EasyChair Preprint

№ 9605

Identification of Text Relevance in Service Desk Systems Using Machine Learning Classical Techniques

Marciel Mario Degasperi, Daniel Cruz Cavalieri and
Fidelis Zanetti de Castro

January 20, 2023

Identificação de Relevância em Textos de Sistemas de *Help Desk* usando Técnicas Clássicas de Aprendizado de Máquina

Marciel Mario Degasperi [*] **Daniel Cruz Cavalieri** [**]
Fidelis Zanetti de Castro [***]

[*] *Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, Campus Serra, ES (e-mail: marciel.deg@gmail.com)*
[**] *Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, Campus Serra, ES (e-mail: daniel.cavalieri@ifes.edu.br)*
[***] *Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, Campus Serra, ES (e-mail: fidelis.castro@ifes.edu.br)*

Abstract:
Service Desk systems have a rich information base made up of the history of calls made, which can and should be used as a reference base for subsequent calls. Standard search tools, such as keyword searches, prove to be unfeasible for searching large databases, due to the long query time and the return of results unrelated to the problem. This work aims to investigate the ability of some classical classification algorithms to find the characteristic defined here as "relevance": the characteristic of texts with some knowledge that can be reused. The motivation is that non-relevant texts can be removed early from the dataset, allowing complex algorithms to be employed on a smaller amount of information. In the tests performed, the *Naive-Bayes*, *Adaptive Boosting*, *Random Forest*, *Stochastic Gradient Descent*, *Logistic Regression*, *Support Vector Machine*, and *Light Gradient Boosting Machine* classifiers were used. The classifiers showed accuracy below 0.8, indicating that, in this scenario, other more efficient approaches should be used.

Resumo: Sistemas de *Help Desk* fornecem uma rica fonte de informações composta pelo histórico de atendimentos realizados, a qual pode e deve ser utilizada como fonte de consulta para atendimentos seguintes. Ferramentas comuns de busca, como buscas por palavras-chave, mostram-se inviáveis para busca em grandes bases de dados, devido a longo tempo de consulta e o retorno de resultados não relacionados ao problema. A proposta deste trabalho é investigar a capacidade de alguns algoritmos clássicos de classificação em encontrar a característica aqui definida como "relevância": a característica de textos com algum conhecimento que possa ser reutilizado. A motivação é que os textos não relevantes possam ser removidos antecipadamente da base de dados, permitindo que algoritmos complexos possam ser empregados em uma quantidade menor de informações. Nos testes realizados foram utilizados os classificadores *Naive-Bayes*, *Adaptive Boosting*, *Random Forest*, *Stochastic Gradient Descent*, *Logistic Regression*, *Support Vector Machine* e *Light Gradient Boosting Machine*. Os classificadores apresentaram acurácia abaixo de 0,8, indicando que, neste cenário, outras abordagens mais eficientes devem ser utilizadas.

Keywords: Machine Learning; Natural Language Processing; Service Desk Systems; Classification

Palavras-chaves: Aprendizado de Máquina; Processamento de Linguagem Natural; Sistemas de *Help Desk*; Classificação

1. INTRODUÇÃO

Os sistemas *Help Desk* atuam como um elo entre as empresas e os clientes, através de um sistema que possibilita a compilação e resolução dos problemas, bem como os registros da solução apresentada e correlação com outras situações (Boscolo, 2009). Com isso, cria-se uma ampla base de dados para a empresa, a qual permite gerenciar os problemas, resolvendo-o na sua raiz diminuindo custos operacionais. O *Help Desk* pode centralizar uma diversidade de informações e áreas de atendimento, tornando-se assim um ponto chave na administração e na solução de problemas (Cavalari and Costa, 2005).

Com o crescimento da base de atendimentos no decorrer do tempo, a busca por informações torna-se complexa. Uma forma simples de obter um conjunto de respostas para uma consulta de usuário é determinar quais documentos em uma coleção contém um certo conjunto de palavras-chave da consulta. Todavia, isto pode não ser suficiente para

atender ao usuário, pois a presença de documentos não relevantes entre os documentos retornados por uma consulta é praticamente certa. O principal objetivo desses sistemas deve ser recuperar o maior número possível de documentos relevantes e o menor número possível de documentos não relevantes (Baeza-Yates and Ribeiro-Neto, 2013). Neste cenário, verifica-se a necessidade da utilização de ferramentas de apoio na análise dos atendimentos de forma a auxiliar a classificação e recuperação de informações no conjunto de dados. Como estes dados são compostos basicamente por textos descritivos, dá-se a necessidade de propor e realizar sistemas computacionais baseados em processamento de linguagem natural.

