MIS772

Predictive Analytics

Workshop: Intro to Classification

Decision trees and k-NN with holdout validation

DEAKIN BUSINESS SCHOOL

AACSB ACCREDITED

EFMD EQUIS ACCREDITED

Workshop Plan

Objectives:
The task is to create a predictive model classifying all Danish AirBnB rental properties into "cheap" (price/night < $100) and "expensive". Visualize properties located in the Danish region of Northern Jutland by generating a new attribute.

Data Set:

AirBNB-DK.csv (From Unit Website)

Acknowledgements: *http://tomslee.net*

Method:
Attend the workshop, follow the tutor's demo and instructions, take notes. Note that the class and online seminar will be recorded and their videos linked to the CloudDeakin topic for later access and study.

1 **Prepare data for modelling**

(a) Load data from file
(b) Filter out data with missing values
(c) Generate new attributes, using function expressions
(d) Discretize an attribute
(e) Select predictors and a label attribute
(f) Set role (label)

2 **Train and validate decision tree model**

(a) Split data into training and validation partition
(b) Add a ***k-NN*** or ***Decision Tree*** model and train it
(c) Apply the model to validation data
(d) Measure and interpret the model and performance
(e) Save the RM process (use versions)

3 **Discuss, extend and reflect**

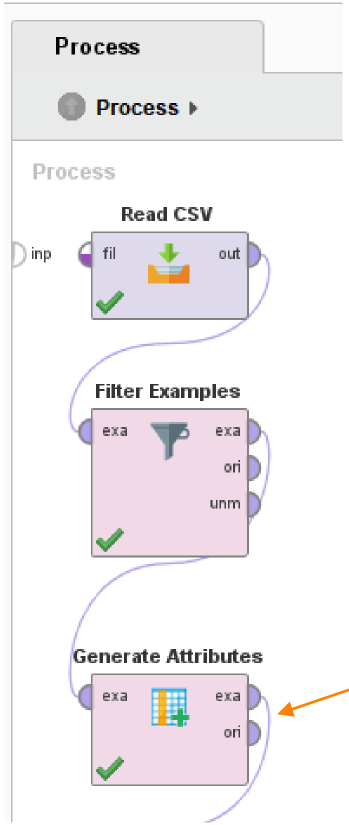
Visualise geospatial data

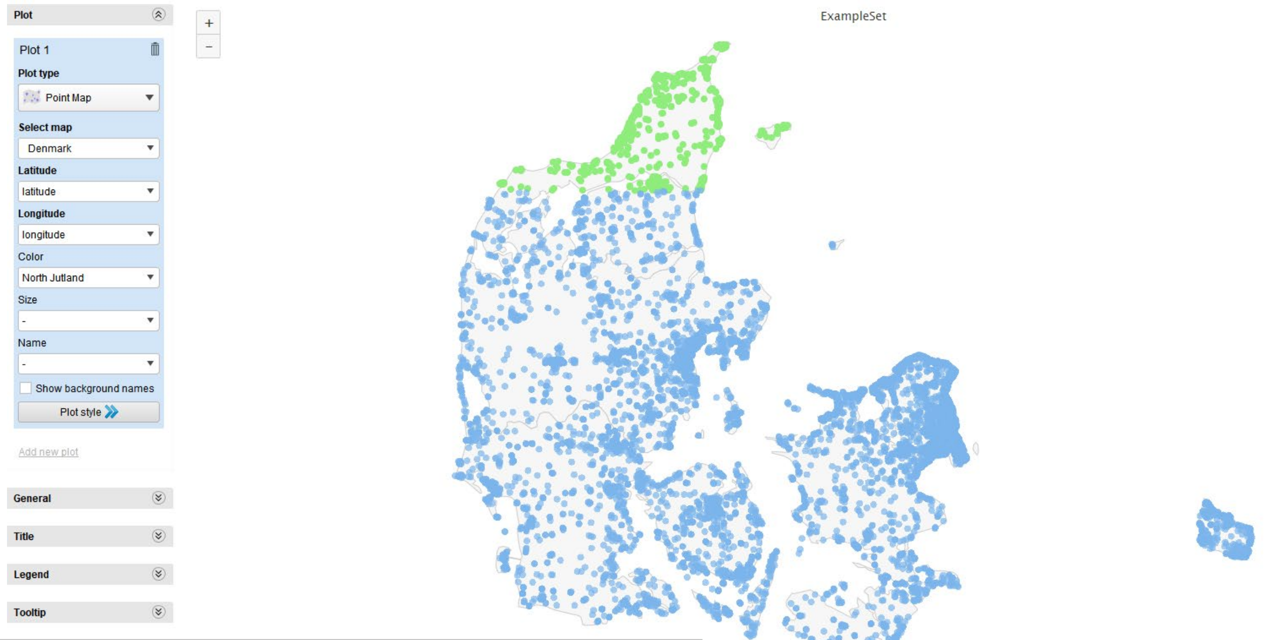
Watch a lecture demo. Follow the tutor. Initially create an RM process responsible for data pre-processing, similar to the one below. It will load the data from a .csv file, filter out data with missing attributes, generate new attributes (including identifying North Jutland properties). Visualize properties located in North Jutland using a point map.

