Principles of Machine Learning(CS4011)
Programming Assignment#2

S Ram Ananth
ME15B153

October 10, 2017

Feature Extraction

1 Prinicipal Component Analysis

1.1 Goal

Perform PCA on the provided dataset DS3,which is a 3-Dimensional dataset with 2 classes, and extract 1 feature. Use the data in this projected space to train linear regression with indicator random variables.Use the learnt model to classify the test instances. Report per-class precision, recall and f-measure.Report the 3-D plot of the dataset and the plot of the dataset in the projected space along with the classifier boundary.

1.2 Approach

The 3-D plot of the datapoints in DS3 is plotted with datapoints colored classwise. Principal Component Analysis(PCA) is performed on this dataset and one feature along direction of maximum variance is extracted and datapoints are then projected onto that dimension.

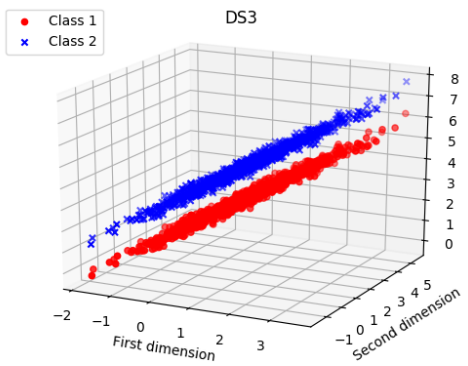

Figure 1: Plot of datapoints in DS3

Linear regression with indicator random variables is trained on this projected space and all values above a threshold (1.5 in this case) are classified as Class 2 and rest as Class 1. The decision boundary is obtained as follows:

$$y = \beta_0 + \beta_1 x$$

$$x_{decision} = \frac{y_{threshold} - \beta_0}{\beta_1}$$

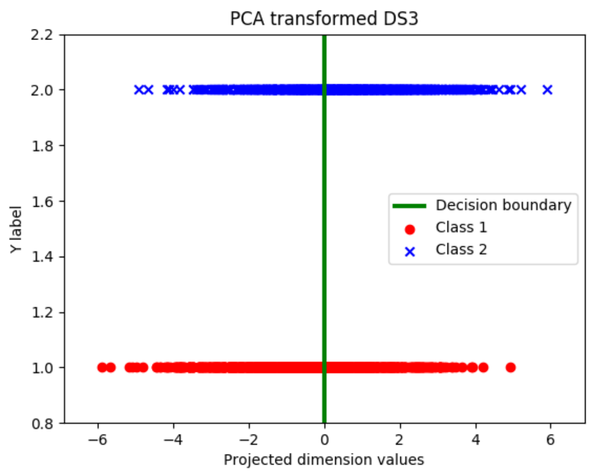

Figure 2: Plot of datapoints in the projected dimension found by PCA

1.3 Result

The Performance metrics are as follows.

Class	Accuracy	Precision	Recall	F-Measure
1	0.3825	0.38	0.39	0.39
2		0.38	0.38	0.38

2 Linear Discriminant Analysis

2.1 Goal

Perform LDA on the provided dataset DS3,which is a 3-Dimensional dataset with 2 classes, and extract 1 feature. Use the data in this projected space to train linear regression with indicator random variables.Use the learnt model to classify the test instances. Report per-class precision, recall and f-measure.Report the 3-D plot of the dataset and the plot of the dataset in the projected space along with the classifier boundary.

2.2 Approach

The 3-D plot of the datapoints in DS3 is plotted with datapoints colored classwise (As shown in Figure 1). Linear Discriminant Analysis(LDA) is performed on this dataset and one feature, along direction that minimises within-class variance and maximises between-class variance ,is extracted and datapoints are then projected onto that dimension.

Linear regression with indicator random variables is trained on this projected space and all values above a threshold (1.5 in this case) are classified as Class 2 and rest as Class 1. The decision boundary is obtained as follows:

$$y = \beta_0 + \beta_1 x$$

$$x_{decision} = \frac{y_{threshold} - \beta_0}{\beta_1}$$

2.3 Result

The Performance metrics are as follows.

