Grape Leaf Disease Identification using Machine Learning Techniques

Introduction:

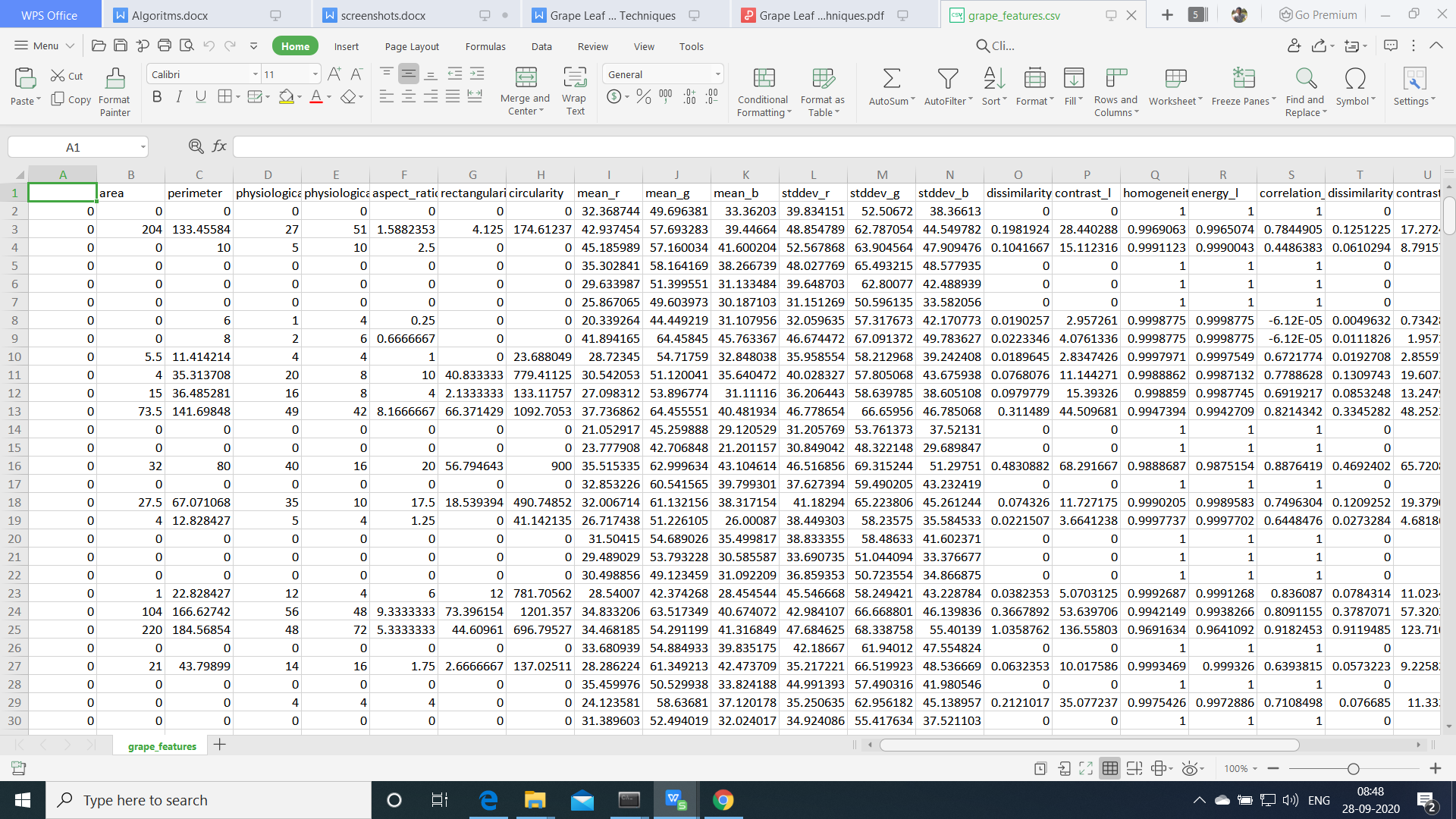
Indian Economy is highly dependent on agricultural productivity of the country. Grape is very commercial fruit of India. It can easily be grown in all tropical, sub-tropical and temperate climatic regions. India has got different types of climate and soil in different parts of the country. This makes grapevines a major vegetative propagated crop with high socioeconomic importance. The grape plant will cause poor yield and growth when affected by diseases. The diseases are due to the viral, bacteria and fungi infections which are caused by insects,rust and nematodes etc., These diseases are judged by the farmers through their experience or with the help of experts through naked eye observation which is not accurate and time consuming process. Early detection of disease is then very much needed in the agriculture and horticulture fifield to increase the yield of the crops. We have proposed a system that can detect and identify diseases in the leaves of the grape plants.

Data Description:

For grape leaf disease prediction we take grape features datain csv format. This dataset consists twenty five columns and 4089 records. Out of twenty five columns twenty three columns are attributes and last column is class. The class column consists of either Black\_rot or Esca\_(Black\_Measles).

**Github location**

Contains 4 categories



**Algorithms:**

**Support Vector Machine**

“Support Vector Machine” (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle" \t "https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/_blank) which can be used for both classification or regression challenges. However,  it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_1.png)

Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

**Logistic regression**

Logistic regression is named for the function used at the core of the method, the logistic function.

The [logistic function](https://en.wikipedia.org/wiki/Logistic_function), also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

1 / (1 + e^-value)

Where e is the [base of the natural logarithms](https://en.wikipedia.org/wiki/E_(mathematical_constant)) (Euler’s number or the EXP() function in your spreadsheet) and value is the actual numerical value that you want to transform. Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.



