INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Text mining aplicado à gestão de fundos públicos

Luís Henrique Broncas Chinita

Mestrado em Ciência de Dados

Orientador:
Doutor Ricardo Daniel Santos Faro Marques Ribeiro, Professor Associado, Iscte - Instituto Universitário de Lisboa

Co-Orientador:
Doutor Luís Miguel Martins Nunes, Professor Associado, Iscte - Instituto Universitário de Lisboa

novembro, 2022

BUSINESS SCHOOL

TECNOLOGIAS E ARQUITETURA

Text mining aplicado à gestão de fundos públicos

Luís Henrique Broncas Chinita

Mestrado em Ciência de Dados

Orientador:
Doutor Ricardo Daniel Santos Faro Marques Ribeiro, Professor Associado, Iscte - Instituto Universitário de Lisboa

Co-Orientador:
Doutor Luís Miguel Martins Nunes, Professor Associado, Iscte - Instituto Universitário de Lisboa

novembro, 2022

Esta página foi intensionalmente deixada em branco

Agradecimentos

Agradeço aos professores Ricardo Ribeiro e Luís Nunes, profissionais e pessoas de valor, pelas quais tive o gosto de ser orientado. A sua partilha de conhecimento, disponibilidade e apoio prestado tiveram um papel crucial nesta dissertação. Quero agradecer especificamente a elevada regularidade de reuniões de orientação que facilitaram a contínua progressão do trabalho.

Agradeço a todos os professores, colegas e outras pessoas envolvidas no processo que trabalharam no projeto de investigação "IA-SI - Inteligência Artificial na Gestão de Incentivos", ao qual esta dissertação se encontra associada.

Agradeço ao IAPMEI e à AICEP, entidades que facultaram os dados utilizados nesta dissertação.

Agradeço aos meus pais por me apoiarem em todas as etapas da minha vida. Por me empurrarem para o que julgam ser o melhor caminho, por me forçarem a sair de zonas de conforto, por me encorajarem e finalmente por confiarem em mim e me permitirem confiar sempre neles.

Agradeço à Catarina, a minha namorada. Está sempre lá. Faz-me querer ser melhor. Conhece-me quase tão bem como eu próprio. Ajuda-me a não perder o foco nos desafios a que me proponho. É a pessoa com que de uma forma ou de outra, tudo faz sentido, nem que seja só para nós.

Agradeço aos meus restantes familiares e amigos. Contribuem para o meu crescimento, para a minha felicidade, para o meu presente e mais importante para o nosso futuro.

Todos fizeram parte desta etapa.

Parte deste trabalho foi desenvolvido no âmbito do projeto de investigação POCI-05-5762-FSE-000231, designado por "IA-SI - Inteligência Artificial na Gestão de Incentivos".

A realização deste trabalho foi parcialmente financiada por fundos nacionais através da FCT - Fundação para a Ciência e Tecnologia, I.P. no âmbito do projeto UIDB/04466/2020 (ISTAR).

Esta página foi intensionalmente deixada em branco

Resumo

Este trabalho tem como objetivo analisar documentos textuais submetidos por empresas portuguesas no momento de candidatura a programas de incentivos empresariais públicos. Com esta análise pretende-se extrair e selecionar variáveis relevantes, presentes nos textos, que possuam poder preditivo em relação a futuras ações das empresas candidatas aceites, no decorrer dos projetos. O objetivo concreto é a predição da anulação de projetos com fundos atribuídos, durante a sua duração prevista. Para realizar esta análise foi necessário criar uma cadeia de classificação de texto na qual são aplicadas variadas técnicas de processamento da língua natural, extração e seleção de variáveis, seleção e utilização de classificadores, e métricas de avaliação dos resultados. Foram utilizadas técnicas de referência de extração de variáveis como a extração de valores TF e TF-IDF e foram igualmente levadas a cabo experiências de extração de variáveis baseadas em geração de tópicos, análise de similaridade textual, análise de diversidade lexical, exploração de vocabulário específico, entre outros tipos de análise do conteúdo textual. A exploração de variáveis criadas a partir destas experiências mostra-nos características escondidas nos dados, como por exemplo, o facto de se verificar uma maior incidência de projetos com elevados níveis de similaridade em certos distritos do país. O principal objetivo foi alcançar o melhor desempenho possível nas métricas obtidas através da matriz de confusão (taxa de acerto; precisão; cobertura; F1-Score) na predição da anulação de projetos. Os melhores resultados da predição de anulação foram obtidos por um conjunto de variáveis provenientes de diversos métodos de extração e utilizando o algoritmo Classificador Naïve Bayes: 79% de taxa de acerto; 77% de precisão; 71% de cobertura; 74% de F1-Score. Neste trabalho é assim demonstrado o proveito da mistura de variáveis provenientes de diferentes métodos de extração de variáveis.

Palavras-chave: Fundos públicos; Text mining; Extração de variáveis; Análise de dados; Classificação de texto

Códigos de Classificação JEL:
O31 – Inovação e invenções: processos e incentivos
O38 – Políticas governamentais
C53 – Métodos de predição e previsão; Métodos de simulação

Esta página foi intensionalmente deixada em branco

Abstract

This work aims to analyze the textual documents presented by Portuguese companies when applying for business incentive programs. This work intends to extract and select relevant features, present in the texts, which have predictive power in relation to future actions of the companies whose projects were accepted, during the projects. The concrete goal is the prediction of the cancellation of the projects with allocated funds, during their expected duration. It was necessary to create a text classification pipeline which applies natural language processing, various features extraction and selection techniques, classification algorithms and evaluation metrics. Many feature extraction techniques were used, such as classical techniques as TF and TF-IDF values generation, as also other experiments as topic generation, similarity analysis, lexical analysis, identification of specific vocabulary used, among other analysis of textual content that were also carried out. The feature analysis can show us hidden characteristics in the data, such as the fact that there is a preponderance of projects with high levels of similarity in certain districts of the country. The main objective, regarding the perdition of cancellation of the projects, was achieving the best possible performance, for that there were used the confusion matrix metrics (accuracy; precision; revocation; F1-Score). The best prediction results were obtained by a set of features from different extraction methods together with the use of the Naïve Bayes Classifier algorithm: 79% accuracy; 77% precision; 71% recall; 74% F1-Score. Therefore, it is shown the advantages of mixing features from different extraction methods on this text classification application.

Keywords: Public funds; Text mining; Variable extraction; Data Analysis; Text classification

JEL Classification Codes:

O31 – Innovation and Inventions: Processes and Incentives

O38 – Government Policies

C53 – Forecasting and Prediction Methods; Simulation Methods

Esta página foi intensionalmente deixada em branco

Índice

Índice de Figuras

Índice de Tabelas

Esta página foi intensionalmente deixada em branco

Introdução

Esta dissertação foi desenvolvida em associação com o projeto de investigação "IA-SI - Inteligência Artificial na Gestão de Incentivos", desenvolvido em colaboração pelo ISCTE com a AICEP (Agência para o Investimento e Comércio Externo de Portugal) e o IAPMEI (Agência para a Competitividade e Inovação), instituições com responsabilidades na gestão e distribuição de fundos públicos às empresas nacionais. O projeto propôs-se a aplicar inteligência artificial às atividades desenvolvidas por estas organizações. Este teve como propósito a redução da quantidade de burocracia envolvida nestas atividades e do tempo de recebimento, pelas empresas, dos incentivos concedidos ao abrigo do Portugal 2020. O objetivo concreto definido foi o desenvolvimento de um protótipo de *software* capaz de produzir *scorings* de risco que pudessem auxiliar as atividades de gestão dos incentivos. Tal como enuncia o IAPMEI em relação ao projeto, "visa a melhoria da eficiência e fiabilidade dos objetivos e da gestão de riscos na fase de candidatura e análise de pedidos de pagamento das empresas beneficiárias dos sistemas de incentivos do Portugal 2020, com recurso a técnicas de aprendizagem automática." (*IAPMEI e AICEP aplicam AI na gestão de incentivos Portugal 2020*, 2019).

O projeto utilizou como dados as informações contidas nas candidaturas de empresas aos programas de incentivos e outras informações relacionadas com o decorrer dos projetos provenientes das candidaturas aceites. Nem todos estes dados se encontram estruturados, existindo uma parte considerável de informação contida em textos. Esta situação cria o problema de como analisar e extrair informação destes dados de forma eficiente. Atualmente, no contexto de progressiva utilização dos dados como ferramentas de auxílio à tomada de decisão de forma automática, várias organizações deparam-se com o problema concreto de terem grandes quantidades de conhecimento no formato de informação textual pouco estruturada (Dengre et al., 2020). Tal como é referido na pesquisa de Dengre et al. (2020), o *text mining* apresenta-se então como solução. Esta disciplina torna-se relevante nestas circunstâncias por constituir um conjunto de técnicas e ferramentas que permitem a transformação eficiente destes textos em informação relevante, concisa e estruturada, pronta para ser utilizada de forma automática.

1.1. Objetivo

O objetivo desta dissertação define-se assim pela aplicação dos conteúdos da disciplina de *text mining* aos textos das candidaturas das empresas a programas de incentivos públicos, de forma

a extrair informação relevante para a predição de comportamentos futuros destas empresas durante o desenvolvimento dos projetos financiados pelos programas de incentivos. O objetivo concreto definido é a predição da anulação de projetos com base na informação textual das suas candidaturas.

Foram conduzidas experiências de análise de similaridade entre candidaturas, extração de tópicos presentes no conjunto de textos, estudo de vocabulário específico utilizado em subdivisões dos dados, análise de variedade lexical, identificação de entidades presentes nos textos, análise de classes gramaticais presentes nos textos, análise de preenchimento dos vários campos textuais das candidaturas, caracterização de dimensão textual e utilização de algoritmos de *machine learning* e *deep learning* na predição de anulação de projetos durante o seu decorrer. Este trabalho integra-se na grande conjuntura de ações tomadas relativas ao desenvolvimento do projeto "IA-SI - Inteligência Artificial na Gestão de Incentivos". Indica-se ainda que para esta dissertação apenas foram utilizados dados disponibilizados pelo IAPMEI.

Definem-se como questões de investigação para esta dissertação: "Quão bem se pode predizer a anulação de projetos apenas com base na informação textual dos formulários de candidatura?"; "É vantajoso para a predição de anulação de projetos a utilização de variáveis provenientes de diferentes métodos de extração de variáveis?".

1.2. Metodologia

A metodologia de organização do projeto utilizada foi *Cross-Industry Standard Process for Data Mining*, (CRISP-DM). Esta metodologia é composta pelas etapas: análise do problema, definição dos objetivos e definição do plano de projeto; compreensão dos dados e análise exploratória; preparação dos dados (incluindo pré-processamento dos dados, extração de variáveis e seleção de variáveis); modelação; avaliação de resultados; conclusões do projeto. Estas etapas interligam-se de forma cíclica possibilitando um aproveitamento do conhecimento adquirido nas várias fases para o desenvolvimento contínuo e integral do projeto.

Sendo que a análise de *text mining* desenhada para esta dissertação culmina na predição/ classificação dos projetos segundo uma variável-alvo, o processo de trabalho nos dados foi então organizado como uma cadeia de classificação textual. O desenvolvimento de uma cadeia de classificação textual trata-se de uma abordagem especifica às etapas de "Preparação dos dados" e "Modelação" do CRISP-DM.

Foi montada uma cadeia de classificação de texto para desenvolver a análise de *text mining*. Como enunciam Kowsari et al. (2019), o processo de classificação de texto é composto por

quatro etapas principais. Este tópico acaba por ser bastante amplo pois, segundos os autores, podemos então abordar a extração de variáveis, seleção de variáveis, seleção de classificadores utilizados (ou seja, os melhores algoritmos de classificação para os dados em causa) e finalmente as métricas de avaliação de resultados. Também não pode ser esquecido nesta sequência, um passo inicial igualmente importante, o pré-processamento do texto, isto pois em muitos algoritmos estatísticos e probabilísticos, ruído e variáveis desnecessárias podem provocar efeitos adversos na performance dos sistemas (Kowsari et al., 2019).

1.3. Estrutura do Documento

Este documento encontra-se estruturado na forma imediatamente descrita. Na próxima secção, "Enquadramento conceptual", são estudadas as estruturas e composição das cadeias de classificação de texto da literatura relacionada. Esta análise encontra-se subdividida pelas fases do processo previamente enunciadas. São abordadas as técnicas e métodos mais comuns e as suas vantagens e desvantagens. É também, por vezes, feita referência a situações inovadoras dentro de cada etapa, caso estas existam na literatura analisada. Após este capítulo vem a "Contextualização dos dados", neste capítulo é referida a origem dos dados, a seleção dos dados pretendidos de entre os disponíveis, e é feita uma caracterização da informação selecionada. Seguidamente, na "Metodologia" são indicadas as fases deste projeto, bem como todas as experiências efetuadas, seu propósito e os pormenores necessários para a sua execução. Indica-se o pré-processamento utilizado, as experiências de extração de variáveis, qual a escolha de método de seleção de variáveis, de algoritmos de classificação e as suas minudências, bem como, que métricas foram utilizadas para avaliar os resultados obtidos. Segue-se o capítulo de "Análise de resultados e discussão" onde são apresentados os resultados alcançados. São exibidas características das variáveis extraídas e resultados de predição dos algoritmos de classificação. Por fim, são expostas as conclusões, o contributo deste estudo, as limitações do mesmo e as propostas de trabalho futuro.

Esta página foi intensionalmente deixada em branco

Capítulo 2- Enquadramento conceptual

2.1. Metodologia

Para selecionar a literatura relevante para elaborar o enquadramento conceptual desta dissertação, foi utilizado o método Revisão Sistemática de Literatura. Como fonte de pesquisa selecionou-se a plataforma Scopus (https://www.scopus.com), descrita pela própria como "o maior banco de dados de resumos e citações de literatura revista por pares". A *query* utilizada para encontrar literatura relevante encontrou 168 documentos. Dos 168 foram retirados todos os documentos com as seguintes características: mais de três anos e menos de uma citação por ano, em média. Ficaram assim 162 documentos disponíveis. De seguida, procedeu-se a uma classificação de cada um dos 162 documentos como "relevante" ou "não relevante". A classificação foi baseada nos títulos e resumos dos documentos. Foram aplicados, posteriormente, outros critérios como: avaliar como "não relevante" documentos focados em *sentiment analysis*, documentos em que a análise se foca em textos curtos e documentos cuja análise se foca em textos escritos numa língua estrangeira que não inglês. Estes critérios têm como objetivo selecionar apenas literatura que trate a aplicação de técnicas e métodos de classificação de texto a dados com a maior semelhança possível às candidaturas a fundos públicos analisadas nesta dissertação. A língua inglesa trata-se de uma exceção obrigatória derivado da enorme quantidade de literatura apenas disponível em inglês. Depois da classificação atribuída, foram selecionados apenas os documentos classificados como "relevantes". Dos 162 artigos, restaram apenas 56. Dos 56 documentos selecionados, após uma pesquisa pelos mesmos, foi apenas possível aceder ao conteúdo de 48.

2.2. Análise de conteúdos

Segundo Reddy & Chaudhary (2021), o *text mining* é importante no desenvolvimento de várias atividades, sendo que possui múltiplas aplicações, tais como:

- *Information retrieval;*
- *Language identification;*
- *Opinion mining and sentiment analysis;*
- *Spam filtering;*
- *News article classification;*
- *Webpage classification;*
- *Creating suggestion and recommendations;*

- *Automated QA's;*
- *Text summarization.*

No entanto, qualquer que seja o domínio da sua aplicação, o seu emprego baseia-se, por norma, sempre numa cadeia de processos bem definida e estrturada. Grande parte dos artigos analisados no contexto da área de aplicação de *text mining* a classificação textual, constituem uma aplicação de técnicas a diferentes tipos de *corpus*, sendo comum o objetivo concreto de alcançar o melhor desempenho possível, espelhado nas métricas de avaliação de resultados. Faz então mais sentido analisar a estrutura e construção da cadeia de classificação, do que o tipo de aplicações ou os dados em causa.

Seguidamente, são apresentadas as fases integrantes do processo de construção de uma cadeia de classificação de texto. Em cada fase são focadas técnicas e métodos encontrados na literatura a seu respeito.

2.2.1. Pré-processamento

Quanto ao pré-processamento identificam-se várias técnicas utilizadas pela maioria dos autores. A tokenização é utilizada pela globalidade dos autores como requisito inicial do processo de pré-processamento de texto. A tokenização é a primeira técnica aplicada, esta consiste numa partição do texto em palavras, frases, símbolos ou elementos relevantes, de seu nome *tokens*. A tokenização é necessária na identificação de palavras chave no texto (Verma et al., 2014). De seguida, as técnicas mais comuns observadas em trabalhos relacionados foram: expressões regulares, normalização, remoção de pontuação, remoção de *stopwords*, lematização e *stemming*.

Elhadad et al. (2020) aplicam expressões regulares como forma de limpar o texto de palavras não inglesas bem como de qualquer símbolo, já Ali et al. (2021) utilizam expressões regulares como meio de averiguar e corrigir erros de ortografia.

Normalização ou estandardização, de acordo com Reddy & Chaudhary (2021), trata-se de uma etapa de pré-processamento de texto que consiste em nivelar todos os *tokens* permitindo que estes sejam posteriormente analisados de forma equivalente. A normalização elimina acentos e transforma todas as letras em maiúsculas ou minúsculas. Kowsari et al. (2019) afirmam ser esta a forma mais comum de lidar com a variação inconsistente de letras maiúsculas e minúsculas num texto. Esta técnica tem como desvantagem: impossibilitar a interpretação fidedigna de algumas palavras, tais como siglas e acrónimos.

Pontuação e caracteres especiais são essenciais no que diz respeito ao entendimento e interpretação de um texto por parte de um humano. No entanto, quando estes caracteres são analisados por um algoritmo de classificação, normalmente, interferem na sua interpretação acerca do conteúdo alvo de análise. Por esta razão, Kowsari et al. (2019) aconselham a sua remoção como parte do processo de pré-processamento do texto. Vários autores aplicam este método de limpeza nos seus conjuntos de dados textuais (Ali et al., 2021; Alsaidi et al., 2020; Asim et al., 2019; Bayrak et al., 2022; Chai et al., 2013; Du et al. 2019; Kumar et al., 2021; Rustam et al., 2020; Reddy & Chaudhary 2021; Triantafyllou et al., 2020). Contrariamente, Zhang & Lee (2006) consideram legitimo considerar pontuação e caracteres especiais como *tokens*, especialmente na classificação de *emails* de *spam*. Pois, no *spam* é comum inserir pontuação ilógica nos textos com o intuito de enganar os algoritmos de filtragem de *emails*. Nestes casos a análise da pontuação é crucial na deteção deste tipo de *emails*.

A remoção de *stopwords* é outra técnica de limpeza de texto largamente utilizada pelos autores do conjunto de artigos selecionados. *Stopwords* tratam-se de palavras demasiado comuns no conjunto de documentos que estejamos a estudar. De acordo com Amer & Abdalla (2020), as *stopwords* devem ser removidas do texto durante o processo de pré-processamento. A razão apresentada para a remoção destas palavras é que a sua elevada frequência na maioria dos documentos, torna-as irrelevantes como variáveis classificadoras dos documentos. Estas palavras podem ser identificadas por um limite máximo de documentos em que surgem, sendo classificadas como *stopwords* as que excedem esse limite ou utilizando uma lista pré-definida de *stopwords*. Sendo esta uma técnica reconhecida como benéfica na classificação de texto, é normalmente apenas declarado que foi utilizada sem dissertar muito sobre o tema.

Muitas línguas têm variações da mesma palavra (por exemplo, em grau, género, número e tempo). O *stemming* converte as palavras para a sua raiz. Um dos algoritmos mais usados nesta tarefa é o algoritmo de Porter–Stemmer (Verma et al., 2021). A lematização remove os sufixos e os prefixos de forma a reduzir as palavras ao seu lema. O lema difere da raiz pois os lemas correspondem a palavras presentes no dicionário (Skenderi et al., 2021).

