

**UNIVERSIDADE CATÓLICA DE PETRÓPOLIS
CENTRO DE ENGENHARIA E COMPUTAÇÃO
PROGRAMA DE MESTRADO PROFISSIONAL EM GESTÃO DE
SISTEMAS DE ENGENHARIA**

Rafael Ribas Aguiló

Detecção de Fraude de ICMS Utilizando Redes Neurais Artificiais

Petrópolis – RJ

2022

Rafael Ribas Aguiló

Detecção de Fraude de ICMS Utilizando Redes Neurais Artificiais

Dissertação apresentada ao Programa de Mestrado Profissional em Gestão de Sistemas de Engenharia da Universidade Católica de Petrópolis, na área de concentração em Sistemas de Engenharia – Modelagem Computacional, como requisito parcial para a obtenção do título de Mestre em Gestão de Sistemas de Engenharia.

Orientador: Prof. Dr. Giovane Quadrelli

Petrópolis – RJ

2022

CIP – Catalogação na Publicação

A283d Aguiló, Rafael Ribas.
Detecção de fraude de ICMS utilizando redes neurais
artificiais / Rafael Ribas Aguiló. – 2022.
85 f. : il.

Dissertação (Mestrado Profissional em Gestão de
Sistemas de Engenharia) - Universidade Católica de
Petrópolis, 2022.

Orientação: Prof. Dr. Giovane Quadrelli.

Linha de pesquisa: Modelagem Computacional.

1. Fraude. 2. ICMS. 3. Redes Neurais. I. Quadrelli,
Giovane (Orient.). II. Título.

CDD: 006.32

Universidade Católica de Petrópolis (UCP)

Bibliotecária responsável: Marlena H. Pereira – CRB7: 5075

UNIVERSIDADE CATÓLICA DE PETRÓPOLIS
Centro de Engenharia e Computação
Programa de Mestrado Profissional em Gestão de Sistemas de Engenharia

“Detecção de Fraude de ICMS Utilizando Redes Neurais Artificiais”

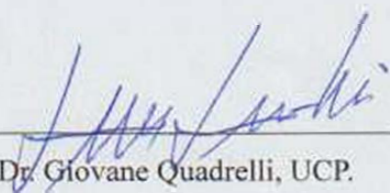
Mestrando: **Rafael Ribas Aguiló**

Orientador: **Giovane Quadrelli**

Coorientador: **Fábio Lopes Licht**

Petrópolis, 16 de dezembro de 2022.

Banca Examinadora:



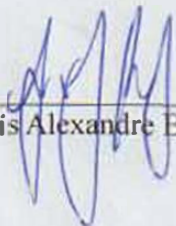
Prof. Dr. Giovane Quadrelli, UCP.



Prof. Dr. Fábio Lopes Licht, UCP.



Prof. Dr. Bruno Richard Schulze, UCP.



Prof. Dr. Luis Alexandre Estevão da Silva, IPJBRJ.

Dedico este trabalho aos meus pais, Gabriel e Maria, que me ensinaram o caminho certo para tudo e que o conhecimento, uma vez adquirido, ninguém consegue lhe tirar; aos meus irmãos, Gabriel, Alexandre e Jéssica, que me incentivaram o tempo todo; à minha esposa, Flávia, a grande responsável por esta conquista, quem me deu o incentivo inicial e todo o apoio necessário durante essa longa jornada; aos meus filhos Guilherme e Eduardo, aos quais espero dar um grande exemplo através deste esforço e pedir a compreensão sobre os momentos de ausência mútua que passamos por este período.

AGRADECIMENTOS

Ao professor Dr. Giovane Quadrelli por todo o seu entusiasmo, apoio, inspiração e dedicação na condução desse trabalho de pesquisa do programa de pós-graduação. Aos professores Fábio Licht, Dr. Renato Portugal e Dr. Nélcio Domingues Pizzolato, que me inspiraram e deram o apoio necessário, sem restrições, durante todo o curso. À Tatiane Teodoro Correia, por todo seu suporte irrestrito para nos atender em qualquer necessidade durante todo o tempo.

Aos meus colegas de trabalho, Alexandre Mendonça, Fábio Brabo, Mauro Sato e Yu Shu, que apoiaram, ajudaram e viabilizaram a transformação deste projeto em ferramenta de trabalho da SEFAZ-RJ.

“O maior perigo da ignorância é crer que a busca do conhecimento não é
necessária.”

Rafael Ribas Aguiló

RESUMO

Existe, atualmente, um considerável número de empresas que fraudam o ICMS, onde empresas de fachada são criadas para emitir notas fiscais falsas, com o propósito de gerar crédito tributário às empresas compradoras. Elas são conhecidas como empresas “noteiras”. Identificar e impedir o funcionamento dessas empresas, o mais rápido possível, é primordial para evitar a perda de receitas no sistema de arrecadação do Estado do Rio de Janeiro. A fim de oferecer uma ferramenta ao grupo de inteligência da SEFAZ-RJ para identificar fraudadores do ICMS do Estado do Rio de Janeiro, esta dissertação propõe um método para apontar as informações fiscais mais relevantes e identificar o modelo de rede neural de classificação mais eficaz bem como definir o menor período de coleta das informações fiscais necessárias a antecipar, ao máximo, a identificação dos fraudadores. Em busca de alcançar este objetivo, foi utilizado o método estatístico de “*Seleção de Características*” para a identificação das variáveis mais relevantes a serem utilizadas no modelo de redes neurais para classificação das empresas, conhecido como *perceptron* de multicamadas, que foi criado, treinado e algumas de suas arquiteturas comparadas em termos de eficiência e eficácia com a intenção de se chegar no modelo mais adequado na identificação dos fraudadores e constituir os fundamentos básicos para a identificação precoce de empresas “noteiras”. O resultado obtido neste estudo tem o potencial de contribuir, não só em favor do estado do Rio de Janeiro, mas, também, em favor de qualquer outro estado brasileiro, pois, trata-se de um problema nacional, além de gerar uma maior e mais rápida recuperação da sua receita, melhorando sua situação financeira e, conseqüentemente, permitindo mais investimentos no bem-estar de sua população.

Palavras-chave: Fraude, *Fuzzy*, ICMS, Noteira, Redes Neurais, SMOTE

ABSTRACT

There are currently a considerable number of companies that defraud ICMS tax collection, where shell companies are created to issue counterfeit invoices, with the purpose of generating tax credit to the purchasing companies. They are known as “noteiras” companies. Identifying and preventing the operation of these companies as soon as possible is essential to avoid the loss of revenue in the collection system of the State of Rio de Janeiro. In order to provide a tool to the intelligence group of SEFAZ-RJ to identify ICMS fraudsters in the State of Rio de Janeiro, this dissertation proposes a method to point out the most relevant tax information and identify the most effective classification neural network model as well as to define the shortest period of collection of the tax information necessary to anticipate the identification of fraudsters as much as possible. In order to achieve this objective, the statistical method of “Feature Selection” was used to identify the most relevant variables to be used in some models of neural networks for classification of companies, using *multi-layer perceptron*, which was created, trained and some of its architectures compared in terms of efficiency and effectiveness with the intention of reaching the most appropriate model for the identification of fraudsters and to provide the basis for the early identification of such companies.

The result obtained in this study has the potential to contribute, not only in favor of the state of Rio de Janeiro, but also in favor of any other Brazilian state, because it is a national problem, besides generating a greater and faster recovery of its revenue, improving its financial situation and, consequently, allowing more investments in the well-being of its population.

Palavras-chave: Fraude, *Fuzzy*, ICMS, Noteira, Redes Neurais, SMOTE

LISTA DE ILUSTRAÇÕES

Figura 1 - Histórico de abertura e fechamento de empresas no Estado do Rio de Janeiro.....	2
Figura 2- Comparação da evolução de arrecadação de ICMS no RJ entre 2017 e 2021	4
Figura 3 - Processo normal de recolhimento de ICMS	5
Figura 4 - Processo de fraude onde não recolhe o ICMS devido	6
Figura 5 - Quantidade de Publicações por Palavras-Chave.....	12
Figura 6 - Visão geral da inteligência artificial	15
Figura 7- Sistema de lógica nebulosa	16
Figura 8- Representação de um neurônio natural	19
Figura 9 - Representação de um neurônio artificial.....	19
Figura 10 - Processo do aprendizado de máquina.....	21
Figura 11 - Processo da rede neural	22
Figura 12- Tipos de Máquinas de Aprendizado	22
Figura 13 - Grafo da arquitetura de uma rede neural com duas camadas escondidas	23
Figura 14 - Diagrama em blocos da aprendizagem com um professor (supervisionada)	24
Figura 15 – Diagrama de blocos de uma rede neural destacando um único neurônio na camada de saída	26
Figura 16 Rede neural exemplo	26
Figura 17- Revista Forbes - Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says.....	29
Figura 18- Funções de pertinência da variável Idade.....	33
Figura 19- Funções de Pertinência de Compras	34
Figura 20- Funções de Pertinência de Arrecadação	35
Figura 21 - Funções de Pertinência de Grau Fraudadora	37
Figura 22- Correlação das variáveis independentes com a variável dependente “Noteira”	40
Figura 23 - Pontos sintéticos (s1 a s5) gerados pelo SMOTE ao longo das linhas de conexão entre um ponto (ponto preto denotado por Q_i) e seus k vizinhos mais próximos (pontos pretos)	43

Figura 24- Representação da Descida do Gradiente (com o objetivo de minimizar a função de custo)	44
Figura 25- Exemplo de arquitetura da rede neural multi-layer <i>perceptron</i>	47
Figura 26 - Processo de aprendizado de uma rede neural	48
Figura 27- Técnica <i>k-fold</i> para treinamento e teste da rede neural	49
Figura 28- Matriz confusão	50
Figura 29- Curva característica de operação do receptor do modelo de classificação	53
Figura 30- Tela exemplo do sistema SARF	56

LISTA DE TABELAS

Tabela 1 - Termos Pesquisados.....	11
Tabela 2- Revisão da Literatura - Publicações Seleccionadas	13
Tabela 3- Dados disponíveis.....	30
Tabela 4- Conjuntos Nebulosos de Idade	33
Tabela 5- Conjuntos Nebulosos de Compras x Vendas.....	34
Tabela 6- Conjuntos Nebulosos de Arrecadação x Vendas	34
Tabela 7- Exemplos de Regras do Sistema Nebuloso de Inferência	35
Tabela 8- Conjuntos Nebulosos de Grau Fraudadora.....	36
Tabela 9- Regras do Sistema Nebuloso de Inferência	37
Tabela 10- Exemplos de entradas e saídas do sistema nebuloso	38
Tabela 11- Variáveis mais relevantes seleccionadas após o resultado da análise de correlação.....	41
Tabela 12- Quantidade de Dados Antes e Depois do SMOTE.....	43
Tabela 13- Modelos de Classificação Estudados.....	46

LISTA DE ABREVIATURAS E SIGLAS

CT-e	Conhecimento de Transporte Eletrônico
DECLAN-IPM	Declaração Municipal para o Índice de Participação dos Municípios
DeSTDA	Declaração de Substituição Tributária, Diferencial de Alíquota e Antecipação
DUB-ICMS	Documento de Utilização de Benefícios Fiscais do ICMS
EFD	Escrituração Fiscal Digital
GIA-ICMS	Guia de Informação de Apuração do ICMS
GIA-ST	Guia Nacional de Informação e Apuração do ICMS Substituição Tributária
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
ICMS	Imposto sobre Circulação de Mercadorias e Serviços de transporte interestadual e intermunicipal e de comunicação
MDF-e	Manifesto Eletrônico de Documentos Fiscais
NF	Nota Fiscal
NF-e	Nota Fiscal Eletrônica
RNA	Rede Neural Artificial
ROC	<i>Receiver Operating Characteristic</i>
SARF	Sistema de Análise de Regularidade Fiscal
SEFAZ	Secretaria de Fazenda
ST	Substituição Tributária
UCP	Universidade Católica de Petrópolis

1.	INTRODUÇÃO.....	1
2.	CONTEXTUALIZAÇÃO E MOTIVAÇÃO.....	3
2.1	O ICMS.....	3
2.2	O SISTEMA DE ARRECADAÇÃO DO ICMS NO ESTADO DO RIO DE JANEIRO.....	3
2.3	DEFINIÇÃO DO PROBLEMA.....	5
2.4	JUSTIFICATIVA E IMPORTÂNCIA	6
2.5	QUESTÕES DA PESQUISA	9
2.6	OBJETIVOS	9
2.7	METODOLOGIA DE PESQUISA	10
3.	REVISÃO DA LITERATURA.....	11
3.1	TRABALHOS RELACIONADOS.....	11
4.	APRENDIZADO DE MÁQUINA.....	14
4.1	LÓGICA NEBULOSA (FUZZY LOGIC)	15
4.2	REDES NEURAIS ARTIFICIAIS.....	18
4.2.1	HISTÓRICO	18
4.2.2	APLICAÇÃO	21
4.2.3	TIPOS DE REDE NEURAL	22
4.2.4	ARQUITETURAS MULTI-LAYER PERCEPTRON	23
4.2.5	FUNÇÕES DE ATIVAÇÃO	24
4.2.6	TREINAMENTO E APRENDIZADO	24
5.	DESENVOLVIMENTO DA SOLUÇÃO	29
5.1	SOBRE DOS DADOS	29
5.2	A OBTENÇÃO DOS DADOS.....	30
5.3	O PROBLEMA NA ROTULAÇÃO DOS DADOS	32
5.4	UM MODELO NEBULOSO PARA AJUDAR NA ROTULAÇÃO DOS DADOS	32
5.5	A IDENTIFICAÇÃO DAS INFORMAÇÕES MAIS RELEVANTES	39
5.6	O PROBLEMA DAS AMOSTRAS DESBALANCEADAS.....	42
5.7	PADRONIZAÇÃO DOS DADOS.....	43
6.	O MODELO ATUAL DE DETECÇÃO DE FRAUDE	45

7.	O MODELO PROPOSTO	46
7.1	PROCESSO DE CONSTRUÇÃO.....	47
7.2	TREINAMENTO E VALIDAÇÃO	48
8.	RESULTADOS.....	50
8.1	ANÁLISE DOS RESULTADOS	50
9.	CONCLUSÕES	54
10.	SUGESTÕES DE TRABALHOS FUTUROS	56
	REFERÊNCIAS BIBLIOGRÁFICAS.....	57
	APÊNDICE A – SCRIPT PYTHON DO SISTEMA FUZZY PARA ELIMINAR RUÍDOS NA ROTULAÇÃO DOS DADOS SOBRE EMPRESAS NOTEIRAS	62
	APÊNDICE B – SCRIPT PYTHON DA REDE NEURAL PARA A CLASSIFICAÇÃO DE EMPRESAS NOTEIRAS	69

1. INTRODUÇÃO

O ICMS, imposto sobre circulação de mercadorias e serviços de transporte interestadual e intermunicipal e de comunicação, previsto no artigo 155, II, da Constituição da República de 1998 e está em vigor desde 1989, regulamentado pela Lei Complementar nº 87/1996, é a principal fonte de recursos do estado. É com os recursos advindos deste imposto que o estado paga professores, policiais militares e civis e demais funcionários. O imposto financia também a construção e melhorias de escolas estradas e saneamento básico. Além disso, vinte e cinco por cento, do que é arrecadado, são repassados às prefeituras de todos os municípios do estado para pagarem os professores municipais, agentes comunitários, postos de saúde e a merenda escolar. Percebe-se, assim, que a importância desse imposto é enorme e a manutenção de sua arrecadação é essencial para a saúde de toda a cadeia dependente dele. Por isso, mais do que a busca de novas receitas, é importante evitar as perdas.

A lei do IPI, no art. 72, diz que *“Fraude é toda ação ou omissão dolosa tendente a impedir ou retardar, total ou parcialmente, a ocorrência do fato gerador da obrigação tributária principal, ou a excluir ou modificar as suas características essenciais, de modo a reduzir o montante do imposto devido a evitar ou diferir o seu pagamento.”*. Desse modo, é necessário que o trabalho de fiscalização seja aperfeiçoado para se ter maior eficiência e eficácia na identificação de fraudadores que prejudicam a economia do estado e promovem uma concorrência desleal.

Segundo (SAYEG, 2003), “Detectar as sonegações têm sido uma tarefa árdua e cada vez mais difícil para os fiscos envolvidos. O universo econômico-mercadológico contemporâneo, apoiado em sistemas de informação cada vez mais complexos, evolui numa escala sem precedentes, criando um "gap" entre o potencial do ente fiscalizador para detectar a sonegação e o potencial do contribuinte para praticá-la.”.

Com o aumento de abertura de novas empresas a importância da fiscalização ganha ainda mais destaque. Segundo o (SEBRAE, 2022), em 2021 tivemos um saldo positivo de 267.043, entre as aberturas e fechamentos de empresas

no Estado do Rio de Janeiro. A Figura 1 mostra o aumento desse volume diminui ainda mais a capacidade da fiscalização estadual.

Figura 1 - Histórico de abertura e fechamento de empresas no Estado do Rio de Janeiro



Fonte: (DataSebrae, 2022)

Contudo, atualmente, há uma grande variedade de informações disponíveis, decorrentes de vários tipos de declarações, movimentações de notas fiscais e outros que podem ajudar a fiscalização a perceber a ocorrência de fraudes. Porém, o grande volume de informações tornou humanamente impossível lidar com todas elas para se chegar a uma conclusão. Fazer esse trabalho manualmente, mesmo com a ajuda de computadores e software, é uma tarefa exaustiva devido à imensa quantidade e variedade de informações, assim como todas as possíveis combinações que poderiam indicar diferentes comportamentos de fraude. É preciso que máquinas aprendam a fazer isso automaticamente. O aprendizado de máquina é uma ferramenta multidisciplinar e aplicável em diversos campos devido à sua importante propriedade de "aprender a partir de dados de entrada com ou sem um professor" Haykin (1999). É através da utilização dessa característica, combinada com redes neurais artificiais que este estudo propõem a definição dos fundamentos básicos para identificar os dados mais relevantes, o menor período possível de coleta de informações fiscais das empresas e o modelo de rede neural mais eficiente e eficaz para a identificação de empresas "noteiras" no Estado do Rio de Janeiro.

2. CONTEXTUALIZAÇÃO E MOTIVAÇÃO

2.1 O ICMS

O ICMS é o imposto que incide “sobre operações relativas à circulação de mercadorias e sobre prestações de serviços de transporte interestadual e intermunicipal e de comunicação, ainda que as operações e as prestações se iniciem no exterior”, (BRASIL, 2022), previsto na Constituição de 1988, é aplicado quando a posse de uma mercadoria é transmitida a outro dono, ou seja, quando a mercadoria é vendida ou o serviço é prestado e o consumidor passa a ser o titular do produto ou atividade. De acordo com (SABBAG, 2017), “o fato gerador indica quaisquer atos ou negócios, independentemente da natureza jurídica específica de cada um deles, que implicam a circulação de mercadorias, assim entendida a circulação capaz de realizar o trajeto da mercadoria da produção até o consumo.”, onde se entende circulação por “a mudança de titularidade jurídica do bem (não é mera movimentação “física”, mas circulação jurídica do bem). O bem sai da titularidade de um sujeito e passa à titularidade definitiva de outro” (SABBAG, 2017).

