UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

BRUNO FERREIRA DA SILVA

UTILIZAÇÃO DE APRENDIZAGEM DE MÁQUINA PARA CLASSIFICAÇÃO DE E-MAILS EM CATEGORIAS RELEVANTES

TRABALHO DE CONCLUSÃO DE CURSO

SÃO CARLOS
2021

BRUNO FERREIRA DA SILVA

UTILIZAÇÃO DE APRENDIZAGEM DE MÁQUINA PARA CLASSIFICAÇÃO DE E-MAILS EM CATEGORIAS RELEVANTES

Trabalho de Conclusão de Curso apresentado ao Engenharia de Computação da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Alexandre Levada
Universidade Federal de São Carlos

SÃO CARLOS
2021

Dedico:

À minha mãe, Maria Ferreira da Silva.

Ao meu pai, Severino Alves da Silva.

Às minhas irmãs: Alexandra e Vanessa.

Aos meus sobrinhos: Lucas, Isabella, Rafaella e Lara.

AGRADECIMENTOS

Primeiramente agradeço aos meus pais, Severino e Maria, que sempre tentaram me guiar em todos os momentos difíceis dessa caminhada. Eles foram os principais responsáveis pela minha formação acadêmica.

Agradeço às minhas irmãs Alexandra e Vanessa pelo apoio, convivência e aprendizado.

Agradeço imensamente à minha namorada Larissa e aos seus pais, Zilda e Celso, pela hospitalidade, paciência, carinho e por todos os momentos de descontração.

Agradeço ao professor Dr. Alexandre Levada pela orientação, oportunidade e paciência.

Agradeço também à todos os amigos de universidade que partilharam comigo alguma parcela deste tempo como aluno do curso de Bacharelado em Engenharia de Computação.

Erros são, no final das contas, fundamentos da verdade. Se um homem não sabe o que uma coisa é, já é um avanço do conhecimento saber o que ela não é. (JUNG, Carl).

RESUMO

SILVA, Bruno. UTILIZAÇÃO DE APRENDIZAGEM DE MÁQUINA PARA CLASSIFICAÇÃO DE E-MAILS EM CATEGORIAS RELEVANTES. 2021. 44 f. Trabalho de Conclusão de Curso – Engenharia de Computação, Universidade Federal de São Carlos. São Carlos, 2021.

Uma das principais ferramentas de tecnologia utilizadas atualmente para a troca de informações é o serviço de e-mail. Entretanto, o gerenciamento do alto volume de informações recebidas é um dos grandes desafios encontrados na utilização desse serviço nas instituições públicas e privadas. A classificação automatizada de textos tem sido considerada um método essencial para atender um alto volume de informações textuais que temos que lidar diariamente. É cada vez mais comum a resolução de problemas por meios eletrônicos e automáticos devido à diminuição de trabalho manual e custos. Pensando nisso, empresas que recebem solicitações de suporte por e-mail tentam reduzir o tempo de atendimento utilizando algoritmos de aprendizado de máquina para a classificação de textos enviados por e-mail. Este estudo tem como objetivo identificar a capacidade dos algoritmos de aprendizado de máquina em determinar corretamente as categorias, utilizando uma base de e-mails previamente rotulada. Para os resultados foram calculadas as médias e desvios padrão das métricas mais utilizadas em aprendizagem de máquina, assim como o tempo de execução dos quatro algoritmos utilizados. Os resultados se mostraram satisfatórios e funcionais.
Palavras-chave: Machine Learning. Text Classification. Natural Language Processing. E-mail categorization. Artificial Intelligence. Supervised Learning.

ABSTRACT

SILVA, Bruno. USE OF MACHINE LEARNING TO CLASSIFY E-MAILS IN RELEVANT CATEGORIES. 2021. 44 f. Trabalho de Conclusão de Curso – Engenharia de Computação, Universidade Federal de São Carlos. São Carlos, 2021.

One of the main technology tools currently used to exchange information is the email service. However, managing the high volume of information received is one of the major challenges encountered in using this service in public and private institutions. Automated text classification has been considered an essential method to handle a high of textual information that people have to deal with on a daily basis. Problem solving by electronic and automatic means is increasingly common due to the reduction of manual work and costs. With that in mind, companies that receive support requests via email have been trying to reduce service time by using machine learning algorithms to sort texts sent via email. This study aims to identify the ability of machine learning algorithms to correctly determine categories, using a previously labeled database. The results were calculated as means and standard deviations of the most used metrics in machine learning, as well as the execution time of the four algorithms used. The results showed themselves to be satisfactory and well functional.

Palavras-chave: Machine Learning. Text Classification. Natural Language Processing. E-mail categorization. Artificial Intelligence. Supervised Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DC	Departamento de Computação
PLN	Processamento de Linguagem Natural
AM	Aprendizagem de Máquina
IA	Inteligência Artificial
AP	Aprendizado profundo
NB	Naïve Bayes
RF	Random Forest
RL	Regressão Logística
SVM	Support Vector Machine
NLTK	Natural Language Toolkit
NLP	Natural language processing - Processamento de Linguagem Natural
WE	Word Embedding
BOW	Bag of Words
TF-IDF	Term Frequency-Inverse Document Frequency
TF-IDF-1	Term Frequency-Inverse Document Frequency Unigrama
TF-IDF-2	Term Frequency-Inverse Document Frequency Bigrama

SUMÁRIO

1 INTRODUÇÃO

O Aprendizado de Máquina (ML) é cada vez mais utilizado em diversas empresas como forma de classificar dados com o mínimo de esforço humano. Segundo estimativa da nada (a), a inteligência artificial (IA) poderá gerar investimentos de US$ 464 milhões para o Brasil em 2021. A utilização de e-mails é cada vez mais presente no cotidiano dos brasileiros. Esta aceleração na comunicação por meios eletrônicos atrai empresas para uma solução valiosa: como gerenciar um alto volume de e-mails recebidos por clientes e fornecedores.

Poucas empresas não possuem o uso de e-mails corporativos para atender seus clientes. A Serasa Experian, líder em informações e responsável pela maior base de dados da América Latina, é uma das grandes empresas que também enfrenta o desafio de gerenciar esse alto volume de e-mails. As solicitações eletrônicas seguem um padrão textual e os responsáveis pelo atendimento geralmente possuem *script* de atendimento para resolução dos problemas. Ainda assim, esta atividade demanda um tempo considerável dos colaboradores da empresa para ler, categorizar, analisar, solucionar os problemas reportados e por fim monitorar o atendimento via e-mail. Todo ano, somente em uma área da empresa contendo cinco colaboradores, são recebidos aproximadamente 38 mil e-mails. Cada e-mail é lido e distribuído para os colaboradores escolhidos aleatoriamente ou entregue aos responsáveis pela demanda. Este processo de gerenciamento manual de e-mails consome um tempo valioso da equipe e não há indicadores objetivos para este atendimento (número de solicitações, categorização do atendimento, cliente solicitante e tempo de atendimento). Há, portanto, o interesse em automatizar o processo de categorização de e-mails com o mínimo de esforço humano, para assim elevar o ganho em eficiência operacional, diminuir a possibilidade de erro humano e direcionar os esforços da empresa para uma abordagem proativa às necessidades dos clientes.

A classificação automática de textos é uma das áreas mais estudadas e utilizadas para realizar as tarefas mencionadas acima. Esta classificação permite atribuir automaticamente um rótulo previamente definido à documentos textuais. Para realizar a classificação automática de textos, podem ser utilizados sistema especialistas ou de algoritmos de aprendizado de máquina. Em um sistema especialista, são criados conjuntos de regras por meio de testes condicionais, baseados na frequência ou presença de um conjunto de palavras, que definem a categoria de um documento. Porém estas regras são difíceis de serem atualizadas e mantidas sem a presença dos especialistas deste domínio. A alternativa mais indicada e abordada neste trabalho para o contexto informado é o uso de algoritmos de aprendizado de máquina.

Há dois principais tipos de aprendizagem que podem ser tratados na Aprendizagem de Máquina (AM): supervisionada e não supervisionada. A aprendizagem supervisionada ocorre quando a partir de um conjunto de dados rotulados previamente definido deseja-se encontrar uma função que seja capaz de predizer rótulos desconhecidos. Por outro lado, na abordagem não supervisionada o conjunto de dados utilizado não possui nenhum tipo de rótulo. Neste

trabalho, foram utilizados uma base de e-mails própria do autor com uma abordagem de AM supervisionada para classificar os e-mails em categorias relevantes.

O objetivo dos algoritmos de aprendizado de máquina é aprender, generalizar e extrair padrões ou características das classes das coleções com base nos documentos textuais e rótulos. Apesar dos rótulos serem definidos com a intervenção humana, os modelos de classificação de textos com o uso de aprendizado de máquina exigem um menor esforço, são mais rápidos de ser construídos e podem ser utilizados em diferentes aplicações.

Para atingir uma alta performance, o classificador deve considerar os dados mais recentes de treinamento e distinguir os textos do título e corpo de e-mail. Esta distinção é relevante porque tem uma distribuição diferente de conteúdo e, portanto, provavelmente podem ser tratados de modo distinto na modelagem.

1.1 Objetivo Geral

O objetivo principal deste trabalho é desenvolver uma abordagem computacional para classificar automaticamente os documentos de texto oriundos de mensagens eletrônicas, em categorias relevantes utilizando aprendizado de máquina.

1.2 Objetivo Específico

Os objetivos específicos são:

- Rotular manualmente uma base de e-mails para o estudo;
- Pesquisar o estado da arte sobre as técnicas que utilizam aprendizagem de máquina para categorização de texto;
- Aplicar técnicas de pré-processamento eficazes às classificações textuais;
- Treinar e testar os dados utilizando quatro modelos de Aprendizado de Máquina para classificação textual;
- Calcular as métricas de avaliação para todos os modelos utilizados;
- Analisar e comparar os resultados, observando quão precisos os algoritmos de aprendizado de máquina são para classificação de documentos como os tratados neste trabalho.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma: No Capítulo 1 apresentamos a definição do problema, os objetivos e alguns conceitos básicos sobre classificação de documentos. No Capítulo 2 vemos detalhes sobre os últimos trabalhos, o estado da arte no que diz respeito ao Processamento Natural de Linguagem, Inteligência Artificial e Aprendizagem de Máquina. No Capítulo 3 são apresentadas as metodologias abordadas na pesquisa. No Capítulo 4 inicialmente são discutidos os testes e realizado uma avaliação e comparativa dos resultados obtidos.

No Capítulo 5 descrevemos as conclusões deste trabalho, comentamos suas contribuições e sugerimos direções futuras de pesquisa.

2 REVISÃO DE LITERATURA

Este capítulo apresenta os principais conceitos de mineração de textos e inteligência artificial abrangendo o processamento e preparação dos textos, os algoritmos de classificação utilizados e os métodos de avaliação dos resultados gerados. O levantamento do estado da arte leva em consideração artigos envolvendo classificação automática de textos e classificação com grande número de classes.

2.1 Inteligência Artificial

Inteligência Artificial (IA) é uma ciência tecnológica, que pesquisa e desenvolve métodos, técnicas e aplicações para simular, estender e expandir a teoria da inteligência humana. IA é um ramo da ciência da computação que tem como objetivo entender a essência da inteligência e produzir uma nova máquina inteligente capaz de ter reações similares à inteligência humana. As pesquisas no campo da inteligência artificial incluem robótica, reconhecimento de fala, reconhecimento de imagem, processamento de linguagem natural e sistemas especialistas (NING; YAN, 2010). De acordo com o autor Islam et al. (2009) "é um sistema computacional que possui conhecimento e comportamento humano com habilidades como por exemplo: aprender, inferir, julgar, resolver o problema, memória, conhecimento e entender a linguagem natural humana. Com base no trabalho dos autores Hosea, Harikrishnan e Rajkumar (2011), Inteligência Artificial inclui:

- Jogos eletrônicos: programar computadores para jogar jogos como xadrez e damas, os quais demandam algoritmos baseados em raciocínio lógico;
- Sistemas especialistas: programar computadores para tomar decisões em situações da vida real (por exemplo, alguns sistemas especialistas ajudam médicos a diagnosticarem doenças baseados em sintomas);
- Linguagem natural: programar computadores para entender a linguagem natural humana;
- Redes neurais: sistemas que simulam inteligência pela tentativa de reproduzir os tipos de conexões físicas que ocorrem em cérebros de animais, e
- Robótica: programar computadores para perceber e reagir a outros estímulos sensoriais

Para os pesquisadores da Inteligência Artificial, a mente humana funciona como um computador, e por isso o estudo dos programas computacionais é a chave para se compreender alguma coisa acerca de nossas atividades mentais. Podemos construir programas que imitem nossa capacidade de raciocinar, de perceber o mundo e identificar objetos que estão à nossa volta, e até mesmo de falar e de compreender nossa linguagem (TEIXEIRA, 2019).

Segundo Kaufman (2019) a inteligência artificial refere-se a um campo de conhecimento associado à linguagem e à inteligência, ao raciocínio, à aprendizagem e à resolução de problemas. A IA propicia a simbiose entre o humano e a máquina ao acoplar sistemas inteligentes artificiais ao

corpo humano (prótese cerebral, braço biônico, células artificiais, joelho inteligente e similares), e a interação entre o humano e a máquina como duas "espécies"distintas conectadas (homem-aplicativos, homem-algoritmos de IA). Tema de pesquisa em diversas áreas - Computação, Linguística, Filosofia, Matemática, Neurociência, entre outras -, a diversidade de subcampos e atividades, pesquisas e experimentações, dificulta descrever o estado da arte atual. Os estágios de desenvolvimento bem como as expectativas variam entre os campos e suas aplicações, que incluem os veículos autônomos, reconhecimento de voz, games, robótica, tradução de linguagem natural, diagnósticos médicos, assim por diante.

2.2 Processamento de Linguagem Natural

Processamento de Linguagem Natural (PNL) consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em alguma linguagem natural. Há três problemas principais relacionados à construção de programas entendem a linguagem natural: o primeiro está relacionado com o pensamento, o segundo com a representação e significado das entradas linguísticas, e o terceiro está relacionado com o conhecimento mundial. Dessa forma, um sistema PNL pode começar no nível das palavras – através da determinação da origem e estrutura – então pode passar para o nível das sentenças – para determinar a ordem das palavras, gramática e significado das sentenças – então para o contexto do ambiente geral. Em um determinado contexto, uma palavra pode ter um significado e pode estar relacionada a diversas outras (CHOWDHURY, 2003a). De acordo com Feldman (1999), a distinção entre os sete níveis utilizados por pessoas para extrair significado de textos ou linguagem escrita é fundamental para o entendimento da Linguagem Natural:

- Nível Fonético ou Fonológico: o qual trata da pronúncia;
- Morfológico: que trata das menores partes da palavra que carregam significado;
- Léxico: que interpreta os significados de palavras individuais;
- Sintático: que trata da estrutura gramatical das sentenças;
- Semântico: que trata de revelar os significados das palavras em diversos níveis;
- Discursivo: que trata da estrutura dos diferentes tipos de texto;
- Pragmático: que lida com o conhecimento advindo do mundo fora do conteúdo dos documentos;

Chowdhury (2003b) divide os trabalhos e pesquisas relacionados ao processamento da linguagem natural em três categorias:

- Análise léxica e morfológica;
- Semântica e análise do discurso;
- Abordagens baseadas no conhecimento e ferramentas para PNL

Para que textos possam ser aplicados a um classificador, é necessário submetê-los a uma etapa de pré-processamento e depois representá-los numericamente. Com a representação numérica, é possível utilizar algoritmos de aprendizado de máquina para dividir os e-mails em categorias diferentes categorias. A Figura 1 apresenta este processo em um diagrama de blocos.