How can this be done?

What results does it produce?

How does it work?





Plot type: Point Map

Select map: Denmark

Latitude: latitude

Longitude: longitude

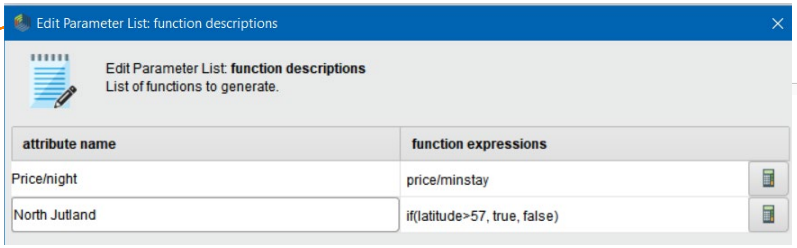
Color: North Jutland

Size: -

Name: -

□ Show background names

Plot style »



Edit Parameter List: **function descriptions**
List of functions to generate.

attribute name	function expressions
Price/night	price/minstay
North Jutland	if(latitude>57, true, false)

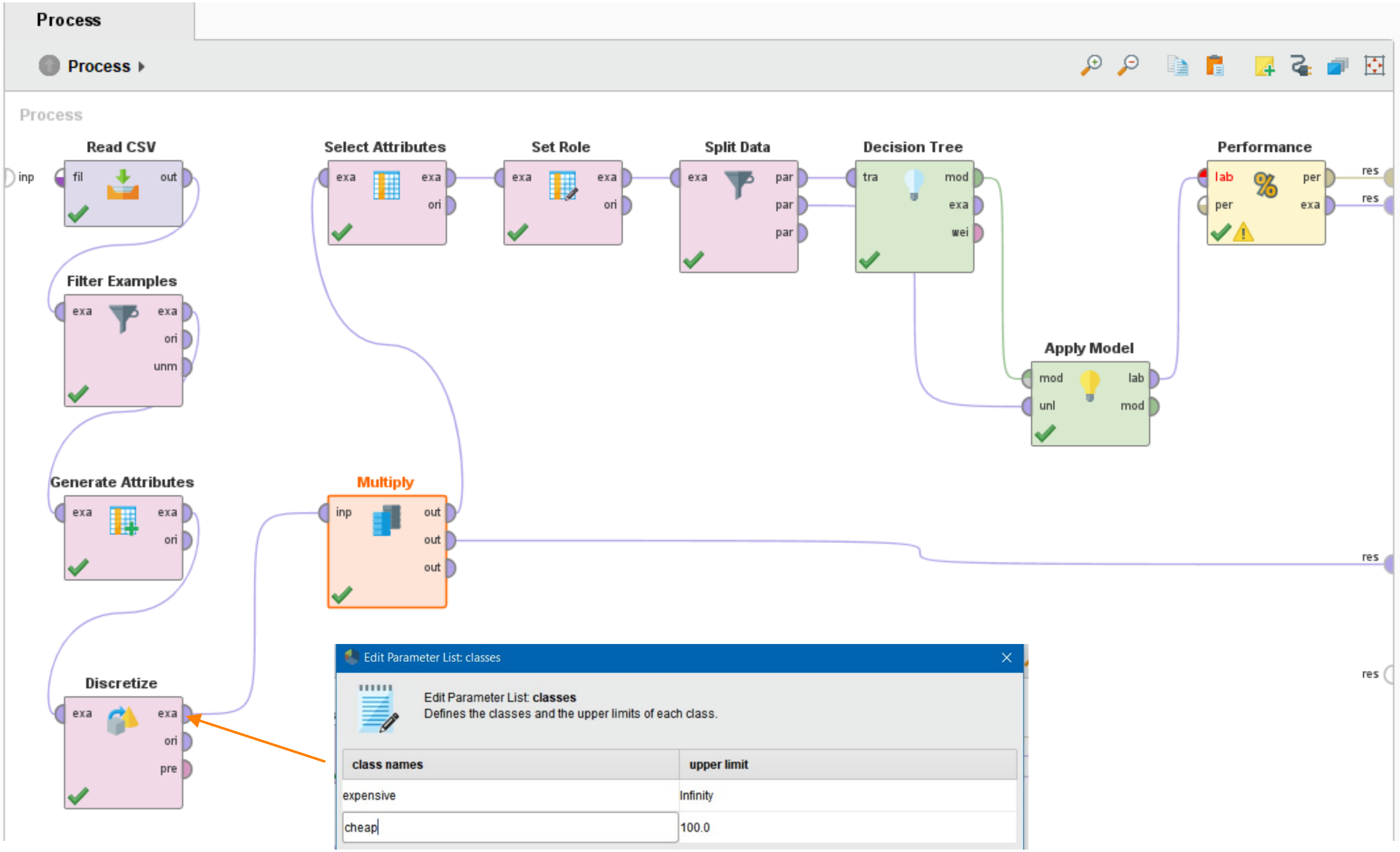
Discretize attribute

Discretize the price/night as cheap (<$100) or expensive.

How can this be done?

What results does it produce?

How does it work?



Edit Parameter List: classes

Edit Parameter List: **classes**
Defines the classes and the upper limits of each class.

class names	upper limit
expensive	Infinity
cheap	100.0

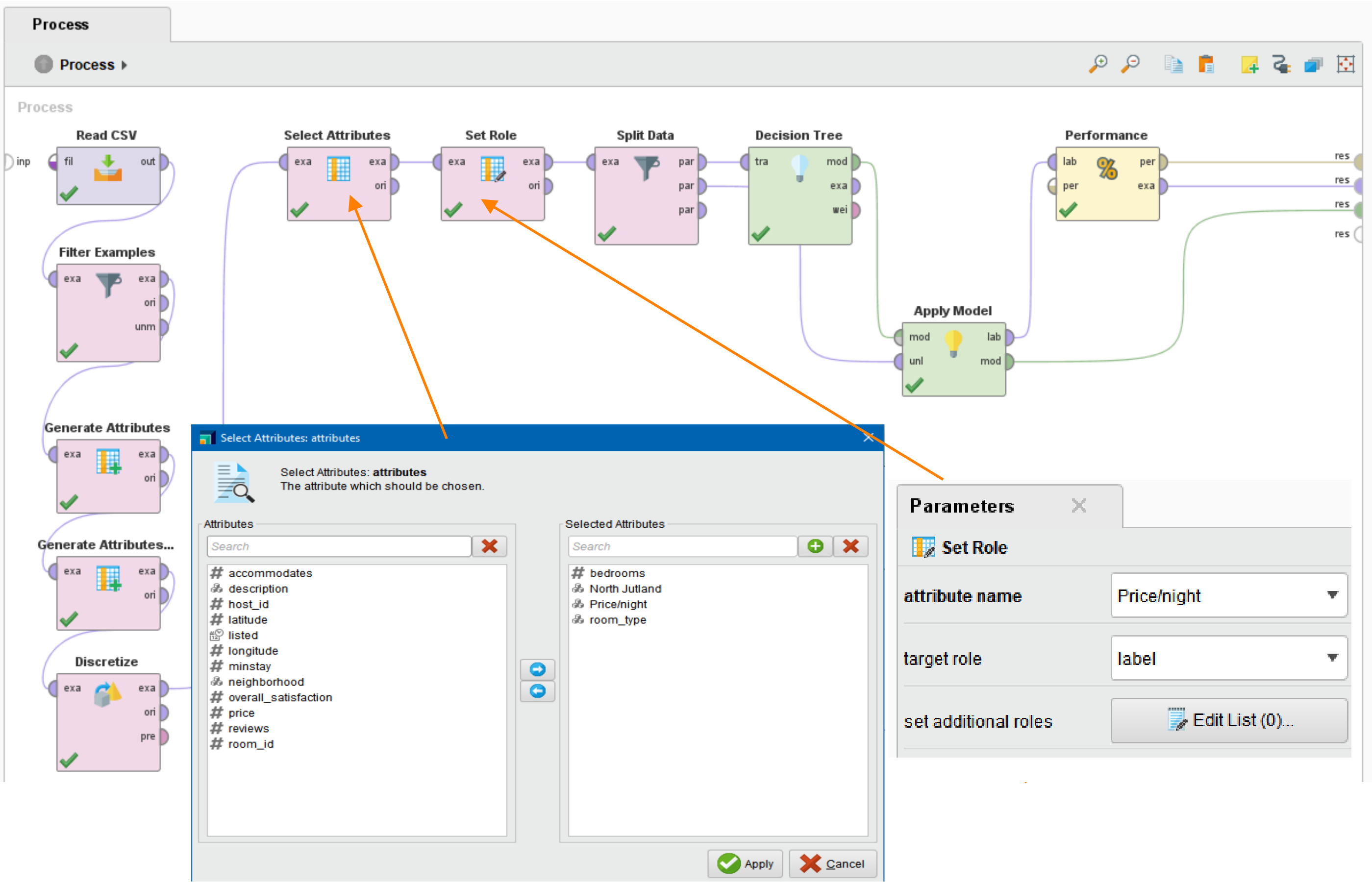
Select predictor attributes and set role

