UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS DOIS VIZINHOS
CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS

LUCAS DYBAX DE ANDRADE

CLASSIFICAÇÃO DA FAIXA DE PESO DE PRODUTOS COM DEEP LEARNING E BERT

TRABALHO DE CONCLUSÃO DE CURSO

DOIS VIZINHOS
2022

LUCAS DYBAX DE ANDRADE

CLASSIFICAÇÃO DA FAIXA DE PESO DE PRODUTOS COM DEEP LEARNING E BERT

CLASSIFICATION OF PRODUCTS WEIGHT RANGES USING DEEP LEARNING AND BERT

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de Especialista em Ciência de Dados.

Orientador: Prof. Dr. Rafael Gomes Mantovani

DOIS VIZINHOS
2022

LUCAS DYBAX DE ANDRADE

CLASSIFICAÇÃO DA FAIXA DE PESO DE PRODUTOS COM DEEP LEARNING E BERT

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de Especialista em Ciência de Dados.

Data de aprovação: 05/novembro/2022

Rafael Gomes Mantovani
Doutorado
Universidade Tecnológica Federal do Paraná - Câmpus Apucarana

Jefferson Tales Oliveira
Doutorado
Universidade Tecnológica Federal do Paraná - Câmpus Pato Branco

Rafael Alves Paes de Oliveira
Doutorado
Universidade Tecnológica Federal do Paraná - Câmpus Dois Vizinhos

DOIS VIZINHOS
2022

Dedico este trabalho aos meus pais André e Dionise por todo o apoio em minha trajetória acadêmica e profissional.

AGRADECIMENTOS

Gostaria de agradecer aos meus pais, André Ferreira de Andrade e Dionise Maria Dybax de Andrade, que me apoiaram durante toda minha trajetória acadêmica e profissional. Além disso, me educaram com ótimos valores e me instigaram a atingir metas as quais jamais achei que seriam possíveis.

À Universidade Tecnológica Federal do Paraná, pela oportunidade de cursar a Especialização em Ciência de Dados, enriquecendo muito meus conhecimentos teórico e técnico sobre diversas vertentes da área de dados, com professores extremamente qualificados e sempre prontos à sanar quaisquer dúvidas.

Ao meu orientador, professor Rafael Gomes Mantovani, por ter me apoiado durante toda esta jornada do curso, principalmente na realização deste trabalho de conclusão, me mostrando sempre a intuição científica e os conceitos-chave necessários para a implementação do meu estudo.

Por fim, a todos os meus professores e colegas que fizeram parte do período em que estive estudando na instituição, compreendido de Março de 2021 a Outubro de 2022.

Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável para aprender a conhecer a beleza libertadora do intelecto para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer. (Albert Einstein)

RESUMO

Dado o crescimento exponencial do comércio virtual nas últimas décadas e da criação de plataformas de *marketplace*, as operações logísticas cada vez mais buscam automatizar e melhorar a previsibilidade de entrega e custos de seus itens. Um problema encontrado pelas plataformas que oferecem esse serviço é a correta estimação do peso-frete de um produto, informação que afeta desde o planejamento logístico de empacotamento até entrega por algum veículo no endereço do cliente. Com muitos vendedores cadastrando seus itens erroneamente, dimensões erradas geram ineficiências operacionais em entregas. As soluções de aprendizado de máquina servem, nesse contexto, para utilizar dados de itens já vendidos e aferidos a fim de categorizar aqueles que ainda não foram: espera-se que produtos similares tenham características de peso e dimensões próximas. Dada a insurgência recente de classificadores de aprendizado profundo, como o BERT, e tornando-se estado da arte em problemas de classificação textual, foram propostos experimentos para avaliar a aplicabilidade dos mesmos no problema de estimativa de peso de itens de entrega. Experimentos foram realizados comparando BERT com soluções já existentes baseadas em aprendizado de máquina tradicional em um ambiente de produção de uma empresa varejista. Os modelos propostos foram avaliados utilizando dados com e sem pré-processamento, etapa que é comum a resolução de problemas baseados com redes neurais artificiais. Os resultados obtidos mostraram um desempenho preditivo superior dos classificadores baseados em BERT quando comparado aos modelos tradicionais. No entanto, o valor de acurácia balanceada de 0.63 obtido pelo melhor classificador, mesmo que superior à todas *baselines*, indica que há muito espaço para melhorias antes de que a solução seja factualmente implementável. A análise das predições errôneas do modelo indicam que uma melhor etapa de pré-processamento dos dados textuais, diretamente alinhado à definição do problema, seria útil para melhorar o desempenho preditivo dos modelos induzidos.

Palavras-chave: Classificação de textos. Processamento de Linguagem Natural. BERT. Logística operacional.

ABSTRACT

Given the exponential growth of e-commerce in recent decades and the creation of *marketplace* platforms, logistics operations increasingly seek to automate and improve the predictability of delivery and costs of their items. A recurrent problem encountered by the platforms that offer this service is the correct estimation of the freight weight of a product, an information that affects everything from the logistical planning of packaging to last-mile delivery to the customers' addresses. With many sellers wrongly listing their items, relying on wrong dimensions lead to operational inefficiencies in delivery. Machine learning solutions serve, in this context, to use data from items already sold and measured to categorize new itens that have not been sold yet: similar products are expected to have similar weight, dimensions and characteristics. Given the recent insurgency of BERT classifiers as a state of the art in textual classification problems, different experiments were proposed to evaluate their applicability in the weight estimation problem. Experiments were performed comparing BERT against existing solutions based on traditional machine learning in the production environment of a retail company. The proposed models were evaluated using data with and without the preprocessing, a pipeline step common to solutions based on artificial neural networks. The obtained results suggest a superior performance of the BERT-based classifiers compared to traditional models. However, the balanced accuracy of 0.63 in the best classifier, even being superior to all *baselines*, indicates that there is much room for improvement before the solution is factually implementable. The study of the missclassified instances also indicates that better data preprocessing, directly aligned with the problem definition, would be useful to improve the performance of the best estimator.

Keywords: Text Classification. Natural Language Processing. BERT. Operational Logistics.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
BAC	*Balanced Accuracy by Classes* ou Acurácia Balanceada por Classes
CNN	*Convolutional Neural Networks* ou Redes Neurais Convolucionais
CV	*Cross Validation* ou Validação Cruzada
DL	*Deep Learning* ou Aprendizado Profundo
DNN	*Deep Neural Networks*
DT	*Decision Tree* ou Árvore de Decisão
EDA	*Exploratory Data Analysis* ou Análise Exploratória de Dados
FN	Falsos Positivos
FP	Falsos Negativos
GS	*Grid Search* ou Busca em Grade
HP	Hiperparâmetro
IA	Inteligência Artificial
IDF	Frequência Inversa de Documentos
K-NN	*K Nearest Neighbors* ou K - Vizinhos mais Próximos
LSTM	*Long Short Term Memory*
ML	*Machine Learning*
MLP	*Multilayer Perceptron* ou Redes Neurais Multicamadas
NB	Naive Bayes
NLP	*Natural Language Processing* ou Processamento de Linguagem Natural
PLN	Processamento de Linguagem Natural
ReLU	*Rectified Linear Unit* ou Unidade Linear Retificado
RF	*Random Forest* ou Floresta Aleatória
RL	Regressão Logística

RS	*Random Search* ou Busca Aleatória
SGD	Gradiente Descendente Estocástico
TF	Frequência de Termos
TF-IDF	Frequência de Termos versus Frequência Inversa de Documentos
TM	*Text Mining* ou Mineração de Textos
UTFPR	Universidade Tecnológica Federal do Paraná
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos

SUMÁRIO

1 INTRODUÇÃO

A digitalização do comércio varejista trouxe uma nova camada de problemas relacionados à logística dos produtos. Antes, a cadeia de processos terminava geralmente na chegada do produto à loja. Entretanto, agora o processo de transporte de um produto pode ligar todas as pontas da jornada da compra, desde a indústria ao consumidor final. Um produto pode sua origem em uma loja, centro de distribuição, indústria ou de um depósito ao consumidor, tendo soluções distintas utilizadas em cada um dos métodos possíveis. Com essa digitalização, há uma grande quantidade de dados sendo gerada em transportadoras, pelos correios, em sistemas ERP (do inglês, *Enterprise Resource Planning*, sistemas focados na gestão de recursos) e similares. Como consequência, os processos de logística estão sendo documentados digitalmente, muitas vezes em sistemas transacionais, e em ambientes de fácil recuperação de conteúdo.

Falhas nesse processo geralmente acontecem quando os dados necessários para execução são de má qualidade, como no caso das dimensões de produtos cadastradas erroneamente, gerando custos e planejamento de frete errados. O erro no cadastro de produtos é problema comum principalmente em plataformas que oferecem o serviço de *marketplace*, no qual vendedores terceirizados utilizam de uma plataforma e sistema logístico centrais para divulgação e distribuição de seus produtos. Uma maneira comumente utilizada para mitigar estes erros é a classificação de um produto em determinada faixa de peso, utilizando dados de produtos similares já vendidos e aferidos.

Um dos possíveis caminhos para a classificação destes itens é suportado pela extração de suas **características textuais**, como título e descrição do produto. Para isso, são utilizadas técnicas da área de **Processamento de Linguagem Natural (PLN)** (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), que alia conhecimentos de Ciência da Computação, Mineração de Dados, Linguística, entre outros diversos subtópicos. Iniciada com a algoritmização determinística de regras linguísticas, atualmente a PLN adota também conceitos de Inteligência Artificial (IA), como Aprendizado de Máquina (AM) e Aprendizado Profundo (*Deep Learning) - DL*, para obtenção de seus objetivos (MARSLAND, 2015).

1.1 Problema de Pesquisa

O problema a ser estudado neste trabalho é a classificação de itens em faixas de peso, utilizando seus títulos como características textuais. Na literatura, este tipo de problema pode ser generalizado como um caso em que há **classificação textual**. Existem diversos estudos abordando esse tipo de problema, com diferentes tipos de arquiteturas e soluções sendo aprimorados nos últimos anos (Wu et al., 2016; Devlin et al., 2018; GONZÁLEZ-CARVAJAL; GARRIDO-MERCHÁN, 2020). Geralmente, o método empregado está associado com a tarefa de classificação de sentimentos (quando se adquire um produto) mas pode ser replicado para

qualquer tipo de classificação supervisionada. A sequência de etapas necessária para solução dos problemas, chamada de *pipeline*, segue sempre uma estrutura semelhante: coleta de dados, extração de características, seleção de modelo(s) classificatório(s) e posterior avaliação de desempenho(MITCHELL, 1997).

