Escola Nacional de Administração Pública

Usando Inteligência Artificial (IA) na classificação de Licitações: um caso prático

Trabalho de Conclusão de Curso
apresentado como parte dos requisitos para
obtenção do grau de Especialista em
Ciência de Dados aplicada a Políticas Públicas.

Aluno: Glauber Volkmer

Orientador: Prof. MSc. Maxwell Sarmento de Carvalho

Brasília – DF
Setembro/2022

Escola Nacional de Administração Pública

Usando Inteligência Artificial (IA) na classificação de Licitações: um caso prático

Aluno: Glauber Volkmer

Controladoria-Geral da União (CGU)

Palavras-chave: licitação de obras, classificação, inteligência artificial.

Resumo

O avanço da digitalização de documentos e processos na Administração Pública trouxe novas possibilidades.

A ferramenta “Alice” da CGU, que realiza análise automatizada de artefatos textuais como editais e termos de referência, necessitava classificar antes da sessão pública da licitação quais tinham como objeto obras e serviços de engenharia, de forma a permitir uma análise antes de haver um vencedor para o certame.

Técnicas de inteligência artificial permitem que uma aplicação faça esta classificação de forma análoga a uma pessoa, “lendo” a descrição dos itens do edital.

O presente artigo apresenta a metodologia e os resultados de uma aplicação (software) desenvolvida em Python para fazer esta classificação a partir da descrição dos itens, tendo alcançado uma taxa de acerto de 99.22% e F-1 Score de 0.92.

Este artigo também traz o referencial básico para desmistificar o tema Inteligência Artificial a um público-alvo com diferentes áreas de formação.

Introdução

O avanço da digitalização de documentos e processos dentro da Administração Pública brasileira, para além das vantagens e desvantagens mais imediatas verificadas pelos usuários dos serviços públicos, trouxe também novas possibilidades que surgem constantemente com o avanço da tecnologia.

Um dos processos que passaram a ocorrer eletronicamente nesta digitalização é a realização de pregões. A digitalização em si não resolve completamente a questão do acesso a alguma informação específica desejada, dado que o volume de dados gerado é em escala tal que torna inviável a indivíduos acompanharem tudo o que ocorre sem o uso de ferramentas de tratamento de dados, busca e pesquisa cada vez mais sofisticadas.

Neste contexto, mas adentrando no âmbito do controle, está a ferramenta "Alice" da Controladoria-Geral da União (CGU). "Alice" é um acrônimo de "Analisador de Licitações, Contratos e Editais", implementado para auxiliar o auditor na busca, categorização e análise automatizada de artefatos textuais (não estruturado), tais como editais, termos de referência e outros documentos.

Uma necessidade identificada do Alice era classificar, antes da sessão pública, quais pregões tinham como objeto obras e serviços de engenharia. Destaque para "antes da sessão pública", pois o objetivo é que a classificação permita a execução de análises e trilhas de auditoria de obras antes que seja adjudicado um vencedor, momento em que a correção de impropriedades torna-se mais custosa. Não há dados estruturados de qualidade que tragam esta informação nos sistemas governamentais em que são realizados os pregões eletrônicos. Não antes da sessão pública, pelo menos.

Avanços obtidos no campo da Inteligência Artificial (IA), especialmente no processamento de linguagem natural (*Natural Language Processing* - NLP), tornarem esta tecnologia cada vez mais factível e aplicável na Administração Pública. Sendo linguagem natural um texto falado ou escrito na forma que seria utilizada na comunicação entre pessoas.

O processamento de linguagem natural, especificamente com uso de técnicas de Aprendizado de Máquina (*Machine Learning*), torna possível a uma aplicação determinar se o pregão tem como objeto obra ou serviço de engenharia de forma análoga à maneira que uma pessoa faria, simplesmente "lendo" a descrição dos itens.

E como parâmetro para definir que a aplicação está classificando com sucesso as licitações estipulou-se uma taxa de acerto mínimo, ou acurácia, de 95% e F-1 Score mínimo de 0.9.

Desta forma, os objetivos deste trabalho são:

- Demonstrar os resultados possíveis que a tecnologia atual de IA pode trazer no âmbito da Administração Pública por meio da apresentação de uma solução real implementada com IA e seus resultados (acurácia >= 95% e F-1 Score >= 0.9).
- Apresentar e desmistificar o tema Inteligência Artificial, e suas subdivisões (*Machine Learning/Machine Learning*), a um público-alvo com áreas de formação bastante diversas, dado que estas tecnologias passarão a estar cada vez mais presentes no dia a dia dos envolvidos com Administração Pública no Brasil.

Cabe destacar que este documento foi escrito tendo como público-alvo leitores oriundos da Administração Pública, com formações acadêmicas em diferentes áreas, em oposição a leitores especializados em Inteligência Artificial ou mesmo Ciência da Computação.

1. Fundamentação Teórica

O presente documento está baseado, principalmente, nos conceitos de (i) Licitação, (ii) Obra/Serviço de engenharia e (iii) Inteligência Artificial. A conceituação e a contextualização dos termos utilizados visa evitar ambiguidades e a extrapolação do conceito para além da definição utilizada no texto.

1.1 Licitação

Dado que esta é um documento voltado para a Administração Pública, e este é um conceito muito ligado a ela, suficiente dizer que o conceito utilizado é o proveniente do Direito Administrativo. Como exemplo, dentre vários autores consagrados, podemos citar a definição de Celso Antônio Bandeira de Mello:

> *Certame que as entidades governamentais devem realizar e no qual possibilitam a disputa entre os administrados interessados em com elas travar determinadas relações de conteúdo patrimonial, para promover a escolha da proposta mais vantajosa às conveniências públicas. O Instituto da licitação finca-se na idéia de competição, a ser travada de forma isonômica entre os que preencham os requisitos necessários ao bom cumprimento das obrigações que se comprometem.*
>
> (MELLO, 2011)

1.2 Obra

O termo “obra” possui vários conceitos associados, variando desde obras literárias, artísticas até obras civis. Neste trabalho, quando citada a expressão obra ou serviço de engenharia, está se referindo a um conceito encontrado, por exemplo, na Lei 8.666/93: “Obra é toda construção, reforma, fabricação, recuperação ou ampliação, realizada por execução direta ou indireta. (BRASIL, 1993)

1.3 Inteligência Artificial (IA)

O conceito de “inteligência artificial” é, por larga margem, o conceito com maior dificuldade de definição. Neste texto, previsivelmente, as definições utilizadas são todas da área da Ciência da Computação.

Uma das evidências desta dificuldade é que várias definições foram propostas ao longo de décadas, sendo que até hoje nenhuma parece satisfazer integralmente aos anseios dos pesquisadores.

Pode-se citar, por exemplo, uma das primeiras questões sobre Inteligência Artificial formulada por Alan Turing (1950), considerado o “pai” da computação por sua “Máquina de Turing”, em sua obra "*Computing Machinery and Intelligence*” (TURING, 1950), na qual ele pergunta: “as máquinas podem pensar?”.

Na busca pela reposta, Turing desenvolveu o que ficou conhecido como “Teste de Turing”, no qual um humano faria um interrogatório e avaliaria se as respostas teriam sido fornecidas por uma máquina ou por uma pessoa. Caso a máquina se fizesse passar por uma pessoa com sucesso, ela teria passado no “Teste de Turing”.

No entanto, definições mais modernas de IA descartam a necessidade de emular um humano para que um sistema seja considerado “inteligente”. A IA pode solucionar um problema à sua maneira, de maneira distinta a um humano, e ainda sim ser considerada inteligente.

Segundo McCarthy (2004), IA é a ciência e a engenharia de se construir máquinas inteligentes, especialmente programas de computador inteligentes. Está relacionada com a tarefa similar de usar computadores para entender a inteligência humana, mas a IA não precisa se restringir a métodos que são biologicamente observáveis. (tradução nossa)

Há ainda subdivisões do que é considerado Inteligência Artificial, sendo que as divisões abordadas neste trabalho foram baseadas naquelas utilizadas pela IBM (2022a). Inclusive, o crédito por

cunhar o termo “machine learning” é de um dos seus colaboradores, Arthur Samuel, em pesquisa de 1959 em torno do jogo de damas (IBM, 2022b).