Logistic Function

Now that we know what the logistic function is, let’s see how it is used in logistic regression.

# Naive Bayes Classifiers

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) \* P(d2|H) and so on.

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

# Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**Decision tree Algorithm:**

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree,testing the attribute specified by this node,then moving down the tree branch corresponding to the value of the attribute as shown in the above figure.This process is then repeated for the subtree rooted at the new node

## **AdaBoost Classifier**

Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set. Adaboost should meet two conditions:

1. The classifier should be trained interactively on various weighed training examples.
2. In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.

**Steps for Machine Learning Algorithms**

1. Install Anaconda Latest Version
2. Open anaconda Prompt
3. Conda create -n tf python=3.7
4. Conda activate tf
5. Install require softwares

scikit-image==0.17.2

scikit-learn==0.23.2

pandas==1.1.1

matplotlib==3.3.1

Pillow==7.2.0

plotly==4.10.0

opencv-python==4.4.0.42

spacy==2.3.2

lightgbm==3.0.0

mahotas==1.4.11

matplotlib==3.3.1lightgbm==3.0.0

mahotas==1.4.11

nltk==3.5

matplotlib==3.3.1

xgboost==1.2.0

Jupyter

1. Activate environment for jupyter notebook(For execute the in jupter notebook)

python -m ipykernel install --user --name=

1. Goto project Directory

Note: For Text related project. Need to Download

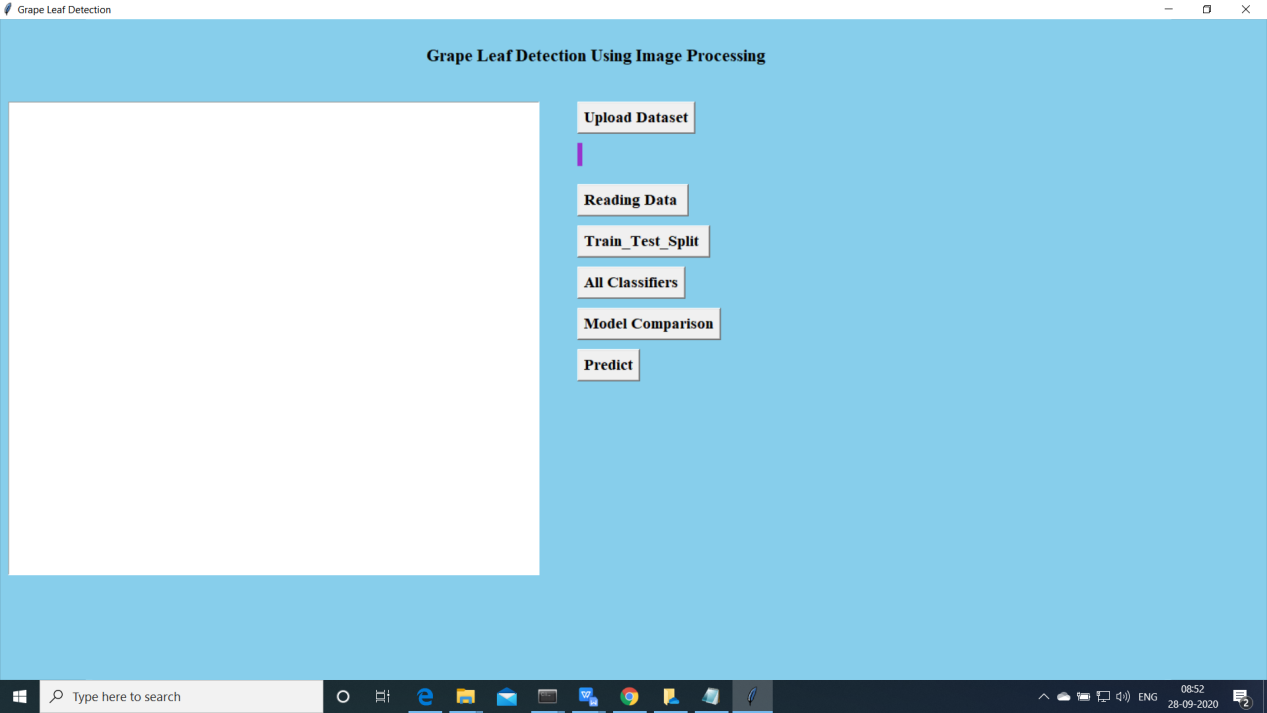
1. Open anaconda Prompt
2. Python
3. Import nltk
4. Nltk.download()

**Project Development Modules:**

1. **Data Collection:**Collect sufficient data samples and legitimate software samples. 
2. **Feature Extraction**:For each image extract the features ubsing image processing and save in ‘.csv’ extension 
3. **Train and Test Modelling: Split the data into train and test data Train will be used for trainging the model and Test data to check the performace**
4. **Modelling:** SVM Navie bayes, Random FOrest,KNN,Ada boost, Decision tree, Ada boost with randomforest . Combine the training using machine learning algorithms and establish a classification model.

