UNIVERSITY OF TARTU
Faculty of Social Sciences

School of Economics and Business Administration

Solomiya Branets

Detecting money laundering with Benford's law and machine learning

Master's thesis

Supervisor: Lenno Uusküla

Tartu 2019

Name and signature of supervisor.....

Allowed for defense on .....
(date)

I have written this master's thesis independently. All viewpoints of other authors, literary sources and data from elsewhere used for writing this paper have been referenced.

.....
(signature of author)

Acknowledgements

I would like to thank everyone, who contributed to this master's thesis in any way. I am sincerely grateful to my supervisor Lenno Uusküla for consistent help with the paper and revising it as many times as needed in short terms. Additionally, I would like to thank my employer Monese Ltd for providing the data and facilities to conduct the research and especially, to Märten Veskimäe for the chance to work on this topic, invaluable guidance in the idea development and implementation. I also thank my referee Nataliia Ostapenko for the detailed review of the thesis when it was not yet ready and for reviewing it second time as a favour. Special thanks to my friends Ivan Slobozhan and Pavel Tertychny for methodological suggestions, helpful advice and comments on the paper, and Gianluca Pichierri for proofreading. Finally, I thank the people who are not explicitly mentioned in this acknowledgement letter but who helped me along the way.

Abstract

The thesis develops a new tool that detects money laundering criminals and can be used by financial institutions. It builds on the basis of Benford's Law and machine learning techniques, applied to the banking data: transactions, carried out by private customers of a mobile bank. The developed algorithm is shown to outperform the traditional rule-based approach.

Keywords: anti-money laundering, benford's law, machine learning, logistic regression, random forest, xgboost

Contents

1 Introduction

Over the past 10 years, an amount of $26 billion in fines has been imposed for non-compliance with Anti-Money Laundering (AML), sanctions and Know Your Customer (KYC) regulations. Since 2018, Danske Bank has been in the spotlight, through the Estonian branch of which large amounts of money had flowed between 2007-2015. Only in 2018, regulators also fined many other banks, including Deutsche Bank, ABLV Bank Latvia, 1MDB, Goldman Sachs, Pilatus Bank, US Bancorp, Commonwealth Bank of Australia, ING, Rabobank and UBS. The main reason is a failure to detect suspicious transactions and weak anti-money laundering controls.

Banks fail to investigate thousands of alerts every month, sometimes because of a shortage of employees, sometimes because of an inability to design a proper screening system and sometimes on purpose. Currently, banks are obliged to investigate every transaction above a certain threshold and those made by sanctioned individuals, but this approach is extremely inefficient as it generates massive amounts of false alerts and is not able to catch the majority of fraudsters. One thing is clear: there is a need for a more intelligent system, that is much harder to trick than by breaking one big transfer into many small or changing a letter in a surname.

Meanwhile, past years were also rich in events and developments in Artificial Intelligence and processing power of computers has grown exponentially, which gives an opportunity to apply them to a huge amount of data that banks have and increase the accuracy of fraud and money laundering detection while reducing costs. Correspondingly, there exist old and proven methods to detect anomalies, like Benford's law, which is still widely used in finance, accounting and other areas. It states that in many naturally occurring collections of numbers, the leading significant digit is likely to be small. The number one appears as the most significant digit about 30% of the time, while nine appears as the most significant digit less than 5% of the time. Bank transaction data is likely to be a naturally occurring collection of numbers so is expected to follow the distribution, in contrast to money laundering processes.

This study combines these two areas and tests the possibility of automated money laundering detection with the help of Benford's law and modern machine learning techniques. Data analyzed is the individual transaction amounts data provided by mobile bank Monese Ltd. The dataset consists of more than 30 million transactions carried out by more than 150,000 customers that were active in the year 2018 and in January of 2019. For each customer, fractions of transactions that start with and have second digit 0,1, . . . ,9 are calculated, as well as several types of distances between observed and Benford's distribution. The dataset is highly imbalanced, containing 0.34% of money laundering criminals.

Applied techniques cover traditional supervised and unsupervised machine learning methods, such as logistic regression, random forest classifier and XGBoost classifier from the supervised basket and k-means clustering, DBSCAN and isolation forest for unsupervised exploration. In addition, under- and oversampling is performed with the help of the random undersampling method, SMOTE and SMOTENN algorithms. Moreover, data is visually explored for overall fit to the Benford's law, and transaction amounts distribution is indeed very close to the expected one, but money laundering cases follow the law to a lesser degree.

The classifier built with supervised learning achieved much better performance in comparison to the rule-based approach, having the area under precision-recall curve 0.1799. It can be used by any financial institution, is simple and requires only information about transaction amounts. In addition, clusters containing money laundering are identified.

The paper contributes to the literature in three ways. First, it shows that Benford's law is helpful for money laundering detection on the micro level in the banking field. Such approach was not yet used for banking transaction data, but just once in a legal area by Badal-Valero et al. (2018), where it has proven its value. Second, it implements latest developments in data science and demonstrates their superiority to the algorithms, used by Patil et al. (2018), Jurgovsky et al. (2018) and Bhattacharyya et al. (2011), showing that boosting in addition to bagging outperforms bagging alone due to sequential tree building and errors correction. In addition, this analysis is based on high-quality real-world data, compared to Patil et al. (2018), whose dataset consisted of just 1,000 transactions and Fiore et al. (2017), who used a publicly available dataset with features, which are the principal components resulting from principal components analysis applied to the original features.

The rest of the paper is organized as follows: Section 2 reviews fraud detection approaches and the use of Benford's law in the literature, Section 3 focuses on methodological issues. In Section 4 one can find a descriptive analysis of the dataset and finally, Section 5 discusses the empirical results of training machine learning models. Section 6 concludes the paper.

2 Literature review

2.1 Money laundering and anti-money laundering regulations

Money laundering is an act of concealing the identity or source of money obtained in an illegal way in order to make them look like they are legally obtained so it can be used without suspicion. Criminal activity includes illegal arms selling, smuggling, human trafficking, drug

trafficking, terrorist activity, bribery, embezzlement, financial crimes, etc.

There are 3 stages of money laundering[1]:

1. Placement, when the criminal introduces the income from illegal activity to the financial system. This is done by dividing the large sum of money into smaller amounts and depositing them into bank accounts or by buying financial instruments.

2. Layering, when the money introduced in the economy is then covered by a number of fund conversions, changing its form and making it difficult to find out the original source of the money.

3. Integration, in which the funds reenter the legitimate economy. The launderer might choose to invest the funds into real estate, luxury assets, or business ventures.

During the placement stage, these criminals are the most vulnerable to being caught because placing large amounts of cash in the legitimate financial system can raise officials' suspicions. Sometimes it is also the case during the layering stage when the account is very active, but the activity itself doesn't make much sense in the eyes of the official. Hence, it is crucial to understand the specific patterns used for money laundering as it will help to build strategies to identify the occurrence of such patterns.

Some of the patterns that have been identified by Palshikar et al. (2014) as suspicious include very regular ATM withdrawals, bursts of activities in a short period of time (especially in previously inactive accounts), many cash deposits that are broken down to sums below the reporting threshold and deposited in multiple banks under different names, as well as cash withdrawals from a bank in one place, re-depositing it in a bank in another place and then transferring it to a third location. Similarly, suspicious are various bank transfers from country to country, where transfers are protected by secrecy law. Some schemes imply using casinos when money is transferred from the criminal's offshore bank account to a casino in some tourist centre abroad. The casino pays the money in chips, the chips are then cashed in and the money is sent back to the criminal's domestic bank account where it can be explained as the result of good luck. If a firm launders money, that might be with over-invoicing, when goods and services are charged at a much higher rate and funds are transferred legitimately to a different company, or shell corporations that look like legitimate businesses accept money for goods and services which might not have been delivered. When the company involved in international trade of goods and services deposits large sums of cash in its domestic accounts, it should be a signal as well.

[1]Read more about money laundering on the website of The Financial Action Task Force: http://www.fatf-gafi.org/faq/moneylaundering/

Typically states as security providers are the most interested in effective anti-money laundering, so banking is one of the most regulated fields. Looking from the banks perspective, banks want to obtain and maintain their licence and to have a good reputation as a safe institution. The most important AML compliance laws are The Financial Action Task Force's (FATF) Recommendations, The United States' Bank Secrecy Act (BSA), European Union's Fourth and Fifth Anti-Money Laundering Directives, Hong Kong Monetary Authority's (HKMA) Guideline on Anti-Money Laundering and Counter-Financing of Terrorism and Monetary Authority of Singapore's (MAS) Notices on the Prevention of Money Laundering and Countering the Financing of Terrorism. According to all of them, financial institutions are obliged to establish AML compliance programs that must include the development of internal controls, designation of an AML compliance officer, an ongoing employee training program and scheduled independent audits. Internal controls should at least include requesting and verifying the client's proof of identity when establishing a business relationship, monitoring transactions made by agents under sanctions, from high-risk countries and politically exposed persons. The FATF recommendations require carrying out due diligence procedures when transaction amount exceeds 15,000 USD/EUR or international transactions exceeding 1,000 USD/EUR, while BSA obliges to report every transaction in the sum of USD 10,000 or more to the US authorities. Failure to comply with AML regulations may result in monetary fines or criminal charges.[2]

The approach by which transactions above a certain threshold are flagged is called 'rule-based approach'. Banks are free to implement some other rules, for instance, based on the number of cash withdrawals within some period of time. The problem with this approach is that it is extremely inefficient as it generates massive amounts of false positives. All flagged transactions then have to be reviewed manually, which is quite expensive, catches only a relatively small fraction of fraud cases and is not able to detect more complex fraudulent patterns. That is why some banks are looking towards modern approaches, such as Advanced Analytics and Machine Learning.

To understand the inefficiency discussed above, one can look at the rule-based statistics from Danske Bank, which is generally relevant for the whole industry. According to it, 99.5% of all cases the bank was investigating were not fraud related, the number of false alerts, generated per day is up to 1200 and the fraud detection rate is 40%.[3]

[2]For example, in the UK, failure to disclose suspicious transactions is an offense that could result in a maximum prison term of 5 years in addition to fines. Also, in 2018 Estonian police arrested 10 in connection with Danske money laundering case.

[3]Read more how Danske Bank fights fraud with deep learning and AI on:
http://assets.teradata.com/resourceCenter/downloads/CaseStudies/CaseStudy_EB9821_Danske_Bank_Fights_Fraud.pdf

The closest to money laundering detection on micro level scientific literature available is studies that focus on credit-card fraud (credit-card fraud occurs when criminals steal credit cards or use a lost or stolen one for online or offline payments), such at Patil et al. (2018), Jurgovsky et al. (2018), Bhattacharyya et al. (2011), Nami and Shajari (2018), Fiore et al. (2017). The key pattern of such type of scam is the normal card activity changing to a suspicious one. Obviously, a criminal tries to make use of the card as soon and possible before the victim realizes and blocks it. It makes sense to take a closer look at the methodology and results of these articles, in order to grasp similarities in behavioural patterns.

The most popular method among them is definitely random forest classifier, implemented by Patil et al. (2018), Jurgovsky et al. (2018), Bhattacharyya et al. (2011), Nami and Shajari (2018) and it is the case not only in fraud detection but in various fields since it usually performs the best way, is simple and efficient. Jurgovsky et al. (2018) use it as a baseline and compares to long-short term memory (LSTM) neural network. A second most popular method is logistic regression, used by Patil et al. (2018) and Bhattacharyya et al. (2011), which is also not a surprise since logistic regression is a well developed and well-performing model widely used in economics and other fields. The other methods are support vector machine, as in Bhattacharyya et al. (2011) and k-nearest neighbourhood classifier as in Nami and Shajari (2018). For fraud detection in slightly different fields, such as communications, also deep convolutional neural networks were used by Chouiekh and Ibn-Elhaj (2018).