Existem trabalhos que abordam o tema desenvolvidos com base em Aprendizado de Máquina (AM) tradicional (MARSLAND, 2015) e, mais recentemente, em Aprendizado Profundo (*Deep Learning* - DL) (AGGARWAL, 2018). Há, atualmente, diversos trabalhos comparativos entre as duas abordagens, visto que o estado da arte para processamento textual parece apoiar-se hoje majoritariamente em modelos e transformadores embasados em DL, assim, as soluções baseadas em AM tradicional atuam como *baselines*. Para o problema específico de produtos em logística, não existe vasta literatura formal. Considerando um cenário em que já existe uma solução em produção que utiliza métodos tradicionais de AM para categorização de produtos, este trabalho investiga a viabilidade de uma nova solução baseada no uso de DL, mais especificamente em transformadores `BERT` (Devlin et al., 2018), do inglês *Bidirecional Encoder Representations from Transformers*.

1.2 Justificativa e Contribuições

Considerando que a classificação de produtos em faixas de peso generalizadas é de grande importância na mitigação de erros operacionais relacionados a frete, a automatização dessa solução é um passo lógico a ser tomado. Para este cenário, considera-se que já existe um *pipeline* de AM tradicional em execução e operação em uma loja de varejo, e com o uso desta incorrem diversas restrições computacionais e operacionais. Como forma de aperfeiçoamento do processo existente, propõe-se a criação de um novo modelo que use das soluções mais robustas do atual estado da arte.

Quando há necessidade de classificação textual, comumente também existe a necessidade de testar diversos tipos de modelos e escolher algum que tenha bom desempenho preditivo. As vezes é comum também separar um problema de classificação multi-classe em diversos problemas menores, a fim de especificar mais os conjuntos de dados trabalhados (por exemplo, segregar itens de uma mesma categoria em um conjunto separado e criar um preditor específico desta) e melhorar a performance das soluções. Dessa prática, derivam diversos tipos de problemas, sendo os principais a dificuldade de manutenção de código para situações que dependem de muitos modelos e o tempo de treinamento exigido para treinar e testar um grande volume de problemas fracionados. Mais recentemente, a crescente adoção do uso de transformadores `BERT` em arquiteturas de redes neurais tem demonstrado resultados excepcionais em todos os tipos de problema de classificação de texto, sem necessidade da manutenção de diversas soluções simultâneas. Assim, tais soluções globais, que trabalham com todas as categorias simultaneamente, serão utilizadas como base para o presente trabalho.

1.3 Objetivos

O objetivo principal deste trabalho é desenvolver um modelo único, baseado em DL, que classifique os produtos de um *marketplace* em faixas de peso, sem a necessidade de segregação ou especificação dos dados. Tal solução é necessária pela dificuldade de manutenção dos diversos modelos que hoje empregam-se em produção.

Especificamente, objetiva-se:

- Classificar dados textuais utilizando técnicas de aprendizagem profunda;
- Aplicar abordagens recentes de PLN para extração de características dos dados textuais;
- Comparar o desempenho da solução com aquelas que foram anteriormente adotadas, em cenários semelhantes de treino e teste;

1.4 Organização do Trabalho

O trabalho segue a seguinte organização: o Capítulo 2 discorre sobre os fundamentos teóricos utilizados para o desenvolvimento do trabalho; o Capítulo 3 apresenta algumas das soluções já utilizadas para a área de mineração e classificação de textos presentes na literatura, assim como estudos comparativos na mesma abordagem da solução aqui proposta; o Capítulo 4 descreve a metodologia experimental aplicada no trabalho; o Capítulo 5 apresenta e discute os principais resultados obtidos após a realização dos experimentos; e por fim, as considerações finais e propostas de trabalhos futuros são abordadas no Capítulo 6.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, é apresentada a fundamentação teórica necessária para compreensão dos experimentos realizados neste trabalho. Em primeiro momento, serão discorridos conceitos sobre Processamento de Linguagem Natural e Aprendizado de Máquina. Depois, serão apresentados alguns conceitos derivados da interseção das duas áreas, mais especificamente relacionados à Aprendizagem Profunda e aos transformadores utilizados nas tarefas de PNL que utilizam de redes neurais.

2.1 Processamento de Linguagem Natural

Processamento de Linguagem Natural (PLN) (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), do inglês *Natural Language Processing* (NLP), é uma área de pesquisa iniciada por volta da década 1950 como interseção de Inteligência Artificial e Linguística. A PNL começou como algo diferente da extração de características textuais, mas ambas áreas convergiram ao longo do tempo. Além disso, tópicos de análises gramáticas e léxicas foram integrados à área. A natureza não restrita e imensa da PLN levou a área a um uso cada vez maior de técnicas estatísticas, em detrimento do que antes era escrito em regras manuais.

Nesse âmbito, quando a PNL é associada a métodos modernos de AM, é comum que exista uma etapa de pré-processamento textual que combine lexicografia, estatística e computação, objetivando extrair o melhor significado possível de cada palavra ou sentença digitada (HADDI; LIU; SHI, 2013). Algumas técnicas comumente aplicadas são:

- **Uniformização textual:** uniformiza o texto em questões de escrita, geralmente normalizando os dígitos em um tipo de caixa (alta ou baixa). A depender do problema, não há necessidade de atribuir significados diferentes a uma mesma palavra escrita de maneiras diferentes (como a inicial em maiúsculo), por isso a implementação desta etapa.
- **Remoção de ruídos:** remove dados do texto que não possuem relevância na extração de características. Quais dados serão removidos dependem do escopo do problema. Exemplos clássicos são espaços excessivos, pontuação, tags HTML, números, etc..
- **Remoção de palavras vazias** (*stopwords*): remove termos que não possuem valor semântico no caso de extração de valores por frequência. São artigos, pronomes, preposições, que fazem parte da estrutura gramatical das frases, com altas frequências de aparição em textos. Esses termos são desnecessários para extração de características justamente por não serem discriminativos;
- **TF-IDF:** alguns autores descrevem essa etapa como uma etapa de *filtragem*, ao contrário das técnicas anteriores que seriam processos de *transformação*. Nesse tipo de filtragem, são atribuídos pesos de acordo com a frequência dos termos no documento analisado,

ponderados pela frequência do mesmo termo em todos os documentos analisados no conjunto de dados. Tal ponderação ocorre para dar mais relevância a termos que apareçam mais vezes em um documento específico e, ao mesmo tempo, menos no contexto geral, pois indicam palavras mais determinantes na classificação daquele documento (ou tipo de documento). Palavras que aparecem muito em todos os documentos, independentemente de categoria, teriam importância reduzida, já que são mais frequentes e tem pouco poder discriminativo.

2.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM) (SAH, 2020), em inglês *Machine Learning* (ML) é uma área de estudos que desenvolvem programas computacionais, baseados em algoritmos específicos, que aprendem e melhoram com base na experiência. É geralmente visto como uma sub-área da Ciência de Dados, pois contribui com muitos algoritmos preditivos, e técnicas de experimentação. Os algoritmos de AM permitem a decisão autônoma dos sistemas, sem interferência humana, a partir da descoberta de padrões ocultos em dados complexos. Existem diversos tipos de algoritmos, categorizados primariamente pela maneira que o aprendizado é conduzido, pelo tipo de dados manipulado, e pelo tipo de problema resolvido. As categorias mais tradicionais são: algoritmos supervisionados; não-supervisionados e; de reforço. Há soluções híbridas entre essas macro-categorias. Para o desenvolvimento do presente trabalho, o contexto do problema aproxima-se das soluções providas por aprendizagem supervisionada.

2.2.1 Aprendizagem Supervisionada

O aprendizado supervisionado ocorre quando os dados estão em formato de variáveis de entrada e têm definido(s) valor(es) esperado(s) de saída. O algoritmo aprende a função de mapeamento dos valores de entrada à saída esperada. Para isso, é necessário ter dados de saída o suficiente para que a melhor função possível seja aprendida, por isso o nome *supervisionado*, visto que são aprendidos padrões a partir de dados existentes no conjunto entradas-saídas. A aprendizagem supervisionada pode ser subdividida em tarefas de classificação e problemas regressão.

- **Tarefas de Classificação:** a variável de saída é de um (ou mais) valor categórico conhecido, como "positivo"ou "negativo";
- **Tarefas de Regressão:** a variável de saída é um valor real ou contínuo. Por exemplo, uma "quantidade"ou "preço".

Existem diversos algoritmos utilizados na aprendizagem supervisionada (NASTESKI, 2017), os mais utilizados nos trabalhos de classificação textual:

- **Árvores de decisão:** classificadores expressos como uma partição recursiva do espaço de instância. É formado por nós que sucessivamente dividem o espaço de instância em duas ou mais partes, de acordo com uma função discreta aplicada aos dados de entrada. Essa divisão continua sendo transmitida dentre nós sucessivos até um ponto em que as amostras contidas nesse nó sejam declaradas como representativas de determinada classe. Assim, esse nó passa a ser uma "folha". Todas as amostras novas são posteriormente alimentadas na árvore, percorrem os sub-espaços de decisão e acabam em determinada folha, sendo então classificadas com o valor atribuído a esta durante o treinamento.
- **Naïve Bayes:** a classificação bayesiana é um método supervisionado e estatístico baseado no Teorema de Bayes. É assumido um modelo probabilístico que permite a captura da incerteza ao determinar probabilidades dos resultados e assumir a independência entre os preditores. Por sua natureza probabilística, por muitos anos este modelo esteve à frente do estado da arte em tarefas ligadas à classificação de textos.

2.2.2 Aprendizagem Profunda e Redes Neurais Artificiais

A aprendizagem profunda, do inglês *Deep Learning* (VARGAS; MOSAVI; RUIZ, 2017), surgiu como um novo campo de pesquisa na subárea da AM. o DL possui duas características principais: processamento não-linear em múltiplas camadas ou estágios; seguida de uma etapa de aprendizagem supervisionada, podendo ser não-supervisionada, ou uma mistura de ambas. A primeira característica refere-se à uma camada de processamento ter a saída da camada anterior como sua entrada, utilizando de uma hierarquia entre as camadas para determinar a importância dos dados como úteis ou não. O segundo ponto é habilidade de explorar esta características de processamento distribuído em diferentes tarefas de aprendizado supervisionado.

