Identificação de possíveis fraudes em transações bancárias utilizando algoritmos de Machine Learning

Luiza Silva Conte, Silvestre Calero Leite
Orientador: Henrique Dezani, Coorientador: José Alexandre Ducatti

e-mail:
luiza.conte@fatec.sp.gov.br; silvestre.leite@fatec.sp.gov.br;
henrique.dezani@fatec.sp.gov.br; jose.ducatti@fatec.sp.gov.br

Resumo: O uso de *Machine Learning* é crescente em diversas áreas de pesquisa, com diversas finalidades, como automatizar tarefas complexas ou fazer predições. A fraude em pagamentos realizados *online* é um problema que ainda é frequente e acarreta diversos impactos negativos, tanto para os agentes financeiros, quanto para seus clientes. O projeto propõe a aplicação de técnicas de *Machine Learning* para auxiliar na detecção de transações fraudulentas, com sua utilização é possível predizer se uma transação de pagamento tem potenciais chances de ser fraudulenta ou não.

Palavras-chave: Transação bancária, fraudes, *python* e *Machine learning*

Abstract: *The use of Machine Learning is increasing in several areas of research, with different purposes, such as automating complex tasks or making predictions. Fraud in payments made online is a problem that is still frequent and has several negative impacts, both for financial agents and for their customers. The project proposes the application of Machine Learning techniques to assist in the detection of fraudulent transactions, with its use it is possible to predict whether a payment transaction has potential chances of being fraudulent or not.*

Keywords: *Banking, Frauds, Python and Machine Learning*

1 Introdução

A inovação tecnológica ocasionou uma grande mudança no ambiente de negócios ao longo dos anos, em especial o financeiro, objetivando se tornar mais produtivo e eficiente para as empresas, ao mesmo tempo em que seu alcance quase se tornou ubíquo para os clientes. Nesta situação, os clientes contam com serviço acessível onde é cada vez mais simples se realizar transações financeiras.

Esta mudança no ecossistema de transações financeiras se deu por meio das tecnologias de transformação digital, que podem ocasionar em alguns problemas relacionados à sua utilização como, por exemplo, o aumento de fraudes e crimes cibernéticos. De acordo com o relatório de fraudes globais da empresa Kroll para os anos de 2017 e 2018 (CARVALHO, 2018) no Brasil, 89% das empresas do setor financeiro sofreram algum tipo de incidente cibernético, por exemplo, fraudes, ataques a seus sistemas informatizados ou roubos de informação e 91% dessas empresas acreditam que a exposição à fraude aumentou. Esses dados evidenciam a grande incidência de eventos fraudulentos em operações financeiras informatizadas.

Visando inibir esse número crescente no aumento das fraudes e crimes cibernéticos, pode-se identificar duas linhas principais de ação. Na primeira delas a abordagem consiste no

aumento da segurança nas transações bancárias e acesso a informações. Na segunda busca-se operações que não são habituais para o usuário verdadeiro. Dessa forma, é possível identificar que uma operação não autorizada está em curso.

O objetivo principal desse trabalho é identificar transações fraudulentas por meio de *Machine Learning*, a fim de lidar com grande volume de dados, distinguir as fraudes das transações normais e evitar maiores prejuízos aos agentes financeiros.

2 Justificativa

A escolha do presente tema justifica-se pelo crescente uso de meios tecnológicos para realização de transações bancárias e o aumento de golpes aplicados que, segundo levantamento da Federação Brasileira de Bancos (Febraban) divulgado pela Agência Brasil (2022), cresceram 165% no primeiro semestre de 2021. Diante do cenário de pandemia os meios tecnológicos se tornaram a alternativa de grande importância, muitas vezes para um público que não estava habituado com tal. Por isso, utilizar de ferramentas que possam identificar e prevenir essas situações se faz cada vez mais necessário para a realização de transações bancárias seguras.

3 Objetivo

Identificar possíveis fraudes em transações financeiras feitas por meio de *Machine Learning*.

