Prevenção de fraudes com cartões de crédito utilizando o classificador ingênuo de Bayes.

1st Gabriel Pierre Carvalho Coelho
Centro de informática
Universidade Federal de Pernambuco
Recife, Brasil
gpcc@cin.ufpe.br

2nd Klarissa Andrade de Morais
Centro de informática
Universidade Federal de Pernambuco
Recife, Brasil
kam@cin.ufpe.br

3rd Luan Eustáquio Lopes de Farias
Centro de informática
Universidade Federal de Pernambuco
Recife, Brasil
lelf@cin.ufpe.br

***Abstract* — Este trabalho busca através do estudo de uma base de dados, prever se uma transação com cartão de crédito é fraudulenta ou não. A proposta consiste em procurar e utilizar uma base de dados contendo o histórico de transações com cartões de crédito, integrando exemplos positivos e negativos(i.e., fraudulentas e legítimas), e utilizar esses dados para treinar o modelo. O algoritmo Ingênuo de Bayes será utilizado para classificar novas transações com base nas características das transações fraudulentas e legítimas já registrados na base de dados.**

Palavras-chave* — *Detecção de fraudes, transações com cartão de crédito, Classificador Probabilístico, Teorema de Bayes, Python.

I. INTRODUÇÃO

Nos últimos anos, o uso de cartões de crédito em transações financeiras tem apresentado um crescimento significativo, impulsionado pelo aumento da popularidade do comércio eletrônico e das transações sem contato físico. No entanto, esse aumento também tem gerado um grande desafio para as empresas e instituições financeiras: a prevenção de fraudes em transações de cartões de crédito. As fraudes nesse tipo de transação representam um problema crescente, causando prejuízos financeiros significativos para as empresas e consumidores, além de abalar a confiança no uso desse meio de pagamento.

Nesse cenário, o uso de técnicas de aprendizado de máquina tem se apresentado como uma solução eficiente para a prevenção de fraudes em transações de cartões de crédito. Diante disso, este trabalho propõe uma solução baseada em algoritmos de aprendizado de máquina, utilizando o algoritmo Ingênuo de Bayes da biblioteca Scikit-Learn, além de testes com algumas variações do algoritmo da mesma biblioteca, a fim de definir o que se encaixa melhor no modelo. A pretensão é construir um modelo capaz de analisar transações de cartões de crédito e classificá-las como fraudulentas ou legítimas com base em uma análise probabilística.

II. OBJETIVO

O objetivo deste trabalho é propor uma solução para prevenção de fraudes em transações de cartões de crédito com base em aprendizado de máquina, em particular, o algoritmo ingênuo de Bayes. A ideia é construir modelos capazes de analisar transações de cartões de crédito e classificá-las como fraudulentas ou não com base em uma análise probabilística. Para isso, serão utilizadas técnicas de pré-processamento de dados, como normalização e seleção de características, e os algoritmos serão treinados e avaliados em conjuntos de dados públicos de transações de cartões de crédito. Além disso, o trabalho propõe uma comparação entre o desempenho dos algoritmos desenvolvidos, a fim de verificar qual deles apresenta melhores resultados nos cenários avaliados.

III. JUSTIFICATIVA

A fraude em transações com cartões de crédito é um problema crescente que causa prejuízos financeiros significativos para empresas e consumidores, além de abalar a confiança no uso desse meio de pagamento. A utilização de modelos preditivos para detecção de fraudes em transações com cartão de crédito é uma abordagem promissora para prevenir e reduzir o impacto dessas fraudes.

Diversos trabalhos na área de aprendizado de máquina têm sido realizados com o objetivo de detectar operações suspeitas, como o trabalho de Ali et al. [1], que propõe uma nova abordagem utilizando redes neurais convolucionais.

Um exemplo recente que destaca a importância da prevenção de fraudes em transações com cartões de crédito é o caso da Netshoes. A empresa foi condenada a pagar uma multa de R$ 500 mil por um vazamento de dados de mais de 2 milhões de clientes [2]. Esse tipo de violação de dados pode levar a prejuízos financeiros e de reputação para a empresa, além de colocar em risco a privacidade e segurança dos clientes. O desenvolvimento de um modelo preditivo eficaz pode contribuir para a melhoria da segurança dos sistemas de pagamento eletrônico e prevenção de perdas financeiras. Portanto, este projeto justifica-se pela importância da detecção de fraudes em transações com cartão de crédito e pela necessidade de desenvolver modelos preditivos eficazes para prevenir e reduzir o impacto dessas fraudes.

