INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Felipe Leonardo Sarmento da Silva

PREVISÃO DE DEMANDA POR TRANSPORTE AÉREO BASEADA EM PROCESSAMENTO DE LINGUAGEM NATURAL E DEEP LEARNING

Trabalho de Graduação
2021

Curso de Engenharia de Infraestrutura Aeronáutica

CDU 656.71

Felipe Leonardo Sarmento da Silva

PREVISÃO DE DEMANDA POR TRANSPORTE AÉREO BASEADA EM PROCESSAMENTO DE LINGUAGEM NATURAL E DEEP LEARNING

Orientador

Prof. Dr. Marcelo Xavier Guterres (ITA)

ENGENHARIA DE INFRAESTRUTURA AERONÁUTICA

São José dos Campos
Instituto Tecnológico de Aeronáutica

2021

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

Silva, Felipe Leonardo Sarmento da
Previsão de demanda por transporte aéreo baseada em processamento de linguagem natural e deep learning / Felipe Leonardo Sarmento da Silva.
São José dos Campos, 2021.
90f.

Trabalho de Graduação – Curso de Engenharia de Infraestrutura Aeronáutica– Instituto Tecnológico de Aeronáutica, 2021. Orientador: Prof. Dr. Marcelo Xavier Guterres.

1. Demanda por Transporte Aéreo. 2. Deep Learning. 3. Processamento de Linguagem Natural. I. Instituto Tecnológico de Aeronáutica. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

SILVA, Felipe Leonardo Sarmento da. **Previsão de demanda por transporte aéreo baseada em processamento de linguagem natural e deep learning**. 2021. 90f. Trabalho de Conclusão de Curso (Graduação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

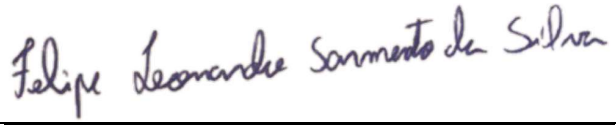
CESSÃO DE DIREITOS

NOME DO AUTOR: Felipe Leonardo Sarmento da Silva
TITULO DO TRABALHO: Previsão de demanda por transporte aéreo baseada em processamento de linguagem natural e deep learning.
TIPO DO TRABALHO/ANO: Trabalho de Conclusão de Curso (Graduação) / 2021

Felipe Leonardo Sarmento da Silva
Rua Gisele Martins, 680
12.236-500 – São José dos Campos–SP

PREVISÃO DE DEMANDA POR TRANSPORTE AÉREO BASEADA EM PROCESSAMENTO DE LINGUAGEM NATURAL E DEEP LEARNING

Essa publicação foi aceita como Relatório Final de Trabalho de Graduação



Felipe Leonardo Sarmento da Silva

Autor

Marcelo Xavier Guterres (ITA)

Orientador

Prof. Dr. João Cláudio Bassan de Moraes
Coordenador do Curso de Engenharia de Infraestrutura Aeronáutica

São José dos Campos, 19 de novembro de 2021.

A meus amigos e família, sem os quais esse trabalho não seria possível.

Agradecimentos

Agradeço a minha mãe Ana e a minha irmã Fernanda por serem os meus maiores exemplos na vida.

Um agradecimento aos amigos que estiveram ao meu lado durante essa caminhada, desde a turma 2 até a civil 20, sem vocês não seria possível concluir esse curso.

Em especial agradeço a Ellen, por todo companheirismo e compreensão durante esses últimos anos e por me ajudar com tempo para escrever e revisar esse trabalho.

Por fim agradeço aos meus professores, em especial ao meu professor e amigo Guterres por toda a paciência e dedicação me orientando neste trabalho.

"Mas quem não morre de melancolia

morre por não nascer".

— Autor desconhecido

Resumo

-

Abstract

Air transport is an essential service for the country's development and for society, so it is of paramount importance that the State and private institutions carry out strategic planning in order to maximize the sector's efficiency, for this the forecast of demand for air transport is fundamental piece. There are some methods in the literature for demand forecasting, such as econometric and gravitational, but all of them have their limitations. This work proposes a new method for the task, based on deep learning and natural language processing, for which a database with newspaper news related to the airline industry from 2006 to 2018 was created using web scraping. This database was associated with ANAC's annual demand data for air tickets, and this associated database served as input to the neural network that, through regression, performs demand forecasts. The results showed that the method is effective in carrying out the proposed task and that newspaper news related to air transport has enough linguistic content to accurately predict the demand for air transport.

Lista de Figuras

FIGURA 1.1 – Exemplo de sazonalidade nos dados de demanda, padrões de comportamento que se repetem em épocas específicas do ano. 22

FIGURA 3.1 – Representação exemplo de uma rede neural com duas camadas ocultas. 37

FIGURA 3.2 – Representação exemplo das relações de um neurônio ou nó de uma rede neural 38

FIGURA 3.3 – Conforme a matriz azul desliza para baixo uma nova matriz é preenchida com dimensão menor que a inicial 39

FIGURA 3.4 – Processo de pre-processamento de documentos em linguagem natural. 40

FIGURA 3.5 – Exemplo de um esquema de representação one-hot para um vocabulário de nove palavras. Embeddings de palavras são lidos como linhas dessa tabela e são predominantemente compostos de zeros para cada palavra. 41

FIGURA 3.6 – Exemplo de palavras representadas por word embeddings de 3 dimensões 42

FIGURA 3.7 – Neste exemplo o mecanismo de atenção se concentra em "The animal"para codificar a representação da palavra "it" 43

FIGURA 3.8 – Exemplo de uso de um modelo pré-treinado pelo BERT em uma tarefa de classificação 44

FIGURA 3.9 – Exemplo gráfico de situações de sobreajuste, ajuste ótimo e subajuste dos dados 44

FIGURA 3.10 –Exemplo gráfico de relação entra qualidade e a complexidade do modelo. Deve-se ajustar o modelo a fim de minimizar os erros no conjunto de teste e no conjunto de treino, ou seja, minimizar tanto o erro sistemático quanto o aleatório 45

FIGURA 3.11 –Cabeçalho e primeiras linhas da tabela contendo as notícias para a tag ”turismo” 47

FIGURA 3.12 –Cabeçalho e primeiras linhas da tabela final para a tag ”economia” . 50

FIGURA 3.13 –Visão geral da arquitetura Pytext 52

FIGURA 4.1 – Evolução do loss ao longo dos epochs para a base de treino do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo. 56

FIGURA 4.2 – Evolução do loss ao longo dos epochs para a base de treino do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo. 56

FIGURA 4.3 – Evolução do loss ao longo dos epochs para a base de treino do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo. . . . 57

FIGURA 4.4 – Evolução do loss ao longo dos epochs para a base de testes do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo. 57

FIGURA 4.5 – Evolução do loss ao longo dos epochs para a base de testes do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo. 58

FIGURA 4.6 – Evolução do loss ao longo dos epochs para a base de testes do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo. . . . 58

FIGURA 4.7 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo. 59

FIGURA 4.8 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo. 59

FIGURA 4.9 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo. 60

FIGURA 4.10 –Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo. 60

FIGURA 4.11 –Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo. 61

FIGURA 4.12 –Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo. 61

FIGURA 4.13 –Evolução do loss ao longo dos epochs para a base de treino para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo. 62

FIGURA 4.14 –Evolução do loss ao longo dos epochs para a base de testes para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo. 63

FIGURA 4.15 –Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo. 64

FIGURA 4.16 –Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo. 64

FIGURA 4.17 –Evolução do loss ao longo dos epochs para a etapa de treino do modelo final. Em azul os valores para o grupo ”economia”, em cinza ”turismo”e em verde ”aéreo” 65

FIGURA 4.18 –Evolução do loss ao longo dos epochs para a base de validação utilizando o modelo final. Em azul os valores para o grupo ”economia”, em cinza ”turismo”e em verde ”aéreo” 66

Lista de Tabelas

TABELA 2.1 – Recorte da literatura sobre previsão de demanda por transporte aéreo 33

TABELA 3.1 – Demanda por transporte aéro no Brasil em número de passageiros pagos 48

TABELA 4.1 – Erro quadrático médio para os 3 grupos da base de dados utilizando o modelo final 66

Lista de Abreviaturas e Siglas

ANAC	Agência Nacional de Aviação Civil
PIB	Produto Interno Bruto
NLP	Natural Language Processing
DAC	Departamento de Aviação Civil
ASK	Available Seat Kilometer
RPK	Revenue Seat Kilometer
IPCA	Índice de Preços no Consumidor
IPEA	Instituto de Pesquisa Econômica Aplicada
KNN	K-Nearest Neighbors
CFNAI	Chicago Fed National Activity Index
MLP	Multilayer Perceptron
NLTK	Natural Language Toolkit
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
LSTM	Long Short TermMemory

Sumário

1 Introdução 16

1.1 Problema de Pesquisa 19

1.2 Objetivo 19

1.2.1 Objetivos Específicos 20

1.3 Justificativa 20

1.4 Delimitação do tema 21

1.5 Estrutura do trabalho 22

2 Fundamentação Teórica 24

3 Metodologia 35

3.1 Referencial Metodológico 35

3.1.1 Aprendizado de máquina 35

3.1.2 Aprendizado profundo 36

3.1.3 Processamento de linguagem natural 39

3.2 Base de dados 46

3.2.1 Descrição dos dados 46

3.2.2 Obtenção dos dados 48

3.2.3 Preparação e Pré-processamento 49

3.3 Aprendizado profundo 51

4 Resultados 55

4.1 Configuração e Calibragem do Modelo 55

4.1.1 Divisão da Base de Dados 55

4.1.2 Título x Texto 55
4.1.3 Número de Epochs 62
4.2 Avaliação do Modelo 65
4.3 Discurssão 66

5 Conclusão e Pesquisas Futuras 68
5.1 Conclusão e Pesquisas Futuras 68

Referências 70

Apêndice A – Extração dos links 74

Apêndice B – Limpeza dos links 81

Apêndice C – Separa os links por ano 83

Apêndice D – Captura o texto dos links 84

Apêndice E – Junta os textos no dataframe final 86

Apêndice F – Prepara e pré-processa os textos 88

1 Introdução

Desde o seu surgimento, a aviação comercial vem exercendo um papel cada vez mais importante no transporte de pessoas e carga pelo mundo, promovendo turismo, comércio e a economia de forma geral. De acordo com (LOVATTI, 2018), o surgimento de novas tecnologias torna voar cada vez mais rápido, simples, acessível e seguro. Assim, progressivamente, o transporte aéreo se consolida como um serviço essencial e peça fundamental para o contínuo desenvolvimento do país, de forma que tanto Estado como instituições privadas têm a responsabilidade de realizar um planejamento estratégico que direcione suas formas de atuação, a fim de garantir o máximo benefício para a sociedade como um todo.

Um dos principais desafios para qualquer organização está na previsão de elementos chave para seus objetivos estratégicos. No setor de aviação não é diferente; fabricantes de aeronaves, companhias aéreas, administradoras de terminais aeroportuários, governos e empresas atuantes na cadeia de abastecimento da aviação devem conciliar seus planejamentos com a previsão de tráfego aéreo futuro. Subestimar a demanda pode gerar congestionamento nos aeroportos e no tráfego aéreo, sobrecarregar o sistema e os agentes envolvidos, e possivelmente reduzir a qualidade do serviço oferecido e a satisfação dos passageiros. Por outro lado, superestimar a demanda, de forma geral, pode resultar em gastos excessivos e até mesmo em desperdício de recursos, como assentos não ocupados em um voo.

Companhias aéreas capazes de realizar boas previsões se tornam menos reativas as mudanças do meio e mais proativas, aumentando a sua eficiência, maximizando os lucros em momentos favoráveis e reduzindo a turbulência em momentos de declínio. Na esfera pública, uma boa previsão da demanda garante políticas públicas mais eficientes, gestão otimizada de recursos e, em última instância, contribui para o desenvolvimento do país e o bem-estar da sociedade. Dessa forma, é imprescindível que decisores trabalhem com os modelos mais avançados e confiáveis de previsão de demanda disponíveis.

Os primeiros modelos de previsão de demanda por transporte aéreo surgem simultaneamente aos primeiros anos de operação da aviação comercial e aos primeiros grandes aeroportos, contudo as primeiras previsões se mostraram pouco exatas com o tempo.

(MALDONADO, 1990) comparou a previsão de demanda contida no plano diretor de 22 aeroportos da região de New England nos Estados Unidos com os dados reais para o período de 1973 a 1986. Ele mostrou que, dentre as previsões com horizonte de tempo de 5 anos, as estimativas do plano diretor desviavam-se na faixa de 36% a 96%, ou seja, as previsões se mostraram ser até 36% menores ou 96% maiores do que os volumes de tráfego reais, enquanto que as previsões de 15 anos, apresentaram um desvio de 34% a 210%.

Parte da dificuldade de se obter boas previsões de demanda está na grande volatilidade do número de passageiros transportados. (NEUFVILLE; BARBER, 1991) analisaram a volatilidade do tráfego aéreo nos 38 maiores aeroportos dos EUA durante o período de 1968 a 1988. Eles concluiram que a volatilidade não diminuiu ao longo do período, tendo inclusive aumentado com a lei de desregulamentação das companhias aéreas de 1978. Dentre as possíveis explicações, está a dinamicidade do mercado de aviação; aéreas experimentando novas operações e estratégias de competição, surgimento de novas empresas e frequentes fusões e alianças de companhias aéreas levando a reconfigurações de rotas e redes.

As constantes inovações e o vanguardismo nos avanços tecnológicos do setor aéreo mostram-se outros complicadores a tarefa de previsão de demanda. De acordo com (GODOY, 1997), quanto maiores as mudanças tecnológicas de um setor, maiores são as chances de mudança nos relacionamentos entre as variáveis e o objeto da previsão, dificultando as previsões de demanda baseadas na identificação de padrões nesses relacionamentos. Ainda de acordo com (GODOY, 1997), outro fator agravante para a previsão é a alta elasticidade-preço da demanda. Quanto mais elástica a demanda, maior é a variação da mesma em função de fatores como o preço médio do bilhete e renda do consumidor. Isso significa que, para elementos de primeira necessidade, como alimentos e remédios, cuja elasticidade é baixa, pode-se obter previsões de demanda mais acertadas do que para elementos de segunda necessidade sob a ótica do consumidor, como viagens aéreas a lazer por exemplo.

Embora a elasticidade-preço da demanda no setor aéreo seja alta, o modelo econométrico tem sido um dos mais utilizados para previsão na área. Em seu trabalho, (JORGE-CALDERÓN, 1997) identifica os fatores geoeconômicos como os principais responsáveis pela demanda por transporte aéreo, sendo a renda e o tamanho da população as principais variáveis utilizadas para representar esses fatores na literatura.

Ao longo dos anos, surgiram muitos estudos de previsão de demanda no setor aéreo e com isso outras variáveis foram incorporadas na literatura, dentre elas: Yield, PIB, frequência de serviço, tamanho das aeronaves, distância das viagens, capacidade do terminal aeroportuário, poder de compra da população, taxa de câmbio, consumo de energia, disponibilidade de crédito, preço do combustível e eventos do setor aéreo no geral.

Em sua maioria, os dados das variáveis utilizadas nessas previsões são de fontes estruturadas, o que naturalmente leva a uma limitação quanto a quantidade e disponibilidade

desses dados. De acordo com (DAS; KUMAR, 2013) mais de 80% dos dados disponíveis no mundo estão na forma não estruturada, como textos, áudios, imagens e vídeos, sendo boa parte disso devido ao advento do Big Data. A expressão Big Data se refere a um fenômeno global caracterizado pela grande quantidade e variedade de dados que são gerados em alta velocidade por diversas fontes nos dias de hoje.

Em consequência ao Big Data, o Aprendizado de Máquina, subcampo da engenharia e da ciência da computação dedicado ao reconhecimento de padrões em dados, se destaca como poderosa ferramenta, capaz de explorar e analisar novas fontes de dados e assim resolver novos problemas e gerar novos insights para problemas antigos. De acordo com (SIEGEL, 2013) os dados que coletamos hoje nos permitem ver coisas que até pouco tempo atrás eram grandes demais para enxergarmos.

