A Predictive Model for the Detection of Clients Suspicious Behavior

Marcelo Leon[1(✉)], Fidel Shagñay[2], Claudia Rivas[1], and Fabricio Echeverria[1]

[1] Universidad Ecotec, Guayaquil, Ecuador
marceloleon11@hotmail.com
[2] Universidad Central del Ecuador, Quito, Ecuador

Abstract. The purpose of this work is to identify the suspicious behavior of the clients of a financial institution. Financial institutions use rule-based systems to detect unusual transactions. These systems focus on individual transactions or simple transaction patterns. Due to this, the need arises to detect suspicious behavior using machine learning since many of the machine learning algorithms are designed to capture complex patterns. Descriptive analysis and predictive analysis are used to detect suspicious behavior. Within the descriptive analysis, outliers are sought in transactional movements. For the predictive analysis, we start from a set of alerts with the label of whether or not they were reported by the Compliance Unit. As a result of the descriptive model, a set of customers that have behaved in an unusual way is obtained and as a result of the predictive model, the alerts that should be reported are predicted. It is concluded that the techniques: descriptive analysis, descriptive analysis and the rule-based system can complement each other, since they focus on different aspects of the identification of unusual transactions and should not be considered as exclusive alternatives.

Keywords: Rules-based systems · Machine learning · Predictive analysis · Transactional movements · Alerts

1 Introduction

Financial institutions use rule-based computer systems to detect unusual or out-of-the-ordinary transactions; these systems use rules to detect unusual transactions. These systems focus on individual transactions or simple transaction patterns and can be breached simply by knowing the thresholds, so they may not be sufficient to detect complicated transaction patterns [1].

The present work is focused on detecting suspicious behavior by applying machine learning algorithms [2]. In addition, rule-based systems focus on individual transactions or simple transaction patterns, so they may not be sufficient to detect complicated transaction patterns.

With the development of high-speed computing, machine learning can analyze large amounts of data quickly. Many machine learning algorithms are designed to capture

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13381, pp. 294–312, 2022.
https://doi.org/10.1007/978-3-031-10548-7_22

complex patterns such as nonlinear relationships between a dependent variable and explanatory variables, where linear rules fall short [3, 4].

A challenge in modelling suspicious behavior is that they are rare events. The rarity of an event refers to the situation where the events of interest are few relative to non-events. These events are infrequent, but their effects can be just as detrimental. Because of their rare occurrence, the information associated with the events tends to be masked by the dominant non-events, which imposes challenges for the model to effectively classify the various signals in the data and accurately separate the two types of events. Because of these problems, models can perform sub optimally in discriminating events from non-events when events are rare [5].

For the construction of the models, the transaction database of a financial institution in Ecuador was used. For the descriptive model, a cluster analysis is applied to find unusual transactional behavior that deviates from normal behavior. For the predictive model, different machine learning algorithms such as random forests, extreme gradient boosting, support vector machine and stacking classifier are applied.

The relevance of this work lies in the use of statistical methods for the detection of suspicious behavior that serve as a tool for the Compliance Unit to locate unusual transactions.

In order to identify suspicious behavior, the transactional movements made by natural customers registered in the financial institution's database are analyzed. The transactions analyzed are those registered in the period February 2019 to September 2019 in the savings account, credit portfolio and fixed-term investment modules. As the financial institution has branches in some provinces of Ecuador, the geographical scope is nationwide.

The financial institution currently uses a rule-based system to detect unusual transactions. The rules are basically conditions with thresholds defined by the Compliance Unit. These systems focus on individual transactions and can be breached simply by knowledge of the thresholds, so they may not be sufficient to detect complicated transaction patterns. Due to these drawbacks, the use of statistical methods for the detection of unusual behavior of the financial institution's customers is proposed. The study does not cover the analysis of credit card transactions and transactional movements of legal clients.

2 Methodology

This paper uses applied research as it developed a descriptive model and a predictive model to identify suspicious behavior of customers of a financial institution.

For the descriptive model, the unsupervised learning technique and a cluster analysis were used to find transactional behavior that deviates from normal behavior or, in other words, to detect unusual behavior [6].

For the predictive model, the supervised learning technique was used in which the alerts that were reported by the compliance unit were labelled [7]. Once the alerts were labelled, several machine learning algorithms were applied and the models with the best performance were used in a meta-classifier. The benefit of the meta classifier is that it can leverage the capabilities of a variety of well-performing models in a classification

task and make predictions that perform better than any individual model. The level of research employed was classificatory and predictive. Classificatory because it seeks to establish unusual customer behavior according to their transactional movements. Predictive because it seeks to define whether or not an alert will be reported based on the characteristics of the alert and the customer.

For the descriptive model, the data available correspond to the transactional movements recorded between February 2019 and September 2019 for the Savings Accounts, Credit Portfolio, Clients and Fixed-Term Investments modules. As the financial institution has branches in some provinces of Ecuador, the geographical scope is nationwide.

For the predictive model, the data available are the records of alerts generated between February 2019 and September 2019 for alerts such as Customers with Multiple Transfers Received, Customers with Frequent Deposits of Fixed Term Deposits, among others. As the alert records are generated from transactional movements, the geographical scope was also nationwide. The study will not cover the movements of the credit card module.

