University of Southampton

Outlier and Anomaly Detection Methods with Applications to the 2021 Census

Zoheir Sabeur, Gianluca Correndo and Galina Veres

IT Innovation Centre, Department of Electronics and Computer Science

Paul A. Smith and James Dawber

Southampton Statistical Sciences Research Institute

University of Southampton

Executive Summary

The Office of National Statistics (ONS) contracted the University of Southampton to conduct research concerning the use of statistical and data science methods for the automatic detection of outliers and anomalies in Census data. This project considered both Census 2011, which was based mostly on traditional survey methods, and Census 2021, which was mostly conducted using online surveys. The ONS has since given us permission to publish the findings of this project. This information is licensed under the Open Government Licence v3.0. To view this licence, visit http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

This research project is in close collaboration with ONS and has run under two phases. These are:

Phase 1: Literature review & selection of methods for detection of outliers and anomalies in Census 2021 data

Phase 2: Prototype demonstrator of outlier and anomaly detection in Census 2021 data

This document is the final report of phase 2 concerning the testing of statistical and data science methods, selected during phase 1, for the detection of outliers and anomalies in Census 2021 data. These methods were investigated using synthetic data perturbations to simulate anomalies which are likely to occur in the census data. The data perturbation strategies were decided in consultation with experts at ONS.

The second phase of the project has been conducted with the following procedures in consultation with ONS:

- The acquisition of experimental data in this study was achieved by accessing "2011 Census Microdata LA data", which were made available by the UK Data Service.
- The experimental 2011 census data had already been processed and cleaned, with no expected anomalies to be present. Thus, in order to assess our anomaly detection methods on the census data, anomalies needed to be synthetically added in this project.
- Several discussions with ONS experts led us to strategize on how to synthetically add anomalies in the data. Data perturbations were performed in accord with real errors that occurred in the previous Census.
- A significant number of selected potential methods (both statistical and data science-based) for the detection of outliers and anomalies were investigated using the newly perturbed 2011 Census data. Their respective performances for the detection of census data anomalies were obtained.
- Benchmarking for the Spark implementations of the selected outlier detection methods was performed. This early testbed experiment revealed scalability trends over increasing volumes and complexities of census records.
- The various outlier detection scripts were integrated onto the Jupyter environment as the first prototype demonstrator for ONS.
- Three major research programmes have been identified for future studies: *Methods for Census Data Perturbation*, *Outlier and Anomaly Detection Methods and Machine Learning Strategies,* and *Methods Scalability using Spark Technology*. Each of the topics are discussed in Section 6 with future recommendations.

Contents

1. Introduction

1.1 Census Data Description

The Census data used for this project were the Census 2011 microdata individual safeguarded samples, obtained from the "2011 Census Microdata LA data" made available by the UK Data Service[1] [1][2]. This dataset provides an intermediate level of detail with less restrictive access limitations. The available data were comprised of 2,848,149 individuals from processed Census data. 120 variables were included (excluding the unique case number), with a mix of standard and derived variables. Since the data had been processed, all count, date and free text variables were represented as categorical data. As shown below in the table updated from the Phase 1 report, this resulted in most variables being nominal with some ordinal variables.

Table 1: Census microdata 2011 variables by data type

	Census section		
Data type	**Household**	**Individuals**	**Visitors**
Count	NA	0	NA
Date	NA	0	NA
Binary	NA	4	NA
Ordinal	NA	15	NA
Nominal (>2 groups)	NA	101	NA
Nominal with free text option (e.g. Other)	NA	0	NA
Free text	NA	0	NA
Total	**NA**	**120**	**NA**

1.2 Data perturbation

As the Census data had already been processed and cleaned, no anomalies were expected to be present. Anomalies therefore needed to be synthetically introduced in order to assess anomaly detection methods. This was done by perturbing the data in accordance with real errors that occurred in the previous census. Two such errors were assessed:

- The first perturbation introduced to the data was based on an error in which individuals born in the 1980s were given years of birth in the 1890s. Individuals between the age of 21 and 31 at the time of the 2011 Census were hence recorded as between 111 and 121 years of age. Although all ages over 90 were grouped together in the available Census microdata, this error still caused anomalous data points- e.g. 90+ year olds living with dependent children. This perturbation is later referred to as the '198x ->189x' error.

- The second perturbation introduced was based on an error in which small "census districts" (CDs) were completely lacking any data, and were consequently left with empty rows in the dataset. This can be caused by errors such as postal mistakes. As the Census microdata only identified geographic areas down to the Local Authority District (LAD) level, each LAD was divided into 50 distinct CDs using synthetic area classifiers. Following this, three additional CDs were created within three arbitrarily chosen LADs. Each of these CDs contained 150 blank individual records, with the only non-empty variables being case number, LA group, and the synthetic CD group.

These two examples of data errors were used to find the extent to which anomaly detection methods can detect such errors.

[1] https://census.ukdataservice.ac.uk/get-data/microdata

2. Investigated Statistical Methods - Outlier and Anomaly Detection

2.1 Statistical distance measures

Statistical approaches using distance measures were found to be rather impractical to use, for two main reasons. Firstly, the data were solely nominal and ordinal; as discussed in the Phase 1 report, it is challenging to adapt continuous multivariate distance measures such as the Mahalanobis distance for categorical data. This is due to the difficulty in capturing correlation and covariance between variables, which is otherwise straightforward for continuous data using well-defined metrics. This leads onto the second issue: while some distanced-based methods are capable of measuring covariation for categorical data(such as the methods in [3-5]), they are so complex that no package was available in R to run them. Meanwhile, the distance measures that were available were too simple to provide any insight. Although code could have been constructed based on the articles describing the more complex methods, it would have been a heavy investment in time for a method that may be useless in the context of census data. Nevertheless, these methods could be investigated in more detail in the future.

2.2 Multiple Correspondence Analysis

The statistical method that proved to be most useful for identifying anomalies was Multiple Correspondence Analysis (MCA). MCA is a dimension reduction method, acting as a nominal data version of the more widely known Principal Component Analysis (PCA) or Factor Analysis methods used for continuous variables. These techniques are very useful for identifying key patterns in datasets with large numbers of variables. They provide insight into the relationships between variables as well as between units, and for MCA, between categories.

A good example which demonstrates the usefulness of these techniques is the decathlon, a ten event track and field athletics discipline. The four track events are 100m, 400m, 1500m, and 110m hurdles, and the six field events are long jump, high jump, shot-put, discus, javelin, and pole vault. It can be expected that athletes who perform particularly well at 100m might also do well in 400m; likewise, strong shot-putters are likely to perform well in other throwing events. This shows inherent correlation structures between the variables. A PCA (since the data would be continuous in this case) would reduce the dimensions of the data down to show that two variables instead of ten may represent the correlation structure rather well. In this example, these two variables would represent the speed and strength of the athletes. Understanding this structure could then lead to finding anomalous athletes; for example, athletes with generally high speed and lower strength that do very well in shot put, an event which typically favours stronger athletes. This particular athlete could thus be considered an anomaly in terms of shot-put ability. MCA can be used similarly to inspect possible anomalies in Census data. To understand the technical details of MCA, the textbook Greenacre & Blasius [6] is suggested.

Before anomalies can be inspected using MCA, the dimensions of the data must be understood. An exploratory MCA was hence conducted on the clean, unperturbed census microdata. This was done using the R package 'FactoMineR' [7]. Due to the computational demands of the method and the large number of records, only the first one million records were selected. As MCA does not accommodate ordinal variables, all variables were considered nominal. Non-applicable levels (left as blanks) were considered in the analysis as a distinct group, rather than as missing data. Variables with only one response level were removed from the analysis. This left 96 variables for the initial MCA.

The results of this initial MCA identified one particular variable having a dominant effect: the *popbasesec* variable, which identifies whether an individual is a usual resident, student living away, or short-term resident, caused a small percentage of clear outliers. This variable predictably has substantial implications for other variables, primarily by causing a large number of 'not applicable' responses where the individual is not a usual resident. However, as 98.5% of records were usual residents, the MCA was recalculated without any non-usual residents to remove this dominant effect. As desired, this second MCA showed no single dominant variable and was used as the basis of subsequent analysis.

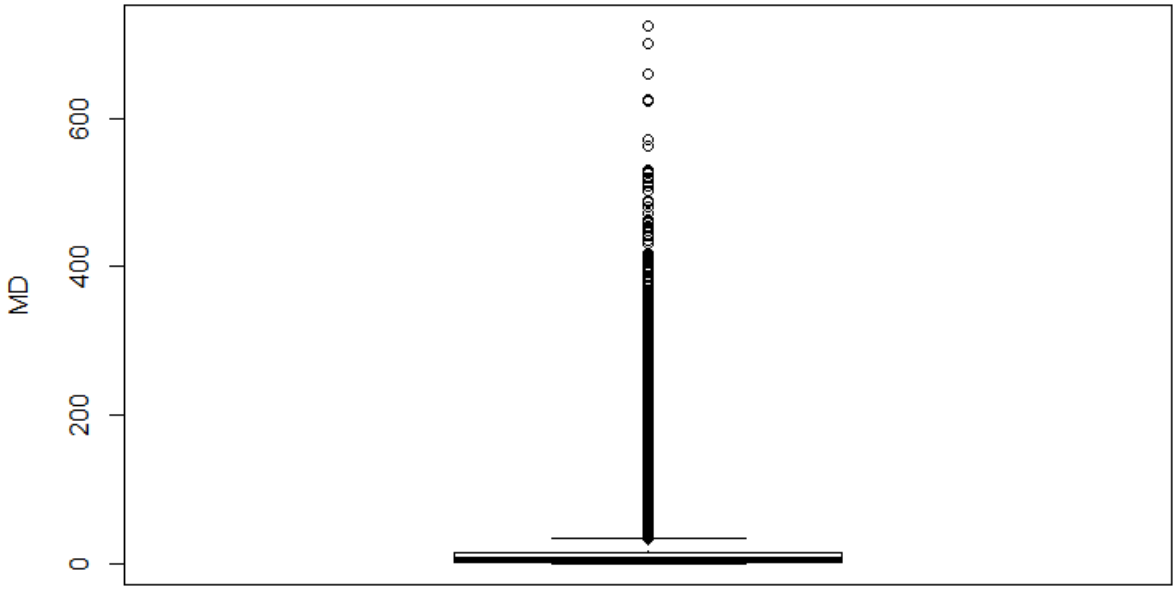
The aim of this preliminary analysis was to identify what the MCA-reduced dimensions represented, or if they even represented something identifiable. Each dimension was therefore assessed to see which variables, and which levels within variables, were highly weighted in a given dimension. These levels and variables were inspected and identified as a single factor if possible. This was done systematically until further dimensions could not be identified clearly. The dimensions were identified like so:

Table 2: example of MCA dimension characterisation		
Dim.	**Dim. Identity**	**Characterised by**
1	Children	Student, dependent, single with NAs for any economic variables
2	Retirees	Retired, pensionable household
3	Foreigners	"Other" passport and national identity, foreign language
4	Welsh	Country and region identifiers
5	One person household	Several variables identify this
6	Workers at home	Several variables identify this
7	Self-employed	Similar to previous dimension except self-employed
8	Full-time students	Also associated highly with Polish
9	Middle-eastern Muslims and Polish	Country of birth and religious status
10	FT students with no fixed workplace	Several variables identify this
11	No fixed workplace, part-time workers	Variable levels for builders and construction site workers are high too
12	Irish	Several variables identify this

The identification of these dimensions unlocked a greater capacity to detect anomalies. This is firstly due to the MCA, which created a continuous variable along each dimension for each record. Working with these transformed continuous variables rather than the raw categorical variables made it much easier to detect outliers. Secondly, these dimensions can be used to intuitively find anomalies: for example, an individual with high values in the first two dimensions is suggested to be both a child and a retiree. Furthermore, rather than trying to detect anomalies within each dimension, it is possible to get the Mahalanobis distances (MDs) for each record relative to the centroid of all records. This gives an overall measure of how peculiar the record is based on the twelve dimensions and their correlation structure.