O objetivo deste estudo é estudar a sensibilidade de técnicas de classificação conhecidas na identificação de relevância de textos. Para este estudo, um atendimento é considerado "relevante" se em seu texto houver conhecimento que possa ser reutilizado como auxílio na resposta de outros atendimentos.

Na seção 2 é apresentado uma breve revisão bibliográfica dos conceitos e ferramentas empregados neste artigo. Na seção 3 são apresentados os materiais e métodos utilizados. Já na seção 4 são exibidos os resultados encontrados. Por fim, na seção 5 são apresentadas as conclusões e próximos passos da pesquisa.

2. REVISÃO BIBLIOGRÁFICA

Nesta seção serão apresentados os conceitos e ferramentas relevantes ao trabalho.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma importante área de interesse em inteligência artificial e ciência da computação, cuja importância cresce paralelamente à quantidade de informações textuais digitalizadas sem precedente na história. Como um campo de estudo científico, o PLN engloba conceitos de ciência da computação, linguística e matemática com o objetivo principal de traduzir a linguagem humana em comandos que podem ser executados por computadores. Embora existam regras básicas de PLN em vigor, a linguagem natural é difícil de entender. Como exemplo, a linguagem humana nem sempre segue suas próprias regras. É possível que as palavras omitidas e gramática ruim não interfiram na comunicação entre duas pessoas. Para computadores, no entanto, isto é um grande desafio (Kang et al., 2020).

De acordo com Kang et al. (2020), o PLN pode ser dividido em quatro etapas: pré-processamento, vetorização, treinamento do modelo e avaliação do modelo. O pré-processamento de texto visa obter um texto "limpo", removendo símbolos e outros elementos que possam gerar ruído na análise subsequente. Após o pré-processamento do texto, é selecionado um algoritmo que converta palavras em uma representação numérica, como vetores. Com base nesses vetores de palavras, é possível aplicar algoritmos para treinar um modelo que pode resolver o problema real. Por fim, após o treinamento do modelo, é necessário sua avaliação para aferir sua qualidade e sua aplicabilidade em outros cenários.

2.2 Algoritmos de Vetorização

Os computadores, embora sejam capazes de executar uma ampla gama de tarefas envolvendo multimídia, ainda são máquinas executando instruções sobre números no sistema binário. Dessa forma, para possibilitar a classificação de textos com recursos computacionais, é necessário a seleção de alguma estratégia de tradução que possa representar os conjuntos de palavras do documento em forma numérica, sem que haja perda de características importantes. Estas estratégias são encapsuladas em algoritmos de vetorização (Singh and Shashi, 2019).

Entre os algoritmos de vetorização mais comuns, pode-se citar o TF-IDF. Mais detalhes destes algoritmos são apresentados nos tópicos a seguir.

TF-IDF Um dos procedimentos geralmente adotado para a vetorização de textos é a representação usando a abordagem *bag-of-words*. Nesta abordagem, cada documento é representado como um vetor de palavras que ocorrem no documento. Quando o termo está presente no documento o valor é 1, caso contrário 0 (Rajaraman and Ullman, 2011).

Uma forma comum de categorização de textos é identificando a ocorrência de palavras-chave que caracterizam os documentos sobre um tema em específico. Por exemplo, artigos sobre beisebol tenderiam a ter muitas ocorrências de palavras como "bola", "bastão", "arremesso", "correr" e assim por diante. O algoritmo *Term Frequency times Inverse Document Frequency*, ou simplesmente TF-IDF, é uma medida estatística que indica a importância de uma palavra em relação ao documento (Rajaraman and Ullman, 2011). Assim, é possível substituir cada palavra de um texto por um número que indica sua importância no contexto do documento.

2.3 Algoritmos de Classificação

Os algoritmos de classificação são métodos usados para identificar novas classes de observações com base em dados de treinamento. O algoritmo aprende a partir de conjuntos de dados ou observações e então classifica novas observações em várias classes ou grupos (Kowsari et al., 2019). A seguir são expostos os algoritmos de classificação utilizados neste trabalho.

Naive-Bayes Em probabilidade e estatística, o teorema de Bayes descreve a probabilidade de um evento, baseado em um conhecimento a priori que pode estar relacionado ao evento. O classificador Naive-Bayes é uma aplicação direta do Teorema de Bayes. O termo *naive*, ("ingênuo") se refere à premissa central do algoritmo de que os atributos considerados são não correlacionados entre si (Swinburne, 2004).