Another dimension where methodology should be discussed is dealing with imbalanced datasets varying from 0.172% of fraudulent transactions in Fiore et al. (2017) to 30% in Patil et al. (2018). An imbalanced dataset is one where the number of observations belonging to one group or class is significantly higher than those belonging to the other classes. In this case classic machine learning algorithms tend to treat the minority group as noise and thus to show a bias for the majority class. Solutions in the literature are cost-sensitive learning proposed by Badal-Valero et al. (2018), random undersampling by Jurgovsky et al. (2018), Bhattacharyya et al. (2011), synthetic minority over-sampling technique (SMOTE) and generative adversarial network (GAN) by Fiore et al. (2017).

As far as empirical results are concerned, in order to measure how well the classifier performs, diagnostic tools such as accuracy, precision and recall are widely used. Description and explanation of all these terms and also other quality assessment methods can be found in the section that describes the methodology. The best result in identifying credit card fraud, such as recall at the level 77% and 93% precision are obtained by Patil et al. (2018) with the help of random forest algorithm, but too little information is provided about the dataset, for example by how many accounts were all 1000 transactions made (account spe-

cific variables are used), also no methods to prevent overfitting applied. Fiore et al. (2017) also did not specify by how many cardholders transactions were made and whether variables are cardholder-related since the features are the principal components resulting from principal components analysis. They report recall at the level 72% and precision varying between 91% and 97%. Nami and Shajari (2018) report 96% accuracy (with 93% baseline accuracy) after using KNN along with the dynamic random forest. In case of interesting use of convolutional neural networks by Chouiekh and Ibn-Elhaj (2018) in telecommunications, accuracy 82% is reported but was not specified how many fraudulent events are in the dataset. LSTM network by Jurgovsky et al. (2018) results in the area under precision-recall curve 0.242 for offline transactions and 0.4 for online transactions, which suggests the reverse dependence between precision and recall and is in accordance with Bhattacharyya et al. (2011), who manage o report precision 7%-61% and recall 24%-81% for different models, but always in reverse dependence.

2.2 Benford's law and its applications in various fields

This paper studies a relatively new approach, based on Benford's law, which has definite advantages over the above-mentioned methods, such as universality, simplicity and little information required.

In 1881 Canadian–American astronomer and mathematician Simon Newcomb published the 'Note on the Frequency of Use of the Different Digits in Natural Numbers', where he states that the fact that 'the ten digits do not occur with equal frequency' must be evident to those who make 'much use of logarithmic tables, and notice how much faster the first pages wear out than the last ones.' He states, that in naturally occurring numbers 'the first significant figure is oftener 1 than any other digit, and the frequency diminishes up to 9' (Newcomb, 1881). 57 years later American electrical engineer and physicist Frank Albert Benford rediscovered it in his article 'The law of anomalous numbers', where he states that 'the frequency of first digits follows closely the logarithmic relation:

$$F_a = log(\frac{a+1}{a}), \tag{1}$$

where F_a is the frequency of the digit a in the first place of used numbers. This relationship results in the digit frequencies, summarized in Table 1.

Benford (1938) performs the digital analysis on the lengths of rivers, U.S. populations, physical constants, molecular weights, entries from a mathematical handbook, the street addresses of the first 342 persons listed in American Men of Science, death rates etc and con-

Table 1: Frequency of Digits in First and Second Places

Digit	First Place	Second Place
0	0.000	0.120
1	0.301	0.114
2	0.176	0.108
3	0.125	0.104
4	0.097	0.100
5	0.079	0.097
6	0.067	0.093
7	0.058	0.090
8	0.051	0.088
9	0.046	0.085

Source: Benford, 1938

cluded the validity of the logarithmic relation. Of course, not all the datasets exactly followed the given frequencies, but at least they all followed the pattern of monotonic decline.

Since the 'rediscovery' different scientists tried to explain the law mathematically and intuitively and there are several probable versions of why the law makes sense. Whyman et al. (2016) suggests, that the higher is the integer, the longer the sequence is needed to get a specific frequency of this integer. For instance, in order to meet 'one' as a first significant digit 111 times the sequence from 1 to 999 is needed. In order to get 111 'fives', one would need a sequence of 1, ...,4999 and in order to have 111 'nines', the sequence of 8999 numbers should be taken. And obviously naturally occurring numbers tend to have a rather lower upper limit for practical reasons. Another explanation by Chang (2017), that does not really contradict the previous one states that a process which grows at a constant rate tends to stay longer at lower digits. For instance, if the price increases from 10$ to 20$ it is 100% growth, but from 80$ to 90$ is just 12,5%, which explains the intensive use of Benford's law in finance and accounting. One more proof is given by Hill (1995). He found, that if one grabs random numbers from random distributions, the digits of these numbers will conform closely to Benford's distribution. Hill names it a 'distribution of distributions'. It was also shown that Benford's law is scale-invariant and base-invariant, which means that it equally holds for different kinds of measurement units, even if one is converted to the other.

Since then Benford's law is widely used and there are many examples of its application. For examples, some studies aim to confirm its validity, like Nigrini (2015), who shows that daily stock returns have a near-perfect fit to Benford's Law or Hickman and Rice (2010), who prove that US crime statistics follow the law at national and state level, but do not follow at

local level and different types of crime shows different conformity. According to Hindls and Hronova (2015), digits of national accounts data of the Czech Republic in 2013 follow very well, and so does reported financial data of 10 industries across 6 countries in 2000-2014, except for some issues with first significant digit 1, as shown by Shi et al. (2017). Likewise, Castellano et al. (2016) demonstrate that the first digit of financial annual single reports in the year 2012 is Benford distributed, with revenues violating it the most, as well as daily changes in sovereign credit default swaps quotes, that overall show much evidence of conformity, but the Law is rather systematically violated in the case of most liquid products, according to Henselmann et al. (2012).

Other researchers use it for investigating the quality of the data, testing it for abnormalities and identifying fraud or manipulations, for example, de Marchi and Hamilton (2006) detect problems in survey data about the emission of different chemicals by comparing them with measurement data reported by the environmental protection agency and Benford's distribution. Such analysis shows that monitored chemical concentrations follow a monotonically decreasing distribution, as well as self-reported data, except for lead and nitric acid - two heavily regulated chemical plants. While checking survey data from rural households, Judge and Schechter (2007) discover that data from the United States seems to conform better than the data from developing countries and such information as income, number of animals owned and hectares of land owned follows the Law closer than crop data, which is less important to know exact amounts by heart.

According to the initial scrutiny of electoral data on vote counts in officially published voting units in USA, Puerto Rico and Venezuela, performed by Pericchi and Torres (2012), the second digit law is compellingly rejected only in the Venezuelan referendum and only for electronic voting units, all the manual elections show support for the second digit of the Law. Moreover, the USA 2004 elections show a remarkable fit to the first digit of Benford's law. The law was also used by Rauch et al. (2011) for investigating the quality of deficit data reported to Eurostat by EU member states, which shows that the aggregate data conforms well with Benford's law, but Greece having the highest deviation. Analysis of European micro income data by Villas-Boas et al. (2017) demonstrates overall conformity, but central European countries conform better than eastern European, Austria is the closest to Benford's law and Greece the furthest, followed by Ireland and Slovakia and it makes sense because all three of them have been facing economic-financial problems recently. As shown by Castellano et al. (2016), one of the findings of monitoring daily changes in sovereign credit default swaps quotes is that Greece again follows a different path compared to the other European countries and the data may have been objects of "manipulation" after 2010.

Analysis of municipal income tax size distributions in Italy by Ausloos et al. (2017) show rather questionable concordance between income taxes of Italian regions and the theoretical statement of Benford's law, but there are discrepancies at a regional level, which are in line with the heterogeneous nature of Italian regions under a socio-economic point of view. Grammatikos and Papanikolaou (2016) test the presence of fraudulent practices in the U.S. banking industry by analyzing various variables from financial reports and reveal the largest deviations only between the expected and the actual ROA and ROE occur in the crisis period. Check for anomalies in yearly aggregated tax income data of all the Italian municipalities by Mir et al. (2014) surprisingly for authors shows excellent aggregated compliance with Benford's law. Diekmann (2007) demonstrates published regression coefficients to be approximately Benford distributed or at least follow a pattern of monotonic decline, but the first digits of fake data also exhibit a pattern of monotonic decline, while second, third, and fourth digits are distributed less in accordance with Benford's law. Cho and Gaines (2007) also apply the Benford's law for investigating statistics of political campaign financing, whose fit gets worse over the years.

All the above-mentioned datasets allow to generally predict whether data manipulation took place or not, but do not allow to identify the exact manipulator since one agent is basically one data point in the whole dataset. There is only one study aiming at catching money laundering by analyzing operations of the Spanish firm and its suppliers, some of which are proven to be money launderers. Badal-Valero et al. (2018) achieve the area under the ROC curve at the level of 0.789.

As far as methodology is concerned, one obviously starts with calculating the frequencies and comparing them to the expected ones visually or in a tabular form. The next step is statistically measuring conformity of the observed distribution to the 'expected' according to Benford. χ^2 goodness of fit test is often widely used for this purpose, for instance by Diekmann (2007), de Marchi and Hamilton (2006), Judge and Schechter (2007), Pericchi and Torres (2012), Durtschi et al. (2004), Rauch et al. (2011), Hindls and Hronova (2015), Shi et al. (2017), Villas-Boas et al. (2017), Castellano et al. (2016), Ausloos et al. (2017), Henselmann et al. (2012), Grammatikos and Papanikolaou (2016), Mir et al. (2014). Test statistics is calculated by the formula (2) and accordinate p-value is taken for 8 degrees of freedom.

$$\chi^2(P_e, P_o) = \sum_{i=1}^{9} \frac{(p_o^i - p_e^i)^2}{p_e^i}, \tag{2}$$

where p_o^i is the observed proportion of a particular digit in the dataset and p_e^i is the expected proportion of a particular digit according to Benford's law.

This test is very sensitive to slight divergences from the Benford's distribution and tends to reject the null hypothesis that two datasets come from the same distribution even if they are fairly similar. Even Benford's calculations show a different level of convergence for different datasets and this is the reason why some researchers, like Torres et al. (2007) and Hickman and Rice (2010), are just satisfied with just observing a pattern of monotonic decline. However, one can also find a decent amount of other methods of measuring the distance between the observed and Benford's distributions in the literature. For instance, χ^2 - statistics divided by sample size, calculated by Rauch et al. (2011) and Grammatikos and Papanikolaou (2016). Many authors, such as Hickman and Rice (2010), Hindls and Hronova (2015), Henselmann et al. (2012), Grammatikos and Papanikolaou (2016) and Badal-Valero et al. (2018) instead of just frequencies, used a so-called digital Z-test, outlined by Nigrini, which takes into consideration sample size:

$$Z_i = \frac{|p_o^i - p_e^i| - (\frac{1}{2n})}{\sqrt{p_e^i \frac{1-p_e^i}{n}}}. \tag{3}$$

Denominator indicates the standard deviation for a particular digit, n is the number of observations and the term $\frac{1}{2n}$ is a continuity correction factor and is used only when it is smaller than the absolute value term.

Other methods are Kolmogorov-Smirnov test by de Marchi and Hamilton (2006) and Rauch et al. (2011), Kuiper's modified Kolmogorov-Smirnov goodness-of-fit test by Judge and Schechter (2007) and Rauch et al. (2011), Chebyshev's distance or maximum absolute difference by Judge and Schechter (2007), Shi et al. (2017) and Castellano et al. (2016), which helps to spot extreme deviations and is calculated the following way:

$$d_c(P_o, P_e) = \max_{i=1\ldots9} |p_o^i - p_e^i|. \tag{4}$$

Castellano et al. (2016) measured Kullback and Leibler's divergence:

$$d_{KL}(P_o, P_e) = \sum_{i=1}^{9} p_o^i ln(\frac{p_o^i}{p_e^i}), \tag{5}$$

which is the expectation of the log difference between the probability of data in the original distribution with the approximating distribution. In other words, it measures how much information is lost while approximating two distributions.