O conceito de DL está intrinsecamente relacionado às **Redes Neurais Artificiais (RNAs)** (OTTER; MEDINA; KALITA, 2020). Nestas, existem diversos nós (neurônios) interconectados, recebendo entradas e fornecendo saídas. Cada neurônio nas camadas de saída realiza operações de somas ponderadas nos valores recebidos em suas entradas e gera saídas usando funções de transformações simples. Os pesos dessas funções são reajustados de acordo com os erros que cada exemplo sofre no processo de saída da rede, utilizando geralmente processos de correção conhecidos como retro-propagação (usando as derivadas dos erros a cada nó). As redes são comumente distinguidas por como os nós se conectam e pela numerosidade de camadas arquitetadas. Algumas arquiteturas comuns a redes neurais são as seguintes:

- **Redes Neurais Diretas:** redes básicas, nas quais todos nós são organizados em camadas sequenciais e recebem entradas somente de nós das camadas anteriores;
- **Redes Neurais Convolucionais:** recebem esse nome em derivação das operações convolucionais da matemática e processamento de sinais. Elas usam funções (filtros) para análise simultânea de diferentes atributos nos dados. Geralmente permite a captura de dependências espaciais e temporais nos dados;

- **Redes Neurais Recursivas:** esse tipo de rede compartilham os pesos para reajuste verticalmente (entre as camadas). O nome advém da aplicação recursiva desse mesmo conjunto de pesos nas entradas anteriores. Possui bastante utilização em tarefas de PLN;
- **Redes Neurais Recorrentes e LSTMs:** são tipos de desdobramentos de redes recursivas. Também muito utilizadas em PLN. As redes recorrentes possuem "memória" de elementos anteriores ao processar novos exemplos, considerando por exemplo, a palavra anterior ao definir pesos para a atual. As Redes de Memória de Curto Prazo Longo (do inglês, LSTMs) são redes recorrentes nos quais os nós objetivam reter, esquecer ou expor informações específicas de curta ou longa duração. Como as redes recursivas comuns tendem a ter os padrões de curto prazo diluídos no longo prazo, as LSTMs fornecem um mecanismo de retenção de informações importantes em ambos horizontes de tempo (curto e longo), enquanto informações irrelevantes no longo prazo podem ser esquecidas.
- **Mecanismos de Atenção e Transformadores:** para tarefas que envolvam a saída de textos, é comum o uso de pares de codificadores e decodificadores. Uma rede neural de codificação produz um vetor de valores com tamanho pre-determinado, ao passo que uma rede de decodificação retorna um texto de tamanho variável baseado neste vetor. A codificação comum, no entanto, acontece simplesmente transformando uma sequência em um vetor de tamanho limitado, sem consideração de quais características do texto são mais importantes para a rede. A partir desse problema, surgiram os mecanismos de atenção, utilizados para garantir que a rede "saiba" para quais características observar durante a priorização de importância na codificação da entrada. Esse processo é feito olhando os estados ocultos da rede durante a propagação dos exemplos e atribuindo pesos conforme adequado. Um transformador é um comitê de modelos de codificação-decodificação em sequência, com instâncias de atenção própria em cada codificador/decodificador e atenção cruzada entre codificadores e decodificadores. Assim, conexões entre palavras ou frases são geralmente melhor aprendidas e representadas nos vetores de características do texto. Transformadores são utilizados extensivamente nas soluções de estado da arte para tarefas de PLN.

2.2.3 BERT

Um desdobramento de transformador que extensivamente vem ganhado espaço por atingir as melhores medidas de desempenho para problemas de PNL é o *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2018). O BERT é um tipo de transformador pré-treinado em vastos vocabulários de palavras. O principal diferencial deste para outros transformadores é que ele não está preso à restrição de ser treinado unidirecionalmente em contexto, da esquerda para direta ou na direção contrária. Nele, os mecanismos de atenção permitem que haja extração bidirecional de contextos, observando palavras anteriores e posteriores à atual da série.

Existem duas arquiteturas de BERT pré treinadas:

- uma chamada *base*, com 12 blocos de transformadores, tamanho oculto de 768 e 12 *heads* de auto-atenção (nomenclatura dada aos módulos de atenção executados em paralelo), totalizando 110M de parâmetros;
- a segunda, BERT *large*, possui 24 blocos transformadores, tamanho oculto de 1024 e 16 *heads* de auto-atenção, totalizando 340M de parâmetros.

A entrada do BERT é denominada "sequência", e precisa ser realizada em forma de *tokens*: pode ser uma sentença ou um par de termos no formato (pergunta, resposta). A tokenização utilizada é a *WordPiece Embedding* (Wu et al., 2016) com algumas alterações. Esse tipo de incorporação leva em consideração um token, sua posição e segmento para atribuir valores numéricos às sentenças. O pré-treino do BERT é realizado em duas tarefas. A primeira, chamada pelos autores de *masked LM*, trata-se da predição de palavras no meio de uma sentença: parte aleatória dos tokens da sequencia são mascarados e o modelo tenta predizer qual palavra está no lugar da máscara. A segunda tarefa é uma predição de próxima sentença, na qual o transformador precisa responder se a sentença apresentada posteriormente à atual é realmente a próxima sentença do texto. Existe ainda uma outra etapa opcional de ajuste fino, na qual o modelo pode ser treinado com entradas e saídas diretamente relacionadas ao problema contextualizado para o último ajuste de pesos.

2.2.4 Desbalanceamento de Classes

Um problema comum a tarefas de classificação é o desbalanceamento de classes. A maior parte dos algoritmos assume uma distribuição relativamente balanceada dentre o número de exemplos de cada classe do problema original. Em casos de desbalanceamento, essa distribuição não se verifica e existem muito mais exemplos de determinada classe (ou classes) frente às outras. Como há menos amostras das classes mais raras, as regras de classificação destas costumam ser rasas, não descobertas ou ignoradas. Assim, exemplos de teste que pertençam a essas classes têm maior taxa de erro em suas classificações que aqueles pertencentes à classe dominante. A depender do problema, predizer corretamente dados pertencentes às classes minoritárias é mais importante que a predição dos casos comuns (SUN; WONG; KAMEL, 2011). Nesses casos, existem algumas métricas de desempenho ajustadas a partir do balanceamento da frequência de cada classe no conjunto de teste (a serem explicadas mais à frente).

Grande parte dos algoritmos de AM tradicionais têm desempenho drasticamente reduzido quando submetidos a um cenário de desbalanceamento de classes. Técnicas comuns para resoluções deste problema são o *resampling* dos conjuntos de treino, de maneira a apresentar amostras balanceadas na fase de aprendizado do algoritmo, e o uso de algoritmos atrelados à técnica de *boosting*, que visa associar pesos maiores às amostras que possuem maior importância na identificação.

2.2.5 Métricas de desempenho

As métricas de desempenho são utilizadas para avaliar o desempenho de um algoritmo na tarefa para a qual ele foi treinado. Este teste é feito em amostras nunca "vista" anteriormente, seja na etapa de treinamento ou validação, de modo a simular um caso real. Caso a etapa de teste não seja conduzida, há a chance do algoritmo acertar as predições justamente por já ter visto aquele exato problema anteriormente. Em tarefas de classificação, como existem finitas categorias de classe, é possível obter métricas a partir de conceitos de **acertos** ou **erros** em cada classe de predição. Nesses casos, não há ordenação contínua de quão longe ou perto do valor real a predição estava, apenas avaliação categórica de se ela foi certa ou não.

Algumas terminologias importantes de serem relembradas são as seguintes:

- **Verdadeiro Positivo (VP)**: amostras preditas como de determinada classe e realmente pertencentes à mesma;
- **Falso Negativo (FN)**: amostras preditas como pertencentes a outras classes, mas que na realidade pertencem à classe atual sendo avaliada;
- **Falso Positivo (FP)**: amostras que foram preditas como pertencentes à classe atualmente avaliada, mas que na realidade pertencem às outras classes do problema;
- **Verdadeiro Negativo (VN)**: todas as amostras preditas como pertencentes às outras classes e que de fato o são (em um problema multi-classe, mesmo que haja erro na classe final predita pelo algoritmo, desde que ele ao menos acertado que não faz parte da classe atual de análise, é contabilizado como verdadeiro negativo.

Essas terminologias são originárias de problemas de classificação binários, mas seus conceitos são facilmente deriváveis à classificação multi-classe, considerando que avaliamos cada classe isoladamente frente às restantes. Dito isto, algumas métricas podem ser derivadas a partir da interação destas quatro categorias. Dentre as mais utilizadas em problemas de classificação de textos, estão as seguintes:

- **Acurácia:** Pode ser abstraída como a porcentagem de acertos de um modelo. Consiste da razão da quantidade de acertos de um modelo (VP + VN) em relação ao total de predições realizadas (VP + VN + FP + FN). No caso de problemas multi-classe, os acertos são computados como a soma de todos os verdadeiros positivos de cada classe. A equação 1 apresenta o cálculo em um problema binário;

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

- **Acurácia Balanceada por Classe**, originalmente *Balanced Accuracy per Class* (BAC): calculada pelo somatório do total de acertos em determinada classe em relação a todos os exemplos da mesma classe, para cada classe possível, dividida pelo número total de classes do problema. É uma medida útil em problemas desbalanceados, pois leva em conta o acerto em cada classe versus a frequência total da classe no momento da avaliação.

