Predictive Analytics: Decision Tree

Peter Irungu Mwangi

SCT321-C004-2079/2018

MIT 3102 Business Intelligence

Task:

{Create a csv for square root of numbers from 1 to 350 and save it as squareroot.csv.

Using this data predict the square root of 500. (Use decision tree to predict the square root of 500)}

Introduction

Predictive analytics can be defined as a form of advanced analytics, which uses both new and historical data to forecast activity, trends and behavior. This involves application of statistical analysis techniques, analytical queries and automated machine learning algorithm to data sets, to create a predictive model(s) which places a numerical value/score on the likelihood of a given event happening.

The predictive analytics relies heavily on advanced algorithms and methodologies, this includes logistic regression models, time series analysis and decision trees.

For this task, will use decision tree to predict the square root of 500 from the dataset of square roots of numbers ranging from 1 (one) to 350 (three hundred and fifty).

Decision Tree

A decision tree refers to a graph that uses a branching method to illustrate every possible outcome of a decision. It is arguably the most popular classification technique in the data mining arena.

Decision trees include many input variables that may have impact on the classification of different patterns, this variables are known as attributes. A tree consist of branches and nodes. Branches represents the outcome of a test to classify a pattern using one of the attributes. A leaf node ate the and represents the final class choice for a pattern of branches from the root node to the leaf node (this can be represented as a complex if statement)

The basic idea behind a decision tree is, it recursively divides a training set until each division consists of an example(s) from one class.

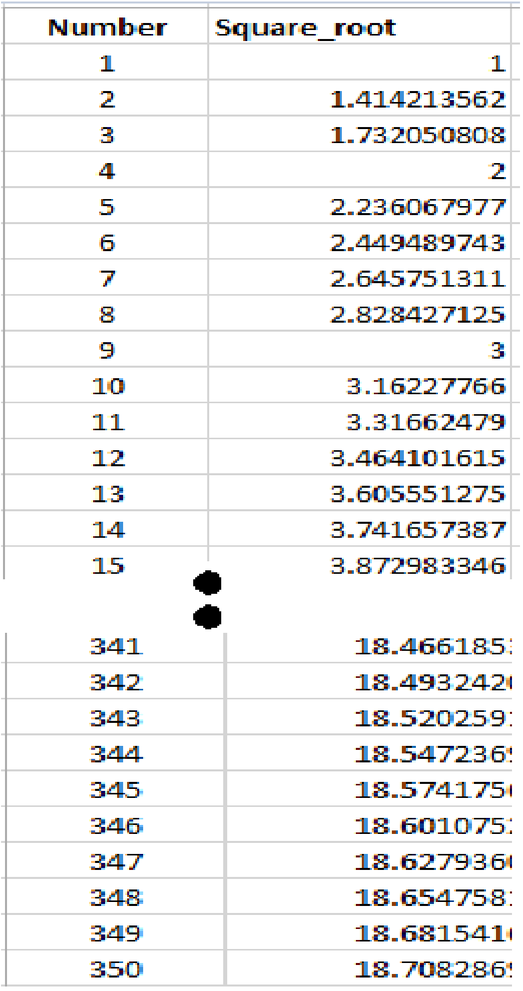
A general algorithm for building a decision tree is as follows:

1) Create a root node and assign all of the training data to it.
2) Select the *best* splitting attribute.
3) Add a branch to the root node for each value of the split. Split the data into mutually exclusive (no overlapping) subsets along the lines of the specific split and move to the branches.
4) Repeat steps 2 and 3 for each and every leaf node until the stopping criteria is reached (e.g., the node is dominated by a single class label).

Solving the task using decision tree

{Create a csv for square root of numbers from 1 to 350 and save it as squareroot.csv. Using this data predict the square root of 500. (Use decision tree to predict the square root of 500)}