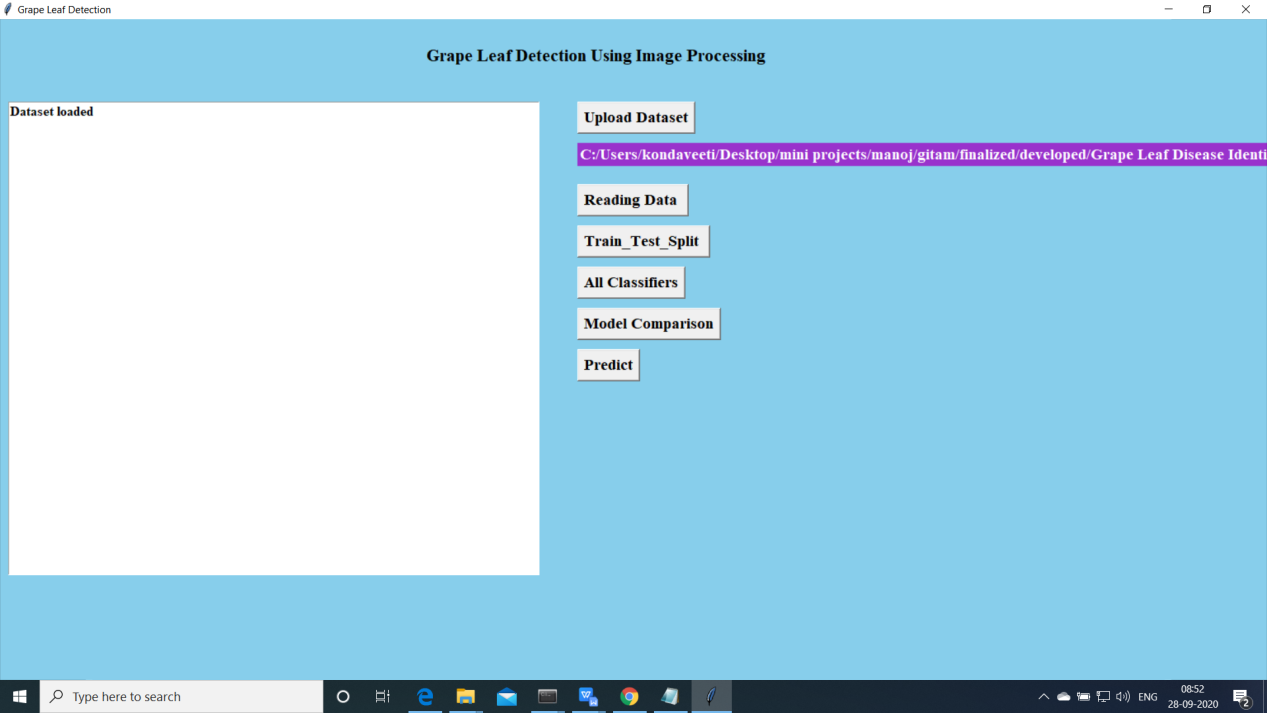
**Execution Steps:**

1. Open anaconda Prompt
2. Conda activate tf
3. Goto Project Directory
4. Python featureextract.py
5. Python final.py

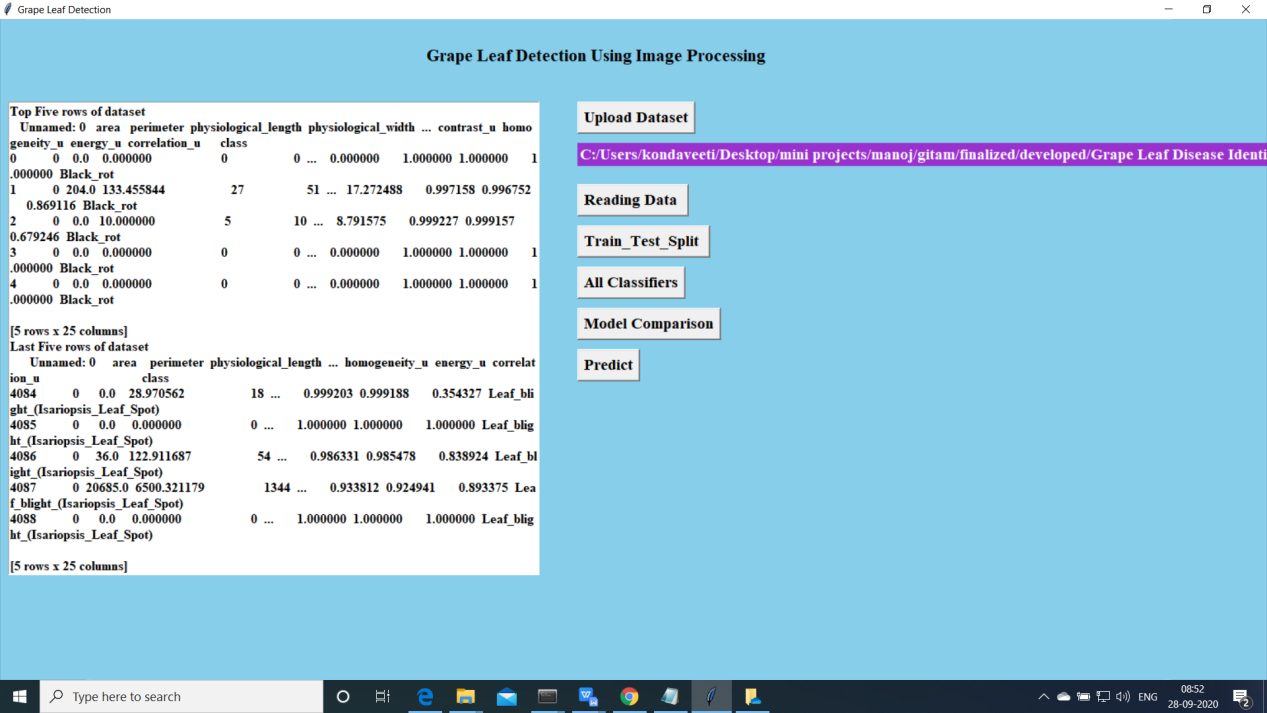


Above screen will be opened.

