PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

NÚCLEO DE EDUCAÇÃO A DISTÂNCIA

***Lato Sensu* Postgraduate Degree in Artificial Intelligence and Machine Learning**

Rômulo Ponciano da Silva Freitas

EXPLICABILITY AS SUPPORT FOR FRAUD DETECTION IN FINANCIAL TRANSACTIONS THROUGH ARTIFICIAL INTELLIGENCE

Rio de Janeiro, Brazil
April 2022

Rômulo Ponciano da Silva Freitas

EXPLICABILITY AS SUPPORT FOR FRAUD DETECTION IN FINANCIAL TRANSACTIONS THROUGH ARTIFICIAL INTELLIGENCE

Course Completion Work presented to the Specialization Course in Artificial Intelligence and Machine Learning, as a partial requirement for obtaining the title of Specialist

.

Rio de Janeiro, Brazil
April 2022

SUMMARY

1. Introduction 4

2. Context 5

3. Problem Description and Proposed Solution 6

4. Data Collection 7

4.1. Data Feature 7

4.2. Data Attributes 8

4.2.1 step 9

4.2.2 type 9

4.2.3 finance attributes 10

4.2.4 class attribute 11

5. Data Processing/Treatment 11

5.1. Programming language and libraries 11

5.2. Problems in the dataset 12

5.2.1 Removal of non-significant records 12

5.2.2 Discretization of attributes 12

5.3.3 Attribute correlation 13

5.3.4 Attribute normalization 14

5.3.5 Class balancing 15

6. Experiments and Analysis with Machine Learning 15

6.1 Model evaluation 15

6.2 Dataset split 16

6.3 Study on dimensionality reduction 16

6.4 Decision Tree 17

6.5 k-Nearest Neighbors 18

6.6 Multi-Layer Perceptron 19

7. Results discussion 20

8. Conclusion and Future Work 22

8.1 Conclusion 22

8.2 Future Work 23

9. Links 24

10. References 24

1. Introduction

Since the beginning of banks, there have been people trying to steal customer information to withdraw money from accounts. In the beginning, identity theft to impersonate another person using personal information and forging signatures was common. However, with the advancement of technology, it has become increasingly common to use methods to break into accounts through data leaks and access to obvious passwords [6, 8].

Currently, among the different types of fraud involving financial institutions, such as identity theft and credit card cloning, there is one related to access to a person's bank account. When the criminal gains access to the account, he withdraws the victim's money via in-person withdrawals or online transactions [1].

The growing increase in the use of cell phones and banking systems to carry out transactions between accounts has caused an increase in the number of account thefts and data leaks from customers of financial institutions around the world [2]. In Brazil, with the arrival of the Pix transaction system, there was a 39% increase in the number of flash kidnappings in São Paulo alone [3]. These problems even forced the Central Bank to adopt measures together with the country's financial institutions [4].

To solve these problems, the Machine Learning area evolved and carried out several studies and projects in large banks with the aim of training predictive models and using them in a production environment. However, this strategy faces several problems. Among them, one of the best known is the use of extremely unbalanced datasets [5, 7].

In line with this, the uncertainty present in predictive models creates the need to use predictive models as decision support tools where the final decision is up to someone. However, many predictive models create ways to make their predictions through black box solutions [9, 10]. In other words, solutions that we cannot easily infer.

Therefore, this project seeks to evaluate and compare the use of black box predictive models with those that have greater interpretation capacity in their results. By using easily explainable models, we can even receive feedback from users of these tools that help us improve the model, whether by reclassifying records for a future update or discarding rules that can lead the model to biased predictions [9].

2. Context

Nowadays, in addition to preventive security measures and user awareness, financial institutions try to identify this type of fraud through the training of classificatory predictive models trained through machine learning algorithms, as everyone is susceptible to a kidnapping like these. However, as there is a much greater number of valid financial transactions than those identified as fraud, the data collected ends up divided in a completely unbalanced way [5, 6].

The problem of such unbalanced financial data is, in fact, one of the factors that most negatively affects the training and use of models trained for this purpose [7]. Furthermore, these models end up being used as decision support tools because they are known to be imperfect and lead to erroneous predictions [8].

For this reason, it is very important that the predictions made by the models can be interpreted in some way by those who will use the support, even if it is a prediction directly sent to the customer, for example, “we identified a suspicious transaction in your account. Do action x to confirm or y to block.”

This need for interpretation in model prediction has generated a growing interest in a subarea called Explainability in Artificial Intelligence (XAI) [9]. With studies in this area, it was possible to separate models between those called explainable and black box models. The first concerns models that provide an interpretation of their predictions, whether through association rules, probability or even logical sequence of data. The second concerns models where we cannot easily infer the reasons behind their predictions, such as deep learning models [10, 11].

3. Problem Description and Proposed Solution

Currently, many studies and practices focus on fraud detection in financial transactions and other studies focus on the explainability of predictive models trained through machine learning algorithms [12, 13]. However, as described previously, it is important to have the ability to interpret predictions in this case. Not only to improve the model's own learning in a future update with more accurate data, but also to increase the understanding of the problem for the user who will come across the result of the prediction.

Therefore, the objective of this course completion work is to demonstrate the difficulties of working with unbalanced data, training several models for the purpose of predicting fraud in financial transactions and comparing the results between black box models and those with greater ease of explainability of its predictions. In this way, the work follows the flow shown in Figure 1.

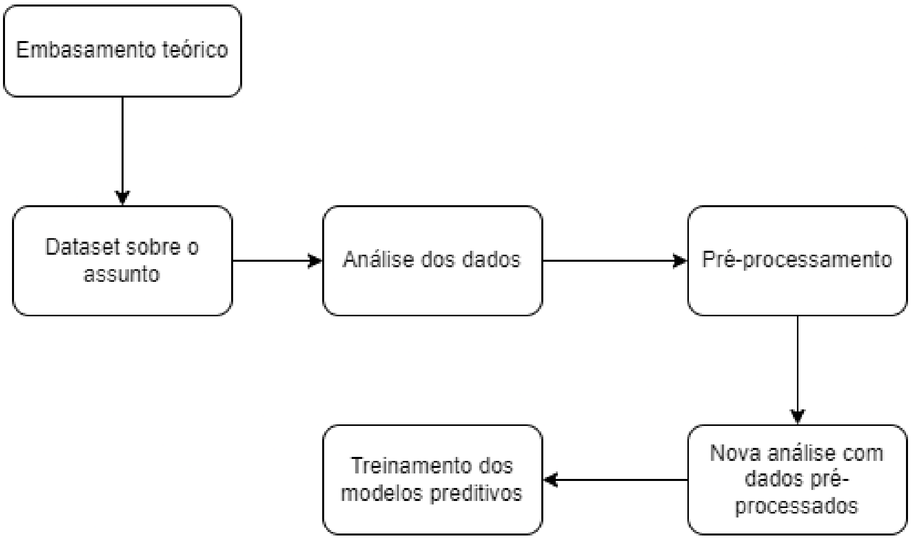


Figura 1. Fluxo do projeto

In Figure 1, it can be seen that the first step after providing a theoretical basis for the problem was the search for a database that represented financial transactions containing valid and fraudulent transactions. With the database in hand, it was necessary to analyze this data, which, in turn, revealed the necessary pre-processing. After pre-processing, there was a new analysis of the processed data and, finally, the training of the predictive models and a final analysis of the results obtained were carried out.