O princípio da não cumulatividade, estabelecido na lei do ICMS, define que “será não-cumulativo, compensando-se o que for devido em cada operação relativa à circulação de mercadorias ou prestação de serviços com o montante cobrado nas anteriores pelo mesmo ou outro Estado ou pelo Distrito Federal”.

2.2 O SISTEMA DE ARRECADAÇÃO DO ICMS NO ESTADO DO RIO DE JANEIRO

O Decreto Lei nº 10/1975, em seu artigo 1º, instituiu que a Inspetoria Geral de Finanças seria o órgão responsável pelo Sistema Estadual de Administração Financeira e Contabilidade do Estado do Rio de Janeiro. Em 2003, o Decreto nº 32.621 alterou a estrutura do Poder Executivo do Estado do Rio de Janeiro, nomeando a Secretaria de Estado de Fazenda com um dos Órgãos de Ação Setorial de Governo e logo depois, dividindo-a em Secretaria de Fazenda em Secretaria de Estado de Receita e Secretaria de Estado de Finanças, sendo a Contadoria Geral do Estado considerada como órgão de atividade fim da Secretaria de Estado de Finanças. Em 2007 retorna à estrutura do Estado a Secretaria de Estado de Fazenda através do

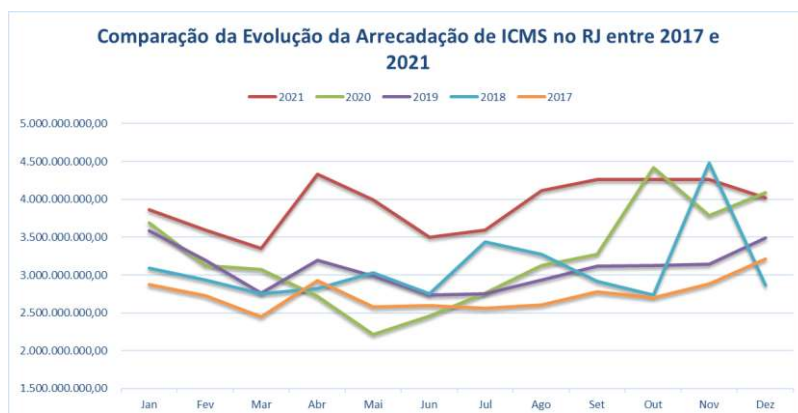
Decreto 40.613 de 15 de fevereiro de 2007, com a Contadoria Geral do Estado compondo um dos Órgãos de Assistência Direta do Secretário. Recentemente, em 2018 a denominação foi alterada para Secretaria de Estado de Fazenda e Planejamento – SEFAZ.

Sendo o ICMS o principal tributo estadual, sua importância é elevada e os esforços para manter a boa saúde da sua arrecadação é imprescindível. Em 2021, sua participação no total da arrecadação do estado foi de 69,67%, segundo o relatório contábil e de gestão fiscal relativo à prestação de contas, feito pelo departamento de contabilidade geral do Estado do Rio de Janeiro (CONTABILIDADE GERAL RJ, 2022).

A

Figura 2 mostra a evolução da arrecadação do ICMS, no Estado do Rio de Janeiro, entre os anos de 2017 e 2021. É importante salientar que o aumento de arrecadação entre 2020 e 2021, de cerca de 20%, se deve aos efeitos do cenário socioeconômico em consequência da pandemia da COVID-19 que afetou seriamente o exercício de 2020 além do aumento da inflação em 2021 que contribui na base de incidência do imposto. Não obstante aos problemas de ordem natural da economia, houve um esforço arrecadatário promovido pelo governo através da equipe econômica e fiscal.

Figura 2- Comparação da evolução de arrecadação de ICMS no RJ entre 2017 e 2021



Fonte: elaborado pelo autor com base nos dados em

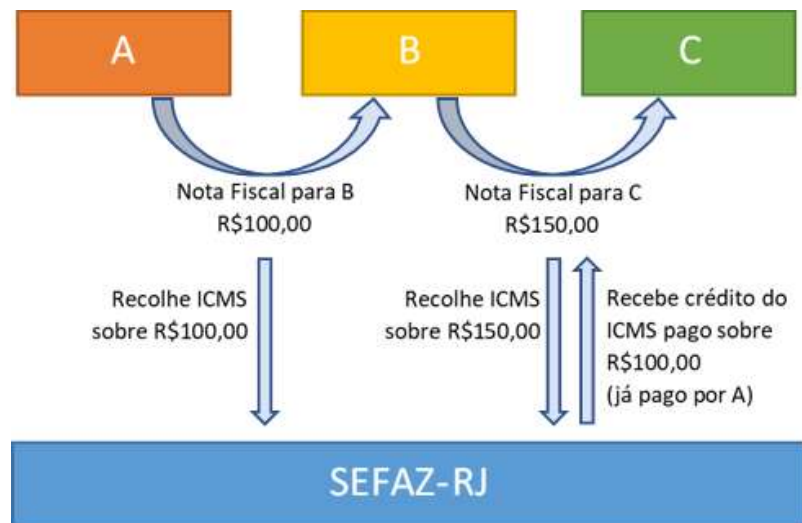
<http://www.fazenda.rj.gov.br/sefaz/faces/oracle/webcenter/portalapp/pages/navigation-renderer.jsp?datasource=UCMServer%23dDocName%3A100780>

2.3 DEFINIÇÃO DO PROBLEMA

Nos últimos anos, empresas, conhecidas como “noteiras”, vêm fraudando o ICMS por meio de uma prática que se aproveita do direito ao crédito do valor do imposto devido pela empresa fornecedora, responsável pelo recolhimento inicial desse tributo. Num exemplo de funcionamento normal de arrecadação, onde uma empresa A vende uma mercadoria a uma empresa B que, por sua vez, vende a mesma mercadoria, beneficiada ou não, a uma empresa C, a empresa A recolhe ICMS por conta da venda à empresa B e está também recolhe o ICMS, agora sobre o valor final, ou seja, parte do imposto estaria sendo cobrado novamente. Acontece que de acordo com a história do ICMS (YAMAO, 2014), seguindo o princípio da não-cumulatividade, créditos são gerados à empresa B de modo que, após a apuração de suas entradas e saídas de mercadorias, esta recolha ao estado somente o imposto sobre a diferença entre o que entrou e saiu.

A Figura 3 ilustra o processo normal de recolhimento de ICMS.

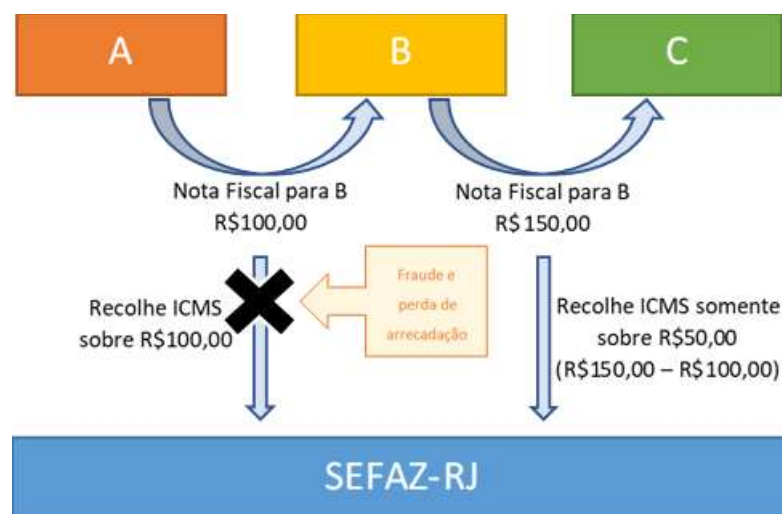
Figura 3 - Processo normal de recolhimento de ICMS



Fonte: elaborado pelo autor

A fraude tem início quando as empresas fantasmas (A) são criadas com o único intuito de emitir notas fiscais “frias”, ou seja, sem que aconteçam vendas reais, contra outras empresas (B) para que esta se beneficie dos créditos anteriormente citados. O problema acontece, efetivamente, quando a empresa emissora não paga os impostos devidos e encerra ou abandona suas atividades, nunca recolhendo o ICMS devido, como destacado na Figura 4.

Figura 4 - Processo de fraude onde não recolhe o ICMS devido



Fonte: elaborado pelo autor

Identificar tais contribuintes é um processo trabalhoso que depende da função humana para realizar diversos cruzamentos de dados em busca de padrões de comportamento que se adequam ao perfil das empresas “noteiras” para, então, se chegar à decisão de elencar as empresas que devem passar por um processo de fiscalização com o objetivo de se confirmar sua existência e sua idoneidade.

2.4 JUSTIFICATIVA E IMPORTÂNCIA

Poucos estudos sobre redes neurais aplicadas à detecção de fraudes de empresas “noteiras” com perdas de arrecadação para o Estado que chegam à casa dos bilhões. A grande quantidade de notícias e artigos apontam o alto volume de ocorrências deste tipo de fraude em todo território nacional ocasionando um gigantesco prejuízo ao erário. Uma busca utilizando o Google, com o critério de busca somente em sites oficiais, encontramos certas de 206 notícias únicas e aproximadamente 20.500 resultados não únicos. Seguem alguns exemplos das notícias e suas estimativas de prejuízos causados:

- I. Operação Maçarico V é deflagrada na capital e outros 11 municípios (Notícias SEFAZ-RJ, 2019)*
- II. Sefaz-RJ realiza nova operação para combater empresas “noteiras” (SEFAZ-RJ, 2019)*
- III. Auditores Fiscais vão vistoriar 48 estabelecimentos na capital e em outros 23 municípios (Notícias SEFAZ-RJ, 2020)*
- IV. Fiscos de SP e RJ fazem operação conjunta para combater fraudes de R\$ 600 milhões no ICMS (SEFAZ-SP, 2021)*
- V. Secretaria da Fazenda e Planejamento deflagra Operação Plassein (SEFAZ-SP, 2020)*
- VI. Empresas “noteiras” são identificadas na região de Ponta Grossa (SARTORI, 2022)*
- VII. Operação em Minas de combate a empresas “noteiras” (MINAS, 2018)*
- VIII. Operação nacional combate empresas que emitem notas frias (SEFAZ-MA, 2018)*

- IX. *Empresas “noteiras” são alvo de operação da Fazenda em Santa Catarina (sef.sc.gov.br, 2019)*
- X. *Quebra-gelo: Fazenda faz operação para recuperar R\$ 210 milhões aos cofres do Estado (SEFAZ-SC, 2020)*
- XI. *SEF promove encontro para debater medidas de combate às empresas “noteiras” (CRCSC, 2019)*
- XII. *Operação recupera R\$492 milhões em créditos tributários sonegados (SEEC-DF, 2022)*
- XIII. *Fraude de R\$ 3,2 bi no setor de sucatas é exposta pelo ES em Encontro Nacional de Inteligência (SINDIFISCAL-ES, 2019)*
- XIV. *MP baiano participa de operação que combate sonegação fiscal em oito estados e no Distrito Federal (MP-BA, 2020)*
- XV. *Experiências bem-sucedidas do país no combate às empresas “noteiras” são compartilhadas no Encontro Nacional em João Pessoa (SEFAZ-PB, 2019)*

É obrigação do Estado fiscalizar a arrecadação dos impostos a fim de promover o fluxo de entrada de caixa para o funcionamento da máquina pública. Além disso, o trabalho de fiscalização é importante para evitar a concorrência desleal, visto que a empresa, que não paga tributos, tende colocar seus preços mais baixos e, assim, prejudicando as empresas honestas.

O trabalho de apuração do ICMS de uma empresa requer dados sobre a movimentação de suas mercadorias. O registro dessas informações é obtido através de leis que obrigam as empresas a declararem essa circulação, sujeitando as empresas a multas pesadas caso descumpram a obrigação. A nota fiscal eletrônica (NFe), o conhecimento de transporte eletrônico (CTe) e a nota fiscal de consumidor eletrônico (NFCe), entre outras, são os meios pelos quais a Secretaria de Fazenda obtém os dados que servirão de fonte de análise.

Contudo, devido ao grande volume de informações e, com isso, de terem sistemas diferentes e isolados em ambientes de alta disponibilidade para cada tipo de informação, reunir todos esses dados para uma análise não é tarefa fácil.

Sobre as finanças do Estado, o Rio de Janeiro teve um impacto enorme com a pandemia de 2020 a 2022. O fechamento de milhares de empresas causou

uma grande redução na arrecadação de ICMS. Além disso, o acordo do regime de recuperação fiscal, assinado entre o Estado e a União, prevê o pagamento de uma dívida de mais de 180 bilhões.

Vem sendo observado, nacionalmente, o crescimento de empresas criadas somente com o intuito de emitir notas fiscais falsas a fim de gerar créditos tributários a outras empresas. O alto volume desse tipo de fraude vem causando prejuízos enormes ao estado e isso gerou a necessidade de ações para recuperar esse tipo de perda.

Por isso, a SEFAZ-RJ criou um grupo especializado de fiscalização para identificar empresas “noterias”. Este grupo começou seu trabalho de obtenção dos dados e de análises de acordo com a experiência dos auditores fiscais envolvidos.

Porém, observou-se que a tarefa de levantamento de informações e sua análise para uma possível ação fiscal era bastante trabalhosa, demorada e semiartesanal, pois utilizavam técnicas de cruzamento de dados baseada na teoria tradicional de conjuntos e ferramentas de banco de dados comuns.

Dentro desse contexto, este trabalho vem oferecer, ao grupo especializado da SEFAZ-RJ, um modelo de análise de empresas “noteiras” que visa a acelerar o processo e definir, objetivamente, as variáveis mais relevantes, identificando novas empresas fraudadoras a partir do seu comportamento histórico fiscal.

2.5 QUESTÕES DA PESQUISA

Este estudo foi elaborado com o propósito de responder às seguintes questões fundamentais:

- Q1- Quais são as informações fiscais do contribuinte mais relevantes para análise e identificação da fraude?
- Q2- Qual é o modelo de rede neural mais adequado para detectar a fraude com acurácia aceitável?

Q3- Qual é o menor período de coleta de dados fiscais necessários para detectar a fraude o mais precoce possível?

2.6 OBJETIVOS

O objetivo deste trabalho é definir o modelo de redes neurais artificiais mais adequado para a SEFAZ-RJ detectar empresas “noteiras” com a menor quantidade de informações fiscais possível, identificando as informações mais relevantes para identificar a fraude, avaliando a eficiência e eficácia de modelos de redes neurais aplicados, visando a constituir os fundamentos básicos para a identificação precoce desse tipo de empresa de modo que seja possível fazê-lo em qualquer unidade federativa do Brasil.

2.7 METODOLOGIA DE PESQUISA

A forma de abordagem deste trabalho envolve o método qualitativo pois busca o modelo de redes neurais mais adequado em relação ao grau de acerto produzido, e quantitativo, tendo em vista que busca analisar a quantidade mínima de informações relevantes bem como o período mínimo de coleta de dados para identificar a fraude com nível de certeza satisfatória.

Do ponto de vista da natureza do trabalho, a pesquisa é aplicada, porque produz o conhecimento de quais informações são mais relevantes, qual método de classificação, utilizando redes neurais, é mais adequado e qual é o menor período de obtenção de informações.

Quanto a seus objetivos podemos classificá-la em pesquisa exploratória, onde busca-se maior familiaridade do tema com o objetivo de desenvolver uma ferramenta que auxilie na classificação e consequente estabelecimento de medidas mitigadoras para os riscos identificados.

3. REVISÃO DA LITERATURA

3.1 TRABALHOS RELACIONADOS

A busca por artigos científicos foi realizada nas seguintes bases de artigos acadêmicos como Portal Capes, Google Acadêmico, Scielo, USP e BDTD. Os conjuntos de palavras-chave utilizadas foram utilizadas na seguinte ordem:

- fraude icms empresas “noteiras” redes neurais
- fraude icms empresas “noteiras”
- fraude icms redes neurais
- fraude icms
- empresas “noteiras”
- redes neurais

A Tabela 1 mostra os resultados obtidos em cada base.

Tabela 1 - Termos Pesquisados

PALAVRAS-CHAVE	PORTAL CAPES	GOOGLE ACADÊMICO	SCIELO	USP	BDTD - Biblioteca Digital Brasileira de Teses e Dissertações	Selecionados
fraude icms empresas “noteiras” redes neurais	0	0	0	0	0	0
fraude icms empresas “noteiras”	0	5	0	0	0	0
fraude icms redes neurais	0	115	0	0	0	4
fraude icms	2	7,960	0	0	5	0
empresas “noteiras”	0	14	0	0	0	0
redes neurais	1,380	82,300	375	100	4,485	0
Total	1,382	90,394	375	100	4,490	4

A Figura 5 mostra em forma de gráfico de barras a quantidade de publicações por palavras-chave.

Figura 5 - Quantidade de Publicações por Palavras-Chave



Fonte: elaborado pelo autor

A

Tabela 2 detalha as publicações selecionadas.

Tabela 2- Revisão da Literatura - Publicações Selecionadas

#	Título	Ano	Autores	País	Periódico / site	Método / Pesquisa	Contexto / Resultados Obtidos
1	ESTRATÉGIAS PARA COMBATER A SONEGAÇÃO FISCAL: UM MODELO PARA O ICMS BASEADO EM REDES NEURAIS ARTIFICIAIS	2020	FN de Oliveira LPG dos Santos	Brasil	Revista de Gestão, Finanças e Contabilidade v. 10 n. 1 (2020): jan./abr doi:10.18028/rgfc.v10i1.7474	Quantitativo / Qualitativa	A pesquisa teve como objetivo desenvolver um Sistema de identificação de Risco de Contribuintes baseado em redes Neurais para auxiliar a Administração Tributária Estadual na identificação de Contribuintes mais propensos a assumir a condição de sonegadores do ICMS Na fase de treinamento, a rede apresentou um índice de acerto de 71% na classificação dos contribuintes passíveis de autuação (ou não). Em relação aos contribuintes que foram autuados, a performance foi de 94%
2	PREVISÃO E AVALIAÇÃO DO DESEMPENHO DOS CONTRIBUINTES DO ICMS DO ESTADO DO CEARÁ UTILIZANDO AS REDES NEURAIS ARTIFICIAIS	2020	Sérgio Ricardo Alves Sinsando Marcos Airton de Sousa Freitas	Brasil	Revista Econômica do Nordeste Volume 37, número 1 ISSN: 2357-9226	Estudo de caso	Elaborar uma proposta alternativa de avaliação do desempenho dos contribuintes do ICMS do Estado do Ceará, utilizando as redes neurais artificiais (RNA), capaz de fornecer previsões mais confiáveis que aquelas apresentadas pelo modelo estatístico atualmente utilizado pela Secretaria da Fazenda do Estado do Ceará Deixou-se de contemplar um grande número de redes com outros tipos de topologias. Procurou-se, acima de tudo, garantir que o processo de aprendizagem não se constituísse em uma forma de “decorar padrões de previsão”. Neste trabalho, nenhum fenômeno de overfitting ou underfitting foi observado.
3	Redes neurais artificiais aplicadas à identificação de riscos de inadimplência fiscal de ICMS e ISS no Distrito Federal	2019	Vinícius Di Oliveira Prof. Dr. Ricardo Matos Chaim	Brasil	Dissertação (Mestrado Profissional em Computação Aplicada)—Universidade de Brasília, Brasília, 2019	Quantitativo / Explicativa	Verificar como o uso de redes neurais artificiais pode auxiliar na identificação de riscos de inadimplência fiscal de ICMS e ISS No grupo de 7.573 empresas selecionadas para verificação 2.471 apresentaram divergências de declarações e pagamentos, 33% do total, ou seja, um forte indicio de não recolhimento de imposto declarado. Assim, diante dos resultados observados foi considerada convalidada a relevância da classificação feita pela RNA validando o modelo
4	UMA REDE NEURAL ARTIFICIAL DE MÚLTIPLAS CAMADAS APLICADA	2004	Geraldo Galdino de Paula Junior Marcos Renato Moreira Silveira Raul Fonseca Neto	Brasil	XXXVI—SBPO, Anais, São João Del-Rei, Minas Gerais (2004)	Quantitativo / Qualitativa	Construir, treinar e testar um sistema neural para utilizá-lo na qualificação e quantificação da sonegação fiscal de ICMS, de empresas inscritas no cadastro da Secretaria de Estado da Fazenda de Minas Gerais, no sistema de recolhimento Débito/Crédito. Os resultados obtidos demonstram que foi alcançada uma metodologia bastante poderosa para aprimorar a eficiência da ação fiscal. É evidente que se chegou a um estágio que demonstra também o quanto é possível ampliar o que se fez para se chegar a uma sofisticação muito maior.

