Introduction to Data Science / Data Intensive Computing (CIS 4930/6930)

Project 4

Submitted By: Rajat Koujalagi

UFID : 61382944

INTRODUCTION

The project deals with building a classifier using a training set of images and show how well it generalizes for the test set. Accuracy is calculated with the class given in the test set. Image data is classified based upon the pixel values. The classification model being used is Decision Tree. K-Means algorithm is used for clustering.

TASK 1

1. The image file (10X10 pixel matrix) is converted to a vector size of 300. Every image has a label, which defines to what class it belongs to. Decision tree is applied, with GINI index as its splitting criterion. This image file is converted to a csv using Java. For each csv file the R array, G array and B array is read. To determine the bin in which each it falls is found as follows: new value= (maximum(Array)- currrentvalue)/(maximum(Array)- minimum(Array))*16
2. The appropriate histograms were determined with this formula. Then a line (vector) is generated a line in the training data file for each image. The G array is then appended to R array and then the B array. Then the label is added at the end.

Operator	**Explanation**
ReadCSV	Reads the CSV files (test and train)
Apply Model	Applies an already learnt or trained model on a Sample Set.
Performance	Used for statistical performance evaluation of classification tasks.
Decision Tree	Generates a Decision Tree for classification of both nominal and numerical data.

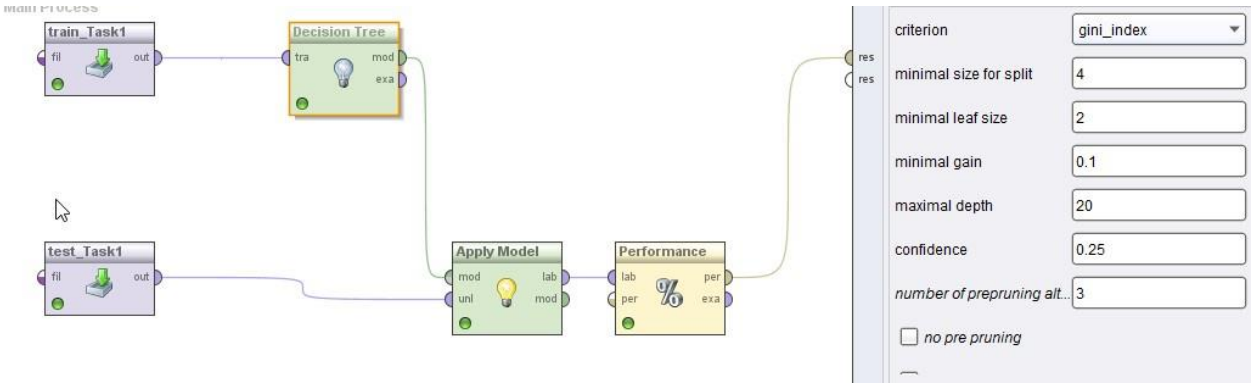

Fig : Design in RapidMiner

PerformanceVector

PerformanceVector:
accuracy: 89.17%
ConfusionMatrix:

True:	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6
Class_1:	14	0	0	2	0	0
Class_2:	0	19	0	0	0	0
Class_3:	1	0	17	0	0	0
Class_4:	2	1	0	18	1	0
Class_5:	0	0	1	0	19	0
Class_6:	3	0	2	0	0	20

Fig : Results using RapidMiner

Conclusion

The results did not change after tuning the parameters in the decision tree. The poor performance of this approach is due to the dimensionality of the feature vector (more irrelevant information).

TASK 2

The training file is read from the Task 1. For each data set a 48 length integer array (feature vector) is created initial value 0.

2. For each line the corresponding feature vector is updated as:

A 48 length histogram is represented as an array. An array is created for the dataset. The counter is set to zero. Iteration is done through the array, with index as the counter and the value is then updated for the histogram.

Now the array has a histogram feature vector that is consistent with the task 2 representation.

The above process is repeated for all the datasets and the corresponding feature vector is written to a new file with the label appended. The same processing is done for the test file as well.