4. Data Collection

For the purpose of this work, it is important that the dataset is unbalanced and is in the context of the problem: fraud in financial transactions. In this aspect, the aim of the work is not to generalize the result to the whole world or a specific country, but rather to present the results of analysis and training on an unbalanced dataset and with the aim of comparing different machine learning algorithms in relation to its quality and explainability.

For this reason, the database was searched and found on the Kaggle website[9.1]. This site is one of the most famous data repositories for the availability and challenges related to Big Data involving Data Science and Machine Learning. The dataset was made available by collaborator Vardhan Siramdasu under a publicly open data license and accessed on 02/25/2022 through Kaggle[9.2].

4.1. Data Features

The database has more than 6 million records (6362620) and 10 columns. Each record represents a transaction in a period of time where the lines are ordered chronologically within a period of 30 days. The columns are arranged on the base according to the following characteristics:

step (Integer): represents the chronological time at which the transaction occurred within the 30-day period. For example, step 1 means it occurred in the first hour of the 30 days; step 73 means it occurred in the 73rd hour (1st hour of the third day). In this dataset, the same hour can contain 0 or n transactions.

type (String): represents the type of that transaction. The possible types are: CASH-IN, CASH-OUT, DEBIT, PAYMENT or TRANSFER.

amount (Double): financial amount of the transaction, regardless of the type

nameOrig (String): code of the originating client from which the transaction originates

oldbalanceOrg (Double): financial amount in the originating account before the transaction.

newbalanceOrig (Double): new financial amount in the originating account after the transaction is carried out

nameDest (String): destination customer code where the transaction is intended

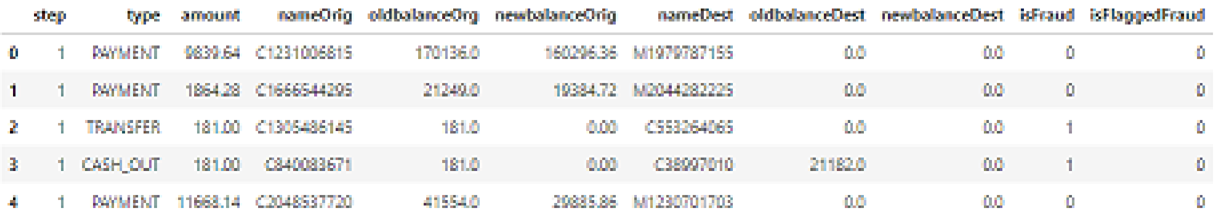
oldbalanceDest (Double): financial amount in the target account before the transaction

newbalanceDest (Double): financial amount in the destination account after the transaction is carried out

isFraud (Boolean): target class of the dataset. Value 1 means that the transaction is a fraud, while value 0 means it is not a fraud.

isFlaggedFraud (Boolean): attribute created to simulate an algorithm that marks a transaction as possible fraud if, and only if, the transaction is for a value greater than 200K.

Figure 2 demonstrates the first 5 lines present in the dataset.



	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Figura 2. Cinco primeiras linhas presentes no dataset

Furthermore, the dataset has a rule about its limitation and already specified in the problem: customers who have a code starting with M are customers who do not have financial information about the target account for that transaction (oldbalanceDest and newbalanceDest are always zero)

4.2 Attributes distribution

This section presents the distribution and statistical characteristics of each of the attributes present in the dataset.

4.2.1 step

We can see from Figure 3 that the attribute distribution is not a normal distribution. With an average of 243.4 and a standard deviation of 142.33, there is a large number of transactions concentrated between the 1st and 8th hour. After a drop in the number of transactions, there are new peaks between the 110th and 410th hour.

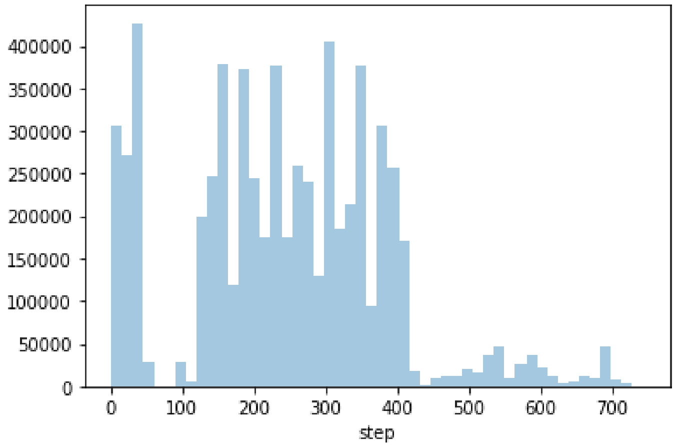


Figura 3. Distribuição do atributo step

Step values vary between 1 and 743, where 25% of these values are up to value 156, 50% up to value 239 and 75% up to value 335. Reinforcing, once again, that most transactions took place before from hour 335.

4.2.2 type

We can see from Figure 4 that there are many more PAYMENT and CASH_OUT transactions than TRANSFER and DEBIT transactions. The CASH_IN type has an average amount around the largest.

