XX ENANCIB
21 a 25 Outubro/2019 – Florianópolis
A Ciência da Informação e a era da Ciência de Dados

ISSN 2177-3688

GT-8 – Informação e Tecnologia

COMPARAÇÃO ENTRE ALGORITMOS DE CLASSIFICAÇÃO APLICADOS NA PREDIÇÃO DE NOTÍCIAS DE JORNAIS *ON-LINE*

COMPARISON BETWEEN CLASSIFICATION ALGORITHMS APPLIED IN ON-LINE NEWSPAPERS NEWS PREDICTION

Lúcia Helena de Magalhães – Universidade Federal de Minas Gerais /IF SUDESTE MG
Fernanda Fernandes Matos - Universidade Federal de Minas Gerais
Renato Rocha Souza - Universidade Federal de Minas Gerais /FGV

Modalidade: Trabalho Completo

Resumo: Na internet existem informações imensuráveis e muitas possibilidades ainda não exploradas, como por exemplo, os artigos de notícias publicados no idioma português. Assim, buscar novos recursos, que sejam capazes de recuperar manchetes, de vários portais, e disponibilizá-las em classes, seria uma possibilidade muito eficiente e sofisticada para explorar informações. Deste modo, esta pesquisa tem como objetivo comparar os resultados obtidos pelos algoritmos Máquina de Vetor de Suporte, Árvore de Decisão, Regressão Logística, Floresta Aleatória, *Naive Bayese AdaBoost* na classificação de notícias, coletadas dos principais jornais *on-line*. Diferentes métodos são empregados para a classificação de textos, porém, podem apresentar desempenhos diferentes, sendo importante testá-los para verificar suas eficácias e assim, escolher o classificador que apresentar os melhores resultados. Utilizou-se para treino uma amostra de 50 notícias relacionadas a quatro assuntos diferentes (biologia, economia, eletricidade e futebol) e para o teste um corpus contendo 10 notícias. Os corpora foram coletados no dia 17 de julho de 2019. As métricas utilizadas para avaliar os algoritmos foram a Acurácia, Precisão, Revocação, F1-score e a Área sob a Curva de Característica de Operação do Receptor. Com base nos resultados das avaliações, conclui-se que os classificadores apresentaram excelentes resultados na predição desse tipo específico de base textual, com exceção do algoritmo *Naive Bayes* que não conseguiu alocar nenhuma notícia na classe correta.

Palavras-Chave: Classificação de textos; Algoritmos; Medidas de avaliação de sistemas de classificação.

Abstract: On the Internet, immeasurable information and many possibilities have not been yet explored, such as the news articles published in the Portuguese language. Therefore, finding new features that are capable of retrieving headlines from various news portals and making them available in classrooms is a very efficient and sophisticated way to explore information. Thus, this research aims to compare the results obtained by the algorithm of the Vector Support Machine, Decision Tree, Logistic Regression, Random Forest, Naive Bayes and AdaBoost in the classification of news texts, collected from the main *on-line* newspapers. Different methods are used for the classification of texts, however, they may present different performance rates. Therefore, it's important to verify their efficacies to choose the classifier that present the best results. A sample of 50 news items related to four different subjects (biology, economics, electricity and soccer) was used for training, and a corpus containing 10 news items was used for testing. The corpora were collected on July 17, 2019, and the metrics used to evaluate the algorithms were Accuracy, Precision, Revocation, F1-score and Area under the Receiver Operating Characteristic Curve. Based on the results of the evaluations, it was concluded that the classifiers presented excellent results in the prediction of this specific type of textual base, except for the Naive Bayes algorithm that could not allocate any news in the correct class.
Keywords: Text Classification; Algorithms; Measures for the evaluation of classification systems.

1 INTRODUÇÃO

As notícias de jornais *on-line* contêm uma enorme quantidade de dados em formato não estruturado que pode ser extraída e transformada em informações valiosas de acordo com a exigência do usuário (LAMA, 2013). Essa massa de dados tem aumentado cada vez mais, pois todos os dias, quintilhões de bytes de dados são gerados provenientes do uso de mídia social e interações digitais (DAS, 2017). Essa questão é cada vez mais importante no mundo dos negócios e da sociedade e fez com que a grande rede mundial de computadores se tornasse um interessante objeto de estudo.

Na internet existem informações imensuráveis e muitas possibilidades ainda não exploradas, como por exemplo, os artigos de notícias publicados no idioma português. Essas publicações são fontes importantes de informação que mantêm as pessoas atualizadas com os acontecimentos atuais do mundo (LAMA, 2013). Porém, muitas vezes, a notícia de um único portal não é suficiente para obter todo o conhecimento desejado. Assim, é necessário recorrer a vários sites em busca de manchetes semelhantes, todavia, essa tarefa não é tão simples. Segundo Liu (2007), os noticiários *on-line* geram diariamente uma grande quantidade de informes e para fornecer um serviço de notícias integrado, os artigos coletados deveriam ser dispostos em uma hierarquia de tópicos. Mas como essas notícias podem ser organizadas? Uma das possibilidades seria empregar um grupo de editores humanos para fazer o trabalho,

no entanto, a organização manual é dispendiosa e demorada, o que se torna inadequada para notícias e para outras informações que são sensíveis ao tempo.

Assim, a classificação seria uma opção para organizar artigos de notícias de acordo com categorias predefinidos. Por conseguinte, buscar novos recursos, que sejam capazes de recuperar manchetes de vários portais de notícias, e disponibilizá-las em classes, seria uma possibilidade muito eficiente e sofisticada para explorar informações. Por isso, a importância de um estudo na área de classificação de textos, com o objetivo de desenvolver estratégias para classificar, automaticamente e de forma inteligente, informações importantes recuperadas dos principais sites de notícias e representá-las em classes, facilitando, deste modo, o acesso aos informes atuais.

Com esse intuito, este trabalho tem como objetivo comparar os resultados obtidos pelos algoritmos Máquina de Vetor de Suporte, em inglês *Support Vector Machine* (SVM), Árvore de Decisão, Regressão Logística, Floresta Aleatória, *Naive Bayes* e *AdaBoost* na classificação de notícias publicadas no idioma português. Para a avaliação do desempenho dos classificadores foram usadas as métricas Acurácia (*Accuracy*), Precisão (*Precision*), Revocação (*Recall*), *F1-score* e Área sob a Curva *Receiver Operating Characteristic* (ROC).

Diversos métodos são empregados para a classificação de textos, porém, podem apresentar desempenhos diferentes, sendo importante testá-los para verificar suas eficácias e, desta forma, selecionar o classificador que apresentar os melhores resultados. Para isso, foram utilizadas 60 amostras de notícias relacionadas a quatro assuntos (futebol, economia, eletricidade e biologia) coletadas em junho de 2019. Para testar os classificadores, foram coletadas 10 notícias usando para a busca a palavra-chave “Copa América 2019” com o intuito de verificar se o algoritmo seria capaz de predizer que essas notícias pertencem à classe futebol.