Judge and Schechter (2007), Shi et al. (2017) and Villas-Boas et al. (2017) calculate well-known and intuitive Pearson correlation coefficient between the empirical proportions of first digits in the data and those predicted by Benford. There are a few different approaches with

Euclidean distance, which is one of the most basic and widely used distance measures:

$$d_e(P_o, P_e) = \sqrt{\sum_{i=1}^{9}(p_o^i - p_e^i)^2}, \tag{6}$$

as calculated by Judge and Schechter (2007). Normalized Euclidean distance by Rauch et al. (2011) looks like following:

$$d^*(P_o, P_e) = \frac{\sqrt{\sum_{i=1}^{9}(p_o^i - p_e^i)^2}}{\sqrt{\sum_{i=1}^{8}(p_e^i)^2 + (1 - p_e^9)^2}}. \tag{7}$$

It is euclidean distance divided by the maximum possible distance, which is also in the studies of Judge and Schechter (2007), Cho and Gaines (2007) and Shi et al. (2017). Advantage of such approach is that division by the maximum value converts the distance to a score bounded by 0 and 1. Mean Absolute Difference (MAD) was measured by Hindls and Hronova (2015) and Henselmann et al. (2012):

$$d_{mad}(P_o, P_e) = \frac{\sum_{i=1}^{9}|p_o^i - p_e^i|}{9}, \tag{8}$$

which is intuitive and interpretive, since shows how far on average each digit is from expected distribution. According to Judge and Schechter (2007), Rauch et al. (2011), and Shi et al. (2017), one can calculate the absolute value of the difference between the average of the empirical FSD distribution μ_o and the average of Benford's FSD distribution (μ_e = 3.4402) divided by the maximum possible difference:

$$d_a(P_o, P_e) = \frac{|\mu_o - \mu_e|}{9 - \mu_e}, \tag{9}$$

which is most often in disagreement with other measures and tests. Entropy measure might be used as by Villas-Boas et al. (2017):

$$E(P_o) = \frac{1}{9}\sum_{i=1}^{9} ln(p_o^i), \tag{10}$$

which does not take into account expected distribution, but only measures the entropy of observed one.

Diekmann (2007) conducted an experiment in which subjects were asked to fabricate data

and compared the real reported and fake data conformity to Benford's law and Badal-Valero et al. (2018) even made an empirical test based on simulation, which calculates the degree of global fit of the data eliminating the sample size effect. The idea is to draw B samples from the Benford's distribution with the same size as our actual sample. Secondly to compute for each of these new samples the χ^2-distance to the expected distribution. And finally to calculate the p-value as the proportion of times the sample B distances computed exceeds the χ^2-distance obtained from the observed sample. A huge advantage of such approach is that it accounts for sample size and randomness when the sample is relatively small.

Some of these distance metrics are adopted in the empirical part (see Sections 3 and 5) of this study and verified whether they make any use for money laundering detection. The hypothesis to be tested in practice is that money laundering activity is not naturally originated and thus is not distributed in accordance with Benford's law.

3 Methodology

In this Section, I discuss methodological questions: how to measure the conformity to Benford's law in our case, what computational methods to use for classification and how to deal with the high imbalance of the dataset.

3.1 Distance measures

As seen above, there is plenty of different distance measures in the literature. Taking into account computational constraints, we chose the basic distance from each 'block', that resulted in the following distances between observed and 'expected' distribution to be measured for each customer:

- χ^2- distance, as in Equation 2;
- Euclidean distance, as in Equation 6;
- mean absolute distance, as in Equation 4;
- maximum absolute difference, as in Equation 8;
- Pearson correlation coefficient ;
- Kullback and Leibler's divergence, as in Equation 5.

All the distances are measured for first and second digit distributions separately. Worth mentioning, that standardization or normalization is not required here, since all the numbers are fractions, that belong to an interval from zero to one.

3.2 Machine learning methods

Machine learning is a field of artificial intelligence describing algorithms which are able to learn from data and therefore adapt their behaviour. The beauty of machine learning is that it is able to learn without being explicitly programmed. Based on the methodology there are 2 main groups of algorithms in machine learning field: traditional machine learning and deep learning. Deep learning is a subset of machine learning, in which artificial neural networks adapt and learn from vast amounts of data.

With respect to the type of task, machine learning algorithms are divided into three categories: supervised learning, unsupervised learning and reinforcement learning. Supervised learning algorithms build a model during a training phase in which they receive the input data and the corresponding output data. Datasets that contain information one is trying to predict are called labelled. Once they have been trained, those algorithms should subsequently be able to predict accurate outputs using unseen input data only. Thus, the aim of those algorithms is to learn an accurate way to match input data to output data. Linear regressions, logistic regressions, decision trees are examples of supervised learning. Unsupervised learning, on the contrary, makes use of unlabelled data by trying to achieve various goals. One may look for hidden patterns, try to cluster similar data points together or seek outliers in a dataset. Reinforcement learning is somewhat different from the two - it targets the learning of a decision process by presenting to the algorithm an environment in which it can perform a set of actions leading to a final reward. The agent is learning by trial and error (minimizes the cost function) using feedback from its own actions and experiences.

For this study the data is labelled, meaning that each customer is marked as *genuine* or *fraudulent*, so I am going to apply traditional supervised machine learning. Nonetheless, the labels might be biased towards the majority class (see data description in Section 4), so it also makes sense to discover internal patterns and similarities with the help of unsupervised learning in order to find criminals who might not have been caught yet.

Among supervised methods logistic regression, random forest and XGBoost algorithms are going to be implemented. Logistic regression and random forest were selected because they are widely in use in similar literature (see Section 2) and XGBoost is proven in practice

to be the best performing classifier for different data science problems.[4]

Taking a closer look at them, one can notice that logistic regression is based on the concept of probability while random forest and XGBoost are tree-based algorithms. So, logistic regression is a linear model for classification that aims to find such weight vector w that maximizes the likelihood of a heuristic model to be the 'real' one. It measures the relationship between dependent and independent variables by estimating probabilities using a logistic function (cumulative logistic distribution). The log-likelihood function of logistic regression is as below:

$$logL(x,y;w) = \sum_{i=1}^{n} [y_i log(p(x_i;w)) + (1-y_i) log(1-p(x_i;w))], \tag{11}$$

where x_i - predictors

w - coefficients

$p(x_i;w)$ - prediction

y_i - observed label $\in \{0,1\}$.

The cost function we want to minimize is the opposite of the log-likelihood function:

$$J = \sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1), \tag{12}$$

where now $y_i \in \{-1,1\}$ and c is the intercept.

It might be used with regularization term to prevent overfitting by making the coefficients smaller (overfitting is the case where the the model fits perfectly on the training examples, but does badly on the test examples. The reason behind is that the model is too complicated and learns noisy patterns in training data):

- L1 penalty:

$$\min_{w,c} ||w||_1 + C(\sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1))$$

- L2 penalty:

$$\min_{w,c} \frac{1}{2} w^T w + C(\sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1))$$

where C is inverse of regularization strength. Regularization [5] adds an additional cost to our

[4] Read why does XGBoost win "every" machine learning competition on http://tiny.cc/o7m46y

[5] A regression model that uses L1 regularization technique is called "lasso regression" and model which uses L2 is called "ridge regression".

cost function that increases as the value of the parameter weights w increase. One can think of it as adding bias if the model suffers from high variance (overfits the training data).

While logistic regression fits a single line to divide the space into two, a decision tree bisects the space into smaller and smaller regions. So, when two classes are separated by a non-linear boundary, the trees can capture the division better. A decision tree is a flowchart, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label. A tree is built by splitting the dataset into subsets based on attribute values. This process is repeated on each derived subset and when all the data points in subset at a node have the same value of the target variable the recursion is completed. Decision trees have a nice "if ... then ... else ..." construction which makes it fit easily into a programmatic structure. Example of a decision tree is depicted in Figure A-1 in Appendix A. Decision tree based algorithms are considered to be one of the best and mostly used supervised learning methods. Methods like random forest and gradient boosting tree-based algorithms are being popularly used in all kinds of data science problems.

The problem with a decision tree is vulnerability to overfitting and random forest, proposed by Breiman (2001), was designed to solve it. Random forest builds multiple decision trees in parallel and merges them together to get a more accurate and stable prediction. Combined with bagging, developed by Breiman (1996), it randomly selects observations and features for each decision tree, which helps to reduce overfitting significantly. A schematic illustration of random forest classifier is in Figure A-2 of Appendix A. Yet, nowadays, the most popular method is random forest combined with bagging and boosting (for example XGBoost). Gradient boosting was first proposed by Freund and Schapire (1996) and the idea is to train a classifier sequentially, each trying to correct its predecessor and then also merge them together. After each iteration the residuals are calculated and those data points that have high residuals are assigned a higher probability to be selected for the next tree.

Among unsupervised methods k-means clustering, proposed by Lloyd (1982), DBSCAN, proposed by Ling (1972), and isolation forest, proposed by Liu et al. (2009), are going to be implemented. K-means is chosen since being quite popular in the literature and successfully used by Nami and Shajari (2018), while Isolation forest and DBSCAN are relatively new techniques, gaining attention. Advantage of DBSCAN is that it does not require one to specify the number of clusters, it can even find a cluster surrounded by a different cluster. K-means tends to find clusters of similar density, so we would expect it to not be very precise while DBSCAN might work in 'high precision' areas. Isolation forest was introduced quite recently and seems to be promising in anomaly detection.[6]

[6]See the presentation at PyData London 2018 conference: http://tiny.cc/30f56y

Looking closely at them, one can see that the k-means clustering algorithm attempts to split a given data set into a fixed number (k) of clusters. Initially, k number or so-called centroids are chosen. Each centroid is an existing data point in the given input data set, picked at random, such that all centroids are unique. Then it assigns each data point to the closest corresponding centroid, using the standard Euclidean distance. After that for each centroid, the mean of the values of all the points belonging to it is calculated and the mean value becomes the new value of the centroid. The process of classification and centroid adjustment is repeated until the values of the centroids stabilize. The final centroids will be used to produce the final clustering of the input data.

DBSCAN stands for density-based spatial clustering of applications with noise and works in a completely different way: first the user chooses two parameters, a positive number epsilon and a natural number minPoints. The algorithm begins by picking an arbitrary point in the dataset. If there are more than minPoints points within a distance of epsilon from that point, (including the original point itself), it considers all of them to be part of a ”cluster”. It then expands that cluster by checking all of the new points and seeing if they too have more than minPoints points within a distance of epsilon, growing the cluster recursively if so. Eventually, it runs out of points to add to the cluster. Then it picks a new arbitrary point and repeats the process. Now, it's entirely possible that a point that is picked has fewer than minPoints points in its epsilon ball, and is also not a part of any other cluster. If that is the case, it's considered a ”noise point” not belonging to any cluster.

Isolation forest is completely different from k-means and DBSCSAN and is built on the basis of decision trees. In the tree, each split is based on selecting a random variable, and a random value of that variable. Then each observation is given an anomaly score based on how long ‘path’ through the tree it takes for this observation to travel. The shorter the path - the more likely it to be an anomaly. The parameter to be arbitrarily chosen is the number of trees to build and contamination - the proportion of outliers in the data set, which is used when fitting to define the threshold on the decision function.

Before moving on and applying chosen machine learning algorithms one has to bear in mind that almost all datasets containing fraud are highly imbalanced and there is a need to do something about it.

3.3 Imbalanced dataset

Almost all supervised machine learning methods, when trained on a highly imbalanced dataset, have a natural tendency to pick up the patterns in the popular classes and ignore the minor ones, since it's the easiest way to achieve the requested metric. That is why under-

sampling and oversampling techniques are in use. Following the literature in the field we are going to apply:

- cost-sensitive learning
- random undersampling
- synthetic minority oversampling technique (SMOTE)
- SMOTEENN (SMOTE and edited nearest neighbours (ENN))

To explain briefly, cost-sensitive learning means that in the loss function higher value is assigned to data points from minor class. Hence, the loss becomes a weighted average, where the weight of each sample is specified. According to X Ling and Sheng (2010), this technique is expected to improve model performance. Random Undersampling does not make any changes in the loss function but balances the dataset by reducing the size of the major class. By keeping all samples in the rare class and randomly selecting a number of samples in the major class, a balanced new dataset can be used for further modelling. For instance, Jurgovsky et al. (2018) implied selecting a random subsample from a major class with 0.9 probability to be picked and 0.1 from minor class, which worked out well.