The target population is only the customers who transact in the different modules of the transactional system, which as of February 2019 are around 70,000 and generate around 1,600,000 transactions per month.

Regarding the registration of alerts between February 2019 and September 2019, there are around 6,700 of which 90 are reported by the compliance unit.

The method and approach used was quantitative because it requires statistical analysis, a cluster analysis was performed to identify unusual transnationality and for the predictive analysis different machine learning algorithms were applied to find patterns to determine whether an alert is reported or not.

The temporal dimension of the study is cross-sectional as it took place at a specific time. For the descriptive and predictive model, transactions in a given month are considered.

3 Modeling

This section describes the development of the descriptive and predictive models.

3.1 Data Exploration

We proceed to explore the data used to build the models. The variables considered for the transactional movements of savings accounts and the verification of missing values are presented in Table 1.1.

It is observed that the variables “CodeDocReserved”, “CauseCode”, “Description” have many missing values so they are removed from the data structure.

The variables considered for the transactional credit flows and the missing value check are presented in Table 1.2. It is noted that there are no missing values in the data structure.

The variables considered from the transactional flows of fixed-term investments and the missing value check are illustrated in Table 1.3. It is observed that the variable “CodeDocReserved” has many missing values so it is removed from the data structure.

The variables considered from the transactional alert records and the missing value check are presented in Table 1. It is observed that the variable "Alert" has many missing values and is therefore removed from the data structure.

Table 1. Missing values: (1.1) in savings accounts, (1.2) in appropriations, (1.3) in fixed-term investments, (1.4) in alert records.

Variables (1.1)	Variables (1.2)	Variables (1.3)	Variables (1.4)
CodeTransaction	CodeTransaction	CodeTransaction	Registration Code
CodeAccount	InvestmentCode	InvestmentCode	CodeAlert
Date	TransactionType	TransactionType	Alert (317)
Hour	CashValue	CashValue	Client Code
TransactionType	ValueCheck	ValueCheck	Description
TotalAmout	Date	Date	Date
CashValue	DateHour	DateHour	Time
ValueCheck	CodeDocument	CodeDocument	Total Value
BalanceAccount	CodeDocReserved (1782)	CodeDocReserved (1782)	Office Code
CodeDocument	Posted	Posted	RiskLevel
CodeDocReserved (1369637)	Reversed	Reversed	Condition
ReservedIndicator	CodeOfficeOrigin	CodeOfficeOrigin	CodePhase
CheckNumber	CodigoOficinaDestination	CodigoOficinaDestination	phases
UserCode	UserCode	UserCode	UAF List
CodeOfficeOriigin	RetentionValue	RetentionValue	PepList
CodigoOficinaDestination	ValueCheck	ValueCheck	
PostedIndicator	Channel	Channel	
CauseCode (2045223)			
Description (768202)			
TypeMovement			
Channel			
PaymentMethod			

3.2 Modern Descriptive Model

Descriptive analysis or unsupervised learning aims to find unusual anomalous behavior that deviates from average behavior [8]. This norm can be defined in several ways. It can be defined as the behavior of the average customer at a snapshot in time, or as the average behavior of a given customer over a particular time period, or as a combination of both. When used in the detection of suspicious behavior, unsupervised learning is often referred to as anomaly detection, as it aims to find anomalous and therefore suspicious observations. In the literature, anomalies are commonly described as outliers or exceptions [9, 10].

In the construction of the descriptive model we proceed to create a table of customers who have transacted in a given cut-off, in this case for the month of February 2019. This table is constructed on the basis of the transactional movements of savings accounts and customer tables.

The data structure is analyzed by the variable "CodeProfession", the aim is to examine homogeneous groups as it indicates about peer group analysis [11]. The professions with the highest concentration of clients, it can be observed that the profession with code 28 has 17.649 clients which represents about 25%. And the professions with the lowest concentration of clients, are 58, 42, and 38 with only one client each. These professions are grouped into a single set based on a threshold (the threshold can be defined based on a number of clients e.g. 500 or based on a percentage e.g. 0.5%.) (Table 2).

Table 2. Professions with: (4.1) the highest concentration of clients, (4.2) the lowest concentration of clients.

Index	Code profession	Customer number	Percentage	Index	Code profession	Customer number	Percentage
0	28	17649	25.5095	72	36	3	0.00433614
1	81	8497	12.2814	73	66	2	0.00289076
2	82	8076	11.6729	74	63	2	0.00289076
3	80	4310	6.22958	75	44	2	0.00289076
4	88	3969	5.73671	76	43	2	0.00289076
5	26	3605	5.21059	77	35	2	0.00289076
6	79	3397	4.90995	78	27	2	0.00289076
7	78	2209	3.19284	79	58	1	0.00144538
8	03	1461	2.11170	80	42	1	0.00144538
9	16	1387	2.00474	81	38	1	0.00144538
10	64	1290	1.86454				

The resulting profession codes are: p26, c0, p65, p77, p74, p80, p78, p79, p81, p71, p73, p70, p72, p19, p82, p03, p17, p16, p08, p28, p75, p18, p83, p64, p29, p88, p21. Codes starting with the letter "p" are the codes of professions that are above the threshold and codes starting with the letter "c" are the product of the cluster analysis (professions below the threshold).

