PLANT DISEASE DETECTION

ML PROJECT PHASE 2

* R.C.SIDDARTHA – AM.EN.U4CSE20156
* B.S.V.VARAPRASAD – AM.EN.U4CSE20116
* CHETAN ADITYA DASARI –AM.EN.U4CSE20119
* DATLA ANINDH VARMA –AM.EN.U4CSE20121
* E C JAYESH –AM.EN.U4CSE20123
* K KRISHNA CHAYTANYA –AM.EN.U4CSE20136

1. **PROBLEM STATEMENT:**

Classifying the plants into different categories based on diseases by taking leaf images as input by using machine learning algorithms.

1. Data sets:

[plantdisease](https://www.kaggle.com/datasets/emmarex/plantdisease)

[Apple disease](https://www.kaggle.com/datasets/hsmcaju/d-kap)

[plant pathology](https://www.kaggle.com/c/plant-pathology-2020-fgvc7/overview)

1. Preparing Data: (for plant pathology dataset)

We took plant pathology dataset which has 1821 items containing 4 classes namely

1. Rust (622)

2. Scab (592)

3. Multiple Diseases (713)

4. Healthy (516)

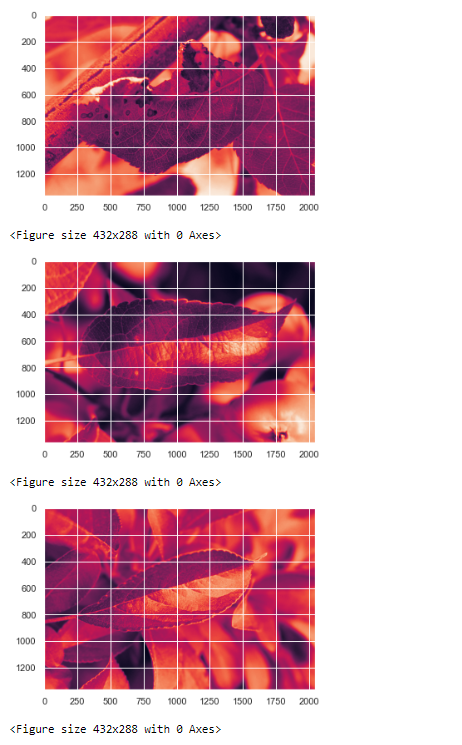
They were all given as one data set named images and a csv file train.csv which contains which id belongs to which class. We separated the images into 4 folders named on their respective class.

By using cv2.imread() we read the images and converted the images into HSV and grey scale.

In order to extract features without using deep learning we used haralick from sklearn.imagepreprocessing. we extracted 16 features consisting of 13 features from haralick and 3 dominant colors from the image matrix

1. Angular second moment
2. Contrast
3. Correlation
4. Sum of Squares
5. Inverse difference moment
6. Sum average
7. Sum variance
8. Sum entropy
9. Entropy
10. Difference variance
11. Difference entropy
12. Information measures of correlation
13. Maximal Correlation Coefficient
14. Dominant color 1
15. Dominant color 2
16. Dominant color 3

Image preprocessing:



1. Python Packages: (Plant Pathology)

* Numpy for mathematical calculations
* Pandas for reading, creating, writing data-frames or data
* Os for accessing folders and contents in folders and navigating through directories
* Natsort for sorting the images based on their id
* Matplotlib for plotting histogram graphs, color scaling
* Cv2 for reading images and color scaling the images
* Tqdm for showing the progress of for loop
* Sklearn for model selection, normalization, confusion matrix, accuracy, preprocessing, confusion matrix, train-test\_split, one vs rest classifier, roc curve, roc auc curve etc.
* Warnings in order to avoid warnings shown by the system.
* Mahotas inorder to extract features from the picture matrix.
* Itertools for using cycle in order to iterate through arrays.

1. Learning Algorithms:

We used algorithms such as :

* K- nearest neighbors

Knn algorithm is used for classification. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line.In our data no of nearest neighbours possible are 706 at maximum.After we had splitted the data into train test split we found test score and train scores.We also found accuracy with knn model.We also plotted the graph between number of neighbours and Accuracy.

* Support vector machine

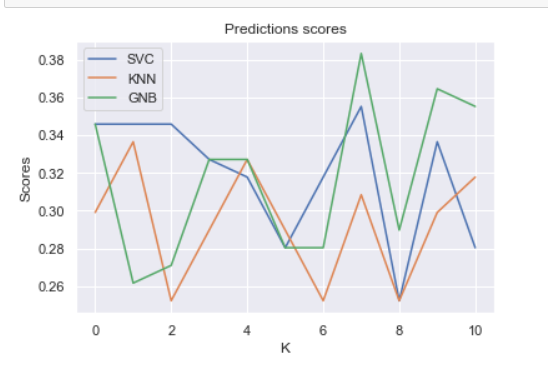
I imported support vector machine from sklearn.svm library.

We used support vector machine for our dataset to find accuracy and for classification.It can also predict unknown data.

* Gaussian Naïve bayes classifier

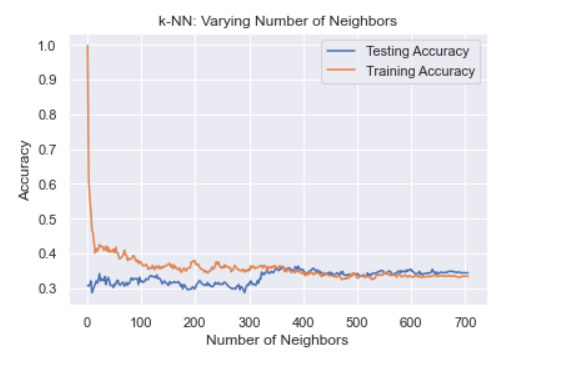
I used this algorithm for my dataset because it can make predictions fast.It is used for solving classification problems.Based on bayes theorem.As the name suggests it predicts on the basis of probability of an object.Here we imported Gaussian Naïve bayes classifier from sklearn.naive\_bayes library.

Create models and estimate their accuracy on unseen data using the specified ML algorithms:

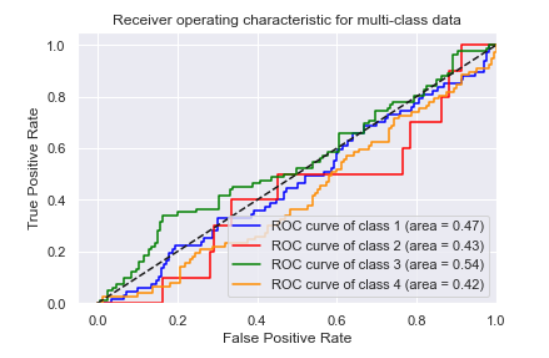


Predictions on dataset:

KNN



ROC CURVES FOR MULTICLASS CLASIFICATION:



Chart, line chart

Description automatically generated