A IA pode ser dividida em *Narrow AI* (IA Limitada – tradução nossa), ou *Weak AI* (IA Fraca – tradução nossa), e *Strong AI* (IA Forte – tradução nossa). De maneira bem resumida, a IA Limitada é treinada e focada em tarefas específicas, sendo que a IA Forte é mais genérica.

A IA Forte pode ser ainda dividida em *Artificial General Intelligence* (AGI), ou Inteligência Artificial Geral (tradução nossa) e *Artificial Super Intelligence* (ASI), ou Super Inteligência Artificial (tradução nossa).

Uma AGI é uma forma teórica de IA na qual sua inteligência seria equivalente à de um humano, sendo autoconsciente com a habilidade de resolver problemas, aprender e planejar o futuro. Já uma ASI, também ainda teórica, teria uma inteligência muito superior a um humano.

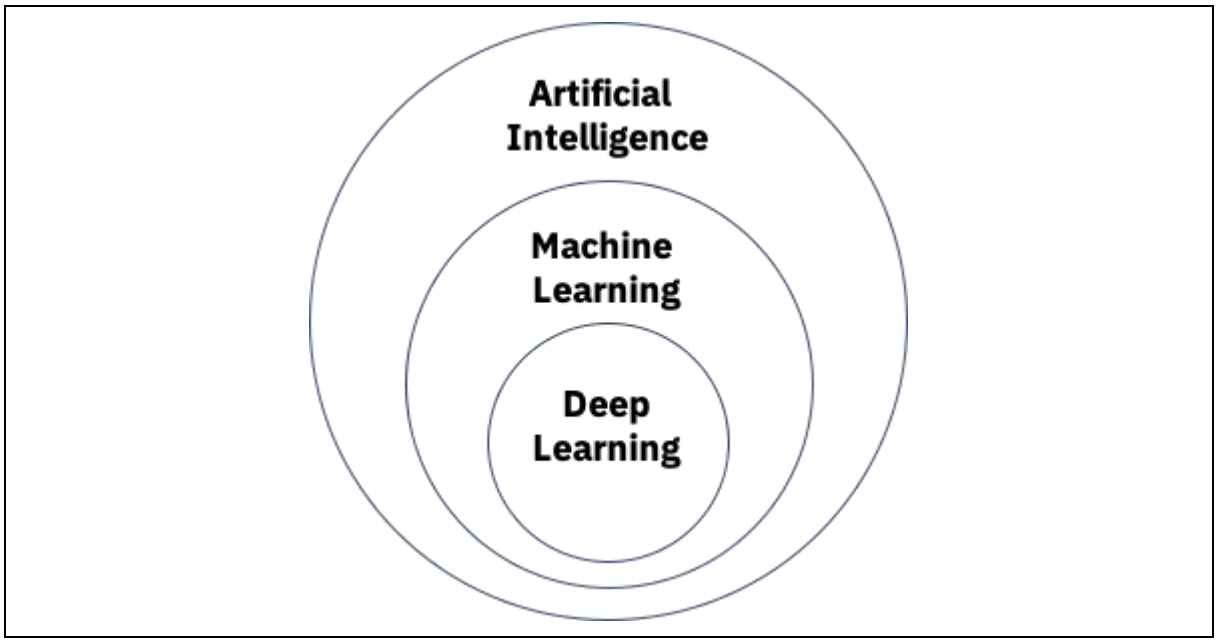
Neste trabalho, qualquer referência ao uso de “Inteligência Artificial” na solução apresentada está se referindo ao conceito de *IA Limitada* (Narrow IA), ou seja, um sistema treinado e focado para executar uma tarefa específica.

A definição de Inteligência Artificial pode parecer intuitiva, a princípio, mas ao tentar-se definir seu escopo e seus limites, várias dúvidas surgem e exigem a utilização de novos termos e conceitos, muitos deles bastante especializados como *Machine Learning* e *Deep Learning.*

1.3.1 Machine Learning (ML) x Deep Learning (DL)

Considerando que as duas expressões são muitas vezes utilizadas como se sinônimos fossem, cabe iniciar esta seção destacando as diferenças entre elas. Apesar de ambas serem áreas da Inteligência Artificial, na verdade o *Deep Learning* é uma subárea de *Machine Learning*:

Figura 1: *Deep Learning* é uma subárea de *Machine Learning*



Fonte: IBM (2022a)

De acordo com a UC Berkeley *School of Information* (2020), *Machine Learning* envolve o uso de aprendizado estatístico e métodos de otimização que permitem aos computadores analisar dados e identificar padrões. Em geral, é usado para fazer uma predição ou classificação, contando para isso com uma função de erro, para avaliar a precisão do modelo, e um processo de otimização, para se ajustar melhor aos pontos de dados.

Em outras palavras, a lógica da programação utilizada em um sistema de *Machine Learning* é diferente da lógica de um software tradicional. No método tradicional um programador (humano) cria (programa, codifica) um conjunto de regras para gerar respostas (resultados) a partir do processamento dos dados de entrada.

Já os algoritmos criados a partir do *Machine Learning* são formados a partir dos dados de entrada e dos resultados esperados, sendo que o sistema cria as regras internas que levam aos resultados. Ou seja, no lugar de um programador humano criando um "programa" (sequência de instruções) é um algoritmo de ML que cria o "programa" a partir dos dados.

E um modelo do tipo *Deep Learning* também é um modelo de *Machine Learning*, ou seja, é um modelo que cria as próprias regras internas que levarão aos resultados, mas que para isso faz uso de redes neurais (*neural networks*). O termo *Deep* (profundo) vem do fato de que são redes neurais com mais de três camadas (*layers*), incluídas as camadas de entrada (*input*) e saída (*output*).

A principal diferença de funcionamento entre um algoritmo *Deep Learning* e um algoritmo *Machine Learning* genérico (sem ser *Deep Learning*), está em **como** eles aprendem.

No *Deep Learning* muito do processo de extração de características dos dados é automatizado, enquanto um algoritmo de *Machine Learning* genérico ainda requer significativa intervenção humana no aprendizado, no qual um humano (especialista em ML) determina a hierarquia dos dados de forma a entender as diferenças entre os dados de entrada. Por esta razão que algoritmos de ML genéricos normalmente exigem dados mais estruturados.

Esta diferença torna o *Deep Learning* mais apto para lidar com grandes bases de dados ou, como notou o Prof. Lex Fridman do *Massachusetts Institute of Technology* (MIT), podemos pensar no *Deep Learning* como sendo uma ML escalável (capaz de crescer, de lidar com dados em grande escala).

A solução apresentada neste trabalho é uma solução de *Machine Learning*, mas sem fazer uso de redes neurais, portanto, sem ser do tipo *Deep Learning*.

1.3.2 Treinamento Supervisionado x Não-Supervisionado

Conforme visto na definição dos conceitos de Inteligência Artificial e de *Machine Learning*, os algoritmos de ML aprendem em um processo não muito diferente dos humanos, eles aprendem por observação durante seu treinamento. Mas existem dois tipos de treinamento para algoritmos de ML: treinamento supervisionado e não-supervisionado.

Treinamento Supervisionado

A principal diferenças ente os dois é o uso de conjunto de dados rotulados (*labeled data*). O treinamento supervisionado é aquele que faz uso de um conjunto de dados rotulados. (IBM, 2022d), enquanto o não- supervisionado dispensa seu uso. Esta diferença é tão fundamental que basicamente é a própria definição dos dois tipos de treinamento.

Com relação ao que vem a ser um dado rotulado ou o que significa rotular um dado, a *Amazon Web Services* (AWS, 2022) define que rotular dados é o processo de identificar dados (imagens, arquivos de texto, vídeos, etc) e adicionar um ou mais rótulos (*labels*) informativos que forneçam contexto para que o modelo de ML possa aprender a partir deles. Por exemplo, podem indicar se uma foto é de um carro ou de um pássaro, quais palavras foram ditas em um áudio ou se uma imagem de raio-x contém um tumor.