Uma das principais tendências do Aprendizado de Máquina e da manipulação de dados não estruturados é o Processamento de Linguagem Natural (PLN) ou Natural Language Processing (NLP), como é mais conhecido. O NLP é uma área que se dedica a desenvolver tecnologias capazes de entender e extrair informações relevantes diretamente da linguagem usada pelos seres humanos. As aplicações dessa área são as mais diversas, a análise de textos disponíveis online podem servir para relacionar a experiência vivenciada por um hóspede e a sua satisfação com o hótel (KO, 2018), prever a volatilidade do mercado de ações (SCHUMAKER; CHEN, 2009) e estimar as vendas de bilheteria na indústria cinematográfica (DUAN *et al.*, 2008) e (JOSHI *et al.*, 2010). Verificou-se também que as publicações de notícias on-line têm conteúdo linguístico suficiente para prever os ganhos e o retorno das ações de uma empresa (TETLOCK *et al.*, 2008) e (CANTO, 2020). E, como demonstrado por (GHOSE; IPEIROTIS, 2011), associando o conteúdo do texto com as características do autor, é possível estimar a utilidade e o impacto econômico de avaliações de produtos on-line.

No campo do transporte aéreo, o processamento de linguagem natural já foi usado para classificar e identificar similaridades em relatórios de incidentes e acidentes na aviação de forma automática (TULECHKI, 2015), para tradução e transliteração em tempo real de fraseologia ATC em inglês para o idioma bengali (PAUL; PURKHYASTHA, 2020) e, mais recentemente, para agrupar e classificar de forma automática julgamentos legais de consumidores contra companhias aéreas (SABO *et al.*, 2021).

Ao longo dos últimos anos, o processamento de linguagem natural vem se provando uma poderosa ferramenta capaz de realizar diferentes tarefas e resolver problemas relevantes de diversas áreas, incluindo o transporte aéreo. Como demonstrado, no setor aéreo um dos tópicos mais importantes a ser trabalhado e de ordem estratégica é a previsão de demanda: um problema de identificação de padrões e que depende de inúmeros fatores, muitos dos quais possivelmente ainda desconhecidos e presumivelmente disponíveis em fontes não estruturadas de dados. Ainda, de acordo com (BATES, 2007), a demanda por

transporte aéreo se diferencia das demandas convencionais por ser uma demanda derivada, ou seja, a viagem aérea não costuma ser o produto final desejado pelo cliente, e sim um meio para outro objetivo. Nesse sentido, surgem relações socioeconômicas complexas, que precisam ser singularmente analisadas para que a demanda seja modelada de maneira correta. Desta forma, a previsão de demanda por transporte aéreo se mostra uma tarefa promissora para a abordagem baseada em Aprendizado de Máquina e NLP.

1.1 Problema de Pesquisa

O processamento de linguagem natural, enquanto subcampo do aprendizado de máquina, vem liberando seu poder em uma ampla gama de aplicações. Sua abordagem baseada na exploração e análise de dados não estruturados, neste caso na forma de texto, vem permitindo ao NLP resultados, em muitos casos, mais favoráveis do que os métodos tradicionais. Dada a alta complexidade da tarefa de previsão de demanda aérea e sua relação com diversos fatores, a pergunta que este estudo se propõe a responder é: é possível prever demanda por transporte aéreo por meio de processamento de linguagem natural?

Hoje há um enorme volume de dados textuais em forma de linguagem natural sendo gerados diariamente, muitos de alguma forma relacionados ao setor de aviação e que têm o potencial de serem utilizados em diversos processos de melhoria, inclusive na previsão de demanda. Uma das maiores fontes desses dados são as notícias jornalísticas, tais notícias retratam diariamente o cenário político, social e econômico do país, além de informar sobre acontecimentos relevantes a esfera pública. Dentre todas as notícias, é possível filtrar apenas as que têm algum tipo de relação com o transporte aéreo e seus condicionantes políticos e socioeconômicos, porém, após o filtro, ainda é preciso determinar se essas notícias possuem conteúdo suficiente para gerar boas previsões.

1.2 Objetivo

O objetivo principal deste trabalho é criar uma ferramenta capaz de prever a variação de demanda por transporte aéreo a partir de dados textuais, mais especificamente de notícias relacionadas a transporte aéreo extraída de jornais em português brasileiro. A proposta é apresentar um modelo específico para a tarefa, baseado em redes neurais artificias orientada a realizar regressão.

Dessa forma, o desafio se encontra em fornecer um algoritimo único juntamente de uma base de dados textuais específica para o setor, que permitam uma extração eficiente das informações contidas nesse tipo de fonte e, por fim, propor uma metologia confiável de processamento de linguagem natural capaz de prever a demanda por transporte aéreo

com exatidão.

1.2.1 Objetivos Específicos

Com base na definição do objetivo principal e da metodologia proposta para alcançá-lo, foram definidos os seguintes objetivos intermediários a fim de direcionar o trabalho:

- Revisar métodos tradicionais de previsão de demanda por transporte aéreo;
- Formar uma base de notícias relacionadas a transporte aéreo robusta e confiável para treino e validação do modelo e também disponibilização na literatura;
- Garantir a replicabilidade e escalabilidade do modelo

1.3 Justificativa

Por meio dos objetivos apresentados, este trabalho busca contribuições teóricas e metodológicas para a literatura, introduzindo novas variáveis ao estudo da demanda aérea e propondo um novo método para a sua previsão.

A incorporação de novas variáveis a análise, neste caso notícias em texto relacionadas a transporte aéreo, permite a formação de novas ideias e compreensões sobre o tema. Ao analisar o problema sob outro ponto de vista, é possível expandir o entendimento da literatura sobre a natureza da demanda por transporte aéreo e suas relações com elementos da nossa sociedade. Hoje, há um fluxo cada vez maior de informações sendo publicadas na internet diariamente, principalmente por meio de notícias, com isso nota-se uma grande fonte de dados que, até o momento, é subutilizada. Com o tratamento adequado, esses dados podem ser relevantes tanto para o estudo de demanda aérea, quanto para os estudos no setor aéreo como um todo. Dessa forma, espera-se que esse estudo sirva de inspiração para que futuros trabalhos na área façam proveito dessa extensa fonte de dados.

Quanto ao método, esse estudo busca agregar ao incluir as potencialidades de uma abordagem baseada em Aprendizado de Máquina e Processamento de Linguagem Natural. Em última instância, busca-se contribuir para a difusão do estudo de processamento de linguagem natural e a aplicação de suas tecnologias no setor de transporte aéreo. O uso do NLP vem crescendo constantemente no meio acadêmico e profissional pela sua alta capacidade de estruturar e extrair informações de dados altamente não estruturados. Para contribuir com esse campo em ascensão, este trabalho, por meio de um estudo sobre previsão de demanda por transporte aéreo a partir de notícias relacionadas ao setor aéreo, busca proporcionar uma ferramenta para governos – em todos os níveis – e instituições

privadas capaz de otimizar suas análises e auxiliar nas tomadas de decisão que definirão o futuro do transporte aéreo no Brasil.

1.4 Delimitação do tema

O processamento de linguagem natural pode trabalhar com qualquer tipo de idioma e dialeto. Hoje existem centenas no mundo, cada um com uma estrutura única de sintaxe e regras gramaticais. Este trabalho limita-se ao processamento de textos escritos em português brasileiro, pelo fato desse formato de dados ser mais abundante no contexto estudado e também possibilitar o trabalho com uma base de dados mais específica.

Os textos utilizados nesse trabalho restringem-se a notícias veiculadas por jornais brasileiros relacionadas a transporte aéreo. Essa restrição faz-se necessária para garantir a confiabilidade dos dados usados e um direcionamento para o treino da rede, usando apenas conteúdos relevantes para o estudo em questão. A correta seleção dos dados que serão usados como entrada é importante para aumentar a eficiência do algoritmo.

Existem algumas formas de se trabalhar com processamento de linguagem natural atualmente, as mais comuns são:

- Utilização de Léxicos;
- Machine Learning;
- Deep Learning;

A ferramenta desenvolvida neste trabalho é baseada em deep learning. A escolha por essa técnica foi motivada principalmente pelos trabalhos feitos por (RINALDO; NAGANO, 2019) e (CANTO, 2020). No primeiro é utilizada uma rede neural profunda (DNN) não sequencial com três camadas de incorporação de palavras paralelas para prever a votação de políticos no Congresso Brasileiro, baseando-se na transcrição de seus discursos. No segundo, é utilizada a ferramenta PyText, um framework para modelagem de processamento de linguagem natural baseado em Deep Learning, para a criação de um modelo capaz de prever o impacto (valorização ou desvalorização) de notícias do mercado brasileiro nas ações da bolsa de valores.

No que diz respeito a demanda por transporte aéreo, a demanda tratada neste trabalho refere-se a demanda agregada do mercado aéreo brasileiro, considerada aqui como a soma do número de passageiros pagos em voos nacionais e internacionais com origem ou destino ao Brasil. Este estudo trabalha com as suas variações anuais, ou seja, o quanto a demanda aumenta ou diminui de um ano para o outro. Essa delimitação temporal é importante para

anular o efeito de sazonalidades que ocorrem em períodos específicos do ano e também para garantir que as notícias sejam aproveitadas em seu período de maior impacto, logo após sua publicação. Ainda, de acordo com (GODOY, 1997), previsões de curto prazo, cobrindo o período de 1 ano, apresentam vantagens na tomada de decisões táticas e operacionais pertinentes ao transporte aéreo, como alocação de aeronaves às rotas, planejamento da manutenção das aeronaves e lançamento de campanhas promocionais ou de vendas.

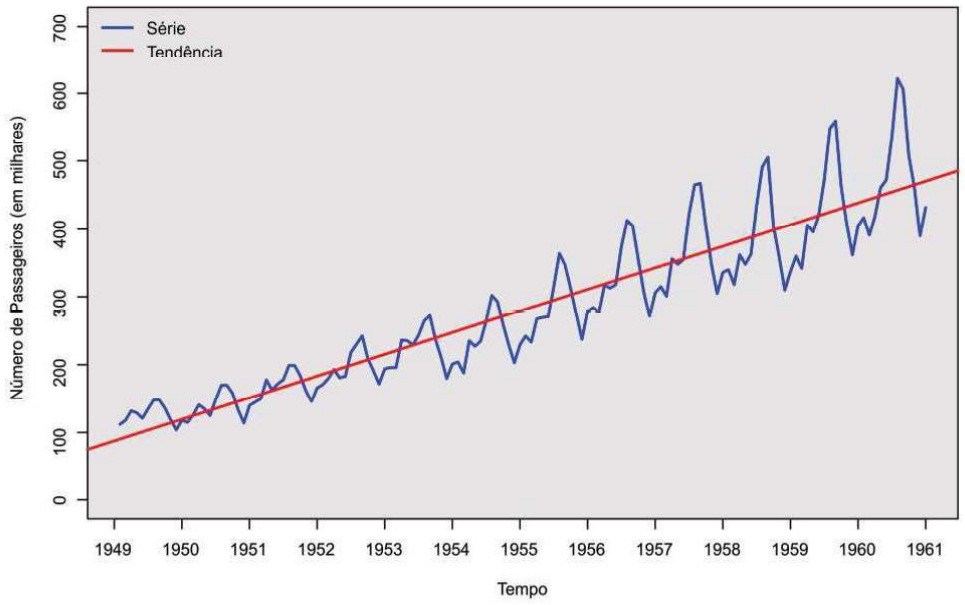


FIGURA 1.1 – Exemplo de sazonalidade nos dados de demanda, padrões de comportamento que se repetem em épocas específicas do ano.

Fonte: (Oper Data, 2019)

1.5 Estrutura do trabalho

O presente trabalho é constituído de 5 capítulos ordenados da seguinte forma:

No Capítulo 1 foi realizada a introdução ao assunto, abordando a definição do problema, a justificativa da escolha do tema, os objetivos definidos para o estudo e as delimitações do mesmo.

O Capítulo 2 apresenta a fundamentação teórica utilizada no trabalho, conceitua previsão de demanda por transporte aéreo e trata dos métodos presentes na literatura.

A metodologia é abordada no Capítulo 3, onde é apresentado o desenvolvimento do algoritmo, detalhado o processo de obtenção do conjunto de dados, método de pré-processamento e criação da base de treino, teste e validação do modelo.

O Capítulo 4 aborda os fundamentos de aprendizado profundo, redes neurais e NLP

usados no trabalho além dos resultados obtidos e da interpretação dos mesmos.

Por fim, o Capítulo 5 apresenta uma conclusão acerca do estudo proposto, apresentando também sugestões que futuramente podem ser aplicadas para a evolução do tema na área de transporte aéreo e possíveis melhorias do modelo.

2 Fundamentação Teórica

Previsões de demanda baseiam-se em identificar padrões, relações ou tendências. Uma vez que um desses fatores seja observado e compreendido, mesmo que parcialmente, é possível projetar o comportamento da demanda com relação ao futuro, possibilitando previsões acerca da mesma. Em suma, trata-se de conhecer o passado para projetar o que virá a acontecer no futuro.

Os métodos tradicionais de previsão de demanda por transporte aéreo são quantitativos, os quais incluem os modelos de tendência, que englobam a tendência simples e séries temporais, além destes há os métodos de regressão causal, sendo o econométrico o mais usado. A sofisticação desta última metodologia está relacionada à disponibilidade das informações socioeconômicas do mercado local (população, consumo de energia e PIB, entre outros), denominadas variáveis explicativas (COSTA *et al.*, 2008).

Os modelos baseados em séries temporais trabalham com previsões a partir da continuação de dados históricos, no caso, a variação da demanda por transporte aéreo no passado serve de base para a projeção da demanda futura. Por sua vez, os modelos explanatórios atuam baseando-se em como as variáveis independentes se relacionam com a variável dependente, por exemplo, como o preço do barril de petróleo, PIB e tamanho da população afetam a demanda por transporte aéreo.

Pouco presentes na literatura, porém importantes para o entendimento do tema, os métodos qualitativos podem ser divididos em 3 grupos principais de acordo com (LOVATTI, 2018): Julgamento pessoal, Pesquisa de mercado e Delphi

Julgamento pessoal se baseia no conhecimento de executivos, gerentes e analistas de rotas, que através de sua experiência e informações de mercado, economia e turismo são capazes de prever a movimentação futura de passageiros.

A Pesquisa de mercado funciona através de entrevistas com o público alvo na tentativa de identificar tendências e padrões de comportamento. Na prática, são aplicados questionários a uma amostra do mercado demandante.

Por fim, o método Delphi busca obter a previsão com base em um consenso de especialistas na área em questão. Por meio de um processo iterativo, os especialistas respondem

um questionário acerca da previsão de demanda estudada. Os resultados são reunidos, compilados e repassados aos especialistas, para que esses possam revisar suas respostas. O procedimento se repete até que se chegue a um consenso.

É importante ressaltar que existem diversas abordagens e técnicas diferentes para previsão de demanda e, embora elas possam ser aplicadas em conjunto a fim de se complementarem, cada uma possui suas vantagens e desvantagens a depender: do horizonte de tempo que se espera que a previsão seja capaz de ser relevante, da quantidade e do tipo de dados que se tem disponível, da precisão e nível de detalhe que se espera chegar com os resultados, e dos custos, seja com recursos para pesquisa ou prazo para entrega, que pode até mesmo determinar se o estudo é viável ou não.

A seguir são descritos diferentes estudos relacionados a previsão de demanda por transporte aéreo. É possível perceber que, ao longo dos anos, diferentes métodos foram aplicados e diversas variáveis foram estudadas. A literatura relacionada ao tema é bastante extensa e diversificada. Como cada país apresenta especificidades em sua relação com transporte aéreo, optou-se por dar destaque aos trabalhos relacionados ao mercado nacional de aviação, permitindo assim um maior entendimento das idiossincrasias do setor aéreo brasileiro e sua relação com a demanda. Entretanto, alguns trabalhos relacionados ao mercado internacional foram incluídos devido a sua alta contribuição metodológica ao tema, demonstrando como a tarefa de previsão de demanda pode ser abordada de diversas formas diferentes.

(BRONS *et al.*, 2001) estuda quais são os fatores determinantes para a elasticidade-preço na demanda por transporte aéreo. Para isso, os autores realizam uma metanálise reunindo 37 estudos em que um ou mais fatores da elasticidade-preço demanda foram estimados. Os trabalhos referem-se ao período de 1956 a 1998 e, em sua maioria, dizem respeito ao transporte aéreo estadunidense e europeu.

A média do compilado de estudos aponta para uma elasticidade-preço de -1,146, o que significa que, para uma redução no preço da passagem de 1%, tem-se um aumento médio na demanda de 1,146%, representando um aumento mais do que proporcional. O desvio padrão da distribuição de elasticidades do compilado é de 0.619.

