UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Uma análise da decisão de consumo sobre cartão de crédito consignado dos servidores federais brasileiros

Thiago Cristian de Souza

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Uma análise da decisão de consumo sobre cartão de crédito consignado dos servidores federais brasileiros

Thiago Cristian de Souza

USP - São Carlos

2023

Thiago Cristian de Souza

Uma análise da decisão de consumo sobre cartão de crédito consignado dos servidores federais brasileiros

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientadora: Profa. Marislei Nishijima

USP - São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

S729a Souza, Thiago Cristian
Uma análise da decisão de consumo sobre cartão de crédito consignado dos servidores federais brasileiros / Thiago Cristian Souza; orientadora Marislei Nishijima. -- São Carlos, 2023.
49 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. . I. Nishijima, Marislei, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

RESUMO

Souza, Thiago Cristian. Uma análise da decisão de consumo sobre cartão de crédito consignado dos servidores federais brasileiros. 2023. 47 p. Trabalho de conclusão de curso (MBA em *Inteligência* Artificial e *BigData* – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Com a dificuldade de alguns servidores públicos em adquirir créditos no mercado, ou com as altas taxas administradas pelos bancos, o crédito consignado tornou-se a solução mais viável, por ser uma linha de crédito com juros baixos, a partir do momento em que o banco tem garantias de recebimento, já que pode descontar o valor mensal diretamente da folha de pagamento do cliente. Após a sua criação do mercado em 2003, o mercado se agitou e tentou se organizar para ofertas estes tipos de produto, criando um enorme ecossistema em volta da oferta deste produto. Entretanto, com o passar dos anos e o avanço da tecnologia, as formas de abordagem ao servidor foi melhorando e tornando-se cada vez mais invasivas, seja pela utilização de discadoras automatizadas, disparos automáticos de *SMS*, whatsapp, e-mails. O contato direto com o servidor acabou ficando muito intenso, fazendo-se necessário a intervenção de órgãos como a FEBRABAN para tentar diminuir o assédio, devido à alta taxa de reclamação pela forma como os servidores são abordados para serem oferecidos estes produtos. Com o intuito de melhorar esta abordagem, o objetivo deste estudo é criar um algoritmo capaz de detectar comunidades de servidores federais a fim de reconhecer e recomendar os melhores clientes a serem contatados para adquirir o cartão consignado com vistas a diminuir tanto a quantidade de abordagens ao cliente, quanto as abordagens improdutivas. O algoritmo proposto foi realizado no modelo do *XGBoost*. Foi realizado um tratamento dos dados, estudo dos melhores parâmetros de entrada até chegar no melhor resultado proposto pelo projeto.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CPF	–	Cadastro Nacional de Pessoa Física
DL	–	Deep *Learning*
FEBRABAN	–	Federação Brasileira de Bancos
IA	–	Inteligência Artificial
ID	–	Index
INSS	–	Instituto Nacional de Seguridade Social
ML	–	*Machine Learning*
MLP	–	Perceptron Multi-Camadas
SMS	–	*Systems Management Server*
SIAPE	–	Sistema Integrado de Administração de Pessoal
UF	–	Unidade federativa
URL	–	*Uniform Resource Locator*
XGBoost	–	Biblioteca de software de código aberto que fornece um framework de *"gradient boosting"*
WL	–	W*eak Learners*

SUMÁRIO

15

1 INTRODUÇÃO

O crédito consignado se tornou a solução de funcionários públicos que não conseguiam obter crédito no mercado ou só conseguiam com a taxa de juros alta. Esta linha de crédito permite que estes servidores possam ter acesso a crédito com juros baixos, isso ocorre porque o banco emprestador tem garantias por meio de descontos mensais das parcelas direto na folha de pagamento do cliente.

Entretanto, esta linha de crédito criou um ecossistema enorme de sistemas de informação em torno deste produto com a finalidade de ajudar bancos a oferecerem tal linha de crédito. Com o avanço da tecnologia e a melhoria das formas de abordar o cliente, como por exemplo, a utilização de discadoras automatizadas para potenciais clientes, gerou-se um intenso acesso direto aos clientes, fazendo-se necessário a intervenção de órgãos como a FEBRABAN para tentar diminuir o grau de assédio aos funcionários públicos, aposentados e pensionistas aptos a adquirir este tipo de produto.

Houve um grande avanço dos estudos nas áreas de IA, *machine learning* e *Big Data.* Isto trouxe investimentos em aprimoramento e disseminação dos mecanismos de armazenamento, análise e utilização de dados por parte de empresas públicas e privadas no mundo todo. Este avanço pode aumentar e melhorar os resultados de muitos negócios por meio do conhecimento dos dados.

1.1 Motivação

Os produtos bancários têm uma alta taxa de reclamação em fóruns apropriados no que se refere à forma como os servidores públicos são abordados para oferecimento de produtos relacionados com crédito consignado. Muitas vezes, o cliente nem pode mais adquirir o cartão consignado, seja por já ter saído do emprego, ou por não ter margem para contratar, já ter adquirido anteriormente o produto, ou outros problemas, mas mesmo assim, estes clientes continuam sendo contactados inúmeras vezes durante o dia. Ligações telefônicas aleatórias que incomodam muito os clientes, assim como atrapalha a vida dos correspondentes bancários, que perdem a maior parte do tempo filtrando contatos improdutivos.

1.2 Objetivo

O objetivo deste trabalho foi criar um algoritmo capaz de detectar comunidades de servidores federais a fim de reconhecer e recomendar os melhores clientes a serem contatados para adquirir o cartão consignado. Dessa maneira, reduzir a quantidade de abordagens a clientes com baixa probabilidade de consumir este produto e, portanto, reduzir abordagens improdutivas e que possam incomodar clientes.

1.3 Justificativa

Conhecer e entender a base de dados e compreender o comportamento dos servidores na contratação deste serviço bancário, é essencial para evitar abordagens excessivas aos potenciais emprestadores. Agrupar tais tomadores de empréstimos de forma a poder predizer quais os melhores servidores poderiam ser contatados para ter uma melhor aceitação de contratação do produto.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA

2.1 Considerações iniciais

Neste capítulo são apresentadas a fundamentação teórica com uma explicação dos agentes e formatos que fazem parte do processo de crédito consignado e alguns trabalhos relacionados a credito consignado disponíveis na literatura.

2.2 Fundamentação teórica

2.2.1 Empréstimo consignado

Esta modalidade de empréstimo foi regulamentada por meio da Lei 10.820 de 17/12/2003 e se refere à uma modalidade de empréstimo em que o valor das parcelas para o pagamento do empréstimo é descontado diretamente na folha de pagamento do funcionário público (Coelho et al. 2015). Desta forma, o banco tem uma garantia muito efetiva e direta de que o valor emprestado ser á pago, oferecendo supostamente crédito a taxas de juros menores do que as usuais de mercado. Assim, funcionários públicos que não tinham acesso ao crédito, por inúmeros motivos, como por exemplo, estar com o nome negativado ou ter uma renda muito baixa, passaram a ter acesso a esta linha de crédito com juros mais baixos. Outra vantagem do crédito consignado, é que o emprestador não precisa ser correntista do banco para a realização do empréstimo.

Nesta modalidade de empréstimo consignado, que foi a primeira a ser criada, existe um valor máximo a ser descontado na folha de pagamento do servidor em percentual do salário mensal, que corresponde ao valor da soma de todas as parcelas de todos os empréstimos consignado ativos. Como o valor das parcelas é descontado diretamente na folha de pagamento, existe um controle rígido do máximo percentual permitido para a realização deste empréstimo consignado.

2.2.2 Cartão consignado

O cartão consignado corresponde a uma modalidade de empréstimo consignado adicional aos servidores públicos via um cartão ao qual o pagamento mínimo da fatura é descontado de forma automática direto na folha do cliente. Por este motivo, a taxa de juros aplicada nestes cartões pode ser também bastante reduzida em relação aos créditos

disponíveis no mercado. Existe também a possibilidade adicional de sacar uma porcentagem do valor do limite do cartão, a juros relativamente baixos, já que um valor mínimo é sempre descontado na folha de pagamento. Relativamente ao crédito consignado, o cartão permite um aumento do limite do percentual do salário que os servidores podem empenhar em empréstimos.

Em nível de comparação, pela garantia de pagamento através do desconto automático direto na folha de pagamento, o valor máximo dos juros que pode ser cobrado no cartão de crédito consignado para servidores federais é de 2,73% ao mês, já no cartão de crédito normal, este valor em geral ultrapassa 12% ao mês.

2.2.3 Empresas de consignado

O ecossistema criado em torno do crédito consignado consiste de vários agentes, tais como os bancos comerciais e as empresas que oferecem empréstimo consignado e que são os correspondentes bancários de um ou mais bancos. Estes últimos agentes têm autorização dos bancos para oferecer os seus produtos bancários, fazendo a intermediação entre o cliente e o banco. Além disso, geralmente trabalham no modelo *callcenter*, que através de listas de clientes, disparam inúmeras ligações, torpedos de voz, SMS e mensagens no *whatsapp* a fim de conseguir contactar o cliente e oferecer o empréstimo ou o cartão consignado a ele.

