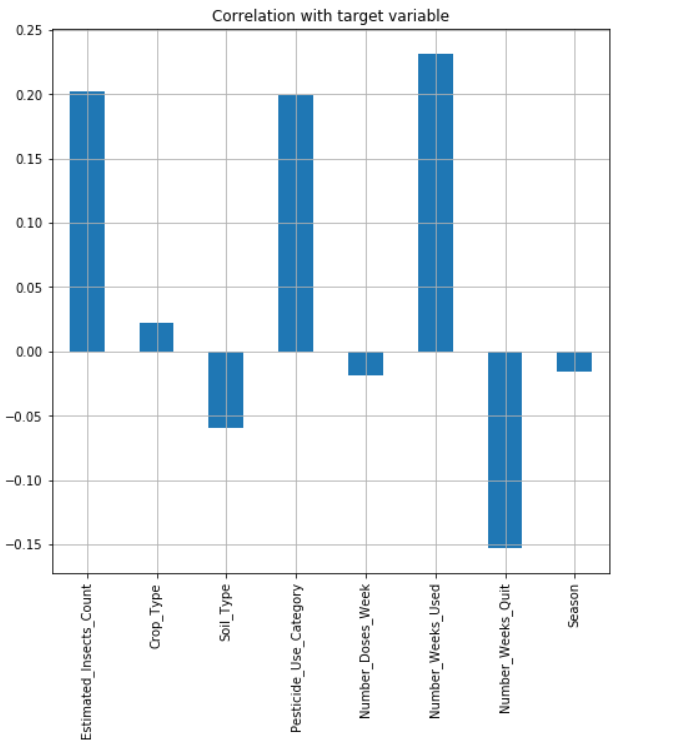




From the above heatmap, we can see that the strongest positively correlated columns are ‘Estimated\_Insect\_Count’ and ‘Number\_Weeks\_Used’ and the strongest negatively correlated columns are ‘Number\_Weeks\_Quit’ and ‘Pesticide\_Use\_Category’.

We also plot a corrwith plot of pandas DataFrame to compare the correlation of each column with the target variable ‘Crop\_Damage’. In this plot pairwise correlation is computed between rows or columns of DataFrame with rows or columns of Series or DataFrame. Before computing the correlations, DataFrames are aligned along both axes.



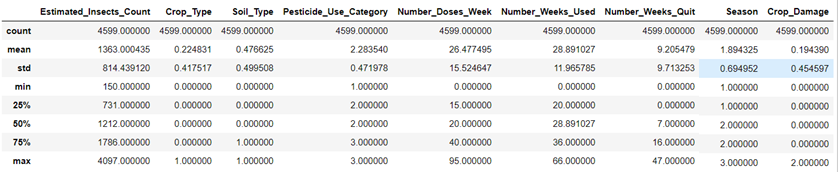


From the above plot, we can see that ‘Soil\_Type’, ‘Number\_Doses\_Week’, ‘Number\_Weeks\_Quit’ and ‘Season’ columns are negatively correlated to the target variable with ‘Number\_Weeks\_Quit’ having the strongest negative correlation. On the other hand, ‘Estimated\_Insect\_Count’, ‘Crop\_Type’, Pesticide\_Use\_Category’ and ‘Number\_Weeks\_Used’ columns are positively correlated to the target variable with ‘Number\_Weeks\_Used’ having the strongest positive correlation.

**2.4) Summary Statistics**

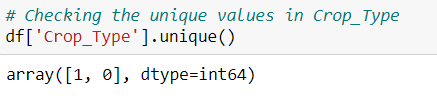
We perform describe() method of pandas DataFrame on the dataset generate descriptive statistics on the dataset. Descriptive statistics includes summarizing the central tendency, dispersion and shape of the datasets distribution.



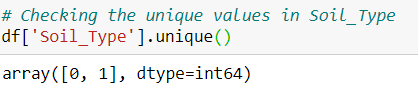


From this, we can see that the difference between 75% and max is higher in ‘Estimated\_Insect\_Count’, ‘Number\_Doses\_Week’, ‘Number\_Weeks\_Used’ and ‘Number\_Weeks\_Quit’, and hence outliers are likely to be present.

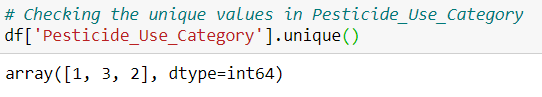
Next, we check the unique values in the columns.



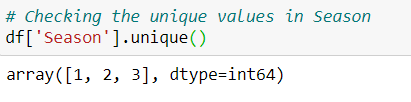
**The dataset includes two category of crops labeled as ‘0’ and ‘1.’**



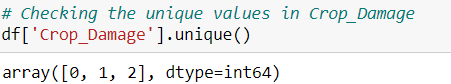
**The crops can be planted in two different categories of soil labeled as ‘0’ and ‘1’.**



**This column describes the use of pesticide in the field, labeled as ‘0’ (Never), ‘1’ (Previously Used), and ‘2’ (Currently Using).**



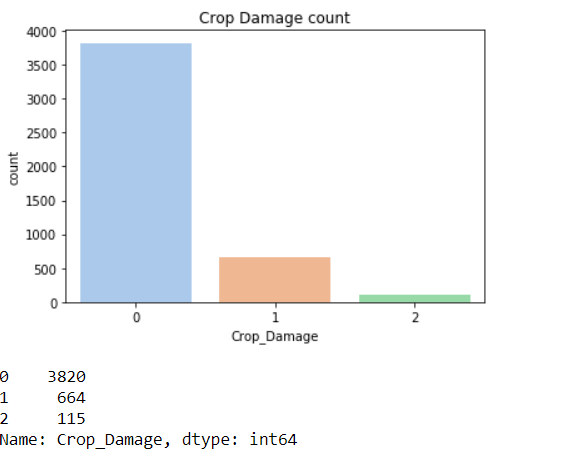
**This dataset describes the season of harvest for the crop, labeled as ‘1’, ‘2’ and ‘3’.**