4 Fundamentação Teórica

4.1. Transações Bancárias e Fraudes

4.1.1. Transações Bancárias

Transação bancária é todo procedimento que envolva troca de recursos, podem ser feitas tanto por pessoa física, quanto jurídica, segundo Redação IUGU (2022). Alguns exemplos de transações são: transações em espécie, em cartões ou pelas instituições bancárias.

4.1.2. Fraudes

Segundo Supremo Concursos (2022) as fraudes nas transações financeiras consistem em métodos de estelionato, em que são praticados contratos ou convenções, promovendo uma falsa concepção de vantagem ilegal para si ou para outros. Por exemplo da utilização de dados alterados propositalmente com o intuito de furto ou golpe.

4.2. Python

Com o intuito de suprir uma necessidade e gerar uma linguagem de programação *script* simples, no início da década de 1990, Guido Van Rossun criou a versão de abertura do Python,

até então conhecida por Modula-3. Atualmente, a linguagem é uma tendência mundial devido sua facilidade de uso, podendo ser aplicada em diversas áreas do desenvolvimento, como *Data Science*, desenvolvimento *Back-End*, *scripts*, aplicação *web*, entre outros cenários.

Neste projeto foi utilizado esta linguagem de programação, pois, segundo o Science Academy (2020), apesar da simplicidade *Python* é uma linguagem de alto nível, tendo em vista que não há dificuldades com a sintaxe, direcionando o foco do desenvolvedor para a resolução que será implementada.

Com sua implementação simples, a prototipagem pode ser feita em plataformas de interface *web*, como utilizado no projeto, o *Google Colab*, que permite que as células sejam executadas em blocos separadamente, tornando simples a forma de manuseio e correção, além da utilização de uma máquina virtual, o que alivia o *hardware* local sendo instanciada no servidor da plataforma *web*. Outra vantagem da utilização do *Python* são as inúmeras bibliotecas disponíveis para tratamento de dados, entre elas estão*: Pandas, PySpark, NumPy, SciPy*, entre outras.

Devido à popularidade do *Python*, foram criadas inúmeras bibliotecas com as mais diversas funções, neste projeto foram utilizadas as bibliotecas *Pandas* e *PySpark*;

• *Pandas,* segundo Bruna Mulinari, é uma biblioteca que auxilia nas manipulações de estruturas por meio de medidas estatísticas como a média, a mediana, o desvio e os percentis, que nos permite analisar o comportamento da nossa informação em exibição. *Pandas* pode ser utilizado em operações para manipular tabelas numéricas e séries temporais, além do alinhamento automático ou explícito dos dados, tratamento flexível e simplificado de dados ausentes, uso de operações, combinações e operações relacionais, informações estatísticas, séries temporais e visualização de dados.

• *PySpark* segundo a própria documentação do projeto, é uma interface, escrita em *Python,* para o *Apache Spark.* Que permite escrever aplicações Spark utilizando *APIs Python* que interagem com o *core* do *Spark*. É uma biblioteca utilizada para processar grandes quantidades de dados, frequentemente utilizada em *Big Datas* (Banco de Dados). Além da função que implementada no *core* do *Spark* e estão disponíveis para serem usadas via *API*.

4.4. *Machine Learning*

De acordo com Oracle (2022), *Machine learning* é um subconjunto da Inteligência Artificial em que os sistemas aprendem de acordo com os dados que consomem. Esta aprendizagem pode ser aplicada no reconhecimento de padrões, análises preditivas, melhoras no desempenho de sistemas, combate a fraudes, entre outros. Os métodos de aprendizagem do ML são subdivididos em: Não Supervisionado, Supervisionado e Por Reforço.

Um conceito necessário para entendimento, é o conceito das variáveis, são elas: independente/preditora (são fornecidas para o aprendizado e exercem influência sobre a variável que se deseja encontrar) e dependente/alvo (está é o resultado desejado).

