Revista

FAC INFO

2020 | Vol 3

FACULDADE DE TECNOLOGIA E INOVAÇÃO

Senac

Faculdade de Tecnologia e Inovação Senac-DF
Revista Eletrônica

FAC INFO

ISSN: 2236-9058

Vol. 3, No. 1, 2º semestre/ Ano 2020

Equipe técnica:

Diretor Faculdade de Tecnologia e Inovação Senac-DF:
Prof. Dr. Luís Afonso Bermúdez

Comitê Editorial:

- Demóstenes Jonatas de Azevedo Júnior - Mestre;
- Denise Maria dos Santos Paulinelli Raposo- Mestre;
- Eliane Ferreira – Doutora;
- Janaína Leonardo Garcia – Mestre;
- Marcelo Persegona – Doutor;
- Rogério Lopes – Mestre.

Editora:
Janaína Leonardo Garcia

Revisão Gramatical:
Eliane Ferreira

Diagramação:
Janaína Leonardo Garcia

Capa:
Assessoria de Comunicação (ACM). SENAC-DF

Sumário

EDITORIAL:

É com grande alegria que apresentamos o terceiro volume da Revista Eletrônica FACINFO da Faculdade de Tecnologia e Inovação Senac do Distrito Federal, com periodicidade de publicação semestral, é destinada a publicar produções acadêmicas produzidas pelo corpo discente e docente desta instituição, com temáticas acerca da Tecnologia da Informação.

Apresentam-se neste terceiro número, dois artigos de discente da pós-graduação e dos docentes desta IES, com a intenção de disseminarmos o conhecimento nesta área que tanto vem crescendo em nosso país. Buscamos também, tornar públicas as informações e dados discutidos nos nossos diversos cursos de pós-graduação em de 2020. O ano em que o mundo mudou por conta da Pandemia de SARS Covid-19 e os seres humanos aprenderam a viver de maneira diferente do habitual. A educação superior contou com grandes adaptações nas formas de ensino-aprendizagem e desta forma, estamos extremamente contentes de mesmo nesse cenário repleto de dificuldades, ter a oportunidade de construirmos, em parceria com nossos alunos e professores essa revista.

Acreditamos que estes trabalhos serão capazes de cativar os leitores desta edição e que serão capazes de criar expectativa para as próximas publicações.

Boa Leitura!

APRESENTAÇÃO:

A revista Fac Info da Faculdade de Tecnologia e Inovação Senac- DF tem como missão publicar artigos, e resultados de pesquisas desenvolvidas por seus professores e alunos. Assim, pretende contribuir para o estímulo à pesquisa e para divulgação e geração do conhecimento na área de tecnologia da informação. . Os contatos com o corpo editorial da revista Fac Info da Faculdade de Tecnologia e Inovação Senac-DF podem ser feitos pelo e-mail demostenes.azevedo@df.senac.com.br.

Aproveite nosso conteúdo!

ARTIGOS:

1) RECONHECIMENTO FACIAL PARA IDENTIFICAÇÃO NO ACESSO AO DATA CENTER

Ilton do Nascimento Castro
Marcos Antônio Rodrigues da Silva Júnior

RESUMO

O artigo aborda o estudo e criação de um algoritmo de reconhecimento facial usando o método *Fisherface*, que tem como objetivo principal analisar imagens faciais de uma base de dados, utilizando uma câmera (webcam) e dessa forma retornar os dados de identificação de pessoas, quando estas estiverem cadastradas na base. As etapas de criação do algoritmo consistem em: detectar faces, coletar as imagens, treinar e reconhecer. Em sua essência, este algoritmo fará extração de características essenciais e peculiares de cada uma das imagens coletadas, através de técnicas de reconhecimento facial, trazendo uma maior efetividade para identificação de faces. Devido ao cenário atual de pandemia (COVID-19), esta forma de biometria ajudará a evitar o contato com equipamentos de acesso, que não estejam devidamente higienizados em um ambiente Data Center e, consequentemente, trazendo mais segurança para identificar intrusos em áreas que exigem autenticação.

Palavras-chave: 1. Reconhecimento Facial; 2. Acesso; 3. Biometria; 4. Segurança Cibernética; 5. Detecção facial; 6. Centro de Dados; 7. COVID-19.

ABSTRACT

The article discusses the study and creation of a facial recognition algorithm using the Fisherface method, which has as main objective, to analyze facial images from a database, using a camera (webcam) and in this way return the identification data of people, when they are registered in the database. The steps to create the algorithm consist of: detecting faces, collecting images, training and recognizing. In essence, this algorithm will extract essential and peculiar characteristics from each of the collected images, through facial recognition techniques, bringing greater effectiveness to face identification. Due to our current pandemic scenario (COVID-19), this form of biometrics will help to avoid contact with access equipment, which is not properly sanitized in a Data Center environment and consequently bringing more security when identifying intruders in areas that require authentication.

Keywords: 1. Facial recognition; 2. Access; 3. Biometrics; 4. Cybersecurity; 5. Facial detection; 6. Data Center; 7. COVID-19.

Introdução

As buscas pela segurança de dados, principalmente em ambientes destinados ao armazenamento de recursos tecnológicos, trouxeram um crescimento contínuo, voltado às tecnologias que fazem identificação de pessoas através da face. Tecnologias que permitem associar imagens a pessoas, marcar regiões especiais do corpo e relacionar dados a um contexto específico em uma base dos dados, estão acessíveis a preços muito altos no mercado (KUROIWA; CARRO, 2015, p.2).

O reconhecimento facial é uma técnica de biometria baseada em traços do rosto humano. Esse processo é realizado através de ligações algorítmicas de traços, tamanhos e formas, podendo ser citado, como exemplo, uma distância exata entre partes do rosto, tamanho da arcada dentária, distância entre olho e boca, tamanho do crânio, entre outros detalhes (OKABE; CARRO, 2014, p.2).

O reconhecimento facial é uma tecnologia que pode ser utilizada em um sistema de biometria capaz de ser usada para fins de acesso, o que é essencial para as questões de segurança nos dias atuais. O fato de essa tecnologia ser empregada em locais públicos e privados possibilita a facilidade de minimizar limitações jurídicas e comprovar acessos de pessoas em horários distintos (OKABE; CARRO, 2014, p.2).

O objetivo deste estudo é enfatizar e demostrar a importância do uso de um *framework* específico, com funcionalidades de detecção de pessoas nos sistemas de acesso por biometria de reconhecimento facial, utilizando o método *Fisherface*, demonstrando que essa tecnologia pode ser aplicada para identificar acessos e evitar intrusões em um ambiente Data Center (KUROIWA; CARRO, 2015, p.2).

A Figura 1 abaixo representa o sistema de reconhecimento facial no Data Center.

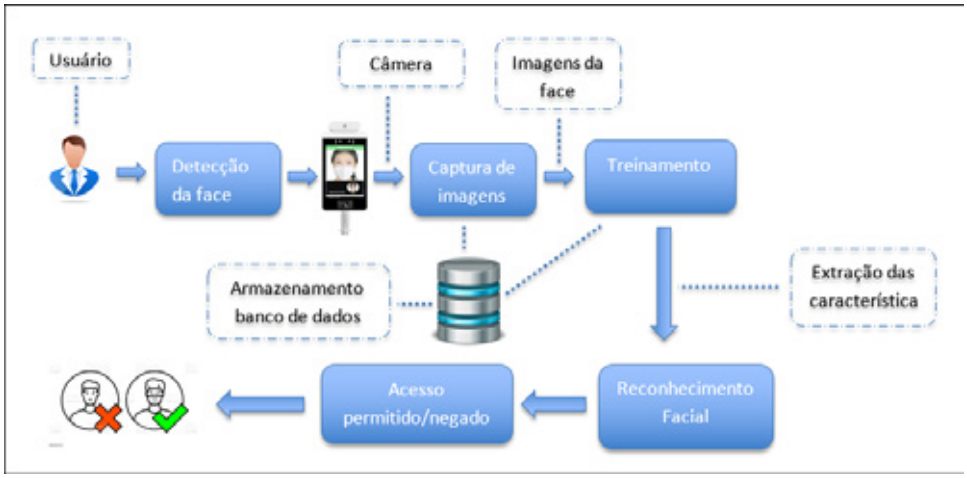


Figura 1 - Ilustração do sistema de reconhecimento facial no Data Center.

Este trabalho foi organizado em 6 seções. Na primeira será descrito a história sobre a tecnologia de reconhecimento facial, como surgiu, quais são os benefícios e malefícios dos sistemas que usam essa tecnologia nos dias atuais, quais são as áreas de aplicação dessa tecnologia. Na segunda, será feita uma descrição sobre o método *Fisherface* e sobre as técnicas básicas empregadas no método, a Análise Discriminante Linear (LDA) e *Principal Component Analysis* (PCA). Na terceira, serão

apresentadas as características dos componentes utilizados no estudo dessa tecnologia e quais são as funcionalidades de cada componente. Na quarta seção, será abordada a implementação do algoritmo de reconhecimento facial e todas as suas etapas de criação do início ao fim. Na quinta, serão mostradas quais foram os resultados positivos e negativos, após a execução do programa. E, por último, na sexta seção, apresentam-se as conclusões sobre o artigo.

1. Referencial teórico

O francês Alphonse Bertillon, em meados do século XIX, propôs um método que se baseava em um conceito de que as medidas do corpo de um ser humano adulto não mudavam com o tempo (BERTILLON, 1896, *apud* SILVA, 2015, p.20).

Na década de 1960, surgiu um sistema de reconhecimento facial semiautomático, onde, através de uma base de dados com imagens de 10 pessoas, era possível fazer uma análise de forma interativa, mostrando a foto de cada pessoa e, caso o sistema respondesse de forma negativa, era possível receber um retorno com o nome da pessoa a ser identificada. O sistema conseguiu usar matriz de pesos e medidas para melhorar sua precisão a cada apresentação de uma nova imagem. Através dessa interação, era possível aplicar um treinamento assistido para fins de teste. Após apresentação dessas imagens 250 vezes, o sistema conseguiu de forma assertiva a classificação de todas essas imagens (TAYLOR, 1967, *apud* SILVA, 2015, p.20).

Na década seguinte, foi desenvolvido um sistema bem mais evoluído. Esse sistema começou a utilizar 22 duas características para melhoria do reconhecimento, tais como abertura do olho e largura da boca, sombra dos olhos (GOLDSTEIN; HARMON; LESK, 1971, *apud* SILVA, 2015, p.20).

O intuito era tornar o resultado de reconhecimento aproximado ao feito por seres humanos, através da extração de características da imagem da base, por meio de uma ordem com características e limiares. A título de curiosidade, os experimentos feitos com 255 faces atingiram 70% de efetividade (SILVA, 2015, p.20).

O método *Fisherface*, com o objetivo de redução de dimensionalidade de imagens, aplica uma técnica chamada LDA. Esse método teve uma efetividade considerável em bases de dados que contêm imagens com problemas de variações de luminosidade (BELHUMEUR; HESPANHA; KRIEGMAN, 1997, *apud* SILVA, 2015, p.21).

> No âmbito da extração de características o método de PCA foi utilizado, por ter um sucesso em sistemas de reconhecimento facial, além de estar bem difundido no mercado, lembrando que também diminui a dimensionalidade e mantendo apenas as características de maior importância e descartando as de menor relevância. A técnica PCA gera um conjunto de imagens do domínio que é selecionado para treinamento, na etapa seguinte é criado um espaço de faces e posteriormente uma base de dados dos usuários é criada. (BISSI;2018, p.28).

1.1 Benefícios e malefícios dos sistemas que usam essa tecnologia

A procura pela segurança em ambientes onde são alocados recursos tecnológicos e dados em geral está em constante crescimento, por esse motivo surge a necessidade de gerenciamento de

identidades. Uma tecnologia importante e em constante crescimento é a biometria, especificamente o reconhecimento facial, que pode nos trazer vários benefícios, mas também, se não forem tomados os devidos cuidados, podem trazer prejuízos enormes.

Outro risco iminente neste ano de 2020 é de contágio com a doença COVID-19. A disseminação desse vírus acontece principalmente do contato com superfícies infectadas que não foram devidamente higienizadas.

Uma das ações que contribuem para a infecção da doença COVID-19 é tocar em olhos e nariz após o contato com alguma superfície infectada (MCINTOSH, 2020, p.3). Nesse ponto, pode-se dizer que o reconhecimento facial será de crucial importância, pois, ao apresentar a face em sistemas de acesso, será evitado o contato físico com possíveis locais por onde circulam o vírus.

Nos dias atuais, está-se lidando com uma situação atípica, que é a pandemia do COVID-19, em que um dos principais meios de contaminação está na má sanitização de equipamentos que possam ser contaminados. Então, nesse aspecto, o reconhecimento facial traz um grande benefício, pois não é preciso tocar em equipamentos que dependem de uma validação para identificar o acesso em alguns lugares, como, por exemplo, Data Centers.

Outro benefício está no reconhecimento automático de pessoas que trabalham em um ambiente corporativo com muito fluxo de funcionários, pois pode facilitar de forma mais rápida a liberação, registro e gestão dos mesmos. Para fins segurança do ambiente, essa tecnologia é essencial, pois o contato físico pode ser substituído em alguns aspectos a fim de evitar possíveis ameaças.

O reconhecimento facial já é utilizado em dispositivos móveis, como, por exemplo: em smartphones para desbloqueio de tela, assim como plataforma de aplicativos de motorista de carro, que também já utilizam a tecnologia.

A tecnologia de reconhecimento facial evita fraudes e traz facilidade nos processos de integração com outros sistemas, visto que os seres humanos possuem características peculiares e, quando os dados são compartilhados de forma segura, observando todos os quesitos legais, tornam-se fortes aliados no processo de integração com o ambiente tecnológico.

Assim como outras tecnologias, tem-se também problemas em relação ao reconhecimento facial. Hoje, com a evolução de aplicativos e dispositivos de dados, pode-se ter, de uma forma não-autorizada, o compartilhamento de informações, ou seja, roubo de dados por pessoas mal-intencionadas, que tem o objetivo de criar contas falsas, efetuar cadastros em sites de compras, denegrir a própria imagem de pessoas na rede mundial de computadores e também o uso impreciso e discriminatório, o que pode ser um malefício, pois oferece o risco de pessoas inocentes serem confundidas com criminosos em qualquer tipo de ambiente, causando um dano moral e psicológico.

O ambiente físico de uma organização que possui dados importantes pode ser alvo de criminosos, que colocam em risco toda a infraestrutura de armazenamento desses dados. Outro problema que pode ser encontrado nessa tecnologia são riscos à privacidade, devido ao envolvimento de dados pessoais utilizados para seu funcionamento, o direito fundamental de um cidadão pode ficar comprometido.

1.2 Áreas de aplicação dessa tecnologia

As áreas de aplicação dessa tecnologia são diversas. O foco deste trabalho está na ideia de usar essa tecnologia para identificação de funcionários, em equipamentos de autorização de um Data Center. A aplicação dessa tecnologia está empregada há tempos em vários setores do dia a dia, como, por exemplo, empresas de valores que guardam bens valiosos de outras organizações, escritórios corporativos que devem guardar seus dados para que sejam acessados apenas por pessoas que realmente devem ter esse acesso, carteira nacional de habilitação do Distrito Federal para evitar fraudes, casas lotéricas, casas de câmbio, mercados, conveniências, eventos de grande magnitude, como exposições de tecnologia, são todos exemplos de áreas de aplicação.

