

**Assignment 1 Report
Data Mining**

CSE 5334 Section 001

Student Name:

**Ahir Kunjal (1001877263)
Sheth Devanshi (1001959019)
Gope Pankaj (1001990547)**

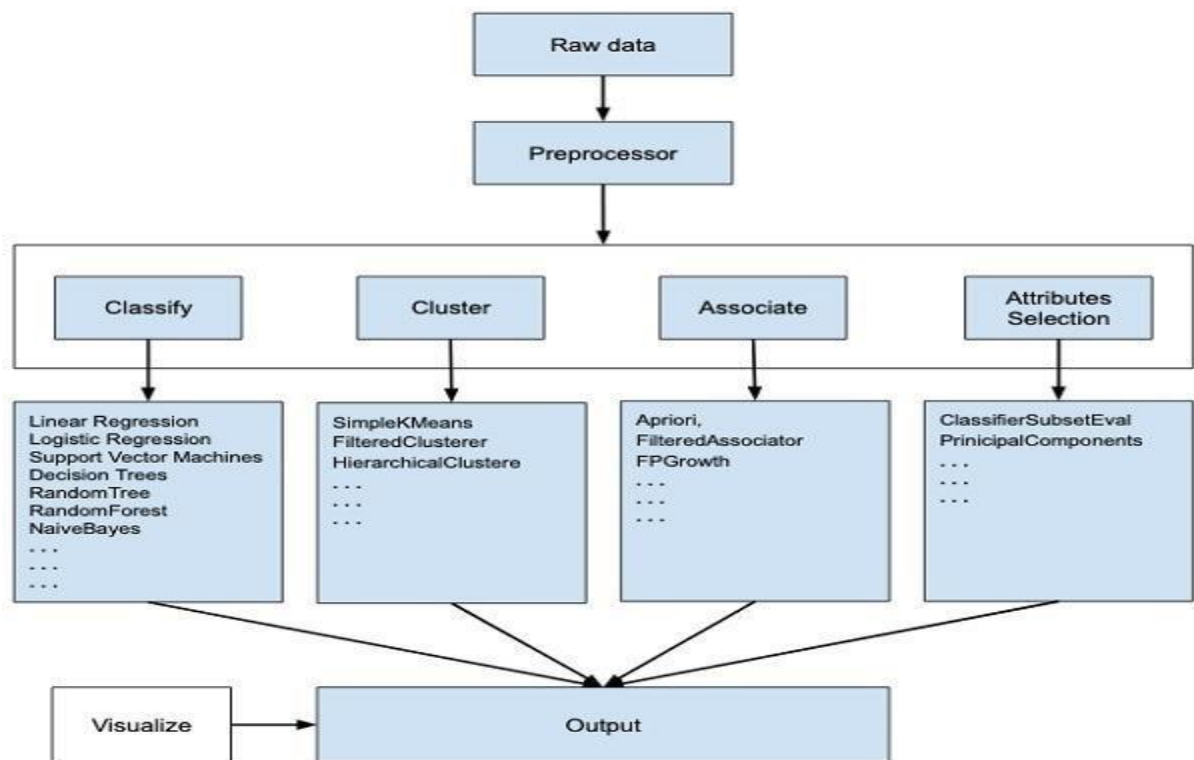
Professor: Dr Elizabeth D Diaz

Using Weka

Weka

Introduction

WEKA is an open source program that includes data preparation tools, implementation of numerous Machine Learning algorithms, and visualization tools to help you build machine learning approaches and apply them to real-world data mining situations.



Glimpse of dataset

In this assignment we are working on housing dataset.

Attributes:

- price
- lotsize
- bedrooms
- bathrms
- stories
- driveway
- recroom
- fullbase
- gashw
- airco
- garagepl
- prefarea

The following is a detailed definition of the attribute data type:

@attribute " numeric
@attribute price numeric
@attribute lotsize numeric
@attribute bedrooms numeric
@attribute bathrms numeric
@attribute stories numeric
@attribute driveway {yes,no}
@attribute recroom {no,yes}
@attribute fullbase {yes,no}
@attribute gashw {no,yes}
@attribute airco {no,yes}
@attribute garagepl numeric
@attribute prefarea {no,yes}

