Irving Fisher Committee on
Central Bank Statistics

BANK FOR INTERNATIONAL SETTLEMENTS

Ninth IFC Conference on "Are post-crisis statistical initiatives completed?"

Basel, 30-31 August 2018

A machine learning approach to outlier detection and imputation of missing data[1]

Nicola Benatti,
European Central Bank

[1] This paper was prepared for the meeting. The views expressed are those of the authors and do not necessarily reflect the views of the BIS, the IFC or the central banks and other institutions represented at the meeting.

A machine learning approach to outlier detection and imputation of missing data

Nicola Benatti

In the era of ready-to-go analysis of high-dimensional datasets, data quality is essential for economists to guarantee robust results. Traditional techniques for outlier detection tend to exclude the tails of distributions and ignore the data generation processes of specific datasets. At the same time, multiple imputation of missing values is traditionally an iterative process based on linear estimations, implying the use of simplified data generation models. In this paper I propose the use of common machine learning algorithms (i.e. boosted trees, cross validation and cluster analysis) to determine the data generation models of a firm-level dataset in order to detect outliers and impute missing values.

Keywords: machine learning, outlier detection, imputation, firm data

JEL classification: C81, C55, C53, D22

Contents

Introduction

In a world made of data, different outlier detection techniques have been applied to different types of data. From the most standard techniques such as the deletion of observations in the tails of the distributions, to more tailor made variable-specific threshold-setting methods, data analysts always find difficulties in identifying a general framework for outlier detection which can be easily applied to a whole dataset without compromising its integrity. Such analysis tended to become even more difficult in the recent years with the increase in availability of very large datasets. These contain large numbers of variables for which it becomes tedious to set variable-specific thresholds while their distributions are linked to each other by construction.

An outlier is generally considered as an observation which is significantly distant from the other considered observations. Since in a dataset variables are often partially related with each other, we can consider an outlier a data entry which is lying far from the others on a n-dimensional space, where n is the number of variables in the dataset.

It is important to clarify that an outlier is not necessarily a mistake. A simple example could be a sport champion who excels in her discipline. She is an outlier and she is definitely not a mistake. All we can do then, is to rank observations by their likelihood of being anomalies and start analysing, knowing both which are their expected values and how the reported values have been generated. If necessary we have to look for information which is not contained in our dataset in order to confirm or reject an observation.

The literature tends to distinguish between three dimensions according to which to define an observation as outlying.

The first one is the distribution of a specific variable under observation. In economics literature in particular, data cleaning processes often consist in dropping the lowest and highest percentiles of each considered variable, leading in this way to the deletion of a fixed percentage of information, part of which may not only be correct, but also might have major economic meaning.

A less drastic approach is the one of eliminating observations which exceed a certain number of standard deviations from the mean, assuming in this way that the data is following a symmetric distribution. Since this is often not the case, the mean can also be substituted by the median to account for the skewness of the distribution.

Single variable distribution-based techniques are quite simple to implement but will only allow spotting extremely high or extremely low observations, excluding the potential outliers within the distribution.

For this reason distance based techniques started being applied to large datasets, such as k-nearest neighbourhood which clusters observations into groups and flags as outlying observations which lay the furthest away from any recognised group. These techniques can be completely data driven (unsupervised machine learning) but, by themselves, do not exploit the fact that having available the whole dataset we want to find outliers in, this machine learning algorithm can be implemented instead in a supervised fashion and provide information on the data generating model behind it.

There is then another way of detecting outliers which I am going to present in this paper. The idea of this technique is, like in cluster analysis, to look at joint distributions but, additionally, to exploit the information available in a supervised way and estimate the data generation model of a specific variable. The model will then provide the data analyst with a set of suggested features (i.e. which features are driving the determination process of a specific variable), their importance, the fitted values for the estimated variables (i.e. to be used for data imputation) and ranks the observations by their likelihood of being mistakes. Once the data analyst has these pieces of information, she can use them to prioritise the analysis of potential errors.

This framework combines the supervised and unsupervised frameworks under a probabilistic model and reduces assumptions on the data structure to its minimum allowing also for the estimation of the expected values which can be furthermore used to impute missing observations.

Construction of the dataset and data description

In order to show an application of the technique described in this paper, I use the iBACH dataset of non-financial corporations' balance sheet data. The dataset has been collected on a stable form since December 2017 by the Working Group Bank for the Accounts of Companies Harmonized (WG BACH), under the mandate of the European Committee of Central Balance Sheet Data Offices (ECCBSO).

The Working Group BACH is active since 1987 in the collection and analysis of aggregated balance sheet data of non-financial corporations and its members are the Central Balance Sheet Data Offices of European countries. While the collection, the analysis and the dissemination of these data has always been carried on at an aggregate or meso level (i.e. information has always been aggregated by country, industry, year, size class and quartile), in June 2017 the working group BACH received the mandate to create a task-force for the collection of individual company data (iBACH) that can be shared among researchers of the participating institutions.

Since February 2018 this dataset was disseminated by the European Central Bank (ECB), which functions as a hub for the data collection, to its internal users and to the WG BACH representative members who are in charge of disseminating and promoting the dataset in their countries. The iBACH dataset contains yearly balance sheet, financial and demographic variables of non-financial corporations. It currently covers six EU countries (Belgium, Spain, France, Italy, Portugal and Slovakia).

It has to be mentioned that this type of data surely helps my analysis since, by construction, data generation models (i.e. joint distributions) of several balance-sheet variables are the same (or very similar) across countries and years; which means that accounting rules generally hold.

It is also important to mention that, both before and after reaching the ECB, the quality of these data is checked extensively by applying a set of accounting validation rules which highlight a first set of reported mistakes. The data is then corrected and resubmitted to the ECB iteratively until the number of spotted accounting mistakes is minimised.

