Course: **Machine Learning**

Student: **Roman Vantukh**

Lecturer: **prof. Caianiello**

Homework 1

Task:

Write a program *predict(TS,QS): OUT ;*

TS,QS, and OUT are text files where

1. TS contains lines of n+1 integers separated by commas
2. QS contains lines of n integers and a "?" in the last position
3. OUT contains lines of n+1 integers

The idea is that you make up a prediction/learning algorithm that decides what integer to choose to substitute for the "?" for each sample in the QS, given the feature values in the sample and the information in the TS.

Idea learning process (algorithm)

For this homework as a learning algorithm I've chosen **least squares method**. This approach is a standard method in regression analysis and could be applied to current task as well.

Let $(x_{i1}, \dots, x_{ik}, y_i) \in R^{k+1}, i = 1, \dots n$ – training set (TS) of n samples with power of sample space equal to k + 1. And for each sample $x_i = (x_{i1}, \dots, x_{ik})$ we have target $y_i \in R$. Consider function

$$f = f(x) = a_0 + a_1 x_1 + a_2 {x_2}^2 + \cdots + a_k {x_k}^k, x = (x_1, \dots, x_k)$$

as a way to finding our target y_i for given x_i:

$$y_i = f(x_i) = a_0 + a_1 x_{i1} + a_2 {x_{i2}}^2 + \cdots + a_k {x_{ik}}^k$$

So, our learning process is equal to finding coefficients $a_j,\ j = 0, \dots, k$ such that

$$S(a_0, \dots, a_k) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \to min$$

In other words

$$\begin{cases} \frac{dS}{da_0} = 0 \\ \frac{dS}{da_1} = 0 \\ \dots \\ \frac{dS}{da_k} = 0 \end{cases}$$

Let compute partial derivatives of S in respect to $a_j, j = 0, \dots k$

$$\begin{cases} \frac{dS}{da_0} = \sum_{i=1}^{n} 2\left(a_0 + \left(\sum_{j=1}^{k} a_j {x_{ij}}^j\right) - y_i\right) = 0 \\ \frac{dS}{da_1} = \sum_{i=1}^{n} 2x_{i1}\left(a_0 + \left(\sum_{j=1}^{k} a_j {x_{ij}}^j\right) - y_i\right) = 0 \\ \dots \\ \frac{dS}{da_k} = \sum_{i=1}^{n} 2{x_{ik}}^k\left(a_0 + \left(\sum_{j=1}^{k} a_j {x_{ij}}^j\right) - y_i\right) = 0 \end{cases}$$

By solving the system of equation above by using Gaussian elimination method, we will get coefficients $a_j,\ j = 0, \dots, k.$

The error function that was used:

$$E = \sum_{i=1}^{n} |f(x_i) - y_i|$$

Implementation details

For developing this learning algorithm I used C# programming language. File with source called Program.cs and it is in a folder called MachineLearning_T1. Also in the root of *.zip you will find:

- TS.txt – contains training set (300 samples)
- QS.txt – contains queries
- OUT.txt – contains output of learning process by given queries.

Total error for given samples $E = 97$; average error: $\frac{E}{|TS|} = \frac{97}{300} \approx 0.32$