Viewer													
Relation: Housing													
No.	1:	2: price	3: lotsize	4: bedrooms	5: bathrms	6: stories	7: driveway	8: recroom	9: fullbase	10: gashw	11: airco	12: garagepl	13: prefarea
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Nominal	Nominal	Nominal	Nominal	Nominal	Numeric	Nominal
1	1.0	42000.0	5850.0	3.0	1.0	2.0	yes	no	yes	no	no	1.0	no
2	2.0	38500.0	4000.0	2.0	1.0	1.0	yes	no	no	no	no	0.0	no
3	3.0	49500.0	3060.0	3.0	1.0	1.0	yes	no	no	no	no	0.0	no
4	4.0	60500.0	6650.0	3.0	1.0	2.0	yes	yes	no	no	no	0.0	no
5	5.0	61000.0	6360.0	2.0	1.0	1.0	yes	no	no	no	no	0.0	no
6	6.0	66000.0	4160.0	3.0	1.0	1.0	yes	yes	yes	no	yes	0.0	no
7	7.0	66000.0	3880.0	3.0	2.0	2.0	yes	no	yes	no	no	2.0	no
8	8.0	69000.0	4160.0	3.0	1.0	3.0	yes	no	no	no	no	0.0	no
9	9.0	83800.0	4800.0	3.0	1.0	1.0	yes	yes	yes	no	no	0.0	no
10	10.0	88500.0	5500.0	3.0	2.0	4.0	yes	yes	no	no	yes	1.0	no
11	11.0	90000.0	7200.0	3.0	2.0	1.0	yes	no	yes	no	yes	3.0	no
12	12.0	30500.0	3000.0	2.0	1.0	1.0	no	no	no	no	no	0.0	no
13	13.0	27000.0	1700.0	3.0	1.0	2.0	yes	no	no	no	no	0.0	no
14	14.0	36000.0	2880.0	3.0	1.0	1.0	no	no	no	no	no	0.0	no
15	15.0	37000.0	3600.0	2.0	1.0	1.0	yes	no	no	no	no	0.0	no
16	16.0	37900.0	3185.0	2.0	1.0	1.0	yes	no	no	no	yes	0.0	no
17	17.0	40500.0	3300.0	3.0	1.0	2.0	no	no	no	no	no	1.0	no
18	18.0	40750.0	5200.0	4.0	1.0	3.0	yes	no	no	no	no	0.0	no
19	19.0	45000.0	3450.0	1.0	1.0	1.0	yes	no	no	no	no	0.0	no
20	20.0	45000.0	3986.0	2.0	2.0	1.0	no	yes	yes	no	no	1.0	no
21	21.0	48500.0	4785.0	3.0	1.0	2.0	yes	yes	yes	no	yes	1.0	no
22	22.0	65900.0	4510.0	4.0	2.0	2.0	yes	no	yes	no	no	0.0	no
23	23.0	37900.0	4000.0	3.0	1.0	2.0	yes	no	no	no	yes	0.0	no
24	24.0	38000.0	3934.0	2.0	1.0	1.0	yes	no	no	no	no	0.0	no

Add instanceUndoOKCancel

Statistical Exploratory Data Analysis

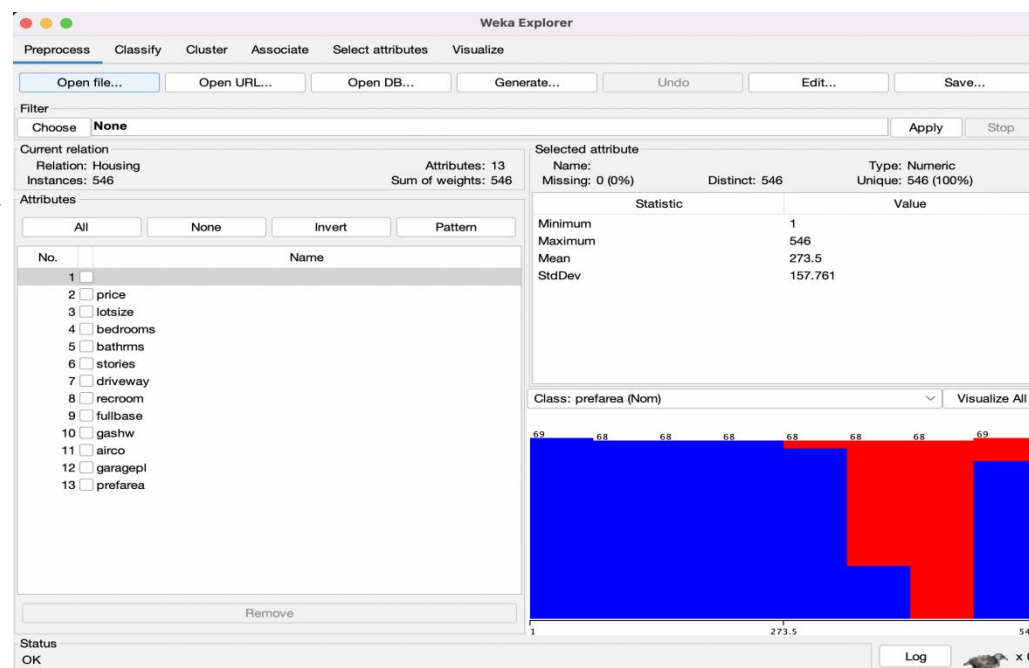
Question 1: How to load the dataset in Weka?

1. Open the Weka GUI Chooser
2. Click the “Explorer” button to open the Weka Explorer.
3. Click the “Open file...” button, navigate to the numeric/ directory and select housing.arff.
4. Click the “Open button”.

The dataset is now loaded into Weka.



Question 2: Find the number of instances and attributes are present in the given dataset.



We can see that there are total 546 instances and 13 attributes. The names of attributes are listed in the attributes section above. The first four attributes are of numeric type while the class is a nominal type with 3 distinct values.