The outliers are then analysed for each profession based on customer transactionality ("TransactionValue" = Total Transaction Value in a given month, "TransactionFrequency" = Number of Transactions in a given month) using the One-ClassSVM algorithm. Once the outliers in the transnationality have been identified, we proceed to perform a cluster analysis using the KMeans technique with these records marked as outliers. Prior to the cluster analysis, binary variables are generated for the categorical variables and quantitative variables are scaled by the theme of scales [12, 13].

Records containing the cluster with the maximum transaction value are defined as high risk and medium risk in the opposite case. This analysis is carried out for each profession code and the result is stored in a data structure. After analyzing all the occupational codes, the variable "Label" is defined according to the level of risk (1: if the risk level is high or medium, 0: for low risk level) (Table 3).

Table 3. Initial data structure with the 2 generated variables (risk, label).

Province	Age	Risk	Label
17	69	Low	0
17	83	Low	0
10	82	Low	0
17	76	Low	0
17	73	Low	0
17	75	Low	0
17	70	Low	0
17	66	Medium	1
17	67	Low	0
17	67	Low	0
17	81	Low	0
17	69	Low	0

Once the label has been defined, a probability for flagging a customer with anomalous behavior is calculated using logistic regression. Table 4 illustrates the grouping of the data by the variable "Label". The data are not balanced, the customers with anomalous behavior represent 4.18% and those without anomalous behavior represent 95.82%, so we proceed to sub-sample the majority class to obtain balanced data.

Table 4. Grouping of data by the variable label.

Index	Label	Quantity	Percentage
0	0	66796	95.8171
1	1	2916	4.1829

The subsampling with the sampling strategy parameter equal to 1, means that the number of records of the minority class will be equal to the records of the majority class.

A logistic regression is then applied and the variable "Probability" is generated, which indicates the probability that a customer has an anomalous behavior (Table 5).

Table 5. Data structure with variable Probability.

Risk	Label	Probability
High	1	1
High	1	0.999815
High	1	1
High	1	0.999921
High	1	0.999999
High	1	1
High	1	0.999828
Low	0	0.530792
Low	0	0.433696
Low	0	0.350096

3.3 Model Predictive

In predictive analytics, the goal is to build an analytical model that predicts a target metric of interest. The target metric is typically used to drive the learning process during an optimization procedure. For the construction of the predictive model, the first task is to define the target measure or variable, which is defined on the basis of the status of the alerts [14–18].

The data structure of the predictive model was created joining the tables AlertRegister, Customer and CustomerNatural. Then, the target variable is defined on the basis of the alert statuses. Table 6 shows the different statuses of the alert records. Based on its analysis, the compliance officer determines whether the alert is reported, which is why the target variable is defined as one if the alert is reported and zero if it is not (status other than reported).

Table 6. Registration status alerts.

Code	State
1	Entered
2	Official analysis
3	Analisys agency
4	Reported
5	Not reported
6	Reopened
7	To discharge
8	Expired

Figure 1 illustrates the percentages representing the statuses of the alert records. The status "Reported" represents 1.2% of the total number of records.

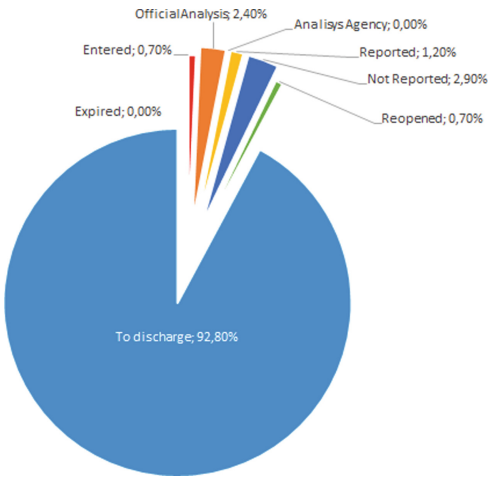

Fig. 1. Percentage of alert registration statuses.

Alerts with status "Entered", "Official Analysis", "Agency Analysis", "Reopened", "Expired" are deleted as these alert records have not been analyzed. The creation of the variable "Reported" was made with a value of 1 if it is reported and 0 otherwise.

The number of records with a value of one in the variable "Reported" is 81, representing 1.2% of the total. As the data are unbalanced, the techniques of under sampling and oversampling are used. The training and test partitions are defined. For this case 70% (4512) of records will be destined for model training and 30% (1934) for model testing.

To imbalance in the training partition (0:4457, 1:55), first the minority class is oversampled (with the sampling strategy parameter equal to 10%, the minority class will be 10% of the majority class), then the majority class is under sampled (with the sampling strategy parameter equal to 100% the minority class will be 100% of the majority class).

This combination of oversampling and under sampling is used because using each technique individually, the model does not perform well and when combining the two techniques the performance of the model improves.