Class	Accuracy	Precision	Recall	F-Measure
1	1.0	1.0	1.0	1.0
2		1.0	1.0	1.0

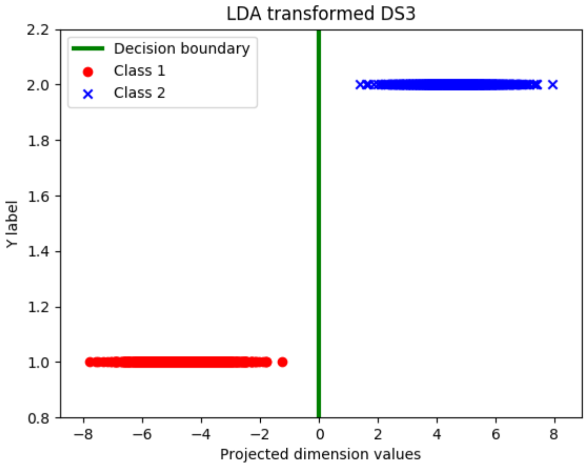

Figure 3: Plot of datapoints in the projected dimension found by LDA

As seen from Figures 2 and 3 and the obtained performance metrics LDA is seen to perform much better than PCA in this case as LDA finds direction that minimises within-class variance while maximising between-class variance unlike PCA which just finds direction of maximum variance.

3 Discriminant Analysis

3.1 Goal

For this experiment, we have to use Iris Dataset (http://archive.ics.uci.edu/ml/datasets/Iris).The goal is to use only petal width and petal length features and perform LDA and visualize the boundaries learnt. Perform QDA on the same data set and visualize the boundaries learnt.

3.2 Approach

The dataset is read with datatype string to ensure labels are read properly. The labels are then encoded to numbers and features are converted to float. The petal length and petal width alone are chosen as features and LDA,QDA is performed based on this. The points are plotted on a color mesh to help better visualise the boundaries learnt by the different classifiers.

3.3 Result

The plots obtained are as follows:

For LDA

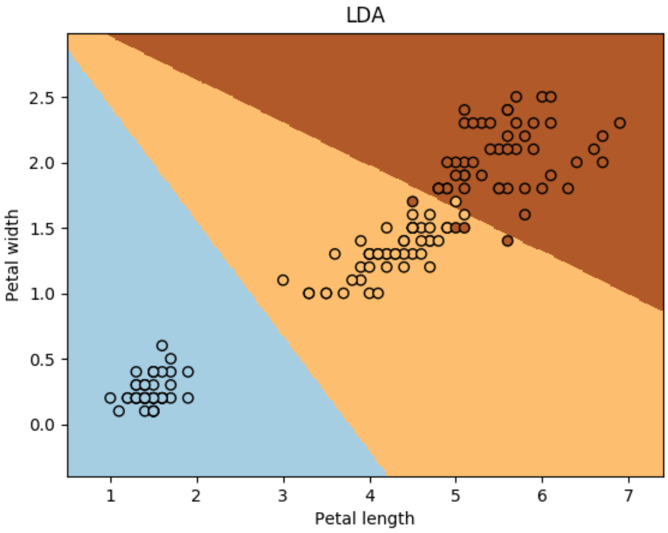

Figure 4: LDA classification on Iris dataset

For QDA

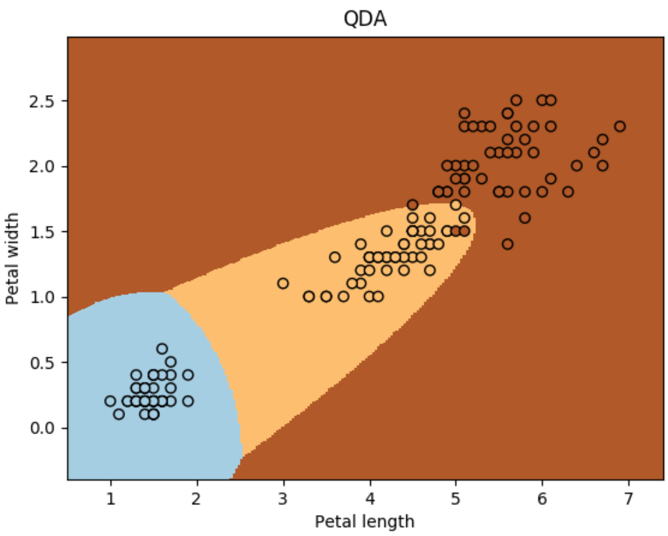

Figure 5: QDA classification on Iris dataset

4 Support Vector Machines

4.1 Goal

The goal is to train an SVM to classify the test images of DS2 into either of the following four categories: coast, forest, inside-city, mountain. Use the training data to build classification models using the following kernels:Linear kernel, Polynomial kernel,Gaussian kernel,Sigmoid kernel and come up with the kernel parameters for the various models with the help of k-fold cross validation.

4.2 Approach

Support vector machines are an important class of machine learning algorithm which is used widely in the industry. The 4 most commonly used Kernels for SVM and their functions are as follows

Kernel Type	Function $K(x_i, x_j)$
Linear	$x_j^T x_i$
Polynomial	$(x_j^T x_i + 1)^d$
Gaussian (Radial Basis Function)	$\exp(-\gamma \|x_i - x_j\|^2)$
Sigmoid	$\tanh(\gamma x_j^T x_i + coef0)$

As seen from the table above each of the kernels use some parameter or the other apart from C that need to be chosen. 5- fold cross validation was performed and accuracy used as an evaluation metric to compare performance of classifiers with different hyperparameters. Hyperparmaters C and γ were varied logarthimically to base 2 and *degree* and $coef0$ hyperparameters were varied linearly. The obtained best-fit models were saved as

4.3 Result

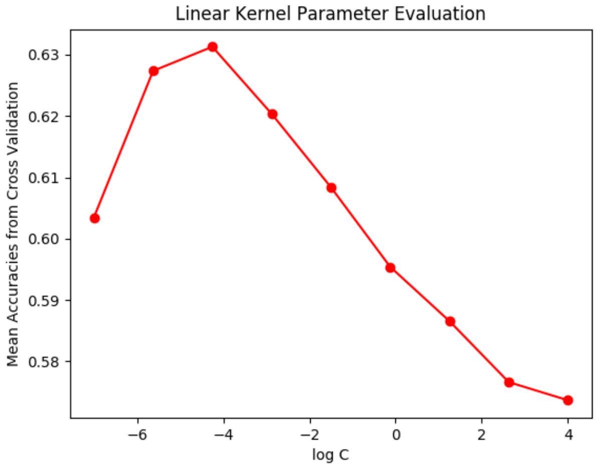

Figure 6: Linear Kernel Parameter Evaluation

The best parameters and their scores were found to be :

Kernel	Best-fit parameters	Mean Validation Accuracy
Linear	C=0.0525560259534	0.6313
Polynomial	C=0.35355339059327379	0.6580
	γ = 0.0078125	
	$degree$ = 4	
Radial Basis Function	C= 2.3784142300054421	0.6630
	γ = 0.0078125	
Sigmoid	C= 0.91700404320467122	0.5656
	γ=0.0078125	
	$coef0$=0	

5 Decision Trees

5.1 Goal

The goal is to use Weka and Mushroom dataset from UCI machine learning repository (https://archive.ics.uci.edu and run J48 Decision Tree algorithm. Report precision, recall and f1- measure and turn in the decision tree learnt by this model

5.2 Approach

The agaricus.csv file is first fitted with column headings obtained through the information provided for the dataset . This csv file is then converted to arff format which is suitable for Weka.