Assim, classes com taxas de acerto muito baixas reduzem fortemente o desempenho final, mesmo se a classe majoritária for predita corretamente. A Equação 2 apresenta a fórmula desta métrica para um problema binário;

$$Acurácia\ Balanceada = \frac{1}{2}\left(\frac{VP}{VP+FN} + \frac{VN}{VN+FP}\right) \tag{2}$$

- **Medida F1**, do inglês *F1 Score*: uma métrica que avalia o desempenho de cada classe de maneira mais ponderada. Considerando a performance isolada de cada classe, existem duas medidas importantes: a taxa de verdadeiros positivos e o valor preditivo de positivos. A primeira consiste do total de acertos da classe (VP) em relação ao total de exemplos da mesma classe (VP + FN). Essa taxa é mais comumente conhecida como revocação (R) na literatura. A segunda é definida como precisão (P), pois denota o total de acertos de determinada classe (VP) em relação ao total de predições da mesma classe (VP + FP). A medida F1 representa a média harmônica entre revocação e precisão, conforme a Equação 3 apresenta. Como esse tipo de média dentre dois números tende a se aproximar do menor, valores mais altos para F1 indicam que tanto precisão quanto revocação tendem a estarem altos.

$$F1 = \frac{2}{1/R + 1/P} \tag{3}$$

2.2.6 Ajuste de Hiperparâmetros

Modelos de AM geralmente possuem dois tipos de variáveis intrínsecas: parâmetros e hiperparâmetros. Parâmetros são variáveis aprendidas durante a etapa de treinamento e internas ao modelo. São eles que "direcionam" as entradas a seus resultados de saída. Em uma regressão linear, por exemplo, os coeficientes independentes e dependentes que melhor se ajustam à distribuição dos dados são parâmetros. Hiperparâmetros são externos aos modelos e definidos antes da etapa de treinamento: eles definem como o modelo irá se comportar durante a etapa de aprendizado. O tamanho máximo de uma árvore de decisão, ou o número de árvores em um comitê são alguns exemplos. Como os hiperparâmetros não são aprendidos na etapa de treinamento, não é possível "acertá-los" com o uso de dados. Por algum tempo, esse ajuste foi feito em bases de tentativa e erro. Posteriormente, surgiram técnicas para otimização desses hiperparâmetros (Liashchynskyi; Liashchynskyi, 2019). Estas técnicas usam a divisão de dados em treino/teste para estimar diferentes configurações de hiperparâmetros. Além disso, retornam a melhor configuração após realizar um número de avaliações durante a busca em um espaço de possibilidades. Existem duas técnicas mais simples para essa tarefa, diferindo-se na maneira que o espaço de busca é definido e testado:

- **Busca em Grade**, ou *Grid Search* (GS): são definidos possíveis conjuntos de valores para os hiperparâmetros, em espaços igualmente intervalados, e posteriormente avaliadas todas as possíveis combinações desses valores. O custo computacional dessa tarefa cresce

exponencialmente conforme maior o espaço de busca, visto que todas as possibilidades são testadas. Pode-se gerar ineficiências pela demora em resposta do melhor modelo e ainda assim ficar preso a um ótimo local;

- **Busca Aleatória**, ou *Random Search* (RS): é definido pelo usuário um espaço de possíveis valores para os parâmetros, e a partir deles são testadas aleatoriamente distintas combinações de hiperparâmetros, respeitando um intervalo de distâncias e de tentativas máximas definidas antecipadamente. Como são testes aleatórios no espaço, é mais provável que a solução final não esteja presa a um ótimo local.

3 TRABALHOS RELACIONADOS

Neste capítulo, são sintetizados alguns dos trabalhos relacionados à tarefa de classificação de textos, com ênfase naqueles que realizaram estudos comparativos de desempenho dentre modelos tradicionais de PNL e soluções que implementam o BERT. Estes trabalhos serão discutidos nas próximas seções.

3.1 Soluções com BERT e/ou comparativas

Qasim et al. (2022) se propôs a estudar como diferentes arquiteturas de BERT, disponibilizadas para uso como modelos pré-treinados), desempenhavam em três tarefas classificatórias: detecção de *fake news* sobre a COVID-19; classificação de textos extremistas; e a separação de *tweets* sobre a COVID-19 em informativos e não-informativos. Foi utilizada uma abordagem de *transfer learning*. O *pipeline* utilizado era composto das seguintes etapas: separação das entradas em treino e validação; pré-processamento textual (remoção de url, remoção de pontuação, normalização e lematização); codificação (utilizando transformadores e extração de características via TF-IDF); treinamento e validação utilizando a incorporação de classificadores BERT. Foram utilizadas como medidas comparativas acurácia, precisão, revocação e o F-score. Em comparações com modelos tradicionais da literatura (árvores de decisão, algoritmos bayesianos, técnicas de *boosting*, os modelos BERT foram estatisticamente superiores a todos. Os modelos baseados em BERT tiveram acurácias acima de 90%, frente à média de 60% das soluções convencionais. Destacam-se os bons desempenhos dos modelos *vanilla* do BERT, as versões *large* e *base*.

Yu, Jindian e Luo (2019) e Hu et al. (2022) desenvolveram dois trabalhos diferentes sobre a classificação com BERT, mas com arquiteturas semelhantes: ambos constituem-se apenas de uma camada de entrada, seguida do codificador BERT e finalizada com uma camada de saída simples *softmax*, que classifica as probabilidades condicionais de cada categoria pré-definida. Em ambos casos, não há pré-processamento de texto anterior ao uso do BERT. Os classificadores novamente obtiveram resultados excelentes, superando aplicações de aprendizado tradicional.

??) sugeriu melhorias e testou algumas estratégias para ajuste fino do BERT em classificação de textos. Dentre as possibilidades, destacam-se o truncamento dos tokens de entrada de maneira a utilizar parte do início e parte do fim da sequência, o uso da última camada do BERT para classificação de textos e a extensão de pré-treinamento para as tarefas e domínios específicos. Com essas abordagens, foram atingidas performance de estado-da-arte em oito diferentes conhecidos datasets de classificação de texto.

González-Carvajal e Garrido-Merchán (2020) desenvolveram um trabalho comparativo entre modelos BERT e abordagens tradicionais de PNL e aprendizado de máquina (TF-IDF +

modelo). Nos experimentos, a abordagem tradicional utilizou a ferramenta *TidfVectorizer* do módulo *sklearn* para o Python 3, seguida da aplicação de um *Predictor* do módulo *automL* ou do *H2OAutoML* do módulo *h2o*, a fim de achar o melhor modelo a ser utilizado para cada tarefa. No caso do BERT, foi utilizado um modelo pré-treinado obtido com o uso do módulo *ktrain*, também para python. Este módulo é uma implementação de alto nível de soluções de DL, utilizando como base o *keras*. Em todos datasets testados (classificação de sentimentos em tweets, análise da veracidade de notícias, classificação de gêneros literários), o modelo BERT pré-treinado e com ajuste fino obteve os melhores resultados dentre todas as opções testadas.

3.2 Considerações Finais

Os trabalhos apresentados até aqui são apenas algumas das soluções e arquiteturas possíveis para a classificação de textos. O foco principal de estudos foi na implementação de algoritmos BERT, devido à caracterização inicial do problema aqui estudado: como já existem modelos treinados e testados utilizando de técnicas de aprendizagem de máquina tradicional em ambiente de produção, faz sentido lógico buscar diretamente soluções que, com vasto apoio literário, atingem os melhores níveis de performance e são estado da arte para o problema direcionado.

Assim, o direcionamento maior faz-se em **como** implementar uma solução semelhante. Como a literatura evidencia, mesmo implementações simples, sem grandes ajustes, são capazes de desempenhar ótimos resultados. Maior parte dos autores conferem esses resultados à força do *transfer learning*, ou seja, dos modelos BERT já serem extensivamente treinados em datasets linguísticos gigantescos antes de serem disponibilizados para uso.

4 METODOLOGIA EXPERIMENTAL

Neste capítulo são explicados o contexto do problema, os métodos utilizados para coleta/extração dos dados e elaboração das possíveis soluções.

4.1 Conjunto de Dados

Os dados utilizados na elaboração deste trabalho são provenientes de um sistema real de uma grande varejista digital. Foram extraídas as informações de cadastro de itens pertencentes ao *marketplace* da companhia, mais especificamente, os títulos e categorias dos produtos, e as dimensões reais de cada um destes itens após a aferição de medida. O conjunto de dados é composto apenas de produtos únicos, ou seja, um mesmo título não aparece duas vezes na lista de observações. Isso ocorre devido à natureza do problema estudado: existe, em produção, um *pipeline* que se aproveita das medidas de produtos já vendidos e replica para próximas vendas dos mesmos. Assim, sempre que um mesmo produto é vendido pela segunda vez, seja pelo mesmo ou outro vendedor, ele já possui medidas associadas e não precisa de qualquer tipo de predição.

Foram extraídos do sistema um total de 930.712 itens com aferições. A variável-alvo que se deseja prever é a "categoria de peso" do produto. Essa classe/categoria é dependente do conceito de "peso-frete", calculado com base nas informações do produto, como: altura, largura, comprimento e peso do produto. A definição de "peso-frete" é apresentada pelo Algoritmo **??**.

---

peso-volumétrico $\leftarrow$ altura $*$ comprimento $*$ largura $* 167$;
if *peso-volumétrico* ≤ 5 **then**
  peso-frete $\leftarrow$ peso;
else
  peso-frete $\leftarrow max($peso, peso-volumétrico$)$;
end

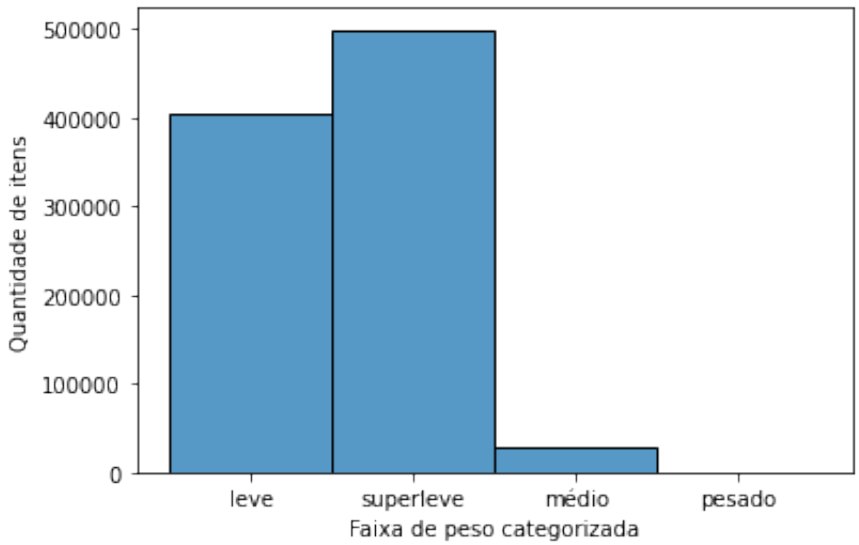
A partir da definição do "peso-frete" um produto é então dividido em categorias: produtos **superleves** têm até 0.5 kg, produtos **leves** tem peso-frete superior a 0.5 e até 5 kg, e produtos **médios** possuem peso-frete superior a 5 kg e menor ou igual a 20 kg. Após essa faixa de peso, os produtos passam a ser categorizados como pesados e não são mais entregues pelo transporte da empresa. Como os pesados estão fora da política de entregas da empresa, todo o planejamento e execução são de responsabilidade total do vendedor que disponibiliza o produto no *marketplace*.