1. Now click on “Upload data ”

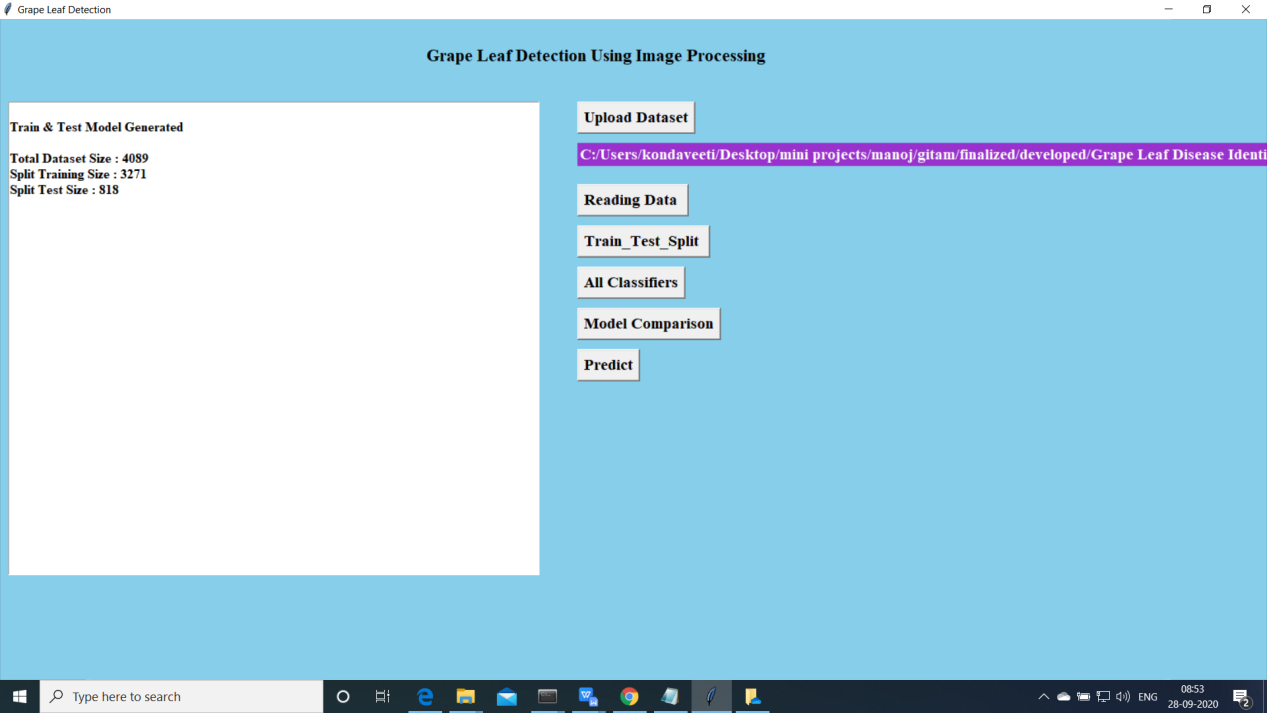


1. Import the data and Preprocee

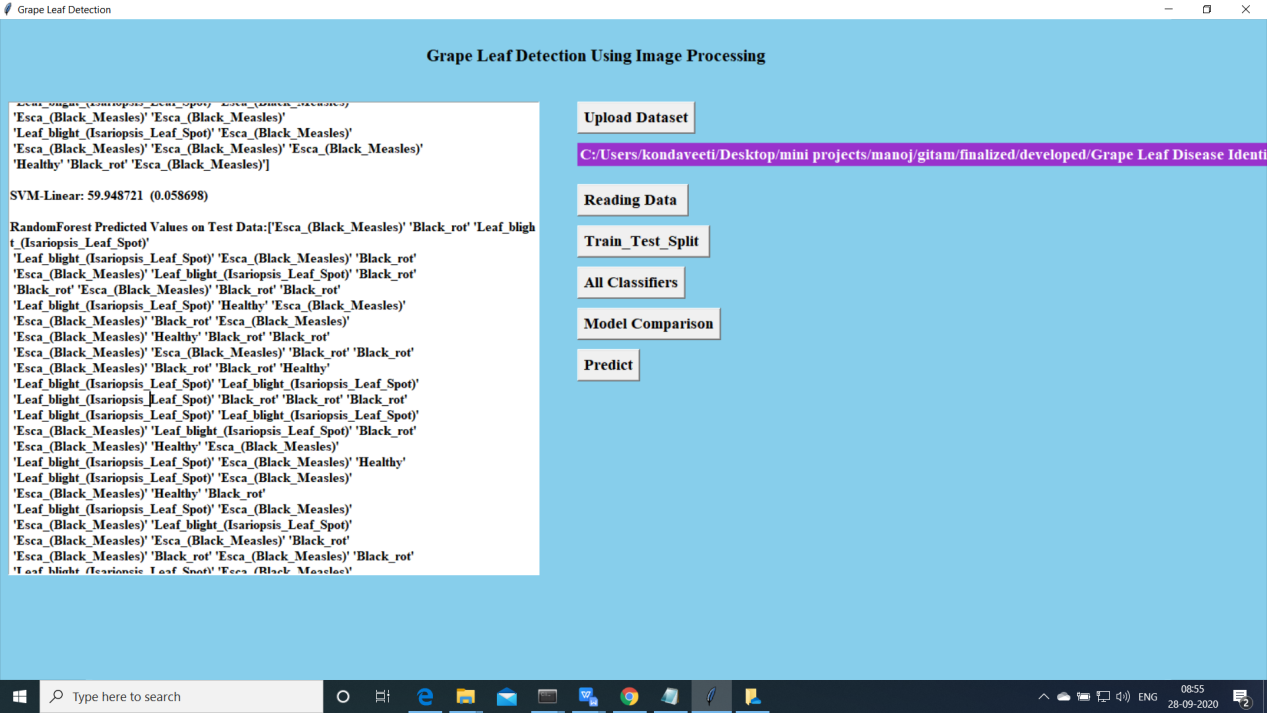


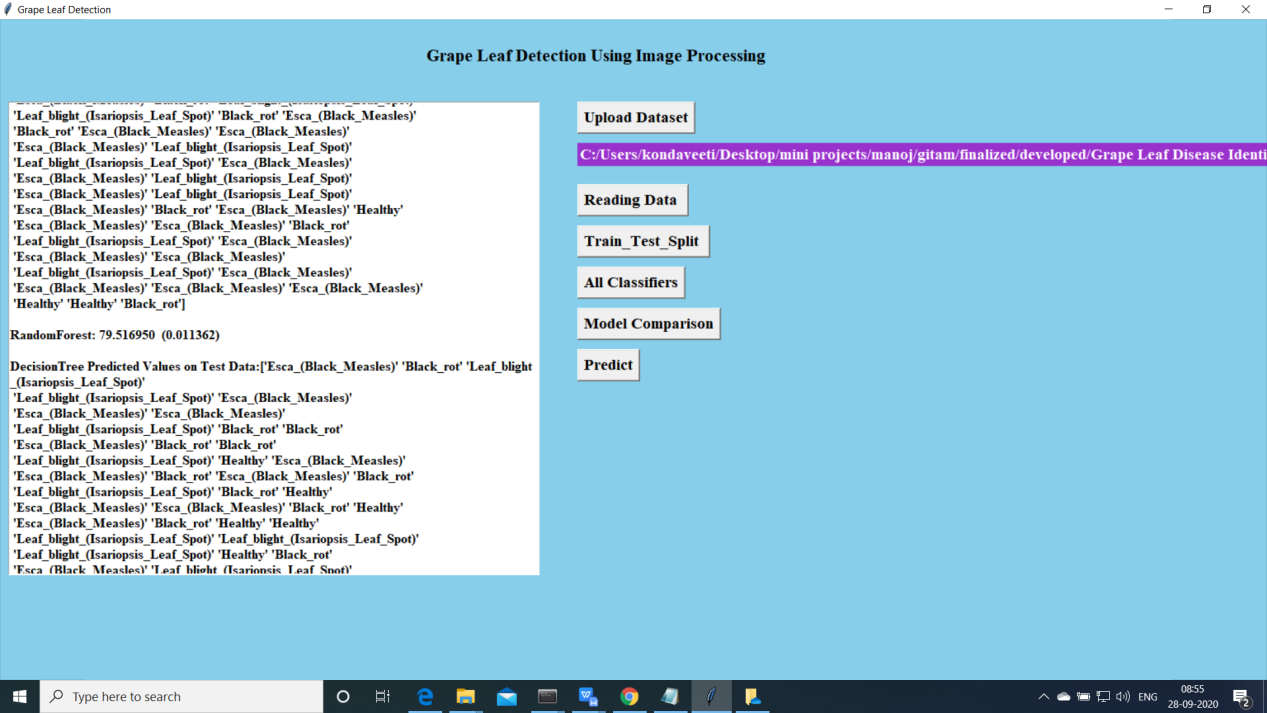
Upload the data and read the basic data information will be shown on the screen

