MACHINE LEARNING

PROGRAMMING ASSIGNMENT - 2

BY:
ASHIMA GARG
PhD19003

Solution 1:

About Cifar 10 dataset:

- The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
- The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.
- Original size of the dataset: ***(50000, 3072)***
- Each row of the array stores a 32x32 colour image. The first 1024 entries contain the red channel values, the next 1024 the green, and the final 1024 the blue. The image is stored in row-major order, so that the first 32 entries of the array are the red channel values of the first row of the image.

Method 1:

➢ **Data Preprocessing Used:**

- Reshape the dataset into (50000, 3, 32, 32).
- For processing using cv2 library, reshape the dataset to (50000, 32, 32, 3) in RGB format.
- Opencv uses BGR format for preprocessing of images. So, convert the above dataset to BGR format.
- Grid Search for SVM training is computationally very expensive for dataset with features of size 50000*32*32*3 = 15,36,00,000.
- Therefore, shrink the image of size 32*32*3 to 16*16*3
- Convert the image to grayscale image
- Execution time for feature extraction of Training data: **1.153sec**
- Execution time for feature extraction of Testing data: **0.203sec**
- Feature extraction results the training dataset to size ***(50000, 256)***
- Feature extraction results the testing dataset to size ***(10000, 256)***

➢ **Model Training and Testing**

Using GridSearchCV, SVC routine from sklearn library to find best parameters for the dataset.

Possible Values of C and kernel used for params of gridsearchcv are:

C= {0.1, 1, 10}, kernel = {‘rbf’, ‘linear’}

Best Params found:

- C = 10 (Inverse of Regularization parameter)
- Kernel = 'rbf'

Training set accuracy = ***36.28%***
Testing set accuracy = ***38.42%***

Execution time for Training: ***2463.492 sec ~ 41.05 minutes.***
Execution time for Testing: ***300.10 sec ~ 5 minute***

Method 2:

About HOG Features:

The HOG descriptor focuses on the structure or the shape of an object. HOG is able to provide the edge direction as well. This is done by extracting the gradient and orientation (or you can say magnitude and direction) of the edges. Additionally, these orientations are calculated in 'localized' portions. This means that the complete image is broken down into smaller regions and for each region, the gradients and orientation are calculated. We will discuss this in much more detail in the upcoming sections. Finally the HOG would generate a Histogram for each of these regions separately. The histograms are created using the gradients and orientations of the pixel values, hence the name 'Histogram of Oriented Gradients'.
A histogram is a plot that shows the frequency distribution of a set of continuous data. We have the variable (in the form of bins) on the x-axis and the frequency on the y-axis. Here, we are going to take the angle or orientation on the x-axis and the frequency on the y-axis.

➢ **Data Preprocessing Used:**

- Reshape the dataset into (50000, 3, 32, 32).
- For processing using cv2 library, reshape the dataset to (50000, 32, 32, 3) in RGB format.
- Opencv uses BGR format for preprocessing of images. So, convert the above dataset to BGR format.
- Grid Search for SVM training is computationally very expensive for dataset with features of size 50000*32*32*3 = 15,36,00,000.
- Extract Histogram of Oriented Gradients(HOG) features of each image by first converting the BGR format to Grayscale image.
- Settings used for **HOG feature extraction** is:
  orientations=9, pixels_per_cell=(8, 8), cells_per_block=(4, 4), block_norm='L2-Hys', visualize=False, transform_sqrt=True
- Execution time for feature extraction of Training data: **15.31sec**
- Execution time for feature extraction of Testing data: **2.22sec**

- Feature extraction results the training dataset to size ***(50000, 144)***
- Feature extraction results the testing dataset to size ***(10000, 144)***

➢ **Model Training and Testing**

Using GridSearchCV, SVC routine from sklearn library to find best parameters for the dataset.