2.2.2. Extração de variáveis

No contexto de mineração textual, os dados textuais carecem de ser convertidos para um espaço estruturado de variáveis que permita a sua análise por parte dos modelos matemáticos dos classificadores. Para o efeito existem dois métodos principais de extração de variáveis de dados não estruturados: "*weighted word techniques*" e "*word embedding techniques*".

A forma mais básica de aplicação do método "*weighted word*" trata-se da extração dos valores de TF (*term frequency*), que basicamente, consiste em atribuir a cada palavra um número que representa a quantidade total de vezes que o termo surge no texto. Outras variações deste método são a extração de um valor booleano que apenas assinala a presença ou ausência do termo e variações que traduzem os valores de TF em escalas logarítmicas. Este método cria vetores de igual tamanho para cada palavra, tendo esses vetores a dimensão do vocabulário do corpus ao qual este é aplicado. São vistas como limitações do método: a larga importância que este dá a palavras com pouco significado, mas comumente utilizadas no diálogo e o facto de desconsiderar a gramática e a ordem de ocorrência dos *tokens*. (Kowsari et al., 2019). Verma et al. (2021) também designam este método por *count vectorizer*. Este método enfrenta também um problema de escala devido ao facto de codificar cada palavra no vocabulário como um vetor *one-hot encoding*. Trata-se do método mais simples de extração de variáveis textuais (Kowsari et al., 2019; Reddy & Chaudhary, 2021; Verma et al., 2021).

TF-IDF, corresponde a uma versão avançada do método TF, sendo que a principal diferença é que representa, não só, a importância de um *token* num documento, mas também a importância deste no conjunto de documentos (Verma et al., 2021). K. Sparck Jones propôs a utilização do método IDF (*inverse document frequency*) associado ao TF, como uma forma de diminuir o efeito nos algoritmos de palavras comuns no corpus, visto estas não terem capacidade de distinguir entre documentos (Jones, 1972, como citado em Kowsari et al., 2019). Tal como declarado no trabalho de Elhadad et al. (2020), sendo TF (t, d) a frequência do *token* t do documento d presente no corpus D e IDF (t, D) a frequência inversa de tem D:

$$\text{TF}-\text{IDF}\,(t, d) = \text{TF}\,(t, d) \times \text{IDF}\,(t, D) \qquad (1)$$

Apesar de TF-IDF ser um método mais tradicional de extração de variáveis, é ainda utilizado em muitos trabalhos recentes (Ali et al., 2021; Du et al., 2019; Elhadad et al., 2020; Kumar et al., 2021; Putong & Suharjito, 2020; Reddy & Chaudhary, 2021; Rustam et al., 2020; Skenderi et al., 2021; Triantafyllou et al., 2020; Wang et al., 2017). Este método continua, no entanto, a ter como limitação o facto de não conseguir relacionar o significado dos termos entre si. Neste contexto surgiram mais recentemente, modelos capazes de incluir conceitos como similaridade entre *tokens* e POS-Tagging (Kowsari et al., 2019).

BOW, sigla que significa "*bag-of-words*", é um conceito que engloba as técnicas de extração de variáveis que não consideram a ordem das palavras, tal como é o caso das técnicas TF e TF-IDF.

Ao contrário dos métodos anteriores que geram representações esparsas dos documentos textuais, as "*word embeddings techniques*" geram representações vetoriais densas, de menor dimensionalidade, que capturam mais adequadamente a semântica do texto. As técnicas mais comuns que aplicam este método, segundo a literatura selecionada são Word2Vec, GloVe, FastText.

GloVe, FastText e Word2Vec são algoritmos de extração de variáveis que utilizam *word embeddings*. Citando Kumar et al. (2021), "*Word embeddings* são representações distribuídas que modelam as propriedades das palavras em vetores de números reais num espaço vetorial predefinido, capturando características e preservando suas relações semânticas". Como resultado dessa representação, as palavras com significados semelhantes têm uma representação semelhante.

O algoritmo Word2Vec visa detetar o significado e relações semânticas entre as palavras através do estudo da coocorrência de palavras num determinado corpus. Este utiliza redes neuronais superficiais com duas camadas ocultas, bem como os modelos CBOW e Skip-gram para criar um vetor para representar cada palavra.

GloVe é um algoritmo desenvolvido como um projeto *open-source* pela Universidade de Standford. A abordagem deste algoritmo é muito semelhante à do método Word2Vec. Sucintamente, cada palavra é representada por um vetor de elevada dimensão que foi treinado com base no contexto semântico da palavra. Este algoritmo utiliza para seu funcionamento o modelo Skip-gram e o método LSA (*latent semantic analysis*).

Skenderi et al. (2021) caracterizam FastText como um modelo que aprende a representar textos ao nível da palavra e do caracter. O FastText surgiu para colmatar a incapacidade dos algoritmos Word2Vec e GloVe, por estes não estarem aptos para lidar com palavras fora dos seus vocabulários. O Facebook surgiu então com o FastText que é fundamentalmente uma extensão do Word2Vec. Este modelo representa as palavras como a soma das suas várias representações *n-gram* correspondentes (Kowsari et al., 2019; Kumar et al., 2021).

Wang et al. (2019) mostraram que *word embeddings* treinados pelo algoritmo Word2Vec superam significativamente o método TF-IDF. Afirmam ainda que em NLP, *word embeddings* são a tecnologia de *deep learning* mais bem-sucedida devido à sua capacidade de capturar propriedades semânticas e sintáticas de elevado nível das palavras.

Abonizio et al. (2020) utilizam no seu estudo uma forma diferente de extração de variáveis do texto. Eles propõem a extração de variáveis independentes da língua dos textos, para deteção de *fake news*. Neste trabalho os autores extraem três tipos de variáveis distintas: "*complexity features*", "*stylometric features*" e "*psychological features*".

Complexity features são variáveis que visam incluir a complexidade geral do texto, tanto ao nível da palavra como ao nível da frase. Alguns exemplos destas são: número médio de palavras por frase, número médio de caracteres por palavra, número total de frases e *type-token ratio* (variável que representa o número de palavras únicas no texto dividido pelo número total de palavras do texto).

Stylometric features englobam tudo o que sejam características gramaticais do texto. Para a extração deste tipo de variáveis utilizam-se várias técnicas de NLP, tais como *part-of-speech* (POS) *tagger* e NER (*name entity recognition*). Exemplos destas variáveis são: rácio de adjetivos no texto, rácio de pronomes no texto, rácio de pontuação no texto, número total de letras maiúsculas e rácio de entidades no texto.

Psychological features tratam-se de variáveis relacionadas a processos cognitivos, tais como a polaridade sentimental, que associa o texto ou frases a categorias como "sentimento positivo" ou "sentimento negativo".

2.2.3. Seleção de variáveis

A classificação de texto é um processo que se pode tornar complexo a nível computacional. Com o aumento da quantidade de dados tratados começam a verificar-se constrangimentos em relação ao tempo de processamento das tarefas, bem como à quantidade de memória consumida. Grandes quantidades de dados geram grandes quantidades de variáveis, que a certo ponto, se tornam pesadas para os computadores e algoritmos suportarem. Daí a importância do problema da redução da dimensionalidade de dados, ou melhor dizendo, de variáveis (Dixit et al., 2020).

Alguns métodos mencionados por Kowsari et al. (2019) para atender ao problema da redução de dimensionalidade são: PCA (*principal component analysis*), ICA (*independent component analysis*), *linear discriminant analysis*, NMF (*non-negative matrix factorization*), *random projection*, *random kitchen sinks*, *johnson lindenstrauss lemma*, *autoencoder* e t-SNE (*t-distributed stochastic neighbour embedding*).

Forman (2002) advoga a importância dos métodos de seleção de variáveis. Constata que uma boa seleção de variáveis melhora a taxa de acerto da classificação, ou equivalentemente, reduz a quantidade de dados do conjunto de treino necessários para obter os resultados desejados nos algoritmos de classificação e que conserva recursos computacionais. Kumar et al. (2021) acrescentam ainda que a seleção de variáveis melhora a interpretação e visualização dos dados.

Amazal & Kissi (2021) enunciam no seu trabalho que o objetivo da seleção de variáveis é representar um documento recorrendo apenas às variáveis mais relevantes para o propósito.

Este processo acaba também por reduzir as variáveis utilizadas no processo de classificação, melhorando o desempenho dos algoritmos de *machine learning*. Amazal & Kissi (2021) dividem as técnicas de seleção de variáveis em duas categorias, "*filter*" e "*wrapper*". O método "*filter*" analisa estatisticamente as variáveis, enquanto o método "*wrapper*" utiliza um algoritmo classificador em que experimenta todas as variáveis e distingue as que apresentam melhores resultados no desempenho do mesmo. Quanto ao método "*filter*", a forma *standard* de proceder ao selecionar variáveis é averiguar o resultado que cada variável atinge em determinada métrica e depois escolher as N variáveis com melhor resultado. As métricas mais comumente utilizadas são: *chi-squared, information gain, odds ratio, (log) probability ratio e document frequency*.

Dixit et al. (2020) sugerem a utilização do método "*filter*" no caso de trabalharmos com grandes conjuntos de dados.

Sikelis et al. (2021) divide os métodos de seleção de variáveis nas categorias: supervisionados, não-supervisionados e semi-supervisionados. Os métodos supervisionados utilizam-se quando os dados possuem *labels*. Neste caso apenas é necessário analisar que variáveis contribuem mais para a previsão correta dos alvos. Os métodos não-supervisionados utilizam-se quando não temos *labels* para o nosso conjunto. Estes descartam variáveis redundantes em dois passos. Primeiro as variáveis são agrupadas em *clusters* através de um qualquer método de similaridade, depois são retidas aquelas que apresentarem correlações mais fortes com as restantes variáveis do grupo, assumindo que essas as mais representativas do *cluster*. Métodos semi-supervisionados aplicam-se quando temos grande conjunto de dados sem valores para a variável-alvo e um pequeno conjunto de dados com valores alvo. Nestes casos aplicam-se métodos não-supervisionados ao conjunto grande e métodos supervisionados ao conjunto pequeno e diz tratar-se do método semi-supervisionado.

Sikelis et al. (2021), tal como Amazal & Kissi, 2021, também distingue as técnicas de seleção de variáveis pela forma como utilizam os algortimos de classificação. No entanto, ao invés de considerarem duas categorias, "*filter*" e "*wrapper*", os autores acresecentam uma terceira, "*embedded models*". Os *embedded models* são uma combinação dos métodos *filter* e *wrapper* e surgiram para tentar culmatar as limtações de ambos. Os métodos *filter*, apesar de fáceis de aplicar, não consideram os enviesamentos dos classificadores. Enquanto os métodos *wrapper* selecionam as melhores variáveis que conseguem para um classificador específico, às custas de imensas iterações de treino do modelo e grande custo computacional. Os *embedded models* integram a seleção de variáveis no processo de treino dos modelos. Um exemplo deste método é o método de poda de árvores de decisão. Sikelis et al. (2021) consideram, em relação

a métodos *filter*, as métricas seguintes como sendo as mais comuns na literatura: *chi-square*, ANOVA, *fisher score*, *pearson correlation coefficient* e *mutual information*.

Os métodos de redução de dimensionalidade/seleção de variáveis mais utilizados no conjunto de artigos selecionado foram *chi-square*, *information gain* e *mutual information*.

Chi-squared é a técnica de seleção de variáveis mais utilizada e é maioritariamente aplicada a dados textuais segundo Rustam et al. (2020). Esta métrica é utilizada para verificar a independência entre uma dada variável e uma dada classe (Achilonu et al., 2021).

IG (*information gain*) é uma métrica também conhecida como MI (*mutual information*) por alguns autores, sendo que por outros, MI é considerado como uma variação de IG (Amazal & Kissi, 2021; Bruni & Bianchi, 2020). Esta mede a diminuição da entropia dada uma variável (Forman, 2002). Bates et al. (2016) afirmam que a métrica MI foi crucial no aperfeiçoamento do seu modelo de classificação. Acrescentam que esta possui a vantagem de permitir a escolha do número de variáveis que se pretende utilizar, ao contrário dos *embedded models* que decidem de forma independente a quantidade de variáveis selecionadas. Méndez et al. (2019) posicionam esta métrica como sendo a mais utilizada em aplicações de filtragem de *spam*.

No trabalho de Méndez et al. (2019) é proposto um novo método de seleção de variáveis com foco na aplicação filtragem de *emails* de *spam*. Este trabalho apresentou um novo método capaz de aproveitar informações semânticas para detetar tópicos nos dados, cuja seleção de variáveis mostrou melhores resultados na classificação do que os métodos *information gain* e *linear discriminant analysis*.

Amazal & Kissi (2021) propõem um método designado por MTF-MI (*maximum term frequency-mutual information*), que se baseia em TF (*term frequency*) e MI, para melhorar a qualidade das variáveis selecionadas. Quando comparado com *chi-squared*, IG e MI, a utilização de MTF-MI permitiu a obtenção de melhores resultados por parte dos algoritmos nos testes levados a cabo neste estudo.

Forman (2002) salientou a excelente performance da métrica BNS (*bi-normal separation*) para atingir bons resultados de *accuracy*, *reccal* e *F-measure*. Em termos de *precision*, IG foi superior no estudo comparativo conduzido. Aconselha a utilização de BNS associada à métrica IG.

2.2.4. Algoritmos de classificação

Kowsari et al. (2019) elegem a fase de escolha do algoritmo de classificação como a principal do processo de classificação de texto. O estudo do conjunto de artigos selecionados evidenciou *logistic regression*, *the naïve bayes classifier*, *k-nearest neighbor*, *support vector machines*,

decision tree, *random forest* e *deep learning neural networks* como os métodos de classificação principais aplicados na literatura. Segue-se uma análise de cada um destes algoritmos de acordo com a literatura relevante para este artigo.

LR (logistic regression) é um dos primeiros métodos de classificação, foi desenvolvido por David Cox em 1958. Este algoritmo é um dos mais simples e já foi aplicado em inúmeras áreas de conhecimento. Trata-se de um método de classificação binário, no entanto também existe a variante *multinomial logistic regression* que atribui mais de duas classificações. Este método tem como limitação, o facto de ter como requisito que todas as observações sejam independentes (Cox, 2018; Guerin, 2016; Huang, 2015 como citado em Kowsari et al., 2019). Elhadad et al. (2020) mostram a validade deste velho algoritmo ao utilizarem-no no seu estudo relativo à deteção de informação enganosa sobre COVID-19. Os autores comparam este algoritmo, que foi um dos que obteve melhores resultados no seu estudo, com uma rede neuronal com apenas uma camada.

O método de classificação NBC (*the naïve bayes classifier*), trata-se de um modelo generativo e é o algoritmo mais tradicional de todos os métodos de classificação. Este popular método tem a seu favor não ser demasiado custoso computacionalmente. Possui também uma versão chamada *multinomial naïve bayes classifier* que permite a classificação por múltiplos *labels*. Este método faz uma forte assunção acerca da distribuição dos dados, o que o limita. É também fortemente afetado pela escassez de dados (Li et al., 2001; Rajan et al., 2017; Soheily-Khah et al, 2018; Wang et al., 2012 como citado em Kowsari et al., 2019). Dixit et al. (2020) destacam a alta sensibilidade deste algoritmo a métodos de seleção de variáveis.

KNN (k-nearest neighbor) é um modelo não-paramétrico fácil de implementar que se adapta a qualquer tipo de conjunto de variáveis. Consegue classificar os dados em várias classes. O seu principal problema é a larga quantidade de armazenamento que necessita no processo de cálculo dos vizinhos mais próximos, especialmente para grandes conjuntos de dados. Carece também da necessidade de encontrar uma função de distância que seja significativa para o contexto, o que o torna bastante dependente dos dados (Patel & Srivastava, 2014; Sahgal & Parida, 2014; Sahgal & Ramesh, 2002; Sanjay et al., 2018 como citado em Kowsari et al., 2019). No estudo de Caccamisi et al. (2020) este mostrou-se ser o algoritmo mais rápido utilizado para classificação.

SVM (support vector machines) é uma poderosa técnica de *machine learning* que se serve de classificadores discriminativos, esta surgiu em 1990. O seu propósito é essencialmente o reconhecimento de padrões e a regressão (Dixit et al., 2020). Encontramos este algoritmo em imensas áreas de *data mining*: *bioinformatics*, classificação de imagens e vídeos, classificação

de atividade humana, segurança e proteção, etc. É também utilizado frequentemente como *baseline* para investigadores compararem as suas novas propostas para a classificação de texto. Este método tem estado em destaque entre os algoritmos de *machine learning* desde que foi criado. A sua principal fraqueza assenta na sua débil explicabilidade de resultados. (Guo, 2014; Karamizadeh et al., 2014 como citado em Kowsari et al., 2019). Zhang & Lee (2006) observam que os algoritmos SVM atingem melhores resultados do que modelos generativos, que estes conseguem ultrapassar o problema da elevada dimensionalidade dos dados e que nas suas experiências em inglês, chinês e grego estes obtêm resultados excecionais em várias tarefas de classificação. Achilonu et al. (2021) escolheram o SVM como um dos métodos de classificação que utilizaram no seu trabalho, por este ter sido um dos melhores algoritmos de classificação de relatório médicos de patologias cancerígenas em vários outros estudos. Bikku et al. (2018) declara o algoritmo RVM (*relevance vector machines*), como uma versão melhorada do SVM, que oferece melhores resultados do que os outros algoritmos, incluindo o SVM.

DT (decision tree) ou árvore de decisão e RF (random forest) são ambos *tree-based classifiers*. As árvores de decisão são sistemas de decomposição hierárquica do espaço de variáveis. Estas têm como objetivo criar uma estrutura hierárquica de importância de variáveis e é precisamente na decisão entre quais variáveis deverão ser atribuídas aos nodos pais e filhos que está o desafio deste algoritmo. As árvores de decisão são algoritmos muito rápido tanto no treino como na predição. Sofrem, contudo, de uma elevada sensibilidade a perturbações nos dados. São também suscetíveis de *overtfitting* (Aggarwal & Zhai, 2012; Giovanelli et al., 2017; Jasim, 2015; Morgan & Sonquist, 1963; Quinlan, 1986; como citado em Kowsari et al., 2019). Ma et al. (2009) consideraram DT como o melhor entre os cinco algoritmos de *machine learning* utilizados no seu estudo de deteção de *phishing emails*. Conway et al. (2009) aplicam um algoritmo baseado em DT no seu trabalho de classificação de relatórios médicos e destacam a oportunidade de exploração dos dados que este algoritmo oferece.

O algoritmo RF utiliza várias árvores de decisão aleatórias em paralelo. Depois de treinadas todas a árvores, o algoritmo calcula as previsões com base numa fórmula matemática designada por *voting*. É um algoritmo rápido de treinar quando comparado com métodos de *deep learning*, no entanto, é moroso na predição (Bansal et al., 2018; Jasim, 2016; Wu et al., 2004 como citado em Kowsari et al., 2019). Achilonu et al. (2021) constata a capacidade deste algoritmo de conseguir atingir bons resultados de forma consistente e denota a sua supremacia em relação a algoritmos que utilizam uma única árvore de decisão.

Nos últimos anos temos assistido à crescente preponderância das abordagens de *deep learning* no campo da classificação de texto, classificação de imagens, NLP, reconhecimento

facial, entre outros. Tudo isto deve-se à capacidade incomparável, por parte destes algoritmos de modelar relações complexas e não lineares nos dados (LeCun et al., 2015 como citado em Kowsari et al., 2019). Du et al. (2019) concluíram através do seu trabalho de classificação de documentos biomédicos que as redes neuronais conservam também a vantagem de serem capazes de apreender variáveis complexas de forma independente. Prusa & Khoshgoftaar (2017) confirmam e acrescentam que esta característica retira do processo de extração de variáveis o enviesamento humano e permite a extração de maior quantidade de informação dos textos. Possibilita ainda que não seja necessário conhecimento especializado na área de estudo para conseguir extrair variáveis relevantes. Segundo Kowsari et al. (2019), existem três arquiteturas de redes neuronais para classificação de texto: DNN (*deep neural networks*), RNN (*recurrent neural networks*) e CNN (*convolutional neural networks*).