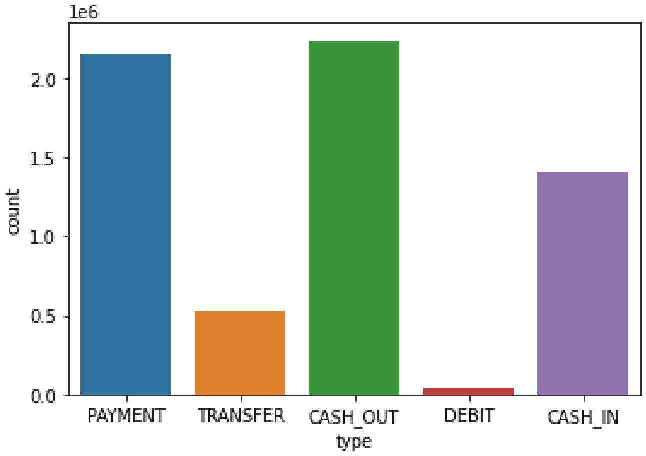
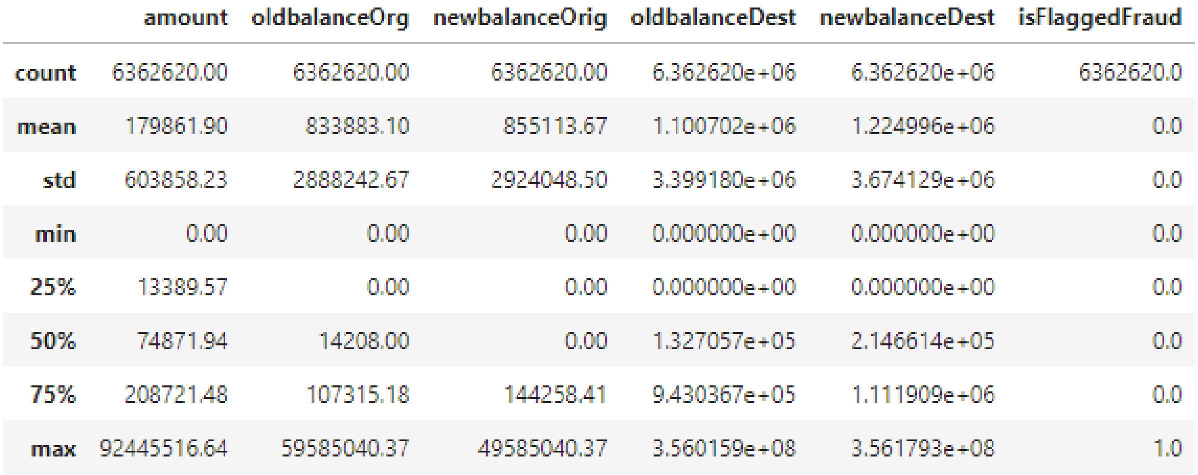


Figura 4. Quantidade de cada tipo de transação no dataset

The absolute values of each type are: 2237500 CASH_OUT, 2151495 PAYMENT, 1399284 CASH_IN, 532909 TRANSFER and 41432 DEBIT.

4.2.3 financial attributes

The attributes referring to financial values were summarized in this section. It can be seen in Figure 5 that the values represent very varied and spaced numbers, from zero to tens of millions.



	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFlaggedFraud
count	6362620.00	6362620.00	6362620.00	6.362620e+06	6.362620e+06	6362620.0
mean	179861.90	833883.10	855113.67	1.100702e+06	1.224996e+06	0.0
std	603858.23	2888242.67	2924048.50	3.399180e+06	3.674129e+06	0.0
min	0.00	0.00	0.00	0.000000e+00	0.000000e+00	0.0
25%	13389.57	0.00	0.00	0.000000e+00	0.000000e+00	0.0
50%	74871.94	14208.00	0.00	1.327057e+05	2.146614e+05	0.0
75%	208721.48	107315.18	144258.41	9.430367e+05	1.111909e+06	0.0
max	92445516.64	59585040.37	49585040.37	3.560159e+08	3.561793e+08	1.0

Figura 5. Tabela de distribuição dos atributos financeiros

It is also important to note the last column of Figure 5. This column represents the metrics of the attribute marked by the automatic algorithm that marked a transaction with a turnover greater than or equal to 200 thousand as possible fraud.

4.2.4 class attribute

With an absolute number of 6354407 valid transactions and only 8213, it is easy to see how unbalanced this dataset is. Only 0.129% of transactions are frauds.

5. Data Processing/Treatment

This section describes how the data was processed and treated, and which tools were used for this process.

5.1 Programming language and libraries

For this project, the Python programming language [9.4] was adopted. This is a language widely adopted by the community when it comes to Data Science or Machine Learning [14]. Due to this wide adoption, the language has become very robust for this problem in conjunction with a variety of libraries to support development.

All processing was done using a Jupyter Notebook. This framework allows the use of the Python language in conjunction with libraries that can be imported and used between text markups. The reason for choosing Jupyter Notebook is to facilitate the explanation and reasoning behind each step from initial data analysis, through processing, to model training.

The libraries used to execute this project were: numpy (array and matrix processing), pandas (efficient data processing, dataframe imports and transformations), matplotlib.pyplot (chart insertion), sklearn.metrics (calculation and display of metrics for machine learning algorithms), sklearn.model_selection.train_test_split (separation of data into training and test sets), sklearn.metrics.confusion_matrix (calculation and display of confusion matrix for analyzing predictive models), sklearn. metrics.classification_report (calculation and display of reports for evaluating predictive models), seaborn (insertion of graphs), joblib (saving and loading predictive models), sklearn.manifold.TSNE (application of the TSNE algorithm), matplotlib.patches.mpatches (insertion of scatter plots), sklearn.tree (application of the decision tree algorithm), sklearn.tree.export_graphviz (export of decision tree assembled from a model), six.StringIO (image export), IPython.display .Image (image display), pydotplus (image export and display), sklearn.neighbors.KNeighborsClassifier (k-NN algorithm application), sklearn.neural_network.MLPClassifier (MLP algorithm application), sklearn.svm (application of SVM algorithm), sklearn.model_selection.GridSearchCV (search for optimal parameters for training predictive models).

5.2 Dataset problems

The data analysis, summarized in section 4.1, demonstrated that the dataset has two major problems: (1) completely unbalanced target class and (2) financial data with a lot of variety and spaced out. In addition to these problems, we can also list some that are common in datasets, such as, for example, (3) the presence of null values or records without meaningful values (in the case of clients that start with M).

5.2.1 Removal of non-significant records

As it is a dataset with millions of rows, problematic data for the algorithms can be removed without major consequences. However, in this context it is necessary to be careful to avoid removing records marked as fraud as much as possible so as not to further reduce the number of records on the unbalanced side.

After removing null records and records with customers that start with the letter M, the dataset now has 4202912 non-fraudulent transactions and 8213 fraud transactions. In other words, no lines related to fraud were removed.

