FRAUD DETECTION FOR FINANCIAL SERVICES

exploiting Big Data Analytics techniques

Big Data Analytics Exam

A.Y. 2018/2019

Author:

Franza Tiziano

Abstract

The focus of this research work is fraud detection on financial transactions exploiting real-time sensitive data, obtained using PaySim, a mobile money simulator dedicated for the same task [1] and developed by Lopez-Rojas and his collaborators in the swedish Blekinge Institute of Technology[1].

First of all, an **Exploratory Data Analysis** (EDA) was carried out to retrieve the first impressions of the dataset through a straightforward look at its structure and then describing the latent information present in the dataset through means of analytical and statistical widely-used tools.

At this point follows the step of **Data Preprocessing**: it is needed to prepare data, carrying the most useful data to be treated for the prediction model and leaving all the information that can be of no use because lacking of correlation with the class field or being redundant with the other selected features (that have higher relevance too).

Thus, a step of **Data Transformation** is considered in order to transform the preprocessed dataset into training and test set, the former that will be useful to the system to learn from the data, the latter one that will be suitable to understand how effectively the system learned on the data.

Consequently to the process of Data Transformation, information is extracted from the data by applying machine learning algorithms in the **Data Mining** phase along the two steps of learning from the training test and evaluating from the test set.

In this last phase, the objective is to show the **Interpretation of the Results** through some visual representations, summing up what were the most interesting results according to the task that we described from the very beginning.

Among all the classifiers, Gradient Boosting Classifier reach the highest Recall value, which means that this classifier is able to classify most of the fraud behaviour, however it is remarkable the fact that the Decision Trees algorithm is able to achieve the highest Precision value with the highest True Positive value too.

[1] https://www.bth.se/eng

1. Introduction

Nowadays, world is in continuous evolution together with the application of even more technological solutions to old problems in our everyday life, including money transfer operations.

During 20th century the most common way to buy goods, invest in services and any other general operation that involves a transfer of money was carried out through paper money. Along with the appearance of online payments systems (such as PayPal[2], Amazon Pay[3], Google Checkout[4], etc...), thus the spread of mobile apps and the credit cards, it has become more and more difficult to protect customers' privacy about their own balances and consequently the handling of money movements too.

In parallel to this evolution, even the kind of performed thefts have moved increasingly from physical attacks to cybercrime attacks on the smartphone directly or indirectly through the internet, exploiting methods like SIM phishing swap (retrieves information by targeting a weakness into the two-factor authentication and two-step verification while sending passwords through the internet), fake support calls (to obtain pin reset) and stolen phones (guaranteeing direct access to the banking service with authentic credentials).

As stated in an Identity Fraud Study, conducted by Javelin Strategy & Research[5] and released in 2018, the percentage of fraud crimes is rising. This phenomenon is due to the evolution of cyber attacks in conjunction with industry's efforts to implement better security. It was reported that in 2017 the biggest source of fraud risks came from online channels dedicated to shopping purposes and increased with the introduction of credit cards.

[2] https://www.paypal.com/en/home
[3] https://pay.amazon.it/
[4] https://pay.google.com/about/business/checkout/
[5] https://www.javelinstrategy.com/coverage-area/2018-identity-fraud-fraud-enters-new-era-complexity

Thus, in the same work it is pointed that cybersecurity attacks in the same year caused a loss of 16.8 billion dollars towards 1.3 million more victims than the previous year. For the very first time in history, the percentage of compromised social security numbers (35%) gets higher than the one for compromised credit card numbers (30%), causing consumers' further loss of trust in institutions.

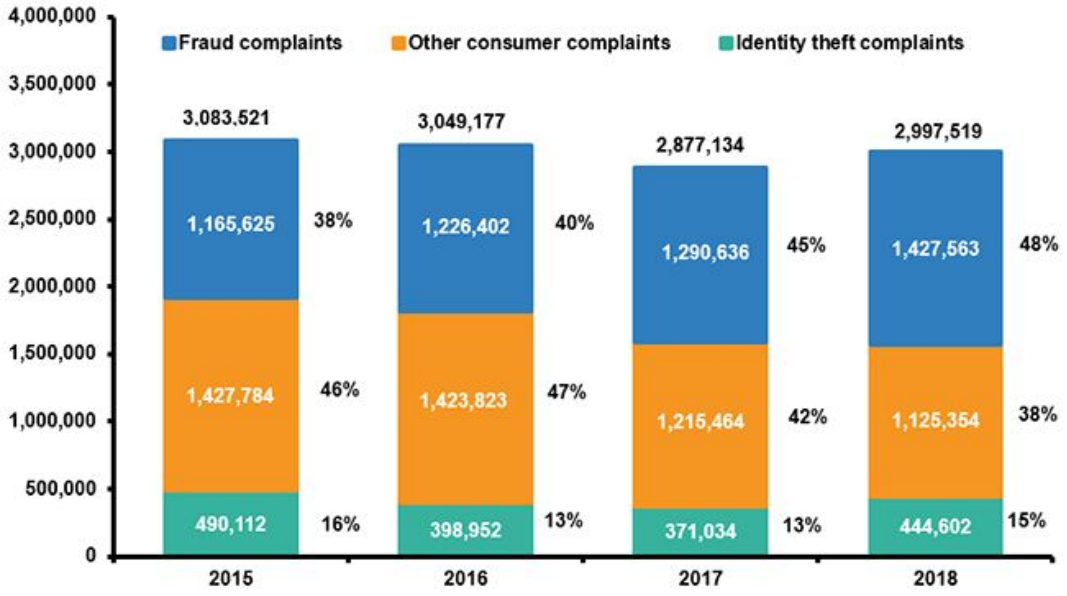


Fig.1 - Consumers' complaints across 2015 to 2018 (does not include "Do Not Call" registry complaints). Source: Federal Trade Commission, Consumer Sentinel Network[6].

In 2019's snapshot the overall fraud rates drop by 15% and the percentage of involved victims falls by 5.66% but still this is felt as a critical problem to be monitored and stemmed in the best way possible. One way to do this is exploiting scientific methods like Big Data Analytics, applied on data which were previously gathered for this specific purpose.

According to Javelin's Prevention, Detection & Resolution Model, the approach to confront the problem is three-fold, beginning with fraud prevention, following with fraud detection and ending up with fraud resolution.

6 https://www.ftc.gov/system/files/documents/reports/consumer-sentinel-network-data-book-2018/consumer_sentinel_network_data_book_2018_0.pdf

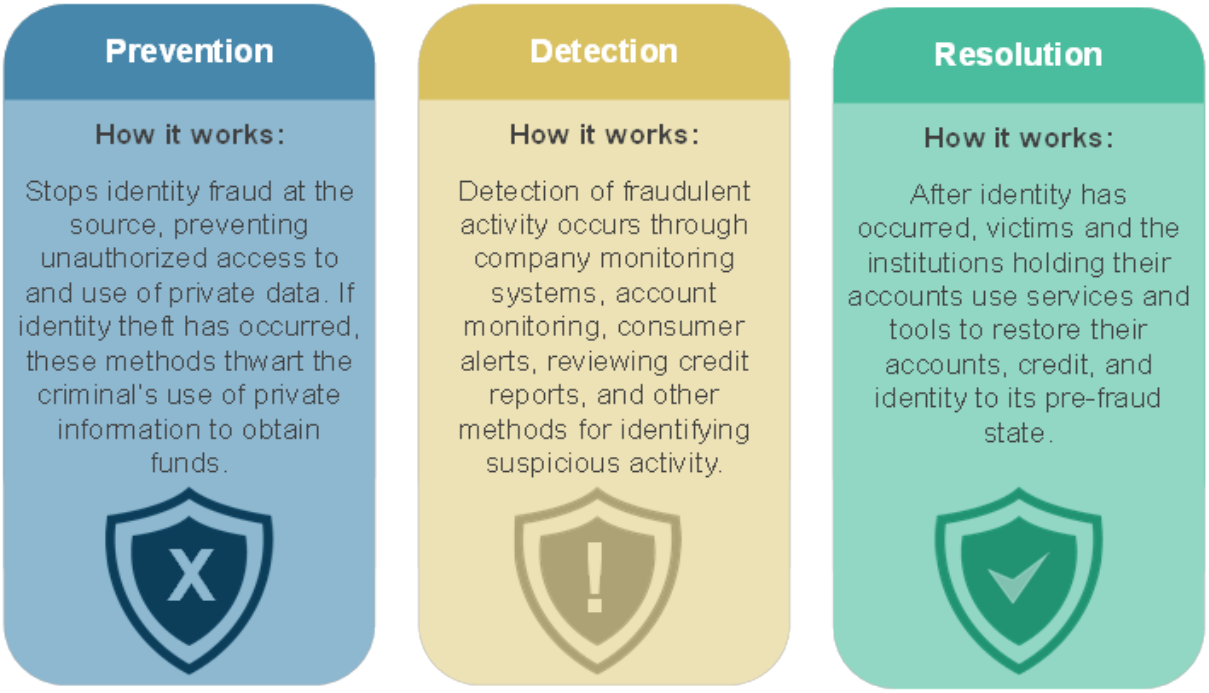


Fig.2 - Javelin's Prevention, Detection & Resolution Model. Javelin Strategy & Research, 2018.

The focus of this research work is fraud detection on financial transactions exploiting real-time sensitive data, obtained using PaySim, a mobile money simulator dedicated for the same task [1] and developed by Lopez-Rojas and his collaborators in the swedish Blekinge Institute of Technology[7].

This report is organized as follows: Section 2 presents the features contained in the dataset from an analytical and a statistical point of view. Section 3 describes the preparation of the data and cleaning. Section 4 pays particular attention to the transformation of the data, which is then used as input to the learning model. Section 5 focuses on the choice and implementation of the classification system that will learn from the yet preprocessed data. Section 6 is dedicated to the exploration of the findings that were gathered during the execution of the experiments. Section 7 summarizes the overall research work, providing insights on potential future work that can be carried out on this system to improve its performance in this field.

[7] https://www.bth.se/eng

2. Exploratory Data Analysis

First of all, an **Exploratory Data Analysis** (EDA) was carried out to gather sense of our data through a straightforward look at its structure and then gathering information through means of analytical and statistical widely-used tools for describing the latent information in the dataset.

This Section will be divided into two subsections. The former will explain where the dataset has been gathered, a general overview followed by a brief exemplified description of its fields.

Conversely, the latter will go deeper in the understanding of the dataset trying to uncover the distributions and statistical information on the features according to the target class (genuine or fraud transaction) that we are interested in forecasting through the learning process.

2.1. The dataset

The dataset adopted to perform research in the domain of fraud detection was made available by Lopez-Rojas on kaggle.com[8] under Creative Commons BY-SA 4.0 license.

It was generated by a simulator called **PaySim** [1] which uses aggregated data from a private dataset to generate a new synthetic dataset (unfortunately, it was not possible to gain the original private dataset since it contains sensitive real data). Regardless of this, the simulator generates the corresponding synthetic dataset gathering the standard genuine transactions, then injecting malicious behaviour in these ones, thus assigning a label to them (0 for genuine transactions, 1 in the other case).

[8] https://www.kaggle.com/ntnu-testimon/paysim1

On the other hand, the original dataset was provided by a multinational company, running in more than 14 countries all around the world. The transactions describe one month time for a mobile money service located in an African country. The size of the synthetic dataset is 1:4 respect to the original one. From now on, we'll be referring the word "dataset" to the synthetic one.