Fonte: elaborado pelo autor

Sobre as publicações selecionadas foram obtidas as seguintes constatações:

- Ratificação de que redes neurais produzem resultados de satisfatórios a ótimos utilizando as variáveis fiscais do contribuinte.
- Predominância de foco na sonegação e não a fraude
- Predominância na criação de modelos de classificação
- Nenhuma publicação com o objetivo de identificar empresas “noteiras”

4. APRENDIZADO DE MÁQUINA

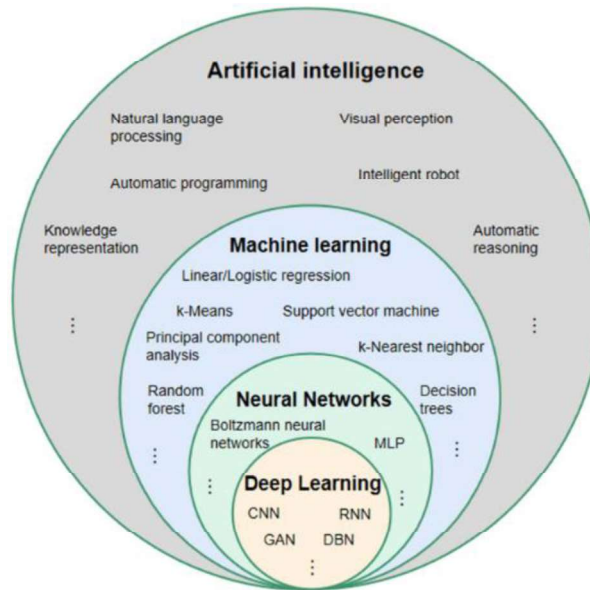
O aprendizado de máquina é um dos vários campos da inteligência artificial, baseado em processos computacionais, que se aprimoram em fazer inferências ao analisar o comportamento de dados e seus padrões ou, como disse (SEYMOUR, 1993) "O campo do aprendizado de máquina está relacionado à questão de como construir programas de computador que melhoram automaticamente com a experiência". Quando o aprendizado ocorre a partir das informações extraídas da sua estatística, dizemos que este é um aprendizado estatístico. A enorme quantidade de dados disponíveis atualmente desafia o trabalho dos estatísticos tanto em quantidade quanto em complexidade. No livro "*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*" (HASTIE, 2009), escrito por três estatísticos da Universidade de Stanford, eles dizem que "Vastas quantidades de dados estão sendo geradas em muitos campos, e o trabalho dos estatísticos é entender tudo: extrair padrões e tendências importantes e entender 'o que os dados dizem'. Chamamos isso de aprendizado a partir de dados."

Enquanto o aprendizado estatístico tem o objetivo de prever as chances de algo acontecer ou não, o aprendizado de máquina tem o propósito de melhorar a precisão da previsão a fim de que se possa tomar uma decisão, com grau de confiança satisfatório, automaticamente e a partir de dados não vistos na fase de aprendizagem.

A

Figura 6 fornece uma visão geral da inteligência artificial, suas especializações e a relação entre elas.

Figura 6 - Visão geral da inteligência artificial



Fonte: (A Comprehensive Review on Radiomics and Deep Learning for Nasopharyngeal Carcinoma Imaging, 2022)

4.1 LÓGICA NEBULOSA (FUZZY LOGIC)

A lógica nebulosa foi fundamentada pela teoria dos conjuntos nebulosos, por (ZADEH, 1965), nos Estados Unidos. É uma proposta que oferece uma maneira, próxima à percepção humana, para traduzir informações imprecisas e vagas em entidades capazes de serem processadas por um computador. Por exemplo, “Arrecada Pouco” ou “Arrecada o Esperado”, que definem os valores percebidos na arrecadação de ICMS de uma empresa.

Na teoria clássica dos conjuntos, para um subconjunto **A** de um universo discurso **U**, um elemento **x** pertencente a **U** se é possível definir uma função de modo que:

$$\begin{cases} F_A(x) = 1 & \text{se } x \in A \\ F_A(x) = 0 & \text{se } x \notin A \end{cases} \quad (1)$$

Na teoria de conjuntos nebulosos, um conjunto nebuloso A num universo discurso U , é definido por uma função de pertinência μ_A , que referencia os elementos de U dentro do intervalo real $[0, 1]$, por um grau de pertinência representado por:

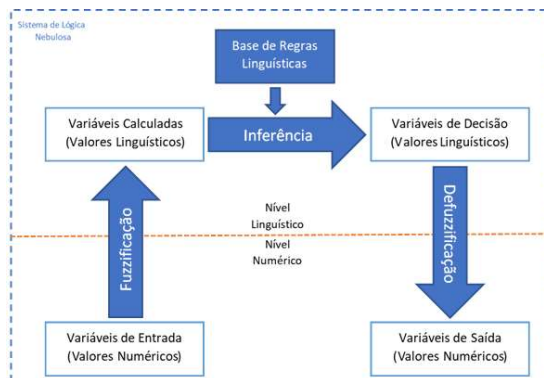
$$\mu_A: U \rightarrow [0, 1] \quad (2)$$

Em outras palavras, isto significa que cada elemento de um conjunto nebuloso é considerado na decisão do resultado de acordo com o seu grau de pertinência àquele conjunto, ou seja, um elemento pode pertencer, simultaneamente a mais de um conjunto.

As funções de pertinência (ZADEH *et al.*, 2018) são escolhidas, apropriadamente, de acordo com o intervalo de uma variável linguística a ser representada. Para isso, é fundamental que um especialista no assunto defina os intervalos numéricos para os seus valores linguísticos.

Um sistema de lógica nebulosa é um processo composto por um conjunto de etapas onde, primeiro, ocorrem transformações de variáveis numéricas em variáveis linguísticas e, depois, utilizando regras construídas através de expressões linguísticas do tipo Se U_1 é A_1 e U_2 é A_2 , então R é B (ZADEH *et al.*, 2018) e, enfim, a transformação das variáveis linguísticas de volta em variáveis numéricas. A Figura 7 mostra o esquema de um sistema nebuloso.

Figura 7- Sistema de lógica nebulosa



Fonte: adaptado de (GONZÁLEZ, 2009)

A entrada são os dados a serem analisados, representados por valores numéricos. O processo de *fuzzificação* é onde ocorre a conversão das variáveis numéricas em variáveis linguísticas com seu respectivo grau de pertinência para cada conjunto definidos pelas funções de pertinência. O sistema de inferência realiza a avaliação do conjunto de regras utilizando a função de implicação representada pela equação (3).

$$\mu_{A \rightarrow B}(u, v) = (\mu_A(u) \wedge \mu_B(v)) \vee (1 - \mu_A(u)) \quad (3)$$

proposta por (ZADEH, 1965) e gera um conjunto nebuloso de saída contendo todos os resultados das regras.

Por fim, o processo de *defuzzificação* utilizará o conjunto nebuloso resultante para obter um valor numérico para cada variável de saída, utilizando o método de *defuzzificação* conhecido como **Método do Centro de Gravidade**, dentre outros existentes, como os citados por (MIZUMOTO, 1998), é definido pela equação (4).

$$w^0 = \frac{\sum_i \mu C(w_i) \cdot w_i}{\sum_i \mu C(w_i)} \quad (4)$$

Para a implementação de um sistema nebuloso poderiam ser sugeridos os seguintes passos:

1. Escolha das variáveis de entrada e saída
2. Definição dos intervalos do universo discurso das variáveis de entrada e de saída
3. Elaboração das regras de inferência
4. Escolha das funções de pertinência
5. Discretização do universo de entrada e saída
6. Escolha do tipo de implicação
7. Implementação do motor de inferência nebuloso

4.2 REDES NEURAIS ARTIFICIAIS

A capacidade de tomar decisões em situações por onde nunca passou é uma forte característica do cérebro humano. Ele toma decisões baseadas em experiências parecidas com um índice incrível de acerto. As redes neurais artificiais, ou redes neurais, foram criadas com o objetivo de imitar o cérebro humano e, conseqüentemente, tentar obter o mesmo sucesso na tomada de decisões, generalizando o problema, ou seja, sem conhecer previamente todas as possibilidades como, normalmente, faz um programa de computador.

4.2.1 HISTÓRICO

O primeiro modelo matemático de um neurônio foi proposto por (MCCULLOCH; PITTS, 1943)

O primeiro modelo de redes neurais artificiais foi criado em 1949 por Donald Hebb, em seu livro de título "*The Organization of Behavior - A Neuropsychological Theory*" (HEBB, 1949), Hebb diz em seu "postulado de aprendizagem" onde afirma que "a eficiência de uma sinapse variável entre dois neurônios é aumentada pela ativação repetida de um neurônio causada pelo outro neurônio através daquela sinapse" e "quando uma célula ajuda repetidamente a disparar outra, o axônio da primeira célula desenvolve botões sinápticos (ou os aumenta, se já existirem) em contato com o soma da segunda célula, onde "soma" refere-se a dendritos e corpo, ou toda a célula exceto seu axônio". No modelo de Hebb a relação entre dois neurônios, ou nós em redes neurais artificiais, se fortalece se os dois neurônios são ativados ao mesmo tempo e enfraquece se forem ativados separadamente. O "peso" é usado para descrever a intensidade dessas relações.

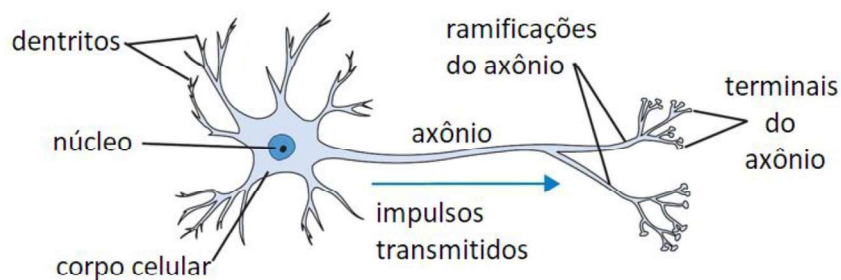
Em 1950, Arthur Samuel, da IBM, desenvolveu um programa de computador para jogar damas. Como o programa tinha uma quantidade muito pequena de memória disponível, Samuel iniciou o que é chamado de poda alfa-beta (alpha-beta *prunning*). Seu design incluía uma função de pontuação usando as posições das peças no tabuleiro. A função de pontuação tentava medir as chances de cada lado vencer. O programa escolhe sua próxima jogada usando uma estratégia

minmax, que acabou evoluindo para o algoritmo minmax. Ele também projetou vários mecanismos, permitindo que seu programa se tornasse melhor. No que Samuel chamou de aprendizado mecânico, seu programa registrava todas as posições que já havia visto e combinou isso com os valores de uma função de recompensa. Arthur Samuel surgiu com a frase "Aprendizado de Máquina" ("*Machine Learning*") em 1952.

Neurônios são unidades de processamento de informação, formados de corpo celular, axônios e suas ramificações, como mostrados na Figura 8, e são simulados, artificialmente, através da entrada X_n , seus pesos W_n , sua função de ativação f e sua saída, segundo , e representado na

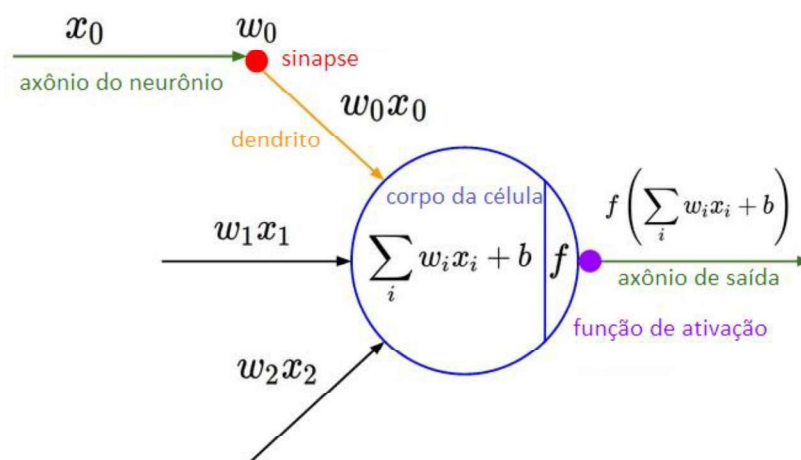
Figura 9.

Figura 8- Representação de um neurônio natural



Fonte: adaptado de <http://cs231n.github.io/neural-networks-1/>

Figura 9 - Representação de um neurônio artificial



Fonte: adaptado de <http://cs231n.github.io/neural-networks-1/>

Neurônios artificiais são dispostos em camadas a fim de formarem as redes neurais. O aprendizado artificial é resultado do treino, ou seja, dos ajustes dos pesos após verificação do erro decorrente da confrontação do resultado obtido pelo modelo com o resultado esperado. Após o processo iterativo do treino, são obtidos os valores ótimos para os pesos, isto é, valores que reduzem o erro ao seu menor valor possível. Assim, pode-se dizer que a rede neural artificial está treinada e pronta a ser submetida a novos dados desconhecidos. A principal vantagem é que, uma vez treinada a RNA, a massa de dados não é mais necessária para estimar um novo resultado, diminuindo muito a exigência de poder computacional numa aplicação prática.

Em 1957, Frank Rosenblatt, no Laboratório Aeronáutico de Cornell, combinou o modelo de interação de células cerebrais de Donald (HEBB, DONALD OLDING, 2005) com os esforços de aprendizado de máquina de Arthur (SAMUEL, 1959) e criou o *perceptron* (ROSENBLATT, 1960). O *perceptron* foi inicialmente planejado como uma máquina, não como um programa. O software, originalmente projetado para o IBM 704, foi instalado em uma máquina personalizada chamada Mark 1 *perceptron*, que havia sido construída para reconhecimento de imagem. Descrito como o primeiro neuro computador de sucesso, o Mark I *perceptron* desenvolveu alguns problemas com expectativas não alcançadas. O *perceptron* não conseguia reconhecer muitos tipos de padrões visuais, como rostos, causando frustração e

paralisando as pesquisas em redes neurais. A rede neural e a pesquisa em aprendizado de máquina lutaram até o ressurgimento nos anos 90.

Em 1967, o algoritmo do vizinho mais próximo foi concebido dando início ao reconhecimento básico de padrões. Esse algoritmo foi usado para mapear rotas e é um dos algoritmos mais antigos usados para encontrar uma solução para o problema do vendedor viajante de encontrar a rota mais eficiente. Donald E. Knuth (KNUTH, 1984) recebeu crédito por ter inventado a “regra do vizinho mais próximo” mas ele credita os méritos ao famoso artigo "*Nearest Neighbor Pattern Classification*" (COVER; HART, 1967).

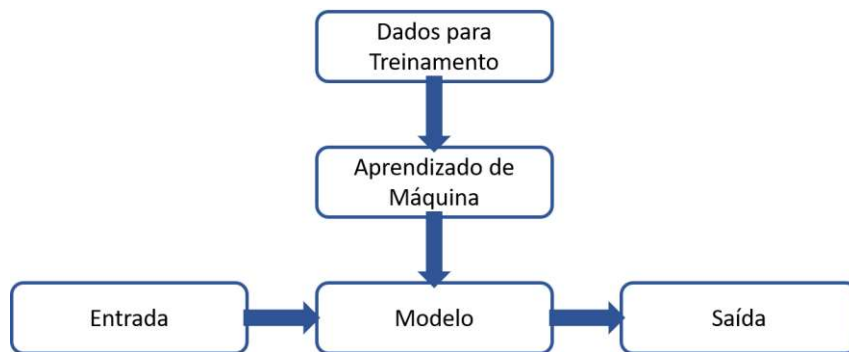
Nos anos 60 foi descoberta a possibilidade de se usar múltiplas camadas, e isso levou ao desenvolvimento de redes neurais *feedforward* o algoritmo *backpropagation*, desenvolvido já nos anos 70. Ele descreve "a propagação reversa de erros", com um erro sendo processado na saída e depois distribuído para trás pelas camadas da rede para fins de aprendizado. A retro propagação agora está sendo usada para treinar redes neurais profundas.

4.2.2 APLICAÇÃO

As redes neurais são excelentes para reconhecer padrões complexos onde se exige a análise de muitas informações ao mesmo tempo. Através do processamento de dados recebidos pela camada de entrada e produzindo o resultado pela camada de saída, as camadas escondidas são responsáveis pela transformação dos dados. Christopher M Bishop (BISHOP, 2006), em seu livro "Pattern Recognition and Machine Learning" diz que "O reconhecimento de padrões tem suas origens na engenharia, enquanto o aprendizado de máquina surgiu da ciência da computação. No entanto, essas atividades podem ser vistas como duas facetas do mesmo campo".

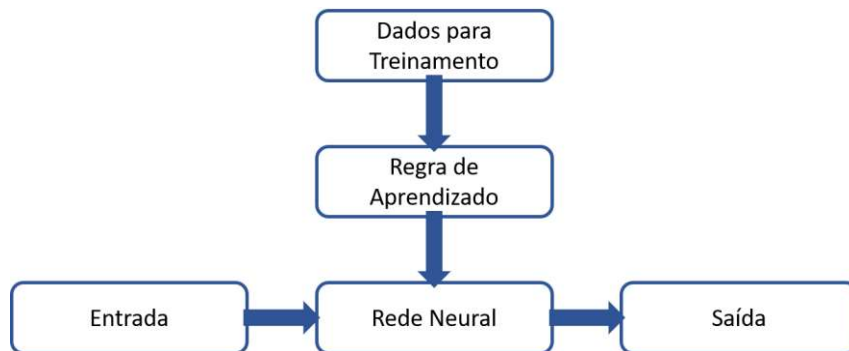
A Figura 10 e a Figura 11 mostram a diferença, mas, ao mesmo tempo, a complementaridade entre o aprendizado de máquina e as redes neurais. Enquanto na primeira não há a intervenção humana, na segunda existe a rotulação dos dados feita previamente, ensinando ao modelo o que é que se deseja.

