

DECISION MAKING USING ID3 ALGORITHM

Mary Slocum*

M.S. Program in Computer Science, Rivier University

Abstract

As we look at Data Mining tools, we see that there are different algorithms used for creating a decision making (or predictive analysis) system. There are algorithms for creating decision trees such as J48 along with algorithms for determining known nearest neighbor (KNN) or clustering when working on pattern recognition. The goal of this paper is to look at one particular decision tree algorithm called Iterative Dichotomiser 3 (ID3) and how it can be used with data mining for medical research. The purpose is to manipulate vast amounts of data and transform it into information that can be used to make a decision.

1 Introduction

This paper is intended to take a small sample set of data and perform predictive analysis using ID3 in order to determine the future services and diagnosis that a patient may encounter. As there is a vast amount of medical information, this study will focus on the risk of heart disease in particular.

At a high-level, the following must be performed on the data:

1. Collect the data
2. Clean the data (explore and evaluate)
3. Analyze the data

Once the data set has been determined, then the necessary algorithm can be used.

2 Collecting the User Data

To collect the data, we start by finding the necessary information on the web. In the UCI repository, (www.cs.umb.edu/~rickb/files/UCI_heart-c.arff), there is a heart disease dataset available which contains risk factors for heart disease.

In an Attribute Relation File Format (ARFF) data set, there are three sets of information: @relation, @attribute, and @data. First, the @relation which is the logical name of the data set. Next, there is a list of attributes where the name of the attribute and possible values are defined. Most importantly, there is attribute #14 where <50 means no level of heart disease and >50_1 to >50_4 represents increasing levels of disease. This attribute data is often referred to as the decision column. Lastly, there is the actual data set (@data) which contains values for each of the attributes.

Below is a subset of this data as an example:

```
@relation cleveland-14-heart-disease
@attribute 'age' real
@attribute 'sex' { female, male }
@attribute 'cp' { typ_angina, asympt, non_anginal, atyp_angina }
```

```

@attribute 'trestbps' real
@attribute 'chol' real
@attribute 'fbs' { t, f}
@attribute 'restecg' { left_vent_hyper, normal, st_t_wave_abnormality}
@attribute 'thalach' real
@attribute 'exang' { no, yes}
@attribute 'oldpeak' real
@attribute 'slope' { up, flat, down}
@attribute 'ca' real
@attribute 'thal' { fixed_defect, normal, reversable_defect}
@attribute 'num' { '<50', '>50_1', '>50_2', '>50_3', '>50_4'}
@data
63,male,typ_angina,145,233,t,left_vent_hyper,150,no,2.3,down,0,fixed_defect,'<50'
67,male,asympt,160,286,f,left_vent_hyper,108,yes,1.5,flat,3,normal,'>50_1'
67,male,asympt,120,229,f,left_vent_hyper,129,yes,2.6,flat,2,reversable_defect,'>50_1'
37,male,non_anginal,130,250,f,normal,187,no,3.5,down,0,normal,'<50'
41,female,atyp_angina,130,204,f,left_vent_hyper,172,no,1.4,up,0,normal,'<50'

```

3 Clean the Data

First, there are a few initial things to note about the data in an ARFF file:

1. Some entries have “?” in them which indicates missing data. We need to decide whether to discard a record if it has missing data, or to replace with the attribute mean.
2. There are 14 attributes, so we need to determine if we want to use all of these or just a subset.
3. As there are many records (303), we need to decide if we want to use a sampling, and we need to determine how large a sampling to use. One rule of thumb is to use 2/3 of the data to train and then 1/3 of the data to test.

Given the complexity of the original data, it was decided to trim it down to a subset with the following for a proof of concept:

1. Only 14 rows of data
2. Only 4 attribute columns plus the decision of column
3. No continuous data. So, rather than data having individual values (such as age with 63, 67, 37, 41, etc.) we would have two or three categories so it would be discrete data.