**This column, i.e. the target variable, describes the damage sustained by the crops, labeled as ‘0’ (Alive), ‘1’ (Damage due to other causes) and ‘2’ (Damage due to pesticides).**

**2.5) EDA**

**2.5.1) Target Variable: Crop\_Damage**

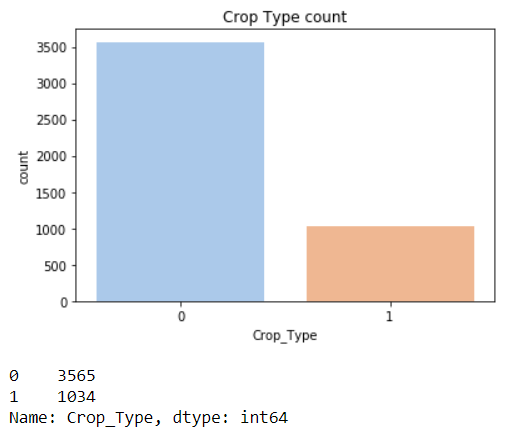


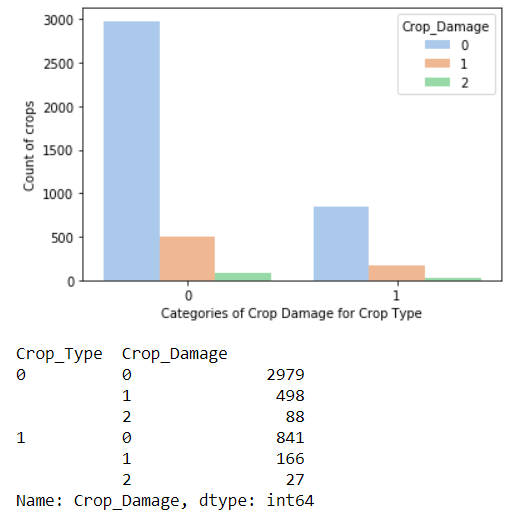
In this project, we are building a machine learning model to predict the values of ‘Crop\_Damage’, i.e. whether a crop has sustained damage due to pesticides or any other factors or not.

In this dataset, 3820 crops out of 4599 are alive, 115 are damaged due to pesticides and 664 are damaged due to other causes. From the plot we can see that the dataset is heavily imbalanced, and hence we will have to make sure we split the dataset properly.

**2.5.2) Crop\_Type**

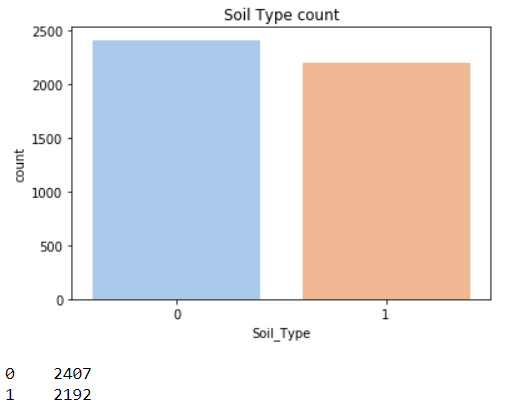
Out of 4599 crops, 3565 are of category ‘0’ and 1034 are of category ‘1’. Out of the 3565 crops of category ‘0’, 83.56% are alive, 2.47% are damaged due to pesticides and 13.97% are damaged due to other causes. On the other hand for crops of category ‘1’, 81.33% are alive, 2.61% are damaged due to pesticides and 16.06% are damaged due to other causes. Crops of category ‘0’ are slightly less likely to be damaged than crops of category ‘1’.