Figura 10 - Processo do aprendizado de máquina



Fonte: elaborado pelo autor

Figura 11 - Processo da rede neural

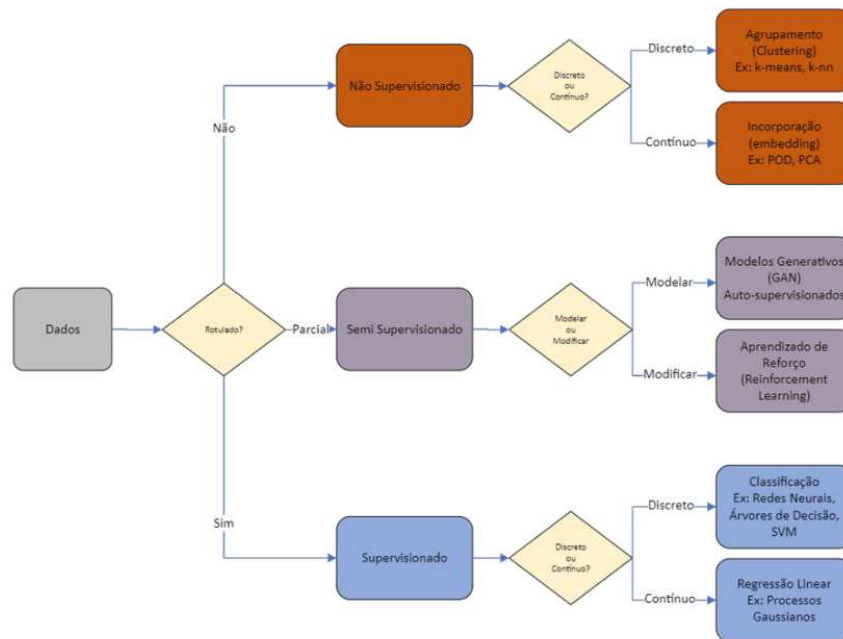


Fonte: elaborado pelo autor

4.2.3 TIPOS DE REDE NEURAL

O tipo de dado que se tem disponível nos guiará a selecionar o tipo de rede neural mais apropriada. A Figura 12 ilustra os modelos disponíveis, dependendo do tipo de dado que se tem.

Figura 12- Tipos de Máquinas de Aprendizado



Fonte: adaptado de https://www.youtube.com/watch?v=0_IKUPYEEyY&t=201s

Os dados disponíveis para a solução proposta pelo autor *** são rotulados, portanto, o tipo de aprendizado a se utilizar é o supervisionado. Os valores são discretos, considerando o valor “1” para empresas classificadas como “noteiras” e “0” para empresas “não “noteiras””. Assim, o tipo de máquina de aprendizado utilizado nesse estudo é uma rede neural artificial de classificação.

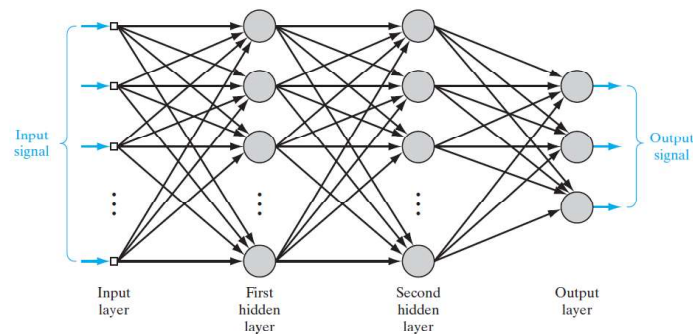
4.2.4 ARQUITETURAS MULTI-LAYER PERCEPTRON

As arquiteturas das redes neurais artificiais do tipo *perceptron* de múltiplas camadas, ou *multi-layer perceptron*, são constituídas por 3 diferentes tipos de camadas: uma de entrada, uma ou mais escondidas, também conhecida como e uma de saída. A

Figura 13 mostra um exemplo de rede neural de múltiplas camadas com 2 camadas escondidas. A camada de entrada é obrigatória e é ela que recebe o estímulo externo, ou seja, as variáveis de entrada. A camada de saída também é

obrigatória e contém o resultado final, ou seja, o valor ou classificação resultado para os valores de entrada. A parte escondida, também conhecidas como intermediárias, ocultas ou invisíveis, pode conter muitas camadas e, quando possui mais de uma camada, a rede passa a ser chamada de rede profunda ou de conhecimento profundo.

Figura 13 - Grafo da arquitetura de uma rede neural com duas camadas escondidas



Fonte : (HAYKIN, 2009)

4.2.5 FUNÇÕES DE ATIVAÇÃO

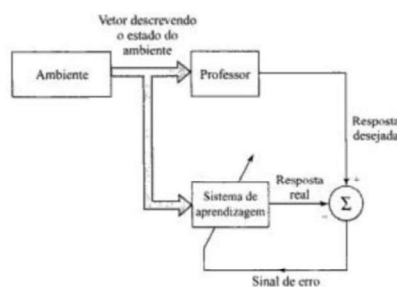
As funções de ativação são responsáveis pela ativação ou não de um neurônio. Elas recebem um conjunto de dados de entrada e geram uma saída.

4.2.6 TREINAMENTO E APRENDIZADO

“O aprendizado supervisionado é uma técnica de aprendizado de máquina em que o algoritmo é primeiro apresentados com dados de treinamento que consistem em exemplos que incluem tanto as entradas quanto as saídas desejadas”, (ZHANG, 2010). Trata-se de um processo indutivo onde a supervisão significa fornecer as respostas baseadas em pré-classificações de dados históricos. São os chamados rótulos, ou seja, a classificação correta que servirá como exemplo de onde se deseja chegar com base nos dados de entrada.

O processo de treinamento para o aprendizado supervisionado consiste em submeter amostras dos dados à rede neural e minimizar o erro, ou seja, a distância do objetivo ou do dado rotulado. “A ideia é medir o erro que a rede comete ao classificar e modificar o peso para que esse erro se torne muito pequeno” (LIN; LEE, 1996). A Figura 14 ilustra o diagrama de blocos do processo de aprendizagem supervisionada.

Figura 14 - Diagrama em blocos da aprendizagem com um professor (supervisionada)



Fonte: (HAYKIN, 2009)

4.2.6.1 BACKPROPAGATION

O algoritmo *backpropagation*, ou retropropagação, foi criado na década de 1970, mas somente em 1986 a sua importância surgiu através do artigo escrito por de (RUMELHART, 1986) com o título de “*Learning representations by back-propagating errors*”. Esse artigo descreve redes neurais onde a retropropagação funciona muito mais rápido do que outras abordagens de aprendizado, o que, somente a partir daí, foi possível usar redes neurais para resolver problemas que antes eram insolúveis. Atualmente, o algoritmo de retropropagação é o mais utilizado no aprendizado em redes neurais.

O método de treinamento *backpropagation*, criado por (RUMELHART, 1986), usado para treinar uma rede neural de *perceptrons* de múltiplas camadas, é descrito por (HAYKIN, 2009) como tendo duas fases distintas: fluxo de sinal, iniciado pelo vetor $v_k(n)$ pela com propagação para frente, ou *forward*, onde o sinal de saída de cada camada é definido pela equação (5):

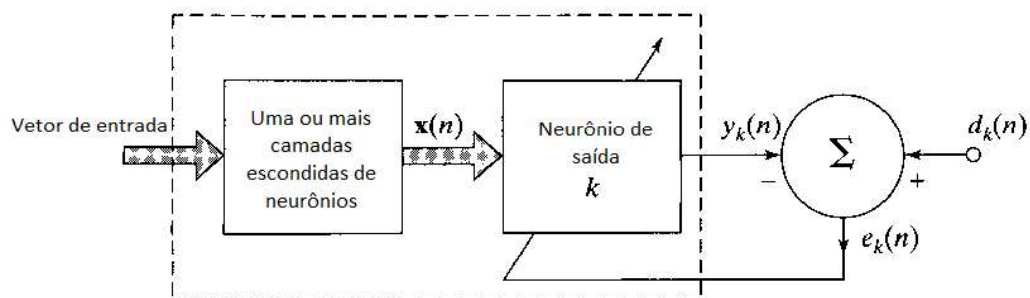
$$y_k = \sum_{j=1}^m v_{kj} x_j \quad (5)$$

Onde o sinal de entrada da próxima camada é o sinal de saída $y_k(n)$ da camada anterior e segue, assim por diante, até a última camada onde está o resultado final da classificação que será comparado com o valor alvo $d_k(n)$ para, então, calcular a diferença entre eles, resultando no erro $e_k(n)$, definido pela função de custo da equação (6).

$$e_k(n) = d_k(n) - y_k(n) \quad (6)$$

A Figura 15 mostra o processo de aprendizagem numa rede neural com um único neurônio de saída.

Figura 15 – Diagrama de blocos de uma rede neural destacando um único neurônio na camada de saída



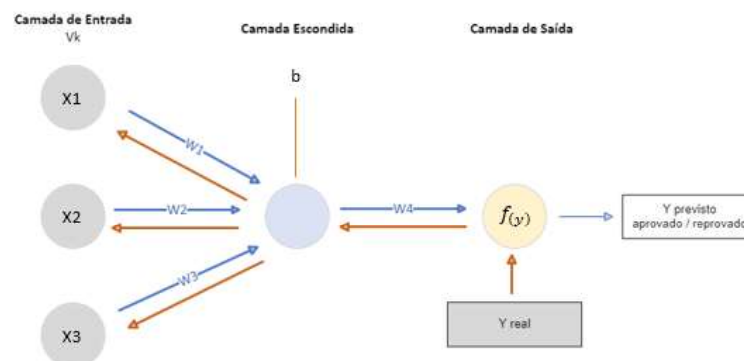
Fonte: adaptado de (HAYKIN, 2009)

O próximo passo é recalcular os pesos e, para isso, é utilizado um método otimizador como o do gradiente descendente, de modo que o próximo resultado se aproxime cada vez mais do alvo. Os novos valores dos pesos têm de ser recalculados

de uma maneira que o erro diminua e isso é feito através do algoritmo de retropropagação.

Podemos dividir todo o processo em 8 passos, utilizando a rede neural da Figura 16 como exemplo, para o entendimento do algoritmo.

Figura 16 Rede neural exemplo



Fonte: adaptado de <https://medium.com/analytics-vidhya/understanding-artificial-neural-network-in-7-steps-86edf61be53e>

Passo 1: inicializa-se os pesos com valores baixos randômicos

Passo 2: submete-se os dados na camada de entrada e obtém-se a saída Y a partir da equação (7).

$$Y = \sum_{j=1}^m v_{kj} x_j + b \quad (7)$$

Assim, aplicando-se os valores, Y pode ser representado pela equação (8):

$$Y = w1 * v[x1] + w2 * v[x2] + w3 * v[x3] + b \quad (8)$$

Onde:

$w1$, $w2$ e $w3$ são os pesos (weights) e b é o viés (bias)

E aplica-se a função de ativação e obtém-se a saída S representada pela equação (9):

$$S = f_{(y)} \quad (9)$$

Passo 3: a saída da camada escondida é enviada como entrada da camada de saída. Esse passo se chama propagação para frente e está destacado em azul. Só existe uma saída porque se trata de uma classificação binária, ou seja, no caso deste estudo, a empresa é ou não “noteira”.

Passo 4: A classificação estimada é, então, comparada com a classificação rotulada, isto é, verifica-se se houve acerto com base na classificação fornecida como dado de treinamento.

Passo 5: Obtém-se a diferença entre o valor previsto e o pretendido para propagá-la pela rede neural no próximo passo.

Passo 6: é necessário recalcular os pesos e o viés para que a diferença encontrada no passo anterior seja minimizada de modo que, na próxima passada, o valor previsto se aproxime do valor pretendido. Isso é feito utilizando-se otimizadores como o gradiente descendente.

Passo 7: calcula-se quais seriam os novos pesos com o objetivo de minimizar a perda, ou o custo, com a equação (10):

$$*W_x = W_x - a \left(\frac{\partial \text{Error}}{\partial W_x} \right) \quad (10)$$

Onde:

* W_x = novo valor do peso

W_x = valor anterior do peso

a = taxa de aprendizado, que é um parâmetro configurável da rede

$\left(\frac{\partial \text{Error}}{\partial W_x} \right)$ = derivada do erro em relação ao peso

Vale salientar que a taxa de aprendizado varia de 0 a 1 e não deve ser muito pequena porque isso faz com que a convergência para os pesos ideais demore muito a acontecer ao mesmo tempo que não deve ser muito grande porque pode provocar a perda do ponto mínimo local da função na busca dos pesos ideais.

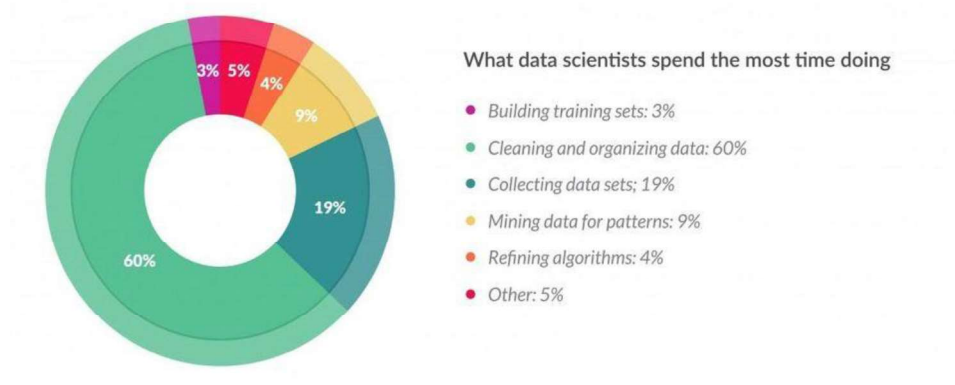
5. DESENVOLVIMENTO DA SOLUÇÃO

5.1 SOBRE DOS DADOS

Os dados são a parte mais importante de um projeto que envolve aprendizado de máquina. Assim, como o cérebro humano, que aprende pela experiência, uma rede neural precisa dessa experiência para aprender. Neste caso, a experiência vem da quantidade e da qualidade dos dados. Quantidade porque, quanto mais experiência, mais se aprende e qualidade porque, quanto mais corretamente classificados são os dados, maior a qualidade do aprendizado e, por fim, melhor será a qualidade da classificação de dados desconhecidos.

É fundamental conhecer bem os dados a fim de buscar o melhor caminho para solução. Também, será necessário garantir que eles estejam numa escala apropriada, num formato útil. "A preparação dos dados é responsável por 80% de todo o trabalho" - (FORBES, 2016), demonstrada na Figura 17.

Figura 17- Revista Forbes - Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says



Fonte: <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#139bb0b16f63>

5.2 A OBTENÇÃO DOS DADOS

Os dados foram obtidos através de vários *pipeline* de dados, com base nas informações fiscais obtidas das declarações econômico-fiscais como EFD ICMS/IPI (Escrituração Fiscal Digital), GIA-ICMS (Guia de Informação e Apuração do ICMS), GIA-ST (Guia Nacional de Informação e Apuração do ICMS Substituição Tributária), DeSTDA (Declaração de Substituição Tributária, Diferencial de Alíquota e Antecipação), DUB-ICMS (Documento de Utilização de Benefícios Fiscais do ICMS) e DECLAN-IPM (Declaração Municipal para o Índice de Participação dos Municípios) utilizando a ferramenta *Apache NiFi*. A Tabela 3 descreve os dados disponíveis que servirão como ponto de partida para a modelagem e treinamento da rede neural.

Tabela 3- Dados disponíveis

VARIÁVEL	DESCRIÇÃO
ID	Identificação da empresa (artificial para fins de sigilo)
ANO	Ano apurado
MÊS	Mês apurado
IDADE	Idade da empresa em meses
VL_TOTAL_NF_DEST_E	Valor total de NF de entrada recebida pelo contribuinte no mês
VL_TOTAL_NF_DEST_S	Valor total de NF de saída recebida pelo contribuinte no mês
VL_ARRECADACAO	Valor arrecadado do contribuinte no mês
QTD_NFE_S	Quantidade de NFE emitidas de saída pelo contribuinte no mês
QTD_NFE_E	Quantidade de notas emitidas de entrada pelo contribuinte no mês
QTD_NOTAS_DEST_S	Quantidade de NFE de saída recebida pelo contribuinte no mês
QTD_NOTAS_DEST_E	Quantidade de notas de entrada recebida pelo contribuinte no mês
MAIOR_VL_NFE_E	Maio valor da NFE emitida de entrada pelo contribuinte no mês

MAIOR_VL_NFE_S	Maio valor da NFE emitida de saída pelo contribuinte no mês
MAIOR_VL_NF_DEST_E	Maio valor da NFE de entrada recebida pelo contribuinte no mês
MAIOR_VL_NF_DEST_S	Maio valor da NFE de saída recebida pelo contribuinte no mês
VL_TOTAL_NFE_E	Valor total de NF de entrada emitida pelo contribuinte no mês
VL_TOTAL_NFE_S	Valor total de NF de saída emitida pelo contribuinte no mês
VL_TOTAL_ICMS_E	Valor total de ICMS na NF de entrada pelo contribuinte no mês
VL_TOTAL_ICMS_S	Valor total de ICMS na NF de saída pelo contribuinte no mês
VL_CONTABIL_ENTRADA	Valor contábil de notas de entrada: $\text{SUM}(\text{NVL}(\text{VL_TOTAL_NFE_E},0) + \text{NVL}(\text{VL_TOTAL_NF_DEST_S},0)) - \text{SUM}(\text{NVL}(\text{VL_TOTAL_NFL_DEST_E},0))$
VL_ICMS_ENTRADA	Valor contábil de ICMS de entrada: $\text{SUM}(\text{NVL}(\text{VL_TOTAL_ICMS_E},0) + \text{NVL}(\text{VL_TOTAL_ICMS_DEST_S},0)) - \text{SUM}(\text{NVL}(\text{VL_TOTAL_ICMS_DEST_E},0))$
VL_ST_ENTRADA	Valor contábil de ST de entrada: $\text{SUM}(\text{NVL}(\text{VL_TOTAL_ST_DEST_S},0) + \text{NVL}(\text{VL_TOTAL_ST_E},0))$
QTD_VEICULO	Quantidade de veículo(s) vinculado(s) ao contribuinte
VL_TOTAL_VEICULO	Valor venal total do(s) veículo(s) vinculado(s) ao contribuinte

Os dados deste estudo estão agrupados pela identificação da empresa, “ID”, o ano e o mês apurados, compreendendo um total de 12.213 registros rotulados, sendo 2.021 como “NOTEIRA” e 10.192 registros como “NÃO NOTEIRA”.

O período de análise dos dados é de 03/2021 a 03/2022, porém existem empresas que não possuem todo o período de dados pois são empresas recentes,

criadas em datas entre esse intervalo da análise. Sendo assim, o período máximo de análise é de 13 meses enquanto o período mínimo é de 1 mês.

A análise é feita por empresa, mês a mês, ou seja, uma mesma empresa pode possuir rótulos diferentes para cada mês, dependendo do seu comportamento financeiro naquele mês.

5.3 O PROBLEMA NA ROTULAÇÃO DOS DADOS

Rotular os dados de empresas como “noteiras” e “não “noteiras”” é fundamental para que uma rede neural seja capaz de aprender corretamente. Contudo, como o trabalho atualmente é manual, a única informação que conseguimos obter foi a de que, num período de análise de 13 meses, determinadas empresas foram identificadas como “noteiras” e outras não. O problema se torna evidente quando empresas “noteiras” possuem um comportamento intermitente ou a partir de certo mês de sua existência, ou seja, ela não cometeu fraude em todos os meses durante o seu funcionamento. Assim, ao rotular todos os meses do seu período analisado, poderemos rotular alguns meses corretamente, mas outros não.