For the purposes of this exercise, a subset of the data for heart disease of just 5 attribute columns (Age, Trestbps (resting blood pressure), Chol (cholesterol), Gender, and decision of whether they have Heart disease) and 14 rows of data, and explicitly putting “Yes” or “No” for the decision columns (rather than having 5 possible decisions) will be used and the information stored in a Comma-Separated Value (CSV) file. Also, for simplicity, the Age column data is changed to not contain contiguous data by using < 50, <60, and <70 as shown in Figure 1.

```

@relation HeartDisease
@attribute 'Age' { <50, <60, <70 }
@attribute 'Trestbps' { <120, <140, <160 }
@attribute 'Chol' { <200, >200 }
@attribute 'Gender' { male, female }
@attribute 'Heart' { No, Yes }
@data
<50,<120,<200,male,No
<50,<120,<200,female,No
<70,<120,<200,male,Yes
<60,<140,<200,male,Yes
<60,<160,>200,male,Yes
<60,<160,>200,female,No
<70,<160,>200,female,Yes
<50,<140,<200,male,No
<50,<160,>200,male,Yes
<60,<140,>200,male,Yes
<50,<140,>200,female,Yes
<70,<140,<200,female,Yes
<70,<120,>200,male,Yes
<60,<140,<200,female,No

```

Figure 1. Modified data set for heart disease.

4 Iterative Dichotomiser 3 (ID3) Decision Tree Algorithm

For the decision tree algorithm, ID3 was selected as it creates simple and efficient tree with the smallest depth. [1] Unlike a binary tree (such as used for Huffman Encoding where there is just a left and right node), the ID3 decision tree can have multiple children and siblings.

The ID3 decision makes use of two concepts when creating a tree from top-down:

1. Entropy
2. Information Gain (as referred to as just gain)

Using these two algorithms, the nodes to be created and the attributes to split on can be determined. The best way to explain these concepts and how they apply to creating a decision tree using ID3 is to show an example.

4.1 Entropy

Entropy is the measurement of uncertainty where the higher the entropy, then the higher the uncertainty. With a decision tree, this measurement is used to determine how informative a node is. [2] Wikipedia best summaries Entropy for the ID3 algorithm with [3]:

$$E(S) = - \sum_{j=1}^n f_S(j) \log_2 f_S(j)$$

Where :

- $E(S)$ is the **information entropy** of the set S ;
- n is the number of different values of the attribute in S (entropy is computed for one chosen attribute)
- $f_S(j)$ is the frequency (proportion) of the value j in the set S
- \log_2 is the **binary logarithm**

An entropy of 0 identifies a perfectly classified set.

4.2 Informational Gain

Information Gain (also known as just Gain) uses the entropy in order to determine what attribute is best used to create a split with. By calculating Gain, we are determining the improved entropy by using that attribute. So, the column with the higher Gain will be used as the node of the decision tree.

Also, Wikipedia has the summary for the equation for Gain [4]:

$$G(S, A) = E(S) - \sum_{i=1}^m f_S(A_i) E(S_{A_i})$$

Where :

- $G(S, A)$ is the gain of the set S after a split over the A attribute
- $E(S)$ is the **information entropy** of the set S
- m is the number of different values of the attribute A in S
- $f_S(A_i)$ is the frequency (proportion) of the items possessing A_i as value for A in S
- A_i is i^{th} possible value of A
- S_{A_i} is a subset of S containing all items where the value of A is A_i

4.3 Sample

The concepts of the ID3 algorithm are best described using an actual example with data. First, we take the entire data set as shown in Figure 2.

Age	Trestbps	Chol	Gender	Heart
<50	<120	<200	male	No
<50	<120	<200	female	No
<70	<120	<200	male	Yes
<60	<140	<200	male	Yes
<60	<160	>200	male	Yes
<60	<160	>200	female	No
<70	<160	>200	female	Yes
<50	<140	<200	male	No
<50	<160	>200	male	Yes
<60	<140	>200	male	Yes
<50	<140	>200	female	Yes
<70	<140	<200	female	Yes
<70	<120	>200	male	Yes
<60	<140	<200	female	No

Figure 2. Entire data set for heart disease risk.

And, we determine the number of “No” and number of “Yes” for the decision column in order to calculate the entropy using the table shown in Figure 3.

