Classification using J48 Decision Trees

1. Introduction

Whenever the outcome is a categorical variable, a lot of machine learning techniques exhibit limitations. It would suffice to say that the options that we have as a data scientist go down significantly when categorical prediction needs to be done. Now, I do understand that a lot of traditional methods can be tweaked in order to accommodate a categorical outcome but its effectiveness would be questionable. I have decided to tackle a classification problem as a part of my individual project in this course. I have used the "adult" dataset with more than 30k instances for training data and 16k instances for test data. I decided to go with decision trees as my approach to classify these instances. The input to my decision tree algorithm were the first 14 variables of my dataset. The last variable was the class variable. I got an overall accuracy of 87.14% on the training set and an accuracy of 83.55% for the test set. The data had a lot of missing values to start with and I had decided to remove these instances altogether both from the training set and the test set. The most determining attribute of our dataset turned out to be "capitalgain" followed by "maritalstatus". The ensuing content of this report explains every aspect of the pre-processing, tools used, analysis and results in detail.

2. Dataset

Source: http://archive.ics.uci.edu/ml/datasets/Adult

No. of attributes: 15

The data had 14 attributes which can be included as predictors to determine the class value. The outcome variable has 2 possible outcomes. The attributes included age, sex, maritalstatus, education, capitalloss, capitalgain etc. The extraction of this data was done by Barry Becker from the 1994 Census database. The prediction task is to determine whether a person makes over 50K a year or not. More detailed info about the dataset can be found at the aforementioned link.

2.1 Preprocessing

It originally had 32,561 instances for training data which was reduced to 30,162 after missing value treatment. The test data was also subjected to a similar missing value treatment and that gave us 15,060 instances from a data set that had 16,281 instances initially. Unnecessary blank spaces were also removed. These blank spaces created issues while feeding the. arff file into Weka.

The reduction of data in both of these datasets was somewhat proportional, so this wouldn't be a huge cause for concern. I decided not to remove a single attribute from the data set as it was quite evident that I shouldn't from the decrease in accuracy while constructing decision trees. Additionally, decision trees have their own inherent attribute selection measure like the "gain ratio" which does the job of selecting relevant attributes.