Há dois tipos de problemas para os quais é utilizado o aprendizado supervisionado: classificação e regressão:

a) Classificação: Na classificação o algoritmo "classifica" o dado em categorias (classes) específicas, como separar imagens de gatos e cachorros. Ou, usando um exemplo mais prático, classificam um email entre spam ou não. Os algoritmos classificadores mais comuns são: classificadores lineares (*linear classifiers*), máquinas de vetor de suporte (*support vector machines* – SVM), árvores de decisão (*decision tree*) e floresta aleatória (*random forest*). A nomenclatura efetivamente utilizada no mundo todo é a original da língua inglesa, mas traduções livres são citadas neste texto com o intuito único de trazer maior contexto à nomenclatura de cada um.

b) Regressão: A regressão é um método que busca entender a relação entre variáveis dependentes e independentes, ou seja, ele cria fórmulas matemáticas as quais, quando alimentadas com os dados de entrada (variáveis independentes), fornecem os resultados (variáveis dependentes). É utilizada para prever valores numéricos como o faturamento de determinado negócio, por exemplo. Exemplos de algoritmos de regressão bastante populares são: regressão linear, regressão logística e regressão polinomial.

Como já deve estar aparente neste momento, o problema abordado neste trabalho de "classificar" as licitações entre "obras/serviços de engenharia" ou "outros" é um problema do tipo classificação.

Treinamento Não-Supervisionado

Ainda de acordo com a IBM (2022d), os algoritmos de treinamento não-supervisionado descobrem padrões ocultos nos dados sem a necessidade de intervenção humana, de onde deriva a expressão "não supervisionado".

Diferentemente do treinamento supervisionado que necessita de dados rotulados, o treinamento não-supervisionado dispensa o uso de rótulos (*labels*) nos dados, sendo normalmente usados para 3 tarefas: agrupamento (*clustering*), associação (*association*) e redução de dimensionalidade (*dimensionality reduction*).

Outras Diferenças entre Treinamento Supervisionado x Não-Supervisionado

Para além da forma como os diferentes métodos aprendem, a seguir são apresentados outros aspectos em que estes diferem entre si:

a) Objetivos: No treinamento supervisionado o objetivo é prever um determinado resultado a partir de novos dados de entrada. Já se sabe de antemão que tipo de resultado será apresentado. Já no treinamento não-supervisionado o objetivo é obter *insights* a partir de grandes volumes de dados. O próprio algoritmo de ML determina o que é diferente ou relevante no conjunto de dados, de forma que o tipo do resultado pode ser surpreendente (não imaginado previamente).

b) **Aplicações:** Enquanto o treinamento supervisionado é ideal para detecção de spam, previsão do tempo, predição de preços, etc, o treinamento não-supervisionado é uma excelente opção para detecção de anomalias, personas de clientes e imagens médicas.

c) Complexidade: O treinamento supervisionado é mais simples, normalmente realizado por meio do uso de linguagens de programação como R ou Python. No treinamento não-supervisionado são necessárias ferramentas poderosas capazes de lidar com grandes quantidades de dados não classificados. Também são computacionalmente complexos em razão do volumoso conjunto de dados de treinamento que requerem.

d**) Desvantagens:** Modelos que fazem uso de treinamento supervisionado podem exigir bastante tempo em seu treinamento, e os rótulos para os dados de entrada e saída requerem especialização (conhecimento). Enquanto isso, modelos não supervisionados podem apresentar resultados muito imprecisos sem uma intervenção humana para validar as variáveis de saída (resultados).

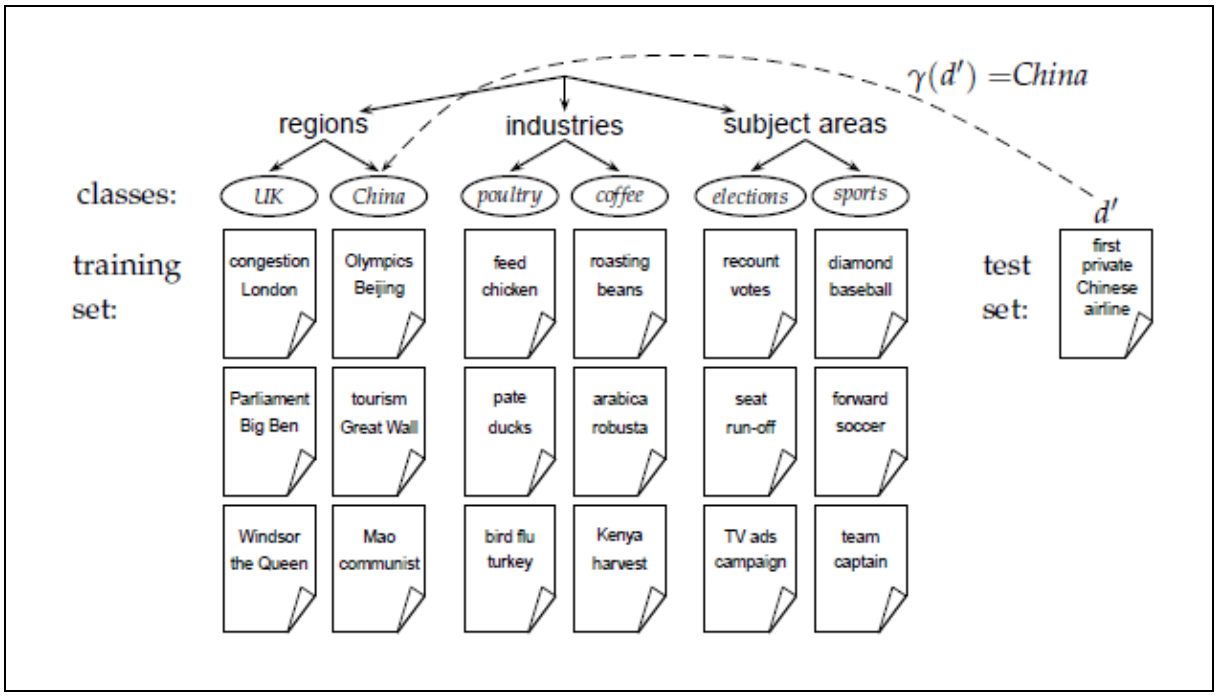
1.3.3 Classificação

Conforme visto na definição de *Machine Learning,* este é frequentemente usado em predições e classificações. Sendo inclusive o problema abordado neste trabalho um problema do tipo classificação (ver subitem “Treinamento Supervisionado”).

Um problema do tipo classificação pode ser definido como, dado um conjunto de classes, aquele no qual busca-se determinar a qual(is) classe(s) dado objeto pertence. Uma classe não precisa ser definida de forma muito restritiva, podendo ser definida de maneira mais abrangente, como um assunto ou área. Normalmente estas classes mais abrangentes são referidas como “tópicos”. (MANNING, RAGHAVAN, SCHÜTZE, 2009)

A imagem a seguir é um exemplo extraído da obra “*An Introduction to Information Retrieval*”, de Manning, Raghavan e Schütze (2009), com exemplos de classificação de documentos em classes:

Figura 2: ***Classificação em classes***



Fonte: Manning, Raghavan e Schütze (2009), p. 257

Na figura anterior, a função classifica o novo documento d' (*first private Chinese airline*) com a classe "China", que é a classificação correta.

As classes podem ter uma estrutura hierárquica, a qual pode ajudar na classificação do objeto. No exemplo da figura, há duas instâncias por categoria, sendo a classe "China" uma instância da categoria "*regions*".

Os modelos podem também definir a classificação como sendo "one-of", na qual o documento é associado a uma única classe, como ocorreu no exemplo da Figura ao associar o documento unicamente á classe "China".

Outro tipo de classificação é o "any-of", na qual um documento pode ser classificado como pertencente a mais de uma classe simultaneamente. Voltando ao exemplo da figura anterior, um documento que trata dos Jogos Olímpicos de 2008 poderia ser classificado como pertencente às classes "China" e "Sports", de modo que pode-se concluir que a classificação "one-of" para a hierarquia do exemplo não é a mais adequada.