A implementação de DNN é um modelo discriminativo treinado que se serve de um algoritmo de *standard back-propagation* e utiliza a função *sigmoid* como função de ativação (Nair & Hilton, 2010 como citado em Kowsari et al., 2019). Prusa & Khoshgoftaar (2017) demonstraram um novo método de criação de representação textual ao nível do caracter através da utilização de redes DNN, que comparados com outros métodos, reduziram o tempo de treino, o consumo de memória e alcançaram melhores performances de classificação.

As RNN consideram a informação vinda de nodos anteriores da rede de uma forma sofisticada que permite uma melhor análise semântica da estrutura dos dados. Estas redes funcionam pela utilização de LSTM *networks* (*long short term memory*) ou GRU (*gated recurrent units*). LSTM são redes muito populares especificas para a tarefa de modelar dados sequenciais, pelo que são adequadas para o processamento de texto. São eficazes em tarefas como sentiment analysis superando o algoritmo DVM em performance (Tang et al., 2015 como citado em Prusa & Khoshgoftaar, 2017). As redes LSTM são, porém, lentas no processo de treino, propensas a *overfitting* e a sua parametrização pode ser considerada complexa (Prusa & Khoshgoftaar, 2017). Bayrak et al. (2022) relatam a boa performance de redes LSTM no seu trabalho. Borges et al. (2019) utilizam redes LSTM na deteção de *fake news* e alertam acerca de estas alcançarem melhores resultados quando aplicadas à análise de textos curtos. As RNN têm sido muito utilizadas em tarefas de reconhecimento de voz, descrição de imagens e outras tarefas de processamento de dados sequenciais (Wu et al., 2020).

CNN são redes neuronais que apesar de terem sido originalmente desenhadas com o propósito de processar imagens têm vindo a ser utilizadas com sucesso na classificação textual (LeCun et al., 1998; LeCun et al., 2015 como citado em Kowsari et al., 2019). Estas são bastante usadas para classificar hierarquicamente documentos (Jaderberg et al., 2016; Lai et al., 2015

como citado em Kowsari et al., 2019). De acordo com Prusa & Khoshgoftaar (2017) este popular tipo de redes mostrou ser um método eficaz de extração de variáveis textuais. Os autores corroboram ainda a grande capacidade destas redes ao afirmarem que uma rede CNN apenas com uma única camada *convolucional* tem melhor performance do que métodos como SVM ou *multinomial naïve bayes classifier*.

As principais limitações das redes neuronais são a ausência de explicabilidade de resultados e o facto de este método requerer grandes quantidades de dados para o seu funcionamento, não sendo possível utilizar estas redes em conjuntos de dados de menores dimensões (Kowsari et al., 2019).

Outros algoritmos menos comumente abordados pela literatura, mas ainda assim presentes, são o XGBoost (*extreme gradient boosting*), AdaBoost (*adaptive boosting*) e K-Means.

2.2.5. Métricas

É essencial perceber a mecânica de cada métrica de avaliação de resultados, isto pois a sua lógica varia. Para uma comparabilidade de resultados coerente é fulcral um entendimento do tipo de informação que as métricas de avaliação de resultados nos transmitem. A matriz de confusão é constituída pelo cruzamento de duas dimensões, "classe verdadeira" e "classe prevista" que gera quatro categorias: verdadeiros positivos (*true positive* - TP), falsos positivos (*false positive* - FP), falsos negativos (*false negative* - FN) e verdadeiros negativos (*true negative* - TN). Estas categorias, através de diferentes formulações matemáticas dos valores da matriz, permitem a criação das métricas *recall*, *precisio*n, *accuracy*, *sensitivity*, *specificity* e *F-measure*, que são normalmente as métricas mais utlizadas (Kowsari et al., 2019; Reddy & Chaudhary, 2021).

Outra métrica também muito utilizada para avaliar a performance dos algoritmos de classificação é a AUC (*area under ROC curve*). Resume-se à medição da área abaixo da curva ROC. Esta métrica não se trata da mais apropriada quando utilizamos dados não balanceados (Kowsari et al., 2019).

2.3. Trabalho relacionado – exemplos e comparação

Vimos como a literatura selecionada organiza o seu processo de aplicação de *text mining* como método de captação de informação relevante a partir de dados não estruturados. Foram descritas as etapas principais de um projeto de text mining *standard* e quais os métodos mais utilizados em cada uma das etapas. Mais abaixo, está exposto um quadro resumo dos métodos aplicados

em dois artigos, que representam bons exemplos de um projeto de classificação de texto. Na Tabela 1 é possível ver as técnicas que alguns autores consideraram racional utilizar em cada etapa da cadeia da classificação de texto nos seus trabalhos, tal como uma breve nota sobre os seus melhores resultados.

Os dois artigos selecionados, como exemplo de uma cadeia de classificação de texto, servem também para comparação com este mesmo trabalho. Comparação de métodos e técnicas utilizadas e resultados, pois são semelhantes em estrutura e objetivos a este mesmo estudo. Além do mais são ambos artigos recentes com uma maior panóplia de técnicas, métodos e meios disponíveis para sua utilização. Os artigos referidos são: "Classification model of contact center customers emails using machine learning", Putong & Suharjito (2020) e "Detecting misleading information on COVID-19", Elhadad & Gebali (2020).

O trabalho de Putong & Suharjito (2020) foca-se na classificação de *emails* de clientes de um *contact center* em quatro categorias. O objetivo principal dos autores neste estudo foi criar uma cadeia de classificação de texto a mais exata (com maior taxa de acerto) que conseguissem. Como pré-processamento utilizaram as técnicas: tokenização; conversão para *Lowercase*; remoção de *stopwords*; *stemming*. Como técnicas de extração de variáveis os autores testaram os métodos TF-IDF e Word2Vec. Não utilizaram técnicas de seleção de variáveis. Finalmente, os algoritmos selecionados foram o SVM, Naïve Bayes e o KNN. Os melhores resultados foram conseguidos combinando a extração de variáveis através do método Word2Vec com o algoritmo de classificação SVM, atingindo uma accuracy de 77.85%.

Em "Detecting misleading information on COVID-19", Elhadad & Gebali propõem-se a criar um modelo de deteção de informações "enganosas". Coletaram dados, tanto verdadeiros como falsos, sendo que os verdadeiros são dados com proveniência de instituições e fontes com elevada credibilidade. De seguida, desenvolveram uma cadeia de classificação de texto com o fim de classificar as informações recolhidas acerca do COVID-19 como verdadeiras ou falsas Este projeto aparenta ser muito completo devido ao elevado número de técnicas que experimenta. Comparando o seu pré-processamento com o respetivo do artigo previamente analisado, Elhadad & Gebali, aplicam não só o mesmo pré-processamento, como ainda, convertem símbolos numéricos para texto, convertem todas as letras para minúsculas à exceção das letras maiúsculas essenciais para a compreensão textual, removem tokens com menos de dois caracteres e utilizam *part-of-speech* para selecionar apenas algumas categorias gramaticais. São também utilizados dez algoritmos de classificação e doze métricas de avaliação de resultados derivadas da matriz de confusão. Os melhores resultados foram

alcançados pelos algoritmos classificadores NN, LR e DT, com resultados de *accuracy* na casa dos 99%.

Tal como os artigos apresentados, também este projeto se propõe a criar uma cadeia de classificação de texto em que o objetivo principal é atingir o melhor desempenho possível na predição feita pelos algoritmos, mais concretamente na predição em relação à anulação dos projetos de incentivos empresariais aceites. Neste artigo são utilizadas diversas técnicas em cada uma das etapas de classificação de texto e são também elaboradas várias experiências no âmbito da etapa de extração de variáveis (que serão discriminadas em capítulos seguintes) de forma a capturar características não linguísticas dos textos que possuam capacidade preditiva. Ao contrário dos artigos analisados e presentes na seguinte tabela, são empregues técnicas de seleção de variáveis na etapa devida.

Tabela 1 - Exemplos da pipeline de classificação de texto

Artigo:	Pré-processamento:	Extração de Variáveis:	Seleção de Variáveis:	Algoritmos:	Métricas de Avaliação:	Melhor Resultado Alcançado:
Classification model of contact center customers emails using machine learning, Putong & Suharjito (2020)	• Conversão Lowercase • Tokenization • Remoção de Stopwords • Stemming	• TF-IDF • Word2Vec	-	• Naïve Bayes • SVM • KNN	• Precision • Recall • F-Score • Accuracy	(Word2Vec + SVM) Accuracy = 77.85%
Detecting misleading information on COVID-19, Elhadad & Gebali (2020)	• Conversão Lowercase • Regular Expressions • Tokenization • Remoção de Stopwords • Remoção de Símbolos • Conversão de símbolos numéricos para números por extenso • Part-of-Speech • Stemming	• TF-IDF • BOW • N-Grams • Word-embeddings	-	• Linear SVM • CNN • Decision Tree • Multinomial Naïve Bayes • Multinomial Naïve Bayes • Bernoulli Naïve Bayes • XGBoost • LR • Esemble RF • Perceptron, • Neural Network	• Accuracy • Error Rate • Precision • Recall • F-Score • AUC • Specifity • Geometric-Mean • Miss Rate • Fall-Out Rate • False Discovery Rate • False Omission Rate	(NN, LR e DT) Accuracy > 99%

Esta página foi intensionalmente deixada em branco

Capítulo 3- Contextualização dos dados

Neste capítulo é feito um enquadramento acerca dos dados envolvidos neste trabalho, de forma a que seja percetível qual a sua origem, quais foram os dados selecionados para serem analisados e são expostas algumas das suas características para melhor entendimento dos mesmos.

3.1. Enquadramento

Como já referido anteriormente, esta dissertação foi desenvolvida em associação com o projeto de investigação "IA-SI - Inteligência Artificial na Gestão de Incentivos", desenvolvido em colaboração pelo ISCTE com a AICEP e o IAPMEI, instituições com responsabilidades na gestão e distribuição de fundos públicos às empresas nacionais. Iremos focar-nos apenas no IAPMEI, pois foram utilizados apenas os seus dados nesta dissertação.

Entre todas as fases do processo de gestão e distribuição de fundos em causa, encontram-se algumas mais importantes para esta dissertação. Procede-se de seguida a uma breve e simples explicação do processo de atribuição de fundos: as empresas interessadas em concorrer aos programas de incentivos disponibilizados pelo IAPMEI candidatam-se aos mesmos; as candidaturas são avaliadas, algumas são aceites e outras não; as candidaturas aceites tornam-se em projetos em desenvolvimento; alguns destes projetos são anulados durante a sua duração, outros são concluídos passada a sua duração total.

3.2. Origem dos dados

O conjunto de dados utilizado nesta dissertação foi cedido pelo IAPMEI, com o fim de permitir o estudo da aplicação de técnicas de inteligência artificial à gestão de incentivos empresariais. Os dados cedidos consistem em informação de empresas cuja candidatura foi aceite, concretamente, os formulários de candidatura e dados acerca do desenvolvimento do projeto respetivo e da sua avaliação. Os dados contam com a informação relativa às 2793 candidaturas aceites entre os anos de 2014 e 2020. Dos 2793 projetos resultantes das candidaturas aceites, alguns já foram concluídos, outros ainda se encontram em fase de desenvolvimento.

3.3. Seleção dos dados

Numa fase anterior do projeto de investigação à qual este estudo está associado, os dados referidos tiveram de ser extraídos a partir de ficheiros com extensão ".xml" para ficheiros no formato ".csv" e estruturados numa base de dados relacional. A base de dados criada conta com 72 tabelas e 1178 campos, com vários tipos de dados. Sendo este um trabalho sobre classificação de texto, foi essencial restringir a análise apenas a campos textuais.

Os campos textuais das candidaturas analisados encontravam-se numa das referidas tabelas. Tabela esta, com 395 campos divididos em 46 categorias. Exemplos dessas categorias são: dados do promotor da candidatura, dados da consultora envolvida na candidatura, dados do projeto, descrição do projeto, descrição física da empresa, entre outras. Os diversos campos guardam valores do tipo *int*, *float* e *string*. Para selecionar os campos relevantes para análise neste artigo, o primeiro passo foi restringir a seleção inicial apenas a campos que só admitissem dados do tipo *string*. Passámos a considerar apenas 184 dos 395 campos. Depois deste passo, os campos foram selecionados com base no seu conteúdo, sendo então selecionados aqueles cujo teor correspondesse a texto não estruturado. Evitamos assim a seleção de campos com dados do tipo *string* irrelevantes, tais como nomes de ficheiros submetidos, por exemplo, tendo em conta que existem vários deste género. Após a filtragem, restaram 99 campos textuais, relativos a 29 das 46 categorias. Os campos textuais, tais como outros campos presentes nos formulários de candidatura encontram-se divididos por categorias tais como "Informações sobre a empresa promotora", "Informações sobre a empresa consultora", entre outras.

3.4. Caracterização dos dados

A caracterização dos dados originais (entenda-se, sem pré-processamento), serviu igualmente como análise exploratória. A análise exploratória teve especial relevância na fase de "Extração de Variáveis", pois revelou facetas caracterizadoras dos dados que poderiam eventualmente representar variáveis não linguísticas com poder preditivo.

Seguidamente são expostos os resultados da caracterização dos dados elaborada segundo quatro perspetivas: data de início de projeto, valores nulos nos dados, dimensão textual e conteúdo textual.

Com recurso à Figura 1, começamos por analisar a distribuição das candidaturas por ano de submissão das mesmas.

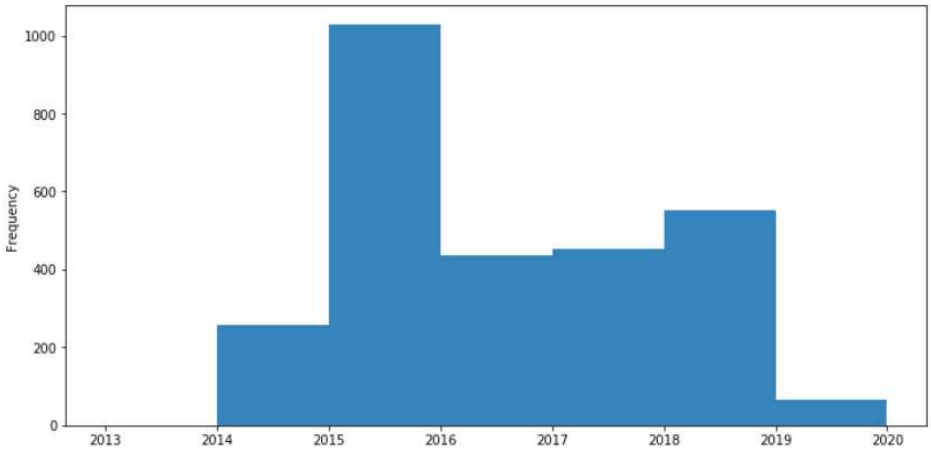


Figura 1 - Distribuição dos projetos por ano de início

Nesta figura verifica-se um máximo de projetos em 2015, bem como um mínimo de projetos em 2019.

Em relação à presença de valores nulos, ou *missing values*, existentes nos campos selecionados, podem observar-se a Figura 2 e a Figura 3. Retratam a distribuição de campos por número de *missing values* e de seguida e a distribuição de projetos por número de *missing values*, respetivamente.

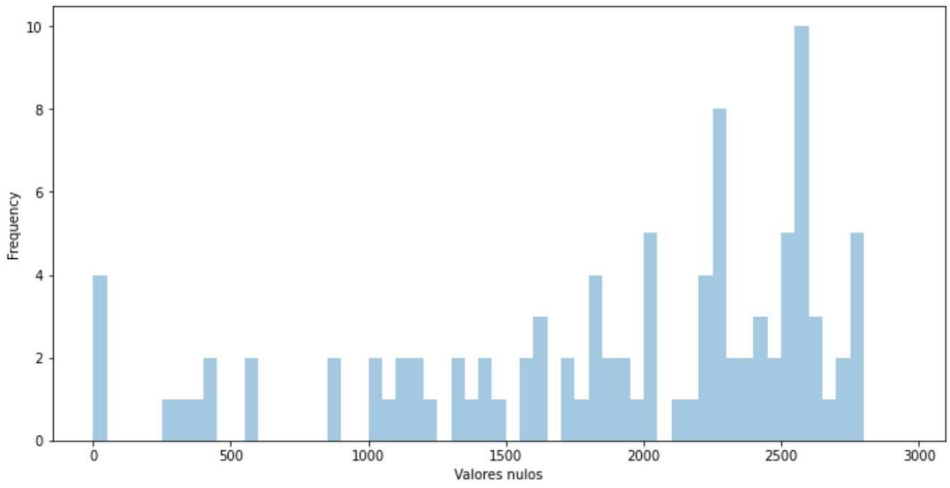


Figura 2 - Distribuição dos campos por nº de valores nulos

O valor mínimo de valores nulos por campo de texto é 0, o valor máximo 2792. O único campo textual preenchido pela totalidade dos projetos é o campo onde consta o nome da empresa promotora da candidatura. Há também outros campos que praticamente nenhum projeto tem preenchidos. A média de valores nulos por campo é 1871

(arredondado às unidades). Combinando esta informação com a visualização da Figura 2, observamos que maior parte dos campos do formulário de candidatura é apenas preenchido por uma minoria de projetos.

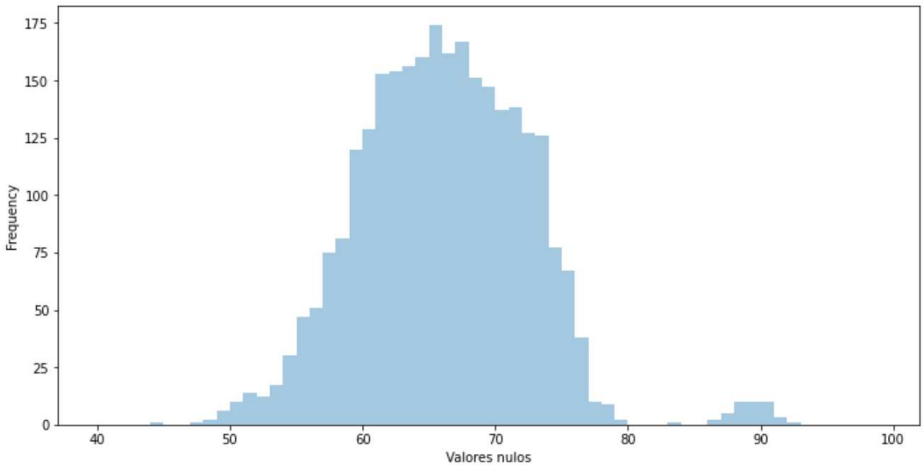


Figura 3- Distribuição dos projetos por nº de valores nulos

O mínimo de valores nulos por projeto é 42 e o máximo de valores nulos por projeto é 92. A média de valores nulos por projeto é 66 (arredondado às unidades), mais de 50%.

Em relação à dimensão textual, foram calculados os números mínimo, médio e máximo de caracteres e *tokens* de cada um dos projetos e são apresentados na Tabela 2. Estes valores refletem a dimensão de um único texto por projeto, este é composto pela agregação de toda a informação textual presente nas variáveis selecionadas como relevantes da candidatura relativa ao respetivo projeto.

Tabela 2 - Nº máximo, mínimo e médio de carateres e tokens dos textos de candidatura dos projetos

	Caracteres	*Tokens*
Max:	135917	20222
Min:	10844	1696
Avg:	58969,01	8873,8

Analisou-se também a distribuição da dimensão textual dos projetos ao longo do tempo (data de início de projeto), por caracteres e por *tokens*.

