Regression on US survey

14 settembre 2020

Andrea Adami

Mariacristina Cattani

We declare that this material, which we now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.

Indice

1 Introduction

This work aims to create a Regression model to predict the total personal income (PINCP) target variable. We want to check whether it is possible to inference on the personal income of the 3.5 million American families who participated to the 2013 survey, certainly not considering the variable PERNP (Universe for Total Person's Earnings) which is directly and strongly correlated with the target variable. The project analyses the use and performance of Regression techniques, such as GBT (Friedman 1999) and Linear regression, in order to compare the results obtained from the models. The work is divided as follows: in the first part the database in question will be presented, then the literature underlying the applied models will be explained, in the third part the methodology and procedure applied in the analysis will be explained. Fourth part reports the actual experiment, concluding the work with the discussion of the results obtained.

2 Dataset

The dataset has been taken from Kaggle (www.kaggle.com/census/2013-american-community-survey). From this link it is possible to download two paths of study: one concerning the population as such and the other one concerning housing. To facilitate data download, both lines of study are divided into two subsections each containing half of the US states; therefore, in section "a" contains the first 25 federal states and "b" the remaining. For the execution of this project, it was decided to take into consideration only the two files concerning the population, where here each row represents a person, and 283 variables concerning age, gender, work situation and everything about a the personal and professional situation are analysed. One of this PWGTP, is a weight associated to each person, because individuals are not sampled with equal probably in this annual survey.

3 Literature review

Linear regression is a technique to identify the linear relationship between independent variables and dependent variables, with one independent variable is:

$$y = \beta_0 x_0 + \beta_1 x_1 + \ ... \ + \beta_m x_m$$

The aim is to use the regression to estimate the functional relation between the variables. The regression coefficient β_i represents the change in response per unit change of the corresponding regressor x_i when the other independent variables are held constant. Finding the residua e_i means calculate $e_i = \hat{y}_i - y_i$, difference between the predicted and true value. A common way to estimate is the residual sum of squares

$$\sum_{i=1}^{n} {e_i}^2$$

We then look for the best regression model capable of minimizing the previous function. There are also more complex regression models where regularization parameters are added to ensure that the model created does not give low errors on the train and high on the test.

3.1 GBM Trees

Gradient-Boosted Trees (GBTs) are ensembles of decision trees. GBTs iteratively train decision trees in order to minimize a loss function. On each iteration, the algorithm uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. The dataset is re-labeled to put more emphasis on training instances with poor predictions. [1] Thus, in the next iteration, the decision tree will help correct for previous mistakes. For doing that it is used a Loss function, usually a L_1 or L_2.

The model is customisable, changing the following parameters:

- *Loss-Function*: is an important parameter, different loss-functions bring to different results;
- *numIterations*: is the total number of trees the model will fit;
- *learningRate*: a coefficient multiplied by consequent trees in order to better fit data.

The Boosting Process can overfit very fast during the training procedure, so a Validation while training is needed, in order to find the optimal stopping point for the number of iteration. The consecutive fitting can be done on the whole training data or on a subset of it, the stochastic selection of the set give a stochastic behaviour of the Gradient descent. [2]

3.2 Linear Regression

Considering a data train X with n examples, the vectors $x_i \in \mathbb{R}^d$ are the training data, for $i \in \{1, \ldots, n\}$ and $y_i \in \mathbb{R}$ the corresponding labels to be predicted. It is possible to solve a convex optimization problem, i.e. finding the minimum of function f that depends on a variable vector w, with d entries:

$$\min_{w \in \mathbb{R}^d} (w)$$

where the objective function is:

$$f\,(w) = \lambda\ R\,(w) + \tfrac{1}{n}\ L(w;\ x_i,\ y_i)$$

The objective function f has two parts: the Regularizer that controls the complexity of the model, and the loss that measures the error of the model on the training data. The fixed regularization parameter $\lambda \geq 0$ defines the trade-off between minimizing the loss and avoiding overfitting. In case of Ridge regression L2 regularization is used otherwise in case of Lasso L1

$$L2 = \parallel w \parallel_2 \text{ or } L1 = \parallel w \parallel_1$$

With average loss or training error known as the mean squared error

$$\tfrac{1}{n} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2$$

4 Methodology

The Experiment starts with the configuration of the Spark (3.0.1) environment in the Google Colaboratory virtual machine. The database located in Kaggle, is then downloaded using the built-in API of Kaggle, using the credentials of the profile. Once obtained the dataset in the VM it has been loaded using the spark.read.csv function, also using the inferschema to guess the data type of the column values, by the way the column dtype have been transformed into Int type, in order to require less space and be faster in the computation. After the creation of the Dataframe it has been done a proper adjustment of the columns, dropping PERNP and using VectorAssembler for providing a new schema with a single column for features and one for target.