2.3 Estudos Relacionados

Como mostrado por Pontes e Lopes (2017), o crédito é visto como um instrumento financeiro que estimula a economia e gera empregos e rendas, afetando diretamente a vida das pessoas de todas as idades, sendo o meio mais rápido e fácil de obtenção de bens. Os autores apontam que em 2003 foi criada a modalidade de crédito consignado, e que esta concepção foi focada principalmente visando auxiliar os idosos aposentados e pensionistas do INSS. Focado também nos servidores públicos e federais, foi criado durante o primeiro mandato do governo do presidente Lula, para fazer a oferta de crédito disponibilizada pelos bancos ser mais acessível e ter menores juros. Primeiramente começou com o empréstimo e ao longo dos anos, foram criando outros produtos, como cartão de crédito, seguro com desconto direto na fatura do cartão, entre outros produtos.

Coelho et al. (2015) estudam o efeito da entrada do crédito consignado no Brasil referente ao período 2003, criação do crédito, até 2008 e concluem que esta forma de crédito reduziu a taxa de juros e aumentou o volume geral de créditos concedidos no país.

Schuh et al. (2019) usando dados de 2004 a 2016 concluem que no longo prazo o empréstimo consignado diminui a taxa de inadimplência de no Brasil.

Particularmente, não foi encontrado na literatura trabalho correlato com o mesmo objetivo deste, apenas trabalhos usando classificação com linguagem de máquina para outros objetivos. Paula et al. (2019) usam modelos logísticos para classificar o risco de *default* em empréstimos no Brasil, incluindo empréstimos consignados. Arram et al. (2023) usam vários algoritmos de aprendizagem de máquina para classificar risco de *default* em empréstimos de cartão de crédito e concluem que a rede neural *perceptron* multicamada (MLP) performa melhor do que incluindo regressão logística, árvores de decisão, e florestas aleatórias.

2.4 Modelos e técnicas de machine learning

Para o problema proposto, utilizamos o *XGBoost* como modelo para encontrar o resultado que encontramos neste projeto, além de utilizar a regressão linear múltipla para ajudar a responder qual o sentido de influência de algumas variáveis utilizadas para realizar o treinamento e classificação do modelo.

2.4.1 XGBoost

Segundo (CHEN; GUESTRIN, 2016), o *XGBoost* é um *boosting* baseado no gradiente e é considerado o estado da arte dos algoritmos de ML que não utilizam as técnicas de DL. Ele realiza utilização de técnicas de *Boosting* , o qual adiciona sucessivamente modelos WL, a fim de corrigir o conjunto de entradas do passo anterior, otimizando e acertando a entrada para o próximo passo, otimizando a função de perda com base na técnica do gradiente e minimizando o risco estrutural (Reis, L. G. Moura, 2022).

2.4.2 Regressão linear múltipla

No estudo proposto, a principal técnica utilizada para chegar no modelo proposto foi o *XGBoost*. Mas o projeto precisava de algumas respostas, como a direção de influência de alguns campos sobre o resultado final. A regressão linear conseguiu nos dar esta resposta, embora os modelos utilizem métodos diferentes.

3 MODELO PARA CLASSIFICAR OS SERVIDORES

3.1 Considerações iniciais

Este capítulo pretende explicar em detalhes como o problema proposto será solucionado. Na sequência das suas subseções, descreve os passos utilizados para a coleta de dados, informando suas fontes de origem, como foi a sua coleta, como foram tratados. Para posteriormente, serem separados em conjuntos de dados utilizados no treinamento e teste em avaliação do modelo que será proposto no trabalho a ser apresentado no projeto.

3.2 Problemática e solução

Através da utilização de bases internas próprias, as empresas de consignado realizam ligações aleatórias para servidores públicos federais, seja de forma manual ou utilizando discadoras automatizadas, o que intensifica o contato telefônico com estes servidores, fazendo com que recebam, às vezes, até mais de 10 ligações por dia[1], de várias empresas diferente. Além de perturbar o servidor, que em geral não tem interesse em adquirir o cartão de crédito consignado, pode causar uma imagem ruim para a empresa de crédito.

A fim de melhorar o contato com os possíveis clientes e a eficiência dos resultados obtidos, este trabalho utilizou o algoritmo XGBoost para realizar uma triagem dos dados dos funcionários públicos antes de realizar uma tentativa de contato. Desse modo, o modelo busca direcionar o contato apenas para servidores com maior probabilidade e interesse em contratar o produto, objeto deste estudo, o cartão consignado. Aumentando, assim, a assertividade dos contatos, diminuindo a quantidade de contatos improdutivos e aumentando as vendas destas empresas.

Sendo assim, iniciamos com a descrição do processo de coleta e tratamento dos dados, seguido da descrição dos problemas enfrentados e como eles foram resolvidos.

3.3 Coleta de dados

O processo de coleta de dados foi baseado em 2 fontes de dados secundários. Os dados vieram de forma estruturada, mas foram necessários vários passos para conseguir realizar a junção destes dados, além das análises dos dados faltantes, *outliers*, duplicados,

1 Conforme informação da empresa privada que forneceu os dados reportados neste estudo.

com pouca relevância para o estudo, como servidores que não podem adquirir produtos consignados, entre outros tratamentos para melhorar a qualidade do modelo.

3.3.1 Construção da base de dados

A primeira fonte, foi uma base comercial particular que foi disponibilizada por uma empresa privada para a qual estou realizando este trabalho. Nesta base, foi possível extrair informações sobre os servidores públicos federais como sexo, idade, estado e se já adquiriu ou não o cartão de crédito consignado.

A segunda fonte, foram 2 bases coletadas no site do portal da transparência do governo[2], onde estão disponibilizados varias bases de dados abertos dos servidores federais. Para este projeto, foram baixadas as bases seguindo o seguinte caminho: Foi acessado a URL do portão citado acima, em seguida clique na aba “Servidores” e no link “Servidores (Ativo e Inativos) e Pensionistas”. Este trabalho, entretanto, explora apenas o banco de dados dos servidores ativos, referente ao exercício de junho de 2023. Então os campos ficaram marcados assim: Exercícios Disponíveis = 2023, Meses Disponíveis em 2023 = Junho, Tipos de planilhas disponíveis em = Servidores_SIAPE. Após isto, clicar em “Baixar”. Com isto será baixado um arquivo no formato zip. Com isto, foi necessário descompactar o arquivo em uma pasta, para ter acesso as 2 bases o qual será utilizada no estudo que será a base de cadastro e a base de remuneração dos servidores.

Após a extração e coleta de dados, foi iniciado o processo de junção das bases de dados, vislumbrando extrair o máximo possível de informações disponíveis de cada servidor. Para isto, primeiramente foi realizado um estudo das bases para identificar as colunas com dados iguais em tabelas diferentes.

Após este estudo, foi constatado que a melhor maneira de juntar estes dados, seria primeiro juntar as 2 bases de dados abertos do governo, e em seguida juntar com a base privada.

Para juntar as 2 bases que vieram do portal da transparência do governo, foi identificado que as 2 possuem a coluna “Id_SERVIDOR_PORTAL”, o qual é utilizado para identificar os dados do servidor dentro do portal. Então ao utilizar esta coluna para realizar a junção das tabelas, ficaram mantidos os dados da tabela de cadastro e trouxe apenas a coluna “REMUNERAÇÃO BÁSICA BRUTA (R$)” com o nome de “salario” para a base nova gerada.

2 Disponível em: <https://portaldatransparencia.gov.br/download-de-dados> Acesso em: Acesso em: 10 set. 2023

Após isto, ficou faltando juntar a base privada com as bases do portal da transparência. Para isto, foram utilizados os campos “NOME”, “CPF” e “MATRICULA” das duas bases para realizar a junção. Lembrando que os dados de “CPF” e “MATRICULA” são em parte mascarados, para não infringir a lei de proteção de dados, foi levado em conta esta informação no momento de realizar a junção. Considerando a grande coincidência necessária entre os números de CPF disponíveis e nomes completos dos indivíduos para ocorrerem duplicidades, a concatenação das informações gerou um banco de pessoas praticamente sem duplicidade. Na sequência, as três colunas identificadoras foram eliminadas dos dados finais alvos deste estudo, pois além de serem dados sigilosos são também dados muito específicos e diferentes para cada servidor, não fazendo sentido utilizar na realização deste estudo. Nesta junção, foram mantidos os dados do portal da transparência com o identificador do indivíduo, com exceção destas três colunas, adicionando as informações de sexo, idade e se tem ou não o cartão consignado.

