Knowledge & the web: exercise session 6

Katrien Laenen Gust Verbruggen Ward Schodts

November 22, 2015

In this report we discuss which algorithms we tried to find a good classification model that predicts if a passenger of the Titanic will survive or not.

We use the data from `titanic_train.csv` to train our model and perform cross-validation for performance assessment. The models used are a **rule set** and a **decision tree**. The rest of this document is structured as follows. First we describe the preprocessing steps used in order to prepare the data for better classification. Next, we describe the learning process for both models. Finally, we compare results for both learning techniques, based on the cross-validation results.

1 Preprocessing

In order to achieve better classification, we preprocess the data. This is done in two phases: feature selection to only preserve relevant features and feature engineering in the form of a discretisation step.

Feature selection Features that we do not think are relevant are `Name`, `Ticket`, `PassengerId`, `Embarked` and `Cabin`. They are removed.

Discretisation The two numerical values are discretised in accordance to some criterium. The `Age` feature into five age classes: child (< 9 years), teen (10 to 15 years), young (15 to 24 years), adult (25 to 59 years) and elderly (> 59 years). To discretise `Fare`, we take a look at a histogram plot, as seen in Figure 1. There are a lot more low fares. The three leftmost bars get their own bin. The chosen discretisation is: 0-10, 10-20, 20-30, 30-100 and 100+.

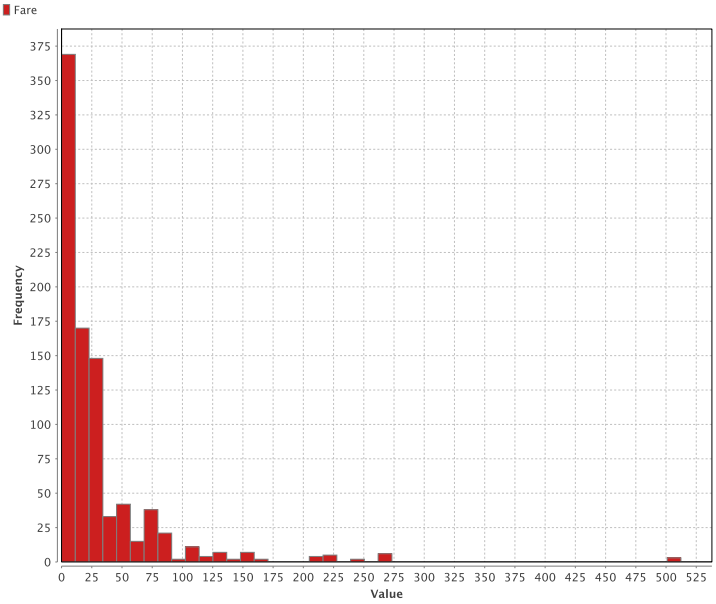

Figure 1: Histogram for the `Fare` feature

2 Rule learning

One thing we tried was to learn classification rules from the training data. Rule learning algorithms come with different options for several aspects of the algorithms. In the exercise session we experimented with two of these.

First, for choosing the conditions of a classification rule, there are two options: one is to add to the rule the condition which increases the accuracy the most, another is to add to the rule the condition which increases the information gain the most. The accuracy criterion option resulted in the following rule set:

- if Parch > 5.500 then 0 (1 / 0)
- if Fare ≥ 10 and Sex = male then 0 (170 / 25)
- if Sex = female then 1 (63 / 197)
- else 0 (171 / 61)

The information gain criterion option resulted in this rule set:

- if Sex = male and Fare ≥ 10 then 0 (170 / 25)
- if Sex = female then 1 (64 / 197)

- if Fare = 10-20 and Parch $\leq$ 0.500 then 0 (66 / 7)
- else 0 (111 / 54)

A second aspect of rule learning is the desired pureness, which is the minimum ratio of the major class in a covered subset in order to consider the subset pure. The standard value for this is 0.9. We experimented with this value set to 0.8 and 1. For the desired pureness set to 0.8 and the accuracy criterion we got the following rule set:

- if Parch $>$ 5.500 then 0 (1 / 0)
- if Fare $\geq$ 10 then 0 (189 / 47)
- if Sex = female then 1 (44 / 175)
- else 0 (171 / 61)

This rule set is very similar to the rule set obtained with the accuracy criterion option and desired pureness of 0.9. The only difference is that in the second rule the *Sex = male* condition now disappeared from the condition part. For the desired pureness set to 0.8 and the information gain criterion we got the following rule set:

- if Sex = male then 0 (360 / 93)
- else 1 (57 / 177)

When comparing this rule set with the rule set obtained for the information gain criterion with the desired pureness set to 0.9, we see that this rule set is a great simplification of the other. For the desired pureness set to 1 and the accuracy criterion we got exactly the same rule set as when setting the desired pureness to 0.9 for that criterion. The same happened for the desired pureness set to 1 and the information gain criterion, which also resulted in the same rule set as when setting the pureness to 0.9 and keeping the information criterion.

Next to the criterion and desired pureness, there are three other parameters for rule learning: sample ratio (which specifies the sample ratio of training data used for growing and pruning), minimal prune benefit (which specifies the minimum amount of benefit which must be exceeded over unpruned benefit in order to be pruned) and use local random seed (which indicates if a local random seed should be used for randomization). We did not experiment with these parameters, but kept the standard values for these, which were 0.9, 0.25 and no respectively.

