

# Mecanismo para Detecção de Transações de Cartão de Crédito Fraudulentas.

Douglas A. Clementino<sup>1</sup>, Rafael de Paulo Dias<sup>1</sup>

<sup>1</sup>Departamento de Informática  
Universidade Federal do Paraná (UFPR) – Curitiba – Brasil

<sup>1</sup>{dac17, rpd17}@inf.ufpr.br

## 1. Introdução

A partir do dataset ‘Credit Card Fraud Detection’ disponibilizado na plataforma Kaggle [ULB 2021], este trabalho buscará descrever processo de construção de uma ferramenta para detecção de transações de crédito fraudulentas, possibilitando a agilização das respostas a esse ataque por parte das credoras.

### 1.1. Motivação

- Detectar operações de crédito fraudulentas.
- Impedir reincidência em aprovação de crédito fraudulenta.
- Propor modelo base para detecção de operações fraudulentas de crédito, que futuramente pode ser aprimorado para cenários reais, propensos a maiores diversidades de ataques (resultando em *Concept Drift*).

## 2. Objetivo

Considerando a existência de mecanismos das credoras de crédito para consulta direta a seus clientes quanto a ocorrência de transações consideradas ‘anormais’ (geralmente em forma de notificação através de seus aplicativos), o grande ponto relacionado a este problema torna-se a maximização na identificação de transações fraudulentas mesmo que a custo da classificação errônea de transações válidas. Dessa forma, criando a possibilidade de consulta a clientes para o máximo de transações fraudulentas possível, minimizando os prejuízos finais.

Assim, além da acurácia geral do problema, buscaremos aprimorar a métrica recall para classe 1 (Fraudes), a fim de identificar o máximo possível de fraudes mesmo que a custo da acurácia geral.

## 3. Visão Geral

### 3.1. Coleta e Pré-Processamento

A base de dados foi coletada diretamente da plataforma Kaggle, ela possui formato CSV e possui as seguintes variáveis:

- **Time:** Intervalo de tempo decorrido desde a primeira amostra de dataset.
- **Amount:** Quantia de crédito solicitada.
- **Class:** Classe da transação, sendo ‘0’ o valor para transações válidas e ‘1’ o valor para transações fraudulentas.

- **v1-v28:** Variáveis escalares que, segundo o próprio fornecedor da base, seriam decorrentes do processo de redução de dimensionalidade a través da técnica PCA (*Principal Component Analysis*). Dessa forma, possibilitando a distribuição dos dados sem comprometer as informações pessoais dos clientes.

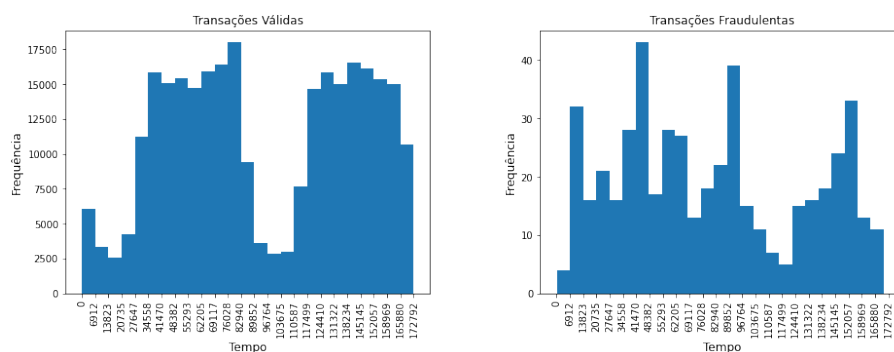
Dessa forma, exceto pelas 3 primeiras variáveis, não é possível efetuar uma devida exploração e combinação de variáveis a fim de enriquecer a base de dados final.

### 3.2. Distribuição das Classes

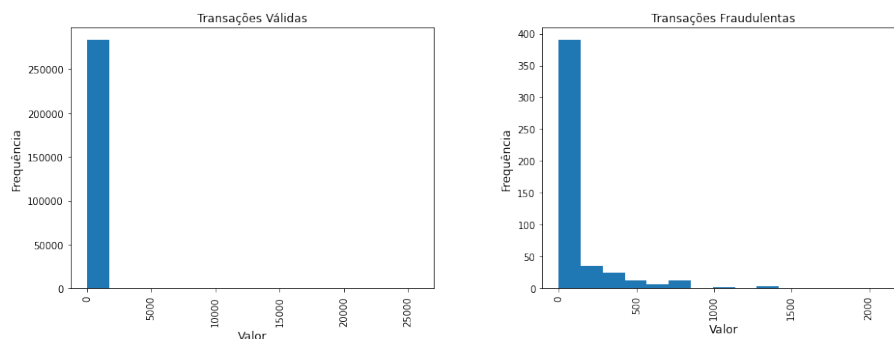
Como descrito anteriormente, a base de dados está dividida entre transações válidas, indicadas pelo valor '0' em variável 'Class' (sendo elas 284.315, representando 99,82% da base total) e as transações fraudulentas, indicadas pelo valor '1' (sendo elas 492, representando 0,17% da base total). Dessa forma, está claramente evidenciado um desbalançamento do dataset.

### 3.3. Análise de Dataset

Considerando que apenas as variáveis 'Time', 'Amount' e 'Class' são diretamente inteligíveis, tentou-se verificar a existência de relações entre 'Classe vs Time' e 'Class vs Amount':



**Figura 1. Histograma de Frequência Transações de cada Classe com relação ao Tempo.**



**Figura 2. Histograma em Transações com relação Classe e Valor.**

Porém, como é possível observar na Figura 1, existem alguns intervalos de tempo contendo algumas concentração maiores na quantidade de fraudes, porém se comparado

com o histograma de transações válidas, exceto pelos intervalos de tempo 6912 à 13823 e 89852 à 96764, a proporção de fraudes tende a acompanhar a proporção de transações válidas, indicando que estes pontos destacados podem ser apenas acaso.

Da mesma forma, Figura 2, observa-se que a grande concentração tanto das transações válidas e todas as transações fraudulentas encontra-se na faixa de até 2000,00 unidades monetárias, indicando a caráter moderado na excussão das fraudes, excluindo esta relação como de destaque para análise.

## 4. Metodologia

Para a criação do mecanismo de detecção de fraudes foram aplicadas as seguintes etapas:

- Escolha de Classificador
- Aprimoramento de Classificador
- Balanceamento Base de Treino

Em todas as etapas foi aplicada a técnica de Kfold [scikit-learn developers 2021a], dividindo o dataset completo em 5 partes, utilizando uma delas (20% de dataset total) como teste e as demais como treino (80% de dataset total), alternando a base de treino até que 5 modelos fossem gerados por experimento. Foram utilizados os parâmetros *shuffle=True* e *random\_state=123* de forma a randomizar os elementos escolhidos em cada agrupamento de forma igual para todos os experimentos, além de manter o caráter reprodutível do trabalho.