Experiments in different settings of GridSearch Parameter

❑ **Settings 1:**

Parameters used from GridSearchCV in this setting:
C= {0.1, 1, 10}, kernel = {'rbf', 'linear', 'poly'}, gamma = {0.1, 1, 10}

Best Estimators of SVC:

- C = 10 (Inverse of Regularization parameter)
- Kernel = 'rbf'
- Gamma = 1

1. Results Using Grid Search
   Best Score: ***61.238%***
   Execution time for Training: ***5367.034 sec ~ 90 minutes.***

2. Results Using Model trained with best parameters
   Train Set Accuracy: ***94.652%***
   Test Set Accuracy: ***63.63%***
   Number of Support Vectors found: ***39248***
   Execution time for Training: ***660.71sec ~ 10 minutes.***
   Execution time for Testing: ***300.81sec ~ 5 minutes.***

3. Results Using Model trained with Support Vectors
   Train Set Accuracy: ***93.19%***
   Test Set Accuracy: ***63.62%***

❑ **Settings 2:**

Parameters used from GridSearchCV in this setting:
C= {0.1, 1, 10}, gamma = {0.1, 1, 10'}

Best Estimators of SVC:

- C = 10 (Inverse of Regularization parameter)
- Kernel = *default-'rbf'*
- Gamma = 1

4. Results Using Grid Search
   Best Score: ***61.238%***
   Execution time for Training: ***2390.91sec ~ 39 minutes.***

5. Results Using Model trained with best parameters
   Train Set Accuracy: ***94.652%***
   Test Set Accuracy: ***63.63%***
   Number of Support Vectors found: ***39248***
   Execution time for Training: ***660.71sec ~ 10 minutes.***
   Execution time for Testing: ***300.81sec ~ 5 minutes.***

6. Results Using Model trained with Support Vectors
   Train Set Accuracy: ***93.19%***
   Test Set Accuracy: ***63.62%***

❑ **Settings 3**
   Parameters used from GridSearchCV in this setting:
   C= {0.1, 1, 10}, kernel = {'rbf', 'linear', 'poly'}

   Best Estimators of SVC:
   - C = 1 (Inverse of Regularization parameter)
   - Kernel = 'linear'
   - Gamma = *default-'auto' which uses (1/n_features)*

7. Results Using Grid Search
   Best Score: ***51.87%***
   Execution time for Training: ***2693.54 sec ~ 45 minutes.***

8. Results Using Model trained with best parameters
   Train Set Accuracy: ***53.708%***
   Test Set Accuracy: ***52.38%***
   Number of Support Vectors found: ***41553***
   Execution time for Training: ***660.71sec ~ 10 minutes.***
   Execution time for Testing: ***301.81sec ~ 5 minutes.***

9. Results Using Model trained with Support Vectors
   Train Set Accuracy: ***53.704%***
   Test Set Accuracy: ***52.37%***

❑ **Settings 4:**
   Parameters used from GridSearchCV in this setting:
   C= {0.1, 1, 10}

   Best Estimators of SVC:
   - C = 10 (Inverse of Regularization parameter)
   - Kernel = *default-'rbf'*
   - Gamma = *default-'auto' which uses (1/n_features)*

10. Results Using Grid Search
    Best Score: ***50.85%***
    Execution time for Training: ***1251.91sec ~ 20 minutes.***

11. Results Using Model trained with best parameters
    Train Set Accuracy: ***52.582%***
    Test Set Accuracy: ***51.49%***
    Number of Support Vectors found: ***45071***
    Execution time for Training: ***660.71sec ~ 10 minutes.***
    Execution time for Training: ***305.81sec ~ 5.1 minutes.***

12. Results Using Model trained with Support Vectors
    Train Set Accuracy: ***47.39%***
    Test Set Accuracy: ***51.49%***

Inferences

- SVM training without HOG feature extraction, and just normalizing the data directly feeding the dataset took a lot of time. 100 samples of Cifar 10 Dataset was used in this setting. Training Accuracy observed ~ 16% (very low).
- Feature extraction technique using Histogram of oriented gradients helped in extract 144 important features which significantly reduced the execution time using grid search.
- *Note:* Limited parameters from Gridsearchcv are used due to time constraints.