Figura 1 – Diagrama de blocos de um categorizador de e-mails

Fonte: O Autor

2.3 Aprendizado de Máquina

A palavra aprender em sua origem latina é composta por "ad" que significa junto e por "prae" (à frente) + "hendere" relacionado a hera, uma planta trepadeira que se prende as paredes para poder crescer, seu sentido original era o de levar para junto de si metaforicamente levar para junto da memória (HARPER et al., 2001).

O processo da aprendizagem, fundamental para a adaptação ao meio em que os seres vivos estão expostos, é resultado de interações entre estruturas mentais e o meio ambiente. No ramo da psicologia, estudiosos utilizam diversas abordagens para representar os conceitos de aprendizagem dos seres humanos. Segundo Skinner (1950), "aprendizagem é uma mudança na probabilidade de uma resposta específica". Este conceito pode ser aplicado para seres vivos e também para máquinas, assim como seres vivos respondem a estímulos do meio, um dispositivo artificial pode ser capaz de retornar respostas a entrada de dados. Uma calculadora, por exemplo é um dispositivo que retorna uma resposta a partir da inserção de dados, porém não gostaríamos que a calculadora aprendesse a realizar as contas de uma maneira diferente retornando resultados diferentes, mas sim que ele aprendesse de maneira autônoma novos cálculos que são capazes de solucionar problemas antigos apurando a probabilidade da solução.

A aprendizagem de máquina, assim como outras disciplinas, possui aspectos teóricos e empíricos. Contudo, o componente empírico sobressai durante o estudo dos algoritmos, sustentando-se no fato de que a maioria dos algoritmos de aprendizagem são muito complexos para uma análise formal. Experimentos envolvem sistematicamente variar uma ou mais variáveis independentes e examinar seus efeitos nas variáveis dependentes, assim realizam-se diversas iterações da etapa de aprendizagem para que sejam medidos os aspectos do comportamento do algoritmo sob diferentes condições (LANGLEY, 1988).

Segundo Langley (1988), a noção de aumento de performance são base de muitas definições de aprendizado, assim várias medidas de performance são variáveis dependentes naturais para os experimentos de aprendizagem de máquina, assim como para o estudo do aprendizado em humanos. O princípio para a avaliação de qualquer método de aprendizagem de máquina são as medidas de performance, pois outras medidas, como por exemplo a de entendibilidade do método, podem ser utilizadas de forma informativa, porém não relevante, observando-se que em alguns casos métodos intuitivamente plausíveis de aprendizagem resultam

em piores performances.

2.3.1 Aprendizagem Supervisionada

Aprendizagem de máquina indutiva ou supervisionada consiste na criação de um classificador capaz de aprender a partir de um conjunto de treinamento e generalizar para as novas instâncias que aparecem (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2007). Durante a classificação supervisionada são apresentados os possíveis resultados recolhidos através de observação, dessa forma ele possuirá um escopo limitado e pré-definido de resultados que serão utilizados como parâmetro e, principalmente, como referência durante a classificação.

Classificação e regressão são dois tipos principais de problemas de aprendizado de máquina supervisionados (GUIDO; MüLLER, 2016). Uma forma fácil de distinguir tarefas de classificação e regressão é perguntar se existe algum tipo de continuidade na saída. Se houver continuidade entre os resultados possíveis, é um problema de regressão.

2.3.1.1 Classificação

Classificação é uma tarefa, na qual as características de um determinado objeto são analisadas e atribuídas a um conjunto predefinido de classes. Nesta tarefe, os objetos a serem classificados são representados por registros em uma tabela de banco de dados ou em um arquivo, e o ato de classificação consiste em rotular os registros em uma determinada classe (BERRY; LINOFF, 2004).

Alguns exemplos de classificação são:

- Classificação de risco de crédito;
- Reconhecimento facial;
- Reconhecimento de fraudes em sistemas;
- Detecção de sintomas de doenças;

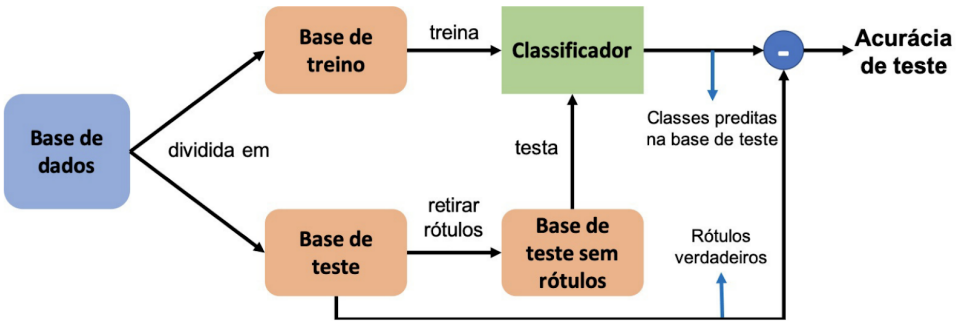
De acordo com Soares (2005), realiza-se o processo de classificação para encontrar relacionamento entre os atributos preditivos e objetivo, tendo assim um conhecimento que seja capaz de antever a classe de um registro ainda não classificado.

Segundo Petermann et al. (2006), o processo de classificação é composto por duas etapas. A primeira é o aprendizado ou treinamento, que é executada com uma base de já classificada. A segunda etapa é a de teste, no qual se utilizam registros que não foram utilizados no treinamento. Basicamente, é feito o mapeamento das entradas e saídas para posteriormente serem utilizadas na tomada de decisão em novos registros, ainda não classificados.

Ainda segundo Petermann et al. (2006), o primeiro passo de classificação é fundamental, pois através dele é possível construir um modelo de dados pré-definidos, através da análise de tuplas de um banco de dados, no qual cada uma dessas tuplas é pertencente de uma classe. O segundo passo é essencial para verificar se os o modelo construído na etapa de treinamento foi suficientemente bom para categorizar, com alto índice de precisão, o conjunto de dados não utilizado na etapa de treinamento. Esta etapa envolve então a predição de classes: os exemplos

da base de teste são apresentados para o modelo treinado para que este realize a predição de suas classes. Ao comparar as classes preditas com as classes verdadeiras da base de teste, é possível medir sua capacidade em classificar corretamente exemplos não vistos durante o treinamento. Pode-se resumir este fluxo pela figura 2

Figura 2 – Fluxo resumido de problemas de Classificação



Fonte: Escovedo et al. (2020)

2.3.1.2 Regressão

Tarefas de regressão tem como objetivo prever um número contínuo, ou um número de pontos flutuantes em termos de programação (ou números reais em termos matemáticos). Um exemplo de tarefa de regressão é prever a renda anual da pessoa a partir do local onde mora, nível de escolaridade, idade e profissão.

Os algoritmos de aprendizagem que relacionam um conjunto de atributos de entrada com uma ou mais saídas contínuas, que podem assumir qualquer valor dentro de um intervalo, são chamados de algoritmos de regressão. O exemplo mais simples deste tipo de modelo é o simples ajuste de pontos a uma curva $y = f(x)$, onde pode-se associar um único atributo x com uma saída y. Porém, assim como no caso da classificação, pode ser necessário associar a saída com uma série de atributos em um vetor x (FONTANA,).

2.3.2 Aprendizagem Não Supervisionada

Os casos em que permanece a necessidade de análise dos dados mesmo sem eles estarem rotulados são considerados de aprendizado não supervisionado, onde se aprende sem um professor. Desconhecer a "resposta certa" é o melhor caminho para o aprendizado não supervisionado (DAUMÉ III, 2012).

A classificação através do método não supervisionado não apresenta exemplos da saída desejada, dessa forma o algoritmo deve possuir recursos para superar essa falta de respostas através da apresentação dos resultados classificados de acordo com categorias formuladas por ele próprio.

2.3.2.1 Clusterização ou Agrupamento

Para a extração de conhecimento de uma base de dados é fundamental separar estes dados em forma de grupos (*clusters*) que possuam algum significado relevante para a análise. A criação destes grupos é necessária para que se realize uma melhor investigação do grande volume de dados.

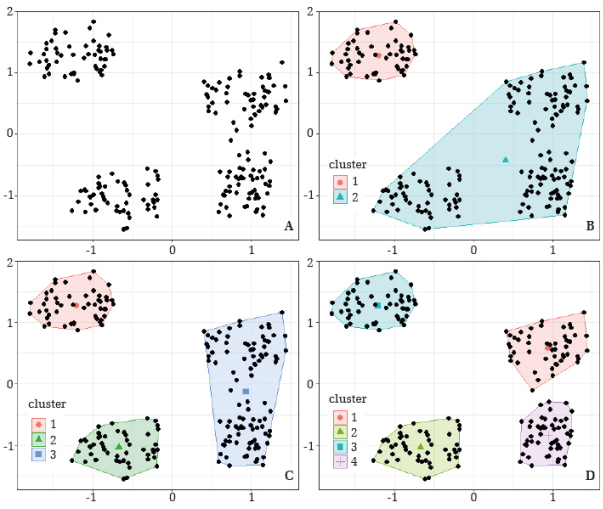
A Clusterização é o método de identificação de grupos de dados semelhantes em um conjunto de dados. A tarefa de dividir a população ou os pontos de dados em vários grupos, de modo que os pontos de dados nos mesmos grupos sejam mais semelhantes a outros pontos de dados no mesmo grupo do que os de outros grupos é chamada de clusterização. O objetivo desta abordagem é segregar grupos com traços semelhantes e atribuí-los a clusters.

De um modo geral, clusterização pode ser dividido em dois subgrupos:

- Cluster Rígido: No cluster rígido, cada ponto de dados ou pertence a um cluster completamente ou não.
- Cluster flexível: No cluster flexível, em vez de colocar cada ponto de dados em um cluster separado, uma probabilidade ou probabilidade de que o ponto de dados esteja nesses clusters é atribuída.

A aplicação da técnica de agrupamento normalmente ocorre quando deseja-se realizar uma análise estatística ou a generalização dos dados de forma exploratória. Logo, esta é uma técnica necessária a análise de informações de mesmas características sem a presença de informações irrelevantes. Neste cenário, tanto a similaridade entre os termos dos grupos quanto a dissimilaridade devem ser investigadas.

Figura 3 – (A) Gráfico de dispersão de dados não categorizados. (B) Agrupamento utilizando o algoritmo K-means com 2 clusters (k=2). (C) Agrupamento utilizando 3 clusters (k=3). (D). Agrupamento utilizando 4 clusters (k=4)



Fonte: FERNANDES Fernando Timoteo; CHIAVEGATTO FILHO (2019)

Na figura 3 (A), o grupo apresenta o conjunto de dados originais como entrada. No

exemplo (B), são formados dois grupos, em (C) três grupos e, por fim, em (D) são formados quatro grupos. Os grupos foram formados de acordo com os critérios estabelecidos e as distâncias euclidianas entre os termos.

K-Means e DBScan são exemplos de algoritmos que utilizam o método de clusterização.

2.3.3 Aprendizagem por Reforço

Métodos de aprendizagem por reforço (SUTTON; BARTO et al., 1998) tratam de situações onde um agente aprende por tentativa e erro ao atuar sobre um ambiente dinâmico. Desta maneira, não é necessária uma entidade externa que forneça exemplos ou um modelo a respeito da tarefa a ser executada: a única fonte de aprendizado é a própria experiência do agente, cujo objetivo formal é adquirir uma política de ações que maximize seu desempenho geral (MONTEIRO; RIBEIRO, 2004).

Segundo Ribeiro (1999), a aprendizagem por reforço é um paradigma computacional de aprendizagem em que um agente aprendiz procura maximizar uma medida de desempenho baseada nos reforços (recompensas ou punições) que recebe ao interagir com um ambiente desconhecido .

2.4 Mineração de Textos

A mineração de dados ou descoberta de conhecimento em bancos de dados, conhecida como KDD (*Knowledge Discovery in Databases*), trata da descoberta de informação útil presente em grandes bases de dados através da identificação de regras e padrões nesses conjuntos. Na maioria das vezes, esses conjuntos possuem especificidades, logo, as ferramentas de mineração devem ser utilizadas de forma interativa e não automática gerando diversas etapas para o processo, como as listadas (FAYYAD et al., 1996):

- Identificar o Domínio
- Preparação dos dados
- Mineração
- Processamento dos resultados
- Utilização dos resultados

A aprendizagem de máquina durante a mineração de dados acaba sendo combinada com as áreas de estatística e bancos de dados. Apesar de haver controvérsias e considerações de similaridade entre as duas, pode-se citar como diferença da aprendizagem de máquina e KDD que na primeira área há uma busca pelas estruturas que estão por trás dos dados e contribuem para sua formação, enquanto no KDD os dados são a base para o estudo e o interesse nos resultados não depende de estruturas por trás desses dados. Além disso apesar da aprendizagem de máquina ser parte do núcleo da mineração de dados elas são coisas diferentes (FAYYAD et al., 1996).

2.5 Categorização de Textos

Textos são uma forma natural para armazenar informação. Por outro lado, sua mineração é uma tarefa difícil e multidisciplinar que possui potencial comercial. As disciplinas estudadas nesse trabalho serão as relacionadas especificamente com a classificação dos textos.

Pesquisadores da área de aprendizagem de máquina e recuperação de informação e desenvolvedores de aplicações são profissionais que precisam trabalhar com grandes quantidades de documentos, por isso a classificação de textos é objeto de interesse para ambas as classes. Tal inclinação é potencializada pela conectividade aumentada e disponibilidade de diversas bases de dados contendo documentos de diferentes tipos e com variadas quantidades de informação.

A recuperação de informação e pesquisa em bases de dados com grande quantidade de textos pode ter duas aproximações, na primeira constroem-se ferramentas robustas para a pesquisa nessas bases ou na segunda solução constroem-se ferramentas robustas para estruturar esses dados tornando a pesquisa mais simples (SEBASTIANI, 2005).