Para o restante dos itens, essa categorização é importante no planejamento de qual será o modal de transporte, a malha logística a ser percorrida e o custo estimado de carregamento: produtos de uma mesma faixa têm certas variações de custos conforme suas dimensões, mas

para eles é provisionado um valor (com margem) que considera os possíveis padrões. O valor real é ajustado após a aferição e então os provisionamentos podem ser devolvidos ou cobrados a mais do vendedor. Para categorias diferentes, o valor dos provisionamentos pode divergir significantemente do real, gerando ineficiências e problemas operacionais relacionadas ao valor em caixa provisionado pela empresa e/ou vendedor. Além disso, produtos superleves podem ser entregues em modais específicos e mais baratos (como bicicletas e outros veículos não-motorizados), enquanto as outras categorias necessitam que a última milha seja percorrida com veículos mais potentes.

A Figura 1 mostra a distribuição de classes do dataset amostrado. Para cada uma das classes (superleve, leve e médio) é mostrada a quantidade de exemplo. Uma característica sobressalente na figura é o forte desbalanceamento dentre as categorias, com pouca prevalência da classe minoritária (produtos com peso médio). Isso é um indicativo de que métricas de modelos que consigam trabalhar com dados desbalanceados possam ser explorados, caso contrário as predições da classe minoritária podem ser insatisfatórias.

Figura 1 – Distribuição de classes do *dataset* utilizado nos experimentos deste trabalho.



Fonte: Autoria própria.

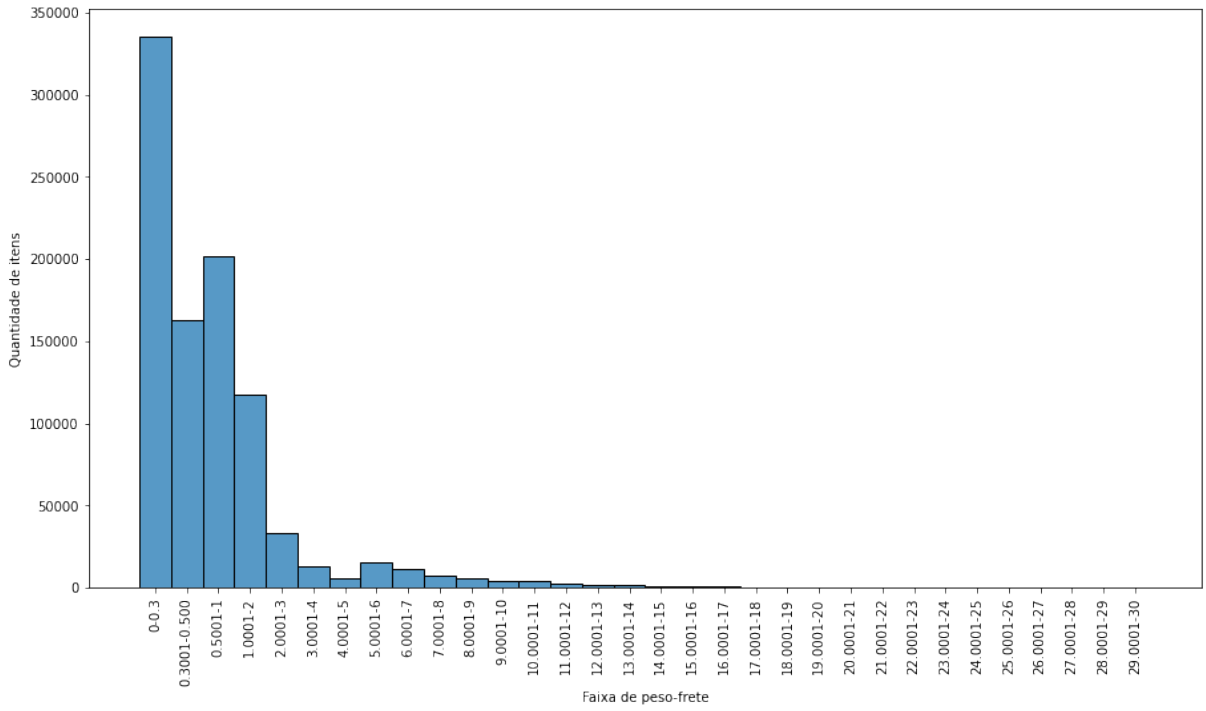
4.2 Análise Exploratória dos Dados

Antes de realizar os experimentos propriamente ditos, foi conduzida uma Análise Exploratória de Dados, do inglês *Exploratory Data Analysis* (EDA), a fim de se compreender melhor características intrínsecas do problema. Nesta primeira etapa do *pipeline* aplicado, extraímos algumas características simples e possivelmente relevantes para a resolução do problema. Como a única característica preditiva do problema é textual, a EDA verificou a distribuição de classes do problema, tanto na categorização da Figura 1, como considerando outras possíveis categorias.

Essa maior separação em categorias pode ser vista na Figura 2. No universo dos produtos superleves, é visível que a concentração de itens considera os produtos cujos pesos estão mais próximos a zero. A tendência geral do conjunto é a diminuição da quantidade de itens de acordo com o aumento na faixa de peso.

Posteriormente, foi aplicada uma vetorização via TF-IDF para caracterização do dataset, seguida da aplicação de uma técnica de Análise de Componentes Principais, do inglês *Principal Component Analysis (PCA)*. Os dois primeiros componentes, e mais descritivos, foram usados para gerar a Figura 3, na tentativa de verifica a separabilidades das classes originais do problema. É identificável que não há uma clara separação classes, tendo diversos grupos de exemplos sobrepostos. Alguns itens da classe média até acabam ligeiramente separados das demais na região mais inferior do gráfico, porém as classes leve e superleve basicamente se misturam. Isso indica que o problema não é fácil, e espera-se que classificadores lineares não sejam eficientes na resolução do problema. Por isso, serão propostas algumas soluções já utilizadas para problemas similares e que consideram a complexidade dos problemas relacionados à textos e linguagem.

Figura 2 – Distribuição de classes do *dataset* estudado em faixas mais discriminadas de peso-frete.

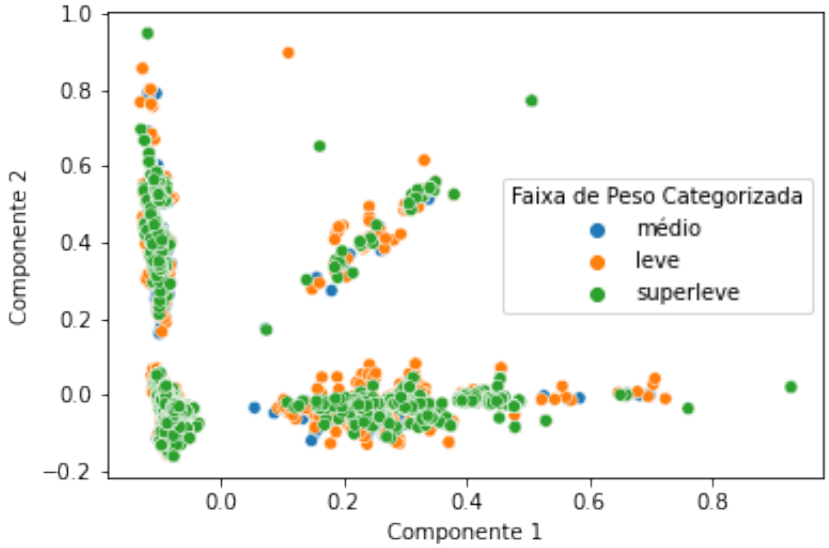


Fonte: Autoria própria.

4.3 Pipeline Implementado

Experimentos foram realizados na tentativa de gerar um modelo único para classificação multiclasse. Dois diferentes modelos baseados em BERT foram experimentados. O que diferencia

Figura 3 – Aplicação de PCA bidimensional no *dataset* estudado.



Fonte: Autoria própria.

cada modelo é o pré-processamento aplicado ao *dataset*, isto é, como os dados textuais são *tokenizados*:

- **modelo1**: não realiza nenhum tipo de pré-processamento nos dados de entrada, apenas faz a normalização dos caracteres em caixa baixa;
- **modelo 2:** realiza uma etapa de pré-processamento com remoção de tags html, remoção de pontuações e acentuações, tokenização dos dados, remoção de caracteres alfanuméricos e *stopwords*.

Em ambos os casos os dados passam posteriormente pelo tokenizados BERT, recebendo tanto truncação quando *padding* para normalizar os tamanhos dos vetores de entrada. Para que não haja interferência desses passos no aprendizado do modelo, as máscaras de atenção também são retornadas para serem usadas como entrada do modelo preditivo. Dependendo do modelo gerado, cada um possui um tamanho diferente de unidades na camada de entrada: o modelo sem pré-processamento possui 100 unidades, enquanto o modelo com pré-processamento possui 80.

Após a camada de entrada, ambos modelos seguem a mesma arquitetura sequencial: uma camada de *embedding* utilizando o transformador *BERT base* pré-treinado, uma camada de *pooling* 1D dos resultados, e três camadas densas de tamanhos 128, 32 e 3, com *dropout* de 10% entre as duas primeiras. A classificação é realizada com uma operação de *softmax* simples para as três possibilidades de saída.

Devido ao desbalanceamento, o treinamento dos modelos usou uma técnica simples de *downsampling*: as duas classes majoritárias são reduzidas até o tamanho da classe minoritária. Os experimentos foram repetidos 5 vezes, gerando 5 modelos para cada solução. A avaliação

foi realizada em outra porção do conjunto de dados, nunca vista nos conjuntos de treinamento, para evitar problemas de *data leakage*. O conjunto de teste mantém o desbalanceamento do problema real, respeitando a distribuição dos dados de cada categoria.

4.4 Baselines

Como *baselines* para resultados dos modelos propostos, foram usados modelos de Árvore de Decisão e Naïve Bayes com hiperparâmetros pré-definidos. Tal escolha é feita baseada no fato de que estes dois algoritmos compõem hoje a solução atual do ambiente de produção da empresa. No entanto, a aplicação dos algoritmos é distinta.

A implementação atual da empresa usa um modelo específico para predição de itens de cada sub-família de itens, seguindo a divisão mais granular do departamento. Como existem milhares de subfamílias, derivam dessa solução milhares de modelos, cada um treinado especificamente para um pequeno conjunto de itens. Tal solução é de dificílima manutenção, tanto pela dificuldade em identificar modelos que estão com desempenho inferior, como quanto pelo custo computacional de retreinar e gerar modelos mais adequados para todas as subdivisões de itens.