4.1 Preprocessing

The pre processing have been done in three steps. First null entries are converted in zeroes, in order to have zeroes entering in the models. Such that multiplication with coefficients does not give any problem, then Dimensionality reduction is applied, using PCA, at the end both features and target are scaled.

The PCA has been performed due to the fact that in the original dataset there are a really high number of variable, lot of which without an appreciable correlation with the target variable. So we decided to keep only 20 features. This number have been chosen after a single shot of correlation of variables with the target, and counting the variables with a correlation higher than a threshold. PCA aims to project the values onto a new subspace with dimensions equal to or smaller than the original one. It is a linear transformation technique used for the extraction of the characteristics of a multivariate set, obtaining a set of main variables, while maintaining the maximum variation of information. The eigenvectors are calculated from the variance-covariance matrix; that is, the main components orthogonal to each other.

MaxAbsScaler is used on feature and an "hand made scaling" is performed on target. Feature scaling is essential for machine learning algorithms first to make all variable with different unit comparable, second to better calculate distances between data and last because in the usage of some algorithms, those have a faster convergence. Standardization transforms the data to have zero mean and a variance of 1.

4.2 model definition

For inferring the PINCP variable 2 different models have been fitted, The first one is a linear model. The function used is LinearRegression. The fitting of this model has been done through the help of a grid, in which all the combination of parameters have been settled in order to find the best model to be used. The choose of the model has been performed through a train-validation procedure with $p = 0.8$. The model was later tested on R^2 and RMSE, both on train and test set, willing to check if an overfitting occurred or not.

The second model fitted is a GBTRegressor, implemented using the J.H. Friedman. "Stochastic Gradient Boosting." (1999) algorithm which produces a predictive model as a set of "weak" learner, trees, for fitting the residuals at each boosting iteration. As previously done the model have been fitted through the use of a grid of parameters and a train-validation procedure. Later evaluated on train and test with the same metrics.

4.3 model fitting and metrices

The RMSE is the square root of the variance of the residuals. It calculates the distance from the estimated to the observed data points, so indicates the absolute fit of the model to the data. Low values indicate a better fit, therefore a good predictor. It must also be measured in relation to the type of model used and the number of data points.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

R-squared is a relative measure of adaptation. It is interpreted as how good the regression model fits the observed data, in term of the percentage of variance explained.

5 Results

For the construction of the first model a linear regression was used, modelled using cross validation with validation at 20% in order to predict the PINCP target variable, the scaled features selected following the application of the PCA were used. The results obtained are similar, both on the train and on the test set, not leading to overfitting, resulting in $rmse = 0.00804$ and $R^2 = 95\%$. This result can be seen as a good thing (not to overfit over such many features), but, tells us that a more sophisticated model may be build, giving the model more freedom to improve complexity.

The fitting of GBTRegression given us an $R^2 = 90\%$ both on train and test set, using the most complex model within the possible combination provided. This again suggest the fact that a more complex model may be fitted. Increasing both the mìnumber of boosting iteration and the complexity of the trees.

To calculate the complexity of the Linear model, this was approximated to

$$\mathcal{O}(d^3 + nd^2)$$

The number of train samples can be distributed by decreasing the complexity, but the time complexity can at most be approximated to

$$\mathcal{O}(d^3 + d^2) = \mathcal{O}(d^2(d+1)) \approx \mathcal{O}(d^3)$$

A possible solution is to drastically decrease the number of features, using PCA, in this way the computational time obtained would be of circa one order of magnitude smaller every half of p.

$$\mathcal{O}((\tfrac{d}{2})^3) = \mathcal{O}(\tfrac{d^3}{2^3}) \ = \tfrac{1}{8}\mathcal{O}(d^3) \approx \tfrac{1}{10}\mathcal{O}(d^3)$$

Regarding the computational complexity of GBTRegression, this was calculated taking into account the number of train samples, the number of features and trees. Therefore, the number of train samples can be splitted in different workers, thus duplicating the number of these, the complexity is halved for the same features and tree.

All the work have been coded using the well known libraries of spark, with built-in function in order to provide an easy and automatic way of distribution of Data storage and computation.

Riferimenti bibliografici

[1] Apache Spark 3.0.1 release documentation
https://spark.apache.org/docs/latest/mllib-ensembles.htmlgradient-boosted-trees-gbts

[2] Stochastic Gradient Boosting by Jerome H.Friedman
https://statweb.stanford.edu/jhf/ftp/stobst.pdf