A adesão da tecnologia de reconhecimento facial não está somente no ambiente onde estão os profissionais da área de tecnologia da informação e comunicações, ela expande-se em quaisquer áreas que exigem uma melhor identificação de pessoas para melhoria de registros e prevenção de incidentes de segurança.

2 Método *FisherFace*

O método escolhido foi o *Fisherface*, pois a quantidade no acerto e confiança do algoritmo mostrou-se mais eficiente, conforme dados apresentados na Tabela 2.

Tabela 1 - Eigenface

Acertos	Confiança	Parâmetros
73	5388,98	Padrão
60	2976,57	40, 8000

Tabela 2 - Fisherface

Acertos	Confiança	Parâmetros
80	1645,89	Padrão
76	466,2	3, 2000

Tabela 3 – LBPH

Acertos	Confiança	Parâmetros
66	10,69	Padrão
63	0,77	2,2,7,7,50

Utiliza-se no processo de reconhecimento facial um método chamado *Fisherface*, onde são extraídas imagens de características de cada indivíduo. Segundo Bissi (2018, p. 24), sobre as fisherfaces:

> Similar ao eigenfaces, as fisherfaces podem ser visualizadas como imagens de características onde, as características das fisherfaces são variações de aparência presentes nas imagens de cada indivíduo, tais como variações de luminosidade, poses e expressões faciais. Assim como as imagens no espaço de dados possuem um valor para cada atributo, os vetores e características possuem um valor para cada fisherfaces (p.24).

A Figura 1.1, a seguir, representa a projeção de faces relativa ao *Fisherface.*

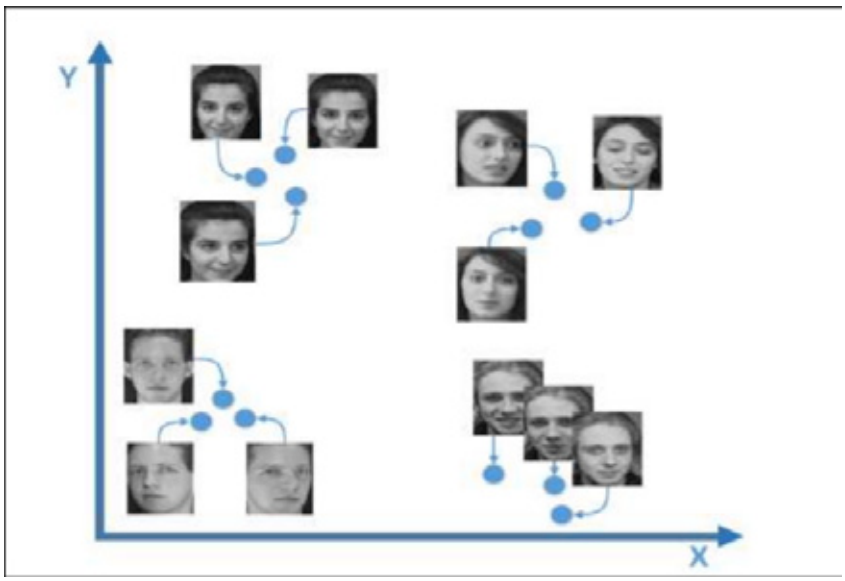


Figura 1.1 - Espaço de Faces Fisherfaces - (BISSI, 2018, *apud* SILVA, 2016, p.25)

A Figura 1.1 descreve perfeitamente o objetivo do Fisherface. De acordo com Bissi (2018, p. 25):

> Temos então que quanto mais diferente for uma pessoa da outra, mais distantes deverão estar suas projeções, e quanto mais parecidas forem as imagens de uma pessoa mais próximas deverão estar suas projeções (p.25).

Sobre o FisherFace, Bissi (2018, p.25) destaca que o algoritmo possui 8 etapas em sua combinação:

> Cálculo da face média por classe: soma de todos os pixels de uma imagem pertencente a mesma classe, dividido pelo total de imagens dessa classe. Cálculo de face média geral: soma de todos os pixels de uma imagem, dividido pelo total de imagens independente da classe. Transformação das imagens em vetores: concatenação de pixels linha a linha, formato um vetor de pixels. Construção da matriz de dispersão intra-classe: quanto as imagens de um indivíduo diferem uma das outras. Construção da matriz de dispersão inter-classe: quanto as imagens de indivíduos distintos diferem umas das outras. Cálculo das fisherfaces: são as fisherfaces do LDA. Cálculo dos vetores de características: vetores de imagens formados pelas características de média de cada fisherface. Cálculo da similaridade: uso de um classificador para pontuar a similaridade entre as imagens, exemplo utilizado, distâncias entre características de uma foto e outra (p.25).

2.1 Técnicas básicas empregadas no método Fisherface

Técnicas biométricas são empregadas na tecnologia de reconhecimento facial, as mesmas consistem na identificação de padrões faciais, tais como: formato da boca, rosto, distância entre olhos, boca, nariz, entre outros (BISSI, 2018*, apud* SILVA; CINTRA, 2015, p.11).

Apesar do *Fisherface* utilizar com maior amplitude a técnica LDA, também será comentado sobre a técnica PCA, que também pode ser aplicada no algoritmo.

2.2 PCA - Principal Component Analysis

Técnica amplamente utilizada no tratamento de características, foi o primeiro método que teve êxito em sua aplicação no processo de reconhecimento facial. Ainda é utilizado amplamente nos dias atuais para comparação de novas propostas na área específica do estudo de reconhecimento facial (BISSI, 2018, *apud* PENTEADO, 2009, p.21).

Referente a PCA, Bissi (2018, *apud* Penharbel, 2005, p.21) destaca que:

> A PCA é uma técnica matemática que descreve um conjunto de dados usando as principais componentes que representam da melhor maneira o conjunto de dados, usados de maneira a reduzir a dimensionalidade dos dados ou então detecção de padrões (p.21).

A organização de um algoritmo de reconhecimento facial precisa levar em consideração técnicas de extração de dados principais da face, com o objetivo de aumentar a efetividade do sistema, evitando, assim, falhas e possíveis problemas de administração da ferramenta que emprega essa tecnologia.

2.3 LDA - Linear Discriminant Analysis

A LDA é outra técnica utilizada no tratamento de dados de imagens no processo de reconhecimento facial. Bissi (2018, pag.22) destaca que:

> Com mesmo objetivo da técnica PCA, temos que o LDA consiste em reduzir a dimensionalidade dos dados visando a classificação. A partir de um conjunto de dados multidimensionais rotulados, o LDA gera um conjunto de dados de menor dimensionalidade que representa as classes dos dados originais (p.22).

Essa técnica é a mais utilizada no algoritmo do *Fisherface*, pois a preocupação com a melhoria na tradução de particularidades entre imagens está amplamente difundida nessa técnica, que foi basicamente uma melhoria do seu antecessor *Eigenfaces*.

O *Fisherface* utiliza a técnica LDA para aumentar de forma bastante significativa a confiabilidade entre as imagens cadastradas na base de dados para o treinamento do algoritmo, justamente pelo foco em reduzir dimensionalidade dos dados, ou seja, quanto maior for a similaridade entre fotos, maior a probabilidade de acerto nas características faciais.

3. Metodologia

O foco deste estudo é apresentar um algoritmo que utiliza o método *Fisherface* para chegar ao objetivo final, que é reconhecer uma face e imprimir alguns dados de cadastro. Abaixo, estão descritos os componentes e funcionalidades objetivas de cada um para melhor compreensão do estudo. São eles:

1. Pycharm: é uma IDE (ambiente de desenvolvimento) utilizado especificadamente para linguagem python, nele será desenvolvido a parte de código para o programa;

2. Linguagem Python: é uma linguagem de programação de alto nível, utilizada para escrita do código do programa de reconhecimento facial;

3. Biblioteca OpenCV: é uma biblioteca de programação com funções computacionais, que servirá de auxílio para criar o algoritmo;

4. *Fisherface*: é o método já mencionado para facilitar a implementação do programa, que será utilizado junto com técnica LDA e scripts ".yml", com o objetivo de treinar o algoritmo;

5. Base de dados de imagens: é uma base de dados local que será preenchida com imagens dos dois integrantes do grupo, elas serão utilizadas para treinar o algoritmo para reconhecer as faces;

6. Laptop com uma webcam integrada e outra separada: utilizado com um laptop para demonstração prática do algoritmo de reconhecimento facial em funcionamento.

4. Implementação do algoritmo

O trabalho inicia neste momento com a parte prática. As etapas de reconhecimento facial são divididas especificadamente em 4: detecção da face, coleta das imagens, treinamento do algoritmo e, por fim, reconhecimento. Cada uma dessas etapas é de extrema importância para que sejam alcançadas todas as expectativas deste estudo.

Abaixo, será feita a descrição de cada etapa de forma mais detalhada para facilitar o entendimento do processo, inclusive do código implementado com toda sua lógica.

4.1 Detecção da face

Conforme o algoritmo apresentando na Figura 2, o mesmo foi projetado apenas para identificação da face, ou seja, o sistema detectou o rosto com um quadrado vermelho assim que o localizou, de acordo com a demonstração na Figura 3.

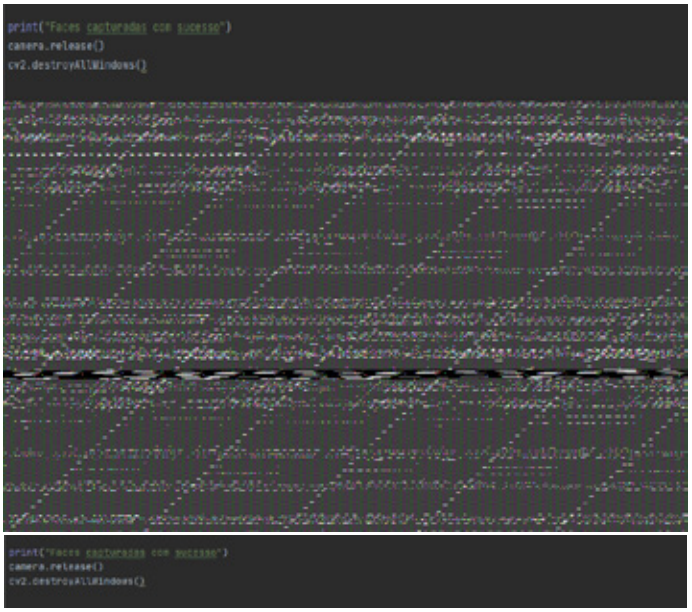


Figura 2 - Algoritmo de detecção das imagens.

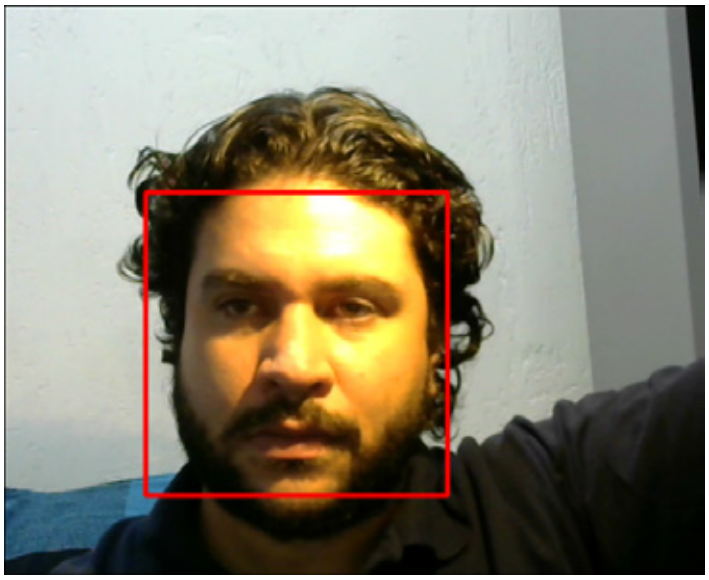


Figura 3 - Detecção de face.

4.2 Coleta das imagens

Nessa etapa é necessário digitar o identificador (ID), para que seja iniciada a captura das imagens pelo algoritmo (necessário apertar a letra “q” do teclado) e, consequentemente, a imagem será salva na base de dados local.

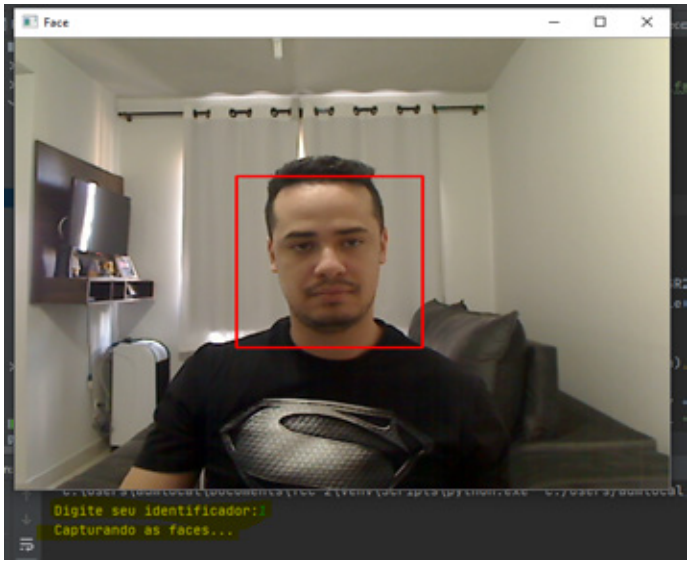


Figura 4 - Coleta de imagens.

4.3 Treinamento do algoritmo

O algoritmo descrito na Figura 6, que realiza o treinamento com as imagens cadastradas na base de dados, cria o arquivo chamado classificadorFisher.yml. Nesse arquivo são escritos os parâmetros do treinamento conforme Figura 5. O código completo de forma pública, com todas as etapas, está disponível em: https://github.com/rdpinformacao/CodigoReconhecimentoFacial.git.

```
%YAML:1.8
---
opencv_fisherfaces:
   threshold: 1.7976931348623157e+308
   num_components: 1
   mean: !!opencv-matrix
      rows: 1
      cols: 48400
      dt: d
      data: [ 1.8072000000000000e+02, 1.7858000000000001e+02,
          1.7546000000000001e+02, 1.7308000000000001e+02,
          1.7186000000000001e+02, 1.7220000000000002e+02,
          1.7078000000000000e+02, 1.6886000000000001e+02,
          1.6644000000000000e+02, 1.6347999999999999e+02,
          1.6024000000000001e+02, 1.5852000000000001e+02,
          1.5775999999999999e+02, 1.5581999999999999e+02,
          1.5288000000000000e+02, 1.4992000000000002e+02,
```

Figura 5 - Arquivo classificadorFisher.yml.

```python
import cv2
import os
import numpy as np


eigenface = cv2.face.EigenFaceRecognizer_create()
fisherface = cv2.face.FisherFaceRecognizer_create()
lbph = cv2.face.LBPHFaceRecognizer_create()

def getImagemComId():
    caminhos = [os.path.join('fotos', f) for f in os.listdir('fotos')]
    #print(caminhos)
    faces = []
    ids = []
    for caminhoImagem in caminhos:
        imagemFace = cv2.cvtColor(cv2.imread(caminhoImagem), cv2.COLOR_BGR2GRAY)
        id = int(os.path.split(caminhoImagem)[-1].split('.')[1])
        ids.append(id)
        faces.append(imagemFace)

        #cv2.imshow("Face", imagemFace)
        #cv2.waitKey(10)

    return np.array(ids), faces

ids, faces = getImagemComId()
#print(faces)
```

```python
print("Treinando...")
eigenface.train(faces, ids)
eigenface.write('classificadorEigen.yml')

fisherface.train(faces, ids)
fisherface.write('classificadorFisher.yml')

lbph.train(faces, ids)
lbph.write('classificadorLBPH.yml')

print("Treinamento realizado")
```

Figura 6 – Código treinamento do algoritmo.

