Athens University of Economics and Business

Department of Management Science and Technology

MSc in Business Analytics—Big Data Systems

Spark Assignment

Associate Professor P. Louridas

Instructions

Your submission will consist of a Scala file containing the code that you will write.

The Problem

You will write a Scala Spark script that creates and evaluates a Random Forest classifier on a given dataset. The dataset is available at `http://mlr.cs.umass.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data` and a detailed description can be found at `https://archive.ics.uci.edu/ml/datasets/Heart+Disease`. The dataset concerns people examined for the presence of heart disease from Cleveland. Each line of the file contains values for the following:

- `age`: age in years
- `sex`: sex (1 = male; 0 = female)
- `cp`: chest pain type
  - 1: typical angina
  - 2: atypical angina
  - 3: non-anginal pain
  - 4: symptomatic
- `trestbps`: resting blood pressure (in mm Hg on admission to the hospital)
- `chol`: serum cholestoral in mg/dl
- `fbs`: fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- `restecg`: resting electrocardiographic results
  - 0: normal
  - 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
  - 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

- `thalach`: maximum heart rate achieved
- `exang`: exercise induced angina (1 = yes; 0 = no)
- `oldpeak`: ST depression induced by exercise relative to rest
- `slope`: the slope of the peak exercise ST segment
  - 1: upsloping
  - 2: flat
  - 3: downsloping
- `ca` number of major vessels (0–3) colored by flouoroscopy
- `thal`: 3 = normal; 6 = fixed defect; 7 = reversable defect
- `num`: diagnosis of heart disease (angiographic disease status)
  - 0: < 50% diameter narrowing
  - 1: > 50% diameter narrowing (yes, I know, there is no indication of what happens when we have exactly 50%)

Missing data are indicated with question marks (?).

Notes

- Although you may write your solution in a standalone Spark program, you do not have to. It is enough to put in a Scala file *all* the code that you use in the Spark shell, so that by using `:load YourFile.scala` I will be able to run it.
- If you want to learn Scala, here are some links:
  - `http://www.scala-lang.org/documentation/getting-started.html`
  - `http://www.scala-lang.org/docu/files/ScalaTutorial.pdf`
- A good book for getting up to speed with Scala fast is *Scala for the Impatient* by Cay Horstmann, Addison-Wesley, 2012. Curiously, this seems to be available online in PDF form at places.
- Documentation about classification trees in Spark is at `http://spark.apache.org/docs/latest/mllib-decision-tree.html`
- Documentation about random forests in Spark is at `http://spark.apache.org/docs/latest/mllib-ensembles.html`.
- From the description of the features, you can see that some of them are categorical. It is worth checking the performance of your model when you treat it as numerical compared to what happens when you treat it as categorical. To do that, the `categoricalFeaturesInfo` variable of the example in the lecture slides must be defined as:

```
val categoricalFeaturesInfo = Map(
  1 -> 2,
  2 -> 4,
  5 -> 2,
  6 -> 3,
  8 -> 2,
  10 -> 3,
  12 -> 4
)
```

That is, you indicate which columns are categorical and how many categorical values there are in each column. When reading the data you must make sure that these columns have values in the range $[0, n]$, where n is the number of classes (so, if a column has values 0, 3, 6, 7, these must be changed to 0–4). Although it is not required in the assignment, it is strongly advised to try to handle categorical features correctly, as this will give you some more leverage into the Scala programming language.

Go for it!