EQUILIBRANDO CLASSES

TÉCNICAS PARA TRATAR DADOS DESBALANCEADOS EM MACHINE LEARNING

Tipo de Atividade: Projeto Prático;

Área de Estudo: Ciência de Dados + Aprendizado Supervisionado;

Membros (Grupo 1): Caique Sidrão Siqueira Barbosa, Rebecka Bocci e Gabriel Lima.

SUMÁRIO

1. INTRODUÇÃO

No mundo da ciência de dados, os profissionais trabalham com diversas bases de dados e, com elas, desenvolvem previsões usando modelos de machine learning. Para obter a melhor eficiência do modelo, os cientistas buscam bases de dados balanceadas, ou seja, aquelas em que as categorias apresentam frequências muito próximas entre si — por exemplo, um dataset que possui uma quantidade de dados fraudulentos próxima da quantidade de dados confiáveis.

Todavia, há datasets que possuem dados desbalanceados — quando uma categoria apresenta uma quantidade muito superior à outra — o que prejudica a análise e o treinamento do modelo, resultando em métricas pouco satisfatórias.

Este trabalho tem como objetivo estudar e aplicar métodos de balanceamento de datasets e compará-los com sua versão desbalanceada.

2. OBJETIVO

O objetivo principal deste trabalho é estudar o que é uma base de dados desbalanceada e os métodos utilizados para balanceá-la, com o intuito de melhorar o desempenho dos modelos de *machine learning* e fornecer previsões mais precisas.

Ao final, serão comparados os diversos métodos entre si e também em relação à base desbalanceada, discutindo-se seus resultados.

3. PESQUISA

3.1. O que são dados desbalanceados?

No mundo dos dados, os analistas estão sempre trabalhando com bases de dados (ou *datasets*), que podem conter inúmeras linhas e colunas.

Um *dataset* desbalanceado é uma base em que as categorias apresentam frequências muito díspares. Por exemplo, um conjunto de dados destinado a classificar pessoas com problemas cardíacos pode conter muito mais registros de indivíduos sem nenhuma dificuldade cardíaca do que daqueles que possuem a condição.

Esse desbalanceamento pode prejudicar o desempenho de modelos de classificação supervisionados, resultando em previsões equivocadas.

Para evitar esses problemas, existem técnicas que auxiliam no balanceamento da base de dados.

3.2. Implicações em usar base de dados desbalanceados

Em uma base de dados desbalanceada, as categorias-alvo apresentam grande discrepância entre si; isso significa que existem muito mais dados para determinados cenários do que para outros. Esse problema impacta especialmente modelos de classificação, como *K-nearest Neighbours* (K-vizinhos mais próximos, KNN) e *Logistic Regression* (Regressão Logística).

Modelos de *machine learning* treinados em dados altamente desbalanceados tendem a ficar enviesados em favor da classe majoritária. Eles se tornam muito eficientes em prever a classe que aparece com maior frequência, mas falham em identificar corretamente as instâncias da classe minoritária. Como consequência, o modelo pode apresentar alta acurácia (quando essa é a métrica principal), mas um baixo *recall* para a classe minoritária — que, em muitos casos, é justamente a classe de maior interesse, como em detecção de fraudes, diagnósticos de doenças raras, entre outros.

3.2.1 Accuracy (Acurácia)

A acurácia é uma métrica que avalia a proporção de acertos feitos pelo modelo, sendo frequentemente utilizada em problemas binários (como *positivo/negativo*, *normal/fraudulento*, *0/1*, entre outros). Seu cálculo é dado por:

$$Acurácia = \frac{Verdadeiros\ Positivos\ (VP/TP) + Verdadeiros\ Negativos\ (VN/TN)}{Total\ dos\ dados}$$

Verdadeiros positivos = VP (True Positives = TP); Verdadeiros Negativos = VN (True Negative = TN)

O modelo pode apresentar uma acurácia alta, mas ainda assim depender excessivamente da classe majoritária (aquela com maior frequência). Isso ocorre porque o modelo acerta grande parte das instâncias da classe majoritária, porém erra com frequência as classificações da classe minoritária — e esses erros têm pouco impacto no valor final da acurácia.

Por isso, em bases de dados desbalanceadas, é fundamental utilizar outras métricas para avaliar o desempenho do modelo, como *Recall* e *Precisão*, que fornecem uma visão mais completa sobre sua capacidade de identificar corretamente a classe de interesse.

3.2.2. Precision (Precisão)

A proporção de verdadeiros positivos entre todas as previsões positivas. Responde: "Das vezes que o modelo previu fraude, quantas foram realmente fraudes?"

$$Precisão = \frac{Verdadeiro\ Positivo\ (VP)}{Verdadeiro\ Positivo\ (VP) + Falso\ Positivos\ (FP)}$$

3.2.3. Recall (Sensitivity ou True Positive Rate)

A proporção de verdadeiros positivos entre todas as instâncias reais da classe positiva. Responde: "Das fraudes reais, quantas o modelo conseguiu detectar?".

$$Recall = \frac{Verdadeiros\ positivos\ (VP)}{Verdadeiros\ positivos\ (VP) + Falso\ Negativos\ (FN)}$$

3.2.4. F1-Score

F1-Score é uma métrica que busca encontrar o meio termo entre recall e precisão, sendo dependente das duas.

$$F1 = \frac{2*precisão * recall}{precisão + recall}$$

3.2.5. AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

Mede a capacidade do modelo em distinguir entre as classes. Um valor mais alto indica melhor discriminação. O AUC-ROC foi útil para comparar a capacidade geral de cada modelo.

3.2.6. Curva Precisão-Recall

Útil para avaliar modelos em bases de dados desbalanceadas, essa métrica mostra como a Precisão e o Recall variam à medida que alteramos o *threshold* de classificação do modelo. Em outras palavras, avalia como acontece a troca de recall por precisão, vice-versa.

3.2.7. Matriz de confusão

Uma tabela que resume o número de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, fornecendo uma visão detalhada do desempenho do modelo em cada classe.

3.2.8. Como comunicar resultados quando a acurácia não conta a história toda?

Quando a acurácia não é suficiente (em casos de desbalanceamento), é crucial comunicar os resultados utilizando as métricas mais adequadas mencionadas acima. Ao apresentar os resultados, é importante:

- Explicar o problema do desbalanceamento: Deixar claro por que a acurácia isoladamente é enganosa e por que outras métricas são necessárias.
- Apresentar a matriz de confusão: A matriz de confusão é uma ferramenta visual poderosa que mostra claramente onde o modelo está acertando e errando em relação a cada classe.
- Focar em Precisão, Recall e F1-Score (ou outras métricas relevantes): Dependendo do objetivo do negócio (minimizar falsos positivos ou falsos negativos), enfatizar as métricas mais importantes. Por exemplo, em detecção de fraude, o recall (não perder fraudes) pode ser mais importante, enquanto em diagnósticos médicos, a precisão (não dar falsos positivos) pode ser crucial.
- Usar a curva ROC e a AUC: A curva ROC e a AUC são ótimas para comparar a performance geral de diferentes modelos ou técnicas de balanceamento.
- Contextualizar os resultados: Relacionar as métricas de volta ao problema de negócio. Por exemplo, explicar o que significa ter um recall de 80% em detecção de fraude (80% das fraudes reais são pegas) e o custo de um falso positivo (investigar uma transação legítima).

Em resumo, a comunicação deve ir além de um único número (acurácia) e fornecer uma imagem completa do desempenho do modelo em relação às classes minoritária e majoritária, utilizando as métricas e visualizações apropriadas.

3.3. Métodos de balanceamento

Abaixo estão listados os métodos de balanceamento mais utilizados:

3.3.1. Undersampling (Subamostragem)

Técnicas que diminuem a frequência dos dados da classe majoritária, ou seja, reduzem o volume do conjunto maior para aproximá-lo do tamanho da classe minoritária.

Isso acaba diminuindo a quantidade total de dados e pode acelerar o treinamento dos modelos de machine learning.

Entre as principais técnicas de *undersampling*, têm-se:

3.3.1.1. Random undersampling

Basicamente a exclusão aleatória das classes majoritárias. Os cientistas de dados podem optar por usar algum raciocínio para excluir ou permanecer certos dados.

3.3.1.2. Three near-miss undersampling techniques

São três técnicas que compõem esse conjunto: na primeira técnica, eles mantêm eventos da classe majoritária que têm a menor distância média para os três eventos mais próximos da classe minoritária em um gráfico de dispersão; no segundo, eles usam eventos da classe majoritária que têm a menor distância média para os três eventos mais distantes da classe minoritária em um gráfico de dispersão; no terceiro, eles mantêm um determinado número de eventos da classe majoritária para cada evento da classe minoritária que estão mais próximos no gráfico de dispersão [12, VIADINUGROHO, 2021].

3.3.1.3. Condensed Nearest Neighbors (CNN) undersampling

A técnica acaba criando um subconjunto e nele é alocado às duas classes: todas as classe minoritárias e as majoritárias que não podem ser classificadas corretamente. O restante é descartado.

3.3.1.4. Tomek Link Method

Esse método tem duas diferenças em relação ao CNN: primeiro, encontra dois eventos, um majoritário e outro minoritário, que estejam com a menor distância entre si em um gráfico de dispersão. Isso são os Tomek Links, ou eventos na fronteiras ou ruídos [12, VIADINUGROHO, 2021]. Normalmente, não elimina grande quantidade da classe majoritária, por isso é geralmente usado com outras técnicas.