Após execução do código, foi apresentada a seguinte mensagem: “Treinamento realizado”.

```
"C:\Users\acmlocal\Documents\TCC 2\venv\Scripts\python.exe" C:/Users/acmlocal/Documents/TCC/Treinamento.py/treinamento.py
Treinando...
Treinamento realizado

Process finished with exit code 0
```

Figura 7 - Finalização do treinamento.

4.4 Reconhecimento

Nessa etapa é realizado o reconhecimento facial. Os dados de cadastros foram exibidos, após a validação positiva do reconhecimento facial. É possível realizar ajustes no código de modo que o algoritmo diferencie as pessoas que possuem cadastro ou não. Segue abaixo a Figura 8 com o reconhecimento do Marcos já cadastrado na base do sistema, e a Figura 10, que mostra uma pessoa que não tem o cadastro na base de dados.

Figura 8 - Reconhecimento facial pessoa 1 cadastrada.

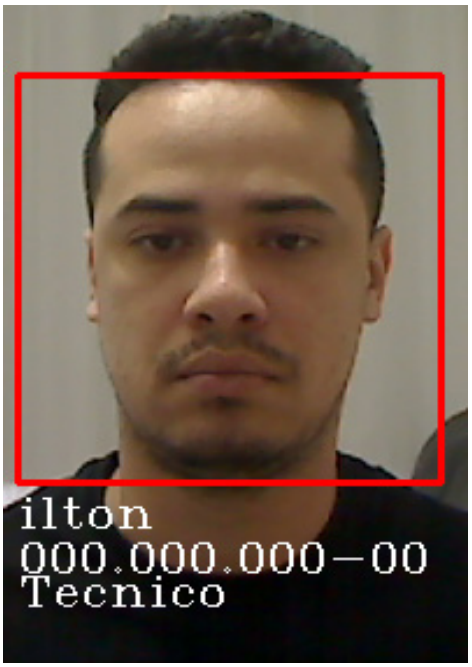


Figura 9 - Reconhecimento facial pessoa 2 cadastrada.

Figura 10 - Reconhecimento facial pessoa 3 sem cadastro.

5. Resultados

Foram realizados testes durante o treinamento do código utilizando o *Fisherface* e foi constado que, dependendo do ângulo e da claridade, o algoritmo não é tão eficaz, pois pode haver variações na detecção das faces que estão na base.

Conforme mostrado no reconhecimento facial das Figuras 11 e 12, houve uma variação na identificação devido à distância, câmera não ser a ideal para captura da foto com qualidade e a iluminação do ambiente não ser suficiente. Esses recursos podem causar problemas na eficiência do algoritmo.

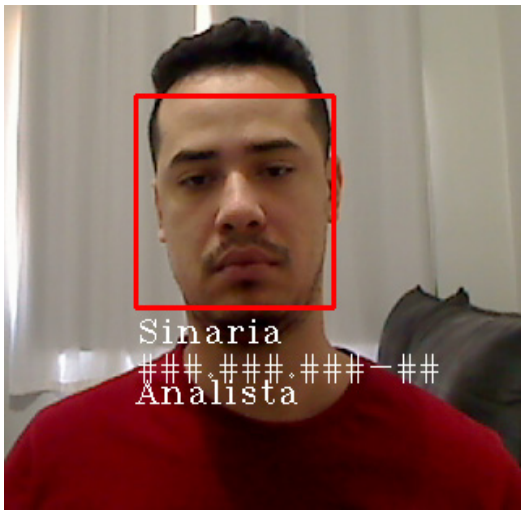


Figura 11 - Variação de precisão do reconhecimento.

Figura 12 - Pessoa não reconhecida na base.

De acordo com a Figura 12, mesmo o usuário cadastrado, o sistema não reconheceu a face do mesmo, pois a precisão do reconhecimento facial foi afetada, por conta da distância da câmera e baixa qualidade da webcam.

Na Figura 13, apresentada a seguir, foram realizados ajustes no código e nas capturas das imagens, as quais foram feitas através de uma webcam de maior qualidade, sendo possível obter o reconhecimento facial da pessoa já cadastrada na base de dados local, alcançando assim o resultado esperado.

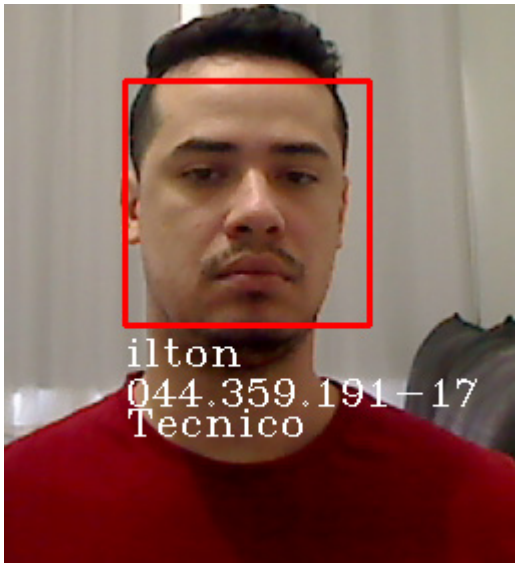


Figura 13 - Ajustes realizados no código.

Executados os testes cobrindo parte da face, identificaram-se falhas, pois o sistema não foi capaz de reconhecer a pessoa que já era cadastrada, conforme a Figura 14.

Figura 14 – Cobrindo a boca.

No exemplo a seguir, na Figura 15, cobriu-se a metade da face, porém, o sistema conseguiu realizar o reconhecimento com sucesso.

Figura 15 – Cobrindo a metade do rosto.

Na Figura 16, cobriu-se os olhos, o sistema apresentou falha e não conseguiu reconhecer a face.

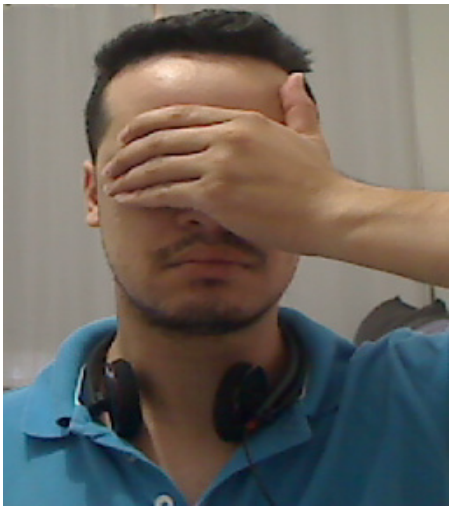


Figura 16 - Cobrindo os olhos.

Figura 17 - Cobrindo o nariz.

Foram realizados testes com pessoas já cadastradas na base do sistema e constatou-se que, mesmo elas utilizando máscaras, o algoritmo desenvolvido foi capaz de identificar a face, conforme demostrado nas Figuras 18 e 19.

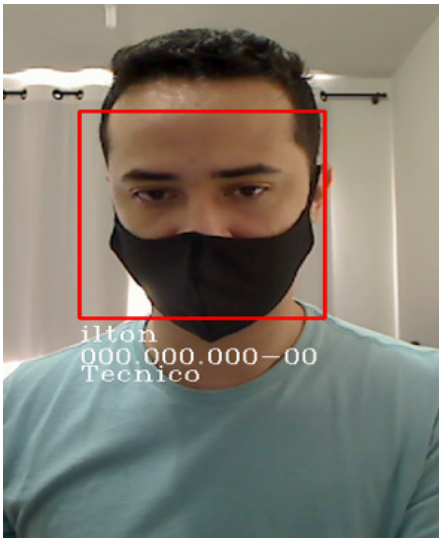


Figura 18 – Com máscara

Figura 19 - Com máscara

Tabela 4 - Webcam utilizada

Nome do componente	Fabricante	Modelo	Resolução máxima câmera
webcam 1	Logitech Europe S.A.	Logitech HD Webcam C270	720p/30qps (1 MP)
webcam 2	Logitech Europe S.A.	Webcam Pro 9000	1600 x 1200 (2 MP)
webcam 3 (integrada)	AsusTek Computer Inc.	Asus Vivobook S400CA	1.3 MP
webcam 4 (integrada)	Dell Technologies	Dell Latitude 7390	1280 x 720 (0,92 MP)

5.1 Resultados positivos

Agilidade e segurança para realizar o acesso ao Data Center, pois não será necessário fazer o reconhecimento da digital, sendo apenas necessário apresentar a face na câmera, onde será realizado o reconhecimento facial. No caso de cadastro positivo, o acesso será liberado ao ambiente, caso contrário, o mesmo será notificado que não tem permissão para acessar, e o registro será efetivado como tentativa de intrusão, para que seja analisado posteriormente, objetivando tomadas de decisões.

5.2 Resultados negativos

Ambiente com a iluminação ruim dificulta o reconhecimento da face e foto mal capturada pode ser um futuro problema para o reconhecimento. Outro fator importante que deve ser levado em consideração é a qualidade de resolução da câmera, pois quanto maior a resolução, mais são as chances de efetividade no reconhecimento facial; quanto menor a resolução, provavelmente as chances de efetividade serão piores, pois nos testes foram identificados que fotos, capturadas da webcam do laptop com a resolução inferior, impossibilitaram a identificação de algumas faces.

Observou-se que o sistema apresenta algumas falhas em determinados pontos. Um exemplo a ser citado é que, ao apresentar para a câmera a foto de uma pessoa já cadastrada na base de dados, existe a possibilidade de o sistema realizar a leitura dessa foto e liberar o acesso.

6. Considerações finais

Esse trabalho foi desenvolvido com base na LGPD, evitando compartilhamento de dados sensíveis de terceiros, com o objetivo de assegurar a proteção de dados.

A LGPD, Lei Geral de Proteção de Dados, atualmente em vigor no brasil, cita em seu Art. 4°, alguns casos em que não se aplica o tratamento de dados pessoais:

> Art. 4º Esta Lei não se aplica ao tratamento de dados pessoais:
> III - realizado para fins exclusivos de: a) segurança pública; b) defesa nacional; c) segurança do Estado; ou d) atividades de investigação e repressão de infrações penais (p.2).

A tecnologia de reconhecimento facial tem seu respaldo pela LGPD, em casos específicos, conforme citado acima, ou seja, não se pode, de forma aleatória, realizar testes com dados faciais de pessoas, visto que essas ações, em casos particulares e sem o consentimento dos usuários, podem gerar sérios problemas com a justiça.

A LGPD, Lei Geral de Proteção de Dados, cita também em seu Art. 11°, Seção II, do tratamento de dados pessoais sensíveis, sobre a:

> g) garantia da prevenção à fraude e à segurança do titular, nos processos de identificação e autenticação de cadastro em sistemas eletrônicos, resguardados os direitos mencionados no art. 9º desta Lei e exceto no caso de prevalecerem direitos e liberdades fundamentais do titular que exijam a proteção dos dados pessoais (p.6).

O cenário relatado acima informa claramente que, mesmo nos sistemas eletrônicos utilizados para autenticação, que é o caso do reconhecimento facial, por lei, estão garantidos a prevenção contra fraude e reforça que esses sistemas de autenticação devem ser seguros, permitindo a privacidade dos dados de pessoas cadastradas.

A importância dos sistemas de reconhecimento facial na segurança, é essencial para os dias atuais, onde é preciso agilidade na identificação de pessoas, para melhoria nos processos de segurança de informação, mais precisamente autenticidade. Garantir o registro de dados e horários de acesso é uma forma de avaliação de possíveis comportamentos fora do padrão, para que sejam feitas tomadas de decisões importantes em cada ambiente.

O estudo realizado sobre os métodos de reconhecimento facial foi suficiente para diagnosticar possíveis falhas que podem ser exploradas por pessoas mal-intencionadas, com o propósito de acessar espaços físicos de empresas e roubarem dados importantes, e assim utilizar como meio de extorsão ou como exposição da imagem da empresa, fazendo com que os danos sejam imensuráveis.

Foi identificado, durante o treinamento dos algoritmos *Fisherface*, que o seu poder de acerto e a sua confiabilidade é bem superior a outros métodos utilizados para essa mesma tecnologia. O *Fisherface* é um algoritmo que utiliza as questões de dimensionalidade de imagens para gerar sempre uma precisão melhor.

Neste trabalho, a criação do algoritmo foi feita com sucesso, em todas as suas etapas, do início ao fim, lembrando que é de extrema importância realizar todos os passos, desde a detecção de faces, coleta das imagens, treinamento do algoritmo até o reconhecimento das faces.

As principais vulnerabilidades encontradas foram: variação de precisão do reconhecimento; pessoa cadastrada na base, porém, o sistema não reconhece e foi realizado teste em que se cobriu parte da face para verificar a precisão do sistema, conforme ilustrado nas Figuras 14 a 17, onde ficou claro que o sistema é falho em alguns pontos.

Algumas vulnerabilidades encontradas no estudo podem ser evitadas com algumas das boas práticas citadas abaixo:

1. Utilizar o algoritmo de reconhecimento facial com outros recursos, para melhoria da efetividade, por exemplo, reconhecimento por voz, reconhecimento por íris, dentre outros;
2. Criar mecanismos que detectem a presença real de uma face, para evitar exposição de fotos pelos celulares ou tablets;
3. Para captura de imagens, utilizar câmeras com boa resolução no mínimo acima de 1 MP;
4. Manter uma distância mínima de 50 cm entre a face e o equipamento;
5. Exigir o consentimento digital das pessoas que serão cadastradas para utilizarem essa tecnologia;
6. Efetuar testes de precisão do algoritmo, a fim de evitar falsos positivos;
7. Melhoria contínua no código, utilizando as melhores práticas disponíveis no mercado de I.A.

O reconhecimento facial criado através de um algoritmo mostrou-se bastante eficaz, reconhecendo as faces colocadas à prova, imprimindo dados de cadastros, que podem ser diversos, como, CPF, RG, cargo. Algumas questões que foram mostradas devem ser analisadas com mais critério, como, por exemplo: devem ser capturadas fotos com qualidade, boa luminosidade, distância da câmera; expressões faciais devem ser registradas; falhas que podem ser exploradas utilizando imagens de pessoas que já estejam cadastradas na base de dados e assim comprometendo a segurança do ambiente.

Foram executados diversos testes e encontrada vulnerabilidade no sistema, pois é possível burlar o sistema, utilizando uma foto de uma pessoa já cadastrada na base, sendo ela impressa ou na tela de celular.

Durante o estudo, também foi realizado teste cobrindo parte da face, em que foi possível identificar que, se cobrir parte do rosto, o sistema ainda continua reconhecendo; porém, ao cobrir qualquer parte como boca, nariz ou olhos, o sistema não consegue identificar a face. Durante o teste, foi possível identificar que tem uma pequena variação na identificação, e isso acontece por conta de fotos tiradas com baixa qualidade, iluminação do ambiente, entre outros fatores, como a quantidade de fotos para realizar o treinamento, sendo que foi testado na primeira vez com 25 fotos e, posterior mente, com 50 fotos, em que foi visível uma melhora na detecção das faces e diminuição de variações e erros.

7. Referências

BISSi, Thelry David. Reconhecimento Facial com os algoritmos Eigenfaces e Fisherfaces, **Bitstream**. 2018. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/22158/3/ReconhecimentoFacialAlgotimos.pdf>. Acesso em: 25 out. 2020.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD).** 2018. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm>. Acesso em: 02 set. 2020.

