**Machine Learning in Agriculture**

Dataset Link: [https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects/blob/master/train\_agriculture.xlsx](http://archive.ics.uci.edu/ml/datasets/Census+Income)

**Problem Defination :** We need to determine the outcome of the harvest season, i.e. whether the crop would be healthy (alive), damaged by pesticides or damaged by other reasons.

**Attribute Information:**



**Target/Dependant Variable :**

**Crop\_Damage:**0,1,2

By looking at target variable it is obvious that we need to use classifier machine learning algorithm for our solution .

Before proceeding furter, first we need to load the data file into dataframe .

After loading the dataset we can see there are 10 columns in datarame with mix data type and 88858 no of rows .



Here in dataset we have ID column which is representing an identifier to row which is not useful in our analysis .So we can drop this column .

**EDA Process:**



Here From above snap we can see there are 9000 missing values present in Number\_Weeks\_Used column .Lets visualize this through heatmap .



From above heatmap we can see there are few missing vaue presents in our dataset .

So it is required to remove those missing value before processding further .

I have dropped Number\_Weeks\_Used column as it contains only missing values .

After removing the missing values lets check heatmap again .



From this above heatmap we got a proof that we do not have any missing values present in our dataset .



Aboove graph tells that if insects count is less then chance of crop alive is more .

Lets find correlation between columns .



We found below observation from above heatmap.

1.Number\_Weeks\_Used is highly -vely correlated with Number\_Weeks\_Quit

2.Number\_Weeks\_Used is highly +vely corelated with Estimated\_Insects\_count

So lets remove remove Number\_Weeks\_Quit feature .

**Data Cleaning:**

Various methods are available to improve the obtained dataset by removing redundant values and handle the missing values. In this dataset, we have 8 attributes each contributing to the result . Lets find skewness and remove skewness if it is > 0.55.

I did not remove outlier as it is removing class/category 2 values of target variable .



Our final dataset looks like below



We have values for Crop\_Damage category 0,1 and 2 .

0 66743

1 11059

2 2056

**Split feature and target variable:**

x=df.drop('Crop\_Damage',axis=1)

y=df['Crop\_Damage']

**Standardize input data:**

Standardization comes into picture when features of input data set have large differences between their ranges, or simply when they are measured in different measurement units (e.g., Pounds, Meters, Miles … etc).

These differences in the ranges of initial features causes trouble to many machine learning models. For example, for the models that are based on distance computation, if one of the features has a broad range of values, the distance will be governed by this particular feature.

To illustrate this with an example : say we have a 2-dimensional data set with two features, Height in Meters and Weight in Pounds, that range respectively from [1 to 2] Meters and [10 to 200] Pounds. No matter what distance based model you perform on this data set, the Weight feature will dominate over the Height feature and will have more contribution to the distance computation, just because it has bigger values compared to the Height. So, to prevent this problem, transforming features to comparable scales using standardization is the solution.

After standardisation data looks like below .



Now we will use classification algorithm for our prediction .

I have used almost all classifier algorithm like Logistic Regression,DecissionTree Classifier,RandomForest Classifier,GaussianNB,KNeighbors Classifier etc .

After testing with all above model ,I have printed all model score in below section .

**Lets find best model from below matrix :**



**Observation :**

The results show that Gradient Boost Classifier and SVC models are very good for accurately predicting the data with 84.71% and 84.41% accuracy.