Existem duas variações da categorização de textos: clusterização e classificação. A categorização (*clustering*) é uma das técnicas mais utilizadas no processo de mineração de dados para descobrir grupos e identificar distribuições de padrões ocultos em uma base de dados. Esta técnica permite agrupar documentos similares em coleções de texto sem que se tenham informações prévias sobre os grupos, portanto não há classes ou rótulos previamente definidos para o treinamento de um modelo. Porém a categorização de textos por meio da classificação ou categorização de documentos consistem em classificar um documento em um conjunto pré-especificado de categorias. Assim, dado um conjunto de categorias, assuntos ou tópicos e uma coleção de documentos de texto, a classificação é o processo de encontrar a categoria correta associada a cada documento. (FELDMAN; SANGER; PRESS, 2007)

Clustering é o processo de examinar uma coleção de "pontos"e agrupar os pontos em "clusters" de acordo com alguma medida de distância. O objetivo é que pontos no mesmo cluster têm uma pequena distância um do outro, enquanto pontos em diferentes clusters estão a uma grande distância um do outro (RAJARAMAN; ULLMAN, 2011). De acordo com Hotho, Nürnberger e Paaß (2005), a clusterização pode ser utilizada para a formação de grupos de documentos com conteúdo similar, os chamados clusters, com conteúdo mais similar dentro do cluster e menos similar entre os diferentes clusters.

Segundo Sebastiani (2005), inicialmente, em meados da década de 70, as aplicações construídas objetivavam a indexação automática de textos para sistemas de recuperação de informações booleanas. Tendo em vista os termos de indexação serem predefinidos, podemos considerar uma forma de classificação onde utilizamos analogamente tais expressões como classes dos sistemas atuais. Com o crescimento de importância e quantidade de documentos nas décadas de 80 e 90, surgiram aplicações de classificação de texto como os filtros de notícias, classificação de patentes e classificação de páginas WEB.

Se o objetivo do estudo é determinar em qual categoria os documentos melhor se enquadram, e já existir um conjunto de dados consistente previamente categorizado, costuma-se

utilizar a abordagem de classificação, ao invés da clusterização. Há duas abordagens para realizar a classificação de documentos: uma é centrada no conhecimento de especialistas, onde é desenvolvida uma solução em forma de regras de classificação, e outra é pela aprendizagem de máquina, na qual um processo indutivo executa um classificador com base no aprendizado obtido a partir do conjunto de dados previamente categorizado.

2.6 Seleção de Atributos

Na dinâmica da aprendizagem de máquinas, o desafio de manter o foco nas informações mais relevantes diante de uma quantidade de dados muito grande se tornou uma importante tarefa. Há grande quantidade de atributos em exemplos encontrados geralmente no material científico ou corporativo, que é mais comum no contexto da aprendizagem de máquina. Diante do avanço da Web, o processo foi inundado com material de baixa qualidade de informação. Neste contexto, o foco na seleção dos atributos mais relevantes para a representação dos dados se torna cada vez mais importante (BLUM; LANGLEY, 1997).

A literatura da aprendizagem de máquina apresenta diferentes conceitos de relevância por que esta geralmente depende do objetivo estudado, na sequência 5 conceitos de relevância que são importantes na tarefa de seleção de atributos (BLUM; LANGLEY, 1997).

- Relevante ao alvo: ao alterar o valor do atributo para determinado documento alvo então a classe a qual estava atribuído também é alterada.
- Fortemente relevante à amostra: o atributo deve necessariamente estar presente na amostra que contém as classes A e B. É semelhante ao relevante ao alvo, mas relacionado a amostra.
- Fracamente relevante à amostra: é considerado fracamente relevante a amostra caso possa ser removido um subconjunto de atributos e ele se torne fortemente relevante.
- Relevante como uma medida de complexidade: compões o menor número de atributos necessários para atingir a ótima performance.
- Útil de forma incremental: caso produza uma melhora na acurácia quando inserido no subconjunto de atributos.

Métodos de filtragem, que não dependem do algoritmo que será utilizado para classificação, geralmente são utilizados em tarefas de classificação de textos. Nesse processo, para remover atributos irrelevantes e selecionar atributos importantes são utilizadas as características do próprio conjunto de treinamento (BLUM; LANGLEY, 1997).

Tratar cada palavra como um atributo é a forma mais simples de indexação dos atributos selecionados em textos, porém a existência de sinônimos ou significados múltiplos de uma palavra complica sua indexação, por isso surgiram métodos mais complexos para seleção de atributos em textos, como a indexação de frases. Nesta última, atributos são extraídos a partir da presença de uma ou mais palavras em um trecho sintático do texto ou o método de clusterização de termos, onde grupos de atributos podem ser substituídos por um único atributo correspondente a sua soma lógica ou numérica (LEWIS et al., 1997).

Técnicas como a apresentada por Ning e Yan (2010) buscam a redução da necessidade de dados pré classificados, muitas vezes escassos, durante a fase de seleção de atributos através de Maximização de expectativa ou EM (*Expectation-Maximization*). A maximização de expectativa consiste na utilização de documentos não classificados ou incompletos, visto que os documentos não classificados carecem de classe, em conjunto com documentos rotulados para aumentar a posterior eficiência do classificador. Para que os documentos não rotulados não acabem prejudicando a etapa de treinamento, é proposto no trabalho citado a inclusão de um fator peso nesses documentos. Outra proposição para aumentar a eficiência da técnica é modelar as classes de acordo com uma mistura múltipla de componentes. Apesar do uso da seleção de atributos não ser exclusiva dos problemas de classificação, principalmente na classificação faz sentido a utilização de classes para supervisionar essa etapa para assegurar que os atributos que melhor se relacionam as classes sejam selecionados (AGGARWAL; ZHAI, 2012).

2.7 Pré-processamento de Textos

A etapa de pré-processamento de textos que consiste em transformar documentos textuais em um formato estruturado, tal como uma tabela atributo-valor, para que possa ser aplicado algoritmos de aprendizado de máquina para extrair conhecimento dessa informação textual (MITCHELL et al., 1997). Porém, essa transformação é um processo custoso e demorado que deve ser feito com cuidado para que o conhecimento adquirido posteriormente seja útil para o usuário final. Esta etapa de pré-processamento é necessária para aumentar a precisão dos algoritmos de AM aplicados a categorizações textuais.

Sistemas de Aprendizado de Máquina raramente conseguem trabalhar diretamente com informações digitais dispersas em documentos textuais devido ao formato não estruturado desses textos. Desse modo, torna-se evidente a necessidade de desenvolver sistemas que possam tratar essa informação não estruturada e transformá-las em informação estruturada, especificamente uma tabela atributo-valor, para possibilitar o uso de sistemas de aprendizado já existentes.

A abordagem *bag of words*, detalhada na subseção 2.8.1, é uma das mais utilizadas em AM para transformar dados estruturados em uma representação atributo-valor, na qual a frequência das palavras (termos), independentes do seu contexto ou significado, são contadas. A partir da contagem é gerada a tabela cujas entradas contém informações relacionadas à frequência de cada palavra.

Na transformação de documentos textuais em tabelas atributo-valor, existem alguns métodos para auxiliar na redução do número de atributos (redução de dimensionalidade) visando melhorar a relevância da informação para a classificação do texto. Esses métodos são brevemente descritos a seguir.

2.7.1 Tokenização

Ao analisar-se a quantidade de elementos que um texto contém pode-se identificar diversos componentes recorrentes sejam eles palavras, pontuação, ou símbolos, durante o processo de tokenização (em inglês, tokenizing) estes elementos são separados e listados, apenas palavras são interessantes para os processos posteriores logo pontuação, espaços ou delimitadores de tabulação servem como separadores durante a formação dos tokens, que são instâncias compostas por uma sequência de caracteres em um documento em particular que é agrupada na forma de uma unidade semântica útil para o processamento (DEBARR; WECHSLER, 2009).

2.7.2 Remoção de stopwords

Algumas palavras não oferecem relevância durante o processo de classificação e devem ser removidas na etapa de pré-processamento. São conectores, preposições, artigos, pronomes e palavras comuns de grande ocorrência e com pouco valor para a classificação que constituem uma lista dos tokens com pouca relevância. Esses tokens variam de acordo com a língua do texto e servem como parâmetro para a formação do dicionário de palavras definitivas para as demais etapas.

A remoção das *stopwords* pode liberar espaço nos vetores e, em alguns casos, melhorar a performance do classificador ajudando na recuperação de informação. A maioria dos termos utilizados são relativamente óbvios e pertencem a um conjunto não muito amplo de palavras, contudo é importante ressaltar que algumas palavras que aparecem nos documentos têm maior valor sintático do que o valor semântico que interessa para o classificador. Pode-se considerar uma stopword aquela encontrada quando se consulta um documento, mas não é relevante se a consulta está ou não relacionada com o documento (WILBUR; SIROTKIN, 1992).

2.7.3 Stemming

O algoritmo responsável pelo stemming remove as variações de uma palavra através da redução para uma mesma raiz ou uma forma comum. Tal ação é considerada de grande importância pelos pesquisadores da área de recuperação de informações, pois palavras que possuem variações pequenas, como tempo verbal ou mudanças entre plural e singular, ao passarem por essa transformação para que seja reduzida a ocorrências de tokens com sentidos similares, aumentam a eficiência da atribuição dos pesos para os termos (LOVINS, 1968).

Na análise morfológica automática, os sufixos da palavra podem ter maior interesse imediato que a raiz da palavra, assim como a contagem da frequência dos termos pode ser relevante para análise matemática ou estatística de um corpus, pois requerem expressões idênticas. Contudo, alguns problemas linguísticos são comuns para qualquer algoritmo que realize stemming não importando sua finalidade (LOVINS, 1968). É importante ressaltar que, de acordo com o algoritmo utilizado para stemming, o resultado pode ser um termo sem

validade gramatical, pois podem ser cortados letras do final ou reduzidos a partes da palavra original, tornando-a sem sentido para leitura.

Como exemplo, ao aplicar *stemming* nas palavras "química", "químico" ou "químicos", o algoritmo converte-as para o *stemm* "químic". Esta técnica permite reduzir a dimensionalidade sem perder informações relevantes do conjunto de dados inicial.

2.8 Conversão de Valores Simbólicos para Numéricos

A manipulação de tipos de atributos diferentes, numéricos e simbólicos, depende da capacidade e necessidade da técnica de aprendizado de máquina utilizada, já que algumas técnicas, como árvores de decisão, podem manipular valores simbólicos enquanto outras, tais como redes neurais, podem manipular somente uma representação numérica de um valor simbólico (NEVES, 2003).

Devido ao fato de todas as técnicas de mineração de dados, ou de aprendizagem de máquina, poderem manipular dados numéricos, mas algumas não poderem manipular dados simbólicos, torna-se preciso aplicar algum método de transformação de valores simbólicos em uma representação numérica apropriada (PYLE, 1999).

Nas subseções seguintes são apresentadas abordagens frequentemente utilizadas para converter palavras em dados numéricos.

2.8.1 Bag of Words

Uma técnica simples e popular usada para representação de texto, em Processamento de Linguagem Natural (PLN), é o modelo do *Bag of Words* (Saco de Palavras). O modelo Bag of words (BOW) é comumente usado em métodos de classificação de documentos onde a (frequência de) ocorrência de cada palavra é usada como um recurso para treinar um classificador (MCTEAR; CALLEJAS; GRIOL, 2016).

O BOW utiliza um conjunto de palavras do texto como entrada nos algoritmos classificadores, sendo cada palavra diferente um atributo do conjunto de dados formado. A instância mais simples de BOW é a seleção de todas as palavras do texto, tornando a dimensionalidade do conjunto de dados, e consequentemente do problema, igual à quantidade de palavras diferentes presente no texto. Desconsideram-se do texto a gramática e até mesmo a ordem das palavras.

2.8.2 TF-IDF

Uma das medidas utilizadas, em mineração de textos e em aprendizado de máquina, para dar peso às palavras presentes nos documentos de uma coleção é a TF-IDF (*Term Frequency-Inverse Document Frequency*, proposta por Jones (1972). Nesta medida, a importância de uma palavra é mensurada de acordo com a frequência com que essa palavra aparece em um documento, logo quanto mais frequente for uma palavra em um documento, maior sua

importância. Entretanto, se, na coleção como um todo, aquela palavra for muito frequente, ela não é tão relevante assim. Portanto, essa medida destaca a pouca frequência do termo (IDF) na coleção e a frequência dele (TF) no documento. Dado um termo t, a frequência do termo é calculada por:

$$Tf(t) = \frac{Número\,de\,ocorrências\,do\,termo\,t\,no\,documento}{Quantidade\,de\,termos\,no\,documento} \quad (1)$$

Essa divisão pelo número de termos no documento se deve ao fato de ser mais provável um número maior de ocorrências de um termo em um texto grande. Assim, é feita essa normalização, calculando-se a frequência relativa. O IDF é uma medida da importância do termo na coleção. Termos muito frequentes têm seu peso diminuído, enquanto o inverso ocorre para termos raros. O IDF é dado por:

$$idf(t) = log_e \frac{Número\,total\,de\,documentos}{Nmero\,de\,documentos\,que\,possuem\,o\,termo\,t} \quad (2)$$

2.8.3 Atributos N-Gramas

Segundo VILELA (2011), um n-grama é uma sequência de n itens dentro de uma frase. Os itens podem ser palavras, letras, sílabas, classificação gramatical das palavras, ou qualquer outra base. Um n-grama de tamanho 1 é chamado de unigrama, de tamanho de 2, de bigrama, de tamanho 3 é chamado de trigrama, de 4 em diante é n-grama . Para uma sequência de palavras, por exemplo "Departamento de Computação da UFSCar", um bigrama de palavras seria: "# Departamento", "Departamento de", "de Computação", "Computação da", "da UFSCar" e "UFSCar #".

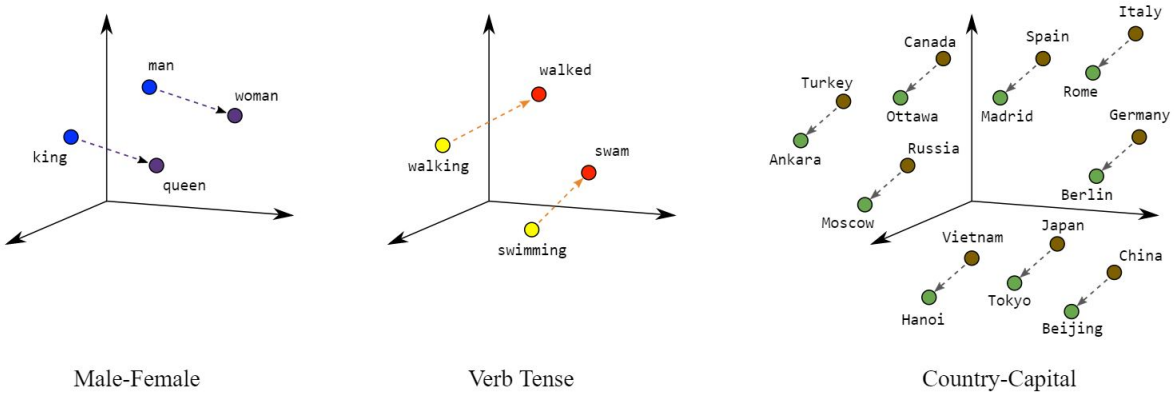
A obtenção de tabelas atributo-valor baseadas em bag-of-words é um dos métodos mais simples de descrição de textos. A maior crítica relacionada a essa representação é o fato de ela considerar cada palavra isoladamente, não conseguindo capturar bem os conceitos que estão expressos em termos compostos por mais de uma palavra. Desse modo, foram propostas descrições baseadas em n-gramas, ou seja, grupos de n palavras consecutivas que são encontradas nos textos. Os unigramas (n=1) constituem os mesmos atributos obtidos pelo método bag-of-words (BRAGA, 2010).