3.3.2 Tratamento da base de dados

Para realizar o tratamento dos dados, após a junção, foram realizados os seguintes passos:

1. Através dos dados da UF de exercício o qual o servidor está em exercício, foi criado uma coluna para informar a região o qual aquele estado pertence.
2. Foram mantidas na base apenas as colunas que contem dados de sexo, idade, regime jurídico vinculado, qual o tipo de vinculo com o órgão, a jornada de trabalho, a situação do vinculo com o órgão, o órgão superior ao qual está alocado, a região que reside e se contêm ou não o cartão de crédito consignado.
3. Foram retiradas as linhas o qual tinham dados com valores nulos, para facilitar e melhorar o treinamento do modelo.
4. Foram retiradas as linhas com dados de idades inferiores a 18 e maiores que 79, o qual não é possível realizar a contratação do cartão consignado.
5. Foram retirados os dados o qual a situação do servidor no órgão não é consignável.
6. Foram retiradas as linhas o qual o regime jurídico do servidor não é consignável.
7. Foram retirados os *outliers* da tabela com salários, onde foram retiradas as linhas o quais os dados de salário estavam acima de 30 mil, neste caso havia agrupamento de todos os salários maiores de 30 mil impossibilitando a identificação do salário do servidor.

Finalizado estes passos, foi iniciado o processo de estudo e preparação dos dados tratados. Ao todo, o tamanho da base total resultante foi de 99.503 linhas para ser utilizada no treinamento e validação do modelo proposto.

Esta etapa foi realizada para consolidar os dados de todas as colunas e suas características das informações tratadas na etapa anterior. Foram analisados e tratados os dados das colunas para melhor funcionamento do algoritmo.

Primeiramente a Tabela 1 descreve as informações de nome, descrição e tipo de dado atual de todas as colunas que serão trabalhadas (cada coluna representa uma das variáveis usadas). Foi incluído um “ID” nas colunas, para facilitar a identificação delas. Em seguida apresento como cada coluna foi preparada e no final apresento outra tabela, a Tabela 2, com as informações finais da base que foram inseridas como entrada de informação para rodar o modelo proposto.

Sendo assim, segue abaixo a primeira tabela descritiva de dados. A Tabela 1 informa como os dados provindos do site do próprio portal da transparência. No site está disponibilizado um dicionário de dados dos servidores[3] para descrever os campos retornados. Também apresentam os campos de dados providos pela da base do parceiro particular, com suas descrições e o atual formato dos dados.

Tabela 1 – Visão geral das colunas antes da preparação dos dados

ID	Campo	Descrição	Tipo de dado
1	tem_cartao	Informa se o servidor tem ou não o cartão consignado. Esta será a variável alvo.	Binário (0 ou 1)
2	sexo	Sexo do servidor, separado em masculino ou feminino	String
3	idade	Idade do servidor	Inteiro
4	salario	Salário bruto do servidor	Decimal
5	regime_juridico	Regime jurídico o qual o servidor esta vinculado	String
6	jornada_trabalho	Jornada de trabalho semanal do servidor, representado em horas por semana.	String
7	tipo_vinculo	Tipo de vínculo do servidor com o órgão de lotação.	String
8	situacao_vinculo	Situação do vínculo do servidor com o órgão de lotação.	String
9	orgsup_locacao	Órgão superior onde o servidor está lotado.	String
10	uf_exercicio	UF do país onde o servidor está locado.	String

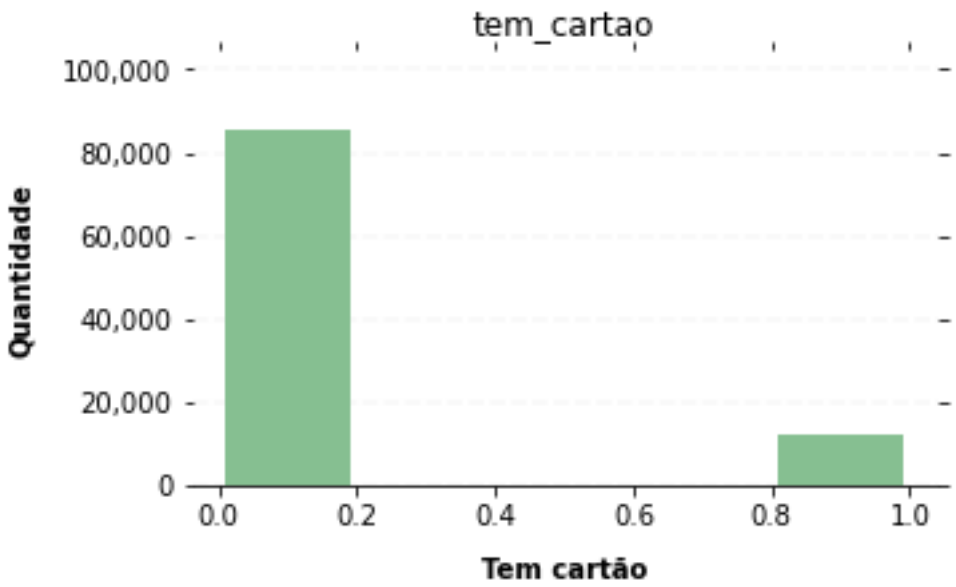
Fonte: Autor

3 Disponível em: https://portaldatransparencia.gov.br/pagina-interna/603422-dicionario-de-dados-servidores-cadastro Acesso em: 10 set. 2023

Na sequência, apresento abaixo descritivamente como cada coluna (variável) foi analisada e preparada para chegar nos dados finais de entrada.

1. tem_cartao - Os dados já estão em formato binário, então não será necessária nenhuma modificação dos dados. Sobre o valor: 1 para quem já adquiriu o cartão e 0 para quem ainda não tem.

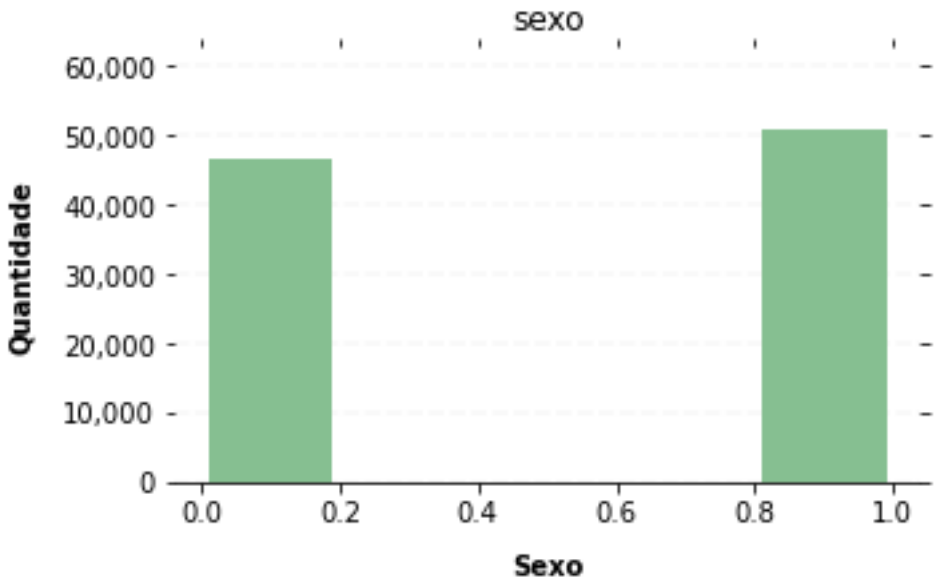
Figura 1 – Quantidade de dados da coluna “sem_cartao”



Fonte: Autor

2. sexo - Os dados estão em formato *string* e foi necessário transformá-los em binário, transformando o valor “masculino” em 1 e feminino em “0”.

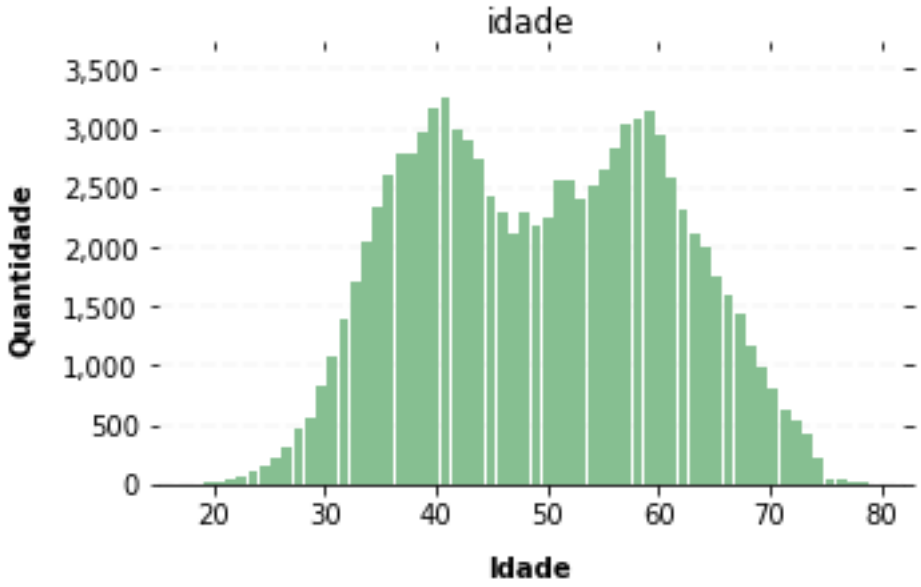
Figura 2 – Quantidade de dados da coluna “sexo”



Fonte: Autor

3. idade - Os dados estão em formato inteiro e por ser uma coluna no qual a ordenação dos dados possui um significado de maior ou menor idade, não se trata de variável categórica, não exigiu transformação da variável.