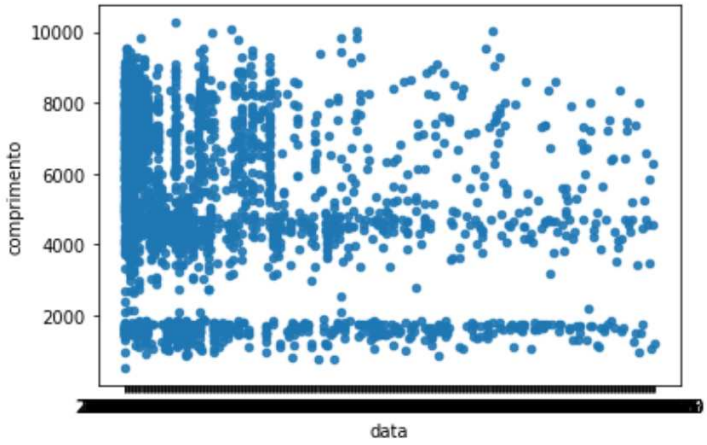


Figura 4 - Distribuição do comprimento textual de cada projeto em tokens por ano de início de projeto

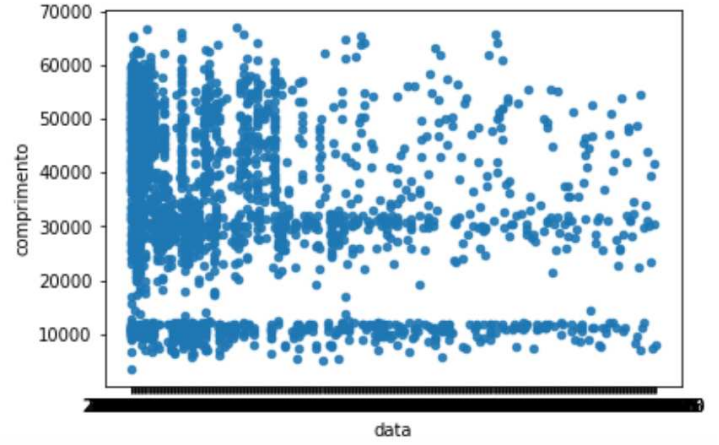


Figura 5 - Distribuição do comprimento textual de cada projeto em caracteres por ano de início de projeto

Para além de ser nítido, através da Figura 4 e Figura 5, a menor densidade de projetos com o passar do tempo, também são percetíveis duas retas horizontais, em ambas as figuras, que mostram uma grande quantidade de projetos com a mesma dimensão textual. As linhas concentram-se nos valores 10000 e 30000, relativamente à Figura 5.

Foram também utilizadas as técnicas TF e IDF como formas de explorar o conteúdo textual dos dados. Esta exploração focou-se em vários níveis de caracterização: conjunto da totalidade dos textos disponíveis, textos de cada campo e textos de cada projeto. Para clarificar, a cada projeto corresponde um texto de cada um dos campos.

Realçar que o papel da técnica TF é contar as ocorrências de cada *token*, sendo que quanto mais elevado o valor de TF, maior preponderância essa palavra tem no texto. Já a

técnica IDF, representa valores que podem ser vistos como uma medida de importância de uma palavra no contexto de uma coleção de documentos.

Foram apuradas as 10 palavras com menor IDF por campo (palavras mais comuns nos vários textos de cada variável), as 10 palavras com maior TF-IDF por campo (palavras que melhor caracterizam os textos de cada campo em comparação com as restantes), as 50 palavras com maior TF (mais frequentes) no conjunto total de textos de candidatura dos projetos (palavras que ocorrem um maior numero de vezes no conjunto total de textos disponíveis), 50 palavras com maior DF (palavras com maior probabilidade de estarem presentes no texto de um projeto) e finalmente a lista ordenada de projetos mais distintos textualmente. A lista ordenada de projetos mais distintos textualmente foi calculada através do somatório dos valores de TF-IDF de todas as palavras presentes no texto de cada projeto. Sendo que os projetos que obtiveram maior valor são distinguidos como os mais distintos e os com menor valor, como os menos distintos.

Em contexto de análise exploratória, estas informações foram analisadas empiricamente como forma de conhecer melhor os dados, no sentido de tomar decisões mais informadas em próximas fases da cadeia de classificação de texto.

Como forma de representar o conteúdo dos dados foram também geradas *word clouds* que facilitam a apreensão do vocabulário preponderante nos textos analisados. São de seguida apresentadas duas *word clouds* representativas dos textos das candidaturas dos projetos, estas diferem entre si, na medida em que, uma delas utilizou a técnica N-grams para agrupar diferentes *tokens* em expressões presentes nos dados em larga escala, a Figura 6, e outra não, a Figura 7.

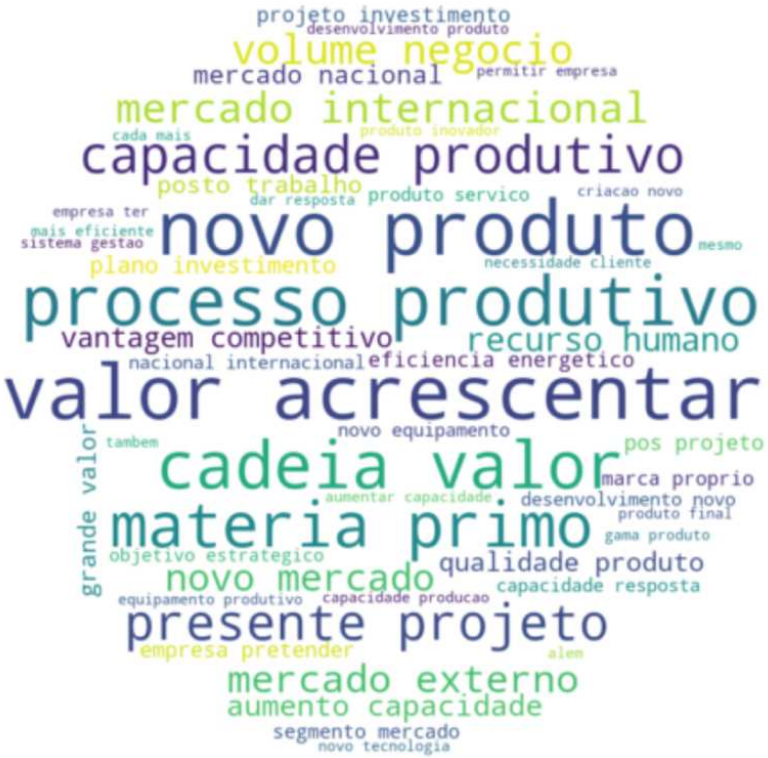


Figura 6- Word cloud dos textos de candidatura de empresas a programas de incentivos, com utilização de N-grams

Figura 7- Word cloud dos textos de candidatura de empresas a programas de incentivos, sem utilizar N-grams

3.5. Caracterização da variável-alvo

Em relação à variável-alvo deste projeto de classificação de texto, esta dispõe de informação acerca de 781 projetos do total de 2793 disponíveis. Destes 781, 328 são projetos anulados e 453 são projetos não anulados. Esta trata-se, portanto, de uma variável binária, na qual o valor 1 assinala um projeto anulado e o valor 0 assinala um projeto não anulado. Na figura seguinte apresenta-se a composição da variável alvo.

Figura 8- Composição da variável-alvo, "Projeto Anulado"

Capítulo 4- Metodologia

Nesta secção é explanado o sistema de organização do trabalho desenvolvido nesta dissertação.

Foi utilizada como estrutura do projeto a metodologia *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Na Figura 9 está exposta a relação das várias fases de que é composta esta metodologia.

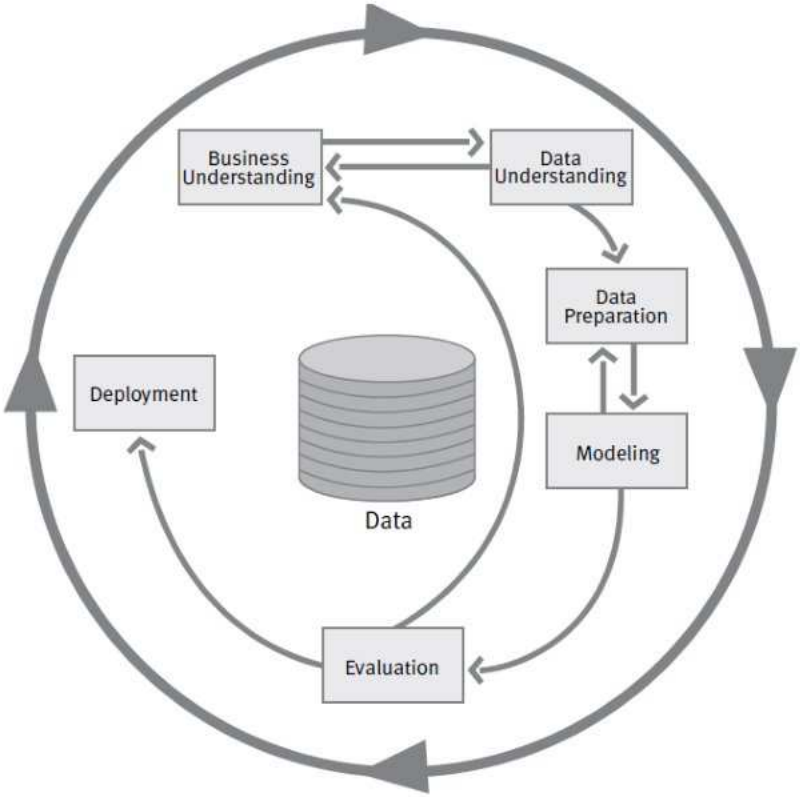


Figura 9 - Fases e estrutura da metodologia CRISP-DM (Fonte: Chapman et al. (2000))

Como é possível observar, as várias etapas de desenvolvimento do projeto interligam-se através de uma configuração cíclica, facilitando uma agregação do conhecimento obtido em cada etapa, de forma à otimização contínua do desenvolvimento integral do projeto.

A cadeia de classificação de texto necessária ao desenvolvimento desta mesma dissertação encaixa-se nas etapas de "Preparação dos dados", "Modelação" e "Avaliação" do CRISP-DM.

No seguimento desta secção são especificadas as ações tomadas para a criação da cadeia de classificação de texto construída. A descrição seguinte encontra-se subdividida

nas seguintes secções: “Pré-processamento dos dados”; “Extração de variáveis”; “Seleção de variáveis”; “Classificação”; “Métricas”.

4.1. Pré-processamento dos dados

Depois de selecionados e caracterizados os dados relevantes para a dissertação, inicia-se então o processo de análise de *text mining*, ou seja, os dados entram na cadeia de classificação de texto. De acordo com a metodologia CRISP-DM, inicia-se a etapa de “Preparação dos dados”. Como etapa inicial do processo de tratamento dos dados, temos o pré-processamento, cuja missão é preparar os dados para a sua utilização em fases seguintes. Tal como já foi referido, o pré-processamento é uma fase crucial para o alcance de bons resultados através da cadeia de classificação de texto gerada.

Foram utilizados dois tipos de pré-processamento. Na primeira versão de pré-processamento foram substituídos os *missing values* pela ausência de texto, utilizou-se a tokenização e agregaram-se os textos por projeto, eliminando as divisões por campo dos textos. Os textos que antes estavam guardados como relativos tanto a um projeto como a um campo de preenchimento, foram agregados por projeto, perdendo assim, a associação ao campo. O objetivo de agregar todos os textos relativos a um projeto num único texto foi a simplificação da estrutura dos dados. Isto, pois, em fases de futura análise, torna-se computacionalmente menos custoso um projeto ser representado apenas por um texto do que cada projeto ser representado por vários textos. Este primeiro pré-processamento teve como fim a extração de variáveis relacionadas com dimensão do texto, com a identificação de entidades mencionadas e com a análise de classes de *tokens* presentes no texto, ou seja, a extração de *complexity and stylometric features*. O segundo tipo de pré-processamento teve o intuito de adicionar ao primeiro tipo limpeza do ruído e condensação de significados das palavras através de lematização, com o intuito de preparar os dados para serem utilizados noutro tipo de experiências. Nesta versão, as primeiras ações tomadas corresponderam à primeira versão de pré-processamento já apresentada. Para a fase seguinte desta segunda versão do pré-processamento foi utilizada uma biblioteca de funções para Python, designada por Stanza (Qi et al., 2020). Recorrendo às suas funções o texto foi novamente tokenizado, foi retirada pontuação, retiradas *stopwords*, procedeu-se à transformação de todos os caracteres para letra minúscula, todos os tokens foram lemmatizados, foram removidas palavras com menos

de três caracteres e foram apenas selecionadas palavras dentro das categorias gramaticais: "Nomes", "Verbos", "Advérbios" e "Adjetivos".

4.2. Extração de variáveis

Nesta etapa deste projeto de classificação de texto, foram testados vários tipos de extração de variáveis, desde métodos mais convencionais até experiências mais exploratórias dos dados. O objetivo foi conseguir extrair variáveis diferenciadas com o maior poder preditivo possível.

Abordando, primeiramente, os métodos observados no capítulo "Enquadramento Conceptual", foram testados como forma de extração de variáveis os métodos: TF e TF-IDF. Depois, para além destes métodos, também foram extraídas variáveis não linguísticas tendo em conta características dos textos através das experiências que são explanadas de seguida.

Uma experiência de extração de variáveis trabalhada foi a utilização de um modelo (probabilístico) de tópicos. O objetivo é descobrir os tópicos/temas latentes numa coleção de documentos textuais. Para tal, este tipo de modelos assume que um documento é uma mistura de tópicos e cada tópico é definido por uma distribuição probabilística de palavras. O modelo de tópicos utilizado para esta experiência foi o Latent Dirichlet Allocation (Blei et al., 2003). Nesta análise usaram-se os projetos disponíveis, 2793. Na criação do modelo de tópicos segundo o algoritmo LDA foi necessário definir o número de tópicos que era pretendido gerar para correr o algoritmo. Para decidir o número de tópicos a pré-definir foram analisados os valores de coerência gerada por modelos com diferentes números de tópicos. A Figura 10 ilustra a variação do valor da coerência com base na variação do número de tópicos.

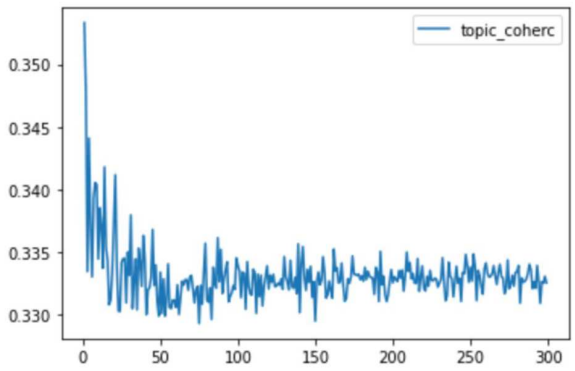


Figura 10- Valor de coerência do modelo de tópicos segundo o número de tópicos gerado

Pretendia-se encontrar um número que apresentasse uma coerência alta e que gerasse tópicos distintos. Com base nesta figura e na análise das relações entre os tópicos dos diferentes modelos gerados, o número de tópicos escolhido foi 14. As preponderâncias de cada tópico, ou seja, valores que expressam o quanto um tópico está representado em cada projeto, foram extraídas como variáveis.

Analisou-se também a existência de vocabulário específico associado às classes dividas pela variável-alvo, projetos anulados e projetos não anulados. Para a análise do vocabulário, recorreu-se à técnica IDF. Este valor pode ser visto como uma medida de importância de uma palavra no contexto de uma coleção de documentos. Quanto maior o valor IDF, mais discriminativa é essa palavra (e, logo, mais importante). Com o objetivo de compreender se existe algum vocabulário mais específico associado à variável-alvo, dividiu-se o conjunto de dados de acordo com esta, tendo-se criado os seguintes subconjuntos: projeto anulados, projetos não anulados. Para compreender qual o vocabulário que melhor caracteriza o conjunto dos projetos anulados começou por se obter as palavras de menor IDF no conjunto dos projetos anulados e não anulados — as palavras com menor IDF em cada um dos conjuntos são as que ocorrem num maior número de projetos em cada conjunto. Em seguida, do conjunto de palavras associado aos projetos anulados retirou-se as palavras associadas ao conjunto dos projetos não anulados, obtendo-se assim o conjunto de palavras que melhor caracteriza os projetos anulados. Através deste procedimento foram identificadas as 10 palavras que melhor caracterizam o conjunto de textos relativo aos projetos anulados. Cada uma dessas palavras originou a extração de uma variável binária que representa a presença ou ausência da palavra nos textos de candidatura dos projetos.

Outra experiência de extração de variáveis deste estudo foi baseada no conceito de similaridade semântica entre textos. A similaridade semântica textual mede o grau de equivalência entre textos, esta é expressa numa escala ordinal que vai de equivalência semântica total até completa disparidade, como enunciam Agirre et al. (2016).

Para estimar a similaridade entre projetos, usou-se a representação vetorial com valores TF-IDF dos textos agregados de cada projeto e calculou-se o cosseno entre pares de vetores. Quanto maior o valor do cosseno do ângulo formado pelos vetores que representam os projetos, mais similares são os textos dos projetos. Com base nas experiências realizadas, as variáveis baseadas na similaridade extraídas são as seguintes:

- valor máximo de similaridade entre projeto em análise e os restantes;
- valor médio de similaridade com os projetos anulados;

- valor médio de similaridade com os projetos não anulados.

Foi também analisada a diversidade lexical dos textos. Diversidade lexical ou LD (lexical diversity), trata-se de uma medida de riqueza lexical. Mas segundo Duran et al. (2004), LD mede mais do que apenas a variedade de vocabulário empregue num texto. Afirmam que conceitos como a riqueza do vocabulário, a flexibilidade e a criatividade verbal também contam no apuramento da diversidade lexical de um texto. Não se trata apenas da dimensão do vocabulário, mas também como este é aplicado.

Para esta análise foram aplicadas as métricas de diversidade lexical MTLD e HD-D aos textos originais (entenda-se sem qualquer tipo de pré-processamento). A escolha destas métricas baseou-se no trabalho de McCarthy & Jarvis (2010), "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment", que aconselham a utilização conjunta das técnicas MTLD e HD-D (esta métrica corresponde a uma versão mais avançada da métrica voc-D). Isto, pois a aplicação simultânea de ambas as métricas permitem uma melhor interpretação dos resultados, evitando conclusões erróneas.

As variáveis extraídas através desta experiência foram os índices de MTLD e HD-D de cada texto. Cada métrica foi gerada 10 vezes para cada texto e de seguida foi apurada a média de cada métrica para cada texto, de forma a criar valores coerentes destas métricas. Este valor foi gerado para cada métrica e foram assim extraídas duas variáveis.

Tal como fazem Iyer & Rose (2019) para a identificação da autoria de textos também neste estudo foram extraídas *linguistic stylometric features*, ou seja, variáveis que consistem na utilização de palavras comuns na escrita. Os autores identificam esta análise como "*lexical usage analysis".* Foram extraídas variáveis que consistem nos valores TF de uma lista de palavras e expressões específicas, concretamente: "e", "mas", "contudo", "no entanto", "apesar de", "se", "também", "mais", "menos", "muito", "bastante".

Foram também extraídas variáveis em relação à dimensão dos textos. As variáveis extraídas foram:

- dimensão do texto em carateres;
- dimensão média de *token* em carateres*;*
- dimensão média de frase em carateres;
- desvio padrão da dimensão de frase em carateres.

Para além destas etapas, explorou-se também a identificação de entidades nomeadas. O reconhecimento de entidades nomeadas, designado por NER (*named entity*

recognition). O algoritmo utilizado para esta tarefa extrai as sequências de palavras que identifica como entidades e classifica-as por tipo: pessoas, organizações, locais ou como uma mistura das anteriores (diferentes abordagens podem utilizar outros tipos de entidades). Os resultados obtidos não foram satisfatórios. Muitas das entidades reconhecidas pelo algoritmo não se tratavam de facto de entidades, sendo apenas palavras com erros ortográficos, sequências de palavras com pontuação pelo meio ou expressões ligadas por falta de espaços no texto. Desta experiência não foram extraídas variáveis.