Os resultados permitiram identificar uma elasticidade-preço consideravelmente menor para os viajantes de classe executiva do que para os da classe econômica, o que indica que os passageiros que viajam a trabalho são menos sensíveis aos preços, e também reforça a relação da renda com a elasticidade. Quanto a localização, esperava-se que os passageiros europeus demonstrassem uma maior sensibilidade aos preços do que os americanos devido a maior disponibilidade de modais de transporte na Europa, o que não se verificou; segundo os autores, a explicação pode estar na renda média, que é menor na Europa e acaba sendo um fator mais determinante. Já quando a análise é feita considerando as distâncias dos

voos, é possível verificar que os de longa distância tendem a ter uma elasticidade-preço mais alta do que os de curta distância, provavelmente pela tarifa maior exigir mais poder aquisitivo dos passageiros quando há variações percentuais. Por fim, uma correlação negativa foi identificada entre a elasticidade-preço e o ano do trabalho, indicando que, possivelmente, os passageiros se tornaram menos sensíveis ao preço ao longo dos anos.

(WEI; HANSEN, 2006) cria uma modelo de previsão de demanda agregada para o tráfico de passageiros em uma rede de aeroportos hub-and-spoke. O modelo baseado em regressão, utiliza as seguintes variáveis: frequência do serviço oferecido pela aérea entre o spoke e o hub, tamanho médio das aeronaves (em número de assentos), número de spokes atendidos pela companhia através do hub, frequência média do serviço entre o hub e os spokes, tarifa média por passageiro pagante, distância média das viagens na rede, número de passageiros locais que voam a partir do hub para qualquer spoke, número total de passageiros que iniciam a viagem em dado spoke, renda da população na área do hub e a capacidade de chegada de aeronave no hub.

De forma geral, os dados abrangem as variáveis de serviço das companhias, as condições socioeconômicas e demográficas dos passageiros locais e capacidade do aeroporto hub. A base de dados utilizada reúne informações do segundo trimestre dos anos 2000 e corresponde a 15 aeroportos hub dos Estados Unidos e 8 companhias aéreas.

Os resultados da regressão apontam para uma maior relevância da frequência de serviço entre o spoke e o hub do que para a rede como um todo, indicando que, em termos de demanda, investimentos no serviço oferecido pelas aéreas na primeira etapa da viagem na rede são mais importantes do que os oferecidos na segunda etapa. Com relação a tarifa cobrada, os resultados demonstram uma correlação negativa com a demanda, como era esperado. Por fim, a distância da viagem apresentou uma correlação positiva com a demanda, indicando que quanto maior a distância, maiores as chances de os passageiros optarem pelo modal aéreo.

(GROSCHE *et al.*, 2007) apresenta dois modelos gravitacionais para previsão de demanda por transporte aéreo entre pares de cidades. Os modelos utilizam variáveis que descrevem a atividade econômica geral e as características geográficas das cidades, e não incluem variáveis que descrevem o serviço aéreo. Assim, os modelos podem ser aplicados a pares de cidades onde há poucos dados disponíveis ou até mesmo onde atualmente nenhum serviço aéreo é fornecido, identificando uma possível demanda latente.

Os dados utilizados referem-se a voos entre a Alemanha e 28 países europeus no período de janeiro a agosto de 2004; são eles: população das áreas de captação da cidade, poder de compra da população dentro da área de influência do aeroporto, PIB do país de cada aeroporto, distância geográfica entre os aeroportos e o tempo de viagem para cada par de cidades. Os resultados revelam um modelo estatisticamente expressivo, com todas

as variáveis significantes ao nível de 1% de significância e um bom ajuste dos dados, apresentando coeficiente de determinação (R^2) igual a 0,761 em seu melhor modelo.

(ROCHA, 2010) pesquisou padrões na demanda por transporte aéreo regional utilizando uma abordagem econométrica clássica a partir de dados referentes ao período de 1997 a 2001 do antigo órgão regulador do transporte aéreo brasileiro, o Departamento de Aviação Civil (DAC). Por meio de regressão linear, (ROCHA, 2010) identifica que apenas os valores de PIB e Yield não são suficientes para proporcionar um bom ajuste dos dados. Ao adicionar a influência da capacidade no transporte de passageiros, expressa em ASK, o R^2 (medida estatística de quão próximos os dados estão da linha de regressão ajustada) apresenta significativa melhora, porém levanta-se também a discussão acerca da validade da regressão, já que ASK e RPK apresentam uma forte endogenia, ou seja, a disponibilidade de assentos não possui uma relação unívoca com o RPK, pois uma parte de sua variação é explicada pela própria variação do RPK.

Vale destacar que em seu estudo, (ROCHA, 2010) também incorporou os efeitos da integração da TAM regional pela TAM Linhas Aéreas em novembro de 2000 incluindo uma variável "dummy" na regressão representando os momentos pré e pós integração. Os resultados mostram que esse evento teve uma influência importante no RPK, muito provavelmente em função da diminuição de assentos disponíveis com o fim da TAM Regional.

A partir de seus resultados, (ROCHA, 2010) é capaz de indicar que havia uma demanda reprimida para a aviação regional, de modo que um aumento na disponibilidade de assentos, por meio de investimentos em infraestrutura e aumento na frequência de voos, é apontado como medida favorável para estimular o mercado.

(CONDÉ, 2011) estudou a demanda por transporte aéreo para a cidade do Rio de Janeiro em 2014, a partir de dados de 2003 a 2009 do BTD (Brazilian Tranportation Database). Utilizando modelos de regressão múltipla, (CONDÉ, 2011) incorporou os dados relativos ao movimento de aeronaves, movimento de passageiros nos aeroportos Santos-Dumont e Galeão para os segmentos doméstico e internacional, yield médio doméstico, PIB em reais, taxa de inflação (IPCA) e a taxa de câmbio real-dólar.

Com a análise dos dados, a autora demonstrou os efeitos da crise no setor aéreo brasileiro ou "apagão aéreo"ocorrido entre 2006 e 2007, com uma redução na movimentação de passageiros internacionais e um ligeiro aumento na movimentação doméstica. Também demonstrou os efeitos do code-share (compartilhamento de vôos) entre a TAM e a Varig entre os anos de 2003 e 2005; com o fim do compartilhamento entre as empresas, houve aumento da concorrência e, consequentemente, redução do yield, ou seja, redução dos preços cobrados pelas companhias, o que levou a um aumento na demanda.

Em seus resultados, (CONDÉ, 2011) identifica uma tendência de crescimento acelerado para o movimento doméstico de passageiros; 4,8%, 3,4% e 6,3% ao ano para os cenários

de manutenção, aumento e redução do yield, respectivamente; enquanto a movimentação de passageiros internacionais se mostra bastante variável para os dados utilizados e sem tendência específica.

(DINIZ, 2013) investigou a adequação de um projeto de ampliação nominal do aeroporto de Marabá usando dados da Infraero para movimentação de passageiros entre 1990 e 2010. Analisando os dados de demanda, foi possível identificar um crescimento abrupto no início de 2007, marcado pelo início das operações da Gol em Marabá. Para explicar a forte mudança na demanda gerada por esse evento, (DINIZ, 2013) incorporou ao seu modelo uma variável categórica associada à entrada da Gol no mercado. Dessa forma, o modelo baseado em regressão linear utilizado, levou em consideração os dados de PIB, Yield e o "efeito Gol".

Em seus resultados, (DINIZ, 2013) previu crescimentos anuais na demanda entre 2011 e 2014 de 7,33%, 11,25% e 14,05% nos cenários "pessimista", "base" e "otimista" (baseados em premissas de evolução do PIB e do yield) respectivamente. Com isso, em uma estimativa realista de acordo com o autor, foi possível identificar uma forte inadequação do projeto, evidenciando a necessidade de ampliação ainda maior das capacidades do aeroporto para atender a demanda.

(FALCÃO, 2013) busca, a partir de um modelo econométrico de previsão de demanda, analisar os impactos do aeroporto de Manaus (SBEG) no turismo da região norte do Brasil. Para isso, utiliza dados de 2003 a 2010 com o objetivo de estimar a demanda para o ano de 2014. Os dados utilizados incluem: movimentação de passageiros no aeroporto para os segmentos domésticos e internacional de acordo com a Infraero, o yield médio doméstico, PIB em reais, a taxa de inflação (IPCA) e taxa de câmbio real-dólar.

Em seu trabalho, a autora também inclui os efeitos de eventos como o code-share entre a Varig e a TAM entre os anos de 2003 a 2005 e a crise no setor aéreo brasileiro ou "apagão aéreo"ocorrido entre 2006 e 2007 por meio de variáveis categóricas. Outro evento incorporado na análise foi a crise financeira mundial do subprime entre 2008 e 2009. Diferente de outras localidades, Manaus teve um ligeiro aumento de demanda tanto doméstica, quanto internacional durante o "apagão aéreo", porém, com o fim do code-share entre a Varig e a TAM, também presenciou um considerável aumento na demanda. Já durante o período da crise do subprime, diferente do esperado, foi possível identificar um ligeiro aumento na movimentação de passageiros tanto internacionais quanto domésticos, evidenciando o pouco impacto da crise no setor aéreo de Manaus.

Por fim, os resultados de (FALCÃO, 2013) apontam para uma demanda de 3,8 milhões de passageiros para 2014, muito além da capacidade declarada na época do aeroporto de 2,5 milhões, demonstrando assim, a necessidade de investimentos na infraestrutura a fim de aumentar a capacidade de operação e atender a demanda prevista.

(LIMA, 2013) também utilizou regressão e o modelo econométrico com a intenção de prever a demanda para o aeroporto Eurico de Aguiar Sales (SBVT) em Vitória (Espírito Santo), com base em dados de 2002 a 2011 da Infraero e ANAC. As variáveis consideradas foram: movimento de passageiros locais, yield médio doméstico nacional (pela não disponibilidade do yield específico) e o PIB em reais, considerando as taxas de inflação.

Ao analisar os dados, (LIMA, 2013) identificou a mesma correlação negativa, recorrente na literatura, entre yield e demanda por transporte aéreo. Ao longo do período estudado, foi possível identificar uma constante redução no yield acompanhada de um também constante aumento na demanda. Da mesma forma, também foi identificada a correlação positiva entre o PIB e a demanda, com o aumento do PIB acompanhado ao aumento da demanda ao longo dos anos. Todavia, foi verificado um elevado aumento na movimentação de passageiros no início de 2011, tendo praticamente dobrado em relação aos anos anteriores. Essa quebra de tendência não pôde ser explicada por nenhuma das variáveis utilizadas pela autora, o que a levou a incluir uma variável categórica "dummy" associada a essa mudança abrupta em seu modelo.

Em seus resultados, (LIMA, 2013) compara diversos modelos de regressão e, de acordo com seus critérios de significância estatística, identifica o modelo que leva em conta os valores de PIB, yield e a "dummy" representando o aumento na demanda no início de 2011 como o modelo mais adequado para previsão no contexto analisado.

(PAMPLONA; OLIVEIRA, 2015) estudaram os impactos do crescimento da demanda de passageiros em um aeroporto compartilhado entre pessoal civil e militar. Para isso, tomou o Aeroporto Internacional de Salvador (SSA) como estudo de caso e usou dados da Infraero, ANAC, Eletrobras (Empresa Brasileira de Energia Elétrica) e IPEA (Instituto de Pesquisa Econômica Aplicada) do período de 2002 a 2013. Os dados utilizados foram: movimentação doméstica de passageiros, consumo de eletricidade local, poder de compra da população e o yield.

Neste estudo, os autores optaram por tomar o consumo de eletricidade como parâmetro de consumo, em substituição ao comumente usado PIB, pela não possibilidade de desagregar o PIB por região. Também optaram por incorporar o poder de compra da população ao modelo, utilizando para isso a quantidade de crédito para consumo dentro do mercado de crédito. A análise inicial dos dados, permitiu identificar uma correlação positiva entre o consumo de energia elétrica e a demanda e entre a disponibilidade de crédito e a demanda, também se constatou uma correlação negativa entre o yield e a demanda.

Aplicando os dados no modelo de regressão proposto, (PAMPLONA; OLIVEIRA, 2015) chegaram a uma previsão de aumento na demanda de, em média, 64730 e 354123 passageiros anuais nos cenários "pessimista" e "otimista", respectivamente, no que diz respeito ao

crescimento esperado para o consumo de energia. Utilizando simulações de cenário relacionados a diferentes expectativas para as variáveis consideradas, (PAMPLONA; OLIVEIRA, 2015) concluíram que a demanda provável para os próximos anos não deveria interferir de maneira significativa nas operações militares realizadas no aeroporto.

(BENDINELLI; OLIVEIRA, 2015) buscam analisar o sucesso do processo de privatização do Aeroporto Internacional de Confins em Belo Horizonte (Minas Gerais) sob o ponto de vista da demanda por transporte aéreo, uma vez que a lucratividade do empreendimento depende fundamentalmente da demanda. Para isso, foi utilizado também um modelo de regressão econométrico, com base nas seguintes variáveis explicativas: PIB, yield, preço do barril de petróleo WTI em dólares e taxa de câmbio real-dólar. Os dados compreendem o período de 2000 a 2012. Os autores também incorporaram em sua análise variáveis categóricas para explicar a variação na demanda provocada por eventos como o code-share entre Varig e TAM e o crescimento acelerado do número de passageiros embarcados e desembarcados domésticos totais no início de 2011.

Para o aeroporto analisado, foi encontrado uma relação de elasticidade em que, para cada 1% de elevação no preço das passagens aéreas, se reduz a demanda em 0,13%. Para a elasticidade-renda da demanda foi encontrado uma relação de 2,05% de aumento na demanda por transporte aéreo para cada 1% de aumento na renda dos passageiros. Com esses dados foi possível concluir que o sucesso do empreendimento de privatização do aeroporto está altamente relacionado ao crescimento econômico do país.

(OLMEDO, 2016) propõe dois métodos para previsão de demanda aeroportuária baseados em técnicas de reconstrução e aprendizagem, e compara os resultados obtidos na previsão de pontos (regressão) e na previsão de sinal (classificação). Os modelos utilizam como dados a série histórica de pousos diários no aeroporto de Palma de Mallorca na Espanha durante o período de janeiro de 2000 a dezembro de 2010.

A abordagem baseada em reconstrução busca prever a evolução da série temporal de aterrisagens, considerando a série como uma projeção unidimensional gerada por um sistema multivariado desconhecido; de forma que o objetivo do algoritmo é desdobrar a projeção de volta para um estado de espaços multivariados com a mesma dinâmica do original. A autora utiliza para isso o KNN (K-nearest neighbors), algoritmo tradicional de aprendizado de máquina para predição local. Já a abordagem de aprendizagem, utiliza redes neurais artificiais, algoritmo também tradicional do aprendizado de máquina, mas voltado para sistemas complexos devido ao alto poder de generalização de uma rede.

De acordo com os resultados, ambos os métodos obtiveram erro quadrático médio normalizado menor do que 1, indicando melhores previsões do que um preditor baseado na média dos valores. A abordagem de reconstrução demonstrou melhores resultados na previsão de sinais, já a abordagem de aprendizagem se saiu melhor na previsão de pontos,

o que indica que a demanda é um sistema complexo dependente de muitos fatores; uma vez que a abordagem de reconstrução é mais adequada para sistemas de baixa dimensionalidade, enquanto que a abordagem baseada em aprendizado é mais apropriada para sistemas complexos, difíceis de serem reconstruídos.

(VARELLA, 2016) propõe um modelo econométrico com o objetivo de prever a demanda para o Aeroporto Internacional de Recife (SBRF), a fim de avaliar a necessidade de expansão do aeroporto. Em seu modelo, (VARELLA, 2016) utiliza dados de 2002 a 2016 e inclui: a movimentação doméstica de passageiros, o PIB, o yield, o code-share entre a Varig e a TAM, a crise financeira de 2009 e o "apagão aéreo" ocorrido entre 2006 e 2007. Diferentemente dos trabalhos anteriores, com foco em outras localidades, a análise inicial dos dados não permitiu encontrar evidências do impacto desses eventos no padrão de evolução de demanda do aeroporto de Recife, porém foi possível identificar uma correlação positiva entre o PIB e a demanda e negativa entre o yield e a demanda, como o esperado.

Com base em diferentes cenários para as variáveis explicativas PIB e yield, o autor previu que de 2015 até 2020 haveria um aumento de demanda na ordem de 6,40% para o cenário pessimista e 16,45% para o cenário otimista no aeroporto de Recife, crescimento modesto segundo o autor e que não justificava uma possível expansão do aeroporto. Apesar das características turísticas da cidade, os resultados encontrados apontaram para uma elasticidade-preço da demanda baixa, para explicar esse resultado o autor levantou a hipótese de a maior parte dos passageiros de Recife serem viajantes em família, que planejam com antecedência e priorizam a segurança e garantia da viagem, não se importando muito com o preço a ser pago.