Create a dataset for square roots of numbers from 1 to 350 as squareroot.csv

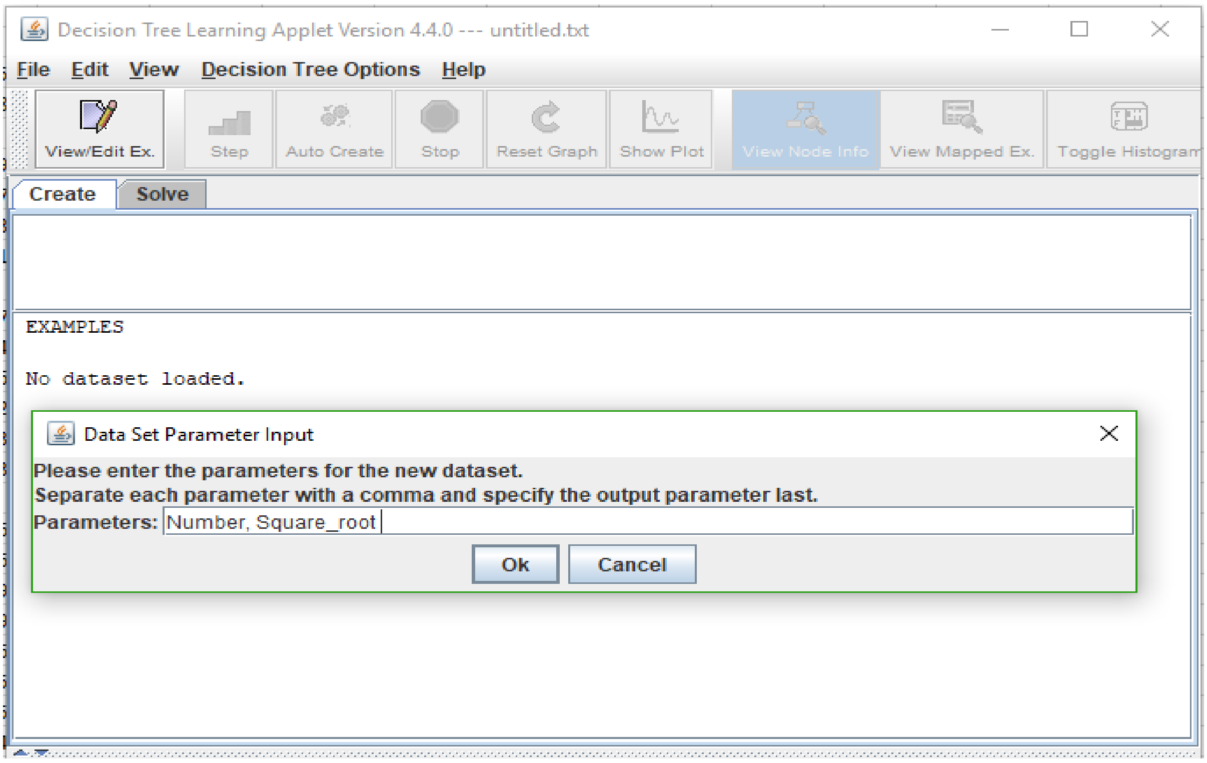


Number	Square_root
1	1
2	1.414213562
3	1.732050808
4	2
5	2.236067977
6	2.449489743
7	2.645751311
8	2.828427125
9	3
10	3.16227766
11	3.31662479
12	3.464101615
13	3.605551275
14	3.741657387
15	3.872983346

⋮

341	18.466185
342	18.493242
343	18.520259
344	18.547236
345	18.574175
346	18.601075
347	18.627936
348	18.654758
349	18.681541
350	18.708286

Create the parameters from the squareroot.csv into the applet as shown below.