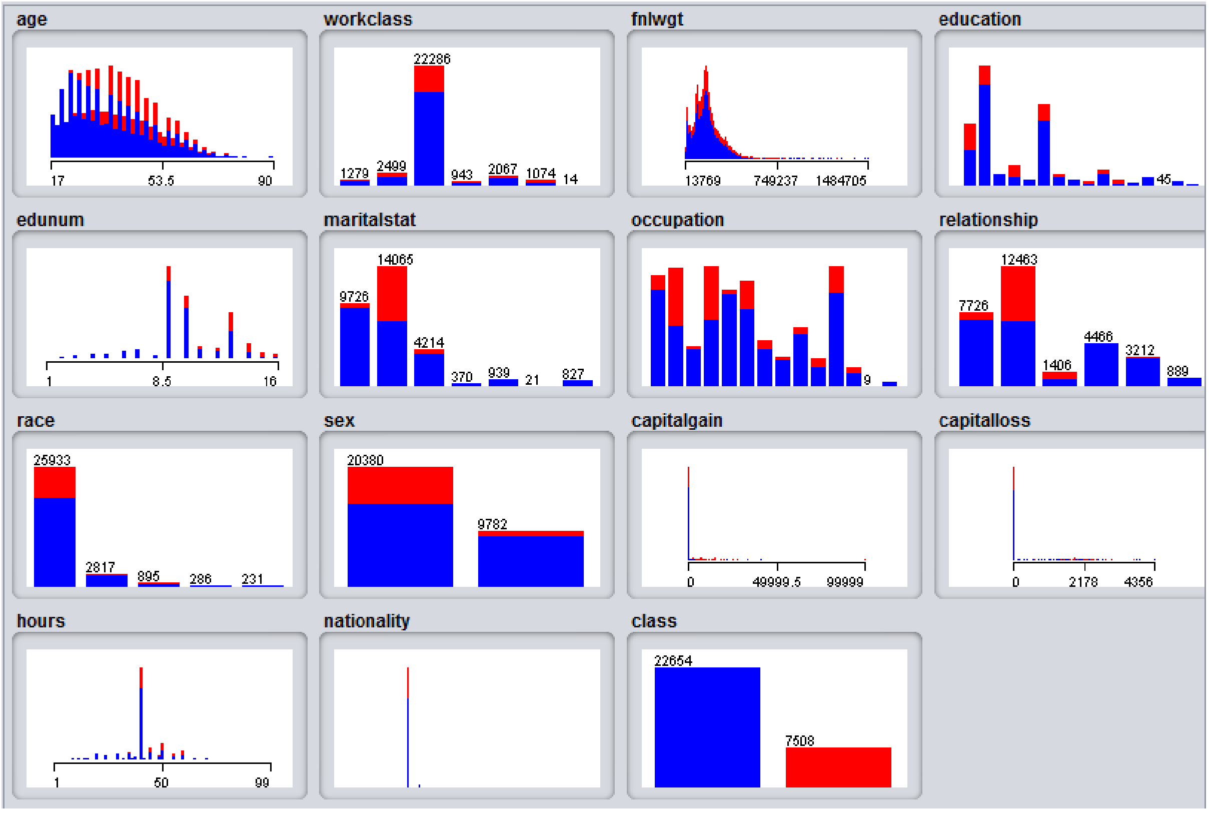
2.2 Converting to ARFF

I used Weka 3.8 as my primary tool in order to build decision trees. Now, for use in Weka, we need to convert the data into .arff format (Attribute Relation File Format). An ARFF file is an ASCII text file that describes a list of instances sharing a set of attributes. Technically, ARFF has 2 different sections, the “header” section and the “data” section.

Since, the .csv needed to be converted to .arff, I added the header to each column in excel and wrote code in Java using the Weka API to do the conversion. The code used to do this conversion is named “*csvtoarff.java*” and can be found in the zip file that I have submitted.

3. Exploratory Data Analysis

Due to the fact that we don’t need to eliminate any variables before feeding data to the decision tree (since, trees have an attribute selection measure built-in), I thought that showing graphical information about each and every attribute that is involved in this whole process would be a nice way to explain all 15 attributes without taking up much space. I used Weka 3.8 to generate these graphs.



The last variable is the “class” I am trying to predict and all the other variables are the predictors. The prediction is whether or not the income of the individual is more than $50k. There were 8 nominal variables and 6 numeric variables that were used to predict the “class”. Age, fnlwgt, capitalgain, capitalloss, edunum and hours were the numeric variables and the rest were nominal.

4. Brief Overview of WEKA 3.8

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Weka is an open source software issued under the GNU General Public License. It can also be applied to big data.

5. Model Selection and Validation

This section describes which algorithm I decided to use for my data from the vast choice of available algorithms.

5.1 J48 Decision Tree

The J48 decision tree is an open source java implementation of the traditional C4.5 algorithms in the Weka data mining tool. It uses gain ratio as an attribute selection measure in contrast to information gain which was the attribute section measure for ID3 decision tree algorithm.

5.2 Training

I have trained the classifier with "*adult_train_15.arff*" using the default configuration for the decision tree i.e. confidence factor 0.25 and a batch size of 100. This yielded an accuracy of 87.14%. Now, Weka has a tendency to randomize the training set that is fed into it and so the accuracy yielded by the GUI slightly differs. This can be easily avoided by generating an ID attribute using the "addfilter" method in Weka and removing the attribute before training. Although, the resulting change in accuracy is very trivial and shows its effects only after the 2nd decimal place.

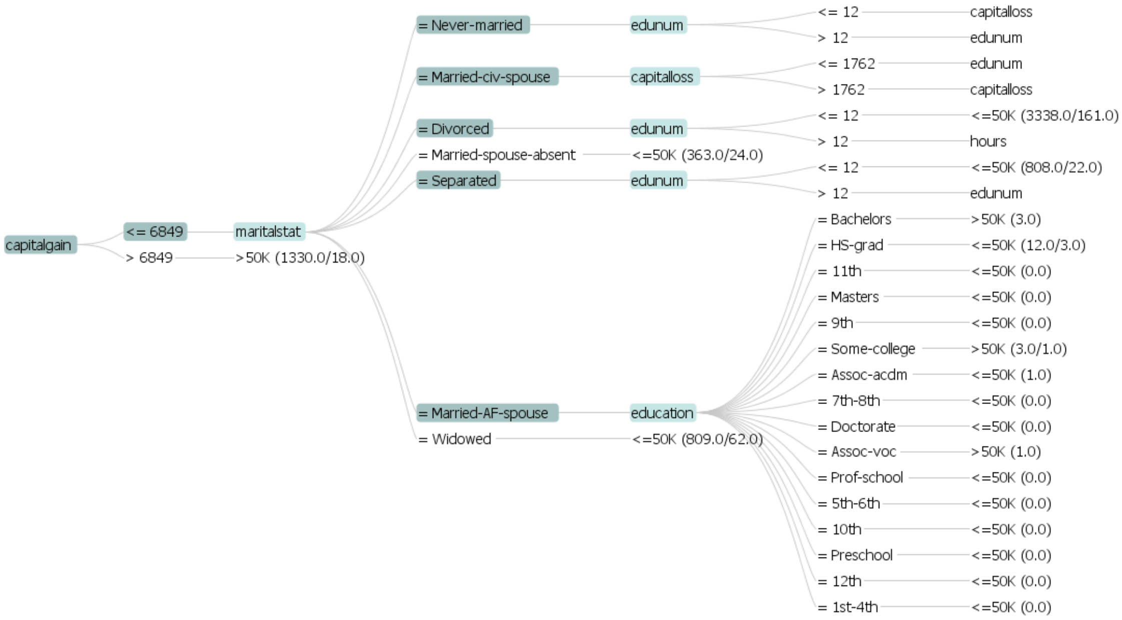
All the code pertaining to training and testing can be found in files "*WekaDecisionTree.java***"** and "*Functions.java"***.**