In the dataset, three main **entities** are involved in the transactions:

- **clients**, who are the normal customers of the system;
- **merchants**, who play a passive role and only serve the clients in specific operations;
- **fraudsters**, who represent the threat to the system and the principal target of this study. This role can be played by other customers, merchants, insiders of the organization, hackers and common thieves.

The main focus is on clients, for this reason data is client-oriented and consequently it does not include many details regarding merchants (unless needed to understand amounts in payments for goods or services).

Transactions in this dataset can be specialized into five main classes:

- **CASH-IN**: the process of increasing the balance of account by paying in cash to a merchant.
- **CASH-OUT**: the process of withdrawing cash from a merchant, which decreases the balance of the account (opposite to the process of CASH-IN).
- **DEBIT**: the process sending the money from the mobile money service to a bank account (similarly to the process of CASH-OUT).
- **PAYMENT**: the process of paying for goods or services to merchants, which decreases the balance of the account and increases the balance of the receiver.
- **TRANSFER**: the process of sending money to another user of the service through the mobile money platform.

The common fraud scenario involves a victim customer, who is unable to access his account anymore. In order to apply the theft, the fraudster will potentially transfer the money on another temporary account (a.k.a. "mule accounts") and then execute a cash-out operation to get hands on the profit and avoid the risk of frozen accounts due to detection, or just paying merchants from the victim's account to obtain highly-valuable goods.

The dataset is 471MB heavy, 178MB compressed. It holds more than 6 million rows and it is organized along 11 fields, here analyzed one by one:

- **Step**: indicates the hour in which a transaction has occurred. Values for this field are integer auto-increment indices in range [1, 744]. The highest value is explainable by the fact that this dataset contains one-month data (in fact 744 equals 24 hours * 31 days). This field allows to the learning model to understand the order in which transactions happen.
- **Type**: identifies the specific kind of transaction. Values for this field are discrete and already explained too: CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER.
- **Amount**: refers to the amount of money involved in the transaction. Values for this field are positive decimals.
- **NameOrig/NameDest**: represent the balance of the two entities who get involved into the transaction. Values for this field are compound literals, which can be split into two parts:
    - the first letter: can be "C" or "M" depending on whether the actor of the transaction is either a customer or a merchant. Merchants can be found only in the "nameDest" field.
    - the numeric id: represents the identifier for the corresponding actor of the transaction. The two fields "nameOrig" and "nameDest" do not share any numeric id.

- **OldBalanceOrg/OldBalanceDest**: represent the balance of the two entities, who get involved into the transaction meanwhile it gets carried out, and that represent the customer who starts the transaction and the customer who ends it respectively.
  As stated into the dataset official description, no information regarding merchants can be found in OldBalanceDest. Values for this field are decimals lower-bounded with 0.

- **NewBalanceOrig/NewBalanceDest**: represent the balance of the two entities, who get involved into the transaction meanwhile it is carried out, that are the customer who starts the transaction and the customer who ends it respectively.
  As stated into the dataset official description, no information regarding merchants can be found in NewBalanceDest. Values for this field are decimals lower-bounded with 0.

- **IsFraud**: is the target class who says whether a transaction is operated by a malicious user or an authentic one. Values for this field are the two integers 0 and 1, the former for genuine transactions and the latter for fraud ones.

- **IsFlaggedFraud**: identifies single transactions that involved more than $200,000, marking them as fraud.

All these details regarding the dataset can be partly observed on the data, but also retrieved in further detail in 2016 Lopez-Rojas Doctoral Thesis [2], concerning the design and development of the financial transaction simulator PaySim.

In the next section we'll be explaining less intuitive insights regarding the dataset and which can be obtained by the use of descriptive statistics and reason about the discovered hidden information.

2.2. Insights on the dataset

The first thing that we have to investigate is whether we are working with a balanced dataset or an imbalanced one. In order to understand this, we need to observe the distribution of the fraud and genuine transactions, inferable from the field "isFraud":

All transactions	Genuine transactions	Fraud transactions
6,362,620 (100 %)	6,354,407 (0.9987 %)	8,213 (0.0013 %)

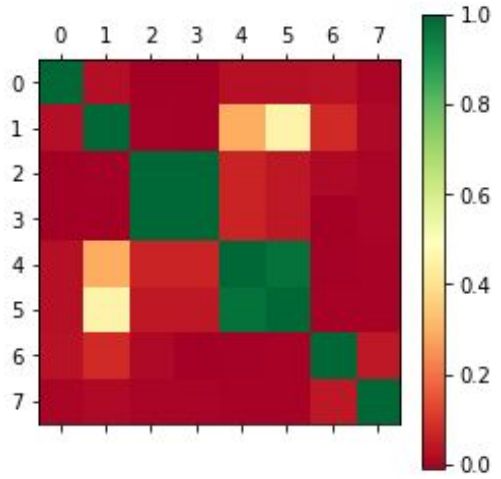
Tab.1 - Overall amounts and percentages for genuine, fraud and whole transactions in the dataset.

As we can notice from the table, the skewness of our dataset we are dealing with an evident **imbalanced dataset** as soon as 99.87% of the transactions are genuine and 0.13% of them are registered as fraudulent. This evidence is confirmed by the highly positive value for skewness (~27.78) for the class field too. Here means that it would be possible to achieve 99.87% of accuracy just exploiting the model that assigns to every transaction the class "genuine", however we would have misleading results generated by highly overfitting the data on genuine transactions, hence failing in catching the fraud ones. We'll try to understand how to deal with it in the next section.

Taking into account this piece of information, we can go through an analysis of the single fields in order to understand how to reduce the dimensionality of the feature space, considering only the features that present high correlation with the fraudulent behaviour - the non/less relevant features will be discarded since useless for the prediction task. The features we're going to cover for the analysis are: the step (hour of a day), the transaction type, the involved amount of money, the ids of the involved customers, their balance account before and after the transaction, the indicator "isFlaggedFraud".

In order to get insights regarding the usefulness of these features towards the prediction task, the tools exploited to do this are the two statistical indicators of covariance and correlation. Bivariate **correlation** will give us is knowledge about potentially interesting trends between the available paired features, on the other hand **covariance** will tell us whether these ones vary together positively, negatively or simply they are likely to vary in a non-uniform way. Such information becomes useful for the feature selection task since we are looking for those ones which are the least interrelated within each other, but still most correlated with the class value (fraud or genuine) as much as the dataset will allow it.

The employed correlation method is the Pearson Correlation Coefficient and the correlation matrix that can is obtained is the following:

Correlation Matrix	Correlation Features
	0 → step 1 → amount 2 → oldbalanceOrg 3 → newbalanceOrig 4 → oldbalanceDest 5 → newbalanceDest 6 → isFraud 7 → isFlaggedFraud

Tab.2 - The Pearson Correlation Matrix with values in range [-1, 1]. Values are truncated to 0 since there are no features in the dataset that appear to be inversely correlated.

At first sight, the Correlation Matrix presents very weak correlation in general among all couples of features. The matrix presents the standard characteristics of every correlation matrix, as such the fact that it is symmetric and the fact

that all the values on the main diagonal have correlation equals 1, this because every feature have 1.0 correlation when coupled with itself.
Apart from these basic considerations, we can observe strong correlation in the following couples of features:

- "oldbalanceOrg" and "newbalanceOrig" with a correlation coefficient of 0.99. This can be justified by the fact that these values are obtained for some of the transactions in combination with the amount value correspondent to the same transaction. Amount can be either positive or negative according to the carried out transaction.
- "oldbalanceDest" and "newbalanceDest" with a correlation coefficient of 0.98. This can be explained by the fact that these values are obtained for some of the transactions in combination with the amount value correspondent to the same transaction.

Thus we can spot relatively good correlation in the following couples of features:

- "amount" and "oldbalanceDest" with a correlation coefficient of 0.29.
- "amount" and "newbalanceDest" with a correlation coefficient of 0.46.

At this moment of the analysis we haven't got many clues to understand this correlation, however it indicates that redundant information exists and can be solved either removing some of them (e.g. "oldbalanceDest" and "newbalanceDest", keeping "amount", or viceversa), or applying the Principal Component Analysis method thus exploiting Eckart-Young Theorem.

In any case, this preprocessing would cause a slight loss of information, in the second case with greater precision but at a much higher computational cost, but we'll explain more on these decisions later in the document.

On the other hand, the correlation between features and the class field is always visibly low: the retrieved values for correlation are summarized in the table on the right. As we can see, the lowest correlation can be observed with the "oldbalanceDest" and "newbalanceDest", while the highest one can be observed with the "amount", "step" and "isFlaggedFraud" feature. The next step is the understanding in detail of all the useful features.

FEATURE	isFraud
step	0.032
amount	0.077
oldbalanceOrg	0.010
newbalanceOrig	-0.008
oldbalanceDest	-0.005
newbalanceDest	0.001
isFlaggedFraud	0.044

Tab.3 - Correlation value for each feature with the class field.

Step. From the generic description of the dataset we don't know anything about this field but the fact that this field indicates hours in a month taken as a sample from real world data, but maybe with statistical analysis of the data we can figure out something more about it. In fact we can obtain some insights regarding the hours of the day and the night by the general distribution of the transactions per hour.

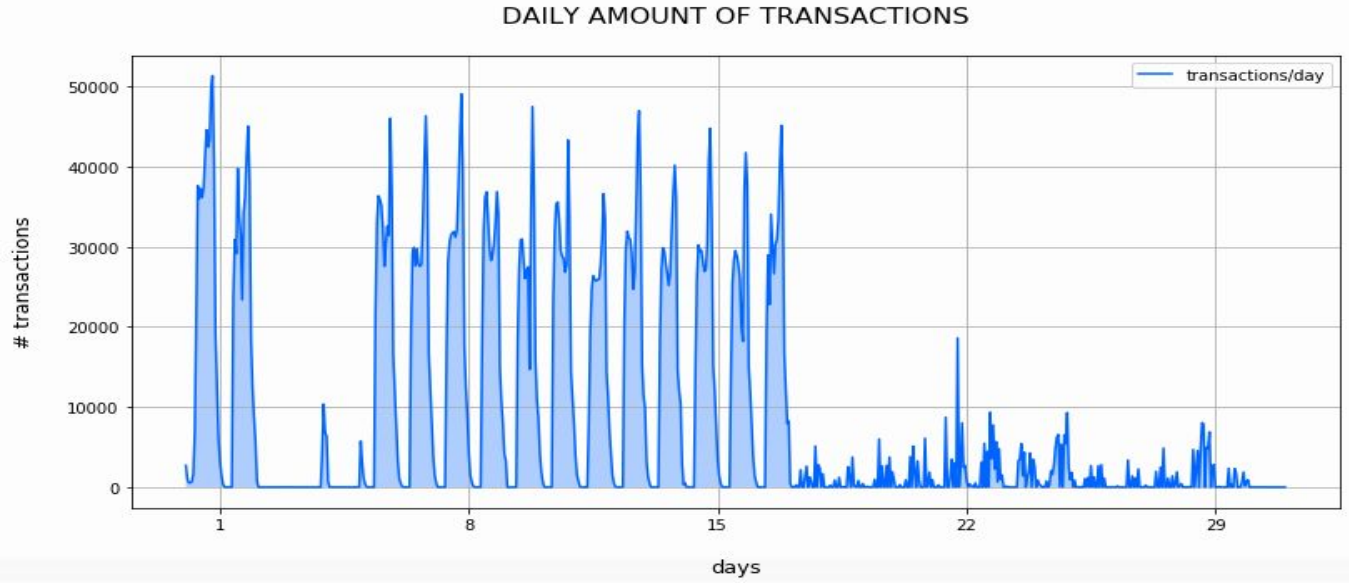


Fig.3 - Number of transactions that occur each hour for each day in the sample month.