A fim de minimizar o ruído que essa rotulação pode causar, foi construído um sistema baseado em lógica nebulosa para identificar registros rotulados como “noteira” mas que não possuem um comportamento tão evidente de “noteira”, baseado em regras básicas e definidas por um especialista no assunto.

5.4 UM MODELO NEBULOSO PARA AJUDAR NA ROTULAÇÃO DOS DADOS

O do sistema nebuloso proposto é, com regras baseadas em algumas poucas variáveis, como idade da empresa, valores de notas fiscais de saída e de entrada e valores declarados ao fisco, descartar os rótulos dos meses que, evidentemente, estão rotulados indevidamente.

O primeiro passo é definir as variáveis de entrada. A "Idade da Empresa" representa a quantidade de meses que uma empresa tem desde a data de sua inscrição e possui três conjuntos nebulosos, como definidos na Tabela 4.

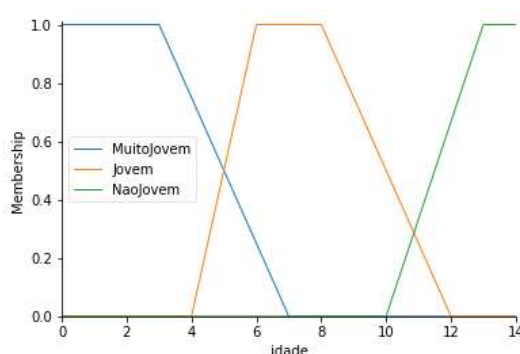
Tabela 4- Conjuntos Nebulosos de Idade

Muito Jovem	Até 7 meses
Jovem	Entre 4 e 12 meses
Não Jovem	A partir de 10 meses

O segundo passo é definir as funções de pertinência utilizadas para cada conjunto. Foram definidas, respectivamente, função decrescente, função trapezoidal e a função crescente, como podem ser vistas na

Figura 18.

Figura 18- Funções de pertinência da variável Idade



Fonte: elaborado pelo autor

A variável "Compras x Vendas" representa a relação entre valores de notas fiscais de entradas e as notas fiscais de saída no mesmo período, e tem 4 conjuntos nebulosos, como descritos na Tabela 5.

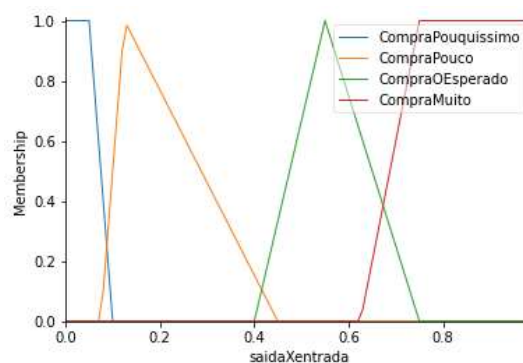
Tabela 5- Conjuntos Nebulosos de Compras x Vendas

Compra Pouquíssimo	Entre 0 e 0,10
Compra Pouco	Entre 0,075 e 0,45
Compra o Esperado	Entre 0,35 e 0,75
Compra Muito	A partir de 0,625

As funções de pertinência utilizadas para cada conjunto foram, respectivamente, função decrescente, função triangular, função triangular novamente e a função crescente, como pode ser visto na

Figura 19.

Figura 19- Funções de Pertinência de Compras



Fonte: elaborado pelo autor

A última variável de entrada utilizada, "Arrecadação x Vendas", representa a arrecadação esperada em relação às notas fiscais de saída, ou seja, às vendas realizadas no mesmo período, e tem 4 conjuntos nebulosos, como mostra a Tabela 6.

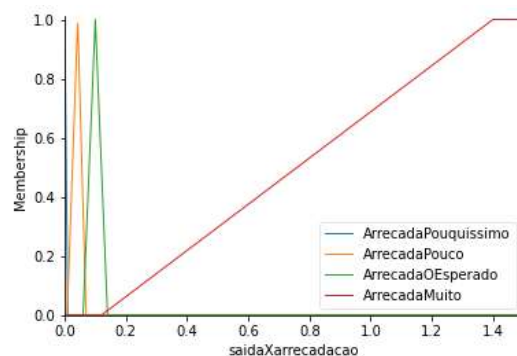
Tabela 6- Conjuntos Nebulosos de Arrecadação x Vendas

Arrecada Pouquíssimo	Entre 0 e 0,01
Arrecada Pouco	Entre 0,008 e 0,08
Arrecada o Esperado	Entre 0,06 e 0,16
Arrecada Muito	A partir de 0,14

As funções de pertinência utilizadas para cada conjunto foram, respectivamente, função decrescente, função triangular, função triangular novamente e a função crescente, como pode ser visto na

Figura 20.

Figura 20- Funções de Pertinência de Arrecadação



Fonte: elaborado pelo autor

O terceiro passo é a definição das regras de inferência. Um conjunto de 48 regras foram determinadas para o sistema e na Tabela 7 são mostradas algumas regras para servir de exemplo. O antecedente é a condição formada pelas variáveis de entrada "Idade Empresa", "Compras x Vendas" e "Arrecadação x Vendas", e o consequente é o resultado indicado por um dos valores da variável de saída "Grau Fraudadora", ou seja, "Baixo", "Médio" e "Alto".

O conjunto completo de regras pode ser observado na implementação de código no APÊNDICE A. A Tabela 7 mostra alguns exemplos das regras do sistema nebuloso.

Tabela 7- Exemplos de Regras do Sistema Nebuloso de Inferência

Se	Então
Idade é "Muito Jovem" e "Compra Pouquíssimo" e "Arrecada Pouquíssimo"	o grau é "Alto"
Idade é "Jovem" e "Compra Pouco" e "Arrecada Pouco"	o grau é "Médio"
Idade é "Não Jovem" e "Compra o Esperado" e "Arrecada o Esperado"	o grau é "Baixo"

Para que seja considerado um sistema nebuloso, os intervalos, definidos pelas funções de pertinência das variáveis de entrada, devem ter áreas de sombra, ou seja, as extremidades dos intervalos devem se sobrepor de modo que esta área de transição demonstre a pertinência de um valor em mais de um conjunto, simultaneamente.

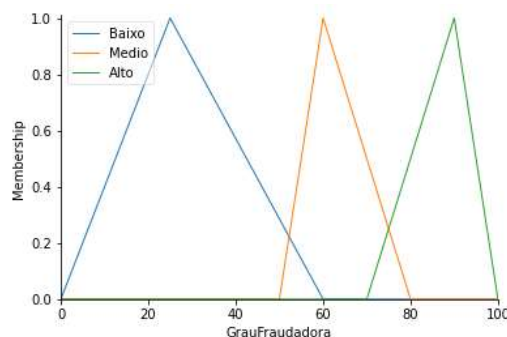
Além das variáveis de entrada, teremos uma variável de saída, chamada “Grau Fraudadora”, que traduz o grau de possibilidade de uma empresa ser considerada como fraudadora, podendo assumir os três valores linguísticos: “Baixo”, “Médio” e “Alto” de acordo com a Tabela 8.

Tabela 8- Conjuntos Nebulosos de Grau Fraudadora

Baixo	Entre 0 e 60
Médio	Entre 50 e 80
Alto	Entre 70 e 100

Três funções de pertinência foram utilizadas para definir as três faixas, "Baixo", "Médio" e "Alto", de saída da variável "Grau Fraudadora". Cada faixa foi definida por uma função triangular, como podem ser vistas na Figura 21.

Figura 21 - Funções de Pertinência de Grau Fraudadora



Fonte: elaborado pelo autor

Definidas as variáveis de entrada e saída e seus universos discursos, construiremos as regras de inferência. O sistema proposto possui 48 regras e, por brevidade e exemplificação, somente serão listadas nove delas, como mostra a Tabela 9.

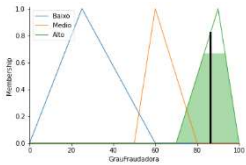
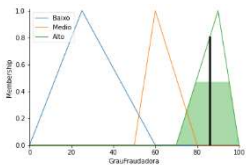
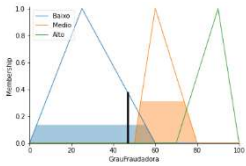
Tabela 9- Regras do Sistema Nebuloso de Inferência

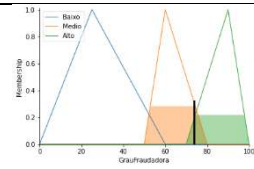
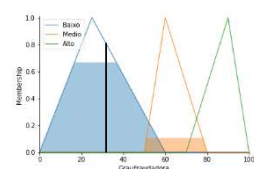
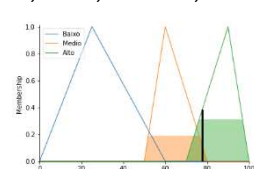
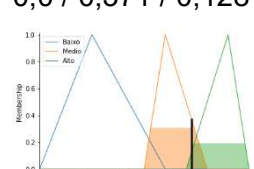
Regra	Variáveis Linguísticas de Entrada			Variável Linguística de Saída
	Idade	Entradas x Saídas	Arrecadação x Saídas	Grau Fraudadora
R1	Muito Jovem	Compra Pouquíssimo	Arrecada Pouquíssimo	Alto
R2	Muito Jovem	Compra Pouquíssimo	Arrecada O Esperado	Médio
R3	Jovem	Compra Pouco	Arrecada Pouquíssimo	Alto
R4	Jovem	Compra Pouco	Arrecada Pouco	Médio
R5	Não Jovem	Compra O Esperado	Arrecada Pouquíssimo	Alto
R6	Não Jovem	Compra Muito	Arrecada Pouco	Médio

R7	Muito Jovem	Compra O Esperado	Arrecada O Esperado	Baixo
----	-------------	-------------------	---------------------	-------

Para cada entrada de dados, todas as regras são avaliadas pelo sistema de inferência que formará um conjunto nebuloso resultado para que, então, seja submetido ao processo de *defuzzificação* a fim de se obter a variável numérica de saída. A Tabela 10 mostra alguns exemplos de valores numéricos de entrada que produzirão suas respectivas saídas numéricas e seu valor linguístico final, resultado da *defuzzificação*.

Tabela 10- Exemplos de entradas e saídas do sistema nebuloso

Entrada	Variáveis Linguísticas de Entrada			Saída	
	Idade (meses)	Entradas x Saídas	Arrecadação x Saídas	Numérica Pertinência Baixo / Médio / Alto	Linguística
E1	12	0,007860	0,000678	0,0 / 0,0 / 0,819 	Alto
E2	9	0,113283	0,005299	0,0 / 0,0 / 0,802744 	Alto
E3	11	2,069762	0, 141350	0,374 / 0,0 / 0,0 	Baixo
E4	12	0,671348	0,071301	0,0 / 0,319 / 0,180	Médio

					
E5	12	0,004138	0,153633	0,809 / 0,0 / 0,0 	Baixo
E6	12	0,014187	0,067587	0,0 / 0,121 / 0,378 	Alto
E7	4	0,042851	0,072284	0,0 / 0,371 / 0,128 	Médio

O resultado da aplicação do sistema nebuloso eliminou 24 ocorrências que poderiam influenciar negativamente o aprendizado da rede neural, uma vez que estes estavam rotulados como “noteiras”, no entanto, de acordo com as regras definidas pelo especialista, essas ocorrências foram classificadas como não “noteiras”.

A correção dos dados é importantíssima para a qualidade do aprendizado. Eliminar o máximo de ruído é fundamental para que a rede neural aprenda o mais corretamente possível.

5.5 A IDENTIFICAÇÃO DAS INFORMAÇÕES MAIS RELEVANTES

A seleção das variáveis mais relevantes é importante para aumentar a qualidade do modelo. Ela minimiza a redundância e maximiza a relevância de cada variável além de otimizar a velocidade de treinamento do modelo por conta da eliminação de variáveis que não iriam influenciar tanto no resultado. Em termos de

aprendizado de máquina, pode-se pensar em como as variáveis de entrada correspondem à sua saída.

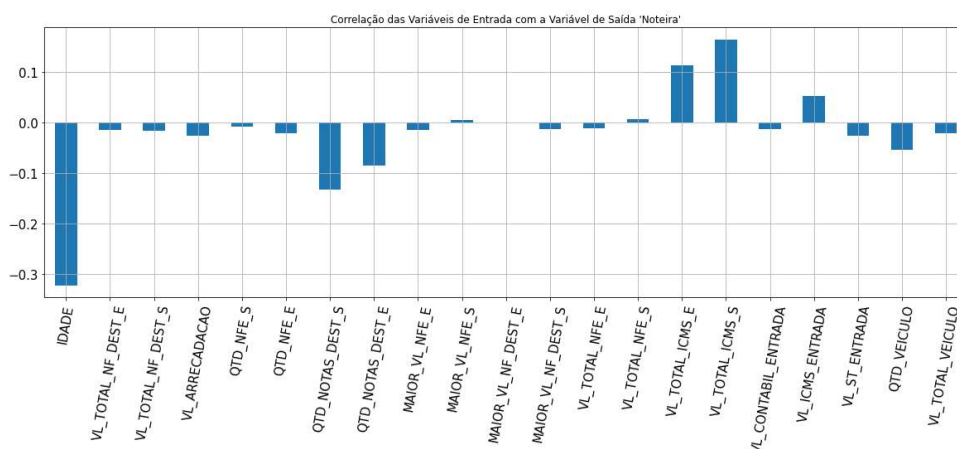
Para analisar a relevância das variáveis e decidirmos, com confiança, quais delas serão utilizadas no modelo de rede neural, calculamos o coeficiente de correlação de Pearson de cada variável de entrada com a variável de saída, isto é, cada uma das variáveis disponíveis na

Tabela 3 com a variável de saída “Noteira/Não Noteira”. Com x_i e y_i sendo os valores das variáveis de entrada X e de saída Y e \bar{x} e \bar{y} sendo, respectivamente, as médias dos valores x_i e y_i , a correlação de Pearson é definida pela equação (11).

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum(x_i - \bar{x})^2)(\sum(y_i - \bar{y})^2)}} \quad (11)$$

A correlação varia de -1 a 1 e um valor positivo indica que a variável independente aumenta enquanto a variável dependente também aumenta. Já um valor negativo indica que enquanto a variável independente diminui a dependente aumenta. Finalmente, quando a correlação é igual a zero, não há relação entre as variáveis ou, quando se aproxima de zero, há pouca relação entre as variáveis. A Figura 22 mostra a correlação de cada variável de entrada com a variável de saída.

Figura 22- Correlação das variáveis independentes com a variável dependente “Noteira”



Fonte: elaborado pelo autor

Como se percebe, no gráfico da Figura 22, existem algumas variáveis com correlação próxima a zero e, desse modo, são de baixa relevância e, portanto, podem ser descartadas. A Tabela 11 mostra as 11 variáveis selecionadas.

Tabela 11- Variáveis mais relevantes selecionadas após o resultado da análise de correlação

VARIÁVEL	DESCRIÇÃO
IDADE	Idade da empresa em meses
VL_TOTAL_NF_DEST_E	Valor total de NF de entrada recebida pelo contribuinte no mês
VL_TOTAL_NF_DEST_S	Valor total de NF de saída recebida pelo contribuinte no mês
VL_ARRECADACAO	Valor arrecadado do contribuinte no mês
QTD_NFE_S	Quantidade de NFE emitidas de saída pelo contribuinte no mês
QTD_NFE_E	Quantidade de notas emitidas de entrada pelo contribuinte no mês

QTD_NOTAS_DEST_S	Quantidade de NFE de saída recebida pelo contribuinte no mês
QTD_NOTAS_DEST_E	Quantidade de notas de entrada recebida pelo contribuinte no mês
MAIOR_VL_NFE_E	Maio valor da NFE emitida de entrada pelo contribuinte no mês
MAIOR_VL_NFE_S	Maio valor da NFE emitida de saída pelo contribuinte no mês
MAIOR_VL_NF_DEST_E	Maio valor da NFE de entrada recebida pelo contribuinte no mês
MAIOR_VL_NF_DEST_S	Maio valor da NFE de saída recebida pelo contribuinte no mês
VL_TOTAL_NFE_E	Valor total de NF de entrada emitida pelo contribuinte no mês
VL_TOTAL_NFE_S	Valor total de NF de saída emitida pelo contribuinte no mês
VL_TOTAL_ICMS_E	Valor total de ICMS na NF de entrada pelo contribuinte no mês
VL_TOTAL_ICMS_S	Valor total de ICMS na NF de saída pelo contribuinte no mês
VL_CONTABIL_ENTRADA	Valor contábil de notas de entrada: SUM(NVL(VL_TOTAL_NFE_E,0) + NVL(VL_TOTAL_NF_DEST_S,0)) - SUM(NVL(VL_TOTAL_NF_DEST_E,0))
VL_ICMS_ENTRADA	Valor contábil de ICMS de entrada: SUM(NVL(VL_TOTAL_ICMS_E,0) + NVL(VL_TOTAL_ICMS_DEST_S,0)) - SUM(NVL(VL_TOTAL_ICMS_DEST_E,0))
VL_ST_ENTRADA	Valor contábil de ST de entrada: SUM(NVL(VL_TOTAL_ST_DEST_S,0) + NVL(VL_TOTAL_ST_E,0))
QTD_VEICULO	Quantidade de veículo(s) vinculado(s) ao contribuinte
VL_TOTAL_VEICULO	Valor venal total do(s) veículo(s) vinculado(s) ao contribuinte

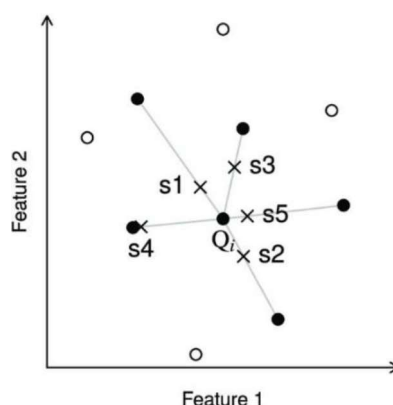
5.6 O PROBLEMA DAS AMOSTRAS DESBALANCEADAS

É natural que se tenha muito menos casos de fraude do que de não fraude. Isso faz com que tenhamos muito menos amostras de casos positivos de fraude, causando um desequilíbrio no treinamento da rede neural. “O desempenho dos algoritmos de aprendizado de máquina é normalmente avaliado usando precisão preditiva. No entanto, isso não é apropriado quando os dados estão desequilibrados e/ou os custos de diferentes erros variam acentuadamente” (CHAWLA, 2002).

Uma maneira de lidar com conjuntos de dados desbalanceados é superamostrar a classe minoritária e a forma mais simples de fazer isso é duplicar ocorrências dessa classe. No entanto, a simples duplicação não adiciona novas informações ao modelo e, portanto, não agrega qualidade ao aprendizado.

Com o objetivo de amenizarmos os problemas que um conjunto de dados desbalanceados provoca no aprendizado de uma rede neural, aplicamos a técnica SMOTE, por (CHAWLA, 2002), (*Synthetic Minority Oversampling Technique*) ou Técnica de Sobreamostragem Minoritária Sintética. Esse método consiste em selecionar exemplos próximos entre as variáveis mais próximas e criar novos pontos entre os pontos existentes. Um exemplo é aleatoriamente selecionado e, então, um vizinho, dentre tipicamente 5 vizinhos, é também selecionado aleatoriamente e, enfim, um exemplo sintético é criado em um ponto aleatório entre os dois exemplos selecionados anteriormente. A Figura 23 mostra como a técnica SMOTE cria as variáveis.