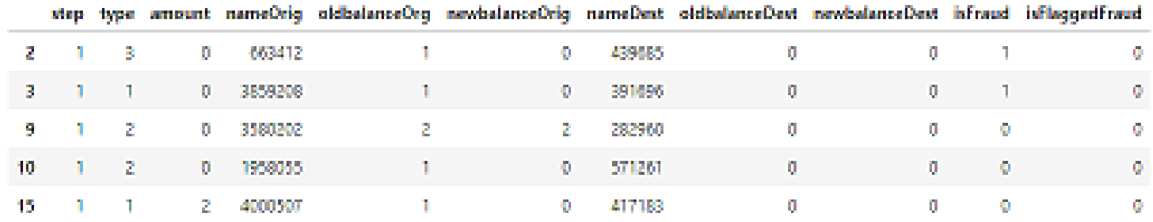
5.2.2 Attributes Discretization

After the presentation in section 4.2.3, the need to discretize the financial attributes due to their great sparsity and variation in values became clear. Discretization is a concept that involves transforming numerical values into a category based on conditions.

In this context, a discretization was applied based on the quartiles demonstrated in section 4.2.3. In this way, the values were distributed into categories that define whether the value is in the first 25%, between 25% and 50%, 50% and 75% or above 75%.

After the conversion, the attributes were as shown in Figure 6. In this same figure it is also important to note that the discretization was also applied to the String columns (type, nameOrig and nameDest) due to the training algorithms that cannot deal with this type of given away.

However, in these cases, the conversion was simpler: each String received a unique number. Thus, customers received the same id regardless of whether it was in the nameOrig or nameDest column.

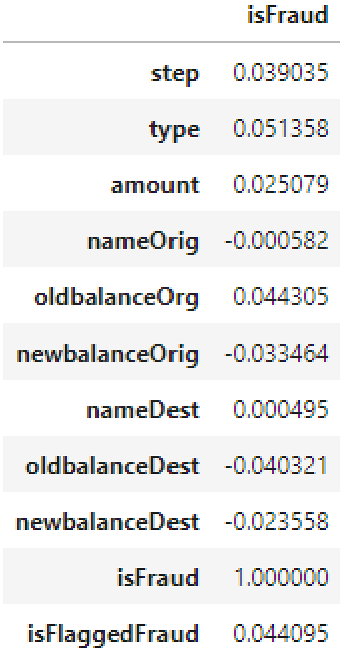


	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
2	1	3	0	663412	1	0	439685	0	0	1	0
3	1	1	0	3859208	1	0	391696	0	0	1	0
9	1	2	0	3580202	2	2	282960	0	0	0	0
10	1	2	0	1958055	1	0	571261	0	0	0	0
15	1	1	2	4000507	1	0	417183	0	0	0	0

Figura 6. Cinco primeiras do dataset após discretização

5.3.3 Attributes correlation

To identify whether other attributes can be removed or whether there is any attribute with a strong relationship with the objective of predicting whether it is a fraud record or not, the Pearson Correlation calculation was used for this analysis. Figure 7 shows the result.



	isFraud
step	0.039035
type	0.051358
amount	0.025079
nameOrig	-0.000582
oldbalanceOrg	0.044305
newbalanceOrig	-0.033464
nameDest	0.000495
oldbalanceDest	-0.040321
newbalanceDest	-0.023558
isFraud	1.000000
isFlaggedFraud	0.044095

Figura 7. Correção de Pearson entre os atributos e a classe

It is known that, according to the Pearson Correlation rule, the closer the result value is to 1 or -1, the greater its positive or negative correlation, respectively. Therefore, it is noted that there is no single attribute or a small set of attributes that has sufficient correlation to be trained separately from the others. However, there are 2 attributes that have a correlation close to zero: nameOrig and nameDest.

It can even be said that these attributes have a correlation value of zero, if we round to 3 decimal places. Therefore, these two attributes were also removed from the dataset.

5.3.4 Attribute normalization

Many Machine Learning algorithms perform distance calculations between attributes. However, it is necessary to equalize these magnitudes so that the algorithms do not consider the step values with more weight than the other attributes, as step can go up to the unit of hundreds while the financial attributes are in the units place.

One way to perform this equalization is through attribute normalization. There are different techniques for attribute normalization and, in this project, we opted for min-max normalization due to its practicality and wide use [15].

5.3.5 Class balancing

There are two basic types of dataset balancing for unbalanced classes: simulate and insert records similar to the desired class or remove records related to the class with the highest number [16, 17].

For the reality of this base and this concept, we cannot disregard outliers as these may precisely be cases of fraud. Furthermore, simulating more random cases can create records of fraud associating behaviors that do not reflect a real possibility and, thus, harm learning in any algorithm.

For these reasons, it was decided to remove most of the records until the imbalance becomes smaller. In other words, the objective is not to completely remove the imbalance, but rather to reduce it. This strategy was adopted due to the study by Krawczyk [18], which classified class imbalance into two categories: Highly unbalanced (1:1000+) and slightly unbalanced (minimum of 1:4).

To reduce the imbalance above 10%, it was necessary to remove most of the dataset. After removal, the dataset was left with 42029 non-fraudulent transactions (83.65%) and 8213 fraudulent transactions (16.35%).

6. Experiments and Analysis with Machine Learning

This section describes the algorithms that were used, the needs for changes in the dataset in each case and the motivations for choosing these trainings.

6.1 Model Assessment

Due to the unbalanced nature of the dataset, it is not recommended to use the accuracy metric as the value to be observed for evaluation. This occurs because the dataset is unbalanced and, therefore, it is natural that the accuracy is high [16, 17].

As the problem is directly linked to the need to correctly identify when a transaction is fraudulent, it is necessary to look more closely at the adjustments related to this class. Furthermore, in case of errors, this context clearly benefits when the model predicts a false fraudulent transaction than the bet.

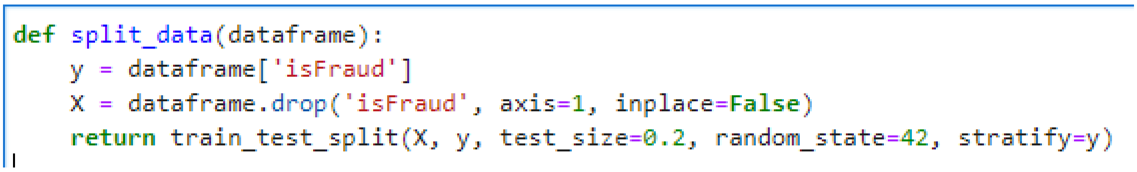
For this reason, we chose to evaluate the model using the Confusion Matrix metrics, which presents exactly the correct hits (positive and negative) and false hits (positive and negative). Another metric directly linked to the Confusion Matrix and also used in the evaluation of this project is the precision and recall value.