- Observations in Settings 1 and 2 differ from Settings 3 and 4 with gamma parameter and it can be observed that gamma in default value of gamma i.e. (1/n_features = 0.0069) works best in the setting. As when gamma is varied in between (0.1, 1, 10), it overfits the training data and hence the training set accuracy of 94% is achieved whereas test set accuracy 63.63%. This is consistent with the fact that the gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.
- Observations could be better if gamma parameter is varied in much larger range as compared to the current settings. But due to time constraints, it is limited in this range.
- RBF kernel or the gaussian kernel is the best kernel found because this kernel can map the input features to much higher dimensions as compared to polynomial or linear kernel.
- In all the four different Settings of GridSearch Parameters, it can be observed that training and test accuracy obtained when trained with the whole dataset of *50K* samples, is almost equivalent to when the model is trained with only support vectors. Thus, it can be inferred that only support vectors affect the model training and testing. At the test time, only found support vectors at the train time is used to test the performance. This is consistent with the fact that lagrangian multiplier exist for only support vectors and for other train samples which are not support vectors, it is zero and thus they don’t affect the model testing.

Solution 2:

Part i) Plot pairwise relations in dataset
Used Wine Dataset of Sklearn Library:

About the Wine Dataset

- **Features/Attributes:**
  1) Alcohol
  2) Malic acid
  3) Ash
  4) Alcalinity of ash
  5) Magnesium
  6) Total phenols
  7) Flavanoids
  8) Nonflavanoid phenols
  9) Proanthocyanins
  10) Color intensity
  11) Hue
  12) OD280/OD315 of diluted wines
  13) Proline

- **Target Names**
  1) Class 0
  2) Class 1
  3) Class 2

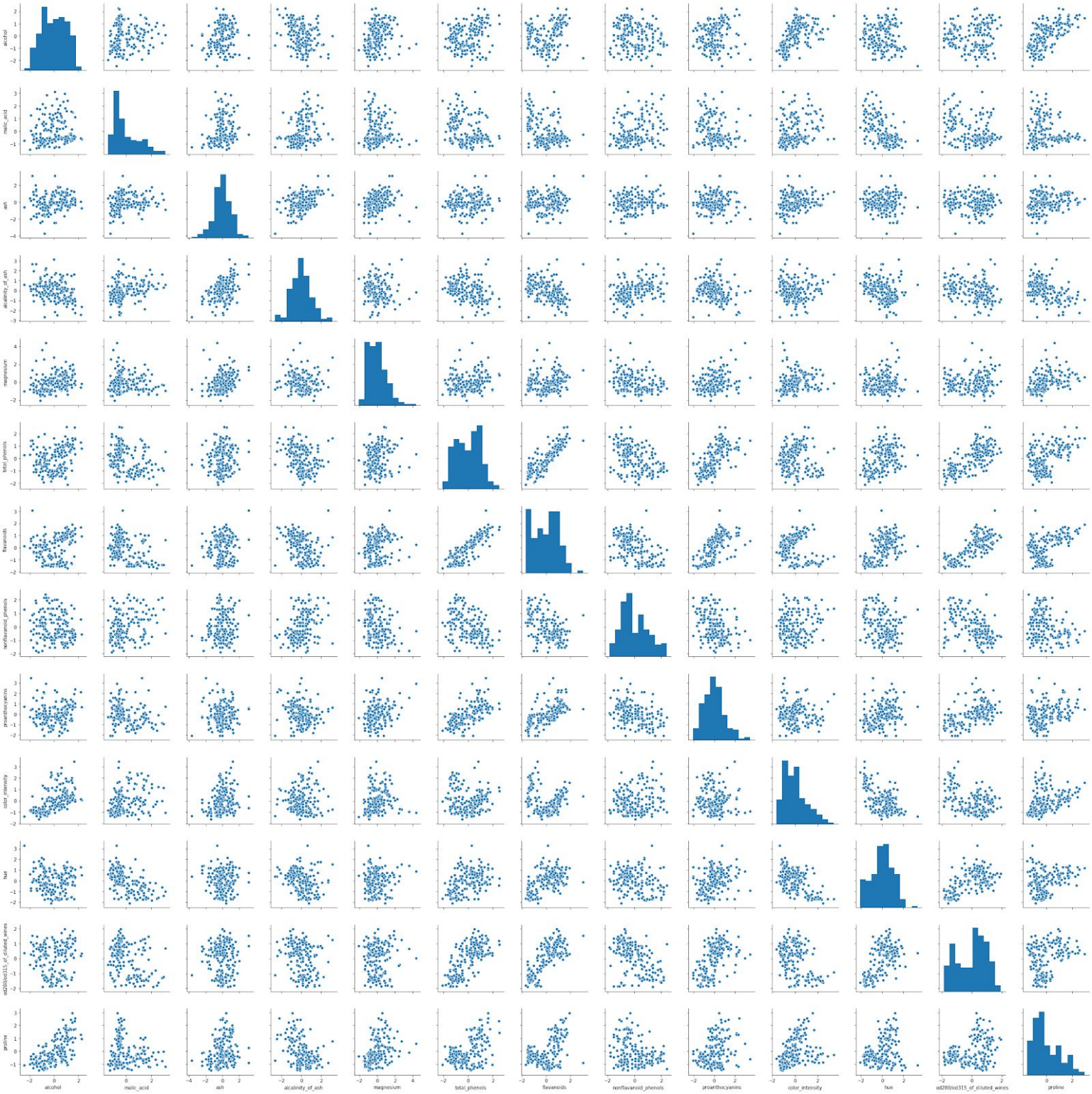
- **Number Of Instances:**
  1) Class 0: 59
  2) Class 1: 71
  3) Class 2: 48