2.8.4 Word Embedding

Word Embedding (WE) é uma abordagem moderna utilizada para representar textos no processamento de linguagem natural, porém no WE cada palavra é representada por um vetor numérico, diferente do BOW que representa cada palavra por um número. O método WE consiste em uma abordagem capaz de identificar as informações semânticas latentes da linguagem, capturando a concorrência de padrões das palavras (MIKOLOV et al., 2013), o qual permite a redução da alta dimensionalidade nos dados e o raciocínio sobre o uso e significado das palavras (SHUANG et al., 2019).

Este algoritmo é semelhante ao *Bag of Words*, na qual as palavras são representadas em vetores grandes e esparsos, porém o WE utiliza vetores densos de tamanho fixo que são capazes de armazenar informações sobre o contexto e significado dos documentos. Cada palavra é representada por um ponto em um espaço multidimensional chamado de *embedding space*. Durante o treinamento, as palavras são definidas por elas mesmas e pelas palavras que a acompanham. Na Figura 4, por exemplo, as seguintes visualizações de *embeddings* reais mostram relações geométricas que capturam relações semânticas como a relação entre os gêneros das palavras (masculino e feminino), entre os tempos verbais entre os países e suas capitais.

Figura 4 – Exemplos de uma abordagem Word Embedding



Fonte: GOOGLEDEVELOPERS (2021)

Segundo Junior (2007) apesar de a identificação de tokens ser uma tarefa simples para o ser humano, ela pode se tornar complexa quando executada por um computador pelo fato de os delimitadores assumirem diferentes papéis. Por exemplo o "ponto", que é utilizado para marcar o fim de uma sentença ou na composição de uma abreviatura. Uma sentença pode ter seus vocábulos dispostos em diferentes ordens as quais podem até mesmo modificar seu significado, são consideradas colocações as palavras que apresentam essa relação. Por exemplo "provocar essa destruição" e "essa destruição foi provocada", por isso é interessante que a tokenização seja composta por todo o conjunto de palavras que podem traduzir uma ideia diferente (SOARES et al., 2008).

2.9 Máquina de Vetores de Suporte (SVM)

As Máquinas de Vetores de Suporte (SVM) são fundamentadas na Teoria do Aprendizado Estatístico, proposta por VAPNIK (1995), cujo objetivo é encontrar um classificador particular com bom desempenho entre todos os classificadores gerados no processo de treinamento. Esta técnica de aprendizado de máquina (ML) é do tipo aprendizado supervisionado.

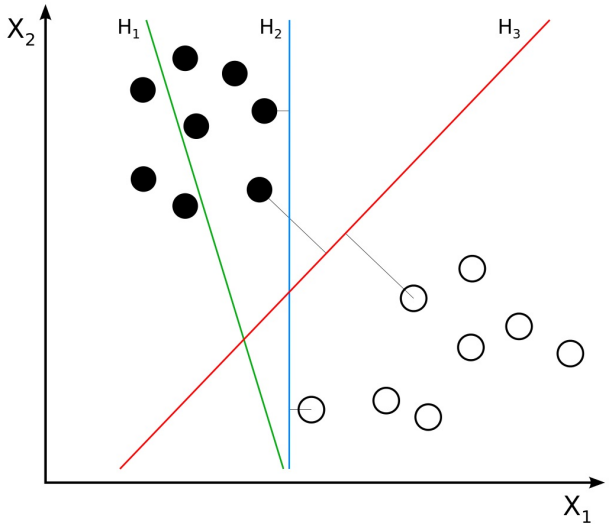
A principal característica do SVM é a boa capacidade de generalização e suporte a dados de grande dimensão. Pode-se dizer que ele implementa um classificador linear cuja função discriminante geral é da forma:

$$f(x,w,b) = sign(w * x - b) \tag{3}$$

Onde: w, é o vetor de peso associado a cada elemento do vetor suporte x, são os elementos (amostras) do vetor suporte Usando-se da equação anterior separam-se dois espaços, um positivo e outro negativo cujas fronteiras são os vetores suporte, pontos da extremidade da margem que servem como fronteira da maximização da mesma. b é um fator compensador que permite aumentar a margem da separação de hiperplanos. Tanto b como w são parâmetros ajustados durante o treinamento.

Os documentos mais próximos ao hiperplano são denominados $supportvectors$ ou vetores de suporte. O objetivo do SVM é encontrar o hiperplano com maior distância entre os vetores de suporte e duas duas regiões. Esse tipo de classificador também é chamado de classificador da margem máxima, pois a soma das duas distâncias entre os vetores de suporte estabelece a margem do classificador. Assim o objetivo do SVM é maximizar a margem M de separação entre os elementos positivos e negativos.

Figura 5 – Exemplo de um classificador SVM



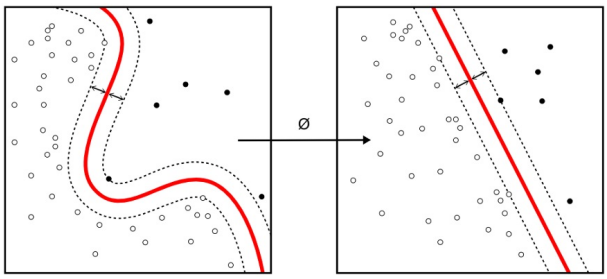
A reta (H3) é a mais distante dos dois grupos, considerando apenas os pontos de cada grupo mais próximos à reta (como indicado pelas linhas cinzas). Após descoberta essa reta, o programa conseguirá predizer a qual classe pertence um novo dado ao checar de qual lado da reta ele está.

Em situações em é preciso classificar nossos dados em mais de duas classes, é necessário buscar algum modo de simplificar o trabalho. A maneira mais utilizada e de menor complexidade é dividir o problema multiclasses em várias classificações binárias, que podem ser um versus um ou um versus todos.

A divisão em um versus um consiste em separar classificações binárias de cada par diferente. Se, por exemplo, houver três classes A, B e C, serão feitas as comparações (A,B), (A,C) e (B,C), e a classe mais votada será a escolhida. Caso o SVM escolher A em (A,B), C em (A,C) e C em (B,C), o resultado será (A, C, C) e a classe escolhida será a C.

No entanto, alguns grupos existentes não podem ser separados somente por hiperplanos. Nesses casos, o SVM Não Linear é utilizado para delimitar as duas classes que traçará uma ou mais linhas retas ou curvas para separar as classes da melhor forma possível. Com a finalidade de separar esses tipos de exemplos, o algoritmo primeiro faz uma transformação não-liner do espaço para depois poder separar os grupos com um SVM linear. Portanto, apesar da separação ser um hiperplano no espaço das *features*, no espaço das entradas a separação é não-linear.

Figura 6 – Exemplo de um classificador Não Linear e Linear



A imagem acima demonstra uma transformação não linear entre o espaço das entradas (à esquerda) e o espaço das features (à direita).

2.10 Random Forest

O *Random Forest* RF é um algoritmo de aprendizado de máquina baseado em árvore de decisão. Uma Árvore de Decisão é utilizada para classificar uma instância baseada em variáveis decisórias seguindo o caminho da raíz até as folhas (BREIMAN, 2001). Os atributos (ou variáveis decisórias) são utilizadas para tomar decisões, e suas respostas formam caminhos específicos em uma árvore.

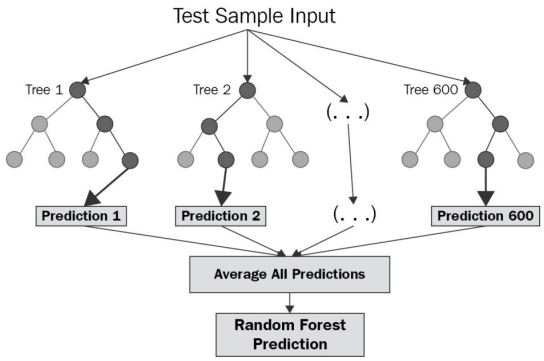
A Arvore de Decisão é uma estrutura de fluxograma semelhante a uma árvore. A estrutura é utilizada para aplicar um conjunto de regras, onde cada nó interno é responsável por testar um atributo, cada ramificação representa os resultados do teste de determinado nó e cada folha indica o rótulo da classe. Essa árvore é gerada através de processo de divisão, guiado por uma medida de satisfação (a entropia). É utilizada ainda, uma abordagem gulosa para decidir que atributos devem ser levados em consideração para dividir o conjunto de dados em uma determinada iteração do método (PACIFICO; BRITTO; LUDERMIR, 2020).

Segundo Britto e Pacífico (2020), e um método que combina um conjunto de Árvores de Decisão, no intuito de evitar o impacto que ruídos e *outliers* podem ter no resultado de uma única Árvore, o que torna o classificador muito mais robusto. O algoritmo combina diversas

Árvores, agregando os votos de diferentes estimadores para decidir a classe final do dado de teste.

Para a categorização, pode-se classificar cada atributo como uma *feature* e cada resposta seria uma possível característica dessa *feature*. A partir de um subconjunto de dados já classificados, uma árvore de decisão poderia ser construída. Devido ao dinamismo semântico da língua portuguesa, a linguagem natural pode variar consideravelmente mesmo quando são utilizadas as mesmas palavras em contextos distintos. Logo, a construção de apenas uma árvore não seria suficiente para a criação de um modelo de aprendizado de máquina eficiente. Por este motivo, o RF propõe que sejam criadas diversas árvores de decisão baseadas em subconjuntos aleatórios de uma base de dados. Dessa forma, a classificação passa a ser baseada em uma "floresta" de decisão, e não somente em uma árvore. Se houver atributos nas árvores de decisão, pode ocorrer o *overfitting*, tornando o classificador pouco genérico, podendo ocorrer erros maiores que o esperado. Para solucionar esse problema, a profundidade de cada árvore é diminuída para maximizar os resultados e generalizar a solução.

Figura 7 – Exemplo do Random Forest



Fonte: Lorena e Carvalho (2007)

2.11 Naïve Bayes

A técnica de Naïve Bayes (NB) é um algoritmo de classificação baseado no Teorema de Bayes, fundamentado na teoria das probabilidades, que assume que cada atributo influenciará de forma independente a classificação de uma nova instância (AMARAL, 2016). Na abordagem NB, supõe-se que, dadas as características, existe independência entre os preditores. O NB então assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro atributo. Segundo Amaral (2016), durante a etapa de treinamento, estabelece-se uma tabela de valores e deve se atribuir um peso para cada atributo em cada uma das classes de classificação. Quando uma nova instância é submetida à classificação, o modelo somará os pesos de cada atributo em cada uma das classes. A classe que somar o maior peso será a classe classificadora do novo item.

O NB segue uma fórmula matemática utilizada para cálculo de probabilidades con-

dicionais, descrevendo a probabilidade de um evento com base no conhecimento prévio das condições que podem estar relacionadas com esse evento:

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)} \tag{4}$$

Onde:

$P(c|x)$ é a probabilidade posterior da classe alvo (probabilidade de x acontecer dado c), $P(c)$ a probabilidade original da classe, $P(x|c)$ é a possibilidade de que a probabilidade da classe preditora seja dada (probabilidade de c acontecer dado x), $P(x)$ é a probabilidade original do preditor (probabilidade de x acontecer).

Para a classificação de texto, os xis são as palavras individuais do documento. O classificador gera a classe com a probabilidade posterior máxima.

2.12 Regressão Logística

Regressão logística trata de uma técnica estatística que procura descrever as relações entre uma variável categórica dependente e uma ou mais variáveis explicativas Everitt (1992).

A regressão logística é um dos principais modelos utilizados quando se deseja analisar dados em que a variável resposta é binária ou dicotômica. Geralmente, mesmo se a resposta de interesse não é inicialmente binária, a resposta de interesse é dicotomizada de modo que a probabilidade de sucesso possa ser estimada com o modelo de regressão logística. A regressão logística se tornou popular por ser flexível do ponto de vista matemático, de fácil utilização e por apresentar interpretação simples de seus parâmetros Giolo (2017).

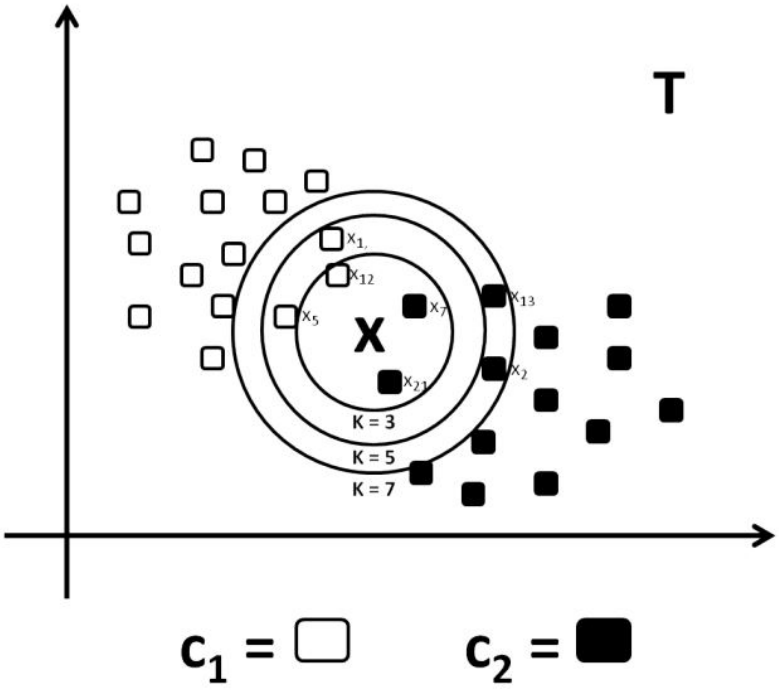
2.13 K-vizinhos mais próximos

O k-NN (*K nearest neighboors*), proposto por Fukunaga e Narendra (1975), é um dos classificadores mais simples de ser implementado e ainda hoje é capaz de obter bons resultados. O k-NN é um classificador onde o aprendizado é baseado na analogia. O conjunto de treinamento é formado por vetores n-dimensionais e cada elemento deste conjunto representa um ponto no espaço n-dimensional. Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador k-NN procura k elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância. Estes k elementos são chamados de k-vizinhos mais próximos. Verifica-se quais são as classes desses k vizinhos e a classe mais frequente será atribuída à classe do elemento desconhecido (FARIA; MONTEIRO, 2015).