6.2 Dataset split

Before carrying out any experiments with Machine Learning algorithms, it was necessary to separate the dataset into training data and test data. This type of approach is highly recommended so that the model can be evaluated in a more realistic way. The model is trained with the vast majority of data and a separate test set is reserved to evaluate it through record predictions that the model did not use during its training previously.

It is important to highlight that this approach is used solely and exclusively to evaluate the model. When the strategy is defined, all data is used to train models that will be put into production.

To separate the dataset, the train_test_split module from the sklearn library was used, as shown in Figure 8. With this method it was possible not only to separate the data by choosing the desired percentage, but also to use the equal partitioning strategy between the classes. In other words, by separating 20% of the data for testing, the algorithm guarantees that it will maintain the same proportion of classes in the dataset.



```
def split_data(dataframe):
    y = dataframe['isFraud']
    X = dataframe.drop('isFraud', axis=1, inplace=False)
    return train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

Figura 8. Método criado para separação do dataset

6.3 Study on dimensionality reduction

Before applying any data to the algorithms, an analysis was made on the dimensionality reduction of the attributes. By reducing the dimension from 9 attributes to 1, for example, it would have gained not only the training process but also a greater application of algorithms to be used and evaluated together with the original dimensionality data.

The study was carried out using the t-Distributed Stochastic Neighbor Embedding (t-NSE) algorithm. This algorithm is a stochastic method to reduce the dimensionality of data and visualize the result.

When applying the algorithm through the sklearn library, the result presented in Figure 9 was obtained. With this figure, it can be seen that there was no clear pattern for the dataset with reduced dimensionality to be applied to other algorithms.

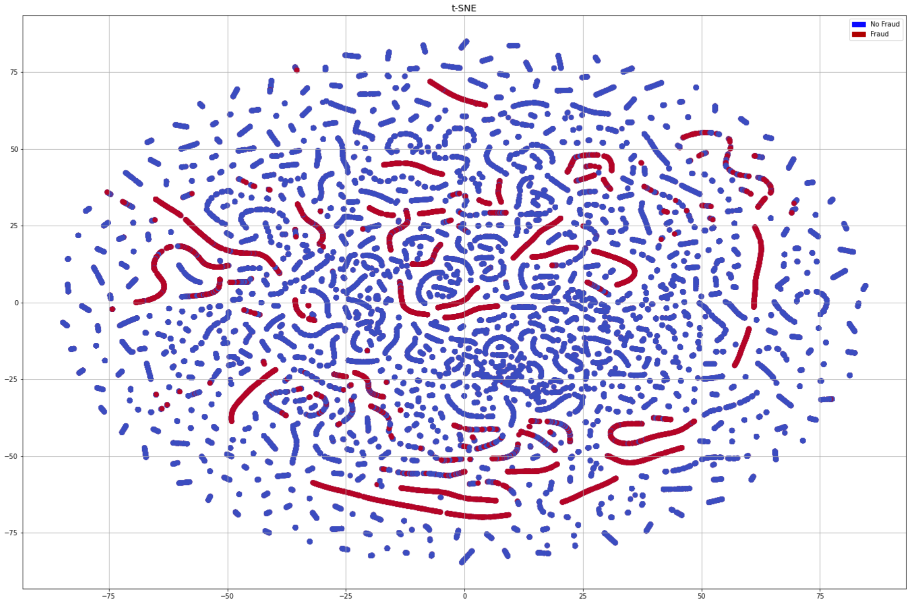


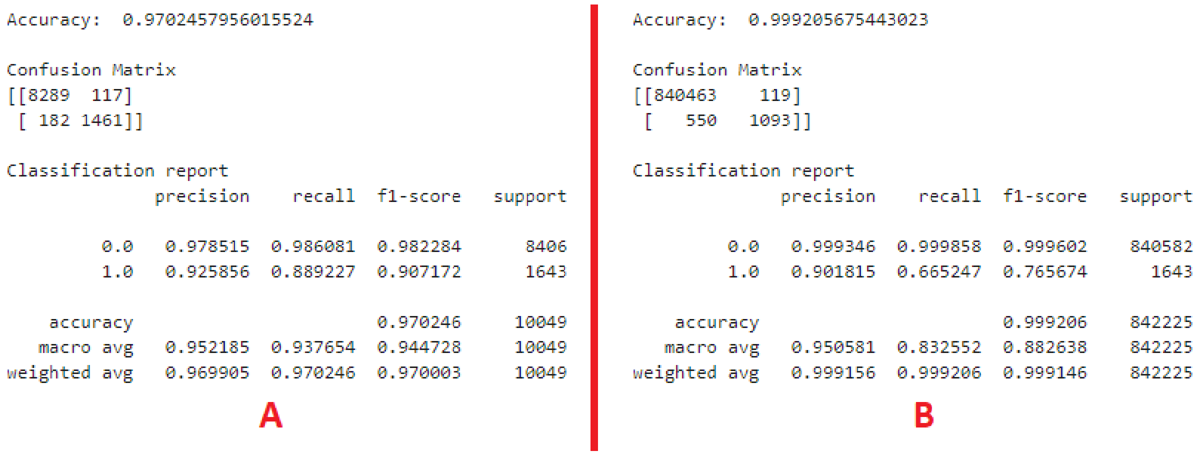
Figura 9. Demonstração do resultado após t-NSE

It is important to highlight that the algorithm was applied with different parameters and this was the clearest result obtained, as shown in the complete notebook with the entire process in code.

6.4 Decision Tree

In general terms, the Decision Tree algorithm consists of training a model that makes predictions based on logical rules created from the training. In this way, the algorithm's prediction occurs by analyzing one or more attributes and checking which branch of choice it should continue with the analysis until it reaches the nodes of the tree where the class of that path taken is found.

To train the model, the sklearn library algorithm was used and the decision tree was trained with 2 bases: full base and reduced base (class with lower imbalance). Figure 10 shows the results of the two training sessions.