Although the data was cleaned of anomalies, a boxplot of MDs from the preliminary MCA is shown in figure 1 below (just the first ten dimensions were used because dimension 11 was not strongly clear and the Irish dimension was perhaps too clear, separating out Irish individuals so well they became the most "anomalous"):

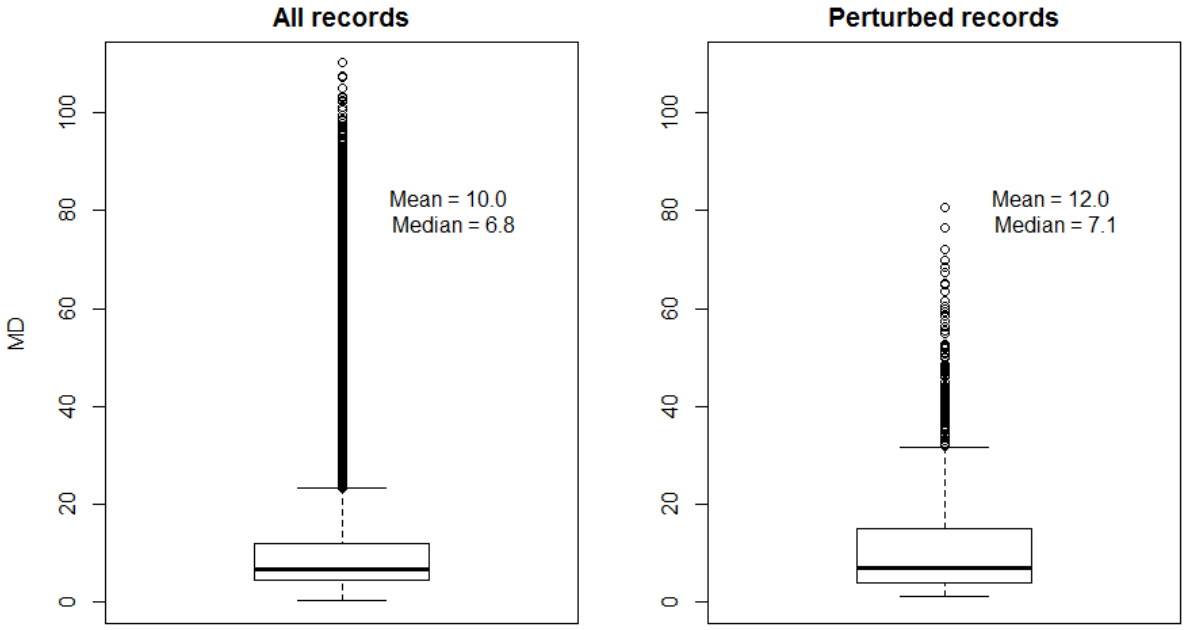
Figure 1 Boxplot of Mahalanobis distances of MCA output



The distribution of these MDs is heavily right-skewed, with the largest values representing records which are most likely to be anomalous. These were mostly records with high numbers on dimensions 3 and 9, but low numbers on dimension 10. These records thus corresponded with foreign full-time students from Poland and Middle-Eastern countries. While such population groups are by no means anomalous, they are slightly uncommon; this method was therefore able to identify peculiar records as potential anomalies. However, further analysis was necessary to find the extent to which actual errors could be detected in the perturbed data.

Ideally, it would be desirable for the MCA and subsequent MDs to reveal distinctly large distances for the perturbed records. For the first data perturbation type with the erroneous birthdate, a total of 1,448 (0.15%) perturbed records were introduced to 985,340 total records (only usual residents were used). A comparison of the MDs between all records and the perturbed records is shown below in Figure 2.

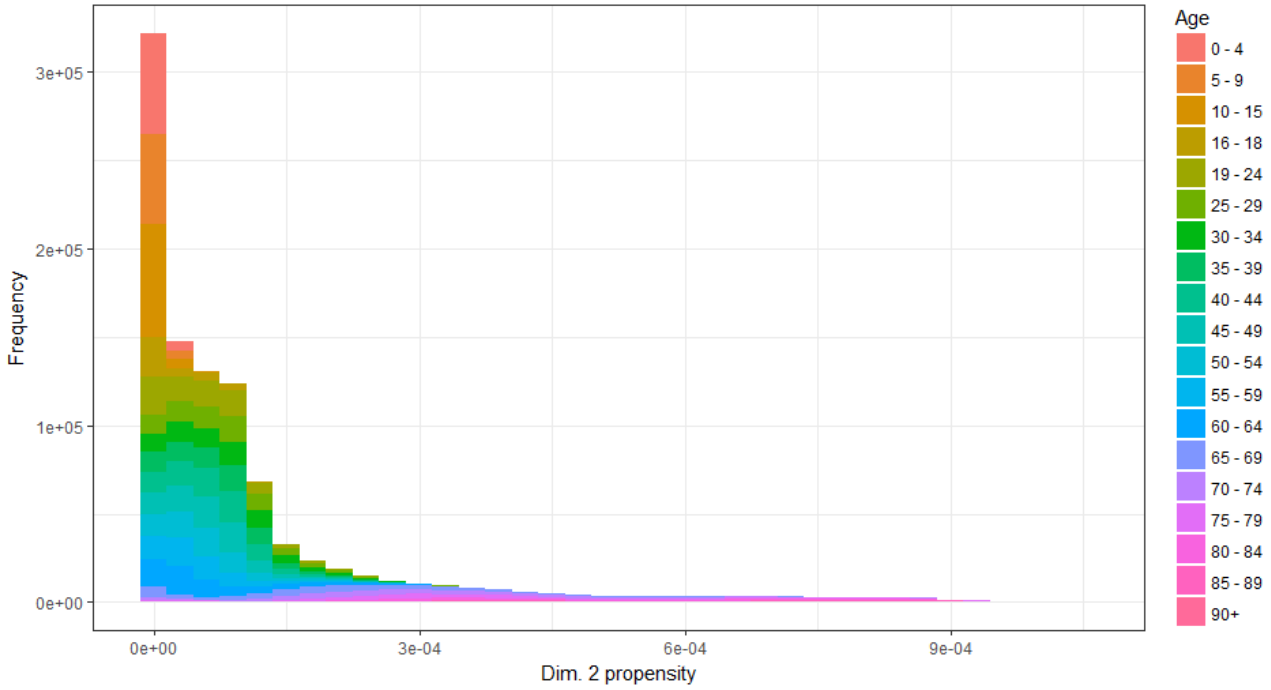
Figure 2 Boxplots of Mahalanobis distances of MCA output comparing perturbed data



Clearly the perturbed records do not have substantially larger distances, showing that using MCA in this way is not a useful anomaly detection tool. However, a more detailed examination of the MCA can be applied: if

we consider the perturbation that is occurring and the dimensions that the MCA uncovers, then we can apply a more focused approach to finding anomalies. If the perturbations represent people 90 years and over who are in fact young adults, then we should expect the second dimension, retirees, to rank them lowly. In other words, if dimension 2 is a propensity score associated with characteristics of retirees, then the perturbed records should have low scores, despite their (perturbed) age. The distribution of the dimension 2 propensity scores shown in Figure 3 indicates a clear relationship with age:

Figure 3 Histogram of 'retiree' dimension 2 by age



This relationship between age and dimension 2 is expected: the older the person, the higher the value for retiree propensity. However, the question remains where the perturbed records lie in this distribution, especially compared to other 90+ year olds. Figures 4 and 5 show how the perturbed records are clearly distinct from other 90+ year olds:

Figure 4 Histogram of 'retiree' dimension 2 within 90+ year olds

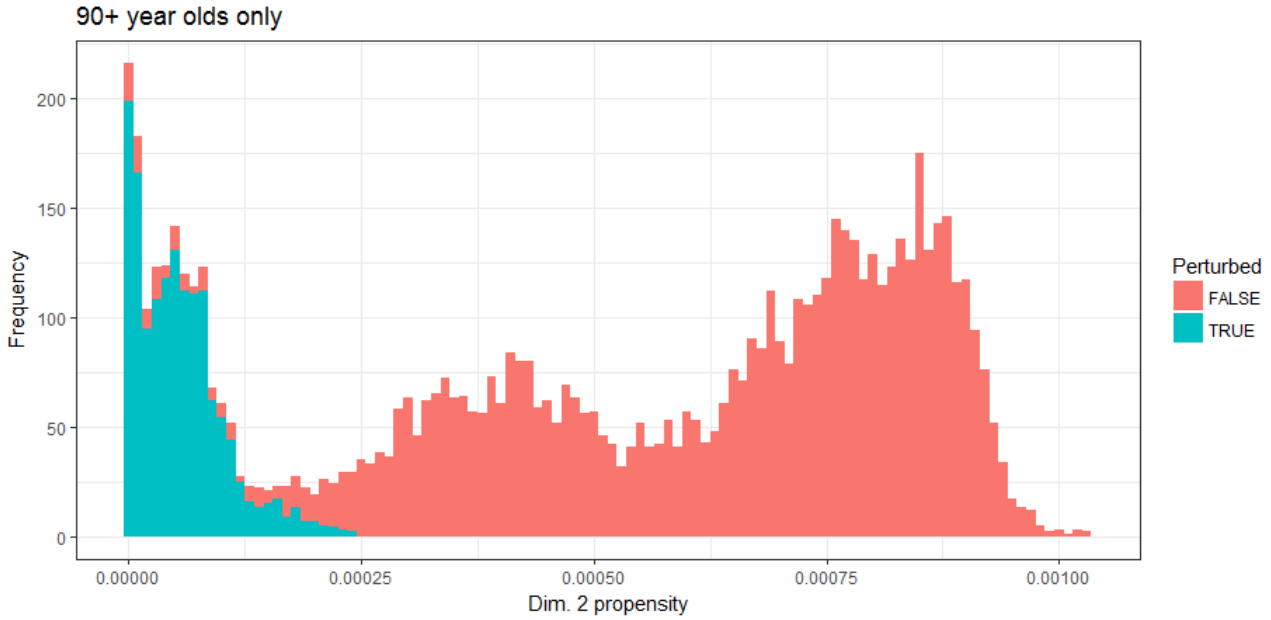
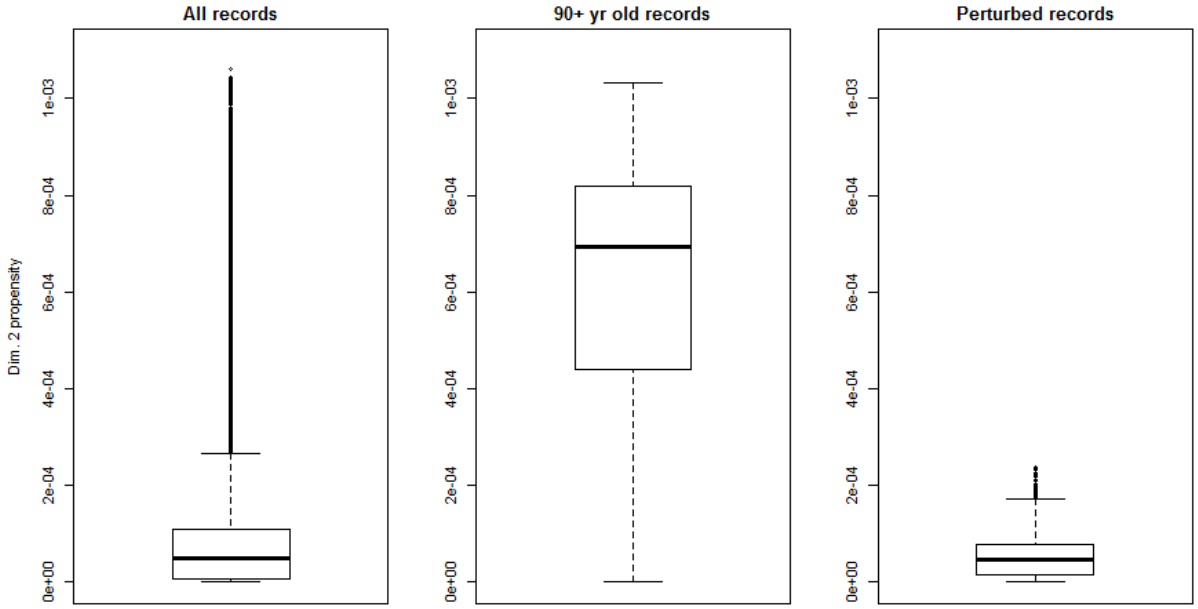


Figure 5 Boxplots of ‘retiree’ dimension 2 propensity scores within different groups

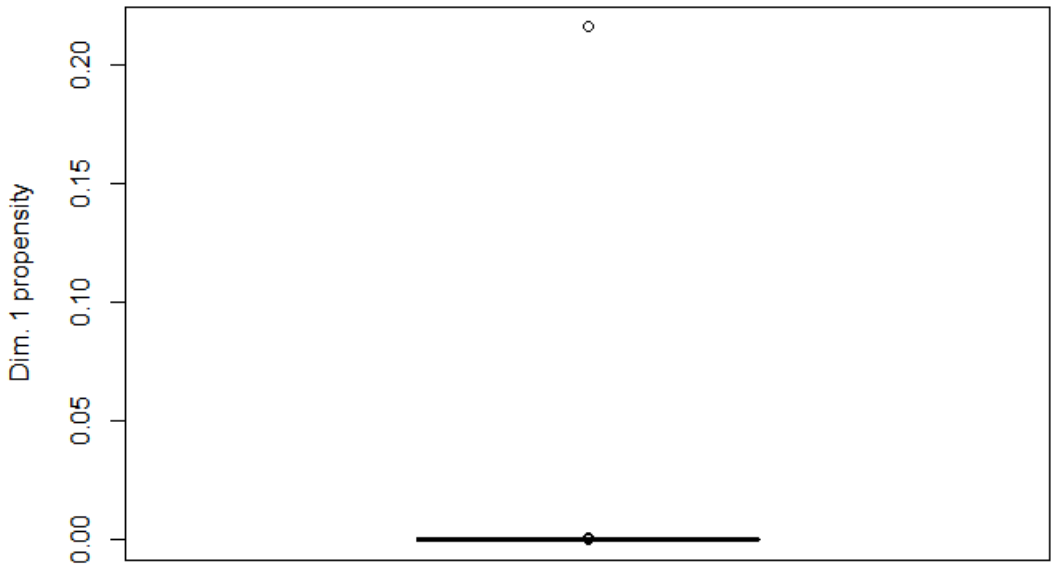


These figures demonstrate that the MCA method on dimension 2 can differentiate the true 90+ year olds from the perturbed records reasonably well. Although there is some cross-over, this is due to the few 90+ year olds that are still economically active. In fact, the “true” 90+ year old with the lowest dimension 2 score works full time, drives to work 20-40km away, is married, and has very good health. it is therefore almost impossible to completely differentiate this record from a person in their twenties. Nonetheless, with prior knowledge of the MCA dimensions, this method can be used effectively for identifying specific potential anomalies.

For the second perturbation relating to missing records in CDs, MCA was expected to be a useful identification method: as all missing CDs shared the same missing values, all variable levels should be perfectly correlated. Aside from a household identifier and geographic information, all other variables were either blank or contained elements suggesting non-response. A total of 22 LADs were selected for testing this perturbation type, including the three LADs containing blank CDs. As the focus was on identifying the blank CDs, it was not important to include too many variables. Five important variables were therefore included to go with the LAD and CD groups. These were combined to create a unique identifier for each CD. The 22 LADs were comprised of 245,851 records in total, 450 (0.18%) of which were empty.

Following the application of an MCA, the first dimension was found to correspond to blank vs non-blank records; this predictably made identifying blank records very straightforward. Figure 6 shows a boxplot of the dimension 1 scores, with the single outlying dot representing all 450 overlapping blank records. Needless to say, this method identifies these anomalies very well.

Figure 6 Boxplot of dimension 1 propensity scores from the MCA



3. Investigated Machine Learning Methods - Outlier and Anomaly Detection

The purpose here was to investigate the ability of machine learning techniques to detect the introduced '198x -> 189x' error, along with any other anomalies present in the sample dataset. The following machine learning techniques were selected for implementation and investigation: K-means clustering, KAMILA, Rule-based approach, SPAD, One-class Support Vector Machine for novelty detection, Pattern Outlier Detector based on Support Vector Machines, and Pattern Outlier Detector based on Boosted Regression Trees. While the majority of these approaches were described in the Phase 1 Report, a short description for new approaches is given below. Given both the nature of the problem and the type of data considered (census data where all anomalies were already filtered out), there was no ground truth to use in training the models. All methods were therefore unsupervised.

According to the document '2011 Census Variable and Classification Information: Part3.pdf', the following standard (key) variables can help to detect the 198x -> 189x error:

- *residence_type*
- *health*
- *hours*
- *disability*
- *marstat*
- *student*
- *empstat*
- *lastyrwrkg*

However, since *residence_type* was uniformly distributed in the sample set of 1mln records, it was not considered. Several further potentially useful standard variables were also available in the sample dataset: *occ_current* (current occupation), *activity last week*, *TERMIND*, and *socmin* (standard occupation classification). Four derived variables were also considered: two perturbed age variables, *ageh0* and *ecopuk11* (derived from age and two other unavailable standard variables) were included in analysis, along with *popbasesec* and *scanddtyp*.

3.1 K-means

The K-means algorithm was run on the first 1mln records using all variables mentioned above, without scaling and with two clusters. The Euclidean distance was calculated between each record and the centre of its assigned cluster. The first 1,000 outliers were selected based on this distance by sorting in ascending order. Of the top 10 anomalies identified, three were introduced errors. The K-means algorithm detected records as anomalies when they met (all of?) the following criteria:

- NA for *disability, ecopuk11, empstat, health, hours*, (and/or?) *lastyrwrkg*
- Claimed to be students living away from home with secondary address as Other
- Belonged to one of the following age groups: 90+ (introduced error), 0-4, 5-9, 50-54, or 65-99

3.2 KAMILA

The KAMILA algorithm was run on the first 1mln records using two clusters and 30 initialisations of initial cluster centres. Since KAMILA requires both categorical and numerical variables, *popbasesec* was considered as categorical, while the remaining variables were scaled and treated as numerical.

On the first 1 mln records, the top 10 outliers detected included three introduced errors. The top outlier records met the following criteria:

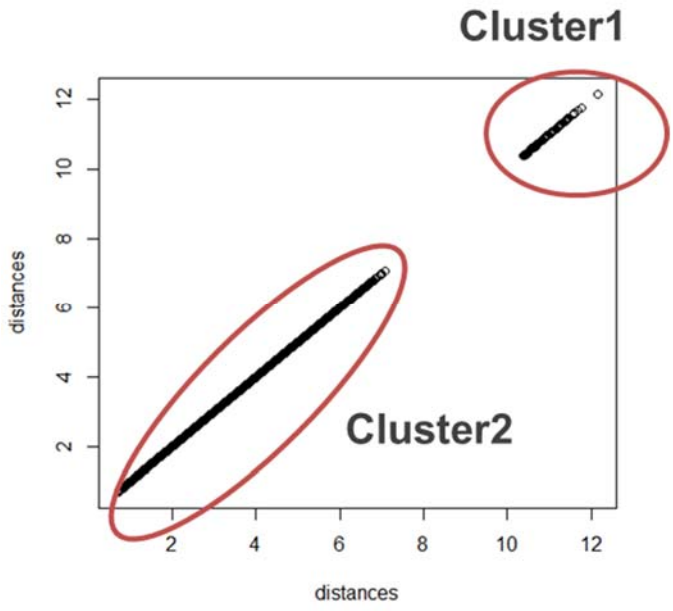
- NA for *disability, ecopuk11, empstat, health, hours*, (and/or?) *lastyrwrkg*
- claimed to be students living away from home
- Belonged to following age groups: 90+ (introduced error), 10-15, 35-44, 45-54, or 65-69.

On the remaining ~1.8mln records, 18 introduced errors were detected in the first 1,000 outliers (17 of which were in the first 25 outliers). The records in the top outliers were:

- Records with NA for *disability, ecopuk11, empstat, health, hours*, (and/or?) *lastyrwrkg*, claiming to be students living away from home with secondary addresses , and belonging to age groups: 90+ (introduced error), 50-54, 60-64, or 70-79

An example of distance-based clusters formed by KAMILA is given in Figure 7. Two clear clusters can be observed, with some visible outliers for both.