In the Figure 3, the indicator of the y axis refers to the amount of transactions while the indicator of the x axis shows the days in the sample month of the dataset. In order to improve the readability of the graph and get insights on the distribution along the weeks, the number of days were divided by 7, which represents the amount of days in a week.

The histogram shows two main trends dominating the first fortnite and the second fortnite each: in the first one there is an alternance in the amount of transactions between fifty thousands and zero while in the second half the amount goes over ten thousands only once, in the end of the third week.

We cannot really explain the radical change in the amount transactions for the second half of the month, due to our limited knowledge on the dataset, although a trend is clearly visible on the histogram: the alternance of highs and lows indicates the probable alternance of day and night between days, supposing that during the night the amount of transactions is lower than during the daylight.

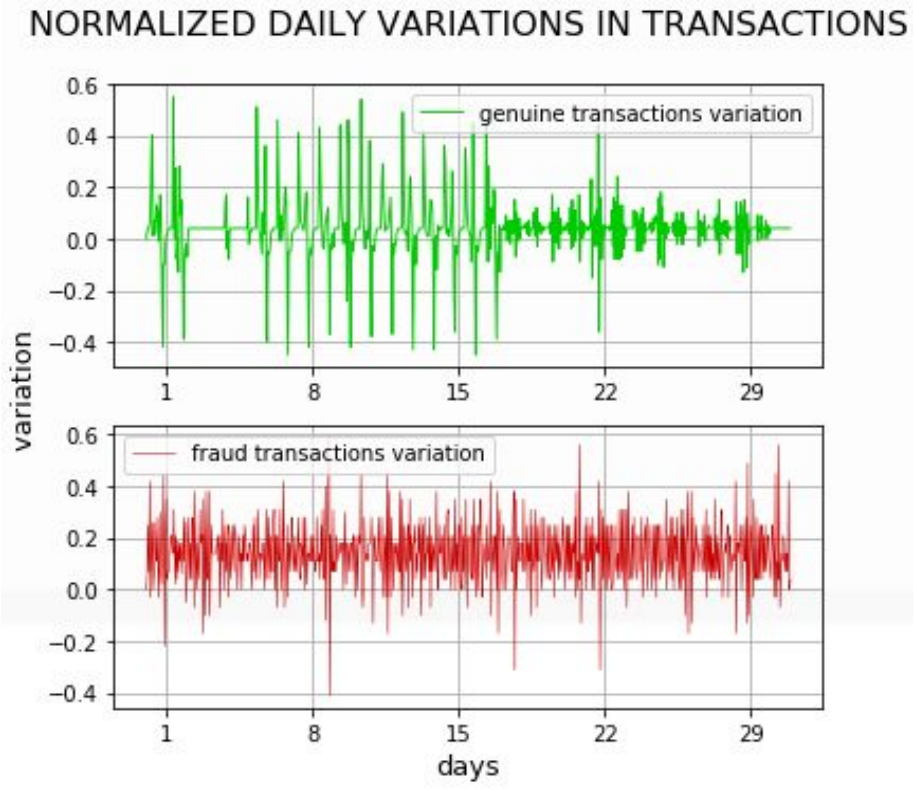


Fig.4 - Normalized daily variation day by day along the sample month.

In support of this claim, we can see that in the first half of the month, almost each week hosts a trend of exactly seven equally distributed periods of alternance.

Hence, a visualization of the daily variation on the amount of transactions was computed separately on fraud and genuine transactions (figure on the previous page). A clear trend is displayed for the genuine transactions, similar to the previous one about daily amount of transactions, although nothing of interest can be reported for fraud transactions. Actually it was possible to expect results like these because of the imbalance of the dataset and the relatively small amount of fraud transactions present in the dataset.

The investigation continued further with the analysis of normalized trends in fraud and genuine transactions grouped:

- on the single days of the month, to monitor overall activity in terms of transactions.
- on the single days of the week, to spot hot days of the week on average and maybe infer the weekend and the rest of the week.
- on the single hours of the day, to spot hot hours of the day on average and maybe infer the hours of dark and the hours of light.

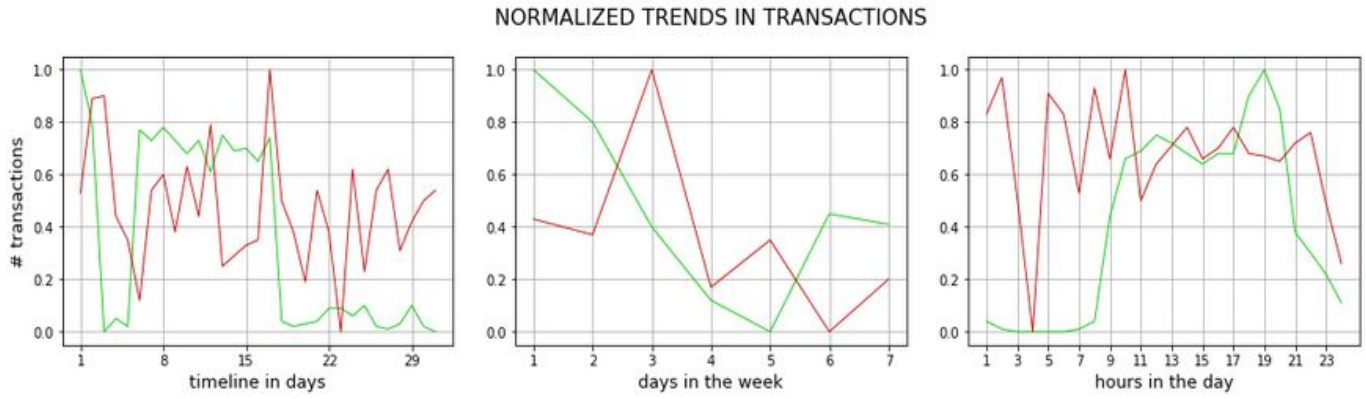


Fig.5 - Genuine and fraud transactions normalized trends along three different timelines: days in the month, days in the week and hours in the day.

The first histogram is the less informative since its knowledge relies on the data of a single month grouped per hour and shows that the hottest period for genuine transactions begins in the last part of the first week, continues for all the second week and lasts till the beginning of the third week. Along with it, trends for fraud transactions tend to be higher in the middle of the week and become lower towards the end of the same week on average.

However it is necessary to take into account that we highlighted weeks as 7-days period: it is possible to gather information and make suppositions about the beginning and the end of the week only with the second histogram. It averages day per day of each week the amount of transactions normalizing and splitting them between fraud and genuine ones. In order to get a generic trend for the overall transactions we have to look at the genuine trend: it presents the high peak on day 1 and a low peak on day 5 with a significant decrease between days 2 and 4 and a remarkable increase between days 7 and 1. In contrast, fraud transactions concentrate on day 3 and have the lower peak in day 6.

The analysis of trends is completed with the third and last histogram that represents the normalized amounts of transactions hour by hour in each day. Because of the number of retrieved observations - 31*n, with n the average number of transactions per day - this represents the most reliable statistic among the three considered. Trends for transactions in general can be associated to the genuine transactions and we can observe that most of these transactions happen between hour 10 and 20 for then dramatically decrease and stay low between hour 23 and 8. It is possible to infer that the position of the hours is realistic as it is but with no certainty.

Generally, fraud transactions seem to have no specific trend or, in case they have, it should not be taken into high consideration due to the quite small amount of available data these fraud statistics rely on, nonetheless having reliable insights about the genuine transactions can give a significant help in finding the fraud ones.

Type. This field indicates the type of transaction, among which we can distinguish payments, debits, cash-ins, cash-outs and transfers. Two kinds of analysis were carried out on this field, that is understanding the main differences in distribution among genuine and fraud transactions and the observation of the correlation among the fields for each kind of transaction separately.

Formerly, the analysis on the genuine and fraud distribution over this field were carried out. The histogram, which summarizes all the information and statistics about distributions of all the available transaction types, can be found below and reveals that this feature is a highly discriminatory indicator for fraud detection since it provides a proof that CASH-IN, DEBIT and PAYMENT operations cannot be part of a malicious behaviour, conversely CASH-OUT and TRANSFER operations do in similar percentages too.

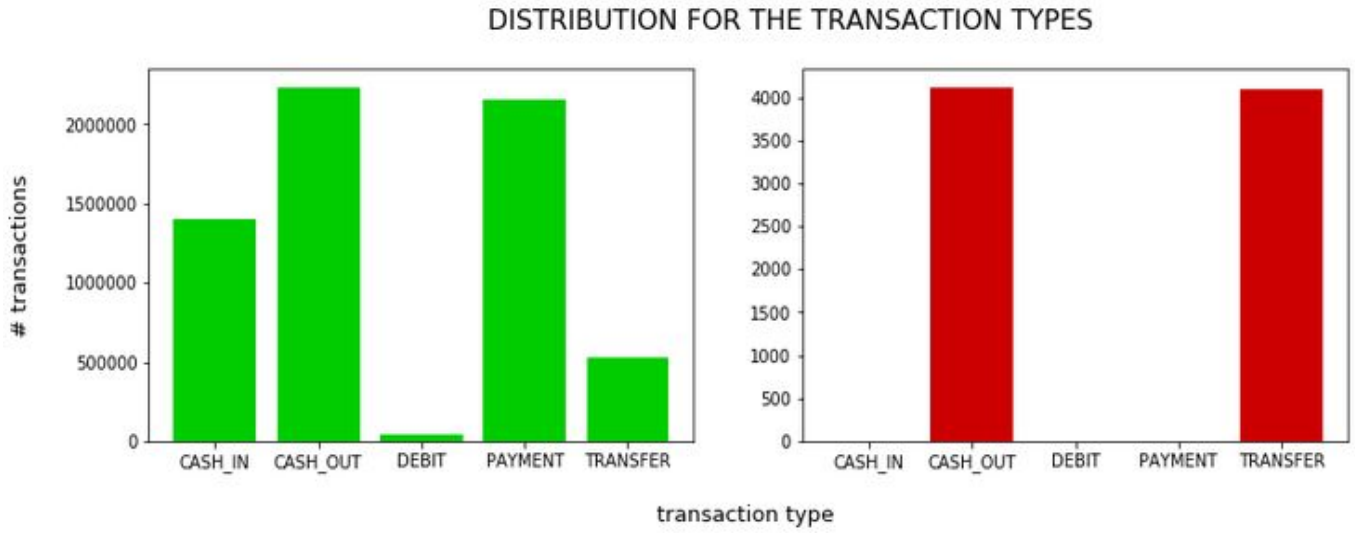
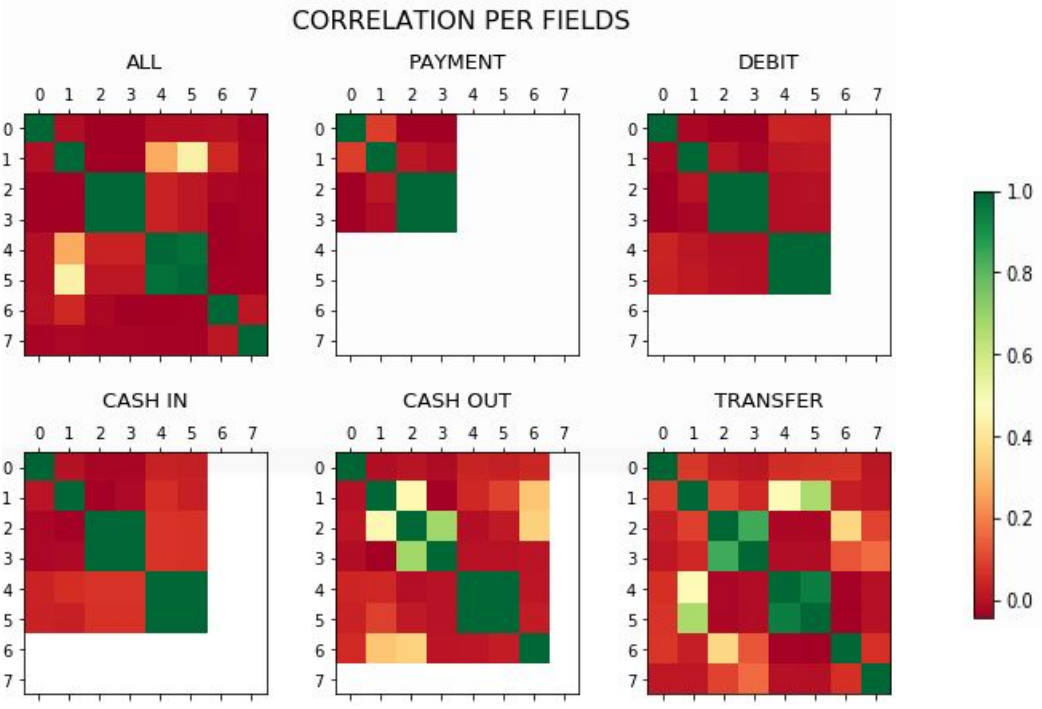