(PLAKANDARAS *et al.*, 2019) apresentam um modelo de previsão de demanda por transporte aéreo, rodoviário e ferroviário para o mercado doméstico nos EUA. O modelo é baseado em econometria e aprendizado de máquina. Os autores utilizam dados do período de 2000 a 2015 e as seguintes variáveis: número de passageiros domésticos em voos regulares, número de milhas por passageiro ferroviário, número de milhas percorridas por veículo (obtidas junto ao departamento de trânsito), preço do querosene de aviação, preço do petróleo bruto (ferroviária e rodoviária), PIB e o CFNAI (Chicago Fed National Activity Index). O CFNAI é um índice mensal criado para medir a atividade econômica geral americana.

O método utilizado baseia-se em SVM (support vector machine), um algoritmo de aprendizado de máquina supervisionado que analisa os dados de entrada e reconhece padrões, podendo ser usado para classificação ou regressão. Os autores avaliaram o modelo para diversas combinações de uso de variáveis e seus achados sugerem que o transporte aéreo é fortemente impulsionado pelos custos do combustível, o que diverge da literatura existente. Por fim, ao comparar os resultados com outros modelos, os autores concluem que o modelo baseado em aprendizado de máquina produz previsões fora da amostra mais

exatas do que os modelos econométricos clássicos.

(FRAZÃO; OLIVEIRA, 2020) analisa a influência da distribuição de renda na demanda por transporte aéreo, utilizando para isso o índice Gini, que mede a desigualdade de renda entre a população. O trabalho tem foco em identificar os impactos na elasticidade-preço da demanda da entrada de novos consumidores no mercado, identificados como a "nova classe média" e frutos da melhor distribuição de renda ocorrida no Brasil no final dos anos 2000.

Os autores consideraram diversos pares origem-destino para estimar a demanda agregada, utilizando dados de 2000 a 2013 que incluem: movimentação doméstica de passageiros, população (média geométrica do par origem-destino), renda per capita (calculada como a média geométrica do PIB no par origem-destino), yield, codeshare entre Varig e TAM, "apagão aéreo", crise financeira de 2009 e a presença de startups low cost (consideradas as empresas Gol e Azul no início de suas operações).

Em seus resultados, foi identificada uma correlação positiva da população, renda e presença de startups com a demanda, enquanto yield, codeshare Varig e TAM, "apagão aéreo" e a crise financeira apresentaram correlação negativa com a demanda, como esperado. Quanto à distribuição de renda, os resultados indicam uma correlação positiva com a elasticidade-preço da demanda, o que resultou em um efeito intensificador no aumento de passageiros transportados com a redução do yield. De acordo com os autores, o efeito da distribuição de renda na demanda abre espaço para crescimento via maior desregulação do mercado, por atrair a participação de low costs e induzir a queda de preços e assim viabilizar a entrada dos novos consumidores.

A tabela 2.1 resume a abordagem, o período dos dados coletados e as principais variáveis usadas em cada estudo.

TABELA 2.1 – Recorte da literatura sobre previsão de demanda por transporte aéreo

Autor e Ano	Dados	Abordagem	Principais Variáveis
Brons et al (2001)	1956-1998	Metánalise	Outros estudos (metánalise)
Wei e Hansen (2006)	2000-2001	Econometria	Yield, Frequência do serviço, Tamanho das aeronaves, Distância das viagens, Renda, Capacidade do terminal aeroportuário
Grosche et al (2007)	2004-2005	Modelo Gravitacional	População, Poder de compra da população, PIB, Distância das viagens, Tempo das viagens
Rocha (2010)	1997-2001	Econometria	PIB, Yield, ASK, Eventos
Condé (2011)	2003-2009	Econometria	Yield, PIB, Taxa de câmbio (real-dólar), Eventos
Diniz (2013)	1990-2010	Econometria	Yield, PIB, Eventos
Falcão (2013)	2003-2010	Econometria	Yield, PIB, Taxa de câmbio (real-dólar), Eventos
Lima (2013)	2002-2011	Econometria	Yield, PIB, Eventos
Pamplona e Oliveira (2015)	2002-2013	Econometria	Yield, Consumo de energia, Disponibilidade de crédito
Bendinelli e Oliveira (2015)	2000-2012	Econometria	Yield, PIB, Preço do combustível, Taxa de câmbio (real-dólar), Eventos
Olmedo (2016)	2000-2010	Aprendizado de Máquina	Série Temporal
Varella (2016)	2002-2016	Econometria	Yield, PIB, Eventos
Plakandaras et al (2019)	2000-2015	Aprendizado de Máquina	PIB, Preço do combustível, Atividade econômica geral
Frazão e Oliveira (2020)	2000-2013	Econometria	Yield, População, Renda, Eventos

Pode-se destacar desse recorte da literatura que diferentes fatores têm influência sobre a demanda aérea, como preço do bilhete, parâmetros macroeconômicos, sociais, políticos e eventos (tanto internos quanto externos ao setor aéreo que de alguma forma o impactam). Muitos desses fatores são de difícil captação, seja por indisponibilidade de dados ou

dificuldade de aproximação. Outra limitação dos modelos causais é a previsão indireta, ou seja, nesses modelos é preciso identificar como e quais fatores influenciam a demanda e, a partir de uma previsão desses fatores (PIB, Yield, Taxa de Câmbio etc), prever a demanda futura. Além de elevar o grau de incerteza sobre a previsão final, esse método é penalizado por possíveis multicolinearidades entre as variáveis explicativas, o que dificulta o entendimento dos efeitos isolados de cada variável sobre a demanda.

Devido aos recentes avanços na área de aprendizado de máquina e processamento de linguagem natural e seus notáveis resultados na literatura de diversas áreas, pretende-se com este trabalho apresentar uma nova abordagem para a previsão de demanda por transporte aéreo, baseado em NLP e redes neurais artificiais para regressão utilizando notícias relacionadas a transporte aéreo de jornais.

3 Metodologia

O método adotado neste estudo tem como objetivo prever a variação de demanda por transporte aéreo a partir de notícias de jornais relacionadas ao setor aéreo. Para isso é adotada uma abordagem baseada em processamento de linguagem natural e redes neurais profundas. A base de dados utilizada é composta por notícias do site G1 de 2006 a 2019. A preparação e o pré-processamento de dados foi feito por meio de programação em linguagem Python e a rede neural foi criada utilizando o framework PyText. A base de dados utilizada e todo as as rotinas implementadas estão disponíveis na página do projeto no GitHub (FELIPELEO1995, 2021).

3.1 Referencial Metodológico

3.1.1 Aprendizado de máquina

A ideia de inteligência artificial foi popularizada por Alan Turing ainda na década de 1950. Para ele, uma inteligência artificial seria a capacidade que uma máquina teria de imitar o comportamento humano. Com o objetivo de "imitar"a inteligência humana, diversos métodos computacionais surgiram e continuam surgindo até hoje baseados no funcionamento do cérebro humano, dentre os mais populares está o aprendizado de máquina.

Nós seres humanos, aprendemos sobre o mundo desde o momento em que nascemos e começamos a observar e interagir com o ambiente ao nosso redor, como se tudo o que os nossos sentidos captam se transformassem em dados para o nosso cérebro e, a partir desses dados, começássemos a identificar padrões e nos tornássemos cada vez mais capazes de realizar tarefas. Baseado nesse conceito, surge o aprendizado de máquina, que segundo sua definição mais usual, é a ciência de fazer com que computadores realizem tarefas sem serem explicitamente programados.

O aprendizado de máquina pode ser dividido em 3 grandes grupos principais:

- **Aprendizado supervisionado**: Dados de entrada são rotulados e o algoritmo aprende

a rotular novos dados a partir de padrões encontrados nos dados de entrada;

- **Aprendizado não supervisionado**: Dados de entrada não são rotulados e o algoritmo aprende a rotular dados de acordo com as similaridades e diferenças identificadas nos dados de entrada;

- **Aprendizado por reforço**: Aqui os dados de entrada não são estritamente necessários e a tarefa não precisa ser previamente explicitada, o algoritmo deve aprender a realizar a tarefa por meio de reforços positivos e negativos a cada uma de suas ações.

Dentre as principais tarefas de aprendizado de máquina supervisionado, pode-se destacar as tarefas de classificação e regressão. As tarefas de classificação têm o objetivo de construir um classificador que determine a classe de novos dados não rotulados, como, por exemplo, um classificador que determina a raça de um cachorro a partir de suas características físicas. Já as tarefas de regressão têm como objetivo construir um previsor de valores numéricos específicos, como, por exemplo, um previsor de demanda por um produto específico a partir de seu preço e suas características.

No campo do aprendizado de máquina não supervisionado, a principal tarefa é a de clusterização, que tem como objetivo agrupar os dados disponíveis de acordo com suas similaridades sem ter rótulos definidos; por exemplo, ao dividir uma grande base de clientes de acordo com seus perfis de consumo, o algoritmo deve sozinho encontrar as relações de proximidade e separar os dados em quantos grupos forem determinados.

Na aprendizagem por reforço, a inteligência artificial assume o papel de agente que interage com o ambiente, a cada interação o algoritmo aprende qual padrão de ação deve ter para realizar determinada tarefa. Nesse tipo de aprendizagem as tarefas são as mais diversas, desde de recomendação de conteúdo em redes sociais até precificação dinâmica de produtos em lojas online.

O processamento de linguagem natural, como um subcampo da inteligência artificial, faz uso de todos os tipos de aprendizado de máquina e suas tarefas. Este trabalho, porém, concentra-se na tarefa de regressão do aprendizado supervisionado.

3.1.2 Aprendizado profundo

O aprendizado profundo ou deep learning, como é mais conhecido, é uma subárea do aprendizado de máquina, baseada em redes neurais artificiais profundas. Inspirado nas redes neurais biológicas, as redes neurais artificiais têm seu funcionamento baseado em neurônios, onde cada neurônio se conecta e troca informações com outro na tentativa de identificar padrões dentre as muitas possibilidades de conexões. As redes neurais são

dividias em camadas, as mais simples possuem uma camada de entrada, uma camada de saída e, entre elas, camadas ocultas. Cada camada é composta por neurônios, também chamados de nós, as diferentes ligações entre os nós formam uma rede. A figura 3.1 mostra uma representação de uma rede neural com duas camadas ocultas.

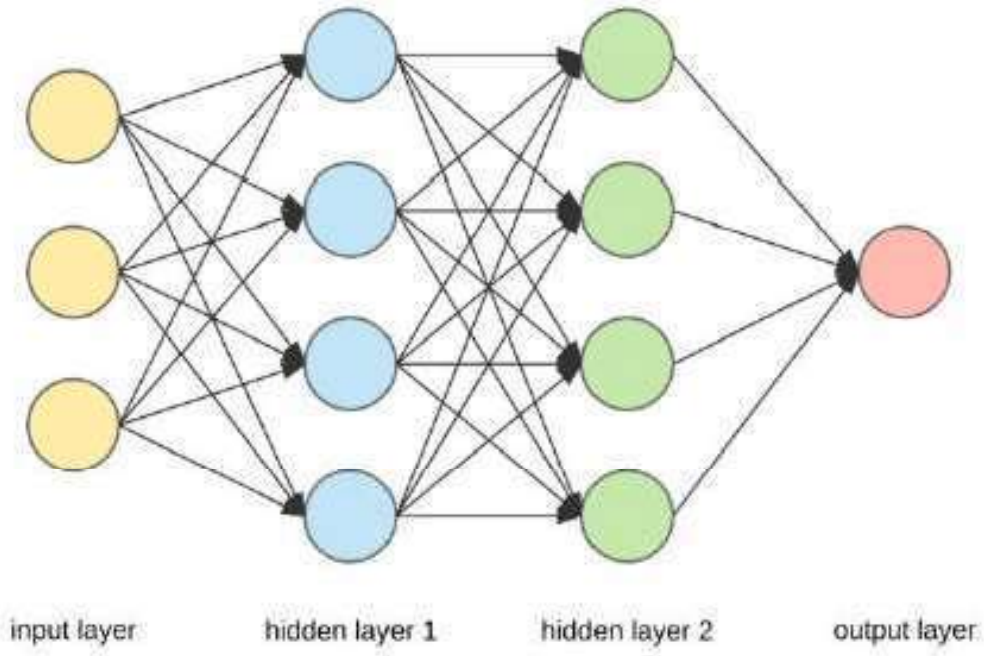


FIGURA 3.1 – Representação exemplo de uma rede neural com duas camadas ocultas.

Fonte: (OGNJANOVSKI, 2019)

As redes neurais têm o objetivo de encontrar a relação que conecta os dados de entrada com a saída esperada, para isso cada nó de uma rede neural representa uma função de ativação, que pode ser interpretada como uma abstração matemática. A função de ativação recebe o somatório das entradas (inputs) multiplicadas por seus respectivos pesos (Weights) e um viés (Bias) e gera uma saída (output) que por sua vez pode servir de entrada para outro nó. Os valores de peso e viés são ajustados durante a etapa de treinamento da rede. A figura 3.2 ilustra essa relação.

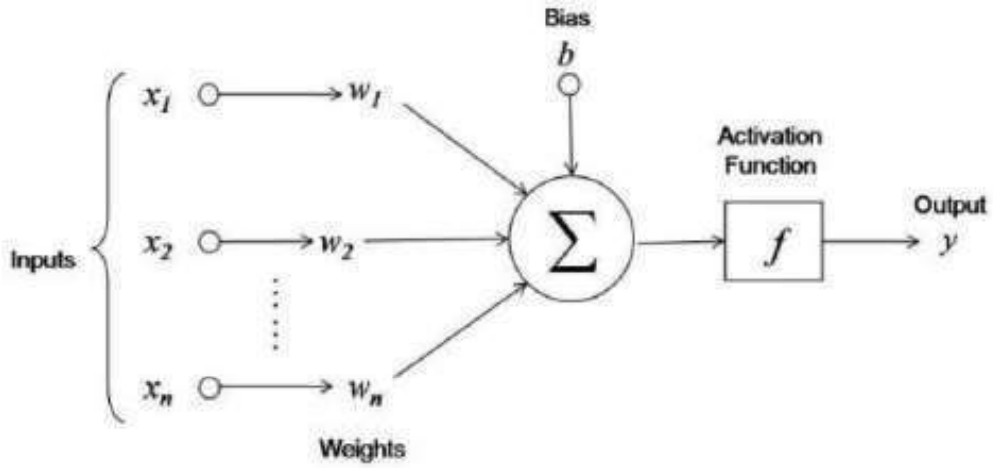


FIGURA 3.2 – Representação exemplo das relações de um neurônio ou nó de uma rede neural

Fonte: (PRATEEK, 2017)

Os valores de peso e viés são ajustados durante a etapa de treinamento por meio de tentativa e erro, até que a rede obtenha uma relação que seja suficientemente boa para relacionar os dados de entrada com a suas respectivas saídas esperadas.

Redes com apenas uma camada oculta são conhecidas como Perceptron e redes com múltiplas camadas ocultas são conhecidas como Multilayer Perceptron (MLP) e são consideradas as redes profundas mais simples. Apesar do seu alto poder preditivo e capacidade de realizar tarefas, as redes do tipo Perceptron ou Multilayer Perceptron apresentam limitações, principalmente quando trabalham com dados de alta dimensionalidade, como fotos e textos. Visando ultrapassar essas limitações, outras arquiteturas de redes profundas foram criadas.

3.1.2.1 Redes Neurais Convolucionais

Redes neurais convolucionais são redes profundas baseadas no processo de convolução. O processo de convolução é uma operação matemática que representa como um sistema linear opera sobre um vetor de entrada. Na prática, os dados de entrada passam por um filtro, também chamado de kernel, que reduz a dimensionalidade dos dados de entrada. A figura 3.3 ilustra a aplicação do filtro aos dados de entrada, reduzindo sua dimensionalidade.