Para o presente trabalho, no qual busca-se classificar as licitações como sendo "obra ou serviço de engenharia" ou "outros", considerou-se o problema como sendo do tipo "one-of".

1.3.4 Processamento de Linguagem Natural

De acordo com a IBM (2022c), *Natural Language Processing* (NLP), ou Processamento de Linguagem Natural, é a área da Inteligência Artificial focada em dar aos computadores a habilidade de entender textos, escritos ou falados, da mesma forma que os humanos entendem.

O NLP combina linguística computacional, ou modelagem da linguagem humana por meio de regras, com estatística e aprendizado de máquina. Juntas, estas tecnologias permitem aos computadores entenderem o significado completo do texto, incluídos aí intenção e sentimento do escritor/orador.

O NLP permite a existência de programas de computador que traduzem textos de uma língua para outra, respondem a comandos de voz e resumem textos volumosos rapidamente.

As primeiras aplicações de NLP eram baseadas em heurísticas, regras criadas por programadores, mas não podiam "crescer", ganhar escala, facilmente a fim de lidar com um fluxo sem fim de exceções ou um volume cada vez maior de dados.

Surge então o "statistical NLP", ou processamento de linguagem natural estatístico, que combina algoritmos e modelos de *machine learning* para automaticamente extrair, classificar e rotular elementos de texto e voz. (IBM, 2022c)

Atualmente, modelos *deep learning* baseados em redes neurais convolucionais (CNN - Convolutional Neural Networks) e redes neurais recorrentes (RNN - Recurrent Neural Networks) permitem que aplicações NLP aprendam enquanto são executadas e extraiam significados cada vez mais precisos a partir de volumes de dados cada vez maiores e sem tratamento (não-estruturados e não-rotulados).

Os modelos atuais permitem a execução mais precisa de tarefas complexas no âmbito do NLP, como por exemplo as tarefas a seguir:

a) ***Speech recognition* ou reconhecimento de voz**: é a tarefa de converter texto falado (áudio) em texto escrito. Por esta razão também conhecida como "*speech-to-text*". O que torna esta tarefa particularmente difícil é a forma como as pessoas falam: rapidamente, emendando palavras, variando as entonações, com diferentes sotaques e, frequentemente, com erros gramaticais

b) ***Speech tagging,* ou classificação gramatical:** é o processo de determinar a classificação gramatical de parte do texto com base em seu uso e seu contexto. É o *speech tagging* que identifica que a palavra "pinta é um verbo na frase "Ele pinta com maestria" e um substantivo na frase "Ele tem uma pinta no nariz".

c) ***Word sense disambiguation* ou desambiguação de palavras:** é a seleção do significado de uma palavra que possui múltiplos significados, por meio de um processo de análise semântica que determina qual o significado mais apropriado dentro do contexto. É a desambiguação que ajuda a distinguir o significado do substantivo "pinta" em "pinta no nariz" (mancha na pele) e em "pinta de Guiness" (unidade de medida de 0,47 litros aproximadamente).

d) ***Named entity recognition* (NEM) ou reconhecimento de entidade nomeada**: é a identificação de vocábulos ou frases como entidades. É a NEM que identifica "Bahia" como um lugar e "João" como um homem, por exemplo.

e) ***Co-reference resolution* ou resolução de correferências:** é a tarefa de identificar se e quando duas palavras se referem a uma mesma pessoa ou objeto. O exemplo mais comum é no uso de pronomes que se referem a pessoas já citadas (exemplo: "ela" = "Mariana"), mas também pode envolver identificar uma metáfora (exemplo: "galinha" = "Mariana" ou "porco" = "João").

f) ***Sentiment analysis* ou análise de sentimento**: é a processo de extrair características subjetivas do texto como atitudes, emoções, sarcasmo, confusão, suspeição, surpresa, etc.

g) ***Natural language generation* ou geração de linguagem natural**: é gerar texto, em linguagem falada ou escrita, a partir de informações estruturadas.

A aplicação desenvolvida no presente trabalho não executa as tarefas descritas acima pois foram utilizadas técnicas de Processamento de Linguagem Natural estatístico.

Considerou-se que para o nível de complexidade do problema apresentado que um método de NLP estatístico era adequado, de modo que não foi feito o uso de bibliotecas de processamento de linguagem natural, como o *Natural Language Toolkit* (NLTK) disponível em Python, por exemplo.

O NLP estatístico utilizado é baseado em frequências de conjunto de caracteres e Máquina de Vetor de Suporte, conforme será descrito no capítulo "Metodologia".

2. Metodologia

Conforme abordado no Capítulo de Conceituação, a solução apresentada neste trabalho é uma solução de *Machine Learning*, na qual foi feito um treinamento supervisionado utilizando dados rotulados.

Para chegar ao produto final, várias etapas foram percorridas ao longo do desenvolvimento. O presente capítulo descreverá as etapas percorridas em ordem cronológica, na medida do possível, dado que foram realizados vários ciclos/iterações do processo completo de modo a aperfeiçoar o produto intermediário de cada etapa.

2.1 Acesso aos Dados

Os dados dos pregões eletrônicos utilizados no treinamento foram obtidos por meio do CGUDATA, o qual possui uma cópia dos dados do DW-SIASG, o sistema que originalmente armazena os dados dos pregões eletrônicos.

O CGUDATA é o ambiente de gestão de dados institucionais da CGU, que centraliza bases de dados antes localizadas em diversos bancos de dados da instituição. A ferramenta é composta por uma infraestrutura de servidores de banco de dados Microsoft SQL Server.

Já o DW-SIASG é um banco de dados centralizado dos sistemas de Compras Governamentais, com dados das compras e contratações efetuadas pela Administração Pública Federal, assim como dos fornecedores do Governo Federal.

As informações DW-SIASG eram atualizadas mensalmente com dados extraídos do Sistema Integrado de Administração de Serviços Gerais (SIASG), do Portal de Compras do Governo Federal (Comprasnet) e do Sistema de Cadastramento Unificado de Fornecedores (SICAF). Foram utilizados os dados do Comprasnet simplesmente por ser este o sistema utilizado para realizar os pregões eletrônicos durante décadas e, portanto, era o sistema que armazenava os dados históricos de milhares de pregões realizados no período, inclusive seus editais e descrição dos itens licitados.

Em 2021 o “Comprasnet” passou a ser chamado de “Compras.gov.br” com novas funcionalidades. Porém, a nomenclatura do sistema que armazena os dados das licitações, ou como ele é estruturado, são irrelevantes para a solução apresentada neste trabalho. Os sistemas governamentais citados aqui, foram mencionados somente em razão de terem sido a fonte dos dados utilizados no treinamento do classificador.

Sua continuidade, ou substituição, não interfere no uso do classificador pois basta informar a descrição do item da licitação em forma textual que o item será classificado, independentemente do sistema no qual a descrição está armazenada.

Ao todo foram acessados na realização deste trabalho um total de mais de 31 milhões (31.576.895) registros de itens licitados no período de 1999 a 2020, referentes a pouco mais de 4.8 milhões de licitações (4.837.722).

2.2 Rotulagem

Como visto anteriormente, a solução de ML implementada fez uso de treinamento supervisionado, ou seja, de dados rotulados. O rótulo utilizado foi uma coluna numérica na qual um valor igual a 1 (um) indicaria que o item era de uma licitação de obra ou serviço de engenharia. Valor igual a 0 (zero) indicaria ser um item de outra categoria.

No entanto este rótulo é inexistente nas bases de dados do DW-SIASG, também inexistente no CGUDATA, de modo que este rótulo precisou ser criado durante o desenvolvimento.

Uma técnica frequentemente utilizada na rotulagem é a classificação realizada por pessoas. Por exemplo, o leitor já deve ter se deparado inúmeras vezes com pedidos de sites como o Google para informar, dentre um conjunto de imagens, quais delas retratam pontes, semáforos, bicicletas, etc,. Enquanto isso é, de fato, uma forma de se verificar que o usuário acessando o site é um humano e não um robô (justificativa dada pela Google para realizar o questionamento), é também uma forma da Google utilizar seus usuários para classificar (rotular) imagens.