Foram também apuradas as classes gramaticais dos *tokens* dos textos. Recorrendo a POS-*tagging*, foram identificadas classes, como por exemplo, Nomes, Verbos, Símbolos, Pontuação, entre outras. Seguidamente foram criadas variáveis que consistem na contagem de *tokens* de cada uma destas classes presentes nos textos.

Outra variável gerada foi o número de *missing values* por projeto. Esta experiência foi feita acreditando que o número de campos deixados por preencher no formulário de candidatura podem dizer algo em relação ao comportamento dos projetos durante a sua duração, nomeadamente acerca da sua eventual anulação.

Foram extraídas variáveis através do campo relativo ao email do promotor da candidatura. Foram geradas variáveis binárias que assinalam se o email em causa tem domínio ".pt", ".com" ou se o seu servidor de correio eletrónico é "live", "gmail", "hotmail", "sapo" ou outro distinto.

Depois de extraídas todas estas variáveis foram criados quatro grupos de variáveis de entrada para os modelos: "variáveis TF" com 3402 variáveis; "variáveis TF-IDF" com 3403 variáveis; "outras variáveis" (designação atribuída a todas as restantes variáveis que não pertencem às categorias de valores TF ou TF-IDF) com 46 variáveis; "todas as variáveis" (grupo que aglomera os restantes grupos) com 6852 variáveis.

As variáveis do grupo "outras variáveis" e as variáveis do grupo "todas as variáveis" foram normalizadas através da função "*normalize*" da biblioteca de funções para Python, ScikitLearn (Pedrosa et al., 2011). Esta etapa teve como objetivo uniformizar a escala de apresentação das variáveis, de forma a evitar que os algoritmos de classificação sejam perturbados pelas diferentes escalas dos valores, isto pois alguns algoritmos são bastante sensíveis a esta característica dos dados. Os dados foram normalizados e não estandardizados pois o método posteriormente definido para a seleção de variáveis não aceita os valores negativos que seriam originados através da utilização do método de estandardização.

4.3. Seleção de variáveis

Antes de proceder à fase de classificação foi necessário passar pela etapa de seleção de variáveis.

Foram filtradas variáveis na extração de variáveis TF e TF-IDF, através das funções da biblioteca de funções para Python, ScikitLearn (Pedrosa et al., 2011), TfidfVectorizer e CountVectorizer. Estas funções permitem aquando da extração de variáveis, definir filtros. Foram definidos filtros relativos à frequência de ocorrência dos *tokens* nos vários documentos (DF, *document frequency*), definiu-se um DF mínimo de 5% e um DF máximo de 95%. Ou seja, foram apenas selecionadas palavras que ocorrem em pelo menos 5% dos documentos e não em mais do que 95% dos documentos.

Foi ainda testada como forma de seleção de variáveis um método *wrapper*, a função RFE (*recursive feature elimination*), da biblioteca de funções para Python, ScikitLearn (Pedrosa et al., 2011). No entanto, os resultados alcançados com esta seleção de variáveis não correram como esperado, pois acabaram por gerar um pior desempenho dos modelos. Outra característica que se destacou acerca deste método foi o facto de ser muito custoso em termos de duração. Daí ter sido descartada a utilização da função RFE.

Outra abordagem utilizada foi a utilização de métodos *filter* como forma de selecionar variáveis. Utilizou-se a função SelectKBest, da biblioteca de funções para Python, ScikitLearn (Pedrosa et al., 2011). Esta função permite escolher as K variáveis estatisticamente mais relevantes à luz de uma métrica escolhida. Foram utilizadas duas métricas, Chi2 e Mutual Information. Este processo foi aplicado aos grupos de variáveis, "variáveis TF", "variáveis TF-IDF" e "todas as variáveis", o grupo "outras variáveis" não entrou nesta etapa de processamento dos dados pelo facto de conter um número bastante mais reduzido de variáveis em comparação com os restantes. O valor K foi definido como metade da dimensão do grupo de variáveis ao qual o processo de seleção de variáveis fosse aplicado. Para os três grupos de variáveis cada uma das métricas elegeu metade variáveis. Foram posteriormente consideradas como variáveis selecionadas para a classificação apenas aquelas elegidas por ambas as métricas. Os grupos de variáveis ficaram então reduzidos para menores dimensões: "variáveis TF" com 914 variáveis; "variáveis TF-IDF" com 901 variáveis; "todas as variáveis" com 1712 variáveis.

4.4. Classificação

Nesta secção, correspondente à etapa "Modelação" da metodologia CRISP-DM, são explicadas quais as experiências de classificação desenvolvidas neste projeto e como se processou a sua elaboração.

Chegou a ser desenvolvida uma experiência de predição da autoria dos textos, associando os textos de cada candidatura à consultora do projeto, tornando a consultora do projeto na variável-alvo a predizer. O objetivo seria então investigar a possibilidade de predizer qual a consultora associada a determinado projeto apenas pelo texto da candidatura. Esta experiência chegou a ser trabalhada, no entanto, foi interrompida pois não vinha a demonstrar bons resultados.

A classificação levada a cabo e correspondente passo final da cadeia de classificação de texto desenvolvida nesta dissertação consiste na predição de anulação de projetos referentes programas de incentivos empresarias.

Como já referido, para classificar os projetos anulados utilizou-se a variável-alvo denominado "projeto anulado".

Optou-se por não balancear a variável-alvo. Esta decisão provém do facto de não querer introduzir dados artificiais na predição através de técnicas de *oversampling,* nem querer perder dados através de balanceamento *undersampling,* visto que o conjunto de projeto para os quais existe informação acerca da variável-alvo é já algo reduzido. Além do mais, os dados apresentam uma distribuição em relação à variável-alvo de 42% de projetos "Anulados" e 58% de projetos "Não anulados", o que não representa um desequilíbrio considerável.

Os dados foram divididos em conjunto de treino e conjunto de teste, numa proporção 80/20. Foi ainda utilizada a técnica de validação cruzada, *k-fold cross validation*, com k=5, nos dados do conjunto de treino, com o intuito de otimizar a parametrização dos modelos sem recurso aos dados de teste.

Os algoritmos de *machine learning* utilizados para a classificação foram LR, NB, RF, SVM e XGBoost. Foram também utilizados algoritmos de *deep learning*, redes neuronais, especificamente, CNN (redes neuronais convolucionais).

Em relação à experiência de predição com redes neuronais, esta foi desenvolvida com recurso à biblioteca de funções Keras (Chollet, 2015). Foram desenvolvidas duas redes, uma rede neuronal, que serve como *baseline,* que utiliza como variáveis de entrada os valores de *term frequency* dos textos (valores TF) e outra rede CNN que utiliza uma

camada de *embeddings* para gerar as variáveis de entrada da mesma. A rede CNN utiliza *embeddings* treinados pela própria rede, não foram utilizados *embeddings* pré-treinados pois todos os testados abrangiam apenas uma baixa percentagem do vocabulário presente nos textos. Ambas as redes foram construídas com recurso ao modelo sequencial suportado pela biblioteca de funções Keras (Chollet, 2015). Foi utilizada a função GridSearchCV de forma a otimizar os hiper-parâmetros das redes construídas e para aplicar o método de *k-fold cross-validation* com k=5.

A rede construída e otimizada para as variáveis constituídas pelos valores de *term frequency* dos textos é constituída por uma camada de input, três *hidden layers* de 30, 10 e 5 nodos respetivamente e uma camada de output. As *hidden layers* são da categoria de camadas "Dense", utilizam como "ReLU" como função de ativação enquanto a camada de output utiliza a função "Sigmoid". Foram otimizados os seguintes hiper-parâmetros: *epochs*; *batch size*; *optimizer*. Como função de perda foi utilizada "*Binary crossentropy*".

A rede CNN tem como camada inicial uma camada de *embeddings*, de seguida uma camada designada por "Conv1D" com função de ativação "ReLU", sendo esta a camada convolucional, seguida pela camada "GlobalMaxPooling1D" cujo propósito é diminuir a dimensão dos vetores de variáveis e por fim uma camada "Dense" com 10 nodos com função de ativação "ReLU" e a camada de output que utiliza a função de ativação "Sigmoid". Os parâmetros relativos ao número de filtros e à dimensão do *kernel* necessários à camada convolucional foram otimizados conjuntamente com os restantes hiper-parâmetros. Foram otimizados os seguintes hiper-parâmetros: *epochs; batch size*; *optimizer*; número de filtros (Conv1D); dimensão do *kernel* (Conv1D); dimensão de embedding; dimensão de vocabulário.

No caso da rede CNN foi necessário mais algum pré-processamento dos dados. Procedeu-se a uma indexação dos tokens presentes nos textos de forma a vetorizar o corpus de texto numa lista de números inteiros. Ou seja, cada *token* passa a corresponder a um número inteiro, fica codificado com uma chave num dicionário. Esta indexação é ordenada de forma decrescente pela frequência dos *tokens* no corpus. O número zero não fica adjudicado a nenhuma palavra. Após este passo é necessário normalizar a dimensão dos textos, utilizou-se o método *padding*, este que consiste em definir uma para os textos e restringir a essa dimensão os textos maiores que a mesma e aumentar a dimensão dos textos menores que a dimensão escolhida acrescentando zeros, que não correspondem a nenhuma palavra. A dimensão escolhida foi 10.000.

Todos os algoritmos foram de seguida otimizados para cada um dos conjuntos de variáveis recorrendo à função GridSearchCV (da biblioteca de funções para Python, Scikit-learn). Na Tabela 3 estão expostos parâmetros otimizados e os valores testados para cada parâmetro para cada algoritmo.

Tabela 3 - Parâmetros otimizados para cada algoritmo

Algoritmo	Parâmetros testados
Regressão logística	C - [0,001, 0,01, 0,1, 1, 10, 100] max_iter - [100, 500] penalty - [none, elasticnet, l1, l2] solver - [newton-cg, lbfgs, liblinear, sag, saga]
Classificador Naive Bayes	var_smoothing - [conjunto de 100 números intervalados numa escala logaritmica]
Floresta aleatória (Random Forest)	bootstrap - [True, False] max_depth - [10, 25, 50, 75, 100, None] max_features - [auto, sqrt] min_samples_leaf - [1, 2, 4] min_samples_split - [2, 5, 10] n_estimators - [200, 500, 1000, 1500, 2000]
Máquina de vetores de suporte (SVM)	C - [0.1, 1, 10, 100] gamma - [1, 0,1, 0,01, 0,001] kernel - [linear, rbf, poly, sigmoid] max_iter - [50000, 100000, 150000, 500000]
Xgboost	booster - [gbtree, gblinear, dar+A11:B22t] scale_pos_weight - [3,2] eta - [0,001, 0,01, 0,1, 0,3] n_estimators - [100, 500, 1000]
Rede Neuronal (NN)	batch_size - [5, 10, 20, 30] epochs - [5, 10, 20]
Rede Neuronal Convolucional (CNN)	num_filters - [32, 64, 128] kernel_size - [5] vocab_size - [10000] embedding_dim - [50, 100, 200] batch_size - [10, 20, 50] epochs - [10, 15, 20, 30] optimizer - [RMSprop, Adam, Adamax, sgd]

Foram posteriormente calculadas as importâncias das variáveis para a predição. Alguns modelos geram estes valores automaticamente, outros não. Para os modelos em que esse não é o caso foi necessário calcular a importância das variáveis com recurso à função "permutation_importance" (da biblioteca de funções para Python, Scikit-learn). Esta função funciona da seguinte forma: através de uma métrica a definir, a função avalia a evolução do erro de predição do modelo, caso uma variável não esteja disponível e depois compara esta avaliação com o erro de predição caso a variável esteja disponível; a função aplica este procedimento a todas as variáveis, as variáveis em que se verificar uma diferença maior entre os valores comparados são as variáveis mais importantes para o modelo segundo este método. A métrica escolhida foi a taxa de acerto.

4.5. Métricas

As métricas de avaliação de resultados utilizadas foram as métricas *standart* provenientes da matriz de confusão: taxa de acerto, precisão, cobertura e *F1-score*.

Esta página foi intensionalmente deixada em branco

Capítulo 5- Análise de resultados e discussão

Esta secção apresenta os resultados do trabalho executado durante esta dissertação. São caracterizados os produtos das experiências de extração de variáveis produzidas e de seguida são apresentados os resultados de predição da variável-alvo "Projeto anulado". Estes constituem o produto final da cadeia de classificação de texto desenvolvida nesta dissertação.

5.1. Resultados das experiências de extração de variáveis

Foram extraídas variáveis através das seguintes experiências efetuadas: criação de um modelo de tópicos; análise de vocabulário específico, análise da similaridade textual entre projetos; análise de diversidade lexical; análise da dimensão de palavras, frases e textos; estudo da utilização de pontuação e palavras especificas; extração de entidades nomeadas; análise gramatical dos textos; observação dos textos de candidatura em falta; análise de endereço de email.

Seguidamente são expostos os resultados provenientes das experiências considerados relevantes. São abordadas as experiências: criação de um modelo de tópicos, análise de vocabulário específico e análise da similaridade textual entre projetos.

5.1.1. Criação de um modelo de tópicos

Dos 14 tópicos nos textos de candidatura. Cada tópico é definido pela ocorrência em certa medida de várias palavras nos textos, consoante a ocorrência de palavras verificada, é atribuído a cada texto um valor que representa a preponderância de cada tópico no mesmo. A partir das palavras mais e menos relevantes para cada tópico segundo o algoritmo LDA, conseguimos interpretar a identidade de cada tópico e atribuir-lhe um rótulo.

Os seguintes encontrados são consideravelmente genéricos, alguns exemplos são: equipamento, inovação, internacionalização, serviços, aumento de capacidade e necessidades. No que diz respeito à variável-alvo, nenhuma das classes apresenta uma relação distinguível com tópicos específicos. Não obstante, as preponderâncias de cada tópico em cada projeto foram extraídas como variáveis, pois nem sempre a interpretação dos algoritmos de *machine learning* correspondem com às perceções humanas de uma mesma variável.

5.1.2. Vocabulário específico

As palavras que melhor caracterizam o conjunto de projetos anulados encontram- se na Tabela 4.

Tabela 4 - Palavras que melhor caracterizam os projetos de acordo com as variáveis-alvo

Vocabulário característico – Projetos anulados
acabamento
candidatura
conceção
digital
efetuar
engenharia
entidade
método
montagem
precisão

Como já referido, cada uma destas palavras originou a extração de uma variável binária que representa a presença ou ausência da palavra nos textos de candidatura dos projetos.

5.1.3. Análise da similaridade textual entre projetos

Nesta experiência, cada um dos projetos disponíveis foi comparado através da sua informação textual com todos os restantes projetos. Desta análise resultaram valores de similaridade entre todos os projetos disponíveis. Foram geradas variáveis descritivas das relações entre projetos, nomeadamente, o valor máximo de similaridade apresentado entre cada projeto e os restantes e a média de similaridade de cada projeto com os restantes projetos segmentados pela variável alvo "projeto anulado".

Esta experiência associa um valor entre 0 e 1 (sendo 0 o valor mínimo possível, significando total dissemelhança e sendo 1 o valor máximo possível, que indica completa correspondência entre textos), a pares de projetos. O valor mínimo verificado de

similaridade entre textos de diferentes projetos é 0 e o valor máximo verificado é 1. Estes valores mostram-nos que existem projetos com textos de candidatura totalmente dispares e também existem textos de candidatura idênticos.

A experiência focada na similaridade entre textos revelou elevados valores de similaridade entre vários projetos. De entre os 2793 projetos, 443 apresentam valores de similaridade acima de 0,9, entre si. Destes 443, 143 fazem parte do grupo de 781 projetos para os quais existe informação acerca do seu estado de anulação (variável-alvo).

Foram analisados pares de projetos constituídos por cada um destes projetos e o projeto com que cada uma destes apresenta a sua máxima similaridade.

Nos dados disponíveis existem 120 pares de projetos que apresentam valor de similaridade textual acima de 0,9, em que pelo menos para um dos projetos existe informação acerca do seu estado de anulação. Destes 120 projetos, 80 têm a mesma empresa como promotora. Este facto não parece ser especialmente relevante pelo facto de que muitos dos campos textuais da candidatura das empresas passam pela descrição da própria empresa promotora. Logo não se trata de uma anormalidade que os textos elaborados pela mesma empresa promotora para diferentes projetos (quando agregados num único texto) sejam semelhantes.

Se excluirmos os pares de projetos que pertencem à mesma empresa promotora, continuam a existir 40 pares de projetos com textos de candidatura com valores de similaridade textual muito elevados.

Ao comparar os elementos de cada par de projetos, os resultados que se destacam indicam que: em 21 destes o distrito da empresa promotora é o mesmo; em 29 pares, os projetos estão associados à mesma consultora; em 21 verifica-se que nenhum dos dois projetos foi anulado e em 19 pares verifica-se que apenas um foi anulado.

Comparam-se de seguida duas distribuições: a distribuição da totalidade de projetos disponíveis por distrito; e a distribuição de projetos com elevados níveis de similaridade com outro projeto, sendo os projetos promovidos por empresas diferentes, por distrito da empresa promotora. A Figura 11 assinala as diferenças apuradas.

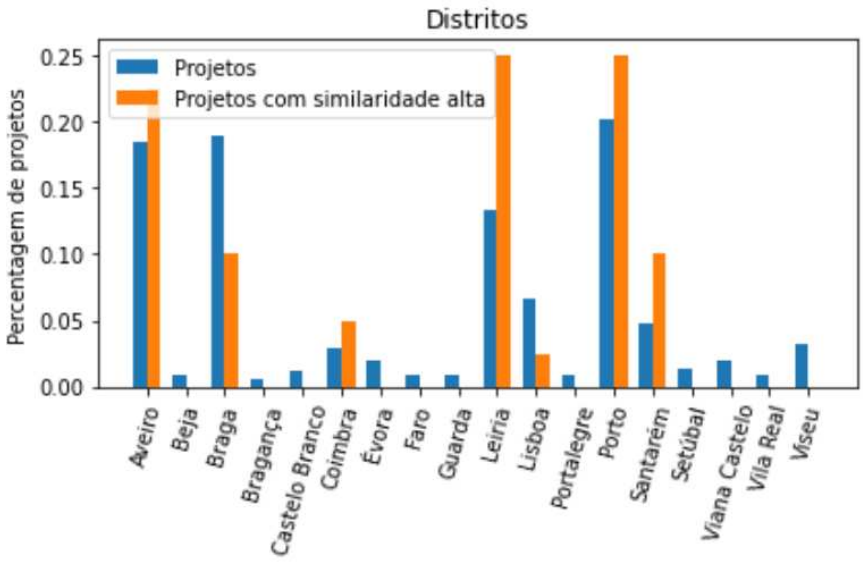


Figura 11 - Distribuições de grupos de projetos por distritos

Verificam-se diferenças na distribuição por distrito entre o grupo selecionado de projetos com similaridade elevada e a distribuição do grupo total de projetos mostrando uma maior concentração de projetos com alta similaridade de empresas promotoras distintas nos distritos de Aveiro, Coimbra, Leiria com especial intensidade, Porto e Santarém.

5.2. Resultados das experiências de previsão de anulações

Nas tabelas seguintes, cada uma relativa a um dos algoritmos de predição utilizados, é possível verificar os valores das métricas de avaliação alcançados pelos algoritmos de classificação para cada um dos conjuntos de variáveis definidos. As tabelas mostram também a combinação de parâmetros que atingiu melhores resultados na predição do conjunto de validação. Os resultados exibidos são relativos à aplicação dos algoritmos otimizados ao conjunto de teste.