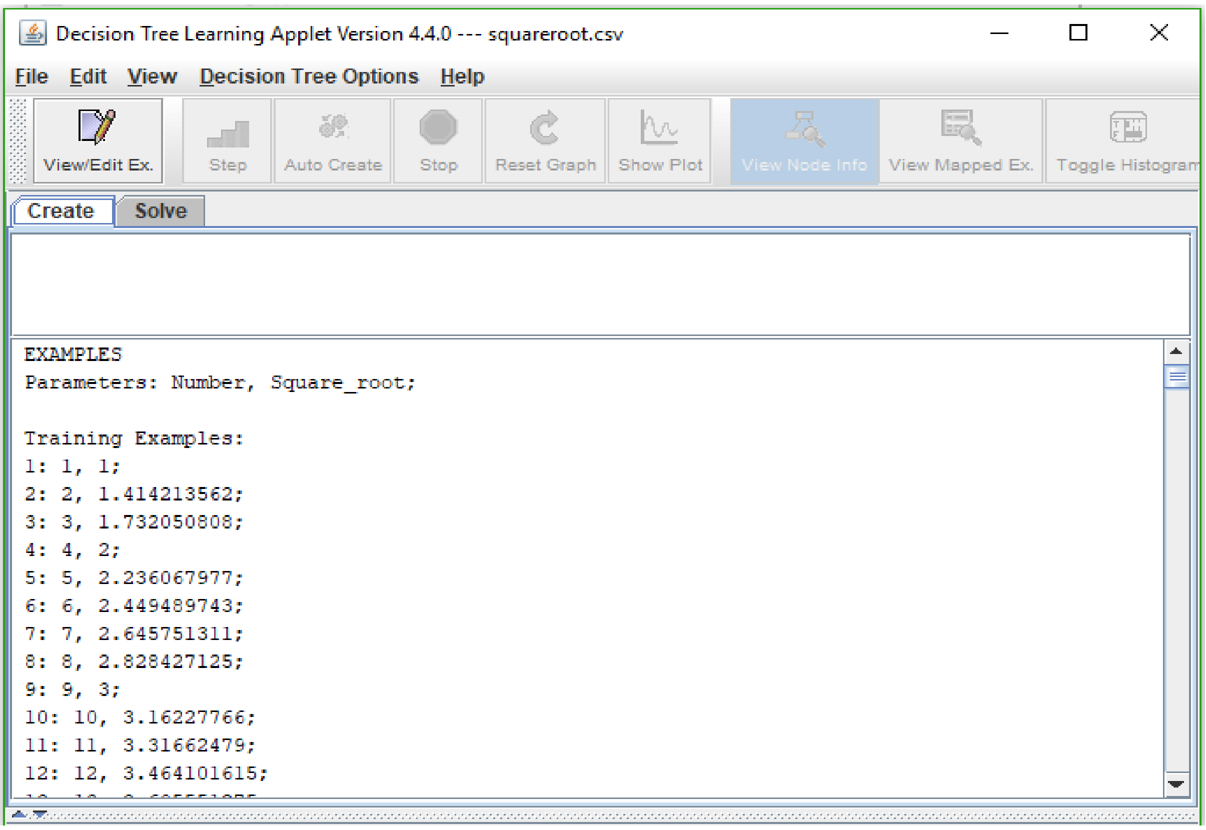


```
EXAMPLES

No dataset loaded.
```

Please enter the parameters for the new dataset.
Separate each parameter with a comma and specify the output parameter last.
Parameters: Number, Square_root

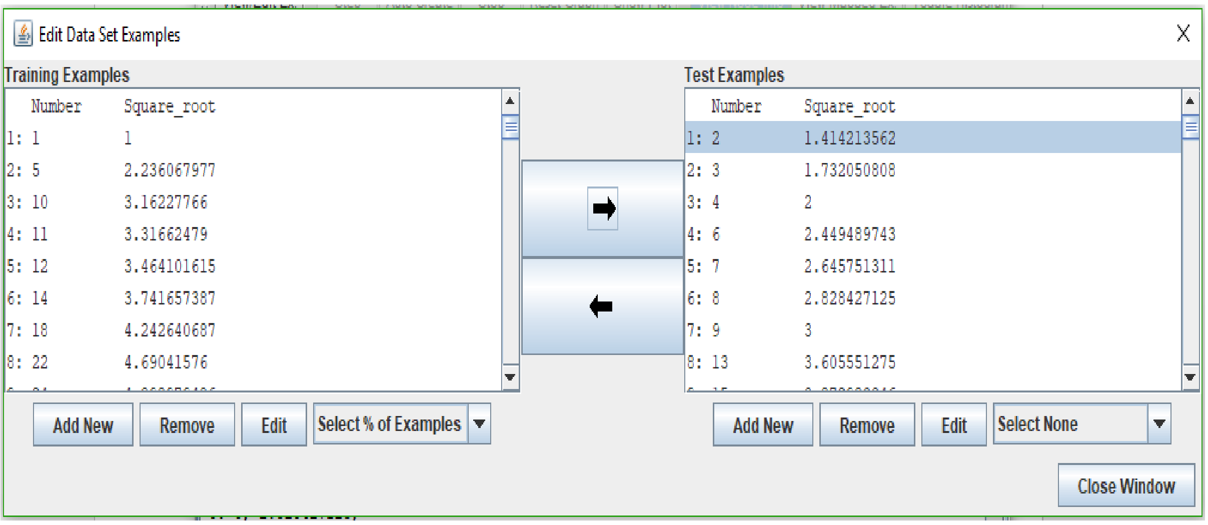
Load the squareroot.csv file to the applet to get the training examples as shown below



```
EXAMPLES
Parameters: Number, Square_root;

Training Examples:
1: 1, 1;
2: 2, 1.414213562;
3: 3, 1.732050808;
4: 4, 2;
5: 5, 2.236067977;
6: 6, 2.449489743;
7: 7, 2.645751311;
8: 8, 2.828427125;
9: 9, 3;
10: 10, 3.16227766;
11: 11, 3.31662479;
12: 12, 3.464101615;
```