Age	Trestbps	Chol	Gender	Heart
<50	<120	<200	male	No
<50	<120	<200	female	No
<60	<160	>200	female	No
<50	<140	<200	male	No
<60	<140	<200	female	No
Total No				5
<70	<120	<200	male	Yes
<60	<140	<200	male	Yes
<60	<160	>200	male	Yes
<70	<160	>200	female	Yes
<50	<160	>200	male	Yes
<60	<140	>200	male	Yes
<50	<140	>200	female	Yes
<70	<140	<200	female	Yes
<70	<120	>200	male	Yes
Total Yes				9

Figure 3. Table dividing up the “Yes” and “No” results.

The following formula can be used to calculate the total entropy, E:

$$E = ((-\text{NumNo}/\text{NumTotal})\log_2(\text{NumNo}/\text{NumTotal})) + ((-\text{NumYes}/\text{NumTotal})\log_2(\text{NumYes}/\text{NumTotal}))$$

So, in this example, the total entropy would be:

$$E = ((-5/14)\log_2(5/14)) + ((-9/14)\log_2(9/14)) = 0.94$$

Next, we take each column and calculate the Gain. Starting with Gender, we look at each of the columns with Male and Female and calculate entropy of each “Yes” and “No” for Gender/Female (6/14) and Gender/Male (8/14) and subtract from the total entropy that we calculated previously. Figure 4 shows just the Gender and Heart Disease data used for this calculation.

$$\text{Gain} = \text{TotalEntropy} - (6/14 \times (\text{EntropyFemale})) - (8/14 \times (\text{EntropyMale})) = 0.048$$

Gender	Heart
male	No
male	Yes
male	Yes
male	Yes
male	No
male	Yes
male	Yes
male	Yes
male	Yes
Total	8

Gender	Heart
female	No
female	No
female	Yes
female	Yes
female	Yes
female	No
Total	6

Figure 4. Tables of Male and Female.

After calculating the Gain for each column, we have determined that Age has the highest gain as it is the largest value, so this becomes the next attribute node. In this initial case, Age will be the root node with each of its possible values (<50, <60, and <70) becoming children as represented in Figure 5.

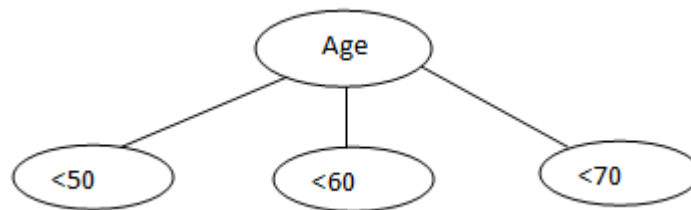


Figure 5. Initial nodes for decision tree.

Now, the next set of nodes or children need to be determined using a recursive process. Each of the subsets of age is looked at. For < 50, the subset of data is shown in Figure 6.

Age	Trestbps	Chol	Gender	Heart
<50	<120	<200	male	No
<50	<120	<200	female	No
<50	<140	<200	male	No
<50	<160	>200	male	Yes
<50	<140	>200	female	Yes

Figure 6. Table for just Age < 50

This process would continue recursively with all subsets until all nodes have the appropriate attribute and all leafs contain a decision. [5]

Finally, after the decision tree is completed, it would be traversed breadth-first for determining a decision.

5 Implementation

One possible implementation would be using C++ to create a system as a proof of concept to exercise both Entropy and Gain that would do the following:

1. Initially all of the User Interface (UI) will be disabled except for the Browse button. So, you must select the Browse button and navigate to the Heart data file to load the data set. Then, you should select “Clean File” in order for the basic information on the data set such as Total number of rows, Total number of columns/attributes, number of rows missing data, and percentage of missing data to be displayed as shown in Figure 7.