- **Description of Variables.**

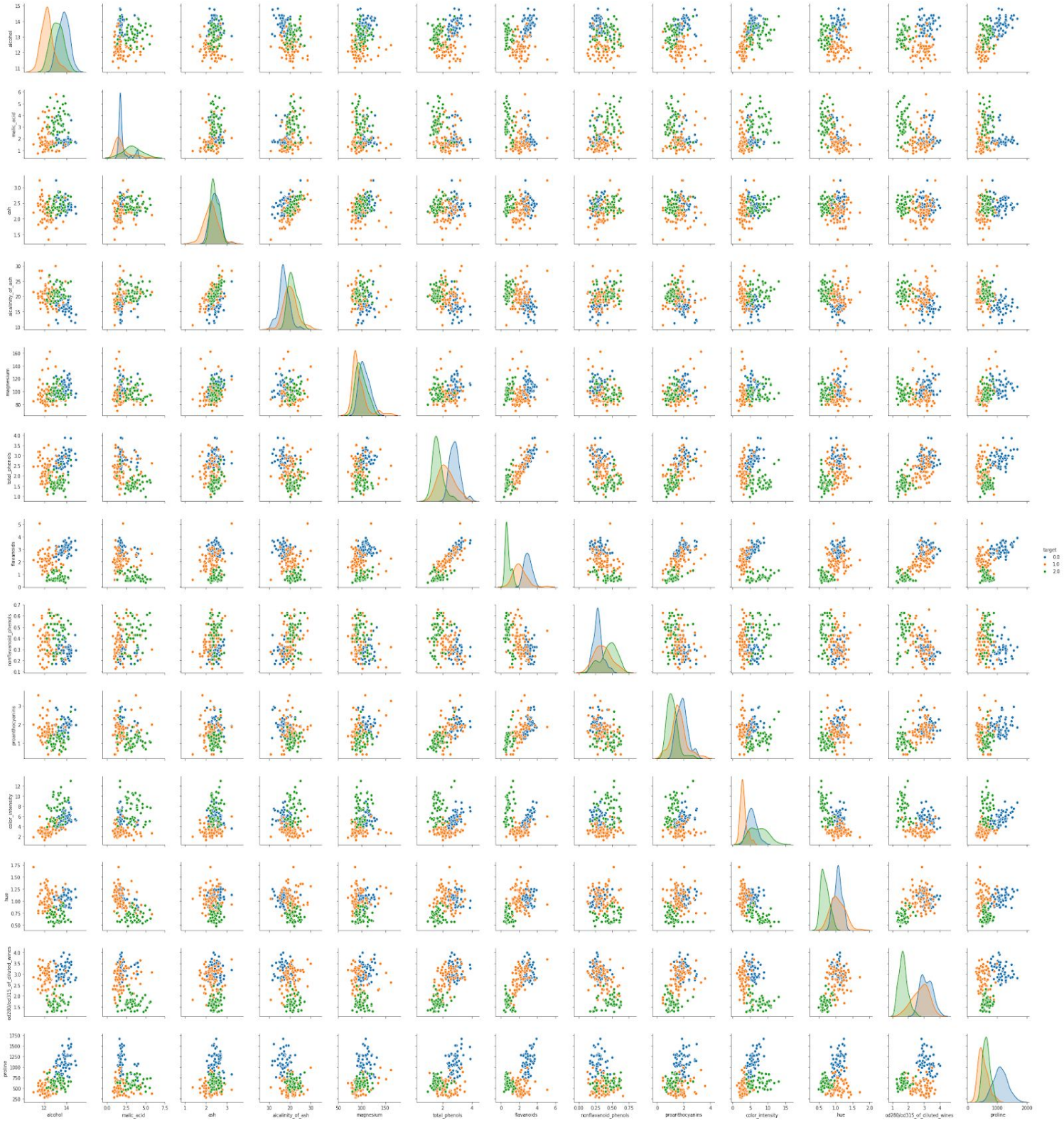
	Min	Max	Mean	Standard Deviation
Alcohol	11.0	14.8	13.0	0.8
Malic Acid:	0.74	5.80	2.34	1.12
Ash	1.36	3.23	2.36	0.27
Alcalinity of Ash	10.6	30.0	19.5	3.3
Magnesium	70.0	162.0	99.7	14.3
Total Phenols	0.98	3.88	2.29	0.63
Flavanoids	0.34	5.08	2.03	1.00
Nonflavanoid Phenols	0.13	0.66	0.36	0.12
Proanthocyanins	0.41	3.58	1.59	0.57
Colour Intensity	1.3	13.0	5.1	2.3
Hue	0.48	1.71	0.96	0.23
OD280/OD315 of diluted wines	1.27	4.00	2.61	0.71
Proline	278	1680	746	315

