

ICT for Health

Laboratory # 3

Decision tree and chronic kidney disease

Monica Visintin

Politecnico di Torino



2018/19

Table of Contents

1 Chronic kidney disease

2 Laboratory # 3

- Prepare the data
- Generation of the decision tree with Python
- Report

Chronic kidney disease [1]

See the slides by Prof. Pagana for a description of the disease.

Table of Contents

1 Chronic kidney disease

2 Laboratory # 3

- Prepare the data
- Generation of the decision tree with Python
- Report

Prepare the data [1]

- Download from https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease the Data Folder and Data Set Description.
- Use Pandas `read_csv` class to import the data frame, but note that:
 - 1 the initial rows of the file `chronic_kidney_disease.arff` contain the description of the features and should be skipped
 - 2 the file is a normal csv file that uses `,` as field separator
 - 3 some rows have an extra `,` so that 26 columns are read instead of 25
 - 4 many data are missing (identified with `?`)
 - 5 some columns contain numerical features, other columns contain categorical features
 - 6 there are “hidden” typing errors (a “yes” becomes a “ yes” with an extra blank or a “ yes”, with an extra tab)

Prepare the data [2]

- You need to perform an initial **cleaning on the data**, which is a task that is almost always needed, takes time, and is frustrating. It is better you get used to this preprocessing phase!
- First option: you clean the data **manually**, by editing the original csv file (it works if you have very short files...)
- Second option: you **exploit the arguments of Pandas** `read_csv`.
 - ① You can specify that the separator is `,` by writing `sep=','`
 - ② You can skip the first 5 lines (for example) by writing `skiprows=5`
 - ③ You can specify that no header is present in the file (i.e. that there is no first row with the names of the features) by writing `header=None`
 - ④ You can specify that `?` is not a number by writing `na_values=['?', '\t?']`
 - ⑤ You can specify the feature names by writing `names=feat_names` where `feat_names` is a list that contains exactly as many elements (typically strings) as the columns in the file (example `feat_names=["a","b","c"]`)

Prepare the data [3]

- Even if the hierarchical classification based on mutual information works with **categorical data**, the **Scikit Learn implementation requires numerical data** (a clear limitation of Python). It is then necessary to map all the values of **categorical features into numbers**. You can do this exploiting the method `replace` of Pandas Dataframes
- The file has 24 features whereas the 25-th column is the class to be estimated (either “ckd” -chronic kidney disease- or “notckd”)

Management of missing data

- 1 remove from the dataset all rows (patients) for which you have less than 20 valid data
- 2 select the rows (patients) with 25 valid data, and generate matrix X (with 25 columns, including the last one, storing the class) and normalize it
- 3 case of a row with **just one missing value**: if the missing value is in column j , then (a) define `data_train` equal to X without column j and `y_train` equal to column j of X ; (b) find the optimum weight vector w of linear regression using ridge regression with Lagrangian multiplier λ equal to 10; (c) regress the missing value in the row (remember to denormalize); (d) if the missing value is that of a categorical feature, then round the floating point number you get from regression into an integer number, check that the integer value you get is in the range of integer values for that feature
- 4 extend the method to the rows in which the missing features are more than one
- 5 get the cleaned dataset with N_p rows and $N_f = 25$ columns without missing values (N_p is the number of patients of the original dataset with at least 20 valid features)

Classification tree in Python [1]

- Scikit Learn (library of Python) has the function that implements C4.5 (or a similar algorithm) for the hierarchical classification
- The Scikit Learn class that implements hierarchical/decision trees is `tree.DecisionTreeClassifier('entropy')`
You must first instantiate an object of that class, and then perform the training:

```
clf = tree.DecisionTreeClassifier("entropy")  
clf = clf.fit(data, target)
```

where `data` corresponds to the first 24 columns of the cleaned dataset and `target` is the 25th column of the cleaned dataset

Classification tree in Python [2]

- To **view the decision tree**,
 - use in the Python script:

```
dot_data = tree.export_graphviz(clf,out_file="Tree.dot",  
feature_names=feat_names,class_names=target_names,filled=True,  
rounded=True,special_characters=True)
```

where `feat_names` is the list with the names of the features (this produces a nicer tree) and the other arguments are used for aesthetic reasons
 - in the Windows/Linux shell give the command:

```
dot -Tpng Tree.dot -o Tree.png
```

which generates the file `Tree.png` with the tree
- Use the other methods of `DecisionTreeClassifier` (read the manual at <http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier>) to produce more information about the dataset (for example the “importance” of each feature, etc.)

Report

- Report due by December 12th 2018 at midnight, upload it in folder “elaborati” of the class webpage
- Maximum number of pages for the report is 4 (concise and complete)
- Remember to:
 - use filename xxxxxxx_report_3.pdf for the report and filename xxxxxxx_report_3.zip for the zipped file that includes the python script; xxxxxx is you matricola number (otherwise the grade is 0 out of 5)
 - use the spellchecker
 - number the figures and write the caption (the figure title is NOT the caption)
 - refer to the figures in the text using the figure number
 - number the pages and justify your text
 - use sections, subsections, etc
 - read your report 3 times before you submit it
- Your report must be understood by a professor who teaches machine learning but does not know our notation, does not know about the dataset, does not know what the lab is about.