Na classificação de textos, o classificador Naive-Bayes é um método popular devido à sua eficiência computacional e bom desempenho na predição (Chen et al., 2009). Neste classificador, cada documento é visto como uma coleção de palavras, onde a posição de ocorrência de cada palavra não é considerada. Por ser muito simples e rápido, possui um desempenho relativamente maior do que outros classificadores. Além disso, o Naive-Bayes só precisa de um pequeno

número de dados de teste para concluir classificações com uma boa precisão (Frank and Bouckaert, 2006).

Adaptive Boosting "*Boosting*" é uma técnica proposta na década de 90, onde diversos classificadores pouco acurados são combinados, de forma a produzir um novo algoritmo poderoso. Dentre as várias implementações de *boosting* existentes, o *Adaptive Boosting*, ou AdaBoost é comumente mais usada (Friedman et al., 2000).

O conceito básico por trás do AdaBoost é definir os pesos dos classificadores e treinar a amostra de dados em cada iteração de forma que garanta as previsões precisas de observações incomuns. Em cada iteração, ele tenta fornecer um ajuste ótimo para esses exemplos, minimizando o erro de treinamento para que a próxima iteração faça uma classificação mais precisa (Kowsari et al., 2019).

Random Forest Uma árvore de decisão é um preditor que estima o rótulo associado a uma instância viajando de um nó raiz de uma árvore para uma folha. Em cada nó no caminho da raiz para a folha, o filho sucessor é escolhido com base na divisão do espaço de entrada. Normalmente, a divisão é baseada em uma das características ou em um conjunto predefinido de regras (Shalev-Shwartz and Ben-David, 2014). As árvores de decisão podem ser usadas para descobrir recursos e extrair padrões em grandes bancos de dados que são importantes para discriminação e modelagem preditiva. Essas características, juntamente com sua interpretação intuitiva, são algumas das razões pelas quais as árvores de decisão serem usadas extensivamente para análise exploratória de dados (Myles et al., 2004).

A fim de mitigar as limitações conhecidas nas *Decision Trees*, Ho (1995) propôs a *Random Forest*, ou Floresta Aleatória. Trata-se de uma combinação de árvores de decisão de tal forma que cada árvore é construída em um subespaço aleatório do espaço de características. Árvores em diferentes subespaços generalizam sua classificação de maneira complementar, e sua classificação combinada é melhor do que a classificação individual.

Stochastic Gradient Descent Na matemática, "Gradiente Descendente" é um algoritmo de otimização iterativa de primeira ordem para encontrar um mínimo local de uma função diferenciável. A ideia é dar passos repetidos na direção oposta do gradiente da função no ponto atual, pois esta é a direção de descida mais acentuada (Mikolov et al., 2013).

O *Stochastic Gradient Descent*, geralmente abreviado por SGD, pode ser considerado como uma aproximação estocástica da otimização de gradiente descendente, uma vez que substitui o gradiente real (calculado a partir de todo o conjunto de dados) por uma estimativa do mesmo (calculada a partir de um subconjunto de dados selecionado aleatoriamente), reduzindo assim a carga computacional (Sra et al., 2012).

Logistic Regression A regressão logística é um algoritmo de classificação supervisionado. Em um problema de classificação, a variável de saída pode assumir apenas valores discretos para um determinado conjunto de entradas. O modelo de regressão logística constrói um modelo de regressão para prever a probabilidade de uma determinada entrada de dados pertencer à categoria numerada como 1. Assim como a regressão linear assume que os dados seguem uma função linear, a regressão logística modela os dados usando a função sigmóide (Levy and O'Malley, 2020).

A regressão logística é uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de dados, um modelo que permita a predição de valores tomados por uma variável categórica a partir de uma série de variáveis explicativas (Rymarczyk et al., 2019)

Support Vector Machine O classificador *Support Vector Machine*, ou SVM, toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte. Dados um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, um algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra (Cortes and Vapnik, 1995).

Um modelo SVM é uma representação de exemplos como pontos no espaço, mapeados de maneira que os exemplos de cada categoria sejam divididos por um espaço claro que seja tão amplo quanto possível. Os novos exemplos são então mapeados no mesmo espaço e preditos como pertencentes a uma categoria baseados em qual o lado do espaço eles são colocados (Shalev-Shwartz and Ben-David, 2014).

Light Gradient Boosting Machine *Gradient Boosting Decision Tree* é um algoritmo de aprendizado de máquina popular e possui algumas implementações eficazes, contudo sua eficiência e escalabilidade ainda são insatisfatórias para bases de dados extensas.

O objetivo da *LightGBM* é reduzir o número de instâncias de dados e o número de recursos, acelerando assim o processo de treinamento. Para isso, duas técnicas foram criadas: *Gradient-based One-Side Sampling* (GOSS), que propõe a redução do volume de dados realizando amostragem aleatória nas instâncias com gradientes pequenos, enquanto mantém todas as instâncias com gradientes grandes, e o *Exclusive Feature Bundling* (EFB), que agrupa recursos exclusivos. Com estas abordagens, o *LightGBM* consegue resultados até 20 vezes mais rápido do que seu antecessor, com praticamente a mesma precisão. (Ke et al., 2017).