Fig.11 - Overall distribution for the transaction type for genuine and fraud transactions.

Moreover, the probability that TRANSFER operations are involved in hacking/fraudulent attacks is higher than for CASH-OUT ones as soon as the amount of these is much higher than the TRANSFER ones found among the genuine transactions.

Going down in details of the fraud transaction types, the exact cardinalities for the CASH-OUT and TRANSFER transactions are 4116 and 4097 respectively. Looking at the fraud transactions ordered by time, each TRANSFER fraud operation is followed by a paired CASH-OUT fraud operation in the same hour too, which can let us infer how fraud transactions operate in this dataset: first the fraudster transfers the money to another account - maybe a disposable one created on purpose, or maybe belonging to another user (we'll catch this shade later in the accounts analysis) - and then retrieves all the money with a CASH-OUT operation. The exploited account is just needed to make his tracks disappear (potentially framing other people).

Furthermore it is noticeable that almost 1:1 ratio shows that 0.3% of the times (13/4116) the CASH-OUT operation in a fraud context is executed without having the money transferred on another account first, but this does not affect the standard trend of TRANSFER --> CASH-OUT operation in sequence.



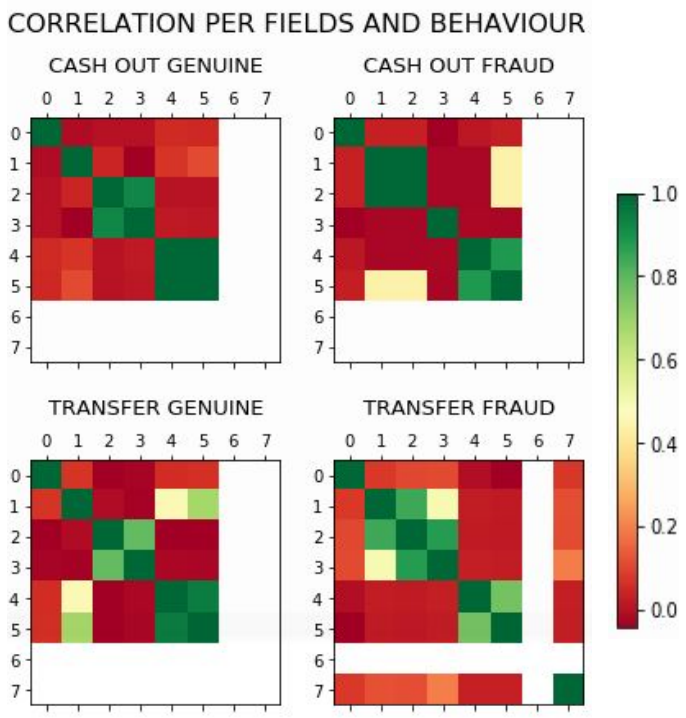
Tab.5 - The Correlation Matrix for the generic dataset and for each specific type of transaction.

N.B.: Available correlation features are step (0), amount (1), oldbalanceOrg (2), newbalanceOrg (3), oldbalanceDest (4), newbalanceDest (5), isFraud (6), isFlaggedFraud (7).

Latterly, getting insights on the correlation among the fields for each kind of transaction can reveal useful to better understand the data since the previous analysis on correlation in the entire dataset can be misleading even if gives us a general impression of the dataset as a whole. Histograms related to this investigation can be found in Tab.5 at the previous page.

We can see that correlation values for balances (old and new) for each transactions are always paired with high values. Further, PAYMENT, DEBIT and CASH-IN transactions are quite similar in correlation values, while CASH-OUT and TRANSFER operations present some dissimilarities, among which we can underline the interesting correlation between:

- "amount" and "oldbalanceOrg" fields, "isFraud" and both "amount" and "oldbalanceOrg" fields for CASH-OUT operations;
- "amount" and both "oldbalanceDest" and "newbalanceDest" fields, "isFraud" and "oldbalanceOrg" fields for TRANSFER operations.



Since the unexpected correlations were found for the specific transactions potentially affected by fraudulent behaviour, it can be interesting to get more insights about the differences between genuine and fraud transactions, maybe finding out some correlation which involves behaviour too.

Tab.6 - The Correlation Matrix for genuine and fraud CASH-OUT and TRANSFER transactions.

N.B.: Available correlation features are step (0), amount (1), oldbalanceOrg (2), newbalanceOrg (3), oldbalanceDest (4), newbalanceDest (5), isFraud (6), isFlaggedFraud (7).

Heatmaps show that such dissimilarities are due to the presence of the fraudulent transactions. In particular the main differences that characterize the behaviour in the two types of transactions are the following:

- CASH-OUT transactions: fraud ones have the correlation values for “newbalanceDest” and both “amount” and “oldbalanceOrg” higher than the genuine ones, for “amount” and “oldbalanceOrg” higher than the genuine ones, “newbalanceOrig” and “oldbalanceOrg” lower than the genuine ones.
- TRANSFER transactions: fraud ones have the correlation values for “amount” and both “oldbalanceDest” and “newbalanceDest” lower than the genuine ones, for “amount” and both “oldbalanceOrg” and “newbalanceOrig” higher than the genuine ones.

Only evident variations have been highlighted in this analysis: lesser ones have been neglected. These insights about differences in correlation can be used for the design of the prediction model.

Amount. Statistics that can be carried out to gain insights about this feature include the analysis over average values for this field over the whole dataset, then focusing only on fraud and genuine transactions separately. Then, with the help of standard deviation we can quantify how much the data are spread out around the averages that we have previously calculated, while quantiles can help too for the understanding of the distribution of the data under analysis.

All pieces of information are issued in the Tab.7 located at the next page. First of all the value relative to the “count” metric is expected to be much higher for genuine transactions since it represents the amount, and we also know that the dataset is heavily imbalanced.

METRIC (statistical)	**GENUINE** (transaction)	**FRAUD** (transaction)
count	6,354,407	8,213
mean	178,197	1,467,967
std dev	596,237	2,404,253
min	0.01	0
25%	13,368	127,091
50%	74,684	441,423
75%	208,364	1,517,771
max	92,445,520	10,000,000

Tab.7 - Statistical metrics describing both genuine and fraud transactions along with their differences.

Then the "mean" metric shows us insights regarding the fact that fraudulent transactions involve more money during the transactions, further remarked by the values obtained for the quantiles.

Hence, we computed the values for "quantile" metrics and we can compare these with the average value to get some insights regarding the skewness of the distributions. In this case, for genuine transactions (but even more for fraud transactions) it is reasonable to have a positive high skewness due to the fact that the average value is nearer to the 75% than to the 50%, which means that there are lots of values which are lower than the average and some very high values which allow the average to increase and generate the positively skewed distribution that we observe in these data.

About this fact, standard deviation can give us some more insights: its value tells us how much the values are spread out from the average. Both have relatively high values for this metric, fraud transactions even more respect to the genuine ones, although this value is highly affected by the overall amount of transactions for both, which makes the value obtained for genuine transactions more reliable than fraud ones. The problem with the difference in these statistics can be reconducible to the imbalance of the dataset.

Unexpected values can belong to the “min” metric, which is fixed to 0, and this means that there are some fraud transactions that involve no money, which sounds awkward enough in a financial banking context. The number of transactions that assume the mentioned value have been counted and are 16 for the fraud transactions. However this value is explainable since there are some fraudulent transactions that have been detected while these were being carried out and were consequently halted by the financial system. Every change with the relative account was rollbacked and the logs (our dataset) was updated with a total amount of 0 for the detected fraud transactions - this explanation can be found in the official PhD Thesis of the dataset creator [2].

A higher value for the max metric for genuine transactions instead of fraud transactions is appropriate enough and gives us no further insights of interest for our research task, hence it is not considered.

OldbalanceOrg, NewbalanceOrig, OldbalanceDest, NewbalanceDest.
Further, the Exploratory Data Analysis moved towards these fields, that represent the old and the new balance of the actor who started the transaction and the old and the new balance of the actor who ended the transaction respectively.

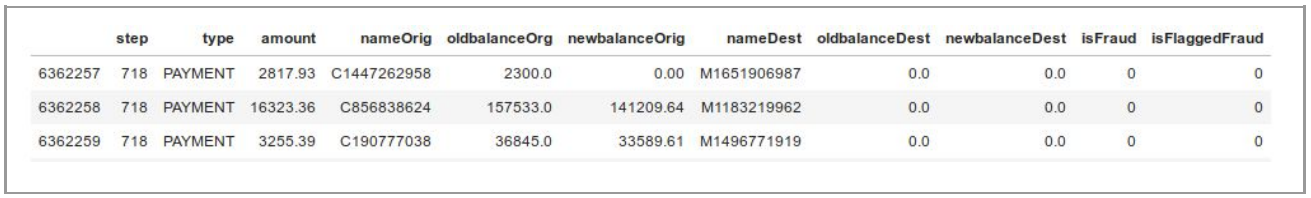
Among these fields, some correlation seems to exist and it is usually associated to the “amount” field. As we anticipated in the introduction, some hidden rules, still not so evident at first sight and which the single transactions follow, can be inferred by looking more deeply in detail at the data present in the dataset for each transactions.

These rules vary according to the type of transaction. For this reason, we are going to analyze the resulted rules according to the type they are associated, following:

- PAYMENT. This kind of transaction involves merchants, towards which the customer buys goods or services. Hence, the values for “oldbalanceDest” and “newbalanceDest” are always 0 because the dataset does not record the balance alterations of the merchant. Still on the customer’s old and new balance applies the following relation:

  “ oldbalanceOrg - newbalanceOrig □ amount “.