Desta forma, neste trabalho investigamos a possibilidade de se usar um modelo único capaz de prever itens independentemente de suas subclassificações. Assim, os experimentos realizados consideram apenas modelos globais, isto é, o desempenho a ser comparado é de modelos que realizam predições para **todas** as subcategorias de itens em uma única implementação. Adicionalmente, foi também utilizado como *baseline* um classificador baseado na classe majoritária, dado o desbalanceamento de classes presente no conjunto real, qualquer modelo que seja induzido precisa ser superior a tal abordagem.

4.5 Reprodutibilidade dos Experimentos

Os experimentos foram realizados na linguagem `Python`, versão 3.7.1. Foi usado também o pacote de bibliotecas do `Anaconda`, versão 4.10.1, destacando-se: `Numpy`, `Pandas`, `Matplotlib`, `Seaborn`, `Scikit-learn`, `Tensorflow`, `Keras`. O valor de semente aleatória utilizado nos experimentos foi 42. Como os dados utilizados tratam de informações confidenciais reproduzidas com alteração da empresa fornecedora, a reprodutibilidade e acesso às fontes originais pelo público geral foram impossibilitadas.

5 RESULTADOS

Neste capítulo são discutidos os principais resultados obtidos e suas implicações diretas. Também serão buscadas possíveis causas que expliquem o desempenho dos melhores modelos.

5.1 Desempenho Geral dos Classificadores

O desempenho geral de todos os modelos e estratégias avaliadas é apresentado na Tabela 1. Na Tabela são reportados os valores de acurácia balanceada por classe (BAC) obtidos por cada modelo no conjunto de testes. O melhor resultado é indicado em negrito na coluna dos valores de BAC.

Tabela 1 – Acurácia balanceada (BAC) dos classificadores no conjunto de teste. O melhor resultado está destacado em negrito.

Modelo	**BAC**
Naïve Bayes + PCA	0.33
Árvore de Decisão + PCA	0.35
Árvore de Decisão	0.37
Naive Bayes	0.37
Classificador Majoritário	0.53
BERT com pré-processamento textual	0.58
BERT sem pré-processamento textual	**0.63**

Fonte: Autoria própria.

Desta visão geral pode-se notar que os valores de BAC da maior parte dos modelos é baixa. O desbalanceamento das classes pode ser uma explicação para o desempenho ruim dos métodos inicialmente testados. Esta é, inclusive, uma das justificativas para a implementação da solução anteriormente em produção, que criava um classificador para cada subcategoria de itens: o desbalanceamento é vastamente reduzido dentro de cada conjunto específico. No entanto, testar cada um dos modelos para cada classe é custoso e se faria inviável na implementação de redes neurais.

Ainda se tratando das *baselines*, é interessante notar que o desempenho dos modelos que utilizam os componentes de PCA como entrada é inferior aos modelos que não o fazem, e utilizam apenas a matriz de TF-IDF. Esse resultado é parcialmente esperado, visto que a classificação de textos, quando realizada desta maneira, acaba sendo uma ponderação da frequência de palavras. Com menos componentes, perde-se o valor de termos especificamente ponderados. Assim, o uso dos textos completamente vetorizados acaba fornecendo mais características a serem aprendidas.

Observando os classificadores baseados no transformador BERT, percebe-se que ambos os modelos apresentaram os melhores resultados dentre todos os testados. Ainda assim, as

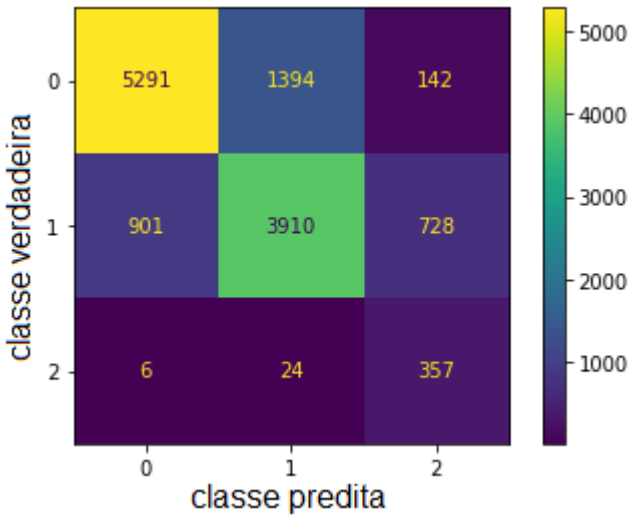
performances não são tão boas quanto o esperado, com pouco mais de 0.6 de BAC no melhor dos casos. Novamente, é interessante notar que o pré-processamento dos dados textuais acaba reduzindo a performance do modelo. Para entender melhor o comportamento da melhor solução, nas seções seguintes serão analisadas as predições e buscados possíveis indícios de melhoras a serem feitas.

Como os melhores modelos são baseados no transformador BERT, não há muitos ajustes de hiperparâmetros a serem feitos: a taxa de aprendizado recomendada é padrão e a arquitetura da rede é em grande parte pré-definida. As únicas adições na rede de melhor desempenho foram as camadas densas, de *droupout* e *softmax* na saída. Assim, é interessante tentar entender padrões, comportamentos dos modelos obtidos, ou indícios de características que estejam afetando o desempenho do modelo, em vez de ajustes pontuais em arquitetura.

5.2 Analisando as Predições dos Modelos

Dado que o classificador BERT sem pré-processamento textual foi o que demonstrou melhor desempenho no conjunto de testes, as predições deste serão as analisadas para entendimento dos erros cometidos. A Figura 4 mostra todas as predições realizadas pelo melhor classificador no conjunto de testes. As classes 0 a 2 referem-se à ordem de peso crescente, sendo a primeira superleve e a última leve. A partir dessa matriz, pode-se calcular as métricas para cada classe e compará-las para entender o que faz o desempenho geral estar abaixo do esperado.

Figura 4 – Matriz de confusão das predições do melhor modelo baseado em BERT para o conjunto de teste.



Fonte: Autoria própria.

As classes 1 e 2, quando observadas isoladamente, obtém precisão de 85% e 73%, respectivamente. A terceira classe (médios), no entanto, apresenta apenas 29% de precisão, mas 92% de revocação. Isso significa que a maioria dos itens pertencentes à classe 2 são corretamente

classificados, mas muito itens também não pertences a esta classe são erroneamente atribuídos a ela. Na matriz de confusão, isso é claramente visível pela quantidade de exemplos pertencentes às duas outras classes e preditos como médios. Além disso, também há certo conflito entre predições errôneas nas classes 1 e 2, com itens de uma atribuídos à outra. É interessante checar se há alguma característica que esteja interferindo no desempenho (como, por exemplo, os itens errados estarem próximos de algum limiar de peso).

5.2.1 **Itens erroneamente classificados como Médios**

Iniciando a análise do problema mais perceptível, tentamos encontrar as causas desse fenômeno. Na Figura figura 4 podemos perceber que ocorrem duas situações: itens superleves classificados como médio (142 exemplos) e itens leves classificados como médio (728 exemplos). A segunda situação tem maior ocorrência do que a primeira.

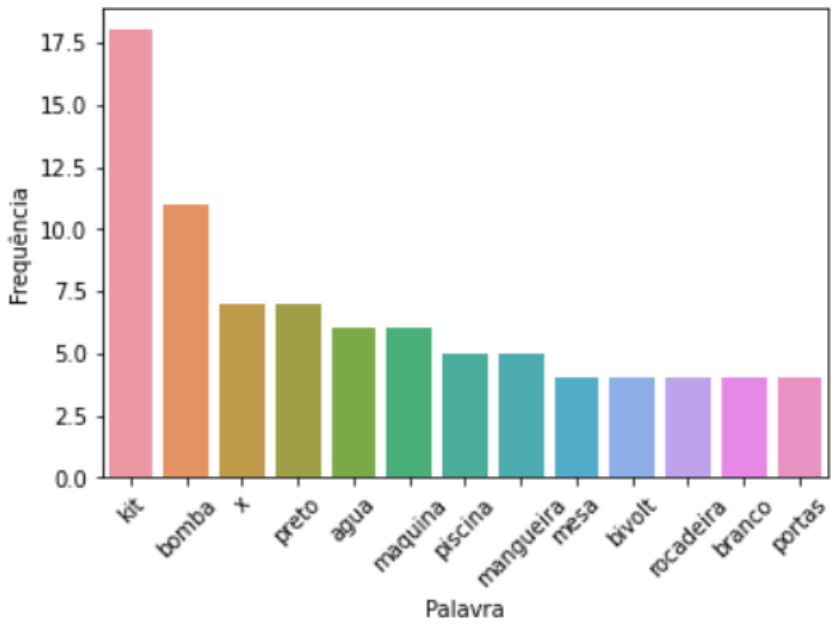
Como o modelo utilizado é baseado em um transformador, todo o processo de criação e ponderamento de características é realizado endogenamente. Ademais, como o BERT baseia-se no aprendizado por transferência, parte dos pesos de cada característica já existe antes mesmo dos treinos serem realizados, e são apenas ajustados durante o treinamento. Assim, fica difícil saber exatamente qual característica pode estar atrapalhando o desempenho do modelo. Essa é uma desvantagem desse tipo de aplicação, remetendo ao conceito de *caixa preta*, pois não se tem controle sobre os aspectos da indução dos modelos de DL.

Uma possível investigação a ser feita, dada a contextualização passada, é verificar as características exógenas ao modelo. Nesse caso, é possível buscar os títulos dos produtos, que são as características originais do problema, e verificar se há algum tipo de palavra que comumente aparece nos itens erroneamente classificados como pertencentes à classe de produtos com peso-frete médio (classe 2).

A Figura 5 mostra a frequência das palavras contidas nos títulos dos exemplos de itens superleves classificados erroneamente como médios. No top-5 de termos mais frequentes do ranking aparecem as palavras: “kit”, “bomba”, “x”, “preto” e “água”. Embora o total de itens classificados errados é 142, nenhuma palavra aparece em mais de 22 itens. Além disso, o transformador BERT utiliza mecanismos de atenção que conseguem atribuir sentido à n-gramas diversos, então esse resultado isoladamente pode não significar algo determinante. No entanto, algumas coisas são identificáveis: sabe-se, por exemplo, que a palavra “x” não é algo que tem valor para classificar um item, dado que não oferece nenhum tipo de valor semântico. Esse pode ser um primeiro indício de que talvez seja necessário fazer algum tipo de pré-processamento específico, mais alinhado à definição do problema.