The dataset contains the following 66 variables:

Demographics:	**Income statement:**
Address	Operating revenue
Address 2	Financial expenses
Firm's status of activity	Net turnover
City	Financial income
Consolidation code	Extraordinary income
Country	Costs of goods sold
Identification number	Labour costs
Legal form	Depreciation and amortization
Name	Interests on financial debt
Legal form in the national codification	Tax on profit
Number of employees	EBIT, net operating profit
Number of months for the account exercise	EBITDA
Localisation information	Profit and loss
Sector of activity (NACE code)	Profit and loss before taxation
Year	Cash flow
Year of incorportation	Value added
Year of liquidation of the firm	
Zipcode	
Liabilities:	**Assets:**
Shareholders funds	Total assets
Capital, reserves, earnings and other equity instruments	Total current assets
Total liabilities	Total fixed assets
Short-term debt	Intangible fixed assets
Other current liabilities	Tangible fixed assets
Total current liabilities	Financial fixed assets
Long-term debt	Total inventories
Total non-current liabilities	Trade receivables
Short-term debt owed to credit institutions	Other receivables
Long-term debt owed to credit institutions	Other current receivables
Trade credit	Deferred assets
Payments received on account of orders, current	Deferred current assets
Deferred liabilities, current	Other financial assets, current
Deferred liabilities, non current	Cash and bank
Provisions	
Investment accounts:	
Acquisition less sales and disposals of intangible assets	
Acquisition less sales and disposals of tangible fixed assets	
Acquisition less sales and disposals of financial fixed assets	

The panel structure of the dataset is as follow:

Number of firms reporting

	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
BE		204,625	218,707	233,180	250,392	264,474	284,327	297,899	326,480	344,480	362,762	377,386	382,669	349,034
ES							450,538	447,540	459,076	450,146	443,527	583,081	560,570	324,701
FR	184,812	192,206	198,107	208,534	225,408	233,267	233,865	244,843	260,670	260,565	250,048	253,758	257,950	261,051
IT	492,472	517,464	540,517	560,140	582,993	600,656	613,021	624,235	634,278	629,865	627,317	621,722	618,177	464,353
PT	16,920	17,547	15,176	342,588	357,480	367,237	366,806	365,821	373,230	373,500	378,731	382,779	390,730	392,030
SK												99,389	99,584	97,869

In spite of the data quality checks mentioned above, several data quality issues remain, both in terms of non-plausible values reported and values which are not reported at all.

For these reasons I follow the most recent literature on data science techniques for estimation in order to determine at first the data-generation model and its features for each numeric variable reported, later the expected value of each observation in the dataset and finally which observations have are most likely to be errors.

Given the high computational power needed to run the algorithms described in the next chapter, the original dataset was filtered to select a random sample of 10% of the overall observations. This subsample consists of 2,345,338 observations on which the proposed model is trained and subsequently tested.

Since not all countries report demographic variables and not all these would add information but rather slow down the process, I removed part of the demographic columns from the analysis. On the other hand, I define the set of variables that have to be considered as categorical (e.g. sector, country and legal form).

Last but not least, in order to train and test the models that I am estimating for each variable, I create a training sample picking up randomly 2/3 of the observations. The remaining 1/3 of observations is kept as testing sample. In this way I will be able to calibrate the algorithm on 2/3 of the data and test the results on the remaining 1/3.

Empirical strategy

While in the pages above I describe the general principles that I apply in my methodology, it is now time to go more in details on the various steps of the detection process.

In the paragraphs below I will first briefly described how I use well-known algorithms in a combination and with a scope not yet covered by the literature. I will

first outline the XGBoost algorithm by T Chen, C Guestrin (2016) used to determine the data-generation model, producing the sets of features and their importance for each estimated variable and the fitted values of the observations. I will then use the GridSearch algorithm to select the models' hyper-parameters that minimize my objective function. Following I propose using the DBSCAN algorithm (density based clustering algorithm) for flagging outlying observations. Last but not least I introduce the concept of feature additive ranking (FAR) to determine the reported cells which are most likely to be reported wrongly.

Extreme Gradient Boosting

In order to estimate the data-generating model, the importance of its features and to predict the response variable y_i given the set of explanatory variables $\boldsymbol{x} = \{x_1, \dots, x_n\}$, I use the Extreme Gradient Boosting (XGBoost) technique developed by Chen and Guestrin.

The objective of the algorithm is, given the training data, to find the best parameters to minimize the expected value of an objective function $F(\theta)$ which is composed by a training loss $L(\theta)$ and a regularization part $\Omega(\theta)$:

$$F(\theta) = L(\theta) + \Omega(\theta)$$

Xgboost is based on tree-ensembles technique. The concept behind it is that y_i is classified into different leaves of a tree given the values of $\boldsymbol{x}_i$. These trees are weak learners and normally have a very few splits. Since a single tree would not be strong enough to properly estimate y_i, the prediction of different trees are summed up together into a tree ensemble model.

The authors of XGBoost developed it as an additive learning algorithm which means that one tree is added at every iteration of the algorithm, the principle being that each new tree predicts the residuals of errors of the (sum of) the previous trees:

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

At this stage, the algorithm adds the best tree available at each iteration, but in order to improve the performance, the authors of XGBoost added a regularization term which penalises complexity.

Moreover, Chen and Guestrin prune trees so that a tree will not be added to the model if the gain is smaller than gamma, preventing overfitting.

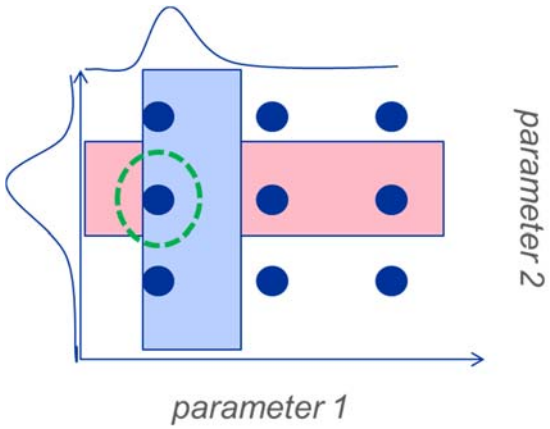
Last but not least, once the algorithm has successfully run on the data, it provides the contribution (importance) of independent variables on the determination of the dependent variable, which turns out being extremely useful for the user to understand the data generating model.

Xgboost is available as a ready-to-run Python package (xgboost) which includes a wide set of hyper-parameters including the type of model that one wants to specify (classification, linear, logit, poisson), the subsample on which to train the model, the number of estimators, the maximum depth of a tree and many others.

More information regarding the methodology and the package can be found on the Xgboost website (http://xgboost.readthedocs.io/en/latest/tutorials/model.html).

Hyperparameter optimisation

One of the usual criticisms to machine learning techniques is that the researcher biases the algorithm by imputing his personal judgment when choosing its hyperparameters. Recent literature investigates the possibility of setting the hyperparameters of a machine learning algorithm by testing on the data which one is the optimal set of hyperparameters for the model.



Gridsearch, the algorithm available within the python scikit-learn python package, exhaustively searches through a subset of specified hyperparameters. The algorithm trains the machine learning algorithm with each set of possible combination (Cartesian product) of hyperparameters and evaluates the performance on the testing set by picking the tuple that goes in the estimation model with the highest explanatory power.