IV. CLASSIFICADOR INGÊNUO DE BAYES

Para prosseguirmos na análise dos dados, será empregado o algoritmo classificador de Bayes, o qual foi desenvolvido pelo matemático inglês Thomas Bayes de mesmo nome.

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} : \quad \text{Bayes theorem,}$$

Figura 1. Teorema de Bayes.

O método se baseia na independência entre os fatores, por isso é conhecido como "ingênuo" e descreve com boa precisão a classificação dos mesmos , essa precisão, e poder empregar um número menor de dados e mesmo assim produzir um resultado preciso é o motivo a qual ele é empregado em machine learning e por consequência o utilizaremos no projeto.

Na equação da figura 1, P(x) e P(y) representam a probabilidade de um evento "x" e outro "y" ocorrerem, enquanto a notação P(y|x) simboliza a probabilidade do evento "y" ocorrer após ter ocorrido um evento x.

V. METODOLOGIA

A. Dataset

Coleta de dados: Para este projeto, será utilizado o conjunto de dados *Credit Card Fraud Detection*, disponibilizado publicamente, que contém informações de transações de cartões de crédito realizadas por titulares europeus em setembro de 2013. O conjunto de dados apresenta transações que ocorreram em dois dias, totalizando 284.807 transações, dentre as quais 492 foram rotuladas como fraudulentas e as demais como legítimas. A base de dados é altamente desbalanceada, com a classe positiva (fraudes) representando apenas 0,172% do total de transações. O conjunto de dados contém apenas variáveis numéricas resultantes de uma transformação PCA (Principal Component Analysis). As variáveis V1 a V28 são as componentes principais obtidas com o PCA, enquanto as variáveis *"Time"* e *"Amount"* não foram transformadas. A variável *"Time"* representa o tempo em segundos entre a transação atual e a primeira transação registrada no conjunto de dados, enquanto *"Amount"* representa o valor da transação. A variável *"Class"* é a variável de resposta e assume o valor 1 em caso de fraude e 0 em caso contrário.

Pré-processamento de dados: Para lidar com as particularidades do conjunto de dados *"Credit Card Fraud Detection"*, altamente desbalanceado, serão utilizadas técnicas de amostragem, como subamostragem ou sobreamostragem, para equilibrar as classes e evitar a dominância da classe negativa no conjunto de dados. O objetivo é selecionar as características mais relevantes e tratar dados ausentes ou duplicados de forma adequada, avaliando o impacto das técnicas de pré-processamento no desempenho dos modelos de detecção de fraudes.

Divisão dos dados: Os dados serão divididos em conjuntos de treinamento, e teste, em uma proporção de 80% e 20%, respectivamente.

B. Tecnologias e desenvolvimento

Ferramentas utilizadas: Para este projeto, utilizaremos da linguagem de programação *Python*, que tem se mostrado útil para trabalhos envolvendo computação científica. O passo a passo dos cálculos do classificador, bem como o das probabilidades serão implementados através de recursos da linguagem e de suas bibliotecas, como a *Numpy, Pandas, Scikit Learn e Matplotlib*. Sendo a última a fornecedora de artifícios para a exibição desses dados através de conjuntos de métodos para a geração dos resultados graficamente. Essas bibliotecas nos fornecerão funções para tratamento de números, manipulação

dos dados e aprendizagem de máquina.

Treinamento dos modelos: Serão treinados modelos de detecção, utilizando a variação do algoritmo ingênuo de Bayes da biblioteca Scikit-Learn que melhor se encaixe com o nosso conjunto de dados.

Ambiente de desenvolvimento: O projeto será desenvolvido no ambiente do Google Collaboratory, onde realizaremos a escrita do código e análise dos resultados obtidos.