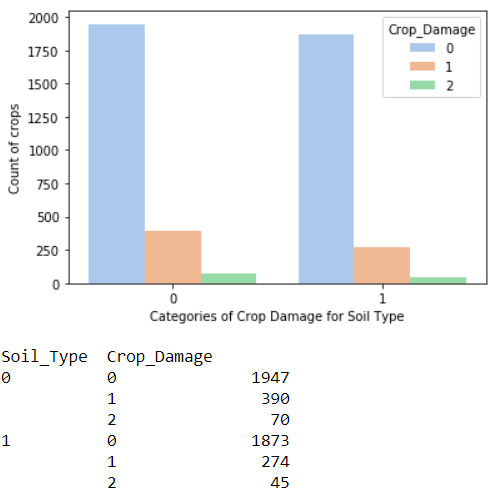




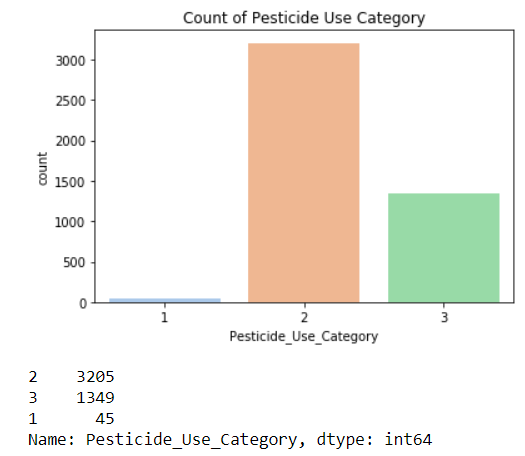
**2.5.3) Soil\_Type**

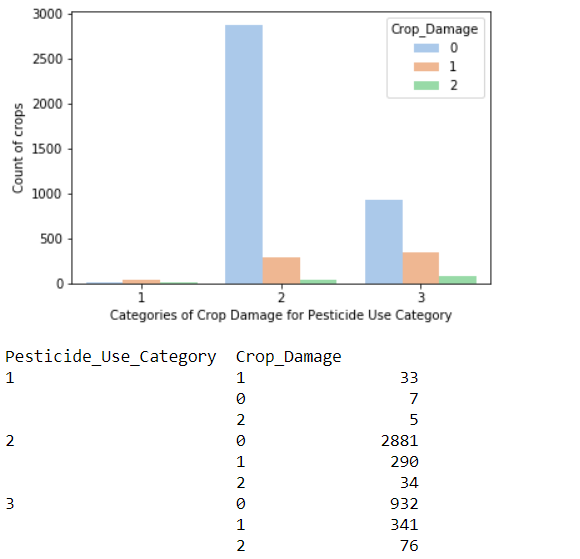
Out of 4599 crops, 2407 were planted in soil of category ‘0’ and 1034 2192 were planted in soil of category ‘1’. Out of the 2407, 80.89% are alive, 2.91% are damaged due to pesticides and 16.2% are damaged due to other causes. For crops planted in soil of category ‘1’, 85.45% are alive, 2.05% are damaged due to pesticides and 12.5% are damaged due to other causes. Therefore, we can say that crops planted in soil type ‘1’ are more likely to be alive than crops planted in soil type ‘0’.





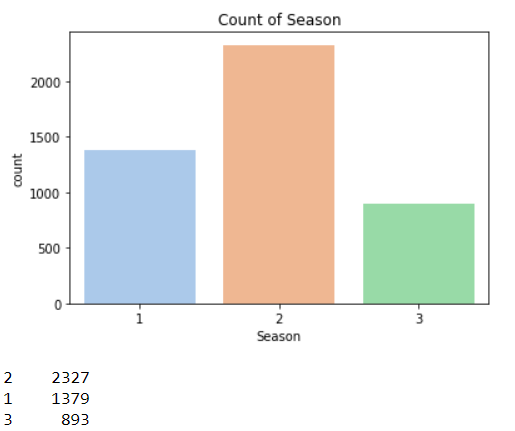
**2.5.4) Pesticide\_Use\_Category**

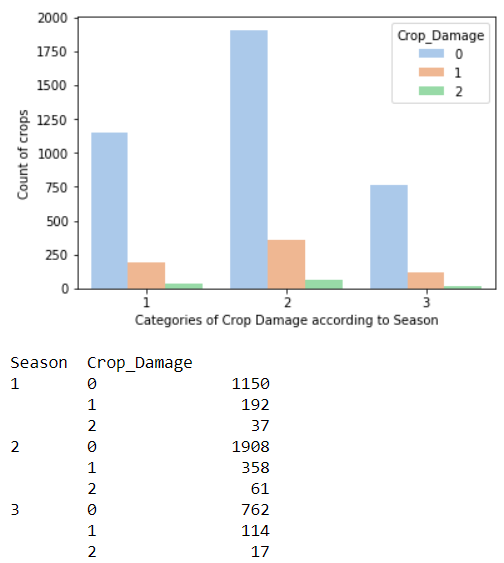




Out of the 4599 crops, 45 have never been treated by pesticides, 1349 have previously been treated by pesticides and 1349 are currently being treated by pesticides. Of those that have never been treated by pesticides, 15.56% are alive and 84.44% have been damaged. Of those that have been treated by pesticides previously, 89.89% are alive, 1.06% are damaged by pesticide and 9.05% are damaged due to other causes. And of the crops that are currently being treated by pesticides, 69.09% are alive, 5.635% are damaged due to pesticides and 25.28% are damaged due to other causes.

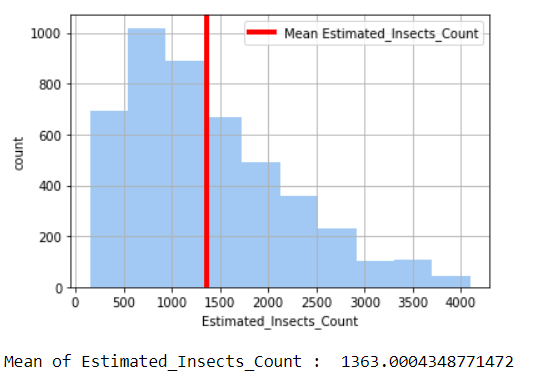
**2.5.5) Season**

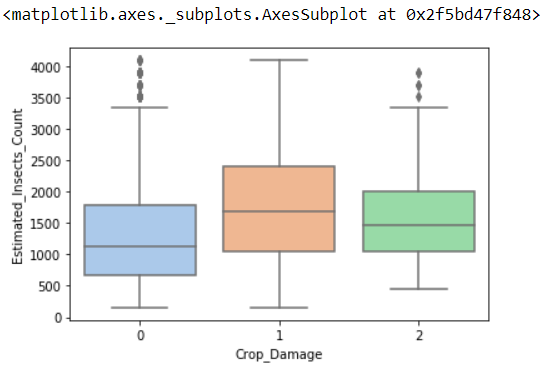




The crops listed in the dataset are harvested in different seasons that are labeled by ‘1’, ‘2’ and ‘3’. Of 4599, 1379 crops are harvest in season ‘1’, out of which 83.4% are alive, 2.68% are damaged by pesticides and 13.92% are damaged by other factors. 2327 are harvested during season ‘2’, of which 82% are alive, 2.62 %are damaged due to pesticides and 15.38% are damaged due to other causes. The remaining 893corps are harvest during season ‘3’. Of the 893 remaining, 85.33% are alive and 14.67% are damaged. From this we can say that season of harvest affects the damage done to the crops even if on a small scale.