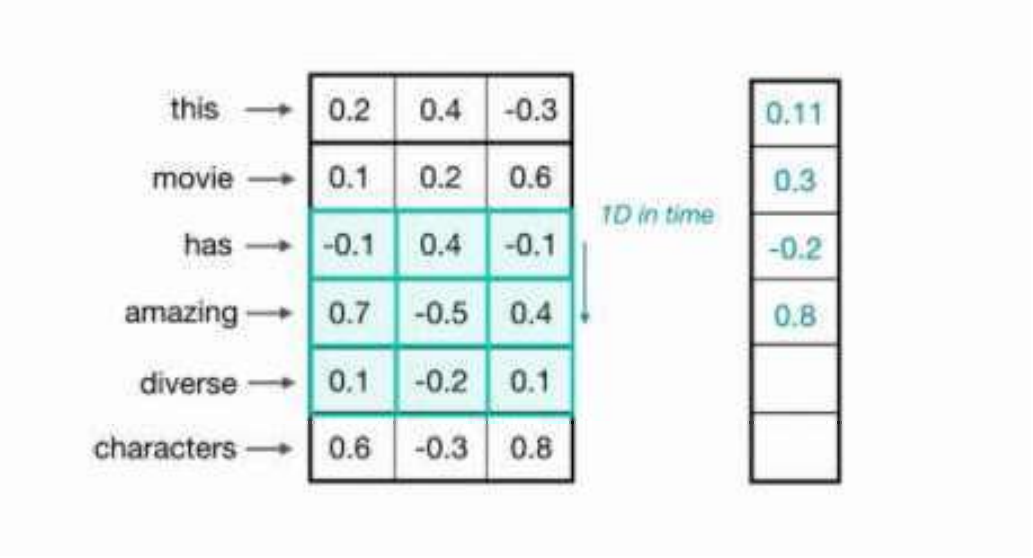


FIGURA 3.3 – Conforme a matriz azul desliza para baixo uma nova matriz é preenchida com dimensão menor que a inicial

Fonte: (CARNEIRO, 2020)

No exemplo acima, os dados de entrada estão representados na forma de matriz, sendo cada palavra um vetor de números reais. Neste caso o kernel é representado por uma matriz de dimensão menor que a matriz de entrada e, por meio da multiplicação de matrizes, uma nova matriz é gerada com dimensão menor que a inicial. Durante as etapas de treinamento de uma rede neural convolucional, os kernels são constantemente atualizados para melhor extrair as características principais da matriz de entrada.

Ao final do processo de convolução, também pode ser aplicado o Max pooling, que consiste em extrair o maior valor e mais relevante da matriz de saída, no caso do uso em textos tende a ser a palavra de maior destaque na sentença de entrada. Após essa etapa, os dados estão prontos para serem trabalhados em uma rede do tipo MLP, por exemplo.

3.1.3 Processamento de linguagem natural

Línguas naturais são aquelas que evoluem naturalmente por humanos devido ao uso e repetição, sem planejamento. Contrapondo as linguagens estruturadas, como a matemática ou a linguagem de programação, a linguagem natural é de mais difícil análise e processamento por um computador, tornando a tarefa de análise e processamento de linguagem natural praticamente inviável para algoritmos tradicionais, mas uma área promissora para o aprendizado de máquina.

O objetivo do processamento de linguagem natural é fazer com que a máquina seja capaz de entender e trabalhar com a linguagem humana, extraindo informações valiosas ao transformar dados não estruturados em estruturados. De forma a transformar textos

em linguagem natural em representações computacionais prontas para serem utilizadas, uma série de etapas deve ser observada de modo a promover a eficácia do modelo. A Figura 3.4 exemplifica etapas utilizadas no pré-processamento de documentos.

FIGURA 3.4 – Processo de pre-processamento de documentos em linguagem natural.

Fonte: (KATIUSKA *et al.*, 2018)

O texto bruto, da maneira que é encontrado na base de dados, é tratado como Corpus e passa pela primeira etapa; a preparação. Nessa etapa os dados são filtrados, de forma a permanecer na base apenas os textos de interesse ao modelo, depois são convertidos em um formato específico de acordo com cada modelo (CSV, TSV, entre outros tipos de armazenamento de dados).

A etapa de pré-processamento de dados textuais requer o uso de ferramenta computacional. Hoje, existem algumas ferramentas que suportam as principais tarefas de pré-processamento, das quais pode-se destacar a biblioteca NLTK ( ) para Python que é especializada em manipulação de dados textuais. Dentro da etapa de pré-processamento existem algumas subetapas que merecem atenção especial, são elas: (i) tokenização, (ii) remoção de ”stop words”, (iii) case-folding e (iv) radicalização e lematização.

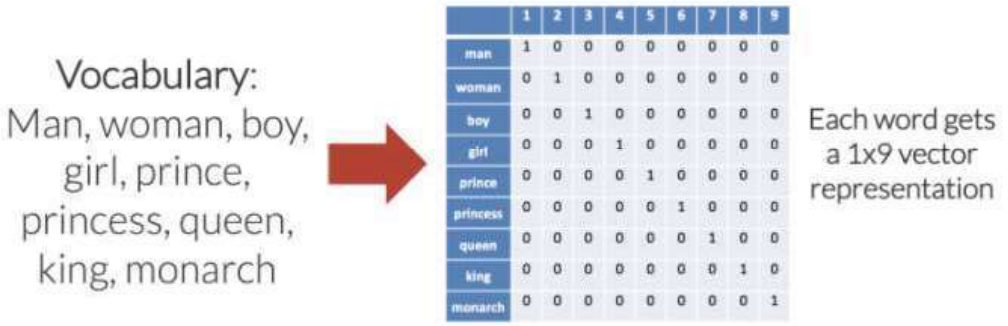
- **Case-folding**: Consiste em converter todo o texto para uma mesma caixa de caractere a fim de unificar pares como ”Avião”e ”avião”em uma mesma representação padrão. Idealmente busca-se unificar palavras distintas, mas de mesmo significado em uma forma equivalente, como os pares ”avião”e ”aeroplano”. Comumente se implementa listas relacionando palavras equivalentes, porém os custos de implementação são muito altos por depender de agentes humanos.

- **Tokenização**: Um token pode ser definido de forma abstrata como uma unidade de texto cujo valor semântico é útil para um dado propósito e, logo, depende da aplicação (KATIUSKA *et al.*, 2018). A princípio, para o computador, o texto de entrada é uma grande sequência de caracteres. A tokenização tem o objetivo de agrupar os caracteres em tokens ignorando certos caracteres de pontuação. Na prática, os tokens podem ser palavras, caracteres, sinais de pontuação ou sentenças, a depender do tokenizador usado.

- **Remoção de stop words**: De modo usual, stop words se referem a artigos, conjunções e preposições em textos em português brasileiro. Essas palavras têm um baixo poder discriminativo para as tarefas no geral, pois aparecem com muita frequência em todos os textos.

- **Radicalização e Lematização**: A Lematização trata de extrair o "lema"de cada palavra de acordo com o seu significado no dicionário, ou seja, trata-se de uma análise morfológica para extrair a palavra base, palavras como "aprendi", "aprenderam", "aprenderão"e "aprenderiam"são transformadas em "aprender". Já a radicalização busca extrair o radical das palavras, no caso "aprender"e todas as suas variações se tornariam "aprend".

Na etapa de representação, diversas técnicas diferentes podem ser implementadas com o objetivo de representar os textos computacionalmente. As técnicas mais simples representam cada palavra do texto como um vetor estático, de forma que a posição associada a palavra no vetor é preenchida com 1 e as demais posições com 0, esse tipo de representação é conhecida como one-hot encodding. A figura 3.5 exemplifica a aplicação dessa metodologia.



	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

FIGURA 3.5 – Exemplo de um esquema de representação one-hot para um vocabulário de nove palavras. Embeddings de palavras são lidos como linhas dessa tabela e são predominantemente compostos de zeros para cada palavra.

Fonte: (SHANE, 2018)

Apesar de bastante práticas, as representações one-hot apresentam a desvantagem de não contemplar a relação entre as palavras em sua representação. Nesse tipo de representação todos os vetores são ortogonais entre si, ou seja, o produto interno entre dois vetores ou duas palavras quaisquer terá resultado igual a 0, mesmo que as palavras representem entidades similares, como "aeroporto"e "aeródromo", por exemplo. Devido a essa limitação, esse tipo de representação perdeu espaço para as representações baseadas em word embeddings.

A representação por meio de word embeddings tem o objetivo de incorporar em sua representação as analogias entre as palavras, suas relações semânticas, relações sintáticas e os demais significados do contexto. Nesse tipo de representação, as palavras são representadas por vetores de números reais com dimensão n, ou seja, podem ser interpretadas como pontos em um espaço euclidiano de n dimensões. A figura 3.6 representa um exemplo de word embedding para n igual a 3.

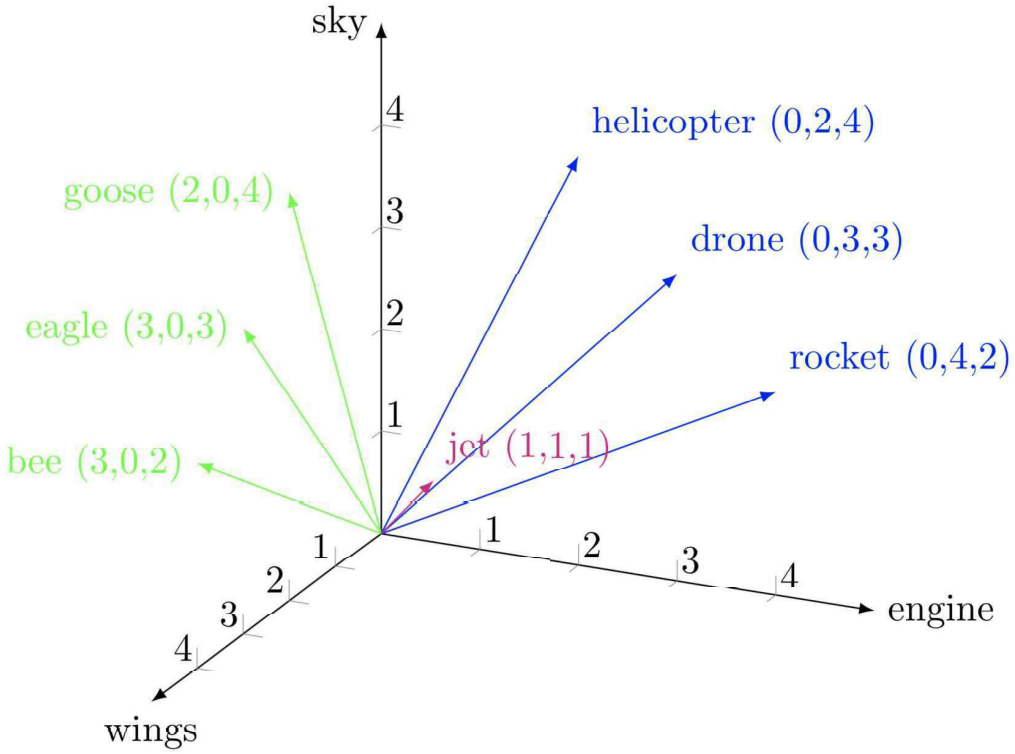


FIGURA 3.6 – Exemplo de palavras representadas por word embeddings de 3 dimensões

Fonte: (DESAGULIER, 2018)

Diferentes frameworks podem ser utilizados para gerar word embeddings, porém o BERT hoje se destaca como estado da arte no campo. BERT é o acrônimo para Bidirectional Encoder Representations from Transformers, que pode ser traduzido como codificador biderecional de representações baseado em transformadores. De maneira simplificada, BERT é uma rede neural profunda baseada na arquitetura Transformers capaz de gerar word embeddings de forma bidirecional.

A arquitetura Transformers vem revolucionando a área de processamento de linguagem natural desde seu lançamento em 2017. Ela foi criada especialmente para lidar com dados de entrada sequenciais (como texto em linguagem natural) e se baseia no mecanismo de auto atenção (Self-Attention) para computar as representações de sua entrada. Esse mecanismo funciona pesando a influência de diferentes partes dos dados de entrada, ou seja, dentro de uma mesma sentença, a representação de cada palavra vai ser influenciada

pelas palavras anteriores e posteriores a ela. Essa influência é ponderada de acordo com as palavras que são identificadas como mais relevantes para aquela representação em questão de acordo com o treinamento da rede. A figura 3.7 exemplifica essa relação.

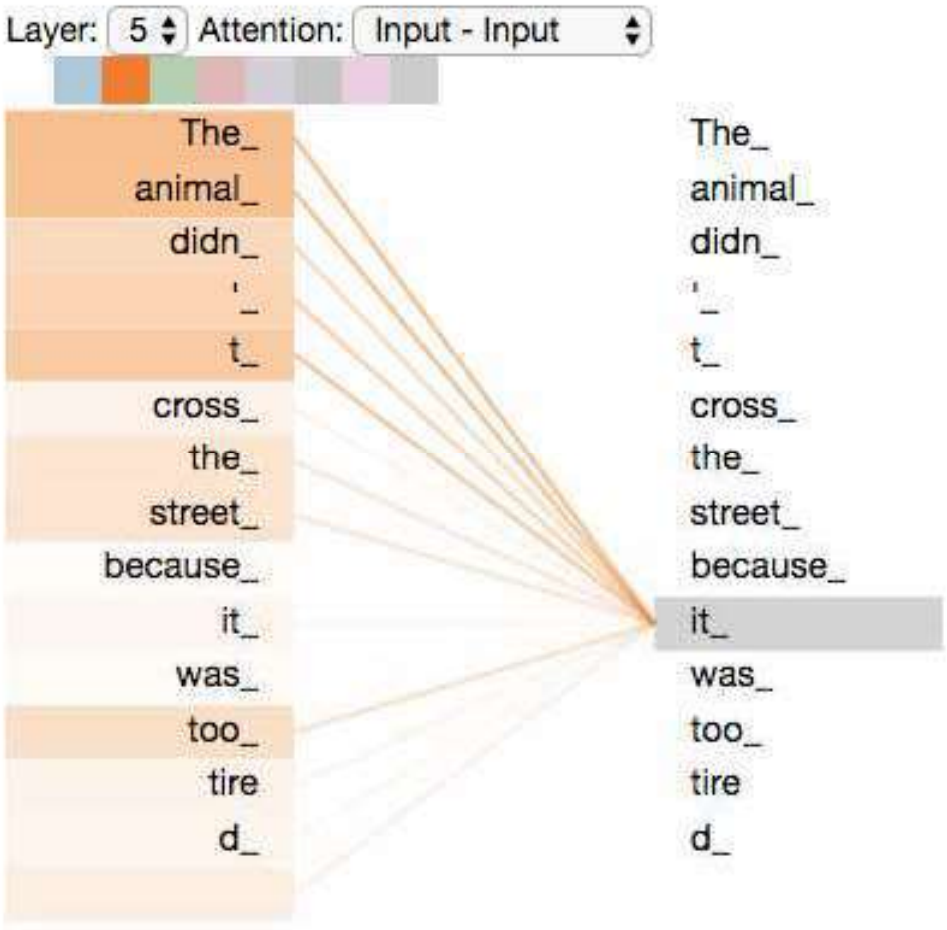


FIGURA 3.7 – Neste exemplo o mecanismo de atenção se concentra em ”The animal”para codificar a representação da palavra ”it”

Fonte: (ALAMMAR, 2018)

Baseado em transformadores, o BERT é uma técnica de código aberto criada pelo Google para fazer o pré-treinamento de modelos de processamento de linguagem natural baseada em redes profundas. Na prática, ele gera representações de palavras que podem ser usadas em outros projetos a partir da sua importação, poupando assim necessidade de processamento que muitas vezes são sequer viáveis para computadores comuns. A figura 3.8 exemplifica o uso de um embedding pré-treinado pelo BERT em uma tarefa de classificação.

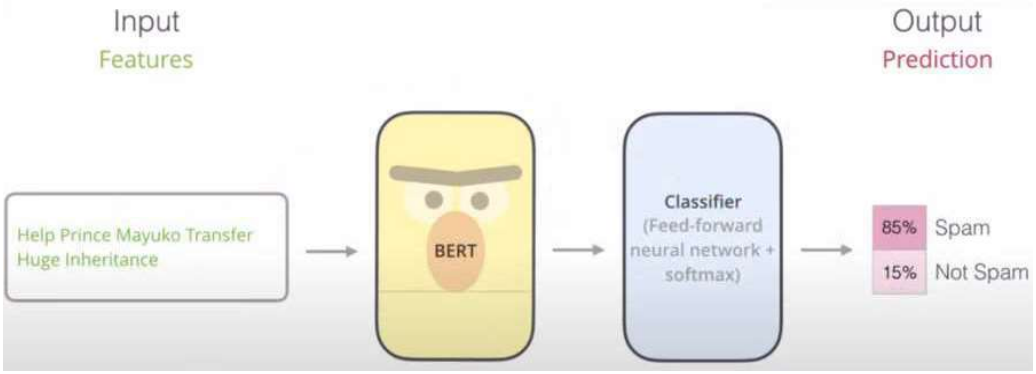


FIGURA 3.8 – Exemplo de uso de um modelo pré-treinado pelo BERT em uma tarefa de classificação

Fonte: (RIBEIRO, 2020)

Após passar pelas etapas de pré-processamento e representação, os dados estão prontos para servirem de entrada para a rede neural que de fato irá realizar a tarefa; neste caso, regressão. Diversas arquiteturas de redes podem ser usadas, a depender do tipo de tarefa que se pretende realizar, da quantidade e dos tipos de dados de entrada. Para cada arquitetura de rede existem hiperparâmetros a definir, como quantas camadas ocultas a rede terá e quantos nós serão usados em cada camada.