4.4.1. *Machine Learning* Não Supervisionado

No método Não Supervisionado, segundo Hewlett Packard Enterprise (2022), a abordagem utilizada é mais independente, desta forma ele aprende sem a necessidade de uma orientação próxima e constante, ou seja, seu treinamento não possui rótulos ou saída específica definida.

4.4.2. *Machine Learning* Supervisionado

O método Supervisionado é aplicado quando tenta encontrar relação entre uma(s) variável(is) preditora e uma variável alvo, de acordo com Hewlett Packard Enterprise (2022). Neste modelo, as variáveis devem ser rotuladas e/ou com valores corretos, para que a análise feita seja a mais assertiva.

A Regressão linear para Damaceno (2020), é um conceito estatístico muito utilizado no *Machine Learning* Supervisionado, em que é feita através de um diagrama de dispersão, dessa forma, por meio de valores contínuos no diagrama, é traçado uma reta calculando sua correlação.

4.4.3. *Machine Learning* Por Reforço

Para Data Science Academy (2022), o método Por Reforço é treinado para tomar uma série de decisões, desta forma seu aprendizado é por meio de metas em ambientes incertos. Neste método o sistema age por tentativas e erros, sendo recompensado ou penalizado, seu objetivo é reduzir ao máximo as penalidades, maximizando as recompensas.

5 Trabalhos Similares

Atualmente as instituições financeiras fazem uso medidas de segurança baseadas em Inteligência Artificial (IA), para assegurar que seus clientes não estejam sendo alvos de nenhuma fraude. Esse uso abriu as portas para *Machine Learning*, que consiste em fazer um reconhecimento de padrões (MARYALENE, 2017). O *Machine Learning* possibilitou o desenvolvimento de diversos métodos, que podem ser utilizados em muitas áreas de estudo, como financeira, processamento de imagens, medicina, entre outras. Alguns dos métodos mais conhecidos são árvores de decisão, regressão logística, algoritmos de agregação, redes neurais e aprendizado profundo. Em seguida, trabalhos que aplicam algumas dessas técnicas.

MATTOS (2021) propõe a aplicação de técnicas de *Machine Learning* para predizer se uma transação de pagamento online tem potenciais chances de ser fraudulenta ou não. Para tal, utiliza os algoritmos de Regressão Logística e de Árvore de Decisão, em seus resultados o modelo de Regressão Logística apresenta melhor desempenho ao classificar os dados atingindo uma acurácia e precisão de 95%.

SOUZA (2021), avaliou modelos preditivos aplicados a um conjunto de dados de transações de cartões de crédito previamente processado, com o objetivo de identificar a abordagem mais adequada para predições de fraudes em transações de cartões de crédito, a metodologia que foi aplicada envolveu analisar os hiper parâmetros dos algoritmos, com o propósito de otimizar os resultados e realizar uma classificação para cada subconjunto da base de dados.

6 Metodologia

6.1. Tipo de pesquisa

O trabalho foi feito por meio da montagem de um artigo que busca trazer por meio de *Machine Learning*, metodologia a fim de identificar fraudes em transações bancárias, portanto este artigo possui uma metodologia qualitativa.

6.2. Métodos utilizados

O projeto utiliza o conceito de *Machine Learning*, por meio da plataforma *Google Colab*, a linguagem de programação foi *Python* e suas bibliotecas, como *Spark* e *Pandas*, a base de dados *Fraudulent Transactions Data* e o conceito de *machine learning* aplicado à regressão**.**

6.3 Base de Dados

A Base de Dados utilizada para treinar o algoritmo sobre reconhecimento de transações fraudulentas, a *Fraudulent Transactions Data* (Dados de Transações Fraudulentas), é uma base disponibilizada ao público pela *Kaggle*. Esta base possui o formato de uma tabela, que consistem 10 colunas e 6.362.620 linhas, dentre elas estão as variáveis, que estão dispostas abaixo:

-*step*: mapeia uma unidade de tempo no mundo real. Neste caso, 1 passo é 1 hora com total de 744 etapas (simulação de 30 dias).
-*type*: *CASH-IN, CASH-OUT*, DÉBITO, PAGAMENTO e TRANSFERÊNCIA.
-*amount*: valor da transação em moeda local.
-*nameOrig*: cliente que iniciou a transação
-*oldbalanceOrg*: saldo inicial antes da transação
-*newbalanceOrig*: novo saldo após a transação
-*nameDest*: cliente que é o destinatário da transação
-*oldbalanceDest*: destinatário do saldo inicial antes da transação.
-*newbalanceDest*: novo destinatário do saldo após a transação.
-*isFraud*: São as transações feitas pelos agentes fraudulentos dentro da simulação. Neste conjunto de dados específico, o comportamento fraudulento dos agentes visa lucrar ao assumir o controle das contas dos clientes e tentar esvaziar os fundos transferindo para outra conta e depois sacando do sistema.
-*isFlaggedFraud*: O modelo de negócios visa controlar transferências massivas de uma conta para outra e sinaliza tentativas ilegais. Uma tentativa ilegal neste conjunto de dados é uma tentativa de transferir mais de 200.000 em uma única transação.

7 Desenvolvimento

A princípio o desenvolvimento foi dividido em uma série de etapas para facilitar o entendimento de todo o processo de construção do algoritmo e resultados. Dentre os tópicos abordados, temos:

- Instalação das bibliotecas
- Conexão com ambiente de nuvem
- Leitura da base de dados
- Análise e limpeza de dados
- Treinamento do modelo
- Avaliação do modelo

- Impacto das características no aprendizado

7.1 Algoritmo

7.1.1 Instalação das bibliotecas

O desenvolvimento de algoritmos em *Python* possibilita a instalação das bibliotecas necessárias de forma rápida e simples, a figura 1 registra todas as importações feitas.

```
# Instalação das bibliotecas
!pip install pyspark;

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn import metrics
from pyspark import SparkFiles
from pyspark.sql import SparkSession
from google.colab import drive
from sklearn.tree import DecisionTreeClassifier
```

Figura 1: Implementação das importações do modelo

7.1.2 Conexão com ambiente de nuvem

Visando poupar processamento, utilizamos as ferramentas fornecidas pelo Google, para o desenvolvimento, o *Google Colab* como IDE do *Python* que instancia nos servidores do Google uma máquina virtual para o processamento e o *Google Drive* para o armazenamento dos códigos implementados e documentação. O que possibilitou trabalhar com um arquivo de 6.362.620 linhas de forma simplificada e ágil, em que o *Google Colab* conecta de forma simples o ambiente de desenvolvimento ao *Google Drive*. Para isso, o Google solicita uma verificação de conta para montar o ambiente, com isso, quando é rodado o bloco de código observado na figura 2, o Google gera um link de validação, que exige o *login* na conta selecionada. Após confirmação da conta é gerado um *token* de acesso fornecido pelo Google que permite o acesso ao conteúdo do arquivo .csv.

```
[ ]  # Conexão  com ambiente de nuvem
     drive.mount('/content/drive')
```