```
Accuracy:  0.9702457956015524

Confusion Matrix
[[8289  117]
 [ 182 1461]]

Classification report
              precision    recall  f1-score   support

         0.0   0.978515  0.986081  0.982284      8406
         1.0   0.925856  0.889227  0.907172      1643

    accuracy                       0.970246     10049
   macro avg   0.952185  0.937654  0.944728     10049
weighted avg   0.969905  0.970246  0.970003     10049
```

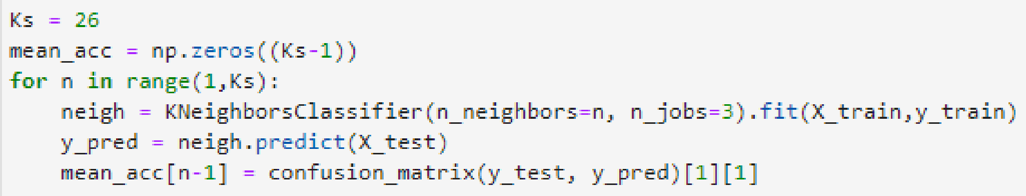
```
Accuracy:  0.999205675443023

Confusion Matrix
[[840463    119]
 [   550   1093]]

Classification report
              precision    recall  f1-score   support

         0.0   0.999346  0.999858  0.999602    840582
         1.0   0.901815  0.665247  0.765674      1643

    accuracy                       0.999206    842225
   macro avg   0.950581  0.832552  0.882638    842225
weighted avg   0.999156  0.999206  0.999146    842225
```

Figura 10. Árvore de Decisão - Comparação entre base reduzida (A) e base completa (B)

6.5 k-Nearest Neighbors

In general terms, the k-NN algorithm seeks to separate classes into clusters and, when performing the prediction, find the cluster to which the input data belongs. To perform this task, the algorithm performs some distance calculation between points.

As this is an algorithm with a high variety of parameters, it was decided to leave the library's own defaults and change only the number of k neighbors. To discover the best k for the dataset, a simple iteration algorithm was run between different k's to compare the result based on the number of true positive cases (fraud).



```python
Ks = 26
mean_acc = np.zeros((Ks-1))
for n in range(1,Ks):
    neigh = KNeighborsClassifier(n_neighbors=n, n_jobs=3).fit(X_train,y_train)
    y_pred = neigh.predict(X_test)
    mean_acc[n-1] = confusion_matrix(y_test, y_pred)[1][1]
```

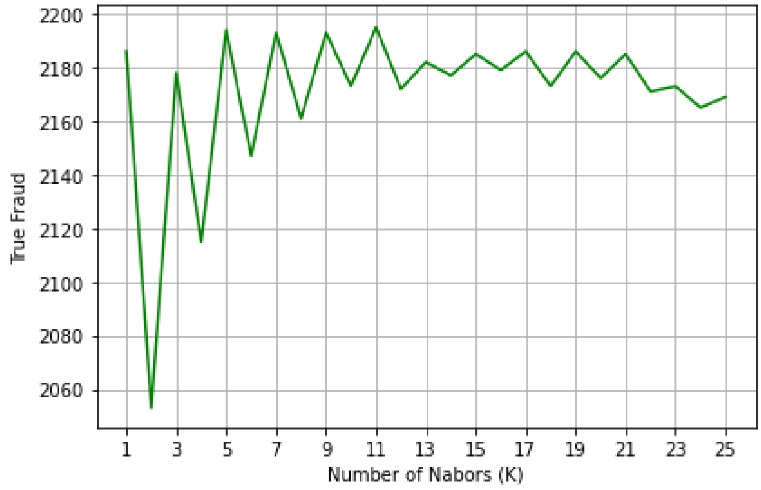
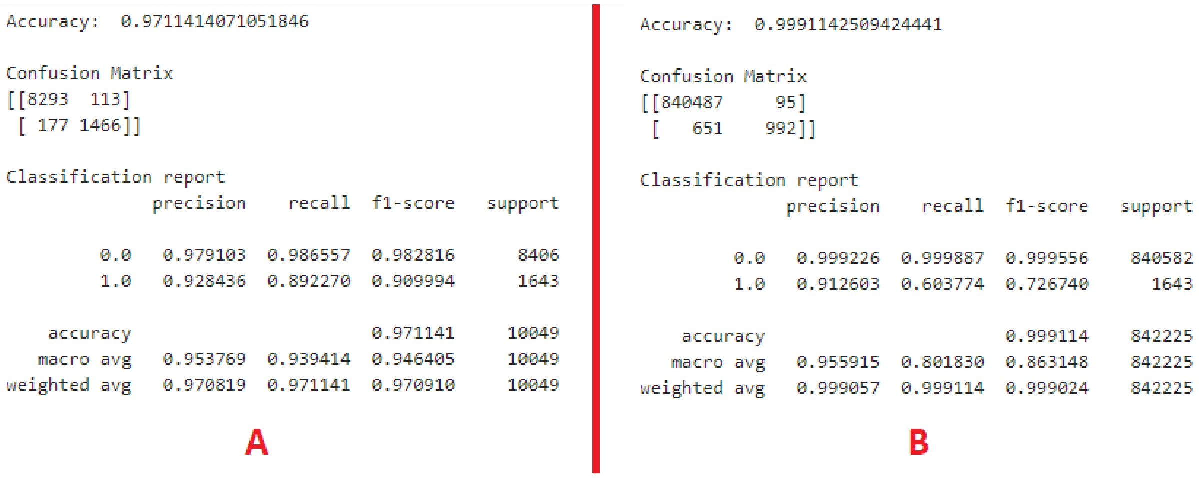


Figura 11. Comparação de True Fraud entre diferentes k's

When observing Figure 11, it can be seen that the best for k is 5, 7, 9 or 11. After this value, there is a drop in the number of correct answers. For this reason, the final model was trained with the value of k=5 and obtained the results shown in Figure 12.



