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.

2. 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.

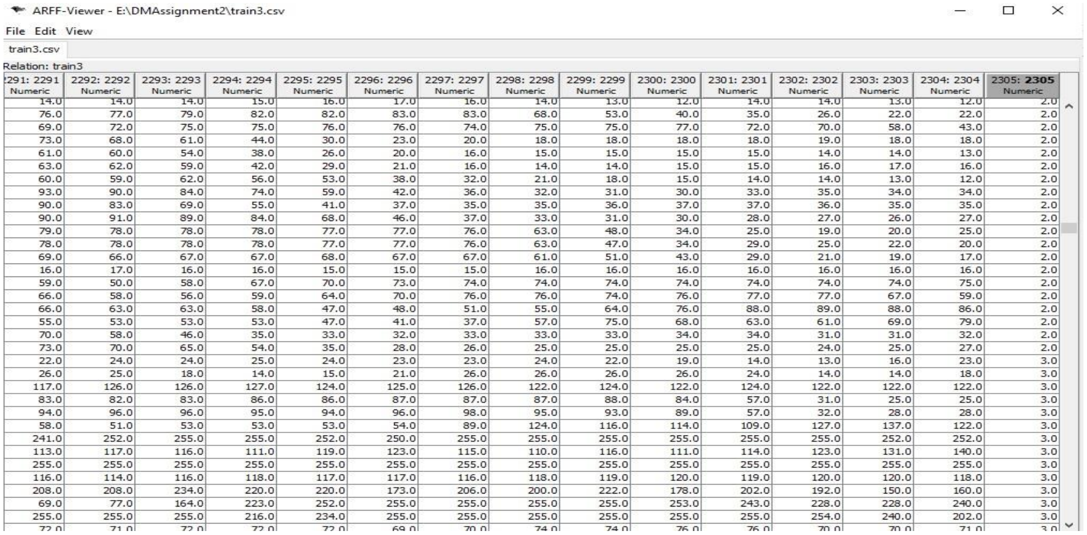
3. Preprocessing

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.