DELL. Manual do proprietário do Dell Latitude 7390, **Dell**. 2020. Disponível em: <https://www.dell.com/support/manuals/br/pt/brbsdt1/latitude-13-7390-laptop/latitude_7390-om/camera-specifications?guid=guid-71e89bdf-d54c-478c-bb0e-0cee4bceace2&lang=en-us>. Acesso em: 03 out. 2020.

ICECAT. Logitech QuickCam Pro 9000 webcam 1600 x 1200 pixels USB Preto Prateado, **Icecat**. 2019. Disponível em: <https://icecat.biz/br/p/logitech/960-000053/webcams-quickcam+pro+9000-896740.html>. Acesso em: 02 out. 2020.

KITAMURA, Celso. Compilador (compiler). 2017. Disponível em: <https://celsokitamura.com.br/compilador/>. Acesso em: 18 jun. 2020.

KUROIWA, Bruno Tsutsumi; CARRO, Silvio Antonio. Detecção de intrusão com reconhecimento facial em imagens geradas por câmeras de segurança, **Revistas Unoeste**. 2015. Disponível em: <http://revistas.unoeste.br/index.php/ce/article/view/1424/1459>. Acesso em: 09 jun. 2020.

LEDA, Ana. Algoritmos e Lógica de Programação, **Slideshare Rodfernandes**. 2009. Disponível em: <https://pt.slideshare.net/rodfernandes/material-de-apoio-de-algoritmo-e-lgica-de-programao>. Acesso em: 13 jun. 2020.

LOGITECH. Hd-webcam-c270, **Logitech**. 2020. Disponível em: <https://www.logitech.com/pt-br/product/hd-webcam-c270>. Acesso em: 03 out. 2020.

MCINTOSH, Kenneth. Doença de Coronavírus 2019 (COVID-19), **Ebserh**. 2020. Disponível em: <http://www2.ebserh.gov.br/documents/1688403/5111980/4.pdf/49227786-d768-470e-9ea2-7e021aa96cc9>. Acesso em: 09 set. 2020.

OKABE, Rogerio Kazuhiro; CARRO, Silvio Antonio. Reconhecimento facial em imagens capturadas por câmeras digitais de rede, **Revistas Unoeste**. 2014. Disponível em: <http://revistas.unoeste.br/index.php/ce/article/view/1307/1425>. Acesso em: 11 jun. 2020.

PEREIRA, Ana Paula. O que é algoritmo, **Tecmundo**. 2009. Disponível em: <https://www.tecmundo.com.br/programacao/2082-o-que-e-algoritmo-.htm>. Acesso em: 15 jun. 2020.

SILVA, Ivson Soares José da. Reconhecimento facial em imagens de baixa resolução, **Bitstream**. 2015. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/16367/1/disserta%c3%a7%c3%a3o_jiss_ci%c3%aanciadacomputa%c3%a7%c3%a3o.pdf>. Acesso em: 20 jun. 2020.

ZAMBARDA, Pedro. Ultrabook-asus-vivobook-s400ca, **Techtudo**. 2013. Disponível em: <https://www.techtudo.com.br/review/ultrabook-asus-vivobook-s400ca.html>. Acesso em: 02 out. 2020.

ZENICOLA, Filipe Luiz Braga. Sistemas de Reconhecimento Facial, **Micti**. 2013. Disponível em: ≤http://eventos.ifc.edu.br/micti/wp-content/uploads/sites/5/2014/08/ESTUDO-SOBRE-METODOS-DE-RECONHECIMENTO-FACIAL-EM-FOTOGRAFIAS-DIGITAIS.pdf>. Acesso em: 10 ago. 2020.

2) Classificando ocorrências (chamados) em uma empresa de telefonia nacional: aplicando mineração de texto em chamados de TI.

Tássio da Silva Costa e Edilberto Silva

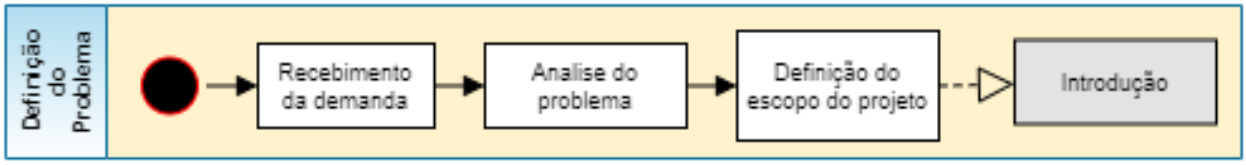
Resumo. Neste trabalho será feito um modelo de classificação de ocorrências de TI em uma grande empresa de telefonia, que gerencia as ocorrências e chamado no help desk embasada na ITIL. Tais ocorrências são divididas em dois grupos, os incidentes e as atividades recorrentes. Busca-se neste artigo estudar a aplicação de técnicas de Inteligência Artificial para classificar automaticamente tais ocorrências, pois há um elevado número de registros errôneos que deveriam ser incidentes, mas são registrados como Atividade Recorrente e vice-versa, causando enorme prejuízo para a gerência do Help Desk.

Palavras-chave: Help Desk; classificação automática; Incidente, Atividade Recorrente; IA**.**

Abstract.
In this work a model of classification of IT occurrences will be made in a large telephone company, which manages the occurrences and is called on the help desk based on ITIL. Such occurrences are divided into two groups, incidents and recurring activities. This article seeks to study the application of Artificial Intelligence techniques to automatically classify such occurrences, as there are a high number of erroneous records that should be incidents are registered as Recurring Activity and vice versa, causing enormous damage to the Help Desk management

1. INTRODUÇÃO

Figura 1: Fluxo de Atividades – Definição do Problema.



Fonte: Elaborado pelos autores (2020).

Este artigo tem como alvo um estudo de caso de uma grande empresa do ramo telefônico de abrangência nacional.

Esta empresa tem como pilar oferecer o melhor serviço e melhor atendimento ao seu cliente, seja por suporte telefônico, ou digital (Chatboot, internet ou APP).

Esse ramo é um dos mais competitivos do mercado, o nível de exigência dos clientes é muito alto e a empresa é constantemente avaliada pelo órgão regulador, Anatel, logo, o nível de serviço e excelência no trato com os clientes deve ser sempre o melhor. Essa entrega está em todas as diretorias e gerências da companhia, principalmente a diretoria de tecnologia, que tem toda a sua entrega focada na melhor experiência do seu cliente final.

Tendo total preocupação com a qualidade do serviço a empresa tem gerências que fazem auditoria interna, umas dessas é a gerência de TI, que audita, é claro, chamados da diretoria de TI para que se possa avaliar o nível de serviço e o tempo de resolução dos chamados dessa gerência, solicitações feitas por usuários de sistemas e de equipamentos para as áreas solucionadores da empresa.

A diretoria de tecnologia da companhia segue os conceitos do ITIL [18], e a gerência responsável pelos incidentes da empresa faz o acompanhamento dos chamados com o objetivo de verificar se os chamados foram abertos com a categoria correta, se as resoluções dos chamados são passiveis de criar mudanças e solucionar os problemas na sua causa raiz, fazendo com que estes chamados não voltem a ser abertos novamente.

Esses números são avaliados e buscam-se formas de se reduzirem os incidentes, pois um incidente indica que o trabalho de um ou vários colaboradores está sendo afetado, e isso gera prejuízo para empresa, já que incidentes podem afetar até mesmo uma venda de um plano de telefonia ou internet ao cliente final.

A classificação das ocorrências de TI feitas de forma errada pode afetar de forma muito negativa e drástica o desempenho de atividades dos colaboradores como, também, o funcionamento de alguns sistemas críticos da empresa, como sistemas de vendas, por exemplo.

Caso um chamado relatando um problema em um sistema crítico seja feito de forma errada, o tempo de atuação da equipe solucionadora pode ser prejudicado e isso pode acarretar perdas de vendas para empresa. Isso afeta diretamente os números dos indicadores da gerência responsável por manter o sistema ativo e saudável.

O usuário, sem saber, classifica o chamado de forma errada, e acaba afetando uma cadeia inteira de áreas solucionadoras, quer seja de sistemas ou de infraestrutura.

A gerência de incidentes faz essas análises à mão, o que gera muito custo e horas para analisar tantos chamados, mesmo pegando uma margem amostral, é muito trabalhoso.

Essas análises deveriam ser mais amplas, abordar mais chamados, pois o incidente pode causar um grande prejuízo para a empresa; um incidente pode representar um colaborador de vendas sem trabalhar porque seu computador parou de funcionar; um colaborador do setor de pregões não vai participar de um pregão porque seu notebook não está funcionando; ou a empresa pode parar de arrecadar, pois um sistema de vendas está indisponível. Ou seja, todo incidente é importante, deve ser investigada sua causa raiz e ter uma solução definitiva.

A escolha de um incidente aberto de forma errada para ser analisada pela equipe de auditoria gera uma perda de tempo, perda de esforço, assim como um incidente categorizado de forma errada, como uma atividade recorrente, pode deixar de ter uma causa raiz solucionada, fazendo com que esse chamado seja aberto novamente no futuro gerando prejuízos para a companhia.

Por conta dos problemas na seleção de chamados para análise, foi requisitado pela gerência uma forma de se conseguir selecionar os incidentes de verdade, independentemente de como esteja categorizado, para ser auditado. Com a disponibilidade das novas tecnologias disponíveis hoje, isso é possível, analisar cada chamado aberto, lendo sua descrição e classificando o chamado como sendo

um incidente ou não, com o uso da mineração de texto e da inteligência artificial isso pode ser muito bem aplicável.

Hoje a mineração de texto tem sido visada como investimento nas empresas, pois a necessidade de entender seus clientes é muito importante, como eles se comportam, do que eles gostam, quais suas reclamações. [11]

Umas das grandes vantagens da mineração de texto é a compreensão e o entendimento de documentos na troca de mensagens, pois antes era necessário fazer uma leitura de um documento para saber sobre o que se tratava ao passo que hoje a mineração de texto avalia as principais palavras do documento por meio de combinações de palavras com estatísticas. Com isso é possível criar algoritmos que descrevem com precisão sobre o assunto tratado naquele documento e ainda avaliações de atendimentos em chats ou telemarketing.[12]

Pergunta-chave:

Quais são os termos/palavras-chaves presentes nas descrições dos chamados abertos que os classificam/agrupam em chamados corretos (atendidos) e errados (cancelados)?

Objetivo Geral

O objetivo principal deste trabalho é utilizar a mineração de texto e suas técnicas para criar um algoritmo de aprendizagem de máquina que permita reclassificar de maneira semiautônoma os chamados abertos para a área de microinformática, a fim de serem auditados conforme sua classificação.

Chamados de um certo tipo de classificação sejam auditados por sua real classificação.

2. TRABALHOS RELACIONADOS

Figura 2: Fluxo de Atividades – Pesquisa de Trabalhos Correlatos.

Fonte: Elaborado pelos autores (2020).

Existem diversos trabalhos relacionados ao uso de técnicas de mineração de texto para classificação, os mais comuns são para classificar documentos, por exemplo, o trabalho de triagem de denúncias à CGU[1], porém, o que diferencia os trabalhos são as formas de extração dos dados, pois as técnicas e algoritmos de mineração de texto são parecidos.

No trabalho de Santos e Fernando (2013), houve erros de classificação que poderiam prejudicar a aprendizagem do algoritmo, foram criados três corpus, dois criados a partir das extrações diretas e um criado através de dados conferidos manualmente e classificados corretamente. O corpus B foi

criado a partir do corpus A, que foi classificado manualmente; o corpus B seguiu com as classificações extraídas; o corpus C foi criado com a mescla entre os corpora A e B. Cada Corpus continha 10 mil avaliações, a partir dessas bases foram aplicados três pré-processamentos e criados 3 novos corpora para treinamento:

- Um corpus sem nenhum tratamento;
- Um corpus com padronização de termos;
- Um corpus com padronização de termos e correção ortográfica;
- Um corpus com padronização de termos, correção ortográfica e stemming.

O processo de correção das palavras foi feito utilizando o projeto "GNU Aspell" [4], com ajuda dessa ferramenta foi criado um dicionário de palavras com gírias, abreviações e erros mais comuns de escrita, as correções foram feitas de forma manual.

A utilização de *stopwords* foi utilizada em todos os corpus, pois é um procedimento que é reconhecidamente eficaz e utilizado em quase todas as análises e tratamentos de mineração de texto [3].

Todos estes *corpora* foram criados para avaliar o impacto dos processamentos na análise e no resultado final do processo. Esse processo foi adotado neste artigo para observar se teria melhoras.

Foi criada uma tabela de frequência relativa para que sejam criadas classificações de termos referentes ao texto, é dada um peso a cada palavra ou conjunto de palavras para aquele texto, *Target,* técnica conhecida como TF-IDF. Com isso obteve-se que o melhor resultado foi o *corpus* 3, com padronização de termos e correção ortográfica utilizando o algoritmo SVM do Weka.[19]

No trabalho feito por W. Augusto (2016), foi realizada uma análise de sentimentos utilizando mensagens do Twitter. As mensagens foram pegas a esmo, aleatórias, foram recolhidas quatrocentos e noventa e quatro mensagens, onde foi separado 80% desses dados para treinamento, 10% para validação cruzada e os outros 10% restantes para o teste. Esse trabalho forneceu uma boa perspectiva de como se trabalhar com NLP (Processamento de Linguagem Natural).

Foi feita a criação de um corpus separando os twittes em positivos e negativos. Na etapa seguinte foram realizados tratamentos de limpeza utilizando o pacote NLTK, remoção de stopwords e steamming. Com a biblioteca NLTK, foi feito um treinamento com o algoritmo Naïve Bayes, usando o Scikit-Learn foi realizada validação cruzada após o treinamento, com esses procedimentos o nível de acerto final ficou em 91%. Muito baseado nessa referência foi aplicado os tratamentos e testes neste artigo

No trabalho realizado por Lima e Guerra (2018), foi utilizado o modelo word2vec para fazer classificação. Baseado em palavras e contextos utilizando técnicas de CBOW e Skip-gram, foram utilizadas mensagens recolhidas do Twitter do segundo semestre de 2014, visto que as mensagens, por terem apenas 140 caracteres, seria difícil associar palavras a contextos e assim recolheram também mensagens de avaliações para agregar as mensagens do twitter. As mensagens foram separadas por assuntos, as mensagens do twitter estavam divididas em três assuntos: Hobbit3 (filme), Archeage

(Jogo) e iPhone6 (smartphone). As mensagens de avaliações seguiram a linha de assunto do twitter: filmes, jogos digitais e celulares e acessórios. A técnica de Word2Vec também será aplicada a este trabalho a fim comparar as técnicas TF-IDF e Word Embeddings, onde será aplicada a arquitetura do *Skip-gram* (técnica de prever um contexto baseado em uma palavra-chave) [13].

Com essas bases, foram criados 2 modelos word2vec, um utilizando técnicas de CBOW e outro de Skip-gram[8]. Os modelos para o twitter tinham janelas de contexto de tamanho 2 e para as avaliações onde os textos eram maiores foi utilizada janelas de tamanho 10 com frequência mínima de 40 vezes por palavra, ou seja, cada palavra deveria aparecer no mínimo 40 vezes. Os modelos foram gerados utilizando uma biblioteca do python, gensim.