```
Accuracy:  0.9711414071051846

Confusion Matrix
[[8293  113]
 [ 177 1466]]

Classification report
              precision    recall  f1-score   support

         0.0   0.979103  0.986557  0.982816      8406
         1.0   0.928436  0.892270  0.909994      1643

    accuracy                       0.971141     10049
   macro avg   0.953769  0.939414  0.946405     10049
weighted avg   0.970819  0.971141  0.970910     10049
```

```
Accuracy:  0.9991142509424441

Confusion Matrix
[[840487     95]
 [   651    992]]

Classification report
              precision    recall  f1-score   support

         0.0   0.999226  0.999887  0.999556    840582
         1.0   0.912603  0.603774  0.726740      1643

    accuracy                       0.999114    842225
   macro avg   0.955915  0.801830  0.863148    842225
weighted avg   0.999057  0.999114  0.999024    842225
```

Figura 12. k-NN - Comparação entre base reduzida (A) e base completa (B)

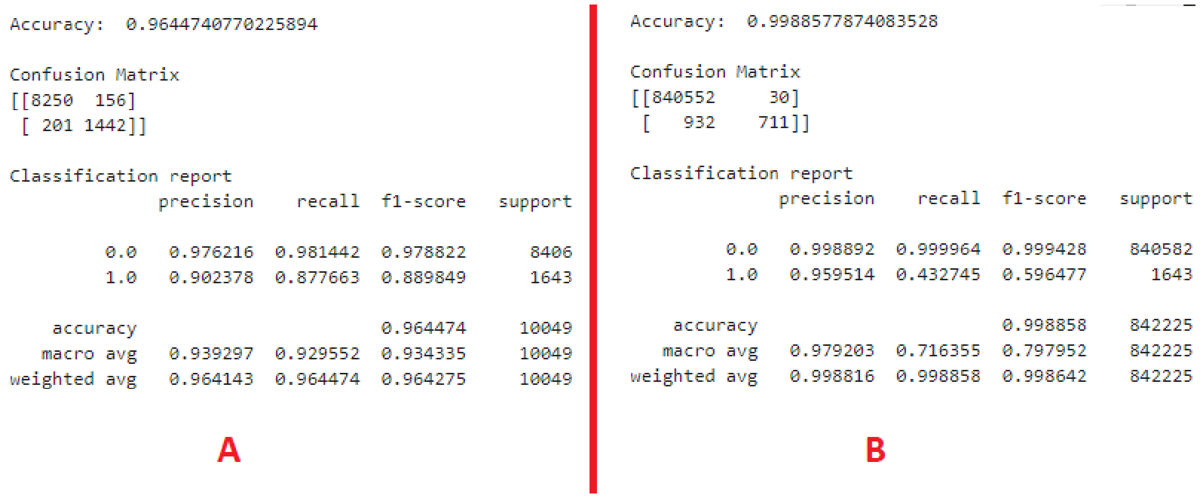
6.6 Multi-Layer Perceptron

MLP consists of an algorithm that learns to find the target class by creating a function and an alpha constant. Furthermore, as the name suggests, it is a multi-layer classifier. Finally, there are different methods for carrying out learning.

As it is a complex algorithm with many possible combinations of parameters, the optimal parameter search method called GridSearch [9.4] was used. This method tests the combination of possible parameters and returns the best configurations based on the chosen acceptance criteria. Again, as we are not looking at accuracy, the chosen evaluation method was recall.

Thus, the model was trained with 3x1000 layers, logistic activation function, ADAM method, an initial alpha of 0.0005 and a maximum of 1000 iterations. However, it should be noted that these optimal parameters were found based on the possibilities created for executing GridSearch. And, due to hardware limitations, it was not possible to test more possibilities.

Figure 13 demonstrates the comparative result between the reduced and full base models.



```
Accuracy:  0.9644740770225894

Confusion Matrix
[[8250  156]
 [ 201 1442]]

Classification report
              precision    recall  f1-score   support

         0.0   0.976216  0.981442  0.978822      8406
         1.0   0.902378  0.877663  0.889849      1643

    accuracy                       0.964474     10049
   macro avg   0.939297  0.929552  0.934335     10049
weighted avg   0.964143  0.964474  0.964275     10049
```

```
Accuracy:  0.9988577874083528

Confusion Matrix
[[840552     30]
 [   932    711]]

Classification report
              precision    recall  f1-score   support

         0.0   0.998892  0.999964  0.999428    840582
         1.0   0.959514  0.432745  0.596477      1643