From the example dataset squareroot.csv, edit the set examples to have a training examples and test examples. To achieve this, I have randomly selected 50% of the training example to create a test example dataset as shown below.



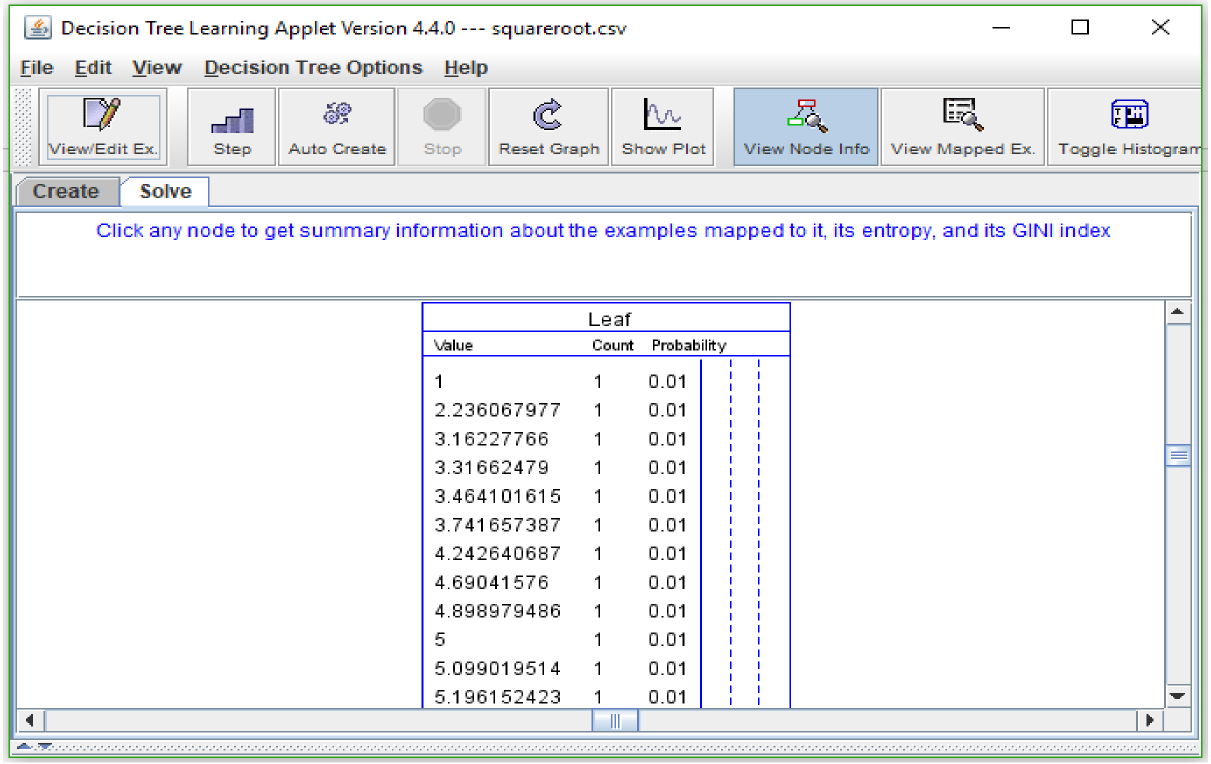
Training Examples

```
   Number      Square_root
1: 1           1
2: 5           2.236067977
3: 10          3.16227766
4: 11          3.31662479
5: 12          3.464101615
6: 14          3.741657387
7: 18          4.242640687
8: 22          4.69041576
```

Test Examples

```
   Number      Square_root
1: 2           1.414213562
2: 3           1.732050808
3: 4           2
4: 6           2.449489743
5: 7           2.645751311
6: 8           2.828427125
7: 9           3
8: 13          3.605551275
```

After separating the two datasets. Clicked on solve, and from here am able to get the summary information of the node created, as shown below.