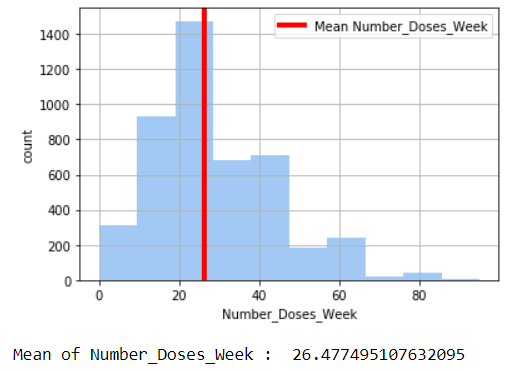
**2.5.6) Estimated\_Insect\_Count**

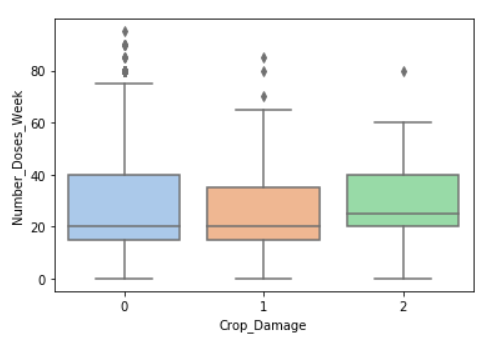




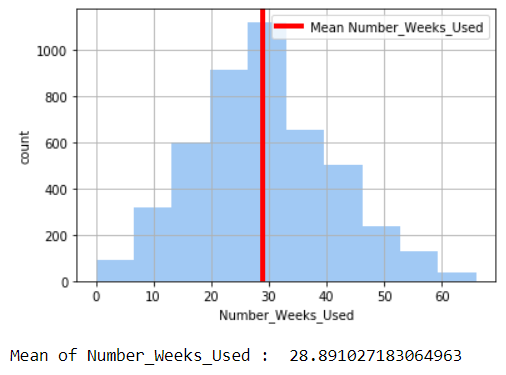
Estimaed\_Insects\_Count column is right skewed. The mean estimated count of insects in the field is 1363. From the boxplot we can see that there are outliers in the column. The average estimated insect count for crops that are alive is the lowest and for crops that are damaged due to factors other than pesticides is the highest. Hence, we can say that a low estimated insect count will most likely mean that the crops will be in better health.

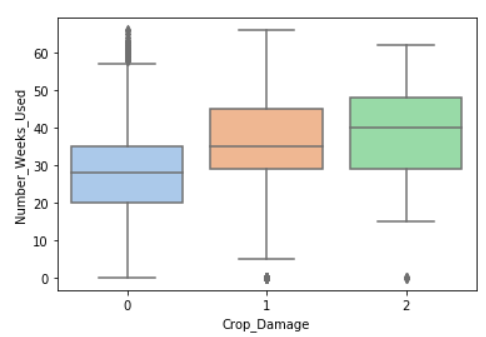
**2.5.7) Number\_Doses\_Week  
Number\_Doses\_Week column is slightly right skewed. On an average approximately 26 doses of pesticide are given every week. The column also has a few outliers. From the boxplot we can see that the average number of doses of pesticides per week is lower and almost for crops that are either alive or have been damaged by factors other than pesticides. But, for crops that have been damaged by pesticides the number of doses per week is higher. Hence, we can say that if the number of doses per week are high, than the crops are more likely to get damaged due to the pesticides.**