Os vetores de cada base foram submetidos aos classificados Naive Bayes e SVM utilizando validação cruzada de cinco partições, foram realizadas cinco classificações, para estes procedimentos foi utilizado o Scikit-Learn, o modelo gerou classificadores de filmes para avaliar os tweets de Hobbit3, o classificador de jogos para avaliar os tweets de Archeage e o modelo de celulares para avaliar os tweets de iPhone6.

Utilizando a métrica F-measure, verificou-se que, utilizando a base de filmes para classificar os tweets de hobbit3, foi muito melhor que utilizar a própria base de dados extraídas do twitter, utilizando o SVM houve taxas de acerto de 93% e utilizando Naive Bayes o acerto foi de 88%[8].

Os resultados da base de comentários utilizando SVM também foram melhores para classificar comentários do jogo Archeage 79% contra 74%. Para a base do iPhone, também foi melhor os resultados, SVM 83% contra 80% do Naive Bayes.

Foi aplicado neste artigo algumas das técnicas apresentadas por Santos e Fernando (2013), W. Augusto (2016) e Lima e Guerra (2018).

Foi feita uma separação dos arquivos de treinamento e *corpus* diferentes, no caso foi aplicado dois datasets para treino, um completo, com a descrição do usuário, apenas aplicada técnicas de limpeza, e outro dataset com curadoria Neste foi realizado a classificação de forma manual e deixando apenas um problemas por chamado [2].

Também foram aplicadas técnicas de tratamento de texto utilizando a biblioteca NLTK do Python, treinamento com algoritmo Naive Bayes e outras técnicas do Scki-Kit Learning [7].

E com o intuito de aplicar técnicas um pouco mais modernas, foram aplicadas técnicas de classificação utilizando word2vec, com a utilização desta técnica foi possível aplicar técnicas de classificação através de semântica utilizando a arquitetura do *skip-gram.* [8]

Utilizando esses trabalhos como referência e aplicando suas técnicas, os resultados foram bastante satisfatórios, assim como os trabalhos relacionados.

2.1. *Mineração de texto*

A mineração de texto é uma vertente da mineração de dados, onde se recuperam informações para gerar aprendizagem ou para tomada de decisões estratégicas utilizando banco de dados ou de procedimentos estatísticos. A mineração de texto é um conjunto de métodos para descobrir informações

dentro de bases textuais, pode ser classificada como uma extensão do data mining.

No geral a mineração de texto é diferente de mecanismos de buscas onde o usuário sabe o que quer encontrar, no TM o usuário tem a possibilidade de fazer descobertas desconhecidas. A mineração de texto veio em decorrência da necessidade de se descobrir informações em textos [5].

Com a crescente informalização da informação os bancos de dados estão cada vez maiores, nestes estão contidas várias informações importantes como tendências e padrões que podem auxiliar na tomada de decisões estratégicas das empresas, não importando seu tamanho e alcance. Muitas técnicas foram desenvolvidas para auxiliar na extração dessas informações, uma delas foi a o KDD (knowledge discovery in database) e KDT (Knowledge Discovery in Textual Databases) [6].

2.2. *PLN*

O PLN é baseado em processamento de textos baseados na forma de comunicação humana levando em conta as formas, referências, estrutura e significados, de forma mais clara e direta é dizer que o computador ou um robô vai conseguir se comunicar com um ser humano, nem sempre de forma perfeita, com formação de sentenças, sons ou palavras [6].

3. METODOLOGIA

Este artigo seguiu as recomendações do método CRISP-DM [9][10]:

- Compreensão do negócio;
- Compreensão dos dados;
- Preparação dos dados;
- Modelagem;
- Avaliação.

Na execução deste trabalho foram utilizadas 2 máquinas: Desktop com um processador Intel core I5 3300, SSD 240GB e 32GB de Ram DDR3 1600MHZ; Notebook com processador Intel Core I7 7500, SSD 500GB e 8GB de Ram DDR4 2133MHZ.

Para realização deste trabalho foram utilizadas as seguintes ferramentas:

- Python – O Python é uma ferramenta utilizada para várias tarefas como desenvolvimento Web e científico numérico, principal motivo para utilização neste trabalho.
- NLTK - Biblioteca para criação de programas visando PLN em Python.
- Scikit Learn - Biblioteca que trabalha com várias técnicas de machine learning em Python.

- H2o.ai – Biblioteca que trabalha com várias técnicas de machine learning, será utilizada para gerar modelos baseados em Word2Vec.
- Oracle – SGBD (Banco de dados que utiliza linguagem SQL, Structured Query Language, ou Linguagem de Consulta Estruturada). Onde foi feita a extração dos dados e um pré-processamento dos dados.

3.1. *Compreensão do Negócio*

Figura 3: Fluxo de Atividades – Compreensão do negócio.

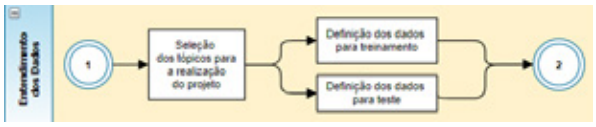
Fonte: Elaborado pelos autores (2020).

Nesta etapa é importante conhecer bem o negócio e o problema que é preciso resolver.

Neste artigo o problema exposto é de se obter o melhor algoritmo para categorizar uma reclamação feita por um usuário a respeito de problemas de funcionamento de seu desktop ou notebook.

3.2. *Compreensão dos dados*

Figura 4: Fluxo de Atividades – Entendimento dos Dados.



Fonte: Elaborado pelos autores (2020).

Nesta etapa é importante conhecer bem os dados, saber do que cada variável se trata, avaliar quais aspectos podem fazer a diferença no resultado final.

Os dados para análise deste projeto foram todos extraídos da base de dados da ferramenta de registro de chamados, ARSystem (https://www.bmc.com), da empresa analisada.

Foi feito um select no banco de dados da ferramenta, utilizando PL/SQL foi extraído do banco Oracle um arquivo CSV contendo o número do chamado, descrição do chamado e a categoria a qual o chamado pertence (TARGET).

Foi feita a extração dos Top 5 chamados de hardware. A escolha dos top 5 é feita através da média de chamados abertos dos últimos 30 dias, sysdate – 30, porém, foi utilizado chamados de 2016 até o momento.

A gerência de microinformática é dividida em 4 regionais:

- RJ-SP;
- MG-SUL;
- CO;
- NO-NE.

A Regional RJ-SP contempla os estados do RJ e SP, essa regional tem apenas estes dois estados, pois a filial RJ é a que possui o maior número de colaboradores da empresa e SP o menor, assim a gerência consegue se equilibrar no tratamento dos problemas das duas regionais.

A Regional MG-SUL contempla o estado de MG e os estados do Sul (MG, ES, PR, SC, RS), esta regional concentra um bom número de ativos e chamados por conta de um call center localizado na regional PR.

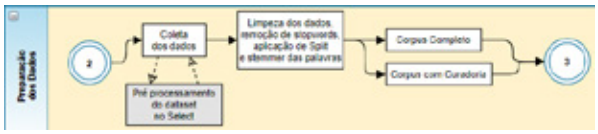
A Regional CO contempla exatamente os estados da região centro-oeste e o Distrito Federal (DF, GO, MT, MS, TO). Essa é a segunda maior regional da empresa, essa regional tem o segundo maior número de colaboradores, e ativos de microinformática, isso se deve ao fato dessa regional suportar dois call centers, um em GO e outro em MS.

Por fim, a regional NO-NE, essa regional é a menor em número de ativos de microinformática e colaboradores, assim pôde ficar juntos todos os estados do norte e nordeste em uma única regional (AC, AM, RR, RO, AP, PA, MA, PI, BA, CE, RN, PB, PE, AL, SE).

Para a base de treinamento foram extraídos 66.666 registros, para o teste foram extraídos 6.063 chamados. Dos chamados utilizados para treinamento 44,09% são referentes da região CO (29.396), isso demonstrar o impacto que os dois call centers geram nos indicadores dessa regional.

3.3. *Preparação dos dados*

Figura 5: Fluxo de Atividades – Preparação dos Dados.



Fonte: Elaborado pelos autores (2020).

Nesta fase é feita a inspeção, organização e a preparação dos dados, modo em que as informações serão definidas na base de dados, os formatos e quais campos serão utilizados.

Seguindo a literatura e procedimentos adotados em artigos já mencionados neste texto, foi realizado alguns ajustes na base de dados.

No momento da extração dos dados da base, foi feito um pré-tratamento no campo "Descrição", o *select* foi feito já transformando o texto para minúsculo e retirando todos os parágrafos, todas as quebras de linha foram transformadas em um espaço em branco.

No python, usando funções da biblioteca NLTK, foram aplicados outros tratamentos ao texto como:

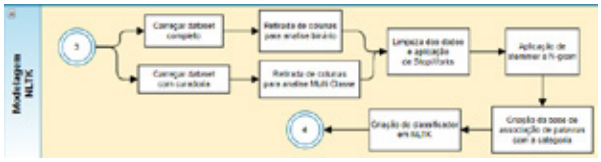
- Remoção de números;

- Remoção de pontuação;
- Remoção de acentuação, símbolos e cê-cedilha;
- Remoção de espaços em branco duplos;
- Feito split (técnica onde a frase é quebrada e as palavras são separadas);
- Aplicação do StopWords (técnica onde são removidas palavras que não agregam ao treinamento, como artigos, preposições e etc.);
- Aplicação do stemmer (técnica que deixa apenas os radicais das palavras).

Após estes tratamentos o arquivo foi submetido aos algoritmos para construção de modelos de treinamento.

3.4. *Modelagem*

Figura 6: Fluxo de Atividades – Modelagem NLTK.



Fonte: Elaborado pelos autores (2020).

Figura 7: Fluxo de Atividades – Modelagem SciKit Learn.

Fonte: Elaborado pelos autores (2020).

Nesta fase é feita a construção dos modelos onde será realizado o aprendizado da máquina, a modelagem geralmente é dada em diversas iterações, são feitos diversos modelos usando vários parâmetros e ajustes, não tem como saber qual é o melhor modelo e o melhor algoritmo, é preciso testar todas as opções disponíveis.

Neste trabalho foram elaborados vários algoritmos para treinar a máquina com o intuito de classificar um texto de acordo com o TARGET, COMPLEMENTOABERTURA.

Foram utilizadas as seguintes bibliotecas para treinar os algoritmos:

- NLTK;
- SciKit- Learn;
- H2o.

Para as bibliotecas NLTK e SciKit-Learn, foram feitos os treinamentos baseados na relevância

de palavras para cada *Target*, técnica do TF-IDF [14].

Na biblioteca H2o, foram criados algoritmos utilizando a técnica do Word2vec baseados no modelo skip-gram [15].

Nos algoritmos de TF-IDF, foi aplicado o método de frequência N-gram, esta técnica busca fornecer uma maneira mais eficaz de classificar documentos e textos buscando associações de *Target* utilizando pequenos contextos e não palavras únicas [16].

Para os modelos do SciKit-Learn foram utilizados os seguintes N-grams:

- UniGram;
- BiGram;
- TriGram;
- UniBigram;
- UniTrigram.

Para os modelos feitos no NLTK, os N-grams utilizados foram:

- UniGram;
- BiGram;
- TriGram;

No NLTK não foram utilizados os N-grams UniBiGram (blocos de uma palavra mais um bloco de Bigramas) e UniTrigram (blocos de uma palavra com mais um bloco de Trigramas) devido à limitação computacional.

3.4.1 Corpus Multi Classe completo

No primeiro momento foram feitos os treinamentos utilizando a base multi classe, ou seja, em um único arquivo de treinamento estavam todos os 5 Target (BATERIA, HD, MEMÓRIA, MOUSE, TECLADO).

Os Algoritmos utilizados para as classificações utilizando a biblioteca SciKit-learn foram:

- Random Forest;
  - Número de árvores = 1000;
  - Profundidade de árvores = 3;
  - Quantidade de saídas por nó = 2.
- Gradient Boosting;
  - Número de estágios de execução =1000;
  - Profundidade de árvores =5;
  - Aleatoriedade das iterações = 0.
- Support Vector Machines;
  - Configuração padrão (Devido à limitação computacional).
- Naive Bayes
  - Configuração padrão (Devido à limitação computacional).

O algoritmo utilizado na biblioteca NLTK foi o Naive Bayes.

3.4.2 Corpus Multi Classe com curadoria

Foi utilizado os mesmos parâmetros dos algoritmos como mostrado no tópico 3.4.1, porém, o corpus está menor, foram separados 100 chamados de cada categoria, feita curadoria deles a fim de testar o desempenho de classificação dos algoritmos.

3.4.3 Corpus Binário Completo

Para este treinamento foi utilizado um corpus separado por complementos de abertura, foram criados cinco corpora, um para casa *Target,* caso a frase pertencesse aquela classe era marcado o *Target* = 1, se não, 0.

As configurações dos algoritmos permaneceram as mesmas, conforme item 3.4.1, a diferença está na criação dos modelos, foi criado um modelo específico para cada complemento de abertura. Para cada classe foi criado um algoritmo utilizando uma técnica, exemplo: Para a classe MOUSE, TECLADO, MEMÓRIA, HD E BATERIA foram criados quatro modelos, um para cada algoritmo (Random Forest, GBM, SVM e Naive Bayes).

3.4.4 Corpus Binário com curadoria

Foi utilizado os mesmos parâmetros dos algoritmos mostrados no tópico 3.4.3, porém, com o corpus menor, como já informado no tópico 3.4.2.

3.4.5 Corpus Binário Completo para CBOW

Figura 8: Fluxo de Atividades – Tratamento dos dados – H2o

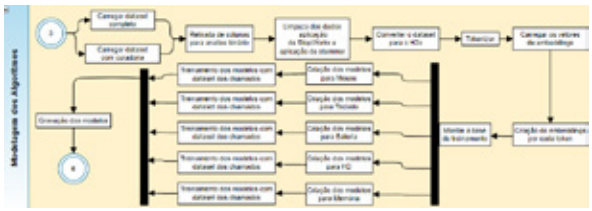
Fonte: Elaborado pelos autores (2020).

Para este treinamento foi utilizado o mesmo corpus do item 3.4.3, um dataset completo com quase 70 mil registros separados por categorias binárias, porém, a preparação no word2vec é diferente, primeiro é criado um dicionário de similaridade, os arquivos foram importados para um dataframe no h2o, onde foi criado o dicionário seguindo os seguintes parâmetros:

- Taxa de envio de amostras = 0.0
- Épocas = 20
- Tamanho do Vetor = 400
- Frequência mínima de palavras = 5

Desta forma obtém-se uma matriz de similaridade entre os termos, essa matriz será utilizada no momento de aprendizado com os algoritmos.

Figura 9: Fluxo de Atividades – Modelagem dos algoritmos – H2o.



Fonte: Elaborado pelos autores (2020).

Para obter um nível de comparação igualitário, foram utilizados os mesmos algoritmos do scikit-learn no h2o, porém, foi utilizado um recurso adicional disponível neste framework.

O H2o possui um classificador automático, onde é passado os arquivos de treinamento e o próprio framework faz um treinamento a fim de indicar qual o melhor algoritmo para o problema exposto.

Foi passado para o H2oAutoML o arquivo de treinamento de cada complemento de abertura, utilizando um vetor de 400 posições, podendo fazer o treinamento por seiscentos segundos.

Desta forma, o h2o indicou utilizar o XGBoost e o Deep Learning.

Figura 10: Algoritmos indicados pelo H2o.

Fonte: Elaborado pelos autores (2020).

Portanto, para a biblioteca H2o foram utilizados os algoritmos:

- Random Forest;
- Gradient Boosting;
- Support Vector Machines;
- Naive Bayes;
- XGBoost;
- Deep Learning.