K geralmente é um número ímpar quando o número de classes é 2. Se K $= 1$, o algoritmo é conhecido como o algoritmo do vizinho mais próximo. Esse é o caso mais simples. Supondo que P1 é o ponto para o qual o rótulo precisa prever, deve-se achar o ponto mais próximo de P1 e, em seguida, o rótulo do ponto mais próximo atribuído à P1.

A Figura 8 mostra um modelo de classificação com K-NN e a influência do valor de k na classificação de novas instâncias. O conjunto T é usado para classificar a instância desconhecida x. Dada a configuração espacial dos exemplos de treinamento, valores diferentes de k classificam o novo exemplo com classes diferentes. Para os valores k = 3 e k = 7, a instância é classificada como pertencente à classe c2. Já para k = 5, a classe c1 é atribuída à nova instância (MOURA, 2015). Isso mostra a influência da escolha do valor de k no algoritmo K-NN.

Figura 8 – Classificação de uma instância desconhecida com k-NN.



Fonte: Moura (2015)

2.13.1 Cálculo da distância

Existem diversas métricas de distância, e a escolha de qual usar varia de acordo com o problema. Uma das métricas mais utilizadas é a distância Euclidiana, descrita pela equação 5.

$$D_E(p,q) = \sqrt{(p_1 - q_1)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{5}$$

2.14 Distribuição Normal

A distribuição normal, conhecida também como distribuição gaussiana, é uma das mais importantes distribuições contínuas. Sua importância se deve a vários fatores, entre eles podemos citar o teorema central do limite, o qual é o resultado fundamental em aplicações

práticas e teóricas. O teorema central do limite garante que mesmo os dados que não possuem distribuição de acordo com uma normal, a média deles converge para uma distribuição normal conforme o número de dados aumenta. Na prática, diversos estudos têm como resultado uma distribuição normal. Podemos citar, por exemplo, a altura de uma determinada população: geralmente, a altura segue uma distribuição normal.

Definição: Uma variável aleatória contínua X tem distribuição Normal se a sua função densidade de probabilidade for dada por:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], \quad x \in (-\infty,\infty). \tag{6}$$

2.15 Testes Paramétricos e Não Paramétricos

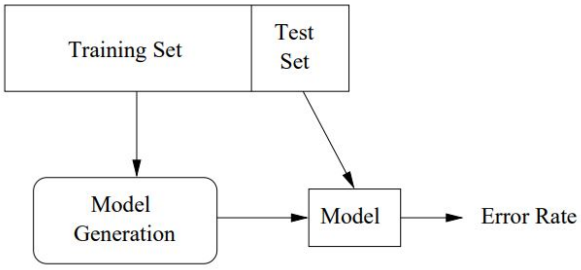
Os testes estatísticos podem ser divididos em dois grandes grupos, conforme fundamentem ou não os seus cálculos na premissa de que a distribuição de frequências dos erros amostrais é normal, as variâncias homogêneas, os efeitos dos fatores de variação são aditivos e os erros independentes. Se estas condições ocorrerem, provavelmente a amostra seja simétrica, terá um ponto de máximo, centrado no intervalo de classe onde está a média da distribuição, seu histograma de frequências terá um contorno paramétrico e o desenho terá uma forma de sino da curva normal. O cumprimento desses requisitos condiciona o pesquisador aos testes paramétricos, uma vez que, se forem preenchidos, ele poderá utilizar a estatística paramétrica, cujos testes são em geral mais poderosos do que os da estatística não-paramétrica, e consequentemente devem ter a preferência do investigador quando o seu emprego for permitido (CAMPOS, 2002). Os termos paramétrico e não-paramétrico referem-se à média e ao desvio-padrão, que são os parâmetros que definem as populações que apresentam distribuição normal.

2.16 Métodos de Avaliação

As medidas mais comuns para a avaliação dos resultados da classificação de texto são: revocação (*recall*), precisão (*precision*), acurácia (*accuracy*) e F1-Score. Essas métricas utilizam como base os Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN).

Quando o modelo prevê um caso positivo corretamente, nós temos um caso de Verdadeiro Positivo; caso o seu modelo determine que uma classe é verdadeira, quando na verdade não é, nós temos um caso de Falso Positivo. Quando o modelo prevê um caso negativo corretamente, temos um caso de Verdadeiro Negativo. O oposto também pode ocorrer. Quando o modelo diz que a classe não é verdadeira, mas na verdade é, temos um caso de Falso Negativo, também conhecido.

Figura 9 – O conjunto de treinamento ajuda a construir o modelo, e o conjunto de testes o valida



Fonte: Rajaraman, Leskovec e Ullman (2014)

A Figura 9 ilustra a arquitetura de treino e teste. Assumimos que todos os dados são adequados para o treino, ou seja, as informações de cada classe ou, no caso estudado pelo trabalho em questão, as informações de cada categoria dos e-mails. Uma fração do conjunto de dados disponível é separada para o conjunto de testes. Os dados restantes podem e devem ser utilizados verificar se o classificador apresenta resultados aderentes a proposta. Como sabemos a classe de cada elemento de teste, pode-se dizer quantitativamente se o modelo apresenta uma taxa de erro menor que a pré-determinada. Caso a taxa de erro seja maior que a esperada, deve-se utilizar técnicas especificas de cada algoritmo, como, por exemplo, a utilização de balanceamento das classes, diminuição nos ruídos dos dados de treinamento ou aumento do conjunto de dados para a etapa de treinamento do modelo.

2.16.1 Validação Cruzada

Validação cruzada é um procedimento estatístico bastante comum no âmbito de mineração de dados e Aprendizado de Máquina. A validação cruzada é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados. Este procedimento consiste na partição de uma amostra de dados em subconjuntos enquanto os demais são guardados para uma subsequente confirmação e validação da análise inicial. O conjunto de dados inicial é chamado de conjunto de treinamento e os outros são chamados de conjuntos de validação ou teste. Caso utilizem-se 3 partições para a validação cruzada, significa que o conjunto de dados será separado em 3 conjuntos distintos e no final (após a etapa de treino) será feita uma avaliação sobre as métricas. O procedimento é repetido circularmente para todas as partições, onde cada uma é utilizada uma única vez para o treinamento. O resultado final pode ser a média das avaliações. Com esse procedimento, evita-se resultados tendenciosos ao se utilizar apenas um espaço amostral (TAVARES; LOPES; LIMA, 2007).

2.16.2 Precisão

Precisão é a razão entre as observações VP previstas corretamente e o total de observações julgadas como positivas, ou seja, o divisor é a soma de VP com FP. Pode-se ter

verdadeiros positivos

$$Precisão = \frac{VP}{VP + FP} \tag{7}$$

2.16.3 Acurácia

A acurácia é a medida de desempenho mais intuitiva (BIRD; KLEIN; LOPER, 2009), pois é uma proporção do resultado previsto corretamente com o total de observações do conjunto. Com esta medida, pode-se dizer quantas classificações foram de fato classificadas corretamente, independentemente se foram determinadas como positivas ou negativas. Esta métrica é definida pela razão entre o que o modelo acertou e a soma de todas as classificações.

$$Acurácia = \frac{VP + VF}{VP + VF + FP + FN} \tag{8}$$

2.16.4 Revocação

Também chamado de Sensibilidade nos casos para classificação binário, a revocação se define como medir a capacidade do modelo sabendo-se a proporção de VP que conseguimos acertar de todas os documentos da classe real, em outras palavras, quão bom meu modelo é para prever apenas os positivos. Portanto é definido como a razão entre verdadeiros positivos sobre a soma de verdadeiros positivos com negativos falsos.

$$Recall = \frac{VP}{VP + FN} \tag{9}$$

2.16.5 F1-Score

Mesmo com as medidas de precisão e recall sendo de grande utilidade para avaliações de desempenho de classificadores, às vezes se faz necessária uma medida única, para que se possa comparar de forma direta dois ou mais classificadores. E essa é a ideia da medida F1-Score que é uma medida baseada na média harmônica dos valores de precisão e recall.

$$F1 - Score = 2 * \frac{precisão * recall}{precisão + recall} \tag{10}$$

2.16.6 Matriz de Confusão

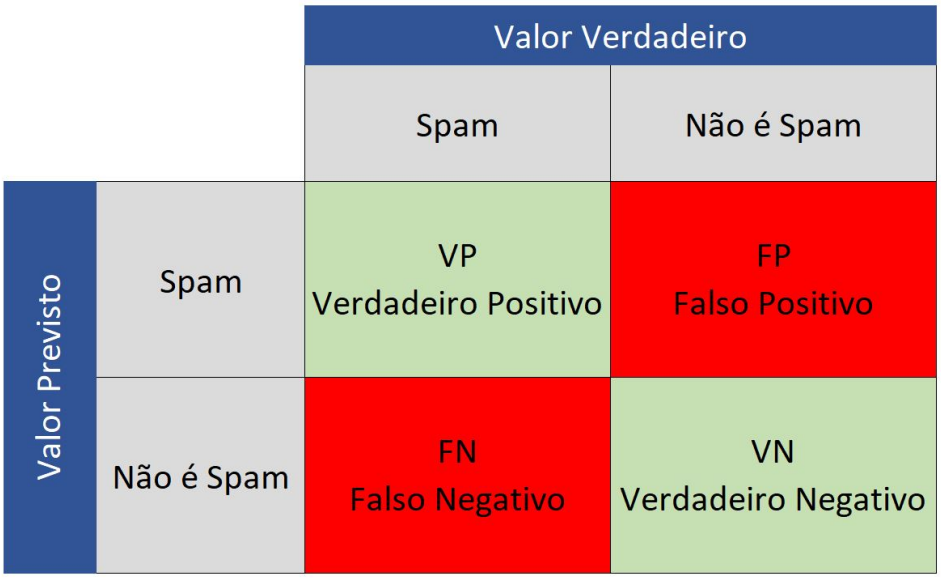
No campo do Aprendizado de Máquina uma matriz de confusão é uma tabela que permite a visualização do desempenho de um algoritmo de classificação (DUDA; HART; STORK, 2001).

Segundo Chipman et al. (2004), a matriz de confusão é um método amplamente utilizado para analisar resultados. Ela separa classe por classe, a relação entre os dados de referência conhecidos e os resultados correspondentes de uma classificação automatizada.

A matriz de confusão facilita a visualização do número de classificações corretas e do número de classificações preditas para cada classe, de um determinado conjunto de exemplos, segundo o classificador em análise. Torna-se uma ferramenta útil para analisar a qualidade do classificador no reconhecimento de exemplos de diferentes categorias (HAN; KAMBER, 2006).

A matriz de confusão fornece o número de VP, VN, FP e FN, além das métricas de acurácia e revocação. Como exemplo, a figura 10 mostra uma matriz de confusão com classificações de e-mails para determinar se é Spam[1] ou não. As quantidades de cada classificação são apresentadas nos locais de "VP", "VN", "FP" e "FN". Conforme mostrado nas seções anteriores, com estes valores é possível, por exemplo, calcular as métricas de acurácia, precisão, revocação e F1-Score,

Figura 10 – Exemplo de Matriz de Confusão: Classificação de Spams



Fonte: O autor

2.16.7 ROC

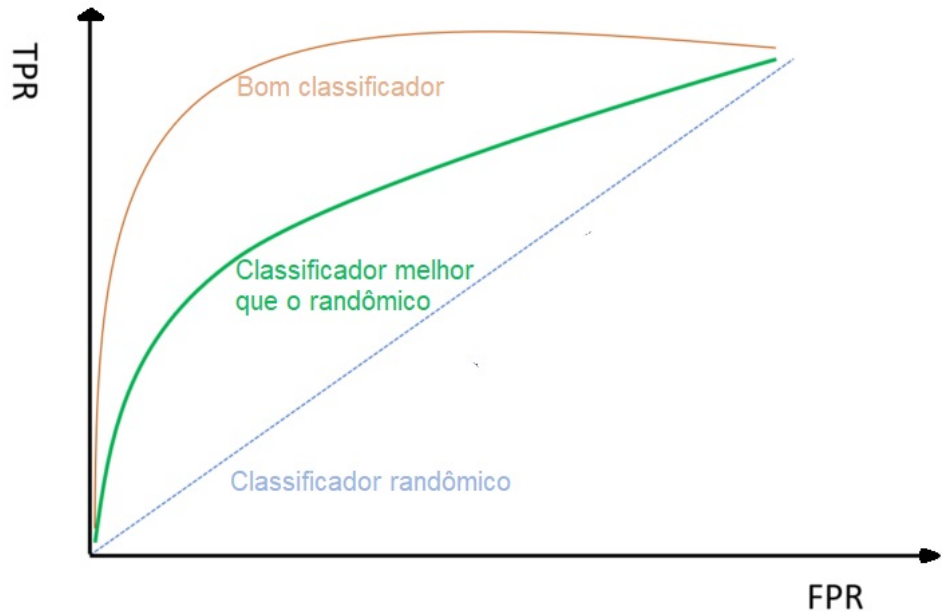
Receiver Operating Characteristics (ROC) é uma métrica muito utilizada para modelos de classificação. Curvas ROC são ferramentas que avaliam a sensibilidade da técnica. São baseadas na relação entre a fração de verdadeiros positivos (TPR) e a fração de falsos positivos (FPR).

Na análise do desempenho de modelos classificatórios são aplicados diversos estimadores estatísticos, e um dos mais utilizados é a curva ROC (Receiver Operating Characteristic), que consiste em uma representação gráfica da performance de um modelo de dados quantitativos segundo sua taxa de sensibilidade (fração dos verdadeiros positivos) e a fração dos falsos positivos (1-especificidade), segundo diferentes valores do teste.

[1] Spam é a prática que consiste em utilizar meios eletrônicos para enviar mensagens que não foram solicitadas

A curva ROC é o gráfico traçado com TPR no eixo y e FPR no eixo x para todos os limites possíveis. Tanto o TPR quanto o FPR variam de 0 a 1. Um bom classificador terá uma curva mais distante da linha do classificador randômico. Para quantificar um bom classificador de um mau usando uma curva ROC, é feito por AUC (Área sob a Curva). A partir do gráfico percebe-se que um bom classificador terá AUC maior do que um classificador ruim, pois a área sob a curva será maior para o primeiro.

Figura 11 – Curva Roc



Fonte: O autor

3 METODOLOGIA

Este capítulo descreve em profundidade os métodos e técnicas, assim como os dados e ferramentas que foram usados para conduzir o estudo de categorização de texto e construção automática da base de e-mails estudada, abordando modelos clássicos de aprendizado de máquina.

Todos os algoritmos implementados utilizaram a linguagem de programação Python[1] em conjunto com as bibliotecas do Scikit-Learn[2]. Scikit-learn é um módulo Python que integra uma ampla gama de algoritmos de aprendizado de máquina de última geração para problemas supervisionados e não supervisionados de média escala. Este pacote se concentra em levar o aprendizado de máquina para não especialistas usando uma linguagem de alto nível de uso geral (PEDREGOSA et al., 2011).