The hyperparameters of XGBoost estimated by the Gridserach algorithm here are:

- Max_depth: maximum depth of a tree, increase this value will make the model more complex / likely to be overfitting.
- Eta: step size shrinkage used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features and eta actually shrinks the feature weights to make the boosting process more conservative.

- Subsample: subsample ratio of the training instance. Setting it to 0.5 means that XGBoost randomly collected half of the data instances to grow trees and this will prevent overfitting.
- N_estimators: Number of boosted trees to fit.

Two main issues need to be raised here. The first is that XGBoost presents a much wider set of hyperparameters which I set as default due to the fact it would be too computationally intense to specify them all. The second issue is that Gridsearch is looking for the best solution among the combinations of parameters in the portfolio I chose based on common practice but it is not sure that the minimisation process yields a global minimum. A better but more computational demanding approach would be to use a random search algorithm or a Bayesan search algorithm, increasing however significantly the time to run the process.

The medians of the parameters chosen by Gridsearch for all variables estimated are the following:

max_depth=5,

learning_rate=0.1,

subsample=0.7,

n_estimators=1500

Flagging outlying observations

In the two paragraphs above I described two processes relatively easy to find in the literature, even in combination with each other. What instead was not widely investigated yet is how by using XGBoost and Gridsearch algorithms a data analyst can detect outliers.

Having estimated the data generating model behind each y_i of our set, I estimate the fitted values for each observation and compute the estimation residuals.

At this stage I apply a specific cluster analysis algorithm called Density-based spatial clustering of applications with noise (DBSCAN by Ester, Kriegel, Sander and Xu 1996) on the vectors of residuals and relative residuals (i.e. the residuals as percentage of the reported value) of each model and flag as outlier the observations which are the most distant from their nearby neighbours.

This award-winning algorithm has the great advantage of isolating on the set of absolute and relative residuals those in a low density area. Compared to KNN techniques, in particular, it allows not to set the number of clusters to find and to flag as outliers the datapoints that lay in low-density areas , while the parameters to be specified are the physical distance ε (which is variable-specific), and minPts which is the desired minimum cluster size which I set to 3.

Feature-additive ranking

While the technique above is widely used to spot outliers in distributions, I now apply it to the residuals of a regression problem. In this way I will be able to spot observations which do not fit into my data generation model. However, what might happen is that a value spotted as outlier is not an actual reporting error while one of its determining features is instead reported erroneously (i.e. one of the independent variables that feed the model). In other words, when spotting an outlying observation, we cannot assume that it is an actual mistake until we check if any of the other variables that generate it are wrongly reported or not.

For example, if the identity A=B+C+D identified by the XGBoost algorithm is not respected, it could be because of a mistake in reporting any of the variables among A, B, C or D.

In order improve the ranking of potential errors in the data, I introduce the feature-additive ranking technique (FAR) which sums up the contribution of any variable by unit of observation (in my case entity-year) each time (in each model) this is flagged as an outlier.

$$FAW_{v,j,y} = \sum_{k=0}^{n} f_{j,y,k} * i_{v,j,y,k} * w_k$$

where:

v=variable

j=firm

y=year

k=outlier model

FAW=feature additive weight

f=outlier flag

i=importance

w=model weight

Using the FAW score, I then rank the information per unit of observation by the highest sum of FAW scores, highlighting the cell (variable-firm-year) supposed to be the most important in determining an outlying observation. This technique allows data analysts to prioritise their investigation on the "most-likely-to-be wrong" variable, cutting drastically investigation times.

Results

Having such a large dataset of balance sheet data and such computationally expensive algorithms, the procedures to flag outliers for each single variable runs

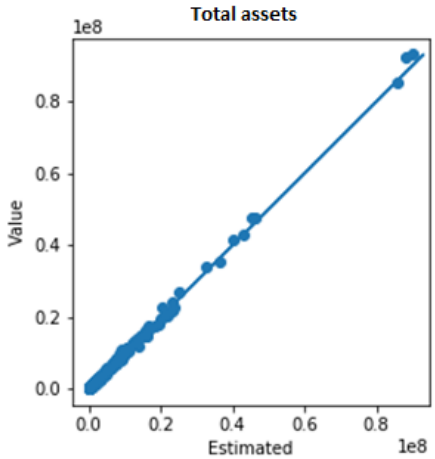
on a standard laptop for more than a week before providing the final calibrated models.

For each variable analysed we now know the importance of each other variable (feature) in its determination and we store the calibrated data generation model to be used for further estimations of fitted values.

The overall performance of XGboost is extremely high and the median R-squared when fitting the calibrated models on the test dataset has a median of 0.8757. Given such a high performance, the system can be used for data imputation when missing values are reported.

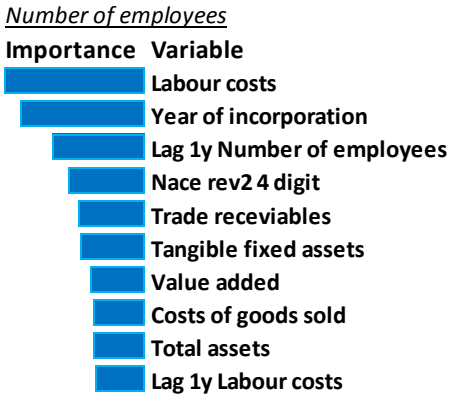
For example, the importances of the first eleven variables estimating total assets of firms rank as follows:

When looking at the results of the estimation of total assets on the test sample, which, just to remind the reader is not the dataset on which the algorithm was calibrated, the algorithm predicts perfectly each reported value and the observations all lay on the bisector:

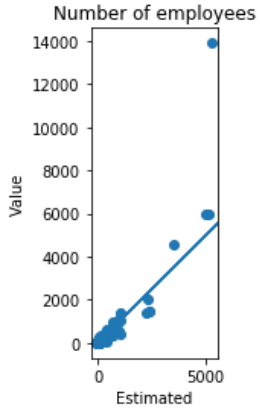


In the case of total assets, the variable is usually extensively checked by a set of validation rules before the data arrives at the ECB which make sure no observation is indeed reported erroneously.

When estimating of the number of employees, instead, the ten most important variables are the following:



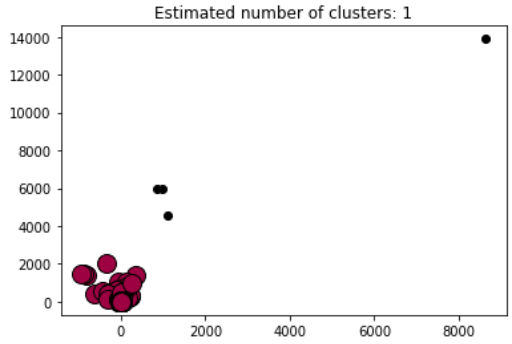
In this case it is easy to spot that certain actual values are far from the bisector line, given that the number of employees is not a balance sheet variable and does not have validation rules run a-priori on the data.



