



US Census dataset

YONGWE JEAN-LUC

January 12, 2017

Contents

Introduction

Basically humans live in a three dimensional world , that implies that they can understand data in one-dimension, two-dimension but also in three-dimension. Sometimes it requires to think beyond that n-dimensional world and the human brain can get easily overflowed , so it is preferable to use machine learning for it to become easily to recognize pattern in data that is not obvious at first sight.

US census dataset is a dataset assemble to be able to " drawn" the profil of people that earns more than 50000 dollars/year or less troughout differents variables (42 exactly) such as AGE , level of Education , SEX , Employment situation , Marital situation , Skin tone....Yearly earnings.

The choosen technology to analyse our dataset was Python because it is very versatile and modular langage that allows the user to either simply manage and explore his dataset (preliminaries statistics and graphical vizualisation) and even pull off some more exotics insights by using advanced statistics.

Approach

The approach to the problem was made steps by steps which are the following:

STEP 1

The target variable of our dataset is a two modalities-variable coded as " - 50000." and "50000 + ." , that we are going to recode into 1 if the person yearly earnings are above 50000 dollars and 0 if the person yearly earnings are under 50000 dollars.

STEP 2

The dataset is coming into two parts one where we are going extract any kind of pattern present and a second one to check the credibility of it.

In order not to let our analysis to be biaised by values that may be missing , the idea here is to clean the documents but first we need to know how much data are missing..

At first we recode the original dataset opened up as a csv file to code missing values as *NaN* .
To count the number of *NaN* elements in our dataset we need to count the frequency of every modalities in every variables , once we find the frequencies of *NaN* element we make a simple addition of all of them and divide it by the dimensions of the dataset (number of rows X number of columns).

Second we run a simple basic statistic univariate description:(see the figure below)

```
##### % STATISTIC AND UNIVARIATE AUDIT #####
% missing values: 3.68 %
=====
statistical description of the quantitative variables :
      index      var1      var3      var4 \
count 199523.000000 199523.000000 199523.000000 199523.000000
mean  99761.000000  34.494199   15.352320   11.306556
std   57597.473217  22.310895   18.067129   14.454204
min    0.000000    0.000000    0.000000    0.000000
25%   49880.500000  15.000000    0.000000    0.000000
50%   99761.000000  33.000000    0.000000    0.000000
75%  149641.500000  50.000000   33.000000   26.000000
max   199522.000000  90.000000   51.000000   46.000000

      var6      var17      var18      var19 \
count 199523.000000 199523.000000 199523.000000 199523.000000
mean   55.426908   434.71899   37.313788   197.529533
std   274.896454  4697.53128  271.896428  1984.163658
min    0.000000    0.000000    0.000000    0.000000
25%    0.000000    0.000000    0.000000    0.000000
50%    0.000000    0.000000    0.000000    0.000000
75%    0.000000    0.000000    0.000000    0.000000
max   9999.000000  99999.00000  4608.000000  99999.000000

      var25      var31      var37      var39 \
count 199523.000000 199523.000000 199523.000000 199523.000000
mean  1740.380269   1.956180   0.175438   1.514833
std   993.768156   2.365126   0.553694   0.851473
min   37.870000    0.000000    0.000000    0.000000
25%  1061.615000    0.000000    0.000000    2.000000
50%  1618.310000    1.000000    0.000000    2.000000
75%  2188.610000    4.000000    0.000000    2.000000
max  18656.300000    6.000000    2.000000    2.000000

      var40      var41      target
count 199523.000000 199523.000000 199523.000000
mean   23.174897   94.499672   0.062058
std   24.411488    0.500001   0.241261
min    0.000000   94.000000   0.000000
25%    0.000000   94.000000   0.000000
50%    8.000000   94.000000   0.000000
75%   52.000000   95.000000   0.000000
max   52.000000   95.000000   1.000000
```

STEP 3

Only 4percent of the data are missing , the next step is to erase those data in order to have a more managable dataset.

A closer look shows that 8 columns (variables) are concerned , in a sense that they some *NaN* elements. The dimensions of the dataset are 199523 rows and 42 colmun which give us 8379966 elements.

There are 8 variables concerned by the missing data meaning 1596184 elements on that grid , a quick calculations shows , if we go about the columns this is 20percent of the data that will be passed on , so the decision is to go by the rows , where we will clearly loss less data .

STEP 4

Dataset Analysis

The dataset is containing 42 variables , according to the statistical description figure only 15 of them are numericals or quantitave type (if you will) , it would be intersting to include the qualitative variables in our analysis.

To do so we have to recode each one of the modalities of the variables using the features extraction to extract and distinct all the modalities and OneHotEncoder to recode the modalities into vector of 0 and 1. Unfortunately in doing so we realized :

```
#####RECODING MODALITIES DATASET#####
recoding training file : 396 numbers of modalities
recoding test file : 395 number of modalities
```

Not being able to find what went wrong in the process , we decide to report our analysis on solely quantitative data.

We have two files to our disposition one to learn from and another one to validate our discoveries. We decide to compare two clustering algorithm (Decision Tree and Neural Network) and one regression algorithm (Logistic Regression) on the learning file and finally to check both of them with the learning pattern extract by each one of them on the "testing file" and the results are the following :

```
#####DECISION TREE#####
=====MODEL TRAINING=====
matrice de confusion : [[143227  6514]
 [ 5784  4093]]
precision ratio : 0.922953551604
error ratio : 0.0770464483955
=====MODEL TESTING=====
matrice de confusion [[87386  6198]
 [ 5776  402]]
taux de succes : 0.879974338927
taux d erreur : 0.120025661073
#####NEURAL NETWORK#####
=====MODEL TRAINING=====
matrice de confusion : [[149736      5]
 [ 9876      1]]
precision ratio : 0.93809595409
error ratio : 0.0619040459096
=====MODEL TESTING=====
matrice de confusion : [[93577      7]
 [ 6178      0]]
precision ratio : 0.938002445821
error ratio : 0.0619975541789
#####LOGISTIC MODEL#####
=====MODEL TRAINING=====
confusion matrix : [[149043  698]
 [ 7865  2012]]
precision ratio : 0.946353168189
error ratio : 0.0536468318109
=====MODEL TESTING=====
matrice de confusion : [[92039  1545]
 [ 6083   95]]
precision ratio : 0.923538020489
error ratio : 0.0764619795112
```

Looking back at the results we can see that the Neural Network is more stable and accurate in term of clustering and the Logistic model is the right one in term of prediction on the learning but a little less accurate than the Neural Network. The decision would be to go with the Neural Network more precise and stable on the long run.

A quick look to the correlation matrix and focusing on the target column we can see than the most correlate variable are the 41st , the 17th and the 31st wich is the 'number of week worked in a year' (var41) , 'capital gain in investment' (var17) and 'number of people working in the household' (var31)

```
#####VARIABLE CORRELATION#####
correlation to target :
index      0.002539
var1       0.135720
var3       0.196190
var4       0.013414
var6       0.024528
var17      0.240725
var18      0.147417
var19      0.175779
var25      0.014463
var31      0.222684
var37      0.040473
var39      0.140930
var40      0.262316
var41      0.014794
target     1.000000
Name: target, dtype: float64
```

From there when can say that the more weeks people worked , the more people having a job in their household and also the capital gain , the more they are incline to have more than 50000 dollars savings yearly.