Synthetic minority oversampling technique (SMOTE) works the other way around: it creates new data points from a minor class. In order to create a synthetic data point, it takes the vector between one of its k neighbours, and the current data point and then multiplies this vector by a random number which lies between 0, and 1. Then the synthetic data point is obtained by adding this to the current data point. The performance of these two sampling methods is compared by Mishra (2007), who shows that random undersampling gives the most balanced results with a good tradeoff between specificity and sensitivity. The issue with SMOTE is that it can generate noisy samples by interpolating new points between marginal outliers and inliers. This problem can be solved by cleaning the space derived from over-sampling and results in SMOTEENN algorithm.

SMOTEENN performs over-sampling using SMOTE and under-sampling using edited nearest neighbours (ENN). The ENN method removes the instances of the majority class whose prediction made by KNN (K-nearest neighbourhood) method is different from the majority class. ENN method can remove both the noisy examples as borderline examples, providing a smoother decision surface. Mduma et al. (2019) conduct their experiment using SMOTEEN and random undersampling and show the superiority of the latter.

3.4 Quality assessment metrics

Several evaluation metrics are widely used for quality assessment, such as accuracy, area under the ROC (Receiver Operating Characteristic) curve, and area under the precision-recall curve. One of the most universal metrics is accuracy. It simply measures the number of correctly predicted samples over the total number of samples. Using the same notations as Juba and Le (2017):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{13}$$

where True Positive (TP) predicts an event when there was an event, True Negative (TN) predicts no event when in fact there was no event, False Positive (FP) predicts an event when there was no event and False Negative (FN) predicts no event when in fact there was an event.

Accuracy can be misleading if the number of samples per class in the data is unbalanced. For example, for a dataset with two classes, where the first class is 90% of the data, and the second completes the remaining 10%, if the classifier predicts all dataset as belonging to the first class, the accuracy reported will be of 90% while this classifier is in practice useless. So, measuring accuracy makes sense only when the class labels are uniformly distributed and sometimes accuracy just does not give enough insight about the model performance. In this case, such metric as the area under the ROC curve to be considered.

Receiver operating characteristic (ROC) curve is a plot of false positive rate:

$$FPR = \frac{FP}{FP + TN}, \tag{14}$$

on the x-axis versus the true positive rate:

$$TPR = \frac{TP}{TP + FN}, \tag{15}$$

on the y-axis for a number of different threshold values. This metrics also works well for balanced datasets, since it gives equal value for both classes, as shown by Fawcett (2006).

In contrary, for highly imbalanced datasets, according to Davis and Goadrich (2006), the precision-recall curve is the most relevant metric. Precision is the fraction of instances of interest among the retrieved instances:

$$Precision = \frac{TP}{TP + FP}, \tag{16}$$

while recall is the fraction of relevant instances that have been retrieved over the total amount

of relevant instances in the dataset:

$$Recall = \frac{TP}{TP+FN}. \tag{17}$$

Hence, a Precision-Recall curve is a plot of the Recall on the x-axis and the Precision on the y-axis for different thresholds, similar to the ROC curve. The fact that in computing precision and recall there is never use of the true negatives makes it suitable for imbalanced dataset. [7]

Sometimes precision, recall or combination of them, as well as sensitivity:

$$Sensitivity = \frac{TP}{TP+FT}, \tag{18}$$

and specificity:

$$Specificity = \frac{TN}{TN+FP}, \tag{19}$$

are reported separately for one threshold (usually 0.5), but areas under the curves are definitely more informative, since they take into account all possible thresholds.

4 Data

Dataset is comprised of anonymized individual customer transaction amounts data, that was kindly provided by Monese Ltd, a British startup that offers personal bank accounts. A large sample set consists of 31,766,662 transactions carried out by 172,260 customers that were active in 2018 and in January of 2019. Money laundering criminals account for 0.34% of the customers. The labels might be somewhat noisy due to the specifics of business processes, a backlog of AML investigators or delays in crime confirmation from the authorities. Sometimes money laundering is not discovered at all since there is no victim to report the crime.

Before going into feature engineering and modeling, I check if the first and second digits of all transactions are Benford distributed. The visual representation of the first significant digit distribution is depicted in Figure 1 and of the second significant digit in Figure 2.

First of all, the first significant digit shows a monotonic decline, as predicted by Benford. Moreover, its fit is quite close to an expected one. According to Benford, digits '1','2' and '3' should account for 30.1%, 17.6% and 12.5% respectfully and the observed numbers almost perfectly match: 31.2%, 17.5% and 12,7%. As for the rest of the digits, it is expected that the number should slowly decrease from '4' accounting for 9.7% till 9 accounting for 4.6% while

[7] Read more about evaluation metrics on http://www.davidsbatista.net/blog/2018/08/19/NLP_Metrics/

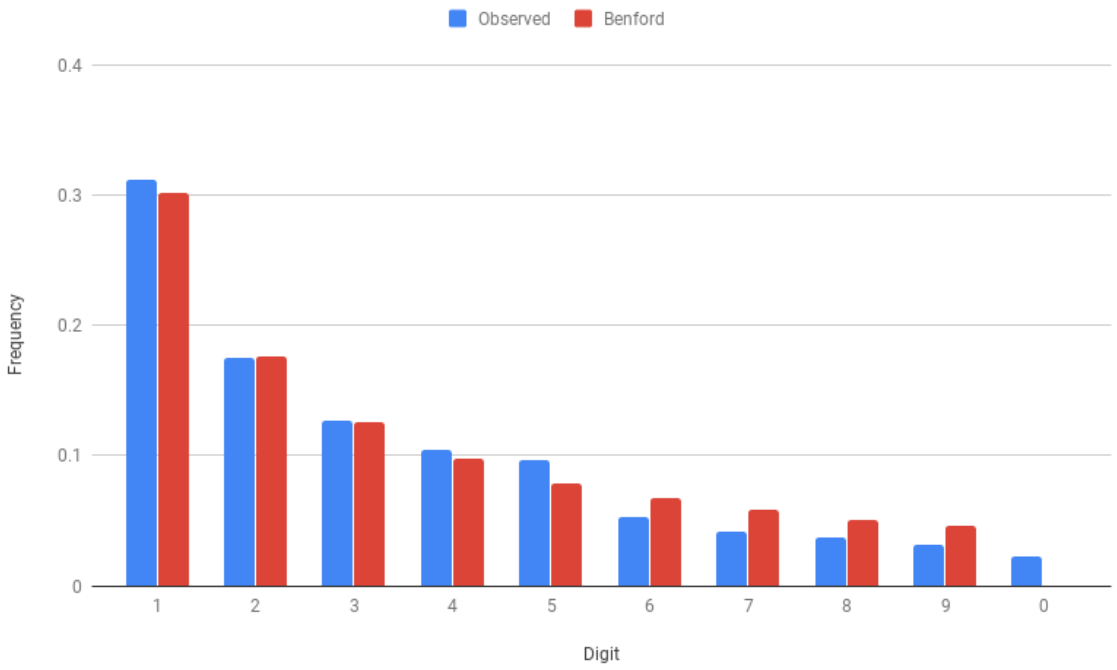


Figure 1: Distribution of the first significant digit

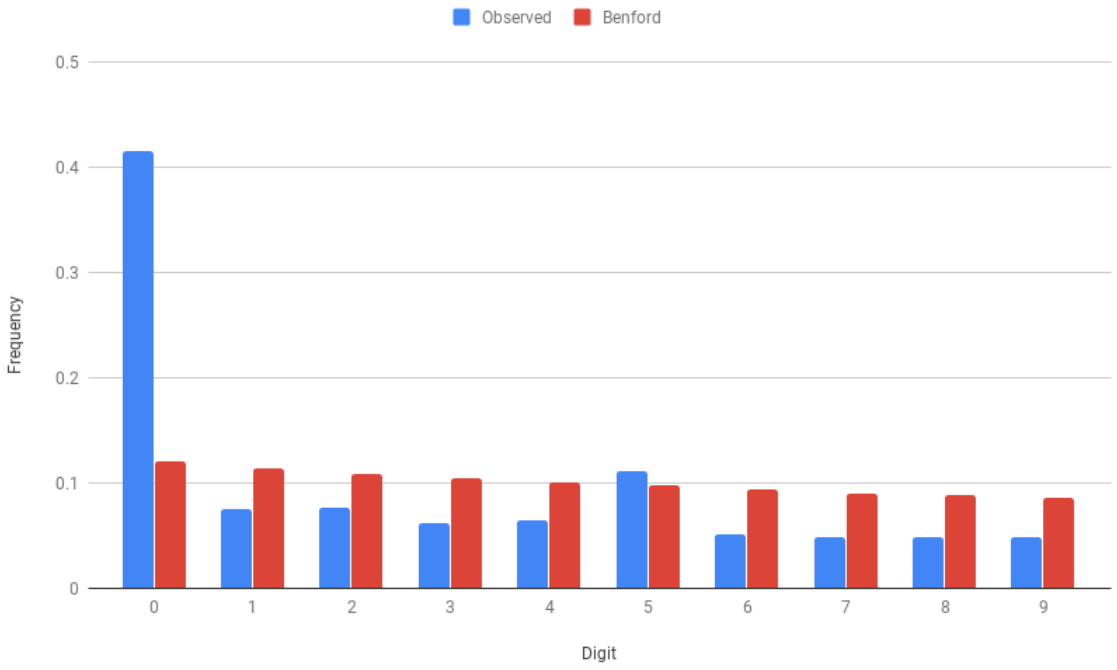


Figure 2: Distribution of the second significant digit

in the observed dataset fractions slightly deviate. The fact that some customers pay monthly fee 4.95 or 14.95 EUR/GBP might explain why fractions of digits 1 and 4 are a little bit higher than expected. Fractions of higher digits 6-9, on the contrary, are less than expected. That might be due to the fact that 2.2% of transactions start with zero, while Benford did not account for such case. Overall visual inspection of the first significant digit suggests close conformity to Benford's law.

As for the second significant digit, its conformity is less obvious, due to a high amount of transaction with second digit 0. Most likely these are all customers who use the account for savings or cash withdrawal and round the amounts due to practical or technical reasons. The same explanation might be also valid for the second significant digit 5, which exceeds the expected amount as well. And due to the higher amount of zeros and fives in the dataset, all other digits account for smaller frequencies than according to Benford.

Then, the obtained dataset was processed further. First of all, for each customer the share of transactions that have first digit 0,1, ...,9 was calculated. Similarly, for each customer the share of transactions that have second digit 0,1, ...,9 was calculated the same way, as well as fraction of transactions that are less than 10 EUR/GBP. Then, the number of transactions by each customer, the sum of transaction amounts and several types of distances between expected Benford's and observed distribution were calculated for both first and second digits: χ^2-distance, maximum absolute difference, Euclidean distance, mean absolute distance, Pearson correlation and Kullback and Leibler's divergence. All the customers that made less than 22 transactions were removed from the dataset, following the logic that according to Benford's distribution in order to have at least one transaction with first digit 9 (the least frequent digit according to Benford's law, that accounts for 4.6 % of a dataset), the data for each customer should consist of at least 22 transactions. It makes distances calculations more relevant. Customers with the small number of transactions account for around 33% of a dataset. Detailed variables description and their distributions can be found in Tables B-1 and B-2 in Appendix B.

To explore the predictive power of each variable, boxplots on Figures 3 and 4 exhibit differences in distributions for genuine customers and money laundering criminals.