  This is a disequation because the customer who has bought goods or services from the merchant can partially make the payment through balance transaction and pay the rest using some other means.



	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
6362257	718	PAYMENT	2817.93	C1447262958	2300.0	0.00	M1651906987	0.0	0.0	0	0
6362258	718	PAYMENT	16323.36	C856838624	157533.0	141209.64	M1183219962	0.0	0.0	0	0
6362259	718	PAYMENT	3255.39	C190777038	36845.0	33589.61	M1496771919	0.0	0.0	0	0

Fig.6 - Sample of PAYMENT transactions from the dataset: it shows how relation applies.

- DEBIT. Data regarding the balances of the customers who get involved in “debit” transaction, before and after it takes place, is controlled by the following relation:

  “ (oldbalanceOrg - newbalanceOrig) □ amount = (newbalanceDest - oldbalanceDest) “,

  whereas the left-side part of the relation includes the balance update for whom needs to pay the debt, while the right-side part includes the balance update for whom has to be paid the debt to. The first part of the relation has the same fashion of the “payment” transaction kind, conversely the opposite part (that receives the money) always obtains the total amount at the end of the transaction.

  However this second part does not apply for a total of 4.92% of the DEBIT transactions: it wasn’t possible to figure out the reason why it happens.

We can neither think to a possible correlation with a malicious behaviour since this kind of transaction is not affected by them: there is some other underlying relation that is not so easily inferable.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
6359887	715	DEBIT	3970.86	C437662144	26011.0	22040.14	C1912265217	203629.40	207600.26	0	0
6360026	715	DEBIT	1373.41	C552365302	3195.0	1821.59	C1380482434	7857.34	9230.76	0	0
6360354	715	DEBIT	12543.72	C839662403	45711.0	33167.28	C574299830	552863.02	565406.74	0	0

Fig.7 - Sample of DEBIT transactions from the dataset: it shows how the relation applies.

- CASH-IN. The relation at the basis of the values for the "Orig" customer's balance before and after this transaction happens is the following:

  " (oldbalanceDest - newbalanceDest) □ amount = (newbalanceOrig - oldbalanceOrg) ",

  whereas this time the "Dest" customer can decide whether paying the "Orig" customer using the balance of his account or any other means. In the first case his bank account will become empty after the transaction (if it was not already empty) while in the second case it will make the balance empty (or just leave it empty if it is already), which explains the disequation existing in the relation.

  Same as the previous kind of transactions, there is a percentage of outliers which do not follow this rule: in particular it is 2.43% among all the CASH-IN transactions.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
6361804	718	CASH_IN	29151.75	C1961863238	175066.0	204217.75	C1224091611	152526.95	123375.20	0	0
6361806	718	CASH_IN	62473.20	C1444587517	554.0	63027.20	C71401441	787299.61	724826.40	0	0
6361807	718	CASH_IN	198362.49	C1597937888	26395.0	224757.49	C1936903624	888247.24	689884.76	0	0

Fig.8 - Sample of CASH-IN transactions from the dataset: it shows how the relation applies.

- CASH-OUT. Data regarding the balance of the customer, who needs to pay the amount of money, before and after the transaction happens, is controlled by the same relation that was employed for the DEBIT transaction. Similarly, after the transaction ends, both require that the money are decreased in the balance of the Orig customer and are increased in the balance of Dest customer, following the same rules already explained.

  Outliers of this rule appear for the CASH-OUT operation too, in fact it results in 6.04% of the total transactions of this kind on which this rule does not apply.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
362521	735	CASH_OUT	417103.68	C1450763584	417103.68	0.0	C1377830519	34232.06	451335.75	1	0
362523	735	CASH_OUT	92735.71	C786761311	92735.71	0.0	C570188819	921583.30	1014319.01	1	0
362525	735	CASH_OUT	123146.28	C981071931	123146.28	0.0	C64963279	0.00	123146.28	1	0

Fig.9 - Sample of CASH-OUT transactions from the dataset: it shows how relation applies.

- TRANSFER. These kind of transactions operates in the same fashion as CASH-OUT ones: refer to the previous sections in this list to get insights on the rules underneath them. The correspondent rule is affected by outliers by 6.94% on the overall TRANSFER transactions.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
118855	538	TRANSFER	508537.73	C640267507	16508.00	0.0	C1259964551	147505.82	656043.55	0	0
118863	538	TRANSFER	2459146.39	C702167860	30149.00	0.0	C190204881	0.00	2459146.39	0	0
118928	538	TRANSFER	1617984.20	C11957124	253187.00	0.0	C656281757	0.00	1617984.20	0	0

Fig.10 - Sample of TRANSFER transactions from the dataset: it shows how relation applies.

These relations indicate that some correlation exists between the balances before and after the transactions are carried out, involving the amount: this represents an underlying relationship between these values that can be useful for the final considerations about the feature selection task.

All the insights gained in this step of the Exploratory Data Analysis are summarized in the following table:

TYPE (of transaction)	**INFERRED RULES** (underlying balances before and after the transaction)	**OUTLIERS** (percentage)
PAYMENT	• oldbalanceOrg - newbalanceOrig □ amount	0.00 % (0.00% overall)
DEBIT	• oldbalanceOrg - newbalanceOrig □ amount • amount = newbalanceDest - oldbalanceDest	4.92 % (0.03% overall)
CASH IN	• oldbalanceDest - newbalanceDest □ amount • amount = newbalanceOrig - oldbalanceOrg	2.43 % (0.53% overall)
CASH OUT	• oldbalanceOrg - newbalanceOrig □ amount • amount = newbalanceDest - oldbalanceDest	6.04 % (2.13% overall)
TRANSFER	• oldbalanceOrg - newbalanceOrig □ amount • amount = newbalanceDest - oldbalanceDest	6.94 % (0.58% overall)
		(3.27 % overall)

Tab.8 - Summary table for inferred rules divided per type of transactions and reporting the quantity of outliers which don't follow these rules in shape of percentages locally to the transaction type amount and over the entire dataset.

NameOrig, NameDest. These fields represent respectively the name of the customer who executed the transaction and the one who was addressed the transaction to. The former contains only customers while the latter one can contain both customers and merchants. In particular, these two classes can be recognizable by the first letter, which can be a "C" for customer or an "M" for merchant.

The distribution between merchants and customers for the "nameDest" field is equal to the one between PAYMENT transactions and all the other transactions, since every transaction of this type necessarily involves a merchant in order to be carried out.

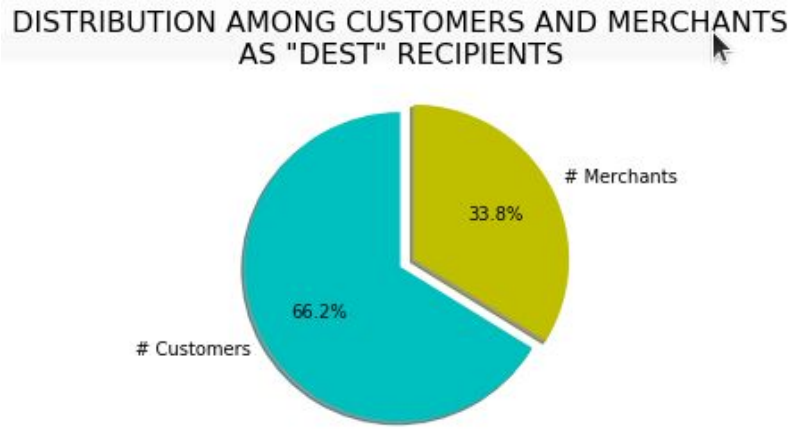


Fig.11 - Visualization of the amounts of merchants and customers as "Dest" recipients for each transaction. Customers are depicted in cyan and merchants in yellow.

Here in the pie chart in Fig.11, this distribution is shown, with 33.8% for the quantity of merchants as recipients and with 66.2% for the generic customers.

Further in the analysis we computed couple of measurements regarding both the fields such as what is the average for each customer involved in the transactions, however this insight will not assure us that the fraudulent behaviour is concentrated on few customers or is just an event that exceptionally happens to standard customers, for this reason we are also interested in understanding fraud transactions whom belong to.

The former question is answered from Tab.9 (next page) and it reports the fact that in general we have a very low values for ratio, which for "Orig" customers is approximately 1 for both fraud and genuine transactions. It is not possible to make out some statistics about the behaviour of these customers, if we know just one transactions for each of them on average, hence it will not be much useful for the prediction task. Something similar happens for "Dest" customers too, even if the value reported for the ratio gains reveals slightly better with 2.3 transactions per customer on average.

		GENUINE	FRAUD
Total Transactions		6,354,407	8,213
"Orig"	Unique Customers	6,345,122 (99.85 %)	8,213 (100 %)
	Transactions / Customers (Ratio)	**~1.0**	**1.0**
"Dest"	Unique Customers	2,719,685 (42.8 %)	8,169 (99.46 %)
	Transactions / Customers (Ratio)	**~2.3**	**~1.0**

Tab.9 - Shows the amount of unique customers per genuine and fraud transaction and how many transactions have been carried out from each customer on average, comparable with the total amount of transactions. Values are computed for both customers who started the transaction and recipients.

The latter question is answered in Tab.9, when focusing on the FRAUD column: ratio is almost 1, both for "Orig" and "Dest" customers. It can't be for sure less than 1 for each instance, otherwise it would neither be in this list, this is why we can say that both the fields "nameOrig" and "nameDest" are of no use for the purpose of this document.

IsFlaggedFraud. This field was considered to indicate all those transactions which were recognized by the recognition system as fraud and have been blocked, then rollbacked all the transactions in the same hour in order to avoid that the fraudulent client reached money and transferred them elsewhere with consequent cashout operation, as we have already seen in the previous chapters of this Exploratory Data Analysis.

We discovered that among all the more than 6 million tuples which can be found in the dataset, only 16 have this field as flagged 1 and the other with 0 value. The level of discriminatory nature for this field is almost null, hence the impact that can have on this analysis is irrelevant.

3. Data Preprocessing

The entire process of Exploratory Data Analysis lets us bring to light most of the latent information hidden in the dataset along the tuples and across all the fields. It highlighted all the potentially relevant correlations between the fields and revealed itself useful for the process of feature selection.

At this point follows the significant step of **Data Preprocessing**: it is needed to prepare data, carrying the most useful data to be treated for the prediction model and leaving all the information that can be of no use because lacking of correlation with the class field or being redundant with the other selected features (that have higher relevance too).

This phase of Data Preprocessing will go along two different steps that will be carried out in sequence, which are:

- the former one will consider some minor fixes to carry out on the names of the fields, consequent feature selection and the following removal of some fields, which proved less purposeful during the Exploratory Data Analysis phase;
- the latter one will involve the problem linked to the imbalance of the dataset, detected at the beginning of the Exploratory Data Analysis, and that can lead to potential alterations regarding the amount of tuples employed for the training step of the model and consequently for the step of test too.

In the end, the final result of the preprocessing phase will be the updated dataset, ready to be input in the further designed model. We're going to approach these steps one by one in the next two subsections.

3.1. Preprocessing along fields

This is the first step of the Preprocessing phase and will involve questioning what are the fields relevant for our research task according to what we already have discovered to be a useful indicator of the class field ("isFraud").