C. Avaliação dos modelos e análises dos resultados

Avaliação dos modelos: Com base no conjunto de dados altamente desequilibrado, avaliaremos o desempenho dos modelos de detecção de fraudes utilizando as métricas da biblioteca scikit learn, observando principalmente sua *precisão, recall e F1-scor*e. É importante também avaliar a relevância e a importância das características utilizadas no modelo, considerando que as características originais das transações não estão disponíveis devido a questões de confidencialidade.

Comparação de desempenho: Será realizada uma comparação de desempenho entre o algoritmo de predição aplicado em diferentes abordagens do conjunto de dados, a fim de verificar em qual delas o algoritmo apresenta melhores resultados na detecção de fraudes em transações de cartões de crédito.

D. Escolha do algoritmo de Naive Bayes

MultinomialNB: Essa variação é adequada para dados discretos e é comumente usada em problemas de classificação de texto. Como o nosso conjunto de dados contém características contínuas, o MultinomialNB provavelmente não é a melhor opção.

BernoulliNB: Essa variação é adequada para dados binários. No entanto, nosso conjunto de dados tem características contínuas, então o BernoulliNB também não seria a melhor escolha.

GaussianNB: Essa variação assume que os atributos seguem uma distribuição normal (gaussiana). Como as características do nosso conjunto de dados são contínuas e parecem ter uma distribuição próxima à normal após o pré-processamento, o GaussianNB é uma opção adequada. Logo, optaremos por usar este algoritmo em nosso experimentos.

VI. ANALÁLISE EXPLORATÓRIA DOS DADOS

Antes de utilizar os dados para a construcão do Classificador Ingenuo de Bayes, iremos realizar uma Análise Exploratória dos Dados. Para isso iremos utilizar algumas bibliotecas do Python e seguiremos os seguintes passos:

1) **Entendendo a base de dados:** primeiramente, será necessário entender um pouco mais sobre as variáveis presentes no banco de dados e como elas estão dispostas.

   a) **Visualização dos dados:** Dataset é formado pelas colunas time, em seguidas as 28 colunas V1-V28 que passaram por uma transformação PCA, a coluna Amount, e nossa variável de interesse Class, que indica se a transação é fraudulenta (1) ou não fraudulenta (0).

	Time	V1	V2	V3	V4	V5	V6	V7
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941

	V8	V9	...	V21	V22	V23	V24	V25
0	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539
1	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170
2	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642
3	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376
4	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010

	V26	V27	V28	Amount	Class
0	-0.189115	0.133558	-0.021053	149.62	0
1	0.125895	-0.008983	0.014724	2.69	0
2	-0.139097	-0.055353	-0.059752	378.66	0
3	-0.221929	0.062723	0.061458	123.50	0
4	0.502292	0.219422	0.215153	69.99	0

Figura 2. Visualização das classes.

b) **Tipo dos dados:**

Time	float64
V1-V28	float64
Amount	float64
Class	int64

c) **Presença de valores ausentes:** o dataset não possui dados ausentes, o que é um ponto positivo, pois não será necessário fazer procedimentos de preenchimento de valores ausentes

Figura 3. Valores ausentes.

d) **Balanceamento das classes:** Com o gráfico abaixo percebemos que trata-se de um dataset muito desbalanceado

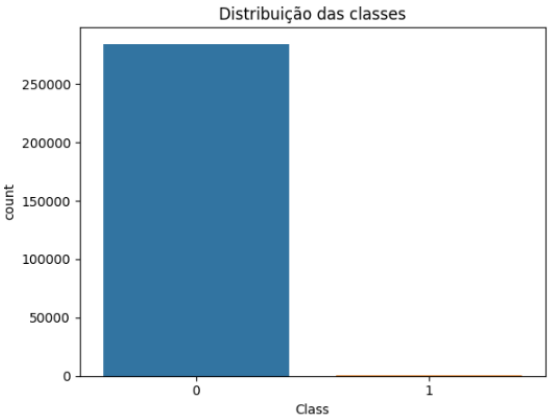


Figura 4. Distribuição das classes.