Feature names are coded as following:

- *p_FD* - share of all transaction that start with D
- *p_SD* - share of all transaction that have second digit D (*p_S-1* - share of transactions that are less than 10)
- *N* - number of transactions

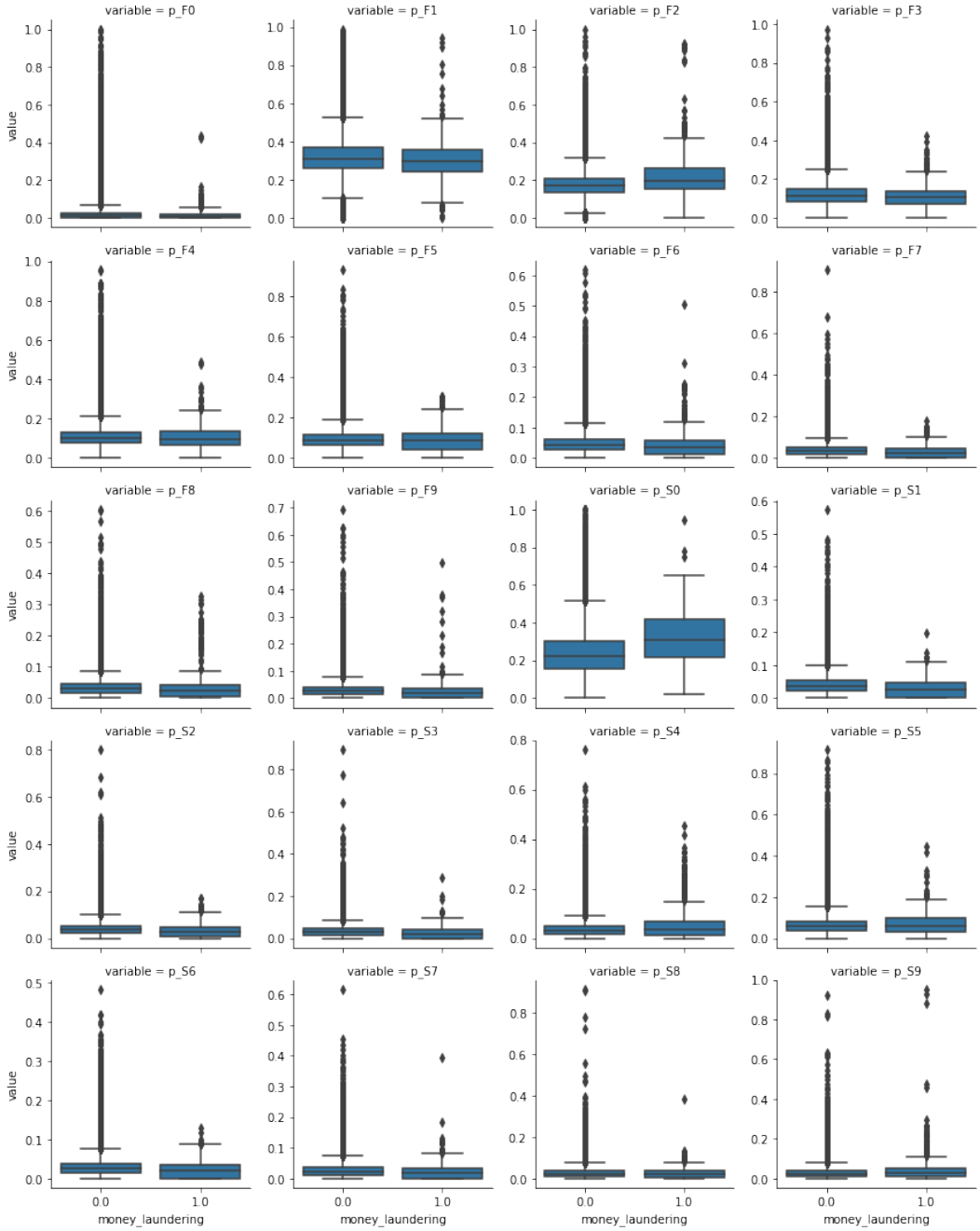


Figure 3: Distributions of Benford related variables

Note: all Benford related features mostly have narrow boxes, which means that half of all data is located within some narrow interval. Whiskers are not wide as well, so the data is not very scattered. Mostly all feature values are within interval 0 - 0.1, except *p_F1, p_F2* and *p_S0*, that have wider distribution and higher median values, compared to other variables. Black dots above or below whiskers are outliers. There is a significant amount of outliers for each variable.

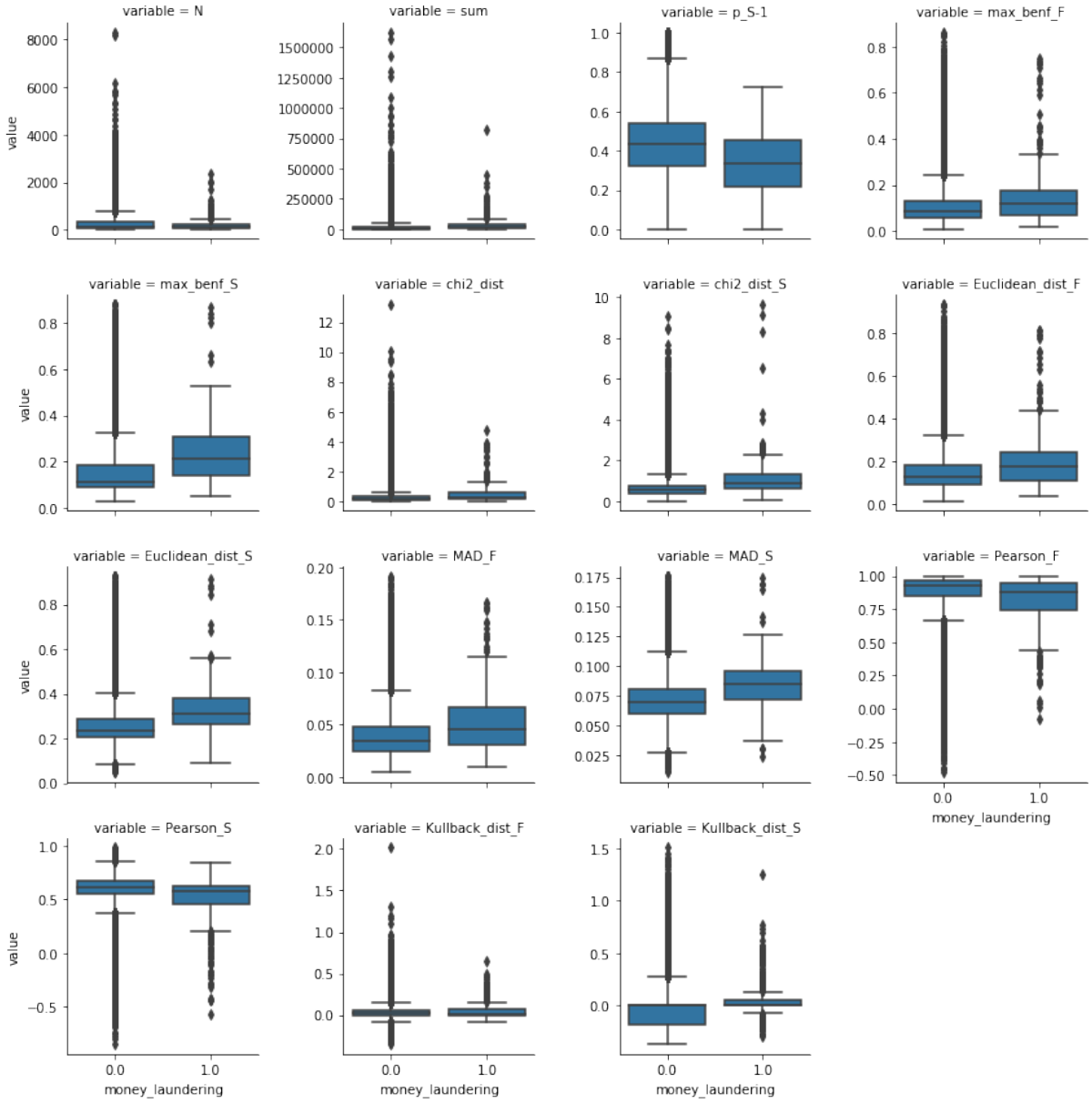


Figure 4: Distributions of secondary variables

Note: values of features, that are derived during data analysis, are more scattered, with different medians and deviations: Kullback distances mostly have low values, while Pearson correlation - high. Other distance measures have middle values. Conclusions, that can be made are that criminals, compared to genuine customers, make less one-digit transactions. Also, all distance measures for them are higher, while the correlation with Benford's distribution is lower. Overall the obtained result exactly matches what one would logically expect.

- *sum* - sum of transaction amounts
- *ml* - labels (stands for money laundering)
- all others indicate specific type of distances between distributions.

As demonstrated, distributions of some variables are indeed different. Among the most significant are variables *p_F2, p_S0, p_S1, p_S-1*, and such distances as maximum absolute difference, Euclidean distance and mean absolute distance. *p_S0* suggests that money laundering criminals do use rounds number more than genuine customers. On the contrary, the average number of one-digit transactions for criminals is much less, which makes perfect sense, as it would require too much effort and cost to break the amounts into such small pieces. Differences in distributions of distances suggest that the overall fit to Benford's law for money launderers is worse.

Due to a large number of outliers in Figure 3, it is not straightforward to see how different are the Benford related features for two groups of customers. Hence, it is reasonable to further explore it with the help of correlation matrix (see Figure 5), which helps to quickly identify incidence patterns and to recognize anomalies. Correlation heatmaps are perfectly suitable for comparing the measurement for each pair of dimension values. In the figure, red colours indicate a positive correlation, blue colours - negative and white - an absence of correlations.

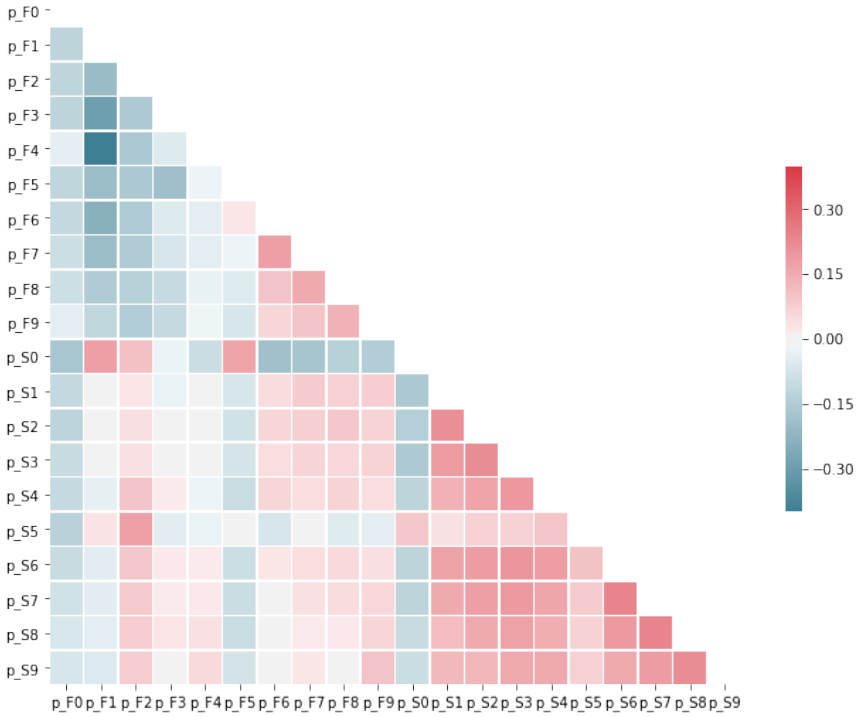
Following the logic of Benford's law, fractions of first digits should be slightly positively correlated between themselves, fractions of second digits should be more correlated between themselves, since, according to Benford, they are all expected to be around 10%. Thus, the correlation matrix should be reddish and smooth. And for genuine customers, it is indeed much smoother than for money launderers, which is rather motley.