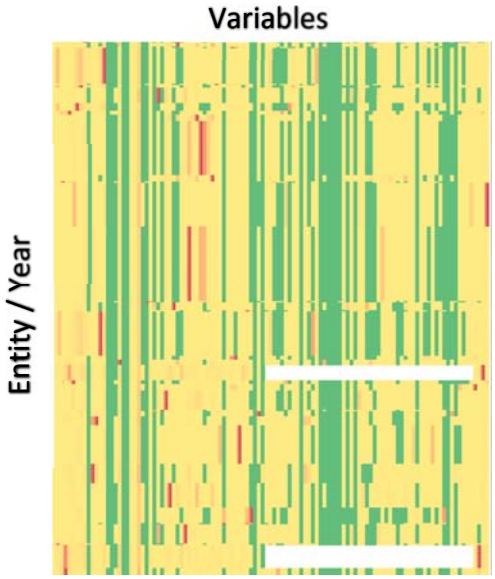
At this stage, the residuals of each variable estimation model are taken separately and analysed using the DBSCAN algorithm in order to flag anomalies in how the dataset fits the model.

In the figure below the black dots represent observations which both in absolute and relative terms lay distant from the value suggested by the data generation model. The red dots instead, represent observations whose residuals from the model are similar with each other and, given the accuracy of XGboost in prediction, are very close to zero.

Number of employees flagged observations



As explained by the methodology above, we do not stop at here. In fact, it could be that the number of employees reported by the firm x is flagged as an outlier but when looking at the time series of the entity, the number reported seems to be perfectly plausible. In fact, it could still be that while the number of employees seems plausible, the labour costs reported by the firm dropped drastically compared to the previous years. For this reason, we check if either of the variables is significantly important in the contribution to the estimation of any other outlying observation by using the FAR technique described above. Once the FAW scores are created, they are inserted in a heat-map which eases the work of the data analyst.



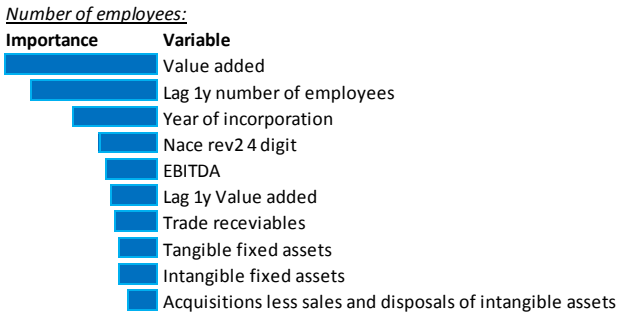
In this way the person analysing the data can not only focus on the outlying variables but also on the subcomponents that are most likely to affect that specific observation.

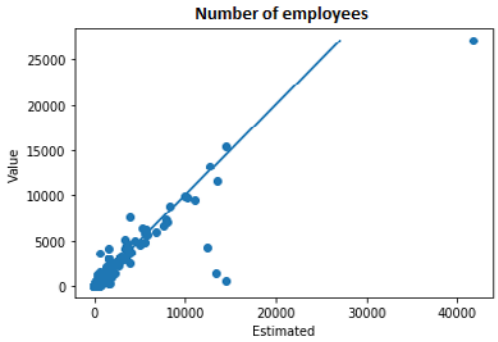
A great value added of this methodology, moreover, is that once the XGBoost algorithm is trained, the estimated model for each variable can be stored and does not need to be re-calibrated every time new data comes in. When new data comes into the system, it can then be evaluated in few seconds by using the rules established in the past.

This characteristic will boost the performance of those investigating these data.

A further empirical application: employment missing data imputation

In order to provide some more taste of what this procedure is capable of, I tested the XGboost model to be used for data imputation of the "number of employees" variable. The "number of employees" is a variable which is often not available in non-financial corporations' datasets because it is not part of the information which has to be reported in a balance-sheet. As we could see above, using the procedures proposed in this paper we can easily and very well estimate this information when the variable "Labour Costs" is available. However it can be that the whole information on employment (both number of employees and labour costs) is missing. We want then to test the estimation of the number of employees without using the information on labour costs. The results that we get are the following:





Although the estimation is not as precise as the estimation proposed when including the Labour cost variable, XGBoost seems to be an extremely adaptive method for estimating missing values, also when some main components are missing. In fact, this methodology improves further the performance of the methodologies previously used based on LASSO algorithm (Novello 2017).

Two general applications of this technique could be, as shown, the imputation missing values (or confirmed mistakes that were removed) but also the creation of a synthetic anonymised datasets for research purposes.

Conclusions

This paper presents an application of a combination of supervised and unsupervised machine learning with a final feature-additive ranking technique in order to spot mistakes in outlying datapoints. The model provides the data analysts with guidance on which variables to prioritise when controlling an outlying observation and which is their expected value given the data generation model identified by the algorithm. Moreover, the methodology described seems to be useful also for additional steps of data quality improvement such as data imputation.

This technique also provides guidance for the construction of new data quality checks that could prevent the submissions of mistakes.

Further improvements to this process are already in the pipeline. In particular, the increase in the sample size as soon as the full deployment of the big-data environment will be finalised at the ECB. Moreover, the comparison of the results obtained with those obtained by using neural networks and multi-target regressions which from the recent literature seem to be even more powerful estimation tools.

Last but not least, the part of the tool which is now based on cluster analysis could become a usupervised classification problem if we could get from the data owners confirmation on whether a spotted potential mistake is actually an error or not. If we could have that information available the problem we are analysing would fundamentally change and we could develop an algorithm which would learn how to classify mistakes instead of outliers.