The first thing to do is changing the name to the of the field "oldbalanceOrg" and replace it with "oldbalanceOrig", since it refers correspondingly to "newbalanceOrig" as well as "oldbalanceDest" and "newbalanceDest". Now we can consider each field one by one and decide whether to bring or leave it.

Step. The correlation between this field and the class field corresponds to 0.032 and it is one of the highest correlated to the class field among all the ones available for the prediction, hence we should take it in high consideration. However it shouldn't be taken as it is since it takes into consideration both the specific month, day by day and hour by hour.

As we have seen, the distribution along the month is much different, splitting into three different trends the amount of transactions (specifically, the first one ends at the half of the same week, the second one ends at the beginning of the third week, the third one lasts till the end of the month), for this reason using the field as it is would result in overfitting the dataset.

Instead, we can take into consideration the possibility of taking apart from this field the information about hour and the information about weekly days that we already obtained through the EDA. Information about the day of month overfits the data because we have one sample per each instance of "day of the month" in the dataset, on the other hand, having information regarding the day of the week will be useful in 4 different periods of the month, even more with the hours of the day since in a month there are 31 repetitions of the hours.

Type. This field is highly discriminatory for the prediction because of the fact that fraud transactions include only two types of transaction, that are the TRANSFER transaction and the CASH_OUT one, usually the former right followed the latter in sequence.

On the other hand, the other three transaction types, namely PAYMENT, DEBIT and CASH_IN, are never affected by any kind of fraudulent behaviour. For these reasons, we will keep this field among the features to consider for the prediction task.

Amount. The correlation between this field and the class field corresponds to 0.077 and it is the highest correlation value among all the observed. This value is quite reasonable since we have seen that fraud transactions involve bigger and bigger amounts of money respect to the genuine one on average, with 1:8 ratio (178,197 and 1,467,967 respectively), despite the imbalance of the dataset and the maximum value for amounts which reveals higher for the genuine transactions. For this reason, we will keep this field among the features to consider for the prediction task.

NameOrig and **NameDest**. The correlation between the class field and "nameOrig" field is approximable with 0 since for each transaction we have a new customer, as a consequence it is not possible to have an estimate for any trend about the customers, for this reason this field will not be considered for the prediction task.

Something similar can be said about the "nameDest" field. Its correlation value with the class field is slightly better than the one for the "nameOrig" just because we have on slightly higher average of 2.33 transactions per unique customer on the entire dataset. Still nothing of high interest can be obtained with such low frequency, for this reason it will not be considered for the prediction task as well as the "nameOrig" field.

OldbalanceOrig and **NewbalanceOrig**. The correlation between the class field and the "oldbalanceOrig" field is 0.010 while the correlation with the "newbalanceOrig" field is -0.008. From the Exploratory Data Analysis we know two main things:

- The former one is that, for most of the transaction instances, these two fields are linked to each other through some means of "heuristic rules", and which vary according to the transaction type. This leads us to carry only one of them, since the information about one gives us insights on the information about the other.
- The latter one regards the fact that these fields are low related to the "amount" field when considering the overall dataset but become tightly related when only TRANSFER and CASH_OUT transactions are considered apart from the other types. Further, we also know that in TRANSFER operations the two fields under examination have a high correlation value while in CASH_OUT operations the same have a low correlation. This leads us to carry both of them, in order to avoid losing useful features to infer the fraudulent behaviour.

For these reasons, it was decided to carry them both in the dataset to be used for the learning process. However we can also think of transforming the two fields in one single field which is the composition of them, and that is obtained by the following formula: "newbalanceOrig - oldbalanceOrig", which highlights the difference in balances before and after the transactions takes place.

FIELD	CORRELATION
step	-0.007
amount	-0.102
oldbalanceOrig	0.221
newbalanceOrig	0.268
oldbalanceDest	0.047
newbalanceDest	0.006
isFlaggedFraud	-0.0002
isFraud	**-0.362**

Tab.10 - Correlation between the new generated field and the others.

The creation of this new feature lead us to investigate on the correlation of the new feature, and which we called “balanceOrig”. Results were very positive: values are summarized in Tab.10 at the previous page.

As we can see, the correlation with the class field is much higher than any other feature, for this reason we’re going to substitute it, hence removing both two fields “oldbalanceOrig” and “newbalanceOrig”.

OldbalanceDest and **NewbalanceDest**. The correlation between the class field and the “oldbalanceDest” field is -0.006 while the correlation with the “newbalanceDest” field is 0.001. From the Exploratory Data Analysis we know that these two values are correlated with the “amount” field.

Trying to generate another feature as done for “oldbalanceOrig” and “newbalanceOrig” is interesting but revealed useless since its correlation with the class field is equivalent to “0.027” and correlation with the “amount” field is “0.846” which makes it of no use to our research work.

IsFlaggedFraud. The correlation between this field and the class field equals to 0.044, which is the second higher correlation value. However, despite this insight, the Exploratory Data Analysis revealed that this field is not discriminatory as it can be thought, as a consequence of the fact that 99% of the values assigned to this field are all 0 for transactions, for this reason we’re not carrying this field in the feature space for the prediction task.

In the end of this Preprocessing step carried out along fields, the obtained feature space set for the prediction task is composed by the following features:

$$FS = \{ \text{“step”}, \text{“type”}, \text{“amount”}, \text{“balanceOrig”} \},$$

while the class set is composed by the following of features:

$$C = \{ \text{“isFraud”} \}.$$

3.2. Preprocessing along tuples

This is the second step of the Preprocessing phase and will deal with the imbalance of the dataset that was detected at the beginning of the Exploratory Data Analysis, and will eventually consider to provide an alteration to the cardinalities of the dataset to rebalance the boolean value for the class field ("genuine" or "fraud").

The imbalance of the dataset is one of the first problems that was detected at the beginning directly from the distribution of the class value among genuine transactions and fraud ones. Using the original dataset will cause overfitting as a main issue, in fact our classification models will assume that in most cases there are no frauds. Differently, what we want for our model is to be certain when a fraud occurs.

According to Kotsiantis et al. in [3], a review about dealing with the imbalance of the datasets published in 2006, it is possible to deal with this issue exploiting one among three main data levels methods:

- Random undersampling, which aims to balance the class distribution through the random elimination of majority class examples.
- Random oversampling, which aims to balance the class distribution through the random replication of minority class examples.
- Feature selection for imbalanced datasets, that proposes a feature selection framework, which selects features for positive and negative classes separately and then explicitly combines them.

All three possibilities are feasible in this research work, the former that would consist in removing part of the genuine transactions in order to rebalance the ratio in the class field "isFraud", the second that can be considered obtaining PaySim from the original authors and simulating new synthetic fraud

transactions to be added inside the dataset, the latter one has been partly already considered in this research work during the Exploratory Data Analysis.

The decision falls on the the first method because it is the most straightforward and reliable, while the second one can be prone to errors if the replication is not carried out correctly. The third method would take into consideration the reconsideration of the entire Exploratory Data Analysis: for this reason it is not taken into consideration. In any case it will be applied onto the test set.

However, before proceeding with Random Undersampling, it is necessary to deal with a couple of issues on the cardinalities of values for each field. In particular, the issues regard the following fields of interest:

- "type": this is the field that distinguishes any transaction among all transaction types. As a consequence of both the facts that we want to distinguish fraud transactions from genuine one (and not the opposite, which is remarkably different), and that PAYMENT, DEBIT and CASH-IN transactions are not affected by any fraudulent behaviour, hence we can neglect the consideration of these in the prediction task.
  In other words, we specifically want to distinguish fraud CASH-OUT transactions from genuine ones and fraud TRANSFER transactions from genuine ones. For this reason, the classifier is not needed for the other transactions type, hence we can remove all the transactions that are tagged with those transaction types.
- "isFlaggedFraud": this field was not carried out in the feature space because the distribution of values it presented was almost null. In the description of the dataset in the PhD thesis of the author it is reported that transactions where "isFlaggedFraud" equals 1 have been altered in the field values, for this reason it's correct to avoid taking the affected transactions into consideration (16 over 6+ million in total).

- “oldbalanceOrg”, “newbalanceOrig”, “old/newbalanceDest”: these fields contained the information regarding the balances before and after the transaction was carried out. During the Exploratory Data Analysis we realized that these values follow some specific “heuristic rules” according to the transaction type they belong to. The “heuristic-ness” of our rules determined some outlier transactions which make in total the 3.27% of the overall transactions.
  We decided to remove the “old/newbalanceDest” fields, and to compress “oldbalanceOrg” and “newbalanceOrig” into the “balanceOrig” field. Now the decision is about having that percentage of outliers or removing them and exploiting rules in some way. Moreover, with the removal of part of the transaction types, the remaining one share the same rule. For this research work, the made decision is to keep that percentage as a consequence of the fact that our rules are somewhat “heuristic”, but it can be interesting understanding whether the removal of that percentage can bring to an increase in performance of the prediction model.

In the end of this Preprocessing step carried out along tuples, the obtained cardinalities for the new dataset are listed in the following table:

TRANSACTIONS	GENUINE	FRAUD
TRANSFER	528,812	4,081
CASH-OUT	2,233,384	4,116
TOTAL	2,762,196	8,197

Tab.11 - Cardinalities for the new dataset among genuine and fraud transactions.

4. Data Transformation

The Data Preprocessing phase allowed us to get rid of useless information through a process of data cleaning first along the fields and then along the tuples, even providing dimensionality reduction of the dataset, in particular over the genuine transactions.

Thus, a step of Data Transformation is considered in order to transform the preprocessed dataset into:

- training set: it will be useful for the system to learn from the data by observing many samples of fraud and genuine transactions;
- test set: it will be suitable to understand how effectively the system learned on the data.

In order to solve this issue, we have to deal with the imbalance between fraud and genuine transactions, already revealed in the Data Preprocessing phase and despite the reduction of dimensionality operated on the genuine set. In fact, at the moment the ratio between genuine and fraud transactions is 1:337.

The method that we have applied is the Random Undersampling and it consists in eliminating random samples from the oversampled part of the dataset.

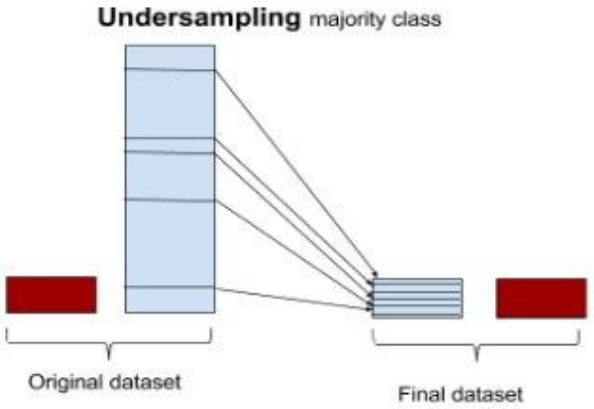


Fig.12 - Random Undersampling Technique carried out on the data, reducing genuine transactions in dimensionality, while leaving fraud ones unchanged in cardinality.

In our case we don't have the problem of temporality of data as soon as we identify only the hour in which the transaction took place through the preprocessing of the "step" field. Specifically we've considered 8,197 random genuine samples, which is the same amount of transactions with fraudulent behaviour.

Moreover, the ratio between TRANSFER and CASH-OUT operations has been kept constant: in this way we are balancing not only the behaviour but also the transaction type for each tuple. Sample transactions were selected independently of their position, avoiding sequences of tuples that may exhibit any kind of ordering.