### 4.1. Escolha de Classificador

Nesta primeira etapa, foram utilizados os classificadores KNN [scikit-learn developers 2021b], Multi-layer Perceptron (MLP) [scikit-learn developers 2021c] e CatBoostClassifier [CatBoost 2021] (Que possui implementação paralela do classificador Random Forest). Cada uma das divisões do dataset foi aplicado a cada um dos classificadores, que estavam sendo utilizados com seus parâmetros padrão. Ademais, foram extraídas as métricas *Accuracy*, *Precision Recall f1-score*, *support*, além das curvas *ROC* e *Precision-Recall* e as respectivas Matrizes de Confusão para cada um dos experimentos.

### 4.2. Aprimoramento de Classificador (*CatBoostClassifier*)

Extraídos os resultados da etapa de **Escolha de Classificador**, obtivemos os melhores resultados para classe de interesse 1 e nas curvas *Precision-Recall* utilizando o classificador CatBoostClassifier (como pode ser observado na **Sessão 5.1**), dessa forma, foram executados experimentos variando-se os parâmetros do classificador em busca de aprimorar o desempenho na classificação da Classe 1 (Fraudes). Ao final, obtendo-se a seguinte variação de parâmetros como melhor resultado:

- **learning\_rate:** 0.0947.
- **max\_depth:** 7.
- **n\_estimators:** 700.0.

Os demais parâmetros foram executados com valores *default*.

### 4.3. Balanceamento de Bases de Treino

Finalmente, como ultima etapa, foi aplicada a técnica de *Synthetic Minority Over-sampling* (SMOTE) [imbalanced-learn developers 2021], que efetua o balanceamento de bases de treino a partir da criação de amostras sintéticas do conjunto minoritário, geradas a partir das amostras reais. Utilizou-se a semente *random\_state=42* para geração de amostras sintéticas. Ao final, extraíram-se métricas e gráficos previamente citados.

### 4.4. Considerações

Foram feitos experimentos a técnica de remoção de *Outliers*, porem isso resultou na remoção majoritária de amostras do subconjunto minoritário (Classe 1) aumentando ainda mais o desbalanceamento, desta forma, optou-se pela não aplicação da técnica sobre o dataset.

Para o CatBoostClassifier, o parâmetro *task\_type = "GPU"* foi utilizado para exploração dos resultados dos parâmetros e comparação com outros modelos (Acelerando os processos de treinamento e testes), porem não consta nos resultados finais pois gera saídas não determinísticas, de forma que os resultados poderiam ser não reproduzíveis.

## 5. Resultados

### 5.1. Comparação de Classificadores

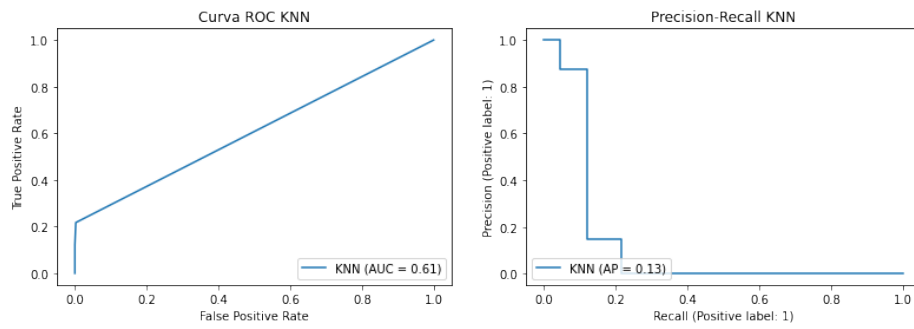
#### 5.1.1. KNN

Accuracy			
0,998			
Classe	precision	recall	f1-score
0	1	1	1
1	0,966	0,06	0,11

**Tabela 1.** Média de acurácia (Total), precision, recall, f1-score por classe (0 Válido, 1 Fraude)

		Valor Predito	
		Válido	Fraude
Real	Válido	56847	0
	Fraude	110	5

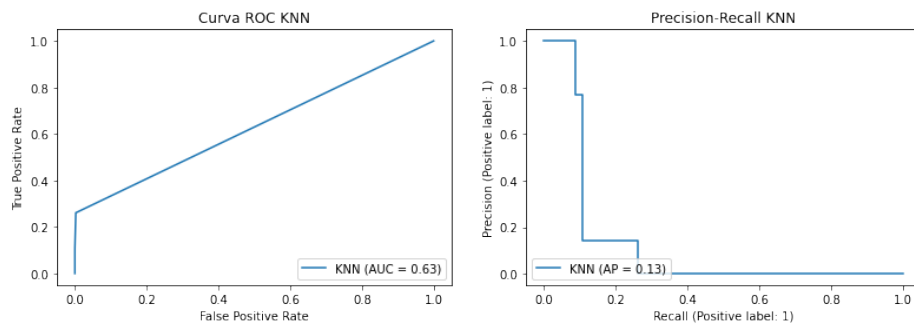
**Tabela 2.** Matriz de Confusão KNN para partição 0 de Kfold.



**Figura 3. Curvas ROC e Precision-Recall de KNN para partição 0 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56870	0
	Fraude	84	8

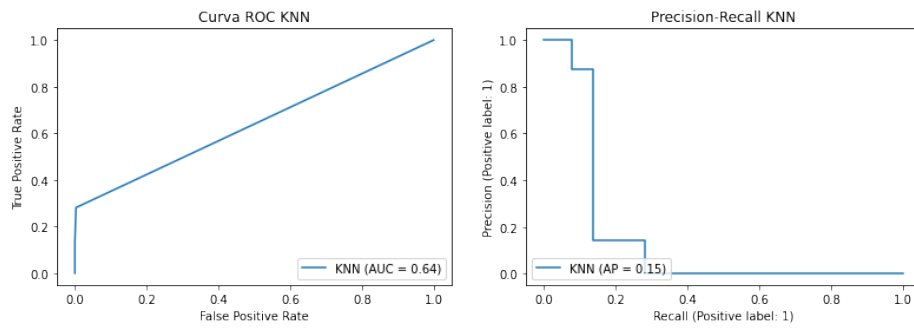
**Tabela 3. Matriz de Confusão KNN para partição 1 de Kfold.**



**Figura 4. Curvas ROC e Precision-Recall de KNN para partição 1 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56858	0
	Fraude	95	8

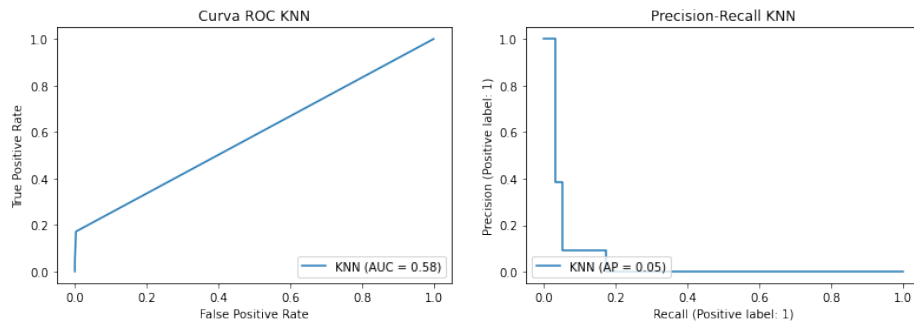
**Tabela 4. Matriz de Confusão KNN para partição 2 de Kfold.**