A Figura 6 mostra o segundo caso: exemplos de itens leves classificados como tenso pes-frete na categoria “médio”. Dos 728 itens classificados erradamente, em mais de 130 deles aparece a palavra “kit” no título do produto. Novamente, aqui aparecem outras palavras sem muito sentido semântico, como: “x”, “preto”, “litros” e “pecas”. Os dados descritos pela figura reforçam novamente que há algum tipo de assimilação de “kit” com a categoria dos itens médios

Figura 5 – Frequências das palavras presentes nos títulos de produtos superleves preditos como médios.



Fonte: Autoria própria.

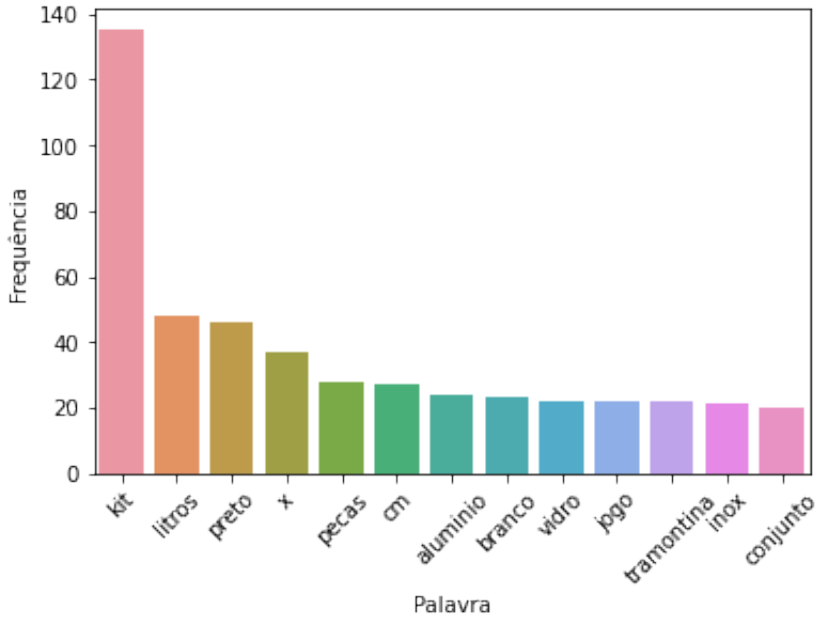
que foi aprendida pelo modelo.

5.2.2 **Erros de classificação entre classes 0 e 1**

A segunda forma mais frequente de erros preditivos do modelo é a classificação errônea dos itens leves como superleves e vice-versa. Tais casos correspondem a maior parte dos erros do modelo. Ou seja, entender sua origem abre um novo leque de possibilidades de como melhorar os modelos preditivos. Nesses casos, além da visualização das palavras mais frequentes nos títulos dos produtos, também faz sentido visualizar outra métrica de negócio: a distribuição de peso-frete dos itens com classificação errada. È possível que os produtos errados estejam muito próximos de um limiar de peso que separe as duas classes. Pode ser, por exemplo, que produtos leves classificados como superleves tenham peso muito próximo à segunda categoria. Se isso ocorre, talvez valha a pena reavaliar a forma de categorização e tentar aproximar o problema por outra ótica.

As Figuras 7 e 8 mostram a distribuição de pesos dos erros que aconteceram nas classes superleves e leves. Em ambos casos, apesar de haverem erros com distintos pesos, é perceptível que há crescimento da frequência conforme mais aproxima-se do limiar da classe oposta. Esse comportamento pode indicar que o classificador está com dificuldades em produtos que têm pesos muito próximos da superfície de decisão. Em teoria, isso pode significar que determinados itens não estão sendo bem aprendidos pelo modelo, apesar de possuírem pesos próximos do que seria a categorização correta. Por exemplo, o modelo pode ter aprendido a classificar determinados tipos de celulares como superleves, mas as variações em modelos de

Figura 6 – Frequências das palavras presentes nos títulos de produtos leves preditos como médios.



Fonte: Autoria própria.

categorias diferentes são erradas por pouca margem. Novamente, como se trata de um modelo baseado em transformadores, é difícil entender exatamente o problema, mas novas tentativas de separação dos dados podem ser relevantes.

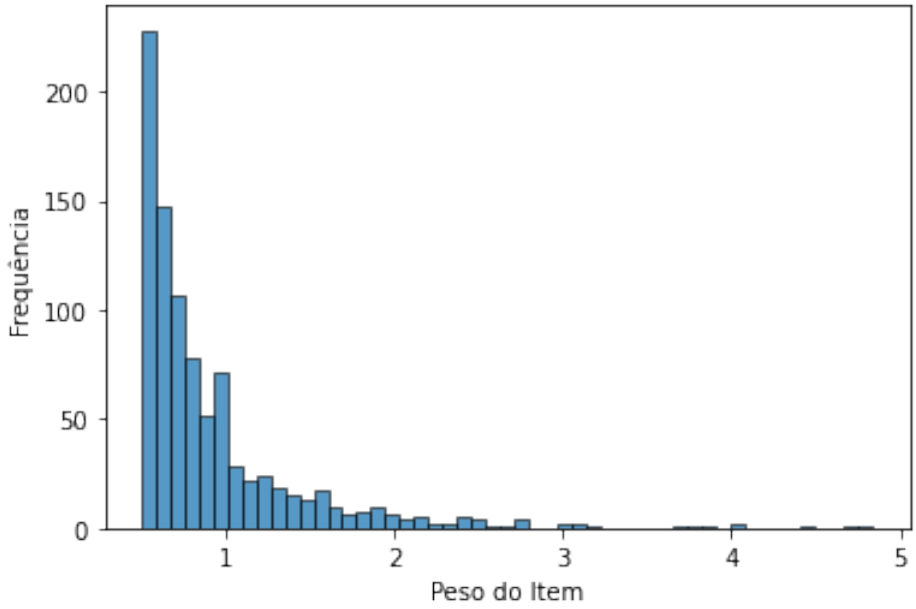
As Figuras 9 e 10 mostram as frequências de termos nas classificações errôneas entre produtos leves e superleves. Nesse caso, não há nenhum grande indicativo além da alta frequência da palavra "kit" nos erros. È importante lembrar que o transformador não funciona exatamente como um TF-IDF, então não é descartada a possibilidade de algum tipo de valor estar associado a esse termo. Aparentemente, há indícios que possíveis melhorias de desempenho estão associadas à melhor definição da característica de entrada (o título do produto).

5.3 Considerações Finais

Sobre os resultados obtidos, seguem algumas considerações:

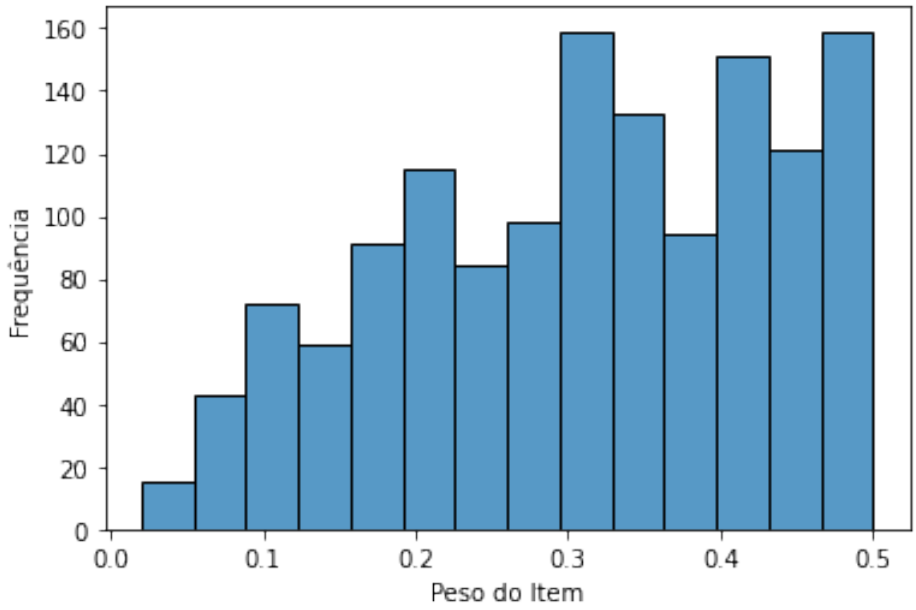
- Como discutido nas seções anteriores, o modelo BERT base sem pré-processamento possui o melhor desempenho dentre os modelos testados. No entanto, os valores de BAC obtidos ainda estão aquém do esperado. Em termos de custos de treinamento e predição, a utilização de TPUs para *fine-tuning* e predição do BERT torna esta uma solução bastante viável, com mais de 8 iterações de retro-propagação realizadas por segundo.
- Uma possibilidade que não foi avaliada nesse trabalho é o uso do modelo `BERT` *large*, versão mais robusta do transformador. A impossibilidade se deu pela quantidade de

Figura 7 – Distribuição de peso-frete dos produtos leves preditos como superleves.



Fonte: Autoria própria.

Figura 8 – Distribuição de peso-frete dos produtos superleves preditos como leves.

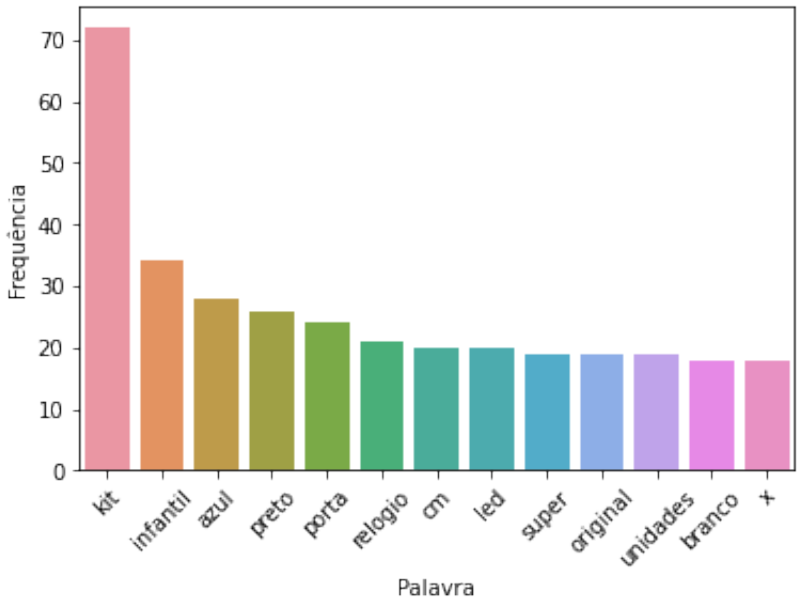


Fonte: Autoria própria.

memória necessária para processar o modelo, muito mais exigente que a versão base. Não foi possível realizar o teste nem em ambiente *on premise* nem nos serviços de nuvem gratuitos disponíveis.