Em um primeiro momento pode-se pensar que quanto mais camadas e mais nós em cada camada, melhor será o treino e consequentemente melhores serão os resultados da rede. Entretanto, da mesma forma que um modelo muito simples pode sofrer com subajuste dos dados, um modelo muito complexo, ou seja, com muitas camadas e muitos nós, pode sofrer com sobreajuste. O subajuste ocorre quando o modelo não consegue capturar a relação entre os exemplos de entrada e os resultados esperados na saída, já o sobreajuste ocorre quando o modelo se ajusta de maneira excessiva aos dados de entrada e com isso perde poder de generalização para prever valores fora da base de treino. A figura 3.9 exemplifica de maneira gráfica as situações de sobreajuste, ajuste ótimo e subajuste dos dados.

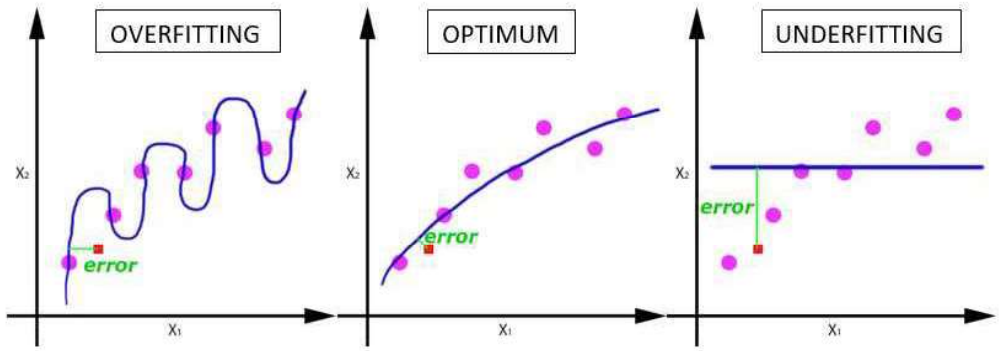


FIGURA 3.9 – Exemplo gráfico de situações de sobreajuste, ajuste ótimo e subajuste dos dados

Fonte: (LUCIAN, 2020)

Outros hiperparâmetros que podem ser citados são o tamanho do passo para cada iteração e a proporção dos dados que serão usados para treino, validação e teste. O passo de cada iteração define a quantidade de exemplos de entrada que a rede irá analisar antes de ajustar seus parâmetros e quantos ajustes serão feitos. Já a proporção entre treino, teste e validação estabelece a proporção dos dados usados para determinar os parâmetros, os hiperparâmetros e avaliar o desempenho do modelo, respectivamente. Em suma, a definição da arquitetura de rede usada e seus hiperparâmetros é essencial para garantir a qualidade do modelo. A figura 3.10 ilustra a relação entre a complexidade e a qualidade de um modelo.

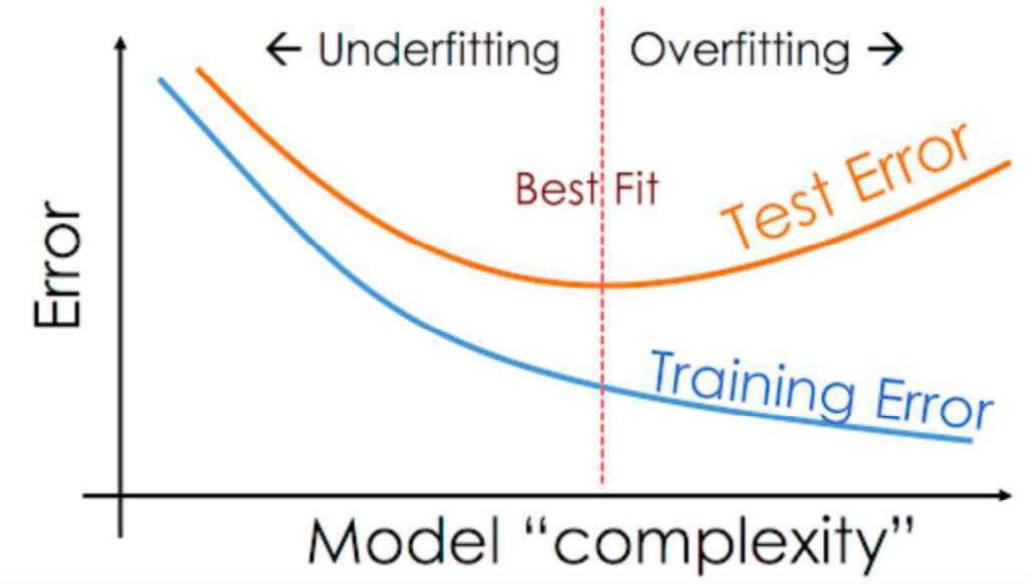


FIGURA 3.10 – Exemplo gráfico de relação entra qualidade e a complexidade do modelo. Deve-se ajustar o modelo a fim de minimizar os erros no conjunto de teste e no conjunto de treino, ou seja, minimizar tanto o erro sistemático quanto o aleatório

Fonte: (KUMAR, 2020)

Devido aos avanços na área de aprendizado de máquina e processamento de linguagem natural e a popularização de seu uso em diversas aplicações, hoje há diversas ferramentas que podem ser usadas para a definição dos hiperparâmetros, escolha da melhor arquitetura e estruturação do modelo, comumente disponibilizadas na forma de frameworks. Frameworks são conjuntos de bibliotecas ou classes implementadas em uma linguagem de programação específica com funções direcionadas para dar suporte a determinado tipo de tarefa. As principais vantagens em se utilizar frameworks em projetos são: agilidade no desenvolvimento, possibilidade de experimentação rápida de diversas opções de modelos, segurança (frameworks são testados por diversos programadores antes de se tornarem padrão na área) e padronização do modelo (importante para a reprodutibilidade).

Desenvolvido pela equipe de inteligência artificial do Facebook e lançado no final de 2018, o PyText se destaca como uma das principais opções de framework para proces-

samento de linguagem natural baseado em redes neuras profundas. O PyText é uma estrutura de modelagem voltada para NLP construída em PyTorch, que por sua vez é umas das principais bibliotecas de aprendizado de máquina e aprendizado profundo para Python. Seus principais destaques são seu baixo nível de abstração, facilidade de alteração de seus componentes e uso das técnicas mais recentes de aprendizado profundo e NLP.

3.2 Base de dados

3.2.1 Descrição dos dados

A base de dados utilizada é composta por 657719 notícias do site G1 e abrange o período de 2006 a 2018. Todas as notícias contêm pelo menos uma das seguintes tags: avião, aéreo, viagem, viajar, turismo, economia, linhas aéreas e aeroporto. A tags foram definidas de forma a filtrar apenas as notícias que, de alguma forma, têm relação com o transporte aéreo. A tag "economia" foi incluída devido à alta relação entre a demanda por transporte aéreo e o cenário econômico nacional, como demonstrou a revisão da literatura. A tag "linhas aéreas" foi incluída com o objetivo de filtrar as notícias relacionadas as companhias aéreas, já que nas notícias são sempre mencionadas como "Gol Linhas Aéreas", "Azul Linhas Aéreas" etc.

O armazenamento da base foi feito por meio de tabelas. Cada uma contém as notícias relacionadas a uma tag, com os dados divididos em 4 colunas: título da notícia, texto da notícia, data de publicação e link. A figura 3.11 exibe o cabeçalho e as primeiras linhas de uma das tabelas

Data	Titulo	Link	texto
01/01/2018	Festa da virada re�ne p�blico em diversos pontos de Manaus	https://g1.globo.com/busca/click?q=turismo&p=431&r=1628127461775&u=https%3A%2F%2Fg1.globo.com%2Fam%2...	Festa de r�veillon na Zona Leste de Manaus \u0000 Foto: Camila Batista/Semcom Festa de r�veillon na Z...
01/01/2018	'Ca�adores de argentinos' oferecem estadia irregular no RS no meio do caminho entre Buenos ...	https://g1.globo.com/busca/click?q=turismo&p=190&r=1628127418318&u=https%3A%2F%2Fg1.globo.com%2Frs%2...	Prefeitura de S�o Gabriel realiza campanha para incentivar formaliza��o de hospedagens \u0000 Foto: Repro...
01/01/2018	Milhares de pessoas lotam a orla de Macei\u0000 para dar boas-vindas a 2018	https://g1.globo.com/busca/click?q=turismo&p=432&r=1628127461781&u=https%3A%2F%2Fg1.globo.com%2Fal%2...	Espet\u0000culo da queima de fogos na orla de Macei\u0000 Cerca de 80 mil pessoas, de acordo com e...
01/01/2018	Festa de r\u0000veillon em Boa Vista re\u0000ne 50 mil pessoas debaixo de chuva no Parque Anau\u0000 ...	https://g1.globo.com/busca/click?q=turismo&p=436&r=1628127464338&u=https%3A%2F%2Fg1.globo.com%2Frr%2...	Festa da virada \u0000 tradicional e ocorre h\u0000 25 anos no Parque Anau\u0000, em Boa Vista \u0000 Foto: Alan Chaves/...

FIGURA 3.11 – Cabeçalho e primeiras linhas da tabela contendo as notícias para a tag "turismo"

Fonte: (SARMENTO, 2021)

Os dados de variação de demanda por transporte aéreo ano a ano foram obtidos diretamente da ANAC. Para este estudo, foi considerada a demanda total de passageiros, incluindo voos domésticos e internacionais dos anos correspondentes aos das notícias. A tabela 3.1 expressa a demanda por transporte aéreo em número de passageiros pagos ao longo dos anos estudados.

TABELA 3.1 – Demanda por transporte aéro no Brasil em número de passageiros pagos

Ano	Nacional	Internacional	Total	Variação
2005	38688977	10303017	48991994	
2006	43090736	10668703	53759439	9,73%
2007	47168910	12093479	59262389	10,24%
2008	50085395	13169708	63255103	6,74%
2009	56930824	12533806	69464630	9,82%
2010	69968162	15359523	85327685	22,84%
2011	81903711	17855206	99758917	16,91%
2012	88472681	18903861	107376542	7,64%
2013	89961353	19751225	109712578	2,18%
2014	95826631	21280165	117106796	6,74%
2015	96093055	21544630	117637685	0,45%
2016	88595389	20920201	109515590	-6,90%
2017	90576791	21890014	112466805	2,69%
2018	93609231	24133026	117742257	4,69%
2019	95101479	24134511	119235990	1,27%

Fonte: (ANAC, 2021)

3.2.2 Obtenção dos dados

Os dados foram obtidos por meio de um Web Crawler construído especificamente para a tarefa. Um Web Crawler, ou rastreador da web, é um programa ou script que navega na internet de forma automática e sistêmica com o objetivo de extrair informações das páginas que visita. O rastreador construído funciona, de forma resumida, acessando a página inicial do site G1, pesquisando por uma das tags no campo de busca, ajustando os filtros de busca do site para que todas as notícias dentro de um intervalo de datas sejam exibidas e capturando os links de cada notícia. Esses links posteriormente são acessados um a um por um algoritmo específico que extrai somente o texto da notícia de cada um dos links e o armazena na base de dados. O processo se repete de forma iterativa percorrendo todas as tags e todo o período de tempo de 2006 a 2018. O pseudocódigo 1 resume o funcionamento dos algoritmos, enquanto aos apêndices A, B, C, D e E apresentam os algoritmos usados em sua forma completa.

Algoritmo 1 Extração de notícias do site G1

Entrada: *link*, *tags*

Saída: *noticias*

```
Acessa a página do G1 pelo link fornecido
para cada tag em tags faça
    Pesquisa por tag no campo de buscas do site e filtra pelas notícias mais relevantes
    para cada semana de 2006 a 2018 faça
        Altera os filtros do site para mostrar as notícias da semana
        para cada noticia da página faça
            noticia ← ttulo, texto, data
            noticias ← noticias + noticia
        fim para
    fim para
fim para
```

3.2.3 Preparação e Pré-processamento

As rotinas de preparação e pré-processamento dos dados foram feitas usando linguagem Python, apoiado principalmente nas bibliotecas: NLTK, pandas e NumPy. O primeiro passo do tratamento dos dados foi remover todas as notícias da base que não possuíam texto (algumas notícias possuem somente título e vídeo). A segunda etapa foi a de limpeza dos textos, de acordo com os procedimentos apresentados no referencial metodológico: case folding, tokenização, remoção das stop words e radicalização e lematização. O pseudocódigo 2 resume o pré-processamento dos dados, enquanto o apêndice F apresenta o código completo.

Algoritmo 2 Pré-processamento dos textos

Entrada: *noticias*

Saída: *noticias_limpas*

```
para cada noticia em noticias faça
    noticia_caixaBaixa ← caixaBaixa(noticia.texto)
    para cada palavra em noticia_caixaBaixa faça
        se palavra não for uma stop word então
            palavra_stemizada_radicalizada ← stem_radi(palavra)
            texto_final ← texto_final + palavra_stemizada_radicalizada
        fim se
    fim para
    texto_final ← removeCaracteresEspeciais(texto_final)
    noticias_limpas ← noticias_limpas + texto_final
fim para
```

A preparação da base de dados seguiu com a união dos dados de variação de demanda

por transporte aéreo com as notícias em si. A junção se deu tomando a data como referência, assim cada notícia passou a ter a variação na demanda aérea do ano subsequente a sua publicação como rótulo associado. Em seguida, a coluna *data* foi excluída da base e manteve-se apenas as colunas texto e variação de demanda por transporte aéreo ou rótulo, neste trabalho utilizadas como entrada e saída ("x" e "y") do modelo. A figura 3.12 exibe o cabecalho e as primeiras linhas da base de dados pré-processada.

# rotulo	A texto
2.18	inflaca med indic prec consumidor ipcs segund seman mai desaceler 055 tax 002 pont percentual abaix...
4.70	setor mov sofr cris cresc 10 ano passos mg setor mov nad contr mar pass mg crise vend cresc 10 munic...
0.45	dol fech qued nest segundafeir 18 perd 024 r 22586 vej cotacaosaib maisacompanh cotaco ambient tranq...
2.69	dol oper qued nest quartafeir 13 recu 1 abaix r 4 apos dad melhor esper sobr comerci chin reviv apet...

FIGURA 3.12 – Cabeçalho e primeiras linhas da tabela final para a tag "economia"

Fonte: (SARMENTO, 2021)

A base de dados, após a formatação necessária para a inserção no modelo de redes neurais, foi aleatoriamente dividida em 3 conjuntos: teste, treino e validação. A proporção

adotada para a divisão foi baseada no que é mais frequente na literatura para trabalhos desse tipo e no Princípio de Pareto, que afirma que 80% das consequências são geradas por 20% das causas (ARNOLD, 2014) . Neste caso, a base de dados foi dividida em 20% para validação e 80% para treino e teste, destes 60% para treino e 20% para testes.

Após esses procedimentos, a base de dados foi dividida em arquivos separados, prontos para servirem de entrada para a rede neural modelada em PyText. O apêndice F apresenta o código utilizado nessa última etapa de preparação.

3.3 Aprendizado profundo

A implementação da rede, o treinamento e a geração dos modelos preditivos foram feitos utilizando-se o framework PyText. PyText é uma estrutura de modelagem de NLP baseada em aprendizado profundo construída em PyTorch, que, por sua vez é uma biblioteca de aprendizado de máquina de código aberto para Python. O PyText tem o funcionamento baseado em módulos, sendo cada um responsável por uma etapa da tarefa realizada pelo modelo preditivo. Essencialmente, cada módulo é uma classe Python diferente e, quando juntas, são capazes de ler os dados de entrada e gerar as saídas. A figura 3.13 descreve a relação entre os principais módulos e o funcionamento do framework.