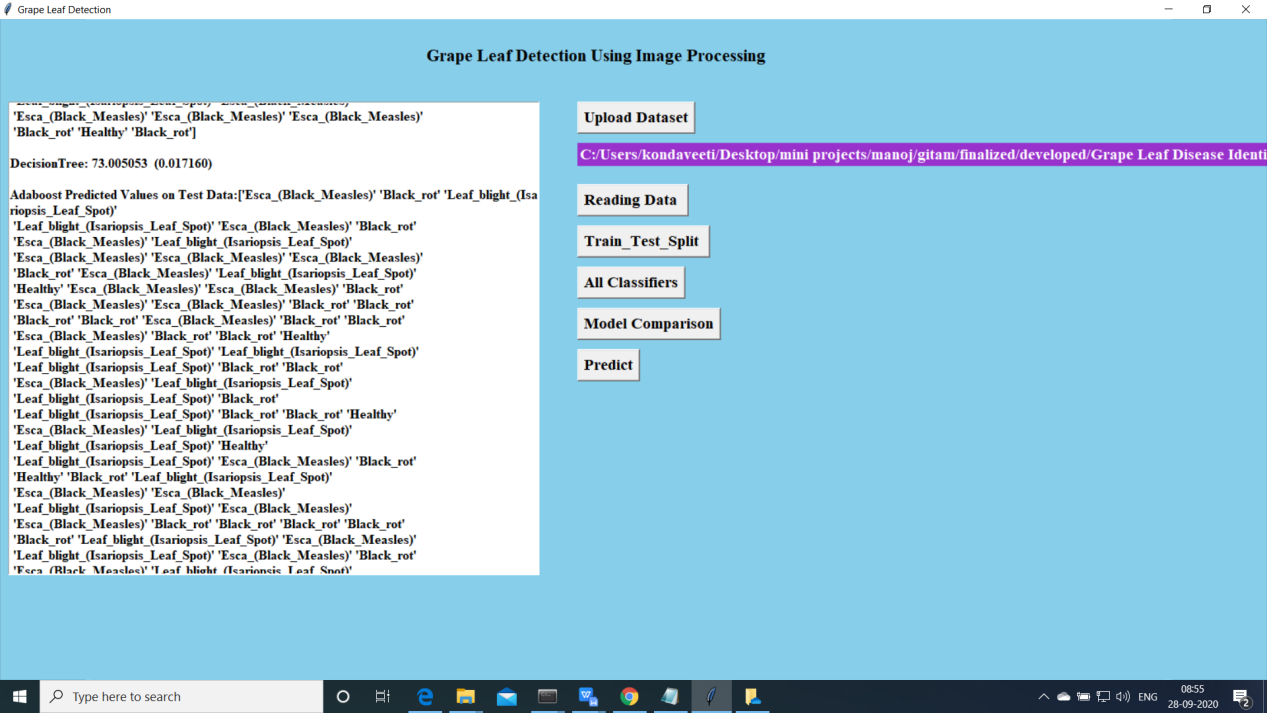
1. Now click on “Train and Test model”. split the data into train and test and traain will be used for training and to tets the performace we are using test data