Com relação aos parâmetros de treinamento no H2o, foram treinados os modelos com os mesmos parâmetros do NLTK e Sci-kit Learn, desta forma ficando mais justo os comparativos entre as bibliotecas.

Para o XGBoost foi utilizado um parâmetro mais próximo possível do GBM, porém, como a própria biblioteca recomendou o algoritmo alguns parâmetros foram utilizados a mais, por exemplo: n-folds e normalize_type.

A diferença em relação aos parâmetros ficou para o Deep Learning, para este algoritmo foi utilizado o *grid,* configuração que permite carregar várias configurações diferentes para teste dos modelos.

No *grid* foi adicionado parâmetros para a quantidade de épocas e quantidade de camadas

ocultas, para todos os cinco complementos de abertura foi utilizada a mesma configuração do *grid*. Para o treinamento foi adicionado a mais o nfolds.

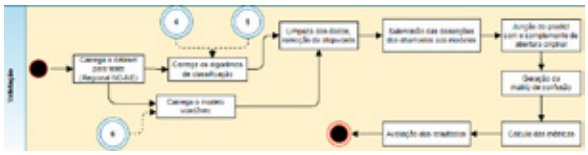
Para estes treinamentos foi adicionado a mais o nfolds.

3.4.6 Corpus Binário com Curadoria

Foi utilizado os mesmos parâmetros dos algoritmos mostrados no tópico 3.4.5, porém, com o corpus menor, como já informado no tópico 3.4.2.

3.5. *Avaliação*

Figura 11: Fluxo de Atividades – Validação



Fonte: Elaborado pelos autores (2020).

Neste tópico será apresentado os resultados obtidos com os modelos construídos.

As métricas utilizadas para avaliar os resultados são baseadas na matriz de confusão, após executar os testes nos modelos foi extraído a matriz e calculado os valores para as métricas [17]:

- Precision = TP/ (TP+FP)
- Recall = TP/ (TP+FN)
- F-Measure = 2 × (precision × recall) / (precision + recall)
- Accuracy = (TP+TN) / (TP + TN + FP + FN)

Onde:

- Verdadeiro Positivo = (TP)
- Falso Positivo = (FP)
- Verdadeiro Negativo = (TN)
- Falso Negativo = (FN)

Para classificar os resultados, foi utilizada a métrica F-Measure, por se tratar de uma média harmônica entre precisão e recall.

3.5.1 NLTK

3.5.1.1 Corpus MultiClasse

Utilizando o classificador Naive Bayes o melhor resultado foi obtido com a base completa, a base com curadoria teve resultados muito inferiores. Tendo referência a métrica F1, os complementos de abertura Bateria, HD, Memória e Mouse tiveram o melhor resultado utilizando Unigram, para o Teclado o melhor foi o que utilizou Bigram:

- Bateria – 0,97
- HD – 0,84

- Memoria – 0,90
- Mouse – 0,99
- Teclado – 1

3.5.1.2 Corpus Binário

Utilizando o classificador Naive Bayes, novamente os algoritmos que usaram a base completa se saíram melhores que utilizando a base com curadoria, dessa vez para todos os complementos o melhor resultado foi obtido utilizando Bigram:

- Bateria – 0,97
- HD – 0,89
- Memoria – 0,92
- Mouse – 0,98
- Teclado – 0,97

3.5.2 SciKit-Learn

3.5.2.1 Corpus MultiClasse

No geral os resultados foram bem esparsos, alguns algoritmos foram melhores com curadoria outros melhores com a base completa.

O melhor resultado para Mouse e Bateria foram na base com curadoria, as outras três tiveram melhor resultados na base completa. Um ponto a se observar que os resultados também foram bem diversificados entre os algoritmos utilizados, nenhum algoritmo foi soberano nos resultados.

Utilizando *corpus* completo:

- Bateria utilizando SVM – 0,97
- HD utilizando RF – 0,86
- Memória utilizando RF – 0,93
- Mouse utilizando SVM – 0,98
- Teclado utilizando SVM – 0,98

Utilizando *corpus* com curadoria:

- Bateria utilizando NB – 1
- HD utilizando NB – 0,88
- Memória utilizando RF – 0,90
- Mouse utilizando SVM – 1
- Teclado utilizando SVM – 0,97

3.5.2.2 Corpus Binário

Os melhores resultados para as bases completas e com curadoria ficaram todos muito parecidos, os percentuais do F1 e precisão muito iguais, alguns resultados obtiveram inclusive as métricas exatamente iguais, optou-se por escolher como o melhor resultado sempre o modelo com

o menor número de N-grams, pois tem a vantagem de ser menos dispendioso para o processamento computacional para treinar e testar o modelo, porém, os resultados ruins foram bem abaixo na base com curadoria, algumas métricas de Trigram ficaram zeradas.

Utilizando *corpus* completa:

- Mouse utilizando RF – 0,98
- Teclado utilizando RF – 0,98
- Memória utilizando RF – 0,92
- HD utilizando RF – 0,84
- Bateria utilizando RF – 0,97

Utilizando *corpus* com curadoria:

- Mouse utilizando RF – 0,98
- Teclado utilizando RF – 0,97
- Memória utilizando NB – 0,85
- HD utilizando SVM – 0,88
- Bateria utilizando RF – 0,98

3.5.3 H2o

3.5.3.1 Corpus Binário

O melhor resultado obtido com o *corpus* completo foram os que utilizaram Deep Learn:

- Mouse – 1
- Teclado – 0,99
- Memoria – 1
- HD – 0,91
- Bateria – 0,97

O melhor resultado obtido com o *corpus* curadoria foi novamente aqueles que utilizaram o Deep Learn:

- Mouse – 1
- Teclado – 0,98
- Memoria – 1
- HD – 0,98
- Bateria – 1

4. CONCLUSÃO

Como apresentado e informado nas avaliações, a diferença no resultado final entre os algoritmos não foi tão grande, ficaram todos muito parelhos, alguns modelos empatados no F1-score, precisão, recall e acurácia fazendo com que o modelo fosse escolhido pela questão computacional, modelo com menor tempo de aprendizagem.

A melhor surpresa ficou para os modelos sugeridos pelo AutoML do H2o, com os testes realizados com o XGBoost e o Deep Learning notou-se um grande benefício, os resultados foram muito bons, o Deep Learning teve uma performance melhor do que qualquer outro algoritmo e o XGBoost teve performance parelha e até melhor em alguns casos, levantando a questão se não seria melhor ter feito todo o trabalho utilizando o modelo de embedding word2vec com o H2o e utilizando os algoritmos propostos pelo H2o, essa hipótese não é pelo resultado final em si, mas pelo esforço de treinamento, pelo esforço computacional para treinar os outros algoritmos, pois os modelos treinados no H2o, mesmo com configurações igual aos modelos treinados no NLTK e no SciKit-Learnin tiveram um tempo de treinamento consideravelmente menor, isso traria muito mais rapidez para a finalização do trabalho.

Após avaliar todos os resultados obtidos, para este cenário, ficou evidente que a base com curadoria não gerou resultados tão diferentes da base completa, o grande esforço feito para realizar uma curadoria não vale a pena, pois este processo é o que gera mais esforço e desgaste. Conforme verificado nas tabelas do apêndice, a taxa da precisão ficou bem alta, ou seja, os modelos estavam acertando muito bem quando era um verdadeiro positivo ou verdadeiro negativo.

Os piores resultados tenderam a aparecer para a classe HD, a princípio o motivo seria o leve desbalanceamento desta classe, porém, ao fazer os treinos com a base de curadoria os resultados se mantiveram, HD tendo taxas menores. Nota-se também que quanto mais números no N-gram o resultado tende a ter uma leve queda de desempenho, dessa forma fazendo refletir se vale a pena usar mais de dois N-grams.

Este trabalho foi de grande valia para o aprendizado prático de mineração de texto e aprendizado de máquina, foi possível aprender técnicas já consolidadas e novas técnicas para este fim. Também comprovou-se ser altamente viável classificar chamados tendo apenas como referência a descrição do chamado, o texto escrito pelo usuário informando qual problema ele está tendo.

5. REFERÊNCIAS

1. H.M.A.A, Helena. 2015. Aplicação de Técnicas de Mineração de Textos para Classificação de Documentos: um Estudo da Automatização da Triagem de Denúncias na CGU. Disponível em: http://repositorio.unb.br/handle/10482/21004. Acesso em: 22 jun. 2019.
2. SANTOS, Fernando. 2013. Mineração de opinião em textos opinativos utilizando algoritmos de classificação. Disponível em: http://bdm.unb.br/handle/10483/7711. Acesso em: 4 set. 2019.
3. R. G. Igor., e M. O. L. O. Thiago. 2006. Etapas do processo de mineração de textos – uma abordagem aplicada a textos em português do brasil. Disponível em: https://www.maxwell.vrac.puc-rio.br/11675/11675_4.PDF. Acesso em: 10 out. 2019.
4. http://aspell.net/.
5. A. E.P. Christian. 2006. A Tecnologia de Mineração de Textos. Disponível em: http://periodicosibepes.org.br/index.php/reinfo/article/view/171/66. Acesso em: 19 jun. 2019.
6. C. N. B. Eliana. 2008. Mineração de Textos. Disponível em: https://revista.pgsskroton.com/index.php/rcext/article/view/2372/2276. Acesso em: 19 jun. 2019.
7. W. Augusto. 2016. Análise de sentimentos do Twitter com Naïve Bayes e NLTK. Disponível em: http://sys.facos.edu.br/ojs/index.php/trajetoria/article/view/135/157 . Acesso em: 18 out.

2019.
8. LIMA, Raul de Araújo; GUERRA, Paulo T. 2018. An Analysis of the Sentiment Classification of Short Messages Using Word2Vec. Disponível https://doi.org/10.5753/eniac.2018.4436. Acessa em: 18 out. 2019.
9. Site IBM. Disponível em: https://www.ibm.com/support/knowledgecenter/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html. Acesso em: 23 jan. 2018.
10. Site Big Data Business. Disponível em: http://www.bigdatabusiness.com.br/se-voce-se-interessa-por-big-data-precisa-entender-o-crisp-dm/. Acesso em: 23 jan. 2018.
11. VIEIRA, C. A. G. 2015. A interdependência entre a identidade da marca e a imagem criada pelos consumidores num contexto de second screening: o caso 5iRTP. Disponível em: https://repositorio.ucp.pt/handle/10400.14/17065. Acesso em: 18 out. 2019.
12. Site take.net. Disponível em: https://take.net/blog/chatbots/chatbot-call-center/. Acesso em: 20 jun. 2019.
13. Pari, Claver; Nunes, Gustavo; Gomes, José. 2019. Avaliação de técnicas de word embedding na tarefa de detecção de discurso de ódio. Disponível em: https://sol.sbc.org.br/index.php/eniac/article/view/9354/9256. Acesso em: 05 jul. 2020.
14. R. Juan. 2003. Using TF-IDF to Determine Word Relevance in Document Queries. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf. Acesso em: 18 out. 2019.
15. R. Xin. 2016. word2vec Parameter Learning Explained. Disponível em: https://arxiv.org/pdf/1411.2738.pdf. Acesso em: 18 out. 2019.
16. B. C. William e M. T. John. 1994. N-Gram-Based Text Categorization. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.3248&rep=rep1&type=pdf. Acesso em: 25 jun. 2020.
17. Hamido, F. et al. Trends in Applied Knowledge-Based Systems and Data Science. Japão: Editora Springer, 2016.
18. https://www.axelos.com/best-practice-solutions/itil/what-is-itil.
19. Weka - data mining software in java. http://www.cs.waikato.ac.nz/ml/weka/. Acesso em: 4 set. 2019.

APÊNDICE A

Arquivos de dados

Base de dados multi classe

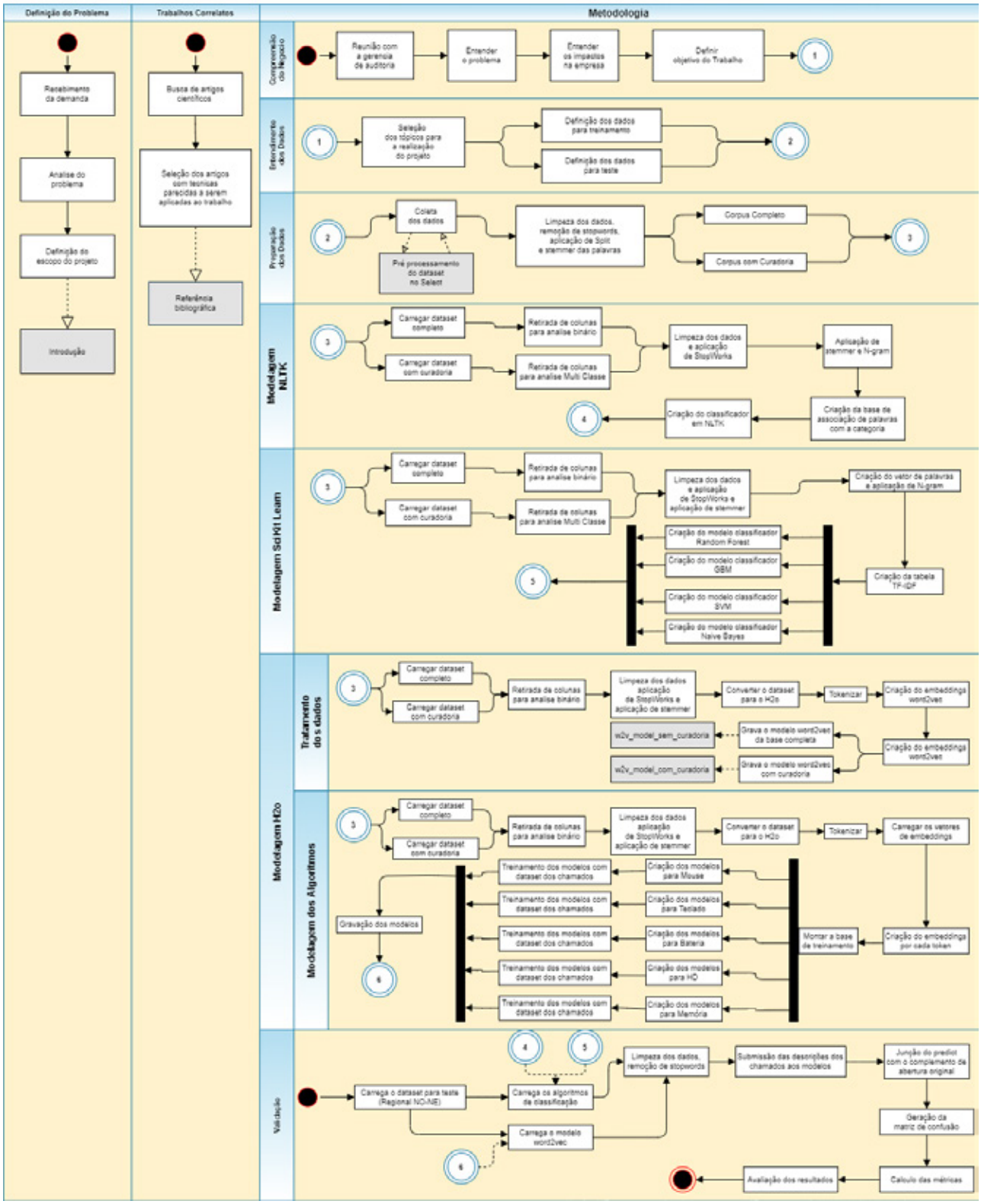
	DESCRICAO	COMPLEMENTOABERTURA
0	a linha do teclado as ve es não funciona	TECLADO
1	a pa está com teclado travando tecla espaç...	TECLADO
2	a teclado	TECLADO
3	a agente de bc dayane questiona que o botão...	TECLADO
4	a bateria aceitando da carga somente	BATERIA
...	...	...
72724	windows não inicia pa ip oidesk	HD
72725	windows não inicia erro inaccessible boot de...	HD
72726	windows não iniciali a	HD
72727	windows não inicializa	HD
72728	windows travando constantemente	HD