2) **Analisando as colunas do dataset:** Como a maioria dos recursos estão no formato PCA, exceto as colunas Time e Amount, examinaremos mais profundamente essas duas colunas

a) **Time:** Olhando o gráfico percebemos que em dois momentos existe um intervalo de tempo em que são realizadas poucas transações. Como o dataset possui transações em um intervalo de tempo de dois dias, podemos inferir que esses momentos com baixa frequência de transações seriam no horário da madrugada

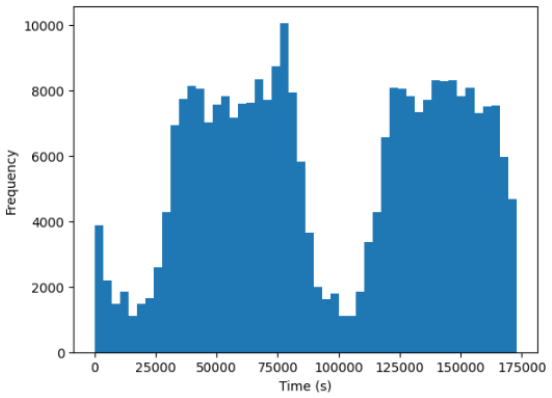


Figura 5. Análise da classe Time.

b) **Amount:** No gráfico abaixo, percebemos que a grande maioria das transações possuem um valor baixo, porém há algumas transações isoladas (outliers) com valores muito alto

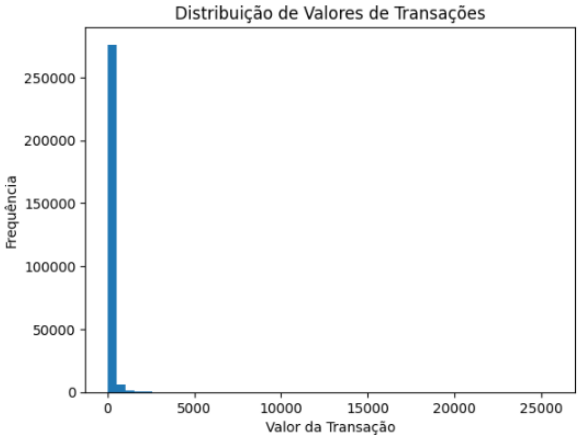
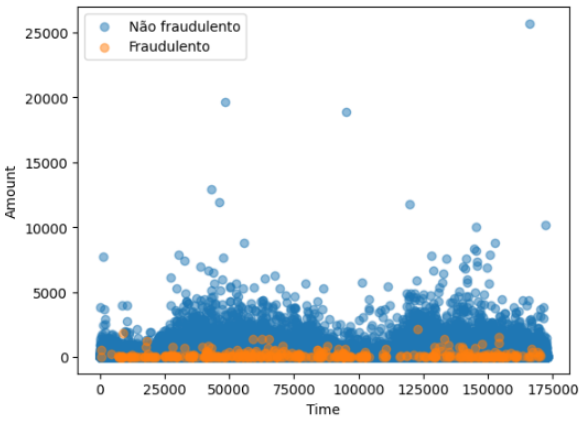


Figura 6. Análise da classe Amount.

c) **Time x Amount de cada classe:** Olhando o gráfico, percebemos que o valor das transações fraudulentas são quase em totalidade pequenos, o que faz sentido dado que existe uma maior cautela dos bancos para transações com valores altos



3) **Correlação entre as colunas:** Olhando a matriz de correlações, podemos perceber que as colunas V2, V4 e V11 são as que parece ter maior correlação com a nossa variável de interesse Class.

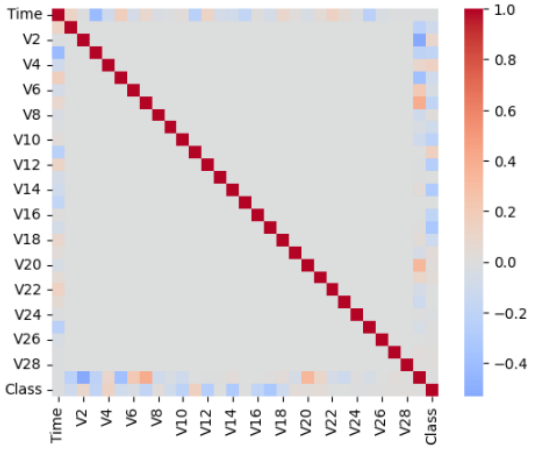


Figura 8. Matriz de correlação.