Several models were built with different algorithms, obtaining better performance with the Random Forest, Extreme Gradient Boosting and Support Vector Machine algorithms [19–22]. The confusion matrix metrics of the three models are presented in Table 7. As the data are unbalanced, the metrics considered to evaluate the performance are: accuracy, sensitivity and F1 score.

Once the optimal parameters of the three models have been found, the Stacking Classifier method is used to combine the three classification models and thus utilize the strength of each individual estimator using its output as input to a final estimator. Figure 2 illustrates the combination of the three models.

Table 7. Metrics of the RF, XGB and SVM models.

Model	Accuracy	Sensitivity	F1 score
Random forests	0.47	0.35	0.40
Extreme gradient boosting	0.38	0.67	0.48
Support vector machine	0.33	0.57	0.42

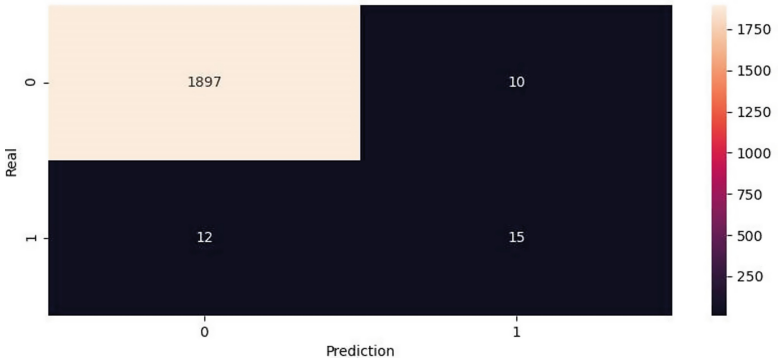

Fig. 2. Confusion matrix of the final model.

The metrics with class 0 (alert was not reported) are good with a sensitivity accuracy and f1 score of 99% one could conclude that it is a very good model, but the metrics for class 1 (alert was reported) should be reviewed because the data is unbalanced and the model was designed to detect alerts that will be reported (Fig. 3).

```
In [89]: print(classification_report(y_true=y_real,y_pred=y_pred))
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1907
           1       0.60      0.56      0.58        27

    accuracy                           0.99      1934
   macro avg       0.80      0.78      0.79      1934
weighted avg       0.99      0.99      0.99      1934
```

Fig. 3. Final model classification report.

Be positive when the alert is reported and negative when the alert is not reported. In the following, the accuracy, sensitivity and f1 score can be interpreted as:

– Accuracy (with a value of 60%) indicates how many of the cases that the model predicts as positive are actually positive. The accuracy takes into account false positives (type I error) i.e. when the model predicts that the alert is reported when it is not actually reported.

– Sensitivity (with a value of 56%) indicates how many of the cases that are actually positive the model can capture or predict as positive. Sensitivity takes into account false negatives (type II error), i.e. when the model predicts that the alert is not reported when it is actually reported.
– The f1 score which can be interpreted as a weighted average of accuracy and sensitivity has a value of 58% is improved by the f1 scores obtained from the individual models (40%, 48% and 42%).

As false negatives (12) are higher than false positives (10) the accuracy is higher than the sensitivity. Table 8 shows the calculation of the metrics of the confusion matrix of the final model.

Table 8. Metrics of the confusion matrix of the final model.

Metric	Calculation	Value
Accuracy	(VN + VP)/(VN + VP + FN + FP) = (VN + VP + FN + FP)/(VN + VP + FN + FP) = (1897 + 15)/(1897 + 15 + 12 + 10)	98.86%
Accuracy	VP/(VP + FP) = 15/(15 + 10)	60.00%
Sensitivity	VP/(VP + FN) = 15/(15 + 12)	55.55%
F1 score	2 * (Accuracy * Sensitivity)/(Accuracy + Sensitivity)	57.69%
Error rate	(FN + FP)/(VN + VP + FN + FP) = (FN + FP)/ (VN + VP + FN + FP) = (12 + 10)/(1897 + 15 + 12 + 10)	1.14%
Specificity	VN/(VN + FP) = 1897/(1897 + 10)	99.47%

4 Discussion of Results

This section presents the results obtained from the descriptive and predictive model. For the presentation of the results, an analytical web application was created in Python with Dash and the application was published on the IIS (Internet Information Services) web server [18, 23–25].

Dash is an open source framework for creating data visualization interfaces. Released in 2017 as a Python library, it has grown to include implementations for R and Julia. Dash helps data scientists create analytical web applications without requiring advanced web development skills [23].

4.1 Descriptive Model

The descriptive model was run for the months of February 2019 to September 2019, where the anomalies for the different months were obtained. Figure 4 shows the anomalies detected for the month of February 2019.