Based on the matrix, rows under such features as *p_F2, p_F4,p_F8, p_S4, p_S9* are very different for genuine and fraudulent customers and thus are expected to give the biggest predictive power. Interesting, that one of the darkest cells in Figure 5(b) is the one, showing a relatively high negative correlation between transactions with second digit zero and second digit nine, which might indicate two different patterns, probably used by different types of criminals. One of them just uses round numbers for practical reasons, while others need to make high transfers below the threshold, which usually looks like a lot of 9,999 EUR/GBP transactions.

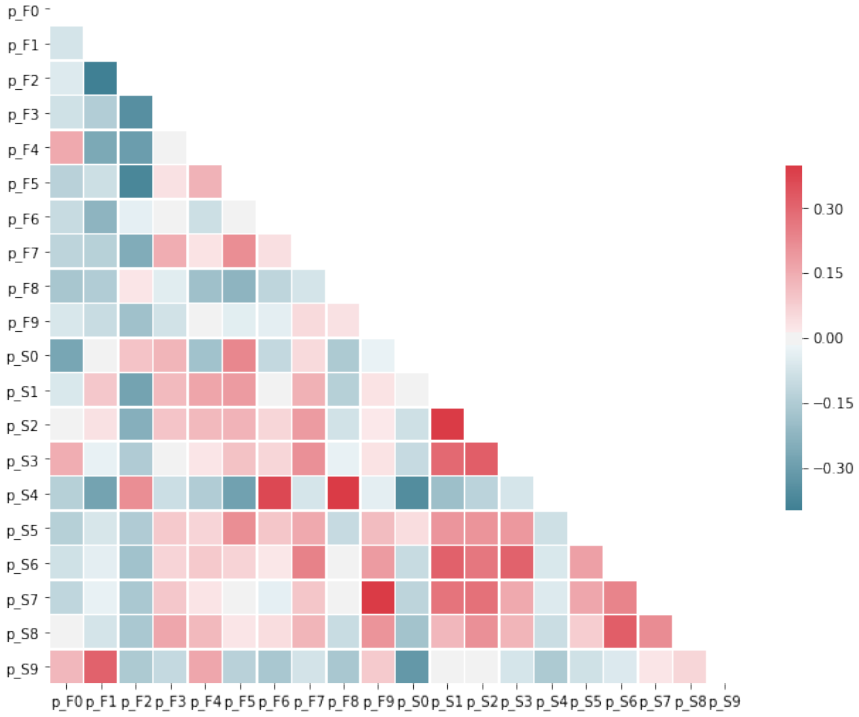
So, based on visual analysis, it can be concluded that the dataset supports the hypothesis that money laundering transactions do not conform with Benford's law and thus are not naturally originated.

5 Results

Visual exploration helped to get some insights about the data. Now, the aim of the modeling part is to build an actual classifier. For supervised learning, the whole dataset was split



(a) for genuine customers



(b) for money launderers

Figure 5: Correlation between variables

into train and test set in the following proportions: random 70% to train set and 30% as a test set.

I use the 4-fold cross-validation technique on the train set in order to choose parameters of the models and to prevent overfitting. K-fold cross-validation is a technique, where the data is divided into k subsets, the holdout method is repeated k times, such that each time, one of the k subsets is used as the validation set and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get total effectiveness of the model. Logistic regression (labeled as *LR*), random forest classifier (labeled as *RF*) and XGBoost classifier (labeled as *XGB*) are trained this way. At this stage, I also apply cost-sensitive learning. Based on a random search of parameters and cross-validation results, the best parameters are chosen and the model trained on all the train set. Then, each algorithm is tested on test data. Test results are displayed in Figure 6.

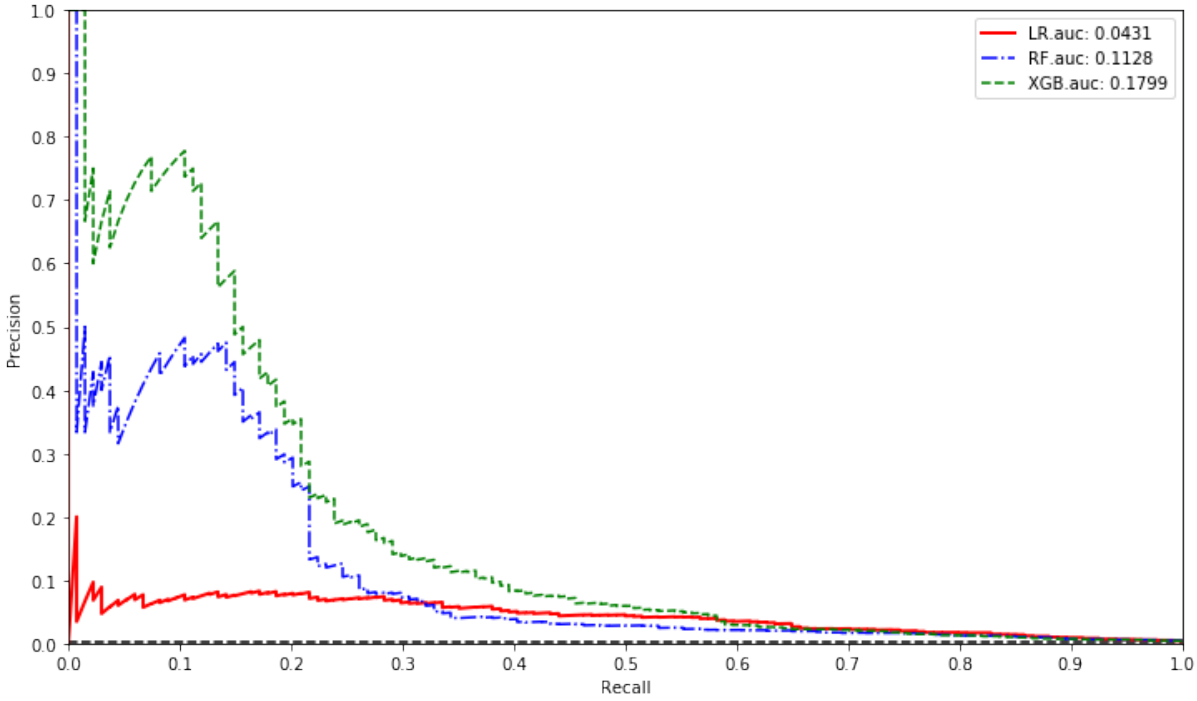


Figure 6: Precision-recall curves for three classifiers

The precision-recall curves are not very smooth due to very few positive cases. One fraudulent customer accounts for a significant part of the recall, so just one or several incorrectly classified customers look significant. At the same time, there are no huge spikes, which means that the classifiers are pretty stable.

Comparing different algorithms between themselves, as expected, XGBoost performs the best way with the area under precision-recall curve 0.1799. This is due to the sequential tree

building. XGBoost is followed by random forest and then by logistic regression, that fails to perform well in high-precision areas, but does its job when the high recall is needed.

To compare XGBoost to the real-world data one can take Danske Banks rule-based performance statistics, described in section 2 as a baseline. As can be seen, the superiority of machine learning methods is obvious. For example, according to Danske bank, it faces 0.5% precision and 40% recall. In contrast, machine learning model would achieve the same 40% recall with around 20 times higher precision at the level 10% and 100% recall with around the same 0.5% precision.

The feature importance according to XGBoost is depicted in Figure 7. The most significant variable turns out to be the cumulative sum of all transaction amounts per customer, which is logical, since more likely for amounts to be laundered. The fact that the sum of transactions is followed by the number of transactions is in accordance to money-laundering specific patterns, such as short bursts of activities, many cash withdrawals, deposits and a lot of money transfers. The other most significant features are all Benford related. It might mean that the classifier is 'intelligent' enough to find linear and non-linear relations between the features on its own and does not need much of 'additional help' in the form of distances. These features are *p_F2, p_F3, p_F4, p_F5, p_F8* from the bunch of first significant digits and *p_S-1, p_S5, p_S9* from second significant digit basket, which are almost exactly the same as were predicted based on correlation matrix in the previous subsection.

In order to improve the performance of XGBoost classifier, I then apply over- and undersampling techniques. The parameters for under- and oversampling methods together with parameters for XGBoost are tuned based on 4-fold cross-validation on train data with the help of *Pipeline* method that is offered by $scikit-learn$ library for Python. The results are displayed in Figure 8. Among these three techniques, the best one is random undersampling with the area under precision-recall curve 0.1581, followed by SMOTE method and then by SMOTEEN. However, not applying any of them is still preferable. The possible reason is that better performance of the classifier cannot be achieved by just balancing the dataset since the classifier already learned existing samples well, but by improving the quality of labels. Yet, one might consider using random undersampling for the area with small precision and high recall, since it decreases computational time without loss in performance.

The reason why labels might confuse algorithms is due to the absence of victims. In most cases, money laundering activities are not reported, so some fraudulent customers in the dataset are labelled as genuine. If improving the labels is not on the table, then it makes sense to apply also unsupervised techniques in order to find possible clusters. And as some labels are available, their presence in the discovered clusters might indicate that the rest of