- Task is to identify the type of alcohol based on the constituents of the wine.
- Since, from the above table it can be inferred that data needs to be standardized. StandardScaler() used from sklearn library to standardize the dataset.

Following pairwise plot shows the correlation between features pairwise and the diagonals of the plot shows the histograms of each feature after standardizing for univariate distributions



Following pairwise plot shows different levels of a categorical variable by the color of plot elements.



Inferences of Dataset

- Pair plot allows to see both distribution of single variables and relationships between the two variables. It plots histograms of the variable which allows us to see the distribution of a single variable and scatter plots allows us to visualize the relationship between the two variables.
- Chemical Flavanoids is not related with feature 'ash', negatively correlated with feature 'alcalinity_of_ash' and positively correlated to all other features with one outlier in each plot.
- Pronline is positively correlated to other features and Class 0 has the highest values of proline.
- Histograms of Proline and malic acid signify that they are slightly right skewed implies mean of these variables is also to the right of this variable which can be verified from the above table.
- Other variables are not correlated to each other.

Part ii) One Vs One and One Vs Rest Classifier

Evaluation Metrics Used:

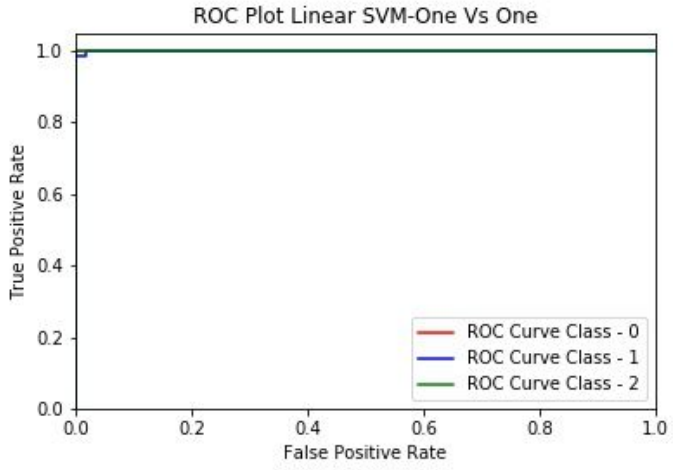
- Train Set Accuracy
- Test Set Accuracy
- F1-Score: (2*Precision*Recall)/(Precision + Recall)
- ROC Curve: Plot between True Positive rate and False Positive rate
  True Positive rate: (TP)/(TP + FN)
  False Positive rate: (FP)/(TN + FP)

❑ ***One vs One***

- **Parameters Used:**
  Sklearn module used- *SVC()* with parameters:
  C = 1.0
  Kernel = 'Linear'

Results:

- **One Vs One**
  Train Set Accuracy Score: 1.0
  Test Set Accuracy Score : 0.981
  F1 Score: 0.981
  Execution Time: 0.0039825 seconds
  ROC Curve:

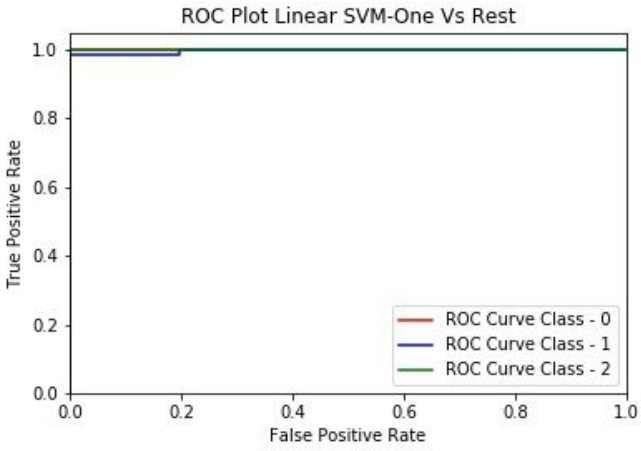


- **One Vs Rest**

  Sklearn module used - *LinearSVC() with default parameters:*
  C = 1.0

  Results

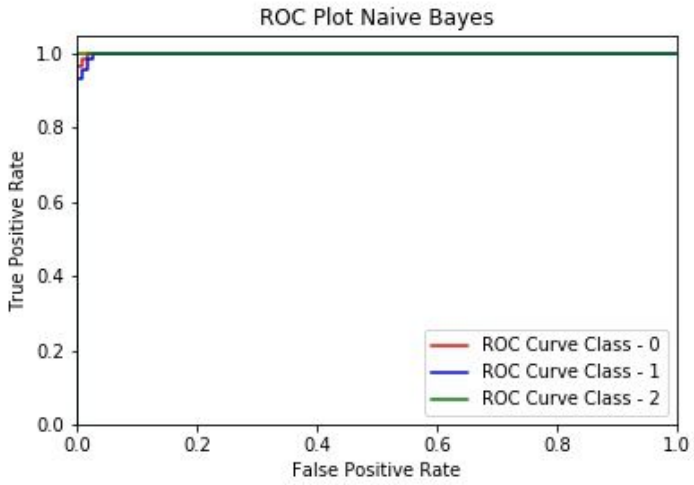
  Train Set Accuracy Score: 1.0
  Test Set Accuracy Score : 0.981
  F1 Score: 0.981
  Execution Time: 0.00697 seconds
  ROC Curve:



Part iii) Gaussian Naive Bayes

Parameters Used: *Default settings of Sklearn library*
priors=None
var_smoothing=1e-09

Results:
Train Set Accuracy Score: 0.9758064516129032
Test Set Accuracy Score : 1.0
F1 Score: 1.0
Execution Time: 0.0020263 seconds
ROC Curve:



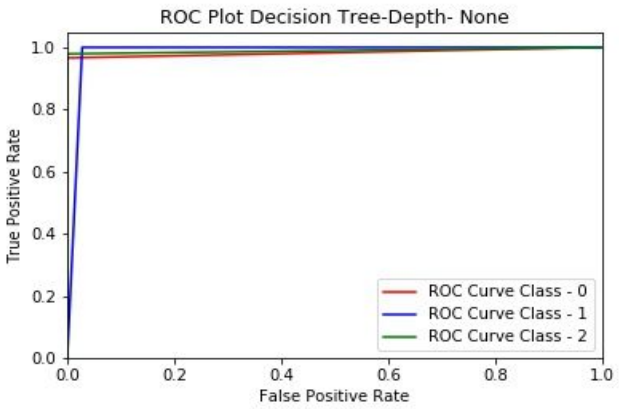
Part iv) DECISION TREE

Observations in different settings of ***max_depth*** parameter in ***DecisionTreeClassifier()*** of Sklearn library.