Base de dados binário

	DESCRICAO	BATERIA	HD	MEMORIA	MOUSE	TECLADO
0	a linha do teclado as ve es não funciona	0	0	0	0	1
1	a pa está com teclado travando tecla espaç...	0	0	0	0	1
2	a teclado	0	0	0	0	1
3	a agente de bc dayane questiona que o botão...	0	0	0	0	1
4	a bateria aceitando da carga somente	1	0	0	0	0
...	...	...	...	...	...	...
72724	windows não inicia pa ip oidesk	0	1	0	0	0
72725	windows não inicia erro inaccessible boot de...	0	1	0	0	0
72726	windows não iniciali a	0	1	0	0	0
72727	windows não inicializa	0	1	0	0	0
72728	windows travando constantemente	0	1	0	0	0

APÊNDICE B

Fluxo de desenvolvimento do artigo



APÊNDICE C

Métricas dos modelos

• NLTK base Multi Classe

		Precision	Recall	F1-score	Accuracy
Bateria	UniGram	0,95	0,98	0,97	1,00
	BiGram	0,00	0,00	0,00	0,92
	TriGram	0,00	0,00	0,00	0,92
HD	UniGram	0,81	0,87	0,84	0,96
	BiGram	1,00	0,38	0,55	0,98
	TriGram	1,00	0,08	0,15	0,88
Memória	UniGram	0,91	0,90	0,90	0,96
	BiGram	0,98	0,25	0,39	0,86
	TriGram	0,97	0,24	0,39	0,86
Mouse	UniGram	0,99	0,99	0,99	0,99
	BiGram	0,55	1,00	0,71	0,66
	TriGram	0,54	1,00	0,70	0,65
Teclado	UniGram	0,99	0,96	0,97	0,99
	BiGram	1,00	0,99	1,00	1,00
	TriGram	1,00	0,95	0,97	0,99

• NLTK base Multi Classe – Curadoria

		Precision	Recall	F1-score	Accuracy
BATERIA	UniGram	0,81	0,96	0,88	0,98
	BiGram	1	0,00	0,01	0,93
	TriGram	1	0,00	0,01	0,93
HD	UniGram	0,52	0,93	0,66	0,88
	BiGram	0	0,00	0,00	0,87
	TriGram	0	0,00	0,00	0,87
MEMORIA	UniGram	0,93	0,63	0,75	0,92
	BiGram	1,00	0,01	0,01	0,81
	TriGram	1,00	0,01	0,01	0,81
MOUSE	UniGram	0,99	0,93	0,96	0,97
	BiGram	0,41	1,00	0,58	0,41
	TriGram	0,40	0,93	0,56	0,39
TECLADO	UniGram	0,99	0,82	0,89	0,96
	BiGram	0,00	0,00	0,00	0,80
	TriGram	0,21	0,05	0,08	0,78

- NLTK base binário

		Precision	Recall	F1-score	Accuracy
BATERIA	UniGram	0,90	0,99	0,94	0,99
	BiGram	1,00	0,94	0,97	1,00
	TriGram	1,00	0,93	0,96	0,99
HD	UniGram	0,57	0,93	0,71	0,90
	BiGram	1,00	0,81	0,89	0,97
	TriGram	1,00	0,80	0,88	0,97
MEMORIA	UniGram	0,71	0,96	0,82	0,92
	BiGram	0,99	0,87	0,92	0,97
	TriGram	0,99	0,86	0,92	0,97
MOUSE	UniGram	0,99	0,98	0,99	0,99
	BiGram	1,00	0,96	0,98	0,98
	TriGram	0,97	0,96	0,96	0,97
TECLADO	UniGram	0,99	0,95	0,97	0,99
	BiGram	1,00	0,94	0,97	0,99
	TriGram	1,00	0,89	0,94	0,98

- NLTK base binário – Curadoria

		Precision	Recall	F1-score	Accuracy
BATERIA	UniGram	0,86	0,95	0,91	0,98
	BiGram	0,36	0,61	0,45	0,89
	TriGram	0,27	0,59	0,37	0,85
HD	UniGram	0,61	0,81	0,69	0,91
	BiGram	0,39	0,25	0,31	0,85
	TriGram	0,36	0,29	0,32	0,84
MEMORIA	UniGram	0,93	0,74	0,82	0,94
	BiGram	0,62	0,19	0,29	0,83
	TriGram	0,55	0,18	0,27	0,82
MOUSE	UniGram	0,99	0,92	0,96	0,97
	BiGram	0,72	0,40	0,51	0,69
	TriGram	0,63	0,29	0,39	0,64
TECLADO	UniGram	0,96	0,79	0,87	0,95
	BiGram	0,56	0,41	0,48	0,82
	TriGram	0,49	0,39	0,44	0,80

- SciKit-Learn base Multi Classe
  - UniGram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.98	0.95	0.97	1,00
	GBM	0.97	0.95	0.96	1,00
	SVM	0.98	0.95	0.96	1,00
	NB	0.97	0.92	0.95	1,00
HD	RF	0.82	0.89	0.86	0,98
	GBM	0.79	0.87	0.83	0,97
	SVM	0.80	0.86	0.83	0,97
	NB	0.88	0.67	0.76	0,97
MEMORIA	RF	0.94	0.91	0.93	0,98
	GBM	0.94	0.90	0.92	0,98
	SVM	0.92	0.91	0.92	0,98
	NB	0.90	0.90	0.90	0,98
MOUSE	RF	0.98	0.99	0.98	0,99
	GBM	0.98	0.98	0.98	0,98
	SVM	0.98	0.99	0.98	0,99
	NB	0.90	0.99	0.94	0,94
TECLADO	RF	0.98	0.97	0.97	0,99
	GBM	0.98	0.97	0.97	0,99
	SVM	0.98	0.97	0.98	0,99
	NB	0.97	0.86	0.91	0,95

  - BiGram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.89	0.74	0.81	0,99
	GBM	0.87	0.74	0.80	0,99
	SVM	0.95	0.72	0.82	0,99
	NB	0.92	0.72	0.81	0,99
HD	RF	0.58	0.59	0.58	0,94
	GBM	0.57	0.57	0.57	0,94
	SVM	0.57	0.59	0.58	0,93
	NB	0.68	0.42	0.52	0,94
MEMORIA	RF	0.89	0.79	0.83	0,96
	GBM	0.89	0.77	0.83	0,96
	SVM	0.89	0.79	0.83	0,96
	NB	0.89	0.76	0.82	0,96
MOUSE	RF	0.88	0.96	0.92	0,92
	GBM	0.88	0.96	0.92	0,92
	SVM	0.87	0.97	0.92	0,92
	NB	0.83	0.98	0.90	0,89
TECLADO	RF	0.94	0.86	0.90	0,95
	GBM	0.94	0.86	0.90	0,95
	SVM	0.96	0.86	0.91	0,95
	NB	0.97	0.84	0.90	0,95

- TriGram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.95	0.55	0.70	0,98
	GBM	0.94	0.55	0.69	0,98
	SVM	0.95	0.55	0.70	0,98
	NB	0.99	0.53	0.69	0,98
HD	RF	0.52	0.21	0.30	0,93
	GBM	0.53	0.20	0.29	0,93
	SVM	0.54	0.20	0.29	0,93
	NB	0.42	0.15	0.22	0,93
MEMORIA	RF	0.89	0.55	0.68	0,94
	GBM	0.89	0.55	0.68	0,94
	SVM	0.89	0.55	0.68	0,94
	NB	0.87	0.54	0.66	0,94
MOUSE	RF	0.67	0.98	0.79	0,77
	GBM	0.67	0.98	0.79	0,77
	SVM	0.67	0.98	0.79	0,77
	NB	0.66	0.98	0.79	0,76
TECLADO	RF	0.92	0.53	0.67	0,87
	GBM	0.93	0.53	0.67	0,87
	SVM	0.93	0.53	0.67	0,87
	NB	0.94	0.52	0.67	0,87

- UniBigram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.98	0.94	0.96	1,00
	GBM	0.97	0.94	0.96	1,00
	SVM	0.99	0.95	0.97	1,00
	NB	0.90	0.95	0.93	0,99
HD	RF	0.80	0.87	0.83	0,97
	GBM	0.78	0.85	0.81	0,97
	SVM	0.78	0.87	0.82	0,97
	NB	0.79	0.62	0.69	0,96
MEMORIA	RF	0.93	0.90	0.92	0,98
	GBM	0.92	0.89	0.91	0,98
	SVM	0.93	0.90	0.91	0,98
	NB	0.89	0.87	0.88	0,97
MOUSE	RF	0.98	0.98	0.98	0,98
	GBM	0.98	0.98	0.98	0,98
	SVM	0.98	0.98	0.98	0,98
	NB	0.93	0.98	0.95	0,95
TECLADO	RF	0.97	0.97	0.97	0,98
	GBM	0.97	0.96	0.97	0,98
	SVM	0.98	0.97	0.97	0,98
	NB	0.97	0.92	0.95	0,97

- UniTrigram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.98	0.94	0.96	1,00
	GBM	0.97	0.94	0.96	1,00
	SVM	0.99	0.95	0.97	1,00
	NB	0.90	0.95	0.93	0,99
HD	RF	0.80	0.87	0.83	0,97
	GBM	0.78	0.85	0.81	0,97
	SVM	0.78	0.87	0.82	0,97
	NB	0.79	0.62	0.69	0,96
MEMORIA	RF	0.93	0.90	0.92	0,98
	GBM	0.92	0.89	0.91	0,98
	SVM	0.93	0.90	0.91	0,98
	NB	0.89	0.87	0.88	0,97
MOUSE	RF	0.98	0.98	0.98	0,98
	GBM	0.98	0.98	0.98	0,98
	SVM	0.98	0.98	0.98	0,98
	NB	0.93	0.98	0.95	0,95
TECLADO	RF	0.97	0.97	0.97	0,98
	GBM	0.97	0.96	0.97	0,98
	SVM	0.98	0.97	0.97	0,98
	NB	0.97	0.92	0.95	0,97

- SciKit-Learn base Multi Classe – Curadoria
  - UniGram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.95	0.95	0.95	0,98
	GBM	1.00	0.95	0.98	0,99
	SVM	1.00	0.95	0.98	0,99
	NB	1.00	1.00	1.00	1,00
HD	RF	0.76	0.90	0.83	0,92
	GBM	0.71	0.95	0.82	0,91
	SVM	0.79	0.90	0.84	0,93
	NB	0.88	0.67	0.76	0,91
MEMORIA	RF	0.94	0.81	0.87	0,95
	GBM	0.94	0.76	0.84	0,94
	SVM	0.90	0.86	0.88	0,95
	NB	0.83	0.90	0.86	0,94
MOUSE	RF	1.00	0.95	0.98	0,99
	GBM	1.00	0.90	0.95	0,98
	SVM	1.00	1.00	1.00	1,00
	NB	0.84	1.00	0.91	0,96
TECLADO	RF	0.94	0.94	0.94	0,98
	GBM	0.94	0.94	0.94	0,98
	SVM	1.00	0.94	0.97	0,99
	NB	0.94	0.88	0.91	0,97

- BiGram

		Precisio	Recall	F1-score	Accuracy
BATERIA	RF	0.47	0.86	0.61	0,77
	GBM	0.46	0.90	0.61	0,76
	SVM	1.00	0.71	0.83	0,94
	NB	0.94	0.76	0.84	0,94
HD	RF	0.78	0.67	0.72	0,89
	GBM	0.76	0.62	0.68	0,88
	SVM	0.82	0.67	0.74	0,90
	NB	0.87	0.62	0.72	0,90
MEMORIA	RF	0.93	0.62	0.74	0,91
	GBM	0.81	0.62	0.70	0,89
	SVM	0.49	0.86	0.62	0,78
	NB	0.93	0.67	0.78	0,92
MOUSE	RF	0.81	0.62	0.70	0,89
	GBM	0.86	0.57	0.69	0,89
	SVM	0.88	0.67	0.76	0,91
	NB	0.74	0.67	0.70	0,88
TECLADO	RF	0.73	0.65	0.69	0,90
	GBM	0.85	0.65	0.73	0,92
	SVM	0.69	0.65	0.67	0,89
	NB	0.43	0.88	0.58	0,78

- TriGram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	1.00	0.48	0.65	0,89
	GBM	1.00	0.48	0.65	0,89
	SVM	1.00	0.48	0.65	0,89
	NB	1.00	0.48	0.65	0,89
HD	RF	0.27	1.00	0.43	0,45
	GBM	0.26	1.00	0.42	0,42
	SVM	0.27	1.00	0.43	0,45
	NB	0.00	0.00	0.00	0,79
MEMORIA	RF	1.00	0.10	0.17	0,81
	GBM	1.00	0.10	0.17	0,81
	SVM	1.00	0.10	0.17	0,81
	NB	1.00	0.10	0.17	0,81
MOUSE	RF	0.75	0.29	0.41	0,83
	GBM	0.67	0.19	0.30	0,81
	SVM	0.75	0.29	0.41	0,83
	NB	0.75	0.29	0.41	0,83
TECLADO	RF	0.50	0.12	0.19	0,83
	GBM	0.33	0.06	0.10	0,82
	SVM	0.50	0.12	0.19	0,83
	NB	0.19	0.88	0.31	0,33

- UniBigram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.95	0.95	0.95	0,98
	GBM	1.00	0.95	0.98	0,99
	SVM	0.95	0.95	0.95	0,98
	NB	0.95	1.00	0.98	0,99
HD	RF	0.76	0.90	0.83	0,92
	GBM	0.67	0.95	0.78	0,89
	SVM	0.79	0.90	0.84	0,93
	NB	0.90	0.86	0.88	0,95
MEMORIA	RF	0.94	0.81	0.87	0,95
	GBM	0.93	0.67	0.78	0,92
	SVM	0.95	0.86	0.90	0,96
	NB	0.90	0.90	0.90	0,96
MOUSE	RF	1.00	0.95	0.98	0,99
	GBM	1.00	0.90	0.95	0,98
	SVM	1.00	1.00	1.00	1,00
	NB	0.91	1.00	0.95	0,98
TECLADO	RF	0.94	0.94	0.94	0,98
	GBM	0.94	0.94	0.94	0,98
	SVM	1.00	0.94	0.97	0,99
	NB	1.00	0.88	0.94	0,98

- UniTrigram

		Precision	Recall	F1-score	Accuracy
BATERIA	RF	0.95	0.95	0.95	0,98
	GBM	1.00	0.95	0.98	0,99
	SVM	1.00	0.95	0.98	0,99
	NB	1.00	1.00	1.00	1,00
HD	RF	0.79	0.90	0.84	0,93
	GBM	0.69	0.95	0.80	0,90
	SVM	0.77	0.95	0.85	0,93
	NB	0.90	0.86	0.88	0,95
MEMORIA	RF	0.95	0.86	0.90	0,96
	GBM	0.94	0.71	0.81	0,93
	SVM	0.94	0.81	0.87	0,95
	NB	0.90	0.90	0.90	0,96
MOUSE	RF	1.00	0.95	0.98	0,99
	GBM	1.00	0.90	0.95	0,98
	SVM	1.00	1.00	1.00	1,00
	NB	0.91	0.95	0.93	0,97
TECLADO	RF	0.94	0.94	0.94	0,98
	GBM	0.94	0.94	0.94	0,98
	SVM	1.00	0.94	0.97	0,99
	NB	0.88	0.88	0.88	0,96