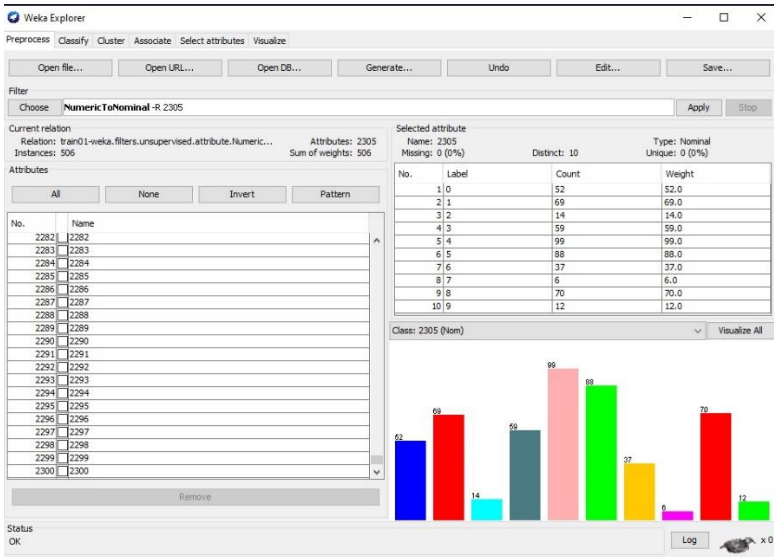
Relation: train3

291: 2291 Numeric	2292: 2292 Numeric	2293: 2293 Numeric	2294: 2294 Numeric	2295: 2295 Numeric	2296: 2296 Numeric	2297: 2297 Numeric	2298: 2298 Numeric	2299: 2299 Numeric	2300: 2300 Numeric	2301: 2301 Numeric	2302: 2302 Numeric	2303: 2303 Numeric	2304: 2304 Numeric	2305: **2305** Numeric
14.0	14.0	14.0	15.0	16.0	17.0	16.0	14.0	13.0	12.0	14.0	14.0	13.0	12.0	2.0
76.0	77.0	79.0	82.0	82.0	83.0	83.0	68.0	53.0	40.0	35.0	26.0	22.0	22.0	2.0
69.0	72.0	75.0	75.0	76.0	76.0	74.0	75.0	75.0	77.0	72.0	70.0	58.0	43.0	2.0
73.0	68.0	61.0	44.0	30.0	23.0	20.0	18.0	18.0	18.0	18.0	19.0	18.0	18.0	2.0
61.0	60.0	54.0	38.0	26.0	20.0	16.0	15.0	15.0	15.0	15.0	14.0	14.0	13.0	2.0
63.0	62.0	59.0	42.0	29.0	21.0	16.0	14.0	14.0	15.0	15.0	16.0	17.0	16.0	2.0
60.0	59.0	62.0	56.0	53.0	38.0	32.0	21.0	18.0	15.0	14.0	14.0	13.0	12.0	2.0
93.0	90.0	84.0	74.0	59.0	42.0	36.0	32.0	31.0	30.0	33.0	35.0	34.0	34.0	2.0
90.0	83.0	69.0	55.0	41.0	37.0	35.0	35.0	36.0	37.0	37.0	36.0	35.0	35.0	2.0
90.0	91.0	89.0	84.0	68.0	46.0	37.0	33.0	31.0	30.0	28.0	27.0	26.0	27.0	2.0
79.0	78.0	78.0	78.0	77.0	77.0	76.0	63.0	48.0	34.0	25.0	19.0	20.0	25.0	2.0
78.0	78.0	78.0	78.0	77.0	77.0	76.0	63.0	47.0	34.0	29.0	25.0	22.0	20.0	2.0
69.0	66.0	67.0	67.0	68.0	67.0	67.0	61.0	51.0	43.0	29.0	21.0	19.0	17.0	2.0
16.0	17.0	16.0	16.0	15.0	15.0	15.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	2.0
59.0	50.0	58.0	67.0	70.0	73.0	74.0	74.0	74.0	74.0	74.0	74.0	74.0	75.0	2.0
66.0	58.0	56.0	59.0	64.0	70.0	76.0	76.0	74.0	76.0	77.0	77.0	67.0	59.0	2.0
66.0	63.0	63.0	58.0	47.0	48.0	51.0	55.0	64.0	76.0	88.0	89.0	88.0	86.0	2.0
55.0	53.0	53.0	53.0	47.0	41.0	37.0	57.0	75.0	68.0	63.0	61.0	69.0	79.0	2.0
70.0	58.0	46.0	35.0	33.0	32.0	33.0	33.0	33.0	34.0	34.0	31.0	31.0	32.0	2.0
73.0	70.0	65.0	54.0	35.0	28.0	26.0	25.0	25.0	25.0	25.0	24.0	25.0	27.0	2.0
22.0	24.0	24.0	25.0	24.0	23.0	23.0	24.0	22.0	19.0	14.0	13.0	16.0	23.0	3.0
26.0	25.0	18.0	14.0	15.0	21.0	26.0	26.0	26.0	26.0	24.0	14.0	14.0	18.0	3.0
117.0	126.0	126.0	127.0	124.0	125.0	126.0	122.0	124.0	122.0	124.0	122.0	122.0	122.0	3.0
83.0	82.0	83.0	86.0	86.0	87.0	87.0	87.0	88.0	84.0	57.0	31.0	25.0	25.0	3.0
94.0	96.0	96.0	95.0	94.0	96.0	98.0	95.0	93.0	89.0	57.0	32.0	28.0	28.0	3.0
58.0	51.0	53.0	53.0	53.0	54.0	89.0	124.0	116.0	114.0	109.0	127.0	137.0	122.0	3.0
241.0	252.0	255.0	255.0	252.0	250.0	255.0	255.0	255.0	255.0	255.0	255.0	252.0	252.0	3.0
113.0	117.0	116.0	111.0	119.0	123.0	115.0	110.0	116.0	111.0	114.0	123.0	131.0	140.0	3.0
255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	3.0
116.0	114.0	116.0	118.0	117.0	117.0	116.0	118.0	119.0	120.0	119.0	120.0	120.0	118.0	3.0
208.0	208.0	234.0	220.0	220.0	173.0	206.0	200.0	222.0	178.0	202.0	192.0	150.0	160.0	3.0
69.0	77.0	164.0	223.0	252.0	255.0	255.0	255.0	255.0	253.0	243.0	228.0	228.0	240.0	3.0
255.0	255.0	255.0	216.0	234.0	255.0	255.0	255.0	255.0	255.0	255.0	254.0	240.0	202.0	3.0
72.0	71.0	72.0	72.0	72.0	69.0	70.0	74.0	74.0	76.0	76.0	70.0	70.0	71.0	3.0