Figure 7 Clusters formed by KAMILA algorithm



Detecting outliers for each cluster produced the following results:

Cluster1 had the largest distances (first 200 outliers) for records meeting the following criteria:

- students , living away from home, all of them having secondary address (student (home or term-time), holiday home, other, another parent/guardian address), with missing values for *disability ecopuk11 empstat health hours lastyrwrkg* (and/or) *carer*.
- This cluster catches 3 of introduced errors

Cluster 2 had the largest distances (first 200 outliers) for:

- People with usual or short-term residents, having holiday home/other secondary address, claiming to be economically inactive with NA for hours, and not working at least from 2010, however at the same time claiming to be employees or self-emp/freelance; records with NA for *empstat, hours, lastyrwrkg*

3.3 Rule-based

A rule-based approach involves designing sets of rules for detecting particular anomalies that are known or suspected to be in the data. To detect the '198x -> 189x' error, the following rules were used:

- Consider only records with age 90+.
- Check the *student* variable- this identifies students who are very old.
- Check *hours*, *empstat* and *lastyrwrkg* – this reveals very old individuals who are still employed and the hours they work.
- Check *scanddtyp* – this detects very old people with unusual secondary addresses, such as an Armed Forces base.

This set of rules allowed the detection of anomalies among 90+ year olds, and was able to pick up our introduced errors along with other possible anomalies. 437 records were affected by the introduction of the '198x -> 189x' error. Some statistics based on the rules above revealed the following:

- 146 90+ year olds claimed to be students, including 94 (21.5%) of the introduced anomalies. The remaining 52 anomalous records were in the original data sample.
- Of the 90+ year olds not claiming to be students, 746 claimed to work full or part time. This included 258 (59%) introduced errors, while the remaining 488 anomalies were present in the original dataset.

When applied to the data sample, the rules above were able to catch 80.5% of introduced errors. Furthermore, by applying different rules, even more anomalies could be detected: this included 24 children aged 0-3 cohabiting with a partner, one 90+ year old and three 80-89 year olds claiming to have an Armed Forces base as a secondary address, and a surprisingly small number of records from prisons in LAs with at least one detention centre.

3.4 SPAD

The simple probabilistic anomaly detector (SPAD for short) estimates a single multivariate probability as the product of univariate probabilities, with the assumption that univariate attributes are independent of each other. The frequency of each attribute value and the number of unique values for each attribute are used to compute the probability of a record having a given attribute value using the Laplace-corrected estimate:

$$\hat{P}(x_i) = \frac{f(x_i) + 1}{n + w_i}$$

Where $f(x_i)$ is the occurrence frequency of x_i in the dataset, n is the number of records in the dataset, and w_i is the number of possible values for x_i.

The underlying assumption in SPAD is that rare records are either anomalies or simple outliers. In this second phase of research the SPAD algorithm was used on unperturbed data to check the type of outliers produced, specifically in the context of age anomalies introduced by the '198x-189x' error.

The categorical variables used in this case are the same used for the other cases:

- *health*
- *hours*
- *disability*
- *marstat*
- *student*
- *empstat*
- *lastyrwrkg*
- *carer*
- *popbasesec*
- *scaddtyp*
- *ecopuk11*
- *ageh*

Once the combined probabilities of each record had been computed, all instances were ranked in ascending order, with the most unlikely records assumed to be anomalies. By running the algorithm on the unperturbed data, we could inspect the top 30 instances to see what type of records were returned; this is summarised in Table 2 below. Note that although many records seem duplicated, they are distinct rows from the input census data; the record ID was stripped off from the analysis.

Table 2 First 30 outliers returned by SPAD algorithm.

health	hours	disability	marstat	student	empstat	lastyrwrkg	carer	popbasesec	scaddtyp	ecopuk11	ageh	p
NA	NA	NA	3	1	NA	NA	NA	2	7	NA	11	-37.44818
NA	NA	NA	3	1	NA	NA	NA	2	7	NA	11	-37.44818
NA	NA	NA	3	1	NA	NA	NA	2	4	NA	5	-37.33718
NA	NA	NA	3	1	NA	NA	NA	2	4	NA	5	-37.33718
NA	NA	NA	3	1	NA	NA	NA	2	4	NA	5	-37.33718
NA	NA	NA	3	1	NA	NA	NA	2	4	NA	5	-37.33718
NA	NA	NA	3	1	NA	NA	NA	2	3	NA	5	-37.04213
NA	NA	NA	4	1	NA	NA	NA	2	-8	NA	7	-36.56488
NA	NA	NA	4	1	NA	NA	NA	2	-8	NA	7	-36.56488
NA	NA	NA	4	1	NA	NA	NA	2	-8	NA	7	-36.56488
NA	NA	NA	4	1	NA	NA	NA	2	-8	NA	10	-36.46263
NA	NA	NA	4	1	NA	NA	NA	2	-8	NA	9	-36.46119
NA	NA	NA	4	1	NA	NA	NA	2	-8	NA	5	-36.24112
NA	NA	NA	6	1	NA	NA	NA	2	-8	NA	4	-36.11206
NA	NA	NA	1	1	NA	NA	NA	2	-8	NA	19	-35.59867
5	3	2	6	1	1	1	3	3	6	8	14	-35.59627
NA	NA	NA	6	1	NA	NA	NA	2	-8	NA	10	-35.48954
NA	NA	NA	5	1	NA	NA	NA	2	-8	NA	8	-35.32571
NA	NA	NA	5	1	NA	NA	NA	2	-8	NA	8	-35.32571
NA	NA	NA	5	1	NA	NA	NA	2	-8	NA	8	-35.32571
NA	NA	NA	5	1	NA	NA	NA	2	-8	NA	8	-35.32571
NA	NA	NA	6	1	NA	NA	NA	2	6	NA	16	-35.30013
NA	NA	NA	4	1	NA	NA	NA	2	4	NA	7	-35.25309
NA	NA	NA	4	1	NA	NA	NA	2	4	NA	8	-35.24131
NA	NA	NA	5	1	NA	NA	NA	2	-8	NA	10	-35.23525
NA	NA	NA	4	1	NA	NA	NA	2	4	NA	6	-35.20260
NA	NA	NA	4	1	NA	NA	NA	2	4	NA	6	-35.20260
NA	NA	NA	4	1	NA	NA	NA	2	4	NA	10	-35.15084
NA	NA	NA	1	1	NA	NA	NA	2	1	NA	4	-35.11769
NA	NA	NA	5	1	NA	NA	NA	2	-8	NA	5	-35.01373

We can see that the 30 most unlikely records (which we assume to be outliers) have many fields set as **NA.** Codes for the non-NA values are given below:

marstat (*marital status*):

- 3: In a registered same-sex civil partnership
- 4: Separated, but still legally in a same-sex civil partnership
- 5: Divorced / formerly in a same-sex civil partnership which is now legally dissolved
- 6: Widowed / surviving partner of a same-sex civil partnership
- (only one single)

student:

- 1: 'Yes', whether usual resident, student living away or short-term resident(?)

scanddtyp (*type of second address*):

- 1: Armed Forces base address
- 3: Student's home address
- 4: Student's term-time address
- 6: Holiday home
- 7: Other
- -8: Multi tick

***Ageh* (age category)**:

- 1: 0-4
- 2: 5-9
- 3: 10-15
- 4: 16-18
- 5: 19-24
- 6: 25-29
- 7: 30-34
- 8: 35-39
- 9: 40-44
- 10: 45-49
- 11: 50-54
- 19: 90+

Moreover, we can see that record no.16 has a more detailed description and makes a strong candidate for a potential outlier. His characteristics are as follows:

health: 5- very bad health

hours (*hours worked per week*): 3- full-time, 31 to 48 hours worked

disability: 2- day-to-day activities limited a little

marstat: 6- widowed/surviving partner of a same-sex civil partnership

student: "yes"

empstat (*employment status*): 1- employee

***lastyrwrkg* (last year working)**: 1- in employment

carer: 3- yes, 20-49 hours

***popbasesec* (resident type):** 3- short-term resident

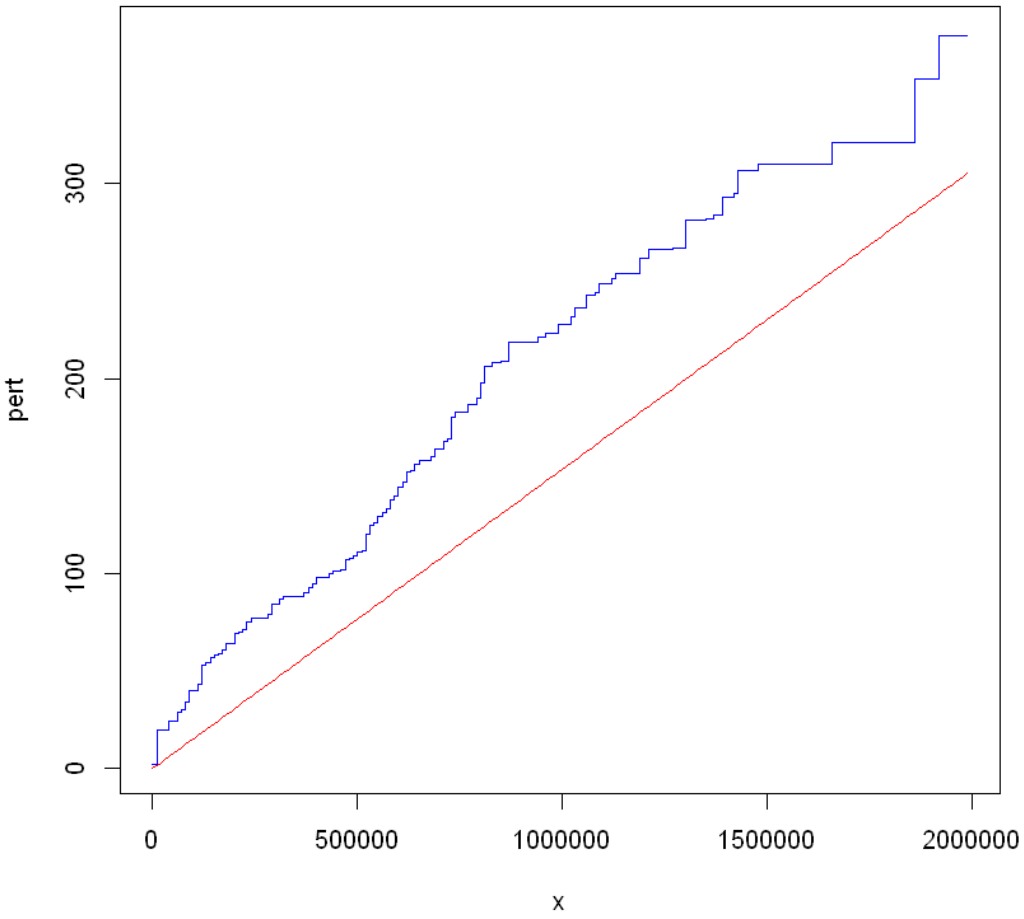
scanddtyp: 6- holiday home

***ecopuk11* (economic activity)**: 8- economically active full-time students, in employment

ageh: 14- 65-69

After using SPAD on the age-perturbed data, we were able to measure the number of introduced errors within the first ***x*** records ranked by the method and take this as a measure of the method's performance. Moreover, we plotted SPAD performances against the null hypothesis of a normally distributed error to see how much more informative the SPAD ranking is (see Figure 8).

