**Data Mining and Machine Learning**

Assignment 2

1. Introduction

In this report, we apply data mining techniques to classify German traffic sign images into ten classes.

1. Dataset Description

The dataset consists of about 12500 instances, each instance represents an image. The number of attributes is 2305 including the class labels. The class labels are numeric values from zero up to 9.

1. Preprocessing

3.1 Merging Attributes and class labels and reducing the size of the dataset.

In this project, MATLAB software is used for merging the features and targets datasets into one dataset. After that, we reduced the size of the dataset and used 40% of the dataset for training and 30% for testing. The MATLAB function dividerand is used to split the dataset into train and test datasets. Also, MATLAB was used to move 30% and 70% of the training data into test data to prepare new test datasets.

3.2 CSV to ARFF Conversion

After that, Weka ARFF viewer was used to convert the datasets from CSV format into ARFF format as show in Figure 1.

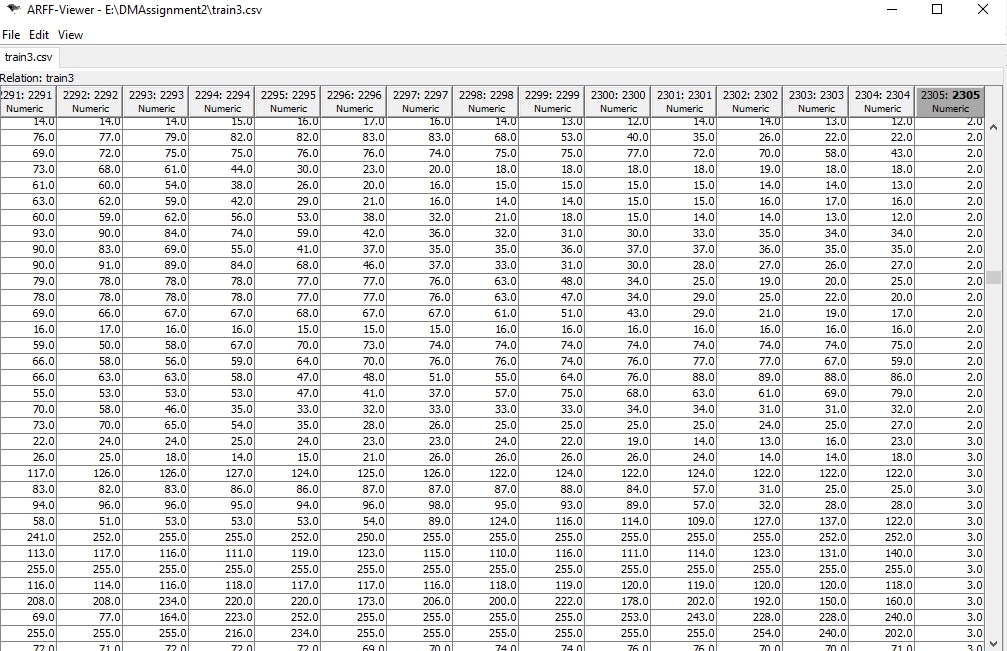


Figure CSV to ARFF Conversion using Weka ARFF Viewer Tool

* 1. Numeric to Nominal Conversion

Then the datasets were loaded into Weak explorer and the NumerToNominal filter was used to convert class labels values type from numeric into nominal. This is a very important step since some classifiers work only on nominal class labels.

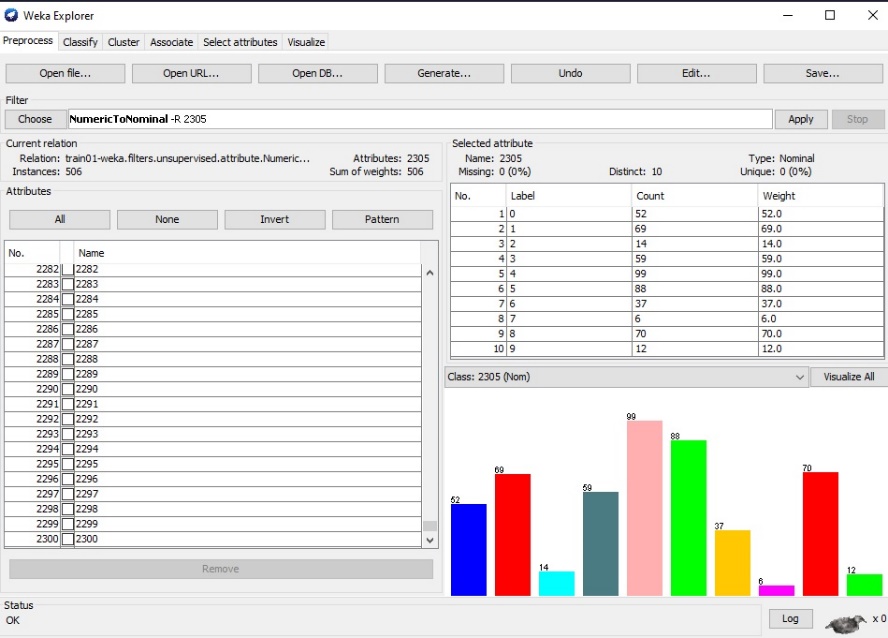


Figure Numeric to nominal conversion

* 1. Feature selection

Since the number of features is about 2305. This number of features will slow the training process of some classifiers such as the Multi-Layer Perceptron (MLP) neural network classifier. Thus, feature selection was carried out using Weka as shown in Figure 3.

