TDT4136 - Artificial Intelligence Methods
Assignment 4 - Decision Trees

Elias Søvik Gunnarsson, MTIØT

March 8, 2021

Problem 1 - Decision Trees

a) The columns witch are considered to be categorical variables with at finite number of possibilities are:

- Pclass
- Sex
- Embarked

Age and **Cabin** are disregarded due to missing values. **Name**, **Ticket** and **Ticket** are continuous variables, as well as **SibSp** and **Parch** under the assumption that both number of siblings/spouses and number of parents/children can be infinite.
Output from running ***dtl_categorical.py***:

```
{'Sex - female': {'female': {'Pclass - 1': {1: {'Embarked - C': {'C': 1},
                                                'Embarked - Q': {'Q': 1},
                                                'Embarked - S': {'S': 1}
                                                                        }},
                             'Pclass - 2': {2: {'Embarked - C': {'C': 1},
                                                'Embarked - Q': {'Q': 1},
                                                'Embarked - S': {'S': 1}
                                                                        }},
                             'Pclass - 3': {3: {'Embarked - C': {'C': 1},
                                                'Embarked - Q': {'Q': 1},
                                                'Embarked - S': {'S': 1}
                                                                    }}}},
 'Sex - male': {'male': {'Embarked - C': {'C': {'Pclass - 1': {1: 0},
                                                'Pclass - 2': {2: 0},
                                                'Pclass - 3': {3: 0}}},
                         'Embarked - Q': {'Q': {'Pclass - 1': {1: 0},
                                                'Pclass - 2': {2: 0},
                                                'Pclass - 3': {3: 0}}},
                         'Embarked - S': {'S': {'Pclass - 1': {1: 0},
                                                'Pclass - 2': {2: 0},
                                                'Pclass - 3': {3: 0}}}}}}
```

Dictionary representation used for creating tree with **Graphviz**.
Accuracy on Training set (**train.csv**) and Test set (**test.csv**) while using decision tree:

```
Accuracy test:  0.8752997601918465
Accuracy train:  0.8431590656284761
```

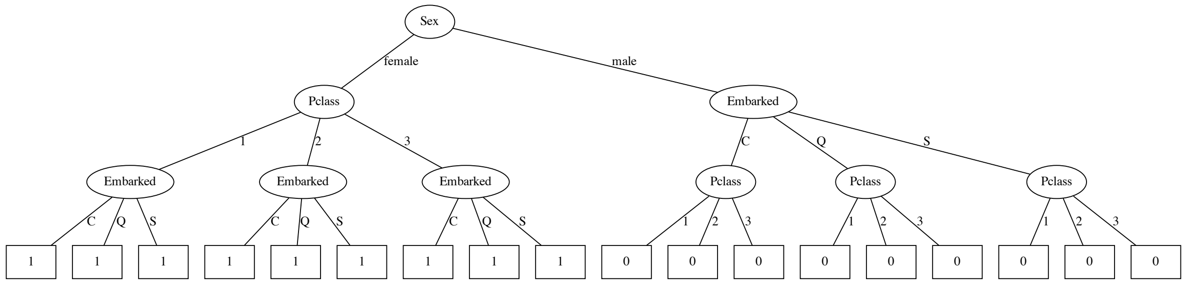

Figure 1: Decision Tree for 1a

b) **Name** and **Ticket** are considered as not relevant since they are not numerical values and can be infinitely different. **SibSp**, **Parch** and **Fare** are to be splitted once, where the plan was to make the best split by evaluating possible split points in the sorted examples where examples have neighbouring examples which have a different classification. Hence choose the split with the best information gain, which was implemented in the **Importance** function. In this way the the attributes can be treated as categorical attributes like in **a)**.

c) A higher test accuracy than train accuracy can explained by the fact that the test set is smaller than the train set, which increases the possibility of higher test accuracy than train accuracy.

In the exercise we were limited to only splitting the continuous attribute *once*. Splitting the continuous attribute more than once can increase the accuracy of the decision tree. However, splitting continuous and integer-valued input attributes is the most expensive part of real-world decision tree learning applications [1]. This can also lead to overfitting if there are too many splits.

One technique which can be used to increase the algorithm's performance is pruning, which can limit overfitting the decision tree by eliminating the irrelevant nodes in the tree. Pruning the tree which results in a smaller tree can decrease the accuracy, but can increase the running performance. This is not an issue in the exercise, but rather larger, more complex and less interpretable trees.

Problem 2 - Missing Values

In the dataset for this exercise we chose to not use the columns with missing values. As stated in the exercise set, this means that we are missing out on a lot of potentially useful information. An alternative approach, which is quite similar to the approach we have already done, is to not disregard the entire column, only the given examples/rows with the missing values. This can be done by simply cleaning the data in the beginning, and can be effective especially in this case where the number of missing values is quite small.

Another approach is to either assign the most common value for the attribute for the missing values or the most common value for the examples with same classification. A third option is to use all possible values i.e. if one example was missing the 'Sex'- attribute, we will use both 'male' and 'female' as values. Each of the possibilities need to be assigned a probability, which will be the weight of the values frequency. The fraction of the example shall then be assigned to each of the descendants in the decision tree. Opposed to the other alternatives which I have discussed, the consequence of a wrong value is weighed out, which can result in a smaller error, whereas choosing entirely wrong value can do more damage, than to simply not use the example.

Bibliography

[1] P. N. Stuart Russell, *Artificial Intelligence: A Modern Approach (4th Edition) (Pearson Series in Artifical Intelligence)*, 4th ed. Language: English, 2020.