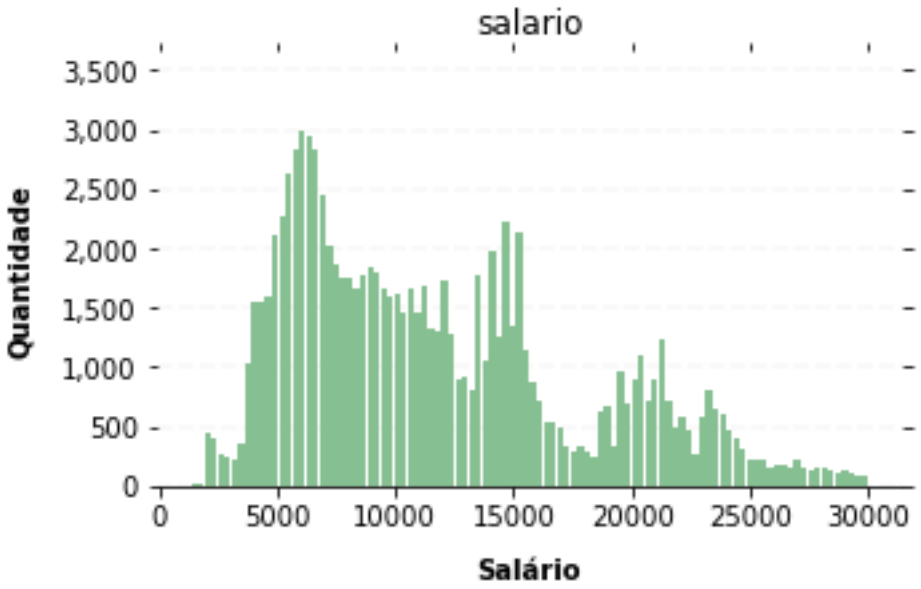
Figura 3 – Quantidade de dados da coluna “idade”



Fonte: Autor

4. salario - Os dados estão em formato decimal e por ser uma coluna o qual os dados representa distância de um para o outro, não foi feita nenhuma alteração.

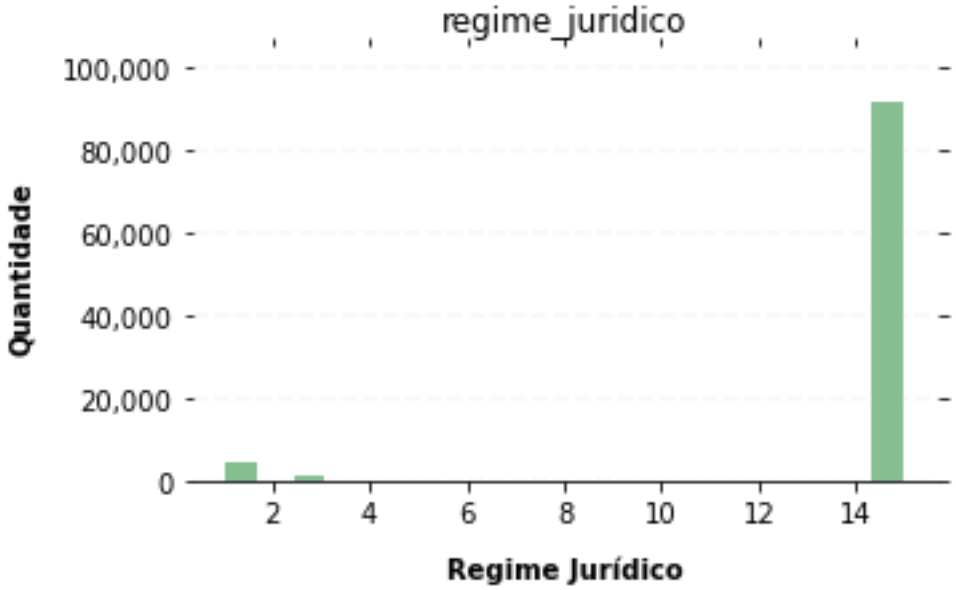
Figura 4 – Quantidade de dados da coluna “salario”



Fonte: Autor

5. regime_juridico - Os dados estão em formato *string* e foi necessário transformá-los em binário. Como 91.657 dos dados era “REGIME JURIDICO UNICO”, a grande maioria dos dados, foi colocado o valor 1 para ele e 0 para ou outros.

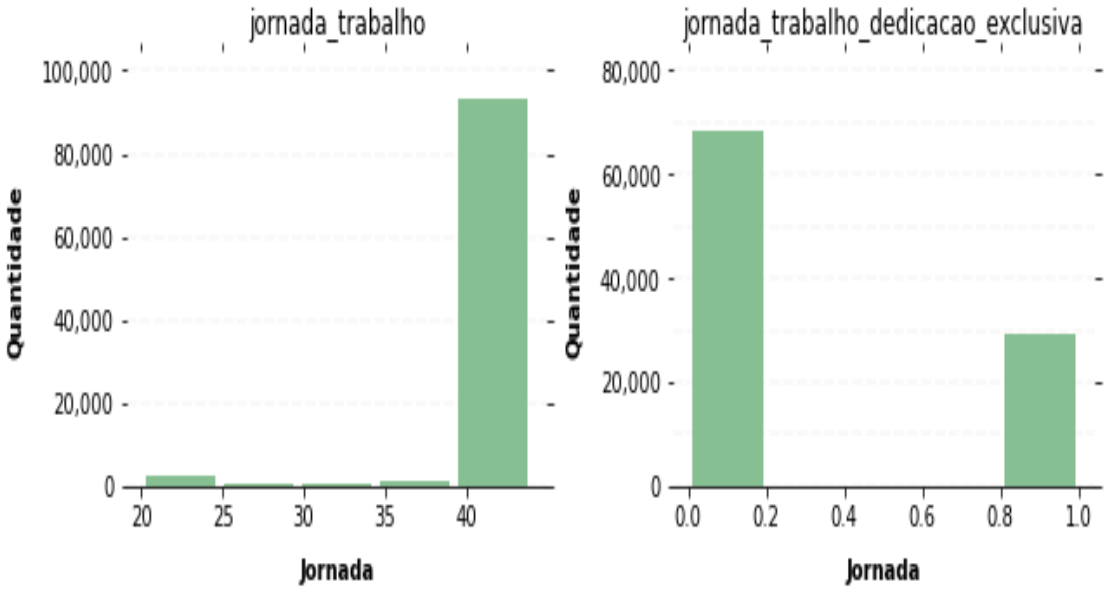
Figura 5 – Quantidade de dados da coluna “regime_juridico”



Fonte: Autor

6. jornada_trabalho - Os dados estão em formato *string* e analisando elas, os 2 primeiros caracteres eram informações da quantidade de horas da jornada de trabalho, menos uma amostra da base de dados que tinha o valor “DEDICACAO EXCLUSIVA”. Após a analise, esta coluna foi transformada em valores inteiros com a informação dos 2 primeiros caracteres, e foi colocado o valor de 40 para o dado “DEDICACAO EXCLUSIVA”, representando as horas trabalhadas por semana de cada servidor. Além disto, já existia uma jornada de trabalho com valor de 40 horas, por isto foi necessário criar a coluna chamada jornada_trabalho_dedicacao_exclusiva em formato binário, adicionando o valor 1 para o dado “DEDICACAO EXCLUSIVA” e 0 para os outros dados, e assim diferenciando os 2 dados que estavam iguais na coluna.