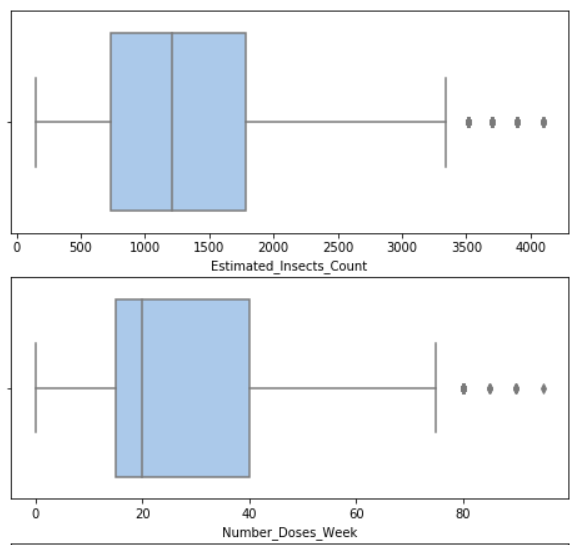
**2.5.8) Number\_Weeks\_Used**

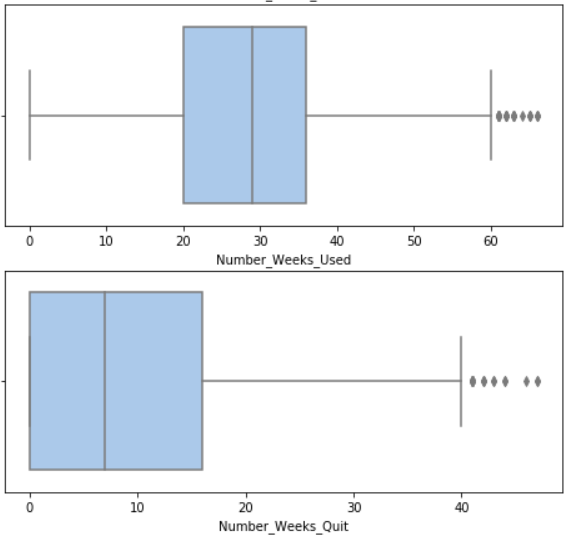




Number\_Weeks\_Used column has a normal distribution. On an average pesticides were used for roughly 29 weeks. From the boxplot we can see that the column has some outliers. For crops that are alive the average number of weeks the pesticides were used is least. On the other hand crops that have been damaged due to pesticides have been treated by pesticides the longest. Hence, we can say that the crops are more likely to be damaged longer the pesticides are used.

**2.6) Outliers**





We use boxplots to plot the outliers in the numerical columns of the dataset. From the above plots we can see that ‘Estimaed\_Insects\_Count’, ‘Number\_Doses\_Week’, ‘Number\_Weeks\_Used’ and ‘Number\_Weeks\_Quit’ columns have outliers. We will have to treat these outliers before building the predictive model.

# EDA Concluding Remarks

* The dataset has features of int64, float64 and object datatype.
* ‘Number\_Weeks\_Quit’ has the strongest negative correlation with the target variable ‘Crop\_Damage’.
* ‘Number\_Weeks\_Used’ has the strongest positive correlation with the target variable ‘Crop\_Damage’.
* The dataset is heavily imbalanced with majority of the crops being alive and with very few of them being damaged by pesticides.
* Most of the crops are of type ‘0’ and only about one-fourth are of type ‘1’.
* Crops planted in soil type ‘0’ are more likely to be damaged by pesticides than the crops planted in soil type ‘1’.
* Crops that have never been treated by pesticides are most likely to be damaged and the crops that have previously been treated by pesticides are the most likely to be alive. On the other hand, crops that are currently being treated by pesticides are not as likely to be damages as the ones that have been never treated by pesticides, but they are more likely to sustain damage than then ones that have been previously treated.
* Crops that are harvested in season ‘3’ are least likely to be damaged by pesticides, while on the other hand, crops harvested in season ‘2’ are most likely to sustain damage from factors other than pesticides.
* Crops in fields that have low estimated insect count, are most likely to be alive and higher the estimated insect count more the crops are getting damaged.
* Crops that have been treated with pesticides for an average of 20 doses per week are more likely to be alive or to be damaged by other diseases. On the other hand, crops that have been treated with pesticides for more than an average of 20 doses per week are more likely to be damaged by pesticides.
* Crops that have been treated by pesticides for an average of around 30 weeks are most likely to be alive, while crops that have been treated for longer than an average of 40 weeks are most likely to be damaged by pesticides.

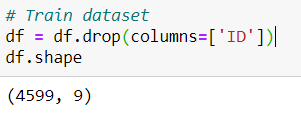
# Pre-processing Pipeline

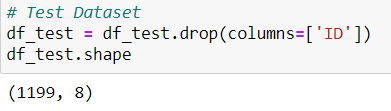
We have to pre-process the data before feeding it to a machine learning algorithm to build a model that will predict crop damage with maximum accuracy.

**4.1) Dropping Columns**

To make the model more accurate we drop ID column that gives us nothing but the unique identification number of the crop which does not affect whether the crop will sustain damage or not.

Since we have 2 datasets, i.e. train and test we have to drop the column in both the datasets.

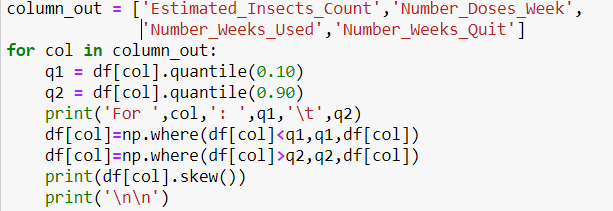


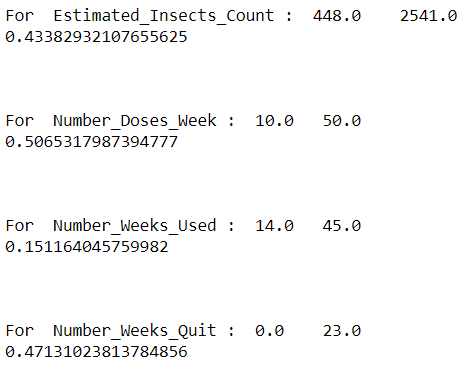


**4.2) Treating Outliers**

To make our model more accurate we have to make sure our data does not have any outliers. Since, we already know we have outliers; we have to treat them before feeding the data into the model.

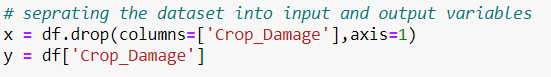
To treat outliers we replace them with the quantile values.





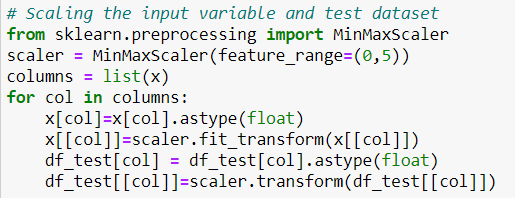
**4.3) Splitting the Dataset into Input and Output Variables**

Before scaling the dataset we have to split the train dataset into input and output variables. We only scale the input variables.