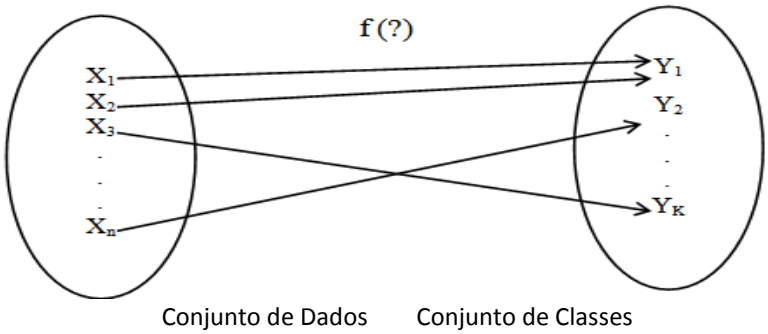
Para melhor entendimento deste artigo, serão conceituados alguns termos e apresentados os principais algoritmos utilizados neste estudo.

2 CLASSIFICAÇÃO

O processo de inserção dos documentos em classes é comumente conhecido como classificação de textos. “A classificação de textos provê um meio para organizar a informação que permite melhor compreensão e interpretação dos dados” (GONÇALVES, 2013, p.278). Dessa forma, com a finalidade de aprimorar os recursos para organização da informação,

estudos foram realizados e algoritmos foram desenvolvidos para a indução automática de sistemas capazes de lidar com problemas de classificação. De acordo com Goldschmidt, Passos e Bezerra (2015, p.89), a tarefa de classificar pode ser compreendida como "a busca por uma função que permita associar corretamente cada registro X_i de um conjunto de dados a um único rótulo categórico $\mathbf{Y_i}$, denominado classe. Uma vez identificada, esta função pode ser aplicada a novos registros de forma a prever as classes em que tais registros se enquadram". A figura 1 ilustra as associações entre registros de dados e suas respectivas classes.

Figura 1: Associações entre registros de dados e classes



Fonte: Adaptada de Goldschmidt, Passos e Bezerra (2015, p.89)

Nesse contexto, Goldschmidt, Passos e Bezerra (2015, p.89) formalizaram a classificação da seguinte forma:

> Considere um conjunto de pares ordenados em que cada par é da forma *(x, f(x))*, onde *x* é um vetor de entrada n-dimensional e *f(x)* a saída de uma função *f*, desconhecida, aplicada a *x*. A tarefa de classificação consiste em, dada uma coleção de exemplos de *f*, obter uma função (hipótese) *h* que seja uma aproximação de *f*. A imagem de *f* é formada por rótulos de classes retirados de um conjunto finito e toda hipótese *h* chamada de Classificador. O aprendizado consiste na busca pela hipótese *h* que mais se aproxime da função original *f*. (GOLDSCHMIDT; PASSOS; BEZERRA, 2015, p.89).

De acordo com Gonçalves (2013, p.278), a classificação de textos provê um meio de organizar a informação que permite melhor compreensão e interpretação de dados. Em suma, nesse processo, o algoritmo de aprendizado constrói um classificador capaz de determinar corretamente a classe de novos exemplos ainda não etiquetados, dado um conjunto de classes e um conjunto de exemplos de treinamento. Na maioria dos casos, a técnica pode usar o conhecimento de um domínio para fornecer alguma informação previamente conhecida como entrada ao indutor. E, após induzido, o classificador é normalmente avaliado e a ação de classificar pode ser repetida, se necessário (MONARD; BARANAUSKAS, 2003).

Assim, "após a etapa de treinamento, tem-se um classificador que deve ser capaz de predizer corretamente o rótulo de novos exemplos, que ainda não foram rotulados" (REZENDE, 2005 apud BORGES, 2012, p.8). De acordo com Monard e Baranauskas (2003), normalmente, um conjunto de exemplos é dividido em dois subconjuntos: o conjunto de treinamento é utilizado para o aprendizado do conceito e o conjunto de testes usado para medir o grau de efetividade do conceito aprendido. "Esses conjuntos são normalmente disjuntos para assegurar que as medidas obtidas, utilizando o conjunto de testes, sejam de um conjunto diferente do usado para realizar o aprendizado, tornando a medida estatisticamente válida" (MONARD; BARANAUSKAS, 2003, p.97). O processo de classificação de textos envolve as etapas de pré-processamento, extração de características, classificação (predição) e a avaliação dos resultados.

2.1 Pré-processamento e Normalização dos dados

Antes de realizar a classificação, os documentos precisam ser processados e normalizados. Isso envolve as seguintes fases:

a) Limpeza dos textos e Remoção de caracteres especiais

As notícias contêm conteúdos desnecessários, como símbolos irrelevantes, caracteres especiais*, tags* XML e HTML que não agregam valor na análise, portanto, eles devem ser removidos. Outra importante tarefa na limpeza e normalização de documentos é remover caracteres especiais como parênteses, colchetes e números, pois eles aumentam o ruído nos textos. Expressões regulares são usadas nesse processo.

b) *Stemming* or *lemmatization*

Segundo Morais e Ambrósio (2007), *stemming* é uma técnica de normalização linguística, na qual as variantes de um termo são reduzidas a uma forma comum denominada *stem* (radical). Isso resulta na eliminação de prefixos, sufixos e características de gênero, número e grau das palavras, reduzindo o número de atributos. Na visão de Monteiro e Gomes et al. (2006), *stemming* consiste na remoção de variações de palavras, tais como plural, gerúndio, afixos, gênero e número, de modo que a palavra fique somente com a *stem*, ou seja, com o radical. Já o processo de lematização é muito parecido com o *stemming*, os afixos

também são removidos, porém, obtém-se o lema da palavra e não o radical. A diferença é que o radical nem sempre é uma palavra lexicograficamente correta; isto é, pode não estar presente no dicionário. Já o lema, sempre estará presente no dicionário (SARKAR, 2019).

c) Remoção das *stopwords*

Stopwords são palavras irrelevantes e insignificantes que aparecem em uma linguagem para ajudar a construir sentenças, mas que não representam nenhum conteúdo nos documentos (LIU, 2007). São palavras comuns, que repetem muitas vezes num texto e não acrescentam e nem retiram informações relevantes (SOLKA, 2007). Normalmente, são artigos, conjunções e preposições que aparecem no corpus e que, apesar de ocorrerem com muita frequência em documentos, elas não são essenciais para dar sentido ao texto, pois são usadas apenas para juntar palavras em uma frase. Devido à sua alta frequência de ocorrência, sua presença apresenta, muitas vezes, como um obstáculo na análise, por isso, a necessidade de removê-las.