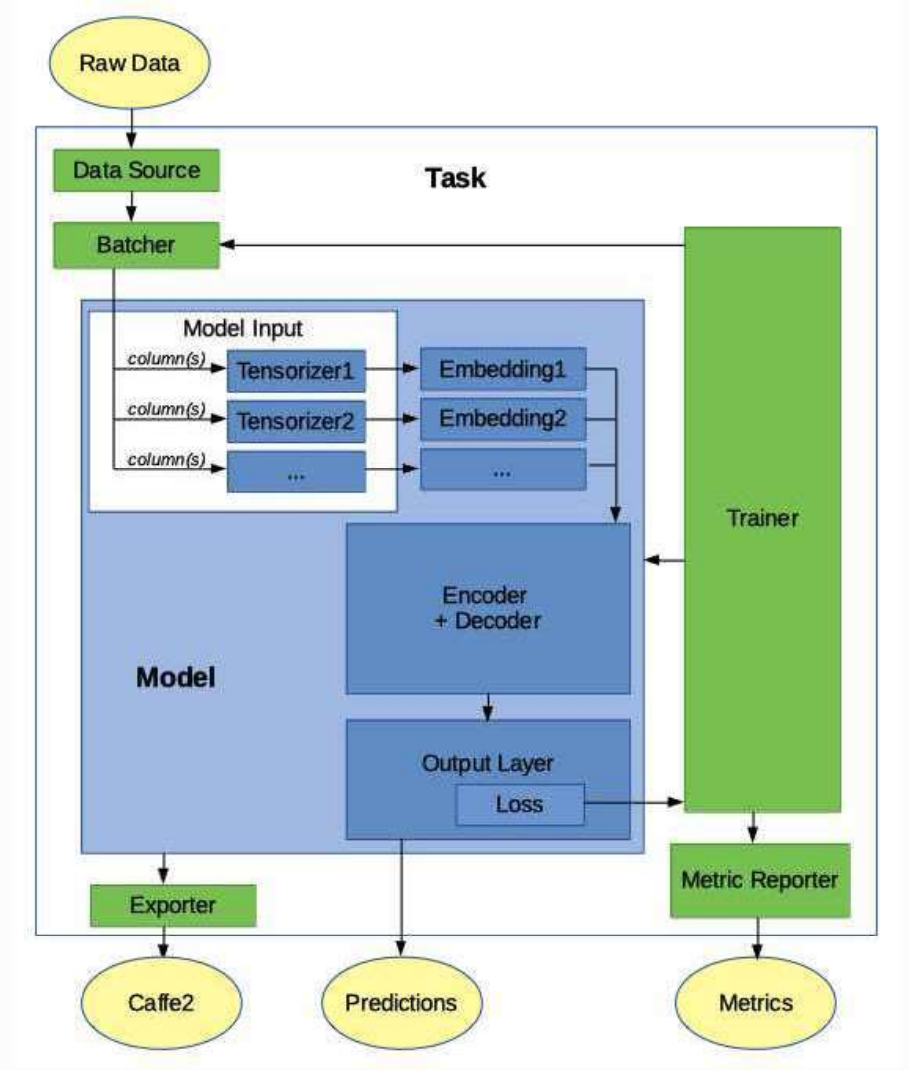


FIGURA 3.13 – Visão geral da arquitetura Pytext

Fonte: (PYTEXT..., 2021)

Para cada módulo, existem diferentes opções de classes a serem utilizadas e também é possível alterar e substituir componentes desses módulos, tudo de acordo com a tarefa e o modelo que se busca implementar. Entretanto, a hierarquia entre eles é fixa, sendo o módulo Task responsável pela interface com o usuário, entrada e saída do modelo, exportar modelos e configurações gerais do treinamento. Enquanto o módulo Model, submódulo de Task, é responsável pelo processamento dos dados e treinamento da rede neural. A seguir é explicado em detalhes o que representa cada um dos componentes presentes na figura 3.13.

- **Raw Data**: Dados de entrada em sua forma "bruta" ou ainda não processados;
- **Data Source**: Neste módulo é feita a leitura dos arquivos de entrada, sendo usadas classes diferentes para a leitura de arquivos em .csv e .tsv, por exemplo. Deve-se especificar os caminhos para os arquivos de teste, treino e validação;

- **Batcher**: Neste módulo, os dados de treino são separados em lotes e repassados para a rede neural. Sendo um dos hiperparâmetros, deve-se especificar o número de lotes a ser usado no modelo;

- **Model Input e Tensorizer**: Aqui o texto de entrada é lido como uma string e transformado em um tensor por uma família de módulos tensorizadores. Cada tensorizador tem um vocabulário próprio que é construído durante um escaneamento inicial dos dados. De acordo com seus vocabulários, cada tensorizador transforma uma linha do texto de entrada em um tensor saída, que basicamente será um vetor com os índices de cada palavra daquela linha do texto mapeadas de acordo com o dicionário;

- **Embedding**: Este módulo é responsável por transformar os tensores criados em Model Input em embeddings, podendo ser utilizados também embeddings pré-treinados para isso, como o caso deste trabalho;

- **Encoder e Decoder**: Neste módulo, encontra-se a rede neural em si que será responsável por identificar padrões nos dados de entrada. Os embeddings são recebidos pelo encoder (codificador), e a tarefa de codificação é feita pelas primeiras camadas da rede, podendo ser do tipo convolucional (CNN) ou do tipo Long Short Term Memory (LSTM). O decoder conta com uma rede do tipo MLP para decodificar os dados de saída;

- **Output Layer**: Aqui, a saída compreensível para o usuário do modelo é definida, para cada dado de entrada é identificado o rótulo mais provável de acordo com o recebido pelo decoder e com os rótulos definidos em Model Input. Neste módulo também é definida a função perda (loss) necessária para o módulo Trainer ajustar o treinamento do modelo;

- **Trainer**: Neste módulo, encontra-se as configurações gerais de treinamento do modelo, como número de epochs de treinamento e semente aleatória e também a função que ajusta os pesos utilizados na rede neural a cada etapa do treinamento;

- **Metric Reporter**: Este módulo é responsável por exibir as diferentes métricas que variam de acordo com o modelo implementado, como accuracy, recall, precision, root mean squared error e loss. Aqui o usuário pode escolher quais métricas deseja exibir;

- **Exporter**: Neste módulo, o modelo final é criado, possibilitando que o mesmo seja usado de forma independente sem a necessidade de novos treinamentos. No caso do PyText o modelo é exportado em formato Caffe2;

- **Caffe2**: Desenvolvido na University of California, Berkeley, Caffe2 é um framework de código aberto pensado para criar modelos de aprendizado profundo, seu funcionamento é baseado na criação de grafos estáticos;
- **Predictions**: Este componente da arquitetura PyText representa as predições geradas pelo modelo a partir da saída da rede neural no componente Output Layer;
- **Metrics**: Este componente representa as métricas do modelo obtidas com o treinamento.

Utilizou-se modelos pré-treinados BERT em português brasileiro disponibilizado pela NeuralMind, startup que atua na área de análise de texto e imagens usando inteligência artificial. O algoritmo da empresa foi treinado usando o BrWaC (Brazilian Web as Corpus). Esse corpus, por sua vez, foi criado a partir de uma metodologia equiparável aos grandes corpus referências internacionais; mais de 60 milhões de páginas foram rastreadas e, dessas, 3,5 milhões foram selecionadas, totalizando 120 mil sites diferentes e 2,7 bilhões de tokens (WAGNER *et al.*, 2018). O modelo pré-treinado foi obtido a partir da página do projeto no GitHub da empresa (NEURALMIND, 2021)

4 Resultados

4.1 Configuração e Calibragem do Modelo

Definido o método, foram feitos diversos testes a fim de se identificar a melhor configuração para a rede neural e os parâmetros mais adequados para o modelo. Os testes foram feitos utilizando a base de treino e a base de testes, reservando assim a base de validação para a avaliação do modelo final, a fim de se evitar viés nos resultados.

4.1.1 Divisão da Base de Dados

A base de dados foi dividida em 3 grupos a fim de facilitar a análise dos resultados e direcionar a discussão sobre os novos insights sobre a teoria que este método pode trazer. Os grupos foram definidos com base no grau de generalidade do conteúdo das notícias, ou seja, as notícias foram divididas em específicas do setor de transporte aéreo e de operação em aeroportos, notícias gerais sobre o setor aéreo e turismo e notícias sobre o cenário político e econômico do Brasil. Neste trabalho, os 3 grupos serão tratados como: aéreo, turismo e economia, respectivamente. As tags foram divididas entre os grupos da seguinte forma:

- **Aéreo**: aéreo, aeroporto, avião e linhas aéreas;
- **Turismo**: viagem, viajar e turismo;
- **Economia**: economia.

4.1.2 Título x Texto

Os primeiros testes foram feitos com o intuito de avaliar a qualidade da previsão quando comparado o uso de todo o texto da notícia com o uso de somente o título como entrada para a rede neural. Com essa verificação, buscou-se identificar se usar somente o resumo da notícia (aqui tido como o título) seria mais eficiente ou traria melhores resultados do

que usar o texto inteiro, que pode conter informações não tão relevantes para o objeto da previsão. Para isso, a rede foi treinada com as duas opções para cada grupo de tags. As imagens 4.1, 4.2 e 4.3 exibem os resultados de loss (no caso a soma dos quadrados residuais) para a base de treino e as imagens 4.4, 4.5 e 4.6 exibem os resultados para a base de teste. Em todos os gráficos é exibido no eixo "x"o número de epochs, que pode ser interpretado como o número de vezes que o algoritmo lê a base de dados, e no eixo "y"o loss correspondente, que representa o erro total cometido pelo algoritmo.

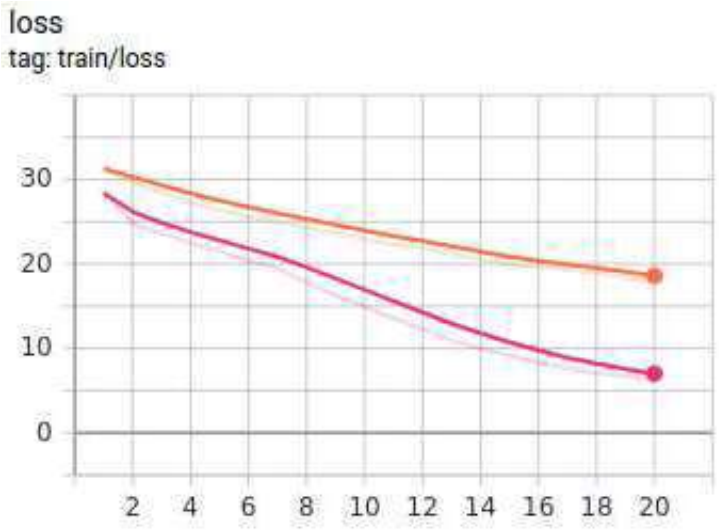


FIGURA 4.1 – Evolução do loss ao longo dos epochs para a base de treino do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo.

Fonte: Autor

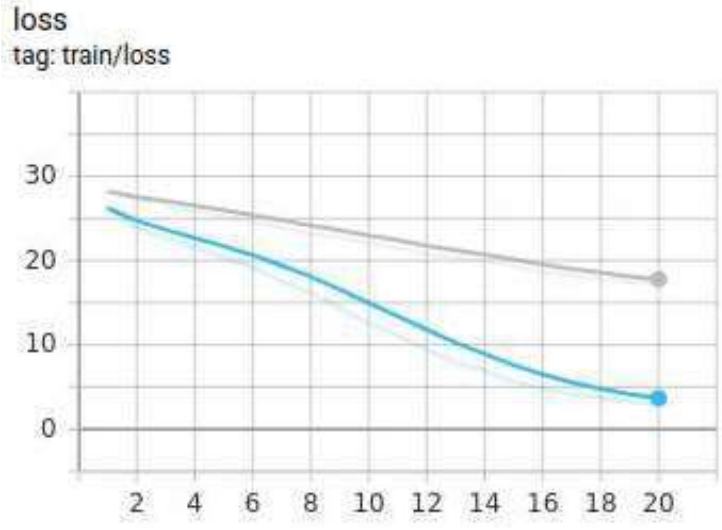


FIGURA 4.2 – Evolução do loss ao longo dos epochs para a base de treino do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo.

Fonte: Autor

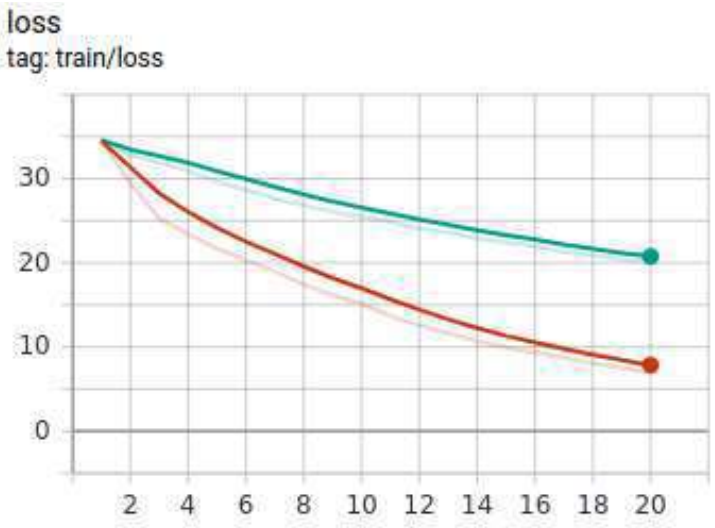


FIGURA 4.3 – Evolução do loss ao longo dos epochs para a base de treino do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo.

Fonte: Autor

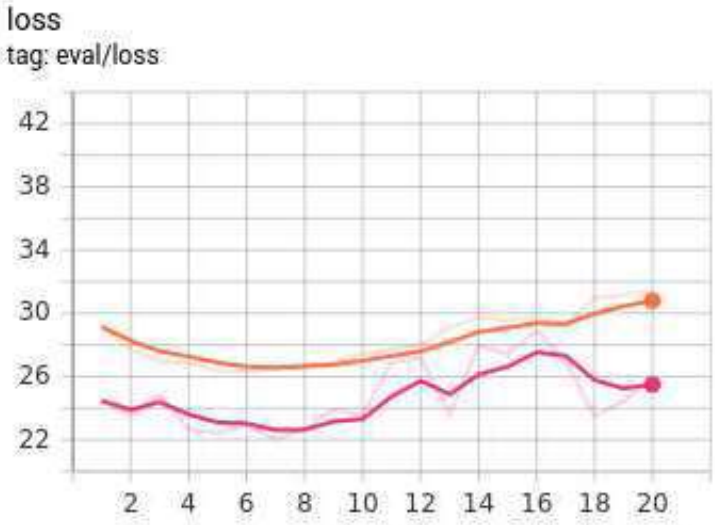


FIGURA 4.4 – Evolução do loss ao longo dos epochs para a base de testes do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo.

Fonte: Autor

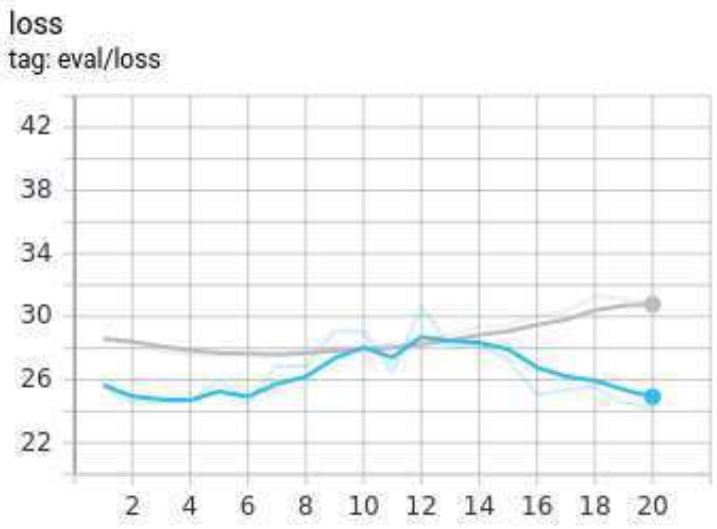


FIGURA 4.5 – Evolução do loss ao longo dos epochs para a base de testes do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo.

Fonte: Autor

FIGURA 4.6 – Evolução do loss ao longo dos epochs para a base de testes do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo.

Fonte: Autor

Como pode ser observado nos gráficos acima, em praticamente todos os epochs de todos os treinamentos para os 3 grupos da base de dados, o valor de loss quando usado todo o texto da notícia foi inferior ao loss de quando usado somente o título, evidenciando assim a melhora na qualidade do algoritmo com o uso de todo o texto.

As figuras 4.7, 4.8 e 4.9 exibem os resultados da correlação de Pearson para a base de treino e as imagens 4.10, 4.11 e 4.12 exibem os resultados para a base de teste. No eixo ”x”dos gráficos temos o número de epochs e no eixo ”y”o coeficiente de correlação de

Pearson correspondente.