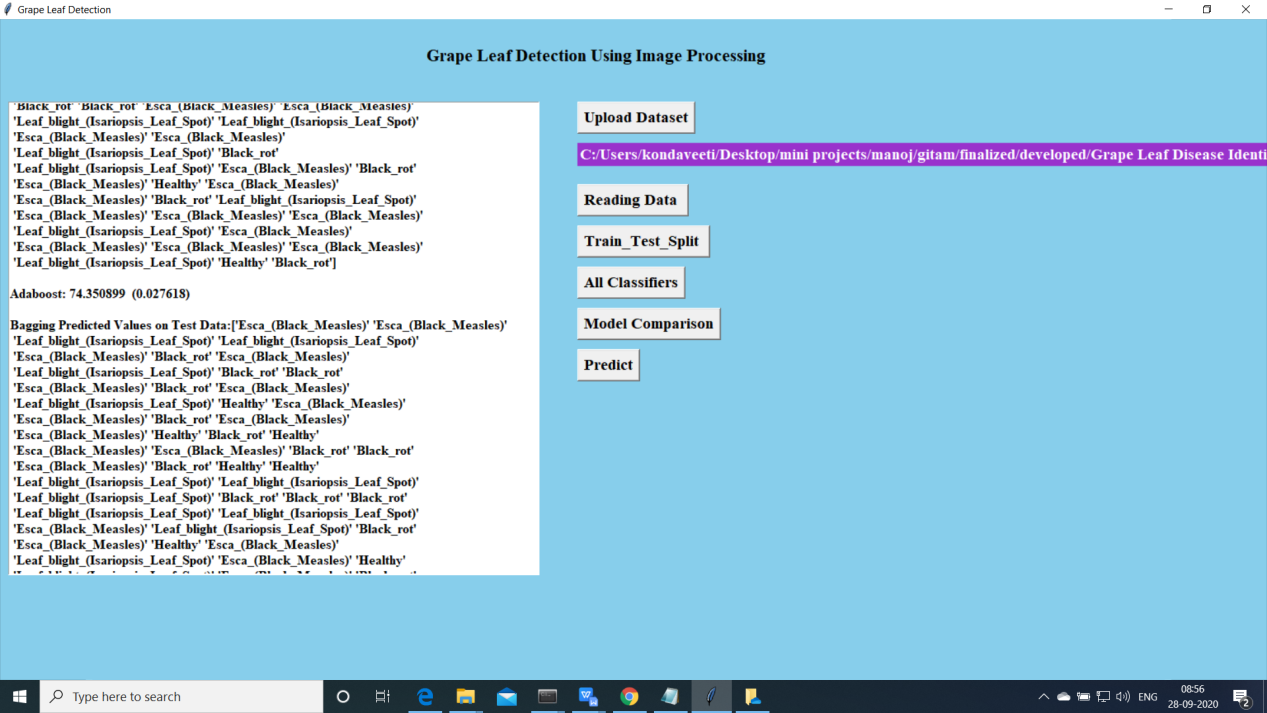


1. Now click on “Run Algoruimns”. Mentioned algorithms will be run on the data