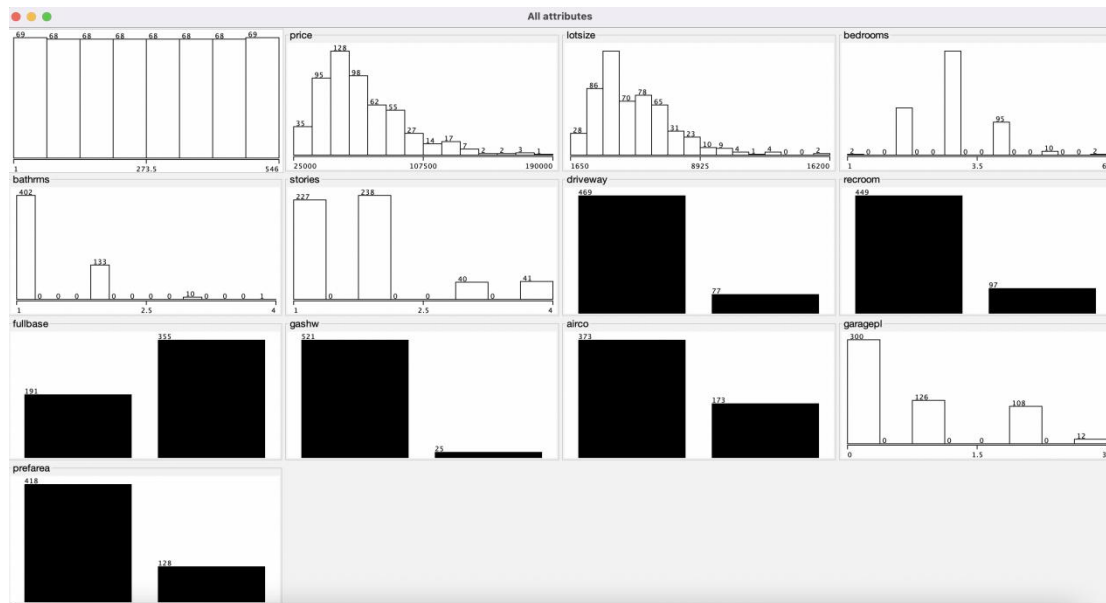
Analyse the dataset

Attribute distributions:

Question 3: How will you visualize all the attributes? Give a brief description of the data.

Click Preprocesses and then click on “Visualize All” button from the selected attributes, you’ll see the graphical distribution of each attribute.

Below is the Graphical distribution of each attribute:-

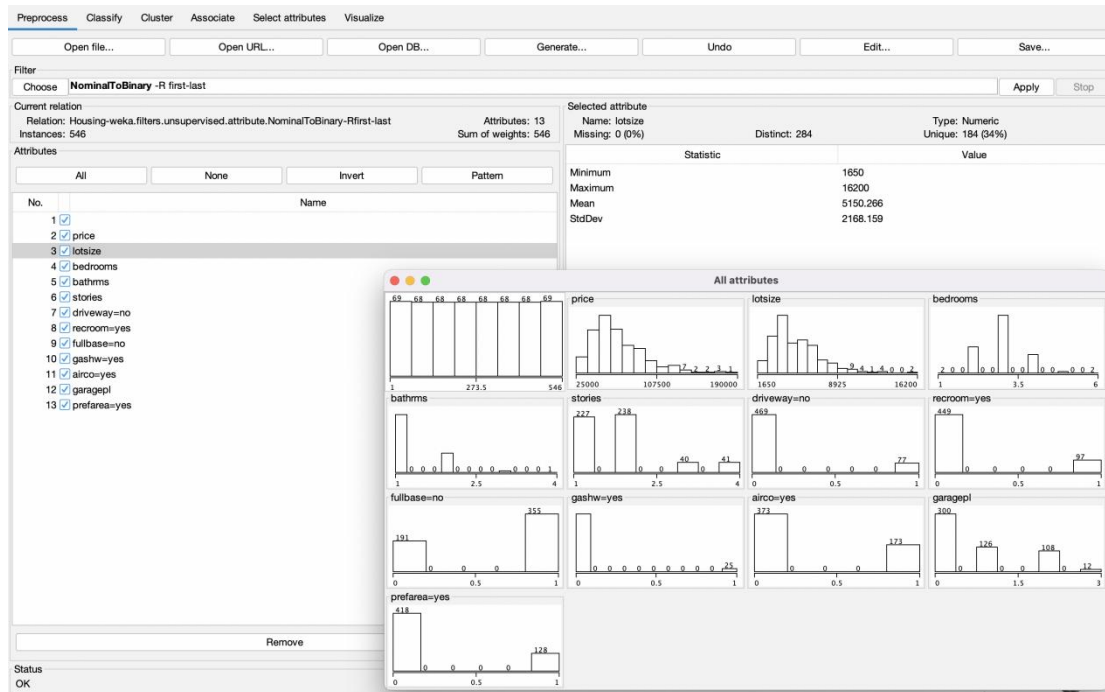


We can notice a few facts about our data:

1. There are no missing values for any of the attributes.
2. 7 inputs are numeric and 6 are binary attribute, and have values in differing ranges.
3. The price, lotsize, bedrooms, bathrms, stories and garagepl attributes are numeric.
4. The driveway recrom, fullbase, gashw, arico, prearea attributes look like a binary-like distribution (two values).

Question 4: How to convert binary data to numeric data? Give a brief description.