3.3.1.5. Edited Nearest Neighbors (ENN)

Com base nos três vizinhos mais próximos, o evento que é da classe majoritária e classificado de maneira errada, baseado nos três vizinhos mais próximos, é removido da classe majoritária.

3.3.1.6. One-sided selection

Combina Tomek Link com CNN, em que o primeiro remove os ruídos nos dados dos eventos de classe majoritários e o segundo os redundantes.

3.3.1.7. Neighborhood cleaning rule

Combina CNN e ENN. O primeiro remove eventos duplicados e o segundo remove eventos ruidosos e ambíguos.

3.3.2. Oversampling (Sobreamostragem)

Técnica usada para aumentar a quantidade de registro de menor frequência. Ele tem a vantagem de ter mais informações do alvo, mas acaba por aumentar a quantidade de dados, o tempo de execução, muitos dados podem estar duplicados e pode resultar em um overfitting. A quantidade de dados de menor frequência é igualada com a de maior frequência.

Dentre as técnicas de oversampling, há:

3.3.2.1. Random Oversampling

Este método seleciona aleatoriamente classes minoritárias e as duplica, aumentando a frequência dos dados minoritários até equivaler com os majoritários.

3.3.2.2. SMOTE (Synthetic Minority Oversampling Technique)

Esta técnica cria dados sintéticos dos eventos da classe minoritária até ter a mesma quantidade da classe majoritária. Seus cálculos se baseiam na linearidade do KNN.

3.3.2.3. ADASYN (Adaptive Synthetic Sampling)

Uma expansão do SMOTE, em que considera a densidade da variável, focando em regiões com menor quantidade de dados minoritários.

3.3.3. Método híbrido

Métodos híbridos são aqueles que juntam dois métodos - um da sobreamostragem e outro da subamostragem - com o objetivo de evitar as fraquezas que esses métodos têm quando isolados.

Os métodos de oversampling dos minoritários normalmente só duplicam os dados, não adicionando nenhuma informação, e os métodos de undersampling da

classe majoritária remove certa quantidade de dados, perdendo os dados que possam ser importantes, os métodos híbridos tentam evitar esses cenários. Normalmente, eles acabam usando ou o SMOTE ou o ADASYN, métodos de criação de dados sintéticos da classe minoritária, somado à alguma técnica de undersampling.

Abaixo, estão alguns exemplos:

3.3.3.1. SMOTE + Tomek Links (SMOTE-Tomek)

Como o nome diz, combina os métodos SMOTE e Tomek, em que o primeiro cria dados sintéticos para a classe minoritária e o segundo remove os links de Tomek da classe majoritária (ou seja, remove os dados da majoritária mais próximo das classes minoritárias). O oversampling é executado antes do undersampling.

3.3.3.2. SMOTE + Edited Nearest Neighbors (SMOTE-ENN)

Esse método utiliza a criação de dados sintéticos do SMOTE e, em seguida, excluí algumas observações de ambas as classes que são identificadas como tendo classe diferente entre a classe de observação e as k classes majoritárias mais próximas [11, VIADINUGROHO, 2021].

3.3.3.3. SMOTE + Neighbourhood Cleaning Rule (SMOTE–NCR)

O SMOTE cria dados sintéticos da minoria, logo em seguida remove eventos duplicados e, por último, eventos ruidosos e ambíguos.

3.3.3.4. ADASYN + ENN

Parecido com SMOTE-ENN, mas agora se baseia na densidade dos dados, focado em regiões com pouca frequência da classe minoritária, para a criação dos dados sintéticos. Logo em seguida, são removidos os dados majoritários que são classificados erroneamente baseados nos três vizinhos mais próximos.

3.3.3.5. ADASYN + Tomek Links

Usando a técnica de ADASYN, cria-se novos dados e o Tomek Links, removendo os dados majoritários próximos dos minoritários.

4. DESAFIOS

O principal desafio encontrado neste projeto foi o desbalanceamento extremo dos dados de fraude de cartão de crédito, onde a classe minoritária (fraude) representa uma proporção muito pequena do conjunto de dados total (aproximadamente 0.0172%). Este desbalanceamento foi superado através da aplicação de técnicas de reamostragem (Oversampling e Undersampling) no conjunto de treino.

Além disso, encontrar métodos de balanceamento adequados, pois, pela quantidade pequena de dados minoritário, acaba que o Oversampling e o Undersampling tenha uma defasagem, ou criando muitos dados repetidos (no caso do sobreamostragem) ou apagando mais que o necessário (subamostragem).

De modo geral, o maior desafio foi a pesquisa e entender sobre como aplicar os métodos estudados. Ademais, por ser uma base muito extensa, não foi possível realizar testes usando o Gridsearch ou até mesmo o K-Folds, por conta de custar muito da máquina e demorar um tempo mais longo, impossibilitando usar essas técnicas.

5. Metodologias

Para este presente estudo, foi usado o dataset, para testes, o Credit Card Fraud Detection (Detecção de Fraude de Cartão de Crédito), retirado da plataforma Kaggle. Nele há dados de 284.807 transações, em que apenas 492 são fraudes, ou seja, 0,172% da base de dados inteira.

Além disso, usamos a tecnologias Python e as bibliotecas: openpyxl (criar editar planilhas excel), imblearn (para os métodos de balanceamento), sklearn (para

a criação dos modelos de machine learning), pandas (para abertura, leitura e análsie de base de dados), matplotlib e seaborn (para criação dos gráficos), kagglehub (importação do dataset) e os (criação e edição de pastas). Para esta ocasião, foram usados dois notebooks para a realização do projeto.

Os notebooks acabam por salvar as métricas calculadas em uma planilha excel, dividindo os dados entre os modelos e os métodos, e salva os gráficos da matriz de confusão, curva Precisão-Recall e curva AUC-ROC em pastas de cada modelo e os arquivos é nomeados: <tipo gráfico>-<modelo>-<método>.png. Os gráficos das curvas são condensados em um para cada método e guardado na raiz da pasta dos modelos (/projeto/modelos/<gráficos da curva>).

Em um dos notebooks foi usado os métodos de SMOTE, Random Oversampling, Random Undersampling; no outro, SMOTE, ENN e SMOTE-ENN. Ambos têm diferenças na separação dos dados.

Os dados foram separados em 20% para testes e 80% para treino. Além disso, foi usado o random_state=42 para padronização dos dados de treino e teste e o escalonamento da coluna *"Time"*(Hora) e *"Amount"*(valor da transação), para manter os padrões nos dados.

5.1. Modelos de aprendizado de máquina e métodos de balanceamentos utilizados

Durante o desenvolvimento do projeto, foram treinados e testados os modelos: KNN, Logistic Regression e Random Forest. Os dois primeiros são sensíveis aos dados desbalanceados, sendo K-vizinhos mais próximos mais afetado do que a Regressão Logística, enquanto o último acaba sendo mais resistente a eles.

O intuito de usar esses três modelos é verificar, antes e depois do balanceamento, o desempenho de cada um e comparar com cada método.

Os métodos utilizados foram o Random Oversampling, SMOTE, Random Undersampling, ENN e SMOTE-ENN.

Em primeira instância, foi usado o Random Oversampling, em que os dados de menor frequência são replicados.

Por ter uma quantidade de dados minoritário baixo, replicá-los geraria um problema de overfitting, por ter diversos dados iguais. Então o SMOTE acaba por ser a melhor opção, em que cria novas informações, ao invés de duplicar, sendo também usado no projeto.

Random Undersampling foi usado para testar a diminuição dos dados em maior frequência. Logo em seguida, foi usado o ENN, com o intuito de tentar remover os elementos ruidosos, não aleatoriamente.

Por último, o SMOTE-ENN foi para analisar quando cria-se dados sintéticos e remove aqueles que podem ser classificados de maneira errada.

5.2. Métricas à serem usadas

Em problemas com dados desbalanceados, a acurácia não é uma métrica adequada, pois se baseia na quantidade de acertos, em que os erros da classe minoritária pode ter pouco impacto nela. Métricas que consideram o desempenho de cada classe individualmente tornam-se essenciais. As métricas mais apropriadas para avaliar esses modelos incluem Recall, Precisão, F1-Score, AUC-ROC e a Curva Precision–Recall.

- Recall irá indicar das fraudes totais, quantas o modelo realmente acertou;
- Precisão irá apresentar, das fraudes previstas, quantas o modelo realmente acertou (ou seja, analisa quantas das fraudes previstas, era na verdade um dado normal);
- F1-Score: encontro harmônico entre o recall e precisão;
- AUC-ROC: analisa a taxa os verdadeiros positivos contra os falsos positivos;
- Precision-Recall: analisa a troca do modelo entre o recall e a precisão.