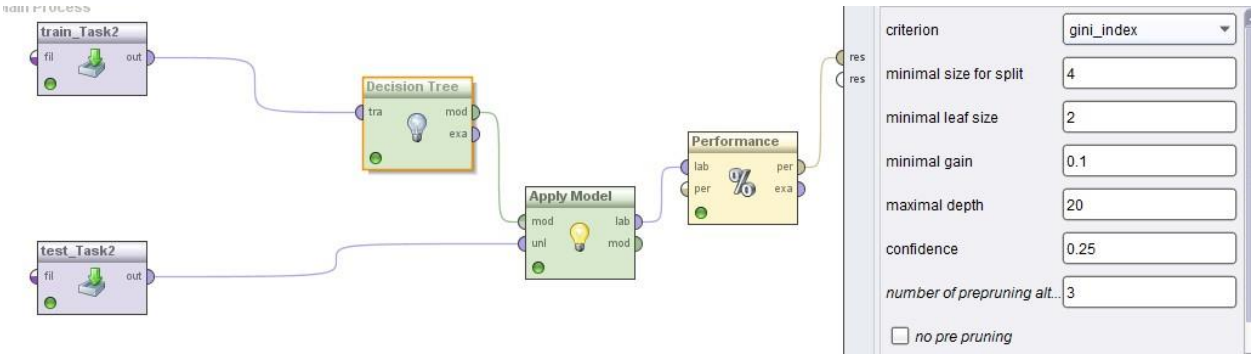

Fig : Design in RapidMiner

PerformanceVector

```
PerformanceVector:
accuracy: 97.50%
ConfusionMatrix:
True:   Class_1 Class_2 Class_3 Class_4 Class_5 Class_6
Class_1:        20      0       0       0       0       0
Class_2:        0       19      0       0       0       0
Class_3:        0       0       20      0       0       0
Class_4:        0       0       0       20      0       2
Class_5:        0       1       0       0       20      0
Class_6:        0       0       0       0       0       18
```

Operator	**Explanation**
ReadCSV	Reads the CSV files (test and train)
Apply Model	Applies an already learnt or trained model on a Sample Set.
Performance	Used for statistical performance evaluation of classification tasks.
Decision Tree	Generates a Decision Tree for classification of both nominal and numerical data.

Conclusion

There is an increase in accuracy as compared to approach 1 as it discards the irrelevant information. With storing the RGB values in bins, dimensionality is reduced of the feature vector and hence an increase in accuracy.

TASK 3

In the first step, the clustering is performed with k-means algorithm with the number of clusters being set to 8, 16 and 32 for each task. After clustering, intermediate files are generated that are used for classification.

Task3.py generates pixel data for clustering. Task3-fv.py generates the feature vector from clustered pixel.

Each data set generates 10x10 pixels of RGB. The pixel data is written to the train file without the label. This process is done for all the data sets. Similar preprocessing is done for the test file. The feature vector is represented as follows:

The feature vector belongs to R^k vector space k being the factor supplied in the clustering phase. ith component : frequency of ith cluster in the labelled data.

Feature vectors are generated for all training samples and then the label is appended.

Similar process is done for the testing data. The model is constructed using training data and directly applied on the test data to label the pixels.

Classification is done with the decision tree.

Operator	**Explanation**
ReadCSV	Reads the CSV files (test and train)
Apply Model	Applies an already learnt or trained model on a Sample Set.
Performance	Used for statistical performance evaluation of classification tasks.
Decision Tree	Generates a Decision Tree for classification of both nominal and numerical data.
Clustering	This operator performs clustering using the *k-means* algorithm.