Figure 8 Number of introduced anomalies against null hypothesis



3.5 One –class Support Vector Machine for novelty detection

One-class Support Vector Machines (SVMs) allow the detection of novelty in a set of records, while automatically identifying the number of such anomalies. Running one-class SVM on 1mln records using R's e1071 package was very slow (more than 24 hours). It was therefore decided to run SVM on packets of data, each containing 50k consecutive records. SVM was tuned automatically for each new packet of data to achieve the best results. The radial basis function proved to be optimal for Census data. The results for two packets of data are presented below to identify any consistency in anomaly detection.

In the first 50k records, one-class SVM detected 50 outliers. The introduced errors were not detected. Examples of the outliers detected include:

- 19-24 years old, economically inactive, non-student, NA *for empstat, hours* and *lastyrwrkg*, short term resident and having secondary address as other – FP.
- 90+ year olds whose activities are limited a lot, claiming to be employed and last year worked before 1991, economically inactive student and not a student at the same time.
- 0-4 year olds claiming to be students while living away from home in term time.

- 0-4 year olds simultaneously claiming to be in very bad health, but with day-to day activities are not limited.
- 85-89 year olds claiming to be full-time students and economically inactive, while at the same time being employees with the last year worked before 2001.

58 outliers were detected in the next 50,001-100,000 records, none of which were introduced errors. Examples of the detected outliers include:

- 0-4 year old students.
- 85-89 year olds in very poor health and limited activity, but still employed with last year worked before 1991.
- 10-15 year old students living away from home, but having a secondary address as other (not student accommodation).
- Older people who claimed to be employees or employers, but last year worked before 1991.

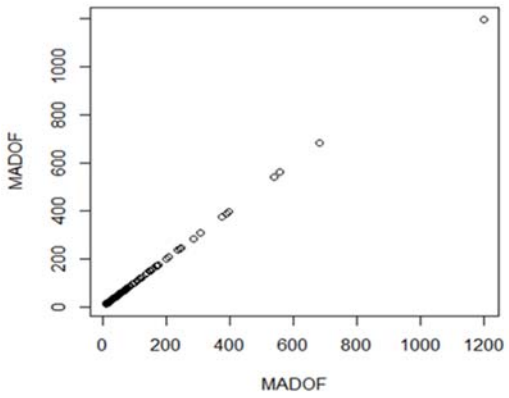
This displays a similarity between outliers which one-class SVM can detect on different packets of records: 0-4 year olds being students, older people in bad health claiming to be working, and contradictions between current employment status last year worked were all detected in both 50k packets.

3.6 Pattern Outlier Detector based on Support Vector Machines

Pattern Outlier Detector (POD) measures interactions between numerical and categorical attributes by finding a pattern among numerical attributes that holds true for a majority of objects with a particular categorical attribute value. It involves the calculation of a Mixed Attribute Outlier Factor (MADOF), which is the function of a Numerical Outlier factor and a Categorical Outlier Factor.

Based on the targeted anomaly, all variables are divided into numerical and categorical classes. In the case of detecting the introduced '198x -> 189x' errors, the variables treated as categorical were *popbasesec*, *student* and *hours*, while the remaining variables were treated as numerical. Three multi-class SVMs were trained, one for each categorical variable, using numerical variables as predictors. Outliers were selected based on n-largest values of MADOF. Since this approach is based on relationships between variables and not distribution, duplicate records including both categorical and numerical variables were removed from consideration. Figure 9 shows that MADOF is a good measure for detection of outliers.

Figure 9 MADOF as a measure of anomaly



After running POD_SVM on the first 100k records, the top 1,000 outliers contained 6 of the 11 introduced outliers present. Examples of detected anomalies in the top 10 outliers include:

- 90+ in bad health (4), but day-to-day activities are not limited, working 31-48 hours a week, and currently employed. This record has the largest MADOF.

- 35-39 years old, full-time student who is economically active, looking for job and employee at the same time, usual resident who lives at Armed Forces base.
- Older people who claim to be economically inactive, have not worked for some time, but simultaneously claim to be an employee or self-employed

In addition to using the first 100k records, we also ran POD_SVM on 50% of records from the sample data chosen at random. While the initial sample size was 1,424,074 records, removing duplicates reduced this to 53,697 unique records. POD_SVM was run overnight, as SVM slows down enormously with data size. In this case we detected 27 introduced errors out of 93 in the first 1000 outliers with the largest MADOF. The outlier examples include:

- Actual 90+ years olds (and older than 65) claiming to be students, living away from home at term time, being single, and NA for the remaining variables.
- 65-69 year olds claiming to be economically inactive/students and non-students at the same time, while being in employment despite a last year of work in 2011.
- 19-24 years old, in good health, claims to be economically inactive/retired and being employee with last year worked in 1991-1995, widowed and owning holiday home.

3.7 Pattern Outlier Detector based on Boosted Regression Trees

Boosted Regression tree (BRT) algorithms aim to improve the performance of a single model by fitting many models and combining them to make predictions [8]. BRTs are significantly faster than multi-class SVMs, while additionally considering interactions between numerical variables (8-way interactions were used in this report). Three BRTs were trained for each categorical variable.

Based on the first 100k processed records, 6 introduced errors out of 11 were present in the top 1,000 outliers based on the largest Mixed Attributes Outlier Factor (MADOF). Example of outliers detected based on largest MADOF include:

- 90+ years old, in very poor health, claims to work FT as employee and currently in employment doing 31-48 hours per week.
- 35-39 years old, in very bad health, economically active FT student and last year worked in 2000.
- 19-24 years old, in bad health, economically inactive/retired, widowed, Armed Forces address is claimed as secondary address.
- 40-44 years old, married, student living away from home, other for secondary address, and NA for all other variables.
- 45-49 years old, in bad health, retired, last year worked in 2002 or 2003 but also claiming to be employee and having another address when working away from home.

3.8 Summary of results

In this section we present the results discussed above in the form of a table (see Table 3) where we list all machine learning approaches tried and the type of contradictions each algorithm is able to detect.

Table 3. Summary of results using Machine Learning techniques

Algorithm	Based on	Type of Anomalies
Rule-based	Rules	Any type of anomaly which can be detected by pre-defined rules
K-means	Natural clusters in data	Records with NA for *disability, ecopuk11, empstat, health, hours, lastyrwrkg* for students, unexpected age
KAMILA	Natural clusters in data	Records with NA *for disability, ecopuk11, empstat, health, hours, lastyrwrkg* for students of unexpected age
		Contradiction between employment statistics and hours together with last year worked

SPAD	Frequency of attributes	Records with NA for *disability, ecopuk11, empstat, health, hours, lastyrwrkg*
		Rare records: e.g. separated but legally in same sex civil partnership
		Contradiction between age (old people) and being student living away from home
		Multi- ticks for secondary address
One-class SVM	Boundaries of classes	Contradiction between employment status and last year worked
		Contradiction between *health* and *disability*
		Contradiction between very young age and being student living away from home
		Contradiction between age (old age) and being student
POD_SVM	Interactions between variables with boundaries for numerical variables	Contradiction between old age and long working hours/employment status
		Contradiction between old age and being student
		Contradiction between employment status and last year worked
		Contradiction between very poor health and employment-related variables
		Contradiction between young age and marital status/employment-related variables
POD_BRT	Interactions between variables with additional interactions between numerical variables	Records with majority of variables NA
		Contradiction between very poor health and employment status/working hours/last year worked
		Contradiction between young age and health/employment-related variables
		Contradiction between health/employment-related variables and secondary address (Army base)

4. Performance and Scalability of Methods for Outlier and Anomaly Detection

4.1 Runtime of available code

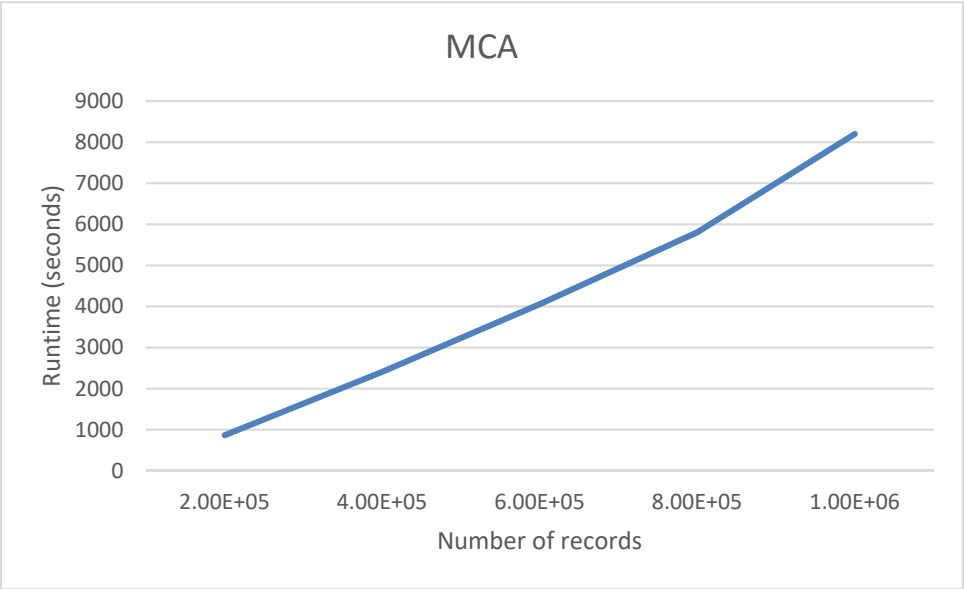
In order to assess the performance and scalability of each of the above-mentioned methods, we ran a basic benchmark where the algorithms' runtimes were measured against different input sizes. The algorithms were tested against the following input sizes: 200k, 400k, 600k, 800k, and 1mln instances. Due to time limitations, the runtimes were only measured once. The execution environment was (with one exception) kept the same, so as to accurately compare runtimes. The benchmark was run, unless stated otherwise, on a virtual machine running Ubuntu 14, with 4 GB of RAM and 8 CPUs. The runtimes are expressed in seconds.