**Figura 5. Curvas ROC e Precision-Recall de KNN para partição 2 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56862	0
	Fraude	96	3

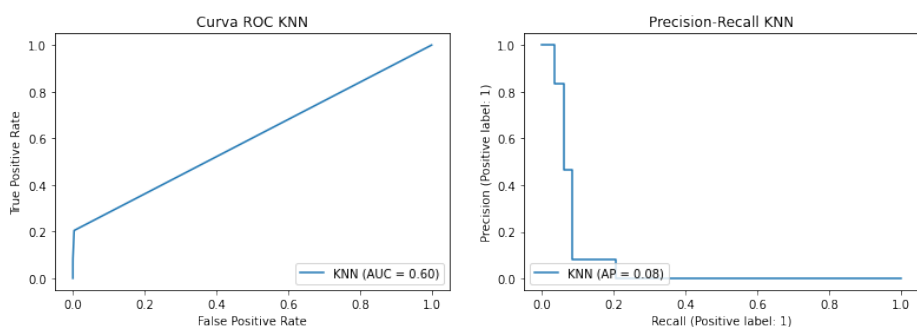
**Tabela 5. Matriz de Confusão KNN para partição 3 de Kfold.**



**Figura 6. Curvas ROC e Precision-Recall de KNN para partição 3 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56877	1
	Fraude	78	5

**Tabela 6. Matriz de Confusão KNN para partição 4 de Kfold.**



**Figura 7. Curvas ROC e Precision-Recall de KNN para partição 4 de Kfold.**

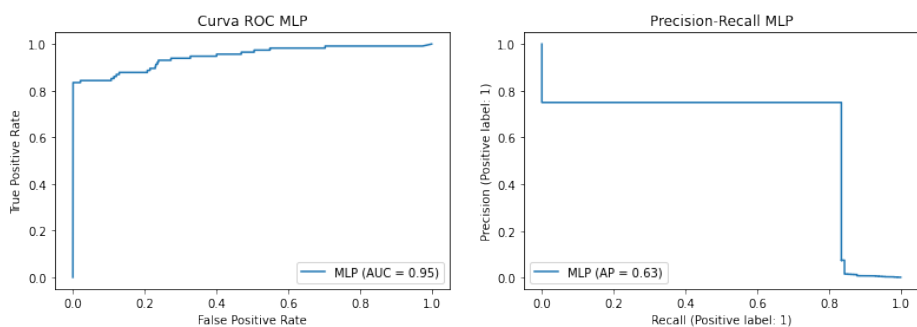
### 5.1.2. MLP

Accuracy			
0,998			
Classe	precision	recall	f1-score
0	1	1	1
1	0,512	0,674	0,548

**Tabela 7. Média de acurácia (Total), precision, recall, f1-score por classe (0 Válido, 1 Fraude)**

		Valor Predito	
		Válido	Fraude
Real	Válido	56769	78
	Fraude	19	96

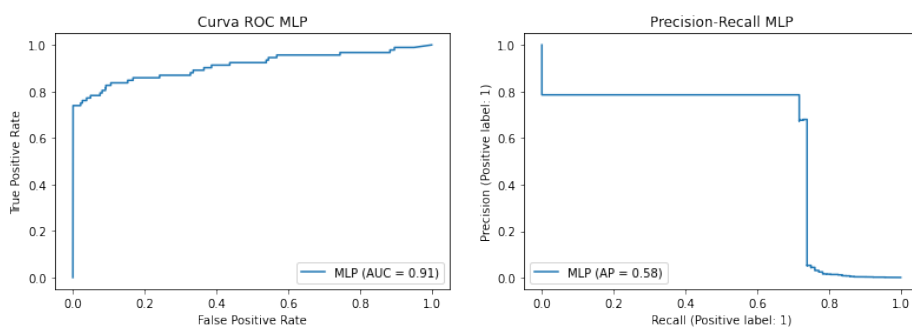
**Tabela 8. Matriz de Confusão MLP para partição 0 de Kfold.**



**Figura 8. Curvas ROC e Precision-Recall de MLP para partição 0 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56799	71
	Fraude	24	68

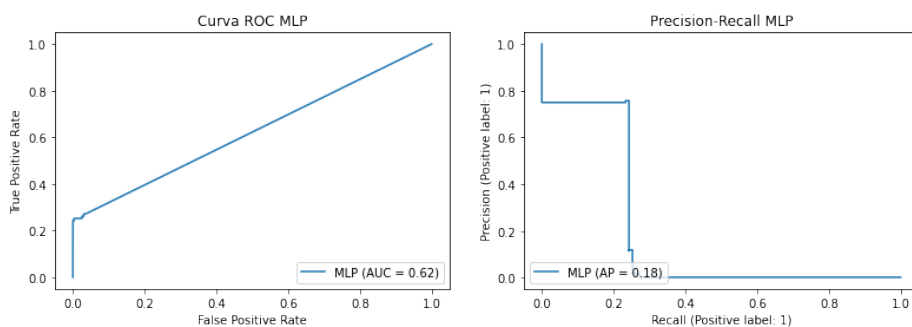
**Tabela 9. Matriz de Confusão MLP para partição 1 de Kfold.**



**Figura 9. Curvas ROC e Precision-Recall de MLP para partição 1 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56843	15
	Fraude	78	25

**Tabela 10. Matriz de Confusão MLP para partição 2 de Kfold.**

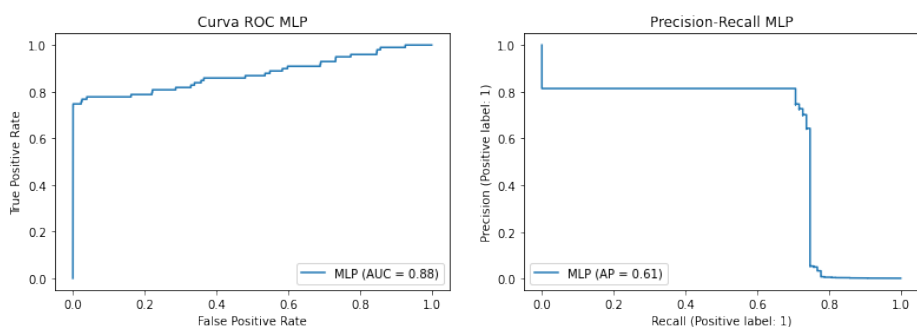


**Figura 10. Curvas ROC e Precision-Recall de MLP para partição 2 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56774	88
	Fraude	25	74