Annex

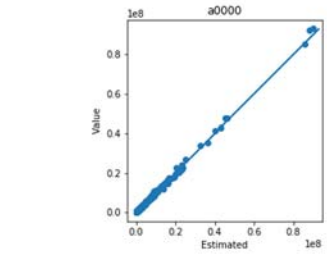
I – Description of variables

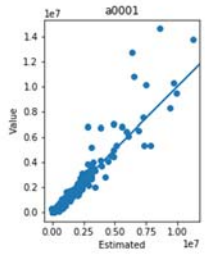
Variable:	Description:
A0000	Total assets
A0001	Total current assets
A1000	Total fixed assets
A1100	Intangible fixed assets
A1200	Tangible fixed assets
A1300	Financial fixed assets
A2000	Total inventories
A3000	Trade receivables
A4000	Other receivables
A4100	Other current receivables
A5000	Deferred assets
A5100	Deferred current assets
A6000	Other financial assets, current
A7000	Cash and bank
DADDRESS1	Address
DADDRESS2	Address 2
DCEASE	Firm's status of activity
DCITY	City
DCONSO	Consolidation code
DCOUNTRY	Country
DID	Identification number
DLEGAL	Legal form
DNAME	Name
DNLEGAL	Legal form in the national codification
DNUMBEREMPL	Number of employees
DNUMBERMTH	Number of months for the account exercise
DREGIO	Localisation information
DSECTOR	Sector of activity (NACE code)
DYEAR	Year
DYINCORP	Year of incorportation
DYLIQUID	Year of liquidation of the firm
DZIPCODE	Zipcode
E0000	Shareholders funds
E1000	Capital, reserves, earnings and other equity instruments
I0001	Operating revenue
I0002	Financial expenses
I0100	Net turnover
I0420	Financial income
I0430	Extraordinary income
I0500	Costs of goods sold

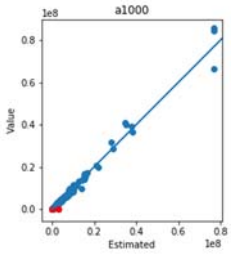
I0700	Labour costs
I0900	Depreciation and amortization
I1000	Interests on financial debt
I1100	Tax on profit
IT001	EBIT, net operating profit
IT002	EBITDA
IT003	Profit and loss
IT004	Profit and loss before taxation
IT005	Cash flow
IT006	Value added
L0000	Total liabilities
L0001	Short-term debt
L0002	Other current liabilities
L0003	Total current liabilities
L0004	Long-term debt
L0006	Total non-current liabilities
L2100	Short-term debt owed to credit institutions
L2200	Long-term debt owed to credit institutions
L4000	Trade credit
L5000	Payments received on account of orders, current
L6100	Deferred liabilities, current
L6200	Deferred liabilities, non current
LP000	Provisions
N1000	Acquisition less sales and disposals of intangible assets
N2000	Acquisition less sales and disposals of tangible fixed assets
N3000	Acquisition less sales and disposals of financial fixed assets

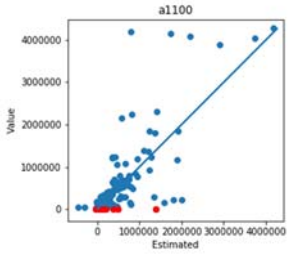
II- Results of the estimation.

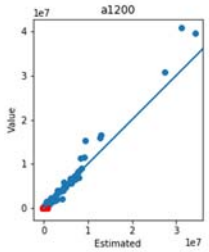
The blue line is the bisector.