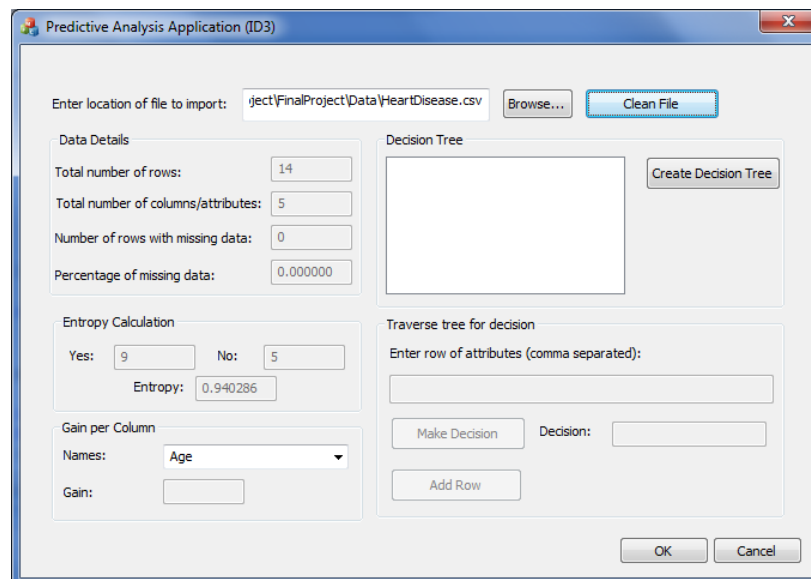


Figure 7. Screenshot after importing data and doing “Clean File”.

2. After “Clean File”, the total entropy is display for the collection, and you can select each column in the “Gain per Column” groupbox to see each individual value as represented in Figure 8.

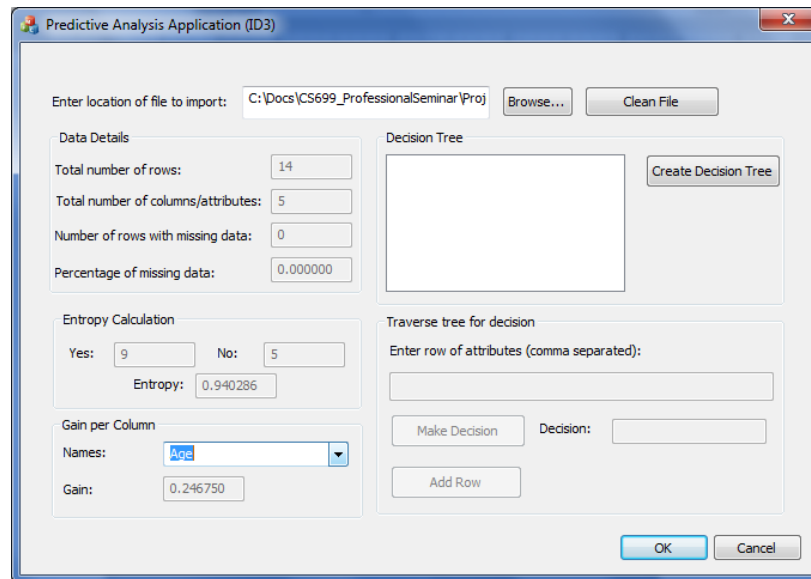


Figure 8. Screenshot showing value of Gain for the Age column.

3. You should see that Age has the largest value for the Gain when looking at each of the columns.
4. Now, you can select “Create Decision Tree” to create a CTreeView visual representation which will need to be expanded manual (as only the first level is expanded automatically) as shown in Figure 9.