VII. EXPERIMENTOS E RESULTADOS

Neste trabalho, foram conduzidos quatro experimentos com o intuito de investigar os principais parâmetros e, com base nisso, realizar uma análise para identificar a abordagem mais eficaz. Além da avaliação da acurácia, foram consideradas métricas adicionais, como recall, que mede a proporção de acertos em relação aos casos positivos reais, precisão, que avalia a proporção de acertos em relação aos casos positivos previstos, e o escore F1, que é uma média harmônica entre recall e precisão. Essas métricas forneceram uma avaliação mais completa do desempenho dos experimentos, permitindo uma análise abrangente dos resultados obtidos.

Experimento 0:

Inicialmente, após a divisão dos dados, que foi comum a todos os experimentos, foi realizado o pré-processamento dos dados. Nessa etapa, verificou-se que o conjunto de dados não continha valores ausentes, portanto não foi necessário aplicar tratamento para esse problema. No entanto, foi realizada a normalização Min-Max na coluna "Amount", com o objetivo de melhorar o desempenho da otimização. Essa normalização mapeou os valores dessa coluna para uma faixa entre 0 e 1, visando obter uma escala adequada para análise. Além disso, é importante ressaltar que não foi realizado um tratamento para o desbalanceamento da classe de indicação de fraude ou não. Essa questão pode impactar na precisão dos resultados obtidos, uma vez que a classe minoritária pode ter menos representatividade e influenciar a capacidade do modelo de generalizar corretamente para ambas as classes. Os dados obtidos foram: 0.15 de Precisão, 0.63 de Recall, 0.24 de F1-score e Acurácia de 0.99

Experimento 1:

Durante o experimento, foram avaliadas duas técnicas para o balanceamento dos dados: SMOTE e RandomUnderSample. O SMOTE gera dados sintéticos na classe minoritária para equilibrar as classes, enquanto o RandomUnderSample realiza uma subamostragem aleatória da classe majoritária para igualar o número de exemplos entre as classes. No entanto, observou-se que o desempenho do RandomUnderSample foi inferior em comparação com o SMOTE. Portanto, optou-se por utilizar este último, que gerou melhores resultados na tarefa de balanceamento das classes indicativas de fraudes. Essa escolha foi baseada na análise comparativa do desempenho das duas técnicas durante o experimento. Então houve o rebalanceamento e equilíbrio do número de exemplos entre classes. Obtendo o seguinte resultado: 0.13 de Precisão, 0.80 de Recall, 0.23 de F1-score e Acurácia de 0.99

Experimento 2:

Realizamos o mesmo pré-processamento aplicado no experimento 1, onde não foram identificados valores ausentes no conjunto de dados e foi realizada a normalização Min-Max na coluna "Amount". Além disso, utilizamos a técnica de validação Kfold para treinar e avaliar o modelo.

A validação cruzada K-fold é uma técnica que divide os dados em 'k' partes iguais, e o modelo é treinado 'k' vezes. Em cada iteração, uma das partes é usada como conjunto de validação e as demais 'k-1' partes são usadas como conjunto de treinamento. Isso evita que o modelo seja avaliado em dados que já foram vistos durante o treinamento e fornece uma estimativa mais confiável do desempenho do modelo em dados não vistos.

Uma diferença significativa do experimento 2 em relação ao experimento 1 foi a adição do hiperparâmetro "var smoothing" na etapa de treinamento do modelo. Esse hiperparâmetro é responsável por controlar a suavização dos dados, evitando o problema de sobreajuste.

Após o treinamento e avaliação do modelo utilizando os hiperparâmetros encontrados no Kfold, observamos uma melhoria em relação ao experimento 1. A inclusão do hiperparâmetro "var smoothing" permitiu que o modelo obtivesse uma melhor generalização dos dados, evitando que ele se ajustasse demasiadamente aos dados de treinamento e não conseguisse generalizar para novos dados. Além disso, a utilização da técnica de validação Kfold possibilitou uma avaliação mais precisa do modelo, uma vez que os dados foram divididos em k partes para treinamento e validação, evitando que o modelo fosse avaliado em dados que ele já havia visto durante o treinamento. Assim, tivemos0’ os seguintes dados:

0.05 de Precisão, 0.88 de Recall, 0.10 de F1-score e Acurácia de 0.97.

