UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA QUÍMICA

ENG07053 - TRABALHO DE DIPLOMAÇÃO EM ENGENHARIA QUÍMICA

Uso de Autoencoder para Predição de Propensão a Empréstimo

Autor: Gabriel Speranza Pastorello

Orientador: Prof. Dr. Jorge Otávio Trierweiler

Coorientador: Rafael Henrique Martello

Porto Alegre, fevereiro de 2024

Gabriel Speranza Pastorello

Uso de Autoencoder para Predição de Propensão a Empréstimo

Trabalho de Conclusão de Curso apresentado à COMGRAD/ENQ da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Bacharel em Engenharia Química

Orientador: Prof. Dr. Jorge Otávio Trierweiler
Coorientador: Rafael Henrique Martello

Banca Examinadora:

Prof. Dr. Marcelo Farenzena, Universidade Federal do Rio Grande do Sul
Dr. Jônathan William Vergani Dambros, Universidade Federal do Rio Grande do Sul
Me. Matheus Funck, Universidade Federal do Rio Grande do Sul

Porto Alegre
2024

AGRADECIMENTOS

Agradeço primeiramente à minha família, meu pai Marco e minha mãe Lúcia por todo o suporte fornecido nestes anos de graduação.

Agradeço também a todo o corpo de professores da Universidade Federal do Rio Grande do Sul que me auxiliou durante essa trajetória, em especial ao professor Jorge e ao coorientador Rafael por todo o suporte fornecido desde o início deste trabalho.

Aos colegas ao longo do curso, agradeço pela amizade e convivência durante essa jornada.

Aos colegas de trabalho, meu agradecimento pelo apoio na realização deste trabalho.

Muito obrigado!

RESUMO

O trabalho consiste na construção de modelos autoencoder para utilização na predição de propensão diária à realização de empréstimo pessoal. O intuito é identificar os clientes mais propensos para direcionar esforços de marketing, como ligações e anúncios em mídias digitais. A realização dessa predição de forma diária é mais adequada à dinâmica de uma Central de Atendimento, por exemplo, que realiza milhares de ligações diariamente. Porém, devido à alta quantidade de dados relacionados ao histórico de ligações, primeiramente um autoencoder foi utilizado para redução de dimensionalidade dessa base, reduzindo o número de variáveis de 120 para 7. Esse procedimento também foi realizado empregando a técnica de Análise de Componentes Principais (PCA) para comparação. Com os dados obtidos nesta etapa e a adição de novas variáveis, como saldo a vencer e limite, um novo autoencoder com 21 variáveis de entrada foi empregado com o objetivo de predizer a propensão de clientes a realização do empréstimo, com todas as predições geradas em um modelo do tipo *Light Gradient Boosting Machine* (LGBM). O conjunto de dados utilizado é anonimizado e proveniente de uma base de dados histórica de uma instituição financeira, sendo altamente desbalanceado, com casos de empréstimos muito mais raros do que casos sem empréstimo (0,22%). Devido à alta diferença entre as classes inerente ao problema, a principal métrica escolhida para avaliação dos modelos foi a área sob a curva de precisão-revocação (PR AUC), pois ela é menos propensa a superestimar o desempenho do modelo quando a classe negativa é predominante. Na etapa de redução de dimensionalidade os resultados obtidos foram muito satisfatórios, com melhor poder preditivo do autoencoder frente ao PCA nas variáveis geradas. Na etapa final de predição, novamente o autoencoder demonstrou superioridade em relação ao PCA, registrando um PR AUC de 0,0194 contra 0,0131. Nesta etapa, os resultados foram inferiores aos alcançados ao utilizar todas as 21 variáveis disponíveis (0,0282), sugerindo que mesmo assim houve alguma perda de informação. Apesar disso, a estratégia de modelo diário se mostrou mais eficiente que a estratégia mensal utilizada atualmente pela instituição em todas as abordagens, evidenciando a qualidade das informações e métodos utilizados. De forma geral, o autoencoder se mostrou uma ferramenta muito útil que pode ser utilizada para o pré-processamento dos dados e fornecimento de informações relevantes para a predição, principalmente em sistemas com alta dimensionalidade.

Palavras-chave: *Autoencoder*, Empréstimo Pessoal, Modelo de Propensão

ABSTRACT

This work consists of constructing autoencoder models for use in predicting daily propensity for personal loan uptake. The aim is to identify the most inclined customers to direct marketing efforts, such as phone calls and digital media advertisements. Performing this prediction on a daily basis is more suitable for the dynamics of a Call Center, for example, which makes thousands of calls daily. However, due to the high amount of data related to call history, initially, an autoencoder was used to reduce the dimensionality of this dataset, reducing the number of variables from 120 to 7. This procedure was also performed using Principal Component Analysis (PCA) for comparison. With the data obtained in this step and the addition of new variables, such as current balance and limit, a new autoencoder with 21 input variables was employed to predict the propensity of clients to take out loans, with all predictions generated in a Light Gradient Boosting Machine (LGBM) model. The dataset used is anonymized and comes from a historical database of a financial institution, being highly unbalanced, with loan cases much rarer than non-loan cases (0.22%). Due to the high difference between the classes inherent to the problem, the main metric chosen for evaluating the models was the area under the precision-recall curve (PR AUC), as it is less prone to overestimating the model's performance when the negative class is predominant. In the dimensionality reduction step, the results obtained were highly satisfactory, with better predictive power of the autoencoder compared to PCA on the generated variables. In the final prediction step, again the autoencoder demonstrated superiority over PCA, recording a PR AUC of 0.0194 against 0.0131. In this step, the results were lower than those achieved using all 21 available variables (0.0282), suggesting that there was still some loss of information. Nevertheless, the daily model strategy proved to be more efficient than the monthly strategy currently used by the institution in all approaches, highlighting the quality of the information and methods used. Overall, the autoencoder proved to be a very useful tool that can be used for data preprocessing and providing relevant information for prediction, especially in systems with high dimensionality.

Keywords: *Autoencoder*, Personal Loan, Propensity Model

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

EP - Empréstimo Pessoal

PCA – *Principal Component Analysis*

RNA – Rede Neural Artificial

MSE - *Mean Squared Error*

LGBM - *Light Gradient Boosting Machine*

AP - *Average Precision*

GOSS - *Gradient-based One Side Sampling*

EFB - *Exclusive Feature Bundling*

ROC - *Receiver Operating Characteristic*

AUC - *Area Under the Curve*

PR - *Precision-Recall*

TP - *True Positive*

FP - *False Positive*

FN - *False Negative*

TN - *True Negative*

KS - Kolmogorov-Smirnov

RFE - *Recursive Feature Elimination*

SUMÁRIO

1 Introdução

O setor de crédito pessoal movimenta trilhões de reais todos os anos e exerce influência direta na vida de milhares de pessoas. No Brasil, esse mercado registrou um movimento financeiro de R$ 2,6 trilhões em 2021, representando um aumento de quase 20% em relação a 2020 (INSTITUTO PROPAGUE, 2021). A busca por esse tipo de crédito tem como principal finalidade a quitação de dívidas em aberto, o que impacta um grande contingente da população brasileira, considerando que 78,3% das famílias possuem dívidas a serem pagas, e dentro desse grupo, 29,1% estão em atraso (CNC, 2023).

Uma das principais frentes desse mercado são os empréstimos pessoais (EPs), modalidade de crédito destinada a pessoas físicas na qual uma quantia em dinheiro é disponibilizada tendo como contrapartida a aplicação de taxas de juros mensais previamente estabelecidas em contrato. Modelos de propensão a compra desse produto são muito interessantes para as instituições financeiras concedentes de crédito, pois ajudam a direcionar os esforços de canais de comunicação, como publicidade online e operação de ligações. Esses estabelecimentos, caracterizados por uma ampla base de clientes, se beneficiam com a identificação dos clientes com maior propensão, que aumenta o retorno sobre investimento (ROI) e impacta diretamente na otimização dos resultados financeiros, uma vez que tempo e mão de obra são recursos escassos e finitos.

Muitas organizações já adotam algum modelo com esse tipo de objetivo, porém em sua forma mais simplificada, onde há um índice, denominado *score* de propensão (variando de 0 a 1, com 1 sendo muito propenso) que é gerado uma vez ao mês com dados estáticos dos meses anteriores, por exemplo. Embora essa seja uma alternativa válida, essa atualização mensal segue uma dinâmica diferente da Central de Atendimento, que realiza milhares de ligações diariamente. Logo, seria interessante a atualização diária dessa propensão levando em conta, por exemplo, valores a vencer do cliente no mês até então, e o próprio resultado de ligações anteriores (se o cliente atendeu, se solicitou retorno após proposta, entre outros).

Essas informações podem ser muito ricas, pois permitem a identificação de um cliente que foi contatado no dia anterior e manifestou desinteresse em um empréstimo, por exemplo. Isso possibilita direcionar os esforços de contato para outros clientes mais propensos, evitando uma nova abordagem a aqueles que já expressaram sua falta de interesse, economizando recursos da empresa e melhorando a experiência do cliente ao evitar abordagens repetitivas e potencialmente incômodas. Essas variáveis não seriam consideradas em um modelo com atualização mensal. O objetivo final é que a inclusão desses dados em uma atualização diária do escore aumente a separação realizada pelo modelo entre propenso e não-propenso e alavanque ainda mais os resultados financeiros.

No entanto, devido à grande disponibilidade de dados relativos ao histórico de ligações, surge a necessidade de uma redução de dimensionalidade, devido à inviabilidade computacional do uso de todos os dados em sua forma bruta. Além disso, ela diminui a complexidade dos dados, o que auxilia na prevenção de sobre ajuste (*overfitting*) no modelo final. Portanto, este trabalho irá realizar técnicas de redução de dimensionalidade visando reduzir o tamanho da base de dados do histórico de ligações, buscando identificar uma representação de baixa dimensionalidade que consiga captar o máximo de informação da base original. A determinação da abordagem a ser adotada neste caso é de suma importância, pois ao mesmo tempo em que reduzir a dimensionalidade traz as vantagens

já citadas, também pode acarretar em perda de informações significativas e assim está intrinsicamente relacionada com a qualidade dos resultados obtidos. O autoencoder é uma técnica que já demonstrou ser eficaz na redução de dimensionalidade em diversos contextos (ARDELEAN, COPORÎIE, *et al.*, 2023), pois se trata de uma arquitetura de rede neural que apresenta a habilidade de aprender e extrair características dos dados, realizando transformações não lineares para gerar uma representação compacta e resumida dos dados de entrada.

Assim, as variáveis geradas nesta etapa serão agregadas a informações adicionais, como saldos pendentes e limites de crédito, para prever a propensão dos clientes a solicitar empréstimos pessoais. Essa predição será realizada tanto com as variáveis geradas no espaço latente do autoencoder quanto considerando a inclusão do erro de reconstrução de um autoencoder treinado exclusivamente com amostras de classe negativa, atuando como um identificador de comportamento, para verificar o poder preditivo dessa informação.

Desta forma, o objetivo deste trabalho é utilizar autoencoders para todo o desenvolvimento do projeto, inicialmente realizando redução de dimensionalidade das variáveis relacionadas ao histórico de ligações, e após utilizá-las em conjunto com outras informações, incluindo o erro de reconstrução, para prever a propensão de clientes realizarem empréstimos pessoais de forma diária. Para ambas as etapas, outras técnicas da literatura serão propostas a fim de comparar seu desempenho com o do autoencoder e validar a adequação dessa técnica para as aplicações em questão.

2 Revisão Bibliográfica

Os conceitos teóricos envolvidos neste trabalho foram revisados na literatura e serão apresentados nesta seção, com destaque para as técnicas de redução de dimensionalidade, o modelo e as métricas de avaliação utilizadas.