    accuracy                       0.998858    842225
   macro avg   0.979203  0.716355  0.797952    842225
weighted avg   0.998816  0.998858  0.998642    842225
```

Figura 13. MLP - Comparação entre base reduzida (A) e base completa (B)

7. Results Discussion

This section consolidates the results presented in section 6. When comparing the confusion matrix between all trained models, it is noted that there was a difference in the results obtained. Figure 14 demonstrates this comparison in terms of predictions related to the fraudulent class.

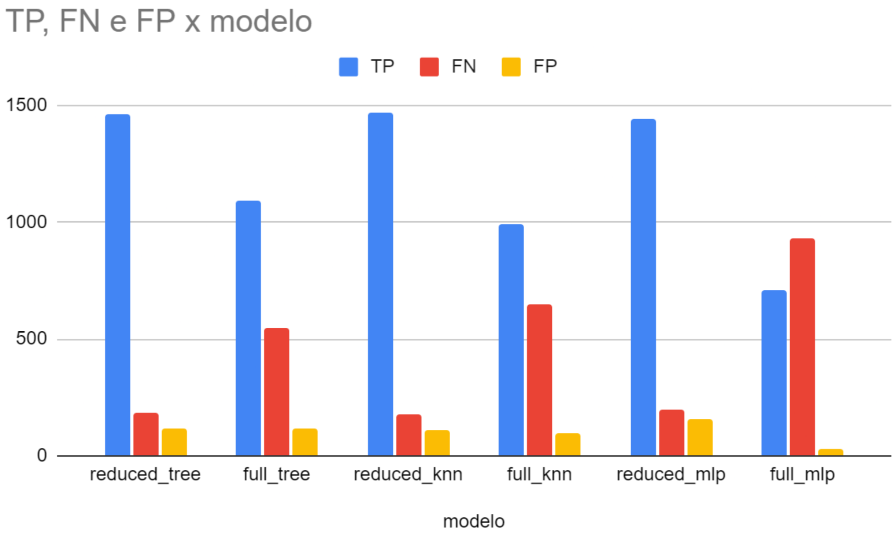


Figura 14. Comparação de TP, FN e FP de cada modelo

The first clear point is the drop in correct predictions (TP, in blue) when changing the model trained with the reduced dataset (reduced_) and the model trained with the complete dataset (full_). Consequently, the number of wrong predictions for the worst case, where the model does not find fraud when it should (FN, in red), increased.

A possible explanation for this result is the gigantic increase in data, leaving the fraud class with less than 0.5% of records. In this case, it is actually plausible and expected that the model will not be able to develop the necessary functions for this prediction. In fact, this result reinforces the need to deal with class imbalance.

Figure 15 presents the same model comparison, but taking into account the precision and recall metrics. Obviously, as the recall has the FN number in its formula, it is expected that it will have a reduction in the cases of models trained with the complete dataset.

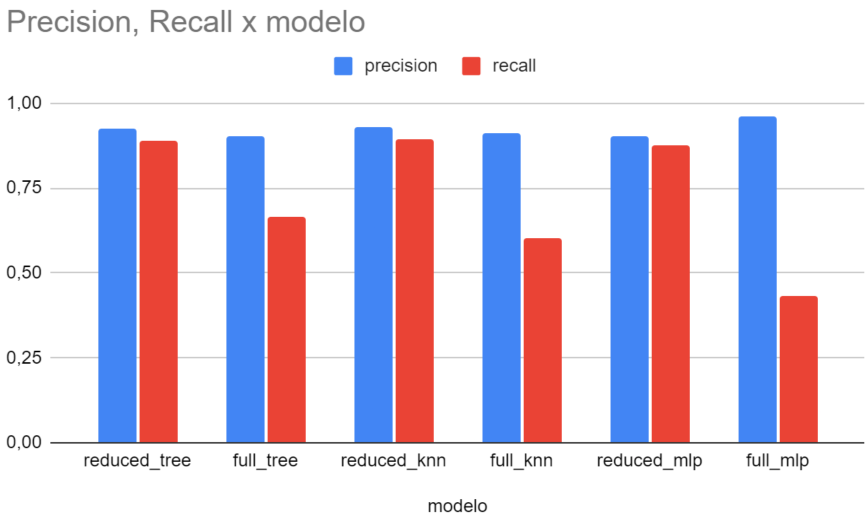


Figura 15. Comparação de Precision e Recall de cada modelo

When looking at Figures 14 and 15, it is easy to see that the models had a very similar result to each other, in relation to the fraud class. In other words, no model had a disparity in either class accuracy or error type (FN and FP).

8. Conclusion and Future Work

8.1 Conclusion

Some conclusions can be drawn from the results obtained in this project. The first point of emphasis is related to the analysis of the dataset. In this context, it is possible to observe that the quality of the dataset directly affected the prediction quality of the model: using an extremely unbalanced dataset generated models with very poor results for predicting the unbalanced class. Therefore, it is always important to carry out good pre-processing on the dataset before defining the algorithm and parameters used.

Another consideration is related to the choice of algorithm for prediction. It is possible to achieve similar results in different models and not just in more sophisticated and complex models like MLP. This type of result means that simpler models with greater explainability can be used as well as others.

So, when analyzing an unbalanced dataset in which we cannot increase the size of the unbalanced class and we know that the result will not be satisfactory, it is worth carrying out an analysis whether the quality of more complex models will remain close to the quality of simpler models. If this is the case, the use of more explanatory predictive models helps in decision making, not only avoiding a greater number of errors, but also helping the user to interpret and understand the problem.

It is worth reflecting on the focus on the explainability of the model for decision making if we have similar results between models when applying this concept in production.

Working with an unbalanced dataset can be a problem without a perfect solution. However, if there is a gap in applying a way to interpret the model's results, its false predictions can be corrected by the user and applied correctly.

8.2 Limitations

It is important to highlight that the objective of this work was to explore the problem of an unbalanced dataset for fraud detection and how the explainability of simple models could support decision making. For this reason, the work has several limitations:

1. Dataset was balanced by removing records and no tests were conducted by adding records;
2. GridSearch used to search for optimal parameters was only carried out to a limited extent due to hardware limitations
3. Deeper network models and more complex algorithms than MLP have not been tested to the end also due to hardware limitations
4. All results extracted from this project are related not only to these limitations, but also to limitations of the dataset itself.

8.3 Future work

Due to the limitations presented in section 8.2, it is clear that the project could still be much further explored:

1. What would the result be like if a method of increasing records, such as SMOTE, were used?
2. Would other neural network algorithms have presented much better results?
3. Would more complete hyper-parameter searches have presented other parameters and, as a consequence, a better result for the MLP?

Many challenges remain open in the field of Machine Learning. These challenges become even clearer and more problematic when trying to apply concepts with data present in everyday life. Therefore, it is necessary to carry out more complete studies and tests.

9. Links

1. https://www.kaggle.com/
2. https://www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction
3. https://python.org
4. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
5. Github containing complete notebook with all exploration, dataset analysis, training and model evaluation: https://github.com/romponciano/fraud-transfer-detection
6. Link to database used: https://www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction

10. Referências

[1] Awoyemi, J; Adetunmbi, A.; Oluwadare, S. **Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis**. Nigeria. IEEE, 2017

[2] FBI Report, **Increased Use of Mobile Banking Apps Could Lead to Exploitation.** Estados Unidos, 2020.

[3] Padin, G. **Pix: após alta de crimes, veja como se proteger de golpes e sequestros**. Brasil. R7, 2021

[4] Máximo, W. **Pix terá medidas de segurança para coibir sequestros e roubos**. Brasil. Agência Brasil, 2021

[5] Ganganwar, V. **An overview of classification algorithms for imbalanced datasets**. India, 2012

[6] Varmedja, D *et. al.* **Credit Card Fraud Detection - Machine Learning methods**. Sérvia, 2019

[7] Makki, S. *et. al.* **An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection**. IEEE, 2019

[8] Perols, J. **Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms**. American Accounting Association, 2011

[9] Gunning, D. *et. al* **XAI—Explainable artificial intelligence**. Science Robotics, 2019

[10] Amina, A.; Berrada, M. **Peeking inside the black-box: a survey on explainable artificial intelligence (XAI).** IEEE, 2018

[11] Wojciech, S; Wiegand, T; Müller, K. **Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models.** arXiv, 2017

[12] Kumar, P. *et. al.* **Credit Card Fraudulent Detection Using Machine Learning Algorithm.** IJREAM, 2020

[13] Arun, D.; Rad, P. **Opportunities and challenges in explainable artificial intelligence (xai): A survey**. arXiv, 2020

[14] Sebastian, R; Patterson, J; Nolet, C. **Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence.** Information, 2020

[15] Kumar, S. *et. al.* **An Improved Fuzzy Min–Max Neural Network for Data Classification.** IEEE, 2019

[16] Pozzolo, A.; Caelen, O.; Bontempi, G. **When is Undersampling Effective in Unbalanced Classification Tasks?** ECML PKDD, 2015

[17] Pozzolo, A. *et. al.* **Calibrating Probability with Undersampling for Unbalanced Classification.** IEEE, 2015

[18] Krawczyk, B. **Learning from imbalanced data: open challenges and future directions.** Prog Artif Intell, 2016.