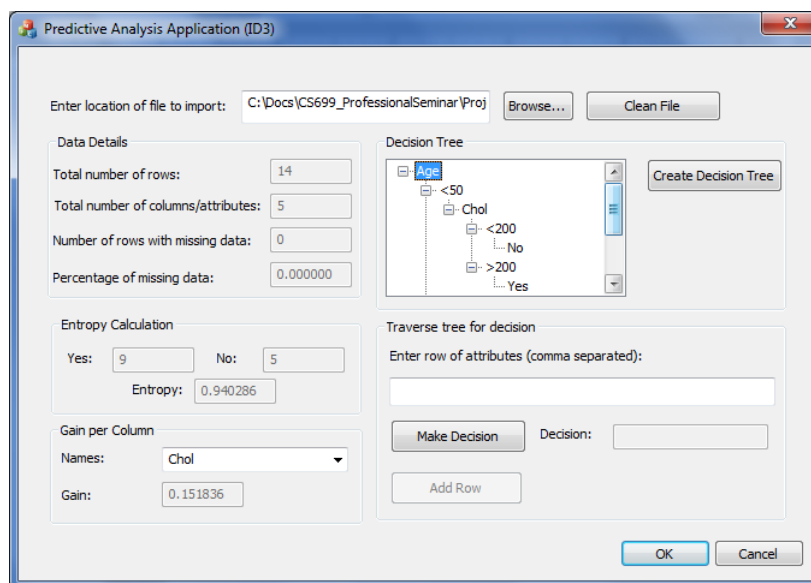


Figure 9. Screenshot after using “Create Decision Tree”.

5. Now, we need to make a decision – so, in the “Traverse tree for decision”, we can enter a row of data for the breadth-first process to occur as displayed in Figure 10.

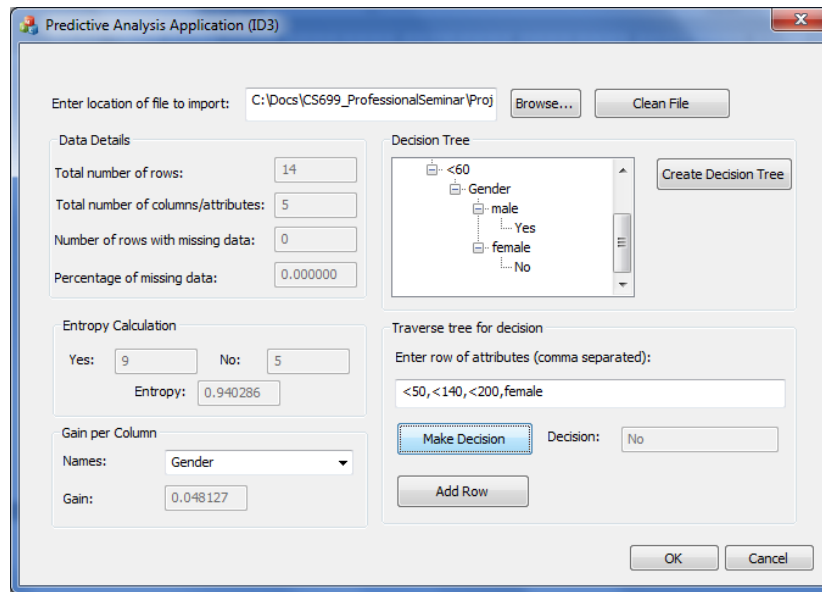


Figure 10. Screenshot of entering attributes and doing a “Make Decision”.

6. As the data was valid and we were able to make a decision, we can now select “Add Row” to add it to the data set.
7. After adding it to the data set, the process is started over again by selecting “Clean File” where the “Data Details” are updated accordingly as shown in Figure 11.