Another operation that can be carried out on the so prepared data is the scaling of the features: as we can see, taken all the possible pairs of features, the dimensionality among them is quite different:

- "step" ranges in values from 0 to 23;
- "type" ranges in values included in the following set: { "TRANSFER", "CASH_OUT" };
- "amount" ranges in values from 0.0 to 92,445,516.64;
- "balanceOrig" ranges in values from -10,000,000.00 to 26,575.26.

However, the decision about whether scaling these amounts or leaving them as they are depend much on the employed machine learning algorithm, for this reason we'll figure this problem later in the data mining phase.

After this operation, data was split into training set and the test set. The fixed ratio employed for training set and test set was 4:1 (80% for training, 20% for test). As a consequence, data became ready to be input in the prediction model.

5. Data Mining

During the Data Transformation phase, we managed to transform the preprocessed data-frame into data, under the shape of training test, useful for the learning process, and test set, needed for the evaluation process, ready to be taken as input by the classifier on purpose for the prediction task.

Consequently to the process of Data Transformation, information is extracted from the data by applying machine learning algorithms in the Data Mining phase along the two steps of learning and evaluating from the training test and the test set respectively.

In this section the discussion will go along two different steps that will be considered one by one, which are:

- the former will consider both the dimensionality of the training/test set and the nature of the features on which the learning process will take place. As a consequence, will follow the process of searching the appropriate algorithms to confront the prediction task purposefully and avoiding issues like under/overfitting;
- the latter one will present the description of the metrics that were employed in order to evaluate the model, and further the results obtained by the two-step process of learning on the training set and evaluating on the test set.

After having gathered all the values for the metrics on each executed algorithm, in the next phase we are going to provide some understanding through the interpretation of the results and potentially new insights which regard the nature of the data.

5.1. Machine Learning algorithms

This is the first step of the Data Mining phase, which consists in understanding of the machine learning algorithms that can be applied to the available data about fraud transactions and the reason why we have considered them for this prediction task.

Machine learning algorithms are specifically selected according to two types of variables:

- the kind of data we are dealing with: in particular the treatment of data and the suitability of an algorithm instead of another changes dramatically according to the nature of the data, in case we are trying to learn on numerical data, decimal data, categorical data and so on.
- the dimensionality of the data: this variable influences a lot the probability of having under/overfitting models, that perfectly describe the dataset but that have problems in predicting brand new data due to the rather low generalization of the model.

First thing that we decided to consider is the kind of data, in particular we need to analyze the kind of data in the feature space and the kind of data in the class field. These are summarized in the following table:

Feature	Kind	Range
step	numerical, integer	[0 ... 23]
type	numerical, integer	{1, 2}
amount	numerical, decimal	[0.0 ... 92,445,516.64]
balanceOrig	numerical, decimal	[-10,000,000.00 ... 26,575.26]

Tab.12 - Summary of the characteristics of the features in the feature space, specifically the kind (numerical - categorical) and the range of the values assumed by the corresponding feature.

Furthermore, we are in presence of a binary classification because the prediction class can assume the two boolean values regarding the fraudulent or genuine behaviour for the transaction.

Despite the high values that we can observe in the range column of Tab.12, the dimensionality of both the data is much lower. As we can recall from the Data Transformation phase, we have a training and test set with an amount of transactions equal to 13,116.

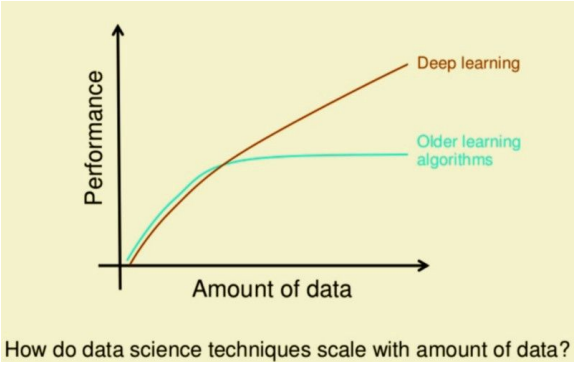


Fig.13 - Older learning algorithms perform better than DL with small-sized data.[9]

The amount of transactions is relatively small, we can consider it as a small dataset to learn on, for this reason we'll be discarding all those algorithms that require a lot of data - such as deep learning techniques - in order to focus on those that are known to provide remarkably good results with small datasets.

The first and two most straightforward machine learning algorithm to consider when dealing with low-dimensioned datasets is the Linear Regression and the Logistic Regression algorithms, which are employed in the supervised learning context and learn from the data through means of linear functions.

The difference between the two is the fact that in the former one the dependent variable is continuous while in the latter one the class field can assume discrete values. As soon as we are dealing with a problem of fraud detection, we are in presence of a boolean outcome ("fraud" or "genuine"), hence we've considered the Logistic Regression algorithm for the task.

9 https://towardsdatascience.com/breaking-the-curse-of-small-datasets-in-machine-learning-part-1-36f28b0c044d

The process for the selection of the appropriate algorithms started from the Scikit Learn space[10], in particular, a cheat sheet that helps specifically for this task. As the Scikit Learn algorithm cheat sheet reported in Fig.13 shows, throughout its path of search to find the best possible algorithm according to the available data, the first discrimination on the dimensionality of the data is at the beginning with less than 50 samples, on the other hand the second one which interests us more is the first split in the Classification section.

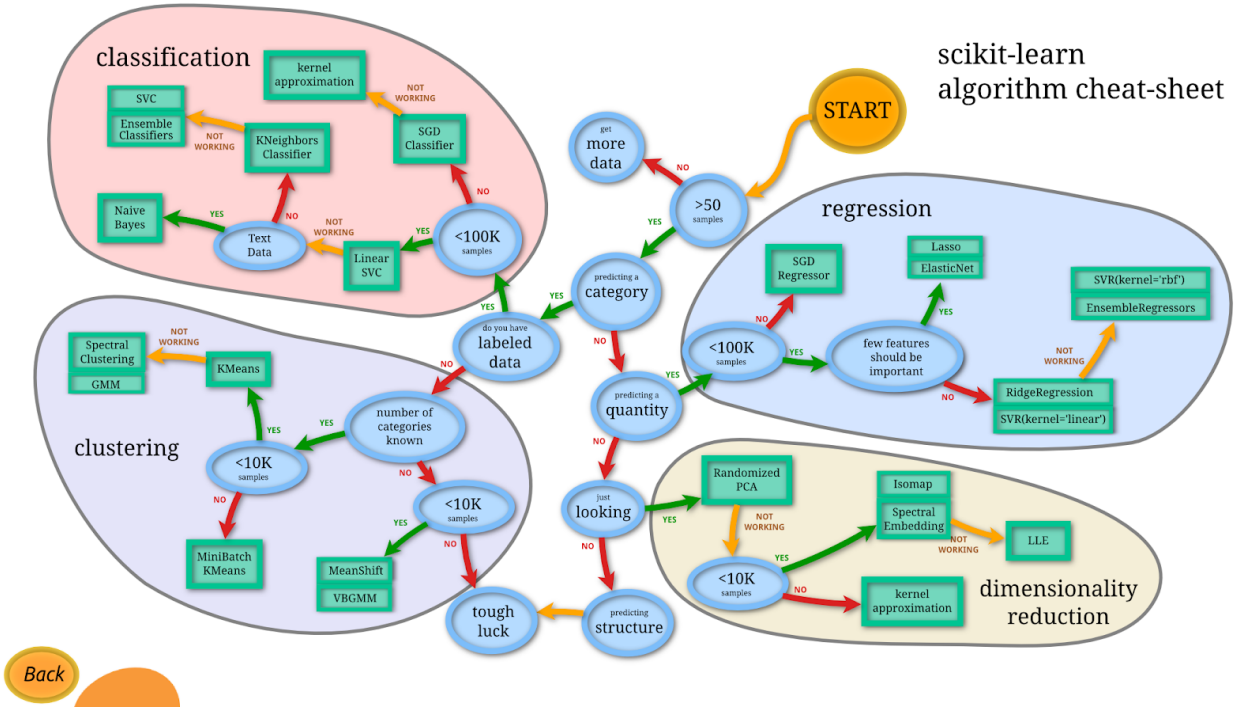


Fig.14 - The interactive scikit-learn algorithm cheat-sheet: it was employed as first source for finding machine learning algorithms appropriate for the research task of this document.

On this path, the examined machine learning algorithms are the Linear Support Vector Classifier, because we're dealing with less than 100,000 transactions, and the K-Neighbors Classifier, since we not concerned about learning on text data.

[10] https://scikit-learn.org/stable/

Apart from this figure, an algorithm which does not appear in this figure and that was tested on our dataset is the Decision Trees algorithm together with one of his ensemble variation, which is the Gradient Boosting Classifier.

Regression Trees were not taken into consideration because those are more suitable for problems in which the dependent variable is a continuous value rather than a discrete one, as it is for our case.

5.2. Metrics and Results

In this section first we're going to describe the metrics adopted for the evaluation of the models that learned on provided data, then we'll summarize the results gathered during the experiments.

On this path, it was possible to train the models on the data according to the five different machine learning algorithms, previously selected on purpose. These algorithms which can be used with small datasets, are prone to overfitting if the right countermeasures are not employed. In our case, during the preprocessing we balanced the dataset in order to avoid overfitting on purpose, for this reason at this advanced point of the research work it is possible for us to use standard metrics to evaluate the models. The metrics which were employed are the following:

- Confusion Matrix: squared 2x2 matrix composed of four principal metrics, that are:
    - True Positive (tp): amount of fraud transactions which were correctly classified by the model as genuine ones.
    - True Negative (tn): amount of genuine transactions which were correctly classified by the model as genuine ones.
    - False Positive (fp): amount of genuine transactions which were wrongly classified by the model as fraud ones.
    - False Negative (fn): amount of fraud transactions which were wrongly classified by the model as genuine ones.

- Precision: metric which measures how effective was the model in spotting the fraud transactions among fraud and genuine. It can be calculated with the following formula:

$$\text{Precision} = \frac{tp}{tp + fp}$$

- Recall: metric which measures how effective was the model in spotting the fraud transactions among all available fraud ones. It can be calculated with the following formula:

$$\text{Recall} = \frac{tp}{tp + fn}$$

- F1: metric which combines Precision and Recall in one single value. It can be calculated with the following formula:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Here in the following table we can observe all the obtained results by each algorithm for the research task:

	TP	FP	TN	FN	Prec.	Rec.	F1
Logistic Regression	1500	139	11	**1628**	0.921	0.993	**0.956**
Linear SVC	1526	113	**190**	1449	0.928	0.884	0.905
K-Nearest Neighbors	1518	121	31	1608	0.93	0.981	0.955
Decision Trees	**1534**	**105**	83	1556	**0.937**	0.949	0.943
Gradient Boosting C	1503	136	19	1620	0.923	**0.988**	0.954

Tab.13 - Summary table of the metrics employed for the evaluation.