2.1 Técnicas de redução de dimensionalidade

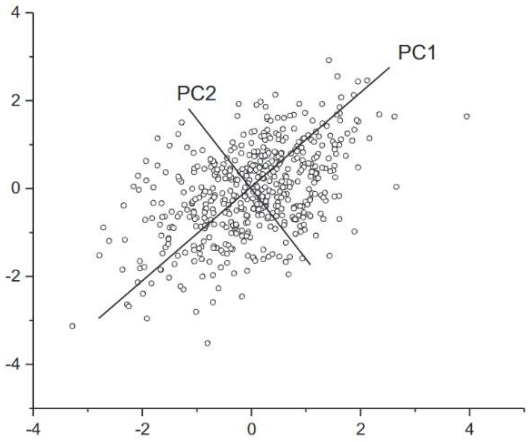
Alguns pontos de atenção devem ser observados quando se deseja trabalhar com bases de dados com muitas variáveis, como por exemplo o custo de armazenamento e principalmente de processamento destes dados (conhecidos como dados de alta dimensionalidade). A redução da dimensionalidade surge como um recurso fundamental no processo de reconhecimento de padrões (JIA, SUN, *et al.*, 2022), visando descobrir o melhor mapeamento dos dados em um número menor de dimensões ao mesmo tempo que minimiza a perda de informação no processo.

A técnica de redução de dimensionalidade que será utilizada é a extração de variáveis (*feature extraction*), que gera novas variáveis a partir das originais. Sua vantagem é que a compactação é mais eficiente, porém, enquanto o conjunto de variáveis original tem um significado evidente, as novas podem perder interpretabilidade (JIA, SUN, *et al.*, 2022). A seguir, duas alternativas para esta técnica serão abordadas.

2.1.1 PCA - *Principal Component Analysis*

A análise de componentes principais, conhecida como *Principal Component Analysis* (PCA) em inglês, é um método não-supervisionado amplamente utilizado para reduzir a dimensionalidade de conjuntos de dados, atribuído a (PEARSON, 1901). Ele funciona extraindo as informações relevantes de um conjunto de dados e as expressando como um novo conjunto de variáveis ortogonais entre si. Essas novas variáveis são combinações lineares das originais denominadas componentes principais.

Figura 2.1: Exemplo de análise de componentes principais (PC)



Fonte: Adaptado de (BJÖRKLUND, 2019)

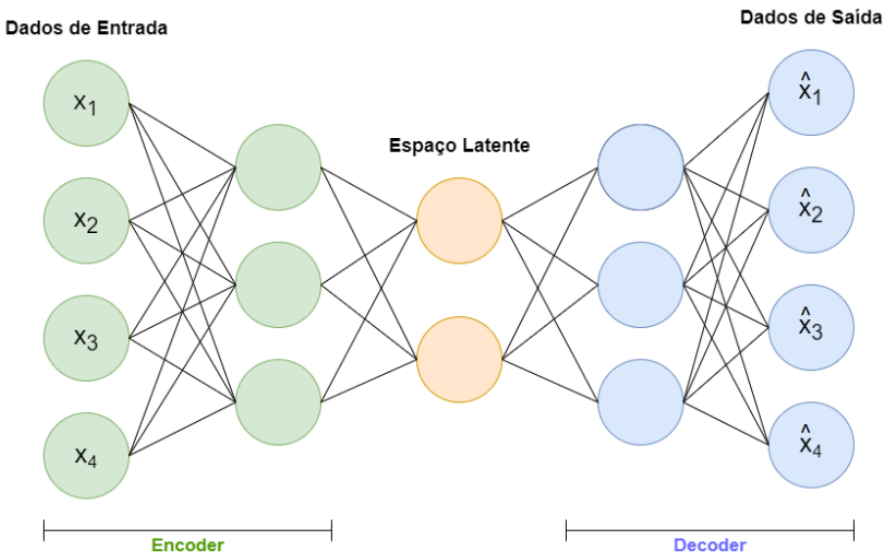
O número de componentes selecionados nessa transformação pode ser determinado de diversas formas, como por exemplo, considerando o número de componentes principais em que a soma da variância explicada acumulada seja de no mínimo 95%. Essa escolha é realizada com o objetivo de minimizar a perda de informações, ao mesmo tempo em que se reduz a dimensionalidade do problema.

A capacidade dos componentes principais em explicar a variância dos dados, eliminar informações redundantes e filtrar ruído propicia seu uso em diversas aplicações, principalmente na etapa de pré-processamento de um problema supervisionado. Entretanto, este método possui algumas limitações, como o fato de não ser capaz de identificar relações não-lineares, e que, por ser um método não-supervisionado, pode descartar informações relevantes para um eventual modelo supervisionado, já que a direção com maior variância não necessariamente é a direção com maior informação.

2.1.2 Autoencoder

Autoencoders são uma técnica de aprendizado de máquina não supervisionado composto por uma rede neural artificial (RNA) que é treinada para reconstruir os dados originais a partir de uma representação de espaço codificado. As RNAs são modelos computacionais inspirados no sistema nervoso central e são compostos por nós (ou neurônios) interconectados que podem aprender a realizar tarefas complexas devido a sua capacidade de realizar transformações não-lineares. Os autoencoders são um tipo especial de rede neural que pode ser usado para reduzir a dimensionalidade de dados, onde o objetivo é reconstruir a entrada ao invés de prever alguma variável alvo. Ao reconstruir as entradas, um autoencoder tenta aprender uma representação condensada dos dados de entrada, processo também conhecido como codificação (*enconding*).

Figura 2.2: Diagrama dos componentes do autoencoder



Fonte: Do Autor

Um autoencoder é composto por:

1) Uma camada de entrada com um vetor de M dimensões que representa os dados de entrada, denotada por $x = (x_1, x_2, \dots, x_M)$.

2) Uma camada de saída, denotada pelo vetor $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_M)$.

3) Uma ou mais camadas ocultas que visam aprender os padrões nos dados de entrada e “codificar” ou “decodificar” as informações essenciais.

4) Uma camada latente que separa o processo de *encoder* e *decoder* e representa a dimensão latente.

Na Figura 2.2 é possível visualizar esses componentes de forma simplificada. As camadas de entrada, chamadas de codificador ou *encoder*, mapeiam os dados de entrada para uma representação de dimensionalidade inferior. Já as camadas de saída (decodificador ou *decoder*) mapeiam a representação de dimensionalidade inferior de volta para a dimensão dos dados originais. A camada intermediária, conhecida como espaço latente, é a representação de menor dimensionalidade que contém informações relevantes dos dados originais, e essa representação constitui uma técnica de extração de variáveis (ARDELEAN, COPORÎIE, *et al.*, 2023).

O autoencoder é treinado visando minimizar o erro entre as reconstruções e os dados originais, nesse trabalho sendo utilizado o erro quadrático médio (MSE, do inglês *Mean Squared Error*). O MSE mede a distância quadrada média entre o valor real (y) e as reconstruções ($\hat{y}$), e possui a vantagem de punir erros maiores ao elevar ao quadrado, ao mesmo tempo que garante que o valor obtido seja sempre maior ou igual a zero. Logo, quanto menor, melhor.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.1}$$

Os erros de reconstrução representam dados de relevância em uma variedade de aplicações, incluindo detecção de falhas, por exemplo. Além disso, a sua utilização como variável pode fornecer informações significativas em tarefas de predição (TELLAECHE IGLESIAS, ÁNGEL CAMPOS ANAYA, *et al.*, 2021). Essa estratégia se fundamenta na suposição de que as amostras da classe minoritária positiva demonstram padrões distintos em comparação ao resto. Isso implica que, ao treinar um autoencoder apenas com instâncias negativas, elas também podem apresentar uma taxa de erro de reconstrução maior. A inclusão do erro como variável tem como objetivo analisar a veracidade dessa suposição, explorando até que ponto esses erros podem contribuir para a capacidade preditiva do classificador.

Por ser uma rede neural, o autoencoder pode ser customizado tanto em número de camadas quanto de neurônios, além de apresentar hiperparâmetros e outras funções como a de otimização e de ativação das camadas a serem definidas. A escolha de cada uma delas será discutida na seção 4.

2.2 Modelo

2.2.1 Light Gradient Boosting Machine - LGBM

Light Gradient Boosted Machine, ou LightGBM, é uma implementação da técnica de *gradient boosting* em árvores de decisão introduzida por (KE, MENG, *et al.*, 2017). *Gradient boosting* é um algoritmo de aprendizado de máquina que visa reduzir o viés nas predições através do treino sequencial de modelos aprendizes fracos, combinando-os em um aprendiz forte. Cada novo modelo é treinado para minimizar a função de perda do modelo anterior utilizando o método do gradiente, sendo possível sua utilização tanto em regressão quanto em classificação.

A métrica de avaliação utilizada foi a precisão média (*average precision*, AP), pois é uma opção já disponível no Python que resume a curva de precisão-revocação utilizando a média ponderada das precisões alcançadas em cada limite (P_n), com o aumento na revocação (R_n) do limite anterior usado como peso para cada limite n.

$$AP = \sum_n (R_n - R_{n-1})\, P_n \tag{2.2}$$

O diferencial do LGBM em relação a outros modelos que utilizam deste mesmo algoritmo são as técnicas de *Gradient-based One Side Sampling* (GOSS) e *Exclusive Feature Bundling* (EFB). GOSS parte do princípio que diferentes instâncias de dados possuem diferentes papéis no cálculo do ganho de informação. Ele mantém as instâncias de dados com maiores gradientes, que irão contribuir mais para o ganho de informação, e descarta aleatoriamente algumas instâncias com baixos gradientes. Assim, consegue preservar a acurácia da estimativa do ganho com um conjunto de dados menor que o original, o que é relevante principalmente quando o valor do ganho de informação varia muito dentro de um mesmo conjunto de dados.

Já a técnica de EFB foi projetada para lidar com dados de alta dimensionalidade, que geralmente são muito esparsos e assim permitem reduzir o número de variáveis com praticamente zero perda. Em um espaço de características esparsas, muitas dessas características são mutuamente exclusivas, ou seja, nunca assumem valores não nulos simultaneamente. As características exclusivas podem ser agrupadas com segurança em uma única característica chamada de *Exclusive Feature Bundle*. Isso resulta em uma melhoria na velocidade de treinamento sem comprometer a precisão (KE, MENG, *et al.*, 2017).

2.3 Métricas de avaliação

A escolha da métrica utilizada para avaliação dos modelos desempenha um papel crucial tanto na orientação da modelagem quanto na avaliação da performance. A base utilizada é extremamente desbalanceada, pois o número de clientes que recebem ligações e não realizam empréstimo é muito maior que o oposto. Logo, essa escolha é ainda mais crucial porque os procedimentos padrão e relativamente robustos utilizados usualmente para classificação binária com dados balanceados, como ROC AUC (*Receiver Operating Characteristic* e *Area Under the Curve*), podem se mostrar altamente inadequados (HE e MA, 2013), resultando em modelos de classificação subótimos que induzem a conclusões enganosas, uma vez que estas medidas são insensíveis a domínios de dados não balanceados (BRANCO, TORGO e RIBEIRO, 2015).

2.3.1 Área sob a curva de Precisão-Revocação – PR AUC

Para análise do desempenho de modelos de classificação binária, diversas métricas utilizam um instrumento conhecido como Matriz de Confusão (Figura 2.3). Esta matriz ilustra os resultados da classificação, e é composta pelos os seguintes elementos:

- **Verdadeiros Positivos** (TP): São as previsões em que o modelo identificou corretamente os exemplos positivos.
- **Falsos Positivos** (FP): Representam as previsões em que o modelo classificou incorretamente exemplos negativos como positivos.
- **Falsos Negativos** (FN): Refletem as previsões em que o modelo identificou incorretamente os exemplos positivos como negativos.
- **Verdadeiros Negativos** (TN): São as previsões em que o modelo classificou corretamente os exemplos negativos.