Resumidamente temos as seguintes etapas para a elaboração do estudo:

- Escolha dos algoritmos para modelagem de treinamento e teste;
- Aquisição da base de e-mails;
- Categorização dos e-mails;
- Pré-processamento dos dados;
- Modelagem;
- Avaliação dos resultados.

Para avaliação dos experimentos realizados, são apresentados alguns conceitos fundamentais para a metodologia. Dentro dessa metodologia são destacadas algumas medidas de avaliação amplamente empregadas na literatura nos estudos de classificação de texto com aprendizado de máquina. Para confirmar se os resultados encontrados com a metodologia, são utilizadas as métricas mais comuns em ML.

Os itens anteriores podem ser divididos em quatro etapas gerais:

Figura 12 – Fluxo completo do desenvolvimento

Fonte: O autor

Os detalhes de cada etapa serão apresentados nas seções seguintes. O capítulo finaliza com considerações pertinentes à escolha da metodologia proposta inicialmente.

[1] https://www.python.org/
[2] https://scikit-learn.org/stable/

3.1 Escolha dos Algoritmos

Os algoritmos escolhidos para categorização dos e-mails foram o Máquina de Vetores de Suporte (SVM), *Naive Bayes Multilinear*, k-Vizinhos Mais Próximos k-NN e o de Regressão Logística.

3.2 Aquisição do Conjunto de Dados

Uma parte essencial presente nos trabalhos que utilizam aprendizado de máquina é a correta escolha ou construção do *dataset*. Neste trabalho, a coleção de textos utilizadas para desenvolvimento dos experimentos é oriunda de e-mails pessoais recebidos entre os anos de 2020 e 2021. Foram utilizados, para este trabalho, 2993 e-mails, os quais contêm a data e hora do recebimento, o assunto, o corpo da mensagem, o remetente e o(s) destinatário(s).

O conteúdo dos e-mails foi armazenado em sua forma bruta e limpa (sem tags HTML), em um arquivo de extensão xlsx[3]. Os arquivos HTMLs de cada e-mail foram armazenados em uma pasta separada para facilitar a classificação manual e a verificação dos resultados.

3.3 Categorização dos E-mails

Para a classificação dos e-mails, foram escolhidas 17 categorias. Ainda na etapa de construção da base de dados, cada e-mail foi classificado de acordo com a escolha do autor, portanto o estudo propõe uma solução de aprendizado de máquina supervisionado. Cada um dos e-mails pertence a uma e apenas uma categoria, mesmo nos exemplos difíceis de distinguir a classificação dentre as previamente selecionadas.

3.4 Pré-processamento dos Dados

Os textos de cada e-mail foram pré-processados para aumentar a precisão dos modelos, conforme orientações na seção 2.7. No pré-processamento é feita a remoção de dados desnecessários, facilitando uma melhor classificação dos e-mails. A biblioteca *BeautifulSoup*[4] foi utilizada para remover *tags* em HTML [5](Linguagem de Marcação de Hipertexto). Além disso, há a remoção das pontuações, dos acentos, dos caracteres especiais e das *stopwords* para aumentar a precisão dos modelos, considerando apenas palavras, ou conjunto de palavras, determinantes para definir a categoria dos e-mails em questão. Com o objetivo de reduzir a dimensionalidade e aumentar a acurácia dos modelos, o processo de *stemming* foi utilizado para eliminar os sufixos das palavras.

[3]XLS e XLSX são versões de arquivo do Microsoft Excel. A versão XLSX é recomendada para o Microsoft Excel 2010 ou superior, já a versão XLS é compatível com versões mais antigas

[4]Biblioteca disponível em https://pypi.org/project/beautifulsoup4/

[5]HTML Define o significado e a estrutura do conteúdo da web. Serve para dar significado e organizar as informações de uma página na web

Para finalizar a etapa de pré-processamento, as palavras, resultantes do processo de stemming, foram vetorizadas utilizando as abordagens de Bag of Words e TF-IDF.

3.5 Modelagem

Conforme mostado na seção 3.1 algoritmos escolhidos para categorização dos e-mails foram o Máquina de Vetores de Suporte (SVM), *Naive Bayes Multilinear*, k-Vizinhos Mais Próximos k-NN e o de Regressão Logística. Foram utilizadas bibliotecas Python fornecidas pelo *scikit-learn*. Diferentes parâmetros foram testados em cada algoritmo utilizado. Para treinamento dos modelos, foram separados 33% dos e-mails para treino e 67% de e-mails para teste.

Cada algoritmo de aprendizagem de máquina executou o modelo três vezes, utilizando duas técnicas distintas (TF-IDF e BOW), para vetorizar os textos. Para a técnica TF-IDF, foram utilizados dois n-gramas distintos, um de dimensão 1 (unigrama) e outro de tamanho 2 (bigrama), ou seja, considerando as duas técnicas, obtiveram-se 12 resultados de categorização dos e-mails. Nos resultados, TF-IDF-1 e TF-IDF-2 representam, respectivamente, as análises do TF-IDF utilizando unigrama e TF-IDF com bigramas.

Para verificar se os algoritmos mantêm o comportamento com diferentes subconjuntos de treino e testo, as execuções dos modelos foram executadas três vezes, portanto temos 36 resultados contendo as respectivas métricas de avaliação.

3.6 Avaliação

A medição de desempenho dos classificadores deste trabalho exige algumas métricas. As métricas utilizadas foram: acurácia, precisão, revocação e o F1-Score. Estas são métricas simples e amplamente utilizadas em diferentes literaturas.

As métricas resultantes das três execuções dos algoritmos de ML foram comparadas para verificar se os resultados são consistentes.

O cálculo das médias e desvios padrão também foram calculados para analisar o comportamento dos algoritmos.

Utilizou-se matrizes de confusão normalizadas para, visualmente, verificar os algoritmos de AM que obtiveram melhores resultados.

Por último, o tempo de execução de cada algoritmo foi analisado após notar um alto intervalo de execução para o algoritmo de Regressão Logística.

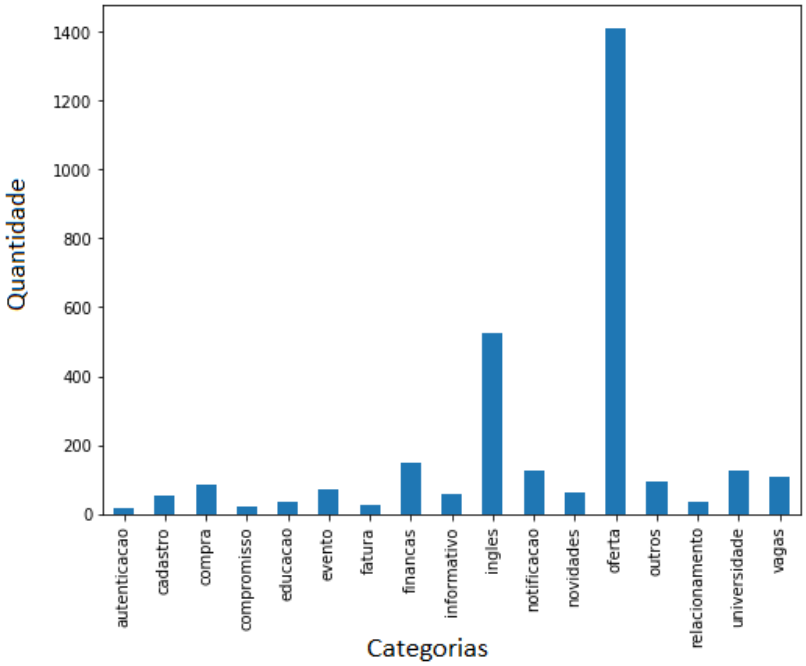
4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

As seções deste capítulo mostram os resultados experimentais da execução de diferentes algoritmos de classificação que se diferenciam pelos algoritmos-base e pela arquitetura de classificação.

Na primeira etapa, o pré-processamento, são removidas palavras irrelevantes para o conjunto de resultados a ser exibido, ou seja, são palavras que não agregam valor significativo para a categorização. As palavras com frequência inferior a cinco não são utilizadas por se tratar de palavras raras, que costumam não influenciar positivamente no processo de categorização, ou então são erros ortográficos.

Os algoritmos de AM utilizaram a mesma base de e-mails, porém houve a validação cruzada, com 3 execuções de cada algoritmo, para diminuir a chance de enviesamento nos resultados. A Figura 13 apresenta a distribuição de categorias dos e-mails utilizados para o estudo.

Figura 13 – Distribuição dos e-mails por categorias



Fonte: O autor

Após a aplicação das funções para a classificação, foi criada uma matriz de confusão para estudar e obter os resultados da análise.

As técnicas Frequência do Termo - Frequência Inversa dos Documentos (TF-IDF) e Saco-De-Palavras (BOW) foram utilizadas para representar vetorialmente os textos de

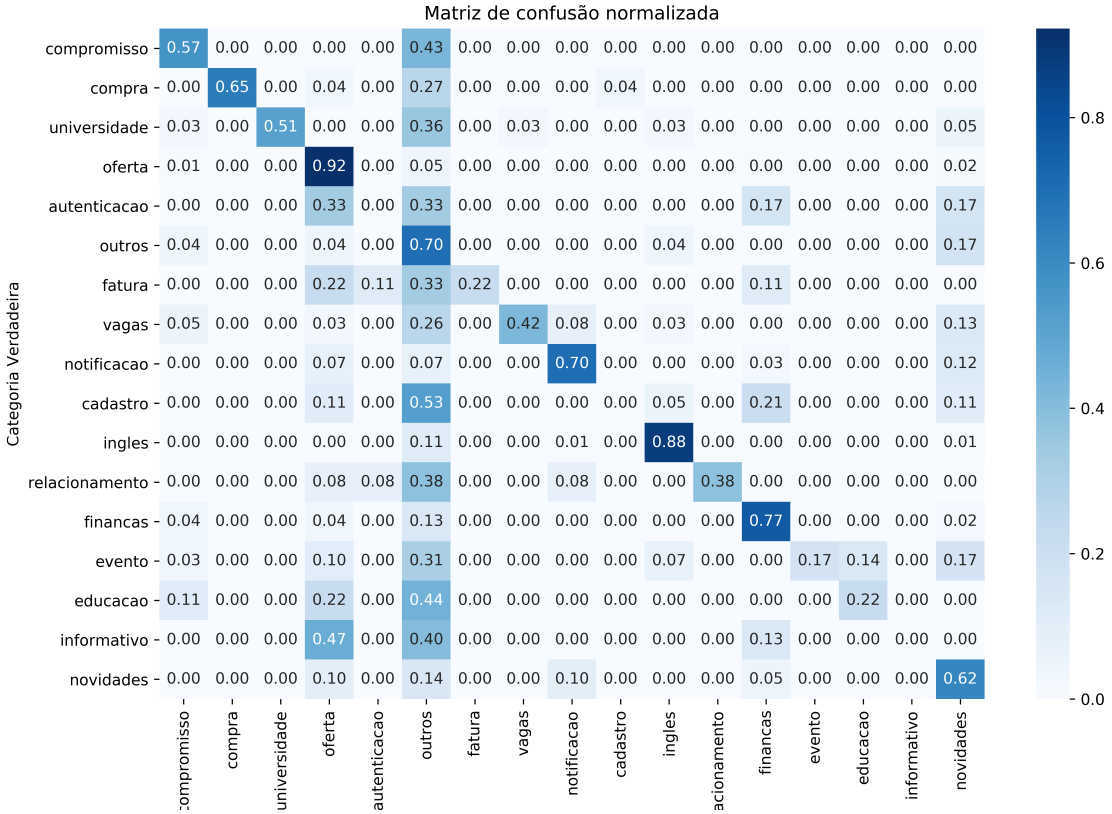
cada e-mail. Além disso, para o TF-IDF, foram utilizados unigramas (TF-IDF-1) e bigramas (TF-IDF-2).

4.1 Modelo K-Nearest Neighbors

Empiricamente, foi determinado que k igual a 10 é o valor com maior capacidade de obter bons resultados no algoritmo k-NN. Os experimentos relatados nesta seção são compostos por três execuções dos classifcador k-NN, com três conjuntos de dados distintos para treino e testes (validação cruzada).

A Figura 14 demonstra a matriz de confusão de uma das execuções do algoritmo k-NN utilizando a vetorização da técnica BOW. Não foi encontrado um motivo razoável para explicar o porquê diversas classes tiveram os resultados impactados pela classe "outros". Talvez seja devido à aleatoriedade de palavras presentes nesta categoria.

Figura 14 – Matriz de Confusão do Modelo KNN de uma das execuções com o vetor BOW



Fonte: O autor

A Tabela 1 mostra as médias e desvios padrão das execuções do k-NN considerando os três vetores utilizados para o trabalho.

As médias obtiveram valores próximos, porém os desvios padrão do TF-IDF-1 e TF-IDF-2 obtiveram maiores variações porque uma das execuções do k-NN em um dos conjuntos

Tabela 1 – Média e desvio padrão obtidos com o algoritmo k-NN com os vetores BOW, TF-IDF-1 e TF-IDF-2

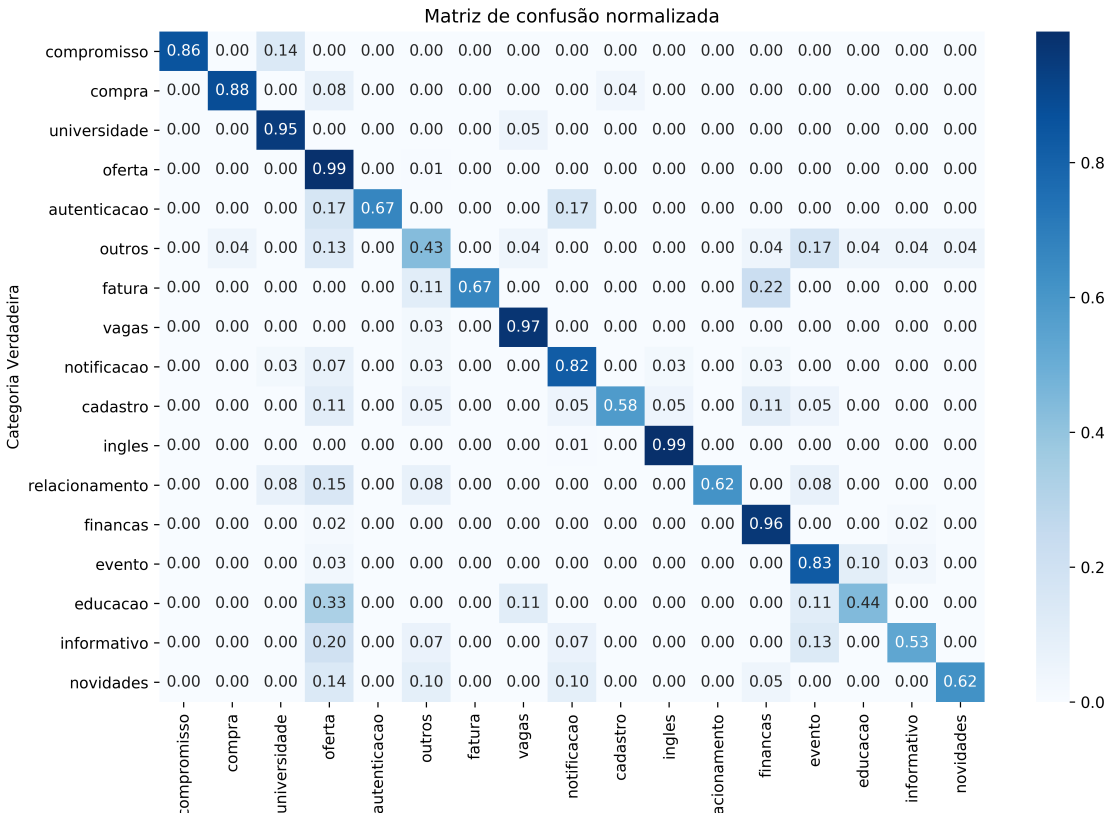
Vetor	Acurácia%	Precisão%	Revocação%	F1-Score%
BOW	76.15 ± 0.46	77.66 ± 0.44	84.50 ± 1.29	76.15 ± 0.46
TF-IDF-1	77.60 ± 8.30	79.52 ± 5.39	87.50 ± 1.94	77.60 ± 8.30
TF-IDF-2	77.53 ± 9.05	79.73 ± 6.10	87.85 ± 1.96	77.53 ± 9.05

de testes e treino, teve uma variação de quase 10% na acurácia.