Figura 2: Conectando com Google Drive

7.1.3 Leitura da base de dados

Com o acesso ao *Google Drive* realizado, foi feito a leitura dos dados com *Spark,* que permite ler e exibir as colunas e informações contidas no arquivo de forma simples.

```
[ ] # Leitura da base de dados
    spark = SparkSession.builder.getOrCreate()
    dataframe = spark.read.csv(SparkFiles.get("/content/drive/MyDrive/TG/data.csv"), header=True, inferSchema=True, sep=",")

[ ] dataframe.createOrReplaceTempView("base")

    query = """
      SELECT *
      FROM base
    """

    spark.sql(query).show()
```

```
+----+--------+---------+-----------+-------------+--------------+-----------+--------------+--------------+-------+--------------+
|step|    type|   amount|   nameOrig|oldbalanceOrg|newbalanceOrig|   nameDest|oldbalanceDest|newbalanceDest|isFraud|isFlaggedFraud|
+----+--------+---------+-----------+-------------+--------------+-----------+--------------+--------------+-------+--------------+
|   1| PAYMENT|  9839.64|C1231006815|     170136.0|     160296.36|M1979787155|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  1864.28|C1666544295|      21249.0|      19384.72|M2044282225|           0.0|           0.0|      0|             0|
|   1|TRANSFER|    181.0|C1305486145|        181.0|           0.0| C553264065|           0.0|           0.0|      1|             0|
|   1|CASH_OUT|    181.0| C840083671|        181.0|           0.0|  C38997010|       21182.0|           0.0|      1|             0|
|   1| PAYMENT| 11668.14|C2048537720|      41554.0|      29885.86|M1230701703|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  7817.71|  C90045638|      53860.0|      46042.29| M573487274|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  7107.77| C154988899|     183195.0|     176087.23| M408069119|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  7861.64|C1912850431|     176087.23|     168225.59| M633326333|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  4024.36|C1265012928|       2671.0|           0.0|M1176932104|           0.0|           0.0|      0|             0|
|   1|   DEBIT|  5337.77| C712410124|      41720.0|      36382.23| C195600860|       41898.0|       40348.79|      0|             0|
|   1|   DEBIT|  9644.94|C1900366749|       4465.0|           0.0| C997608398|       10845.0|      157982.12|      0|             0|
|   1| PAYMENT|  3099.97| C249177573|      20771.0|      17671.03|M2096539129|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  2560.74|C1648232591|       5070.0|       2509.26| M972865270|           0.0|           0.0|      0|             0|
|   1| PAYMENT| 11633.76|C1716932897|      10127.0|           0.0| M801569151|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  4098.78|C1026483832|     503264.0|     499165.22|M1635378213|           0.0|           0.0|      0|             0|
|   1|CASH_OUT|229133.94| C905080434|      15325.0|           0.0| C476402209|        5083.0|       51513.44|      0|             0|
|   1| PAYMENT|  1563.82| C761750706|        450.0|           0.0|M1731217984|           0.0|           0.0|      0|             0|
|   1| PAYMENT|  1157.86|C1237762639|      21156.0|      19998.14|M1877062907|           0.0|           0.0|      0|             0|
|   1| PAYMENT|   671.64|C2033524545|      15123.0|      14451.36| M473053293|           0.0|           0.0|      0|             0|
|   1|TRANSFER| 215310.3|C1670993182|        705.0|           0.0|C1100439041|       22425.0|           0.0|      0|             0|
+----+--------+---------+-----------+-------------+--------------+-----------+--------------+--------------+-------+--------------+
only showing top 20 rows
```

Figura 3: Leitura do arquivo com extensão .csv e exibição dos dados

7.1.4 Análise e limpeza de dados

Com a base de dados a disposição, foi necessário realizar um tratamento na mesma, representado na Figura 4 a análise da quantidade de casos de fraude ou não (quantidade de resultados 0 e 1), em seguida a conversão de campos para números inteiros salvos em uma nova variável (base1).

```
# Análise e limpeza de dados
query= """
      Select isFraud, count(isFraud)
      from base
      group by isFraud
"""

spark.sql(query).show()
```

```
+-------+--------------+
|isFraud|count(isFraud)|
+-------+--------------+
|      1|          8213|
|      0|       6354407|
+-------+--------------+
```

```
[ ] query = """
      SELECT CASE WHEN type = 'PAYMENT ' THEN 0 WHEN type = 'CASH-IN ' THEN 1 WHEN type = 'CASH-OUT ' THEN 2 WHEN type = 'TRANSFER' THEN 3 ELSE 4 END as type,
     amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest,
      newbalanceDest, isFraud
      FROM base
      """

    dataframe = spark.sql(query)
    dataframe.createOrReplaceTempView('base1')
```