No presente trabalho, sendo o desenvolvimento realizado por uma única pessoa, rotular "manualmente" os mais de 31 milhões de itens não era uma alternativa. Mesmo que se fizesse a rotulagem de um grupo menor de itens para o treinamento, como 1 milhão de itens (o quantitativo aproximado utilizado no treinamento), ainda assim é uma quantidade inviável para uma única pessoa rotular "manualmente".

A técnica utilizada então foi uma rotulagem automática realizada por meio da linguagem SQL (Stru*ctured Query Language* ou Linguagem de Consulta Estruturada), uma linguagem utilizada para consultar bancos de dado relacionais, fazendo uso de dados das licitações existentes no banco, inclusive de dados disponíveis somente para licitações já concluídas, como o empenho.

Ainda assim, esta etapa de rotular os dados foi, por ampla margem, a etapa mais trabalhosa e demorada para superar.

Os dados efetivamente utilizados para a definição do rótulo foram os seguintes campos:

- [DS_CMPR_OBJETO_COMPRA] da tabela [D_CMPR_COMPRA], que contém a descrição textual do objeto da licitação;
- [DS_ITCP_ITEM_COMPRA] e [DS_ITCP_COMPL_ITEM_COMPRA] da tabela [D_ITCP_ITEM_COMPRA], que contém a descrição textual de cada item da licitação;
- [DS_ITCP_UNIDADE_FORNECIMENTO] da tabela [D_ITCP_ITEM_COMPRA], que contém a descrição textual da unidade de fornecimento do item;
- [DS_EMPN_NAT_DESPESA] da tabela [D_EMPN_EMPENHO], que contém o código de Natureza da Despesa do empenho da licitação.

A consulta SQL realizada, com mais de 300 linhas, não será detalhada aqui em função do público-alvo do artigo, mas em linhas gerais o que a consulta fez foi, para cada Código de Despesa de uma relação pré-determinada, buscar se havia na descrição do objeto da licitação, ou na descrição do item, uma expressão dentre uma série de possíveis expressões. Caso fosse encontrada uma daquelas expressões, o item seria rotulado como 1 (um), ou seja, obra ou serviço de engenharia.

Alguns códigos de despesa foram agrupados e cada grupo foi associado a uma série de expressões específica para o grupo.

Para facilitar a compreensão do que foi executado, será feito uso de um exemplo, simplificado em relação ao caso real, com o seguinte código de Natureza da Despesa:

- Categoria Econômica: 4 (Despesas de Capital);
- Grupo de Natureza de Despesa: 4 (Investimentos);
- Elemento de Despesa: 51 (Obras e Instalações).

Para este exemplo específico, buscou-se expressões do tipo “reforma”, “obra civil”, “manutenção predial”, dentre várias outras expressões. No caso real, todas as expressões eram do tipo “expressão regular” (regex), fazendo uso de caracteres coringas, principalmente para evitar problemas com acentuação.

O código do exemplo citado foi escolhido justamente por ser o código com o grupo de expressões mais simples, dado que o Elemento de Despesa “51” era bem específico para o contexto. No entanto, outros elementos de despesa mais genéricos exigiram grupos de expressões bem mais complexos, como por exemplo os elementos de despesa: 39 - Outros Serviços de Terceiros - Pessoa

Jurídica; 37- Locação de Mão-de-Obra; 36 - Outros Serviços de Terceiros - Pessoa Física; 35 - Serviços de Consultoria; 92 - Despesas de Exercícios Anteriores.

A relação completa de códigos de Natureza da Despesa utilizados é a seguinte, sendo que "xx" representa qualquer número entre 0 e 99:

- 4.4.xx.51.00 Obras e Instalações
- 4.4.xx.92.00 Despesas de Exercícios Anteriores
- 3.3.90.39.00 Outros Serviços de Terceiros - Pessoa Jurídica
- 3.3.90.37.00 Locação de Mão-de-Obra
- 3.3.90.36.00 Outros Serviços de Terceiros - Pessoa Física
- 3.3.90.35.00 Serviços de Consultoria
- 3.3.90.92.00 Despesas de Exercícios Anteriores
- 4.4.90.39.00 Outros Serviços de Terceiros - Pessoa Jurídica
- 4.4.90.35.00 Serviços de Consultoria
- 4.4.90.92.00 Despesas de Exercícios Anteriores
- 3.3.90.30.00 Material de Consumo
- 4.4.90.52.00 Equipamentos e Material Permanente
- 3.3.90.37.00 Locação de Mão-de-Obra
- 3.3.90.92.00 Despesas de Exercícios Anteriores
- 4.4.90.92.00 Despesas de Exercícios Anteriores

Esta etapa de rotulagem passou por dezenas de iterações, algumas após revisão manual ainda dentro do banco de dados e outras tantas após comparativo do rótulo com a classificação da solução de *Machine Learning* (ML), na qual verificava-se que a classificação de ML estava correta e o rótulo, feito em SQL, incorreto, levando a uma nova versão do código SQL e uma nova versão dos rótulos, reiniciando todo o ciclo.

2.3 Seleção dos Dados para Treinamento

Feita a rotulagem dos dados, verificou-se que dos 31,5 milhões (31.576.895) itens de licitação rotulados, por volta de meio milhão (596.152 ou 1.9%) haviam sido rotulados como sendo obra ou serviço de engenharia (valor 1) e os restantes ~31 milhões (30.980.743 ou 98.1%) rotulados como "outros" (valor 0).

Estes dados revelam uma proporção de quase 52 itens com rótulo “outros” para cada item rotulado como “obra/serviço de engenharia”.

Houve basicamente dois problemas que impediram a utilização dos ~31,5 milhões de itens no treinamento: (i) é computacionalmente inviável para o hardware utilizado no desenvolvimento treinar com este volume de dados e (ii) esta proporção de 52:1 revela um desbalanceamento importante que impacta negativamente no desempenho da solução em ML.

De maneira resumida, um conjunto de dados desbalanceado é definido pela grande diferença de distribuição, 52:1 no caso abordado, das classes de um conjunto de dados. Isto significa que o conjunto de dados possui um viés em direção à classe mais frequente, de forma que um algoritmo treinado com este conjunto de dados terá o mesmo viés em direção a esta classe.

Dito de outra forma, o algoritmo aprenderá que se uma das classes é quase 99% das vezes a classe correta, ele tenderá a classificar desta forma pois parte de uma probabilidade de acerto de 99%, se feita a classificação um número suficiente de vezes (Lei dos Grandes Números).

Para balancear o conjunto de dados de treinamento, reduzindo seu viés, optou-se por trabalhar (não treinar, como será visto mais a frente) com todos os ~0.5 milhão de itens rotulados como obra e número similar de itens rotulados como “outros“, resultando em um conjunto de dados com pouco mais de 1,2 milhão de itens, com distribuição aproximada de 50% para cada uma das duas classes.

Para selecionar os ~0.5 milhão de itens rotulados como “outros”, dentre os ~31 milhões rotulados desta forma, implicando em uma redução de quase 52 vezes na quantidade de itens com teste rótulo, foi feita a ordenação de todos os ~31 milhões por ordem alfabética dos campos [DS_ITCP_ITEM_COMPRA] e [DS_ITCP_COMPL_ITEM_COMPRA], a descrição textual dos itens, e então feita a seleção de um registro a cada 51.

Ao final, o conjunto de dados utilizado foi de 1.203.607 itens, sendo 607.363 rotulados como “outros” e 596.152 rotulados como “obra ou serviço de engenharia”.

2.4 Tratamento dos Dados

Foram realizados alguns poucos procedimentos de tratamento dos dados antes do treinamento do modelo de *Machine Learning*:

a) Retirada de registros com campos nulos, independentemente do campo, para evitar que itens com descrição vazia influenciassem no treinamento, por exemplo;

b) Retirada de caracteres como aspas duplas, neste caso o motivo foi evitar problemas ao exportar os resultados para um arquivo CSV;
c) Substituição de qualquer letra maiúscula pelo seu correspondente minúsculo, dado que o modelo é sensível a esta diferença.