2.4 Métricas de Avaliação

Classificadores necessitam ser treinados para a tarefa na qual serão utilizados. O treinamento consiste em achar os parâmetros que melhor adequam-se aos dados, minimizando o erro entre a predição e o valor original. Esse erro é avaliado usando uma ou várias métricas (Yacouby and Axman, 2020).

Dentre as métricas utilizadas na avaliação de classificadores, podemos citar a Acurária, Precisão, Sensibilidade, *F1-score*, *ROC* e *AUC*. Acurácia é definida como a proporção de instâncias verdadeiras recuperadas, tanto positivas quanto negativas, entre todas as instâncias recuperadas. Precisão mede a proporção de instâncias corretas recuperadas em relação às instâncias recuperadas. Sensibilidade, muitas vezes encontrado na literatura por

seu correspondente em inglês *Recall*, mede a proporção de instâncias corretas recuperadas em relação a todas as instâncias corretas. O *F1-score* é a média harmônica entre precisão e sensibilidade (Dalianis, 2018). Os gráficos ROC (do inglês *Receiver Operating Characteristics Curve*) são gráficos bidimensionais que exibem a relação entre as instâncias verdadeiras positivas e falsas positivas. Por fim, a AUC (do inglês *Area Under Curve*) trata-se da área sob a curva ROC, ou seja, uma representação numérica do mesmo indicador (Zhang et al., 2015).

2.5 Validação Cruzada

Ao implantar um modelo preditivo, é importante entender sua precisão de previsão em testes futuros. Portanto, é essencial que os resultados reflitam boas estimativas e intervalos de confiança precisos. A validação cruzada é uma abordagem amplamente utilizada para essas duas tarefas (Bates et al., 2021).

A validação cruzada opera dividindo a base de dados em dois segmentos: um usado para treinar e o outro usado para validar o modelo. Os conjuntos de treinamento e validação devem ser cruzados em rodadas sucessivas, de modo que cada ponto de dados tenha a chance de ser validado. A forma básica de validação cruzada é a validação cruzada *k-fold*. Os dados são primeiro particionados em k segmentos ou dobras de tamanho similar. Em seguida, k iterações de treinamento e validação são realizadas de tal forma que dentro de cada iteração uma dobra diferente dos dados é mantida para validação, enquanto as $k-1$ dobras restantes são usadas para aprendizado. Após a conclusão, k amostras da métrica de desempenho estarão disponíveis para cada algoritmo. Diferentes metodologias, como a média, podem ser usadas para obter uma medida agregada dessas amostras, ou essas amostras podem ser usadas em um teste de hipótese estatística para mostrar que um algoritmo é superior a outro. (Refaeilzadeh et al., 2016).

3. METODOLOGIA

A seguir são apresentados os materiais e os métodos utilizados no desenvolvimento deste trabalho.

3.1 Ferramentas de Desenvolvimento

Para o desenvolvimento dos testes, foi utilizada a linguagem de programação Python, no ambiente de desenvolvimento Google Colaboratory. Foram utilizados implementações do pacote *sklearn*, que fornece um conjunto de ferramentas para aprendizado de máquina e modelagem estatística, como classificação, regressão, clusterização e redução de dimensionalidade.

3.2 Base de Dados

A base de dados a ser utilizada, inteiramente em português, foi coletada de um banco de dados relacional do sistema *Help Desk Conexos*, software interno da empresa Conexos Consultoria e Sistemas LTDA. Este sistema está em funcionamento desde o ano de 2004, contendo uma considerável coleção de atendimentos prestados pela empresa a seus clientes. O fluxo de interações dos atendimentos deste sistema é similar ao fluxo de fóruns online: a partir de uma solicitação do cliente, que pode ser um questionamento, relato de um problema ou uma solicitação de incremento no sistema, podem haver diversas respostas por parte da prestadora de serviços, bem como diversas réplicas e tréplicas até que o cliente esteja com a solicitação resolvida, momento no qual o atendimento é finalizado. Este formato permite a geração de atendimentos com fluxos complexos, onde frequentemente ocorrem oscilações entre assuntos distintos, dificultando sua classificação. Dessa forma, para os testes iniciais foram selecionados somente os atendimentos onde o fluxo de atendimento é composto por uma pergunta do cliente, uma resposta do prestador de serviços e a finalização do atendimento. Atendendo a estas condições, foram coletados dados dos atendimentos realizados entre os anos de 2019 e 2020, totalizando 11449 atendimentos. Seguindo as diretrizes da Lei Geral de Proteção dos Dados - LGPD (Brasil, 2018), a base foi anonimizada, de modo a impossibilitar a identificação de dados sensíveis.