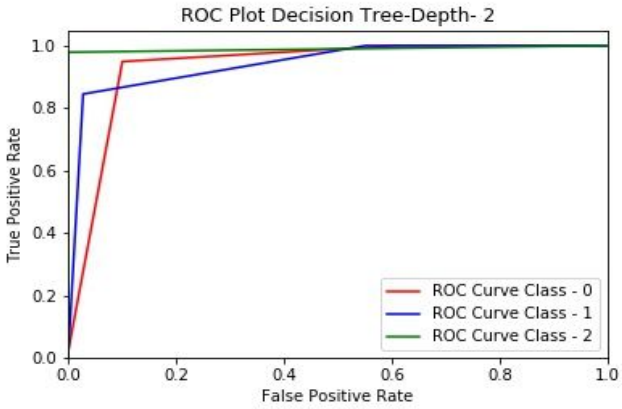
Depth	Train Set Accuracy	Test Set Accuracy	F1 Score	Execution Time(sec)
None(Default)	1	0.94	0.94	0.0029931
2	0.94354838709	0.85185185185	0.851	0.00099683
5	1	0.963	0.963	0.003865
10	1	0.963	0.963	0.00099993

ROC Curves with different values of max_depth parameter:

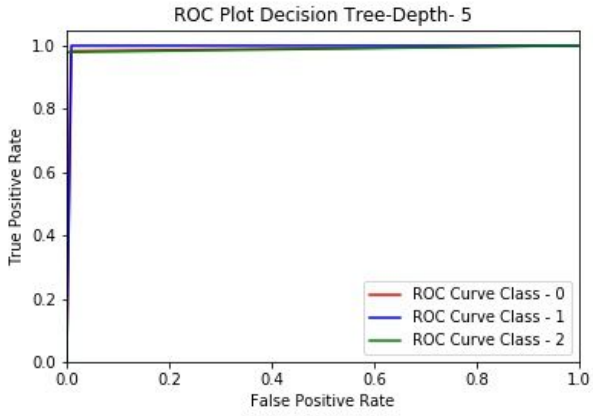
1) Depth = None



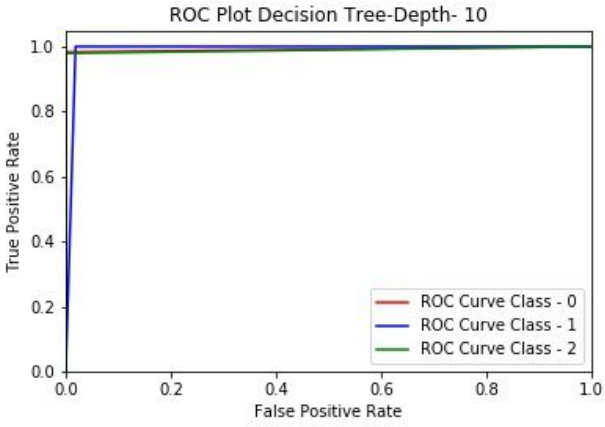
2) Depth = 2



3) Depth = 5



4) Depth = 10



RESULTS

- **Class Wise Train Accuracies of All the models**

	One Vs One	One Vs Rest	Gaussian Naive Bayes	Decision Tree(Depth = 5)
Class 0	1.0	1.0	0.95	1.0
Class 1	1.0	1.0	0.98	1.0
Class 2	1.0	1.0	1.0	1.0

- **Class Wise Test Accuracies of All the models**

	One Vs One	One Vs Rest	Gaussian Naive Bayes	Decision Tree(Depth = 5)
Class 0	1.0	1.0	1.0	0.947
Class 1	0.9523	0.9523	1.0	1.0
Class 2	1.0	1.0	1.0	0.928

- **Comparison of Overall train accuracy, test accuracy and F1 Score:**

	One Vs One	One Vs Rest	Gaussian Naive Bayes	Decision Tree(Depth = 5)
Train Accuracy	1.0	1.0	0.975	1.0
Test Accuracy	0.981	0.9814	1.0	0.962
F1 Score	0.981	0.9814	1.0	0.962

Inferences

- Model trained with gaussian naive bayes performs the best as train set accuracy is 97.5% and test set accuracy is 100%. In case of other models, train set accuracy obtained is 100% whereas it drops to 96-98% when the model is tested on the testing data.
- Class-Wise training accuracies of gaussian naive bayes is 95% for Class 0, 98% for Class 1 and on testing data, 100% accuracy is obtained on all the classes.
- It can be observed that model trained with default parameter of max_depth parameter in Decision tree overfits the model and thus train set accuracy is 100% and test set accuracy drops to 94.44%.