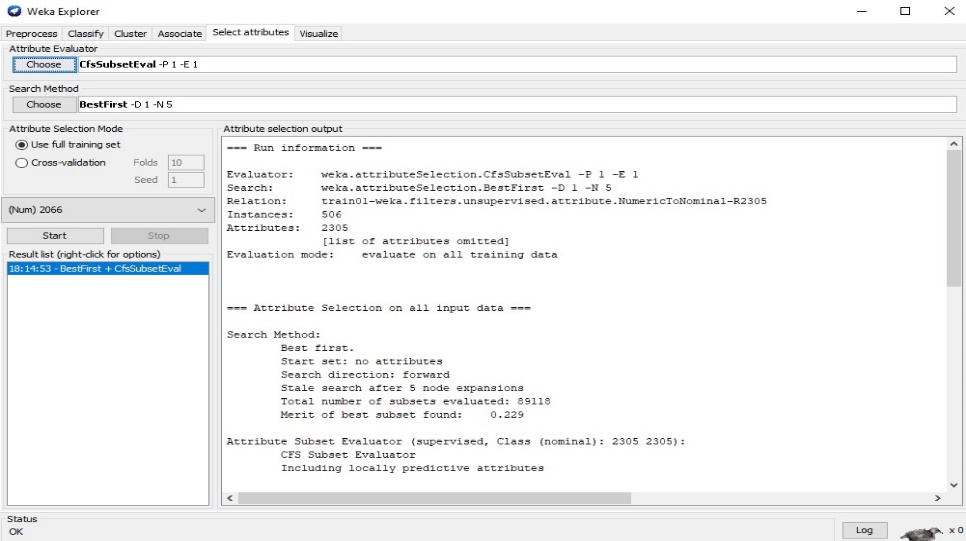


Figure Feature selection

1. Classification and Experimental Results

Five classifiers were used to predict the class labels. Also, Experiments with different configuration settings such as 10 Fold , Using test data, Moving 30% to test dataset, and Moving 70% to test dataset were carried out.

The classifiers are:

Decision trees

1. Classification using Decision Tree
   1. J48 Algorithm

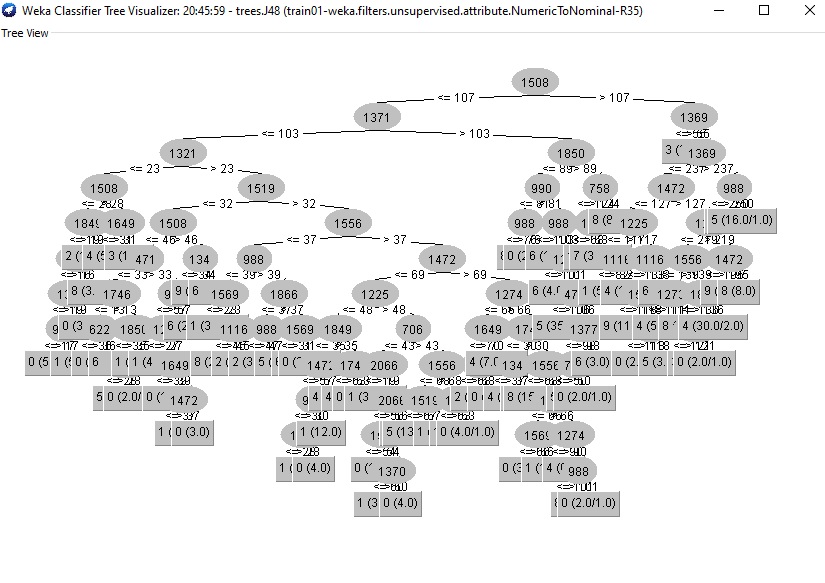


Figure J48 Decision Tree

Table 1 Performance of J48 Decision Tree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 Fold | Using test data | Moving 30% to test dataset | Moving 70% to test dataset |
| Accuracy | 59.8814 % | 64.4737 % | 64.4737 % | 63.0404 % |
| Precision | 0.321 | 0.346 | 0.346 | 0.407 |
| Recall | 0.295 | 0.462 | 0.462 | 0.511 |

* 1. Random Forest

Table 2 Performance of Random Forest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 Fold | Using test data | Moving 30% to test dataset | Moving 70% to test dataset |
| Accuracy | 74.9012 % | 78.1579 % | 78.1579 % | 81.311 % |
| Precision | 0.490 | 0.500 | 0.500 | 0.632 |
| Recall | 0.410 | 0.487 | 0.487 | 0.585 |

* 1. Custom Decision Tree

The custom decision tree selected was the Random decision tree as shown in Figure 5.

