



DEPARTMENT OF COMPUTER SCIENCE

TDT4171 METODER I KUNSTIG INTELLIGENS

Assignment 4 - Decision Trees

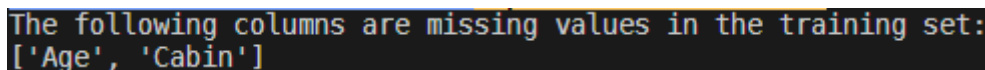
JOSTEIN HJORTLAND TYSSE

Table of Contents

1	Decision Trees	1
a)	Categorical Variables	2
b)	Continuous Variables	3
c)	Discussion	4
2	Missing Values	5

1 Decision Trees

Missing Values As mentioned in the exercise, some columns are missing values. If we use Pandas to check which columns that are missing values in the given data set, we get that *Age* and *Cabin* is missing values, as seen in Figure 1. Running the same test on the test set shows that the same columns are missing values there. Thus, we will ignore these columns for the implementation of the Decision Tree.



```
The following columns are missing values in the training set:
['Age', 'Cabin']
```

Figure 1: Missing values

We begin with classifying the variables as categorical or continuous, and explaining whether they are relevant or not. Table 1 shows a summary of the classification.

Pclass can take values {1,2,3}, and is thus categorical. It may be relevant to the model, as passengers with higher class might have been prioritized when boarding the lifeboats.

Name is not suitable, as it will most likely be unique for every person.

Sex is relevant, as the women and children were prioritized when boarding the lifeboats. The variable is categorical, and can take the values {*male*, *female*}.

Age is an integer-valued variable with potentially infinitely possible values (technically limited by the maximum lifetime of a human). As mentioned earlier, some values are missing from this columns, and we will not use Age in this task. It could have been relevant to include, since children were prioritized.

SibSp is a continuous variable. It is relevant to look at, since it might be the case that people with no siblings ran straight to the lifeboats, but the other ones spent time looking for their family members, and may not have made it.

Parch is continuous, and is relevant for the same reason as SibSp; people with many family members on board may be more likely to not survive because they search for the rest of the family, or even sacrifice themselves by giving their spot in the lifeboat to someone else in the family.

Ticket is continuous, and is unique for each person. The ticket number is not suitable to include in the model, as it should not have any impact on whether a person survived or not.

Fare is continuous. It may be the case that people who had paid a lot for the tickets were prioritized. This might be connected to the Pclass variable, as higher class probably costs more.

Cabin is continuous. Some rows are missing values from the Cabin columns, so we will not use this variable. It would have been interesting to see if people with cabins near lifeboats were more likely to survive.

Embarked is categorical, and can take the values {C, Q, S}. It should not have any impact on the result, as it should not matter where you got on the ship. All that matters is that you were on the ship when it crashed in the iceberg. We will thus not include Embarked in the model.

Variable	Classification	Suitable
Pclass	Categorical	Yes
Name	Continuous	No
Sex	Categorical	Yes
Age	Continuous	Yes
SibSp	Continuous	Yes
Parch	Continuous	Yes
Ticket	Continuous	No
Fare	Continuous	Yes
Cabin	Continuous	Yes
Embarked	Categorical	No

Table 1: Classification of variables

a) Categorical Variables

The relevant categorical variables to look at are **Sex** and **Pclass**. Implementing Decision-Tree-Learning, we can generate the decision tree in Figure 2. Evaluating the tree with the test set, we get about 87.5% accuracy, as seen in Figure 3.

Note that **Pclass** does not actually affect the results, and could be pruned away. Thus, using only one variable, **Sex**, we are able to get 87.5% accuracy by predicting that women survive and men does not. According to Wikipedia, 74% of the women survived, and 20% of the men [1], so it makes sense that this prediction will perform quite well.

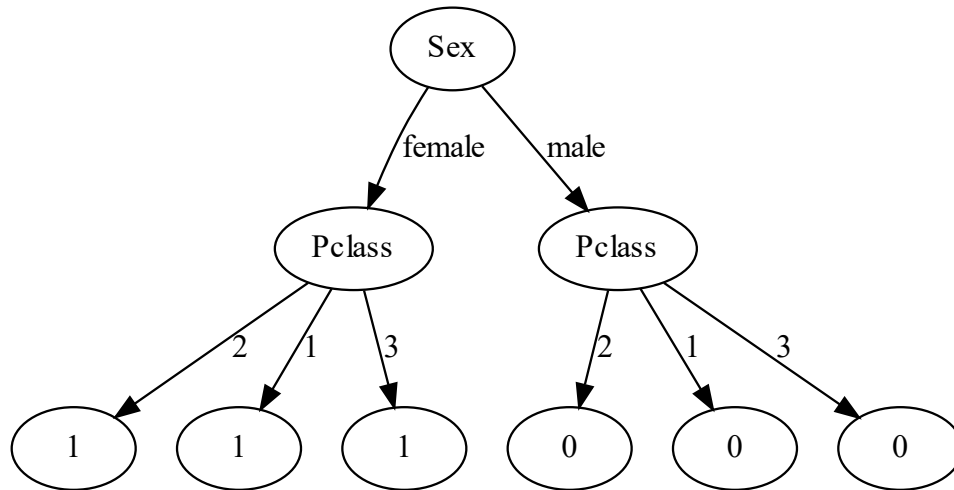


Figure 2: Decision Tree, Categorical variables

Accuracy is 87.52997601918466 %

Figure 3: Score, Categorical variables tree

b) Continuous Variables

The results of testing on the test set, shows that the tree has an accuracy of about 88.5 %, as seen in Figure 5.