4.1.1 Multiple Correspondence Analysis

This particular benchmark ran on a Windows 8 machine with 16 GB of memory.

The runtime of the MCA algorithm (see Figure 10) shows that the implementation has a somewhat linear behaviour.

Figure 10 Multiple Correspondence Analysis runtimes

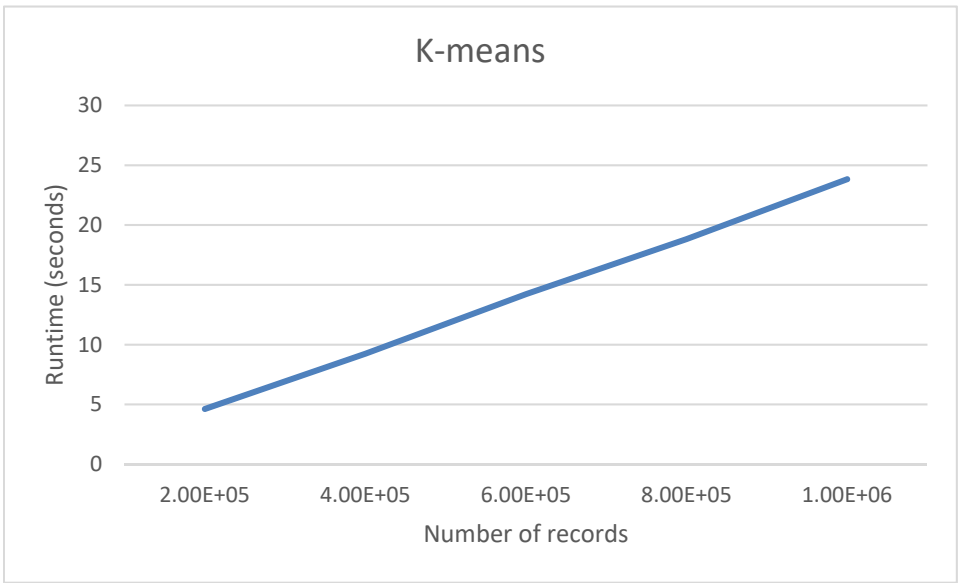


It is worth noting that these are the runtimes based on all possible variables, and a reduction of variables leads to faster runtimes.

4.1.2 K-means

The benchmark for the K-Means algorithm (see Figure 11) shows that the implementation is linearly dependent on the input size.

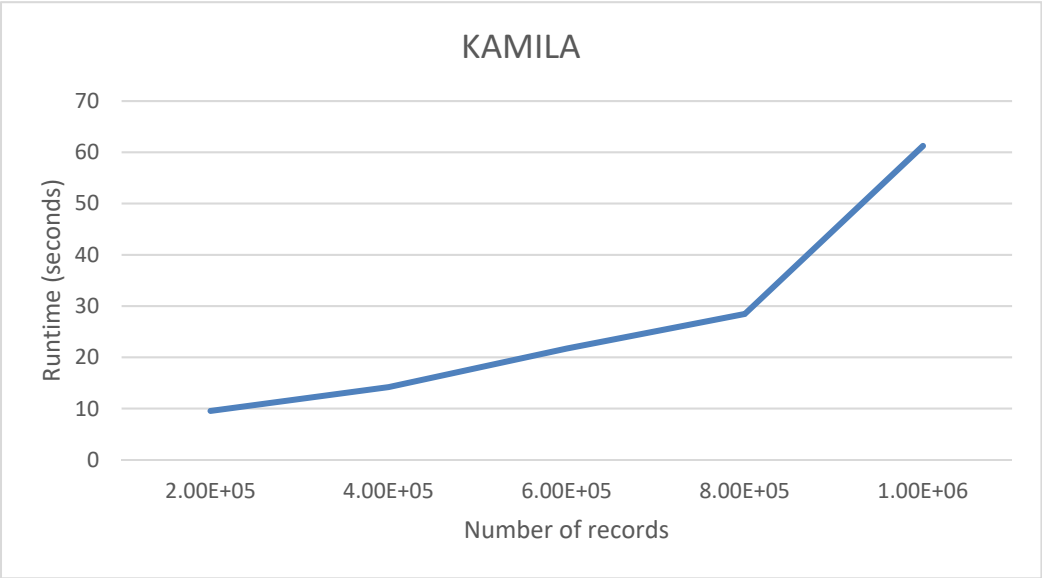
Figure 11 K-Means runtimes



4.1.3 KAMILA

The benchmark for the KAMILA algorithm (see Figure 12) shows the implementation's runtime is non-linear, and that it could increase polynomially based on the input size. However, the runtime for the last experiment (1m records) took only a minute to complete.

Figure 12 KAMILA runtimes



4.1.4 Rule-based

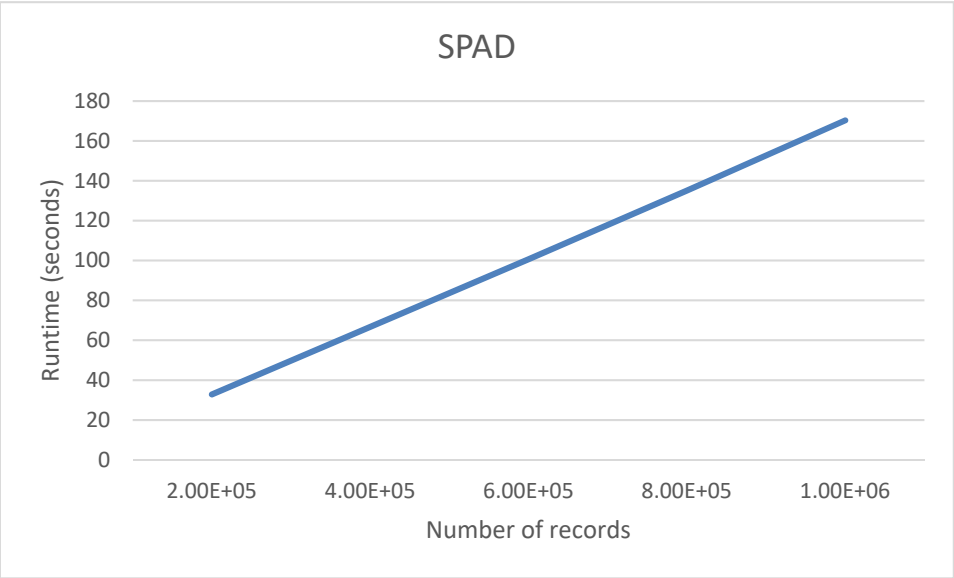
The runtime of an ad-hoc rule-based algorithm (see Figure 13) shows that the implementation has a strange behaviour. Multiple executions are thus required to reason on average times instead.

Figure 13 Rule based runtimes

4.1.5 SPAD

The runtime of the SPAD algorithm (see Figure 14) shows that the implementation is linearly dependent on the input size.

Figure 14 Simple Probabilistic Method runtimes



4.1.6 Support Vector Machine for novelty and pattern detection

The benchmark for the SVM algorithm (see Figure 15) shows the implementation's runtime is linearly dependent on the input size. The benchmark stops at 800k instances, where the execution took more than 4 hours to complete.

Figure 15 Support Vector Machines runtimes (novelty)

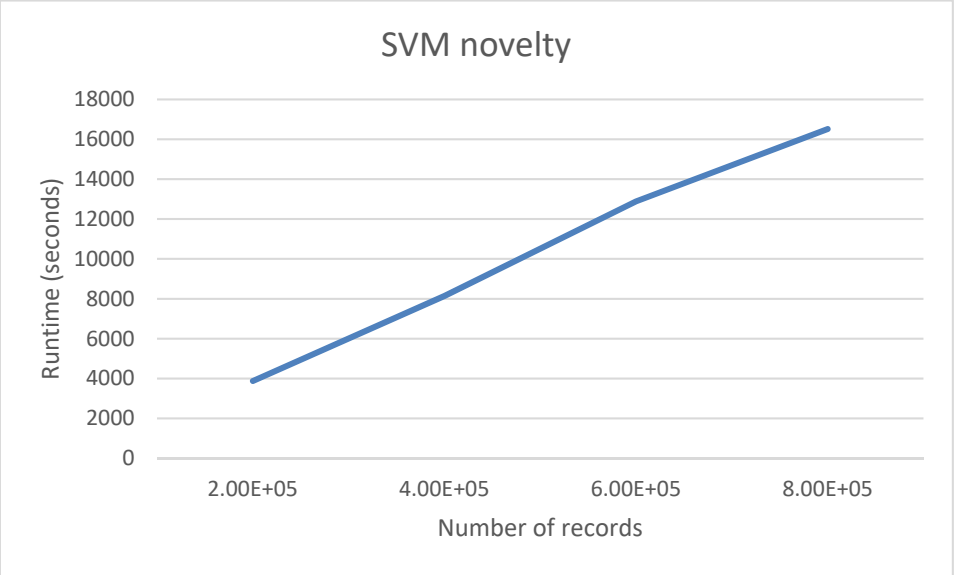
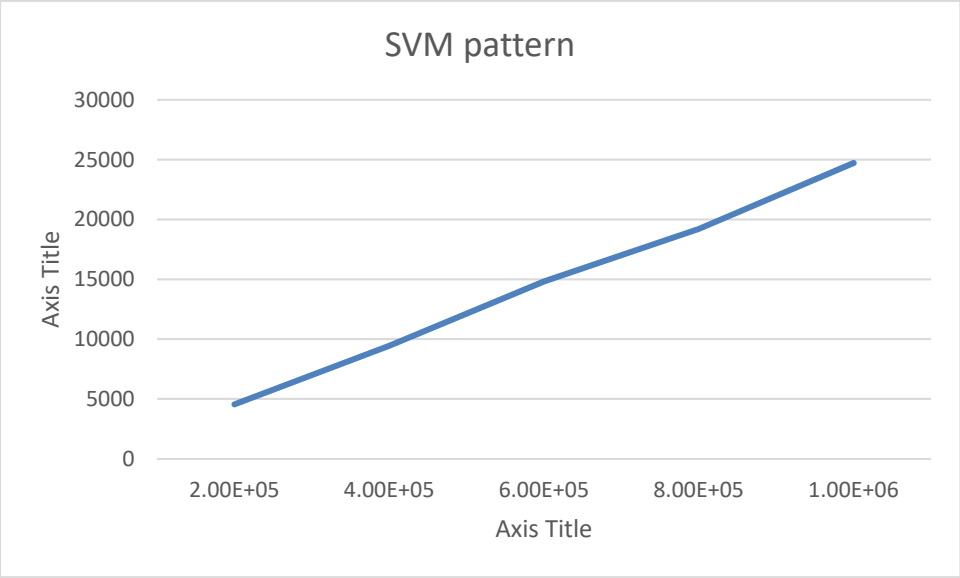