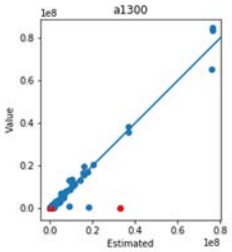


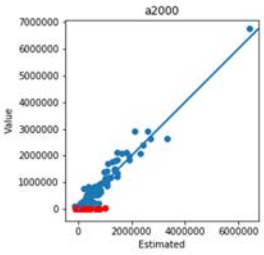


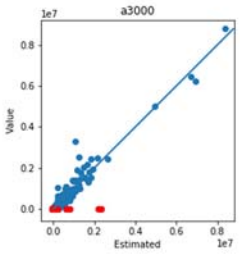


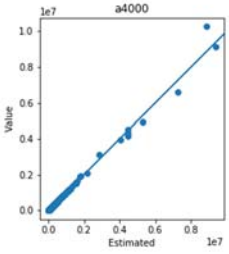


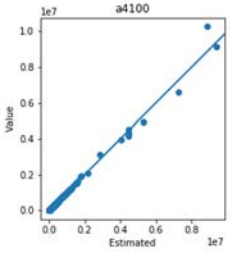


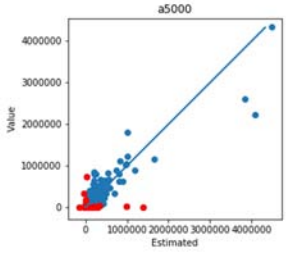


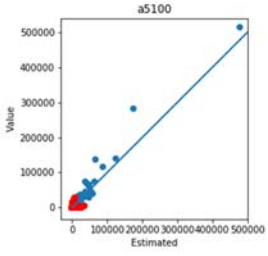


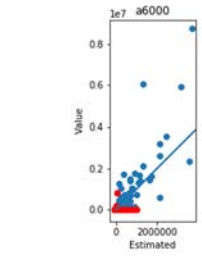


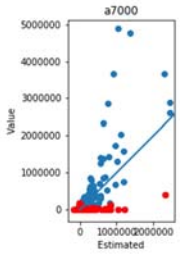


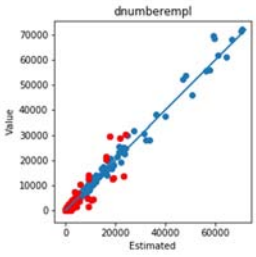


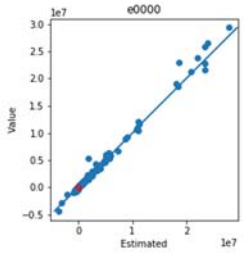


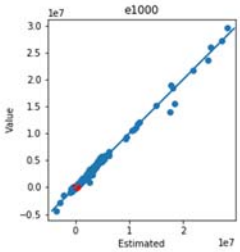


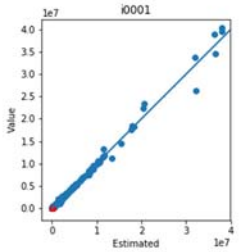

a6000
1e7
Value
Estimated

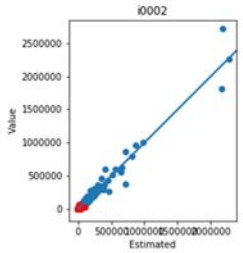

a7000
Value
Estimated

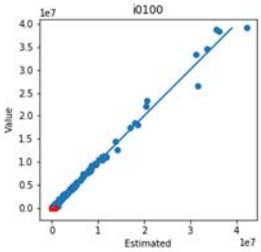

dnumberempl
Value
Estimated

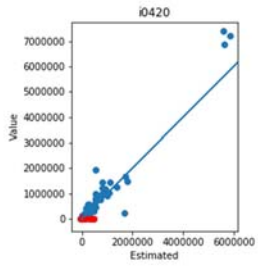

e0000
1e7
Value
Estimated

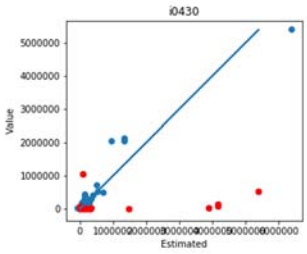

e1000
1e7
Value
Estimated

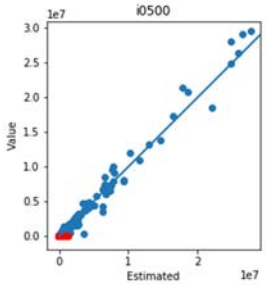

i0001
1e7
Value
Estimated

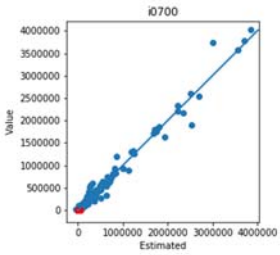

i0002
Value
Estimated

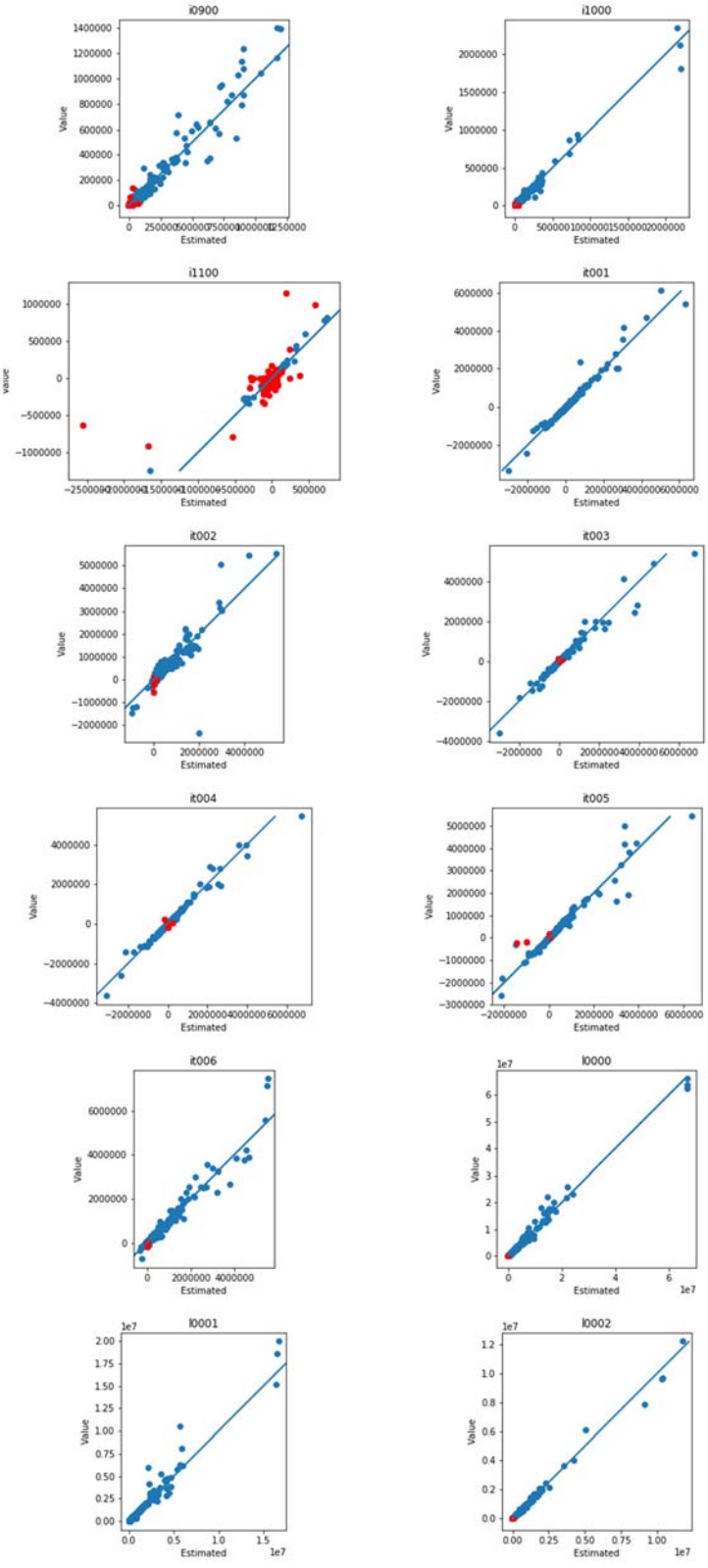

i0100
1e7
Value
Estimated

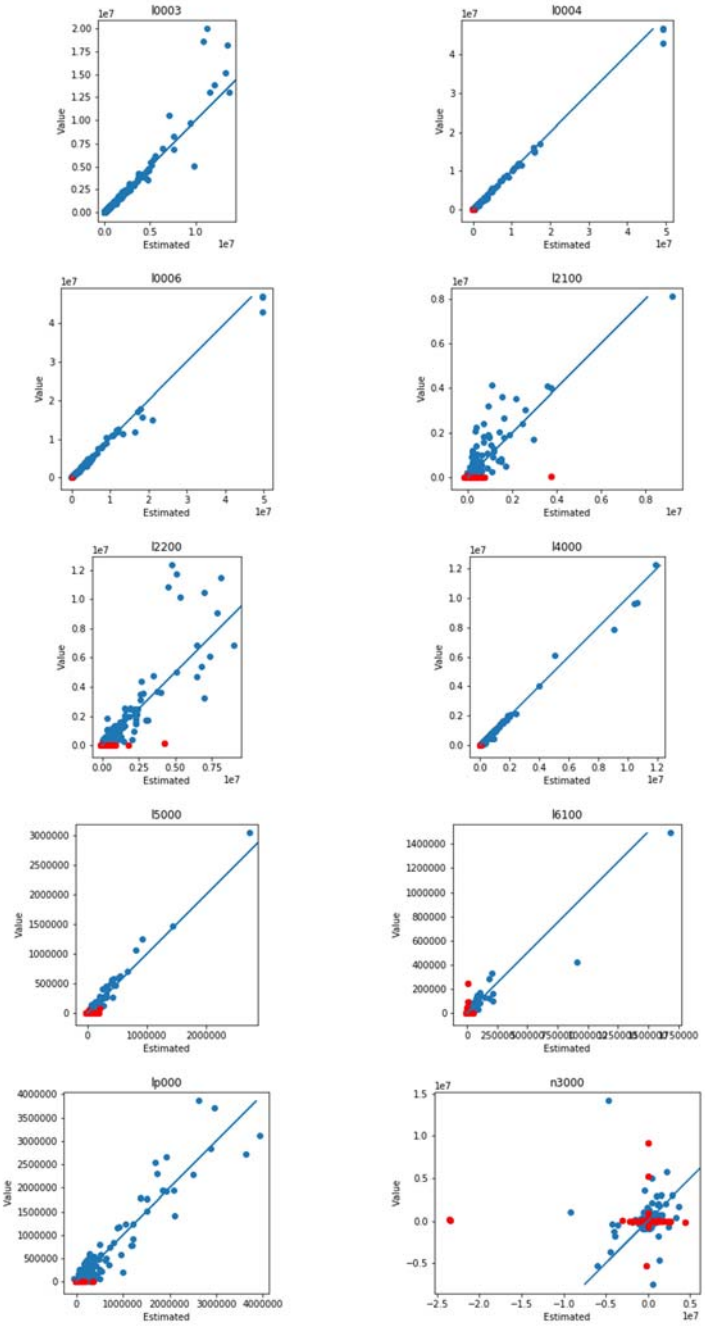

i0420
Value
Estimated


i0430
Value
Estimated


i0500
1e7
Value
Estimated


i0700
Value
Estimated


i0900
i1000
i1100
it001
it002
it003
it004
it005
it006
l0000
l0001
l0002
Value
Estimated


l0003
l0004
l0006
l2100
l2200
l4000
l5000
l6100
lp000
n3000
Value
Estimated
1e7

References

C. Bates, A. Schubert New challenges in labour market statistics: The perspective of a central bank in Europe, Eurostat Conference on Social Statistics 2016

M. Ester, H. Kriegel, J. Sander, X. Xu "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", KDD-96 Proceedings 1996

J. Bergstra, Y. Bengio "Random Search for Hyper-Parameter Optimization", J. Machine Learning Research 2012

T Chen, C Guestrin XGBoost: A Scalable Tree Boosting System, KDD 2016, arXiv:1603.02754

M. Claesen, B. De Moor. "Hyperparameter Search in Machine Learning", arXiv:1502.02127 2015

J. H. Friedman Greedy function approximation: A gradient boosting machine, Ann. Statist. 29 (2001)

V. Hodge, J. Austin A Survey of Outlier Detection Methodologies, Artificial Intelligence Review 2004

A. Novello Computation of Number of Employees for iBACH data, ECB internal memo 2017

Irving Fisher Committee on
Central Bank Statistics

BANK FOR INTERNATIONAL SETTLEMENTS

Ninth IFC Conference on "Are post-crisis statistical initiatives completed?"

Basel, 30-31 August 2018