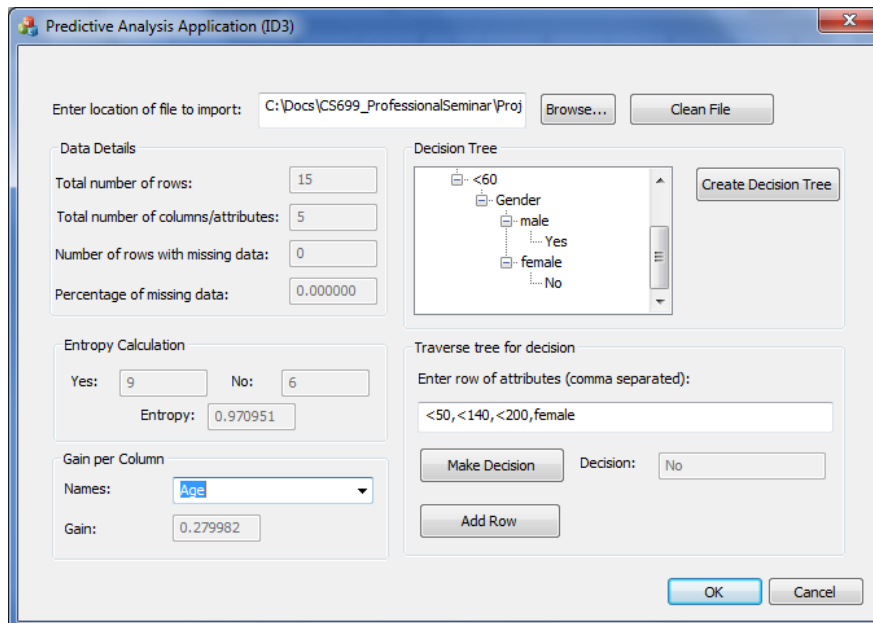


Figure 11. Screenshot after doing an “Add Row”.

8. Now, when we select the different column names under “Gain per Column”, we should see that these values have also changed. For example, Age has changed from 0.246 to now 0.279 with the addition of this particular row.

6 Validating Results

Waikato Environment for Knowledge Analysis (Weka) is a data mining tool that enables us to validate our results with the same data set that we just did by hand and implemented using C++. Using Weka, we create the following J48 decision tree shown textually below and visually in Figure 12.

```
J48 pruned tree
-----

Age = <50
| Chol = <200: No (3.0)
| Chol = >200: Yes (2.0)
Age = <60
| Gender = male: Yes (3.0)
| Gender = female: No (2.0)
Age = <70: Yes (4.0)

Number of Leaves :    5

Size of the tree :    8
```

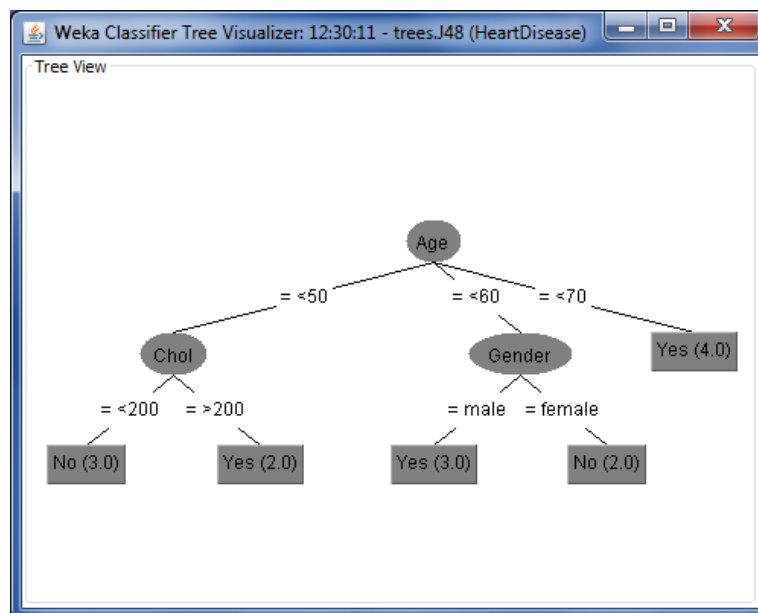


Figure 12. J48 decision tree generated using Weka.

Given that the data is no longer continuous (and smaller set), using both J48 and ID3 provide the same results when comparing Figure 12 to the decision tree produced using our C++ program in Figures 9 – 11, and also seen in Figure 13 which shows the first level of the decision tree.