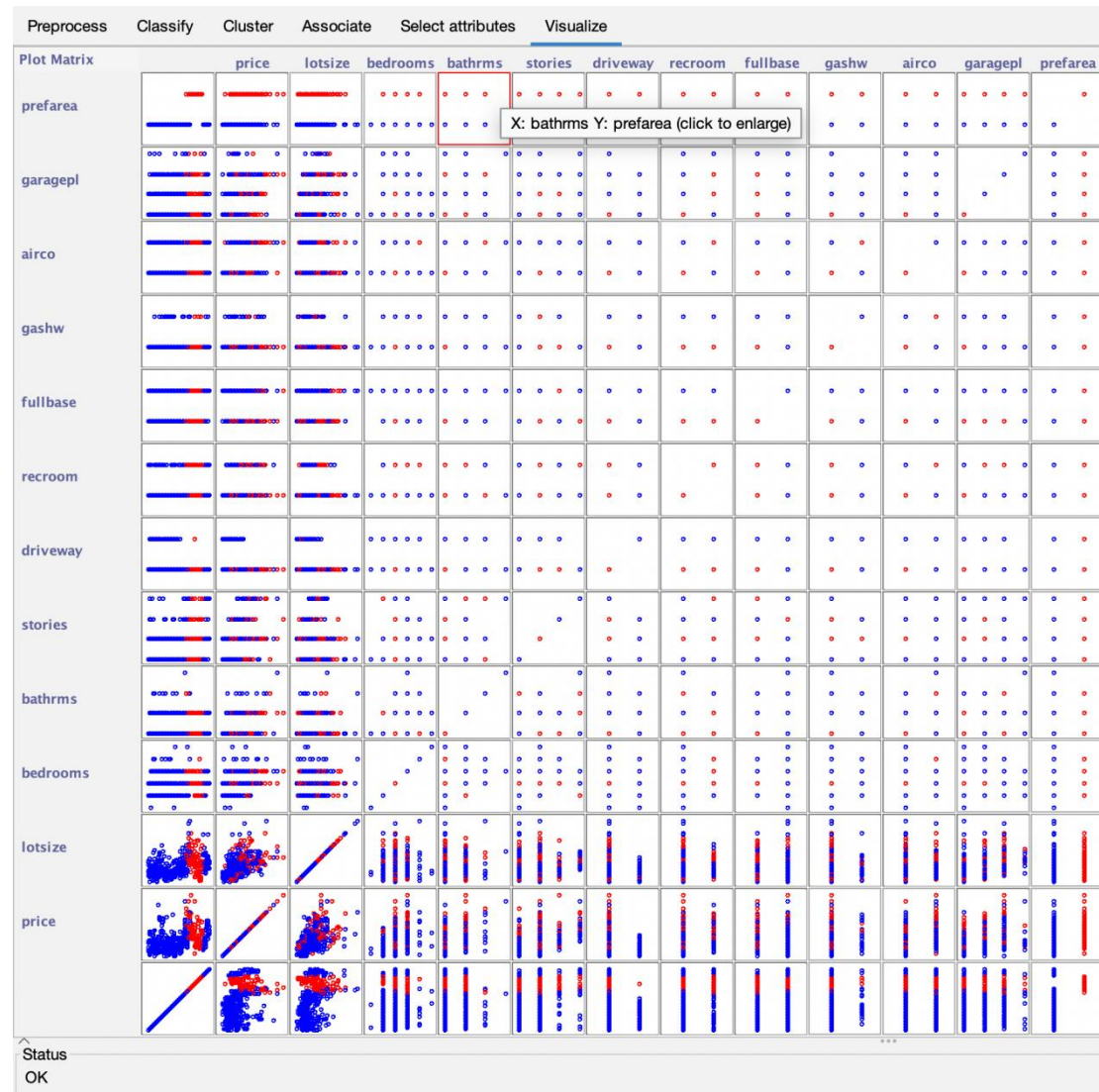
1. Click on Choose filter on Preprocess tab.
 2. Then click on unsupervised filter -> attributes -> "nominal to binary".
- This will visualize the binary data into the numeric form.



We can see that all the binary data is now represented in the form of numeric values.

Question 5: How will you visualize the interaction between attributes? Give a brief description of the data.

1. Click the "Visualize" option to see how the properties interact with one another.
 2. Reduce the "PlotSize" to 50 and modify the window size to ensure that all plots are displayed.
 3. Increase "PointSize" to 3 to make the dots more visible.
- To apply the changes, click the "Update" button.



Looking across the graphs we can see some structured relationships that may aid in modeling such as PRICE vs LOTSIZE and PRICE vs BEDROOMS.

Classifiers

Question 6: Implement top five classifiers and state the accuracy of each.

1. Logistic:

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'Logistic -R 1.0E-8 -M -1 -num-decimal-places 4'. The test options are set to 'Cross-validation' with 'Folds' set to 10. The classifier output shows the following metrics:

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Metric	Value	Percentage
Correctly Classified Instances	445	81.5018 %
Incorrectly Classified Instances	101	18.4982 %
Kappa statistic	0.4747	
Mean absolute error	0.2198	
Root mean squared error	0.3423	
Relative absolute error	61.1423 %	
Root relative squared error	80.7968 %	
Total Number of Instances	546	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.888	0.422	0.873	0.888	0.880	0.475	0.890	0.968	no
	0.578	0.112	0.612	0.578	0.594	0.475	0.890	0.631	yes

=== Confusion Matrix ===

a	b	← classified as
371	47	a = no
54	74	b = yes

You can see that with the default configuration that the Logistic algorithm achieves an accuracy of 81.50%

2. Naive Bayes

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'NaiveBayes'. The test options are set to 'Cross-validation' with 'Folds' set to 10. The classifier output shows the following metrics:

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Metric	Value	Percentage
Correctly Classified Instances	481	88.0952 %
Incorrectly Classified Instances	65	11.9048 %
Kappa statistic	0.6728	
Mean absolute error	0.1397	
Root mean squared error	0.2832	
Relative absolute error	38.8603 %	
Root relative squared error	66.8381 %	
Total Number of Instances	546	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.916	0.234	0.927	0.916	0.922	0.673	0.948	0.985	no
	0.766	0.084	0.737	0.766	0.751	0.673	0.948	0.774	yes

=== Confusion Matrix ===

a	b	← classified as
383	35	a = no
30	98	b = yes

You can see that with the default configuration that the Naive Bayes algorithm achieves an accuracy of 88.09%

3. Decision Tree

The screenshot shows the Weka Explorer interface with the REPTree classifier selected. The classifier output pane displays the following information:

Classifier output

REPTree

< 355.5 : no (235/1) [120/2]
>= 355.5
| < 486.5 : yes (90/5) [41/1]
| >= 486.5 : no (39/0) [21/0]