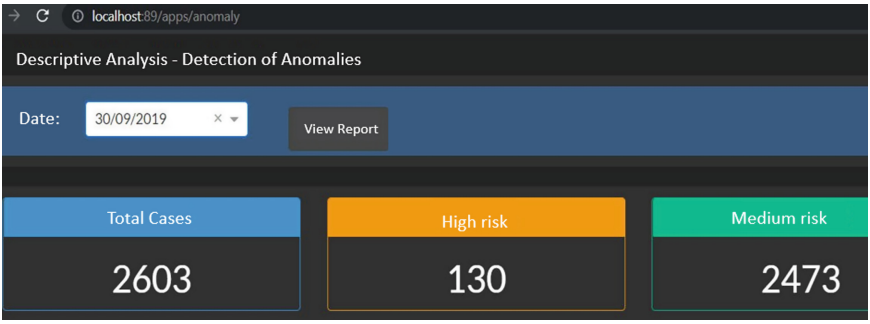

Fig. 4. Final model classification report (web application - Python dash).

Anomalies are categorized with a medium and high risk level. Giving the compliance officer an additional tool to detect unusual transactions. These anomalies do not necessarily represent observations linked to money laundering; these observations require extensive follow-up and validation by the Compliance Officer. Table 9 shows the results of the model for the different months.

Table 9. Anomalies detected (descriptive model - results).

Date	Total cases (anomalies)	High risk	Medium risk
September 2019	2603	130	2473
August 2019	2663	146	2517
July 2019	2686	150	2536
June 2019	2474	189	2285
May 2019	2506	277	2229
April 2019	2705	205	
March 2019	2742	218	2200
February 2019	2630		2433

Since the transaction variable province is available, this variable is used to present the anomalies on a map of the provinces of Ecuador in GeoJSON format. Geographic JSON (or GeoJSON) is an open format for encoding simple geospatial datasets using the JSON (JavaScript Object Notation) standard. Figure 5 illustrates the anomalies by the different provinces of Ecuador, as the financial institution has a greater presence in the provinces of Pichincha and Guayas, where the greatest volume of transactions and therefore the greatest number of anomalies are concentrated.

For September 2019 there are 2603 anomalies of which 1150 correspond to the province of Pichincha.

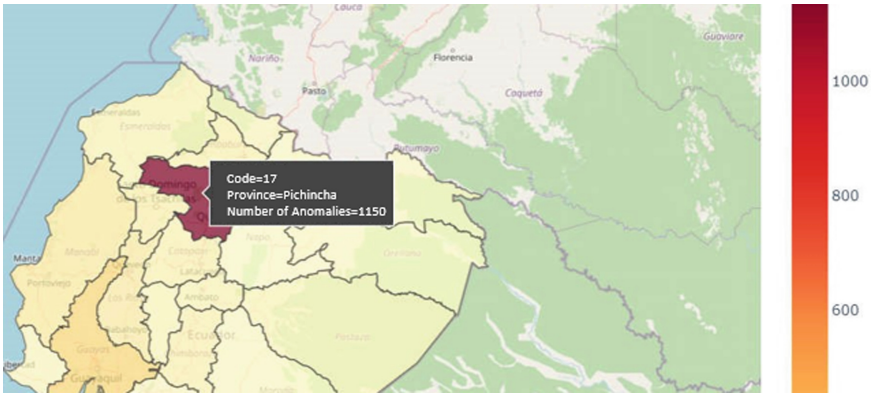

Fig. 5. Anomalies by province.

Figure 6 shows the volume of transactions in millions of dollars and the number of anomalies for the months of February 2019 to September 2019.

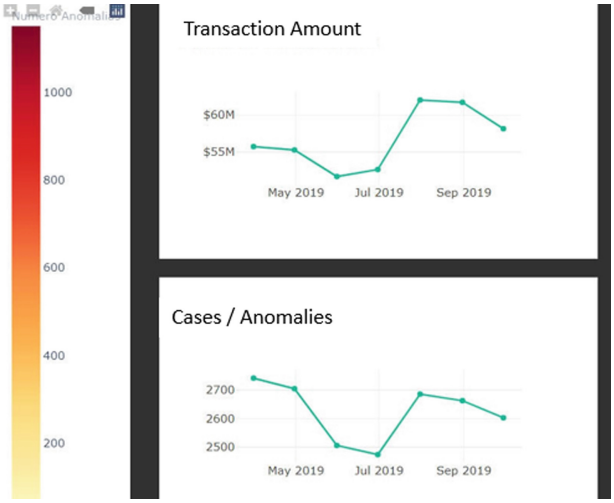

Fig. 6. Transactions and anomalies (Python dash elaboration).

4.2 Predictive Model

The classification metrics of the 3 models generated (Random Forests, Extreme Gradient Boosting, Support Vector Machine) and the final model (Stacking Classifier) are presented below.

Because the data are unbalanced the metrics considered to evaluate the models are: accuracy, sensitivity and f1 score.

The confusion matrix obtained from the model generated with the RF algorithm is illustrated in Fig. 50. It can be seen that there are more false negatives (15) than false positives (9). Table 10 illustrates the metrics of the model generated by the RF algorithm (Fig. 7).

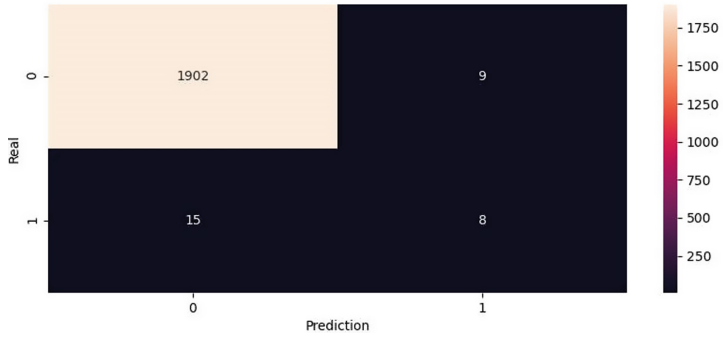

Fig. 7. RF confusion matrix.