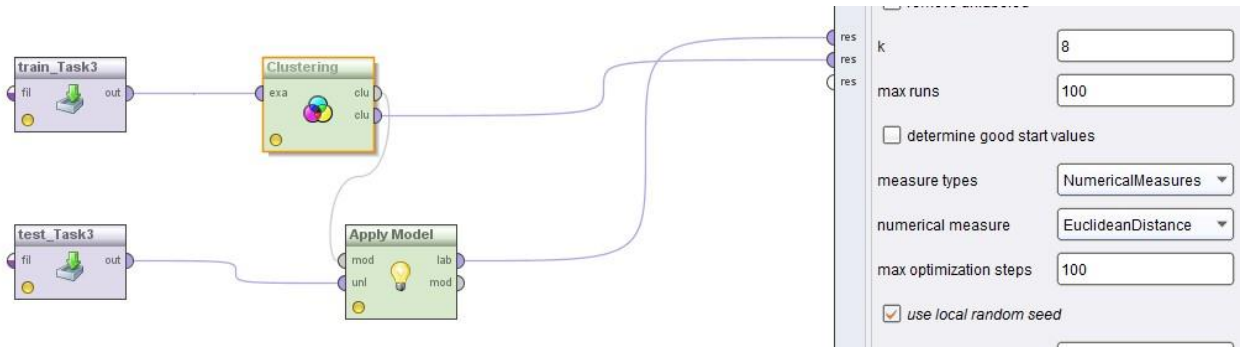

Fig : Clustering in RapidMiner

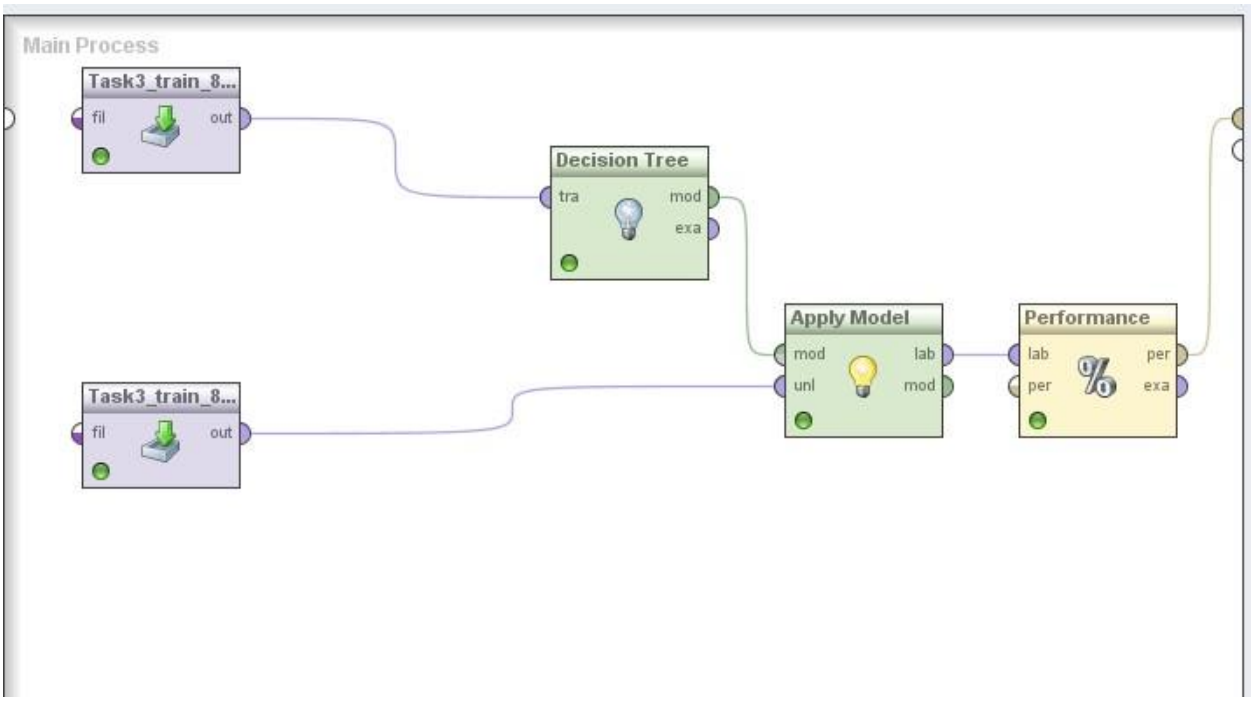

Fig : Classification in RapidMiner

Multiclass Classification Performance | Annotations

Table View | Plot View

accuracy: 100.00%

	true Class_1	true Class_2	true Class_3	true Class_4	true Class_5	true Class_6	class precision
pred. Class_1	20	0	0	0	0	0	100.00%
pred. Class_2	0	20	0	0	0	0	100.00%
pred. Class_3	0	0	20	0	0	0	100.00%
pred. Class_4	0	0	0	20	0	0	100.00%
pred. Class_5	0	0	0	0	20	0	100.00%
pred. Class_6	0	0	0	0	0	20	100.00%
class recall	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	

Fig : k=8

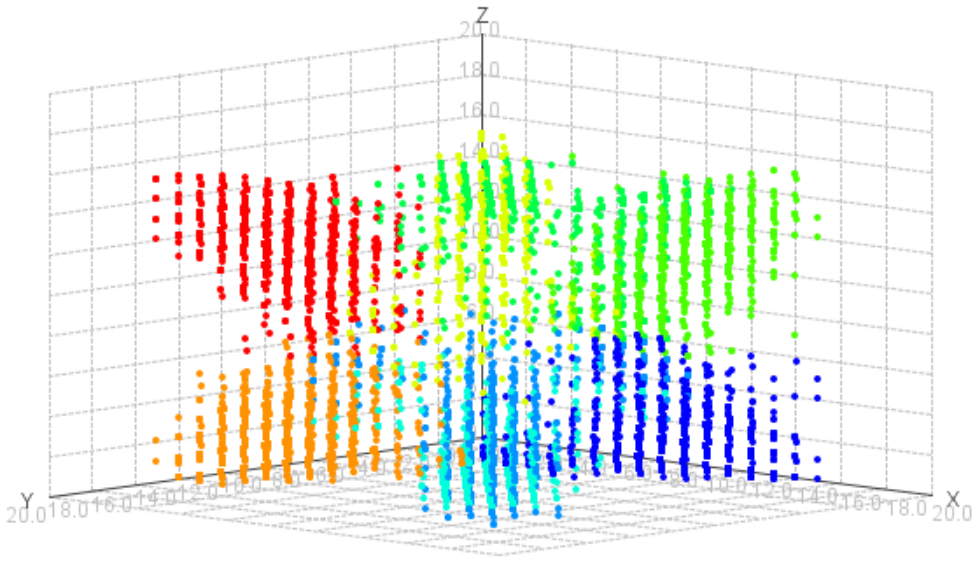

Fig : Scatter 3D plot

Conclusion

The high performance is because of taking into account statistics of the pixel color as it is by clustering it rather that individual intensity. As k=8 produced the best performance it shows that the clustering was drawn from 8 clusters and the constructed cluster model is the true one.