Select predictor attributes and set the role of the price/night category as a label. Run and check the results.

How can this be done?

What results does it produce?

How does it work?



Select Attributes: attributes

Select Attributes: **attributes**
The attribute which should be chosen.

Attributes

- accommodates
- description
- host_id
- latitude
- listed
- longitude
- minstay
- neighborhood
- overall_satisfaction
- price
- reviews
- room_id

Selected Attributes

- bedrooms
- North Jutland
- Price/night
- room_type

Set Role

attribute name	Price/night
target role	label
set additional roles	Edit List (0)...

The End Result: The RM Process

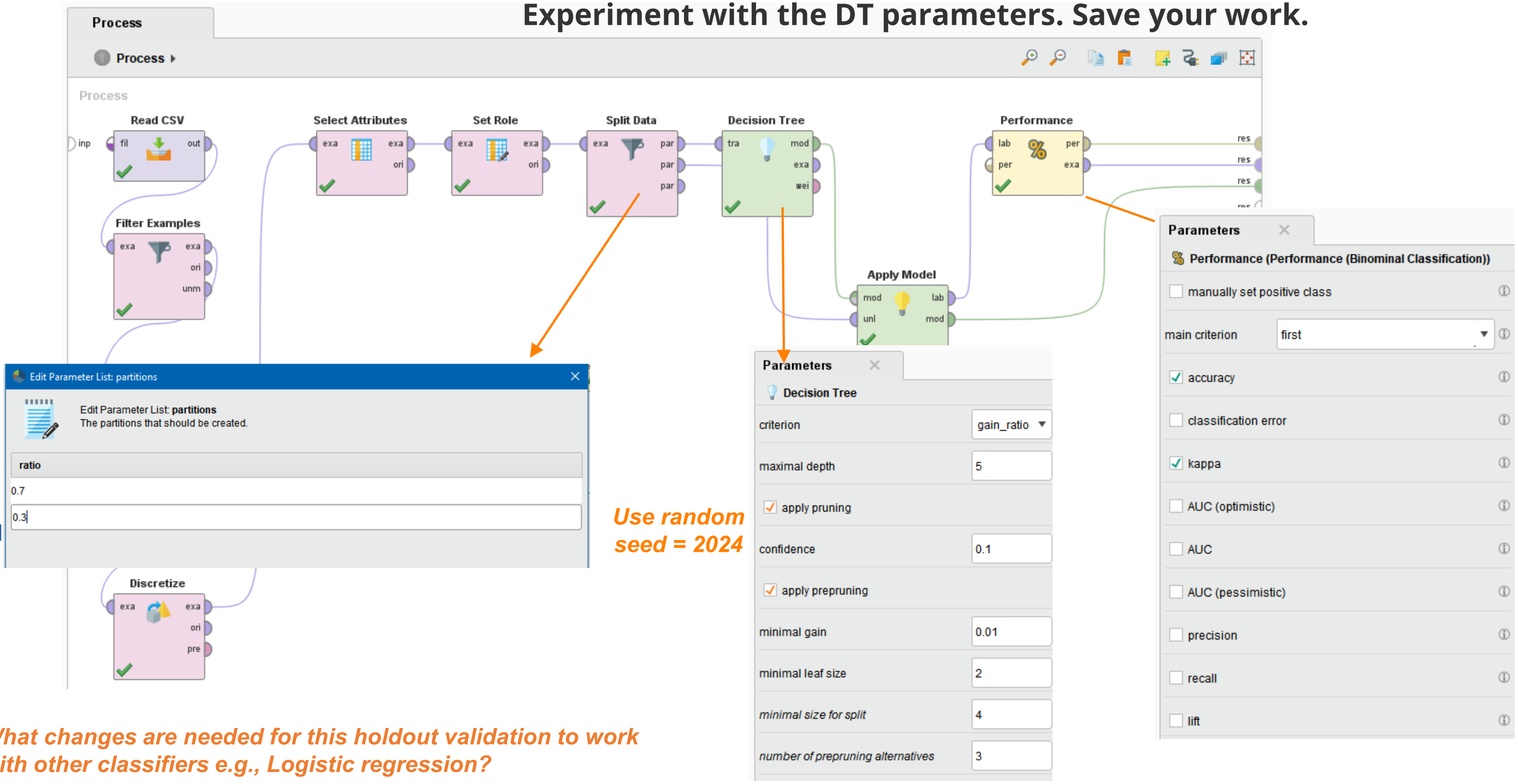
Next we will develop model training and validation. You will add and test one operator at a time. First split data into two parts 70-30%, next add a k-NN or a decision tree, then apply the resulting model to validation data, and finally analyse the predictions using binomial performance (accuracy and kappa). Run and check the results. Experiment with the DT parameters. Save your work.