Observa-se em primeiro lugar a Tabela 5 correspondente ao algoritmo Regressão Logística.

Tabela 5 - Quadro resumo de resultados de classificação do algoritmo Regressão Logística

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste: Taxa de acerto	Precisão	Cobertura	F1-score
Regressão logística	TF	C - 0,001 max_iter - 500 penalty - l2 solver - lbfgs	0,66	0,62	0,52	0,56
	TF-IDF	C - 100 max_iter - 500 penalty - l2 solver - liblinear	0,75	0,71	0,68	0,7
	Outras	C - 0,001 max_iter - 500 penalty - none solver - newton-cg	0,65	0,62	0,44	0,51
	Todas	C - 10 max_iter - 500 penalty - l2 solver - newton-cg	0,72	0,7	0,59	0,64

Este algoritmo obteve como melhores resultados valores nas métricas entre 0,68 e 0,75, utilizando como grupo de variáveis as "Variáveis TF-IDF". Os segundos melhores resultados foram alcançados pela utilização do grupo de variáveis "Todas as variáveis".

Tabela 6 - Quadro resumo de resultados de classificação do algoritmo Classificador Naïve Bayes

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste: Taxa de acerto	Precisão	Cobertura	F1-score
Classificador Naive Bayes	TF	var_smoothing - 8,111308307896873e-08	0,76	0,68	0,79	0,73
	TF-IDF	var_smoothing - 1,873817422860383e-08	0,66	0,62	0,5	0,55
	Outras	var_smoothing - 2,848035868435799e-07	0,61	0,57	0,32	0,41
	Todas	var_smoothing - 0,0002310129700083158	0,79	0,77	0,71	0,74

Tabela 7 - Quadro resumo de resultados de classificação do algoritmo Floresta Aleatória

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste: Taxa de acerto	Precisão	Cobertura	F1-score
Floresta aleatória (Random forest)	TF	bootstrap - False max_depth - 25 max_features - auto min_samples_leaf - 2 min_samples_split - 2 n_estimators - 500	0,7	0,69	0,51	0,59
	TF-IDF	bootstrap - True max_depth - 50 max_features - auto min_samples_leaf - 1 min_samples_split - 2 n_estimators - 200	0,75	0,85	0,5	0,63
	Outras	bootstrap - True max_depth - 10 max_features - auto min_samples_leaf - 1 min_samples_split - 5 n_estimators - 500	0,61	0,55	0,36	0,44
	Todas	bootstrap - False max_depth - 25 max_features - auto min_samples_leaf - 1 min_samples_split - 5 n_estimators - 1000	0,77	0,83	0,58	0,68

Tabela 8 - Quadro resumo de resultados de classificação do algoritmo SVM

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste: Taxa de acerto	Precisão	Cobertura	F1-score
Máquina de vetores de suporte (SVM)	TF	C - 1 gamma - 1 kernel - linear max_iter - 500000	0,68	0,62	0,59	0,6
	TF-IDF	C - 1 gamma - 0,001 kernel - sigmoid max_iter - 50000	0,57	0,25	0,02	0,03
	Outras	C - 0.1 gamma - 1 kernel - linear max_iter - 50000	0,59	0	0	0
	Todas	C - 10 gamma - 0,1 kernel - rbf max_iter - 50000	0,72	0,72	0,55	0,62

As Tabelas 5, 6 e 7 apresentam os resultados de predição dos algoritmos Classificador Naïve Bayes, Floresta Aleatória e SVM, respetivamente, tal como os parâmetros utilizados. Nos três quadros pode notar-se o facto de os melhores resultados para as quatro métricas serem fruto da utilização do conjunto de variáveis "Todas as variáveis". Esta

circunstância mostra que estes algoritmos conseguem tirar melhor proveito de um conjunto de variáveis provenientes de diversos métodos de extração do que variáveis do que de conjuntos de variáveis de igual origem.

O Classificador Naïve Bayes obteve uma gama de resultados nas métricas provenientes da matriz de confusão entre 0,71 e 0,79. O grupo de variáveis que originou os segundos melhores resultados foi "variáveis TF". Os melhores resultados obtidos nas métricas pelo algoritmo Floresta Aleatória métricas compreendem-se entre 0,58 e 0,83. Os seus segundos melhores resultados nas métricas conseguidos por este algoritmo foram ao utilizar o grupo de variáveis "Variáveis TF-IDF". O algoritmo SVM teve os seus melhores resultados compreendidos entre 0,55 e 0,72. Sendo que o grupo de variáveis que proporcionou os melhores resultados foi o grupo de variáveis "Variáveis TF-IDF".

Tabela 9 - Quadro resumo de resultados de classificação do algoritmo XGBoost

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste:			
			Taxa de acerto	Precisão	Cobertura	F1-score
Xgboost	TF	booster - gblinear scale_pos_weight - 3,2 eta - 0,001 n_estimators - 1000	0,7	0,6	0,85	0,7
	TF-IDF	booster - gbtree scale_pos_weight - 3,2 eta - 0,1 n_estimators - 500	0,76	0,73	0,68	0,7
	Outras	booster - gbtree scale_pos_weight - 3,2 eta - 0,3 n_estimators - 100	0,61	0,53	0,61	0,57
	Todas	booster - gbtree scale_pos_weight - 3,2 eta - 0,3 n_estimators - 500	0,7	0,63	0,68	0,66

O algoritmo XGBoost alcançou os melhores resultados utilizando os grupos de variáveis relativos às técnicas de extração TF e TF-IDF. As gamas de resultados ficaram entre 0,60 e 0,85, e entre 0,68 e 0,76, respetivamente.

Tabela 10 - Quadro resumo de resultados de classificação do algoritmo NN

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste:			
			Taxa de acerto	Precisão	Cobertura	F1-score
Rede Neuronal (NN)	TF	batch_size - 30 epochs - 15	0,75	0,69	0,75	0,72

A rede neuronal testada atingiu bons resultados, estando os valores das métricas compreendidos entre 0,69 e 0,75. Esta rede mais simples que apenas utilizou o grupo de variáveis "Variáveis TF" foi utilizada como modelo *baseline,* com o objetivo de comparar os resultados com a rede CNN também experimentada.

Tabela 11 - Quadro resumo de resultados de classificação do algoritmo CNN

Algoritmo	Variáveis	Melhores parâmetros	Resultados para o conjunto de teste:			
			Taxa de acerto	Precisão	Cobertura	F1-score
Rede Neuronal Convolucional (CNN)	Embeddings	num_filters - 128 kernel_size - 5 vocab_size - 10000 embedding_dim - 200 batch_size - 50 epochs - 30 optimizer - RMSprop	0,78	0,86	0,61	0,71

A rede CNN, que utiliza como variáveis *embeddings* treinados pela própria rede, conseguiu suplantar a outra rede neuronal criada. Obteve uma gama de resultados compreendida entre 0,61 e 0,86. Este algoritmo conseguiu atingir resultados melhores do que a maioria dos outros algoritmos testados e ainda apresenta o benefício de gerar as variáveis que utiliza de forma autónoma.

Tendo por critério principal os resultados da métrica taxa de acerto, considera-se assim que os melhores resultados foram apresentados pelo algoritmo de aprendizagem supervisionada, Classificador Naïve Bayes, utilizando o conjunto de variáveis "Todas as variáveis", uma combinação de variáveis extraídas através de vários métodos distintos. O algoritmo atingiu 79% de taxa de acerto. Na Figura 12 podemos observar a matriz de confusão resultante da aplicação deste algoritmo a este conjunto de variáveis.

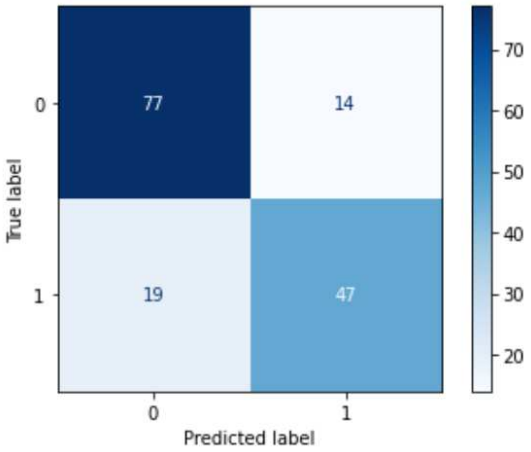


Figura 12- Matriz de confusão resultante da utilização do classificador Naïve Bayes e do conjunto de variáveis "Todas as variáveis"

Foram calculadas as variáveis mais importantes do modelo que atingiu melhores resultados nas métricas. Na Figura 13 podem-se observar as 10 variáveis mais relevantes na predição segundo o método "Importância de permutação de variáveis". O funcionamento do método em causa e o significado da escala de relevância das variáveis podem ser consultados no fim da secção "4.4 Classificação" do capítulo "4. Metodologia".

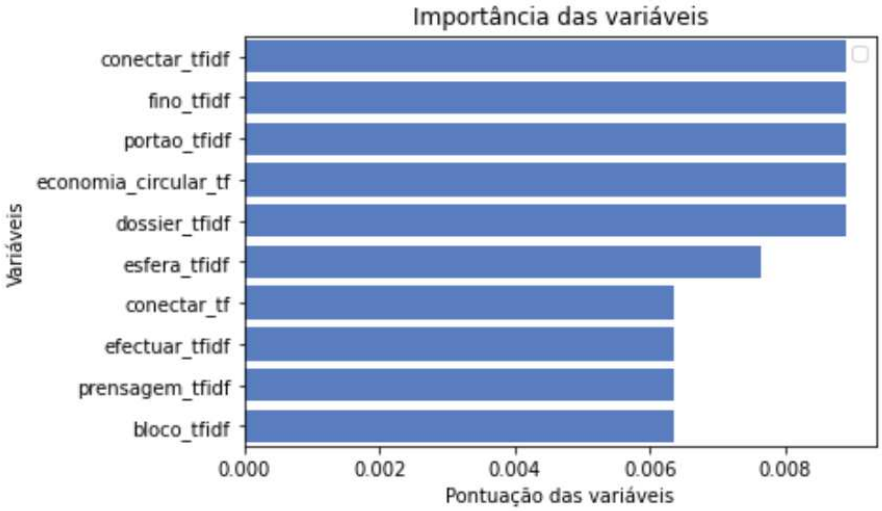


Figura 13 - Variáveis mais relevantes da predição

Algoritmo: Classificador Naïve Bayes; Variáveis: "Todas as Variáveis"

Através da observação da Figura 13 nota-se a maior importância das variáveis extraídas através de métodos mais convencionais, da categoria *bag-of-Words*, sendo que os valores TF-IDF são preponderantes.

Podemos entender por estes resultados que foi benéfico uma utilização mista das técnicas de extração de variáveis pelo método TF e TF-IDF. Apesar dos melhores resultados em vários algoritmos terem sido geradas pela utilização do grupo de variáveis "Todas as Variáveis", sabemos que este grupo é maioritariamente constituído por variáveis TF e TF-IDF e como podemos constatar foram estas que mostraram ter maior poder preditivo. O grupo de variáveis "Outras variáveis" conseguiu alcançar resultados que batessem os outros grupos de variáveis.

Conclusão

Nesta dissertação foram aplicados conhecimentos, técnicas e ferramentas da disciplina de *text mining* a textos provenientes de candidaturas de empresas a fundos públicos aprovadas. O objetivo foi extrair das mesmas informações relevantes para a predição da capacidade das empresas de concluir o projeto e retirar o aproveitamento esperado do mesmo. Ou seja, a predição de se o projeto seria anulado ou não durante a sua duração.

Foi criada uma cadeia de classificação de texto que transformou os dados textuais não estruturados em variáveis prontas a utilizar por algoritmos de *machine learning* e *deep learning*. Neste processo foram utilizadas técnicas de extração de variáveis convencionais como a extração de valores TF e TF-IDF, bem como foram desenvolvidas diversas outras experiências de extração de variáveis com a intenção de captar variadas caraterísticas presentes nos dados.

Neste trabalho é destacada uma certa preponderância de projetos com elevados níveis de similaridade (provenientes de empresas promotoras diferentes) em certos distritos do país. Este resultado demonstra o potencial de exploração que o text mining apresenta e o mostra o potencial de descoberta de informação que pode ser aproveitado futuramente.

Os melhores resultados quanto à predição de projetos anulados foram obtidos pelo conjunto de variáveis que mistura variáveis provenientes de diferentes estratégias, o conjunto designado por "Todas as variáveis", em conjunto com o Classificador Naïve Bayes. Esta combinação atingiu um resultado de 79% de taxa de acerto na predição da anulação de projetos de aplicação de fundos públicos. Conclui-se que nesta aplicação especifica de técnicas de mineração e classificação de texto foi proveitoso a mistura entre técnicas de extração de variáveis, com principal ênfase para as técnicas *bag-of-words*, TF e TF-IDF. Evidenciam-se também os resultados do algoritmo CNN que atingiram bons resultados utilizando como variáveis apenas *embeddings* treinados pela própria rede.

6.1. Contributos

Esta dissertação contribui para o desenvolvimento da aplicação de técnicas de inteligência artificial no setor público de forma prática.

Apresenta-se como um avanço necessário na direção de mitigar riscos na fase de análise de candidaturas de empresas a programas de incentivos públicos. Este trabalho parece tratar-se de uma primeira abordagem ao tema, sendo que não foram encontrados

trabalhos semelhantes na literatura, tendo em conta a combinação entre a abordagem utilizada e o objeto de estudo.

Esta dissertação vem também contribuir para a literatura ao demonstrar mais um exemplo dos resultados que a aplicação da mineração e análise de texto podem apresentar como solução de problemas reais. Contribui com ideias de extração de variáveis a partir de texto. Adicionalmente, fornece informação concreta e concisa acerca do conteúdo das candidaturas a fundos públicos em Portugal.

6.2. Limitações e recomendações de pesquisa futura

Para o futuro, esta dissertação deixa em aberto alguns pontos suscetíveis ao desenvolvimento de um trabalho mais aprofundado.

Uma proposta para a otimização dos resultados da cadeia de classificação de texto desenvolvida seria a otimização da etapa de pré-processamento. Ou seja, experimentar todas as combinações possíveis de técnicas de pré-processamento identificadas no capítulo "Enquadramento Conceptual", com o intuito de gerar melhores resultados nas métricas finais de avaliação de resultados.

A etapa de extração de variáveis deixa bastante espaço aberto para a experimentação de outros métodos de extração de variáveis. Por exemplo, poderia analisar-se com mais foco os que campos textuais que as empresas preenchem e os que não, de forma a perceber se as diferenças de preenchimento da candidatura são relevantes para a predição de anulação de projetos. Outro exemplo de trabalho futuro será a maior exploração da similaridade entre textos.

Quanto à etapa de classificação, poderiam também ser ensaiadas outras arquiteturas de redes neuronais, isto pois, as redes neuronais são algoritmos complexos com um elevado número de parâmetros que lhe dão forma, fator que permite explorar este algoritmo infinitamente. Ainda nesta etapa seria possível continuar a otimizar os algoritmos testados, alargando os testes a mais parâmetros dos mesmos.

A cadeia de classificação de texto gerada poderia também ser aplicada a outras variáveis-alvo. A predição da autoria dos textos de candidatura tendo como variável-alvo as consultoras dos projetos, por exemplo, foi ainda uma experiência iniciada nesta direção que acabou por não ser desenvolvida.

A variável-alvo utilizada contém apenas informação para 781 dos 2793 projetos dos quais existem dados disponíveis, o que limita as predições levadas a cabo.

Outra limitação identificada trata-se do facto de existirem 20 projetos classificados como "não anulados" que ainda se encontravam em curso à data do desenvolvimento deste trabalho. Este facto causar algumas alterações nas predições caso se verifique a anulação de algum destes projetos.

Referências bibliográficas

Abonizio, H. Q., de Morais, J. I., Tavares, G. M., & Junior, S. B. (2020). Language-independent fake news detection: English, Portuguese, and Spanish mutual features. Future Internet, 12(5). https://doi.org/10.3390/FI12050087

Achilonu, O. J., Olago, V., Singh, E., Eijkemans, R. M. J. C., Nimako, G., & Musenge, E. (2021). A text mining approach in the classification of free-text cancer pathology reports from the south african national health laboratory services. Information. An International Interdisciplinary Journal, 12(11). https://doi.org/10.3390/info12110451

Aggarwal, C. C., & Zhai, C. (2012). Mining Text Data. Springer Science & Business Media.

Agirre, Banea, Cer, & Diab. (2016). Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. SemEval-2016. 10th. https://repositori.upf.edu/handle/10230/33534

Ali, M., Yasmine, F., Mushtaq, H., Sarwar, A., Idrees, A., Tabassum, S., BaburHayyat, D., & Rehman, K. U. (2021). Customer Opinion Mining by Comments Classification using Machine Learning. International Journal of Advanced Computer Science and Applications, 12(5), 385–393.

Alsaidi, S. A., Sadiq, A. T., & Abdullah, H. S. (2020). English poems categorization using text mining and rough set theory. Bulletin of Electrical Engineering and Informatics, 9(4), 1701–1710.

Amazal, H., & Kissi, M. (2021). A New Big Data Feature Selection Approach for Text Classification. Scientific Programming, 2021. https://doi.org/10.1155/2021/6645345

Amer, A. A., & Abdalla, H. I. (2020). A set theory based similarity measure for text clustering and classification. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00344-3

Bansal, H., Shrivastava, G., Nguyen, G. N., & Stanciu, L.-M. (2018). Social Network Analytics for Contemporary Business Organizations. IGI Global.

Bates, J., Fodeh, S. J., Brandt, C. A., & Womack, J. A. (2016). Classification of radiology reports for falls in an hiv study cohort. Journal of the American Medical Informatics Association: JAMIA, 23(e1), e113–e117.

Bayrak, S., Yucel, E., & Takci, H. (2022). Epilepsy radiology reports classification using deep learning networks. Computers, Materials and Continua, 70(2), 3589–3607.

Bikku, T., Nandam, S. R., & Akepogu, A. R. (2018). A contemporary feature selection and classification framework for imbalanced biomedical datasets. Egyptian Informatics Journal, 19(3), 191–198.

Borges, L., Martins, B., & Calado, P. (2019). Combining similarity features and deep representation learning for stance detection in the context of checking fake news. Journal of Data and Information Quality, 11(3). https://doi.org/10.1145/3287763

Bruni, R., & Bianchi, G. (2020). Website categorization: A formal approach and robustness analysis in the case of e-commerce detection. Expert systems with applications, 142. https://doi.org/10.1016/j.eswa.2019.113001

Caccamisi, A., Jørgensen, L., Dalianis, H., & Rosenlund, M. (2020). Natural language processing and machine learning to enable automatic extraction and classification of patients' smoking status from electronic medical records. Upsala journal of medical sciences, 316–324.

Chai, K. E. K., Anthony, S., Coiera, E., & Magrabi, F. (2013). Using statistical text classification to identify health information technology incidents. Journal of the American Medical Informatics Association: JAMIA, 20(5), 980–985.

Chapman, P. (2000). CRISP-DM 1.0: Step-by-step Data Mining Guide. SPSS.

Conway, M., Doan, S., Kawazoe, A., & Collier, N. (2009). Classifying disease outbreak reports using n-grams and semantic features. International journal of medical informatics, 78(12). https://doi.org/10.1016/j.ijmedinf.2009.03.010

Chollet, F. (2015). GitHub - keras-team/keras: Deep Learning for humans. GitHub. https://github.com/keras-team/keras

David M. Blei, Andrew Y. Ng, & Michael I. Jordan. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022. https://doi.org/10.5555/944919.944937

Dengre, V., Kheruwala, H. A., & Shah, M. (2020). Comprehensive review of text-mining applications in finance. Financial Innovation, 6(1). https://doi.org/10.1186/s40854-020-00205-1

Dixit, A., Mani, A., & Bansal, R. (2020). Feature selection for text and image data using differential evolution with SVM and Naïve Bayes classifiers. Engineering Journal, 24(5), 161–172.