From these results we can clearly see that Sex, Fare and Parch are features which are important for this task. That Sex and Fare were important features was to be expected. When the Titanic collided with an iceberg, women and children and people of the higher ranks were evacuated first. This is reflected in the data as we see that women and people which payed a higher fare are more likely to have survived. That the number of parents and children is also an important feature can be explained as follows. People in need want to stay close to their loved ones at all costs. But how larger a group, how more difficult it was to get on a lifeboat together. Also, little children and old people need much more assistence to even reach a lifeboat. As such, how more children and parents someone had aboard, how less likely that person was to survive.

3 Decision trees

Apart from rule set learning, we also experimented with decision trees. Since we are working with a mix of nominal and numeric data, this is a suitable model, because each node can use its own *rule* for splitting on an attribute. We will describe how we tweaked the learning parameters in order to achieve better results. The initial tree is shown in Figure 2.

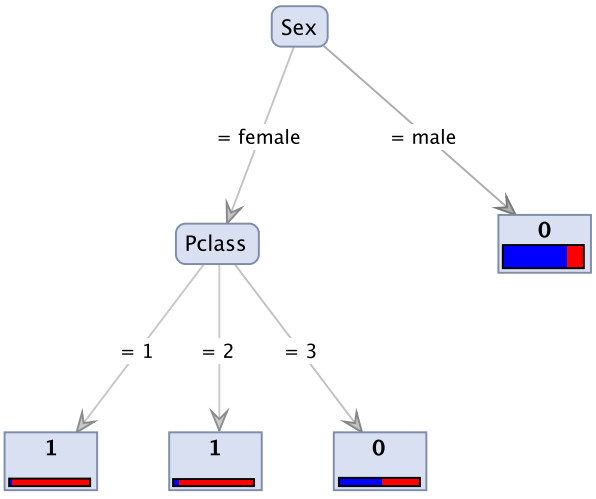

Figure 2: Tree learned with default parameters

The most obvious observation about this initial model is that female passengers with third class tickets are not yet classified very accurately, yet the tree did not split on it. In order to improve on this, we can adjust the

splitting criterion to *accuracy*. It will split nodes such that the accuracy of the tree increases. This results in a very sparse and large tree, which is not shown for exactly that reason. By changing the maximal depth of the tree, we can prevent it from getting too large. A maximal depth of 4 was found to provide good results. The result is shown in Figure 3. As we can see, this

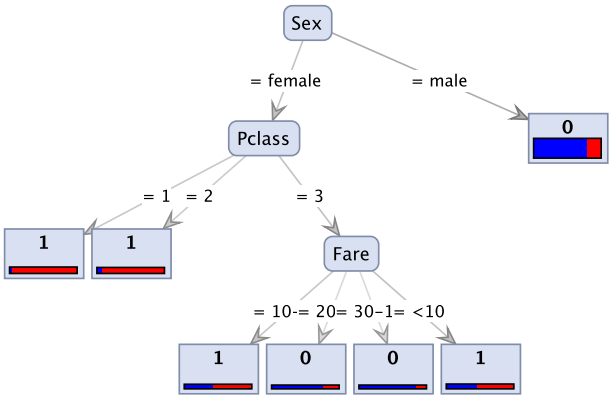

Figure 3: Tree learned with *accuracy* splitting criterion and maximum depth of 4.

achieved the desired result of splitting the undecided node in four new nodes, three of which perform clearly better. The final step is improving prediction for males. While its accuracy is quite high, the recall for surviving males is zero. This probably happens because the gain in criterion is not sufficiently high for the pre-pruning step to split the node. By lowering the minimum gain during pre-pruning, we enable the algorithm to split easier on nodes. Because the `Fare` node is already split until max depth, we expect that only the rightmost node (males) will be further split by this setting. Figure 4 confirms this assumption and shows the final model.

This model is pretty intuitive. For females, Class is the most important deciding factor. Amongst the lowest class women, the ones that payed most for their ticket surprisingly have the lowest survival rate. For males, Age is the most important deciding factor. Children have a far higher survival rate than men.

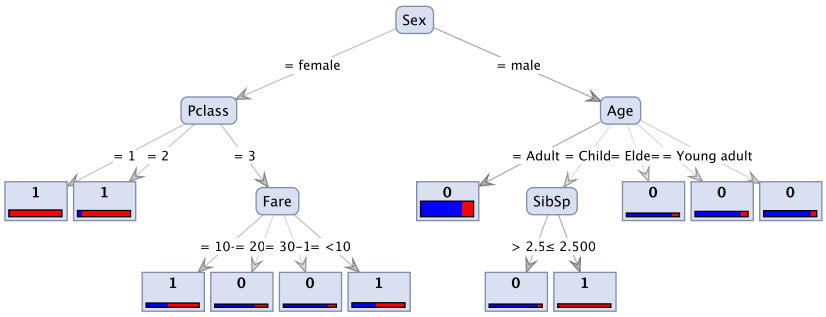

Figure 4: Tree learned with *accuracy* splitting criterion, maximum depth of 4 and minimum information gain of 0.01.

4 Rule learning vs. Decision trees

We conclude this report with a brief summary of our results and our conclusion about the methods. To decide if either Rule learning is better than Decision trees we look at three performance measures, namely: **accuracy**, **precision** and **recall**.

With *accuracy* we mean the proportion of true results (both true positives and true negatives) among the total number of cases examined. For the Titanic case this would mean:

$$accuracy = \frac{\#\text{true dead} + \#\text{true alive}}{\#\text{true dead} + \#\text{false dead} + \#\text{true alive} + \#\text{false alive}}$$

Precision is the fraction of retrieved instances that are relevant:

$$precision = \frac{\#\text{true alive}}{\#\text{true alive} + \#\text{false alive}}$$

In following table we give the accuracy, precision and recall of our best configuration. From these scores we decided what was in our opinion the best method.