We will extend the previous process.

What results does it produce?

How does it work?

Make sure: You do a little, test a little, think a little and save a lot!



Edit Parameter List: partitions

Edit Parameter List: **partitions**
The partitions that should be created.

ratio
0.7
0.3

Parameters

Decision Tree

criterion	gain_ratio
maximal depth	5
☑ apply pruning	
confidence	0.1
☑ apply prepruning	
minimal gain	0.01
minimal leaf size	2
minimal size for split	4
number of prepruning alternatives	3

Parameters

Performance (Performance (Binominal Classification))

☐ manually set positive class

main criterion: first

☑ accuracy

☐ classification error

☑ kappa

☐ AUC (optimistic)

☐ AUC

☐ AUC (pessimistic)

☐ precision

☐ recall

☐ lift

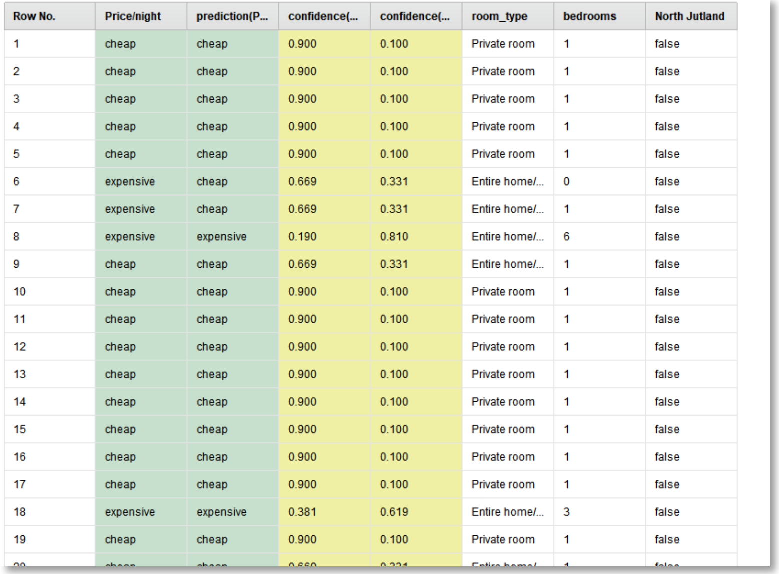
Use random seed = 2024

What changes are needed for this holdout validation to work with other classifiers e.g., Logistic regression?

The End Result: Tables & Charts

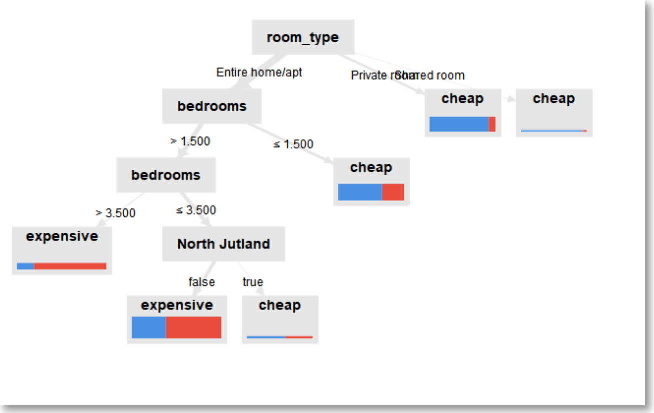
Explore classification results. View the results table – inspect label vs prediction, how should we read the confidence columns. Then, look at the tree model. Analyse the model performance, is it good or bad? Can it be improved? How? How can the skills gained transfer to your assignment?