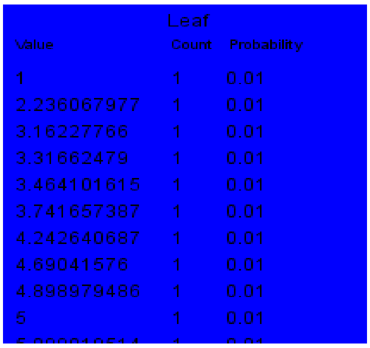


Click any node to get summary information about the examples mapped to it, its entropy, and its GINI index

Leaf

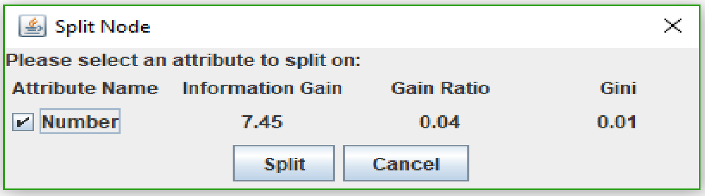
Value	Count	Probability
1	1	0.01
2.236067977	1	0.01
3.16227766	1	0.01
3.31662479	1	0.01
3.464101615	1	0.01
3.741657387	1	0.01
4.242640687	1	0.01
4.69041576	1	0.01
4.898979486	1	0.01
5	1	0.01
5.099019514	1	0.01
5.196152423	1	0.01

To get split node. Selected the node I created, then choose the number attribute to split on as shown below.



Leaf

Value	Count	Probability
1	1	0.01
2.236067977	1	0.01
3.16227766	1	0.01
3.31662479	1	0.01
3.464101615	1	0.01
3.741657387	1	0.01
4.242640687	1	0.01
4.69041576	1	0.01
4.898979486	1	0.01
5	1	0.01



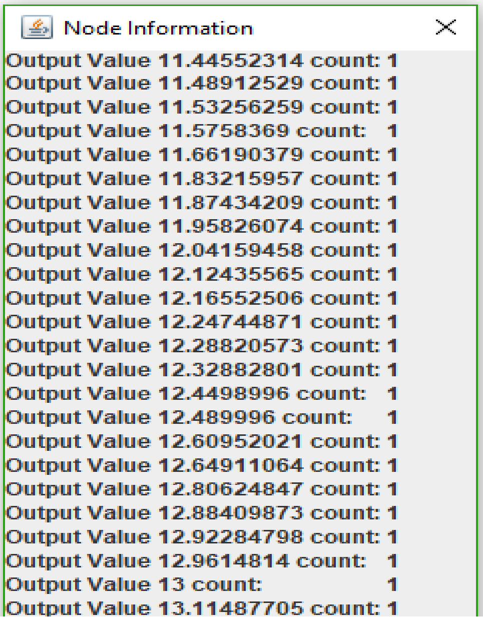
Split Node

Please select an attribute to split on:

Attribute Name	Information Gain	Gain Ratio	Gini
☑ Number	7.45	0.04	0.01

Other information genetrated from the applet

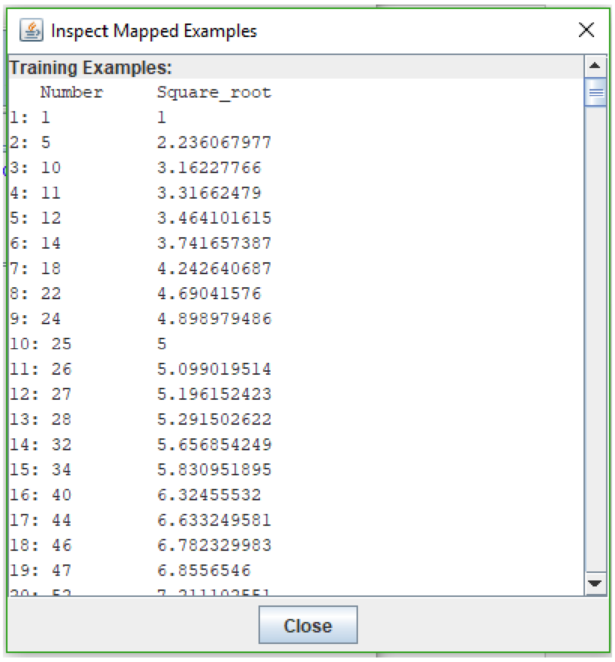
Node information



Node Information

```
Output Value 11.44552314 count: 1
Output Value 11.48912529 count: 1
Output Value 11.53256259 count: 1
Output Value 11.5758369 count:  1
Output Value 11.66190379 count: 1
Output Value 11.83215957 count: 1
Output Value 11.87434209 count: 1
Output Value 11.95826074 count: 1
Output Value 12.04159458 count: 1
Output Value 12.12435565 count: 1
Output Value 12.16552506 count: 1
Output Value 12.24744871 count: 1
Output Value 12.28820573 count: 1
Output Value 12.32882801 count: 1
Output Value 12.4498996 count:  1
Output Value 12.489996 count:   1
Output Value 12.60952021 count: 1
Output Value 12.64911064 count: 1
Output Value 12.80624847 count: 1
Output Value 12.88409873 count: 1
Output Value 12.92284798 count: 1
Output Value 12.9614814 count:  1
Output Value 13 count:          1
Output Value 13.11487705 count: 1
```

Mapped examples of datasets

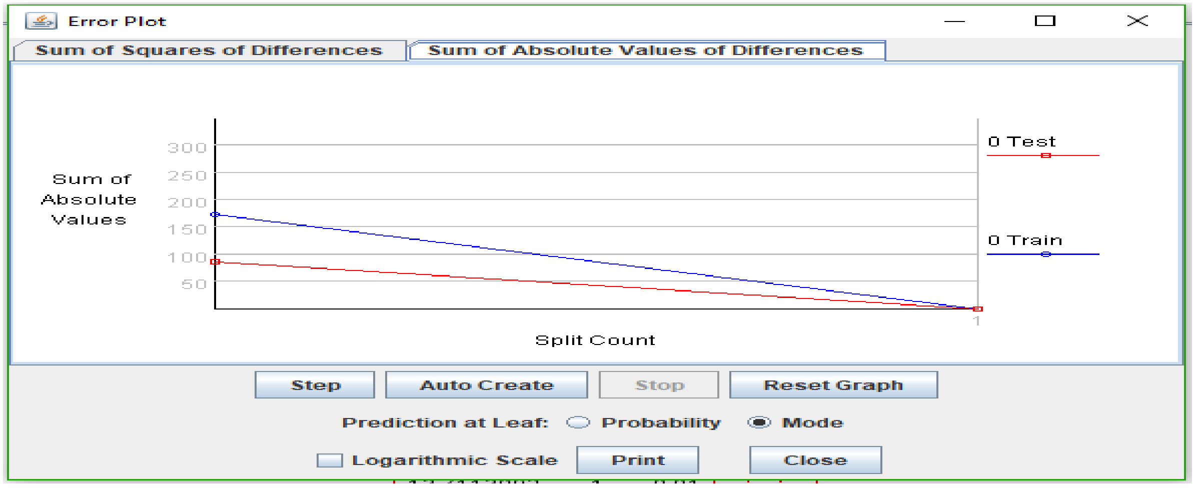


Inspect Mapped Examples

Training Examples:

```
   Number      Square_root
1: 1           1
2: 5           2.236067977
3: 10          3.16227766
4: 11          3.31662479
5: 12          3.464101615
6: 14          3.741657387
7: 18          4.242640687
8: 22          4.69041576
9: 24          4.898979486
10: 25         5
11: 26         5.099019514
12: 27         5.196152423
13: 28         5.291502622
14: 32         5.656854249
15: 34         5.830951895
16: 40         6.32455532
17: 44         6.633249581
18: 46         6.782329983
19: 47         6.8556546
```

Error plot



Prediction at Leaf: ○ Probability ● Mode

□ Logarithmic Scale