- SciKit-Learn base binário
  - Mouse

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,99	0,98	0,98	0,98
	BiGram	0,98	0,92	0,95	0,95
	TriGram	0,96	0,68	0,80	0,83
	Uni-Bigram	0,99	0,98	0,98	0,98
	Uni-Trigram	0,99	0,98	0,98	0,98
GBM	UniGram	0,99	0,98	0,98	0,98
	BiGram	0,98	0,91	0,95	0,95
	TriGram	0,96	0,68	0,80	0,83
	Uni-Bigram	0,99	0,98	0,98	0,98
	Uni-Trigram	0,98	0,98	0,98	0,98
SVM	UniGram	0,98	0,98	0,98	0,98
	BiGram	0,98	0,91	0,95	0,95
	TriGram	0,96	0,68	0,80	0,83
	Uni-Bigram	0,99	0,98	0,98	0,98
	Uni-Trigram	0,99	0,98	0,98	0,98
NB	UniGram	0,92	0,97	0,94	0,94
	BiGram	0,94	0,94	0,94	0,94
	TriGram	0,94	0,68	0,79	0,82
	Uni-Bigram	0,96	0,97	0,97	0,97
	Uni-Trigram	0,96	0,97	0,96	0,97

  - Teclado

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,98	0,97	0,98	0,99
	BiGram	0,96	0,85	0,90	0,95
	TriGram	0,95	0,58	0,72	0,87
	Uni-Bigram	0,98	0,96	0,97	0,98
	Uni-Trigram	0,98	0,96	0,97	0,99
GBM	UniGram	0,98	0,97	0,97	0,98
	BiGram	0,97	0,85	0,91	0,95
	TriGram	0,96	0,57	0,72	0,87
	Uni-Bigram	0,98	0,96	0,97	0,98
	Uni-Trigram	0,98	0,96	0,97	0,98
SVM	UniGram	0,98	0,97	0,98	0,99
	BiGram	0,98	0,84	0,91	0,95
	TriGram	0,97	0,57	0,72	0,87
	Uni-Bigram	0,98	0,96	0,97	0,99
	Uni-Trigram	0,99	0,96	0,97	0,99
NB	UniGram	0,96	0,81	0,88	0,94
	BiGram	0,98	0,84	0,90	0,95
	TriGram	0,95	0,57	0,71	0,87
	Uni-Bigram	0,98	0,88	0,93	0,96
	Uni-Trigram	0,97	0,88	0,93	0,96

- Memória

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,95	0,89	0,92	0,98
	BiGram	0,92	0,76	0,83	0,96
	TriGram	0,92	0,56	0,70	0,94
	Uni-Bigram	0,94	0,88	0,91	0,98
	Uni-Trigram	0,94	0,88	0,91	0,98
GBM	UniGram	0,94	0,89	0,91	0,98
	BiGram	0,92	0,76	0,83	0,96
	TriGram	0,92	0,56	0,70	0,94
	Uni-Bigram	0,94	0,89	0,91	0,98
	Uni-Trigram	0,94	0,88	0,91	0,98
SVM	UniGram	0,94	0,89	0,91	0,98
	BiGram	0,93	0,75	0,83	0,96
	TriGram	0,91	0,56	0,70	0,94
	Uni-Bigram	0,94	0,89	0,91	0,98
	Uni-Trigram	0,94	0,89	0,91	0,98
NB	UniGram	0,87	0,86	0,87	0,97
	BiGram	0,89	0,76	0,82	0,96
	TriGram	0,87	0,59	0,70	0,94
	Uni-Bigram	0,84	0,87	0,86	0,97
	Uni-Trigram	0,85	0,86	0,86	0,97

- HD

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,88	0,79	0,84	0,98
	BiGram	0,67	0,46	0,55	0,94
	TriGram	0,67	0,16	0,26	0,93
	Uni-Bigram	0,87	0,77	0,82	0,97
	Uni-Trigram	0,86	0,76	0,81	0,97
GBM	UniGram	0,86	0,80	0,83	0,98
	BiGram	0,66	0,46	0,54	0,94
	TriGram	0,68	0,17	0,27	0,93
	Uni-Bigram	0,84	0,76	0,80	0,97
	Uni-Trigram	0,84	0,76	0,80	0,97
SVM	UniGram	0,85	0,80	0,82	0,98
	BiGram	0,72	0,37	0,49	0,94
	TriGram	0,84	0,11	0,19	0,93
	Uni-Bigram	0,84	0,77	0,80	0,97
	Uni-Trigram	0,84	0,77	0,80	0,97
NB	UniGram	0,84	0,58	0,69	0,96
	BiGram	0,72	0,36	0,48	0,94
	TriGram	0,61	0,16	0,25	0,93
	Uni-Bigram	0,79	0,57	0,66	0,96
	Uni-Trigram	0,81	0,57	0,67	0,96

- Bateria

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,99	0,95	0,97	1,00
	BiGram	0,96	0,71	0,82	0,99
	TriGram	0,99	0,51	0,67	0,98
	Uni-Bigram	0,98	0,94	0,96	1,00
	Uni-Trigram	0,98	0,94	0,96	1,00
GBM	UniGram	0,97	0,94	0,95	1,00
	BiGram	0,93	0,71	0,81	0,99
	TriGram	0,99	0,50	0,67	0,98
	Uni-Bigram	0,96	0,94	0,95	1,00
	Uni-Trigram	0,96	0,94	0,95	1,00
SVM	UniGram	0,99	0,94	0,96	1,00
	BiGram	0,98	0,70	0,82	0,99
	TriGram	0,99	0,51	0,67	0,98
	Uni-Bigram	0,98	0,95	0,96	1,00
	Uni-Trigram	0,98	0,94	0,96	1,00
NB	UniGram	1,00	0,88	0,93	0,99
	BiGram	0,95	0,70	0,81	0,99
	TriGram	0,99	0,51	0,67	0,98
	Uni-Bigram	0,96	0,93	0,94	1,00
	Uni-Trigram	0,96	0,92	0,94	1,00

- SciKit-Learn base binário – Curadoria
  - Mouse

		Precision	Recall	F1-score	Accuracy
RF	UniGram	1,00	0,95	0,98	0,99
	BiGram	0,92	0,57	0,71	0,90
	TriGram	0,67	0,19	0,30	0,81
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99
GBM	UniGram	1,00	0,95	0,98	0,99
	BiGram	0,85	0,52	0,65	0,88
	TriGram	0,67	0,19	0,30	0,81
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99
SVM	UniGram	1,00	0,95	0,98	0,99
	BiGram	0,86	0,57	0,69	0,89
	TriGram	0,75	0,29	0,41	0,83
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99
NB	UniGram	0,90	0,86	0,88	0,95
	BiGram	0,82	0,67	0,74	0,90
	TriGram	0,75	0,29	0,41	0,83
	Uni-Bigram	0,91	0,95	0,93	0,97
	Uni-Trigram	0,86	0,86	0,86	0,94

- Teclado

		Precision	Recall	F1-score	Accuracy
RF	UniGram	1,00	0,94	0,97	0,99
	BiGram	0,92	0,65	0,76	0,93
	TriGram	0,33	0,06	0,10	0,82
	Uni-Bigram	0,94	0,94	0,94	0,98
	Uni-Trigram	0,94	0,94	0,94	0,98
GBM	UniGram	0,88	0,88	0,88	0,96
	BiGram	1,00	0,65	0,79	0,94
	TriGram	0,33	0,06	0,10	0,82
	Uni-Bigram	0,94	0,94	0,94	0,98
	Uni-Trigram	0,94	0,88	0,91	0,97
SVM	UniGram	1,00	0,94	0,97	0,99
	BiGram	1,00	0,65	0,79	0,94
	TriGram	0,33	0,06	0,10	0,82
	Uni-Bigram	1,00	0,94	0,97	0,99
	Uni-Trigram	1,00	0,94	0,97	0,99
NB	UniGram	0,93	0,76	0,84	0,95
	BiGram	0,71	0,59	0,65	0,89
	TriGram	0,50	0,06	0,11	0,83
	Uni-Bigram	0,88	0,82	0,85	0,95
	Uni-Trigram	0,82	0,82	0,82	0,94

- Memória

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,89	0,76	0,82	0,93
	BiGram	0,93	0,62	0,74	0,91
	TriGram	1,00	0,10	0,17	0,81
	Uni-Bigram	0,89	0,76	0,82	0,93
	Uni-Trigram	0,89	0,76	0,82	0,93
GBM	UniGram	0,89	0,76	0,82	0,93
	BiGram	0,87	0,62	0,72	0,90
	TriGram	1,00	0,10	0,17	0,81
	Uni-Bigram	0,88	0,67	0,76	0,91
	Uni-Trigram	0,88	0,71	0,79	0,92
SVM	UniGram	0,94	0,76	0,84	0,94
	BiGram	0,93	0,62	0,74	0,91
	TriGram	1,00	0,10	0,17	0,81
	Uni-Bigram	0,94	0,71	0,81	0,93
	Uni-Trigram	0,94	0,76	0,84	0,94
NB	UniGram	0,82	0,86	0,84	0,93
	BiGram	0,93	0,62	0,74	0,91
	TriGram	1,00	0,10	0,17	0,81
	Uni-Bigram	0,89	0,81	0,85	0,94
	Uni-Trigram	0,89	0,81	0,85	0,94

- HD

		Precision	Recall	F1-score	Accuracy
RF	UniGram	0,88	0,67	0,76	0,91
	BiGram	0,71	0,57	0,63	0,86
	TriGram	0,00	0,00	0,00	0,79
	Uni-Bigram	0,83	0,71	0,77	0,91
	Uni-Trigram	0,88	0,67	0,76	0,91
GBM	UniGram	0,82	0,67	0,74	0,90
	BiGram	0,74	0,67	0,70	0,88
	TriGram	0,00	0,00	0,00	0,79
	Uni-Bigram	0,88	0,71	0,79	0,92
	Uni-Trigram	0,77	0,81	0,79	0,91
SVM	UniGram	0,84	0,76	0,80	0,92
	BiGram	0,81	0,62	0,70	0,89
	TriGram	0,00	0,00	0,00	0,79
	Uni-Bigram	0,90	0,86	0,88	0,95
	Uni-Trigram	0,86	0,86	0,86	0,94
NB	UniGram	1,00	0,62	0,76	0,92
	BiGram	0,83	0,48	0,61	0,87
	TriGram	0,00	0,00	0,00	0,79
	Uni-Bigram	0,88	0,71	0,79	0,92
	Uni-Trigram	0,93	0,62	0,74	0,91

- Bateria

		Precision	Recall	F1-score	Accuracy
RF	UniGram	1,00	0,95	0,98	0,99
	BiGram	1,00	0,71	0,83	0,94
	TriGram	1,00	0,48	0,65	0,89
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99
GBM	UniGram	1,00	0,95	0,98	0,99
	BiGram	0,94	0,71	0,81	0,93
	TriGram	1,00	0,48	0,65	0,89
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99
SVM	UniGram	1,00	0,95	0,98	0,99
	BiGram	1,00	0,71	0,83	0,94
	TriGram	1,00	0,48	0,65	0,89
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99
NB	UniGram	1,00	0,95	0,98	0,99
	BiGram	1,00	0,71	0,83	0,94
	TriGram	1,00	0,48	0,65	0,89
	Uni-Bigram	1,00	0,95	0,98	0,99
	Uni-Trigram	1,00	0,95	0,98	0,99

- H2O base binário

		Precision	Recall	F1-score	Accuracy
MOUSE	Random Forest	0,94	0,95	0,94	0,94
	GBM	0,95	0,97	0,96	0,96
	SVM	0,97	0,98	0,98	0,98
	Naive Bayes	0,76	0,89	0,82	0,80
	Deep Learn	1,00	1,00	1,00	1,00
	XGBoost	0,96	0,97	0,96	0,96
TECLADO	Random Forest	0,93	0,87	0,90	0,95
	GBM	0,94	0,93	0,93	0,96
	SVM	0,80	0,99	0,88	0,93
	Naive Bayes	0,62	0,76	0,69	0,81
	Deep Learn	0,99	0,99	0,99	1,00
	XGBoost	0,95	0,93	0,94	0,97
MEMORIA	Random Forest	0,96	0,91	0,94	1,00
	GBM	0,96	0,92	0,94	1,00
	SVM	0,99	0,95	0,97	1,00
	Naive Bayes	0,69	0,89	0,78	0,98
	Deep Learn	1,00	0,99	1,00	1,00
	XGBoost	0,97	0,95	0,96	1,00
HD	Random Forest	0,74	0,79	0,77	0,96
	GBM	0,78	0,85	0,81	0,97
	SVM	0,99	0,01	0,02	0,01
	Naive Bayes	0,33	0,83	0,47	0,86
	Deep Learn	0,89	0,92	0,91	0,99
	XGBoost	0,79	0,84	0,81	0,97
BATERIA	Random Forest	0,90	0,87	0,88	0,97
	GBM	0,90	0,89	0,90	0,98
	SVM	0,89	0,91	0,90	0,98
	Naive Bayes	0,73	0,89	0,80	0,95
	Deep Learn	0,97	0,97	0,97	0,99
	XGBoost	0,90	0,91	0,90	0,98

- H2O base binário – Curadoria

		Precision	Recall	F1-score	Accuracy
MOUSE	Random Forest	0,77	0,96	0,86	0,93
	GBM	0,83	0,96	0,89	0,94
	SVM	0,79	0,76	0,78	0,90
	Naive Bayes	0,62	0,84	0,71	0,84
	Deep Learn	1,00	1,00	1,00	1,00
	XGBoost	0,86	1,00	0,92	0,96
TECLADO	Random Forest	1,00	0,76	0,87	0,96
	GBM	0,93	0,82	0,88	0,96
	SVM	0,83	0,71	0,77	0,94
	Naive Bayes	0,48	0,88	0,63	0,81
	Deep Learn	0,97	0,99	0,98	0,99
	XGBoost	0,95	0,86	0,90	0,96
MEMORIA	Random Forest	0,96	1,00	0,98	0,99
	GBM	0,96	1,00	0,98	0,99
	SVM	1,00	0,91	0,95	0,98
	Naive Bayes	1,00	1,00	1,00	1,00
	Deep Learn	1,00	1,00	1,00	1,00
	XGBoost	0,95	0,95	0,95	0,98
HD	Random Forest	1,00	0,81	0,89	0,96
	GBM	0,90	0,86	0,88	0,95
	SVM	0,94	0,76	0,84	0,94
	Naive Bayes	0,51	0,86	0,64	0,79
	Deep Learn	0,98	0,98	0,98	0,99
	XGBoost	0,65	0,81	0,72	0,86
BATERIA	Random Forest	0,83	0,83	0,83	0,92
	GBM	0,82	0,78	0,80	0,91
	SVM	0,85	0,74	0,79	0,91
	Naive Bayes	0,80	0,87	0,83	0,92
	Deep Learn	1,00	1,00	1,00	1,00
	XGBoost	0,95	0,91	0,93	0,97