Table 10. RF confusion matrix metrics.

Metric	Calculation	Value
Accuracy	(VN + VP)/(VN + VP + FN + FP) = (VN + VP + FN + FP)/(VN + VP + FN + FP) = (1902 + 8)/(1902 + 8 + 15 + 9)	98.76%
Accuracy	VP/(VP + FP) = 8/(8 + 9)	47.05%
Sensitivity	VP/(VP + FN) = 8/(8 + 15)	34.78%
F1 Score	2 * (Accuracy * Sensitivity)/(Accuracy + Sensitivity)	40.00%
Error rate	(FN + FP)/(VN + VP + FN + FP) = (FN + FP)/ (VN + VP + FN + FP) = (15 + 9)/(1902 + 8 + 15 + 9)	1.24%
Specificity	VN/(VN + FP) = 1902/(1902 + 8)	99.58%

Because there are more false negatives than false positives the sensitivity is lower than the accuracy. The sensitivity with a value of 34.78 is not very good and indicates that of the cases that are actually positive how many cases the model can capture or predict as positive. The f1 score can be interpreted as a weighted average of the precision and the sensitivity represents a value of 40.00.

The confusion matrix obtained from the model generated with the XGB algorithm is illustrated in Fig. 8. It can be seen that there are more false positives (33) than false negatives (10). Table 11 illustrates the metrics of the model generated by the XGB.

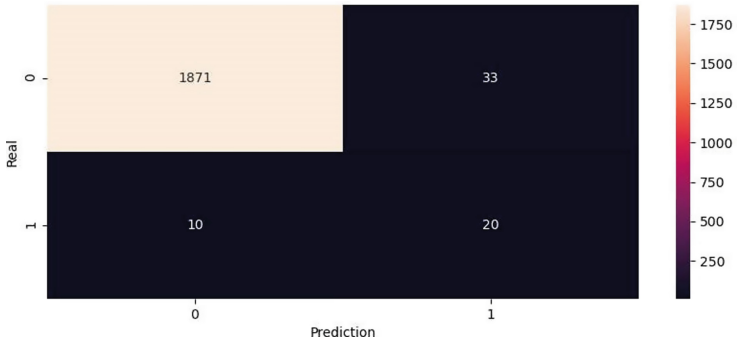

Fig. 8. XGB confusion matrix.

Table 11. XGB confusion matrix metrics.

Metric	Calculation	Value
Accuracy	(VN + VP)/(VN + VP + FN + FP) = (VN + VP + FN + FP)/(VN + VP + FN + FP) = (1871 + 70)/(1871 + 70 + 10 + 33)	97,83%
Accuracy	VP/(VP + FP) = 20/(20 + 33)	37,73%
Sensitivity	VP/(VP + FN) = 20/(20 + 10)	66,66%
F1 Score	2 * (Accuracy * Sensitivity)/(Accuracy + Sensitivity)	48,19%
Specificity	VN/(VN + FP) = 1871/(1871 + 33)	98,26%

Because there are more false positives than false negatives the precision is lower than the sensitivity. The precision with a value of 37.73 is not very good and indicates that of the cases that the model predicts as positive how many are actually positive. The f1 score which can be interpreted as a weighted average of the precision and sensitivity represents a value of 48,19.

The confusion matrix obtained from the model generated with the support vector machine algorithm is illustrated in Fig. 9. It can be seen that there are more false positives (34) than false negatives (13).

Table 12 illustrates the metrics of the model generated by the Support Vector Machine algorithm.

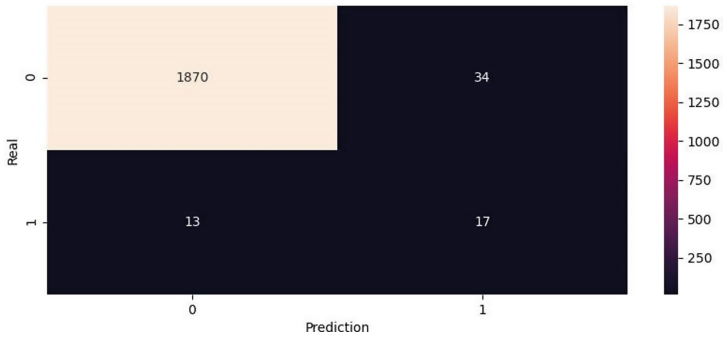

Fig. 9. SVM confusion matrix.

Table 12. SVM confusion matrix metrics.