Figura 2.3: Matriz de confusão

		Valor Predito	
		Negativo	**Positivo**
Valor Real	**Negativo**	Verdadeiro Negativo (TN)	Falso Positivo (FP)
	Positivo	Falso Negativo (FN)	Verdadeiro Positivo (TP)

Fonte: Do Autor

Após a compreensão da Matriz de Confusão, torna-se possível definir dois parâmetros essenciais na avaliação do modelo:

- **Precisão** (*Precision*): dentre todas as classificações de classe positiva que o modelo fez, quantas estão corretas.

$$Precisão = \frac{TP}{TP+FP} \tag{2.3}$$

- **Revocação** (*Recall*): dentre todas as situações de classe positiva esperadas, quantas estão corretas.

$$Revocação = \frac{TP}{TP+FN} \tag{2.4}$$

As métricas discutidas acima são empregadas quando se deseja minimizar o número de erros, já que quantificam essa métrica. No entanto, o cálculo desse erro só pode ser realizado ao impor a condição de que a predição seja expressa no formato binário de 0 ou 1 (em um modelo de classificação binária). Assim, podem ser chamadas de métricas de limite, onde a partir de um determinado valor limiar de probabilidade predita (usualmente 0,5 por padrão), a classificação será atribuída como 1; caso contrário, será classificada como 0. Uma desvantagem importante das métricas de limite é que elas pressupõem conhecimento das condições sob as quais o classificador será implantado, assumindo que o desequilíbrio de classes presente no conjunto de treinamento é aquele que será encontrado ao longo da vida operacional do classificador, pois o limite permanece o mesmo (HE e MA, 2013). Conforme se altera esse limite, também se alteram as classificações preditas das amostras e consequentemente os valores advindos da matriz de confusão.

Uma alternativa às métricas de limite são as métricas de ranqueamento, que estão mais preocupadas em avaliar classificadores com base em quão eficazes eles são na separação de classes. Elas podem ser utilizadas onde se deseja selecionar as melhores *n* instâncias de um conjunto de dados ou quando uma boa separação de classes é crucial, por exemplo. A área sob a curva de precisão-revocação (PR AUC) é uma métrica ideal para uso em classificação binária com classes altamente desbalanceadas (COOK e RAMADAS, 2020), pois testa todos os possíveis valores de limite para gerar a curva de precisão em função da revocação. Essa análise foca especialmente na variação da classe minoritária positiva devido à definição desses indicadores. Por não incorporar verdadeiros negativos, essa avaliação é menos propensa a superestimar o desempenho do modelo em conjuntos de dados onde a classe negativa é predominante. Portanto, se um método tem PR AUC maior do que outro, isso sugere que ele é capaz de manter melhor equilíbrio entre precisão e

revocação independente do limite escolhido, obtendo melhor desempenho na capacidade de distinguir as classes.

Como na maioria das métricas de avaliação, é desafiador estabelecer um valor específico considerado bom ou aceitável que seja aplicável em todas as circunstâncias, geralmente sendo utilizadas como um método de comparação entre diferentes abordagens. No entanto, existem algumas considerações que podem ser úteis na avaliação. Por exemplo, um classificador aleatório, sem nenhum tipo de discriminação entre as classes, obteria um valor de PR AUC equivalente a porcentagem da classe minoritária (COOK e RAMADAS, 2020).

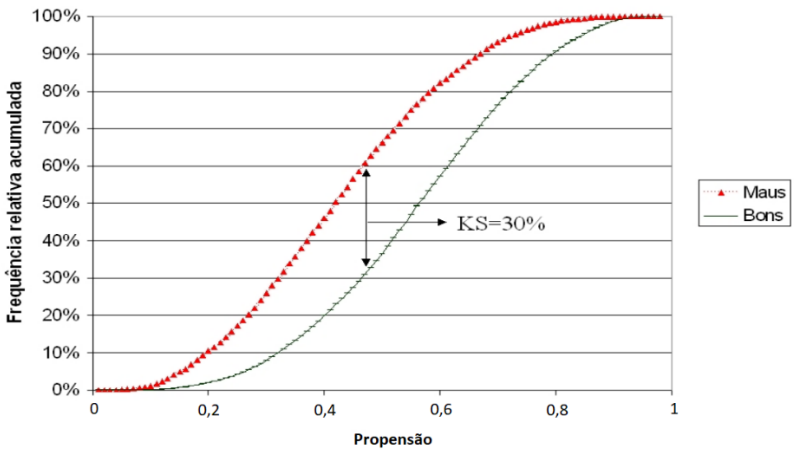
2.3.2 Estatística de Kolmogorov-Smirnov - KS

A estatística de Kolmogorov-Smirnov (KS) é uma métrica para modelos de classificação binária amplamente empregada na avaliação de modelos de escores de crédito. Na estatística não paramétrica, o KS é utilizado para verificar se duas amostras podem ser originadas de uma mesma função de distribuição (CONOVER, 1999). Como métrica de avaliação, se fundamenta na noção da distância entre as distribuições de probabilidades das duas classes, neste caso clientes que efetuaram e que não efetuaram compra. O KS quantifica a maior diferença entre a frequência relativa acumulada dos clientes "maus" (que não fizeram empréstimo), $F_m(s)$, e a frequência relativa acumulada dos clientes "bons" (que fizeram empréstimo), $F_b(s)$. O objetivo é avaliar se as duas amostras de clientes vêm de populações distintas, fornecendo evidências que o modelo está atingindo sua meta de distinguir os dois grupos. A estatística de KS para testes unilaterais, onde a função distribuição dos maus clientes é maior que a dos bons, é definida como:

$$KS = \max_{s}\{F_m(s) - F_b(s)\} \tag{2.5}$$

Em um modelo de bom desempenho, clientes ruins tendem a receber escores mais baixos, enquanto clientes bons recebem escores mais altos. Assim, observa-se que $F_m(s)$ rapidamente se aproxima de 1, enquanto $F_b(s)$ permanece próximo de 0 para um maior número de valores de escore *s*. Consequentemente, quanto mais rápido o crescimento de $F_m(s)$ e quanto mais lento o crescimento de $F_b(s)$, melhor é a qualidade do modelo. É possível observar esse comportamento na Figura 2.4.

Figura 2.4: Exemplo de cálculo da estatística de Kolmogorov-Smirnov



Fonte: Adaptado de (PEREIRA, 2004)

Apesar do modelo em questão não tratar especificamente de escores de crédito, essa métrica pode ser útil para verificar o poder de separação entre as classes. Porém, como é considerado apenas o valor máximo, é importante avaliar o resultado de KS obtido junto com outras métricas para garantir que o modelo não está discriminando clientes bons de ruins apenas em uma certa faixa de escore. Outra análise possibilitada por esse indicador é a área sob a curva dos valores de KS ordenados pelo escore (KS AUC), onde é possível avaliar como a diferença entre as distribuições evolui com o escore. Assim como o PR AUC, tanto o KS como o KS AUC variam de 0 a 1 com valores altos indicando melhor performance.

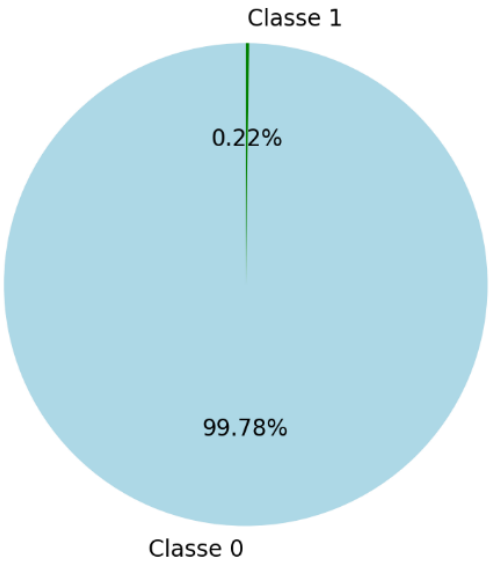
3 Formulação do Problema

Os dados utilizados pertencem a uma instituição financeira, com mais de 3 milhões de instâncias referentes a três meses, ou 92 dias. Atualmente, a instituição já possui um modelo de propensão que gera um escore mensal utilizando informações do histórico dos clientes. Esse escore é utilizado para gerar uma lista de priorização de ligações, onde os mais propensos são priorizados, e até possui um desempenho aceitável quando avaliado mensalmente. No entanto, o grande desafio consiste em possibilitar a atualização diária desse escore, visando gerar resultados mais alinhados com a dinâmica de uma Central de Atendimento. Assim, a ideia é que o novo modelo diário usufrua do escore mensal previamente gerado com a adição de variáveis relacionadas a dois principais grupos: aos valores atuais de compra e limite do mês até então ao histórico de ligações, ambas não presentes no escore mensal atual.

A base de dados referente ao histórico de ligações, na qual será realizada a redução de dimensionalidade, abrange todos os registros de chamadas efetuadas em um período de até 30 dias anteriores ao dia em análise. Para se adequar ao mesmo formato das demais variáveis, essas chamadas foram agrupadas em quatro categorias relacionadas à classificação das ligações. Cada chamada é classificada com base no seu resultado, e seguindo a classificação comumente adotada no mercado, uma chamada pode ser classificada como tendo "alô" quando é atendida por alguém e como tendo "alô efetivo" quando ocorre de fato comunicação com o cliente esperado. É importante mencionar que uma chamada pode ser atendida, mas não resultar em contato efetivo, se a ligação cair ou se o cliente alegar engano, por exemplo. Além disso, foi criada uma categoria adicional dentro das chamadas classificadas como "alôs efetivos", abrangendo situações altamente indicativas de venda, como promessas de compra, por exemplo.

Assim, os quatro grupos formados são: ausência de alô, presença de alô sem efetividade, alô efetivo sem constar nas categorias especiais e alô efetivo constando nas categorias especiais. Esses quatro grupos foram acumulados dia-a-dia ao longo de 1 a 30 dias anteriores, resultando em um total de 120 variáveis.

Figura 3.1: Proporção entre classes na base de dados analisada.



Fonte: Do Autor

Existe forte disparidade entre classes, conforme observável na Figura 3.1, com casos de empréstimos muito mais raros do que casos sem empréstimo. Um cliente foi marcado como classe positiva se, no dia analisado, recebeu uma ligação com contato efetivo e realizou um empréstimo pessoal em até 10 dias (período considerado pela instituição), e classe negativa caso contrário. Essa abordagem retira da base clientes que fizeram EP mas não tiveram influência dos canais de comunicação, o que ajuda a melhor identificar aqueles que realizaram empréstimo total ou parcialmente devido à atuação da Central de Atendimento, assim maximizando o ganho monetário do modelo já que ele atuaria principalmente nesse canal. Essa base foi separada em uma amostra de treino, com 2 milhões de casos, e de teste, com 1 milhão e 333 mil casos. Todo tratamento de dados e criação de modelos foi feito utilizando a linguagem *Python* na versão 3.9.13, e os principais pacotes utilizados foram pandas (versão 2.0.3), numpy (versão 1.21.6), pytorch (versão 2.1.0), scikit-learn (versão 1.3.2) e LightGBM (versão 4.1.0).

4 Metodologia

4.1 Redução de dimensionalidade

Como citado na seção anterior, os dados relacionados ao histórico de ligações inicialmente possuem 120 variáveis. Por causa desse alto número, antes do PCA ou o autoencoder serem utilizados para redução da dimensionalidade, foi feita uma seleção inicial de variáveis baseada na importância dada a elas por sete diferentes métodos: correlação linear, limite de variância, regularização Lasso, floresta aleatória, ensemble, eliminação recursiva de variáveis (RFE) e permutação. Para cada um desses métodos, foi fornecido o conjunto de dados completo com 120 variáveis e as mais relevantes foram selecionadas. Cada técnica está resumida abaixo:

- **Correlação linear**: a análise de correlação estabelece a relação linear entre cada variável e a classe, medida pela magnitude do coeficiente de Pearson. Dessa forma, as 20 variáveis que possuem maior correlação linear com o alvo foram selecionadas.

- **Limite de variância**: o método de limite de variância estabelece um critério para descartar variáveis que possuem uma variância abaixo de um determinado limite. Essa abordagem filtra variáveis com baixas variações, que tendem a ser menos informativas para o modelo. O limite estabelecido foi de 0.1, o que significa que as variáveis onde 10% dos valores são similares foram retiradas, com 61 restantes.