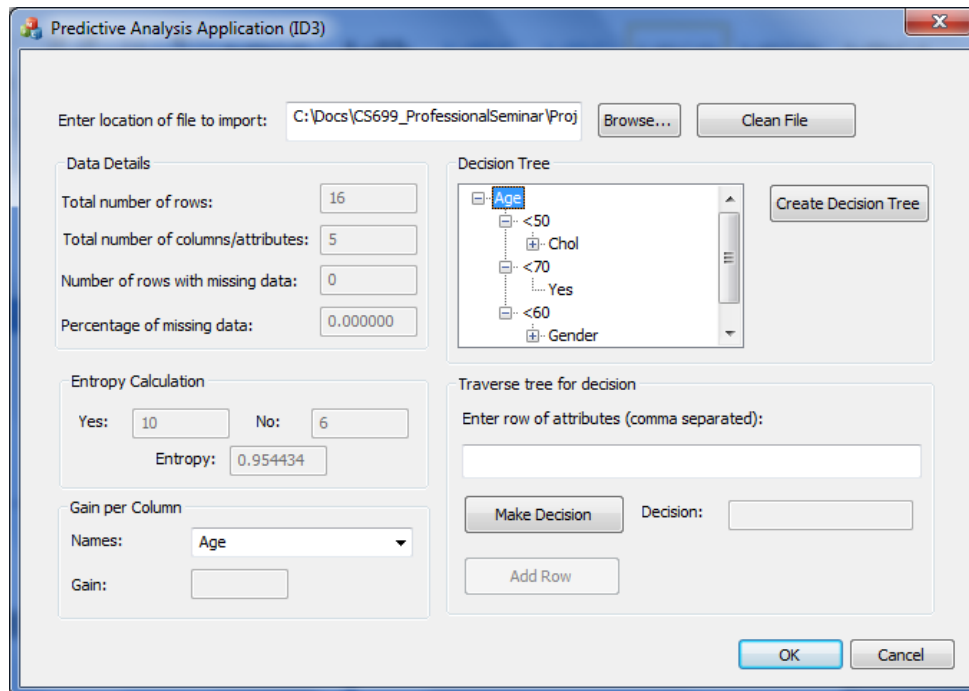


Figure 13. Initial Create Decision Tree showing root node and first set of nodes.

7 Conclusion

In this study, we reviewed the initial data mining task of collecting and cleaning the data prior to the use of the data with the algorithm. For ID3, the two key concepts are Entropy (measurement of uncertainty) and Information Gain (measurement of purity). Using these parameters, we created a top-down tree that we can later traverse breadth-first to make a decision given a new data set.

Though the ID3 algorithm worked well with the non-contiguous data that we created and has the advantage that it generates a smaller depth decision tree, we would want to evaluate this algorithm with a larger and more complicated data set. Also, we might want to consider evaluating different types of decision trees along with clustering algorithms to determine if there is a better approach for the medical industry specifically for determination of the risk of heart disease. In fact, since ID3 was first developed, there is now an improved version called C4.5.

Lastly, there are vast amounts of data available and there are many ways it can be manipulated. So, using these algorithms is an iterative process where processes are always being improved (such as when new attributes are added for considerations – for example, there may other factors for risk of heart disease such as weight or family history). For the medical industry, these decisions can determine if a patient is a high risk for heart disease along with making conclusions as to what insurance coverage a company should give a person based on this risk.

References

- [1] Wei Peng, Juhua Chen, and Haiping Zhou. An Implementation of ID3 -- Decision Tree Learning Algorithm. Retrieved March 10, 2012, from <http://web.arch.usyd.edu.au/~wpeng/DecisionTree2.pdf>.

- [2] Building Classification Models: ID3 and C4.5. Retrieved April 2, 2012, from <http://www.cis.temple.edu/~giorgio/cis587/readings/id3-c45.html#3>.
- [3] ID3 algorithm. Page last modified January 13, 2012. Retrieved March 31, 2012, from http://en.wikipedia.org/wiki/ID3_algorithm.
- [4] ID3 algorithm. Page last modified January 13, 2012. Retrieved March 31, 2012, from http://en.wikipedia.org/wiki/ID3_algorithm.
- [5] CSE5230 Tutorial: The ID3 Decision Tree Algorithm. Monash University, Semester 2, 2004. Retrieved April 10, 2012, from <http://www.csse.monash.edu.au/courseware/cse5230/2004/assets/decisiontreesTute.pdf>.

* **MARY SLOCUM** is a Computer Scientist working on her Masters at Rivier University. She resides in Nashua, NH with her husband and two daughters. She is originally from New York where she worked in NYC before relocating to New England. She enjoys running half marathons and has a black belt in karate.