A Tabela 1 traz alguns dados estatísticos das palavras que compõem a base de dados. Entre as características notáveis verifica-se forte ocorrência de repetição de palavras nos textos.

Tabela 1. Informações descritivas da base de dados.

Indicador	Quantidade
Total de Atendimentos	11449
Total de Palavras	953728
Média de Palavras por Atendimento	83,30
Palavras Distintas	29408
Média de Palavras Distintas por Atendimento	28,90

Uma ferramenta intuitiva que permite uma rápida análise de ocorrências de palavras em um documento é a chamada Nuvem de Palavras. Uma Nuvem de Palavras é uma representação visual da frequência das palavras, de forma que quanto mais comumente o termo aparecer dentro do texto que está sendo analisado, maior a palavra aparecerá na imagem gerada (Atenstaedt, 2012). Na Figura 1 é exibido a Nuvem de Palavras para a base de dados.

Figura 1. Nuvem de Palavras que compõem a base de dados.

A Nuvem de Palavras evidencia uma importante característica da base de dados em questão: a presença expressiva de saudações e termos comuns de apresentação e encerramento, tais como “bom dia”, “prezados” e “gentileza validar”. Para Hajrizi and Nuçi (2020) estes termos, apesar de representarem um gesto respeitoso na comunicação humana, se tornam ruído para os algoritmos de processamento de texto, afetando seu desempenho. Portanto, parece razoável afirmar que tais termos devam ser removidos

da base de dados antes da classificação, visando concentrar as informações e evitar possíveis vieses no treinamento dos classificadores. Essa remoção será detalhada no tópico seguinte.

3.3 Pré Processamento dos Dados

O pré-processamento de texto é uma etapa essencial em qualquer Sistema de Processamento de Linguagem Natural. Esta etapa envolve sua transformação ante sua análise, e engloba estratégias como remover conteúdo irrelevante para a tarefa, aglomerar termos semanticamente relacionados e aumentar a quantidade de informações semânticas capturadas (Hickman et al., 2022).

Neste trabalho, o pré processamento foi realizado pelas etapas abaixo:

- Remoção de caracteres especiais, acentuação e sinais de pontuação;
- Remoção de palavras vazias (ou *stopwords*). Palavras vazias são termos que agregam pouca ou nenhuma informação semântica ao texto, como preposições e artigos;
- Remoção de saudações e termos comuns de conclusão, tais como "bom dia" e "estamos à disposição";
- Substituição de sinônimos por um único termo. Como exemplo, os termos "Documento Fiscal", "Nota de Mercadoria" e "Nota Fiscal Eletrônica" foram substituídos por "Nota Fiscal";
- Substituição de alguns termos por suas siglas, como "Espírito Santo" por "ES";

Na Figura 2 é representado a Nuvem de Palavras após o pré processamento. Podemos notar uma melhor distribuição das ocorrências em termos com sentido contextual, como as palavras "processo", "sistema", "relatório" entre outras.

Figura 2. Nuvem de Palavras que compõem a base de dados após remoção de saudações.

Para treinamento dos algoritmos de classificação, é necessário que os dados estejam previamente classificados, para que os algoritmos possam se embasar durante treinamento e validação dos resultados obtidos. Para tal, um subconjunto de 1084 atendimentos foi classificado, de forma a compor a base de dados de treinamento. A distribuição não balanceada entre as classes na base de dados faz com que os modelos do classificador tendam para classes com muitas amostras de treinamento (Okkalioglu and Okkalioglu, 2022). Desta forma, os atendimentos foram selecionados de forma a compor uma base de dados perfeitamente balanceada.

Na Tabela 2 temos algumas informações estatísticas do subconjunto para treinamento. Uma característica notável é que, em média, atendimentos relevantes possuem uma quantidade de palavras maior do que os atendimentos não relevantes.

Tabela 2. Informações descritivas da base de dados de treinamento.

Indicador	Total	Positivos	Negativos
Número de Atendimentos	1084	542	542
Total de Palavras	88244	52347	35897
Palavras/Atendimento	81,41	96,58	66,23
Palavras Distintas	8355	6167	5194
Palavras Distintas/Atendimento	28,92	29,88	27,97

Na Tabela 3 é exibida a lista das palavras mais usadas na base de dados em sua íntegra, enquanto nas Tabelas 4 e 5 são exibidas as listas das palavras mais usadas no subconjunto de atendimentos relevantes e não relevantes, respectivamente, da mesma base de dados. Pode-se verificar que há uniformidade na distribuição de frequência das palavras. Nota-se também que as bases de dados relevante e não relevante apresentam recorrência de palavras em comum.