6. RESULTADOS E CONCLUSÕES

Os resultados estão na tabela abaixo:

MÉTODO	MODELO	ACC	REC	PRE	F1-SCO	AUC-ROC	PREC-REC
DESB.	*KNN*	0.999	0.806	0.918	0.858	0.943	0.889
DESB.	*LOG*	0.975	0.918	0.060	0.114	0.972	0.763
DESB.	*RF*	0.999	0.816	0.941	0.874	0.963	0.878
SMOTE	*KNN*	0.998	0.878	0.469	0.608	0.953	0.772
SMOTE	*LOG*	0.974	0.918	0.058	0.109	0.970	0.770
SMOTE	*RF*	0.999	0.827	0.861	0.843	0.966	0.863
SMOTE	*RF*	0.999	0.806	0.814	0.810	0.969	0.870
ROS	*RF*	0.999	0.776	0.938	0.849	0.958	0.869
ENN	*KNN*	0.999	0.816	0.842	0.829	0.943	0.864
ENN	*LOG*	0.975	0.918	0.059	0.111	0.952	0.775
ENN	*RF*	0.999	0.846	0.855	0.851	0.952	0.853
RUS	*RF*	0.964	0.908	0.042	0.080	0.978	0.760
SENN	*KNN*	0.997	0.877	0.392	0.542	0.953	0.720
SENN	*LOG*	0.973	0.918	0.055	0.104	0.970	0.786
SENN	*RG*	0.999	0.837	0.766	0.8	0.968	0.869

(OBS: valores estão em decimal e representa a porcentagem de cada métrica)

Legenda:

ACC: acurácia;

REC: recall;

PRE: precisão;

F1-SCO: F1-Score;

AUC-ROC: Area Under the Receiver Operating Characteristic Curve;

PREC-REC: Curva de Precisão e Recall;

SMOTE: Synthetic Minority Oversampling Technique;

ROS: Random Oversampling;

ENN: Edited Nearest Neighbors ;

RUS: Random Undersampling;

SENN: SMOTE-ENN;

KNN: k-nearest neighbors;

LOG: Regressão Logística;

RF: Random Forest.

Olhando os dados acima, é possível ver que ainda se pode ter uma redução de alguma métrica usando determinado método. Dentre todos os conhecimentos aplicados, a precisão foi a mais prejudicada no processo, apresentando diversos casos com valores abaixo dos 50%, ou seja, o modelo está tendo uma redução drástica.

Sem balanceamento: Ao treinar e testar os modelos, foi possível ver resultados melhores que os modelos balanceados. Todavia, isso é porque o dataset tem uma amostra muito pequena das fraudes, em que os modelos acabam por aproveitarem do desbalanceamento para classificar quase tudo como positivo. O Auc-Roc acaba não sendo afetado pelos dados desbalanceados, permanecendo alto; quando balanceado, esse cenário muda um pouco.

Random Oversampling (ROS): Melhorou significativamente o recall da classe minoritária (Fraude), aumentando a capacidade do modelo em detectar fraudes reais. No entanto, a precisão para a classe de fraude foi ligeiramente menor em comparação com SMOTE.

SMOTE: Também melhorou o recall da classe minoritária e apresentou um bom equilíbrio entre precisão e recall para a classe de fraude no modelo de Random Forest, resultando em um F1-Score similar ao ROS. Todavia, nos outros modelos apresentou uma baixa precisão.

Random Undersampling (RUS): Embora tenha alcançado um recall muito alto para a classe minoritária e a maior AUC-ROC, a precisão para a classe de fraude foi muito baixa. Isso indica que o modelo gerou um número excessivo de falsos positivos, o que seria problemático em um cenário real.

Edited Nearest Neighbors (ENN): Tanto o KNN e Random Forest apresentaram valores bons de recall e precisão, mostrando que a capacidade de prever fraudes melhorou. Entretanto, a Regressão Logística apresentou uma piora significativa em relação à precisão.

SMOTE-ENN: A junção de dois métodos para lidar com o dataset desbalanceado, o SMOTE e o ENN, demonstrou resultados bem menores que os métodos separados, mostrando que a junção de dois métodos podem não colaborar para melhor eficácia do modelo.

Percebendo os resultados, os modelos balanceados apresentam métricas menores, mas isso indica que eles não aproveitam mais do desbalanceamento para classificar seus dados.

Os modelos que apresentam precisão baixa, têm dificuldades de prever o que realmente são fraudes, enquanto o recall baixo apresenta pouca previsão delas.

Pode-se levantar __ pontos pelos resultados:

- Modelos treinados com datasets muitos desbalanceados acabam apresentando métricas altas, mas por conta deles aproveitarem do desbalanceamento para a classificação. Ao aplicar os métodos de balanceamento, as métricas diminuem, por conta de não poder tentar apenas classificar como positivo quase sempre;
- Modelos híbridos podem apresentar resultados não muito agradáveis, em comparação aos modelos isolados;
- Random Forest, em todo os teste, acaba sendo ainda a melhor opção para lidar com esse tipo de base de dados, em que os balanceamentos acabam aprimorando ainda mais sua eficiência, apesar de em determinados casos pode gerar resultados não muito agradáveis;
- A opção ideal seria a busca de mais dados, priorizando a classe minoritária, para melhor treinamento dos modelos.

6.1. Objetivos alcançados

Durante o projeto, conseguimos comparar os modelos balanceados com os desbalanceados.

Percebemos que uma quantidade menor de dados pode demonstrar métricas boas, mas enganosas. Quando aplicados os balanceamentos, é possível notar uma certa piora nessas medidas de avaliação, por conta do modelo está lidando com um cenário em que os dados agora se equiparam por quantidade.

Portanto, o projeto mostrou que uma base de dados extremamente desbalanceada, pode causar inconsistências.

Com isso, comprimos os objetivos:

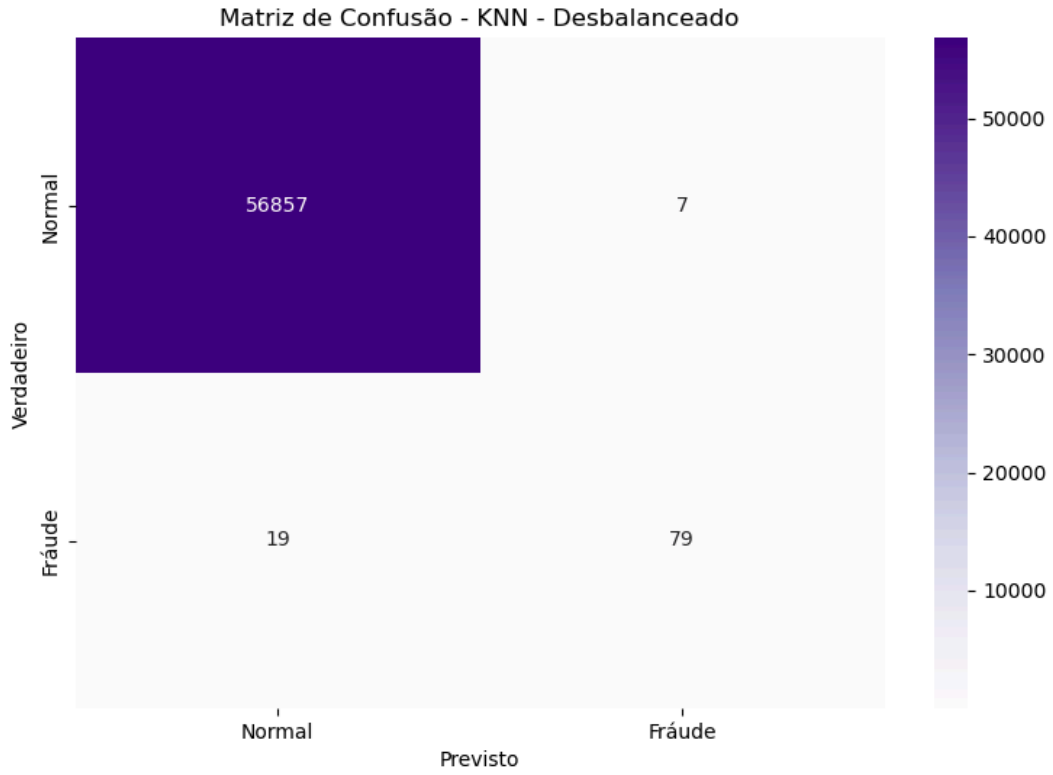
- Estudar os métodos de balanceamento;
- Comparação entre os modelos balanceados e os desbalanceados;
- Entendimento do cenário proposto; e
- Registro do conhecimento adquirido.

Abaixo está a matriz de confusão de cada modelo e o método usado, além do seus gráficos de Precision-Recall e AUC-ROC:

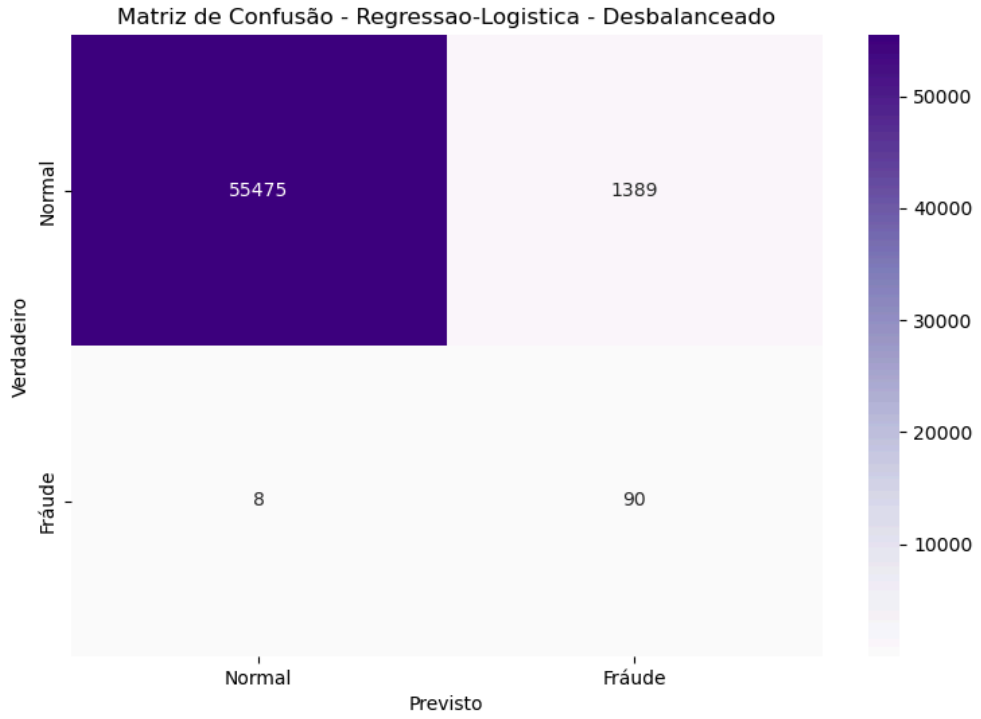
6.2. Desbalanceado

6.2.1. Matriz de confusão

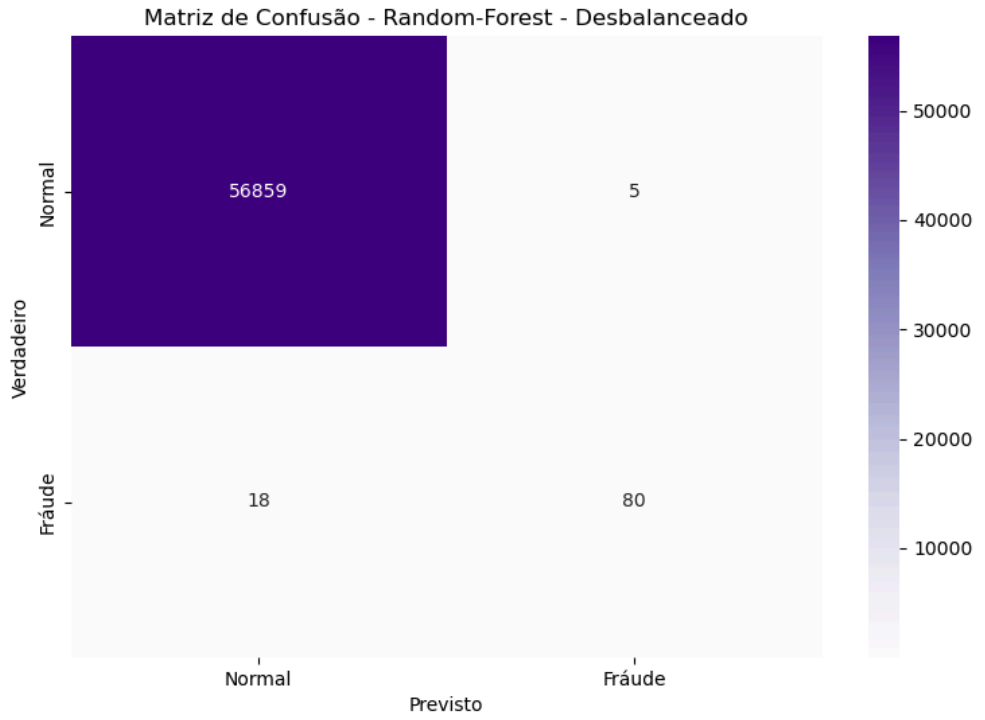
6.2.1.1. KNN



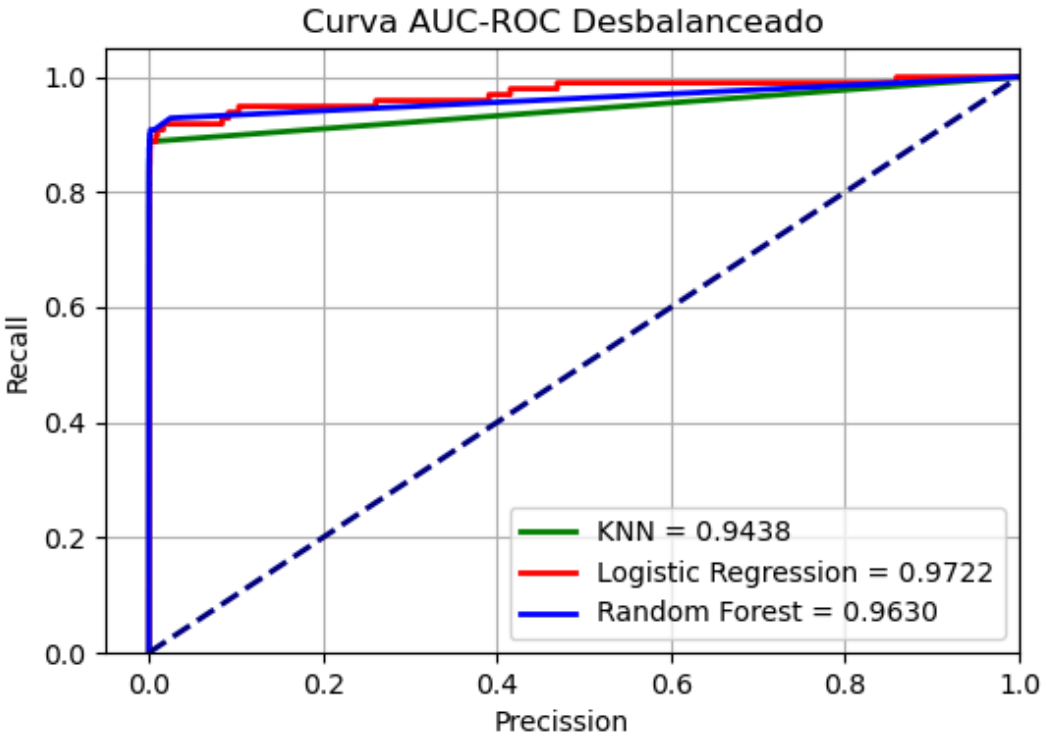
6.2.1.2. Regressão Logística



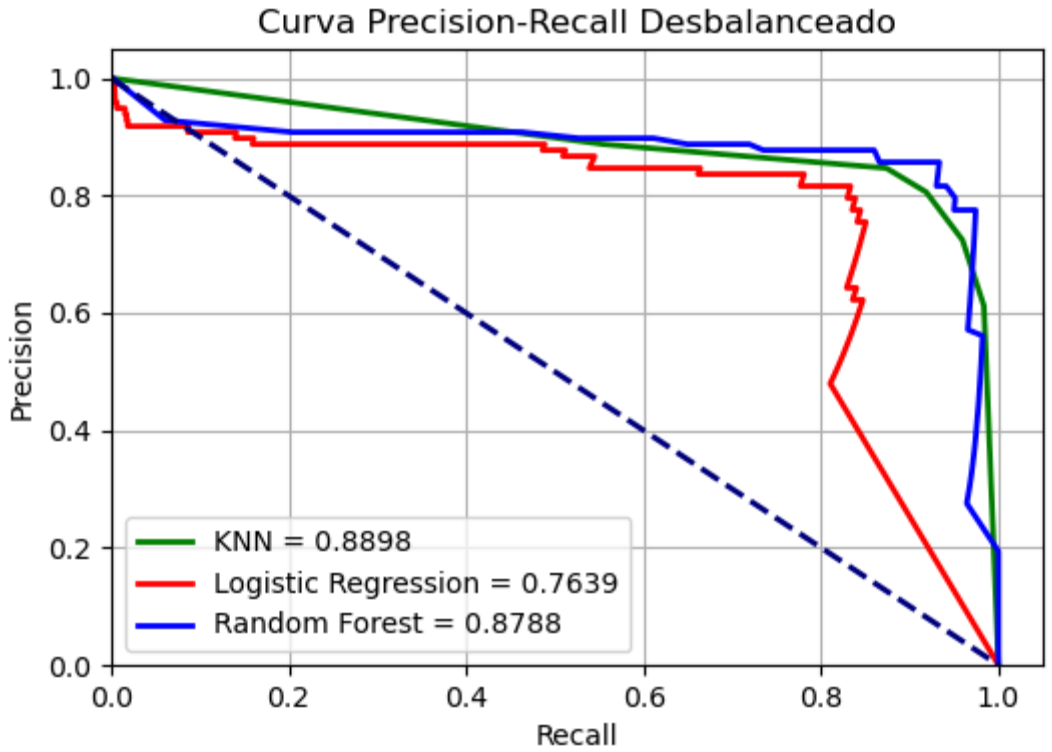
6.2.1.3. Random Forest



6.2.2. Gráfico AUC-ROC



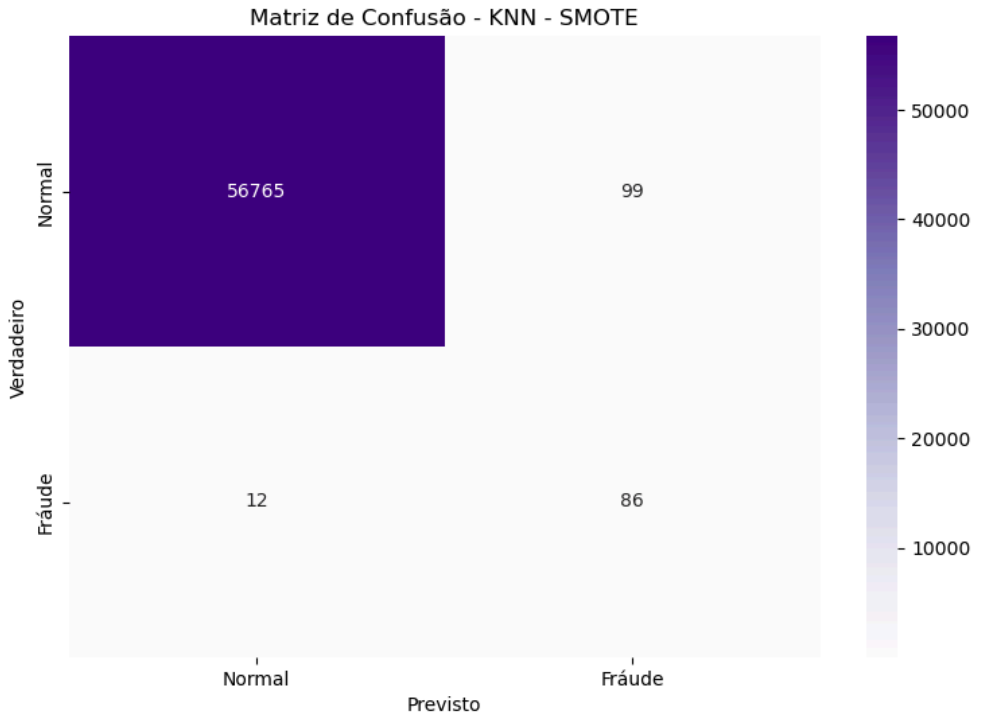
6.2.3. Gráfico Precisão-Recall



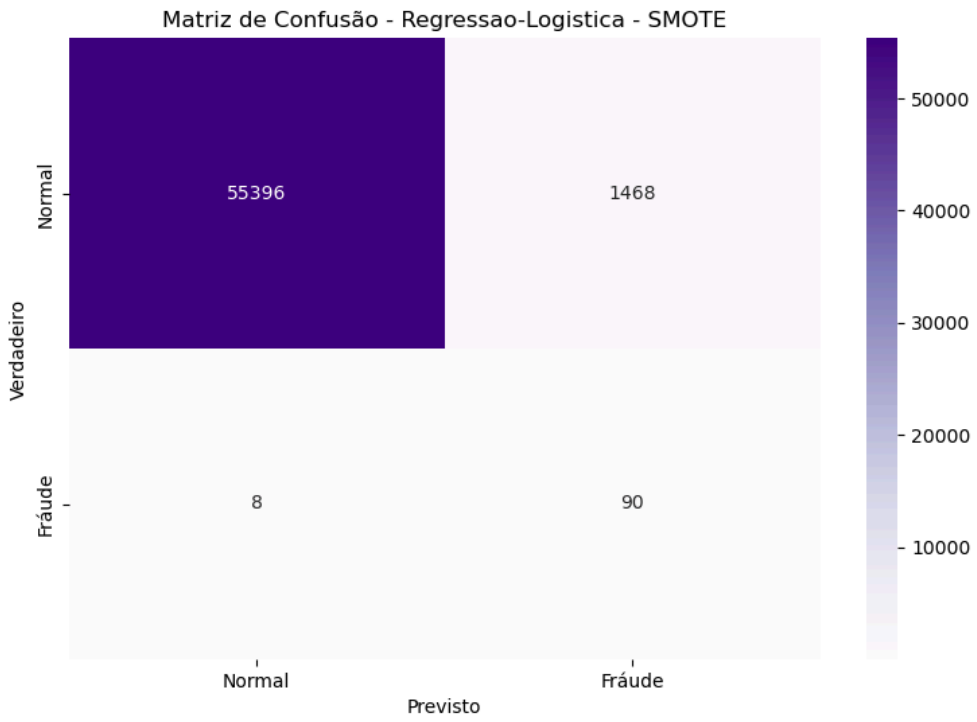
6.3. Oversampling

6.3.1. Matriz de confusão

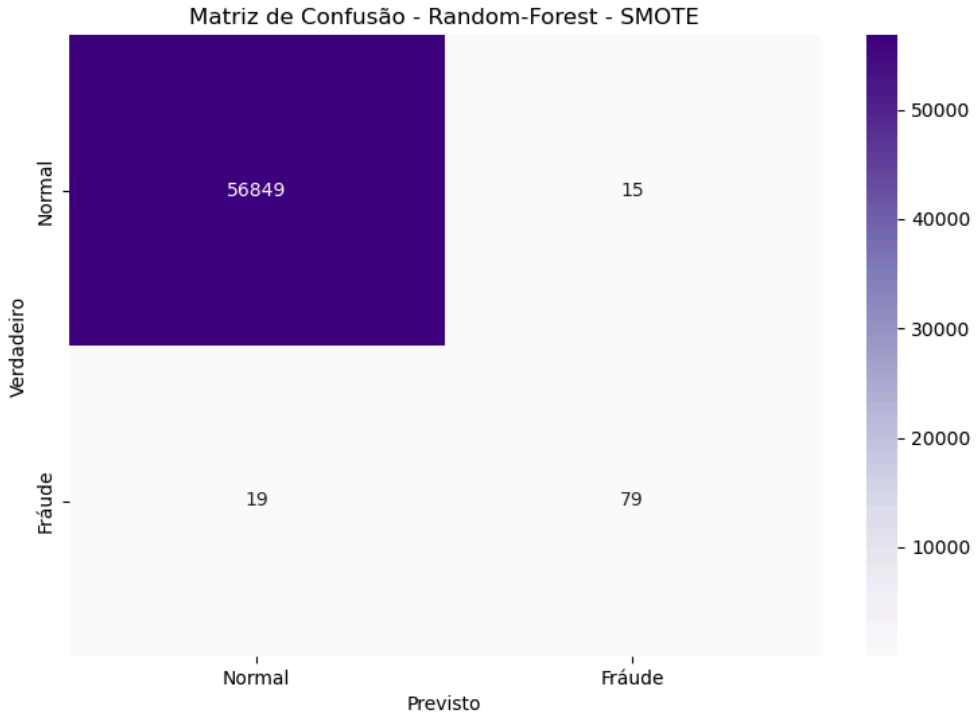
6.3.1.1. KNN



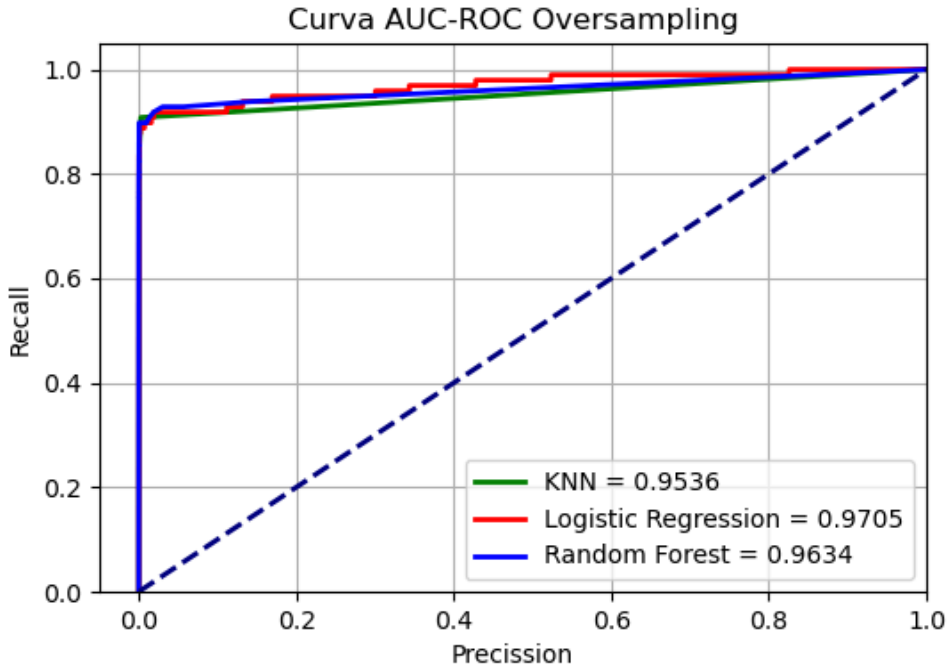
6.3.1.2. Regressão Logística



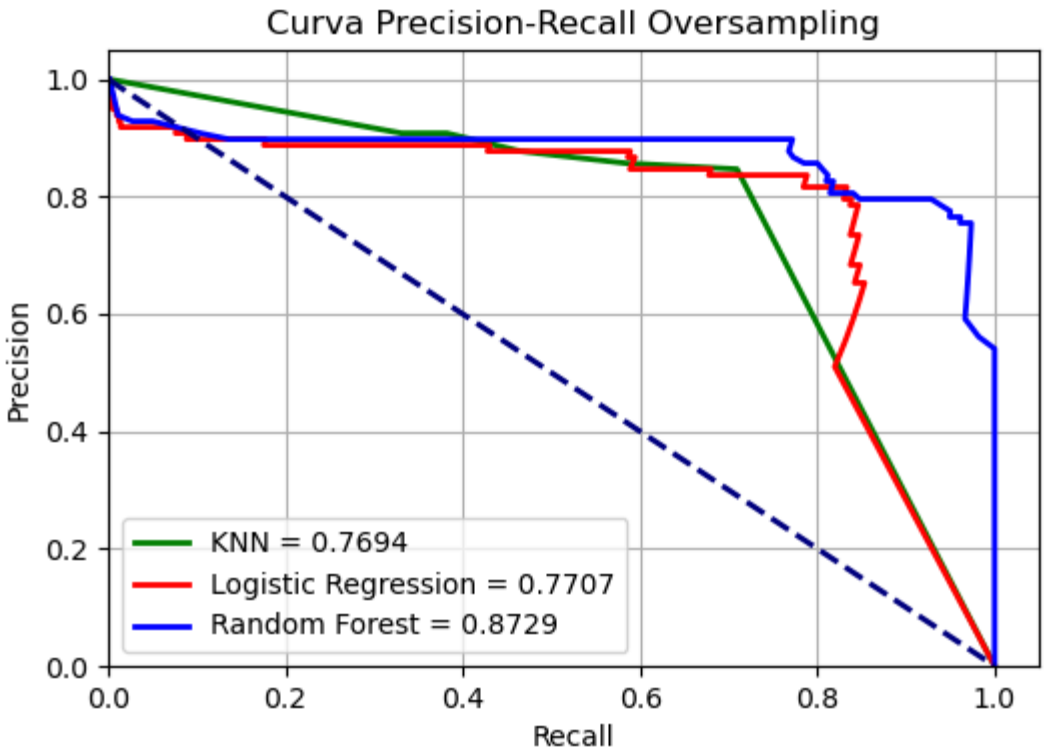
6.3.1.3. Random Forest



6.3.2. Gráfico AUC-ROC



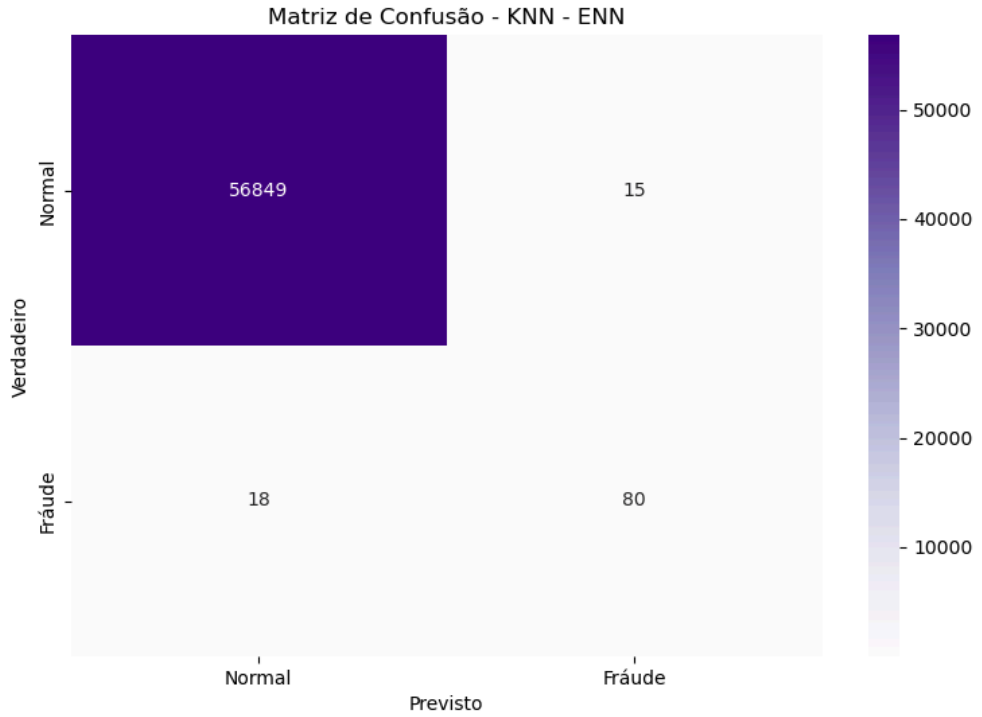
6.3.3. Gráfico Precisão-Recall



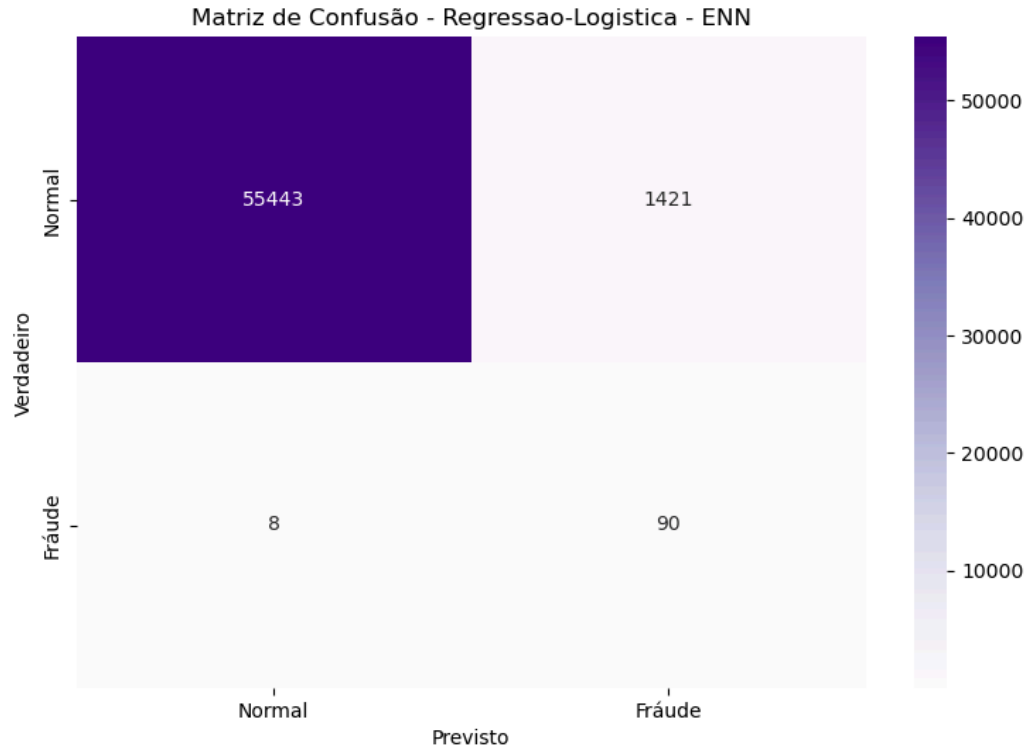
6.4. Undersampling

6.4.1. Matriz de confusão

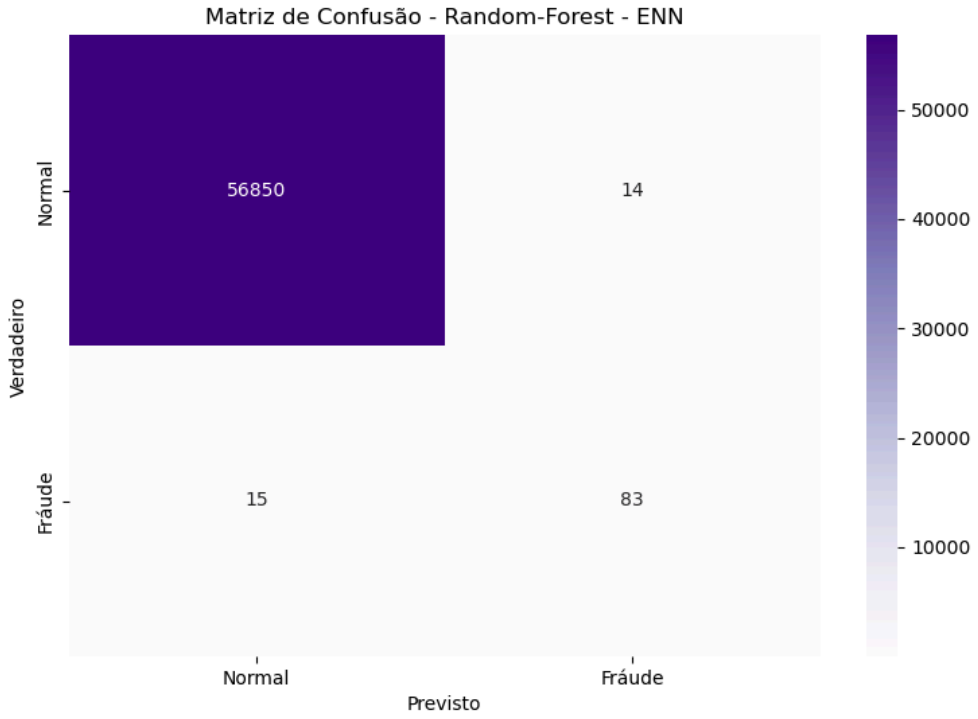
6.4.1.1. KNN