Outras tentativas, como por exemplo a substituição de “ç” por “c” ou a eliminação de acentos ortográficos, resultaram em piora no desempenho do modelo de ML e, por este motivo, foram descartadas.

2.5 Tokenização/Normalização

Antes de efetivamente treinar o modelo escolhido de *Machine Learning* com o conjunto de dados, foi realizada uma “metamorfose” em etapas dos dados.

Uma das etapas é a tokenização, que vem a ser a ação de pegar uma sequência de caracteres e quebrá-la em pedaços, chamados tokens, muitas vezes descartando caracteres no processo como caracteres de pontuação, por exemplo. Durante a tokenização é gerada uma matriz, ou tabela, com a quantidade de cada token no texto.

Segundo Manning, Raghavan e Schütze (2009) um token é “uma instância de uma sequência de caracteres em dado documento que são agrupados como uma unidade semântica útil para o processamento” (tradução nossa).

Há várias opções de tokenização, mas neste trabalho a tokenização foi realizada por meio da classe <sklearn.feature_extraction.text.CountVectorizer> da biblioteca scikit-learn. Scikit-learn é uma biblioteca de M*achine Learning* para a linguagem de programação Python.

Porém, há vários casos em que, por razões semânticas, é preciso que duas sequências de caracteres que não são exatamente iguais sejam tratadas como equivalentes. Manning, Raghavan e Schütze (2009) trazem o exemplo dos tokens “USA” e “U.S.A.” que, embora distintos em sua codificação, para fins semânticos precisam ser tratados como equivalentes. Este processo é denominado normalização.

A normalização realizada neste trabalho foi a partir da classe <sklearn.feature_extraction.text.TfidfTransformer>, que traduz a tabela de frequência de tokens em uma representação normalizada do tipo TF-ID.

O acrônimo “TF-ID”, que significa *term-frequency times inverse document-frequency*, ou frequência do termo vezes o inverso da frequência do documento (tradução nossa) é uma técnica bastante utilizada pois apresenta bons resultados na classificação de documentos TF-ID (2022). A “TF-ID” é uma combinação das técnicas *term-frequency e inverse document-frequency*.

Esta é uma seção que seria mais densamente explorada caso o artigo fosse endereçado a especialistas em ML, mas neste texto a definição superficial das técnicas parece adequada.

A técnica *term-frequency* envolve atribuir um peso (ponderação) a cada termo e o peso dependerá do número de ocorrências do termo no documento. (Manning, Raghavan e Schütze, 2009)

Cabe destacar que as expressões “termo” e “token” não são sinônimos. Um termo é uma classe que representa todos os tokens que contém uma mesma sequência de caracteres. Por exemplo, todos os tokens que contém a sequência de caracteres “xpto” formarão um termo “xpto” e todos os tokens que contém a sequência “qwer” formarão um termo “qwer”, etc.

Já a técnica *inverse document-frequency* é um mecanismo para atenuar o efeito de termos que ocorrem tão frequentemente a ponto de não serem significativos para determinar sua relevância. Trazendo novamente um exemplo de Manning, Raghavan e Schütze (2009), temos que uma coleção de documentos da indústria automobilística provavelmente terá a sequência “auto” em quase todos os documentos.

A *document-frequency (id)* é definida como sendo o número de documentos na coleção que contém dado termo. E a *inverse document-frequency (idf)* é definida como:

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

Sendo “N” o número de documentos na coleção e “df” a *document-frequency* do termo “t”, ou o número de documentos na coleção que contém o termo “t”.

2.6 Algoritmo de *Machine Learning*

Estando os dados, ou as descrições textuais dos itens, convertidos para um formato numérico, como resultado das etapas de tokenização e normalização, finalmente encontram-se prontos para o treinamento do modelo de *Machine Learning* escolhido para classificar os itens: o modelo *Linear Support Vector Classification* da classe <sklearn.svm.LinearSVC>.

A classe Linear *Support Vector Classification* é uma implementação de uma *Support Vector Machine* (SVM), ou Máquina de Vetor de Suporte (tradução nossa), que é um modelo de aprendizado de máquina baseado em espaço vetorial, cujo objetivo é encontrar a fronteira entre duas classes que está o mais distante possível de qualquer ponto nos dados de treinamento, possivelmente descartando alguns pontos como outliers ou ruído (tradução nossa). (Manning, Raghavan e Schütze, 2009)

O desenvolvimento dos modelos classificadores de ML ao longo do tempo resultou em uma série de modelos candidatos: máquinas de vetor de suporte (s*upport vector machines)*, árvores de decisão (*boosted decision trees*), regressão logística (*regularized logistic regression*), redes neurais (*neural networks*) e florestas aleatórias (*random forests*).

A escolha pela *Linear Support Vector Classification* se deu principalmente pelos seguintes motivos:

- SVMs tem sido tradicionalmente utilizadas com sucesso na classificação textual; (Manning, Raghavan e Schütze, 2009, p. 344)
- SVMs são recomendadas para dados de treinamento que serão separados em apenas duas classes, dados que estes podem ser linearmente separados, ou seja, separados por uma reta; (Manning, Raghavan e Schütze, 2009, p. 345)
- A classe *Linear Support Vector Classification* foi implementada com maior flexibilidade de escolha das penalidades e funções de perda (*loss functions*), estando mais apta a lidar com um grande conjunto de dados de treinamento, em comparação com a classe SVC com o parâmetro kernel=’linear’. (LSVM, 2022)
- Baixa complexidade do treinamento.

Importante destacar que, antes da realização do treinamento, o conjunto de dados com pouco mais de 1.2 milhão de itens, já devidamente balanceados (~50:50) entre as classes, foi separado entre conjunto de dados para treino (90%) e conjunto de dados para teste (10%) por meio da função “train_test_split()”.

Ou seja, somente estes 90% dos dados, aproximadamente 1.1 milhão, foram utilizados no treinamento e 10% deles, por volta de 120 mil itens, foram separados para teste. Quer dizer, no momento do teste, após o treinamento, cada item destes ~120 mil é um item inédito para o modelo.

Também é importante destacar que somente a descrição dos itens e a unidade de fornecimento foram utilizadas no treinamento, descartando-se a utilização da descrição do objeto do pregão ou

o valor do item, por exemplo, pois verificou-se, após alguns testes, que o uso destes piorava o desempenho do modelo.

3. Resultados

O presente capítulo apresenta os resultados alcançados pelo modelo desenvolvido ao longo do trabalho, visando identificar o nível de acerto atingido.

Dados de Teste Balanceados

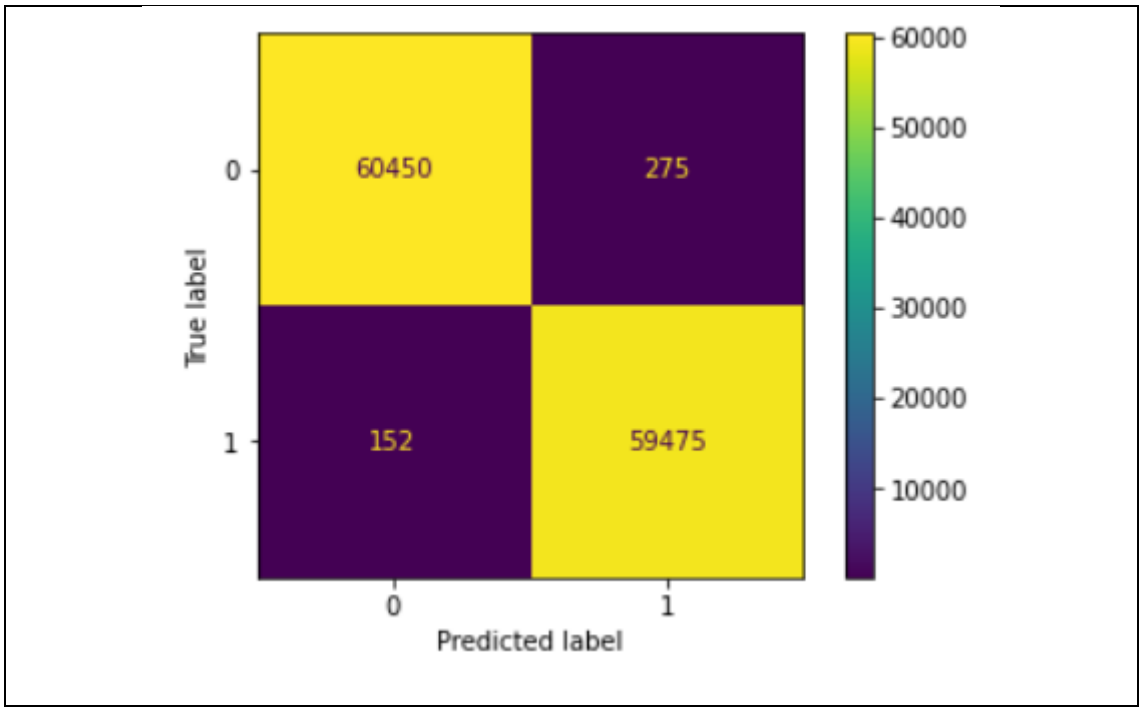
O modelo *Linear Support Vector Classification* utilizado apresentou um score de 0.996 para os dados de teste (10% do total do conjunto de dados balanceado), ou seja, uma acurácia de 99.6%. Ou em outras palavras, em 99.6% dos itens o modelo classificou corretamente se o item era "obra/serviço de engenharia" ou "outros".