**Tabela 11. Matriz de Confusão MLP para partição 3 de Kfold.**

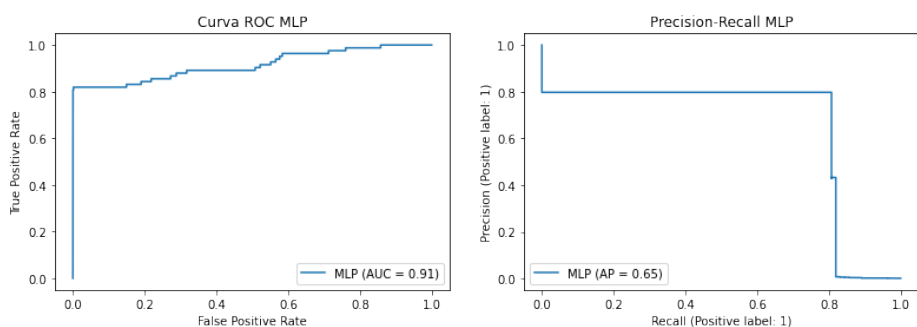




**Figura 11. Curvas ROC e Precision-Recall de MLP para partição 3 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56792	86
	Fraude	16	67

**Tabela 12. Matriz de Confusão MLP para partição 4 de Kfold.**



**Figura 12. Curvas ROC e Precision-Recall de MLP para partição 4 de Kfold.**

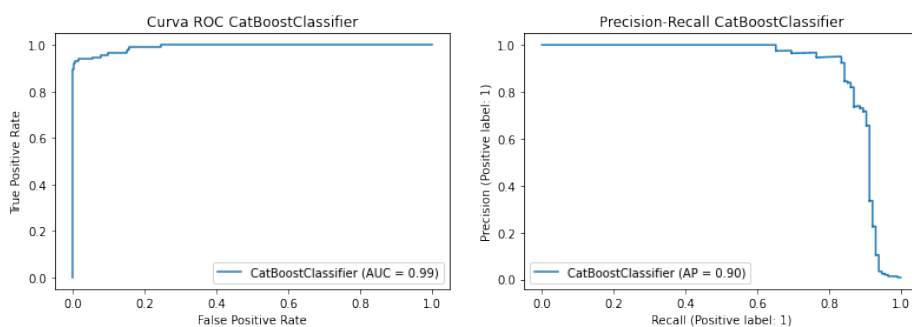
### 5.1.3. CatBoostClassifier

Accuracy			
0,999			
Classe	precision	recall	f1-score
0	1	1	1
1	0,952	0,796	0,866

**Tabela 13. Média de acurácia (Total), precision, recall, f1-score por classe (0 Válido, 1 Fraude)**

		Valor Predito	
		Válido	Fraude
Real	Válido	56841	6
	Fraude	23	92

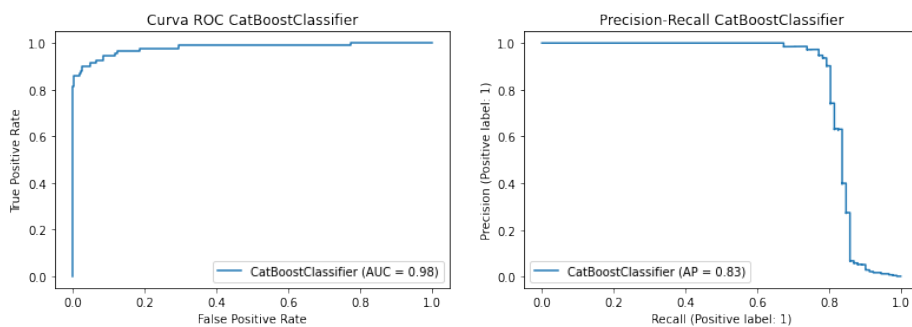
**Tabela 14. Matriz de Confusão CatBoostClassifier para partição 0 de Kfold.**



**Figura 13. Curvas ROC e Precision-Recall de CatBoostClassifier para partição 0 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56867	3
	Fraude	21	71

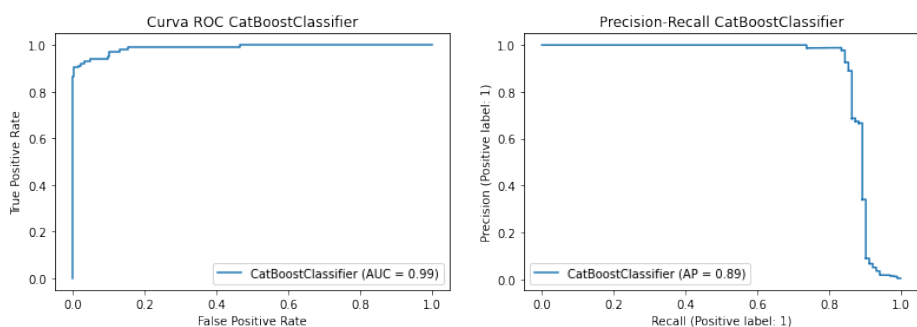
**Tabela 15. Matriz de Confusão CatBoostClassifier para partição 1 de Kfold.**



**Figura 14. Curvas ROC e Precision-Recall de CatBoostClassifier para partição 1 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56857	1
	Fraude	21	82

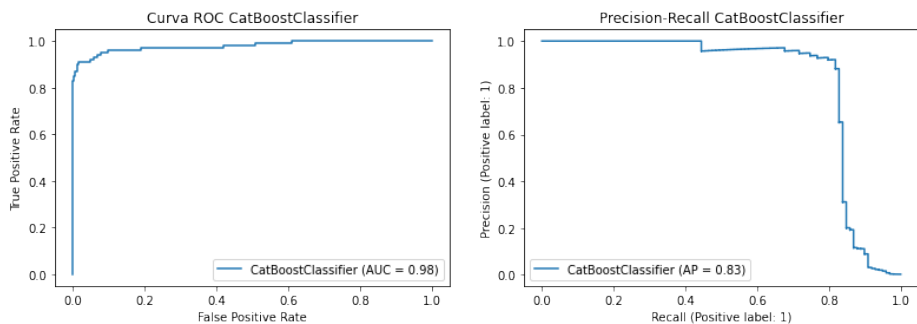
**Tabela 16. Matriz de Confusão CatBoostClassifier para partição 2 de Kfold.**



**Figura 15. Curvas ROC e Precision-Recall de CatBoostClassifier para partição 2 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56857	5
	Fraude	21	78

**Tabela 17. Matriz de Confusão CatBoostClassifier para partição 3 de Kfold.**



**Figura 16. Curvas ROC e Precision-Recall de CatBoostClassifier para partição 3 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56873	5
	Fraude	15	68

**Tabela 18. Matriz de Confusão CatBoostClassifier para partição 4 de Kfold.**

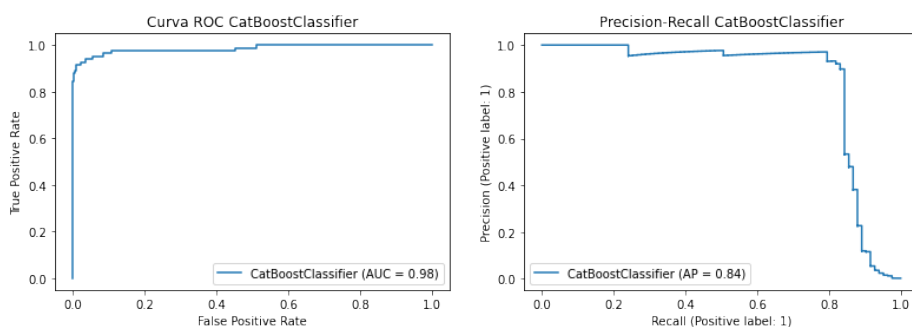


Figura 17. Curvas ROC e Precision-Recall de CatBoostClassifier para partição 4 de Kfold.