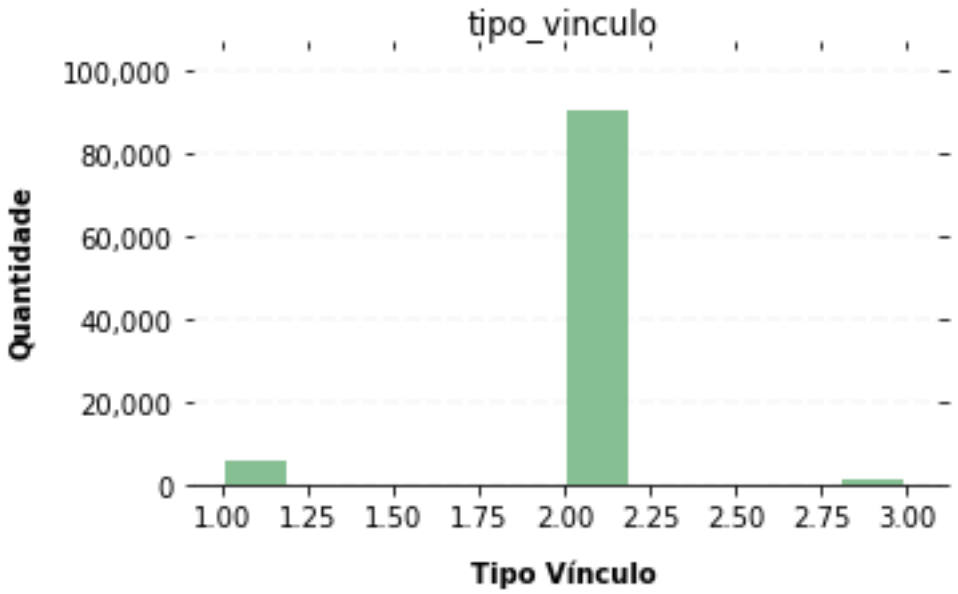
Figura 6 – Quantidade de dados da coluna “jornada_trabalho” e “jornada_trabalho_dedicacao_exclusiva”



Fonte: Autor

7. tipo_vinculo - Os dados estão em formato *string* e foi necessário transformá-los em binário. Como 90.173 dos dados era “Cargo”, a grande maioria dos dados, foi colocado o valor 1 para ele e 0 para ou outros.

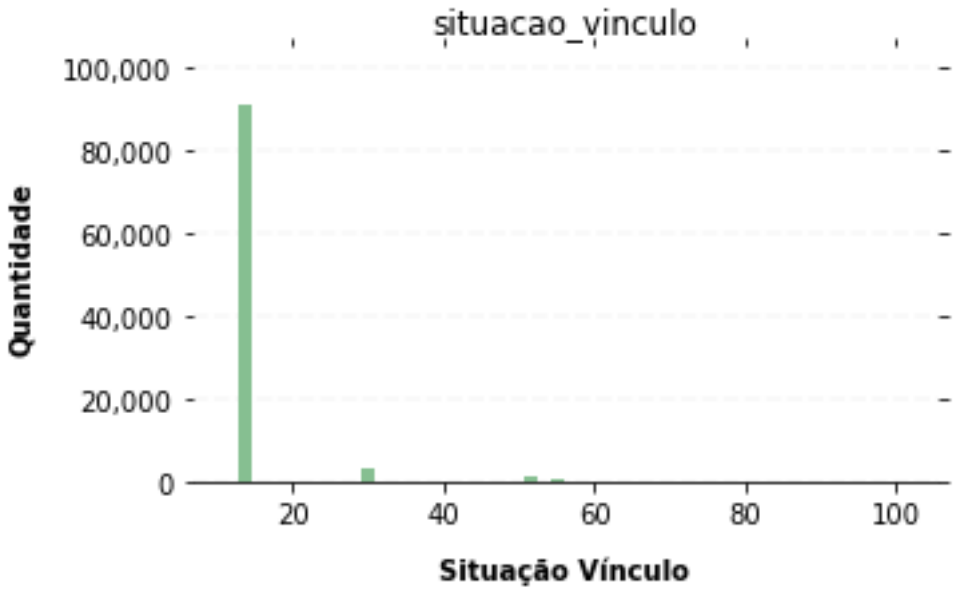
Figura 7 – Quantidade de dados da coluna “tipo_vinculo”



Fonte: Autor

8. situacao_vinculo - Os dados estão em formato *string* e foi necessário transformá-los em binário. Como 90.730 dos dados era “ATIVO PERMANENTE”, a grande maioria dos dados, foi colocado o valor 1 para ele e 0 para ou outros.

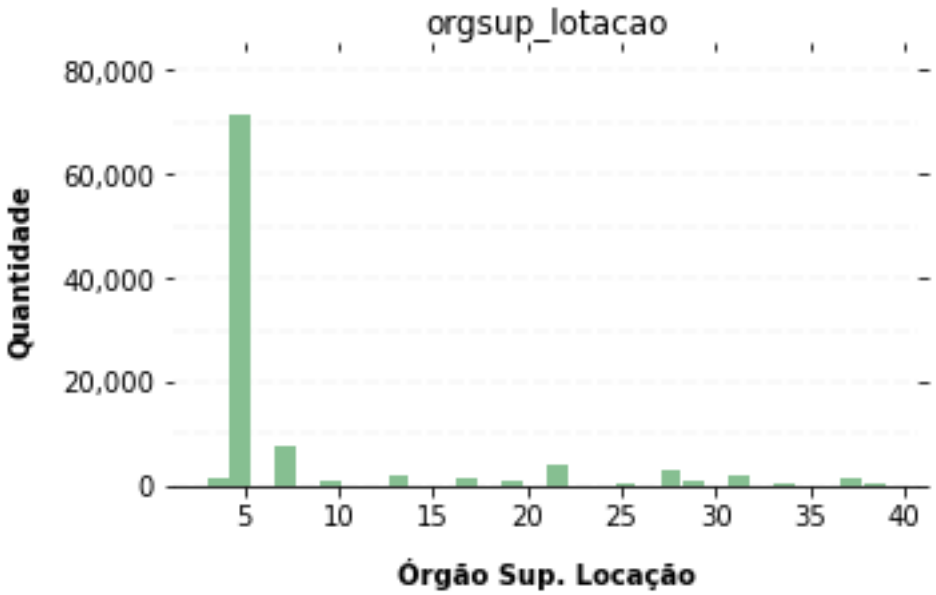
Figura 8 – Quantidade de dados da coluna “situacao_vinculo”



Fonte: Autor

9. orgsup_locacao - Existem 16 dados diferentes nesta coluna e estão em formato *string*. Para tratar, primeiro os dados categóricos foram transformados em números inteiros, e depois foi necessário realizar um *dummy* na coluna, deletando a coluna atual e criando 16 novas colunas com o nome de prefixo “orgsup_lotacao_” + valor.

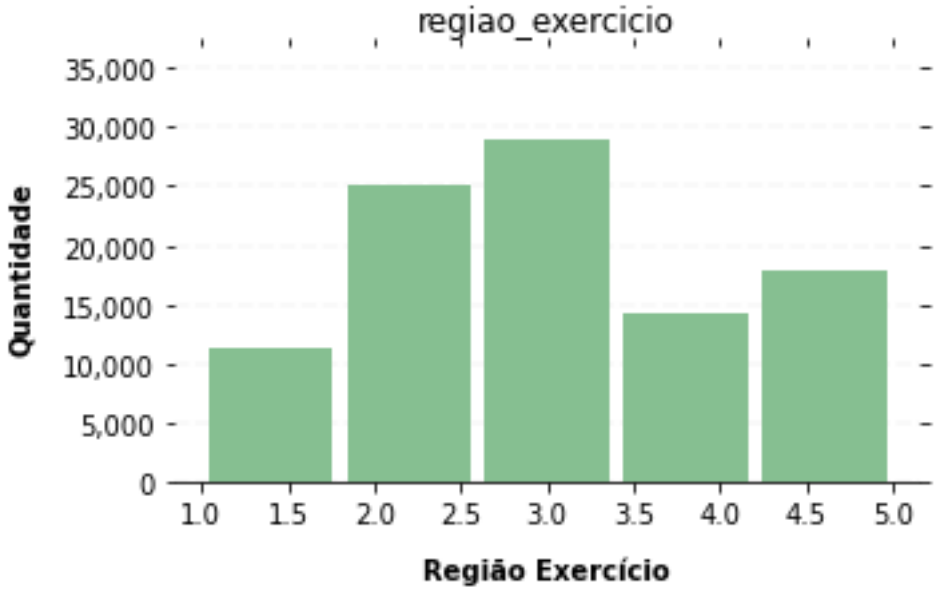
Figura 9 – Quantidade de dados da coluna “orgsup_locacao”



Fonte: Autor

10. uf_exercicio - Existem 27 dados diferentes nesta coluna e estão em formato *string*. Sabendo que cada UF faz parte de uma região do pais, os dados de UF foram substituídos por dados de região no formato inteiro, para o algoritmo trabalhar melhor. O nome da coluna também foi alterada para regiao_exercicio, para fazer mais sentido com o novo dado salvo na coluna. Por existir 5 regiões diferentes nos dados, foi necessário realizar um *dummy* na coluna, deletando a coluna atual e criando 5 novas colunas com o nome de prefixo “regiao_exercicio_” + valor.

Figura 10 – Quantidade de dados da coluna “uf_exercicio”



Fonte: Autor

Com a análise de dados coluna a coluna finalizada, abaixo segue a Tabela 2 com uma visão geral de como ficou a estrutura de dados após a preparação deles com os passos anteriores. Lembrando que a coluna 6 foi dividida em 2, por isso a coluna ID esta mesclada e as colunas 9 e 10 foram feitos o d*ummy* delas.