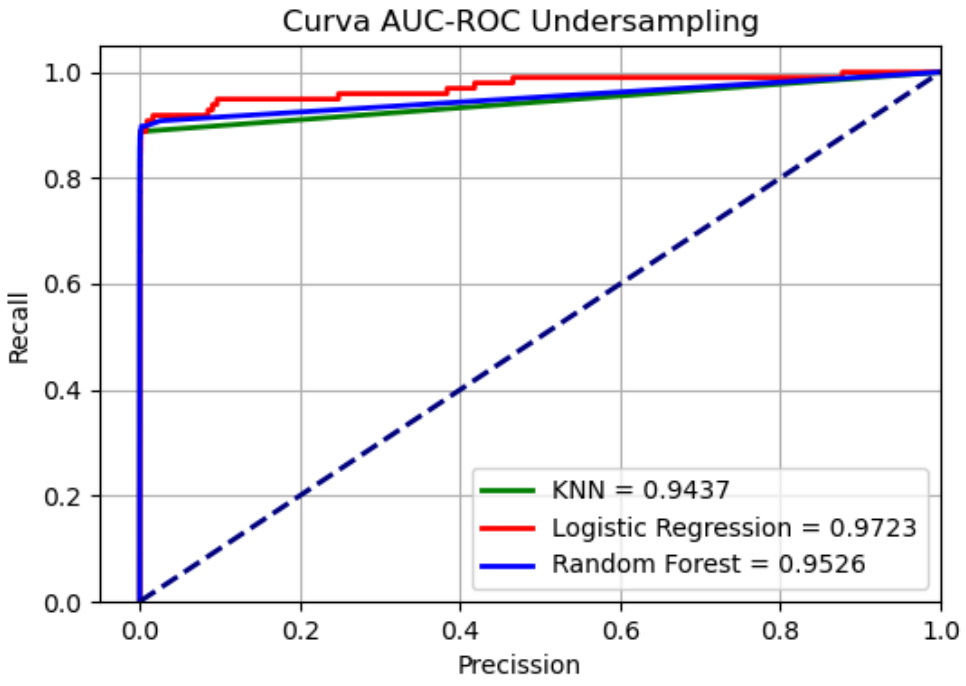
6.4.1.2. Regressão Logística



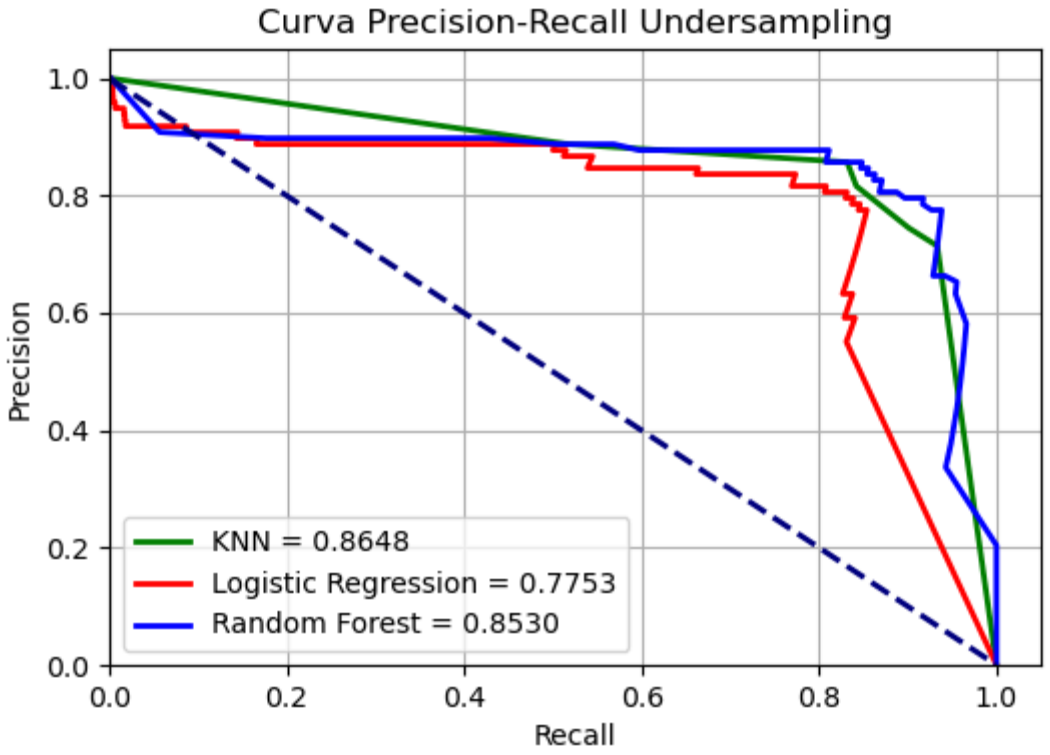
6.4.1.3. Random Forest



6.4.2. Gráfico AUC-ROC



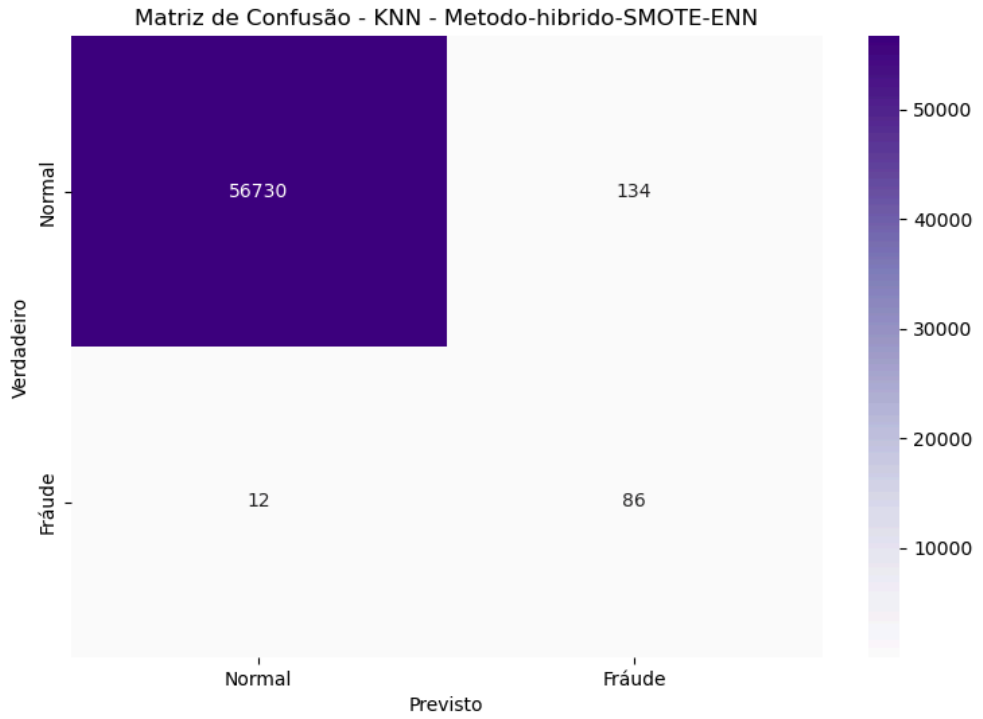
6.4.3. Gráfico Precisão-Recall



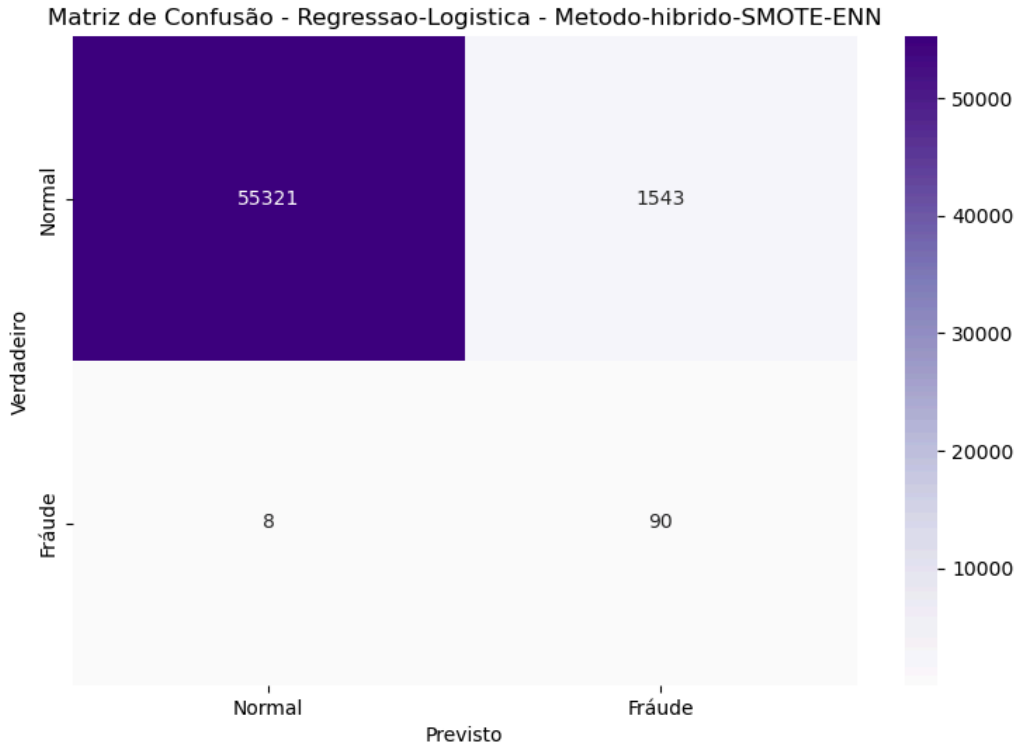
6.5. Método Híbrido

6.5.1. Matriz de confusão

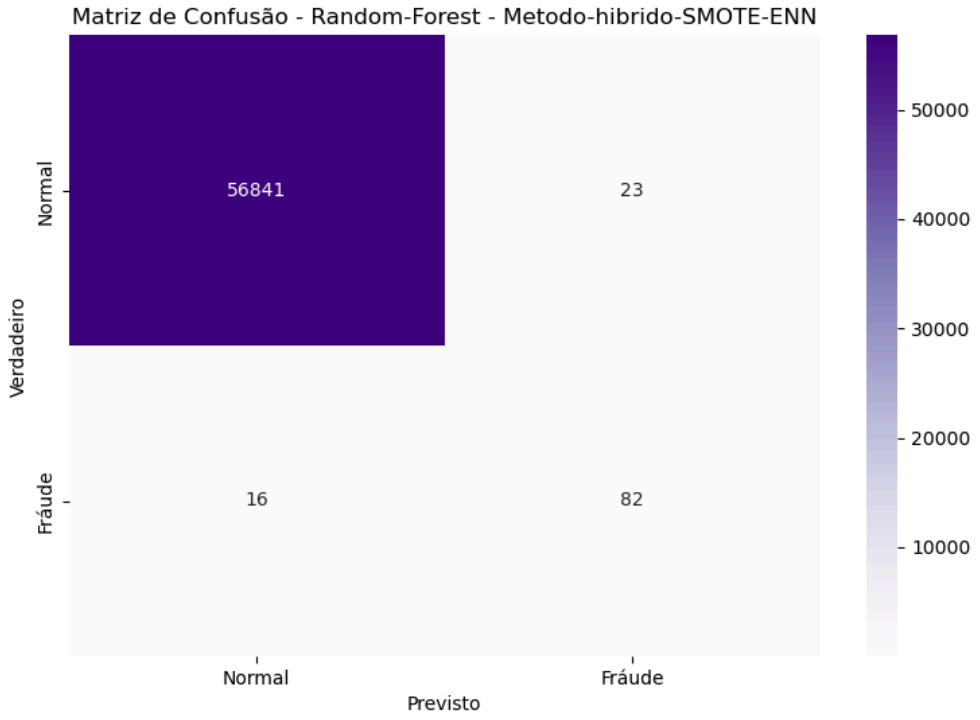
6.5.1.1. KNN



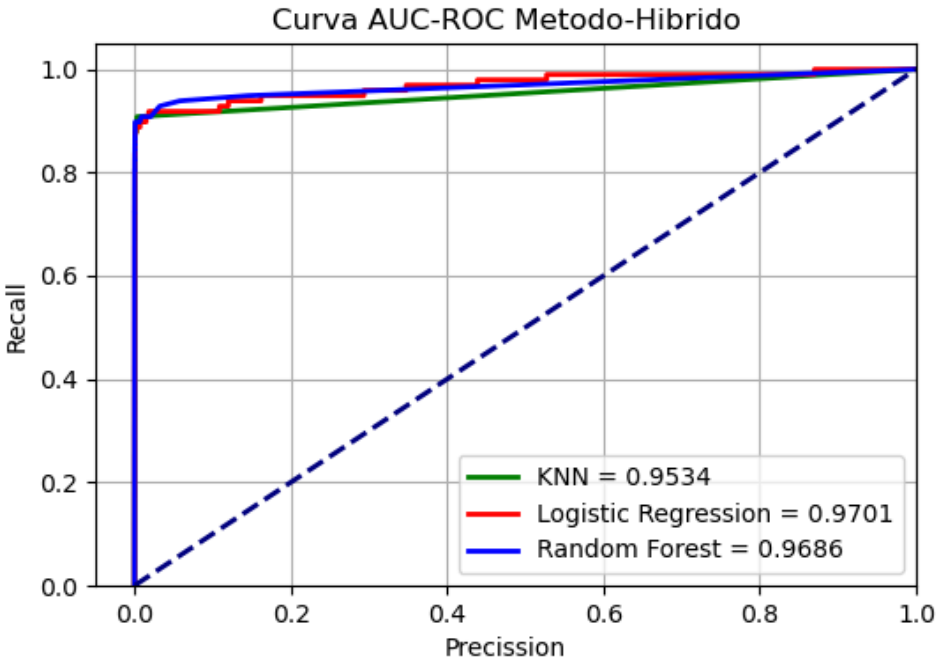
6.5.1.2. Regressão Logística



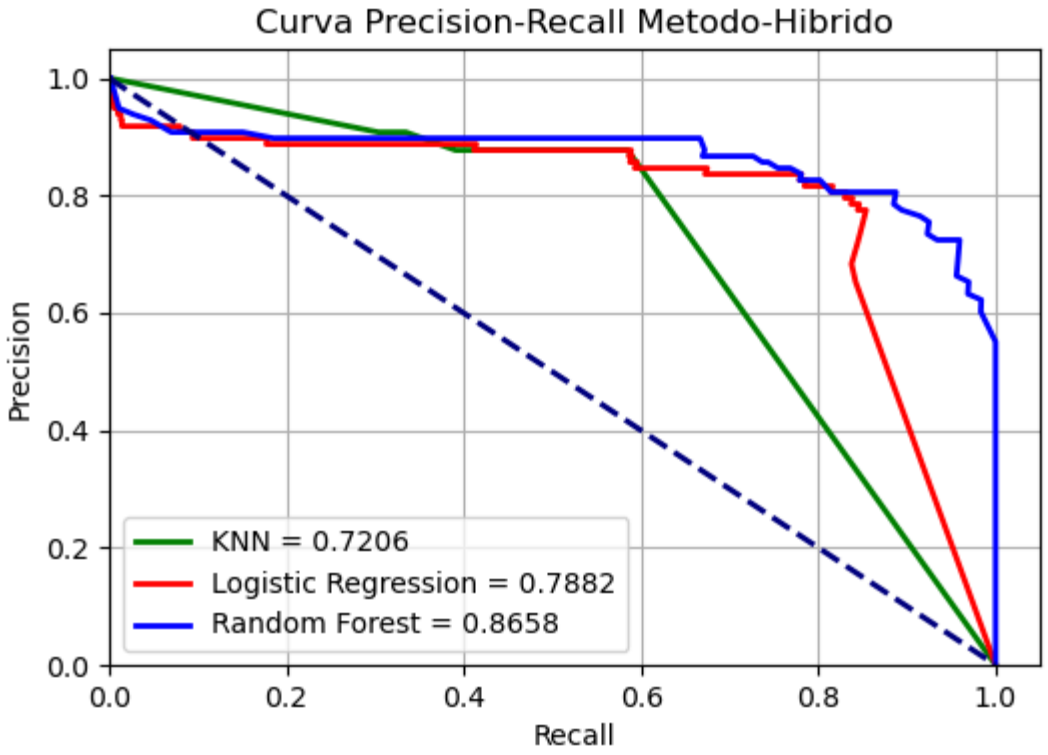
6.5.1.3. Random Forest



6.5.2. Gráfico AUC-ROC



6.5.3. Gráfico Precisão-Recall



7. BIBLIOGRAFIA

ALEMAR, Bernardo. Técnicas para Dados Desbalanceados (SMOTE e ADASYN). Medium, 2023. Disponível em: https://medium.com/@balemar/t%C3%A9cnicas-para-dados-desbalanceados-smote-e-adasyn-f891f9c46c6e. Acesso em: 23 out. 2025. [1]

AURÉLIEN, Géron. Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2. ed. 1005 Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Inc., 2019. 92-100 p. ISBN 978-1-492-03264-9. [2]