2.2 Extração de características

De acordo com Sarkar (2019), em aprendizado de máquina, as características ou recursos são propriedades ou atributos únicos mensuráveis de cada observação ou ponto de dados em um *dataset*. Normalmente, recursos extraídos são usados pelos algoritmos de *Machine Learning* para encontrar padrões de aprendizado que podem ser aplicados em novos pontos de dados para obter insights. Esses algoritmos geralmente esperam características na forma de vetores numéricos. Sendo assim, para a análise das notícias, os documentos precisam ser transformados e representados em um formato numérico para que os algoritmos de aprendizado de máquina possam compreender as informações contidas nos textos. Assim, ao trabalhar com dados textuais, há o desafio adicional de descobrir como extrair recursos numéricos a partir deles. Sarkar (2019) apresenta alguns modelos de extração de recursos em documentos textuais, são eles:

a) Modelo *Bag-of-Words*

Normalmente, nesse modelo, o texto é visualizado como uma sequência de palavras e é dividido em *tokens*. Para isso, o conteúdo de cada documento é decomposto em termos e é verificada a frequência de cada vocábulo no texto. Geralmente, o processo é aplicado em um

conjunto de documentos e a coleção é transformada em uma matriz atributo-valor, na qual cada linha representa um documento do conjunto, e cada documento é descrito pelos valores dos atributos mais representativos da coleção (SOARES; PRATI; MONARD, 2009). O resultado é uma matriz de frequência, que também é conhecida como ***Bag-of-Words*** (BoW), que é a representação numérica do corpus. Nesse modelo, a ordem e a sequência de palavras não são consideradas e "os termos são considerados independentes, formando um conjunto desordenado em que a ordem de ocorrência das expressões não importa" (NOGUEIRA, 2009, p.16). Em suma, o BoW é um nome mais elegante para as matrizes de frequência, é uma forma de representação de texto que computa a ocorrência dos termos em um documento.

b) Modelo of N-Gramas

Um N-Gramas é basicamente um conjunto de palavras de um documento de textos, de modo que esses *tokens* são contíguos e ocorrem em uma sequência. Desta forma, Bi-gramas indicam n-gramas de ordem 2 (duas palavras), tri-gramas indicam n-gramas de ordem 3 (três palavras) e assim por diante (SARKAR, 2019). Esse modelo é apenas uma extensão do BoW.

c) Modelo *Term Frequency – Inverse Document Frequency*

O *Term Frequency Inverse Document Frequency* (TF-IDF) é uma medida estatística usada para avaliar o quão importante uma palavra é para um documento em relação a uma coleção de documentos. Esse modelo é a combinação de duas métricas: *Term Frequency* (TF) e *Inverse Document Frequency (*IDF.). O TF mede com que frequência um termo ocorre em um documento e o IDF mensura o quanto um termo é importante (LIU *et al*. 2007). Segundo Baeza-**Yates** e Ribeiro-Neto (2013, p.35), a primeira forma de ponderação da frequência dos termos foi proposta por Luhn em 1957, e baseia-se na seguinte suposição: "o valor de peso de um termo k_i que ocorre em um documento d_j é simplesmente proporcional à frequência do termo $f_{i,j}$. Isto é, quanto mais frequentemente um termo k_i ocorrer no texto do documento d_j maior será a sua frequência de termo $TF_{i,j}$". De acordo com esses autores, essa hipótese baseia-se na observação que termos com alta frequência são importantes para descrever os tópicos-chave de um documento. Já o TF-IDF é um cálculo que mostra o valor que cada palavra tem em cada texto. A fórmula para computá-lo é a frequência do termo num texto vezes o inverso da frequência desse vocábulo em todos os textos. Deste modo, "uma palavra que

aparece muitas vezes num texto não terá tanto valor se aparecer igualmente muitas vezes nos outros textos. Este método ajuda a distinguir palavras relevantes para o texto de palavras que são comuns" (RODRIGUES, 2016, p.21).

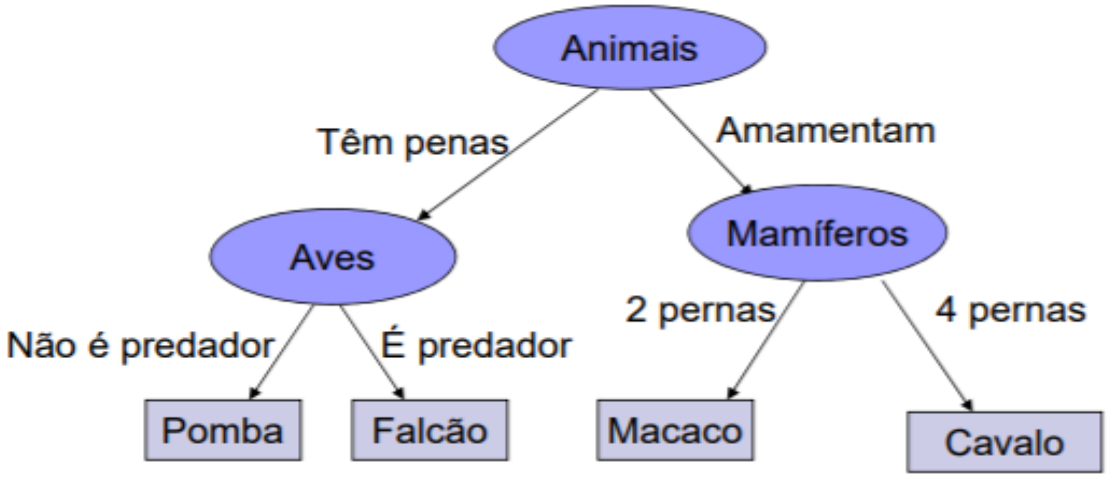
2.3 Algoritmos de classificação

Existem na literatura diversos algoritmos de aprendizado supervisionado para classificação de textos. Segundo Gonçalves (2013), os algoritmos supervisionados de classificação de textos mais representativos são: Árvore de decisão, *Naive Bayes*, SVM, Floresta Aleatória, Regressão Logística e *AdaBoost*. Abaixo, uma descrição dos principais métodos existentes.

a) Árvore de decisão

Para Faceli *et. al* (2017, p.83), "uma árvore de decisão usa a estratégia de dividir para conquistar para resolver um problema de decisão". Um classificador de texto desse tipo é uma árvore em que os nós internos são rotulados pelos termos, os ramos que partem dos nós são definidos pelos testes, levando-se em consideração o peso que o termo tem no teste do documento e as folhas pelas categorias. Segundo Prates (2018), é uma ferramenta de suporte a tomada de decisão que usa um gráfico no formato de árvore e demonstra visualmente as condições e as probabilidades para se chegar a resultados. A Figura 3 ilustra uma árvore de decisão, ou seja, um modo de representar o conhecimento.

