ML Exercises

This lab is split into two part. The first part is very guided and the goal to make you write a program capable of estimating whether a tumor is malign or benign according to a few features collected from a biopsy ! The second part is much more exploratory with several ML tasks on several datasets. One of the goal of this lab is to make you efficient at reading and using the documentation.

1 The winconsin breast cancer dataset

The Winconsin breast cancer dataset contains 699 cases of breast cancers. The dataset is presented here : `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)` and you can directly download the dataset at the following URL : `https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data`.

Our goal here is to determine whether a tumor is *malign* or *benign* using the provided features. Each line of the dataset represents a case and contains 11 numerical values separated by commas. Here is a description of the 11 columns :

#	**Attribute**	**Domain**
1	Sample code number	id number
2	Clump Thickness	1 - 10
3	Uniformity of Cell Size	1 - 10
4	Uniformity of Cell Shape	1 - 10
5	Marginal Adhesion	1 - 10
6	Single Epithelial Cell Size	1 - 10
7	Bare Nuclei	1 - 10
8	Bland Chromatin	1 - 10
9	Normal Nucleoli	1 - 10
10	Mitoses	1 - 10
11	Class :	(2 for benign, 4 for malignant)

1.1 Load the data into Spark

The first step for our ML application consist in reading the dataset.
DO: *Read the data.*
DO: *What does `data` looks like ?*
DO: *What is the schema of the `data` ?*
DO: *In our dataset, how many tumors are benign ? malign ?*

1.2 Splitting into training and testing

To build our model we first need to split our data into a *training* set and a *testing* set. Here we will split according to the usual 1-9 rule, which means that 90% of the dataset will be used for training while 10% will be used to test our model.
DO: *Split `data` into `test` and `train`.*

1.3 Building the model

Building the model is in ML libraries is done by creating an object to train, then fitting this object on the data.
DO: *Read the documentation on decision tree classifiers.*
DO: *Include the relevant packages for a decision tree classifier, create and train a decision tree classifier with your training data (make sure that your model does not read the target attribute).*

1.4 Testing your model

Computing predictions for the test data can be done by applying the model on the test data.

DO: *Compute and store into* `prediction` *the predictions of your model on the test data ?*

DO: *What is the accuracy of our classifier ? This can be computed "manually" (counting the number of correct predictions divided by the total number of predictions).*

DO: *What is the area under ROC of our classifier ? Use the documentation for this.*

1.5 Improving the model

ML models are generally constructed using a set of parameters (number of bins, depth, etc.). ML libraries generally have tools to help you decide which parameters are the best suited for your application. For this you generally need to build a so-called "grid" of parameters and then an "estimator" (that determines a value for each element of the grid). Then the library returns the model that has the best value according to the estimator.

Obviously tuning the parameters will favor "lucky" models. That is you need to split your training dataset to test which is the best model and only once you have a unique model you can use the test dataset to measure the performance of your model. This second splitting is by done by the ML libraries and there are two well-known ways to do it : the train-validation and the cross-validation (see course).

DO: *Use train-validation to determine a good set of parameters.*

DO: *Use cross-validation to determine a good set of parameters.*

DO: *What is the area under ROC of all theses models now ? What is the accuracy ? What is the best model ?*

2 The penguins dataset

The penguins dataset describes measurements of threes species of penguins in three different islands. The dataset is available as a CSV file on Moodle.

2.1 Exploring the data

DO: *Load the data into a dataframe* `raw_penguins` *and look at the shape of data (which columns are numerical, which are categorical, etc.).*

DO: *Remove lines with nulls and stores this into a dataframe* `penguins`*. How many lines did you loose ?*

DO: *Create a dataframe* `num_penguins` *from* `penguins` *where categorical columns are transformed into numerical columns.*

DO: *Make a scatter plot showing bill length in function of flipper length with different colors for the different species.*

DO: *Make a scatter plot showing flipper length in function of body mass with different colors for the different islands.*

2.2 Looking at the data through PCA

DO: *Create a dataframe* `timeless_penguins` *corresponding to the dataframe* `num_penguins` *without the year information.*

Our goal is to look at the PCA of `timeless_penguins` and determine whether the species attribute can be easily inferred from the first two coordinate of the PCA.

DO: *Plot the first two coordinates of the PCA for* `timeless_penguins` *without the species information and make the color of each point depend on the species.*

DO: *You should see 4 clusters. Can you determine what attributes determines these clusters ?*

2.3 Classifying penguins

DO: *Create a dataframe* `anonymous_penguins` *where the information of the island and the year does not appear.*
DO: *Split your data into a training set and a testing set.*
DO: *Train a decision tree. What is its accuracy ?*
DO: *Train a random forest. What is its accuracy ?*
DO: *Train a logistic regression. What is its accuracy ?*

2.4 ROC curves

DO: *Plot the ROC of the models you built. Warning : In Spark, you cannot compute directly the ROC curve.*

3 If you are done, other avaible tasks.

3.1 Improve the classification the Breast Cancer dataset

Consider different models for the Breast Cancer task (e.g. Logistic Regression, Random Forest model and Gradient-boosted tree classifier).

3.2 Regression task over the Wine Quality dataset

Estimate Wine Quality based on chemical components. See `https://archive.ics.uci.edu/ml/datasets/Wine+Quality` for the dataset. The two datasets can be found here : `https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/` (what happens if you train a regression on the white wine dataset and use it for the red wine dataset ?).

3.3 Movie recommendation over the MovieLens dataset

Download the MovieLens dataset here : `https://grouplens.org/datasets/movielens/` and make recommendation using the matrix factorization algorithm.