Figura 23 - Pontos sintéticos (s1 a s5) gerados pelo SMOTE ao longo das linhas de conexão entre um ponto (ponto preto denotado por Q_i) e seus k vizinhos mais próximos (pontos pretos)



Adaptado de: <https://doi.org/10.1371/journal.pone.0190476.g002>

A Tabela 12 mostra a quantidade de dados, por tipo de classificação antes e depois da aplicação da técnica.

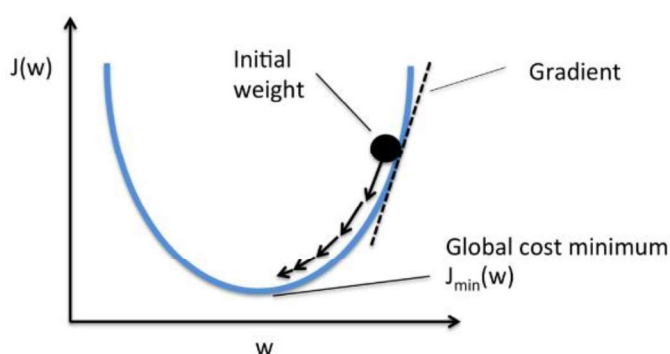
Tabela 12- Quantidade de Dados Antes e Depois do SMOTE

Tipo de Classificação	"noteiras"	Não "noteiras"
Antes do SMOTE	2.021	10.191
Depois do SMOTE	10.191	10.191

5.7 PADRONIZAÇÃO DOS DADOS

A descida de gradiente é um algoritmo de otimização iterativa de primeira ordem para encontrar um mínimo local de uma função. A função de gradiente descendente caminha pelos pontos de dados enquanto é aplicada ao conjunto de dados, passo a passo e, assim, se a distância entre os pontos de dados aumentar, o tamanho do passo mudará e o movimento da função não será suave e, conseqüentemente, o algoritmo precisará de mais passos para encontrar o ponto mínimo da função. A Figura 24 ilustra como funciona uma função de gradiente descendente.

Figura 24- Representação da Descida do Gradiente (com o objetivo de minimizar a função de custo)



Fonte: <https://www.deeplearningbook.com.br/aprendizado-com-a-descida-do-gradiente/>

Portanto, aplicar uma escala nos dados e, conseqüentemente, diminuindo a distância entre seus pontos, é fundamental para o bom desempenho da rede neural. Aplicamos a padronização de dados utilizando a equação (12).

$$x_{\text{padronizado}} = \frac{x - \text{média}(x)}{\text{desvio padrão}(x)} \quad (12)$$

6. O MODELO ATUAL DE DETECÇÃO DE FRAUDE

Atualmente, o trabalho de análise e investigação de fraude de ICMS, feito pela SEFAZ-RJ, é realizado através de ferramentas tradicionais de bancos de dados que utilizam a linguagem SQL, baseada na álgebra relacional criada pelo matemático britânico Edgar F. Codd (CODD, 1970), capaz de realizar as operações básicas da teoria clássica dos conjuntos, como junções e interseções, para fazer todos os filtros e cruzamentos de informações necessários a fim de se chegar à identificação de possíveis fraudadores. A desvantagem em se utilizar tais ferramentas está na sua imprecisão em medir o “meio-termo”. Isso não é possível utilizando teoria clássica dos conjuntos porque ela se baseia na lógica Booleana (BOOLE, 1847), ou seja, são considerados apenas os valores “verdadeiro” ou “falso” para tomadas de decisão.

Um projeto predecessor a este foi construído utilizando lógica nebulosa a partir de regras elaboradas por um especialista em detecção de empresas “noteiras”. A vantagem do sistema é que ele funciona com base na intuição e experiência do perito e, portanto, consegue classificar a maioria das empresas que apresentam dados parecidos.

Contudo, é um sistema estático e que se torna complexo para o especialista definir as regras quando o número de variáveis é muito grande. As regras são definidas e permanecem sempre as mesmas, o que impede o sistema de se adaptar às mudanças de comportamento financeiro das empresas.

7. O MODELO PROPOSTO

O autor deste estudo propõe um modelo de rede neural artificial do tipo *perceptron* de múltiplas camadas *feedforward* utilizando o algoritmo *backpropagation* para o seu treinamento. Para determinar a quantidade de camadas escondidas e de neurônios em cada camada foi utilizado o método de “Validação Cruzada de Pesquisa em Grade” da biblioteca *Scikit-Learn* (PEDREGOSA *et al.*, 2011) , onde foram testados vários modelos de classificação com diferentes parametrizações. A escolha de um modelo de classificação é, normalmente, feita após exaustivos testes, experimentações e análise da sua pontuação que é obtida com base nos parâmetros mostrados na Tabela 13.

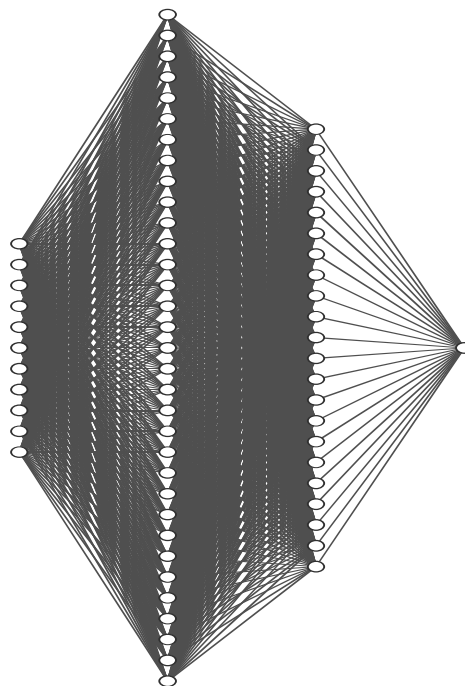
Tabela 13- Modelos de Classificação Estudados

Tipo de Modelo de Classificação	Parâmetros	
	Nome	Valores Testados
<i>Multi-layer Perceptron</i>	<i>max_iter</i>	5.000
	<i>alpha</i>	0,0001; 0,001; 0,01
	<i>hidden_layers</i>	11 -> 7 22 -> 11 33 -> 22 -> 11 33 -> 22
	<i>solver</i>	'lbfgs' 'sgd' 'adam'
	<i>learning_rate</i>	'constant' 'invscaling' 'adaptive'
	<i>activation</i>	<i>identity</i>
		<i>logistic</i>
		<i>tanh</i>
		<i>Relu</i>

O modelo indicado com a melhor performance foi o *multi-layer perceptron* com 11 neurônios na camada de entrada, que correspondem às 11 melhores variáveis

selecionadas anteriormente, 33 neurônios na primeira camada escondida, 22 neurônios na segunda camada escondida e 1 neurônio na camada de saída. No APÊNDICE B pode-se observar o código em Python com todas as etapas de construção do modelo. Para fins de exemplo, a Figura 25 mostra a arquitetura da rede neural utilizada.

Figura 25- Exemplo de arquitetura da rede neural multi-layer *perceptron*



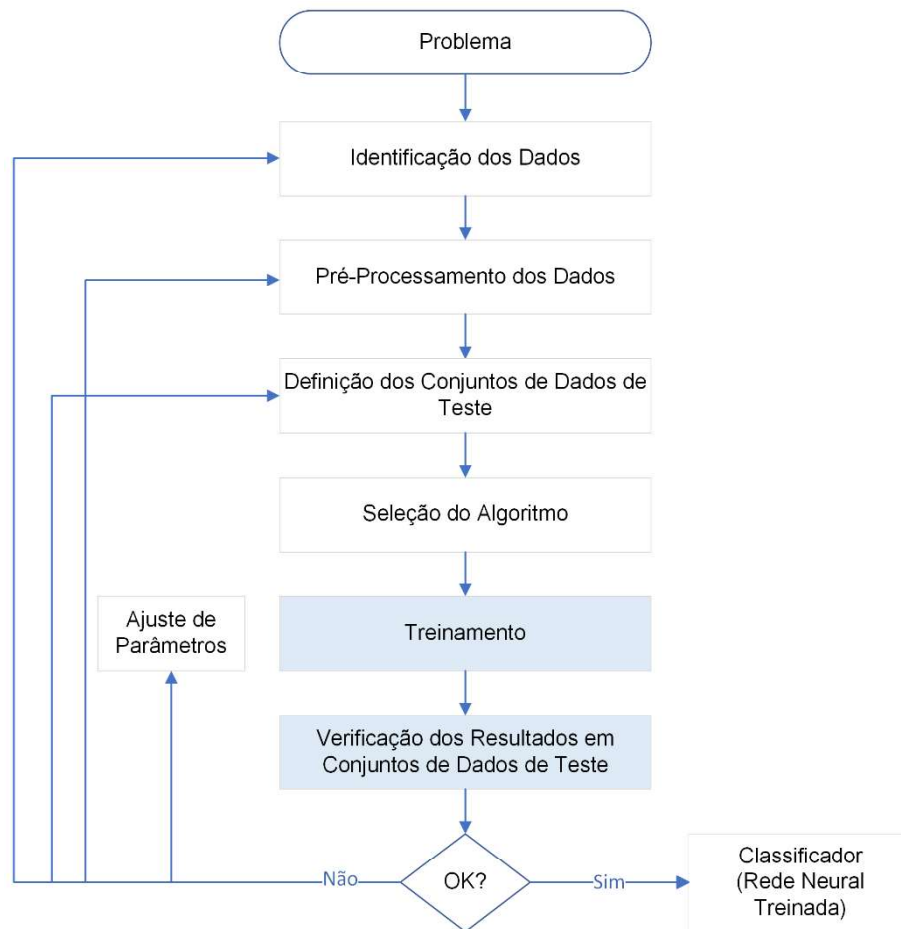
Fonte: elaborado pelo autor através de <https://alexlenail.me/NN-SVG/index.html>

7.1 PROCESSO DE CONSTRUÇÃO

O processo de construção de uma rede neural artificial para classificação, segundo Yagang Zhang (ZHANG, 2010), pode ser dividido nas etapas ilustradas na

Figura 26. Ao final do processo obtém-se o classificador, ou seja, a rede neural treinada com seus pesos e vieses ajustados de acordo com os dados utilizados durante o treinamento.

Figura 26 - Processo de aprendizado de uma rede neural



Fonte: adaptado de (ZHANG, 2010)

7.2 TREINAMENTO E VALIDAÇÃO

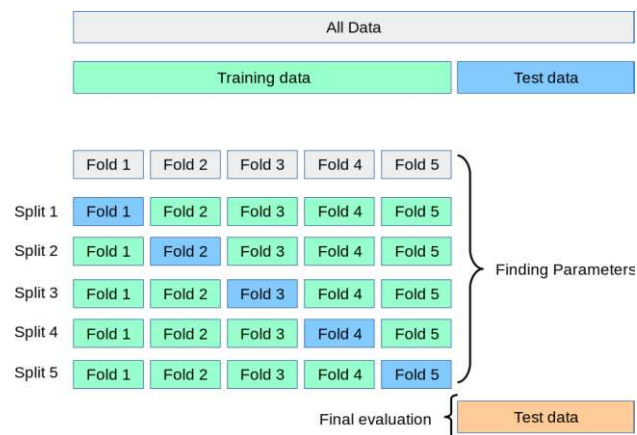
A qualidade do aprendizado de uma rede neural está diretamente relacionada com a qualidade do seu treinamento. Uma rede neural deve conseguir classificar novos dados, ou seja, dados desconhecidos, não vistos durante o treinamento, com um grau satisfatório de acerto. Um modelo que apenas classificasse os rótulos das amostras que acabou de ver teria uma pontuação perfeita, mas falharia em prever qualquer coisa em dados não conhecidos. Essa situação é conhecida como *overfitting*, e pode ser evitada utilizando-se a técnica chamada *k-fold* que embaralha e divide os dados de forma que, a cada treinamento, os dados de treinamento são

inteiramente diferentes dos dados de teste, ou seja, os testes são feitos em dados desconhecidos no momento do treino. Este é o algoritmo básico dessa técnica:

1. Embaralha-se, aleatoriamente , o conjunto de dados
2. Divide-se o conjunto em **k** grupos
3. Para cada grupo único:
 - 3.1. Considera-se este grupo como um conjunto de dados de teste
 - 3.2. Considera-se os grupos restantes como um conjunto de dados de treinamento
 - 3.3. Submete-se o conjunto de treinamento à rede neural e o avalie no conjunto de teste
 - 3.4. Guarda-se a pontuação da avaliação e descarte o modelo
4. Calcula-se a média das pontuações obtidas no passo 3.4

A Figura 27 mostra estratégia da divisão para as etapas de treinamento e teste.

Figura 27- Técnica *k-fold* para treinamento e teste da rede neural



Fonte: https://scikit-learn.org/stable/modules/cross_validation.html

8. RESULTADOS

Os resultados finais obtidos da classificação, de um total de 6.115 casos:

- a) **5.315** casos foram **corretamente** identificados
- b) **800** casos foram **incorretamente** identificados sendo que:
 - a. **583** casos foram identificados como sendo de empresas **noteiras, porém, não eram de fato**;
 - b. **217** casos foram identificados como sendo de empresas **não noteiras, porém, eram de fato**.

8.1 ANÁLISE DOS RESULTADOS

Podemos verificar a qualidade do modelo proposto através da sua matriz-confusão, que mostra a quantidade de acertos e erros positivos e negativos. O melhor cenário, no caso de fraude, é que se tenha o mínimo possível de casos de falsos negativos, ou seja, quando a rede classifica uma empresa “noteira” como “não noteira”. A rede proposta por este estudo consegue alcançar este objetivo, obtendo apenas 3,55% dos casos durante o treinamento. Também é obtida uma boa taxa de falsos positivos, onde uma empresa é identificada como “noteira” que, na verdade, não é. Neste caso o prejuízo seria apenas de fazer uma ação fiscalizatória desnecessária. A Figura 28 mostra a matriz confusão dos resultados obtidos.

Figura 28- Matriz confusão



Fonte: elaborado pelo autor

A partir da matriz confusão, podemos extrair as seguintes métricas do modelo:

- 1- Acurácia: é a quantidade de vezes que o modelo classificou corretamente uma empresa como “noteira” e “não noteira” e é representada pela equação (13) e aplicada na equação (14).

$$\text{Acurácia} = \frac{\text{Verdadeiros Positivos (VP)} + \text{Verdadeiros Negativos (VN)}}{\text{Total}} \quad (13)$$

$$\text{Acurácia} = \frac{2.435 + 2.880}{6.115} = 0,86 \quad (14)$$

- 2- Precisão: de todas as empresas classificadas como “noteira”, quantas são realmente “noteiras” e é representada pela equação (15) e aplicada na equação (16).

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos (VP)}}{\text{Verdadeiros Positivos (VP)} + \text{Falsos Positivos (FP)}} \quad (15)$$

$$\text{Precisão} = \frac{2.435}{2.435 + 583} = 0,80 \quad (16)$$

- 3- Revocação ou sensibilidade: de todas as empresas que realmente são “noteiras”, qual percentual foi identificado corretamente pelo modelo. Ela mede a sensibilidade do modelo e é representada pela equação (17) e aplicada na equação (18).

$$\text{Revocação} = \frac{\text{Verdadeiros Positivos (VP)}}{\text{Verdadeiros Positivos (VP)} + \text{Falsos Negativos (FP)}} \quad (17)$$

$$\text{Revocação} = \frac{2.345}{2.435 + 217} = 0,91 \quad (18)$$

4- *F1-score*: é a combinação de Precisão, Revocação e média harmônica que mede a qualidade geral do modelo ou o quadrado da média geométrica dividido pela média aritmética, representada pela equação (19) e aplicada na equação (20).

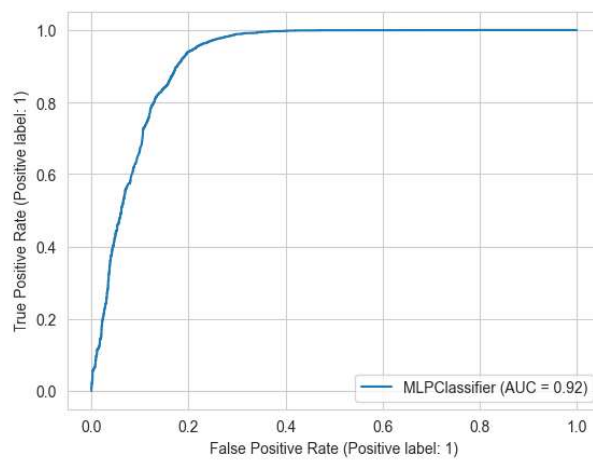
$$F1 = \frac{2 * \text{Precisão} * \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (19)$$

$$F1 = \frac{2 * 0,80 * 0,91}{0,80 + 0,91} = 0,85 \quad (20)$$

Outra maneira de medir e avaliar a qualidade do modelo de classificação é usar o gráfico da curva característica de operação do receptor, ou curva ROC (*Receiver Operating Characteristic*). A curva ROC é criada plotando a taxa de verdadeiros positivos (TPR ou *True Positive Rate*) em relação à taxa de falsos positivos (FPR – *False Positive Rate*), como é mostrado na Figura 29.

A taxa de verdadeiros positivos também é conhecida como sensibilidade, revocação ou probabilidade de detecção.

Figura 29- Curva característica de operação do receptor do modelo de classificação



Fonte: elaborado pelo autor

9. CONCLUSÕES

A análise da matriz-confusão e da curva ROC deixa evidente um nível satisfatório da acurácia e precisão do modelo construído, com índice de acerto satisfatório bem como produzindo menos erros falsos negativos (empresas “noteiras” não detectadas como tal) do que falsos positivos (empresas “não noteiras” detectadas como “noteiras”).

O estudo responde à primeira questão, de quais seriam as informações fiscais do contribuinte mais relevantes para a análise e identificação da fraude, aplicando a análise de correlação entre as variáveis de entrada e a de saída, minimizando a quantidade de cálculos desnecessários e otimizando o modelo.

Responde também à segunda questão, de qual seria o modelo de rede neural mais adequado para detectar a fraude com acurácia aceitável, utilizando a técnica de pesquisa em grade com validação cruzada, que determinou a arquitetura da rede, de 2 camadas escondidas contendo 33 e 22 neurônios cada uma, nesta ordem, ou seja, uma rede com 11 neurônios na camada de entrada, 33 e 22 nas camadas escondidas e 1 neurônio de saída que classifica a empresa como “Noteira” para o valores iguais a 1 (um) ou “Não Noteira” para valores iguais a 0 (zero).

Por fim, responde à terceira pergunta, de qual seria o menor período de coleta de dados fiscais necessários para detectar a fraude o mais precoce possível, concluindo que o valor desta quantidade é de 1 (um) mês, importando muito mais a quantidade total de dados para o treinamento, assim como a sua padronização e balanceamento, do que a quantidade de dados individuais, por empresa de modo que, com base nos dados de empresas já existentes, as novas empresas, mesmo com apenas 1 mês de existência, já poderão ser detectadas como empresas fraudadoras.

Desse modo, este estudo oferece as respostas necessárias para a automatização da detecção de empresas “noteiras” com base nas informações fiscais disponíveis.

O método utilizado permite aos operadores das fiscalizações estaduais obterem uma indicação objetiva e confiável das empresas “noteiras”. O modelo consegue responder à todas as questões iniciais e possui um número mínimo de

falsos negativos, ou seja, da possibilidade de não detectar um caso verdadeiro de empresa “noteira”.