4.2 Modelo Regressão Logística Multiclasse

O modelo de Regressão Logística Multiclasse (RL Multiclasse) apresentou ótimos resultados. Tal fato é mostrado na matriz de convolução normalizada apresentada na Figura 15, obtida em uma das execuções. Quanto mais escura for a diagonal principal da matriz de convolução, melhor o resultado do algoritmo.

Figura 15 – Matriz de Confusão do Modelo Regressão Logística Multiclasse de uma das execuções com o vetor TF-IDF-2



Fonte: O autor

As Tabela (Figura 2), contendo médias e desvios padrão obtidos com o modelo de

Regressão Logística, demonstram a alta assertividade e consistência neste modelo.

Tabela 2 – Médias obtidas com o algoritmo Regressão Logística e vetores BOW, TF-IDF-1 e TF-IDF-2

Vetor	Acurácia%	Precisão%	Revocação%	F1-Score%
BOW	90.76 ± 0.62	89.74 ± 0.96	90.76 ± 0.62	89.81 ± 0.83
TF-IDF-1	91.57 ± 0.70	91.18± 1.37	91.57 ± 0.70	91.01 ± 0.96
TF-IDF-2	91.90 ± 0.36	91.60± 0.83	91.90 ± 0.36	91.39 ± 0.60

4.3 Modelo Naïve Bayes Multinomial

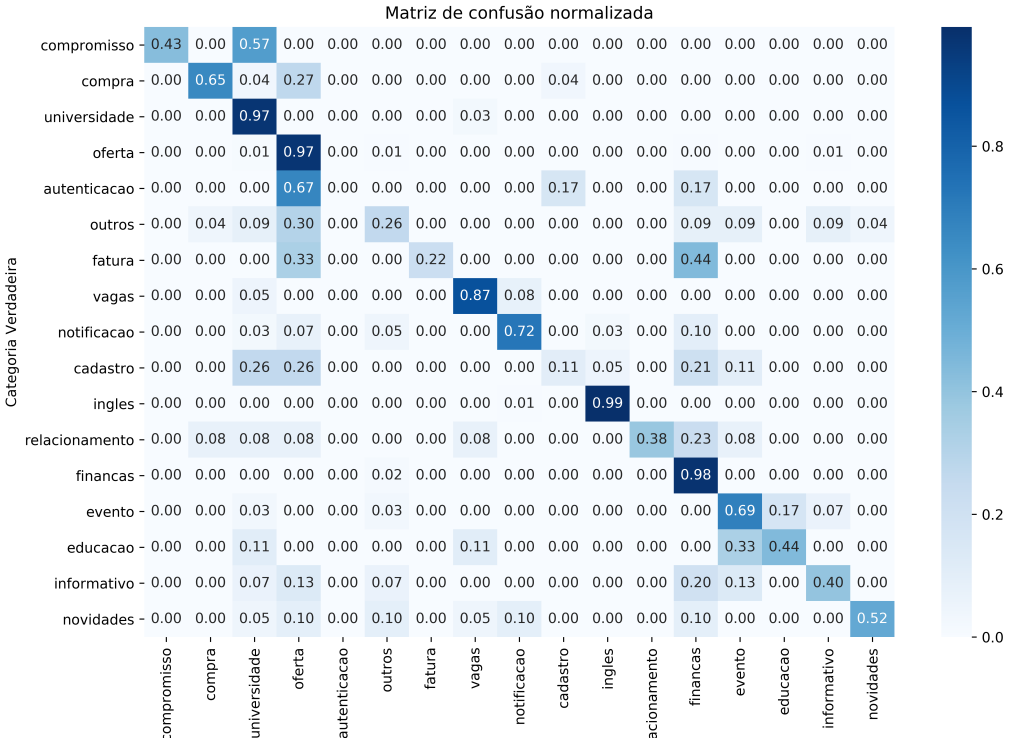
Com resultados melhores que o k-NN, porém abaixo dos resultados obtidos com RL e SVM, o modelo que utilizou o algoritmo Naïve Bayes Multinomial (NB Multinomial) também obteve bons resultados. O NB apresentou melhores resultados ao utilizar o vetor *Bag-of-Words*. As médias (Tabela 3) das métricas apresentadas obtiveram resultados superiores em comparação aos resultados obtidos ao utilizar o TF-IDF-1 e TF-IDF-2.

Tabela 3 – Média e Desvio padrão obtidos com o algoritmo NB Multinomial e vetores BOW, TF-IDF-1 e TF-IDF-2

Vetor	Acurácia%	Precisão%	Revocação%	F1-Score%
BOW	85.76± 1.10	85.00± 2.36	85.76± 1.10	83.96± 1.81
TF-IDF-1	77.83± 1.04	74.18± 3.13	77.83± 1.04	71.27± 1.62
TF-IDF-2	80.63± 0.88	79.17± 3.16	80.63± 0.88	75.81± 1.37

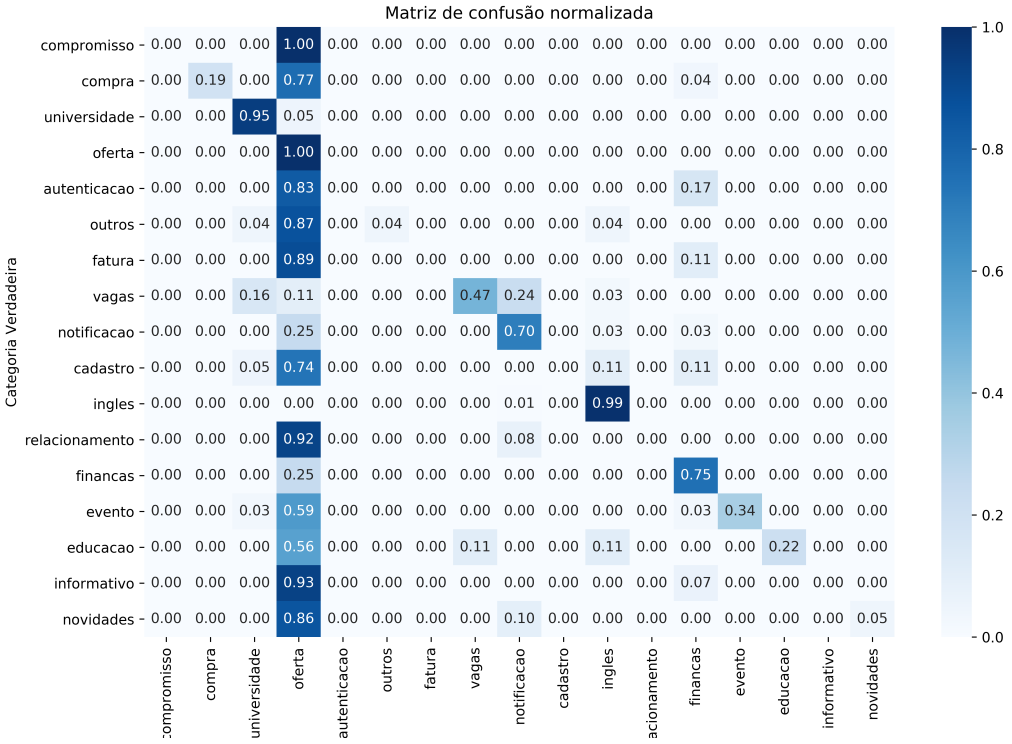
Ao comparar as matrizes de convolução nas Figuras 16 e 17, utilizando os vetores BOW e TF-IDF respectivamente, nota-se o enviesamento por parte do segundo, e maior precisão e sensibilidade do primeiro. Parte considerável dos e-mails foram classificados erroneamente como "oferta" (Falsos Positivos) na utilização do TF-IDF, demonstrando alto enviesamento por parta da classe majoritária.

Figura 16 – Matriz de Confusão do Modelo Naïve Bayes Multinomial com BOW



Fonte: O autor

Figura 17 – Matriz de Confusão do Modelo Naïve Bayes Multinomial com TF-IDF-1



Fonte: O autor

4.4 Modelo SVM

O modelo SVM apresentou resultados similares aos encontrados no de Regressão Logística (RL). Ainda comparando-os, o SVM obteve alta superioridade no tempo de execução para finalizar as classificações. Enquanto o SVM levava menos de 1 minuto, o algoritmo de RL demorava em média 15 minutos para finalizar as classificações.

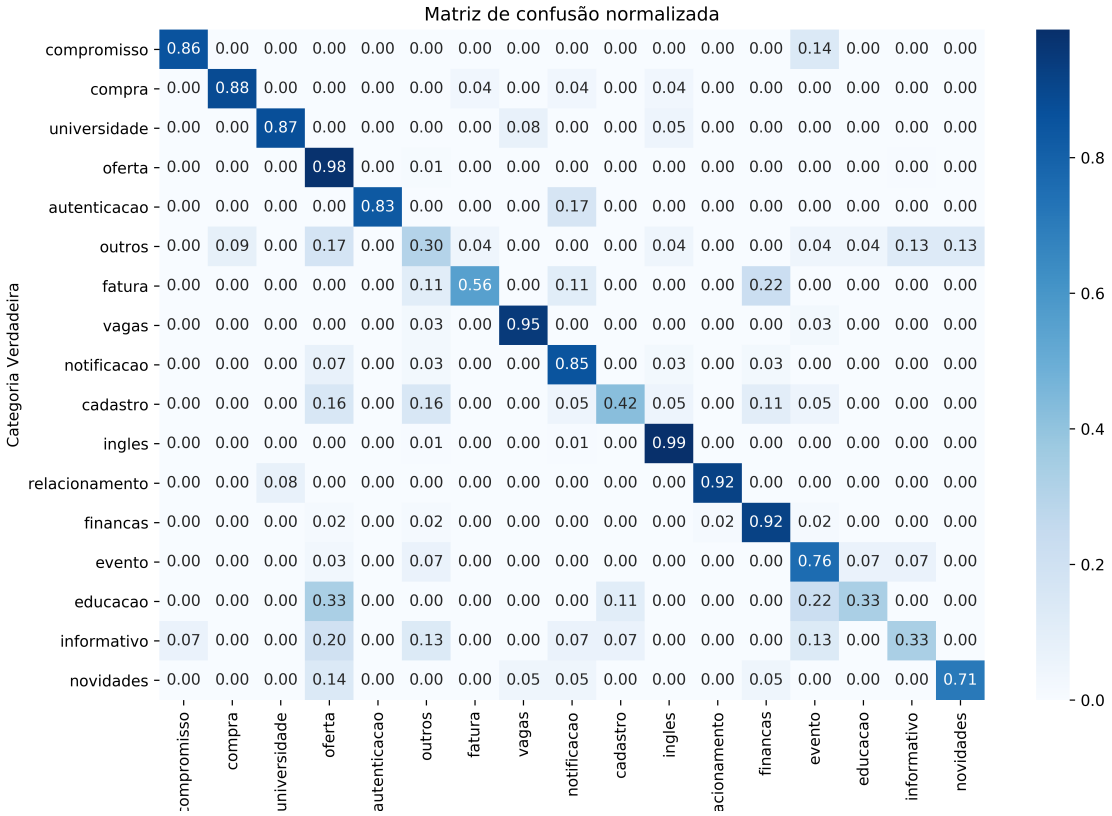
Quase não houve diferença entre os resultados do SVM com os três vetores, conforme constatado na Tabela 4.

A figura 18 apresenta alta taxa de acertos, com exceção das categorias "outros", "educacao" e "informativo".

Tabela 4 – Média e desvio padrão obtidos com o algoritmo SVM e vetores BOW, TF-IDF-1 e TF-IDF-2

Vetor	Acurácia%	Precisão%	Revocação%	F1-Score%
BOW	89.57 ± 0.91	88.91 ± 1.15	89.57 ± 0.91	88.98 ± 1.11
TF-IDF-1	91.50 ± 0.71	90.57 ± 1.15	91.50 ± 0.71	90.63 ± 0.99
TF-IDF-2	91.50 ± 0.10	90.74 ± 0.40	91.50 ± 0.10	90.68 ± 0.28

Figura 18 – Matriz de Confusão do Modelo SVM



Fonte: O autor

4.5 Comparação dos Resultados

Analisando apenas os resultados das métricas dos algoritmos de Regressão Logística e o SVM, a pequena diferença não justifica a escolha entre um e outro. Porém, a diferença no tempo de execução entre esses algoritmos foi considerável, como pode-se observar na tabela 5.

Tabela 5 – Tempo de execução dos algoritmos

Algoritmo	Tempo médio (segundos)	Tempo médio normalizado %
SVM	49	4.68
Naive Bayes	45	4.30
k-NN	62	5.93
Regressão Logística	890	85.09

Apesar de ser uma solução simples em comparação com as demais, o vetor de representação *BOW* obteve melhor resultado em três algoritmos dos quatro analisados. Analisando apenas pelas médias das métricas, o BOW se sobressairia somente no algoritmo SVM, porém o baixo desvio padrão (melhor consistência) frente aos demais foi determinante para sua escolha

no algoritmo k-NN. Os tempos de execução, destacados na tabela 5, e resultados similares nas métricas apresentadas na Tabela (6) determinaram a escolha do BOW também no algoritmo de Regressão Logística Multiclasse (RL Multiclasse).