Tabela 3. Palavras mais recorrentes na base de dados integral após remoção de saudações.

Palavra	Ocorrências	% do Total
sistema	1099	1,32
tela	983	1,18
erro	820	0,98
documento	764	0,92
nfe	715	0,86
processo	556	0,67
caso	544	0,65
codigo	526	0,63
anexo	494	0,59
campo	460	0,55

Tabela 4. Palavras mais recorrentes na base de dados relevante após remoção de saudações.

Palavra	Ocorrências	% do Total
sistema	677	1,35
tela	670	1,33
documento	605	1,20
nfe	551	1,09
erro	457	0,91
processo	391	0,78
codigo	349	0,69
caso	342	0,68
campo	327	0,65
cadastro	323	0,64

3.4 Métodos Propostos

O conceito de relevância é um tema amplamente debatido na Ciência da Informação. De acordo com Baeza-Yates and Ribeiro-Neto (2013), "relevância" é definida como uma métrica de qualidade, ligada à satisfação de um documento atender a um determinado tema. A noção de relevância é humana e não técnica, composto por uma gama de variáveis e, portanto, frágil e de difícil definição.

O conceito de relevância foco deste estudo é abrangente e não restrito a um assunto pré estabelecido. Aqui, considera-se "relevante" os textos de atendimentos cujo

Tabela 5. Palavras mais recorrentes na base de dados não relevante após remoção de saudações.

Palavra	Ocorrências	% do Total
sistema	422	1,28
erro	363	1,10
tela	313	0,95
anexo	228	0,69
base	212	0,64
disposicao	212	0,64
relatorio	208	0,63
caso	202	0,61
codigo	190	0,57
verificar	185	0,56

conteúdo possa ser utilizado como base de conhecimento para outros atendimentos. Aparentemente, este conceito não parece diretamente ligado a palavras-chave, nem à forma como o texto foi estruturalmente redigido. O desafio deste estudo é analisar a sensibilidade de técnicas de classificação, já consolidadas na literatura, para a identificação da relevância nos textos.

Ensaios foram realizados utilizando-se algoritmos clássicos de classificação. Nesta etapa foram utilizados os classificadores *Naive-Bayes*, *Adaptive Boosting*, *Random Forest*, *Stochastic Gradient Descent*, *Logistic Regression*, *Support Vector Machine* e *Light Gradient Boosting Machine*. O objetivo desta etapa é encontrar um classificador viável que possa ser utilizado na separação dos atendimentos "relevantes" dos "não relevantes". Os dados foram divididos em treinamento e teste na proporção 80-20.

4. RESULTADOS

Para os testes foi utilizado um subconjunto da base de dados original, composto de 1084 atendimentos manualmente rotulados. Este subconjunto foi criteriosamente balanceado, com 542 atendimentos classificados como relevantes e os demais como não relevantes. Esta decisão foi tomada pois há problemas conhecidos de alguns algoritmos de classificação com conjuntos de dados desbalanceados, como pode-se ver nos trabalhos de Chawla and Sylvester (2007), Pozzolo et al. (2015), Sundarkumar and Ravi (2015), entre outros. Nos testes foram utilizados o vetorizador TF-IDF e uma série de classificadores clássicos, que são melhor descritos a seguir.

A seguir serão apresentados os resultados utilizando-se os classificadores *Adaptive Boosting*, *Light Gradient Boosting Machine*, *Logistic Regression*, Naive-Bayes, *Random Forest*, *Stochastic Gradient Descent* e *Support Vector Machine*. Na Tabela 6 são exibidos os valores das métricas Acurária, *F1-Score* e AUC dos classificadores, obtidos através de experimentos com validação cruzada com $k = 10$, realizados sobre o conjunto de dados. Verifica-se que os classificadores do experimento apresentaram métricas similares, com destaque para o classificador *Random Forest* que obteve os melhores resultados, atingindo valores de acurária e *F1-Score* de 0,793 e 0,804, respectivamente.

Tabela 6. Métricas de classificadores aplicados ao conjunto de dados.