Figura 2: Exemplo de árvore de decisão



Fonte: Lobo (2010, p.2)

b) Máquinas de Vetores de Suporte

SVM é um conjunto de métodos de aprendizado supervisionado utilizados tanto para classificação quanto para regressão. Segundo Ticom (2007), é um dos mais populares classificadores do tipo linear. Ela implementa a ideia de que seja construído um hiperplano com base no mapeamento dos vetores de entrada em um espaço de características com uma grande quantidade de dimensões. Quando os dados do arquivo de treino são separáveis, a taxa de erro para o SVM pode ser definida pela equação (TICOM, 2007, p.29): $h = R^2 / M^2$, no qual R é o raio da menor esfera que contém os dados de treinamento e M é a margem que significa a distância entre o hiperplano e o vetor de treino mais perto do espaço de características.

c) Random Forest

A Floresta Aleatória é uma técnica que permite obter modelos muito eficazes sem nenhuma preparação de dados ou conhecimento de modelagem (BREIMAN; CUTLER, 2014). Como o nome já diz, o algoritmo cria uma floresta de um modo aleatório. De acordo com Han Kamber e Pei (2012), uma *Random Forest* pode ser descrita como um classificador formado por um conjunto de árvores de decisão $\{h(\mathbf{X}, v_k), k, 1, ..\}$, no qual v_k são vetores aleatórios amostrados de forma independentes, distribuídos igualmente em todas as árvores da floresta. O resultado do processo de classificação é a classe X com maior número de votos dentre todas as árvores consideradas.

d) Naive Bayes

É um método probabilístico, no qual se assume que todas as variáveis são independentes da variável de classificação, o que o torna muito fácil para criar uma rede estruturada e não obriga a geração de um algoritmo de aprendizado. Este classificador se baseia no teorema de *Bayes* com a simplificação de que, após o treinamento, pode ser assumido que as características são independentes para uma dada classe (TICOM, 2007, p.28).

Em reconhecimento de padrões ou classificação, interessa-se em associar a um padrão uma classe. Assim, sejam padrões $\mathbf{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$ e classes $\{\omega_1, \omega_2, ..., \omega_c\}$, a abordagem Bayesiana supõe que as probabilidades de cada classe $P(\omega_i)$ e as densidades de probabilidade condicionais $p(x|\omega_i)$ de **x** com respeito a cada uma das classes $\omega_i, i = 1, 2, ..., c$, são conhecidas. Na ausência de qualquer outra informação, pode-se classificar um padrão **x** como sendo da

classe ω_i de maior probabilidade. Entretanto, dado que **x** foi observado, isto parece uma decisão muito ingênua, pois acertaria a classificação com probabilidade $P(\omega_i)$, porém erraria com probabilidade $\sum_{j \neq i} P(\omega_j)$. Como tem as condicionais, pode-se utilizar o teorema de Bayes e calcular a probabilidade $P(\omega i \,|\mathbf{x})$, ou seja, $P(\omega_i|\mathrm{x}) = \frac{\boldsymbol{P(\omega_i)}\mathbf{p(x|\omega_i)}}{\boldsymbol{p}(\mathbf{x})}$ na qual $P(\omega_i)$ é a priori, $p(\mathbf{x}|\omega_i)$ é a densidade condicional ou verossimilhança, $p(\mathbf{x}) = \sum_{j=i}^{c} P(\omega j)p(x|\omega j)$ é a evidência e $P(\omega_i|\mathrm{x})$ é a posteriori, e tomar a decisão baseada nesses a posteriores (HIRATA, 2007).

e) Regressão Logística

Segundo Nisbet, Elder e Miner (2009), a regressão logística é utilizada para modelar a relação não linear de uma variável dependente e os efeitos combinados de variáveis independentes. Para esses autores, essa relação representa a probabilidade de ocorrência de um evento. Diferente de uma regressão linear simples, os valores observados para a variável dependente, quando colocados num plano cartesiano, não formam uma nuvem de pontos, mas ficam restritos a zero e a 1. Deste modo, o que se faz com a regressão logística é atribuir escores calculados a partir das variáveis independentes, a fim de que, quando a variável dependente pertencer ao grupo zero, ela tenha um baixo escore, enquanto os escores associados às variáveis dependentes do grupo 1 devem ser mais altos (SOARES; REBOUÇAS, 2015).

f) Adaboost

Segundo Nascimento (2011), o *Adaboost* é um algoritmo de aprendizado supervisionado do tipo *boost* que combina um conjunto de funções simples de classificação, denominadas classificadores fracos, para formar um classificador forte. Para a autora, um classificador fraco é uma estrutura simples que contém um vetor de características f, um limiar e uma paridade. Durante o treinamento do classificador fraco deve ser encontrado um limiar que melhor separe o valor de uma característica de exemplos definidos como positivos dos negativos.

2.4 Avaliação dos modelos de classificação

Segundo Sarkar (2019), o desempenho dos modelos de classificação geralmente é baseado em quão bem eles preveem resultados para novos pontos de dados. Assim, os

documentos são classificados a partir de características do texto, como termos ou palavras presentes nas notícias. Baseia-se na análise prévia de um conjunto de amostragem ou de treinamento, contendo objetos corretamente classificados. E o desempenho, normalmente, é medido em relação a um conjunto de teste que consiste em elementos que não foram usados para influenciar ou treinar o classificador. Assim sendo, na avaliação, é necessário escolher métricas corretas que possam pontuar qual é o melhor modelo de classificação. Para isso, recursos são extraídos ao treinar o modelo. As informações obtidas pela amostra já treinada são usadas para fazer previsões de novos textos ainda não etiquetados. Essas previsões são então combinadas com os rótulos reais para ver o quanto o modelo previu. Várias métricas determinam o desempenho de previsão de um modelo, mas este estudo concentrou nas seguintes medidas: Acurácia, Precisão, Revocação, *F1-score* e Área sob a Curva ROC.

a) Acurácia

Segundo Baeza-Yates e Ribeiro-Neto (2013) acurácia é a fração dos documentos de treinamento atribuídos a suas classes corretas pelo classificador. Medidas acuradas ou exatas são aquelas cujo valor médio se aproxima do valor correto. Sarkar (2019) define acurácia como a exatidão geral ou a proporção de previsões corretas do modelo.

b) Precisão

Baeza-Yates e Ribeiro-Neto (2013) conceituam precisão como a fração de documentos relevantes do total recuperados. Segundo Sarkar (2019), precisão é definida como o número de prognósticos feitos que são realmente corretos ou relevantes de todas as previsões baseadas na classe positiva. A capacidade de precisão, ou relevância, está relacionada ao número de documentos recuperados para atendimento das solicitações encaminhadas pelo usuário. Também pode ser mensurada por meio da relação entre os documentos relevantes recuperados e número total de documentos recuperados (FUJITA, 2009).

c) Revocação

Baeza-Yates e Ribeiro-Neto (2013) definem revocação como a fração dos documentos relevantes recuperados. A capacidade de revocação diz respeito ao número de documentos recuperados e pode ser mensurada por meio da relação entre o número de documentos

relevantes sobre determinado tema, recuperados pelo sistema de busca, e o número total de documentos sobre o assunto, existentes nos registros do mesmo sistema (FUJITA, 2009). Para Sarkar (2019), revocação pode ser definida como o número de instâncias da classe positiva que foram corretamente preditas.

d) F1-score

A pontuação F1 é outra medida de precisão que é calculada tomando a média harmônica da precisão e da revocação (SARKAR, 2019). Sendo assim, quando se tem *F1-Score* baixo, é um indicativo de que ou a precisão ou a revocação está baixa.

e) Área sob a Curva ROC

Segundo Prati, Batista e Monard (2008), a análise ROC é um método gráfico para avaliação, organização e seleção de sistemas de diagnóstico e/ou predição. Essa técnica foi introduzida em Aprendizagem de Máquina e Mineração de Dados como uma ferramenta útil e poderosa para a avaliação de modelos de classificação. Ela é particularmente útil em domínios nos quais existe uma grande desproporção entre as classes ou quando se deve levar em consideração diferentes custos/benefícios para os diferentes erros/acertos de classificação.

