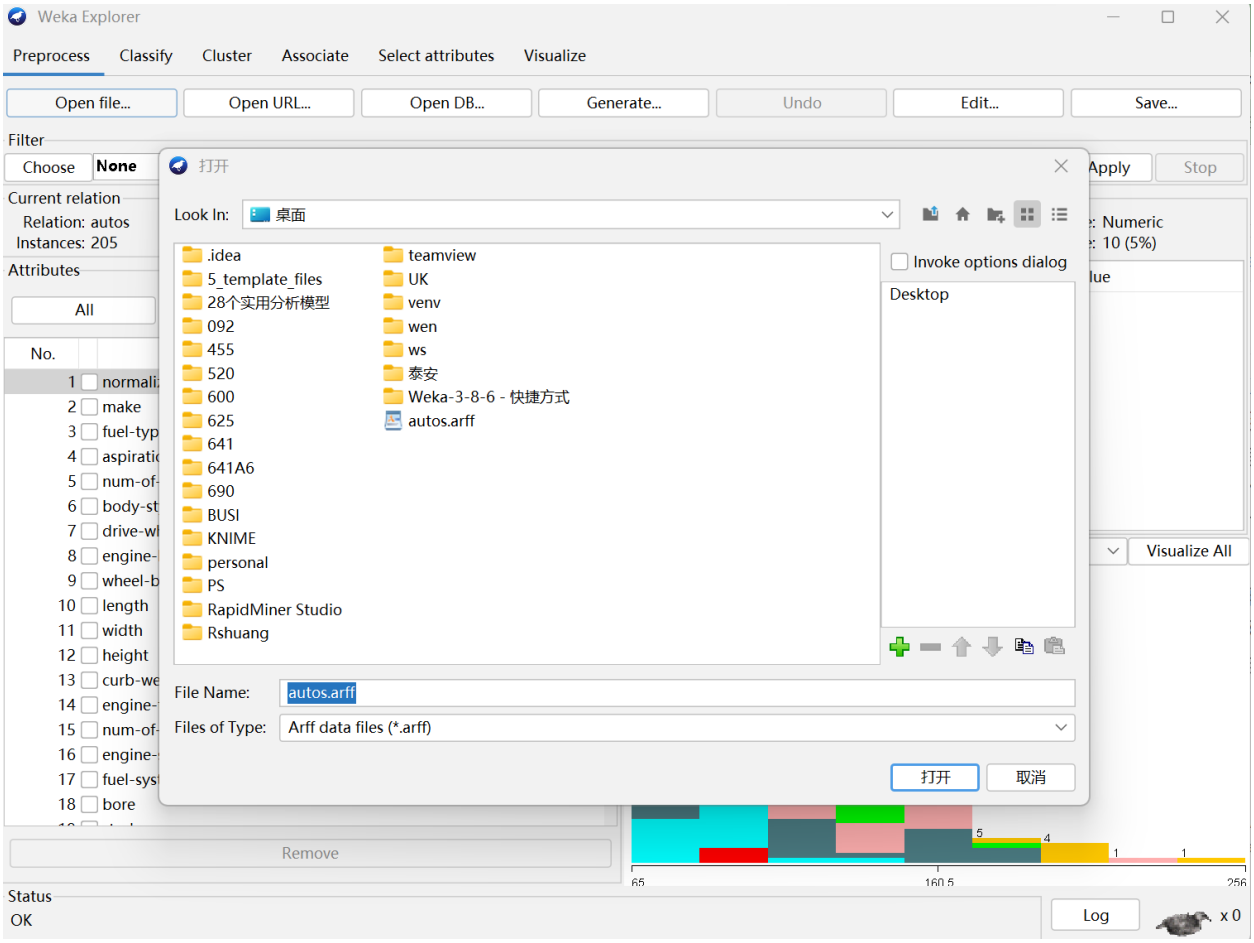
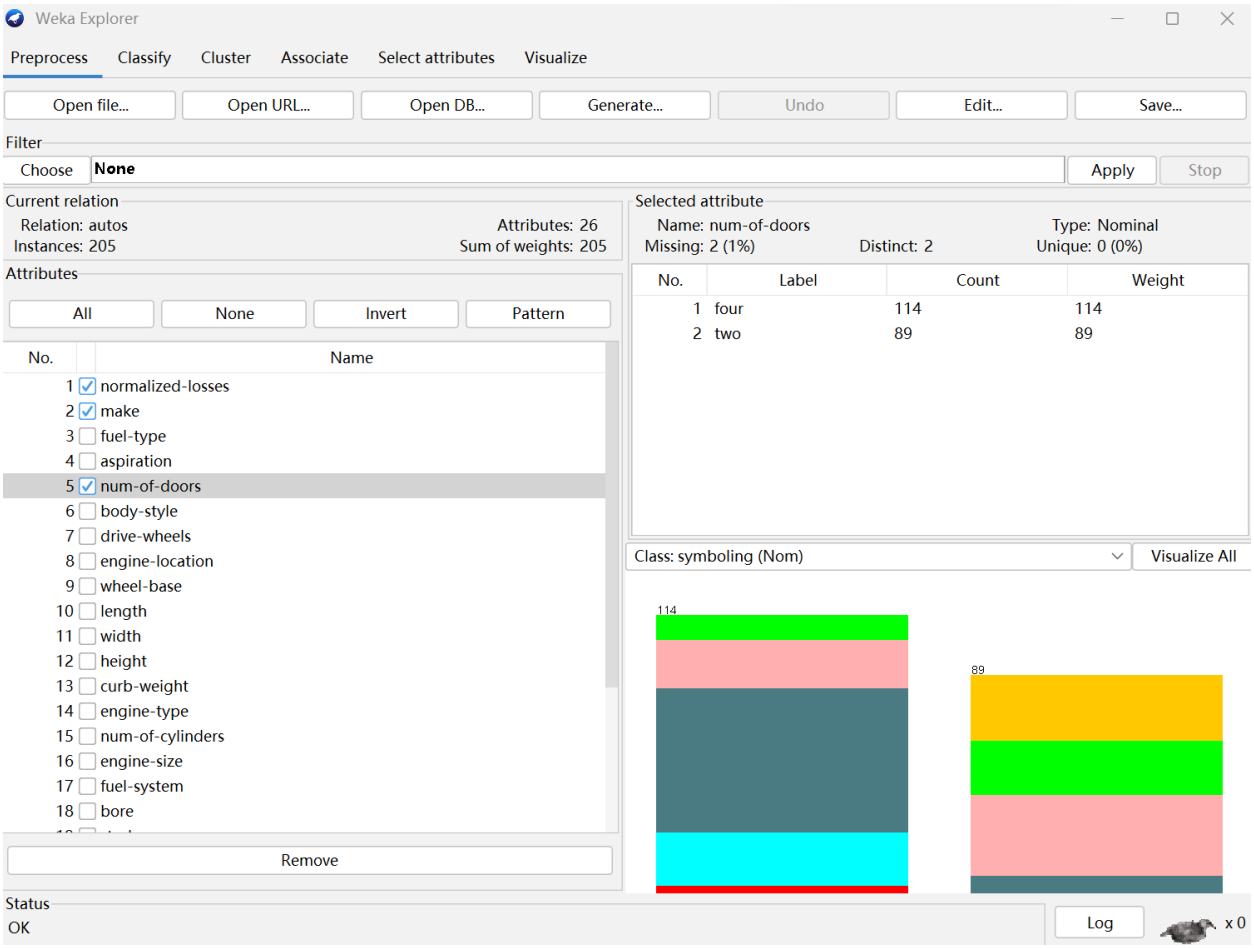
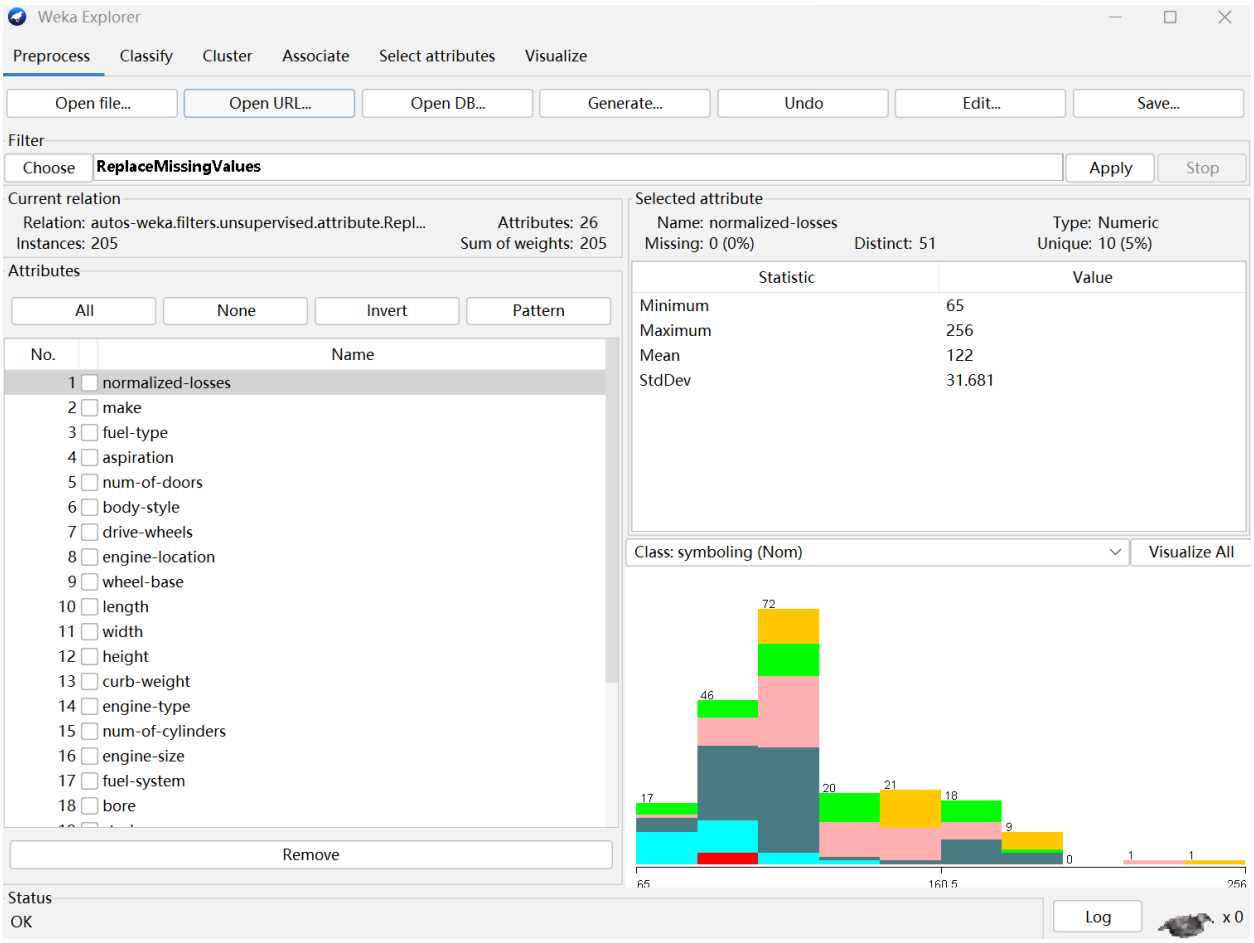
WEKA Challenge

After studying, I started my WEKA challenge. The first challenge I encountered was finding the ARFF files, which I had never used before. I searched and downloaded the dataset for my challenge (autos) from OpenML (an open machine learning platform). The dataset consists of three types of entities: (a) specifications of various features of the car, (b) specified insurance Risk Rating, (c) Normalized loss of use compared to other cars.

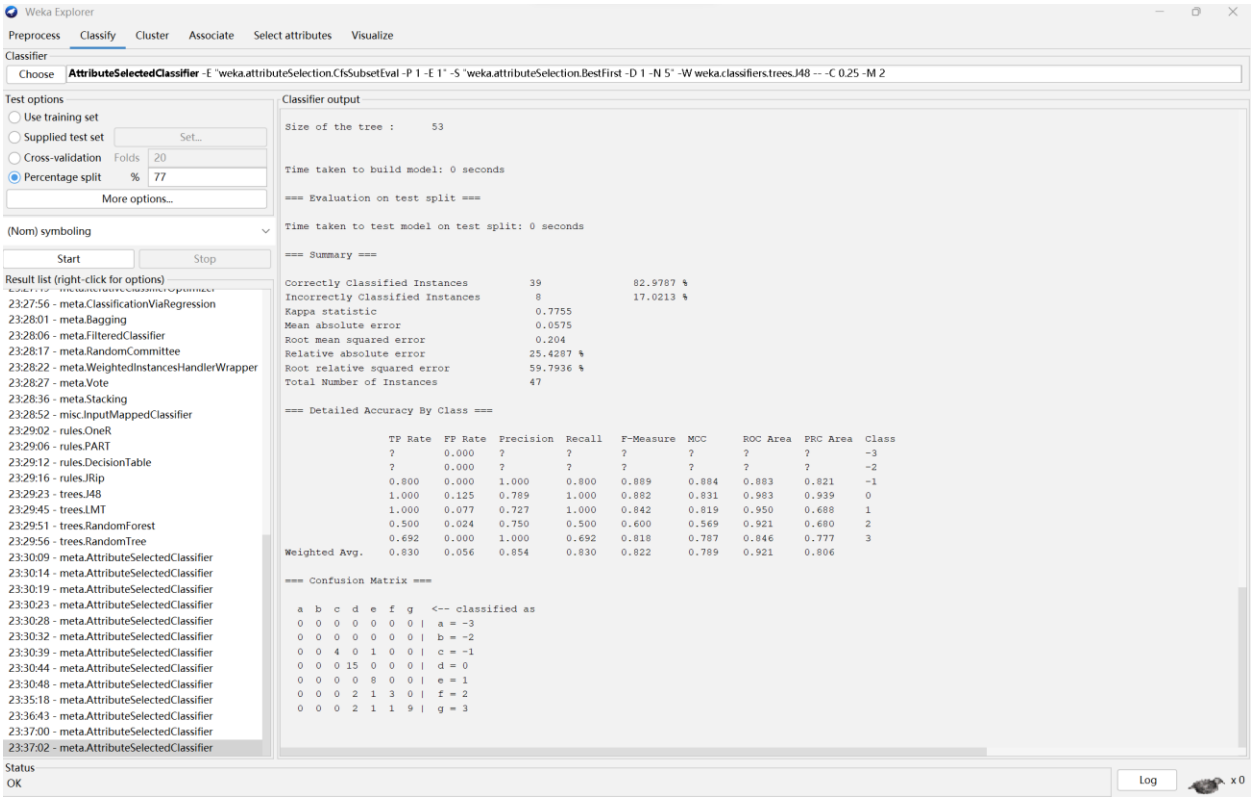


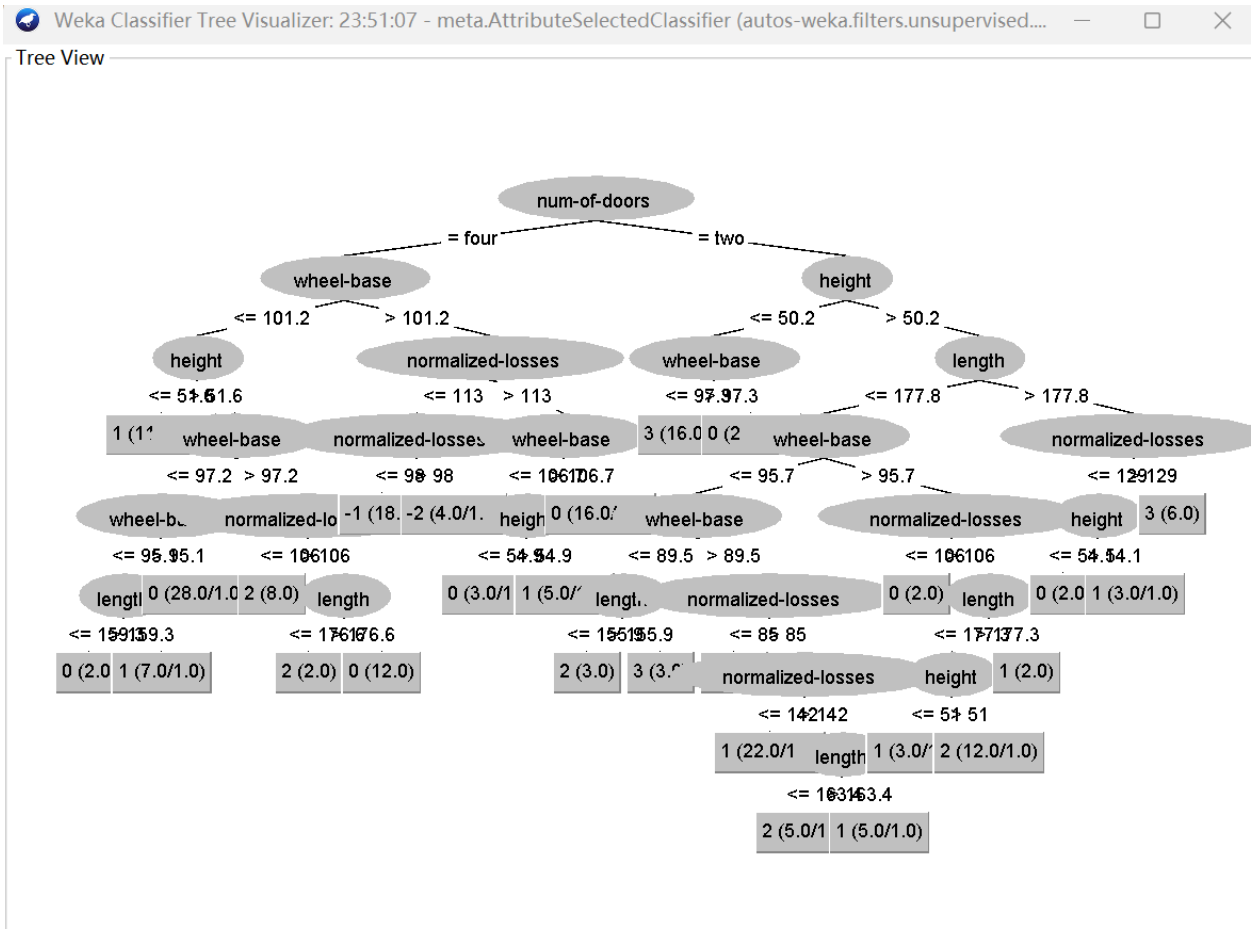
After loading the document into WEKA I need to perform data processing first and we can see that the file has missing values. We can choose unsupervised attributes to replace missing values. ReplaceMissingValues provides a variety of methods to deal with missing values, including using mean, median, mode, fixed value or custom strategy. This flexibility helps choose the most suitable data Alternative methods for gathering and analyzing requirements. The benefit of this is that it helps maintain data integrity, ensuring that every instance in the data set contains complete information.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: autos
Instances: 205
Attributes: 26
Sum of weights: 205
Attributes
All
None
Invert
Pattern
No.