- **Regularização Lasso**: é uma técnica que introduz uma penalização proporcional a soma dos valores absolutos dos coeficientes no processo de otimização, no caso em questão utilizado em uma regressão logística. Esta penalização visa reduzir os coeficientes a valores baixos ou zero, assim reduzindo a probabilidade de sobre ajuste e a aumentando a interpretabilidade do modelo (TIBSHIRANI, 1996). Assim, as 20 variáveis com maior importância para esse modelo foram selecionadas.

- **Floresta aleatória**: as florestas aleatórias (*Random Forests*) são um tipo de algoritmo de aprendizado por agrupamento (*ensemble learning*) baseado em árvores de decisão. As 20 variáveis com maior importância para esse modelo foram selecionadas.

- ***LightGBM***: foi utilizado um classificador LGBM onde as 20 variáveis com maior importância para esse modelo foram selecionadas.

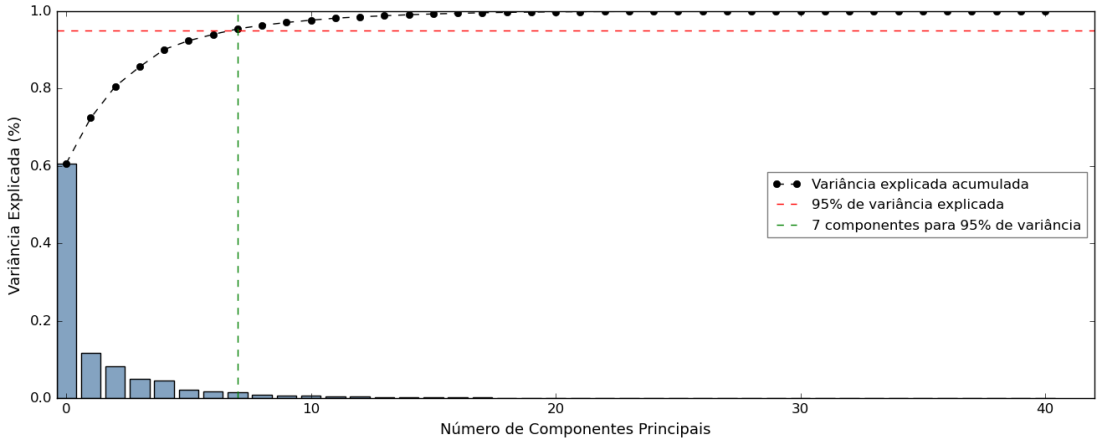
- **Eliminação recursiva de variáveis**: a eliminação recursiva de variáveis (RFE, do inglês *Recursive Feature Elimination*) é uma abordagem que consiste em treinar um modelo com todas as variáveis, identificar a importância de cada uma remover a menos importante, repetindo esse processo até atingir o número desejado, definido como 20. O modelo especificado a ser utilizado para essa eliminação foi um LGBM.

- **Permutação**: o método de Permutação avalia a importância das variáveis ao realizar permutações aleatórias nos valores de cada variável e observando como isso afeta o desempenho do modelo. As 20 variáveis cuja permutação tem maior impacto no desempenho do modelo foram selecionadas.

Todas as variáveis que foram selecionadas por pelo menos dois métodos foram escolhidas para permanecer no modelo, restando 41 variáveis ao final desta seleção inicial.

Desta forma, a camada de entrada do autoencoder diminui de 120 para 41 neurônios, o que aumenta a velocidade de treino enquanto mantém as variáveis mais relevantes. Para definição do tamanho da camada intermediária, foi utilizado PCA para verificar o número de componentes principais necessários para explicar 95% da variância do conjunto de dados. Na Figura 4.1 é possível observar que esse número é de 7 componentes.

Figura 4.1: Curva de variância explicada acumulada para a análise de componentes principais da base de histórico de ligações



Fonte: Do Autor

Para a definição do tamanho da segunda camada, foi realizado uma busca exaustiva (*grid search*) com as duas outras camadas já definidas, onde o melhor desempenho foi alcançado com 27 neurônios. Esta avaliação de desempenho foi medida pelo PR AUC das previsões geradas por um modelo LGBM utilizando hiperparâmetros padrão quando aplicado à base de teste após as transformações realizadas pelo autoencoder. Ao final da busca, a arquitetura do autoencoder foi definida como 5 camadas de 41, 27, 7, 27 e 41 neurônios, respectivamente, seguindo a estrutura mostrada na Figura 2.2. Por ser uma rede neural com mais de três camadas, se caracteriza como uma arquitetura de aprendizado profundo (*deep learning*).

Cada camada realiza primeiramente uma transformação linear dos dados de entrada, depois passando por uma função de ativação, neste trabalho definida como logística (sigmoide), expressa pela equação:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{4.1}$$

A função logística restringe a saída a estar no intervalo entre 0 e 1, o que é útil quando se trabalha com probabilidades ou pixels de imagens, por exemplo. Além disso, ela também possibilita o cálculo de relações não-lineares que ajudam na modelagem complexa da realidade. Na verdade, se a função de ativação dentro de cada camada for linear, as variáveis presentes no espaço latente correspondem diretamente aos componentes principais do PCA (PLAUT, 2018).

O algoritmo de otimização aplicado foi o Adam (do inglês *adaptive moment estimation*), método muito utilizado na literatura (WONG e LUO, 2018) que serve como uma extensão do método de descida do gradiente estocástica. A grande diferença entre o Adam e o método do gradiente é que o Adam mantém taxas de aprendizado adaptativas para cada

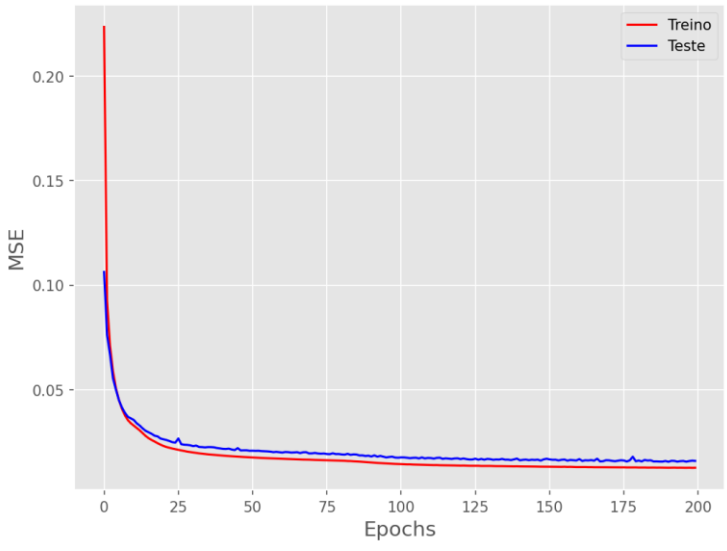
parâmetro da rede. Essa abordagem melhora a eficiência do treinamento, ajustando automaticamente as taxas de aprendizado para diferentes parâmetros, o que tende a resultar em convergência mais rápida e estável.

A definição de alguns hiperparâmetros mais relevantes é necessária para o treinamento do autoencoder. O primeiro é a taxa de aprendizado (*learning rate*), que determina a taxa na qual o modelo atualiza os pesos dos parâmetros a cada iteração durante o treinamento, com valores usualmente entre 0 e 1. Uma taxa de aprendizado mais baixa geralmente resulta em um treinamento mais lento, porém, pode levar a um melhor desempenho do modelo. Neste trabalho, ela foi definida como 0,001.

O tamanho dos lotes (*batch size*) define o número de instâncias de treino que serão utilizados em cada iteração. Já o número de épocas (*epochs*) é o número de passes completos de todas as amostras de treino. Uma época significa que cada amostra no conjunto de treinamento teve a oportunidade de atualizar os parâmetros internos do modelo, composta por um ou mais lotes. Se o tamanho do lote for igual ao número de instâncias de dados, ocorrerá apenas uma iteração por época, caracterizando uma descida do gradiente em lote. Porém, quando o conjunto de dados é muito grande, a descida de gradiente em lote pode ser lenta e exigir um grande consumo de memória. Nesses casos, é vantajoso definir o tamanho dos lotes como um número entre 1 e o tamanho da base, sendo comumente utilizados múltiplos de 2 como 64 ou 128. Neste trabalho, o tamanho dos lotes foi definido como 256, o que significa que ocorreram 7813 iterações a cada época no treinamento.

Para definição do valor de épocas, foi utilizada a técnica conhecida como *early stopping*, onde se monitora os erros de treino e teste através do tempo e se avalia onde estes erros estão estabilizados. O erro no treino tende a diminuir conforme os dados são passados, devido a atualização interna dos parâmetros internos do autoencoder. Na Figura 4.2 é possível ver que a partir de 50 épocas o erro do autoencoder se estabiliza em aproximadamente 0,02, com pequena diferença entre os erros de treino e teste, indicando que não há sobre ajuste significativo.

Figura 4.2: Comparação de erros de treino e teste durante o processo de treinamento do autoencoder para etapa de redução de dimensionalidade



Fonte: Do Autor

Quando os pesos de uma rede neural estão corretamente inicializados, o treinamento e convergência podem ser muito mais rápidos. Para inicialização dos pesos, foi utilizada a inicialização de Xavier, proposta por (GLOROT e BENGIO, 2010). Essa técnica ajusta os pesos iniciais de acordo com uma distribuição normal de variância específica, que leva em consideração o número de entradas e saídas de cada camada. O objetivo é evitar gradientes muito pequenos ou muito grandes, o que poderia desacelerar ou dificultar a convergência do modelo.

As bases de treino e teste foram escalonadas utilizando Standard Scaler, técnica que busca transformar os dados para a mesma escala através da seguinte equação:

$$z = \frac{x - \mu}{\sigma} \tag{4.2}$$

Onde x representa o valor, μ a média e σ o desvio padrão dos dados originais, e z o seu novo valor. Desta forma, a nova variável possui média 0 e desvio padrão 1. Modelos baseados em árvores de decisão, como o LGBM, não são sensíveis a esse tipo de tratamento, pois fazem divisões com base no valor das variáveis, de modo que a sua escala não afeta a capacidade do modelo de encontrar as divisões ideais. Entretanto, em redes neurais como o autoencoder, a escala dos dados importa, visto que se os dados de entrada não estiverem na mesma escala certas variáveis podem dominar o processo de aprendizado. Além disso, ao ter dados padronizados, a função objetivo para o treinamento do autoencoder se torna mais simétrica e regular. Isso geralmente leva a um espaço de busca mais coerente durante o treinamento, facilitando a convergência para uma solução ótima global ao invés de um mínimo local.

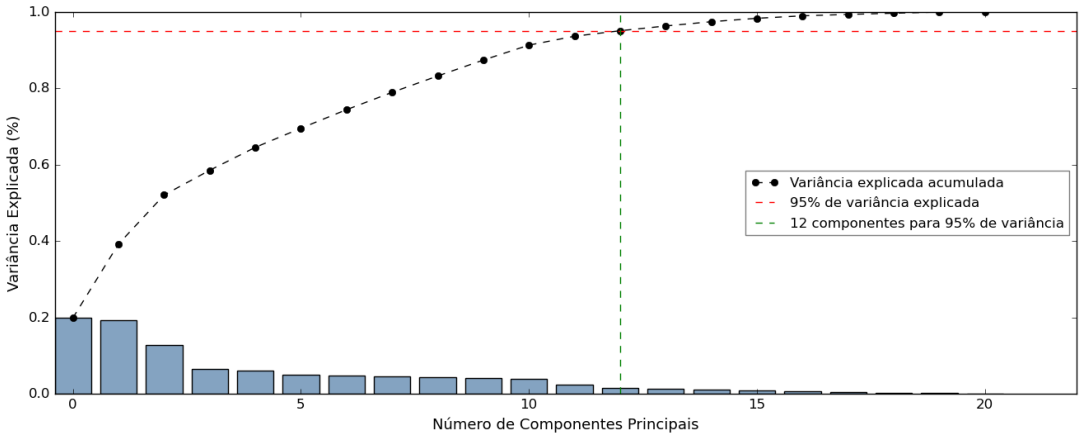
Além da avaliação dos erros de reconstrução, as variáveis geradas pelo PCA e autoencoder foram analisadas quanto a sua capacidade preditiva. Para isso, elas foram utilizadas em um modelo classificador LGBM idêntico com hiperparâmetros padrão e mesma semente aleatória.