3 METODOLOGIA

Para a realização desta pesquisa de natureza quali-qualitativa e de caráter experimental, o primeiro passo foi coletar os informes dos principais jornais *on-line*. Para isso, utilizou-se o *Mediaframe*[1]. O *MediaFrame* é um projeto da Fundação Getúlio Vargas que permite pesquisar e capturar um grande número de notícias dos principais jornais *on-line*. Como é um sistema de acesso restrito, uma outra opção para a coleta dos dados seria o desenvolvimento de um *web crawler* que fosse capaz de rastrear a internet e extrair de forma automatizada as informações desejadas, pois a captura manual seria dispendiosa e demorada.

Para a realização deste trabalho, recuperaram-se dois conjuntos de notícias, um utilizado para treino e o outro para teste. E para o processamento dos textos, usou-se o *Orange Canvas*[2], um software *open source* que além de análise e visualização de dados, possui

[1] https://mediaframe.io
[2] https://orange.biolab.si/

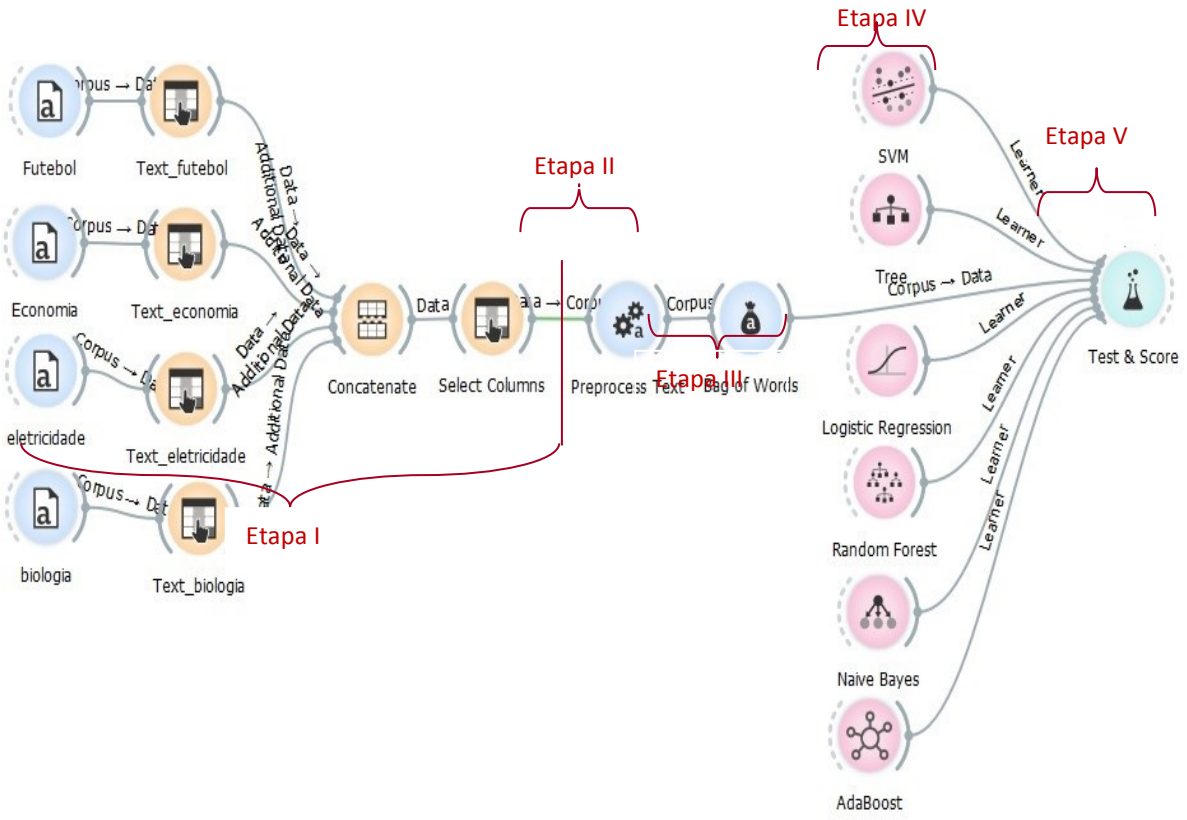
um módulo para *text mining*. Assim, pode-se utilizar os algoritmos de *Data Mining* e *Text Mining* com programação visual (*widgets*) que já vem incorporado no programa, além da possibilidade de incrementar novas funcionalidades através de *scripts* em Python.

O experimento foi feito da seguinte forma: a base total é composta por 60 documentos que foram separados aleatoriamente em bases menores formando o corpus com 13 notícias relacionadas a futebol, 12 sobre biologia, 12 sobre eletricidade e 13 relacionadas à economia, formando uma amostra de 50 notícias que foram usadas para o treino. Para a realização dos testes, foram usadas 10 matérias relacionadas ao tema futebol. Para melhor entendimento, será explanado separadamente as duas fases do processo de classificação: Treino e Teste.

3.1 Treino do modelo

Na fase de aprendizado, o sistema aprende e constrói uma base de conhecimento a partir de um conjunto de amostras que lhe são fornecidos. A figura 4 apresenta o fluxograma da metodologia proposta para a fase de treino na classificação de notícias disseminadas pelos meios de comunicação em massa:

Figura 3: Fluxograma da metodologia na fase de treino



Fonte: elaborado pelos autores usando o software Orange Canvas

As etapas da fase de treinamento, ilustradas no fluxograma, serão descritas para melhor entendimento.

3.1.1 Construção do Corpus de notícias para treinamento

Esta etapa tem como objetivo coletar, dos principais sites de notícias, uma amostra de documentos para a realização da análise. Para isso, utilizou-se o *Mediaframe*. Primeiramente, define-se a palavra-chave usada na consulta. Em seguida, escolhem-se os principais jornais para a busca e determina-se o período desejado em que as notícias foram publicadas.