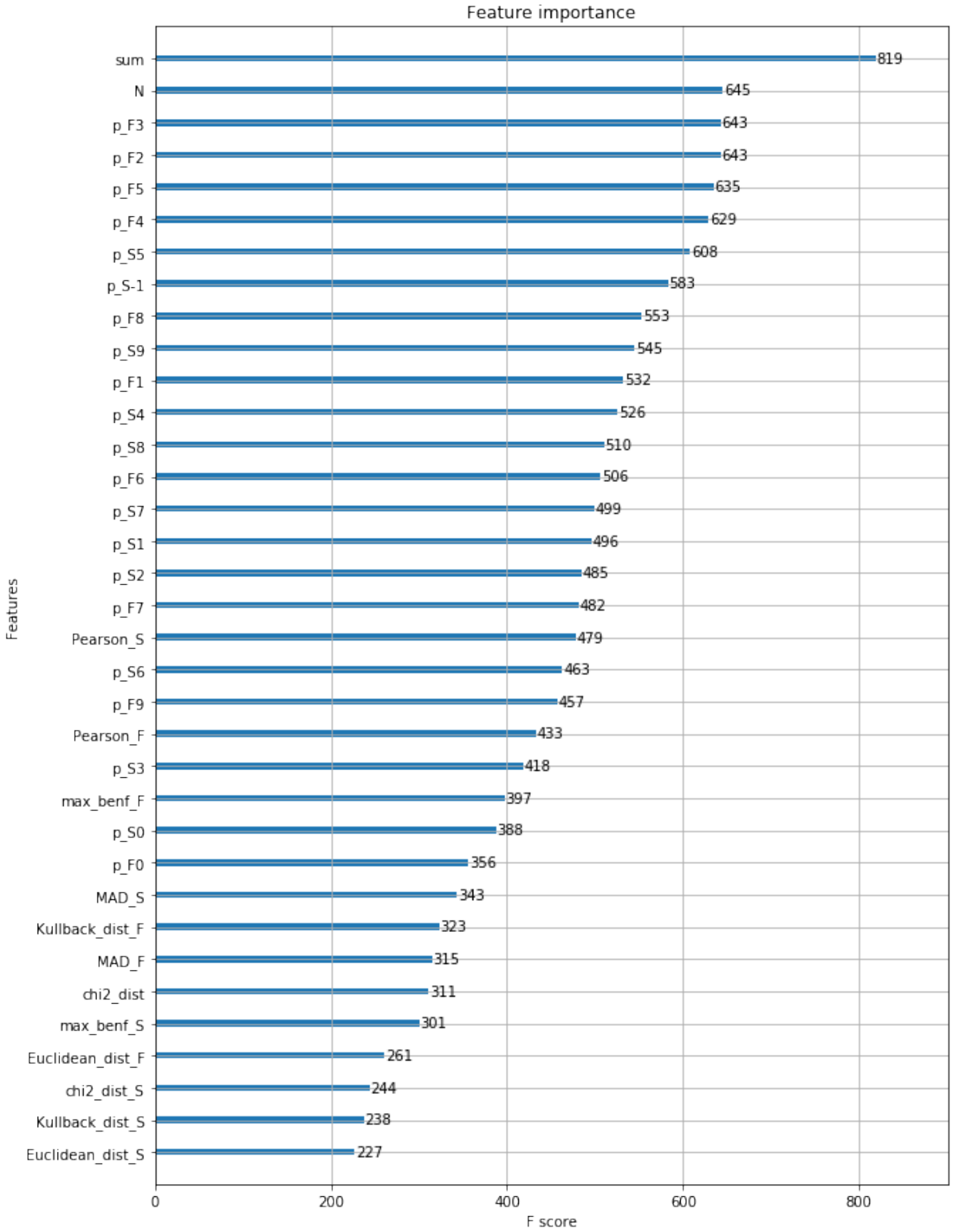


Figure 7: Feature importance according to XGBoost

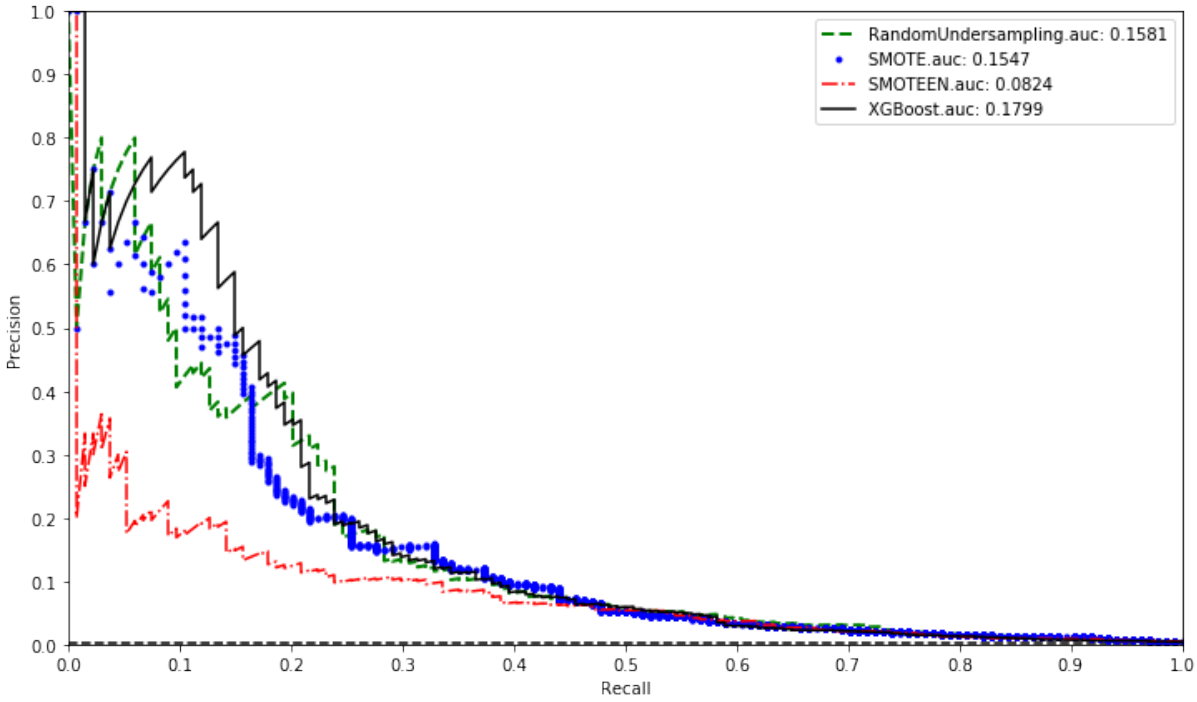


Figure 8: Over-under sampling techniques together with XGBoost

the observations from the same cluster might be similar and worth investigating.

Widely used clustering and anomaly detection methods that handle large amounts of data are k-means clustering, DBSCAN and isolation forest. Since no overfitting is possible, due to no labels required, they were applied to the whole dataset. Moreover, since labels are available, it is also possible to approximately estimate their performance measuring precision and recall. Varying different parameters specific for each algorithm and considering all observations from all minor clusters as 'model alerts', precision and recall for each combination of parameters can be calculated and plotted, as in Figure 9.

Precisely, in case of k-means clustering, all observations from all clusters but a major one are treated as 'model alerts', since all the clusters are relatively same size except one. In case of DBSCAN, 'model alerts' are all observations from all clusters but a major one and all observations from all clusters but two major ones, since the common result of DBSCAN were many clusters: the biggest one, one that is 1,5 - 3 times smaller than the biggest and many tiny clusters with couple dozens of observations. And it often is the case when money laundering counted for 50-90% of these tiny clusters, so in fact, there could be DBSCAN's point on the figure indicating even higher precision. It is done this way in order to keep it simple and avoid bias, but in practice, banks might want to investigate not all minor clusters, but only those with the highest rate of money laundering labels in them. And indeed, a random check of

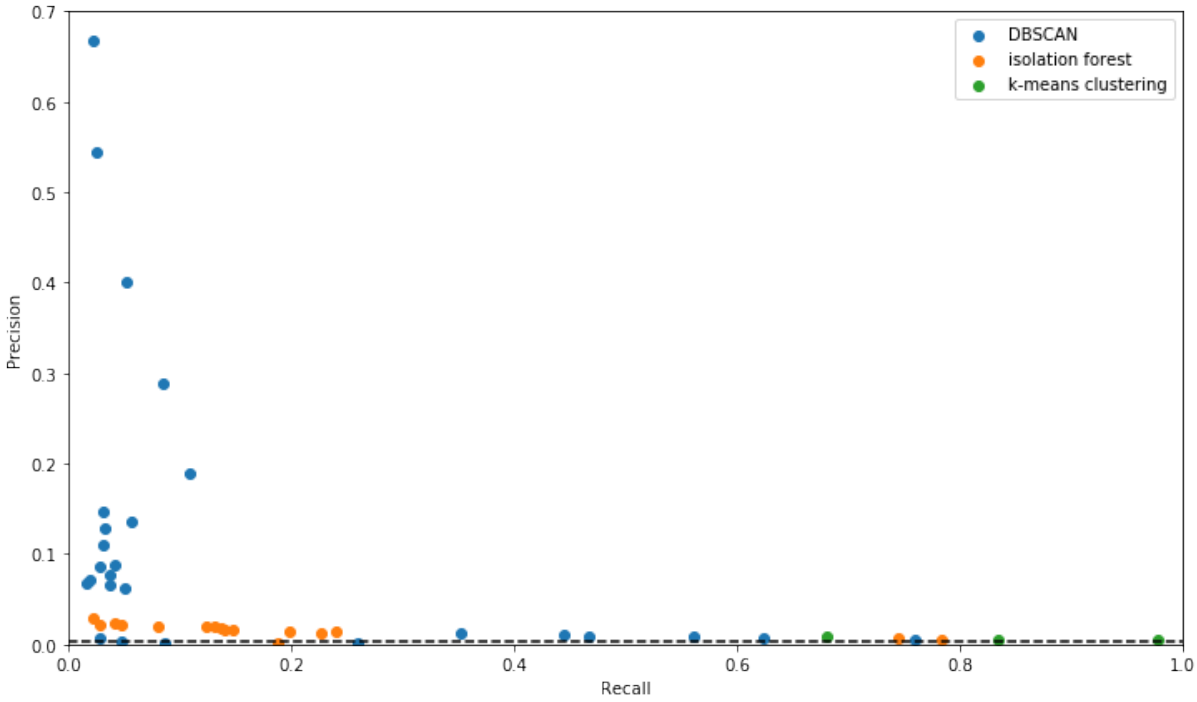


Figure 9: Unsupervised alhorithms performance

customers, that are labelled as genuine, but happened to be in the same cluster with fraudsters suggest that many of them are definitely worth investigation and might change their status in the future.

Overall, the clustering results are rather interesting. A comparative advantage of different methods is visible in different situations: DBSCAN works well mostly in the areas of high precision, k-means clustering may be used for high recall and isolation forest shows low precision and low recall, which makes it 'lose this competition'. DBSCAN looks like the most universal method: varying parameters one can achieve different goals. It definitely makes sense for financial institutions add clustering to the agenda.

Nonetheless, there is a lot more to do. First of all, a qualitative analysis of false positives with AML investigators will help understand where the classifier fails and possibly improve the quality of labels since false positives can, in fact, be incorrectly labelled true positives. Secondly, it might be useful to combine supervised and unsupervised techniques into one, using both benefits of having labels and strength of unsupervised learning to look for similarities. In addition, the machine learning industry is extremely wealthy. For instance, there is plenty of supervised and unsupervised deep learning methods to try, such as multi-layer perceptron, variational autoencoder, generative adversarial networks, self-organizing maps and many others, that can either improve the accuracy of existing techniques or be able to

find completely different types of criminals.

In conclusion, combined Benford's law and machine learning approach is obviously superior to a rule-based one due to different reasons. First of all, machine learning makes the threshold adjustable, so the company can choose precision and recall. 'Benford based' features make feature engineering simple and cheap and having nice numeric properties. Due to a relatively small number of features and the fact that the data for each customer is aggregated, the training is also relatively fast. Since it takes quite a while in terms of time and costs to build a more complicated machine learning model that handles all the personal information and information about transactions, the current approach would definitely work as a starting point. Moreover, Benford's law is not widely known and it is highly unlikely that while laundering money, criminals will purposely fit to it, which makes the developed tool even more valuable.

6 Conclusions

This paper explains the process and patterns of money laundering, describes real-world anti-money laundering solutions and examines recent scientific literature on automatic fraud detection. It also explains the concept of Benford's law and its applications in the literature. Then, it combines the law with supervised and unsupervised machine learning and develops a money laundering detection tool that can be used by financial institutions. The new model allows for a superior screening system, that is also simple and cost-effective.

References

Ausloos, M., Cerqueti, R., and Mir, T. A. (2017). Data science for assessing possible tax income manipulation: The case of italy. *Chaos Solitons Fractals*, 104:238–256.

Badal-Valero, E., Alvarez-Jareno, J. A., and Pavia, J. M. (2018). Combining benford's law and machine learning to detect money laundering. an actual spanish court case. *Forensic Science International*, 282:24–34.

Benford, F. (1938). The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78:551–572.

Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50:602–613.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (2001). Random forests. Statistics Department. University of California.

Castellano, R., Ausloos, M., and Cerqueti, R. (2016). Regularities and discrepancies of credit default swaps: a data science approach through benford's law. *Chaos Solitons Fractals*, forthcoming.

Chang, J. C. (2017). A study of benford's law, with applications to the analysis of corporate financial statements.

Cho, W. and Gaines, B. (2007). Breaking the (benford) law: Statistical fraud detection in campaign finance. *The American Statistician*, 61:218–223.

Chouiekh, A. and Ibn-Elhaj, E. (2018). Convnets for fraud detection analysis. *EProcedia Computer Science*, 127:133–138.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. volume 06.

de Marchi, S. and Hamilton, J. T. (2006). Assessing the accuracy of self-reported data: An evaluation of the toxics release inventory. *Journal of Risk and Uncertainty*, 32:57–76.

Diekmann, A. (2007). Not the first digit! using benford's law to detect fraudulent scientif ic data. *Journal of Applied Statistics*, 34:321–329.

Durtschi, C., Hillison, W., and Pacini, C. (2004). The effective use of benford's law to assist in detecting fraud in accounting data. *J. Forensic Account*, 5.

Fawcett, T. (2006). Introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874.

Fiore, U., De Santis, A., Perla, F., Zanetti, P., and Palmieri, F. (2017). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, 24:148–156.

Grammatikos, T. and Papanikolaou, N. (2016). Applying benford's law to detect fraudulent practices in the banking industry. *Working Paper. Luxembourg School of Finance, Luxembourg.*

Henselmann, K., Scherr, E., and Ditter, D. (2012). Applying benford's law to individual financial reports: An empirical investigation on the basis of sec xbrl filings.

Hickman, M. and Rice, S. (2010). Digital analysis of crime statistics: Does crime conform to benford's law? *Journal of Quantitative Criminology*, 26:333–349.