Tabela 2 – Visão geral das colunas após preparação dos dados

ID	Campo	Tipo de dado
1	tem_cartao	Binário (0 ou 1)
2	sexo	Binário (0 ou 1)
3	idade	Inteiro
4	salario	Decimal
5	regime_juridico	Binário (0 ou 1)
6	jornada_trabalho	Inteiro
	jornada_trabalho_dedicacao_exclusiva	Binário (0 ou 1)
7	tipo_vinculo	Binário (0 ou 1)
8	situacao_vinculo	Binário (0 ou 1)
9	orgsup_lotacao	*Dummy (16* dados)
10	regiao_exercicio	*Dummy (*5 dados)

Fonte: Autor

4 AVALIAÇÃO

4.1 Considerações iniciais

Vamos descrever agora os recursos de hardware e software utilizados, e o algoritmo escolhido, assim como os experimentos, as decisões e as métricas utilizadas para configurar o modelo e chegar no resultado. Por fim, serão exibidos e comparados os resultados obtidos a partir da utilização do modelo selecionado aplicado em dados reais.

4.2 Recursos utilizados

Para programar, configurar, analisar e rodar e extrair o resultado do modelo, foi utilizado um notebook com processador Intel(R) Core(TM) i7-1165G7 @ 2.80GHz de 8 núcleos lógicos, com 12 GB de memória RAM e 120 GB de SSD NVME. O sistema operacional utilizado foi o sistema operacional Linux Ubuntu 22.04.3 LTS x86-64bits. Para manipular os dados, foi utilizado o banco de dados MariaDB, arquivos tipo csv e o LibreOffice Calc. Para programação, foi utilizado a distribuição do Anaconda (Python 3.9.13, 64 bits). Abaixo esta listado as bibliotecas que foram utilizadas no projeto:

* pandas
* xgboost
* numpy
* shap
* sklearn
* warnings
* plotclassification
* pylab
* matplotlib

4.3 Experimentos propostos

A variável alvo deste estudo se refere ao fato do indivíduo possuir ou não um cartão de crédito consignado, que consiste no dado supervisionado. As demais variáveis, listadas acima, são usadas na classificação.

Para realizar os experimentos propostos neste projeto, foi escolhido o algoritmo *XGBoost* para gerar o modelo. Para poder comparar melhor e tentar chegar em melhores

resultados, foram utilizadas duas entradas de dados diferentes para rodar o algoritmo. A primeira com a tabela completa de dados e na outra, foi separado todos os dados que tem cartão da base completa e aleatoriamente foi separado a mesma quantidade de dados que não tem cartão e juntado os dados, ficando ela com 50% dos dados com servidores que tem cartão de crédito consignado e 50% que não tem para levar em conta o problema de desbalanceamento das classes. Além disto, para compreender melhor os resultados, foi utilizado uma regressão linear múltipla para explicar a direção de influência de alguns dados no resultado final.

Para realizar o treinamento e validação dos dados do modelo proposto, é necessário separar uma parte da base para utilizar no treinamento e outra parte para realizar os testes. Neste caso, a base foi separada em 70% para realizar o treinamento e 30% para realizar os testes de validação. Depois de usar a *seed* para gerar o parâmetro *random_state*, mantive este valor constante para garantir a reprodutibilidade deste trabalho. A partir da entrada principal, foi utilizado o valor 1510 como parâmetro *random_state*, que serve para especificar um numero inicial para criar uma sequência pseudoaleatória.

Quando o algoritmo termina de rodar e é realizada a predição, o retorno vem em formato decimal em um range entre 0 e 1. Para facilitar o estudo, o retorno foi transformado em valor binário (0 e 1), sendo que valores maiores que 0,5 foram considerados igual a 1, o restante foi considerado como 0.

4.3.1 Modelo considerando entrada de dados completa

Após a escolha do algoritmo a ser utilizado para gerar o modelo, foi proposta a primeira tentativa para saber como este iria se comportar.

Para realizar este primeiro treinamento inicial, foram utilizados os seguintes parâmetros para configurar o algoritmo:

Tabela 3 – Parâmetros do primeiro treinamento inicial

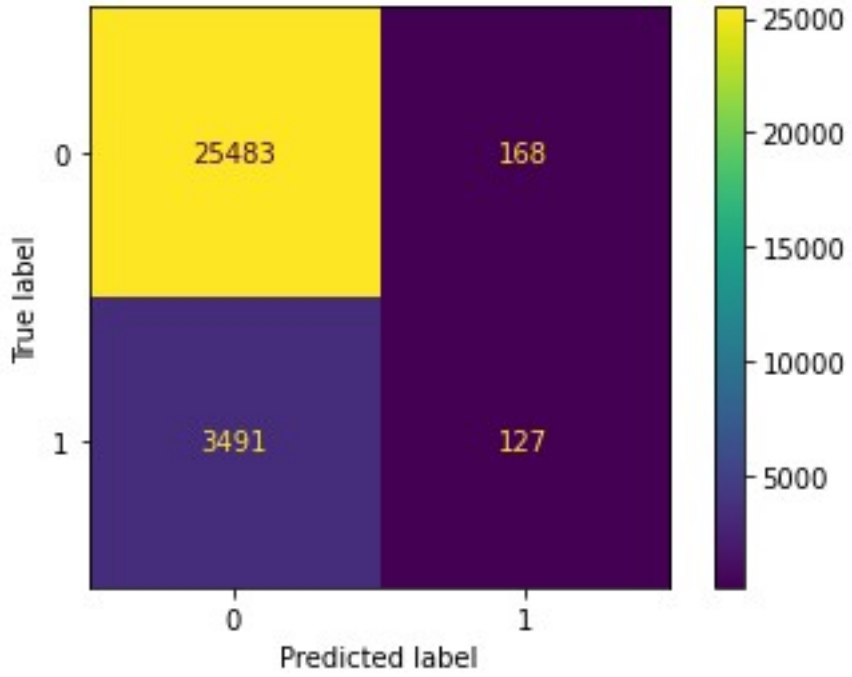
Parâmetro	Valor
learning_rate	0.3
max_depth	2
colsample_bytree	1
subsample	1
min_child_weight	1
gamma	0

random_state	1510
eval_metric	“auc”
objective	Binary:logistic

Fonte: Autor

Como resultado de rodar o algoritmo com os dados de entrada e de parâmetros propostos, chegou-se no resultado da figura 11 e tabela 3 abaixo, ilustrando a matriz de confusão e os dados da predição.

Figura 11 – Matriz de confusão da primeira tentativa de rodar o modelo



Fonte: Autor

Figura 12 – Relatório de classificação da primeira tentativa de rodar o modelo

	precision	recall	f1-score	support
0	0.88	0.99	0.93	25651
1	0.43	0.04	0.06	3618
accuracy			0.87	29269
macro avg	0.66	0.51	0.50	29269
weighted avg	0.82	0.87	0.83	29269

Fonte: Autor

Para melhorar os resultados, o XGBoost tem uma função que utiliza o chamado método GridSearchCV, o qual realiza testes para saber qual os melhores parâmetros a serem passados para o algoritmo para obtenção dos melhores resultados. Para parametrizar a chamada, foi utilizada a seguinte configuração:

Tabela 4 – Parâmetros do primeiro GridSearchCV

Parâmetro	Valor
learning_rate	[0.05,0.3]
max_depth	range(2, 9, 2)
colsample_bytree	[0.5, 1]
subsample	[0.9, 1]
min_child_weight	range(1,5,1)
gamma	[0, 0.1]
random_state	[1510]
n_estimators	range(200, 2000, 200)
booster	["gbtree"]

Fonte: Autor

Após rodar o modelo, o método obteve como resultado os parâmetros a serem utilizados para retornar o melhor resultando. Abaixo seguem os novos parâmetros utilizados para rodar o treinamento:

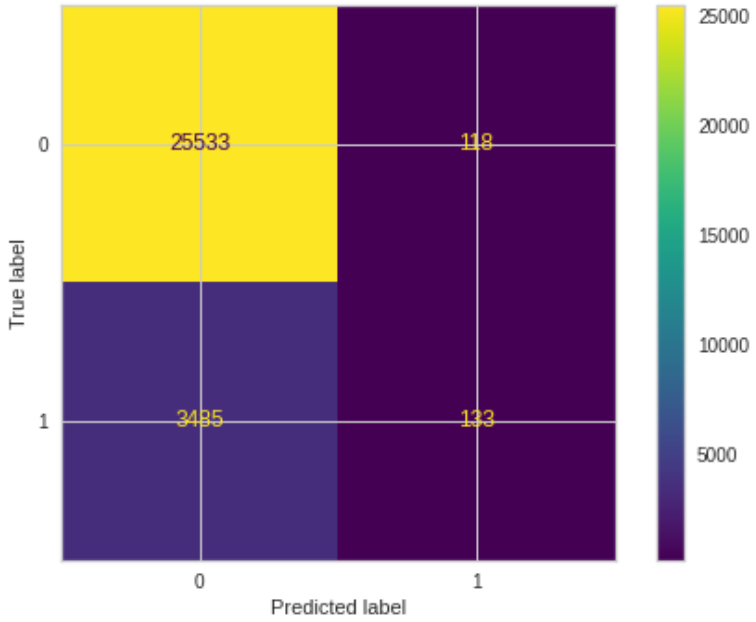
Tabela 5 – Parâmetros do segundo treinamento inicial

Parâmetro	Valor
learning_rate	0.05
max_depth	4
colsample_bytree	0.5
subsample	0.9
min_child_weight	3
gamma	0.1
random_state	1510
eval_metric	“auc”
objective	Binary:logistic

Fonte: Autor

Com esta configuração, o resultado foi:

Figura 13 – Matriz de confusão da segunda tentativa de rodar o modelo



Fonte: Autor

Figura 14 – Relatório de classificação da segunda tentativa de rodar o modelo

	precision	recall	f1-score	support
0	0.88	1.00	0.93	25651
1	0.53	0.04	0.07	3618
accuracy			0.88	29269
macro avg	0.70	0.52	0.50	29269
weighted avg	0.84	0.88	0.83	29269

Fonte: Autor

Percebe-se que foi alcançada uma pequena melhora com as novas parametrizações, entretanto, não foi significativa. Conforme relatório de classificação da primeira e segunda tentativas, houve uma pequena melhora nas métricas de precision, recall e f1-score. No entanto, a matriz de confusão, em porcentagem, praticamente não se alterou.