4.2 Predição de propensão a compra

Para a previsão de propensão a empréstimos, inicialmente foram acrescidas às 7 variáveis geradas anteriormente mais 20 variáveis relacionadas a saldos a vencer, limites até o momento do mês e o escore mensal, totalizando 27 variáveis. O mesmo processo de seleção inicial descrito anteriormente foi aplicado, resultando, ao final, em 21 variáveis de entrada para esta etapa.

A definição das outras camadas foi realizada com uma busca exaustiva avaliada pelo mesmo procedimento apresentado na subseção anterior, a diferença sendo que dessa vez a busca foi realizada para as combinações das duas camadas a serem definidas. A única limitação imposta foi que, ao considerar que o PCA é capaz de capturar 95% da variância dos dados usando 12 componentes, conforme indicado na Figura 4.3, esse número foi estabelecido como o limite máximo para a camada intermediária do autoencoder. Isso se deve à expectativa de que ele consiga um desempenho superior nesse aspecto por conseguir identificar relações não-lineares.

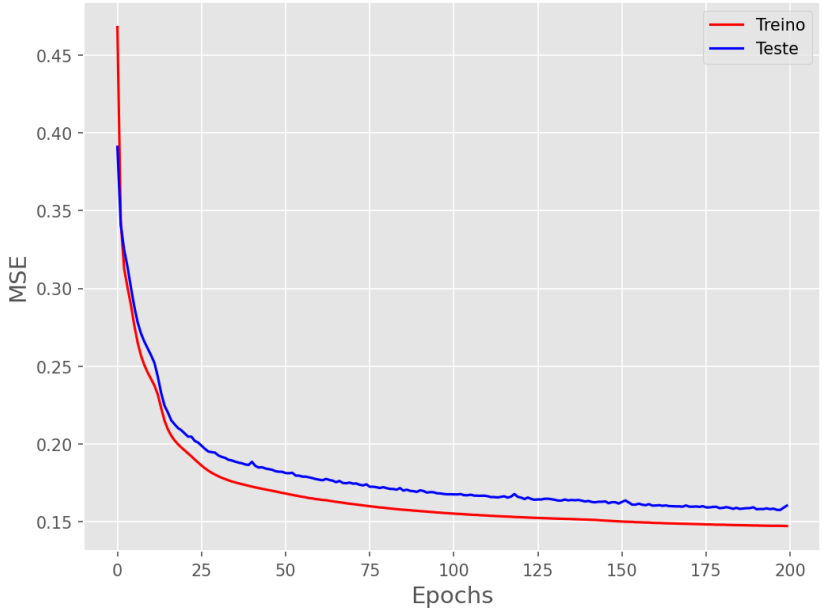
Figura 4.3: Curva de variância explicada acumulada para a análise de componentes principais da base final de predição



Fonte: Do Autor

O melhor resultado foi apresentado pelo autoencoder com 21, 14 e 5 camadas. A inicialização dos pesos, definições do autoencoder, funções de treino, validação e hiperparâmetros do autoencoder permanecem os mesmos da etapa anterior, com a única exceção sendo o número de épocas. Na Figura 4.4 vemos que o autoencoder deste estágio apresentou estabilidade próximo de 100 épocas e apresentou um erro levemente maior que o anterior, estabilizando próximo de 0,16.

Figura 4.4: Comparação de erros de treino e teste durante o processo de treinamento do autoencoder para etapa de predição

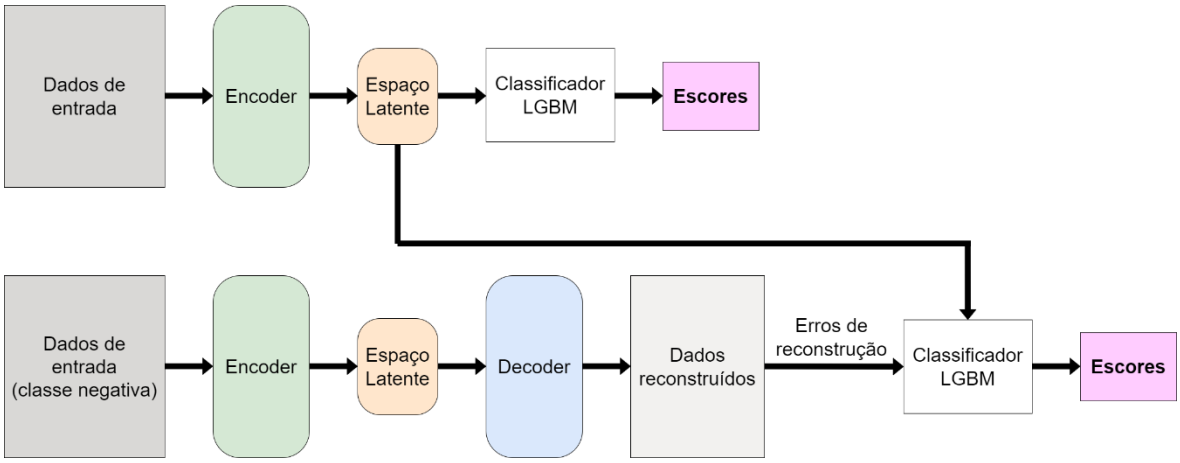


Fonte: Do Autor

Após a definição da estrutura do autoencoder, primeiramente as variáveis do espaço latente foram utilizadas para predição, sendo comparadas com as variáveis geradas pelo PCA com mesma dimensionalidade. Em seguida, os erros de reconstrução obtidos por um autoencoder de mesma estrutura treinado apenas com casos negativos foram utilizados junto às variáveis geradas pelo autoencoder anterior. Essas duas estruturas podem ser

visualizadas na Figura 4.5. Ao final, um modelo utilizando todas as 21 variáveis originais também foi utilizado para contrapor os desempenhos obtidos pelos modelos anteriores.

Figura 4.5: Esquema da abordagem de predição



Fonte: Do Autor

O primeiro procedimento é similar ao realizado para avaliar a capacidade preditiva na etapa anterior, com a diferença das adições das novas variáveis e de uma busca exaustiva nos hiperparâmetros do classificador LGBM, já que desta vez a performance da predição é o resultado de interesse. Essa busca exaustiva foi realizada utilizando uma base de validação, constituída por 20% da base de treino. Os hiperparâmetros testados foram taxa de aprendizado, número de folhas, quantidade mínima de dados em uma folha e fração de variáveis.

A definição de taxa de aprendizado já foi realizada na subseção anterior. O número de folhas refere-se ao número máximo de folhas ou nós terminais em uma árvore de decisão. Controla a complexidade do modelo, influenciando sua capacidade de se ajustar aos dados de treinamento. Valores mais altos podem levar a modelos mais complexos e sujeitos a sobre ajuste. A quantidade mínima de dados em uma folha define o número mínimo de amostras requeridas para formar uma folha (nó terminal) em uma árvore. Isso ajuda a evitar partições em folhas com poucas amostras, contribuindo para a generalização do modelo. Por fim, a fração de variáveis (*feature fraction*) determina a fração aleatória de variáveis a serem consideradas em cada iteração durante a construção da árvore, o que ajuda a controlar a aleatoriedade e a diversidade das árvores no modelo. Ao final da busca, os valores obtidos foram de 0,1, 10, 5 e 1, respectivamente.

Os classificadores LGBM foram treinados com um número máximo de 80.000 árvores. Além disso, foi utilizado o parâmetro de parada antecipada (*early stopping*), configurado para interromper o treinamento caso não sejam observadas melhorias após 800 iterações.

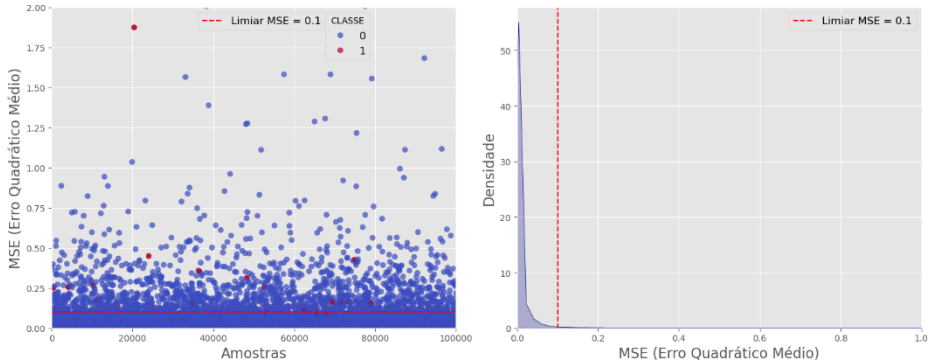
O segundo procedimento da Figura 4.5 busca avaliar a relevância do erro de reconstrução como informação preditiva, assim como a capacidade do autoencoder em distinguir o comportamento entre as duas classes, como descrito na seção 2.1.2.

5 Resultados

5.1 Redução de dimensionalidade

Primeiramente, verificando os erros de reconstrução dos dados de uma amostra de 100 mil casos da base de teste, são obtidos os resultados demonstrados na Figura 5.1. Apenas 1,73% (ou 1730) das amostras ficaram acima de 0,1 de MSE, fronteira indicada pela linha tracejada vermelha, comportamento também evidente na distribuição desse erro, indicando um excelente desempenho na reconstrução. O valor de MSE médio por amostra foi 0,019, também baixo. Casos com classe positiva foram indicados nos pontos em vermelho, mostrando que eles tendem a ter um erro de reconstrução mais afastado de zero, o que, apesar de não ser o objetivo desta etapa, era esperado devido ao seu comportamento distinto.

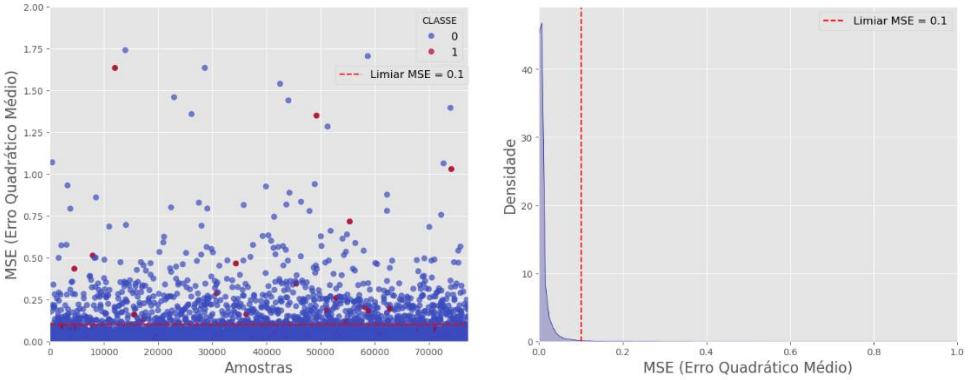
Figura 5.1: Erro quadrático médio (MSE) de reconstrução do autoencoder para a base de teste da etapa de redução de dimensionalidade, por amostra (esquerda) e sua distribuição (direita)



Fonte: Do Autor

Agora, avaliando o desempenho na reconstrução dos dados de entrada de mais de 70 mil amostras totalmente novas, não utilizadas nem para treino nem para teste, são obtidos os erros de reconstrução expressados na Figura 5.2. O comportamento obtido foi extremamente similar, com a porcentagem acima da marca de 0,1 MSE sendo de apenas 1,36%, e o MSE médio por amostra foi de 0,034, validando os resultados obtidos.