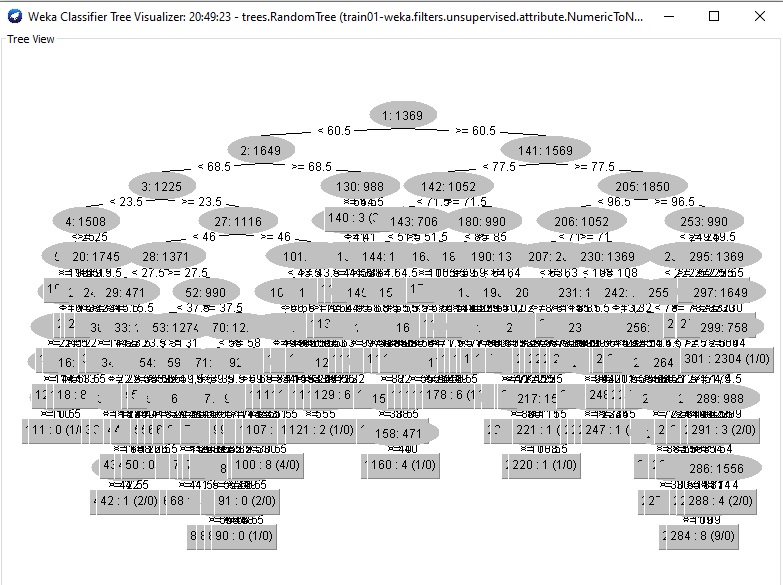


Figure Random Tree

Table 3 Performance of Custom Decision Tree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 Fold | Using test data | Moving 30% to test dataset | Moving 70% to test dataset |
| Accuracy | 55.9289 % | 60.2632 % | 60.2632 % | 61.5063 % |
| Precision | 0.291 | 0.394 | 0.394 | 0.435 |
| Recall | 0.377 | 0.333 | 0.333 | 0.426 |

Part -2 Linear Classifiers and Neural Networks

1. Classification using Linear classification and Neural Network
   1. Linear classification

Table 4 Performance of Linear Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 Fold | Using test data | Moving 30% to test dataset | Moving 70% to test dataset |
| Accuracy | 75.2964 % | 79.4737 % | 75.9124 % | 76.9874 % |
| Precision | 0.333 | 0.400 | 0.382 | 0.507 |
| Recall | 0.279 | 0.359 | 0.323 | 0.404 |

* 1. Multilayer Perceptron (MLP) Classifier

Table 5 Performance of Multilayer Perceptron Neural Network

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 Fold | Using test data | Moving 30% to test dataset | Moving 70% to test dataset |
| Accuracy | 66.996 % | 93.1579 % | 61.4964 % | 61.7852 % |
| Precision | 0.467 | 0.969 | 0.367 | 0.432 |
| Recall | 0.344 | 0.795 | 0.169 | 0.170 |

**Discussion**

Experimental Results show that the Multilayer Perceptron Neural Network achieves the highest classification accuracy of up to about 93.16% outperforming other classifiers.

Another finding is that the Multilayer Perceptron Neural Network achieved the highest accuracy when the classifiers was tested on a separate test dataset and moving some instances from train into test has no effect on the performance of the classifier.

Another finding is that the tested classifiers behave differently for different experiments. For instance, Random Forest classifier achieved the highest classification accuracy with the last experiment when the test dataset is a merge of the original test dataset and 70% of the train dataset. This may indicate that there is a overfitting.

**Recommendations**

Although the traditional feed-forward neural network used in this project achieved a good classification accuracy, it has been trained on dataset of small size. Furthermore, the achieved accuracy is not as high as optimal.

Therefore, a convolutional neural network (CNN) will be the right choice for classifying instances of these dataset.

the CNN will achieve a high classification accuracy as the CNN has achieved higher accuracy in most of computer vision applications recently.

CNN is a deep learning network consists of one or more convolutional layers, one or more pooling layers, and one fully connected layer.

The convolutional layer takes images as input, filters the images, and produces feature maps. The number of feature maps is reduced by the pooling layer. The fully connected layer learns the non-linear relationships between input images and the class labels and performs the classification task.

The limitation of deep learning network is that to produce good results, it requires large dataset.

Another limitation of deep learning network is the high computation and storage cost required by the network.

A solution for the last limitation is implementing the deep learning network as a distributed network.