Name
1 normalized-losses
2 make
3 fuel-type
4 aspiration
5 num-of-doors
6 body-style
7 drive-wheels
8 engine-location
9 wheel-base
10 length
11 width
12 height
13 curb-weight
14 engine-type
15 num-of-cylinders
16 engine-size
17 fuel-system
18 bore
Remove
Selected attribute
Name: num-of-doors
Missing: 2 (1%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 four 114 114
2 two 89 89
Class: symboling (Nom)
Visualize All
114
89
Status
OK
Log
x 0

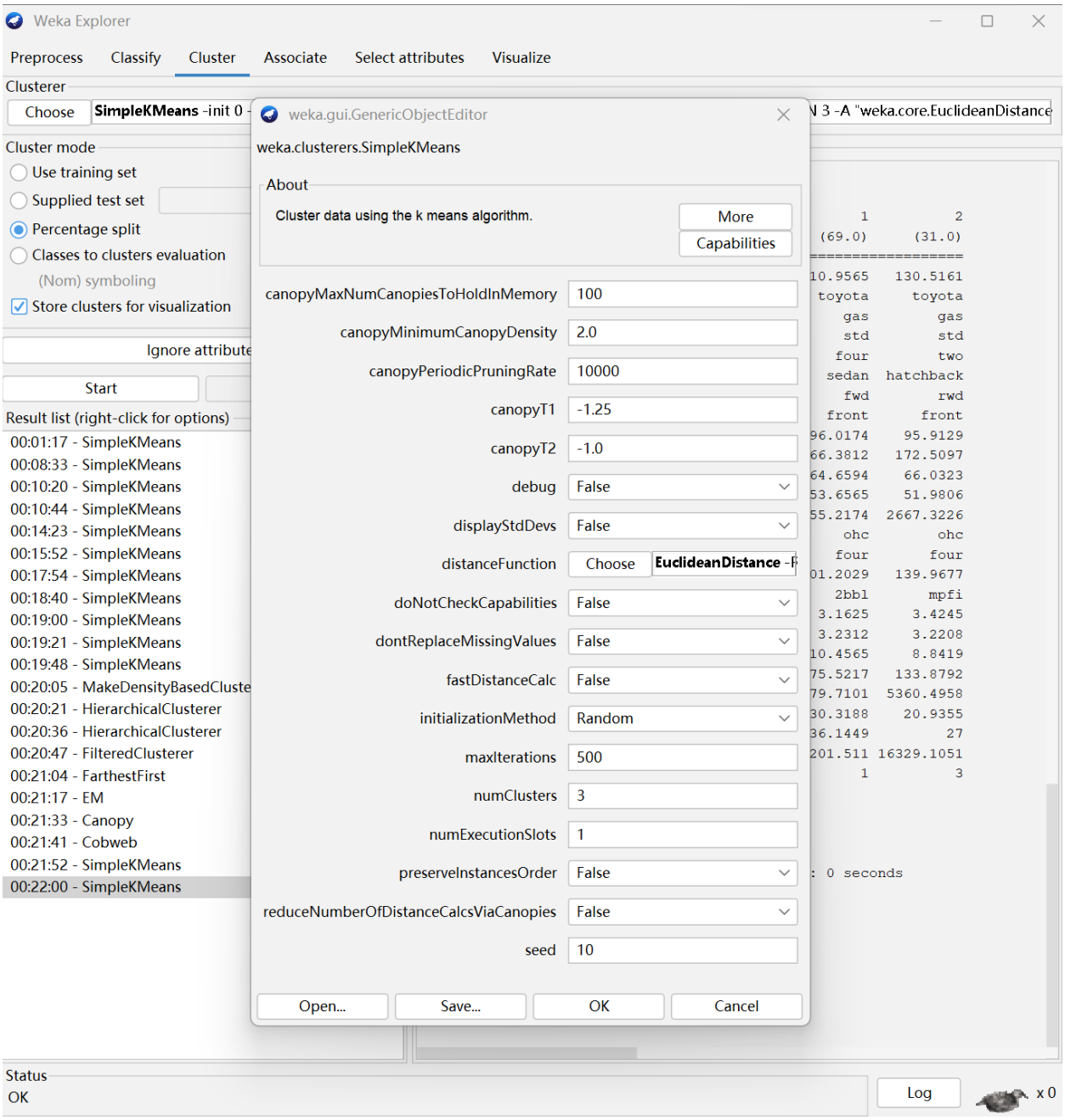


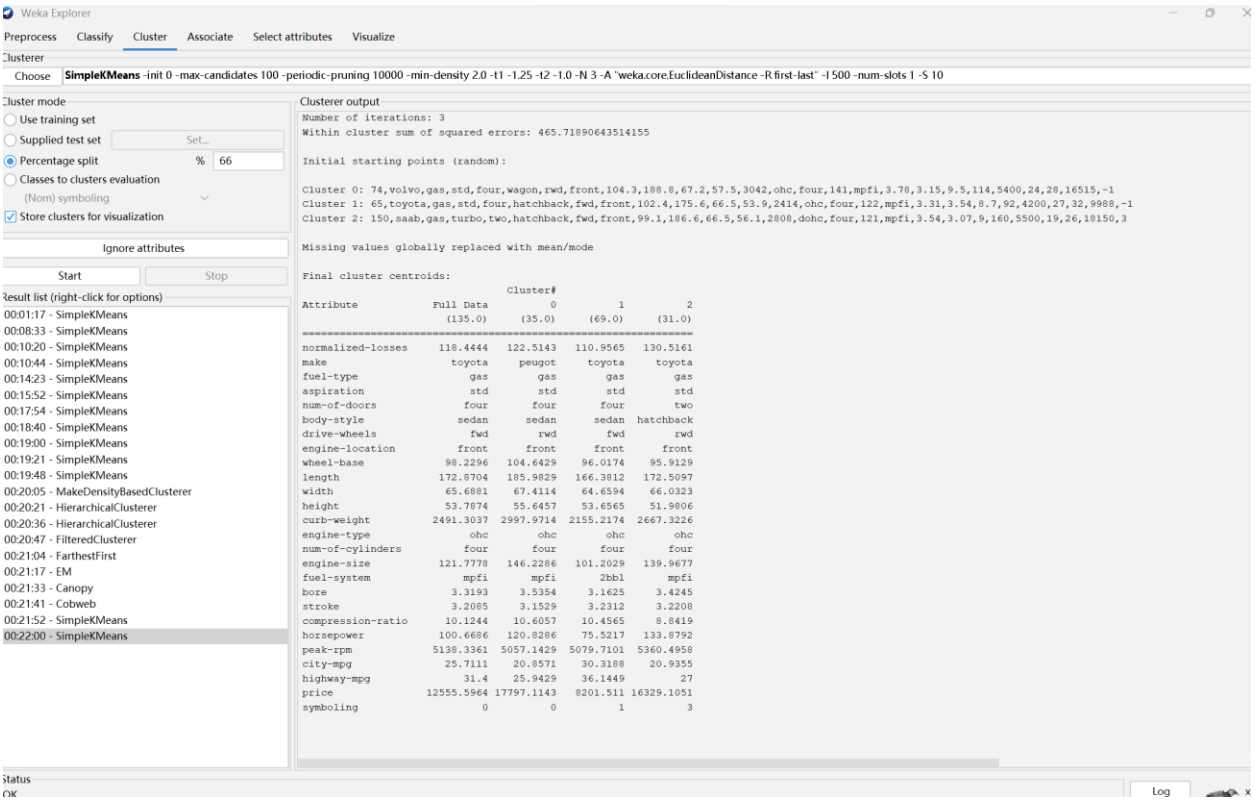
Then I wanted to classify the data. I tried many methods, but the percentage of Correctly Classified Instances was very low. As you can see from the picture, I finally chose AttributeSelectedClassifier for classification. This classification model uses a composite approach, including feature selection and J48 decision trees, to classify the test data and performs well overall. The model selected attributes with predictive capabilities through feature selection, and then used decision trees for classification. It achieved a correct classification rate of about 83%, and the Kappa statistic was 0.7755, indicating that the model has good performance. The confusion matrix provides classification results for different categories and can be used to evaluate the performance of the model in detail. In addition, we can also choose the visualization function to visualize the classification model. We can see the decision tree model and explore the rules and branches of different nodes, which can help us understand how the model classifies data.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
AttributeSelectedClassifier -E "weka.attributeSelection.CfsSubsetEval -P 1 -E 1" -S "weka.attributeSelection.BestFirst -D 1 -N 5" -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
20
Percentage split
%
77
More options...