- Dos modelos e predições estudadas, há indicativos que uma melhor definição da característica de entrada (o título do produto) poderia ser realizada. Isso seria conduzido para

Figura 9 – Frequências das palavras presentes nos títulos de produtos superleves preditos como leves.

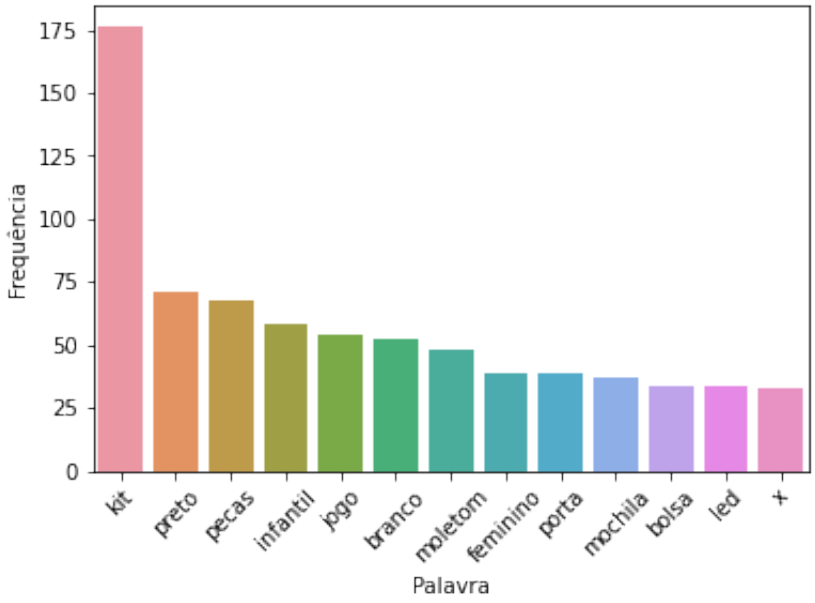


Fonte: Autoria própria.

remover termos que não possuem relevância na classificação de determinado produto. Por exemplo, a palavra kit, que parece atrapalhar muitas predições, poderia ser removida (ou todos os itens vendidos dessa forma poderiam ser excluídos do modelo e testados com outra solução, visto que a aferição de um kit não é unitária). Outras palavras, como adjetivos de cor ou de estado (novo, usado), também não oferecem valor ao classificar um item, pois são características que em teoria não diferenciam mesmos produtos. Um celular azul, por exemplo, não teria diferenças significativas para outro preto de mesmo modelo. Uma roupa infantil, no entanto, é diferente em tamanho de uma adulta, então esse seria um adjetivo importante. Talvez a adição de regras desse tipo, manualmente (em vez do pré-processamento simples e padronizado), seja capaz de melhorar os resultados dos modelos.

- É interessante ressaltar, novamente, a alta revocação do melhor classificador quanto às predições dos itens médios. Com pouquíssimos falsos negativos, talvez seja possível pensar em outro tipo de metodologia para a solução do problema, como o desmembramento em mais modelos e a divisão das soluções em partes.

Figura 10 – Frequências das palavras presentes nos títulos de produtos leves preditos superleves.



Fonte: Autoria própria.

6 CONCLUSÃO

Este trabalho investigou o uso de soluções baseadas no transformador BERT para a classificação de textos, mais especificamente a predição da faixa de peso de um item a partir de seus títulos. A partir dos resultados obtidos, em comparação a baselines que comumente são utilizados para tarefas de NLP, as soluções baseadas em BERT demonstraram desempenho superior, ainda mais considerando o desbalanceamento inerente ao problema estudado. No entanto, apesar de promissores, os resultados ainda não são satisfatórios para uma implementação em produção, devido às necessidades do negócio de ter altos níveis de acurácia para evitar perdas com cálculos errôneos de frete. De fato, as soluções baseadas em aprendizagem de máquina tradicional parecem não aprender com o problema e perdem para um classificador majoritário *dummy*.

O estudo das predições realizadas pelo melhor classificador indica que talvez haja possibilidade de melhorar o pré-processamento de textos com regras manualmente ajustadas, de modo a manter somente características que sejam discriminantes aos itens. O treinamento, quando realizado utilizando TPUs, é rápido e pouco custoso ao sistema, não havendo grandes limitações para a reprodução do experimento com diferentes configurações de características. O aumento da amostragem para melhorar o aprendizado também seria uma solução viável, mas no momento o conjunto de dados representa todo o universo de itens cadastrado em sistema. Assim, espera-se que a engenharia de características seja o melhor fator para ajustar o desempenho em futuras iterações.

6.1 LIMITAÇÕES E TRABALHOS FUTUROS

Apesar dos resultados iniciais não serem suficientes para uma implementação viável da solução em um ambiente empresarial, o desempenho preditivo do classificador BERT frente aos outros baselines é bastante promissor, considerando que o ganho de acurácia foi substancial mesmo sem grandes ajustes no universo de características.

A partir dos resultados obtidos, surgem novas opções a serem exploradas no futuro, incluindo testes e configurações distintas que não foram testados por falta de recursos ou pelo escopo definido inicialmente. Dentre essas opções, destacam-se:

- Foram testadas configurações baseadas apenas no modelo BERT base, por falta de recursos para treinamento da variação large. Espera-se que a performance do classificador large seja maior, e isso poderia ser testado caso houvessem recursos computacionais suficientes;
- Dados os resultados dos classificadores, talvez seja possível a redivisão das classes do problema e o teste de novas configurações para sua solução. Por exemplo, testar

classificadores um-contra-o-resto;

- Não foram testados mais hiperparâmetros ou arquiteturas distintas para a rede neural baseada em BERT. Apesar da maior parte dos HPs ser recomendada, talvez possa haver ganho de performance ao alterar a estrutura das últimas camadas de classificação e adoção de *dropout*. No entanto, não espera-se tanta variação, dado que o transformador em si não possui hiperparâmetros ajustáveis;

- Não houveram grandes esforços de engenharia de características que correspondessem exatamente ao problema estudado. De fato, diversas das possibilidades vieram à tona apenas após o estudo das predições erradas pelo melhor classificador. Dentre fatores que provavelmente melhorem o desempenho do modelo, estão a remoção de palavras que não ofereçam valor discriminatório a um produto (como adjetivos de cor, estado, etc.);

- No escopo de definição do problema, a existência de kits no universo de itens pode atrapalhar a solução do problema real de predição de peso-frete: kits são conjuntos de itens que não são padronizáveis (geralmente cada vendedor possui um kit próprio com distintos itens nele). Assim, talvez a exclusão dessa categoria na solução seja possível. Com isso, seria necessário revisitar quais predições seriam consideradas pelo modelo e aplicadas em produção;

- Não foi comparada a solução BERT com a arquitetura de predição já em produção. Na arquitetura atual, cada subcategoria de itens é treinada em um classificador diferente que testa e classifica apenas os itens desse determinado subconjunto. Essa solução é útil para o desbalanceamento de classes, apesar de custosa em treinamento. Um possível teste que não foi feito é o de custo de treinar um BERT para cada subcategoria e comparar à performance média do melhor classificador *baseline* atualmente em produção.

Referências

AGGARWAL, C. C. **Neural Networks and Deep Learning**. [S.l.]: Springer International Publishing, 2018. Citado na página 15.

Devlin, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv e-prints**, p. arXiv:1810.04805, oct 2018. Citado 3 vezes nas páginas 14, 15 e 20.

GONZÁLEZ-CARVAJAL, S.; GARRIDO-MERCHÁN, E. C. Comparing bert against traditional machine learning text classification. **ArXiv**, abs/2005.13012, 2020. Citado 2 vezes nas páginas 14 e 25.

HADDI, E.; LIU, X.; SHI, Y. The role of text pre-processing in sentiment analysis. **Procedia Computer Science**, v. 17, p. 26–32, 2013. ISSN 1877-0509. First International Conference on Information Technology and Quantitative Management. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050913001385>. Citado na página 17.

HU, Y. et al. Short-text classification detector: A bert-based mental approach. **Computational Intelligence and Neuroscience**, v. 2022, p. 1–11, 03 2022. Citado na página 25.

Liashchynskyi, P.; Liashchynskyi, P. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. **arXiv e-prints**, p. arXiv:1912.06059, dez. 2019. Citado na página 23.

MARSLAND, S. **Machine learning: an algorithmic perspective**. [S.l.]: CRC press, 2015. Citado 2 vezes nas páginas 14 e 15.

MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <https://books.google.com.br/books?id=EoYBngEACAAJ>. Citado na página 15.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, v. 18, p. 544–551, sep 2011. Citado 2 vezes nas páginas 14 e 17.

NASTESKI, V. An overview of the supervised machine learning methods. **HORIZONS.B**, v. 4, p. 51–62, 12 2017. Citado na página 18.

OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning for natural language processing. **IEEE transactions on neural networks and learning systems**, IEEE, v. 32, n. 2, p. 604–624, 2020. Citado na página 19.

QASIM, R. et al. A fine-tuned bert-based transfer learning approach for text classification. **Journal of Healthcare Engineering**, v. 2022, p. 1–17, 01 2022. Citado na página 25.

SAH, S. Machine learning: A review of learning types. 07 2020. Citado na página 18.

SUN, Y.; WONG, A.; KAMEL, M. S. Classification of imbalanced data: a review. **International Journal of Pattern Recognition and Artificial Intelligence**, v. 23, 11 2011. Citado na página 21.

VARGAS, R.; MOSAVI, A.; RUIZ, R. Deep learning: A review. **Advances in Intelligent Systems and Computing**, v. 5, 06 2017. Citado na página 19.

Wu, Y. et al. Google's neural machine translation system: Bridging the gap between human and machine translation. **arXiv e-prints**, p. arXiv:1609.08144, sep 2016. Citado 2 vezes nas páginas 14 e 21.

YU, S.; JINDIAN, S.; LUO, D. Improving bert-based text classification with auxiliary sentence and domain knowledge. **IEEE Access**, PP, p. 1–1, 11 2019. Citado na página 25.