Experimento 3:

Utilizamos o StandardScaler como uma técnica de pré-processamento dos dados. O StandardScaler é uma técnica de normalização que é aplicada aos recursos (ou variáveis) do conjunto de dados, com o objetivo de melhorar o desempenho do modelo de detecção de fraudes em transações financeiras.

O StandardScaler é uma técnica comum de normalização que transforma os dados de forma que eles tenham média zero e desvio padrão igual a um. Isso é feito calculando a média e o desvio padrão de cada recurso e em seguida, subtraindo a média e dividindo pelo desvio padrão. Essa transformação coloca todos os recursos na mesma escala, facilitando a comparação entre eles e melhorando o desempenho do modelo durante o treinamento.

A aplicação do StandardScaler permitiu que os recursos do conjunto de dados fossem normalizados para uma escala padrão, o que pode ter contribuído para o melhor desempenho do modelo de detecção de fraudes em transações financeiras, possibilitando uma aprendizagem mais equilibrada a partir dos diferentes recursos disponíveis. Os resultados foram: 0.97 de Precisão, 0.87 de Recall, 0.92 de F1-score e Acurácia de 0.91

VIII. COMPARAÇÃO E CONCLUSÃO

- **Comparação dos experimentos:**

	Exp1	Exp2	Exp3
Precisão	0.13	0.05	0.97
Recall	0.80	0.88	0.87
F1-score	0.23	0.10	0.92
Acurácia	0.99	0.97	0.91

- **Conclusão:**

Melhor métrica: F1-score.

O f1-score é uma métrica que considera tanto a precisão quanto o recall, sendo adequada para avaliar a performance de um modelo em um conjunto de dados desbalanceado onde a classe minoritária é importante, como é o nosso caso. Além disso, como o objetivo da detecção de fraude em transações de cartão de crédito é identificar corretamente o maior número possível de transações fraudulentas, sem aumentar muito o número de falsos positivos, o f1-score é a métrica ideal para avaliar a qualidade do modelo.

Melhor experimento: Experimento 3.

Como foi visto, o dataset possui um desequilíbrio muito grande em relação a nossa variável de interesse. Lidar com essa característica do dataset foi crucial para os algoritmos de predição. No entanto, no experimento 3, ao realizarmos o balanceamento do conjunto de dados, conseguimos mitigar o problema do desequilíbrio, permitindo que o algoritmo de predição alcançasse uma classificação de qualidade satisfatória.

IX. REFERÊNCIAS

[1] ALI, B. et al. Detecção de fraude em cartão de crédito: uma nova abordagem usando redes neurais convolucionais. IEEE Access, v. 6, p. 36647-36656, 2018. Disponível em: https://ieeexplore.ieee.org/abstract/document/8397371

[2] G1. Netshoes terá de pagar R$ 500 mil por vazamento de dados de 2 milhões de clientes. Disponível em: https://g1.globo.com/df/distrito-federal/noticia/2019/02/05/netshoes-tera-de-pagar-r-500-mil-por-vazamento-de-dados-de-2-milhoes-de-clientes.ghtml

[3] ARAGAW, S. et al. Detecção de fraudes em cartões de crédito usando redes neurais artificiais: uma revisão sistemática. IEEE Access, v. 7, p. 10557-10567, 2019. Disponível em: https://ieeexplore.ieee.org/abstract/document/8730573

[4] PEDREGOSA, F. et al. Scikit-learn: machine learning in Python. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011. Disponível em: https://jmlr.org/papers/v12/pedregosa11a.html

[5] GÉRON, A. Mãos à obra: aprendizado de máquina com Scikit-Learn e TensorFlow: conceitos, ferramentas e técnicas para construir sistemas inteligentes. Rio de Janeiro: Alta Books, 2018. Disponível em: https://www.casadocodigo.com.br/products/livro-machine-learning-python