Size of the tree : 5

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	535	97.9853 %
Incorrectly Classified Instances	11	2.0147 %
Kappa statistic	0.9431	
Mean absolute error	0.0312	
Root mean squared error	0.1394	
Relative absolute error	8.6831 %	
Root relative squared error	32.9037 %	
Total Number of Instances	546	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.993	0.063	0.981	0.993	0.987	0.990	0.943	0.981	0.990	no
0.938	0.007	0.976	0.938	0.938	0.956	0.943	0.981	0.955	yes
Weighted Avg.	0.980	0.050	0.980	0.980	0.980	0.943	0.981	0.982	

=== Confusion Matrix ===

a	b	<- classified as	
415	3	a = no	
8	120	b = yes	

You can see that with the default configuration that the Decision Tree algorithm achieves an accuracy of 97.98%

4. k-Nearest Neighbors

The screenshot shows the Weka Explorer interface with the IBk classifier selected. The classifier output pane displays the following information:

Classifier output

garagepl
prefarea

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	470	86.0806 %
Incorrectly Classified Instances	76	13.9194 %
Kappa statistic	0.6122	
Mean absolute error	0.1407	
Root mean squared error	0.3723	
Relative absolute error	39.1227 %	
Root relative squared error	87.8865 %	
Total Number of Instances	546	

=== Detailed Accuracy By Class ===

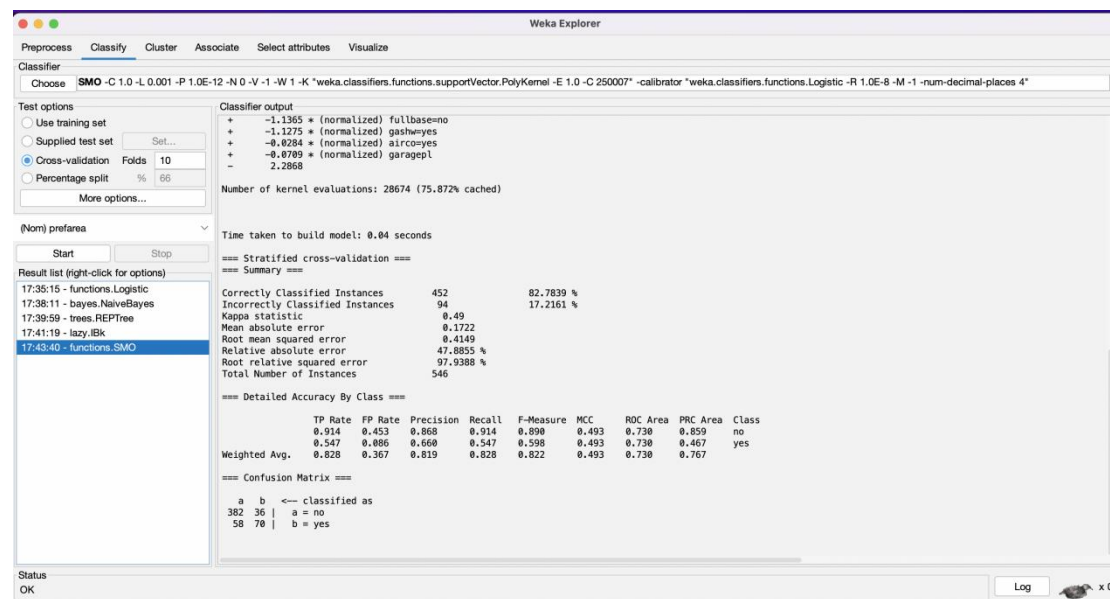
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.909	0.297	0.909	0.909	0.909	0.909	0.612	0.818	0.906	no
0.703	0.091	0.703	0.703	0.703	0.703	0.612	0.818	0.573	yes
Weighted Avg.	0.861	0.249	0.861	0.861	0.861	0.612	0.818	0.828	

=== Confusion Matrix ===

a	b	<- classified as	
380	38	a = no	
38	90	b = yes	

You can see that with the default configuration that the k-nearest neighbors algorithm achieves an accuracy of 86.08%

5. Support Vector machine



The screenshot shows the Weka Explorer interface with the SVM classifier selected. The classifier output pane displays the following results:

```
Classifier output
+ -1.1365 * (normalized) fullbase=no
+ -1.1275 * (normalized) gash=yes
+ -0.0284 * (normalized) airco=yes
+ -0.0789 * (normalized) garagepl
- 2.2868

Number of kernel evaluations: 28674 (75.072% cached)

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 452      82.7839 %
Incorrectly Classified Instances 94      17.2161 %
Kappa statistic 0.49
Mean absolute error 0.1722
Root mean squared error 0.4149
Relative absolute error 47.8855 %
Root relative squared error 97.9388 %
Total Number of Instances 546

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.914	0.453	0.868	0.914	0.890	0.493	0.730	0.859	no	
0.547	0.086	0.060	0.547	0.598	0.493	0.730	0.467	yes	
Weighted Avg.	0.628	0.367	0.819	0.828	0.622	0.493	0.730	0.767	