Classificador	Acurácia	*F1-Score*	AUC
AdaBoost	0,747±0,044	0,742±0,048	0,824±0,033
LightGBM	0,791±0,025	0,794±0,024	0,867±0,036
Logistic Regression	0,789±0,021	0,790±0,024	**0,872±0,030**
Naive-Bayes	0,767±0,034	0,776±0,037	0,861±0,028
Random Forest	**0,793±0,036**	**0,804±0,031**	0,871±0,035
SGD	0,781±0,026	0,787±0,024	0,858±0,026
SVM	0,788±0,018	0,792±0,015	0,864±0,030

Na Figura 3 é exibido as curvas ROC dos algoritmos utilizados. Da mesma forma que as demais métricas, os dados foram coletados em uma validação cruzada, com $k = 10$.

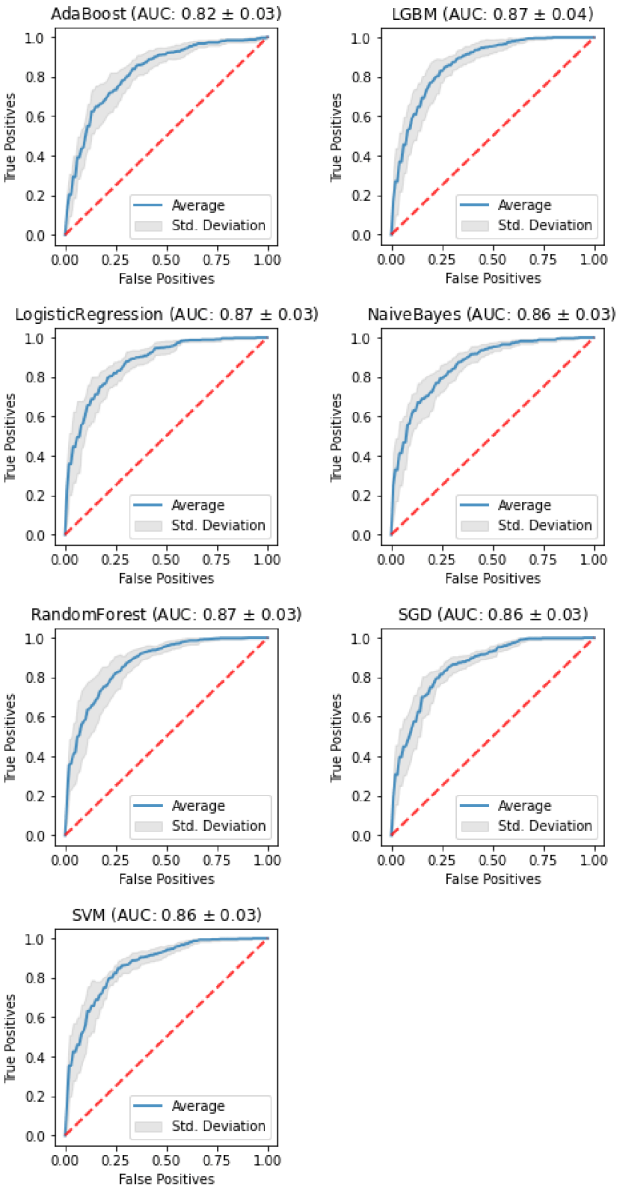

Figura 3. Curva ROC com desvio padrão dos classificadores empregados.

5. CONCLUSÃO

A definição de "relevância" do conteúdo, apesar da aparente simplicidade por trás da classificação binária, é um conceito abstrato. A análise inicial da base de dados mostrou que as características que identificam a relevância são sutis, e envolvem mais do que somente a presença ou não de palavras-chave no texto.

Vários classificadores foram aplicados na tentativa de identificar o conceito de "relevância" dos textos da base de dados. Os algoritmos de classificação clássicos aplicados apresentaram métricas próximas a 80% para a identificação do contexto de relevância. Os classificadores *Random Forest* e *LightGBM* se destacaram na classificação. O classificador *Random Forest* apresentou valores de acurária e *F1-score* de 0,793 e 0,804, respectivamente, enquanto o classificador *LightGBM* apresentou valores para as mesmas estatísticas de 0,791 e 0,794, respectivamente.

Para a tarefa de identificação de relevância dos textos, outras estratégias computacionais devem ser estudadas. Uma sugestão para trabalhos futuros é a utilização de redes neurais recorrentes e aprendizado profundo, como o Word2vec e redes LSTM.

AGRADECIMENTOS

À Cooperação CAPES/FAPES - PDPG, projeto *TIC+TAC*, pelo apoio financeiro (TO 133/2021, Processo Nº 2021-CFT5C).

Aos diretores da empresa Conexos Consultoria e Sistemas LTDA, por fornecer acesso à base de dados, objeto de estudo deste trabalho.

Aos nossos professores e colegas do Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo (Ifes) que forneceram conhecimentos que auxiliaram a pesquisa, embora possam não concordar com todas as interpretações/conclusões deste artigo.