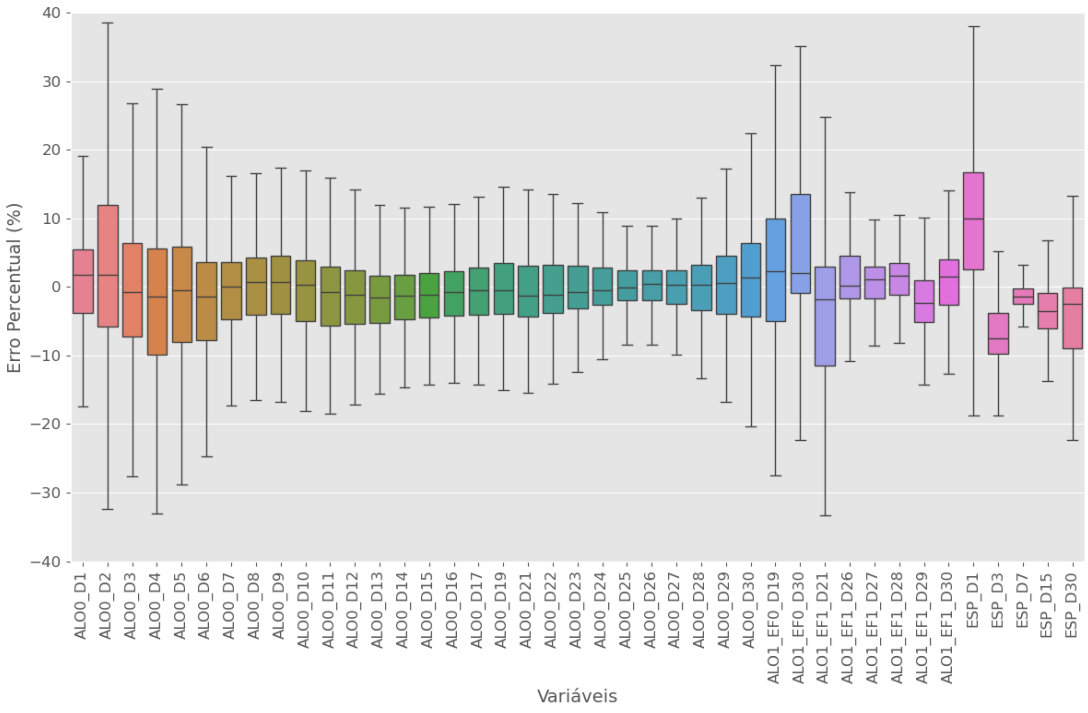
Figura 5.2: Erro quadrático médio (MSE) de reconstrução do autoencoder para base de validação da etapa de redução de dimensionalidade, por amostra (esquerda) e sua distribuição (direita)



Fonte: Do Autor

Na Figura 5.3 estão apresentados os erros de reconstrução individual (em percentual) das 41 variáveis originais da base de teste. Variáveis com maior erro percentual, tanto positivo quanto negativo, representam áreas em que o autoencoder encontrou maior dificuldade em sua reconstrução. Isso indica uma maior variação nos comportamentos dos valores, os quais podem ou não estar relacionados com a classe a que pertencem. Assim, essa visualização ajuda a identificar alguns pontos interessantes na interpretação dos dados utilizados. Por exemplo, no contexto das ligações não atendidas (alô zero), observa-se que a precisão da reconstrução tende a aumentar à medida que o intervalo de tempo considerado se expande. Em outras palavras, o número de ligações próximas ao dia analisado tende a apresentar maior margem de erro, com destaque para o segundo dia que possui o maior intervalo de erro, contrastando com os valores do primeiro dia, que seguem um padrão distinto dos demais no mesmo grupo. Dentro das categorias que envolvem chamadas atendidas, são notáveis as ligações pertencentes ao grupo especial do dia anterior, onde se obteve um erro majoritariamente positivo. Supondo que esse erro esteja ao menos parcialmente relacionado com a diferença de comportamento entre as classes, pode-se inferir que ligações mais recentes tendem a distinguir mais o comportamento entre as duas classes.

Figura 5.3: Erro de reconstrução individual por variável da base de teste

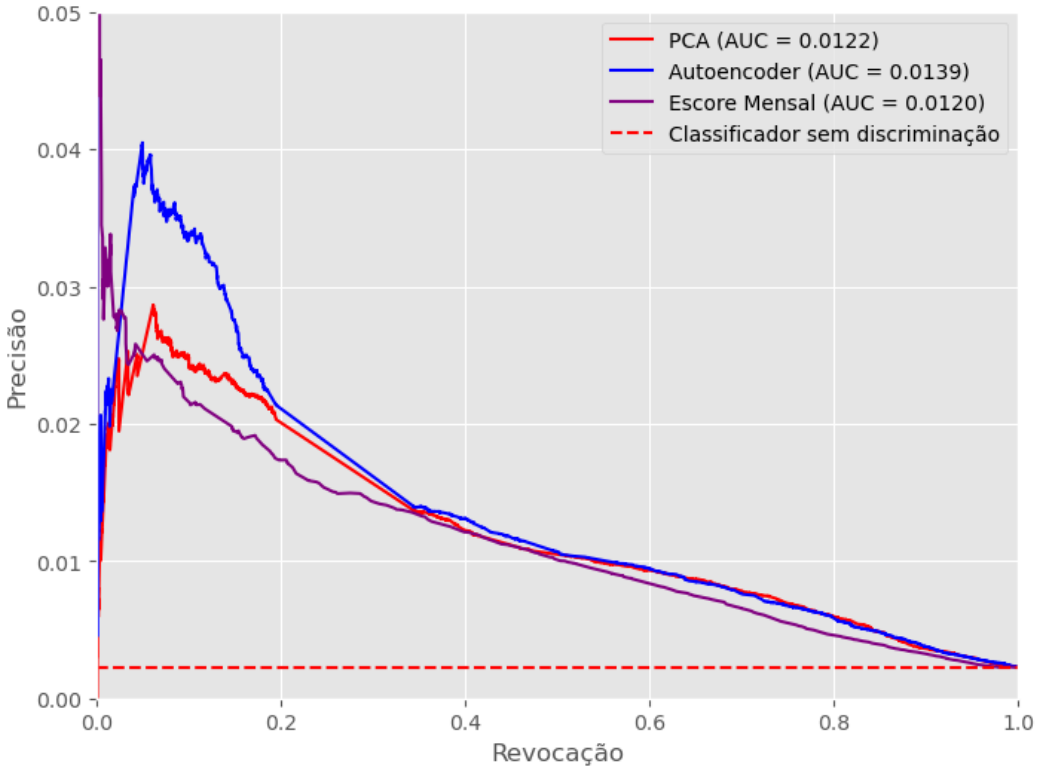


Fonte: Do Autor

Para avaliar as predições dos modelos treinados utilizando as variáveis geradas pelo PCA e autoencoder, na Figura 5.4 constam as curvas de precisão-revocação. Conforme discutido na seção 2.3.1, um classificador aleatório sem discriminação obteria um valor de 0,0022 de área sob a curva de precisão-revocação, seguindo a linha tracejada vermelha. Nessa visão fica claro que ambos os modelos já são capazes de discriminar as duas classes, com o desempenho obtido pelo modelo utilizando as variáveis do autoencoder obtendo um resultado mais de seis vezes maior em valor. É interessante observar que apenas com as variáveis relativas às ligações já é obtido um desempenho melhor que o do escore mensal utilizado atualmente, representado pela curva roxa. Isso atesta a relevância destas

informações para a tarefa de predição, e a capacidade dos métodos utilizados de identificar uma representação de baixa dimensionalidade que consegue captar informações significantes da base original.

Figura 5.4: Curva de precisão-revocação para etapa de redução de dimensionalidade



Fonte: Do Autor

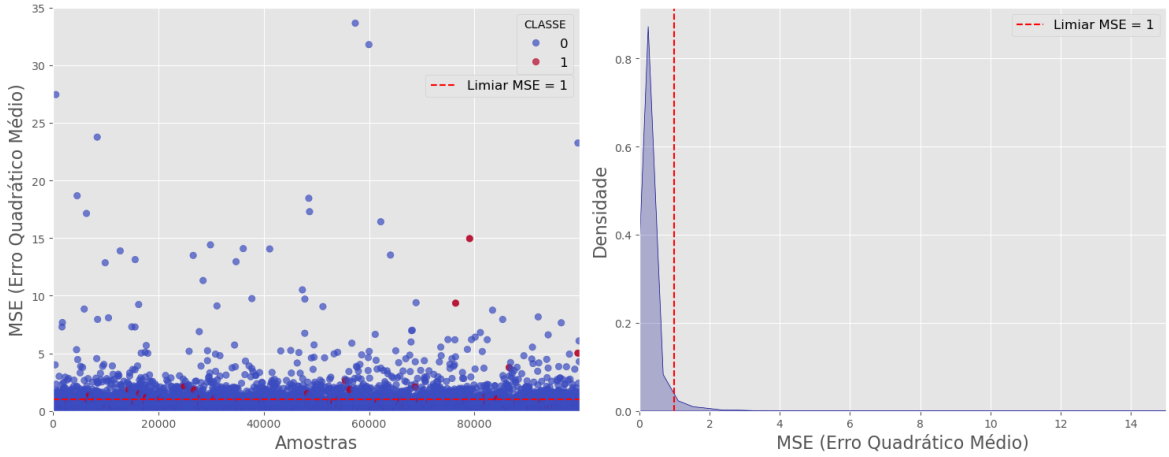
A performance de ambos os modelos foi significativamente melhor na área de menor revocação, indicando maior facilidade na identificação de casos fortemente positivos, com a performance diminuindo para candidatos menos óbvios, o que é coerente. Apesar disso, observa-se uma queda e ascensão da precisão ao longo do gráfico, o que não é comum em gráficos de PR AUC. No entanto, é possível que esse comportamento ocorra, e há outros trabalhos na literatura como o de (COOK e RAMADAS, 2020) que também exibem comportamentos semelhantes. Ao reduzir o limite, a revocação nunca diminuirá, pois só é possível marcar mais exemplos positivos como positivos. Já a precisão está observando todos as instâncias sinalizadas positivamente e, dessas, a fração que é verdadeiramente positiva. Logo, a forma da curva da precisão em termos de limite pode assumir qualquer forma, usualmente tendo alta precisão em limites altos e vice-versa. Na Figura 5.4, observa-se que com um limite de decisão muito próximo de 1 (baixa revocação), há uma faixa em que não existem ou existem muito poucos verdadeiros positivos, resultando em uma queda acentuada na precisão, seguida por um crescimento causado pela identificação correta dessa classe positiva. Essa ocorrência torna-se mais provável de acontecer devido à grande disparidade entre as classes inerente ao problema.

5.2 Predição de propensão a compra

Os erros de reconstrução do autoencoder nesta etapa foram maiores que os encontrados anteriormente, registrando um MSE médio de 0,17 por amostra, como já era esperado pela Figura 4.4. Isso sugere que as variáveis adicionadas possuem uma maior

dispersão entre seus valores em comparação com as da etapa anterior, o que é compreensível, uma vez que as novas variáveis acrescentadas são contínuas, enquanto as anteriores eram inteiras. Apesar desse aumento nos erros, é importante notar que os valores permanecem relativamente baixos, com 98,32% das amostras apresentando MSE abaixo de 1, mostrando que o autoencoder ainda consegue realizar a reconstrução dos dados de maneira satisfatória.