(Nom) symboling
Start
Stop
Result list (right-click for options)
23:27:56 - meta.ClassificationViaRegression
23:28:01 - meta.Bagging
23:28:06 - meta.FilteredClassifier
23:28:17 - meta.RandomCommittee
23:28:22 - meta.WeightedInstancesHandlerWrapper
23:28:27 - meta.Vote
23:28:36 - meta.Stacking
23:28:52 - misc.InputMappedClassifier
23:29:02 - rules.OneR
23:29:06 - rules.PART
23:29:12 - rules.DecisionTable
23:29:16 - rules.JRip
23:29:23 - trees.J48
23:29:45 - trees.LMT
23:29:51 - trees.RandomForest
23:29:56 - trees.RandomTree
23:30:09 - meta.AttributeSelectedClassifier
23:30:14 - meta.AttributeSelectedClassifier
23:30:19 - meta.AttributeSelectedClassifier
23:30:23 - meta.AttributeSelectedClassifier
23:30:28 - meta.AttributeSelectedClassifier
23:30:32 - meta.AttributeSelectedClassifier
23:30:39 - meta.AttributeSelectedClassifier
23:30:44 - meta.AttributeSelectedClassifier
23:30:48 - meta.AttributeSelectedClassifier
23:35:18 - meta.AttributeSelectedClassifier
23:36:43 - meta.AttributeSelectedClassifier
23:37:00 - meta.AttributeSelectedClassifier
23:37:02 - meta.AttributeSelectedClassifier
Classifier output
Size of the tree : 53
Time taken to build model: 0 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 39 82.9787 %
Incorrectly Classified Instances 8 17.0213 %
Kappa statistic 0.7755
Mean absolute error 0.0575
Root mean squared error 0.204
Relative absolute error 25.4287 %
Root relative squared error 59.7936 %
Total Number of Instances 47
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
? 0.000 ? ? ? ? ? ? -3
? 0.000 ? ? ? ? ? ? -2
0.800 0.000 1.000 0.800 0.889 0.884 0.883 0.821 -1
1.000 0.125 0.789 1.000 0.882 0.831 0.983 0.939 0
1.000 0.077 0.727 1.000 0.842 0.819 0.950 0.688 1
0.500 0.024 0.750 0.500 0.600 0.569 0.921 0.680 2
0.692 0.000 1.000 0.692 0.818 0.787 0.846 0.777 3
Weighted Avg. 0.830 0.056 0.854 0.830 0.822 0.789 0.921 0.806
=== Confusion Matrix ===
a b c d e f g <-- classified as
0 0 0 0 0 0 0 | a = -3
0 0 0 0 0 0 0 | b = -2
0 0 4 0 1 0 0 | c = -1
0 0 0 15 0 0 0 | d = 0
0 0 0 0 8 0 0 | e = 1
0 0 0 2 1 3 0 | f = 2
0 0 0 2 1 1 9 | g = 3
Status
OK
Log
x 0



I then wanted to cluster this data, and I chose the k-Means clustering algorithm because it works well with large data sets and is usually able to complete the clustering task in a relatively short time. From the clustering results, the data set is divided into three different clusters (Cluster 0, Cluster 1, Cluster 2), each cluster contains a different number of instances. For each cluster, you see the average values of the individual attributes (Final cluster centroids), and these values reflect the characteristics of each cluster. For example, you can compare attributes such as average vehicle price, engine size, and horsepower among different clusters. By analyzing the average value of each cluster and the composition of vehicle models within the cluster, we can identify similar car groups, which can help with decision-making and insights in application areas such as market positioning, product classification, and customer analysis. However, looking at WCSS, its number is larger, which means that the data points in the clusters are not close enough. According to the visualization, it can also be seen that each cluster is not close enough. I also tried other clustering algorithms, but none of them performed very well.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -
N 3 -A "weka.core.EuclideanDistance
Cluster mode
Use training set
Supplied test set
Percentage split
Classes to clusters evaluation
(Nom) symboling
Store clusters for visualization
Ignore attribute
Start
Result list (right-click for options)
00:01:17 - SimpleKMeans
00:08:33 - SimpleKMeans
00:10:20 - SimpleKMeans
00:10:44 - SimpleKMeans
00:14:23 - SimpleKMeans
00:15:52 - SimpleKMeans
00:17:54 - SimpleKMeans
00:18:40 - SimpleKMeans
00:19:00 - SimpleKMeans
00:19:21 - SimpleKMeans
00:19:48 - SimpleKMeans
00:20:05 - MakeDensityBasedCluste
00:20:21 - HierarchicalClusterer
00:20:36 - HierarchicalClusterer
00:20:47 - FilteredClusterer
00:21:04 - FarthestFirst
00:21:17 - EM
00:21:33 - Canopy
00:21:41 - Cobweb
00:21:52 - SimpleKMeans
00:22:00 - SimpleKMeans
weka.gui.GenericObjectEditor
weka.clusterers.SimpleKMeans
About
Cluster data using the k means algorithm.