Posteriormente, o *Mediaframe* rastreia cada uma das fontes para descobrir os informes relacionados ao assunto pesquisado. Depois, o sistema permite fazer o download do conjunto de notícias recuperado em formato csv. Sendo assim, primeiro, coletou-se a amostra para o treinamento, a coleção foi formada por quatro classes diferentes (economia, futebol, eletricidade e biologia). Cada matéria contém as seguintes informações: id, website, data da publicação, título e o texto. Porém, neste trabalho, foram selecionadas somente as colunas do título e do conteúdo da notícia. Em seguida, os corpora foram concatenados formando um único corpus contendo as quatro classes.

3.1.2 Pré-processamento das notícias

Devido à natureza textual não estruturada, os documentos necessitam de um pré-processamento para serem submetidos a algoritmos de aprendizagem. A transformação dos documentos em uma representação mais adequada é uma etapa de suma importância, visto que a representação desses documentos tem uma influência fundamental em quão bem um algoritmo de aprendizado poderá generalizar a partir dos exemplos (SEBASTIANI, 2002).

Diante disso, os conteúdos das notícias, acompanhados de seus títulos, são processados usando técnicas de pré-processamento de documentos. Esse processo inclui as seguintes etapas:

a) Limpeza

Nesta etapa o texto foi convertido para minúsculo, acentos, *tags* HTML e URL foram removidos. Além disso, usaram-se expressões regulares para eliminar caracteres especiais e números.

b) Tokenização

A tokenização ou *tokenization* é uma etapa importante para o pré-processamento. Nesta fase, as notícias recuperadas pelo *Mediaframe* são submetidas a inúmeras operações para serem representadas estruturalmente, pois antes de realizar qualquer análise, é necessário normalizar os documentos de texto.

Para isso, primeiramente, serão identificadas nas amostras de notícias as palavras mais importantes, isto é, que melhor representam as ideias dos textos. Esse processo de identificação das palavras é conhecido como *tokenization*. O procedimento divide uma unidade de documento em pedaços, denominados *tokens*.

c) Remoção das stopwords

Nesta etapa, realizou-se a eliminação das *stopwords*, que são palavras que têm pouca ou nenhuma significância. Assim, os artigos, os pronomes, as preposições e as interjeições são consideradas *stopwords*. Elas geralmente são removidas do texto durante o processamento, de modo a reter os termos mais relevantes. Como não existe uma lista universal de *stopwords*, cada domínio ou idioma pode ter seu próprio conjunto. Neste trabalho, foram adotadas *stopwords* para o idioma Português. Além disso, para reduzir os termos de pouca relevância nos textos, a lista foi cuidadosamente atualizada com outros vocábulos, tais como verbos genéricos ou substantivos sem muita importância, que foram cuidadosamente selecionados depois de analisar o corpus.

Portanto, a fase de remoção das *stopwords* é importante para reduzir o número de palavras envolvidas nos processos posteriores de análise, de modo a conseguir um melhor desempenho sem perda significativa de informação útil.

d) Stemming

O passo seguinte é identificar e unificar termos que possuam o mesmo significado semântico. Muitas vezes, prefixos e sufixos são anexados a um tronco de palavras para mudar seu significado ou criar uma nova expressão. O objetivo desta etapa é remover esses afixos e retornar as palavras em sua forma básica, ou seja, reduzir o termo a sua raiz.

3.1.3 Representação do Modelo de documentos

Uma importante etapa no processo de classificação é representar o conteúdo do documento sob a forma de expressão matemática para posterior análise e processamento. A representação de textos pode ser feita usando diversas abordagens, entre elas, a BoW. Portanto, nesta etapa, com o corpo de notícias normalizado, uma matriz de característica é construída. Para isso, os *tokens* dos textos são mantidos nos documentos normalizados e as características são extraídas, com base no modelo TF-IDF, de modo que cada característica ocorra em pelo menos 25% dos documentos e no máximo 85% dos documentos. Para controlar a porcentagem, são usadas as frequências mínima e máxima dos termos no documento.

3.1.4 Escolha dos algoritmos de classificação

Nesta pesquisa foi avaliado o desempenho dos algoritmos SVM, Árvore de Decisão, Regressão Logística, Floresta Aleatória, *Naive Bayes* e *AdaBoost.*

3.1.5 Avaliação dos algoritmos de classificação

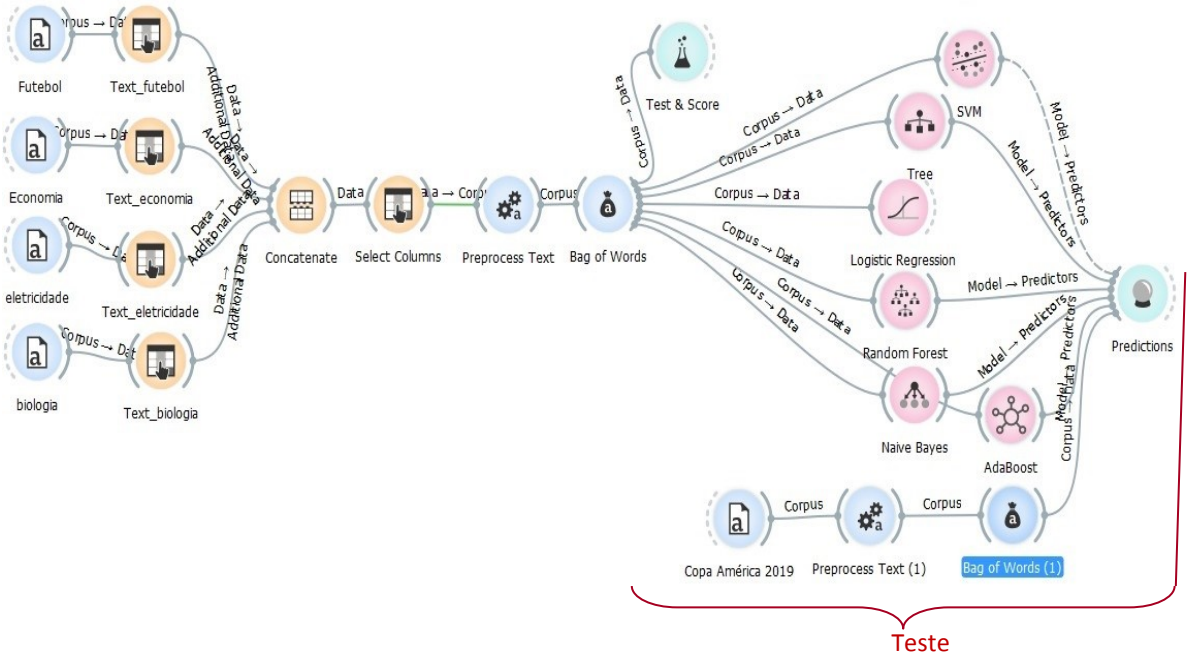
Existem diversas maneiras de se avaliar o processo de classificação como um todo, seja de forma qualitativa ou quantitativa. A utilização de métricas é considerada uma forma quantitativa, ao passo que a utilização do conhecimento de especialistas no domínio é considerada uma forma qualitativa. Para a avaliação dos algoritmos de classificação, utilizaram-se as métricas de Precisão, Acurácia, Revocação, F1-score e Área sob a Curva ROC.