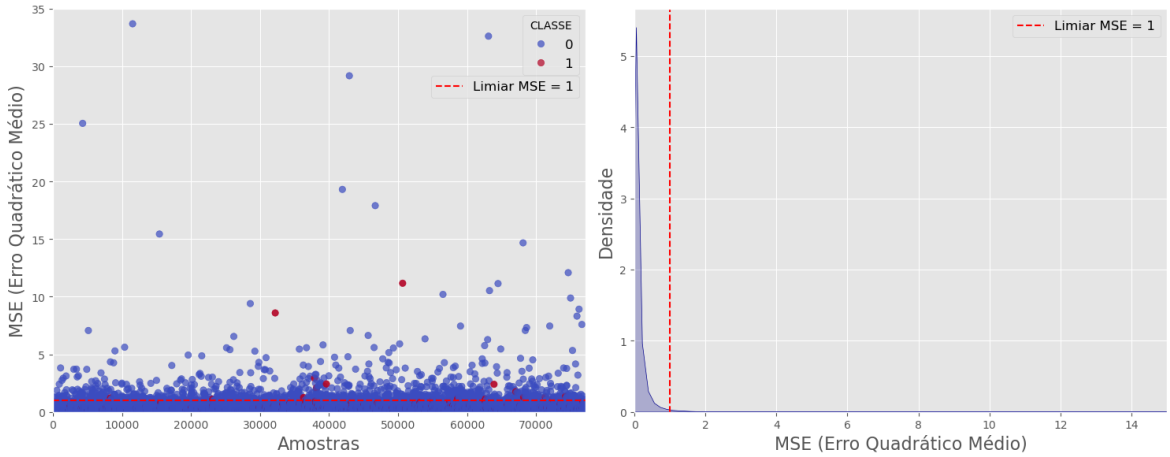
Figura 5.5: Erro quadrático médio (MSE) de reconstrução do autoencoder para base de teste da etapa de predição, por amostra (esquerda) e sua distribuição (direita)



Fonte: Do Autor

Novamente avaliando a reconstrução na base de validação, são obtidos os erros de reconstrução expressados na Figura 5.6. O comportamento obtido foi ainda melhor, com a porcentagem acima da marca de 1 MSE sendo de apenas 1,39%, e o MSE médio por amostra foi de 0,15, corroborando mais uma vez os resultados.

Figura 5.6: Erro quadrático médio (MSE) de reconstrução do autoencoder para base de validação da etapa de predição, por amostra (esquerda) e sua distribuição (direita)

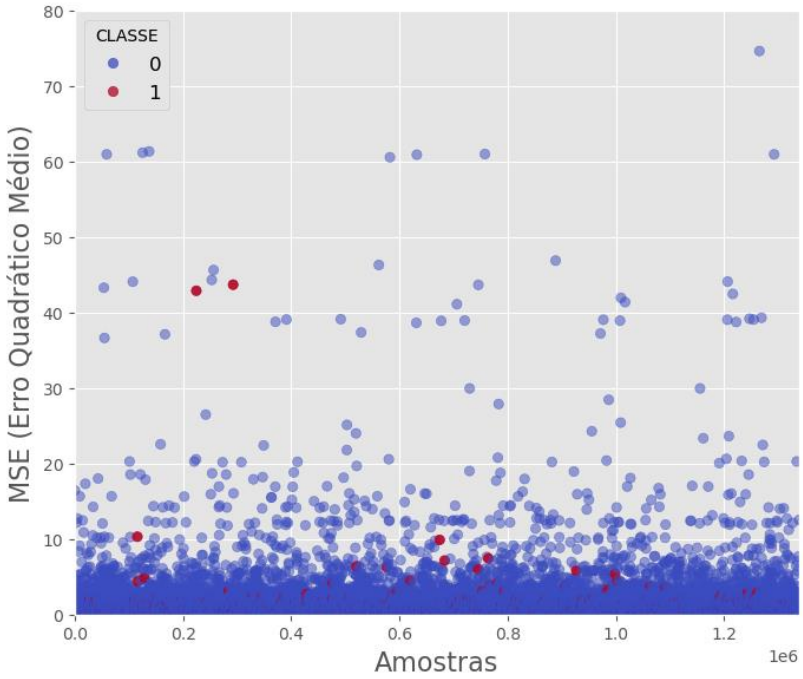


Fonte: Do Autor

No que diz respeito ao autoencoder treinado exclusivamente com instâncias negativas para gerar o erro de reconstrução, os resultados dos erros de reconstrução na base de teste são ilustrados na Figura 5.7. Apesar de amostras positivas terem um MSE médio mais de

duas vezes superior (0,39 contra 0,17), o que indica a capacidade do autoencoder distinguir o comportamento das duas classes, é evidente que várias amostras da classe negativa também exibem um erro significativo, o que pode atrapalhar as predições do modelo.

Figura 5.7: Erro quadrático médio (MSE) de reconstrução por amostra do autoencoder treinado apenas com amostras negativas para base de teste

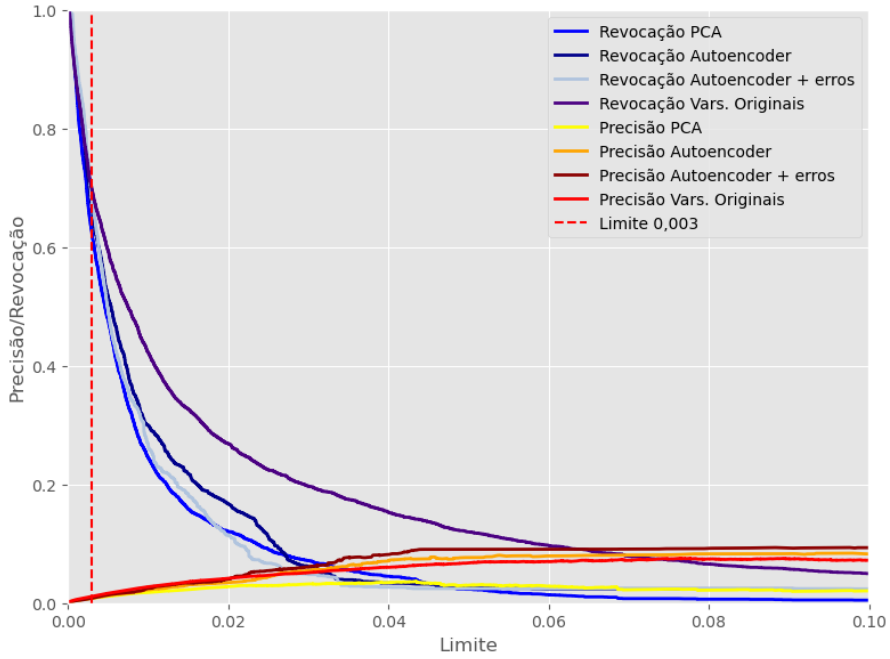


Fonte: Do Autor

A seguir serão avaliadas as métricas e resultados das predições dos modelos. Devido ao acentuado desbalanceamento de classes, métricas de limite são altamente influenciadas pelo limite escolhido, conforme discutido na seção 2.3.1. Para visualizar esse efeito, na Figura 5.8 estão calculadas a precisão e revocação para diferentes valores de limite. Com o limite igual a zero, a revocação é 1 e a precisão é aproximadamente 0, pois há muito mais falsos do que verdadeiros positivos. Conforme o limite aumenta, a precisão tende a aumentar e a revocação a diminuir, pois mais casos que eram considerados positivos se tornam negativos. Para quase todos os limites, a precisão e revocação do autoencoder são maiores ou iguais que as do PCA de mesma dimensionalidade, porém menores que as obtidas usando as 21 variáveis originais.

A definição de um limite “ideal” para ser adotado depende do objetivo do modelo. No cenário abordado, o custo associado aos erros de falsos positivos é baixo (apenas uma ligação que não resulta em compra), mas o valor de verdadeiros positivos é alto (receita imediata). Dessa forma, é preferível otimizar o limite para a revocação, estabelecendo-o em um valor baixo para assegurar a abrangência de todos os potenciais compradores, mesmo que isso resulte em valores menores de precisão.

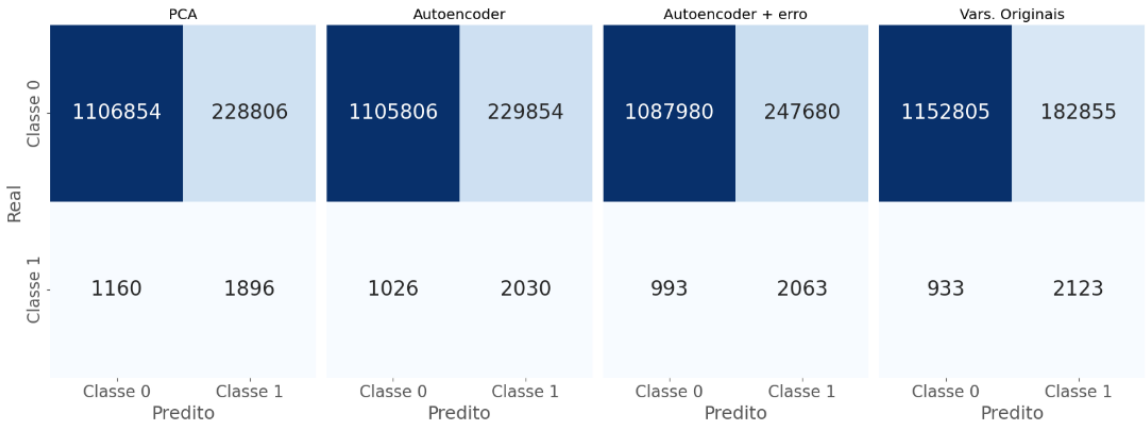
Figura 5.8: Valores de precisão e revocação para diferentes limites



Fonte: Do Autor

Assim, o limite de 0,003 foi escolhido para comparação das matrizes de confusão e valores de precisão e revocação entre os diferentes métodos, encontrados na Figura 5.9 e Tabela 1, respectivamente.

Figura 5.9: Matrizes de confusão das abordagens de PCA, autoencoder, autoencoder + erro e variáveis originais com limite em 0,003



Fonte: Do Autor

Em todas as representações, é evidente a evolução na revocação ao comparar PCA, autoencoder, autoencoder com erros e todas as variáveis originais. Também é possível notar o acúmulo de predições de verdadeiros negativos, o que já era esperado devido à quantidade superior de dados com classe negativa. Entretanto, chama a atenção o número relativamente baixo de verdadeiros positivos, com todos os modelos classificando pelo menos 900 casos positivos como negativos, além dos valores não tão altos de precisão e revocação. No entanto, isso não implica que a separação e assertividade do modelo não é boa. Pode indicar, simplesmente, que estas métricas não são as mais apropriadas para

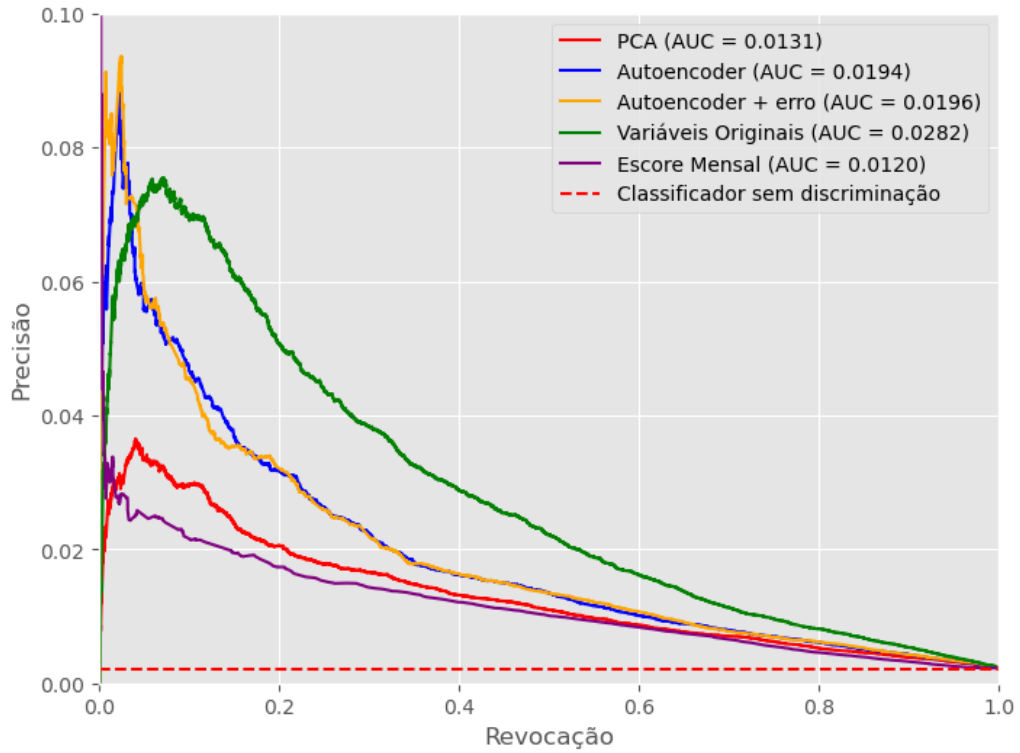
avaliação devido à natureza específica do problema, independente do limite escolhido, ainda mais considerando que o objetivo desta predição é gerar uma lista de priorização onde o importante é a separação das classes, e não a sua predição em formato binário.