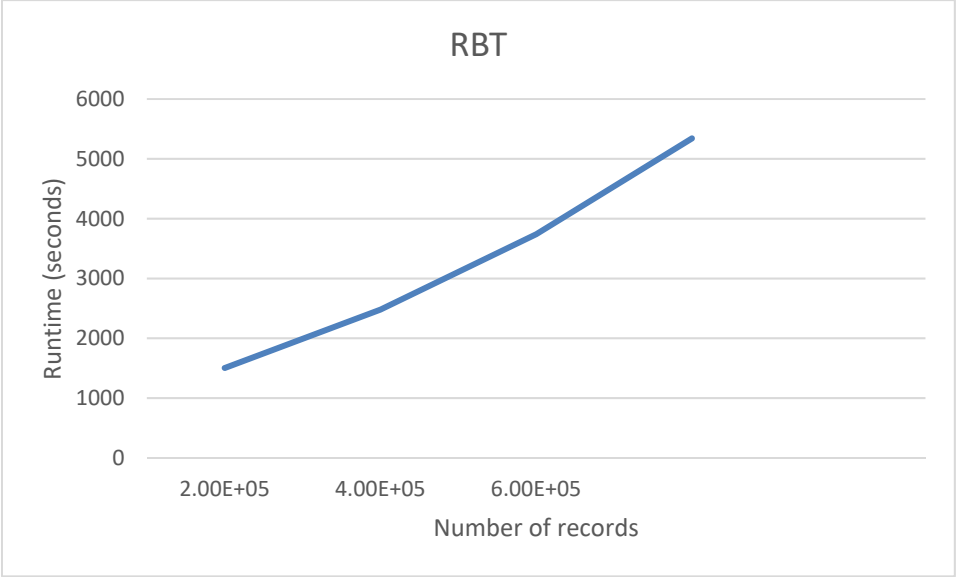
Figure 16 Support Vector Machines runtimes (pattern)



4.1.7 Boosted Regression Trees

The benchmark for the BRT algorithm (see Figure 17) shows the implementation's runtime is linearly dependent on the input size. The benchmark stops at 600k instances, where the execution took more than 1 hour to complete.

Figure 17 Boosted Regression Trees runtimes

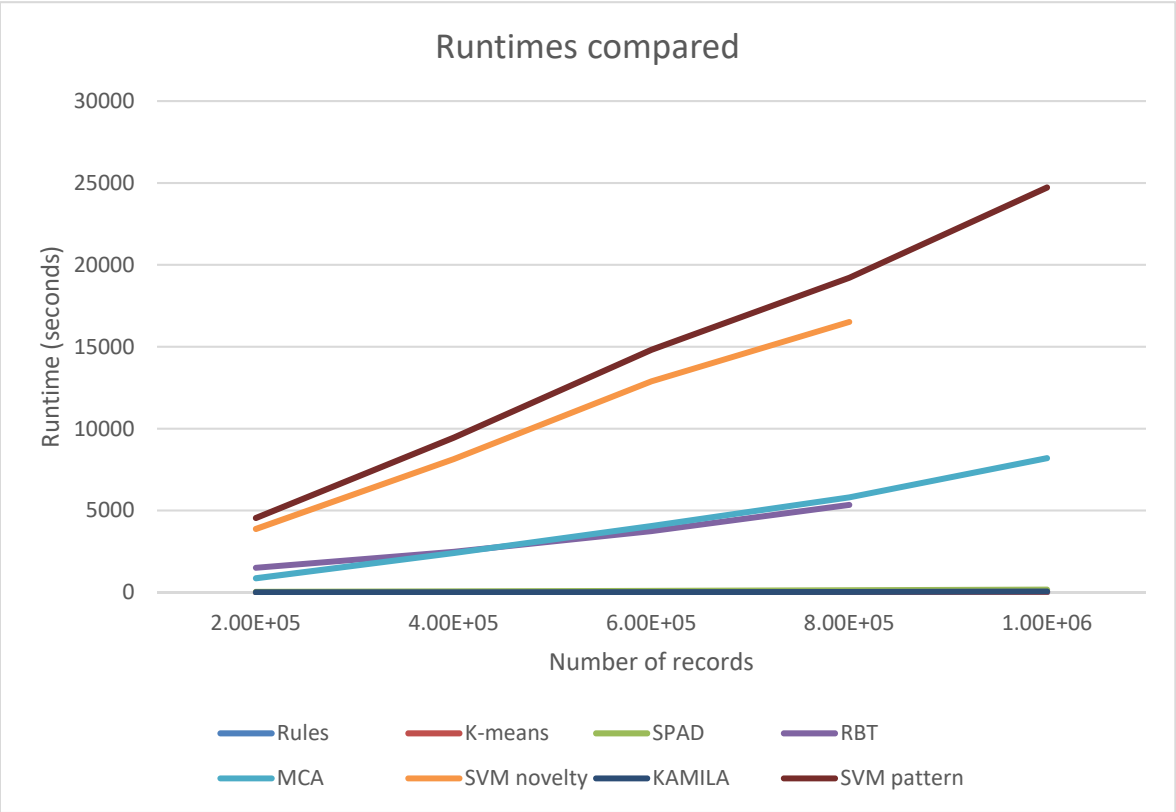


4.1.8 Methods Runtimes (Combined)

When we compare the runtimes of all methods and plot them together, we can clearly see that there are two classes of algorithms (excluding MCA, which ran in a different environment with more memory):

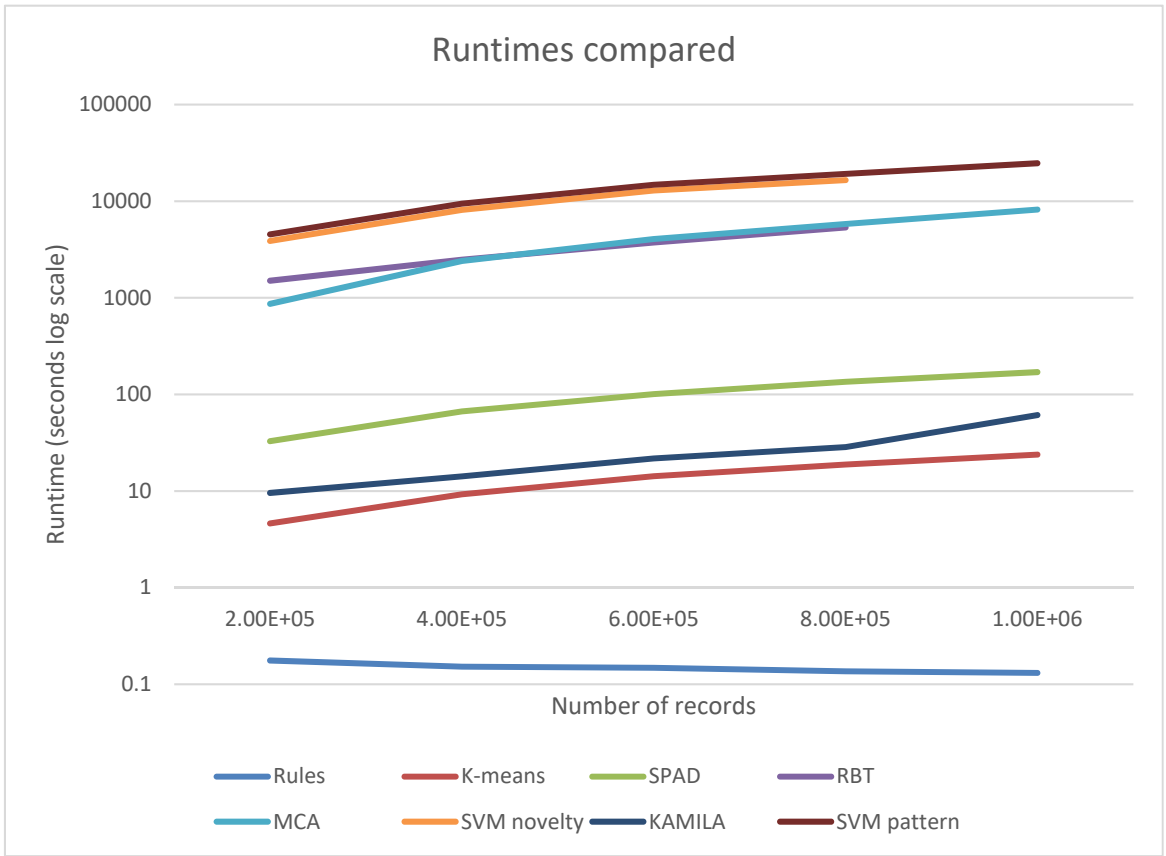
- Lighter implementations whose runtime for 1m records is under 5 minutes: Rules, K-Means, KAMILA, SPAD
- Heavier implementations whose runtime for 1m records is more than one hour: RBT, MCA, SVM

Figure 18 Runtimes compared



If we plot the runtimes in log scale we can more clearly appreciate the relative performances of each method (see Figure 19).

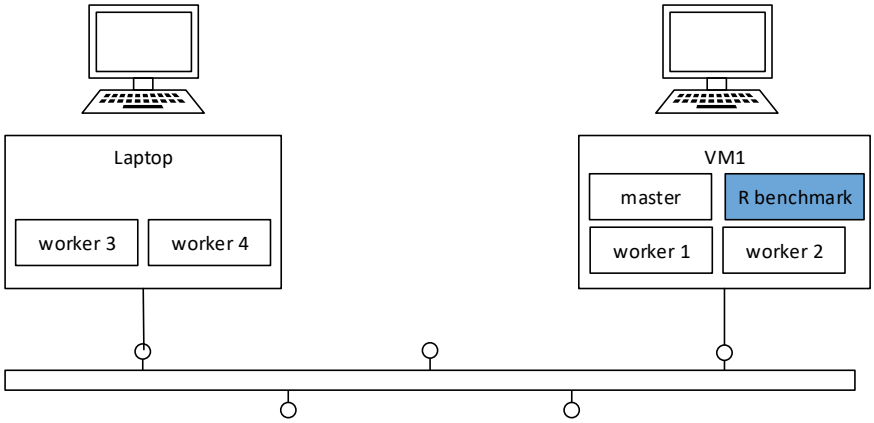
Figure 19 Runtimes compared (log scale)



4.2 Spark implementation

In order to test Spark ML implementations of some of the aforementioned methods (see Section 3.1 and 3.6) used for studying census data, a small testbed has been set up. This testbed is minimal, and is mostly relevant as a proof of concept for the ability to use spark APIs from R for studying census data, rather than as a comparison with the respective non-cloud implementations.

Figure 20 Spark testbed diagram



The testbed consists of an Ubuntu 14.04 VM with 16 GB of memory and a computer with Debian 9 and 8GB of memory. Each physical node runs two spark worker nodes (2 cores and 4GB of memory allocated). The VM also runs the spark master node, the benchmark R code and the data for the census.