Embora os testes de acurácia sejam bons, verifica-se pelo teste F1 que os resultados estão viciados pela classificação da classe majoritária, dado o desbalanceamento das classes: mais de 80% dos servidores não possuem o cartão consignado.

Foi proposta, então, a realização de testes com uma entrada padronizada, conforme proposta no próximo tópico.

4.3.2 Modelo considerando entrada de dados com 50% do tamanho com cada valor

Conforme foi dito acima, esta sessão teve uma alteração nos dados de entrada, quando foram separados todos os dados o qual o servidor tem o cartão de crédito consignado e por outro lado, de forma aleatória, foi buscado a mesma quantidade de linhas de dados o qual não tem cartão, fazendo com que a base de entrada tenha 50% dos dados com cartão e os outros 50% que não tem.

Para o primeiro treinamento, foi utilizado os seguintes valores de parâmetros:

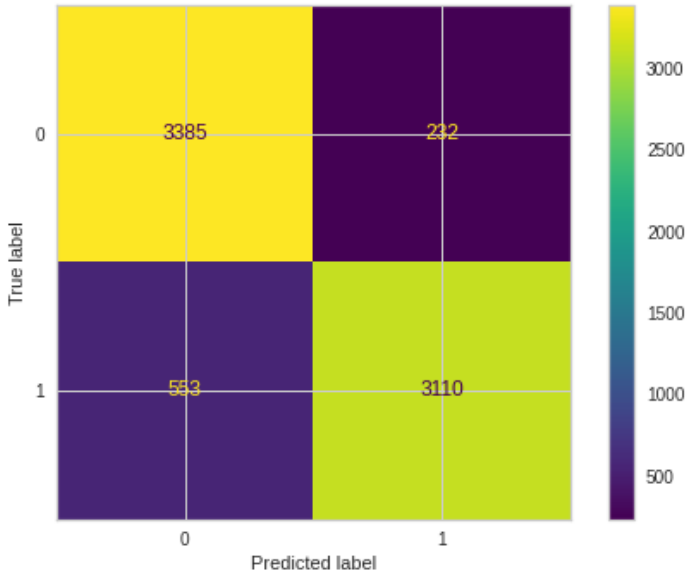
Tabela 6 – Parâmetros do segundo treinamento inicial

Parâmetro	Valor
learning_rate	0.3
max_depth	2
colsample_bytree	1
subsample	1
min_child_weight	1
gamma	0
random_state	1510
eval_metric	“auc”
objective	Binary:logistic

Fonte: Autor

Com esta configuração, o resultado está sendo demostrado na figura 13 que representa a matriz de confusão e na tabela 5, que representa o relatório de confusão.

Figura 15 – Matriz de confusão da primeira tentativa de rodar o modelo com 50% dos dados de cada valor



Fonte: Autor

Figura 16 – Relatório de classificação da primeira tentativa de rodar o modelo com 50% dos dados de cada valor

	precision	recall	f1-score	support
0	0.86	0.94	0.90	3617
1	0.93	0.85	0.89	3663
accuracy			0.89	7280
macro avg	0.90	0.89	0.89	7280
weighted avg	0.90	0.89	0.89	7280

Fonte: Autor

Percebe-se que houve uma melhora nos resultados, apenas padronizando os dados de entrada. Agora, vamos verificar e analisar os melhores parâmetros a serem utilizados com a nova entrada de dados, buscando os melhores resultados:

Tabela 7 – Parâmetros do primeiro GridSearchCV

Parâmetro	Valor
learning_rate	[0.05,0.3]
max_depth	range(2, 9, 2)
colsample_bytree	[0.5, 1]

subsample	[0.9, 1]
min_child_weight	range(1,5,1)
gamma	[0, 0.1]
random_state	[1510]
n_estimators	range(200, 2000, 200)
booster	["gbtree"]

Fonte: Autor

Após rodar o método, teve como resultado os parâmetros a serem utilizados para retornar o melhor resultando do modelo. Abaixo seguem os novos parâmetros utilizados para rodar o treinamento:

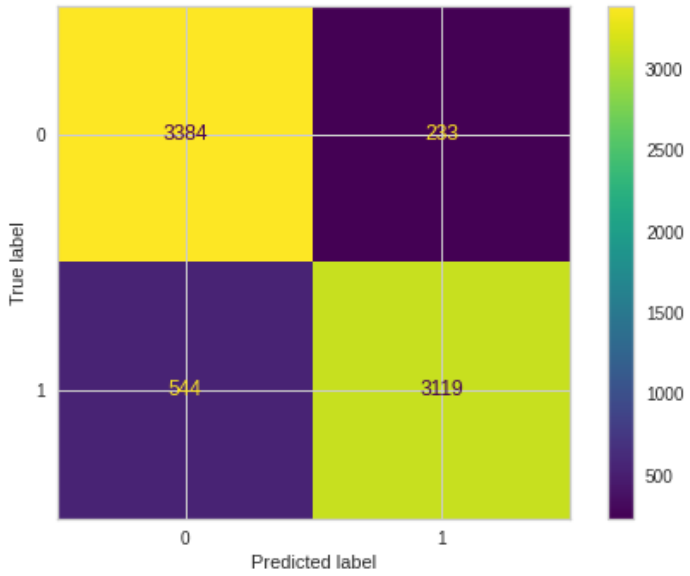
Tabela 8 – Parâmetros do segundo treinamento inicial

Parâmetro	Valor
learning_rate	0.05
max_depth	6
colsample_bytree	0.5
subsample	0.9
min_child_weight	2
gamma	0
random_state	1510
eval_metric	“auc”
objective	Binary:logistic

Fonte: Autor

Com esta configuração, o resultado foi:

Figura 17 – Matriz de confusão da segunda tentativa de rodar o modelo 50% dos dados de cada valor



Fonte: Autor

Figura 18 – Relatório de classificação da segunda tentativa de rodar o modelo com 50% dos dados de cada valor

	precision	recall	f1-score	support
0	0.86	0.94	0.90	3617
1	0.93	0.85	0.89	3663
accuracy			0.89	7280
macro avg	0.90	0.89	0.89	7280
weighted avg	0.90	0.89	0.89	7280

Fonte: Autor

Conforme podemos perceber, não teve melhorias significativas no resultado ao alterarmos os parâmetros de configuração do algoritmo indicados por ele próprio para obter melhores resultados. Neste caso, com as classes balanceadas, verifica-se que o F1-score tornou-se alto para as duas classes.

Por outro lado, percebe-se uma melhora significativa no resultado quando comparado aos dados de entrada completos. Conseguimos visualizar isto ao comparar as figuras 13 com 17, onde a matriz de confusão mostrou valores bem mais altos para os quadros verdadeiro positivo e falso verdadeiro, e valores baixo para falso positivo e falso negativo.

Para adicionar mais informações no retorno e no resultado do modelo, na próxima sessão será demonstrada a utilização da regressão linear múltipla.

4.3.3 Modelo regressão linear múltipla

Para melhorar as explicações do retorno encontrado utilizando o algoritmo *XGBoost*, e para buscar mais informações como verificar o sentido e a influência dos campos nos dados encontrados, também foi realizado um estudo utilizando a regressão linear múltipla.

Como observação, para analisarmos os dados de retorno da regressão linear múltipla mostrados na tabela abaixo, primeiro precisamos informar que os resultados dela pode ser conflitante com os dados retornados na utilização do algoritmo *XGBoost*, por serem modelos diferentes. Mesmo sabendo disto, utilizei a regressão linear para enriquecer os estudos, por fornecer dados complementares o qual o *XGBoost* consegue nos retorna.