Du, J., Chen, Q., Peng, Y., Xiang, Y., Tao, C., & Lu, Z. (2019). ML-Net: Multi-label classification of biomedical texts with deep neural networks. Journal of the American Medical Informatics Association: JAMIA, 26(11), 1279–1285.

Durán, P., Malvern, D., Richards, B., & Chipere, N. (2004). Developmental Trends in Lexical Diversity. Applied Linguistics, 25(2), 220–242.

Elhadad, M. K., Li, K. F., & Gebali, F. (2020). Detecting misleading information on COVID-19. IEEE Access, 8, 165201–165215.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, & Edouard Duchesnay. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), 2825–2830.

Fernandes, M., Sun, H., Jain, A., Alabsi, H. S., Brenner, L. N., Ye, E., Ge, W., Collens, S. I., Leone, M. J., Das, S., Mukerji, S. S., & Brandon Westover, M. (2021). Classification of the disposition of patients hospitalized with COVID-19: Reading discharge summaries using natural language processing. JMIR Medical Informatics, 9(2). https://doi.org/10.2196/25457

Forman, G. (2002). Choose your words carefully: An empirical study of feature selection metrics for text classification: Vol. 2431 LNAI (pp. 150–162).

Giovanelli, C., Liu, X., Sierla, S., Vyatkin, V., & Ichise, R. (2017). Towards an aggregator that exploits big data to bid on frequency containment reserve market. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, 7514–7519.

Guerin, A. (2016). Using Demographic Variables and In-College Attributes to Predict Course-Level Retention for Community College Spanish Students. Em Northcentral University ProQuest Dissertations Publishing (Vol. 10125057). https://search.proquest.com/openview/b7279da30cc1bcdd1205e578dea11ce8/1?pq-origsite=gscholar&cbl=18750

Guo, G. (2014). Soft Biometrics from Face Images Using Support Vector Machines. Em Y. Ma & G. Guo (Eds.), Support Vector Machines Applications (pp. 269–302). Springer International Publishing.

Huang, K. (2015). Unconstrained smartphone sensing and empirical study for sleep monitoring and self-management. Em University of Massachusetts Lowell ProQuest Dissertations Publishing (Vol. 3663970).

https://search.proquest.com/openview/e635fbb48407def4388a551f01995128/1?pq-origsite=gscholar&cbl=18750

IAPMEI e AICEP aplicam AI na gestão de incentivos Portugal 2020. (2019, 11 de Novembro). IAPMEI. https://www.iapmei.pt/NOTICIAS/IAPMEI-e-AICEP-aplicam-projeto-de-AI-na-gestao-de.aspx

Iyer, R. R., & Rose, C. P. (2019). A Machine Learning Framework for Authorship Identification From Texts. Em arXiv [cs.CL]. arXiv. http://arxiv.org/abs/1912.10204

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading Text in the Wild with Convolutional Neural Networks. International journal of computer vision, 116(1), 1–20.

Jasim. (2015). Data mining approach and its application to dresses sales recommendation. Research Gate. https://www.researchgate.net/profile/Dalia-Jasim/publication/293464737_main_steps_for_doing_data_mining_project_using_weka/links/56b8782008ae44bb330d2583/main-steps-for-doing-data-mining-project-using-weka.pdf

Karamizadeh, S., Abdullah, S. M., Halimi, M., Shayan, J., & Rajabi, M. J. (2014). Advantage and drawback of support vector machine functionality. 2014 International Conference on Computer, Communications, and Control Technology (I4CT), 63–65.

Karen, S. J. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. Journal of Documentation, 28(1), 11–21.

Khachidze, M., Tsintsadze, M., & Archuadze, M. (2016). Natural Language Processing Based Instrument for Classification of Free Text Medical Records. BioMed research international, 2016. https://doi.org/10.1155/2016/8313454

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information. An International Interdisciplinary Journal, 10(4). https://doi.org/10.3390/info10040150

Kumar, V., Recupero, D. R., Riboni, D., & Helaoui, R. (2021). Ensembling Classical Machine Learning and Deep Learning Approaches for Morbidity Identification from Clinical Notes. IEEE Access, 9, 7107–7126.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, fevereiro 19). Recurrent Convolutional Neural Networks for Text Classification. Twenty-Ninth AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewPaper/9745

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Li, L., Weinberg, C. R., Darden, T. A., & Pedersen, L. G. (2001). Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics , 17(12), 1131–1142.

Ma, L., Ofoghi, B., Watters, P., & Brown, S. (2009). Detecting phishing emails using hybrid features. UIC-ATC 2009 - Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC'09 and ATC'09 Conferences, 493–497.

McCarthy, P. M., & Jarvis, S. (2010). MTLD, vocd-D, and HD-D: a validation study of sophisticated approaches to lexical diversity assessment. Behavior Research Methods, 42(2), 381–392.

Méndez, J. R., Cotos-Yañez, T. R., & Ruano-Ordás, D. (2019). A new semantic-based feature selection method for spam filtering. Applied Soft Computing Journal, 76, 89–104.

Morgan, J. N., & Sonquist, J. A. (1963). Problems in the Analysis of Survey Data, and a Proposal. Journal of the American Statistical Association, 58(302), 415–434.

Nair, & Hinton. (2010). Rectified linear units improve restricted boltzmann machines. Icml. https://openreview.net/forum?id=rkb15iZdZB

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, 22(2), 199–210.

Patel, D., & Srivastava, T. (2014). Ant Colony Optimization Model for Discrete Tomography Problems. Proceedings of the Third International Conference on Soft Computing for Problem Solving, 785–792.

Prusa, J. D., & Khoshgoftaar, T. M. (2017). Improving deep neural network design with new text data representations. Journal of Big Data, 4(1). https://doi.org/10.1186/s40537-017-0065-8

Putong, M. W., & Suharjito. (2020). Classification model of contact center customers emails using machine learning. Advances in Science, Technology and Engineering Systems, 5(1), 174–182.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. https://doi.org/10.18653/v1/2020.acl-demos.14

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81–106.

Rabbimov, I. M., & Kobilov, S. S. (2020). Multi-Class Text Classification of Uzbek News Articles using Machine Learning. Journal of Physics: Conference Series, 1546. https://doi.org/10.1088/1742-6596/1546/1/012097

Ranjan, Ghorpade, & Kanthale. (2017). Document classification using lstm neural network. Journal of data mining and digital humanities. https://core.ac.uk/download/pdf/230492600.pdf

Reddy, K. R., & Chaudhary, S. (2021). Research challenges in text mining and empirical research directions. Indian Journal of Computer Science and Engineering, 12(3), 752–764.

Rustam, F., Mehmood, A., Ahmad, M., Ullah, S., Khan, D. M., & Choi, G. S. (2020). Classification of Shopify App User Reviews Using Novel Multi Text Features. IEEE Access, 8, 30234–30244.

Sahgal, D., & Parida, M. (2014). Object Recognition Using Gabor Wavelet Features with Various Classification Techniques. Proceedings of the Third International Conference on Soft Computing for Problem Solving, 793–804.

Sahgal, D., & Ramesh, A. (2002). On Road Vehicle Detection Using Gabor Wavelet Features with Various Classification Techniques. Proceedings of the 14th International Conference. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.740.4429&rep=rep1&type=pdf

Sanjay, Nagori, & Sanjay. (2018). Comparing Existing Methods for Predicting the Detection of Possibilities of Blood Cancer by Analyzing Health Data. Int. J. Innov. Res. Sci. https://www.academia.edu/download/57618024/IJIRSTV4I9009_OK.pdf

Sikelis, K., Tsekouras, G. E., & Kotis, K. (2021). Ontology-based feature selection: A survey. Future Internet, 13(6). https://doi.org/10.3390/fi13060158

Skenderi, E., Huhtamäki, J., & Stefanidis, K. (2021). Multi-Keyword Classification: A Case Study in Finnish Social Sciences Data Archive. Information. An International Interdisciplinary Journal, 12(12). https://doi.org/10.3390/info12120491

Soheily-Khah, S., Marteau, P.-F., & Béchet, N. (2018). Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset. 2018 1st International Conference on Data Intelligence and Security (ICDIS), 219–226.

Steffey, D., Cox, D. R., & Snell, E. J. (1990). Analysis of binary data (2nd ed.). Journal of the American Statistical Association, 85(412), 1171.

Tang, D., Qin, B., & Liu, T. (2015). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 1422–1432.

Triantafyllou, I., Drivas, I. C., & Giannakopoulos, G. (2020). How to utilize my app reviews? A novel topics extraction machine learning schema for strategic business purposes. Entropy , 22(11), 1–21.

Verma, P. K., Agrawal, P., Amorim, I., & Prodan, R. (2021). WELFake: Word Embedding over Linguistic Features for Fake News Detection. IEEE Transactions on Computational Social Systems, 8(4), 881–893.

Wang, P., Morgan, A. A., Zhang, Q., Sette, A., & Peters, B. (2007). Automating document classification for the Immune Epitope Database. BMC bioinformatics, 8. https://doi.org/10.1186/1471-2105-8-269

Wang, Y., Khardon, R., & Protopapas, P. (2012). NONPARAMETRIC BAYESIAN ESTIMATION OF PERIODIC LIGHT CURVES. The Astrophysical Journal, 756(1), 67.

Wang, Y., Sohn, S., Liu, S., Shen, F., Wang, L., Atkinson, E. J., Amin, S., & Liu, H. (2019). A clinical text classification paradigm using weak supervision and deep

representation 08 Information and Computing Sciences 0801 Artificial Intelligence and Image Processing 17 Psychology and Cognitive Sciences 1702 Cognitive Sciences. BMC medical informatics and decision making, 19(1). https://doi.org/10.1186/s12911-018-0723-6

Wu, H., Liu, Y., & Wang, J. (2020). Review of text classification methods on deep learning. Computers, Materials and Continua, 63(3), 1309–1321.

Wu, Lin, & Weng. (2004). Probability estimates for multi-class classification by pairwise coupling. Advances in neural information processing systems. https://proceedings.neurips.cc/paper/2003/hash/03e7ef47cee6fa4ae7567394b99912b7-Abstract.html

Yang, H., Nenadic, G., & Keane, J. A. (2008). Identification of transcription factor contexts in literature using machine learning approaches. BMC bioinformatics, 9(SUPPL. 3). https://doi.org/10.1186/1471-2105-9-S3-S11

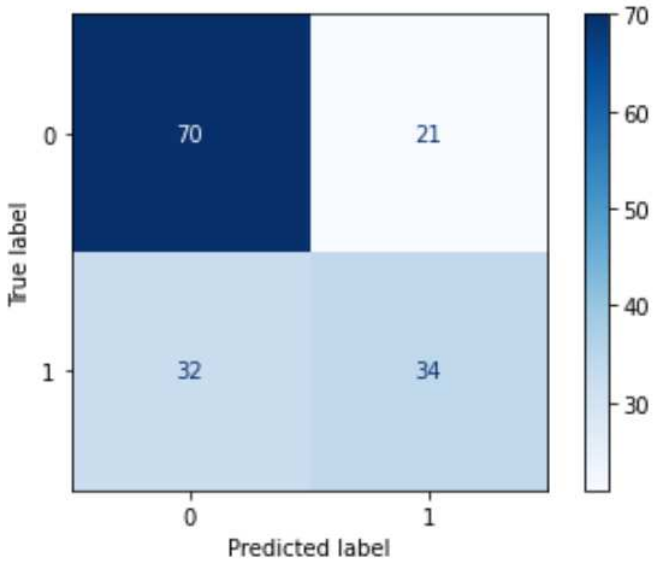
Zhang, D., & Lee, W. S. (2006). Extracting key-substring-group features for text classification. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, 474–483.

Zhang, Y., & Liu, B. (2007). Semantic text classification of emergent disease reports: Vol. 4702 LNAI (pp. 629–637).

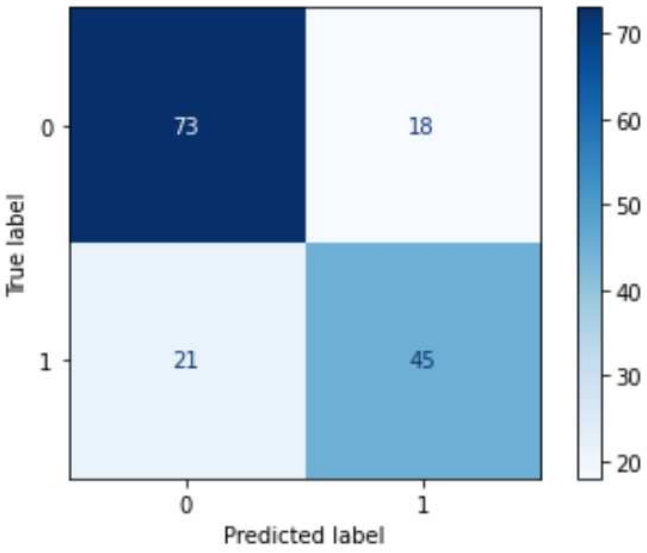
Esta página foi intensionalmente deixada em branco

Anexos

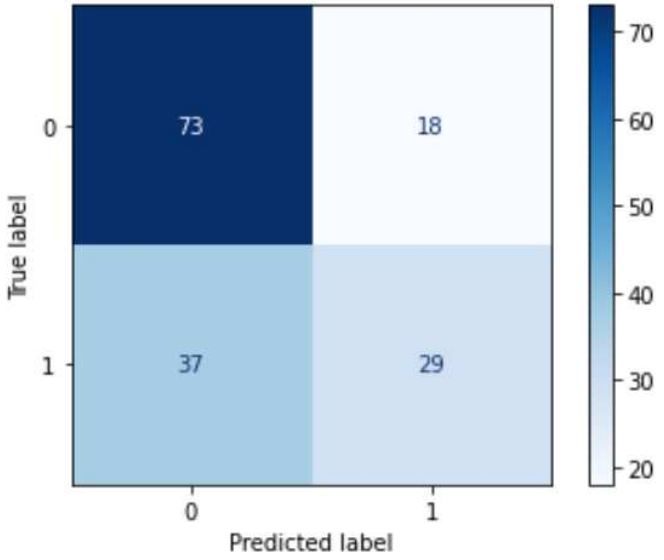
Anexo A. Resultados de classificação do algoritmo "Regressão Logística" para o conjunto de variáveis "TF", no formato de matriz de confusão



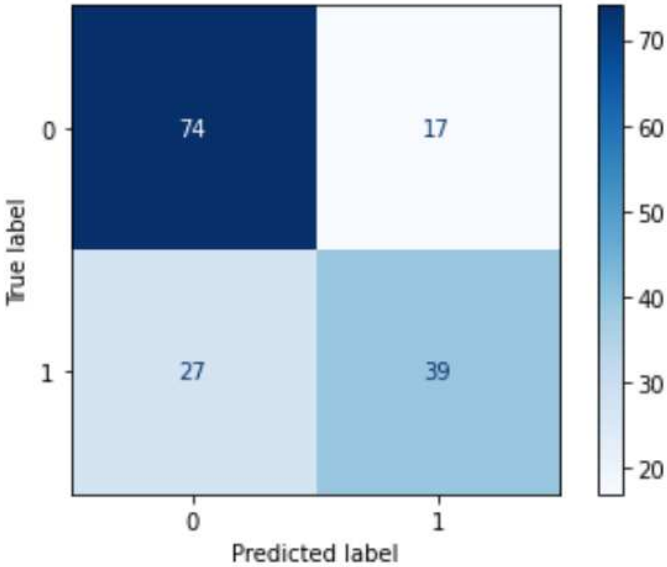
Anexo B. Resultados de classificação do algoritmo "Regressão Logística" para o conjunto de variáveis "TF-IDF", no formato de matriz de confusão



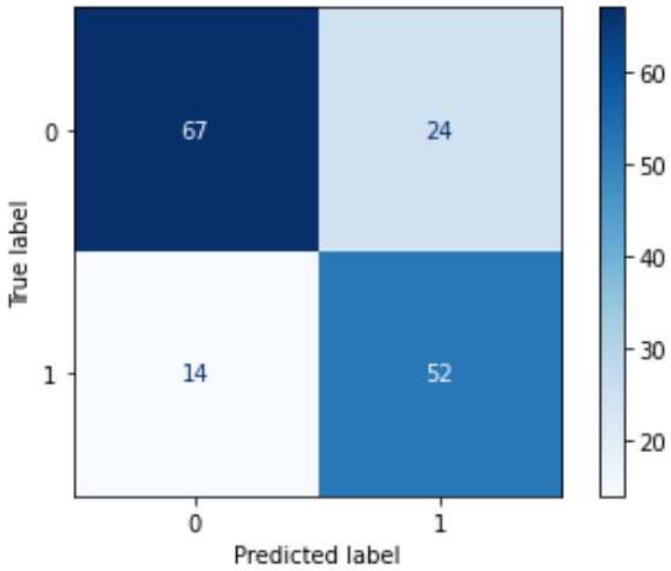
Anexo C. Resultados de classificação do algoritmo "Regressão Logística" para o conjunto de variáveis "Outras", no formato de matriz de confusão



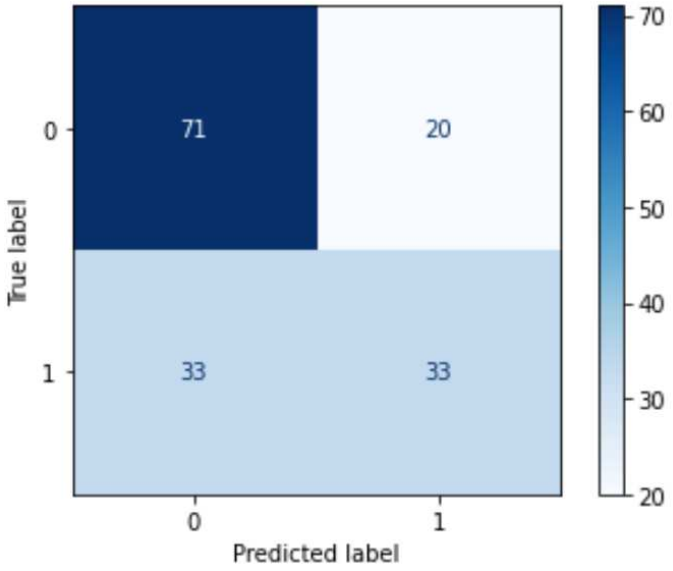
Anexo D. Resultados de classificação do algoritmo "Regressão Logística" para o conjunto de variáveis "Todas", no formato de matriz de confusão



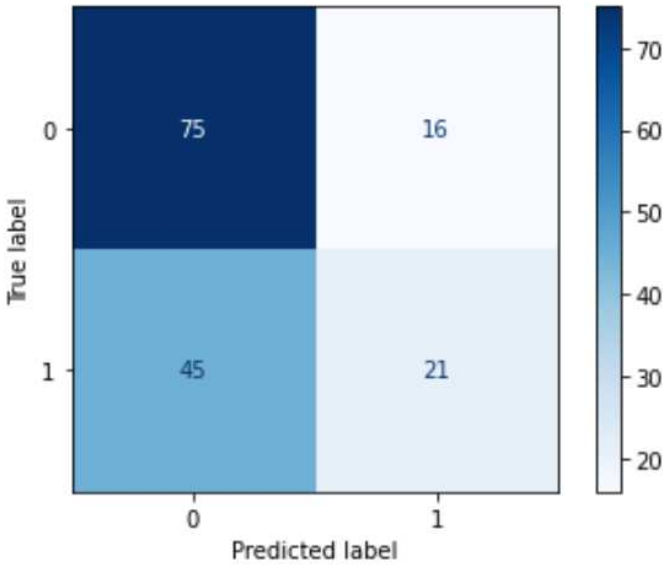
Anexo E. Resultados de classificação do algoritmo "Classificador Naïve Bayes" para o conjunto de variáveis "TF", no formato de matriz de confusão