A acurácia é uma métrica para modelos de classificação que mede a proporção entre o número de classificações corretas e o número de classificações feitas.

Os dados de precisão, recall e f1-score do teste feito com mais de 120 mil itens (10% dos 1.2 milhão) foram os seguintes:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	60725
1	1.00	1.00	1.00	59627
accuracy			1.00	120352
macro avg	1.00	1.00	1.00	120352
weighted avg	1.00	1.00	1.00	120352

A Matriz de Confusão, que permite visualizar de forma gráfica a distribuição dos resultados (Falso Positivo, Verdadeiro Positivo, Falso Negativo, Verdadeiro Negativo) ficou como segue:



Explicando a matriz de confusão, o eixo “X” traz os resultados do modelo de ML implementado (neste caso, 1 = “obra/serviço de engenharia” e 0 = “outros”) enquanto o eixo Y traz o rótulo.

Por exemplo, analisando o quadrante inferior esquerdo da matriz, temos que o modelo classificou como “outros” (valor 0) 152 itens que haviam sido rotulados como “obra/serviço de engenharia” (valor 1), resultando em 152 Falsos Negativos.

Concluindo, dos 120.352 itens, inéditos para o modelo já que estes itens não foram utilizados no treinamento, o modelo:

- Classificou 152 (0.13%) incorretamente como “outros”;
- Classificou 59.475 (49.42%) corretamente como “obra/serviço de engenharia”;
- Classificou 60.450 (50.23%) corretamente como “outros”;
- Classificou 275 (0.23%) incorretamente como “obra/serviço de engenharia”.

Dados de Teste Desbalanceados

Como visto anteriormente na Seleção dos Dados, o conjunto de dados foi selecionado de forma balanceada (50:50) entre as classes de modo a mitigar o viés no treinamento. Contudo, os dados “reais” não são balanceados, em geral eles seguem uma proporção de 52 itens “outros” para cada 1 item de “obra/serviço de engenharia”.

Para saber se o modelo manteria o desempenho com dados desbalanceados foi realizado outro teste com novo conjunto de dados contendo pouco mais de 1 milhão de itens, mas tendo 1.8% dos itens rotulados como “obra” e 98.2% rotulados como “outros”.

Pode-se afirmar que, com relação à acurácia, o modelo manteve o padrão de desempenho alcançando um score de 0.995. Ou seja, uma queda de apenas 0.1% na acurácia em relação ao conjunto de dados balanceado.

No entanto, a acurácia não é uma boa métrica para dados com uma distribuição de classes desbalanceada.

Uma métrica mais adequada para conjunto de dados desbalanceados é o F1-score. O F1-score é uma composição entre precisão e recall. (KORSTANJE, 2021).

Uma boa precisão implica em um modelo que talvez não encontre todos os positivos, mas quando encontra dificilmente estará errado. Já um modelo com bom recall é quase o oposto, é um modelo que é capaz de encontrar praticamente todos os casos positivos, mesmo que identifique alguns negativos incorretamente como positivos.

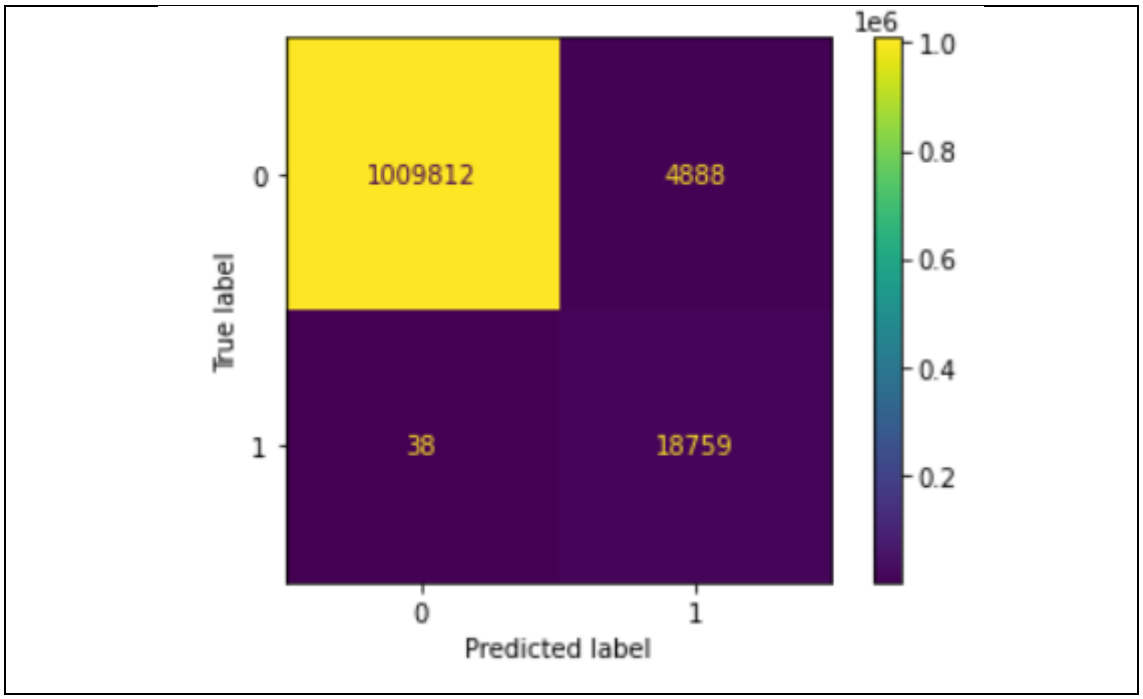
O F1-score é definido como sendo a média harmônica da precisão e do recall, calculado pela seguinte fórmula:

$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

Os dados de precisão, recall e f1-score do teste desbalanceado com mais de 1 milhão de itens (1.033.497) foram os seguintes:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1014700
1	0.79	1.00	0.88	18797
accuracy			1.00	1033497
macro avg	0.90	1.00	0.94	1033497
weighted avg	1.00	1.00	1.00	1033497

A Matriz de Confusão obtida foi:



Percebe-se que a queda no desempenho é quase que exclusivamente decorrente da piora na precisão dos casos positivos, ou seja, o modelo tende a trazer um pouco mais de falsos positivos em comparação com os dados balanceados. Os falsos positivos representam 26% do total de positivos verdadeiros, ainda que o total de falsos positivos (4.888) represente apenas 0,47% do total de itens.

No entanto, o recall permanece com desempenho muito bom, igual a 1, encontrando 99,8% dos casos positivos.

Já para os casos negativos, ou seja, a classe “outros”, tanto a precisão, o recall e o f1-score permaneceram impecáveis com resultado igual a 1.

O F1 Score médio obtido foi de 0.94 e 1.00 para o médio ponderado.

Classificando Por Licitação x Item

Uma característica importante do modelo desenvolvido é que ele faz a classificação por item de licitação e não por licitação. Portanto para um pregão com três itens, o modelo apresentará três classificações.