## 5.2. Tuning CatBoostClassifier

Accuracy			
0,999			
Classe	precision	recall	f1-score
0	1	1	1
1	0,956	0,802	0,872

Tabela 19. Média de acurácia (Total), precision, recall, f1-score por classe (0 Válido, 1 Fraude)

		Valor Predito	
		Válido	Fraude
Real	Válido	56842	5
	Fraude	21	94

Tabela 20. Matriz de Confusão CatBoostClassifier Tuned para partição 0 de Kfold.

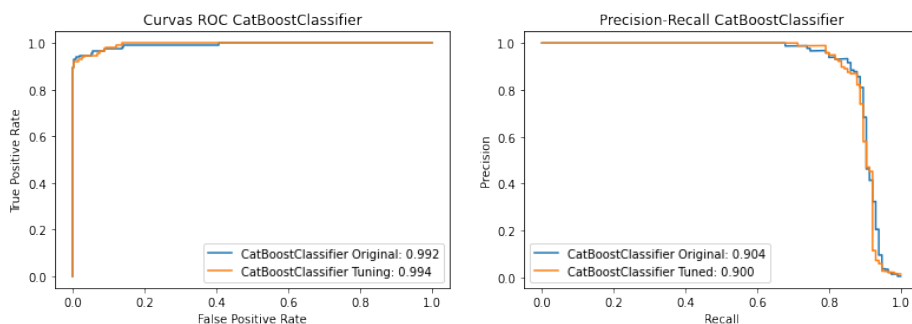
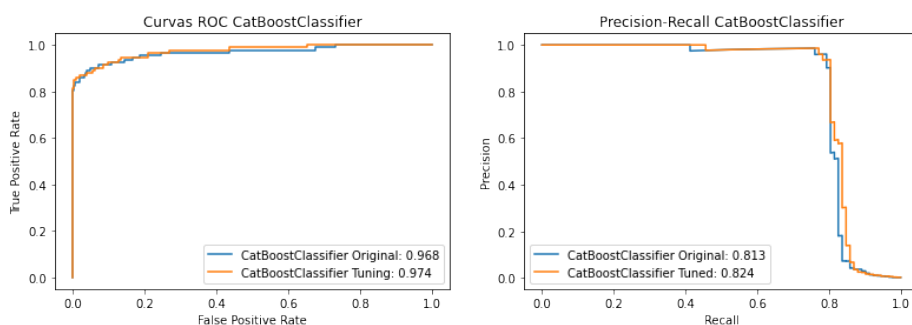


Figura 18. Curvas ROC e Precision-Recall de CatBoostClassifier Default e com Tuning para partição 0 de Kfold.

		Valor Predito	
		Válido	Fraude
Real	Válido	56867	3
	Fraude	21	71

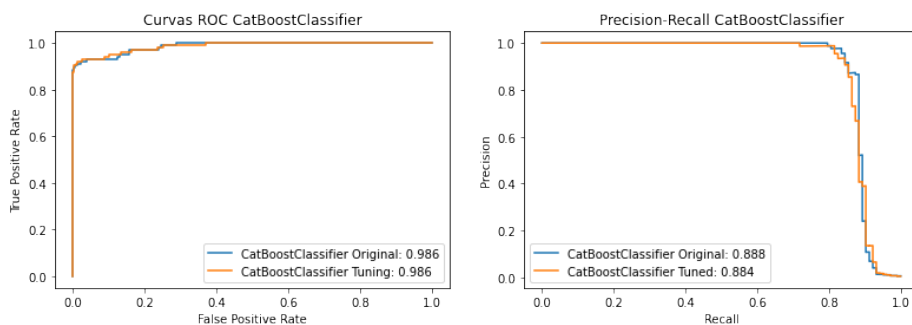
**Tabela 21. Matriz de Confusão CatBoostClassifier Tuned para partição 1 de Kfold.**



**Figura 19. Curvas ROC e Precision-Recall de CatBoostClassifier Default e com Tuning para partição 1 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56857	1
	Fraude	20	83

**Tabela 22. Matriz de Confusão CatBoostClassifier Tuned para partição 2 de Kfold.**



**Figura 20. Curvas ROC e Precision-Recall de CatBoostClassifier Default e com Tuning para partição 2 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56857	5
	Fraude	20	79

**Tabela 23. Matriz de Confusão CatBoostClassifier Tuned para partição 3 de Kfold.**

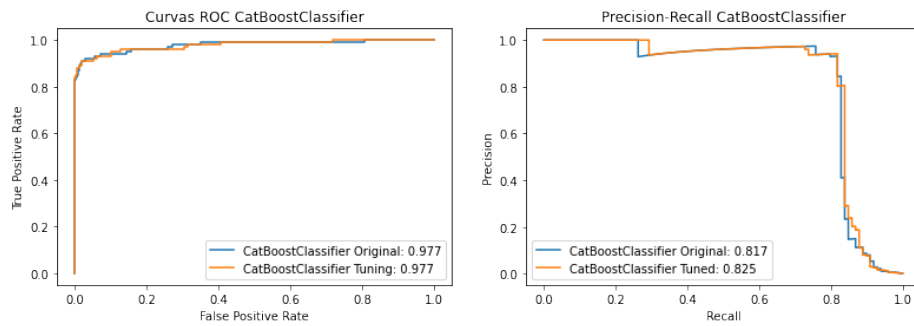


Figura 21. Curvas ROC e Precision-Recall de CatBoostClassifier Default e com Tuning para partição 3 de Kfold.

		Valor Predito	
		Válido	Fraude
Real	Válido	56874	4
	Fraude	16	67

Tabela 24. Matriz de Confusão CatBoostClassifier Tuned para partição 4 de Kfold.

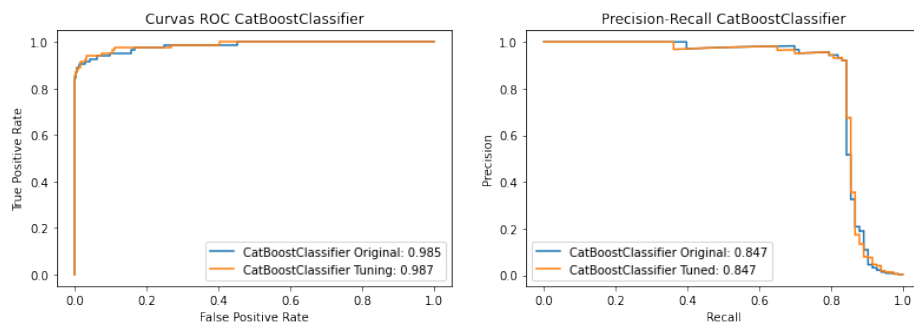


Figura 22. Curvas ROC e Precision-Recall de CatBoostClassifier Default e com Tuning para partição 4 de Kfold.