REFERÊNCIAS

Atenstaedt, R. (2012). Word cloud analysis of the bjgp. *British Journal of General Practice*, 62(596), 148–148. doi:10.3399/bjgp12X630142.

Baeza-Yates, R. and Ribeiro-Neto, B. (2013). *Recuperação de Informação-: Conceitos e Tecnologia das Máquinas de Busca*. Bookman Editora.

Bates, S., Hastie, T., and Tibshirani, R. (2021). Cross-validation: what does it estimate and how well does it do it? *arXiv preprint arXiv:2104.00673*.

Boscolo, V.G. (2009). Sistema de gerenciamento de help-desk.

Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). *Diário Oficial da República Federativa do Brasil*.

Cavalari, G.O.T. and Costa, H.A.X. (2005). Modelagem e desenvolvimento de um sistema help-desk para a prefeitura municipal de Lavras. *Revista Eletrônica de Sistemas de Informação*, 4(2).

Chawla, N.V. and Sylvester, J. (2007). Exploiting diversity in ensembles: Improving the performance on unbalanced datasets. In M. Haindl, J. Kittler, and F. Roli (eds.), *Multiple Classifier Systems*, 397–406. Springer Berlin Heidelberg, Berlin, Heidelberg.

Chen, J., Huang, H., Tian, S., and Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, 36(3, Part 1), 5432–5435. doi:https://doi.org/10.1016/j.eswa.2008.06.054.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. doi:https://doi.org/10.1007/BF00994018.

Dalianis, H. (2018). Evaluation metrics and evaluation. In *Clinical text mining*, 45–53. Springer.

Frank, E. and Bouckaert, R.R. (2006). Naive Bayes for text classification with unbalanced classes. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (eds.), *Knowledge Discovery in Databases: PKDD 2006*, 503–510. Springer Berlin Heidelberg, Berlin, Heidelberg.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2), 337 – 407. doi:10.1214/aos/1016218223.

Hajrizi, R. and Nuçi, K.P. (2020). Aspect-based sentiment analysis in education domain. *CoRR*, abs/2010.01429.

Hickman, L., Thapa, S., Tay, L., Cao, M., and Srinivasan, P. (2022). Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, 25(1), 114–146. doi:10.1177/1094428120971683.

Ho, T.K. (1995). Random Decision Forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, 278–282 vol.1. doi:10.1109/ICDAR.1995.598994.

Kang, Y., Cai, Z., Tan, C.W., Huang, Q., and Liu, H. (2020). Natural Language Processing (NLP) in Management Research: A literature review. *Journal of Management Analytics*, 7(2), 139–172.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4), 150.

Levy, J.J. and O'Malley, A.J. (2020). Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. *BMC Medical Research Methodology*, 20(1), 171. doi:https://doi.org/10.1186/s12874-020-01046-3.

Mikolov, T., Le, Q.V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A., and Brown, S.D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics*, 18(6), 275–285. doi:https://doi.org/10.1002/cem.873.

Okkalioglu, M. and Okkalioglu, B.D. (2022). Afe-mert: imbalanced text classification with abstract feature extraction. *Applied Intelligence*, 1–17.

Pozzolo, A.D., Caelen, O., Johnson, R.A., and Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, 159–166. doi:10.1109/SSCI.2015.33.

Rajaraman, A. and Ullman, J.D. (2011). *Data Mining*, 1–17. Cambridge University Press. doi:10.1017/CBO9781139058452.002.

Refaeilzadeh, P., Tang, L., and Liu, H. (2016). *Cross-Validation*, 1–7. Springer New York, New York, NY. doi:10.1007/978-1-4899-7993-3_565-2.

Rymarczyk, T., Kozłowski, E., Kłosowski, G., and Niderla, K. (2019). Logistic regression for machine learning in process tomography. *Sensors*, 19(15), 3400. doi:10.3390/s19153400.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning - from Theory to Algoritms*. Cambridge University Press.

Singh, A.K. and Shashi, M. (2019). Vectorization of text documents for identifying unifiable news articles. *Int. J. Adv. Comput. Sci. Appl*, 10(7).

Sra, S., Nowozin, S., and Wright, S. (2012). *Optimization for Machine Learning*. Neural information processing series. MIT Press.

Sundarkumar, G.G. and Ravi, V. (2015). A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, 37, 368–377. doi:https://doi.org/10.1016/j.engappai.2014.09.019.

Swinburne, R. (2004). Bayes' theorem. *Revue Philosophique de la France Et de l*, 194(2).

Yacouby, R. and Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, 79–91.

Zhang, X., Li, X., Feng, Y., and Liu, Z. (2015). The use of roc and auc in the validation of objective image fusion evaluation metrics. *Signal processing*, 115, 38–48.