Tabela 1: Comparativo de valores de precisão e revocação com limite em 0,003

	Precisão	**Revocação**
PCA	0,008218	0,6204
Autoencoder	0,008754	0,6643
Autoencoder + erro	0,008260	0,6751
Variáveis originais	0,01148	0,6947

Para melhor avaliar a robustez dos modelos no discernimento entre as classes, na Figura 5.10 constam as curvas de precisão-revocação. A diferença entre os resultados do autoencoder e PCA é ainda maior que na etapa anterior, o que sugere menor perda de informação pelo autoencoder conforme a dimensionalidade do espaço latente diminui. Entretanto, essa perda existe, pois o desempenho do modelo treinado com as 21 variáveis originais foi superior.

Figura 5.10: Curva de precisão-revocação para etapa de predição



Fonte: Do Autor

Além disso, o modelo que incorpora o erro de reconstrução demonstrou um desempenho muito próximo do modelo que não o inclui, exibindo valores ligeiramente superiores que possivelmente não justificam sua aplicação neste problema específico, em

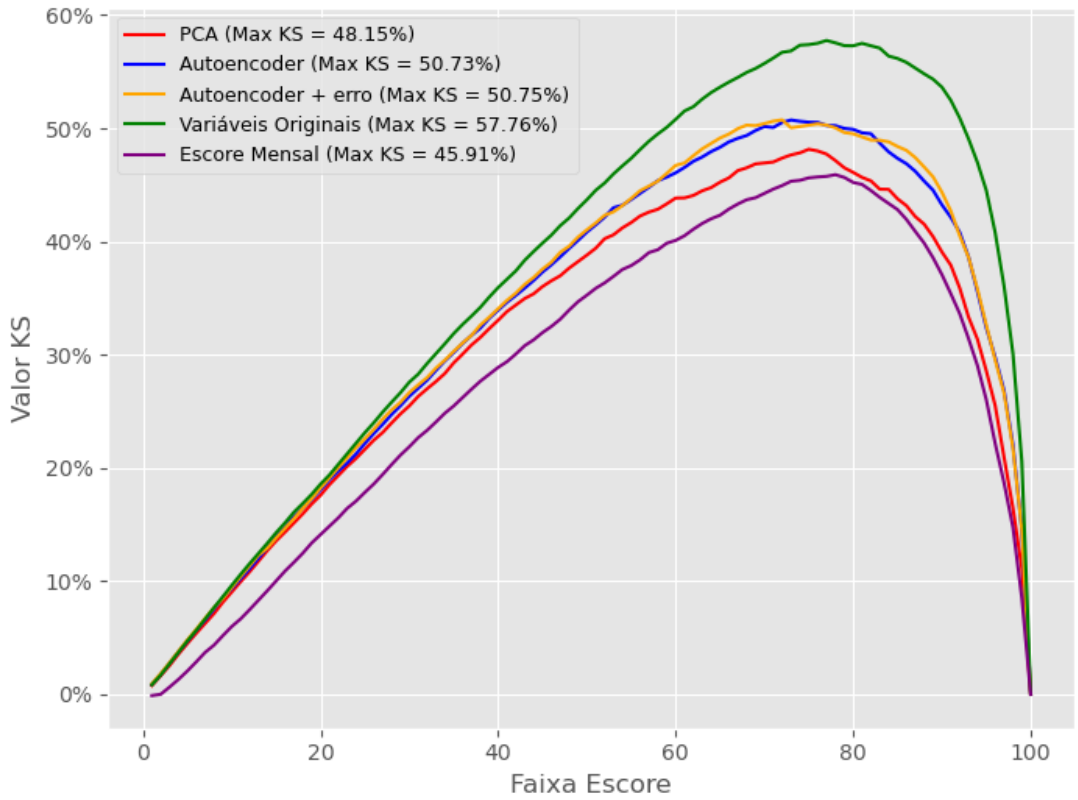
parte devido ao custo computacional elevado associado ao cálculo do MSE. Uma das razões para essa pequena diferença pode ser atribuída à presença de instâncias negativas com alto erro de reconstrução, conforme evidenciado na Figura 5.7, o que pode fazer com que a adição desta informação não seja correspondida com aumento na performance dada a natureza do modelo de árvores LGBM.

O KS, detalhado na Tabela 2, revelou resultados consistentes com a tendência observada. Fica evidente a melhoria no desempenho entre as diferentes abordagens, demonstrado na Figura 5.11 e que se repete em todas as métricas anteriores. Adicionalmente, é possível identificar claramente os picos e as áreas de KS de cada uma. O fato de as curvas estarem levemente deslocadas para a direita sugere que a frequência relativa acumulada da classe positiva está crescendo lentamente, ou seja, os modelos estão fazendo um bom trabalho na identificação de amostras da classe positiva minoritária. Isso corrobora com os resultados apresentados nas curvas de PR AUC, onde os modelos demonstraram um melhor desempenho em situações mais evidentemente positivas, identificadas por escores mais elevados.

Tabela 2: Comparativo de valores de KS para etapa de predição

	PCA	**Autoencoder**	**Autoencoder + erro**	**Variáveis originais**
KS máximo	48,15%	50,73%	50,75%	57,76%
KS AUC	0,3084	0,3281	0,3303	0,3673

Figura 5.11: Curvas de KS para etapa de predição



Fonte: Do Autor

6 Conclusões e Trabalhos Futuros

Neste trabalho, o autoencoder obteve resultados satisfatórios e se mostrou uma excelente técnica para redução de dimensionalidade, produzindo uma representação de dimensão inferior que se mostrou relevante para a predição, podendo então ser aplicada em todas as etapas do processo. Ademais, a estratégia de modelo diário se revelou mais eficaz do que a estratégia mensal atual em todas as abordagens realizadas, alcançando o resultado desejado.

Um dos principais desafios durante o desenvolvimento do trabalho foi a avaliação dos resultados obtidos pelas predições, tanto devido ao desbalanceamento de classes, que requer métricas de avaliação diferentes das usuais, quanto pelo contexto em que essas informações serão utilizadas. Métricas derivadas da matriz de confusão se mostraram menos eficazes em comparação às métricas de ranqueamento para avaliar o desempenho dos modelos. Isso ocorre porque as métricas de limite exigem a definição de um limite, enquanto métricas como PR AUC e KS não requerem tal definição e estão mais alinhadas com o objetivo final, que é criar uma lista de priorização de ligações baseada na propensão do cliente. Como essas métricas consideram a ordem relativa das predições, permitem uma avaliação mais adequada e são mais apropriadas, oferecendo uma representação mais realista sem a necessidade de estabelecer limites arbitrários.

Na etapa de redução de dimensionalidade, o autoencoder obteve baixíssimos erros de reconstrução e atingiu um desempenho melhor ao PCA em todas as métricas avaliadas. Já na etapa de predição, o melhor desempenho foi realizado pela base de dados completa, mostrando que ocorreu perda de informação na redução de 21 para 5 componentes, o que era de certa forma esperado. O uso do erro de reconstrução como variável preditiva não se mostrou muito efetivo para a metodologia utilizada, devido à pequena melhora na performance e da maior capacidade computacional necessária. Porém, se tratando de resultados preditivos, foi na segunda etapa onde o autoencoder demostrou maior superioridade frente ao PCA, revelando-se altamente eficaz em processos que envolvem dados de alta dimensionalidade.

Contudo, sua utilização demanda algumas considerações, como a maior capacidade computacional necessária para o treinamento, bem como a necessidade de definir parâmetros e métodos que não são exigidos no PCA, por exemplo. Em ambientes com baixa dimensionalidade, o uso do autoencoder pode não ser imprescindível. Já em ambientes de alta dimensionalidade, o autoencoder se posiciona como uma das melhores alternativas atualmente. No entanto, a determinação do que constitui baixa ou alta dimensionalidade depende do contexto e da natureza do problema em questão.

Algumas melhorias poderiam ser implementadas em trabalhos futuros, como o desenvolvimento de um método para definição do quanto cada variável em seu formato original contribuiu para a representação obtida no espaço latente, através dos pesos atribuídos por cada neurônio. Isso permitiria maior interpretabilidade do processo realizado pelo autoencoder e poderia ser de ajuda em tomadas de decisão de negócio. Também poderia ser possível a exploração de novas definições dos grupos relacionados à classificação das ligações. Neste trabalho, a redução de dimensionalidade foi feita apenas em variáveis agrupadas em relação ao tempo (dias), com as quatro categorias de ligações já pré-estabelecidas seguindo a definição de alô e alô efetivo. Porém, a base original possui diversas subcategorias relativas ao resultado da ligação que podem ser exploradas.

7 Referências

CNC. **Pesquisa de Endividamento e Inadimplência do Consumidor (PEIC)**. 2023.

INSTITUTO PROPAGUE. **Mercado de crédito em dados**. Instituto Propague. 2021.

ARDELEAN, Eugen-Richard *et al.* **A study of autoencoders as a feature extraction technique for spike sorting**. PLoS ONE 18, 2023.

JIA, Weikuan *et al.* **Feature dimensionality reduction: a review**. Complex & Intelligent Systems (2022), 2022.

PEARSON, Karl. LIII. **On lines and planes of closest fit to systems**. The London, Edinburgh, and Dublin Philosophical, 1901.

BJÖRKLUND, Mats. **Be careful with your principal components**. Evolution, 2019.

TELLAECHE IGLESIAS, Alberto *et al.* **On Combining Convolutional Autoencoders and Support Vector Machines for Fault Detection in Industrial Textures**. Sensors, 2021.

KE, Guolin *et al.* **LightGBM: A Highly Efficient Gradient Boosting**. 31st Conference on Neural Information Processing Systems, 2017.

HE, Haibo; MA, Yunqian. **Imbalanced Learning: Foundations, Algorithms, and Applications**. 1ª. ed. Wiley-IEEE Press, 2013. p. 187.

BRANCO, Paula; TORGO, Luis; RIBEIRO, Rita. **A Survey of Predictive Modelling under Imbalanced Distributions**, 2015.

COOK, Jonathan ; RAMADAS, Vikram. **When to consult precision-recall curves**. The Stata Journal, 2020. p. 131–148.

CONOVER, W. J. **Practical Nonparametric Statistics**. 3. ed. Nova Iorque: John Wiley and Sons, 1999.

PEREIRA, Gustavo H. D. A. **Modelos de Risco de Crédito de Clientes: Uma Aplicação a Dados Reais**. Universidade de São Paulo. São Paulo. 2004.

TIBSHIRANI, Robert. **Regression Shrinkage and Selection via the Lasso**. Journal of the Royal Statistical Society: Series B (Methodological), 1996. p. 267-288.

PLAUT, Elad. **From Principal Subspaces to Principal Components with Linear Autoencoders**. arXiv, 2018.

WONG, Timothy; LUO, Zhiyuan. **Recurrent Auto-Encoder Model for Large-Scale Industrial Sensor Signal Analysis**. 19th International Conference on Engineering Applications of Neural Networks, 2018.

GLOROT, Xavier; BENGIO, Yoshua. **Understanding the difficulty of training deep feedforward neural networks**. International Conference on Artificial Intelligence and Statistics (AISTATS), 2010.