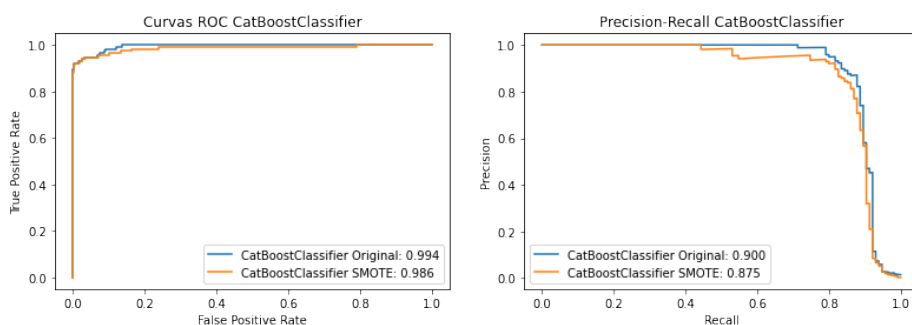
### 5.3. CatBoostClassifier Refinado com SMOTE

Accuracy			
0,999			
Classe	precision	recall	f1-score
0	1	1	1
1	0,712	0,852	0,776

Tabela 25. Média de acurácia (Total), precision, recall, f1-score por classe (0 Válido, 1 Fraude)

		Valor Predito	
		Válido	Fraude
Real	Válido	56809	38
	Fraude	14	101

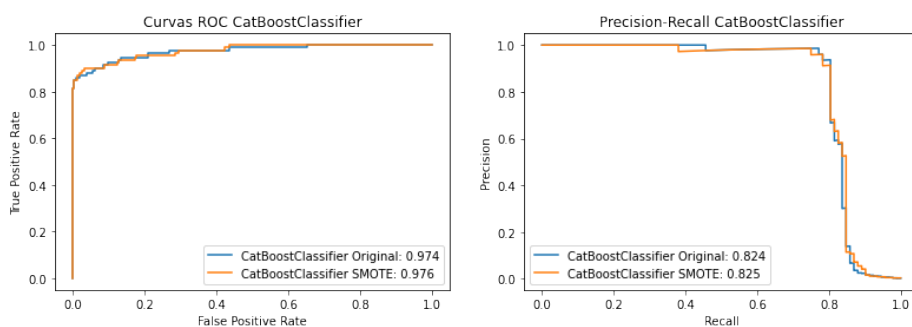
**Tabela 26. Matriz de Confusão SMOTE para partição 0 de Kfold.**



**Figura 23. Curvas ROC e Precision-Recall de CatBoostClassifier-Tuning com e sem SMOTE em base de treino partição 0 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56847	23
	Fraude	18	74

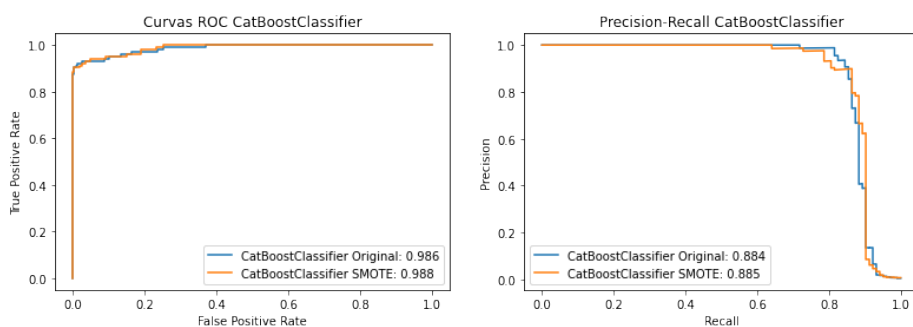
**Tabela 27. Matriz de Confusão SMOTE para partição 1 de Kfold.**



**Figura 24. Curvas ROC e Precision-Recall de CatBoostClassifier-Tuning com e sem SMOTE em base de treino partição 1 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56829	29
	Fraude	12	91

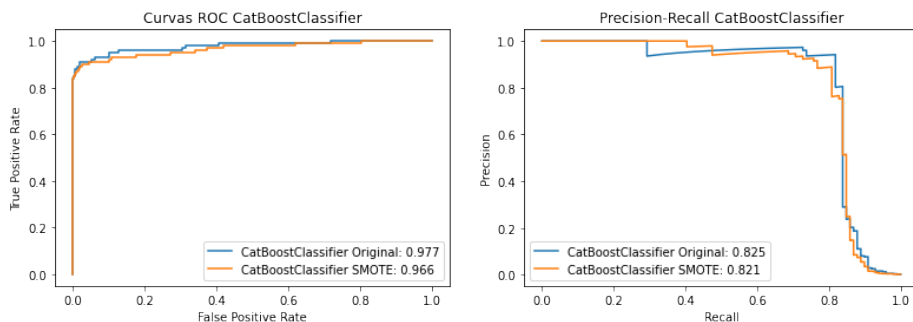
**Tabela 28. Matriz de Confusão SMOTE para partição 1 de Kfold.**



**Figura 25. Curvas ROC e Precision-Recall de CatBoostClassifier-Tuning com e sem SMOTE em base de treino partição 2 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56822	40
	Fraude	16	83

**Tabela 29. Matriz de Confusão SMOTE para partição 3 de Kfold.**

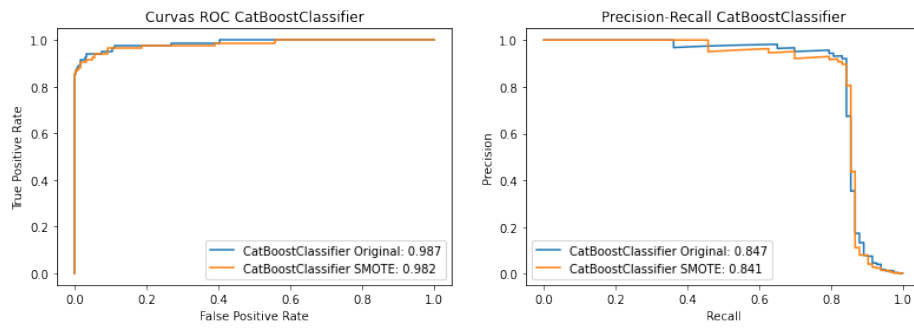


**Figura 26. Curvas ROC e Precision-Recall de CatBoostClassifier-Tuning com e sem SMOTE em base de treino partição 3 de Kfold.**