A machine learning approach to outlier detection and imputation of missing data[1]

Nicola Benatti,
European Central Bank

[1] This presentation was prepared for the meeting. The views expressed are those of the authors and do not necessarily reflect the views of the BIS, the IFC or the central banks and other institutions represented at the meeting.

EUROPEAN CENTRAL BANK

EUROSYSTEM

Nicola Benatti
European Central Bank

A machine learning approach to outlier detection and imputation of missing data

9th IFC Conference 30-31 Aug 2018, Basel

Overview

What is an outlier?

- An outlier is an observation which is significantly distant from the other considered observations.

- Often outliers are identified by assuming the true distribution of each variable separately to be a known one.

- Alternatively, distributional methods are used but they do not suggest the true values of the Observation.

- It is very important that outliers are not automatically considered as errors since extreme cases can still be justified.

- The aim of this analysis is to rank observations that need to be assessed by their likelihood of being errors.

The iBACH dataset

- Balance sheet and profit and loss data of firms collected by the European Committee of Central Balance Sheet Data Offices (ECCBSO) within its WG on Bank for the Accounts of Companies Harmonized (BACH).

- Aggregate database available since several years but firm level data (iBACH) available to participating countries since February 2018.

- 66 numeric variables taken into consideration in the analysis I carry out

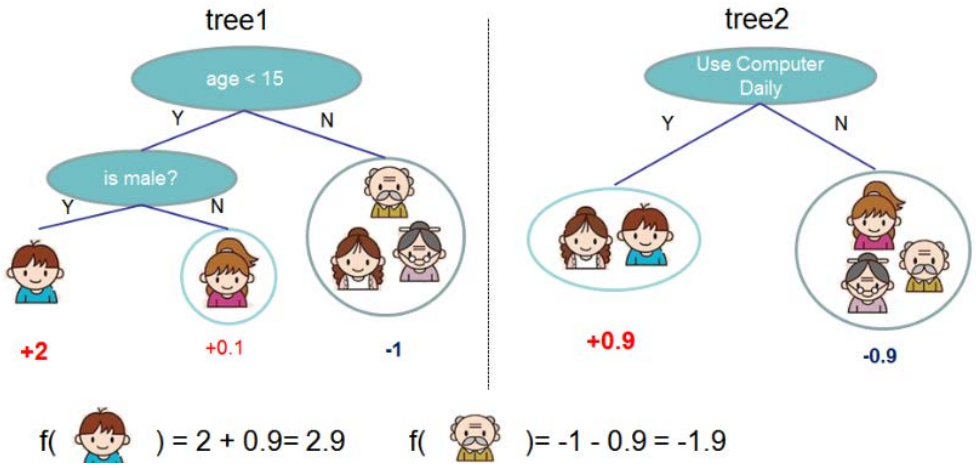
Number of entities

	dyear													
dcountry	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
BE		204,825	218,707	233,180	250,392	264,474	284,327	297,899	326,480	344,480	362,762	377,386	382,669	349,034
ES							450,538	447,540	459,076	450,146	443,527	583,081	560,570	324,701
FR	184,812	192,206	198,107	208,534	225,408	233,267	233,865	244,843	260,670	260,565	250,048	253,758	257,950	261,051
IT	492,472	517,464	540,517	560,140	582,993	600,656	613,021	624,235	634,278	629,865	627,317	621,722	618,177	464,353
PT	16,920	17,547	15,176	342,588	357,480	367,237	366,806	365,821	373,230	373,500	378,731	382,779	390,730	392,030
SK												99,389	99,584	97,869

Sum of n_entities broken down by dyear vs. dcountry. Color shows sum of n_entities. The marks are labeled by sum of n_entities.