3.2 Teste do modelo

Na etapa de teste, o modelo utiliza o conhecimento adquirido na fase de treino para classificar os documentos cujas classes são desconhecidas.

A figura 5 apresenta o fluxograma da metodologia proposta após acréscimo da fase de teste na classificação de notícias coletadas dos principais jornais *on-line*. Neste experimento, realizou-se o teste com uma coleção de 10 notícias relacionadas ao tema futebol.

Figura 4: Fluxograma da metodologia incluindo a fase de teste



Fonte: Elaborada pelos autores usando o software Orange Canvas

As fases necessárias para testar os algoritmos são descritas abaixo:

3.2.1 Construção do Corpus de notícias para teste

Nesta etapa, coletaram-se 10 notícias usando na busca a palavra-chave “Copa América 2019”, assunto relacionado com a classe futebol. De posse do corpus, os textos passaram por todas as etapas do pré-processamento (limpeza, tokenização, remoção das *stopwords* e *stemming*) conforme realizado na coleção de treino. Em seguida, as notícias foram representadas no modelo *BoW*.

3.2.2 Predição

Neste passo, o classificador usa a base de conhecimento adquirida na fase de treinamento para classificar cada item da coleção de teste.

3.2.3 Análise e validação dos resultados

Esta etapa tem como objetivo verificar se o algoritmo foi capaz de predizer se as notícias utilizadas na fase de teste pertencem à classe futebol.

4 ANÁLISE DOS RESULTADOS

Para avaliar o desempenho de técnicas de classificação aplicadas a corpus de notícias, construiu-se um modelo de classificação para testar cinco algoritmos: SVM, Árvore de Decisão, Regressão Logística, Floresta Aleatória, *Naive Bayes* e *AdaBoost*. A tabela 1 mostra os valores de acertos dos algoritmos ao serem submetidos à análise de curvas ROC e avaliados usando as métricas de Precisão, Acurácia, Revocação e *F1-score*.

Tabela 1: Avaliação dos algoritmos de classificação

Método	AUC	Accuracy	F1-score	Precision	Recall
Decision Tree	0.818	0.842	0.770	0.709	0.842
SVM	0.664	0.460	0.446	0.446	0.460
Random Florest	0.953	0.932	0.916	0.934	0.932
Naive Bayes	0.562	0.250	0.208	0.261	0.250
Logist Regression	0.983	0.957	0.954	0.959	0.957
AdaBoost	0.954	0.979	0.979	0.980	0.979

Fonte: Elaborada pelos autores

Observa-se que os algoritmos Floresta Aleatória, Regressão Logística e *AdaBoost* apresentaram um melhor desempenho. O classificador *Naive Bayes* e a SVM tiveram os piores resultados. Nota-se também que muitos dos algoritmos mostraram desempenhos satisfatórios para o problema de classificação de notícias, pois obtiveram precisão e acurácia superior a 0.9. Em contrapartida, *Naive Bayes* obteve um valor baixo em todas as métricas, portanto, não é indicado para esse tipo de textos. Porém se faz necessário novos experimentos com novas coleções, aumentando e reduzindo a quantidade de documentos para, assim, analisar se o desempenho dos algoritmos continua semelhante com outros corpora.

Ao efetuar o teste usando um corpus sem rótulo, os algoritmos apresentaram alta taxa de acerto, a Figura 6 ilustra o resultado.

Figura 5: Resultado da predição

	Tree	Random Forest	Naive Bayes	AdaBoost	SVM	url	publication_date	title	text
1	0.05 : 0.05 → classe (1)	0.00 : 0.00 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.05 : 0.04 → classe (1)	https://g1.glob...	2019-07-07T07:...	Copa América: ...	A grande final ...
2	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.20 : 0.21 → classe (1)	https://epoca.g...	2019-07-07T19:...	Crônica | Brasil ...	Depois de trinta...
3	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.03 : 0.04 → classe (1)	https://brasil.el...	2019-07-07T18:...	AO VIVO | Brasil...	Brasil x Peru se ...
4	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.03 : 0.04 → classe (1)	http://agenciab...	2019-07-04T01:...	Lesão tira Willia...	A Confederaçã...
5	0.05 : 0.05 → classe (1)	0.00 : 0.00 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.02 : 0.04 → classe (1)	https://g1.glob...	2019-07-04T23:...	Grupo Lincoln ...	O grupo Lincol...
6	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.03 : 0.04 → classe (1)	https://brasil.el...	2019-07-06T17:...	AO VIVO | Arge...	Argentina x Chi...
7	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.07 : 0.07 → classe (1)	https://epoca.g...	2019-07-08T08:...	Crônica | Apesa...	Não foi uma ca...
8	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.02 : 0.04 → classe (1)	https://www1.f...	2019-07-07T22:...	Baixe pôster da ...	A Folha oferece...
9	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.02 : 0.04 → classe (1)	https://www1.f...	2019-07-07T22:...	Baixe pôster da ...	A Folha oferece...
10	0.05 : 0.05 → classe (1)	0.00 : 0.07 → classe (1)	0.00 : 1.00 → classe (3)	0.00 : 0.00 → classe (1)	0.03 : 0.04 → untitled (1)	https://brasil.el...	2019-07-05T20:...	Onde e como a...	Argentina x Chi...

Fonte: Elaborada pelos autores usando o software Orange Canvas

Para melhor compreensão da Figura 6, elaborou-se um quadro com as legendas das classes.

Quadro 1: Legenda da Figura 11

LEGENDA	
Classe (0)	Biologia
Classe (1)	Futebol
Classe (2)	Eletricidade
Classe (3)	Economia

Fonte: Elaborada pelos autores

Percebe-se na Figura 6 que os algoritmos Árvore de Decisão, Floresta Aleatória, *AdaBoost* e SVM acertaram 100% ao classificar todas as notícias como pertencente à classe (1), ou seja, futebol. Já o algoritmo *Naive Bayes* errou 100%, pois classificou toda a coleção de notícias como pertencente à classe (3), isto é, economia. Logo, a maioria dos algoritmos conseguiram um excelente desempenho na classificação de um corpus pequeno de notícias.

5 CONSIDERAÇÕES FINAIS

A classificação de textos é uma das mais importantes aplicações do processamento de linguagem natural e tem grande utilidade quando se deseja organizar documentos. Os resultados desta pesquisa mostram que os classificadores tiveram boa performance na classificação de notícias. E, se comparado com a classificação manual, o ganho de tempo é muito grande. Porém, o desempenho é significativo quando se trabalha com um número pequeno de termos, por isso, a importância do pré-processamento, pois essa etapa consegue um ganho significativo da redução da dimensionalidade dos textos. Isso faz com que os algoritmos de aprendizagem de máquina aprimorem a precisão da classificação além de reduzir o custo computacional.