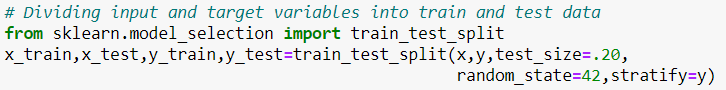
**4.4) Feature Scaling**

Before applying any machine learning algorithm on the dataset, we need to scale the data. Here, we use MinMaxScalar to scale the input parameters. MinMaxScalar transforms the features by scaling each feature to a given range. The estimator scales and translates each feature individually such that it is in the given range. For this dataset, we scale the features in the range 0 to 5. For the train dataset we do both fit and transform, but to scale the test dataset, we do not fit but only transform.



**4.5) Splitting the Dataset into Test and Train**

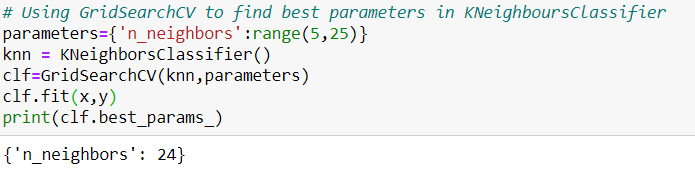
Before building the model we split the train dataset into test and train data. We split it in the ratio 80:20. We use the train data to train the machine learning model which will then predict the results for the test data as well as for the test dataset. The test data will only be used to make predictions and evaluate the model performance in order to selct the best model. We use stratify on y to make sure the y values are evenly split.

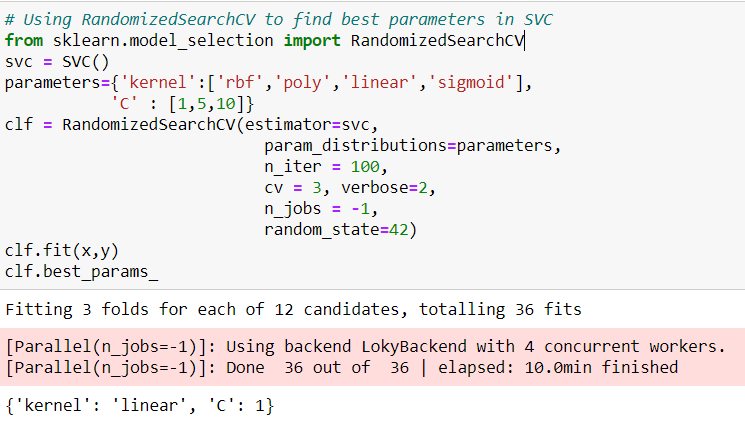


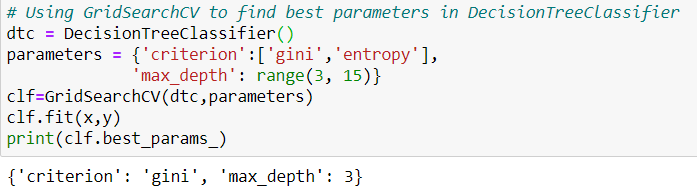
# Building Machine Learning Model

**5.1) Hyperparameter Tuning**

Before running the algorithms, we perform hyperparameter tuning for ‘KNeighboursClassifier’, ‘SVC’ and ‘DecisionTreeClassifier’. We use GridSearchCV and RandomizedSearchCV for tuning the parameters.

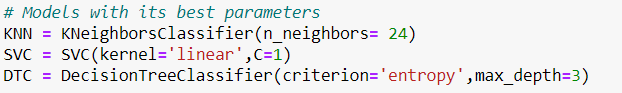


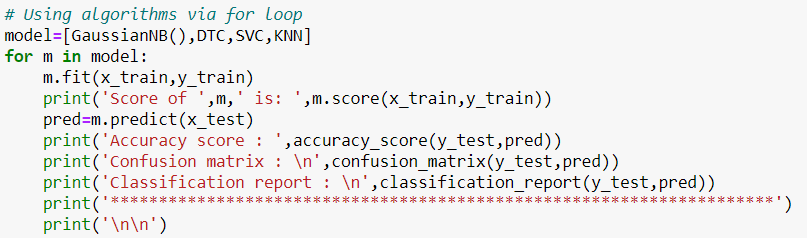




**5.2) Using the Models with the Best Parameters**

On hyperparameter tuning we get the best parameters for the models. We then use the model with these parameters. We use a ‘for’ loop to run all the algorithms. We print out the accuracy, confusion matrix and classification report for each model.