```

=== Confusion Matrix ===
  a  b  <-- classified as
382 36 | a = no
 58 70 | b = yes

```

You can see that with the default configuration that the SVM algorithm achieves an accuracy of 82.78%

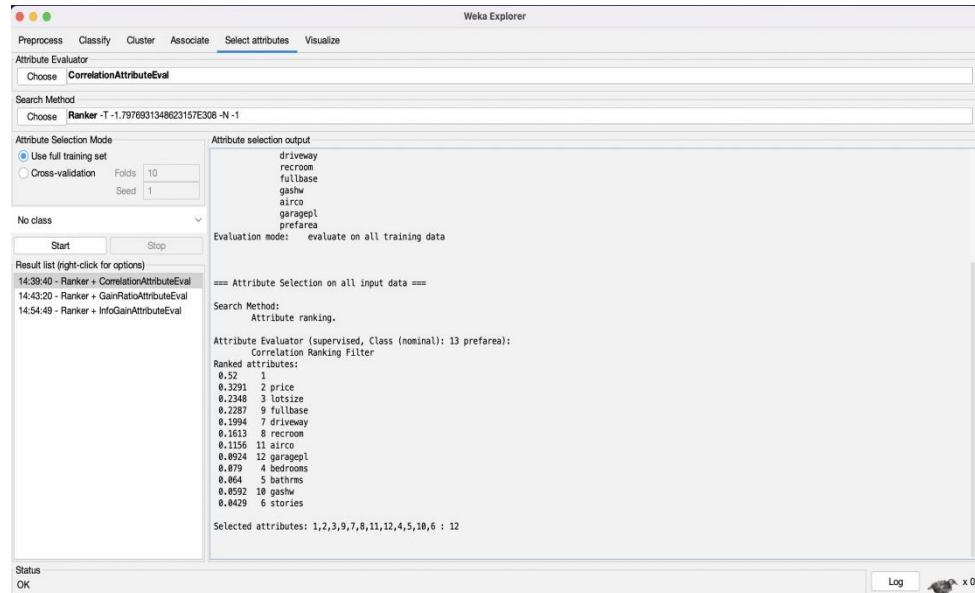
As a result, we may infer that Decision Tree methods provide the highest accuracy of the five algorithms.

Selection Attributes

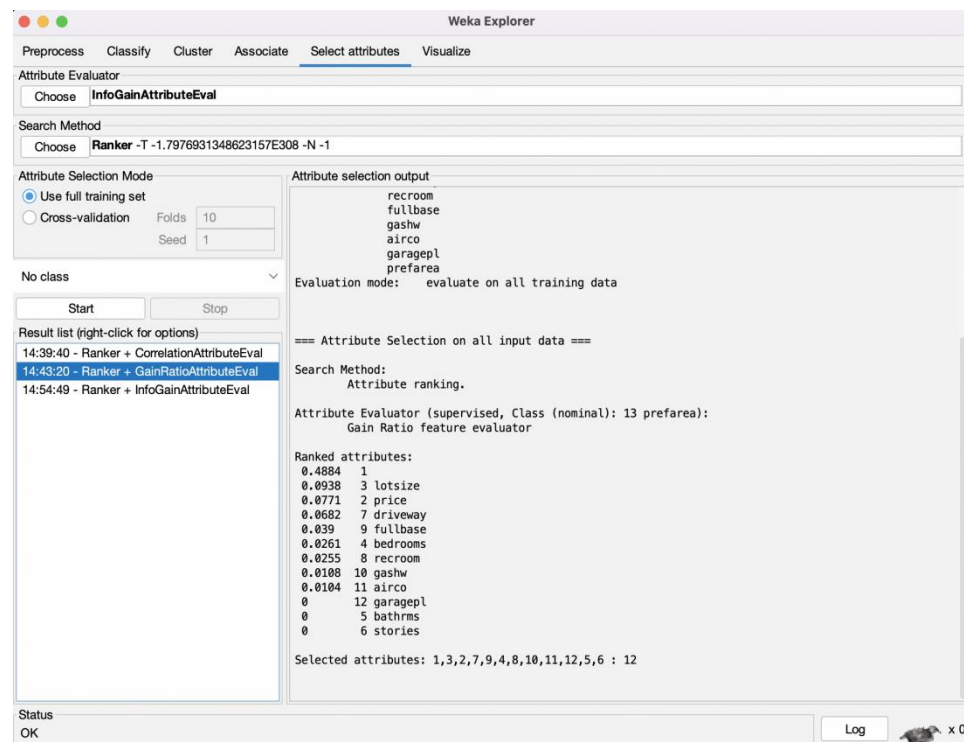
Attribute Selection (ranking)

Question 7: Evaluate any three attributes and find the difference in between them.

Correlation:



Gain Ratio:



InfoGain:

The screenshot shows the Weka Explorer interface with the 'Select attributes' tab active. The 'Attribute Evaluator' is set to 'InfoGainAttributeEval' and the 'Search Method' is 'Ranker -T -1.7976931348623157E308 -N -1'. The 'Attribute Selection Mode' is 'Use full training set' with 10 folds and seed 1. The 'Result list' shows three entries, with the last one selected: '14:54:49 - Ranker + InfoGainAttributeEval'. The 'Attribute selection output' pane displays the following information:

```
recroom
fullbase
gashw
airco
garagepl
prefarea

Evaluation mode:  evaluate on all training data

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

Search Method:
  Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 13 prefarea):
  Information Gain Ranking Filter

Ranked attributes:
0.78566  1
0.11622  2 price
0.10124  3 lotsize
0.04002  7 driveway
0.03642  9 fullbase
0.02127  4 bedrooms
0.01723  8 recroom
0.00936  11 airco
0.0029   10 gashw
0        12 garagepl
0         5 bathrms
0         6 stories

Selected attributes: 1,2,3,7,9,4,8,11,10,12,5,6 : 12
```

The sequence of ranking in all three selection characteristics differs.

Clustering

Question 8: Describe any two of the clustering methods and give the difference between them.

Simple KMeans clustering

The result window displays the centroid of each cluster, as well as data on the number and percentage of instances assigned to each cluster. A mean vector represents each cluster centroid. A cluster may be described using this cluster.

The image displays two screenshots of the Weka Explorer application, illustrating the SimpleKMeans clustering process and its results.

Top Screenshot: The 'Clusterer' tab is selected. The 'SimpleKMeans' clusterer is chosen. The 'Cluster mode' is set to 'Use training set'. The 'Clusterer output' pane shows the following information:

```
=== Run information ===
Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 1000
Relation: Housing
Instances: 546
Attributes: 13
price
lotsize
bedrooms
bathrms
stories
driveway
recroom
fullbase
gashw
airco
garagepl
prefarea

Test mode: evaluate on training data

=== Clustering model (full training set) ===

KMeans
=====
Number of iterations: 3
Within cluster sum of squared errors: 697.1443043407377

Initial starting points (random):
Cluster 0: 94,128000,8500,3,2,4,yes,no,no,no,yes,2,no
Cluster 1: 176,57500,3630,3,2,2,yes,no,no,yes,no,2,no

Missing values globally replaced with mean/mode
```

Bottom Screenshot: The 'Clusterer output' pane shows the final cluster centroids and the clustered instances.

Cluster 0: 94,128000,8500,3,2,4,yes,no,no,no,yes,2,no
Cluster 1: 176,57500,3630,3,2,2,yes,no,no,yes,no,2,no

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (546.0)	Cluster# 0 (173.0)	Cluster# 1 (373.0)
price	273.5	309.8613	256.6354
lotsize	68121.5971	85880.5896	59884.8525
bedrooms	2.9652	3.1387	2.8847
bathrms	1.2857	1.422	1.2225
stories	1.8077	2.185	1.6327
driveway	yes	yes	yes
recroom	no	no	no
fullbase	no	no	no
gashw	no	no	no
airco	no	yes	no
garagepl	0.6923	0.8902	0.6005
prefarea	no	no	no

Time taken to build model (full training data) : 0.03 seconds

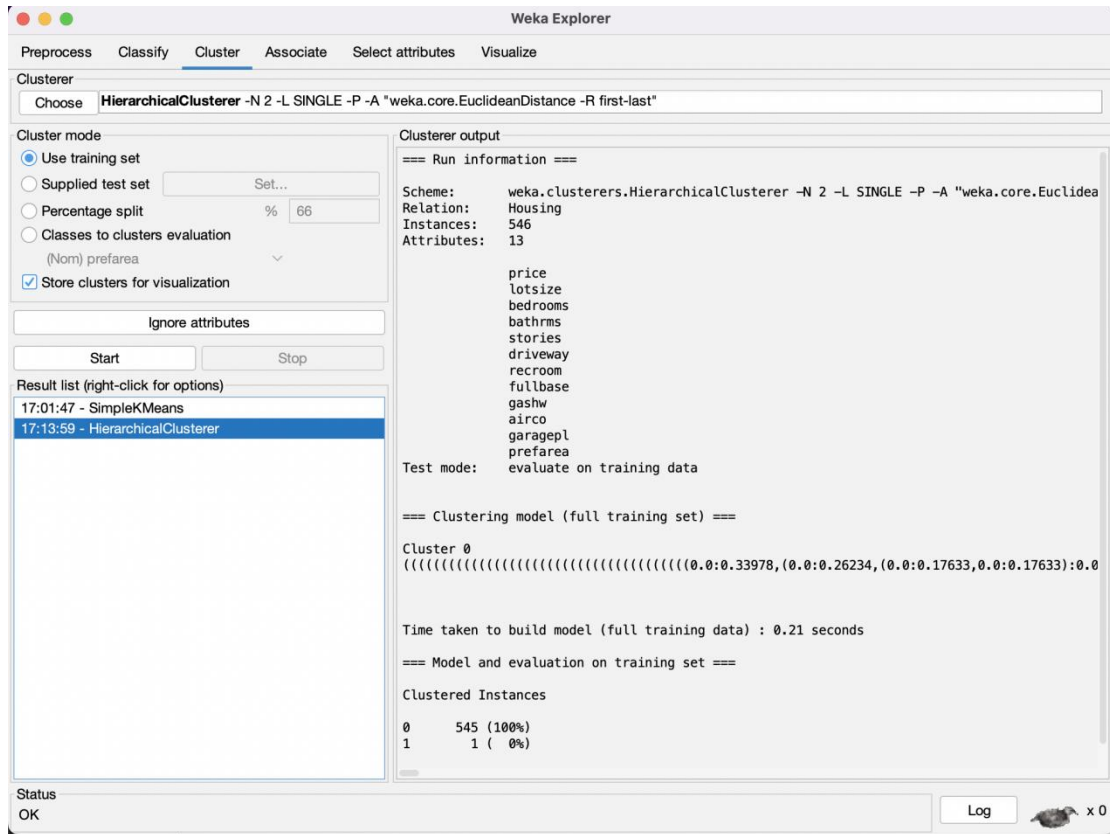
=== Model and evaluation on training set ===

Clustered Instances

Cluster	Instances	Percentage
0	173	(32%)
1	373	(68%)

Hierarchical Clustering:

The resultant window shows the centroid of each cluster as well as information on the number and proportion of instances assigned to each cluster. Each cluster centroid is represented by a mean vector. This cluster can be used to describe a cluster.