Ao analisar os resultados das validações, levando-se em conta a Acurácia, Precisão, Revocação e a Análise da Curva ROC de cada algoritmo, conclui-se que os programas mais indicados para este tipo de base textual, que no caso deste experimento são as notícias, são a Árvore de Decisão, Floresta Aleatória, Máquina de Vetor de Suporte e o *AdaBoost* que, nesse caso, conseguiram 100% de acertos. Já o algoritmo *Naives Bayes* apresentou resultado insatisfatório na predição e é inviável para este tipo específico de base textual.

Por conseguinte, com base nos resultados deste experimento, é possível afirmar que esses modelos constituem uma ferramenta poderosa na classificação de textos, podendo auxiliar leitores e organizadores da informação.

Como trabalhos futuros, seria interessante analisar o desempenho dos classificadores com vários corpora para verificar com mais precisão se a quantidade de documentos e a natureza das notícias influenciam no desempenho dos algoritmos.

REFERÊNCIAS

BAEZA-YATES, R. RIBEIRO-NETO, B. **Recuperação de Informação**: Conceitos e tecnologia das máquinas de busca. Tradução técnica: Leandro Krug Wives, Viviane Pereira Moreira. 2. ed. Porto Alegre: Bookman, 2013. 614p.

BORGES, H. B. **Classificador Hierárquico Multirrótulo Usando uma Rede Neural Competitiva**. 2012. 188f. Tese (Doutorado) – Programa de Pós-graduação em Informática, Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, Paraná, 2012.

BREIMAN, L.; CUTLER, A. **An introduction to random forest for beginners**. 1. ed. Califórnia, Estados Unidos: Salford Systems, 2014. 71p.

DAS. **Formação Cientista de Dados**. Curso de Machine Learning ofertado por Data Science Academy. E-book. 2017. Disponível em https://www.datascienceacademy.com.br.

FACELI, Katti. *et al*. **Inteligência Artificial:** uma abordagem de aprendizagem de máquina. Rio de Janeiro: LTC, 2017.

FUJITA, M. S. L. et al. **A indexação de livros**: a percepção de catalogadores e usuários de bibliotecas universitárias. Um estudo de observação do contexto sociocognitivo com protocolos verbais. São Paulo: Cultura Acadêmica, 2009. 149p.

GONÇALVES, M. Classificação de Textos. In: BAEZA-YATES, R. RIBEIRO-NETO, B. **Recuperação de Informação: Conceitos e tecnologia das máquinas de busca.** Tradução técnica: Leandro Krug Wives, Viviane Pereira Moreira. 2. ed. Porto Alegre: Bookman, 2013. p. 277-338.

HAN, J.; KAMBER, M.; PEI, J. **Data mining**: concepts and techniques. 3. ed. Estados Unidos: Morgan Kaufmann and Elsevier, 2012. 673p.

HIRATA, N. S. T. **Classificador de Bayes**. 2007. Disponível em: http://www.vision.ime.usp.br/~nina/cursos/ibi5031-2007/pr.pdf. Acesso em 06 mar. 2019.

LAMA, P. **Clustering System Based on Text Mining Using the K-Means Algorithm**. 2013. Disponível em: https://www.theseus.fi/bitstream/handle/10024/69505/Lama_Prabin.pdf?sequence=1&isAllowed=y. Acesso em: 10 fev. 2018.

LIU, B. **Web Data Mining**: Exploring Hyperlinks, Contents, and Usage Data. Chicago: Springer, 2007. 532p.

MONARD, M.C.; BARANAUSKAS, J.A. Conceitos de aprendizado de máquina. In: REZENDE, S.O. **Sistemas inteligentes**: fundamentos e aplicações. São Carlos: Manole, 2003. 525p.

MONTEIRO, L. d. O., I. R. GOMES, et al. Etapas do Processo de Mineração de Textos – uma abordagem aplicada a textos em Português do Brasil. In: WORKSHOP DE COMPUTAÇÃO E APLICAÇÕES, 26., 2006, Campo Grande. **Anais do Congresso da SBC**. Campo Grande: Centro Universitário do Pará, 2006. p.78-81

NASCIMENTO, D. N. O. **Classificação Adaboost para detecção e contagem automática de plaquetas**. 2011. 55f. TCC (Graduação em Engenharia de Computação) - Escola Politécnica de Pernambuco - Universidade de Pernambuco, Recife, 2011.

NISBET, R.; ELDER, J.; MINER, G. **Handbook of Statistical Analysis and Data Mining Applications**. Orlando: Elsevier, 2009. 864p.

NOGUEIRA, B. M. **Avaliação de métodos não supervisionados de seleção de atributos para Mineração de Textos**. 2009. 104f. Dissertação (Mestrado) - Programa de Pós-graduação em Ciência da Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação – ICMC-USP – São Paulo, 2009.

PRATES, W. R. **O que é árvore de decisão** (decision tree)? Exemplos em R, 2018. Disponível em: https://www.wrprates.com/o-que-e-arvore-de-decisao-decision-tree-linguagem-r/.

RODRIGUES, H. J. F. **Ferramenta para Text Mining em Textos Completos**. 2016. 50f. Dissertação (Mestrado)- Programa de Pós-graduação Integrado em Engenharia e Computação, Faculdade de Engenharia, Universidade do Porto, Porto, 2016.

SARKAR, D. **Text Analytics with Python:** A Practical Real-World Approach to Gaining Actionable Insights from your Data. Bangalore: Apress. 2019.

SOARES, M. V. B.; PRATI, R.; MONARD, C. WCI 02 Improvements on the Porter's Stemming Algorithm for Portuguese. **Latin America Transactions, IEEE (Revista IEEE America Latina)**, v.7, n.4, p. 472 – 477, ago. 2009.

SOARES, R., A.; REBOUÇAS, S. M. D. P. Avaliação do Desempenho de Técnicas de Classificação Aplicadas à Previsão de Insolvência de Empresas de Capital Aberto Brasileiras. **Revista ADM.MADE**, Rio de Janeiro, ano 14, v.18, n.3, p.40-61, set./dez., 2015.

SOLKA, J. L. **Text Data Mining**: Theory and Methods. Naval Surface Warfare Center Dahlgren Division Statistics Surveys, Vol. 2, p.94-112, 2007.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. **Curvas ROC para avaliação de classificadores**. Disponível em: http://conteudo.icmc.usp.br/pessoas/gbatista/files/ieee_la2008.pdf. Acesso em 18 jul. 2019

TICOM, A. A. M. **Aplicação das técnicas de mineração de textos e sistemas especialistas na liquidação de processos trabalhistas**. 2007, 101f. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciências em Engenharia Civil, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.