Figura 4: Análise dos campos e limpeza de dados

Identificado a quantidade de casos para 0 e 1 (fraude ou não), o comando SQL representado na Figura 5 foi executado visando igualar essa diferença, com esses tratamentos feitos, foi feita a conversão para Pandas e em seguida, aplicada a função *sample*, que retorna uma amostragem aleatória utilizando 70% das linhas, visando evitar casos de *overfitting* (quando nos dados de treino, o modelo tem um desempenho excelente, mas nos dados de teste o resultado é ruim) e o modelo se encontra pronto para treinamento.

```
[8]  query = """
             Select * from ((
               Select * from base1 WHERE isFraud=0 LIMIT 8213)
               UNION ALL (
                 SELECT * from base1 WHERE isFraud=1 Limit 8213))
     """

     dataframe= spark.sql(query)
```

```
df = dataframe.toPandas()

# sample- amostragem aleatória
df = df.sample(frac=0.7)
```

Figura 5: Tratamento dos dados

7.1.5 Treinamento do modelo

A função *head* exibida na Figura 6 foi usada somente para retornar as cinco primeiras linhas do quadro de dados e para dar início ao processo de treinamento, foi atribuído a variável X os dados contidos no *dataFrame* sem a coluna *isFraud,* que indica se a transação em questão é fraudulenta ou não. Em seguida, separamos os dados em treino e teste utilizando *train_test_split*.

```
# Treinamento do modelo
df.head()
```

	type	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud
0	4	9839.64	170136.0	160296.36	0.0	0.0	0
1	4	1864.28	21249.0	19384.72	0.0	0.0	0
2	4	11668.14	41554.0	29885.86	0.0	0.0	0
3	4	7817.71	53860.0	46042.29	0.0	0.0	0
4	4	7107.77	183195.0	176087.23	0.0	0.0	0

```
[ ]  X = df.drop(columns=['isFraud'])
     y = df['isFraud']
```

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Figura 6: Treinamento do modelo removendo a coluna que indica se é ou não fraude

Com os dados já separados, implementamos o classificador da árvore de decisão, através da função *fit* que a partir do conjunto de treinamento (x, y) cria o classificador, em seguida, atribuímos a variável resultado o resultado de *predict* que retorna o valor de regressão para x.

```
[ ]  X_train.shape, X_test.shape

     ((13140, 6), (3286, 6))

[ ]  y_train.shape, y_test.shape

     ((13140,), (3286,))

[ ]
     classificador = DecisionTreeClassifier()
     classificador.fit(X_train, y_train)

     DecisionTreeClassifier()

[ ]  resultado = classificador.predict(X_test)

[ ]  resultado

     array([1, 1, 0, ..., 1, 0, 1], dtype=int32)

     y_test.values

     array([1, 1, 0, ..., 1, 0, 1], dtype=int32)
```

Figura 7: Implementando classificador da árvore de decisão

7.1.6 Avaliação do modelo

Com o modelo treinado, realizamos a avalição dos resultados obtidos, na figura 8 o código utilizado exibe a precisão em cada caso, fraude ou não, e a acurácia do modelo. As métricas utilizadas são, de acordo com Rodrigues (2019):

- A *acuracy* é um indicador geral sobre a performance do modelo.
- A *precision* pode ser utilizada em casos em que os falsos positivos são considerados mais prejudiciais.
- O *recall* pode ser utilizado em casos que os falsos negativos são considerados mais prejudiciais.
- O *F1-Score* trás duas métricas, a *precision* e o *recall*, para uma só, por meio de uma média harmônica.

```
[ ]  # Avaliação do modelo
     print(metrics.classification_report(y_test, resultado))
```

```
                precision    recall  f1-score   support

             0       0.99      0.99      0.99      1630
             1       0.99      0.99      0.99      1656

      accuracy                           0.99      3286
     macro avg       0.99      0.99      0.99      3286
  weighted avg       0.99      0.99      0.99      3286
```