Ficou evidente que o tratamento dos dados foi fundamental para o aumento de qualidade do modelo. A correção das informações utilizando-se um sistema *fuzzy* mostrou-se importante na eliminação de informações que teriam o potencial de “ensinar errado” para a rede neural o que é uma empresa “noteira”. Ainda, a aplicação da técnica SMOTE permitiu evitar que o treinamento da rede neural se tornasse enviesado, equilibrando a quantidade de amostras de empresas “noteiras” e “não noteiras”. Portanto, dar importância à fase de seleção e coleta de dados é fundamental para o sucesso da construção de um modelo de rede neural efetivo.

Assim, este estudo contribui como base para que todas as unidades da federação possam encurtar o seu caminho a fim de obter uma fiscalização antifraude moderna, rápida e confiável.

10. SUGESTÕES DE TRABALHOS FUTUROS

A rotulação inicial, feita manualmente, atribuindo a definição de “Noteira” e “Não Noteira” para todos os meses analisados, independente da empresa ter se comportado alguns meses como “noteira” e outros meses como “não noteira”, definitivamente não contribuiu para o aumento da assertividade do modelo. Apesar da aplicação do modelo de lógica nebulosa ter eliminado alguns registros provavelmente rotulados erroneamente, a situação não é a ideal.

Uma sugestão para trabalhos futuros é a de investir-se mais na qualidade da rotulação dos dados. Mesmo que o trabalho atual seja manual, vale a pena se ater na qualidade das informações registradas e analisadas para utilizações futuras.

Outra sugestão é a de unir o resultado da classificação, produto da aplicação do modelo proposto por este estudo, com o resultado da operação de fiscalização para criar ou atualizar os rótulos e servir como insumo para o próximo treinamento da rede neural. Assim, com uma espécie de retroalimentação, a rede se tornaria cada vez mais eficaz, uma vez que utilizaria como entrada o resultado validado da sua última utilização.

Além disso, é importante construir uma ferramenta de forma a automatizar o processo de fiscalização utilizando o modelo proposto, como o que a SEFAZ-RJ está desenvolvendo, denominado SARF, Sistema de Análise de Regularidade Fiscal, onde utilizará o modelo final, resultado deste estudo, no seu trabalho diário. A Figura 30 mostra uma das telas do sistema SARF.

Figura 30- Tela exemplo do sistema SARF

ID	Nome	Descrição	RMA	Dados	Status	Gráfico
1	Identificação de Empresas Noteiras	Identificar as empresas noteiras através de parâmetros definidos	Noteira - Lógica Fuzzy	Histórico de 12 meses de E x S, A x U e C x C	Ativo	02/08/2021
2	Arrecadação do Setor Elétrico - SARIMAX	Estimador de arrecadação do setor elétrico com algoritmo SARIMAX	Arrecadação do Setor Elétrico - SARIMAX	Arrecadação do Setor de Energia Elétrica	Ativo	01/09/2021
21	Arrecadação Total Mensal - ARIMA	Estimador de arrecadação total mensal usando algoritmo ARIMA	Arrecadação Total Mensal - ARIMA	Arrecadação Total Mensal	Ativo	02/11/2021

Fonte: SEFAZ-RJ

REFERÊNCIAS BIBLIOGRÁFICAS

A Comprehensive Review on Radiomics and Deep Learning for Nasopharyngeal Carcinoma Imaging. **Research Gate**, 2022. Disponível em: https://www.researchgate.net/figure/Relationship-between-artificial-intelligence-machine-learning-neural-network-and-deep_fig3_354124420. Acesso em: 1 dez. 2022.

BISHOP, Christopher M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006.

BOOLE, George. **The mathematical analysis of logic**. [S.l.]: Philosophical Library, 1847.

BRASIL, Constituição Federal do. Artigo Nº. 155 da Constituição Federal do Brasil. **Senado Federal**, 31 out. 2022. Disponível em: <https://legis.senado.leg.br/sdleg-getter/documento?dm=4345148&ts=1594017164326&disposition=inline>.

CHAWLA, Nitesh V. et al. SMOTE: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, 2002. 321-357.

CODD, Edgar F. A relational model of data for large shared data banks. **Association for Computing Machinery**, 1970. 377-387.

CONTABILIDADE GERAL RJ. Relatórios contábeis e de gestão fiscal das finanças públicas do Estado - Exercício de 2021. **Contabilidade Geral RJ - Secretaria de Fazenda do Estado do Rio de Janeiro**, 06 nov. 2022. Disponível em: http://www.contabilidade.fazenda.rj.gov.br/contabilidade/faces/oracle/webcenter/port alapp/pages/navigation-renderer.jspx?_afLoop=106041022716890600&datasource=UCMServer%23dDocName%3AWCC42000030440&_adf.ctrl-state=mb65doj36_9.

COVER, T M; HART, P E. Nearest Neighbor Pattern Classification. **IEEE Trans. Inform. Theory IT**, 1967. 21-27.

CRCSC. Notícias. **crcsc.org.br**, 1 jul. 2019. Disponível em: <https://www.crcsc.org.br/noticia/view/7569>.

DATASEBRAE. **datasebrae.com.br**, 2022. Disponível em: <https://datasebrae.com.br/painel-de-abertura-e-fechamento-de-empresas-rj/>. Acesso em: 23 nov. 2022.

FORBES, Revista. **Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says**. [S.l.]: Revista Forbes, 2016.

HASTIE, Trevor and Tibshirani, Robert and Friedman, Jerome. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer Science & Business Media, 2009.

HAYKIN, Simon. **Neural networks and learning machines / Simon Haykin.—3rd ed.** 3ª. ed. New Jersey: Pearson Education, Inc, 2009.

HEBB, Donald Olding. **The Organization of Behavior**. Quebec, Canadá: John Wiley & Sons Inc., 1949.

HEBB, DONALD OLDING. **The organization of behavior: A neuropsychological theory**. [S.l.]: Psychology Press, 2005.

KNUTH, Donald E. **Computers & Typesetting**. Massachusetts: Addison-Wesley, 1984.

LIN, C. ; LEE, C.. **Neural Fuzzy Systems: A Neuro-fuzzy Synergism to Intelligent Systems**. New Jersey: Prentice Hall P T R, 1996.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, dez. 1943. 115-133.

MINAS, Agência. **hojeemdia.com.br**, 07 dez. 2018. Disponível em: <https://www.hojeemdia.com.br/operac-o-em-minas-de-combate-a-empresas-noteiras-1.677559>.

MIZUMOTO, Masaharu. **Handbook of Fuzzy Computation**. Bristol BS1 6BE, UK: IOP Publishing Ltd, 1998.

MP-BA. MP baiano participa de operação que combate sonegação fiscal em oito estados e no Distrito Federal. **Ministério Público da Bahia**, 10 mar. 2020. Disponível em: <https://www.mpba.mp.br/noticia/50199>.

NOTÍCIAS SEFAZ-RJ. **fazenda.rj.gov.br**, 12 set. 2019. Disponível em: http://www.fazenda.rj.gov.br/sefaz/content/conn/UCMServer/path/Contribution%20Folders/site_fazenda/imprensa/noticias/Secretaria%20de%20Fazenda%20fiscaliza%20empresas%20noteiras.htm.

NOTÍCIAS SEFAZ-RJ. **fazenda.rj.gov.br**, 23 dez. 2020. Disponível em: http://www.fazenda.rj.gov.br/sefaz/content/conn/UCMServer/path/Contribution%20Folders/site_fazenda/imprensa/noticias/Secretaria%20de%20Fazenda%20fiscaliza%20empresas%20noteiras%20na%20Opera%C3%A7%C3%A3o%20Ma%C3%A7arico%20IV.htm.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, 2011. 2825-2830. Disponível em: <https://scikit-learn.org/stable/index.html>.

ROSENBLATT, Frank. Perceptron simulation experiments. **Proceedings of the IRE**, 1960. 301-309.

RUMELHART, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533-536, 1986. ISSN <https://doi.org/10.1038/323533a0>.

SABBAG, Eduardo. **Manual de Direito Tributário**. São Paulo: Saraiva, 2017.

SAMUEL, Arthur L. Some studies in machine learning using the game of checkersSome studies in machine learning using the game of checkers. **IBM Journal of research and development**, 1959. 210-229.

SARTORI, Millena. Empresas noteiras são identificadas na região de Ponta Grossa. **dcmais.com.br**, 07 fev. 2022. Disponível em: <https://dcmais.com.br/ponta-grossa/empresas-noteiras-sao-identificadas-na-regiao-de-ponta-grossa/>.

SAYEG, R. N. Soneração tributária e complexidade. **RAE eletrônica**, 2003. 5.

SEBRAE. Pannel de abertura e fechamento de empresas - RJ. **datasebrae**, 28 ago. 2022. Disponível em: <https://datasebrae.com.br/pannel-de-abertura-e-fechamento-de-empresas-rj/>.

SEEC-DF. **SECRETARIA DE ECONOMIA DO DISTRITO FEDERAL**, 29 jun. 2022. Disponível em: <https://www.economia.df.gov.br/operacao-recupera-r-492-milhoes-em-creditos-tributarios-sonogados/>.

SEF.SC.GOV.BR. **SEFAZ-SC**, 01 jul. 2019. Disponível em: <https://www.sef.sc.gov.br/midia/noticia/2224>.

SEFAZ-MA. **sistemas1.sefaz.ma.gov.br**, 1 dez. 2018. Disponível em: <https://sistemas1.sefaz.ma.gov.br/portalsefaz/jsp/noticia/noticia.jsf?codigo=5302>.

SEFAZ-PB. Notícias. **SEFAZ-PB**, 20 nov. 2019. Disponível em: <https://www.sefaz.pb.gov.br/announcements/8714-experiencias-bem-sucedidas-do-pais-no-combate-as-empresas-noteiras-sao-compartilhadas>.

SEFAZ-RJ. Notícia. **SEFAZ-RJ**, 27 nov. 2019. Disponível em: <https://www.fazenda.rj.gov.br/sefaz/faces/oracle/webcenter/portalapp/pages/navigation-on-renderer.jsp?datasource=UCMServer%23dDocName%3AWCC42000003549>.

SEFAZ-SC. **sef.sc.gov.br**, 2020. Disponível em: <https://www.sef.sc.gov.br/midia/noticia/1736>.

SEFAZ-SP. Notícias. **SEFAZ-SP**, 26 ago. 2020. Disponível em: <https://portal.fazenda.sp.gov.br/Noticias/Paginas/Secretaria-da-Fazenda-e-Planejamento-deflagra-Opera%C3%A7%C3%A3o-Plassein.aspx>.

SEFAZ-SP. Fiscos de SP e RJ fazem operação conjunta para combater fraudes de R\$ 600 milhões no ICMS. **Secretaria de Fazenda de São Paulo**, 24 fev.

2021. Disponível em: [https://portal.fazenda.sp.gov.br/Noticias/Paginas/Fiscos-de-SP-e-RJ-fazem-opera%C3%A7%C3%A3o-conjunta-para-combater-fraudes-de-R\\$-600-milh%C3%B5es-no-ICMS.aspx](https://portal.fazenda.sp.gov.br/Noticias/Paginas/Fiscos-de-SP-e-RJ-fazem-opera%C3%A7%C3%A3o-conjunta-para-combater-fraudes-de-R$-600-milh%C3%B5es-no-ICMS.aspx).

SEYMOUR, Geisser. **Predictive Inference: An Introduction**. Nova Iorque: Springer Science+ Business Media Dordrecht, 1993.

SINDIFISCAL-ES. **sindifiscal-es.org.br**, 24 jun. 2019. Disponível em: <http://www.sindifiscal-es.org.br/noticias/778/fraude-de-r-32-bi-no-setor-de-sucatas-e-exposta-pelo-es-em-encontro-nacional-de-intelig%C3%Aancia>.

YAMAO, Celina. A História do Imposto Sobre Circulação de Mercadorias – do IVM ao ICMS. **Revista Jurídica UNICURITIBA**, p. 36, 2014. Disponível em: <http://revista.unicuritiba.edu.br/index.php/RevJur/article/view/990/681>.

ZADEH *et al.* **Fuzzy Logic Theory and Applications: Part I and Part II**. Berkeley: Kindle, 2018.

ZADEH, L. A. Fuzzy Sets. **Information and Control**, ago. 1965. 338-353.

ZHANG, Yagang. **Supervised Machine Learning**. Vukovar, Croatia: In-Teh, 2010. 14 p. ISBN 978-953-307-034-6.

APÊNDICE A – Script Python do Sistema Fuzzy para eliminar ruídos na rotulação dos dados sobre empresas noteiras

```
1      ### md
2      **Instalando dependências**
3      ###
4      !pip install sklearn
5      !pip install scikit-fuzzy
6      ### md
7      **Importando dependências**
8      ###
9      import numpy as np
10     import skfuzzy as fuzz
11     from skfuzzy import control as ctrl
12     import matplotlib.pyplot as plt
13     import pandas as pd
14     import hashlib
15     import datetime
16     ### md
17     # **Variáveis do Problema**
18
19     ### md
20     **Variáveis de Entrada**
21     ### md
22     1- Idade da empresa em meses
23     ###
24     idade = ctrl.Antecedent(np.arange(0, 15, 1), 'idade')
25     idade['MuitoJovem'] = fuzz.trapmf(idade.universe, [0, 0, 3, 7])
26     idade['Jovem'] = fuzz.trapmf(idade.universe, [4, 6, 8, 12])
27     idade['NaoJovem'] = fuzz.trapmf(idade.universe, [10, 13, 15, 150])
28
29     idade.view()
30     ### md
31     2- Fator Saída X Entrada: valores de emissão de notas fiscais de saída
32     e de entrada
33     ###
34     saidaXentrada = ctrl.Antecedent(np.arange(0, 1, 0.01),
35     'saidaXentrada')
36     saidaXentrada['CompraPouquissimo'] =
37     fuzz.trapmf(saidaXentrada.universe, [0.0, 0.0, 0.05, 0.10])
38     saidaXentrada['CompraPouco'] = fuzz.trimf(saidaXentrada.universe,
39     [0.075, 0.125, 0.45])
40     saidaXentrada['CompraOEsperado'] = fuzz.trimf(saidaXentrada.universe,
41     [0.40, 0.55, 0.75])
42     saidaXentrada['CompraMuito'] = fuzz.trapmf(saidaXentrada.universe,
43     [0.625, 0.75, 1, 10])
44
45     saidaXentrada.view()
46     ### md
47     3- Fator Saída X Arrecadação: valores de emissão de notas fiscais e
48     arrecadação declarada
49     ###
50     saidaXarrecadacao = ctrl.Antecedent(np.arange(0, 1.50, 0.001),
51     'saidaXarrecadacao')
52     saidaXarrecadacao['ArrecadaPouquissimo'] =
53     fuzz.trimf(saidaXarrecadacao.universe, [0, 0.0, 0.01])
54     saidaXarrecadacao['ArrecadaPouco'] =
55     fuzz.trimf(saidaXarrecadacao.universe, [0.008, 0.0425, 0.070])
```

```

46  saidaXarrecadacao['ArrecadaOEsperado'] =
fuzz.trimf(saidaXarrecadacao.universe, [0.060, 0.100, 0.140])
47  saidaXarrecadacao['ArrecadaMuito'] =
fuzz.trapmf(saidaXarrecadacao.universe, [0.120, 1.40, 1.50, 14])
48
49  saidaXarrecadacao.view()
50  ### md
51  # **Variáveis de Saída**
52  ### md
53  Grau de sonegadora
54  ###
55  noteira = ctrl.Consequent(np.arange(0, 101, 1), 'GrauSonegadora')
56  noteira['PoucoProvavel'] = fuzz.trimf(noteira.universe, [0, 25, 60])
57  noteira['Provavel'] = fuzz.trimf(noteira.universe, [50, 60, 80])
58  noteira['MuitoProvavel'] = fuzz.trimf(noteira.universe, [70, 90, 100])
59  noteira.view()
60  ### md
61  # **Regras**
62  ###
63  rules = []
64  ### md
65  Muito Jovem
66  ###
67  rules.append(
68      ctrl.Rule(idade['MuitoJovem'] & saidaXentrada['CompraPouquissimo']
& saidaXarrecadacao['ArrecadaPouquissimo'],
69          noteira['MuitoProvavel']))
70  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaPouco'],
71          noteira['MuitoProvavel']))
72  rules.append(
73      ctrl.Rule(idade['MuitoJovem'] & saidaXentrada['CompraPouquissimo']
& saidaXarrecadacao['ArrecadaOEsperado'],
74          noteira['Provavel']))
75  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaMuito'],
76          noteira['PoucoProvavel']))
77
78  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaPouquissimo'],
79          noteira['MuitoProvavel']))
80  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaPouco'],
81          noteira['MuitoProvavel']))
82  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaOEsperado'],
83          noteira['Provavel']))
84  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaMuito'],
85          noteira['PoucoProvavel']))
86
87  rules.append(
88      ctrl.Rule(idade['MuitoJovem'] & saidaXentrada['CompraOEsperado'] &
saidaXarrecadacao['ArrecadaPouquissimo'],
89          noteira['MuitoProvavel']))
90  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaPouco'],
91          noteira['MuitoProvavel']))
92  rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaOEsperado'],
93          noteira['Provavel']))

```



```

94     rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaMuito'],
95                             noteira['PoucoProvavel'])))
96
97     rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaPouquissimo'],
98                             noteira['MuitoProvavel'])))
99     rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaPouco'],
100                             noteira['MuitoProvavel'])))
101     rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaOEsperado'],
102                             noteira['Provavel'])))
103     rules.append(ctrl.Rule(idade['MuitoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaMuito'],
104                             noteira['PoucoProvavel'])))
105
106
107     ### md
108     Jovem
109     ###
110     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraPouquissimo'] &
saidaXarrecadacao['ArrecadaPouquissimo'],
111                             noteira['MuitoProvavel'])))
112     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaPouco'],
113                             noteira['MuitoProvavel'])))
114     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraPouquissimo'] &
saidaXarrecadacao['ArrecadaOEsperado'],
115                             noteira['Provavel'])))
116     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaMuito'],
117                             noteira['PoucoProvavel'])))
118
119     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraPouco'] &
saidaXarrecadacao['ArrecadaPouquissimo'],
120                             noteira['MuitoProvavel'])))
121     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraPouco'] &
saidaXarrecadacao['ArrecadaPouco'],
122                             noteira['MuitoProvavel'])))
123     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraPouco'] &
saidaXarrecadacao['ArrecadaOEsperado'],
124                             noteira['Provavel'])))
125     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraPouco'] &
saidaXarrecadacao['ArrecadaMuito'],
126                             noteira['PoucoProvavel'])))
127
128     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraOEsperado'] &
saidaXarrecadacao['ArrecadaPouquissimo'],
129                             noteira['MuitoProvavel'])))
130     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaPouco'],
131                             noteira['MuitoProvavel'])))
132     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaOEsperado'],
133                             noteira['Provavel'])))
134     rules.append(ctrl.Rule(idade['Jovem'] &
saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaMuito'],