		Valor Predito	
		Válido	Fraude
Real	Válido	56838	40
	Fraude	12	71

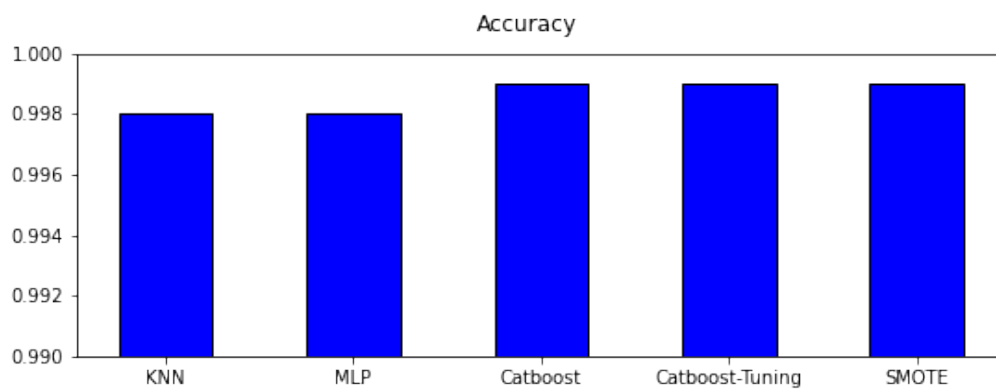
**Tabela 30. Matriz de Confusão SMOTE para partição 4 de Kfold.**





**Figura 27.** Curvas ROC e Precision-Recall de CatBoostClassifier-Tuning com e sem SMOTE em base de treino partição 4 de Kfold.

#### 5.4. Gráficos de Comparação de Médias de Kfold



**Figura 28.** Acuracia dos modelos, resultado obtido pela média dos kfold.

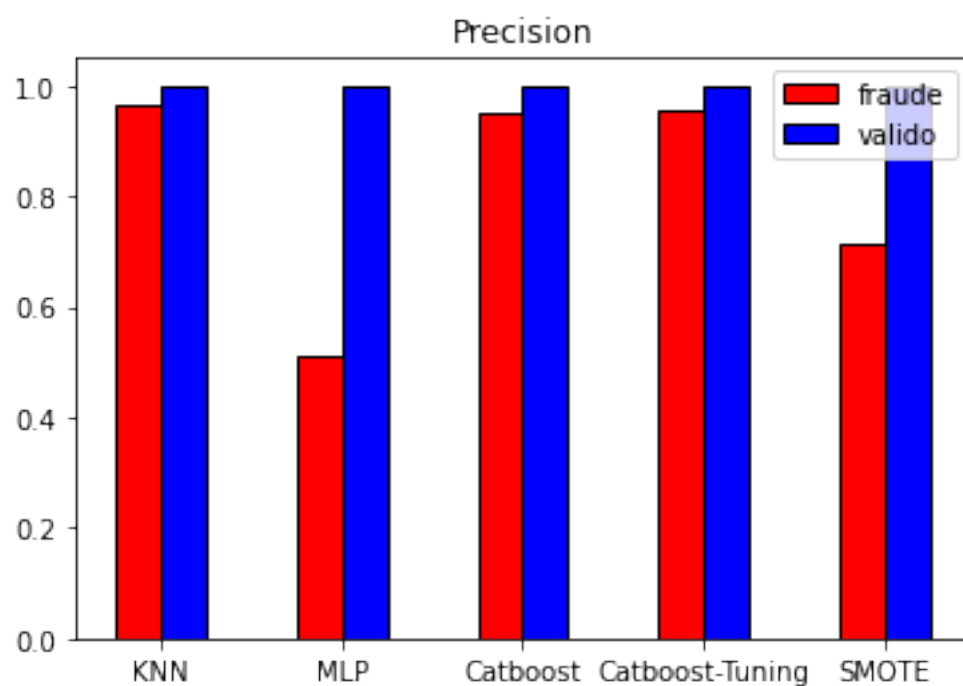


Figura 29. Precisão dos modelos, resultado obtido pela média dos kfold.

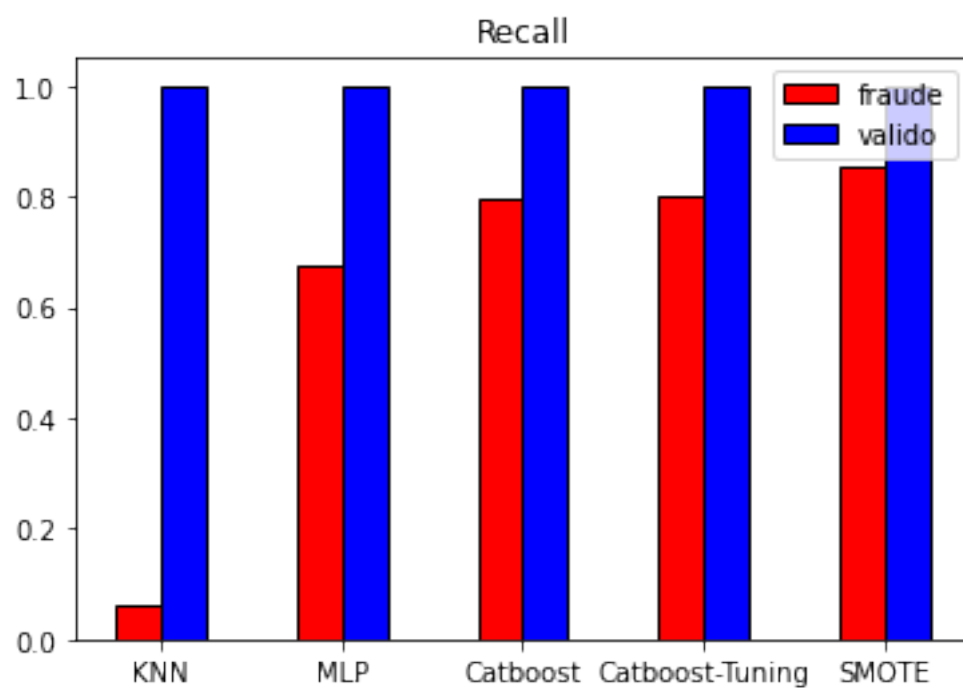
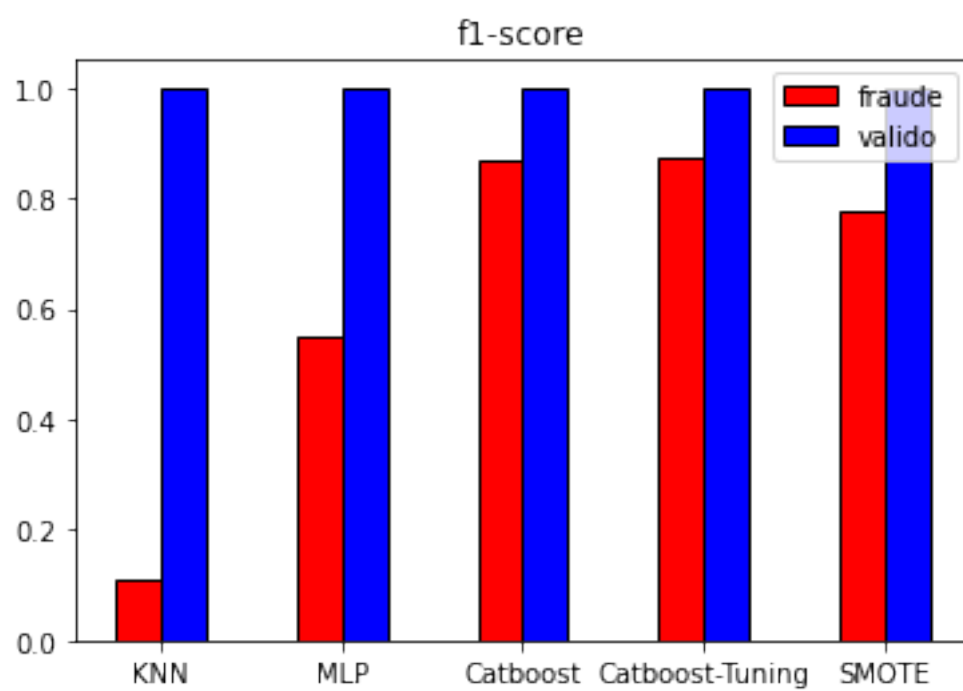


Figura 30. Recall dos modelos, resultado obtido pela média dos kfold.