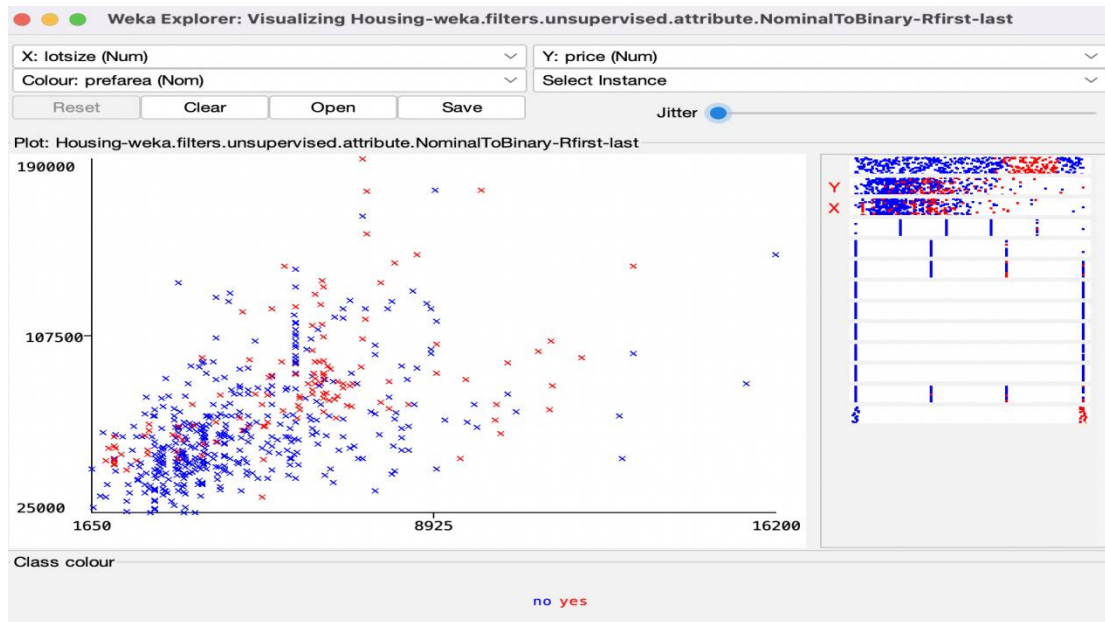
Difference:

1. The performance of K- mean algorithm is better than Hierarchical Clustering Algorithm.
2. The Time Take to build model (full training data) is faster and efficient in Kmeans clustering.
3. The Heirachical clustering is noisy than Kmeans Clustering.

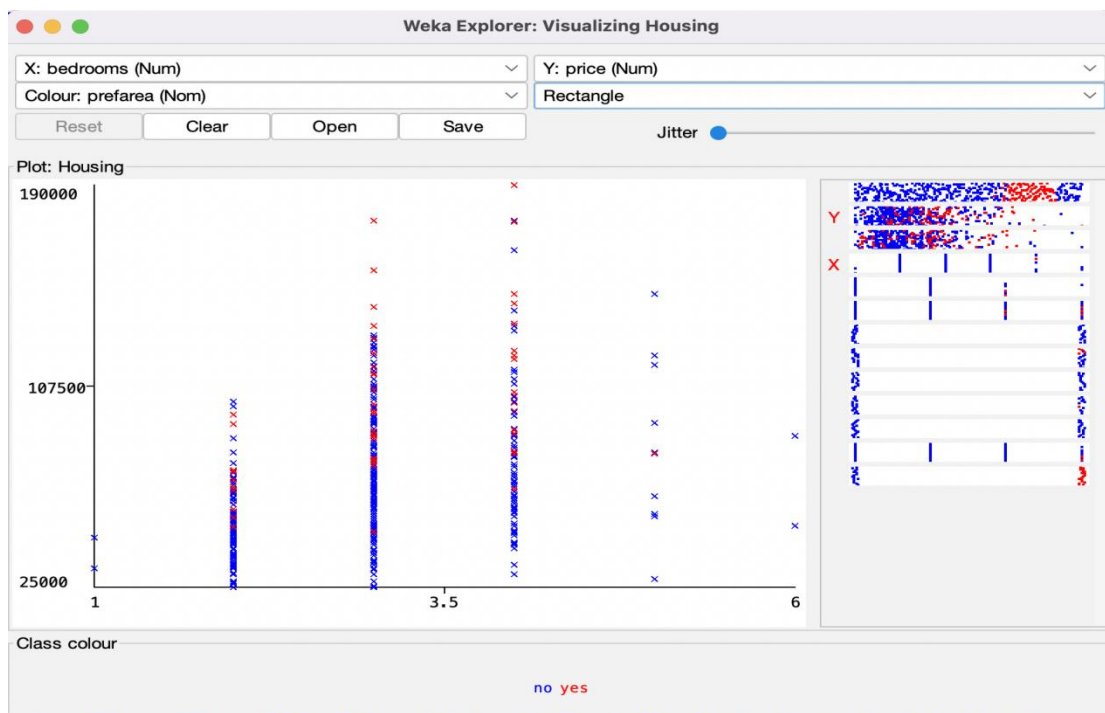
Visualization of Scatter Plot

Question 9: Visualize the scatterplot of Lotsize vs price and Bedrooms vs Price.

lotsize vs price:



Bedrooms vs Price:



Conclusion

WEKA is a strong tool for creating machine learning models. It implements some of the most popular ML methods. It also allows you to preprocess your data before applying these algorithms to it. The supporting algorithms are classed as Classify, Cluster, Associate, and Select characteristics. A beautiful and strong visual representation may be used to visualize the outcome at various stages of processing. This allows a Data Scientist to easily apply several machine learning approaches to his dataset, evaluate the results, and develop the optimal model for ultimate application.

References

<https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/>
<https://www.geeksforgeeks.org/k-means-clustering-using-weka/>
<https://cs.ccsu.edu/~markov/weka-tutorial.pdf>
https://www.tutorialspoint.com/weka/what_is_weka.htm
https://www.tutorialspoint.com/weka/weka_clustering.htm
<http://people.sabanciuniv.edu/berrin/cs512/hws/hw1/WEKA%20Explorer%20Tutorial-REFERENCE.pdf>