More
Capabilities
canopyMaxNumCanopiesToHoldInMemory
100
canopyMinimumCanopyDensity
2.0
canopyPeriodicPruningRate
10000
canopyT1
-1.25
canopyT2
-1.0
debug
False
displayStdDevs
False
distanceFunction
Choose
EuclideanDistance -R
doNotCheckCapabilities
False
dontReplaceMissingValues
False
fastDistanceCalc
False
initializationMethod
Random
maxIterations
500
numClusters
3
numExecutionSlots
1
preserveInstancesOrder
False
reduceNumberOfDistanceCalcsViaCanopies
False
seed
10
Open...
Save...
OK
Cancel
1 2
(69.0) (31.0)
10.9565 130.5161
toyota toyota
gas gas
std std
four two
sedan hatchback
fwd rwd
front front
96.0174 95.9129
66.3812 172.5097
64.6594 66.0323
53.6565 51.9806
55.2174 2667.3226
ohc ohc
four four
01.2029 139.9677
2bbl mpfi
3.1625 3.4245
3.2312 3.2208
10.4565 8.8419
75.5217 133.8792
79.7101 5360.4958
30.3188 20.9355
36.1449 27
201.511 16329.1051
1 3
: 0 seconds
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) symboling
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
00:01:17 - SimpleKMeans
00:08:33 - SimpleKMeans
00:10:20 - SimpleKMeans
00:10:44 - SimpleKMeans
00:14:23 - SimpleKMeans
00:15:52 - SimpleKMeans
00:17:54 - SimpleKMeans
00:18:40 - SimpleKMeans
00:19:00 - SimpleKMeans
00:19:21 - SimpleKMeans
00:19:48 - SimpleKMeans
00:20:05 - MakeDensityBasedClusterer
00:20:21 - HierarchicalClusterer
00:20:36 - HierarchicalClusterer
00:20:47 - FilteredClusterer
00:21:04 - FarthestFirst
00:21:17 - EM
00:21:33 - Canopy
00:21:41 - Cobweb
00:21:52 - SimpleKMeans
00:22:00 - SimpleKMeans
Clusterer output
Number of iterations: 3
Within cluster sum of squared errors: 465.71890643514155
Initial starting points (random):
Cluster 0: 74,volvo,gas,std,four,wagon,rwd,front,104.3,188.8,67.2,57.5,3042,ohc,four,141,mpfi,3.78,3.15,9.5,114,5400,24,28,16515,-1
Cluster 1: 65,toyota,gas,std,four,hatchback,fwd,front,102.4,175.6,66.5,53.9,2414,ohc,four,122,mpfi,3.31,3.54,8.7,92,4200,27,32,9988,-1
Cluster 2: 150,saab,gas,turbo,two,hatchback,fwd,front,99.1,186.6,66.5,56.1,2808,dohc,four,121,mpfi,3.54,3.07,9,160,5500,19,26,18150,3
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1 2
(135.0) (35.0) (69.0) (31.0)
normalized-losses 118.4444 122.5143 110.9565 130.5161
make toyota peugot toyota toyota
fuel-type gas gas gas gas
aspiration std std std std
num-of-doors four four four two
body-style sedan sedan sedan hatchback
drive-wheels fwd rwd fwd rwd
engine-location front front front front
wheel-base 98.2296 104.6429 96.0174 95.9129
length 172.8704 185.9829 166.3812 172.5097
width 65.6881 67.4114 64.6594 66.0323
height 53.7874 55.6457 53.6565 51.9806
curb-weight 2491.3037 2997.9714 2155.2174 2667.3226
engine-type ohc ohc ohc ohc
num-of-cylinders four four four four
engine-size 121.7778 146.2286 101.2029 139.9677
fuel-system mpfi mpfi 2bbl mpfi
bore 3.3193 3.5354 3.1625 3.4245
stroke 3.2085 3.1529 3.2312 3.2208
compression-ratio 10.1244 10.6057 10.4565 8.8419
horsepower 100.6686 120.8286 75.5217 133.8792
peak-rpm 5138.3361 5057.1429 5079.7101 5360.4958
city-mpg 25.7111 20.8571 30.3188 20.9355
highway-mpg 31.4 25.9429 36.1449 27
price 12555.5964 17797.1143 8201.511 16329.1051
symboling 0 0 1 3
Status
OK
Log


Weka Clusterer Visualize: 00:22:00 - SimpleKMeans (autos-weka.filters.unsupervised.attribute.ReplaceMissin..
X: Instance_number (Num)
Y: normalized-losses (Num)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: autos-weka.filters.unsupervised.attribute.ReplaceMissingValues_clustered
256
60.5
65
0
34.5
69
X
Y
Class colour
cluster0 cluster1 cluster2

References

https://www.openml.org/search?type=data&status=active&sort=runs&id=9

https://stackoverflow.com/questions/9821027/missing-values-in-weka

https://www.baeldung.com/cs/weka-data-mining

https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/

https://www.geeksforgeeks.org/k-means-clustering-using-weka/