AZANK, Felipe; GURGEL, Gustavo Korzune. Dados Desbalanceados: O que são e como lidar com eles. Medium, 2020. Disponível em: https://medium.com/turing-talks/dados-desbalanceados-o-que-s%C3%A3o-e-como-evit%C3%A1-los-43df4f49732b. Acesso em: 15 out. 2025. [3]

Classificação: ROC e AUC. Google, Estados Unidos, 2025. Disponível em: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=pt-br. Acesso em: 09 de nov. 2025. [4]

CATUNDA, Heitor. Datasets Desbalanceados: O que São e Como Trabalhar com Eles?. Hashtag, 2022. Disponível em: https://www.hashtagtreinamentos.com/datasets-desbalanceados-ciencia-dados. Acesso em: 15 out. 2025. [5]

MIRANDA, João Vitor De; MIOTO , Ana Clara De Andrade ; PREMEBIDA, Sthefanie Monica. Lidando com o desbalanceamento de dados. Alura, 2022. Disponível em: https://www.alura.com.br/artigos/lidando-com-desbalanceamento-dados?srsltid=A

fmBOoqG3p0x-i37kPJ36oDpBA9TRfPkq8bSYXdy27y7dLXDkqQjiO3-. Acesso em: 18 out. 2025.[6]

PÁDUA, Matheus. Machine Learning -Métricas de avaliação: Acurácia, Precisão e Recall, F1-score. Medium, 2020. Disponível em: https://medium.com/@mateuspdua/machine-learning-m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-e-recall-d44c72307959. Acesso em: 09 nov. 2025. [7]

RIBEIRO, Hector Batista; SILVA, Leandro Oliveira Da; RABÊLO, Ricardo De Andrade Lira. Estratégias para Lidar com Desbalanceamento de Dados em Aprendizado de Máquina. 12. ed. Porto Alegre: Editora SBC, 2024. 79-95 p. ISBN 978-85-7669-602-5. [8]

SANTOS, Dheiver. Método Random Over Sampler você sabe o que é ? e qual importância para classificação?. Medium, 2023. Disponível em: https://medium.com/@dheiver.santos_10420/m%C3%A9todo-random-over-sampler-voc%C3%AA-sabe-o-que-%C3%A9-296403fe0c2d. Acesso em: 09 nov. 2025. [9]

VIADINUGROHO, Raden Aurelius Andhika. Imbalanced Classification in Python: SMOTE-Tomek Links Method: Combining SMOTE with Tomek Links for imbalanced classification in Python. towards data science, 2021. Disponível em: https://towardsdatascience-com.translate.goog/imbalanced-classification-in-python-smote-tomek-links-method-6e48dfe69bbc/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc. Acesso em: 01 nov. 2025. [10]

VIADINUGROHO, Raden Aurelius Andhika. Imbalanced Classification in Python: SMOTE-ENN Method. towards data science, 2021. Disponível em:

https://towardsdatascience.com/imbalanced-classification-in-python-smote-enn-method-db5db06b8d50/. Acesso em: 03 nov. 2025. [11]

WHAT is Undersampling?. Master's in Data Science, Estados Unidos, 2022. Disponível em : https://www-mastersindatascience-org.translate.goog/learning/statistics-data-science/undersampling/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc. Acesso em: 22 out. 2025 [12]