Figura 8: Resultados do treinamento do modelo

7.1.7 Impacto das características no aprendizado

A análise do impacto das características no aprendizado permitiu que realizássemos ajustes no modelo para um resultado mais preciso, ou seja, uma acurácia maior.

```
[ ]  # Impacto das características no aprendizado
     classificador.feature_importances_
```

```
array([0.00844298, 0.12430915, 0.43398507, 0.4078939 , 0.00447547,
       0.02089343])
```

Figura 9: Medindo o impacto de cada coluna no aprendizado do modelo

Essa etapa permitiu identificar e manter somente as colunas mais relevantes, aprimorando o modelo, o conteúdo da Figura 10 exibe a ordenação da maior para a menor relevância no aprendizado, além da representação gráfica desses resultados, que permite a identificação simplificada de possíveis mudanças no algoritmo.

```
[ ] # Relevância das colunas
    features = pd.Series(classificador.feature_importances_, index=X_train.columns).sort_values(ascending=False)
```

```
features
```

```
oldbalanceOrg     0.433985
newbalanceOrig    0.407894
amount            0.124309
newbalanceDest    0.020893
type              0.008443
oldbalanceDest    0.004475
dtype: float64
```

```
[ ] features.plot.barh()
    plt.ylabel('Features')
    plt.xlabel('Importância')
    plt.show();
```

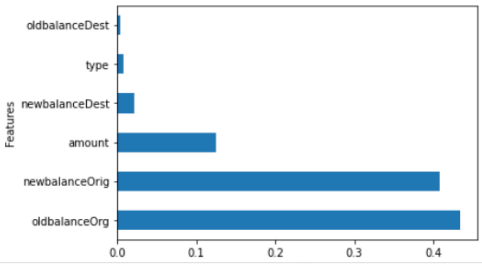

Figura 10: Exibição gráfica da relevância das colunas

8 Conclusão

Assim, foi concluído que a área de Machine Learning é um campo que dispõe de diversas ferramentas que possibilitam a implementação de modelos de forma simplificada com apoio bibliotecas, fóruns e documentações. Dessa forma, foi possível desenvolver um algoritmo para identificação de possíveis transações financeiras fraudulentas, com uma chance de erro, próxima de um valor nulo, o projeto obteve 99% de acurácia em todas as métricas, como *precision*, *recall* e *F1-Score*, ou seja, pra cada 100 avaliações 99 foram assertivas.

Referências

AGÊNCIA BRASIL. Fraudes contra clientes de bancos crescem 165% em 2021. 2022. Disponível em < https://agenciabrasil.ebc.com.br/economia/noticia/2021-10/fraudes-contra-clientes-de-bancos-crescem-165-em-2021> Acesso em 13 de setembro 2022.

ALECRIM, EMERSON. Machine learning: o que é e por que é tão importante. 2018.Disponível em: <https://tecnoblog.net/responde/machine-learning-ia-o-que-e/> Acesso em 12 de setembro de 2022.

CARVALHO, J. Índice da Kroll aponta que 89% das empresas brasileiras já sofreram fraude cibernética. [S.l.], 2018. Disponível em: <https://ipnews.com.br/indice-da-kroll-aponta-que-89-das-empresas-brasileiras-ja-sofreram-fraude-cibernetica/>. Acesso em 11 de setembro de 2022

DAMASCENO, LAURA. REGRESSÃO LIENAR. 2020. Disponível em: <https://medium.com/@lauradamaceno/regress%C3%A3o-linear-6a7f247c3e29>Acesso em 10 de setembro de 2022.

DATA SCIENCE ACADEMY. Deep Learning Book. 2022. Disponível em: <https://www.deeplearningbook.com.br/o-que-e-aprendizagem-por-reforco/> Acesso em 09 de setembro de 2022.

EQUALS. Saiba quais são os principais tipos de transações financeiras. 2018. Disponível em: <https://www.equals.com.br/blog/saiba-quais-sao-os-principais-tipos-de-transacoes-financeiras/> Acesso em 12 de setembro de 2022.