Figure 4: Decision Tree, Continuous variables

Accuracy is 88.48920863309353 %

Figure 5: Score, Continuous variables tree

Only adding Siblings/Spouses

One interesting result can be found when only adding the SibSp continuous variable. In Figure 6 we can note that women on class 3 with more than 2 siblings/spouses are predicted to not survive, otherwise the result is identical to the tree with only categorical variables: women survive, men do not. This supports the initial thoughts of prioritizing higher passenger classes, and that people with many family members on board may be more likely to not survive.

The accuracy is about 88.7 % (see Figure 7), which is actually higher than when we had more variables.

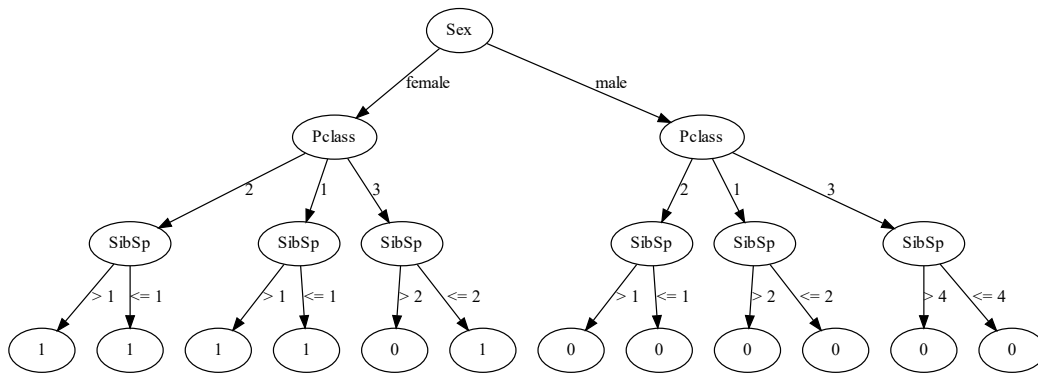


Figure 6: Decision Tree: Sex, Pclass, SibSp

Accuracy is 88.72901678657075 %

Figure 7: Score: Sex, Pclass, SibSp

c) Discussion

Using only categorical variables, we get an accuracy of 87.5%, and including the suitable continuous variables, we get 88.5%. Since we know that women and children were prioritized [1], it is natural that **Sex** is the main deciding variable for survival. It is thus not very likely that the other variables would have a huge impact on the chance of survival, so we would not expect a big increase of accuracy.

Only one percent increase may be a little lower than expected. One reason for the relatively low increase, might be overfitting. We saw that when we decreased the number of input variables, the accuracy became a little bit better, which indicates that the original tree with all the continuous variables suffers from overfitting.

To improve the performance of the tree, we could implement a pruning technique like χ^2 pruning. This technique combats overfitting by pruning nodes that seem irrelevant. By reducing the chance for overfitting, the performance of the decision tree may increase.

Another technique that may improve performance is to use a Wrapper like the Cross-Validation-Wrapper from Russel and Norvig [2], and use our existing DTL algorithm (with some minor modifications like building the tree breadth-first instead of depth-first) as input for the Wrapper.

The wrapper will start with with small, simple models, and gradually increase the complexity. At each iteration, we use cross-validation on the error rates in training and test sets. Normally, the error rate in the test set will decrease at first, but will start to increase when we are overfitting the model [2]. The Wrapper will then find the best candidate for an optimal size for the model, and use this to build a decision tree. Since this approach aims to reduce overfitting, it may lead to an improvement of the performance compared to our original DTL implementation.

2 Missing Values

Age and **Cabin** attributes are missing some values. We have chosen to completely ignore these columns, but they could have improved the accuracy of the decision tree. We will now look at different approaches for handling missing values.

Average

For numerical attributes like **age**, we can find the average age in the data set, and replace the missing **age** values with this. One advantage with this method is that it will not affect the average sample value of the attribute. However, this approach will not be optimal if the different attributes are connected. For example it may be the case that there is a relationship between **age** and **SibSp**, and the *average* approach will not consider that relationship.

Using the *Average* approach requires the attribute to be of a kind that makes sense to make an average of. **Cabin** is not that kind of attribute, as it would not make any sense to find the “average cabin” and assign all missing values to this new average cabin. If we however had data about family relations, it could be possible to assign the missing values to the most frequent cabin of the others in the family.

Delete rows with missing values

One approach is to simply delete all rows with a missing value from the data set. The big advantage for this method, is that it is trivial to implement. There are some drawbacks to this method, since we are deleting data that could have been useful for the model. If there is a connection between the missing values, e.g. it is not random which data is missing, the approach of deleting the rows will introduce bias. By deleting rows, the size of the training set will decrease, and a smaller training set usually results in lower accuracy.

Consider all possible values

This approach pretends that the sample with a missing value instead has all the possible values for the missing attribute. To improve accuracy, the different values are weighted according to its frequency among the examples leading to the node [2]. One advantage with this approach, is that it is likely to be accurate if implemented in a suitable way. It is for example hard to use this approach if you have continuous variables. It is not possible to consider all the infinite values the variable can take, so we cannot use the method.

Look at similar samples

It is possible to look at other samples where the values of the other attributes are similar to the ones in the sample with the missing value. The advantage with this method is that it can be quite easily implemented by creating an ordered set of the samples based on a selection of attributes. This approach will make sense if we assume that the variables are connected to each other. However, one big disadvantage is that the method may increase bias, especially if the samples with missing values are close in the sorted ordered set, since you will end up reusing the same value multiple times.

References

- [1] Wikipedia. *Women and children first*. 2021. URL: https://en.wikipedia.org/wiki/Women_and_children_first (visited on 03/16/2021).

-
- [2] Stuart Russel and Peter Norvig. *Artificial Intelligence, A Modern Approach*. 3. Pearson Education, inc., 2010.