Hill, T. (1995). A statistical derivation of the significant-digit law. *Research Scholars in Residence*, 10.

Hindls, R. and Hronova, S. (2015). Benford's law and possibilities for its use in governmental statistics. *Statistika*, 95:54–64.

Juba, B. and Le, H. S. (2017). Precision-recall versus accuracy and the role of large data sets. Washington University.

Judge, G. and Schechter, L. (2007). Detecting problems in survey data using benford's law. *Journal of Human Resources*, 44.

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He, L., and Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100.

Ling, R. (1972). On the theory and construction of k-cluster. *The Computer Journal*.

Liu, F. T., Ming Ting, K., and Zhou, Z.-H. (2009). Isolation forest. pages 413 – 422.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137.

Mduma, N., Kalegele, K., and Machuve, D. (2019). A survey of machine learning approaches and techniques for student dropout prediction. *Data Science Journal*, 18.

Mir, T. A., Ausloos, M., and Cerqueti, R. (2014). Benford's law predicted digit distribution of aggregated income taxes: The surprising conformity of italian cities and regions. *The European Physical Journal B*, 87:261.

Mishra, S. (2007). Handling imbalanced data: Smote vs. random undersampling. *International Research Journal of Engineering and Technology (IRJET)*, 24(08):318–320.

Nami, S. and Shajari, M. (2018). Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. *Expert Systems with Applications*, 110.

Newcomb, S. (1881). Note on the frequency of use of the different digits in natural numbers. *American Journal of Mathematics*, 4(1):39–40.

Nigrini, M. (2015). Persistent patterns in stock returns, stock volumes, and accounting data in the u.s. capital markets. *Journal of Accounting, Auditing Finance*, 30.

Palshikar, G., Apte, M., and Baskaran, S. (2014). Analytics for detection of money laundering. page 2.

Patil, S., Nemade, V., and Soni, P. (2018). Predictive modelling for credit card fraud detection using data analytics. *Procedia Computer Science*, (132):385–395.

Pericchi, L. and Torres, D. (2012). Quick anomaly detection by the newcomb–benford law, with applications to electoral processes data from the usa, puerto rico and venezuela. *Statistical Science - STAT SCI*, 26.

Rauch, B., Goettsche, M., Brähler, G., and Engel, S. (2011). Fact and fiction in eu-governmental economic data. *German Economic Review*, 12:243 – 255.

Shi, J., Ausloos, M., and Zhu, T. (2017). Benford's law first significant digit and distribution distances for testing the reliability of financial reports in developing countries.

Torres, J., Fernández, S., Gamero, A., and Sola, A. (2007). How do numbers begin? (the first digit law). *European Journal of Physics*, 28(3).

Villas-Boas, S., Fu, Q., and Judge, G. (2017). Benford's law and the fsd distribution of economic behavioral micro data. *Physica A: Statistical Mechanics and its Applications*, 486.

Whyman, G., Shulzinger, E., and Bormashenko, E. (2016). Tintuitive considerations clarifying the origin and applicability of the benford law. *Results in Physics*, 6:3–6.

X Ling, C. and Sheng, V. (2010). Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*.

Appendices

Appendix A

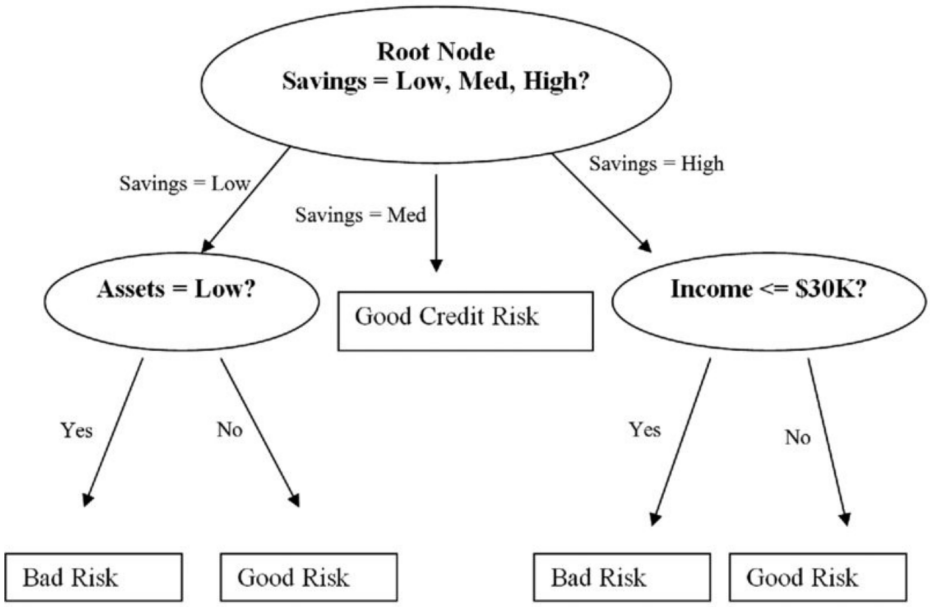


Figure A-1: Example of a decision tree

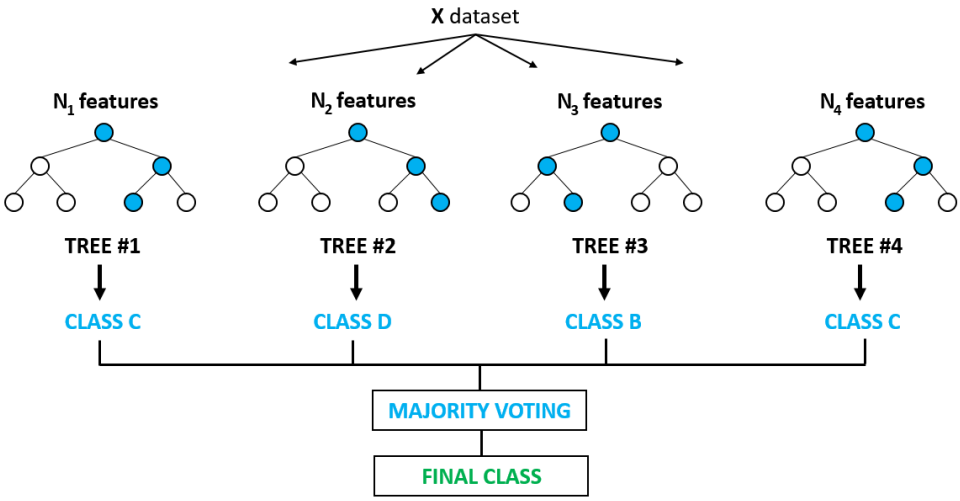


Figure A-2: A schematic of a tree-based classifier

Appendix B

Table B-1: Explanation of variable names

Variable name	Explanation
N	Total number of transactions for each customer
sum	sum of transaction amounts
p_F0	share of transactions that start with 0
p_F1	share of transactions that start with 1
p_F2	share of transactions that start with 2
p_F3	share of transactions that start with 3
p_F4	share of transactions that start with 4
p_F5	share of transactions that start with 5
p_F6	share of transactions that start with 6
p_F7	share of transactions that start with 7
p_F8	share of transactions that start with 8
p_F9	share of transactions that start with 9
p_S-1	share of transactions that have no second digit (are less than 10)
p_S0	share of transactions that have second digit 0
p_S1	share of transactions that have second digit 1
p_S2	share of transactions that have second digit 2
p_S3	share of transactions that have second digit 3
p_S4	share of transactions that have second digit 4
p_S5	share of transactions that have second digit 5
p_S6	share of transactions that have second digit 6
p_S7	share of transactions that have second digit 7
p_S8	share of transactions that have second digit 8
p_S9	share of transactions that have second digit 9
max_benf_F	max absolute difference for the first significant digit
max_benf_S	max absolute difference for the second significant digit
chi2_dist	χ^2-distance between the empirical proportions of first digits in the data and those predicted by Benford
chi2_dist_S	χ^2-distance between the empirical proportions of second digits in the data and those predicted by Benford
Euclidean_dist_F	Euclidean distance between the empirical proportions of first digits in the data and those predicted by Benford
Euclidean_dist_S	Euclidean distance between the empirical proportions of second digits in the data and those predicted by Benford
MAD_F	mean absolute distance for first digit
MAD_S	mean absolute distance for second digit
Pearson_F	Pearson correlation coefficient between the empirical proportions of first digits in the data and those predicted by Benford
Pearson_S	Pearson correlation coefficient between the empirical proportions of second digits in the data and those predicted by Benford
Kullback_dist_F	Kullback and Leibler's divergence between the observed and BL frequencies of first digit
Kullback_dist_S	Kullback and Leibler's divergence between the observed and BL frequencies of second digit

Table B-2: Distribution of all variables

Variable	mean	std	min	0.25	0.5	0.75	max
N	270.118	342.607	22.000	62.000	144.000	338.000	8291.000
sum	19977.266	31491.049	0.230	4120.050	10310.840	24420.485	1622591.000
p_F0	0.023	0.045	0.000	0.000	0.010	0.027	1.000
p_F1	0.317	0.092	0.000	0.260	0.309	0.366	0.988
p_F2	0.176	0.069	0.000	0.135	0.171	0.210	1.000
p_F3	0.125	0.066	0.000	0.085	0.116	0.152	0.972
p_F4	0.112	0.057	0.000	0.079	0.104	0.133	0.958
p_F5	0.094	0.051	0.000	0.064	0.088	0.115	0.934
p_F6	0.047	0.033	0.000	0.027	0.044	0.063	0.619
p_F7	0.038	0.028	0.000	0.020	0.035	0.051	0.907
p_F8	0.034	0.027	0.000	0.018	0.031	0.045	0.605
p_F9	0.030	0.026	0.000	0.014	0.027	0.040	0.693
p_S-1	0.431	0.162	0.000	0.323	0.432	0.540	1.000
p_S0	0.240	0.120	0.000	0.156	0.221	0.301	1.000
p_S1	0.040	0.029	0.000	0.022	0.036	0.053	0.574
p_S2	0.040	0.030	0.000	0.022	0.037	0.054	0.799
p_S3	0.033	0.027	0.000	0.017	0.030	0.044	0.893
p_S4	0.036	0.031	0.000	0.018	0.031	0.047	0.761
p_S5	0.066	0.047	0.000	0.037	0.057	0.083	0.915
p_S6	0.028	0.025	0.000	0.012	0.024	0.038	0.482
p_S7	0.027	0.025	0.000	0.011	0.023	0.037	0.615
p_S8	0.027	0.025	0.000	0.011	0.023	0.036	0.909
p_S9	0.028	0.028	0.000	0.011	0.023	0.038	0.950
max_benf_F	0.106	0.070	0.007	0.059	0.088	0.132	0.861
max_benf_S	0.152	0.096	0.029	0.091	0.114	0.184	0.880
chi2_dist	0.244	0.329	0.002	0.084	0.157	0.290	13.207
chi2_dist_S	0.687	0.514	0.022	0.415	0.557	0.773	9.646
Euclidean_dist_F	0.149	0.080	0.014	0.093	0.131	0.184	0.936
Euclidean_dist_S	0.256	0.084	0.047	0.204	0.238	0.285	0.928
MAD_F	0.039	0.019	0.004	0.025	0.035	0.048	0.192
MAD_S	0.071	0.018	0.010	0.060	0.070	0.081	0.176
Pearson_F	0.872	0.152	-0.481	0.845	0.927	0.964	0.999
Pearson_S	0.592	0.162	-0.854	0.554	0.620	0.677	0.985
Kullback_dist_F	0.039	0.066	-0.355	0.000	0.019	0.061	2.017
Kullback_dist_S	-0.062	0.159	-0.372	-0.180	0.000	0.000	1.507

Non-exclusive licence to reproduce thesis and make thesis public

I, Solomiya Branets,
(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Detecting money laundering with Benford's law and machine learning
,
(*title of thesis*)

supervised by Lenno Uusküla.
(*supervisor's name*)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

author's name Solomiya Branets
dd/mm/yyyy 23.05.2019