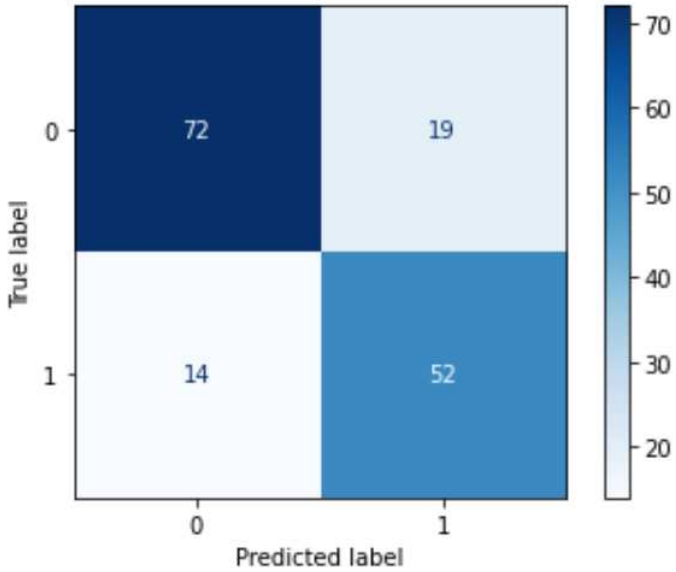
Anexo F. Resultados de classificação do algoritmo "Classificador Naïve Bayes" para o conjunto de variáveis "TF-IDF", no formato de matriz de confusão



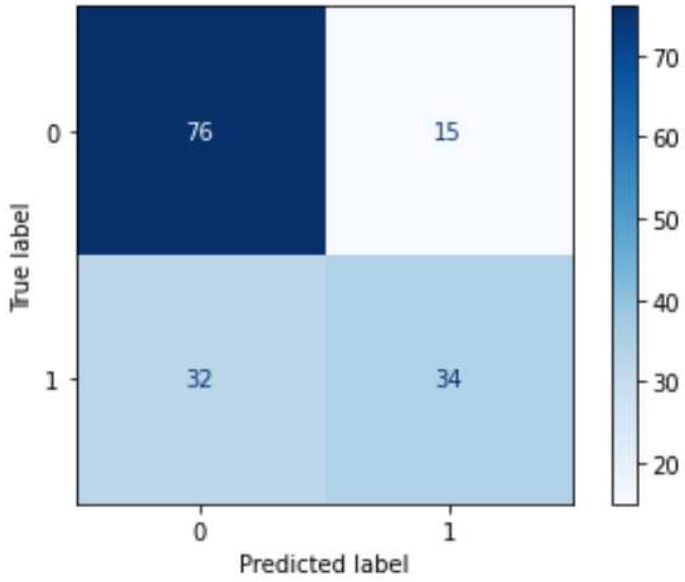
Anexo G. Resultados de classificação do algoritmo "Classificador Naïve Bayes" para o conjunto de variáveis "Outras", no formato de matriz de confusão



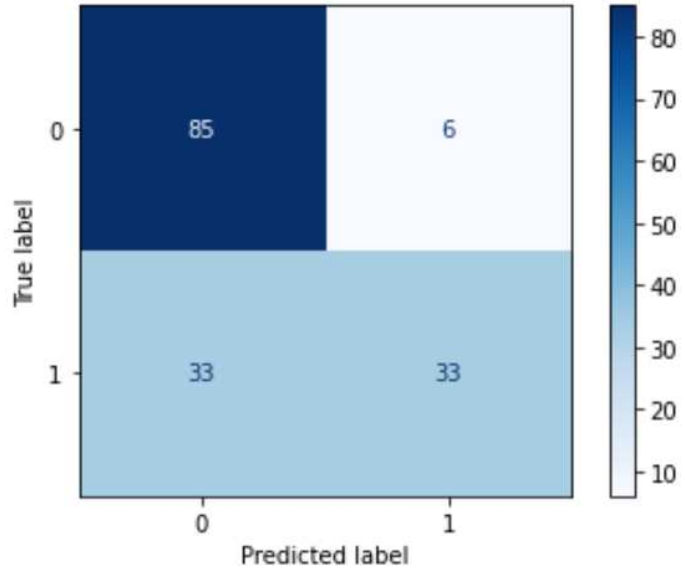
Anexo H. Resultados de classificação do algoritmo "Classificador Naïve Bayes" para o conjunto de variáveis "Todas", no formato de matriz de confusão



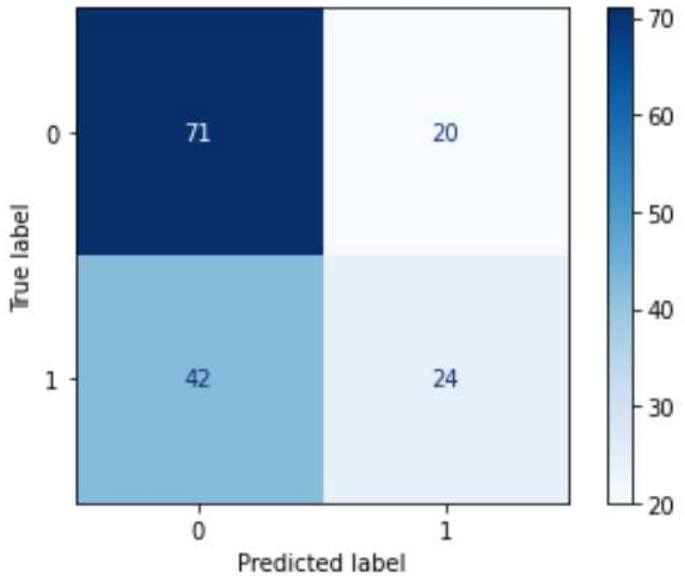
Anexo I. Resultados de classificação do algoritmo "Floresta Aleatória" para o conjunto de variáveis "TF", no formato de matriz de confusão



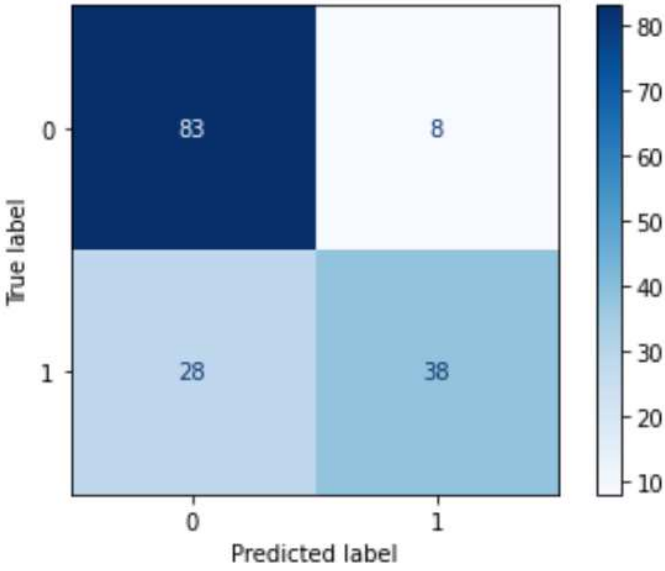
Anexo J. Resultados de classificação do algoritmo "Floresta Aleatória" para o conjunto de variáveis "TF-IDF", no formato de matriz de confusão



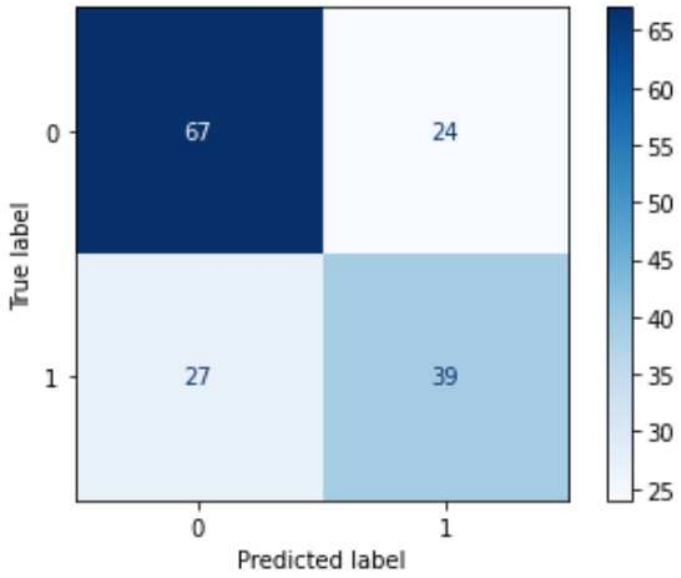
Anexo L. Resultados de classificação do algoritmo "Floresta Aleatória" para o conjunto de variáveis "Outras", no formato de matriz de confusão



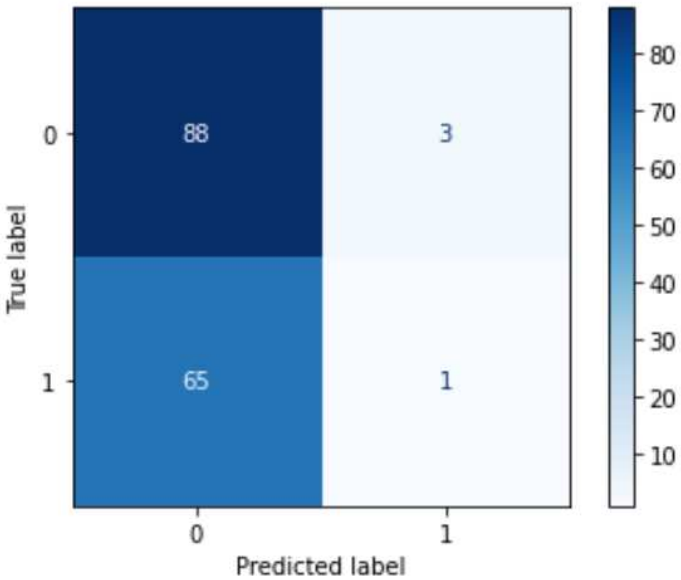
Anexo M. Resultados de classificação do algoritmo "Floresta Aleatória" para o conjunto de variáveis "Todas", no formato de matriz de confusão



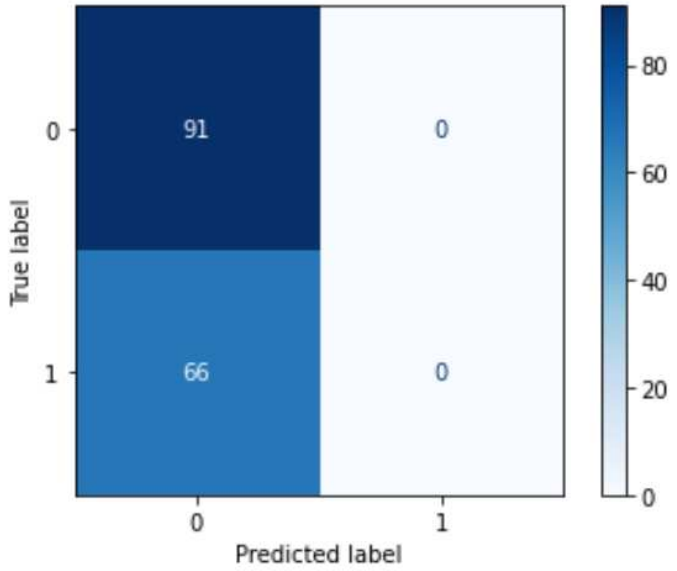
Anexo N. Resultados de classificação do algoritmo "SVM" para o conjunto de variáveis "TF", no formato de matriz de confusão



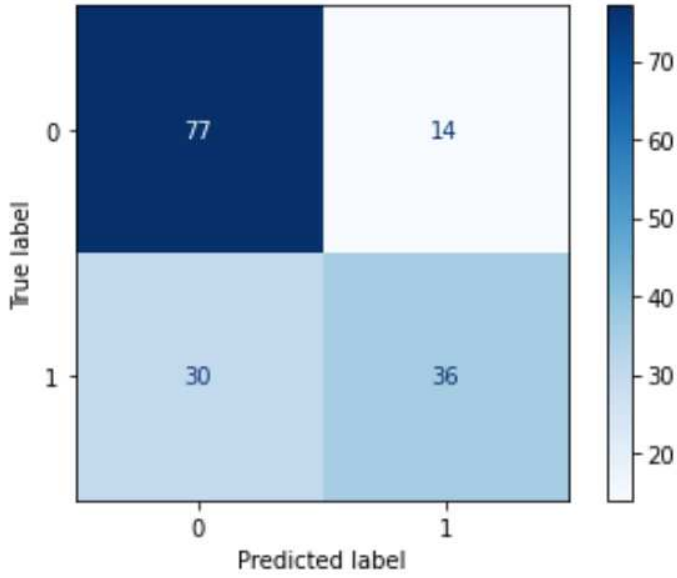
Anexo O. Resultados de classificação do algoritmo "SVM" para o conjunto de variáveis "TF-IDF", no formato de matriz de confusão



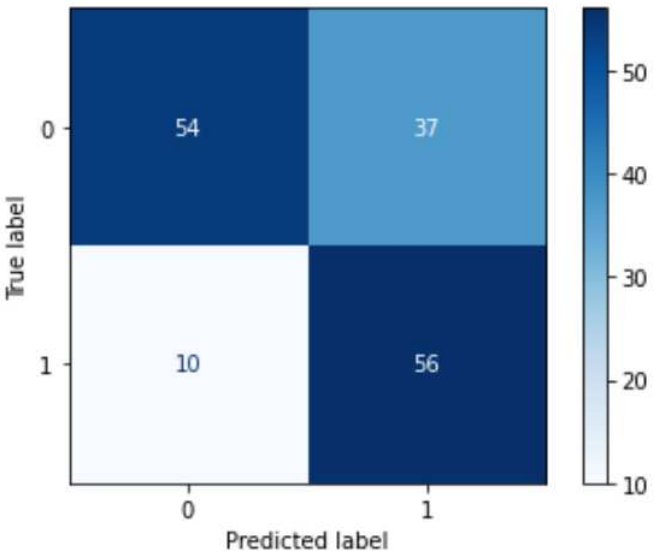
Anexo P. Resultados de classificação do algoritmo "SVM" para o conjunto de variáveis "Outras", no formato de matriz de confusão



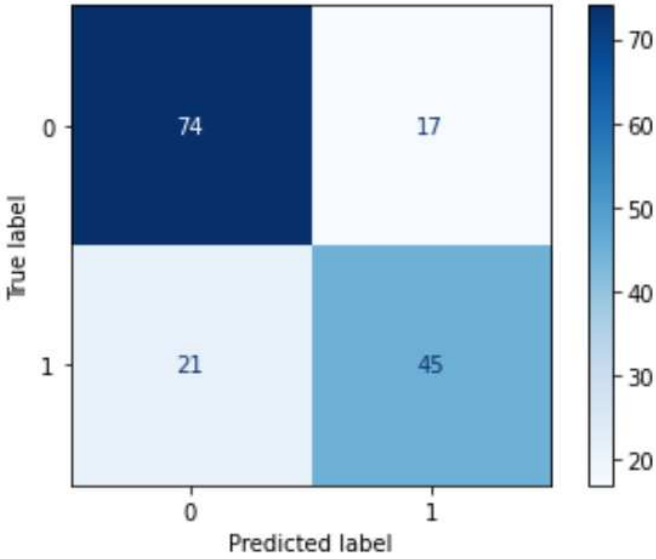
Anexo P. Resultados de classificação do algoritmo "SVM" para o conjunto de variáveis "Todas", no formato de matriz de confusão



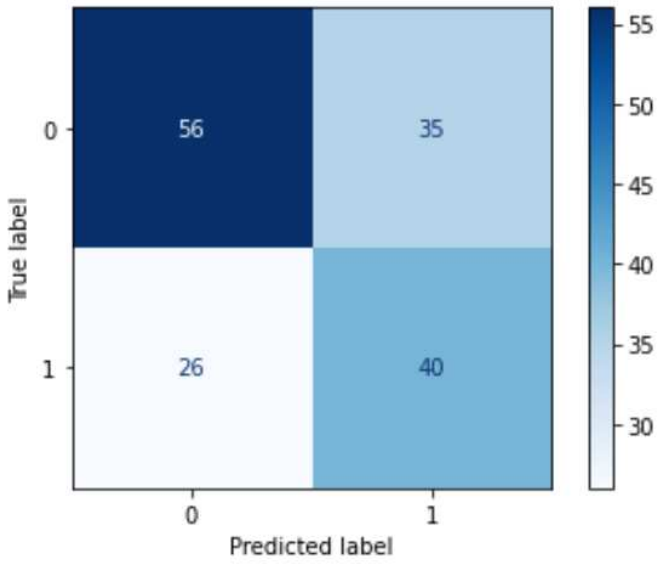
Anexo Q. Resultados de classificação do algoritmo "XGBoost" para o conjunto de variáveis "TF", no formato de matriz de confusão



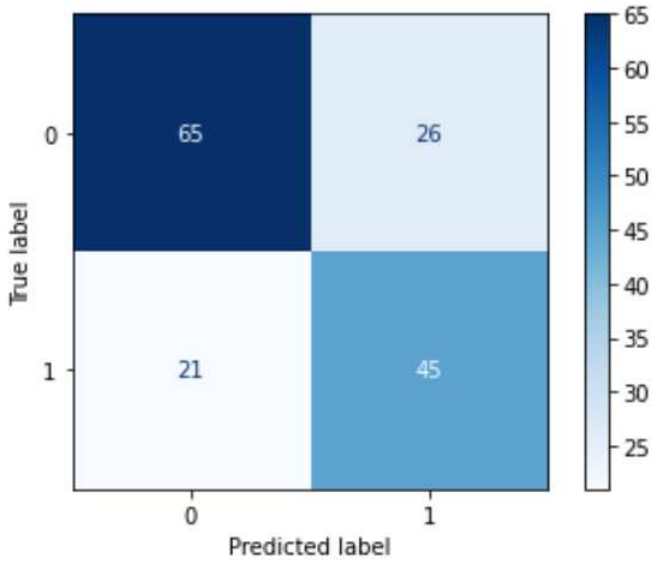
Anexo R. Resultados de classificação do algoritmo "XGBoost" para o conjunto de variáveis "TF-IDF", no formato de matriz de confusão



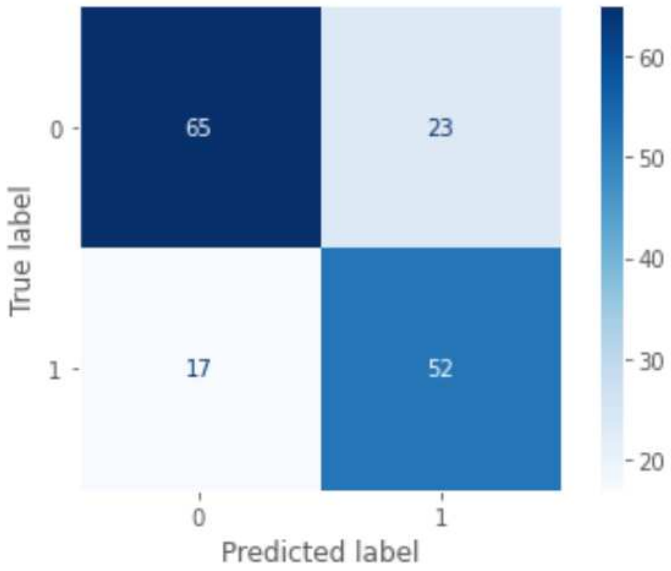
Anexo S. Resultados de classificação do algoritmo "XGBoost" para o conjunto de variáveis "Outras", no formato de matriz de confusão



Anexo T. Resultados de classificação do algoritmo "XGBoost" para o conjunto de variáveis "Todas", no formato de matriz de confusão



Anexo U. Resultados de classificação do algoritmo "NN" para o conjunto de variáveis "TF", no formato de matriz de confusão



Anexo V. Resultados de classificação do algoritmo "CNN" para o conjunto de variáveis "Embeddings", no formato de matriz de confusão