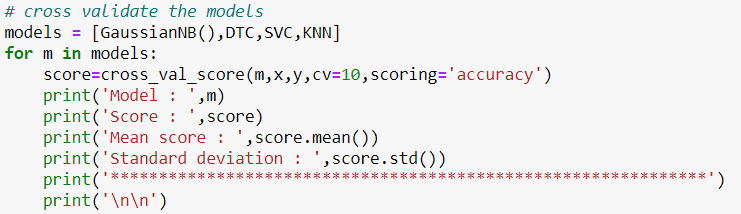


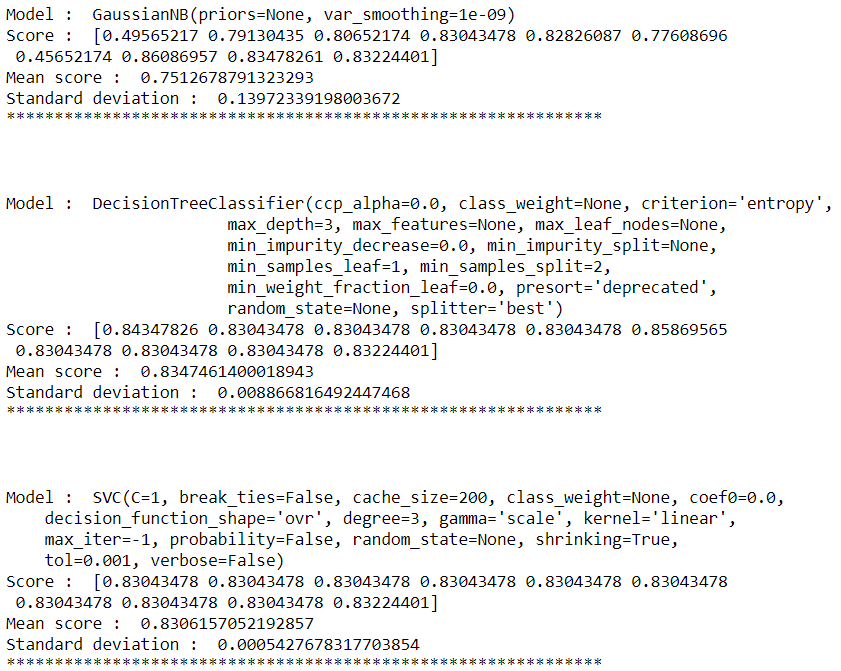
GaussianNB gives an accuracy score of 0.81, DecisionTreeClassifier gives a score of 0.835, SVC gives a score of 0.83 and KNeighboursClassifier gives a score of 0.826.

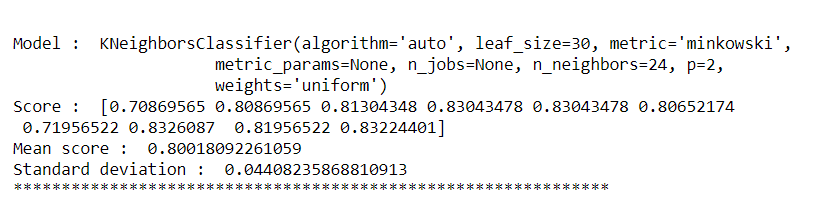
**5.3) Cross validation**

After building the model we cross validate the model to figure out whether the model is overfitting or underfitting. We use the cross\_val\_score() method from the model\_selection library to cross validate the model.

We use a ‘for’ loop to perform cross validation on all the models at once.



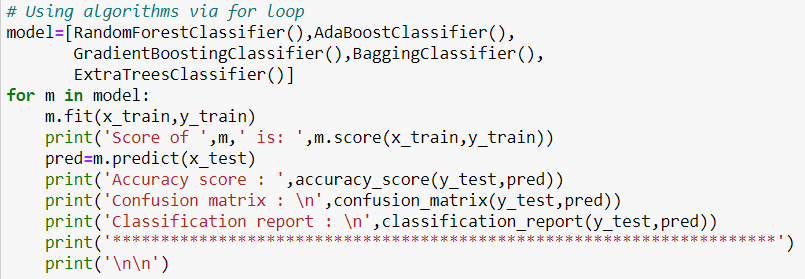




From this we can see that DecisionTreeClassifier shows the least difference in the cross validation score and the accuracy score along with the best accuracy score. Hence, , DecisionTreeClassifier is the best fit among the algorithms tested.

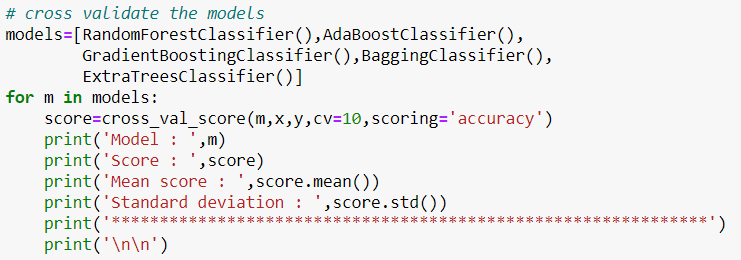
**5.4) Ensemble Models**

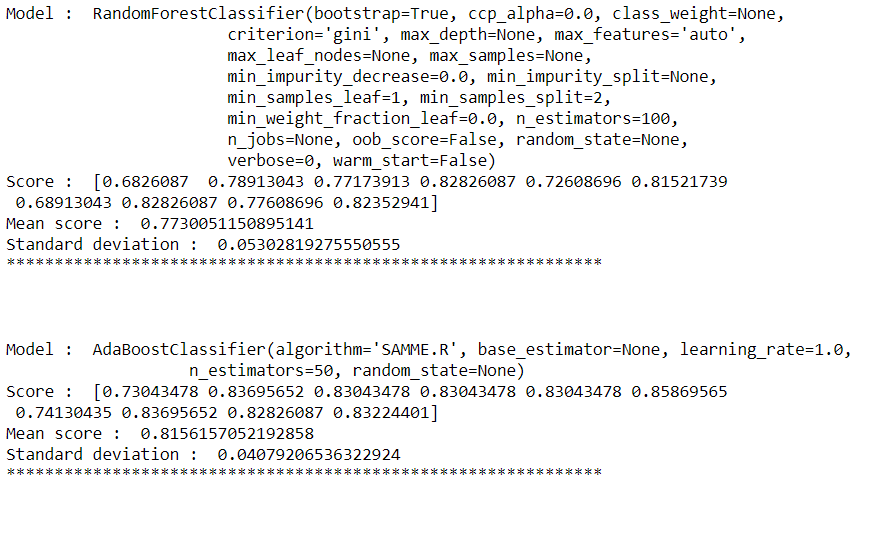
We use several ensemble models on the dataset in hopes of getter better scores. We use ‘RandomForestClassifier’, ‘AdaBoostClassifier’, ‘GradientBoostingClassifier’, ‘BaggingClassifier’ and ‘ExtraTreesClassifier’ models. We use a ‘for’ loop to run all the models at once.