Figura 19 – Resultado da regressão linear múltipla

OLS Regression Results

Dep. Variable:	tem_cartao	R-squared:	0.087
Model:	OLS	Adj. R-squared:	0.087
Method:	Least Squares	F-statistic:	345.2
Date:	Fri, 01 Dec 2023	Prob (F-statistic):	0.00
Time:	23:12:29	Log-Likelihood:	-25817.
No. Observations:	97561	AIC:	5.169e+04
Df Residuals:	97533	BIC:	5.196e+04
Df Model:	27		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.3470	0.012	-28.266	0.000	-0.371	-0.323
sexo	0.0045	0.002	2.220	0.026	0.001	0.009
idade	0.0052	9.48e-05	55.334	0.000	0.005	0.005
regime_juridico	0.0375	0.012	3.186	0.001	0.014	0.061
tipo_vinculo	0.0075	0.004	1.826	0.068	-0.001	0.016
salario	3.518e-09	1.64e-07	0.021	0.983	-3.18e-07	3.26e-07
jornada_trabalho	0.0053	0.000	15.844	0.000	0.005	0.006
jornada_trabalho_dedicacao_exclusiva	-0.1090	0.002	-45.362	0.000	-0.114	-0.104
situacao_vinculo	0.0903	0.011	8.269	0.000	0.069	0.112
regiao_exercicio_1	-0.0216	0.004	-6.061	0.000	-0.029	-0.015
regiao_exercicio_2	-0.0794	0.003	-25.237	0.000	-0.086	-0.073
regiao_exercicio_3	-0.0838	0.003	-26.710	0.000	-0.090	-0.078
regiao_exercicio_4	-0.0928	0.003	-27.113	0.000	-0.100	-0.086
regiao_exercicio_5	-0.0694	0.003	-20.687	0.000	-0.076	-0.063
orgsup_lotacao_3	-0.0302	0.009	-3.285	0.001	-0.048	-0.012
orgsup_lotacao_5	-0.0250	0.004	-6.498	0.000	-0.033	-0.017
orgsup_lotacao_7	0.0680	0.005	13.569	0.000	0.058	0.078
orgsup_lotacao_9	0.0952	0.012	8.060	0.000	0.072	0.118
orgsup_lotacao_13	-0.0914	0.008	-12.177	0.000	-0.106	-0.077
orgsup_lotacao_17	-0.0681	0.008	-8.215	0.000	-0.084	-0.052
orgsup_lotacao_19	-0.0201	0.012	-1.700	0.089	-0.043	0.003
orgsup_lotacao_21	0.0364	0.006	6.105	0.000	0.025	0.048
orgsup_lotacao_25	-0.0222	0.015	-1.447	0.148	-0.052	0.008
orgsup_lotacao_27	0.0233	0.007	3.538	0.000	0.010	0.036
orgsup_lotacao_29	-0.0811	0.012	-6.980	0.000	-0.104	-0.058
orgsup_lotacao_31	-0.0422	0.008	-5.566	0.000	-0.057	-0.027
orgsup_lotacao_33	-0.0413	0.015	-2.790	0.005	-0.070	-0.012
orgsup_lotacao_35	-0.0171	0.038	-0.445	0.656	-0.092	0.058
orgsup_lotacao_37	-0.0602	0.009	-6.607	0.000	-0.078	-0.042
orgsup_lotacao_39	-0.0712	0.013	-5.672	0.000	-0.096	-0.047

Omnibus:	34492.098	Durbin-Watson:	1.997
Prob(Omnibus):	0.000	Jarque-Bera (JB):	92471.216
Skew:	1.977	Prob(JB):	0.00
Kurtosis:	5.668	Cond. No.	1.82e+16

Fonte: Autor

Analisando o retorno, se pegarmos a informação da coluna P>|t|, vemos que a informação importante é que quando o valor for maior que 0.05, significa que o campo não tem uma influência significativa no resultado final do modelo.

Agora analisando os resultados da coluna coef, consegue analisar a direção de influencia do dado no resultado. Abaixo alguns exemplos para melhor explicar esta afirmação.

Pegando como base o campo idade. Por ser um valor muito pequeno, mas positivo, podemos chegar na conclusão que quanto mais velho o servidor for, maior a chance dele adquirir o cartão de crédito consignado.

Agora, analisando o campo sexo, também tem um valor muito pequeno e positivo. O servidor do sexo masculino é mais propenso a adquirir o cartão, lembrando que o valor dele é 1 e feminino = 0.

Agora a coluna tipo_vinculo, a coluna P>|t| tem um valor maior que 0.05, fazendo com que segundo o algoritmo da regressão linear, ele não tem influência sobre o resultado.

Já na coluna salario, chegamos na conclusão que quanto maior o salário, menor a probabilidade de pegar o cartão, pois o coef esta negativo. Mesmo com um valor esta bem irrisório, o dado de salario ainda se mostra significativo para o resultado final.

5 CONCLUSÃO

Neste trabalho foi realizada uma análise usando uma base de dados particular de servidores federais brasileiros. Com isto, foi possível identificar vários campos que poderiam ser utilizados para juntar os dados desta base com dados do governo federal, disponibilizados pelo portal da transparência, enriquecendo ainda mais as informações que a base já continha. Com isto, foi possível adicionar vários campos úteis, além dos que já tinha na base privada, ajudando a melhorar a classificação do modelo.

Este trabalho também ajudou a contribuir na limpeza dos dados uteis para a analise do produto de consignado a ser oferecido que foi proposto no projeto, que é de adquirir o cartão de crédito consignado. Na análise e tratamento dos dados, foram encontradas as informações que determinam se o produto poderia ou não ser adquirido, tais como servidor idade do servidor, gênero e alguns tipos de vínculos não consignáveis, entre outras situações.

Deixando mais clara a utilização da regressão linear múltipla. Ela foi utilizada para ajudar a explicar a direção de importância das variáveis o qual a base continha. O XGBoost mostra a importância relativa de cada variável para classificar, mas faltava saber a direção. E como observação, embora esteja utilizando a regressão linear múltipla, ela não é perfeita para explicar a importância da variável no resultado final, visto que por estarem sendo utilizados metodologias diferentes, alguns resultados podem se mostrar conflitantes. Mas para a analise da direção do efeito, funciona muito bem, me ajudando a realizar uma compreensão mais assertiva em cima dos dados utilizados.

Sobre o resultado final, o modelo com acertos de 89% poderia ter sido maior. Entretanto, os dados que foram adquiridos para a previsão, mesmo através de busca de dados no portão da transparência, ainda são muito gerais, onde a maioria dos servidores tem valores iguais ou muito parecidos. Neste sentido, acredito, que o acerto encontrado foi bem satisfatório, pois permite uma boa separação preliminar de quem está mais propenso a contratar o cartão consignado. O que ajuda as empresas de consignado separar quem realmente pode ter interesse no produto o qual ela quer ofertar, diminuindo os gastos com tentativas de contatos com clientes que não tem interesse nenhum em adquirir o produto.

Em trabalhos futuros, pretende-se aumentar o estudo dos produtos a serem ofertados, como o empréstimo consignado, seguro do cartão consignado, entre outros.

REFERÊNCIAS

Coelho, C. A., De Mello, J. M., & Funchal, B. (2012). **The Brazilian payroll lending experiment**. Review of economics and statistics, 94(4), 925-934.

Arram, A., Ayob, M., Albadr, M. A. A., Sulaiman, A., & Albashish, D. (2023). **Credit card score prediction using machine learning models**: A new dataset. arXiv preprint arXiv:2310.02956.

Paula, D. A. V. D., Artes, R., Ayres, F., & Minardi, A. M. A. F. (2019). **Estimating credit and profit scoring of a Brazilian credit union with logistic regression and machine-learning techniques**. RAUSP Management Journal, 54, 321-336.

Schuh, A. B., Marion Filho, P., & Coronel, D. (2019). **Determinants of the Default Rate of Individual Clients in Brazil and the Role of Payroll Loans**. Economics Bulletin, 39(1), 395-408.

COMIN, C.H. et al. **Complex systems**: features, similarity and connectivity. Physics Reports, n.861, p.1–41, 2020. Doi: 10.48550/arXiv.1606.05400

NEWMAN, M.E.; GIRVAN, M. **Finding and evaluating community structure in networks**. Physical Review E, n.69(2), p.26113-26129, Feb 2004. Doi: 10.1103/PhysRevE.69.026113.

PONTES, ALCIONI; LOPES, PALOMA. **Estratégias de captação e fidelização de clientes de crédito consignado**. Revista Valore,Volta Redonda, 2 (1): 34-50., Junho/2017

GONÇALVES, GUILHERME. **Endividamento pessoal**: Uma análise a partir da utilização do crédito consignado por servidores públicos. Brasília. 2021.

Reis, L. G. Moura. **Aprendizado de máquina na previsão de vazões mínimas a partir de índices de seca e produtos derivados de sensoriamento remoto em escala global.** 2022.

CHEN, T.; GUESTRIN, C. **XGBoost**: A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ago/2016.