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+.

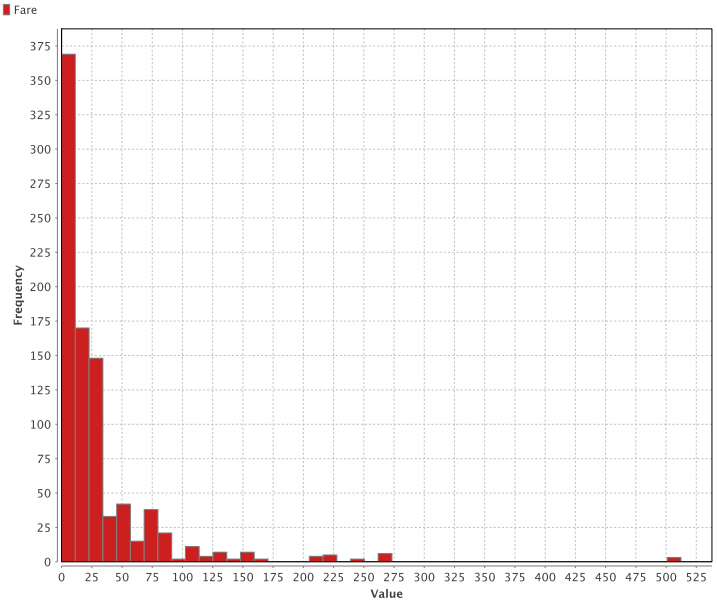

Figure 1: Histogram for the `Fare` feature

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 witch decision trees. First we applied the standard decision tree in RapidMiner, it created a too complicated decision tree, as you can see in figure 1.

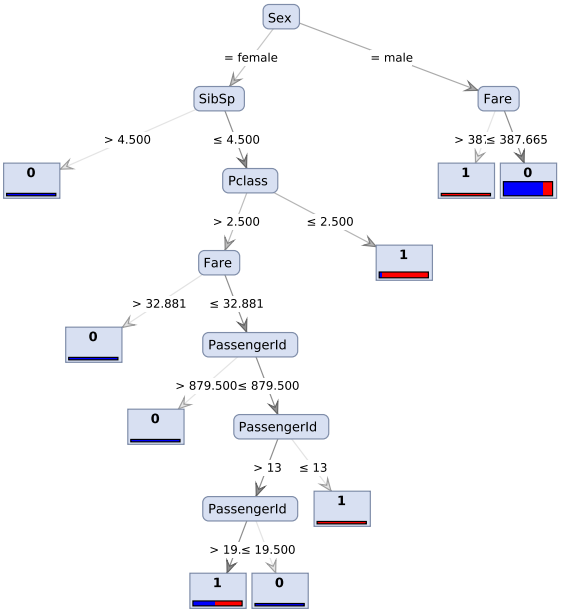

Figure 2: Original decision tree with maximal depth of 20

Next we limited the maximal depth of the decision tree to 3. This gave us figure 2, which is much more usefull.

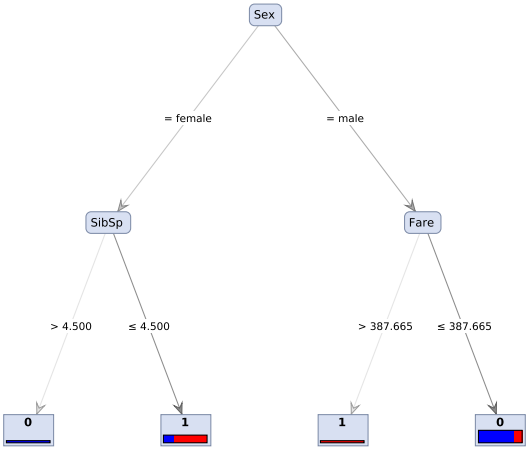

Figure 3: Decision tree with maximal depth of 3

4 Rule learning vs. Decision trees

Settings for Rule learning 1:

- ...

Settings for Rule learning 2:

- ...

Settings for Decision tree 1:

- ...

Settings for Decision tree 2:

- ...

	Precision	Accuracy	Recall
Rule learning 1			
Rule learning 2			
Decision tree 1			
Decision tree 2			

Figure 4: Overview