**Figura 31. f1-score dos modelos, resultado obtido pela média dos kfold.**

#### 5.4.1. Importância de Variáveis CatBoostClassifier-Tuning + SMOTE

##### Análise de Importância de Variável para CatBoostClassifier-Tuning + SMOTE

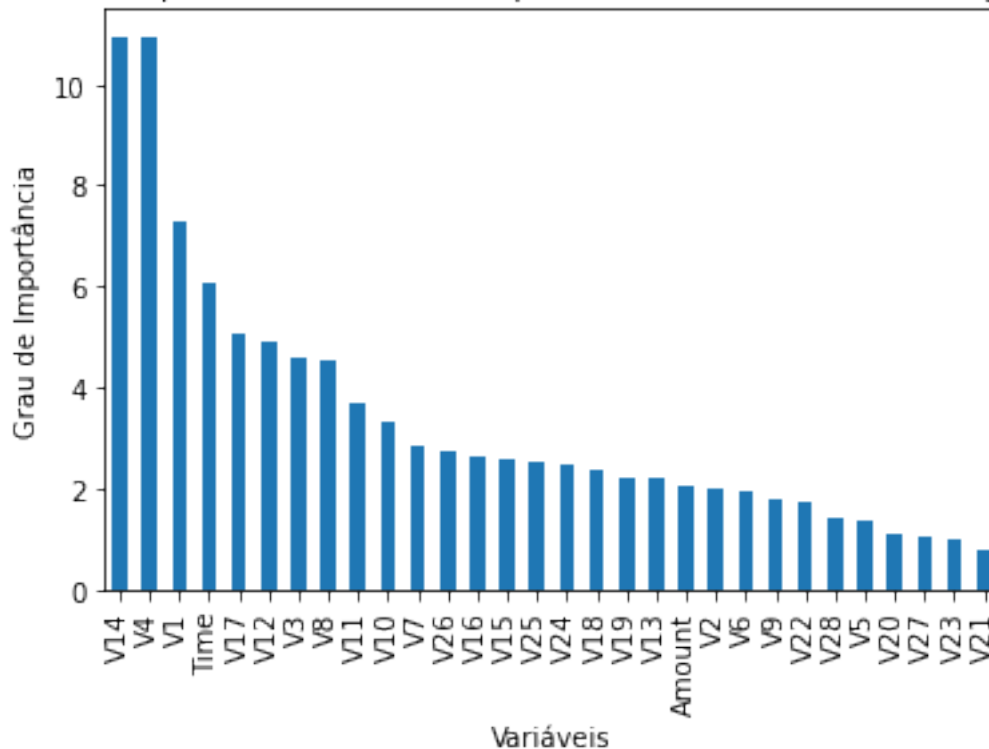


Figura 32. Ordem de variáveis mais importantes para CatBoostClassifier-Tuning + SMOTE, resultado obtido pela média dos kfold.

## 6. Conclusão

Na etapa de escolha de classificador, considerando os resultados da comparação entre os classificadores principalmente as taxas recall, F1-score (**Figuras 28 a 31**), a curva *Precision-Recall*, e a proporção de falsos positivos para classe 1 (falso negativo em matrizes de confusão) fica evidente o classificador *CatBoostClassifiers* foi o que gerou os melhores resultados considerando acurácia geral mas principalmente a precisão na classificação de elementos da classe 1 (Fraudes) no grupo de testes.

Quanto ao tuning da ferramenta CatBoostClassifier, observa-se que houve ganho nos 4 conjuntos de Kfold e perdendo alguma performance no ultimo experimento. Porem pode-se observar pelos gráficos de comparação de precision, recall, e f1-score (**Figuras 29 à 31**) e tabelas contendo estas métricas ( **Tabela 13** para CatBoostClassifier default e **Tabela 19** para CatBoostClassifier-Tuning) que em média houve ganho no tuning de classificador.

Finalmente, nos gráficos de **Figuras 28 à 31** onde se compara a versão CatBoostClassifier-Tuning com e sem a utilização do SMOTE, é possível observar que o balanceamento sintético das bases de teste tem grande impacto na precisão da classificação da classe minoritária (Fraudes), tendo incremento na métrica recall para

Fraudes (diminuição na classificação errônea de entradas da classe 1) em detrimento de alguma perda na precisão.

Além disso, com 'Time' em 4º lugar em importância de variáveis, como pode ser visto em **Figura 32**, talvez anomalias identificadas anteriormente em **Figura 1** tenham contribuído para discriminação, possivelmente indicando alguma sazonalidade tanto em transações válidas quanto em fraudes.

Com o final do experimento, é possível concluir que as técnicas implementadas contribuíram para o desempenho do classificador em relação a classificação de fraudes, tendo como consequência perda de precisão na classificação de transações válidas mas que como descrito em **Sessão 2. Objetivo**, problemas decorridos do deficit de precisão gerados por esta perda podem ser contornados por ferramentas já utilizadas atualmente pelas credora de crédito.

## 7. Reproducibilidade

Notebook disponível em <https://github.com/Daffc/Deteccao-Fraudes-ML>.

## Referências

CatBoost (2021). Catboostclassifier. [https://catboost.ai/en/docs/concepts/python-reference\\_catboostclassifier](https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier). Acessado em: 30/11/2021.

imbalanced-learn developers, T. (2021). Smote. [https://imbalanced-learn.org/stable/references/generated/imblearn.over\\_sampling.SMOTE.html](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html). Acessado em: 30/11/2021.

scikit-learn developers (2021a). sklearn.model\_selection.kfold. [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.KFold.html#examples-using-sklearn-model-selection-kfold](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html#examples-using-sklearn-model-selection-kfold). Acessado em: 30/11/2021.

scikit-learn developers (2021b). sklearn.neighbors.kneighborsclassifier. <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>. Acessado em: 30/11/2021.

scikit-learn developers (2021c). sklearn.neighbors.mlpclassifier. [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html). Acessado em: 30/11/2021.

ULB, M. L. G. (2021). Credit card fraud detection. <https://www.kaggle.com/mlg-ulb/creditcardfraud/>. Acessado em: 09/11/2021.