SIA 3611
Année universitaire 2022-2023

# TP Machine Learning

## TP 1 : Linear Regression

Vassilis.Christophides@ensea.fr
guillaume.renton@ensea.fr

# Table des matières

# Introduction

Linear regression is a family of machine learning algorithms aiming at adjusting a linear model to an ensemble of data. The applications range from signal reconstruction to empirical description. The given dataset was produced by the World Health Organization. It pooled the evolution of 20 features for 15 years and among numerous countries. One of the goals of this TP1 is to manipulate this dataset and try to predict the evolution of Life Expectancy through different variables. Objectives :

- Use and setup an iPython environment
- Manipulate and visualize data
- Implement a simple linear regression
- Apply the aforementioned linear regression
- Compute a $R^2$ on the generated results
- Apply Ridge and Lasso regression

# 1 Use and setup an iPython environnment

## 1.1 To do :

Execute the following cells :

```
a = 3
b = 4
c = a + b
```

```
c = c
```

```
print(c)
```

```
c
```

### Question 1

What is triggering the output display ?

## 1.2 To do :

Execute the following cells :

```
import shutil
import pkgutil

def show_acceptable_modules():
    line = '-' * 100
    print('{}\n{:^30}|{:^20}\n{}'.format(line, 'Module', 'Location', line))
    for entry in pkgutil.iter_modules():
        print('{:30}| {}'.format(entry[1], entry[0].path))
```

```
show_acceptable_modules()
```

### Question 2

What is displayed on the last output ?
Which is the used Python version ? For this first TP, you will need :

- pandas
- matplotlib
- numpy
- sklearn

Are these packages installed in this environnment ?

## 1.3 To do :

Execute the following cell

```
pandas.__version__
```

### Question 3

How would you solve this error ?

## 2 Data manipulation and visualization

### 2.1 To do

Execute the following cells :

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("data/Life_Expectancy_Data.csv")
df = df.dropna()
df.info()
```

### Question 4

Can you explain the different elements printed on the last output ?

```
df1 = df[(df.Country == "France") & (df.Year > 2010)]
print("df1: ", df1)
df2 = df[(df.Country == "France")].Year
print("df2: ", df2)
```

### Question 5

How do you interpret the new DataFrame df1 compared to df ? What represents df2 compared to df1 ?

### 2.2 To code :

What is the range of life expectancy of Belgium between 2004 and 2008 ?

### 2.3 To do :

Compute the correlation among all features

```
print(df.corr())
```

### Question 6

Which seems the most and the least promising values to use as a predictor for life expectancy ?

The function scatter of matplotlib allows to plot two values against each other. Here is the documentation about this function.

### 2.4 To code :

Plot life expectancy against one of your chosen values.

# 3 Simple linear regression

```python
import numpy as np
```

## 3.1 To code :

Select the Life Expectancy and the Income composition of resources of Belarus, Madagascar, India and Lithuania. This new Data Frame will be called df_study

## 3.2 To code :

Implement a linear least square function and apply it on your previously selected data.

## 3.3 To code :

On the same figure, draw the line corresponding to the previous result and the data points corresponding to df_study

## 3.4 To code :

Now, implement a Gradient Descend function def gradDescent(x, y, theta, alpha, iters) where x are the covariates, y the target value, theta the initial weights, alpha the learning rate and iters the number of gradient descent iterations.

## 3.5 To code :

Compute the gradient descend on df_study for 1000 iterations with different values of alpha. You may initialize theta with theta_0 = 0 and theta_1 = 1

## 3.6 To code :

On the same figure, plot the evolution of theta_0 through the iterations for each different values of alpha you chose. Do the same thing for theta_1.

### Question 7

Discuss on the role of alpha.

## 3.7 To code :

Compute the $R^2$ score for the regression on df_study. You can find an sklearn function here.

### Question 8

Is linear regression suited between the two selected variables ?

### Question 9

If not, what would be the relevant regression between these two variables ?

## 4 Multivariable Regression

### 4.1 To do :

Execute the following cells :

```
from sklearn.preprocessing import scale
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge, Lasso
from sklearn.metrics import mean_squared_error

df_study = df[(df.Country == "Belarus") | (df.Country == "Madagascar") |
(df.Country == "India") | (df.Country == "Lithuania")]
y = df_study.Life_expectancy
```

```
X = df_study[[’Adult_mortality’, ’Alcohol’, ’Total_expenditure’,
’Income_composition_of_resources’, ’Schooling’]].to_numpy(dtype=’float64’)

alphas = 10**np.linspace(10,-2,100)*0.5
```

```

ridge = Ridge(normalize = True)
coefs = []

for a in alphas:
   ridge.set_params(alpha = a)
   ridge.fit(X, y)
   coefs.append(ridge.coef_)

ax = plt.gca()
ax.plot(alphas, coefs)
ax.set_xscale(’log’)
plt.axis(’tight’)
plt.xlabel(’alpha’)
plt.ylabel(’weights’)
plt.show()
```

### Question 10

how do you interpret the plot ?

### Question 11

Is it normal that the l2 diminishes with alpha increasing ?

### Question 12

Which is the most relevant variable for Ridge ? Prove and justify your response

## 4.2 To do :

Execute the following cell

```
lasso = Lasso(normalize = True)
coefs = []

for a in alphas:
   lasso.set_params(alpha = a)
   lasso.fit(X, y)
   coefs.append(lasso.coef_)

ax = plt.gca()
ax.plot(alphas, coefs)
ax.set_xscale(’log’)
plt.axis(’tight’)
plt.xlabel(’alpha’)
plt.ylabel(’weights’)
plt.show()
```

### Question 13

Which is the most relevant variable for Lasso ? Prove and justify your response in the following cell

### Question 14

What is the main difference between Ridge and Lasso regression ? For this setup, which is the prefered method to use ? Justify your response

## 4.3 To code :

Following the previous examples, use a sklearn function to compute a multivariable least square method.

## 4.4 To code :

Dealing with too many variables can sometimes be counter-productive and it can be more interesting to remove some features. One way to evaluate the importance of each variable is to compute the f_test whose function is named f_regression in sklearn.

### Question 15

According to the f_test, which variables are the least promising ? Compare it with the correlation of your subset.