Inicialmente era esperado que a Frequência do Termo - Frequência Inversa dos Documentos (TF-IDF) com unigrama (TF-IDF-1) ou bigrama (TF-IDF-2), apontariam melhores resultados, porém, neste trabalho, o BOW foi o que se destacou positivamente.

Tabela 6 – Ordem decrescente dos melhores algoritmos, em decorrência das métricas e tempo de execução

Modelo	Vetor	Acurácia%	Precisão%	Revocação	F1-Score%
SVM	TF-IDF2	91.30±0.91	91.43±1.15	91.30±0.91	91.06±1.11
Regressão Logística	BOW	91.90±0.62	91.60±0.96	91.60±0.62	91.39±0.83
Naive Bayes	BOW	85.76±1.10	85.00±2.36	75.76±1.10	73.96±1.81
k-NN	BOW	76.15±0.46	77.66±0.44	84.50±1.29	76.15±0.46

5 CONCLUSÃO

As mensagens de e-mail, cada vez mais, são utilizadas como meio de comunicação entre as pessoas, seja no contexto pessoal ou profissional. Dado o alto volume de informações que cada pessoa recebe por dia, a necessidade de organizar essas mensagens para não se perder informações importantes é cada vez maior. O problema maior consiste na falta de disponibilidade de tempo em organizar manualmente essas mensagens. Neste cenário, surge a crescente necessidade de que a classificação seja feita de forma automática.

Com este trabalho foi possível construir abordagens de Aprendizagem de Máquina para classificar os documentos de textos, contidos nas mensagens eletrônicas, de forma semi-automática. A humana ainda foi necessária para categorizar previamente a base de e-mails estudada.

Existem diversos algoritmos de Aprendizagem de Máquina amplamente utilizados para categorizar textos. Os quatro algoritmos utilizados, SVM, Naive Bayes, Regressão Logística e k-NN, se mostraram eficazes para resolução do problema estudado.

A técnica de vetorização do *Bag of Words*, apesar de mais simples, obteve resultados melhores que o TF-IDF em 75% dos algoritmos.

Através deste trabalho, verificou-se que a utilização de técnicas de Aprendizado de Máquina é útil na classificação de e-mails em diferentes categorias. Para a classificação de uma base de dados que apresentam grande variedade de informações, o algoritmo Máquinas de Vetores de Suporte (SVM) foi o que apresentou melhor desempenho em relação aos demais algoritmos, com uma acurácia de 91.30%, precisão de 91,43%m revocação de 91,30% e F1-Score de 91,06%. Apesar do algoritmo SVM ter obtido as melhores métricas, os algoritmos restantes, Regressão Logística (RL), Naïve Bayes (NB) e k-Vizinhos Mais Próximos (k-NN), apresentaram uma boa acurácia (91.9%, 85.76% e 76.15% respectivamente). Importante destacar que o algoritmo de Regressão Logística não obteve o melhor resultado devido ao seu alto tempo de execução comparado com os demais. Os resultados das métricas, obtidos com este estudo, indica que os quatro algoritmos testados (SVM, RL, NB e k-NN) podem ser aplicados em casos similares de categorização de mensagens eletrônicas com base no corpo de e-mail.

5.1 Trabalhos Futuros

Para continuação deste trabalho fica definidas algumas possibilidades que poderiam melhorar o estudo e os resultados obtidos:

- Utilizar técnicas de *oversampling* e *undersampling* para diminuir o enviesamento dos resultados para as classes majoritárias;
- Remover uma parcela considerável dos e-mails, até que seja obtido a mesma quantidade de e-mails para cada categoria. Isto equilibraria o número de e-mails por categoria e

diminuiria o problema da base de estudo conter classes majoritárias;

- Utilizar outros dados e metadados que compõem os e-mails, como remetente, assunto e destinatário(s);
- Determinar uma base de dados mais genérica, ou seja, com menos categorias. A categoria "autenticação", por exemplo, também poderia ser classificada na categoria "conta"devido às suas similaridades;
- Utilizar outras métricas para validar os resultados, como a curva ROC e a média macro.

Referências

PEDREGOSA, Fabian and Varoquaux, Gaël and Gramfort, Alexandre and Michel, Vincent and Thirion, Bertrand and Grisel, Olivier and Blondel, Mathieu and Prettenhofer, Peter and Weiss, Ron and Dubourg, Vincent and others. [S.l.]. Citado na página 1.

AGGARWAL, C. C.; ZHAI, C. A survey of text classification algorithms. In: **Mining text data**. [S.l.]: Springer, 2012. p. 163–222. Citado na página 13.

AMARAL, F. **Aprenda Mineração de Dados Teoria e Prática**. Rio de Janeiro: Alta Books, 2016. Citado na página 20.

BERRY, M. J.; LINOFF, G. S. **Data mining techniques: for marketing, sales, and customer relationship management**. [S.l.]: John Wiley & Sons, 2004. Citado na página 7.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. **Artificial intelligence**, Elsevier, v. 97, n. 1-2, p. 245–271, 1997. Citado na página 12.

BRAGA, Í. A. **Aprendizado semissupervisionado multidescrição em classificação de textos**. Tese (Doutorado) — Universidade de São Paulo, 2010. Citado na página 16.

BREIMAN, L. **Random forests**. 2001. 5-32 p. Citado na página 19.

BRITTO, L.; PACÍFICO, L. Uma abordagem de classificação de sentimentos em revisões de livros em português brasileiro usando diferentes métodos de extração de características. In: SBC. **Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2020. p. 116–127. Citado na página 19.

CAMPOS, G. M. **Estatística prática para docentes e pós-graduandos**. [S.l.]: Faculdade de Odontologia de Ribeirão Preto da Universidade de São Paulo, 2002. Citado na página 23.

CHIPMAN, J. W. et al. Mapping lake water clarity with landsat images in wisconsin, usa. **Canadian journal of remote sensing**, Taylor & Francis, v. 30, n. 1, p. 1–7, 2004. Citado na página 25.

CHOWDHURY, G. Natural language processing. **Annual Review of Information Science and Technology**, v. 37, n. 1, p. 51–89, jan. 2003. ISSN 0066-4200. Citado na página 5.

CHOWDHURY, G. G. Natural language processing. **Annual review of information science and technology**, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003. Citado na página 5.

DAUMÉ III, H. A course in machine learning. **Publisher, ciml. info**, v. 5, p. 69, 2012. Citado na página 8.

DEBARR, D.; WECHSLER, H. Spam detection using clustering, random forests, and active learning. In: CITESEER. **Sixth Conference on Email and Anti-Spam. Mountain View, California**. [S.l.], 2009. p. 1–6. Citado na página 14.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification, Hoboken**. [S.l.]: NJ: Wiley, 2001. Citado na página 25.

ESCOVEDO, T. et al. Neuroevolutionary learning in nonstationary environments. **Applied Intelligence**, Springer, v. 50, n. 5, p. 1590–1608, 2020. Citado na página 8.

EVERITT, B. S. **The analysis of contingency tables**. [S.l.]: CRC Press, 1992. Citado na página 21.

FARIA, M. M.; MONTEIRO, A. M. Investigação sobre técnicas de detecção de intrusões em redes de computadorescom base nos algoritmos knn e k-means. 2015. Citado na página 21.

FAYYAD, U. M. et al. Knowledge discovery and data mining: Towards a unifying framework. In: **KDD**. [S.l.: s.n.], 1996. v. 96, p. 82–88. Citado na página 10.

FELDMAN, R.; SANGER, J.; PRESS, C. U. **The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data**. Cambridge University Press, 2007. ISBN 9780521836579. Disponível em: <https://books.google.com.br/books?id=U3EA_zX3ZwEC>. Citado na página 11.

FELDMAN, S. Nlp meets the jabberwocky: Natural language processing in information retrieval. **ONLINE-WESTON THEN WILTON-**, Citeseer, v. 23, p. 62–73, 1999. Citado na página 5.

FERNANDES FERNANDO TIMOTEO; CHIAVEGATTO FILHO, A. D. P. **Perspectivas do uso de mineração de dados e aprendizado de máquina em saúde e segurança no trabalho**. 2019. Disponível em: <https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0303-76572019000101401>. Acesso em: 25 de abril de 2021. Citado na página 9.

FONTANA, É. Introdução aos algoritmos de aprendizagem supervisionada. Citado na página 8.

FUKUNAGA, K.; NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. **IEEE transactions on computers**, IEEE, v. 100, n. 7, p. 750–753, 1975. Citado na página 21.

GIOLO, S. R. **Introdução à análise de dados categóricos com aplicações**. [S.l.]: Editora Blucher, 2017. Citado na página 21.

GOOGLEDEVELOPERS. 2021. Disponível em: <https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space>. Acesso em: 11 de maio de 2021. Citado na página 17.

GUIDO, S.; MüLLER, A. C. **Introduction to Machine Learning with Python: a guide for data scientists.** Rio de Janeiro: O'reilly, 2016. Citado na página 7.

HAN, J.; KAMBER, M. Data mining: Concepts and techniques, 2nd editionmorgan kaufmann publishers. **San Francisco, CA, USA**, 2006. Citado na página 26.

HARPER, D. et al. Online etymology dictionary. 2001. Citado na página 6.

HOSEA, S.; HARIKRISHNAN, V.; RAJKUMAR, K. Artificial intelligence. **2011 3rd International Conference on Electronics Computer Technology**, v. 4, p. 124–129, 2011. Citado na página 4.

HOTHO, A.; NÜRNBERGER, A.; PAASS, G. A brief survey of text mining. In: CITESEER. **Ldv Forum**. [S.l.], 2005. v. 20, n. 1, p. 19–62. Citado na página 11.

ISLAM, M. et al. **Application of artificial intelligence (artificial neural network) to assess credit risk: a predictive model for credit card scoring**. 2009. Citado na página 4.

JUNIOR, J. R. C. Desenvolvimento de uma metodologia para mineração de textos. **Departamento de Engenharia Elétrica, Pontífica Universidade Católica do Rio de Janeiro**, 2007. Citado na página 17.

KAUFMAN, D. **A inteligência artificial irá suplantar a inteligência humana?** [S.l.]: ESTAÇÃO DAS LETRAS E CORES EDI, 2019. Citado na página 4.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, Amsterdam, v. 160, n. 1, p. 3–24, 2007. Citado na página 7.

LANGLEY, P. **Machine learning as an experimental science**. [S.l.]: Springer, 1988. Citado na página 6.

LEWIS, M. et al. **Implementing the lexical approach: Putting theory into practice**. [S.l.]: Language Teaching Publications Hove, 1997. v. 3. Citado na página 12.

LORENA, A. C.; CARVALHO, A. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007. ISSN 21752745. Citado na página 20.

LOVINS, J. B. Development of a stemming algorithm. **Mech. Transl. Comput. Linguistics**, v. 11, n. 1-2, p. 22–31, 1968. Citado na página 14.

MCTEAR, M.; CALLEJAS, Z.; GRIOL, D. Affective conversational interfaces. In: **The Conversational Interface**. [S.l.]: Springer, 2016. Citado na página 15.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **Proceedings of Workshop at ICLR**, v. 2013, 01 2013. Citado na página 16.

MITCHELL, T. M. et al. Machine learning. McGraw-hill New York, 1997. Citado na página 13.

MONTEIRO, S. T.; RIBEIRO, C. H. Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 15, n. 3, p. 320–338, 2004. Citado na página 10.

MOURA, S. d. O. **Uma abordagem para a escolha do melhor método de seleção de instâncias usando meta-aprendizagem**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2015. Citado na página 22.

NEVES, R. d. C. D. d. Pré-processamento no processo de descoberta de conhecimento em banco de dados. 2003. Citado na página 15.

NING, S.; YAN, M. Discussion on research and development of artificial intelligence. In: IEEE. **2010 IEEE International Conference on Advanced Management Science (ICAMS 2010)**. [S.l.], 2010. v. 1, p. 110–112. Citado 2 vezes nas páginas 4 e 13.

PACIFICO, L. D.; BRITTO, L. F.; LUDERMIR, T. B. Reconhecimento de plantas medicinais através de características das folhas e aprendizagem de máquina. In: SBC. **Anais do XIV Brazilian e-Science Workshop**. [S.l.], 2020. p. 17–24. Citado na página 19.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011. Citado na página 28.

PETERMANN, R. J. et al. Modelo de mineração de dados para classificação de clientes em telecomunicação. Pontifícia Universidade Católica do Rio Grande do Sul, 2006. Citado na página 7.

PYLE, D. **Data preparation for data mining**. [S.l.]: morgan kaufmann, 1999. Citado na página 15.

RAJARAMAN, A.; LESKOVEC, J.; ULLMAN, J. D. **Mining Massive Datasets**. [s.n.], 2014. Disponível em: <http://infolab.stanford.edu/~ullman/mmds/book.pdf>. Citado na página 24.

RAJARAMAN, A.; ULLMAN, J. D. **Mining of massive datasets**. [S.l.]: Cambridge University Press, 2011. Citado na página 11.

RIBEIRO, C. H. C. A tutorial on reinforcement learning techniques. In: **Supervised Learning Track Tutorials of the 1999 International Joint Conference on Neuronal Networks**. [S.l.: s.n.], 1999. Citado na página 10.

SEBASTIANI, F. Text categorization. In: **Encyclopedia of Database Technologies and Applications**. [S.l.]: IGI Global, 2005. p. 683–687. Citado na página 11.

SHUANG, K. et al. Convolution–deconvolution word embedding: An end-to-end multi-prototype fusion embedding method for natural language processing. **Information Fusion**, v. 53, 06 2019. Citado na página 16.

SKINNER, B. F. Are theories of learning necessary? **Psychological review**, American Psychological Association, v. 57, n. 4, p. 193, 1950. Citado na página 6.

SOARES, M. V. B. et al. Pretext ii: descrição da reestruturação da ferramenta de pré-processamento de textos. São Carlos, SP, Brasil., 2008. Citado na página 17.

SUTTON, R. S.; BARTO, A. G. et al. **Introduction to reinforcement learning**. [S.l.]: MIT press Cambridge, 1998. v. 135. Citado na página 10.

TAVARES, L. G.; LOPES, H. S.; LIMA, C. R. E. Estudo comparativo de métodos de aprendizado de máquina na detecção de regiões promotoras de genes de escherichia coli. **Anais do I Simpósio Brasileiro de Inteligência Computacional**, p. 8–11, 2007. Citado na página 24.

TEIXEIRA, J. **O que é inteligência artificial**. [S.l.]: E-Galáxia, 2019. Citado na página 4.

VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer-Verlag New York, Inc, 1995. Citado na página 17.

VILELA, P. **DE CS Classificação de sentimento para notícias sobre a Petrobras no mercado financeiro**. Tese (Doutorado) — Tese, 2011. Citado na página 16.

WILBUR, W. J.; SIROTKIN, K. The automatic identification of stop words. **Journal of information science**, Sage Publications Sage CA: Thousand Oaks, CA, v. 18, n. 1, p. 45–55, 1992. Citado na página 14.