GOMES, Pedro César Tebaldi. Regressão Linear: entenda como utilizar. 2019.Disponível em: <https://www.datageeks.com.br/regressao-linear/> Acesso em 10 de setembro de 2022.

GUSTAVO, JOÃO. As vantagens de aprender Python. 2021. Disponível em: <https://medium.com/data-hackers/quais-s%C3%A3o-as-vantagens-de-aprender-python-dd3fbba02c3c#:~:text=Uma%20das%20maiores%20vantagens%20da,ensino%20de%20programa%C3%A7%C3%A3o%20em%20Python>. Acesso em 10 de setembro de 2022.

HARVE. Pandas Python: vantagens e como começar. 2022. Disponível em: <https://harve.com.br/blog/programacao-python-blog/pandas-python-vantagens-e-como-comecar/#:~:text=o%20Pandas%20Python%3F-,Pandas%20%C3%A9%20uma%20biblioteca%20para%20uso%20em%20Python%2C%20open%2Dsource,an%C3%A1lise%20e%20manipula%C3%A7%C3%A3o%20de%20dados.> Acesso em 10 de setembro de 2022.

HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP. Machine Learning. 2022. Disponível em: <https://www.hpe.com/br/pt/what-is/machine-learning.html> Acesso em 11 de setembro de 2022.

LOPES, ALEXANDRE. Aprendizado de máquina. 2020. Disponível em: <https://aprendizadodemaquina.com/artigos/pyspark-entenda-a-engine-do-spark-para-rodar-python/>Acesso em 10 de setembro de 2022.

MARYALENE, L. How Banks Are Working to Protect You From Fraud. [S.l.], 2017. 1 p. Disponível em: <https://money.usnews.com/banking/articles/how-banks-are-working-to-protect-you-from-fraud>. Acesso em 15 de setembro de 2022.

MATTOS, LUCAS. Aplicação de técnicas de machine learning no apoio à detecção de fraudes em pagamentos online. 2021. Disponível em: <https://repositorio.ifsc.edu.br/bitstream/handle/123456789/2502/IFSC_TCC_II_Lucas_Mattos.pdf?sequence=1&isAllowed=y>. Acesso em 15 de setembro de 2022.

ORACLE. O que é Machine Learning?. 2022. Disponível em: <https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/>Acesso em 10 de setembro de 2022.

REDAÇÃO IUGU. O que são transações financeiras e quais os principais meios de pagamento?. 2022. Disponível em: <https://www.iugu.com/blog/transacoes-financeiras#:~:text=Transa%C3%A7%C3%B5es%20financeiras%20s%C3%A3o%20a%20troca,os%20tipos%20e%20como%20funcionam.&text=O%20volume%20de%20transa%C3%A7%C3%B5es%20financeiras,meios%20de%20pagamento%20mais%20consolidados.> Acesso em 10 de setembro de 2022.

RODRIGUES, VITOR. Métricas de Avaliação: acurácia, precisão, recall quais as diferenças?. 2019. Disponível em: <https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c#:~:text=Precis%C3%A3o%3A%20dentre%20todas%20as%20classifica%C3%A7%C3%B5es,harm%C3%B4nica%20entre%20precis%C3%A3o%20e%20recall>. Acesso em 29 de novembro de 2022

SOUZA, TÚLIO CÉSAR LEITE. Estudo de algoritmos de machine learning para predição de fraudes em cartões de crédito. 2021. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/1463> Acesso em 10 de setembro de 2022.

SUPREMO CONCURSOS. Entenda o que configura uma fraude bancária. 2022. Disponível em: <https://blog.supremotv.com.br/entenda-o-que-configura-uma-fraude-bancaria/> Acesso em 11 de setembro de 2022.

VADER, PATTY. Programação em Python e PySpark. 2021.Disponível em: <https://medium.com/team-data-stone/programa%C3%A7%C3%A3o-em-python-e-pyspark-681f4c405850>Acesso em 12 de setembro de 2022.