6. Interpretation of the Results

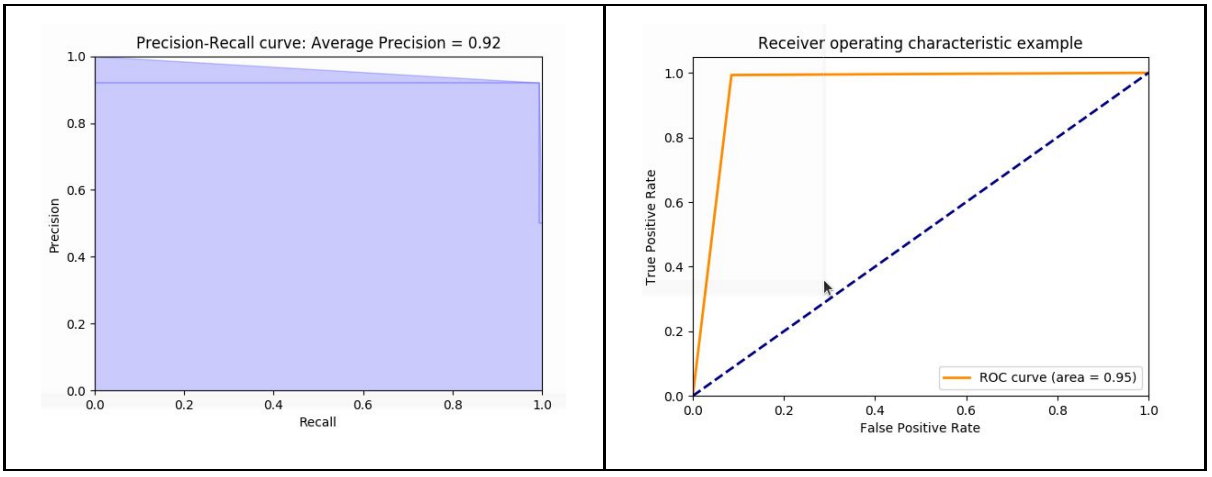
In the previous phase we focused on the process of learning from the data in order to make the classifier able to distinguish fraud transactions from genuine ones through means of machine learning algorithms.

In this last phase, the objective is to show the Interpretation of the Results through some visual representations, summing up what were the most interesting results according to the task that we described from the very beginning.

The visualization of the results is done through two commonly employed plots derived from the metrics that we previously calculated, in particular these are:

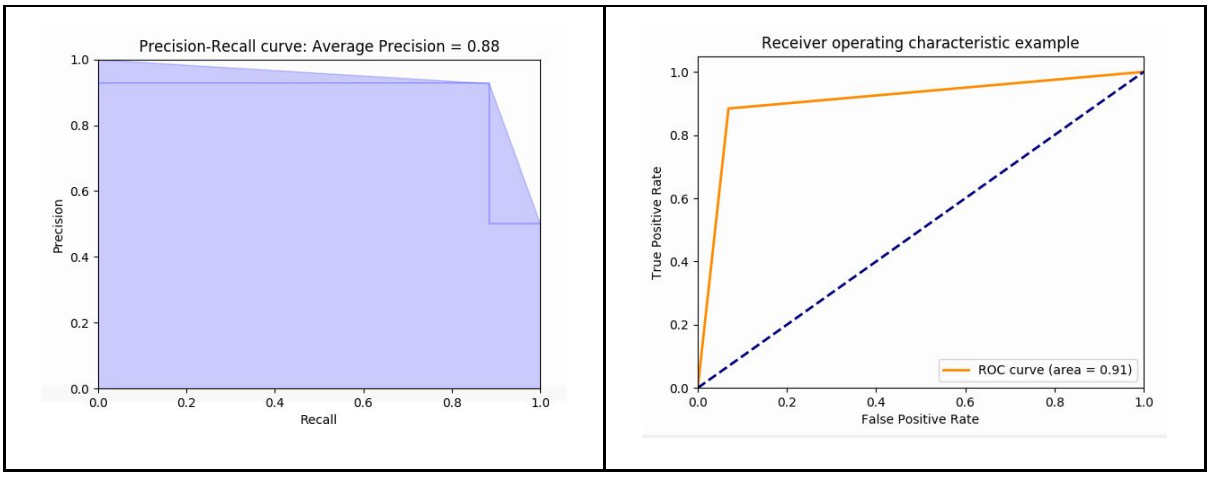
- the Precision-Recall Curve (PRC): evaluates visually the quality of the classifier output according to the trends for the Precision and Recall metrics.
- the Receiver Operating Characteristics (ROC): evaluates visually the quality of the classifier output according to the trends for the True Positive Rate and False Positive Rate.

Logistic Regression



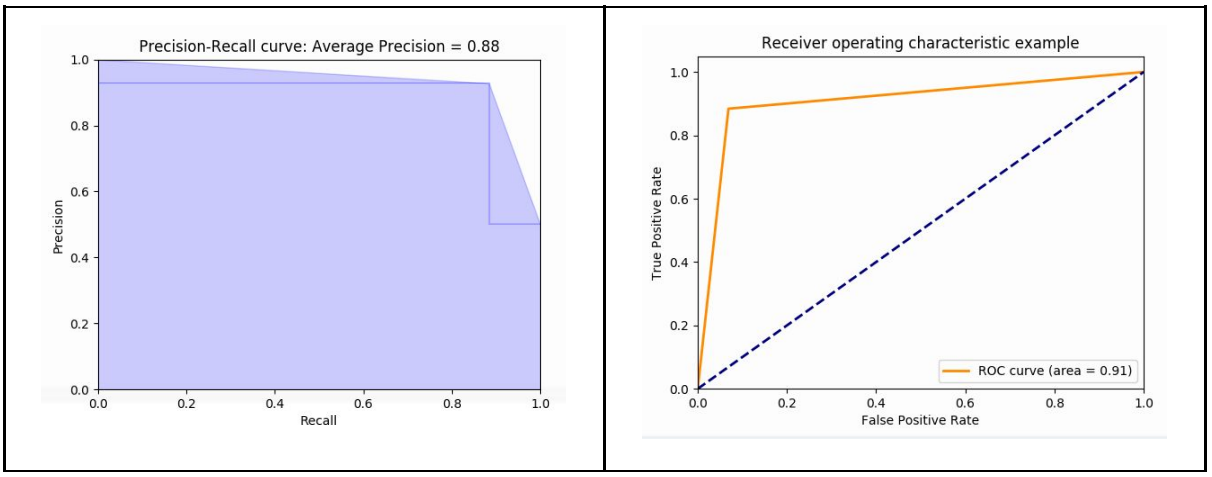
Tab.14 - PRC and ROC for the Logistic Regression.

Linear Support Vector Classifier



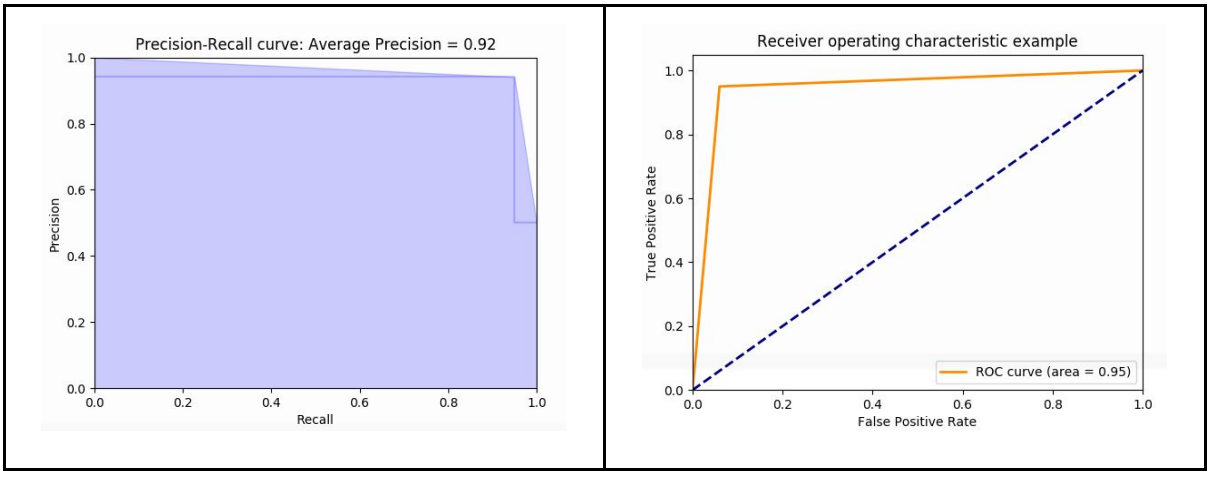
Tab.15 - PRC and ROC for the Linear Support Vector Classifier.

K-Nearest Neighbors



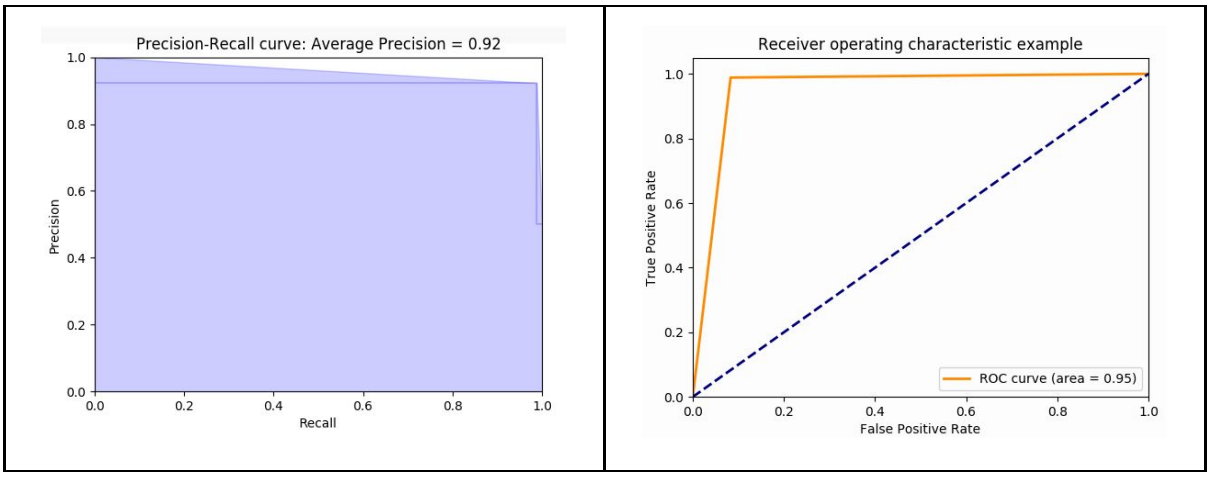
Tab.16 - PRC and ROC for the K-Nearest Neighbors.

Decision Trees



Tab.17 - PRC and ROC for the Decision Trees.

Gradient Boosting Classifier



Tab.18 - PRC and ROC for the Gradient Boosting Classifier.

Regarding the metric results, it is possible to state that every classifier manages to reach good results, however our original purpose has always been catching the most number of fraud transactions and this result is given by the value of the Recall metric.

Among all the classifiers, Gradient Boosting Classifier reach the highest Recall value, which means that this classifier is able to classify most of the fraud behaviour, however it is remarkable the fact that the Decision Trees algorithm is able to achieve the highest Precision value with the highest True Positive value too.

References

[1] E. A. Lopez-Rojas , A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: The 28th European Modeling and Simulation Symposium-EMSS, Larnaca, Cyprus. 2016

[2] Lopez-Rojas, Edgar Alonso. (2016). Applying Simulation to the Problem of Detecting Financial Fraud.

[3] Kotsiantis, Sotiris & Kanellopoulos, D. & Pintelas, P.. (2005). Handling imbalanced datasets: A review. GESTS International Transactions on Computer Science and Engineering. 30. 25-36.