Metric	Calculation	Value
Accuracy	(VN + VP)/(VN + VP + FN + FP) = (VN + VP + FN + FP)/(VN + VP + FN + FP) = (1870 + 17)/(1870 + 17 + 13 + 34)	97,57%
Accuracy	VP/(VP + FP) = 17/(17 + 34)	33,33%
Sensitivity	VP/(VP + FN) = 17/(17 + 13)	56,66%
F1 Score	2 * (Accuracy * Sensitivity)/(Accuracy + Sensitivity)	41,98%
Error rate	FN + FP)/(VN + VP + FN + FP) = (VN + VP + FN + FP) = (FN + FP)/(VN + VP + FN + FP) = (13 + 34)/(1870 + 17 + 13 + 34)	2,43%
Specificity	VN/(VN + FP) = 1870/(1870 + 34)	98,21%

Because there are more false positives than false negatives the precision is lower than the sensitivity. The precision with a value of 33.73 is not as good and indicates that of the cases that the model predicts as positive how many are actually positive. The f1 score which can be interpreted as a weighted average of the precision and sensitivity represents a value of 41.98.

The confusion matrix obtained from the model generated with the stacking classifier method is presented in Fig. 56. It can be seen that the difference between false positives (10) and false negatives (12) is not very marked (Fig. 10).

Table 13 illustrates the metrics of the model generated with the stacking classifier method.

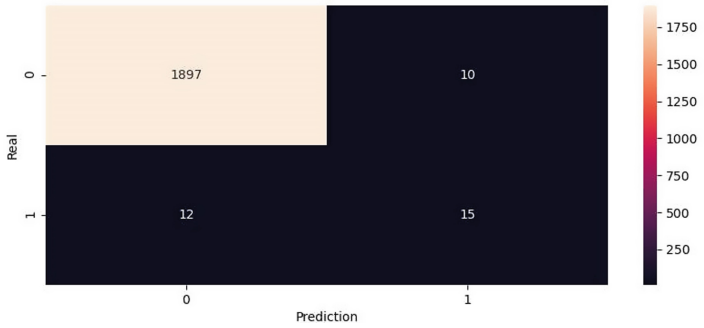

Fig. 10. SC confusion matrix.

Table 13. SC confusion matrix metrics.

Metric	Calculation	Value
Accuracy	(VN + VP)/(VN + VP + FN + FP) = (VN + VP + FN + FP)/(VN + VP + FN + FP) = (1897 + 15)/(1897 + 15 + 12 + 10)	98,86%
Accuracy	VP/(VP + FP) = 15/(15 + 10)	60,00%
Sensitivity	VP/(VP + FN) = 15/(15 + 12)	55,55%
F1 Score	2 * (Accuracy * Sensitivity)/(Accuracy + Sensitivity)	57,69%
Error rate	(FN + FP)/(VN + VP + FN + FP) = (FN + FP)/ (VN + VP + FN + FP) = (12 + 10)/(1897 + 15 + 12 + 10)	1,14%
Specificity	VN/(VN + FP) = 1897/(1897 + 10)	99,47%

Because the difference between false positives and false negatives is not very marked, the accuracy is somewhat equated with the sensitivity. The f1 score which can be interpreted as a weighted average of accuracy and sensitivity represents a value of 57.69 which is higher than the individual models. As the data are not balanced, the precision-sensitivity curve is used instead of the ROC curve as the ROC curve may be optimistic under severe class imbalance, especially when the number of examples in the minority class is small.

To easily measure the success of a classification model based on AUC, Gorunescu [24] presents different ranges for interpreting the AUC of the accuracy-sensitivity curve (Table 14).

Figure 11 illustrates the accuracy - sensitivity curve of the final model.

Table 14. AUC ranges (adaptation [24]).

Interpretation	Balanced	Unbalanced
Random	0.50	0.09
Malo	0.74	0.23
Good	0.84	0.51
Excellent	0.98	0.90
Perfect	1.00	1.00

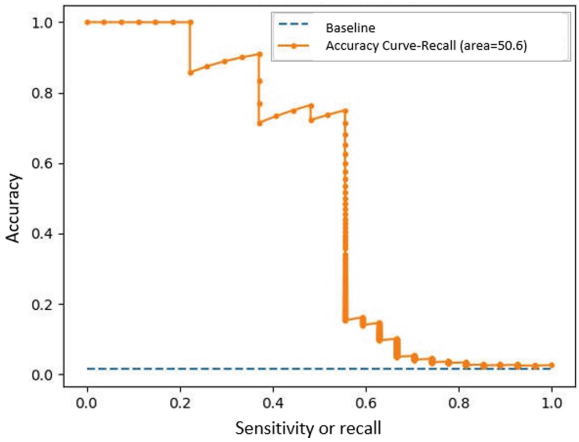

Fig. 11. Accuracy-sensitivity curve of the final model.

According to the AUC rank table and with an AUC of 50.6 it can be said that the final model using the stacking classifier method is relatively good.

5 Conclusions

The descriptive model that was built aims to find unusual anomalous customer behavior that deviates from normal behavior or, in other words, to detect anomalies, so customer transnationality was analyzed for each profession to find anomalies. This resulted in records marked as anomalies with a medium or high level of risk. The anomalies detected do not necessarily represent observations linked to money laundering. Therefore, the use of unsupervised learning for the detection of unusual transactions requires extensive follow-up and validation of the suspicious observations identified.