What are confidence columns and how do they relate to prediction?

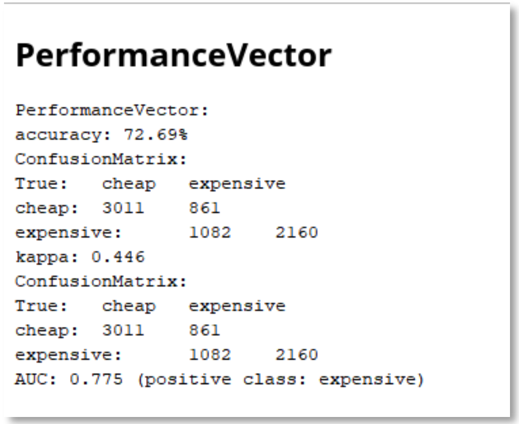


Row No.	Price/night	prediction(P...	confidence(...	confidence(...	room_type	bedrooms	North Jutland
1	cheap	cheap	0.900	0.100	Private room	1	false
2	cheap	cheap	0.900	0.100	Private room	1	false
3	cheap	cheap	0.900	0.100	Private room	1	false
4	cheap	cheap	0.900	0.100	Private room	1	false
5	cheap	cheap	0.900	0.100	Private room	1	false
6	expensive	cheap	0.669	0.331	Entire home/...	0	false
7	expensive	cheap	0.669	0.331	Entire home/...	1	false
8	expensive	expensive	0.190	0.810	Entire home/...	6	false
9	cheap	cheap	0.669	0.331	Entire home/...	1	false
10	cheap	cheap	0.900	0.100	Private room	1	false
11	cheap	cheap	0.900	0.100	Private room	1	false
12	cheap	cheap	0.900	0.100	Private room	1	false
13	cheap	cheap	0.900	0.100	Private room	1	false
14	cheap	cheap	0.900	0.100	Private room	1	false
15	cheap	cheap	0.900	0.100	Private room	1	false
16	cheap	cheap	0.900	0.100	Private room	1	false
17	cheap	cheap	0.900	0.100	Private room	1	false
18	expensive	expensive	0.381	0.619	Entire home/...	3	false
19	cheap	cheap	0.900	0.100	Private room	1	false
20	cheap	cheap	0.669	0.331	Entire home/...	1	false

View the decision tree visualisation. Experiment with the model parameters

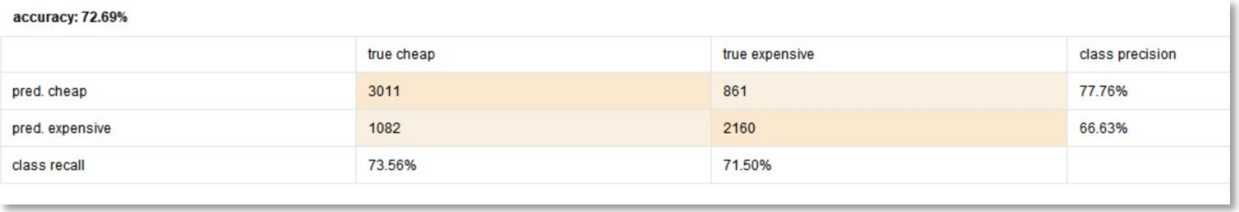


What does it mean?



PerformanceVector

```
PerformanceVector:
accuracy: 72.69%
ConfusionMatrix:
True:   cheap    expensive
cheap:  3011     861
expensive:       1082     2160
kappa: 0.446
ConfusionMatrix:
True:   cheap    expensive
cheap:  3011     861
expensive:       1082     2160
AUC: 0.775 (positive class: expensive)
```



accuracy: 72.69%

	true cheap	true expensive	class precision
pred. cheap	3011	861	77.76%
pred. expensive	1082	2160	66.63%
class recall	73.56%	71.50%	

Check the model performance. What is its accuracy and kappa, can accuracy be trusted? (more on this later)