Para que o modelo seja utilizado para classificar uma licitação, um pregão, uma possível aplicação é considerar que o certame seja classificado como “obra ou serviço de engenharia” quando pelo menos um de seus itens é classificado como tal.

Aplicando este raciocínio para os 1.033.497 itens de licitação do conjunto de dados desbalanceados, fazendo um agrupamento pelo código da compra (código da licitação) temos que estes itens representam um total de 603.749 licitações, sendo 591.598 (98%) licitações rotuladas como “outros” e 12.151 (2%) rotuladas como “obra ou serviço de engenharia”.

Destas 603.749 licitações, temos que em 599.075 licitações a classificação do modelo foi correta, apresentando 99.22% de acurácia. Portanto, em 4.674 licitações a classificação foi incorreta, das quais 36 falsos negativos e 4.638 falsos positivos.

Os dados de precisão, recall e f1-score por licitação foram os seguintes:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	591598
1	0.73	1.00	0.84	12151
accuracy			0.99	603749
macro avg	0.87	0.99	0.92	603749
weighted avg	0.99	0.99	0.99	603749

Constata-se que para as licitações o F1-score da classe “outros” permanece em nível ótimo, ficando igual a 1. Para a classe “obras e serviços de engenharia” a F1-score caiu um pouco, ficando em 0.84 (era 0.88 para os itens).

O F1 Score médio obtido foi de 0.92 e 0.99 para o médio ponderado.

Conclusão

A principal conclusão é que a acurácia superior a 99.2% e o F1 Score de 0.92 (0.99 no ponderado) são indicativos de que o modelo é viável e capaz de produzir bons resultados.

A única observação é que, como a classe “outro” é bem mais frequente que a classe “obra/serviço de engenharia”, o modelo pode trazer alguns poucos falsos positivos, menos de 0.5% (4.888 de

1.033.497) do total de itens classificados foram falsos positivos e menos de 0.8% (4.638 de 603.749) para o total de licitações.

Por outro lado, dificilmente alguma contratação de obra ou serviço de engenharia deixará de ser classificada como tal pelo modelo. O resultado apurado demonstrou que menos de 0.004% (38 de 1.033.497) dos itens e menos de 0.006% (36 de 603.749) das licitações rotuladas como "obra ou serviço de engenharia" deixaram de ser assim classificadas pelo modelo.

Este resultado pode ainda ser melhorado pois, depois da classificação, outros tipos de filtros e validações podem ser aplicados aos itens, fazendo uso do valor (R$), por exemplo.

Certamente que modelo desenvolvido também pode ser aperfeiçoado, melhorando os resultados por meio de:

a) Melhoria nos Rótulos:

Esta melhoria foi muito perceptível ao longo do desenvolvimento em cada iteração do "rotulador" de dados em SQL, principalmente depois de comparar uma amostra de casos em que o rótulo era distinto do previsto pelo modelo, sendo que cada iteração melhorava o desempenho do modelo classificador.

Também não se pode deixar de mencionar que erros no rótulo podem ser "aprendidos" pelo modelo e repetidos no classificador. Embora esta probabilidade tenha sido reduzida significativamente ao longo das dezenas de iterações, ela ainda por estar presente e pode ser mitigada ainda mais com o uso da aplicação e validação posterior.

b) Parametrização do Modelo:

Tanto a etapa de tokenização e normalização quando o treinamento do modelo LSVM, foram realizados com os parâmetros *default* da classe. É possível que novos parâmetros possam melhorar o desempenho do classificador, principalmente com o envolvimento de especialistas em ML mais experientes.

c) Outro Modelo de ML:

Futuros desenvolvimentos com modelos de NLP mais complexos, fazendo uso de redes neurais, por exemplo, podem melhorar o desempenho da solução. Ou mesmo algoritmos de tokenização mais adaptados à língua portuguesa.

Por fim, ainda que se visualize uma longa estrada de possíveis melhorias para a solução apresentada, espera-se que este trabalho tenha obtido êxito em demonstrar a viabilidade do uso desta tecnologia ao descrever um caso prático de uso de IA na Administração Pública.

Referências Bibliográficas

AWS (2022). **What is data labeling?** Amazon Web Services. Disponível em <https://aws.amazon.com/sagemaker/data-labeling/what-is-data-labeling>. Acesso em 30 mai. 2022.

BRASIL (1993). **Lei n° 8.666/1993, de 21 de junho de 1993**. Presidência da República. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l8666cons.htm>. Acesso em: 16 mai. 2022.

IBM (2022a). **Artificial Intelligence (AI).** IBM - International Business Machines Corporation. Disponível em <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>. Acesso em 21 mai. 2022.

IBM (2022b). **Machine Learning e Ciência de dados com IBM Watson**. IBM - International Business Machines Corporation. Disponível em <https://www.ibm.com/br-pt/analytics/machine-learning>. Acesso em 21 mai. 2022.

IBM (2022c). **Natural Language Processing (NLP**). IBM - International Business Machines Corporation. Disponível em <https://www.ibm.com/cloud/learn/natural-language-processing>. Acesso em 08 set. 2022.

IBM (2022d). **Supervised vs. Unsupervised Learning: What's the Difference?** IBM - International Business Machines Corporation. Disponível em <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning>. Acesso em 21 mai. 2022.

KORSTANJE, J. (2021). **The F1-score.** Towards Data Science. Disponível em <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>. Acesso em 20 ago. 2022.

LSVM (2022). **Linear Support Vector Classification.** Scikit-learn - Machine Learning in Python. Disponível em <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>. Acesso em 20 ago. 2022.

MCARTHY, J. (2004). **What is Artificial Intelligence?** Stanford University. Disponível em <https://borghese.di.unimi.it/Teaching/AdvancedIntelligentSystems/Old/IntelligentSystems_2008_2009/Old/IntelligentSystems_2005_2006/Documents/Symbolic/04_McCarthy_whatisai.pdf>. Acesso em 27 mai. 2022.

MANNING C. D.; RAGHABAN P., SCHÜTZE H. (2009). **An Introduction to Information Retrieval.** Cambridge University Press. Cambridge, England. p. 22. Online edition (c) 2009. Disponível em <https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf>. acesso em 19 ago. 2022.

MELLO, C. A. B. (2011). **Curso de direito administrativo**. 28. ed. São Paulo: Malheiros, 2011, p. 528.

TF-ID (2022). **TfidfTransformer**. Scikit-learn - Machine Learning in Python. Disponível em <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer>. Acesso em 20 ago. 2022.

TURING, A. M. (1950). **Computing Machinery and Intelligence.** Disponível em <https://www.csee.umbc.edu/courses/471/papers/turing.pdf>. Acesso em 15 mai. 2022.

UC-BERKELEY (2020). **What Is Machine Learning (ML)?** UC Berkeley School of Information. Disponível em <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>. Acesso em 20 ago. 2022.

CURRÍCULOS

Glauber Volkmer é Mestre em Administração (2018) pela Universidade Federal da Bahia (UFBA). Especialista em Controle Público (2017) pelo Instituto de Contas do Tribunal de Contas do Estado de Santa Catarina (ICON TCE/SC) e Criptografia e Segurança em Redes (2008) pela Universidade Federal Fluminense (UFF). Bacharel em Ciências da Computação (2001) pela Universidade Federal de Santa Catarina (UFSC). Auditor Federal de Finanças e Controle da Controladoria-Geral da União desde 2006.

Contato: glauber.volkmer@cgu.gov.br

Maxwell Sarmento de Carvalho é Mestre em Ciência da Computação (2021) pela Universidade de Brasília (UnB). Especialista em Ciência de Dados e Big Data (2021) pela Pontifícia Universidade Católica de Minas Gerais (PUC Minas). Bacharel em Administração de Empresas (2009) pela Universidade Federal de Minas Gerais (UFMG). Coordenador do Núcleo de Inteligência Analítica e Apoio à Decisão da SMA/ANEEL. Trabalha na ANEEL desde 2011.

Contato: maxwellcarvalho@aneel.gov.br