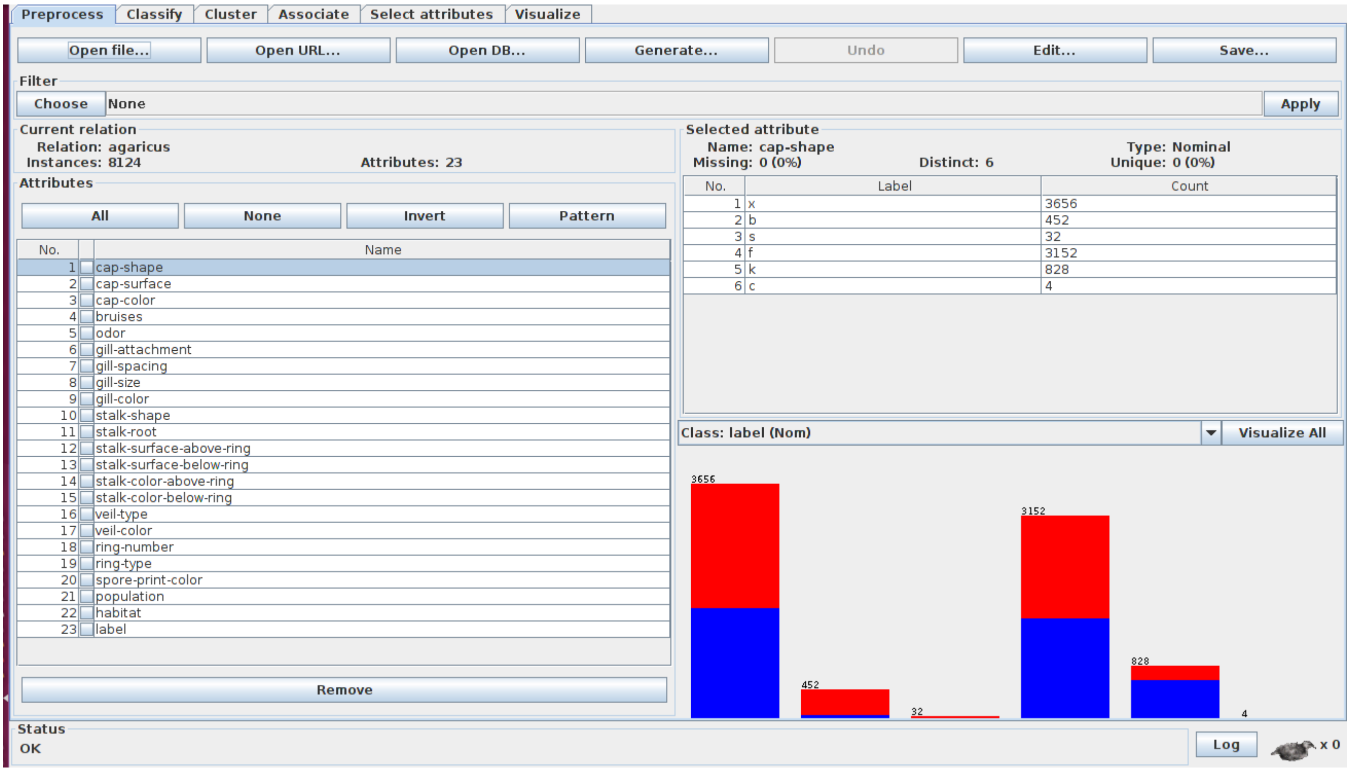

Figure 7: Agaricus Dataset in Weka

Then we use the inbuilt J48 decision tree algorithm that is part of Weka to perform classification with percentage split of 86.16% , corresponding to 1124 test instances of out 8124 instances.