For the development of the predictive model, the main drawback is that the data are unbalanced, i.e. the events of interest are few in relation to the other events, when applying different algorithms to obtain the model, the performance of the models generated is sub-optimal, so the techniques of under sampling (eliminates or combines records of the majority class) and oversampling (creates new synthetic records of the minority class)

are used, and oversampling (creating new synthetic records from the minority class). Using the two techniques in combination yields more effective results than using the two techniques in isolation, so we first oversampled and then under sampled the training partition.

Several models were built with different algorithms, obtaining acceptable performance with the RandomForest, Extreme Gradient Boosting and Support Vector Machine algorithms. To improve the confusion matrix metrics of these three models, the Stacking Classifier method was applied to create a final model and the f1 score of the individual models was improved. It can be concluded that multiple algorithms can capture different parts of the input data space and as such complement each other's deficiencies.

Based on the results obtained, the hypothesis is somewhat rejected as the descriptive, predictive and rule-based system analysis techniques can complement each other as they focus on different aspects of the identification of unusual transactions and should not be considered as exclusive alternatives. An effective suspicious behavior detection system will make use of and combine these different tools, which have different capabilities and limitations, and will therefore reinforce each other when applied in a combined configuration.

References

1. Adewumi, A.O., Akinyelu, A.A.: A survey of machine-learning and nature-inspired based credit card fraud detection techniques. Int. J. Syst. Assur. Eng. Manag. **8**(2), 937–953 (2017). https://doi.org/10.1007/s13198-016-0551-y
2. Sharif, O., Hoque, M.M., Kayes, A.S.M., Nowrozy, R., Sarker, I.H.: Detecting suspicious texts using machine learning techniques. Appl. Sci. **10**(18), 1–23 (2020)
3. Clarke, B., Fokoue, E., Zhang, H.H.: Principles and Theory for Data Mining and Machine Learning. Springer, New York (2009). https://doi.org/10.1007/978-0-387-98135-2
4. Flach, P.: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press, Cambridge (2012)
5. Ross, I.: Exposing Fraud: Skills, Process and Practicalities. Wiley, Hoboken (2016)
6. Danenas, P.: Intelligent financial fraud detection and analysis: a survey of recent patents. Recent Patents Comput. Sci. **8**(1), 13–23 (2015)
7. More, R., Awati, C., Shirgave, S., Deshmukh, R., Patil, S.: Credit card fraud detection using supervised learning approach. Int. J. Sci. Technol. Res. **9**, 216–219 (2021)
8. Bolton, R.J., Hand, D.J.: Statistical fraud detection: a review. Stat. Sci. **17**(3), 235–255 (2020)
9. Young, M.R.: Financial Fraud Prevention and Detection: Governance and Effective Practices. Wiley, Hoboken (2014)
10. Spann, D.D.: Fraud Analytics: Strategies and Methods for Detection and Prevention. Wiley, Hoboken (2013)
11. Baesens, B., Vlasselaer, V.V., Verbeke, W.: Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection. Wiley, Hoboken (2015)
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2005)
13. Swamynathan, M.: Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python. Apress, New York (2017)
14. Shmueli, G., Bruce, P.C., Gedeck, P., Patel, N.R.: Data Mining for Business Analytics: Concepts, Techniques and Applications in Python. Wiley, Hoboken (2020)

15. Vona, L.W.: Fraud Data Analytics Methodology: The Fraud Scenario Approach to Uncovering Fraud in Core Business Systems. Wiley, Hoboken (2017)
16. Géron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Sebastopol (2019)
17. Bowles, M.: Machine Learning in Python: Essential Techniques for Predictive Analysis. Wiley, Hoboken (2015)
18. Julian, D.: Designing Machine Learning Systems with Python: Design Efficient Machine Learning Systems that Give You More Accurate Results. Packt, Birmingham (2016)
19. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324
20. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In: Zhang, C., Ma, Y. (eds.) Ensemble Machine Learning. Springer, Boston (2012). https://doi.org/10.1007/978-1-4419-9326-7_5
21. Carmona, P., Climent, F., Momparler, A.: Predicting failure in the US banking sector: an extreme gradient boosting approach. Int. Rev. Econ. Finance **61**, 304–323 (2019)
22. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A.: A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing **408**, 189–215 (2020)
23. Dabbas, E.: Interactive Dashboards and Data Apps with Plotly and Dash: Harness the Power of a Fully Fledged Frontend Web Framework in Python - no JavaScript Required. Packt, Birmingham (2021)
24. Döble, M., Großmann, T.: Data Visualization with Python: Create an Impact with Meaningful Data Insights Using Interactive and Engaging Visuals. Packt, Birmingham (2019)
25. Layton, R.: Learning Data Mining with Python: Harness the Power of Python to Analyze Data and Create Insightful Predictive Models. Packt, Birmingham (2015)
26. Belorkar, A., Guntuku, S.C., Hora, S., Kumar, A.: Interactive Data Visualization with Python Second Edition: Present Your Data as an Effective and Compelling Story. Packt, Birmingham (2020)
27. Gorunescu, F.: Data Mining: Concepts, Models and Techniques. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19721-5