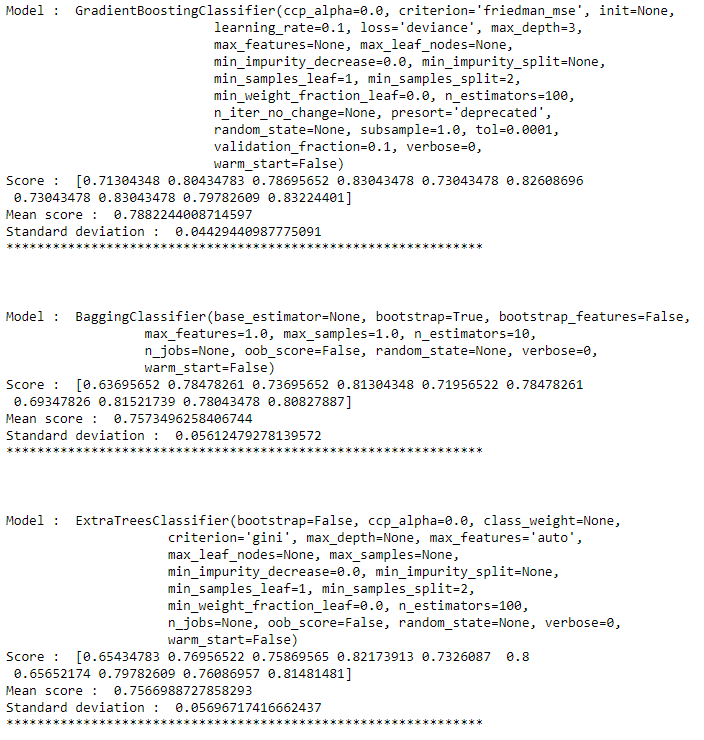


RandomForestClassifier gives an accuracy of 0.82, AdaBoostClassifier give a score of 0.838, GradientBoostingClassifier gives a score of 0.829, BaggingClassifier gives a score of 0.809 and ExtraTreesClassifier gives a score of 0.809.

We then perform cross validation on these models.



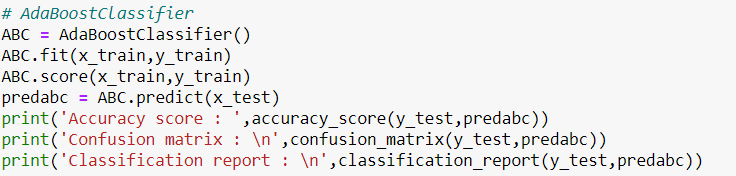




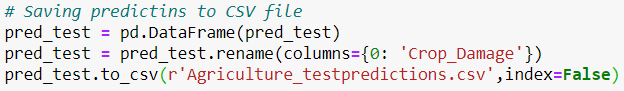
From the above scores we can see that AdaBoostClassifier gives the best results for the dataset.

**5.5) Predicting for Test Dataset**

After testing various models on the dataset, we find that the AdaBoostClassifier model gives the best fit for this dataset. Therefore, we use AdaBoostClassifier as our final model and perform predictions on the test dataset. We then save these predictions to a csv file.







**5.6) Saving**

After getting the model with the best fit, we save the model as a pickle file using the joblib method of externals library.



# Concluding Remarks

As new entries are added to the dataset and the dataset grows, we can retrain the model with the new data to get more accurate predictions. On getting better predictions we can move on to further classify individual factors that cause damage to the crops and not just pesticides. We can further classify the damage as damage due to insects, damage due to rodents and so on. This data can also be used to plan a proper harvesting schedule making sure not to overuse pesticides and other chemicals.

The most relevant features that cause crop damage are:

1. **Soil\_Type: Crops planted in soil type ‘0’ are more likely to get damaged by pesticides then the crops planted in soil type ‘1’. Therefore, the farmers can increase the amount of healthy yield if they make use of soil type ‘1’ in their fields.**
2. Pesticide\_Use\_Category: Never using pesticides causes immense damage to the crops. On the other hand, overusing pesticides also causes damage to the crops. Hence, to get the best possible yield the farmers have to find a balance between overusing and not using pesticides at all.
3. Estimaed\_Insects\_Count: Higher the estimated insect count, higher is the possibility of crops getting damaged. Hence, to avoid damage to the greatest extent possible the farmers have to take measures to keep insects away from the fields. The framers can do this by making sure there are no stagnant water puddles near the fields, or keep LED bulbs lit up on the field to keep the insects away.
4. Number\_Doses\_Week: When the farmer uses more than an average of 20 doses of pesticides per week the pesticides start causing damage to the crops. Therefore, the farmer has to avoid overusing of pesticides to guarantee a healthy yield.
5. Number\_Weeks\_Used: Using pesticides for more than an average of 30 weeks causes more harm than good. The farmer can avoid this by keeping a proper timetable for using pesticides.

In addition to the above listed suggestions, the farmer can use organic measures in place of chemical products.