5.3 Performance

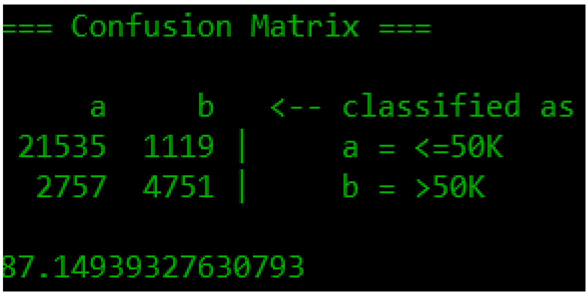
Before attempting to evaluate the model by providing it with a test set. I performed a training using 10 fold cross-validation in order to get an idea of how much the model was overfitting. Cross-validation has proved to be an effective measure in decreasing variance in a model and giving us a more modest value for accuracy; after all, all theories in science are all about predictive performance. This helps us in judging how well the model will perform on unseen data. The results of 10 fold cross-validation turned out to be 86.14%, which was exactly one percent less than what we could obtain with the whole data as a training set at once.

After getting a modest judgement of how well the trained model will perform on unseen data, I proceeded to test my model on completely fresh data. This dataset is "adult_test_15.arff". It has the same number of attributes as the training dataset which was "adult_train_15.arff". The test accuracy obtained is 83.55% which was quite close to the training accuracy and what we I had expected. *The 23rd line in my code (WekaDecisionTree.java) has instructions on how to switch between training and testing accuracy.*

The decision tree that I obtained using prefuse tree (a package available in Weka) for the training data is as follows:



The confusion matrix that I obtained for the training set and test set are as follows:

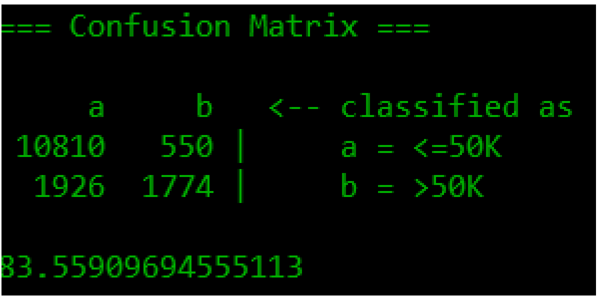


```
=== Confusion Matrix ===

     a     b   <-- classified as
 21535  1119 |     a = <=50K
  2757  4751 |     b = >50K

87.14939327630793
```

(a) Training Confusion Matrix



```
=== Confusion Matrix ===

     a     b   <-- classified as
 10810   550 |     a = <=50K
  1926  1774 |     b = >50K

83.55909694555113
```

(b) Test Confusion Matrix

6. Conclusion

From this whole process, what I can say for sure is that “capitalgain” is one of the most important factors in determining whether a person is going to earn more than $50k. Apparently, income from investments (capital gain) other than regular salary decides the most whether we earn more than 50k or not. To be more specific, we could say that people with a capital gain of more than $6849 were predicted to have an annual income of more than 50k whereas for the other branch it depended on tons of different factors.

The second most important attribute in determining the income category turned out to be “marital status”. Additionally, I was able to achieve the accuracy that is listed in the “data description” section of the data set for C4.5 algorithms at the UCI machine learning repository.

7. References

http://www.cs.waikato.ac.nz/ml/weka/

http://archive.ics.uci.edu/ml/

< >