Figure 1 CSV to ARFF Conversion using Weka ARFF Viewer Tool

3.3 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.



NumericToNominal -R 2305

Current relation
Relation: train01-weka.filters.unsupervised.attribute.Numeric... Attributes: 2305
Instances: 506 Sum of weights: 506

Attributes

No.		Name
2282	□	2282
2283	□	2283
2284	□	2284
2285	□	2285
2286	□	2286
2287	□	2287
2288	□	2288
2289	□	2289
2290	□	2290
2291	□	2291
2292	□	2292
2293	□	2293
2294	□	2294
2295	□	2295
2296	□	2296
2297	□	2297
2298	□	2298
2299	□	2299
2300	□	2300

Selected attribute
Name: 2305 Type: Nominal
Missing: 0 (0%) Distinct: 10 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	52	52.0
2	1	69	69.0
3	2	14	14.0
4	3	59	59.0
5	4	99	99.0
6	5	88	88.0
7	6	37	37.0
8	7	6	6.0
9	8	70	70.0
10	9	12	12.0

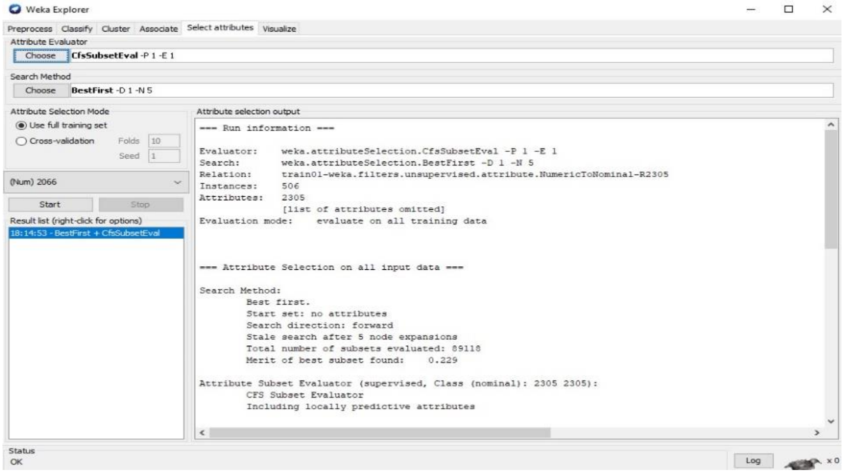
Class: 2305 (Nom)

Status
OK

Figure 2 Numeric to nominal conversion

3.4 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.