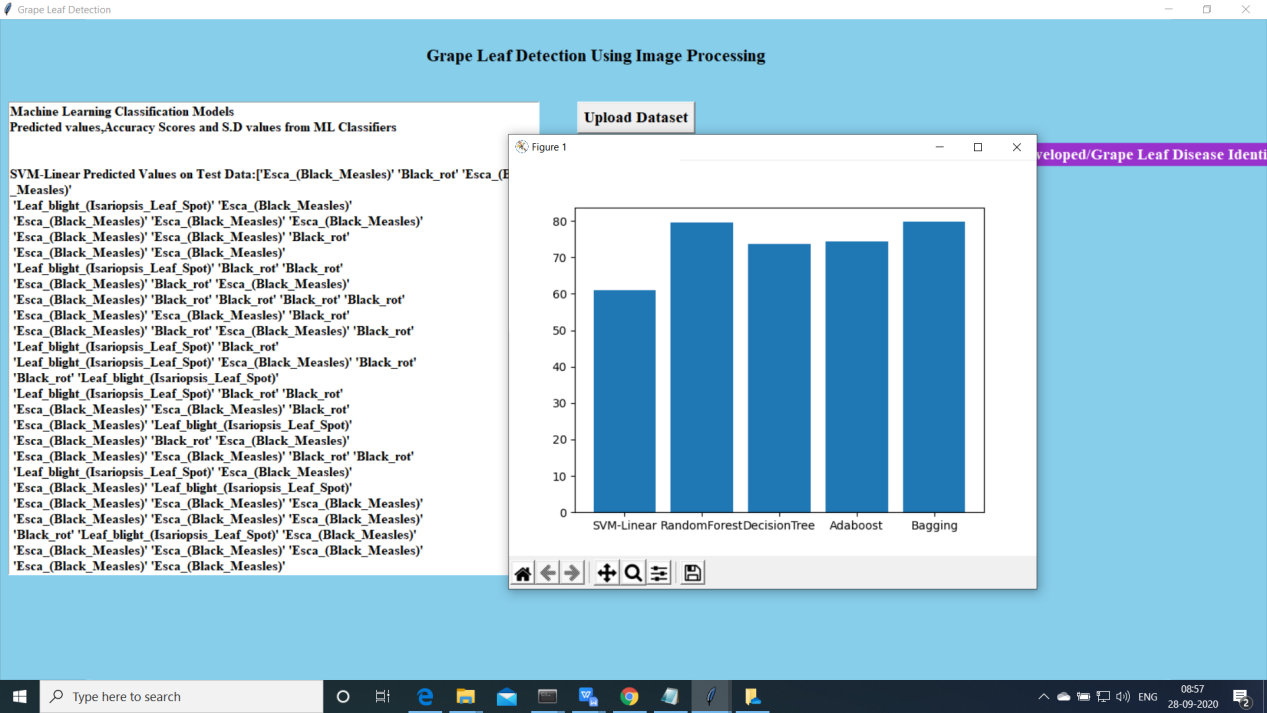




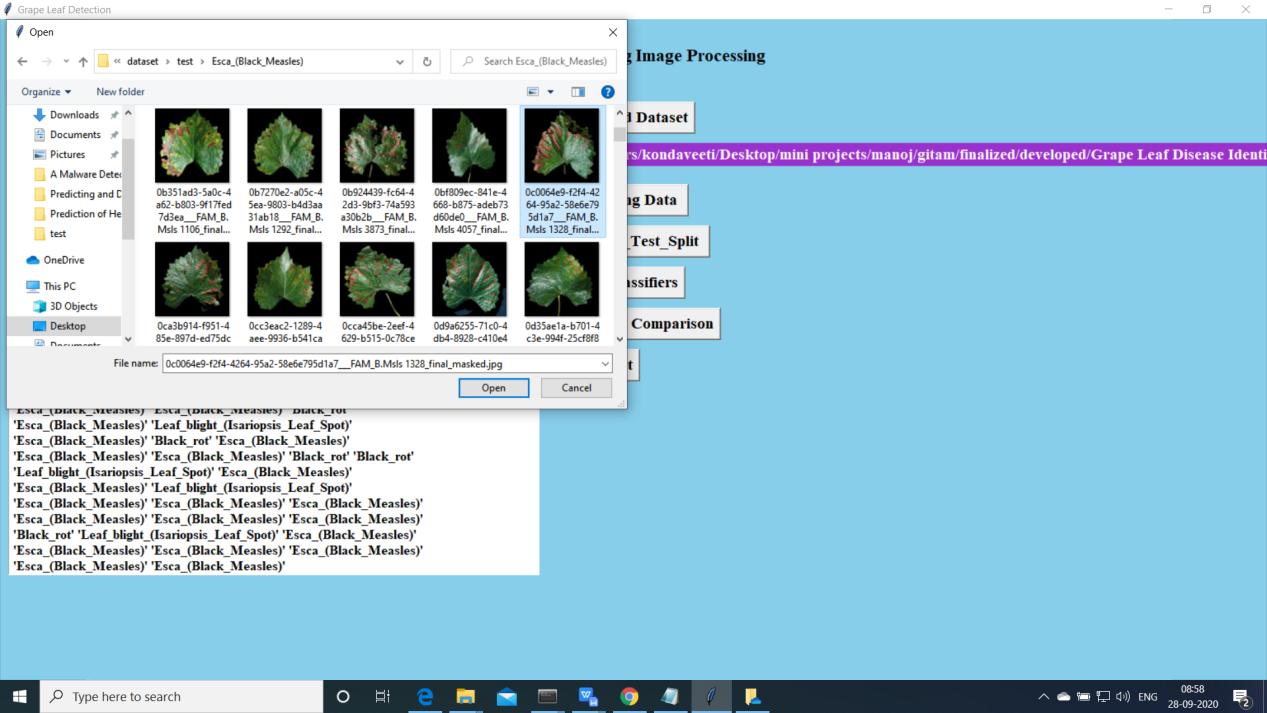


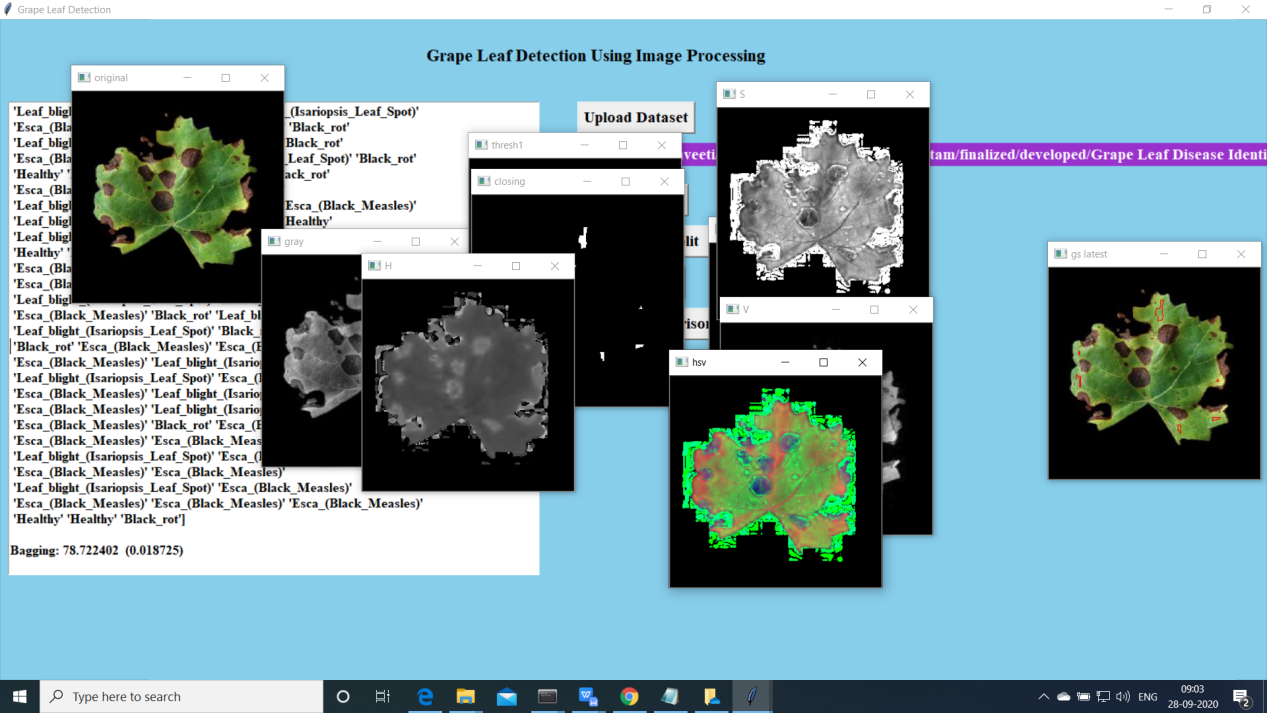


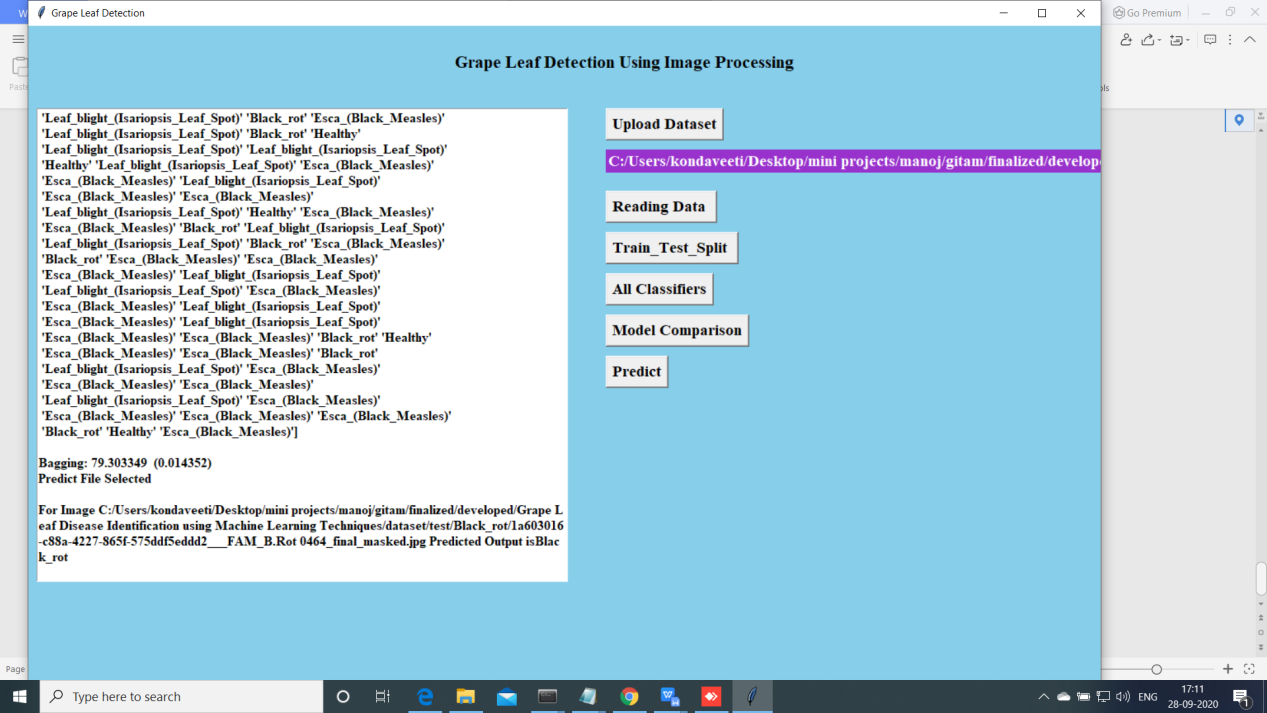
1. Accuracy Comparision for all the models



1. Predict:







Extension Bagging algorithm is performed well compared other ml algortithms