The spark benchmark runs two of the methods included in the census data analysis, namely **spark.kmeans** and **svmLinear** from the **SparkR** API. These two methods are run with the census data, using all available features. The runtimes measured and reported in Figure 21 and Figure 22 are the ones to run the ML method only, with the data already uploaded into the **Spark** cluster.

Figure 21 Runtimes for **spark.kmeans**

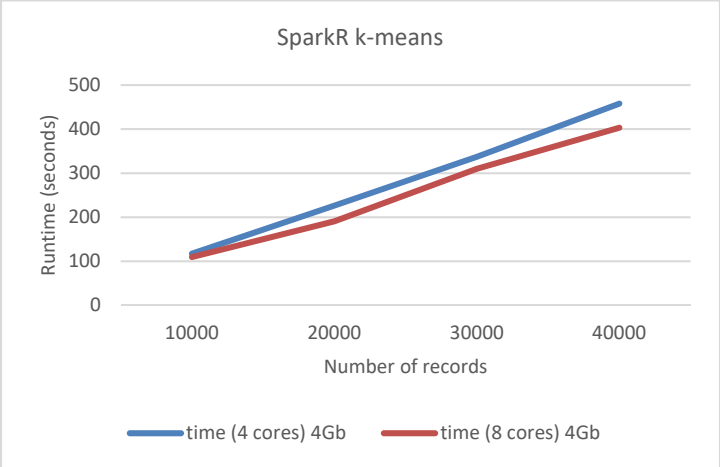
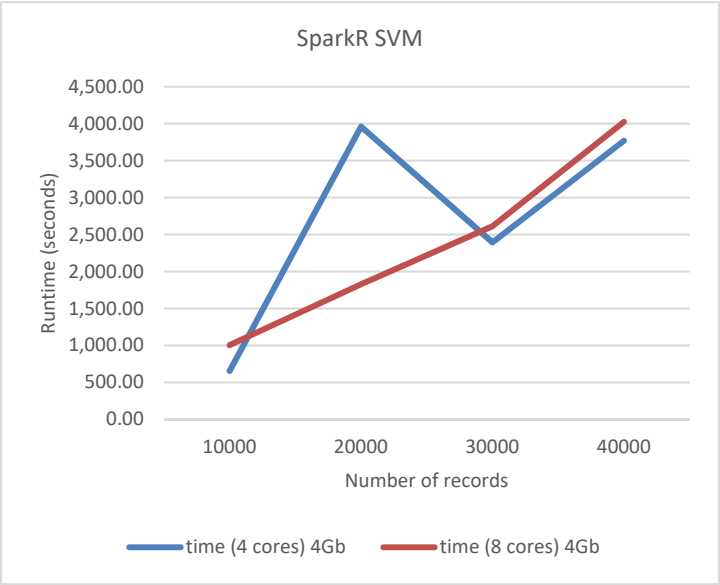


Figure 22 Runtimes for **svmLinear**



Although limited in scope, the testbed shows that using spark does not guarantee a performance boost, and that great care must be taken in planning the use of spark clusters and managing the available computing resources.

5. Integration into Jupyter

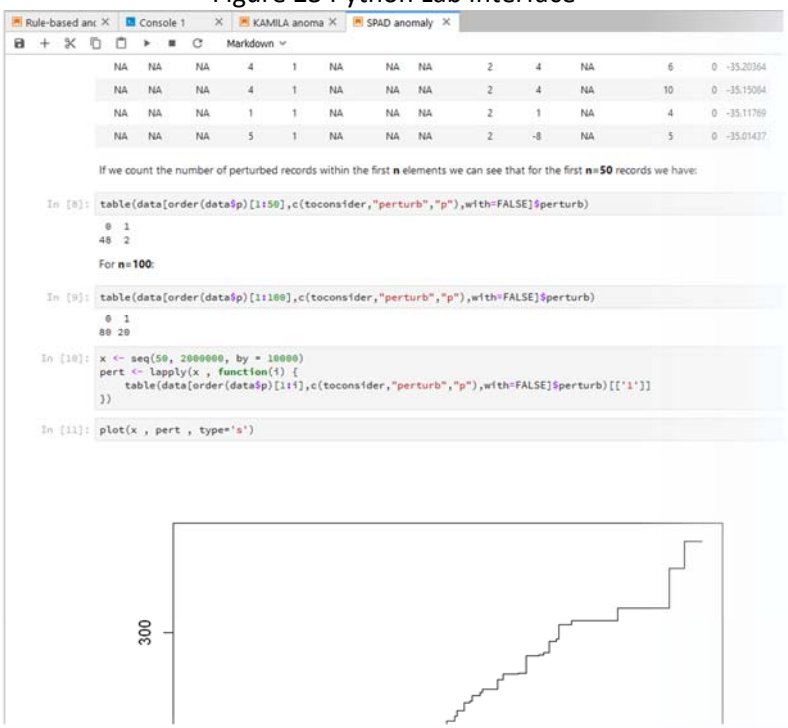
Each outlier and anomaly detection method has been investigated in a separate notebook in which textual description, tables, and graphs complemented the code used for the evaluation. The notebooks have been implemented in Jupyter Lab, running in Python 2.7.12, using R 3.3 kernels (the interface is depicted in Figure 23).

The R 3.3 kernel uses R version 3.3.2 and the necessary packages for running the methods have the following versions:

- **e1071** v1.6-8
- **gbm** v2.1.3
- **kamila** v0.1.1.1

The census data was downloaded and saved under the “data” folder and made available for all notebooks.

Figure 23 Python Lab interface



6. Conclusion and Recommendations

Following the investigation of automated anomaly detection in census data, this study has reached a set of three topics for consideration and recommendations for future studies. These are discussed below:

1- Census Data Perturbation Programmes

The introduction of errors into the census microdata proved to be extremely challenging. The available datasets [1, 2] we acquired had already been edited and processed to remove the most obvious anomalies; however, this process also altered the underlying data distributions. The cleaned data thus did not carry the true statistical signatures of the various classes of embedded anomalies. These would ordinarily manifest themselves with their specific weights (energies) during census processing. We therefore recognize that it is difficult to synthetically introduce many of the types of errors and their respective natural statistical significance which have been identified by ONS for testing anomaly detection methods.

Recommendation 1: Further anomalies should be tested. These anomalies could be introduced either at random, conditionally (based on other variables in the record), or through a two-stage simulation using latent variables to examine the effects of more challenging patterns on anomaly identification.

Recommendation 2: The promising methods should be directly tested on raw census data, where anomalies are present with their significant natural weights, and where new discoveries of correct anomaly detection rates would be expected. The methods will be expected to exhibit greater performance as they become more sensitive to detecting various classes of anomalies in the data.

2- Anomaly Detection Methods and Machine Learning Strategies

The selected detection methods demonstrated some promising results. They distinguished relatively small numbers of unusual observations (with unusual combinations of characteristics) from a larger mass of observations (with shared combinations of characteristics). It is notable that some of these unusual observations are not 'anomalies' in a strict sense, and are just unusual observations in the data. Others however do appear to be discordant, even among the published microdata. This suggests that the 2011 Census processing did not treat all unusual records. This further highlights the need for human intervention in the process, to decide which unusual observations are valid data points and which are anomalies worthy of further investigation. It is notable that our tests identified some common characteristics of anomalous records, which suggests that the methods can identify certain categories of records which are of potential concern.

Recommendation 3: the testing of selected detection methods should be extended to more comprehensive situations and patterns in census data records. The results obtained should be assessed by ONS experts regarding the credibility of the potential anomalies identified; this may indicate outlier groups which would be of concern for Census 2021.

We did not use supervised learning techniques in the exploratory assessment here. This was partly due to our use of a simple rule for introducing anomalies, as the models trained may have discovered the rule we had applied (depending on the proportion of records that were affected). A more realistic test would require some raw data with anomalies already detected. Algorithms could then be applied to identify similar cases.

The unsupervised approach does still allow the identification of unusual features in a dataset. However, it identifies other observations which, while unusual, are still legitimate. Nonetheless, it has the potential to provide candidate errors for review by ONS. It is therefore much more efficient than using experts to manually search for such errors.

Recommendation 4: A two-stage strategy is required for implementing methods on automated anomaly detection. Note that this only concerns the most promising methods identified in this study. Firstly, supervised learning (training) of known anomalies with their known patterns should be exercised. In this way, further anomalies may be discovered and their legitimacy subsequently assessed. Secondly, unsupervised learning should also be applied to automatically aid experts performing the detection of unexpected features in the data. This provides an efficient way of assessing unusualness and classifying anomalies embedded in the data. This two-stage process for identifying and classifying candidate anomalies could facilitate the creation of a more automated error detection approach for the 2021 census.

3- Scalability of the Detection Algorithms using Big Data Spark technology

The data science methods developed here for anomaly detection in census data were benchmarked under a Spark testbed experiment. This early experiment shows the necessary tests that need to be exercised for understanding the processing runtimes under increasing data volumes. Although we worked on a limited sample of data in this study, one expects much larger data volumes and complexities may be encountered in Census 2021. Such complexities may present new problems that require more advanced statistical analysis to overcome.

Recommendation 5: Further advanced Spark testbed experiments will be required to confirm the high performance of the selected detection algorithms. These experiments will also optimise the management of the big data Spark clusters and computing resources.

7. References

1. Office for National Statistics. (2015). *2011 Census Microdata Individual Safeguarded Sample (Local Authority): England and Wales*. [data collection]. UK Data Service. SN: 7682, http://doi.org/10.5255/UKDA-SN-7682-1
2. Office for National Statistics. (2014). *2011 Census Microdata Individual Safeguarded Sample (Regional): England and Wales*. [data collection]. UK Data Service. SN: 7605, http://doi.org/10.5255/UKDA-SN-7605-1
3. M. Otey, A.G., S. Parthasarathy, *Fast distributed outlier detection in mixed-attribute data sets.* Data Min. Knowl. Discov., 2006. **12**(2): p. 203-228.
4. Koufakou, A. and M. Georgiopoulos, *A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes.* Data Mining and Knowledge Discovery, 2010. **20**: p. 259–289.
5. Bouguessa, M., *A practical outlier detection approach for mixed-attribute data.* Expert Syst. Appl., 2015. **42**(22): p. 8637-8649.
6. Greenacre, M. and J. Blasius, *Multiple Correspondence Analysis and Related Methods*. 1 ed. 2006, New York: Chapman and Hall/CRC.
7. Lê, S., Josse, J. & Husson, F. (2008). *FactoMineR: An R Package for Multivariate Analysis*. Journal of Statistical Software. 25(1). pp. 1-18.
8. Elith, Jane, John R. Leathwick, and Trevor Hastie. *A working guide to boosted regression trees.* Journal of Animal Ecology 77.4 (2008): 802-813.