```

```

135         noteira['PoucoProvavel']))
136
137     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraMuito'] &
138         saidaXarrecadacao['ArrecadaPouquissimo'],
139         noteira['MuitoProvavel']))
140
141     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraMuito'] &
142         saidaXarrecadacao['ArrecadaPouco'],
143         noteira['MuitoProvavel']))
144
145     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraMuito'] &
146         saidaXarrecadacao['ArrecadaOEsperado'],
147         noteira['Provavel']))
148
149     rules.append(ctrl.Rule(idade['Jovem'] & saidaXentrada['CompraMuito'] &
150         saidaXarrecadacao['ArrecadaMuito'],
151         noteira['PoucoProvavel']))
152
153     ### md
154     Não Jovem
155     ###
156     rules.append(
157         ctrl.Rule(idade['NaoJovem'] & saidaXentrada['CompraPouquissimo'] &
158         saidaXarrecadacao['ArrecadaPouquissimo'],
159         noteira['MuitoProvavel']))
160
161     rules.append(ctrl.Rule(idade['NaoJovem'] &
162         saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaPouco'],
163         noteira['MuitoProvavel']))
164
165     rules.append(ctrl.Rule(idade['NaoJovem'] &
166         saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaOEsperado'],
167         noteira['Provavel']))
168
169     rules.append(ctrl.Rule(idade['NaoJovem'] &
170         saidaXentrada['CompraPouquissimo'] & saidaXarrecadacao['ArrecadaMuito'],
171         noteira['PoucoProvavel']))
172
173     rules.append(ctrl.Rule(idade['NaoJovem'] &
174         saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaPouquissimo'],
175         noteira['MuitoProvavel']))
176
177     rules.append(ctrl.Rule(idade['NaoJovem'] &
178         saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaPouco'],
179         noteira['MuitoProvavel']))
180
181     rules.append(ctrl.Rule(idade['NaoJovem'] &
182         saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaOEsperado'],
183         noteira['Provavel']))
184
185     rules.append(ctrl.Rule(idade['NaoJovem'] &
186         saidaXentrada['CompraPouco'] & saidaXarrecadacao['ArrecadaMuito'],
187         noteira['PoucoProvavel']))
188
189     rules.append(ctrl.Rule(idade['NaoJovem'] &
190         saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaPouco'],
191         noteira['MuitoProvavel']))
192
193     rules.append(ctrl.Rule(idade['NaoJovem'] &
194         saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaPouco'],
195         noteira['MuitoProvavel']))
196
197     rules.append(ctrl.Rule(idade['NaoJovem'] &
198         saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaOEsperado'],
199         noteira['Provavel']))
200
201     rules.append(ctrl.Rule(idade['NaoJovem'] &
202         saidaXentrada['CompraOEsperado'] & saidaXarrecadacao['ArrecadaMuito'],
203         noteira['PoucoProvavel']))
204
205

```

```

177 rules.append(ctrl.Rule(idade['NaoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaPouquissimo'],
178                     noteira['MuitoProvavel']))
179 rules.append(ctrl.Rule(idade['NaoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaPouco'],
180                     noteira['MuitoProvavel']))
181 rules.append(ctrl.Rule(idade['NaoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaOEsperado'],
182                     noteira['Provavel']))
183 rules.append(ctrl.Rule(idade['NaoJovem'] &
saidaXentrada['CompraMuito'] & saidaXarrecadacao['ArrecadaMuito'],
184                     noteira['PoucoProvavel']))
185
186 ### md
187 Controlador Fuzzy
188 ###
189 noteira_ctrl = ctrl.ControlSystem(rules)
190 noteira_sim = ctrl.ControlSystemSimulation(noteira_ctrl)
191
192 ### md
193 Exemplo de como submeter dados ao sistema Fuzzy e visualizar o grau de
pertinência das variáveis de entrada e de saída
194 ###
195 # Exemplo: submeter dados ao simulador
196 noteira_sim.input['idade'] = 4
197 noteira_sim.input['saidaXentrada'] = 0.168
198 noteira_sim.input['saidaXarrecadacao'] = 0.592
199
200 # Computando o resultado e calculando a saída
201 noteira_sim.compute()
202 print(noteira_sim.output['GrauSonegadora'])
203
204 ###
205 idade.view(sim=noteira_sim)
206
207 ###
208 saidaXentrada.view(sim=noteira_sim)
209
210 ###
211 saidaXarrecadacao.view(sim=noteira_sim)
212
213 ###
214 noteira.view(sim=noteira_sim)
215
216 ### md
217 Carregando a massa de dados
218 ###
219
220 ###
221 #dfs = dataframe empresas suspeitas
222 #dfc = dataframe empresas confirmadas
223
224 ###
225 dfs = pd.read_csv('noteiras_fuzzy_20220118.csv', sep=';',
encoding='utf-8', quotechar='', decimal=',')
226
227 dfs.insert(0, 'ID', dfs['NU_CGC_CPF'].apply(lambda x :
str(hashlib.sha256(str(x).encode('UTF-8')).hexdigest())))
228
229 dfs.drop(['NU_CGC_CPF', 'DT_INICIO_ANALISE', 'DT_FIM_ANALISE',
'DT_INICIO_EMP', 'TOTAL_NF_SAIDA', 'TOTAL_ARRECADACAO'], inplace=True,
axis=1)
230 dfs = dfs.dropna()

```

```

231 dfs.to_csv('empresas_suspeitas_hashid.csv', sep=';', encoding='utf-8',
quotechar='"', decimal=',', index=False)
232
233 dfc = pd.read_csv('Macarico.csv', sep=';', encoding='utf-8',
quotechar='"', decimal=',')
234 dfc.insert(0, 'ID', dfc['NU_CNPJ_CPF'].apply(lambda x :
str(hashlib.sha256(str(x).encode('UTF-8')).hexdigest()))))
235 dfc = dfc[['ID', 'SITUACAO']]
236 dfc.to_csv('empresas_confirmadas_hashid.csv', sep=';', encoding='utf-
8', quotechar='"', decimal='.', index=False)
237
238 ### md
239
240 ###
241
242
243 #
244 ###
245 #
246 #
247 #
248 #
249 df['NEW_FUZZY_NUMBER'] = df.apply(lambda r: 0, axis=1)
250 df['NEW_FUZZY_LABEL'] = df.apply(lambda r: "", axis=1)
251
252 #df.dtypes
253 print(dfs.dtypes)
254 print("=====")
255 print(dfc.dtypes)
256 ###
257 #Das empresas avaliadas como 'Provável' e 'Muito Provável' quais foram
confirmadas como fraudadoras
258 empresas_suspeitas_confirmadas = dfs[dfs['ID'].isin(dfc['ID'].values)
& dfs['FUZZYLABEL'].isin(['Provável', 'Muito Provável'])]
259 empresas_suspeitas_confirmadas['FUZZYLABEL'].value_counts()
260 #141 empresas classificadas como 'Muito Provável' estão na lista de
fraudadoras confirmadas
261 #4 empresas classificadas como 'Provável' estão na lista de
fraudadoras confirmadas
262 ###
263 #Das empresas avaliadas como 'Pouco Provável' quais foram confirmadas
como fraudadoras
264 empresas_nao_suspeitas_confirmadas =
dfs[dfs['ID'].isin(dfc['ID'].values) & dfs['FUZZYLABEL'].isin(['Pouco
Provável'])]
265 empresas_nao_suspeitas_confirmadas['FUZZYLABEL'].value_counts()
266 #Nenhuma empresa classificada como 'Pouco Provável' estava na lista de
empresas fraudadoras
267 ###
268 #Das empresas avaliadas como 'Provável' e 'Muito Provável', quais
ainda não foram confirmadas como fraudadoras e devem ser investigadas
269 empresas_suspeitas = dfs[~dfs['ID'].isin(dfc['ID'].values) &
dfs['FUZZYLABEL'].isin(['Provável', 'Muito Provável'])]
270 empresas_suspeitas['FUZZYLABEL'].value_counts()
271 #12.912 empresas classificadas como 'Muito Provável' são suspeitas e
precisam ser investigadas com 'alta prioridade'
272 #5.927 empresas classificadas como 'Provável' são suspeitas e precisam
ser investigadas com 'média prioridade'
273 ###
274 ###
275 for i in dfs.index:

```

```

292     noteira_sim.input['idade'] = dfs.at[i, 'IDADE']
293     noteira_sim.input['saidaXentrada'] = dfs.at[i, 'SXE']
294     noteira_sim.input['saidaXarrecadacao'] = dfs.at[i, 'SXA']
295     noteira_sim.compute()
296     crisp_value = noteira_sim.output['GrauSonegadora']
297     ling_value = 0
298     ling_label = ""
299     if fuzz.interp_membership(noteira.universe,
noteira['PoucoProvavel'].mf, crisp_value) > 0:
300         ling_value = fuzz.interp_membership(noteira.universe,
noteira['PoucoProvavel'].mf, crisp_value)
301         ling_label = 'Pouco Provável'
302     if fuzz.interp_membership(noteira.universe, noteira['Provavel'].mf,
crisp_value) > 0:
303         ling_value = fuzz.interp_membership(noteira.universe,
noteira['Provavel'].mf, crisp_value)
304         ling_label = 'Provável'
305     if fuzz.interp_membership(noteira.universe,
noteira['MuitoProvavel'].mf, crisp_value) > 0:
306         ling_value = fuzz.interp_membership(noteira.universe,
noteira['MuitoProvavel'].mf, crisp_value)
307         ling_label = 'Muito Provável'
308     #print(row['id'], ling_label, ling_value)
309     dfs.at[i, 'NEW_FUZZYNUMBER'] = ling_value
310     dfs.at[i, 'NEW_FUZZYLABEL'] = ling_label
311     ###
312     dfs.tail()
313     ###
314     #df.to_csv('/content/drive/MyDrive/Colab
Notebooks/Fuzzy/noteiras_fuzzy.csv', index=False, sep=';', encoding='utf-
8', quotechar='"', decimal='.')
315     ###
316     fuzzy_results = df['NEW_FUZZY_LABEL'].value_counts()
317     print(fuzzy_results)
318     #type(fuzzy_results)
319     ###
320     fuzzy_results = df['FUZZYLABEL'].value_counts()
321     print(fuzzy_results)

```

APÊNDICE B – Script Python da rede neural para a classificação de empresas noteiras

```
1     ### md
2     Instalando dependências
3     ###
4     # !pip install numpy
5     # !pip install scikit-learn
6     ### md
7     Importando as dependências
8     ###
9     import time
10    import matplotlib.pyplot as plt
11    import pandas as pd
12    ### md
13    # Passo 1: Carregando a massa de dados
14    ###
15    url = 'Noteiras_NaoNoteiras_202103a202203_str_id.csv'
16    df_origem = pd.read_csv(url, sep=';', encoding='utf-8', quotechar='\"',
17    decimal=',', low_memory=True)
17    cols_x = ['VL_TOTAL_NF_DEST_E', 'VL_TOTAL_NF_DEST_S',
18    'VL_ARRECADACAO', 'QTD_NFE_S', 'QTD_NFE_E',
19    'QTD_NOTAS_DEST_S', 'QTD_NOTAS_DEST_E', 'MAIOR_VL_NFE_E',
20    'MAIOR_VL_NFE_S',
21    'MAIOR_VL_NF_DEST_E', 'MAIOR_VL_NF_DEST_S',
22    'VL_TOTAL_NFE_E', 'VL_TOTAL_NFE_S',
23    'VL_TOTAL_ICMS_E', 'VL_TOTAL_ICMS_S', 'VL_CONTABIL_ENTRADA',
24    'VL_ICMS_ENTRADA',
25    'VL_ST_ENTRADA', 'QTD_VEICULO', 'VL_TOTAL_VEICULO']
26    col_y = ['NOTEIRA']
27    #df_origem = df_origem.drop(df_origem.columns[[0]], axis=1)
28    #df_origem.to_csv(url, sep=';', encoding='utf-8', quotechar='\"',
29    decimal=',', index=False)
30    cols_x = ['IDADE'] + cols_x
31    ###
32    len(df_origem)
33    ###
34    df_origem.info()
35    ###
36    df_origem.describe()
37    ###
38    ncount = df_origem[['NOTEIRA']].value_counts()
39    print(ncount)
40    ###
41    vcount = df_origem[['ID']].value_counts()
42    for i in range(len(df_origem.index)):
43        df_origem.at[i, 'IDADE'] = vcount[df_origem.at[i, 'ID']]
44    ### md
45    # Passo 2: Padronizando os dados
46    ###
47    from sklearn.preprocessing import StandardScaler
48    def do_scale_columns(df, columns):
49
50        for col in columns:
51            scaler = StandardScaler().fit(df[[col]])
52            df[col] = scaler.transform(df[[col]])
53
54        return df
55    ###
```

```

51 df_origem_padronizado = df_origem.copy(deep=True)
52 df_origem_padronizado =
do_scale_columns(df_origem_padronizado, columns=cols_x)
53 ### md
54 # Passo 3: Correlação, selecionando as variáveis mais relevantes
55 ###
56 corr_table =
df_origem_padronizado[cols_x].corrwith(df_origem_padronizado['NOTEIRA'])
57 corr_table
58 ###
59 series = abs(corr_table)
60 series
61 ###
62 def select_best_columns(series_in, max_in):
63
64     result = []
65     s_in = abs(series_in)
66     s_in.sort_values(ascending=False, inplace=True)
67
68     for idx, vl in s_in.items():
69         if len(result) < max_in:
70             result.append(idx)
71
72     return result
73 ###
74 best_cols = select_best_columns(series, 11)
75 best_cols
76 ### md
77 # Passo 3: Correlação
78 ###
79
df_origem_padronizado[best_cols].corrwith(df_origem_padronizado['NOTEIRA'])
.plot.bar(
80     figsize=(20,6),
81     title="Correlação das Variáveis de Entrada com a Variável de Saída
'Noteira'",
82     fontsize=15,
83     rot=80,
84     grid=True)
85
86 ###
87 from sklearn.model_selection import train_test_split
88 from sklearn import metrics
89 from sklearn.neural_network import MLPClassifier
90 from sklearn.ensemble import RandomForestClassifier
91 from sklearn import svm
92 from sklearn.model_selection import GridSearchCV
93 import numpy as np
94 ###
95 X = df_origem_padronizado.loc[:, best_cols].values
96 y = df_origem_padronizado.loc[:, col_y].values.ravel()
97 ### md
98 # Passo 4: Preparação dos dados para o treinamento (SMOTE)
99 ### md
100 SMOTE
101
102 Fatiamento dos dados para treinamento e teste
103 70% -> treinamento
104 30% -> teste
105
106 ###

```

```

107 from imblearn.over_sampling import SMOTE
108
109 Xp = df_origem_padronizado.loc[:, best_cols].values
110 yp = df_origem_padronizado.loc[:, col_y].values.ravel()
111 Xp_train, Xp_test, yp_train, yp_test = train_test_split(Xp, yp,
train_size=0.7, test_size=0.3, shuffle=True)
112
113 smote_balance = SMOTE(random_state=100)
114 X_smote, y_smote = smote_balance.fit_resample(Xp, yp)
115
116 Xp_train_smote, Xp_test_smote, yp_train_smote, yp_test_smote =
train_test_split(X_smote, y_smote, train_size=0.7, test_size=0.3,
shuffle=True)
117
118 print(len(Xp))
119 print(len(yp))
120 print(len(X_smote))
121 print(len(y_smote))
122 print(pd.Series(y_smote).value_counts())
123 ### md
124 # Passo 5: GridSearchCV - Busca da melhor arquitetura de rede
125 ###
126 model_params_smote = {
127     'mlp_classifier' : {
128         'model': MLPClassifier(max_iter=5000, alpha=0.0001,
solver='adam', verbose=True),
129         'params': {
130             'hidden_layer_sizes': [(len(best_cols),
int(len(best_cols)*2/3)),
131                                     (len(best_cols)*2,
len(best_cols)),
132                                     (len(best_cols)*3,
len(best_cols)*2)],
133             'activation': ['identity',
134                             'logistic',
135                             'tanh',
136                             'relu'],
137             'learning_rate': ['adaptive',
138                               'invscaling',
139                               'constant'],
140                             ],
141             'batch_size': [150,
142                             250,
143                             350]
144         }
145     }
146 }
147
148 scores_smote = []
149
150 for model_name, model_params in model_params_smote.items():
151     clf = GridSearchCV(model_params['model'], model_params['params'],
return_train_score=False, verbose=10, n_jobs=3)
152     print("Started model: " + model_name)
153     tic = time.perf_counter()
154     clf.fit(X_smote, y_smote)
155     print("Finished model: " + model_name)
156     toc = time.perf_counter()
157     print(f"Time: {toc - tic:0.4f} seconds")
158     scores_smote.append({
159         'model': model_name,

```



```

160         'best_score': clf.best_score_,
161         'best_params': clf.best_params_
162     })
163
164     df_smote =
165     pd.DataFrame(scores, columns=['model', 'best_score', 'best_params'])
166     scores_smote
167     """ md
168     # Passo 6: Treinando o modelo selecionado
169     """
170     c_smote = MLPClassifier(max_iter=10000, activation='relu',
171                             learning_rate='constant',
172                             batch_size=250, verbose=True,
173                             n_iter_no_change=400,
174                             hidden_layer_sizes=(len(best_cols)*3,
175 int(len(best_cols)*2)),
176                             alpha=0.00001)
177     tic = time.perf_counter()
178     c_smote.fit(Xp_train_smote, yp_train_smote)
179     toc = time.perf_counter()
180     print(f"Time: {toc - tic:0.4f} seconds")
181     """
182     from sklearn.metrics import classification_report
183
184     yp_pred_smote = c_smote.predict(Xp_test_smote)
185     acp = metrics.accuracy_score(yp_test_smote, yp_pred_smote)
186
187     print("Accuracy: ", acp)
188     print(classification_report(yp_test_smote, yp_pred_smote))
189     rsc = metrics.roc_auc_score(yp_test_smote, yp_pred_smote)
190     print(rsc)
191
192     """
193     # from joblib import dump
194     #
195     # dump(c_smote, 'mlp_smote.joblib')
196     """ md
197     # Passo 7: Métricas (matriz-confusão e curva ROC)
198     """
199     import seaborn as sns
200
201     cf_matrix_smote = metrics.confusion_matrix(yp_test_smote,
202     yp_pred_smote)
203     # # Print the confusion matrix using Matplotlib
204     group_names_smote = ['Verdadeiro Negativo', 'Falso Positivo', 'Falso
205     Negativo', 'Verdadeiro Positivo']
206     group_counts_smote = ["{0:0.0f}".format(value) for value in
207     cf_matrix_smote.flatten()]
208     group_percentages_smote = ["{0:.2%}".format(value) for value in
209     cf_matrix_smote.flatten()/np.sum(cf_matrix_smote)]
210     labels_smote = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in
211     zip(group_names_smote, group_counts_smote, group_percentages_smote)]
212     labels_smote = np.asarray(labels_smote).reshape(2, 2)
213     sns.heatmap(cf_matrix_smote, annot=labels_smote, fmt='', cmap='Blues')
214     """
215     len(Xp_test_smote)
216     """
217     from sklearn.metrics import RocCurveDisplay
218     """

```

```
215 fpr_smote, tpr_smote, _ = metrics.roc_curve(yp_test_smote,
yp_pred_smote)
216 auc_smote = metrics.auc(fpr_smote, tpr_smote)
217 #display = RocCurveDisplay(fpr=fpr_smote, tpr=tpr_smote,
roc_auc=auc_smote)
218 #display.plot()
219 RocCurveDisplay.from_estimator(c_smote, Xp_test_smote, yp_test_smote)
220 plt.show()
221 %%%
```