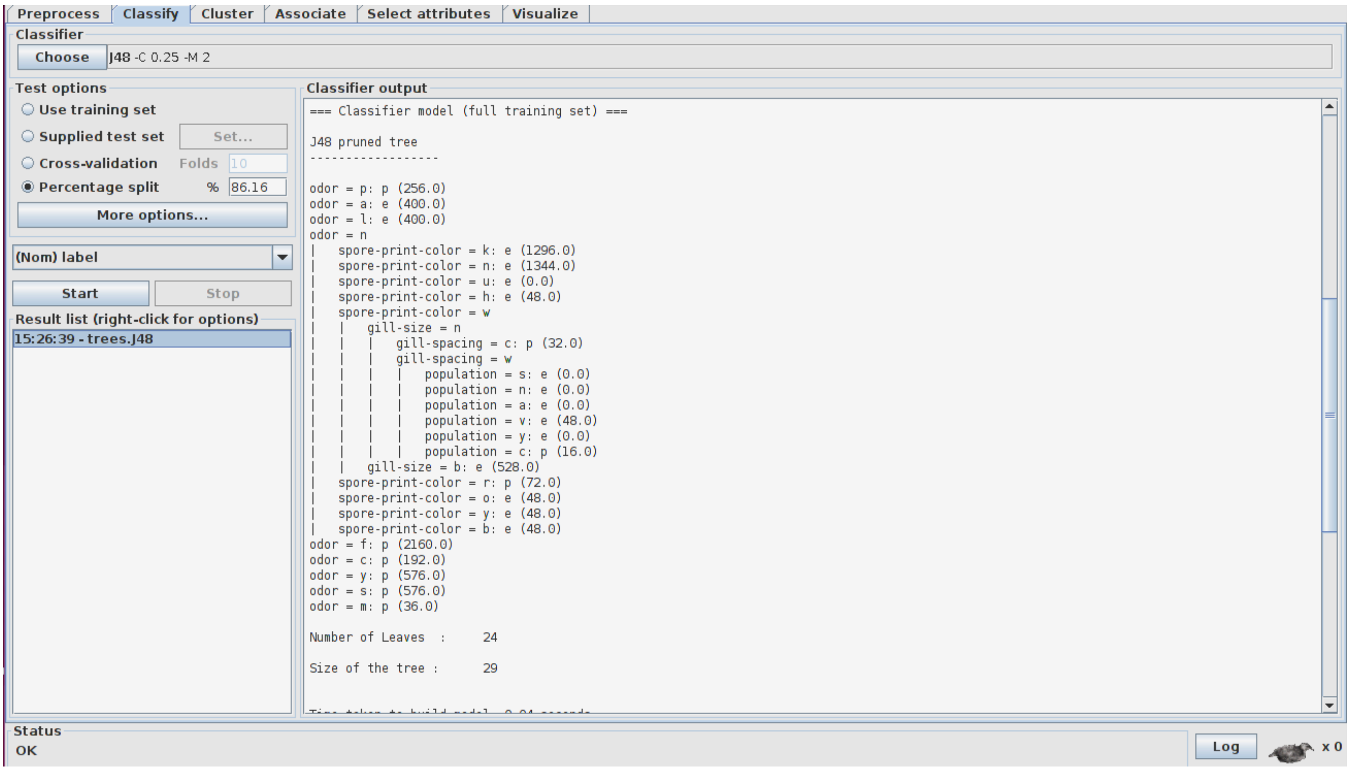

Figure 8: Model learnt by J48 algorithm on Agaricus Dataset

5.3 Result

The evaluation done on the test set is as follows:

Class	Accuracy	Precision	Recall	F-Measure
p	1.0	1.0	1.0	1.0
e		1.0	1.0	1.0

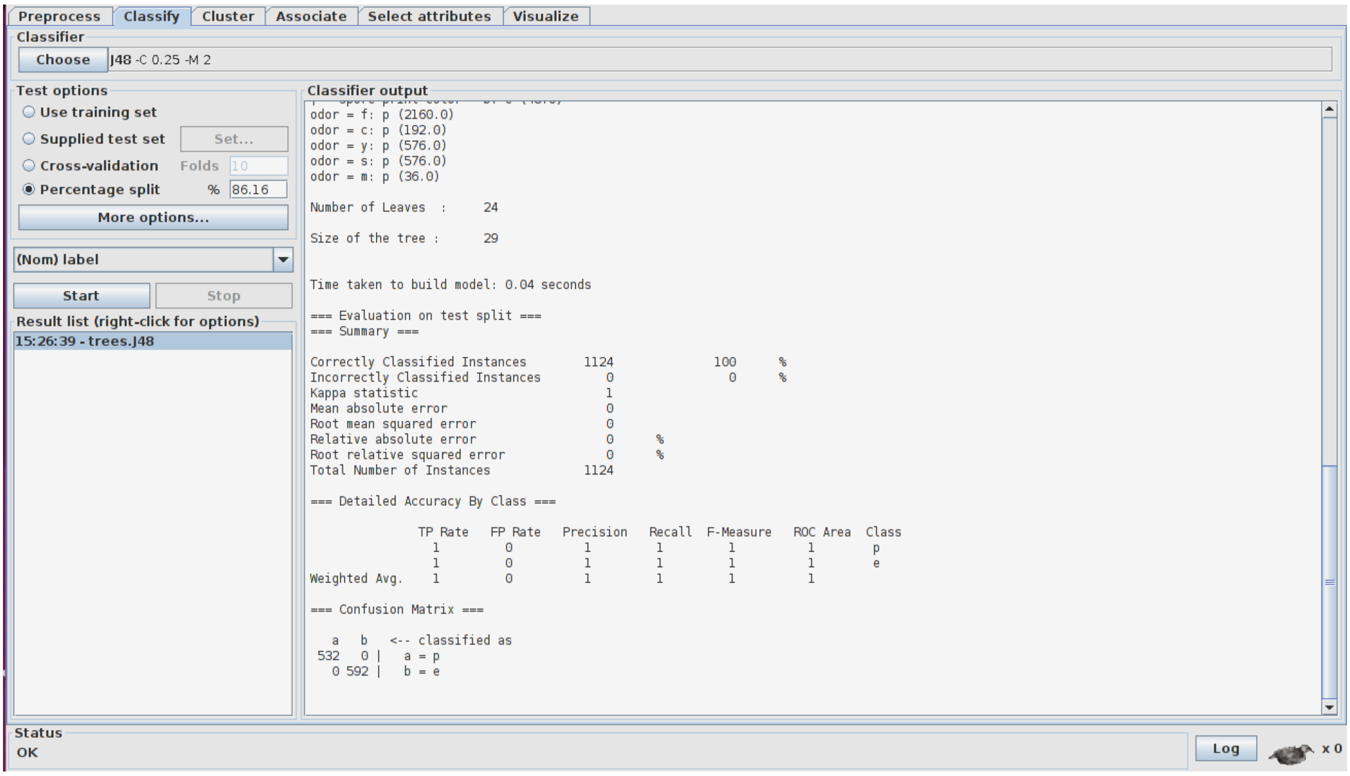

Figure 9: Evaluation on test set of Agaricus Dataset

MinNumObj is the minimum number of instances per leaf. If we keep increasing it performance worsens because we are restricting it from forming a fully grown tree with each leaf containing say only one instance.

When we set reduced error pruning to be true, all error metrics like RMSE,Mean Absolute Error,F-Measure,Precision,Recall,etc seem to be more than the case where same paramters are used without reduced error pruning and number of leaves and tree size is less for case with reduced error pruning.

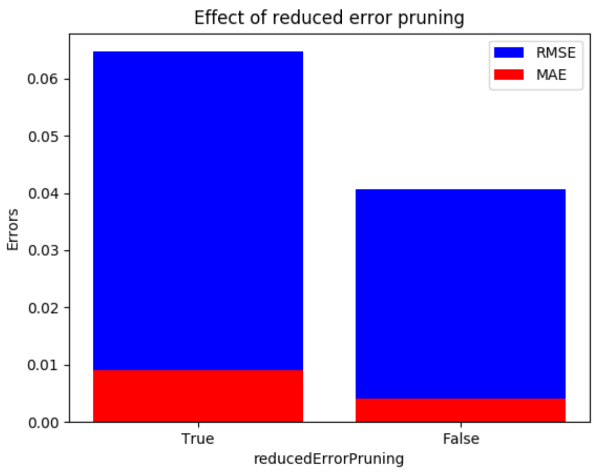

Figure 10: Effect of Reduced Error Pruning

From the splits mentioned in Figure 8 we can say that important features are **odor,spore-print-color,gill-size,population,gill-spacing.**