Attribute Evaluator: **CfsSubsetEval** -P 1 -E 1

Search Method: **BestFirst** -D 1 -N 5

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Num) 2066

Result list (right-click for options)

18:14:53 - BestFirst + CfsSubsetEval

Attribute selection output

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval -P 1 -E 1
Search:       weka.attributeSelection.BestFirst -D 1 -N 5
Relation:     train01-weka.filters.unsupervised.attribute.NumericToNominal-R2305
Instances:    506
Attributes:   2305
              [list of attributes omitted]
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 89118
        Merit of best subset found:    0.229

Attribute Subset Evaluator (supervised, Class (nominal): 2305 2305):
        CFS Subset Evaluator
        Including locally predictive attributes
```

Status
OK

Figure 3 Feature selection

4. 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

1.1.J48 Algorithm

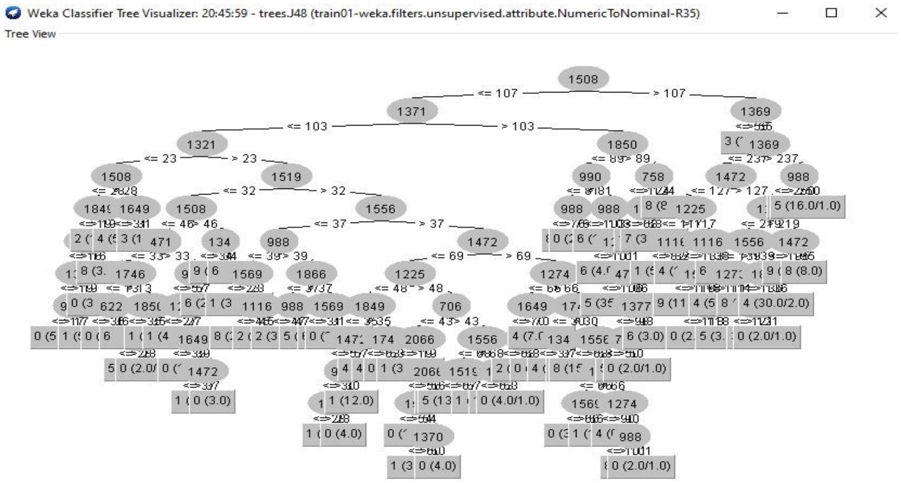


Figure 4 J48 Decision Tree

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

1.2.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

1.3.Custom Decision Tree

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

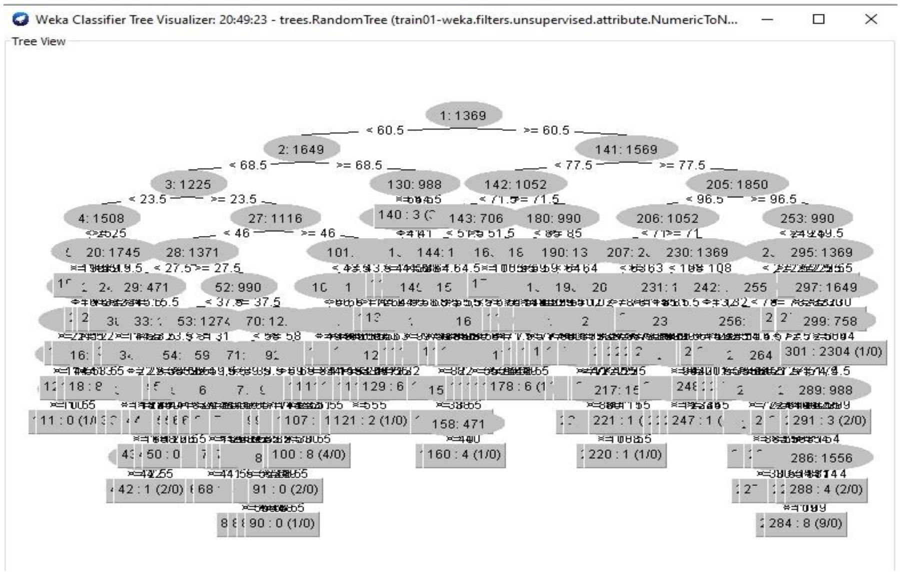


Figure 5 Random Tree

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

2. Classification using Linear classification and Neural Network

2.1.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

2.2.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.