Estimation: XGBoost, Gridsearch

- The estimation technique used is extreme gradient boosting (Chen 2016, in the python package xgboost)



- The hyperparameters are set using a Gridsearch algorithm (M. Claesen, B. De Moor 2015, in the python package Gridsearch) which iterates over a tuple of values and choses the optimal set for the following hyperparameters of xgboost: max depth, eta, subsample, number of estimators

Detection: Distance measures and importance averaging

- **Outlier flagging methods using estimation residuals:**
  - K-nearest neighbour on absolute and relative distance from true value
  - Distribution based on both absolute and relative distance from true value

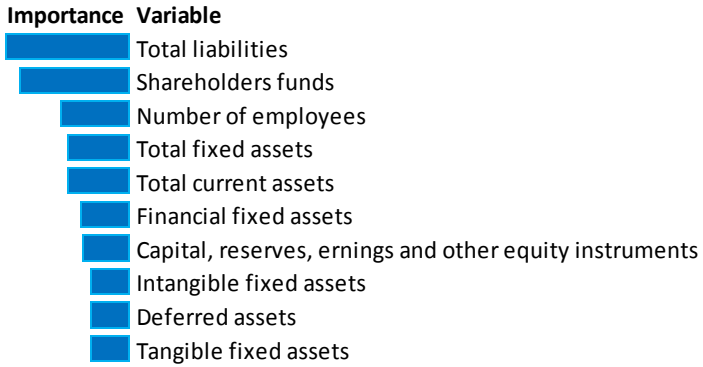
- **Importance averaging:**
  - While causality is confirmed, it might not be clear where the error comes from

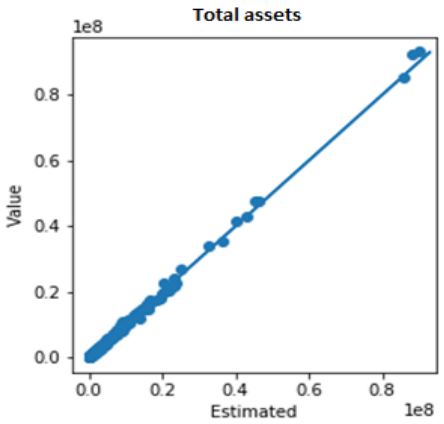
    A=B-(C*D) is false
    Which variable among A, B, C and D is wrong?

  - For each firm/year I sum the contribution of each variable to the model of detected outliers and create a ranking of “most-likely-to be wrong”.

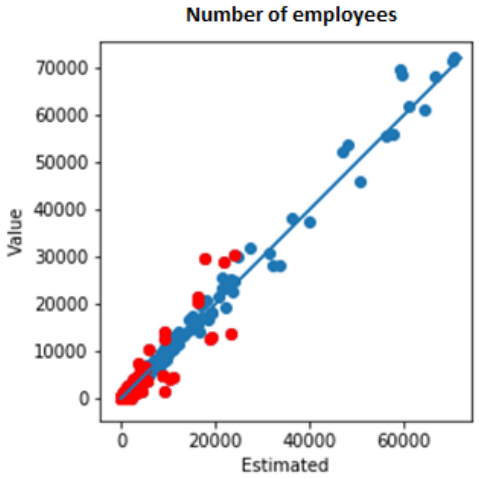
Results

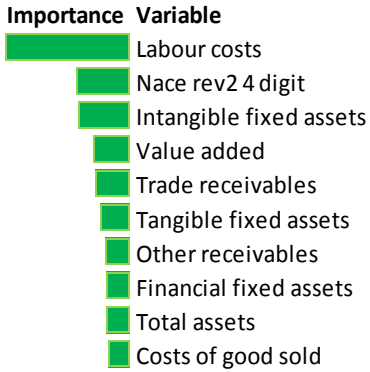
The algorithm allows to accurately estimate all variables analysed.





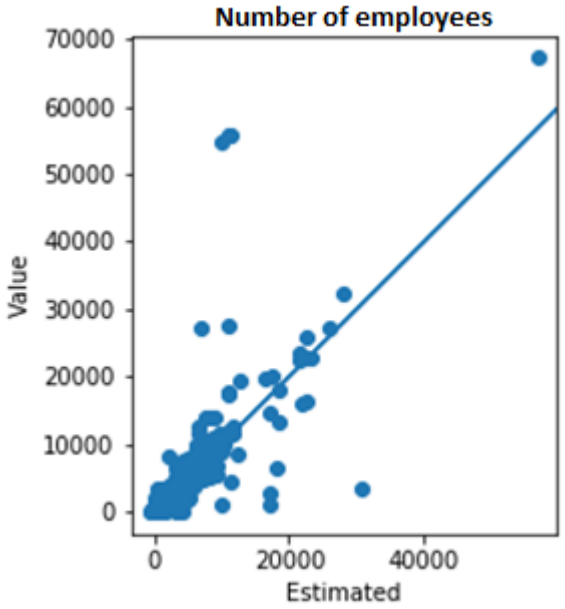
The outliers detected re sent to the NCBs to be investigated, ranked by likelihood of being errors.

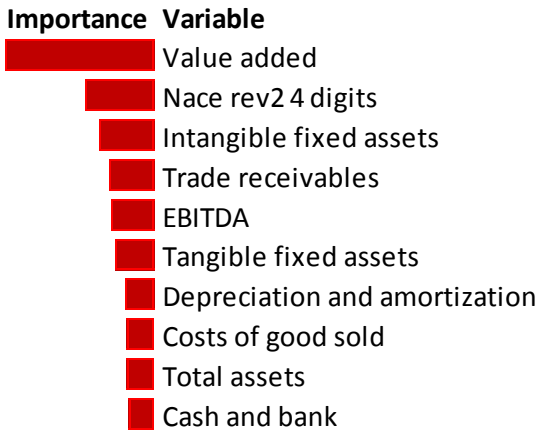




Imputation

The same methodology can be used to estimate the missing values in the dataset. As an exercise, when estimating employment, forcing out the labour costs variable, the estimation still over-performs the methodology used previously internally.





Conclusions

- This paper presents an application of a combination of supervised and unsupervised machine learning with a final feature-additive ranking technique in order to spot mistakes in outlying datapoints.
- The methodology described seems to be useful also for additional steps of data quality improvement such as data imputation.
- This technique also provides guidance for the construction of new data quality checks that could prevent the submissions of mistakes.

Further improvements:

- The increase in the sample size.
- The inclusion of lagged variables would allow for using long-term-short-term memory frameworks.
- The comparison of the results with neural networks and multi-target regressions.
- Inclusion of the confirmation on whether a spotted potential mistake is actually an error or not to transform the distance measure into a classification problem.