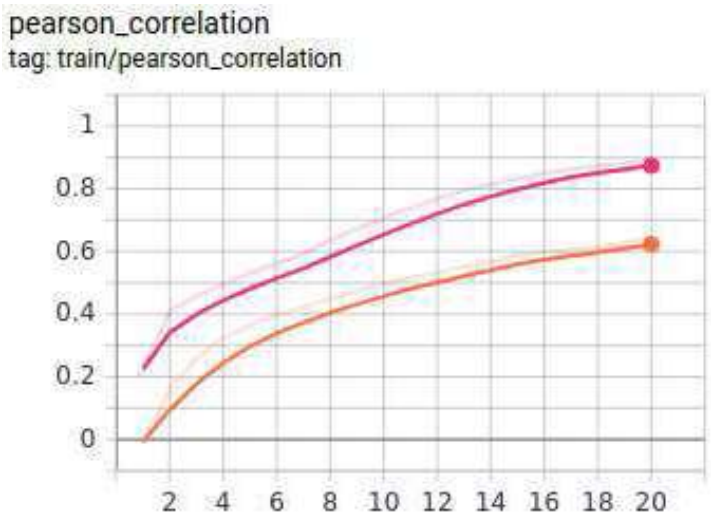


FIGURA 4.7 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo.

Fonte: Autor

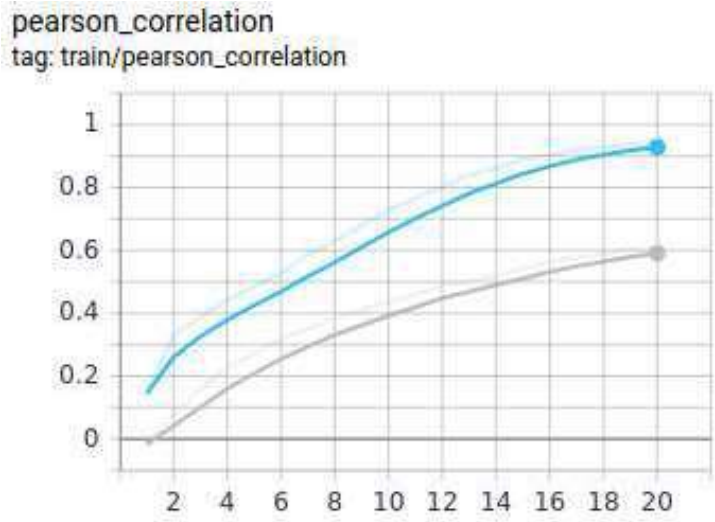


FIGURA 4.8 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo.

Fonte: Autor

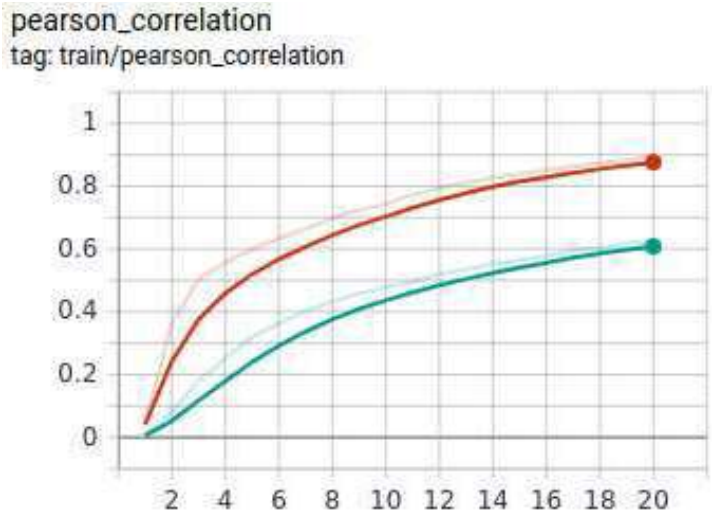


FIGURA 4.9 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo.

Fonte: Autor

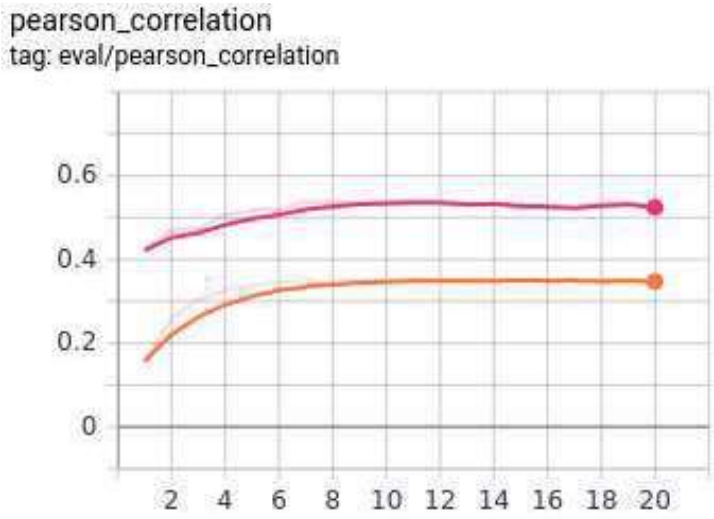


FIGURA 4.10 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes do grupo aéreo. Em laranja os valores correspondentes ao uso só do título e em rosa os valores correspondentes ao uso do texto todo.

Fonte: Autor

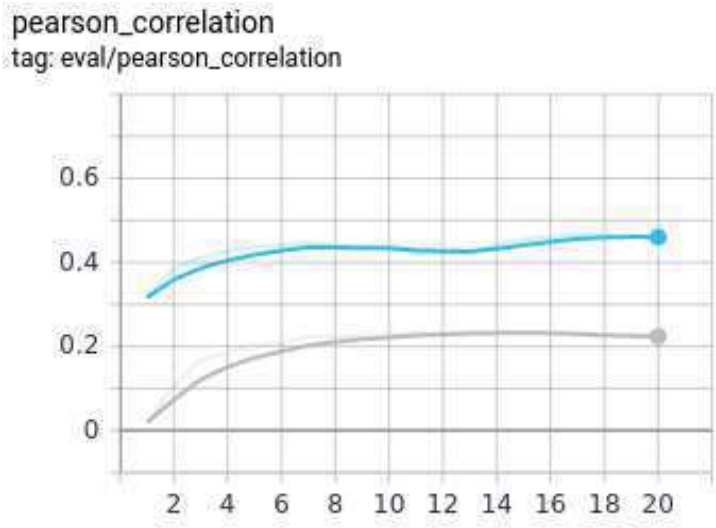


FIGURA 4.11 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes do grupo turismo. Em cinza os valores correspondentes ao uso só do título e em azul os valores correspondentes ao uso do texto todo.

Fonte: Autor

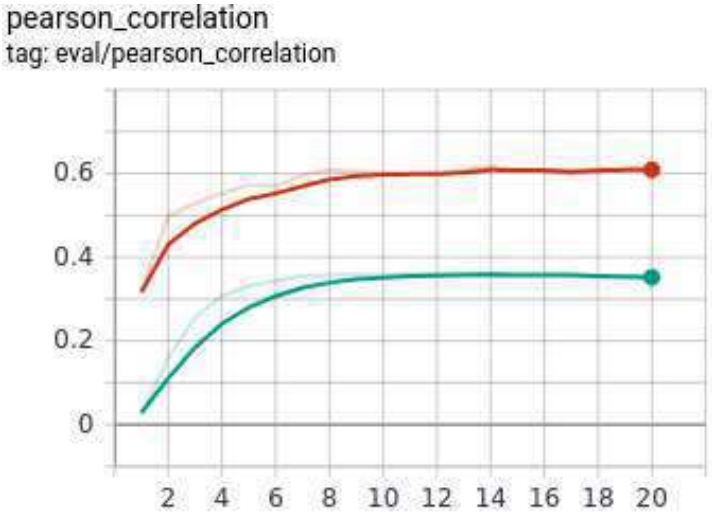


FIGURA 4.12 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes do grupo economia. Em verde os valores correspondentes ao uso só do título e em laranja os valores correspondentes ao uso do texto todo.

Fonte: Autor

Como pode ser observado acima, os valores dos coeficientes de correlação de Pearson confirmam o melhor desempenho do modelo que usa todo o texto como entrada do que o modelo que usa somente o título da notícia. Os valores indicam uma maior correlação entre o texto e o objeto alvo da previsão, ou seja, a variação da demanda por transporte aéreo. Assim, definiu-se para a continuidade do trabalho usar todo o texto da notícia, em detrimento de somente o título.

4.1.3 Número de Epochs

Epochs ou épocas de treinamento em uma rede neural, descreve o número de vezes que o algoritmo lerá todo o conjunto de dados durante o treinamento. Testes com o objetivo de se identificar o melhor número de epochs são necessários para ajustar o modelo, pois para diferentes tarefas, um mesmo número de epochs pode ser alto o suficiente para o algoritmo se ajustar demais aos dados de treinamento e perder generalização para dados gerais, como também pode ser insuficiente para o algoritmo identificar os padrões necessários para uma boa previsão. A figura 4.13 exibe resultados de loss para um treinamento com 20 epochs para os 3 grupos da base de dados de treino, enquanto a figura 4.13 exibe os mesmo resultados, mas para a base de testes.

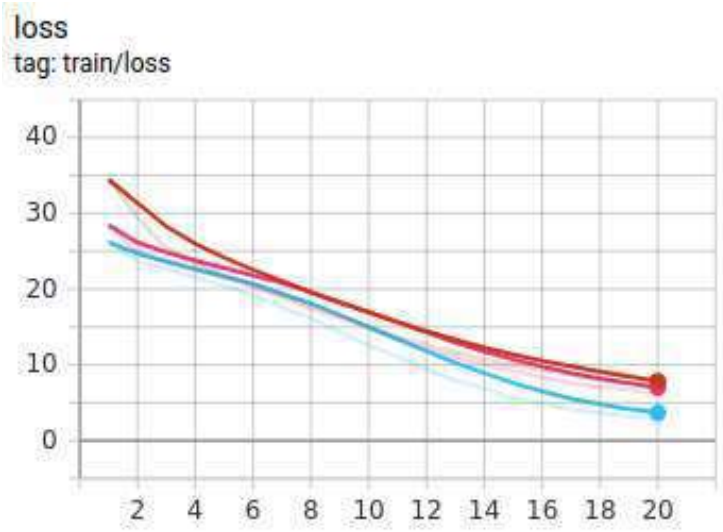


FIGURA 4.13 – Evolução do loss ao longo dos epochs para a base de treino para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo.

Fonte: Autor

FIGURA 4.14 – Evolução do loss ao longo dos epochs para a base de testes para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo.

Fonte: Autor

Como esperado, para a base de treino, o valor de loss apenas decresce com o aumento do número de epochs. Isso ocorre por que, quanto mais vezes o algoritmo lê a base de dados, menor é o erro cometido por ele naquela base. Já a base de testes apresenta uma variação de loss menos previsível. Isso ocorre devido aos fenômenos de underfitting e overfitting mencionados no referencial metodológico. Analisando o gráfico é possível identificar que cada grupo apresenta um valor ótimo de número de epochs, porém não é interessante usar exatamente esses números, pois a base que será usada para avaliar o modelo será a de validação, ainda não "vista" pelo algoritmo. Dado isso optou-se por identificar o valor com melhor desempenho médio nos 3 grupos; neste caso, número de epochs igual a 4 com loss médio igual a 24,39, porém ainda é necessário identificar a evolução da correlação entre as variáveis antes de se definir o número. As figuras 4.15 e 4.16 apresentam os valores do coeficiente de correlação de Pearson para as basese de treino e teste, respectivamente.

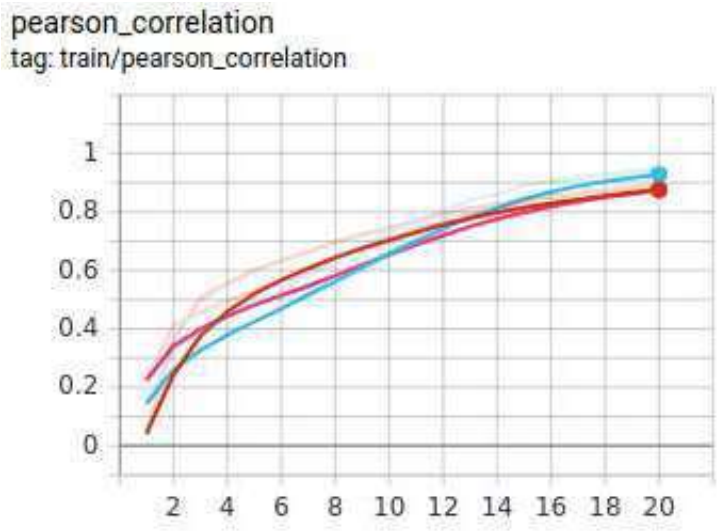


FIGURA 4.15 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de treino para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo.

Fonte: Autor

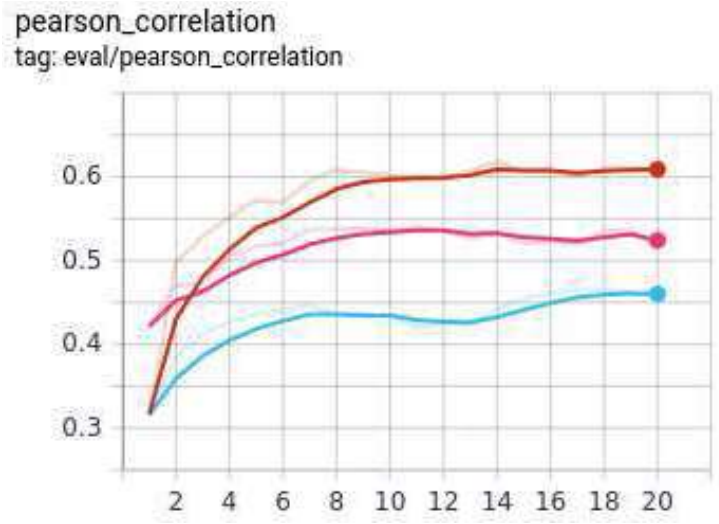


FIGURA 4.16 – Evolução do coeficiente de correlação de Pearson ao longo dos epochs para a base de testes para os 3 grupos da base de dados. Em laranja economia, em azul turismo e em rosa aéreo.

Fonte: Autor

Como é possível observar nos gráficos, é identificado uma correlação cada vez maior entre as variáveis explicativas e a variável explicada à medida que o número de epochs aumenta na base de treino, pelo mesmo motivo que o loss também só diminui na mesma base. Já para a base de testes é identificado uma certa estabilidade a partir de 9 epochs, não havendo ganho significativo na correlação identificada entre as variáveis. Dado isso, para a definição do número de epochs para o modelo final, decidiu-se identificar o menor valor médio de loss dentre todas as opções a partir de 9 epochs, chegando-se assim ao número de 19 epochs, com um loss médio de 24,83, bem próximo do valor ótimo de 24,39

identificado com 4 epochs.

4.2 Avaliação do Modelo

Definidos os parâmetros mais adequados para a realização da tarefa, tem-se o modelo final estabelecido. Para a avaliação do modelo utilizou-se a base de validação, separada no início dos treinamentos e ainda não "vista" pelo algoritmo. As figuras 4.17 e 4.18 apresentam os resultados de loss do modelo para os grupos "economia", "turismo" e "aéreo" para a etapa de treino e de validação final, respectivamente.

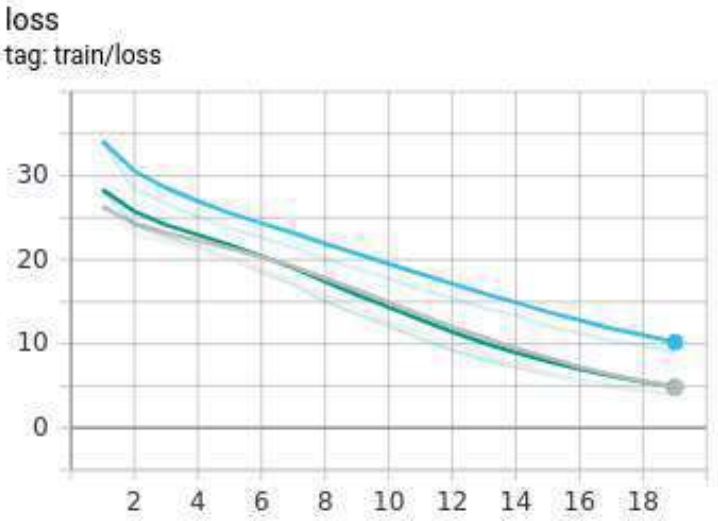


FIGURA 4.17 – Evolução do loss ao longo dos epochs para a etapa de treino do modelo final. Em azul os valores para o grupo "economia", em cinza "turismo"e em verde "aéreo"

Fonte: Autor

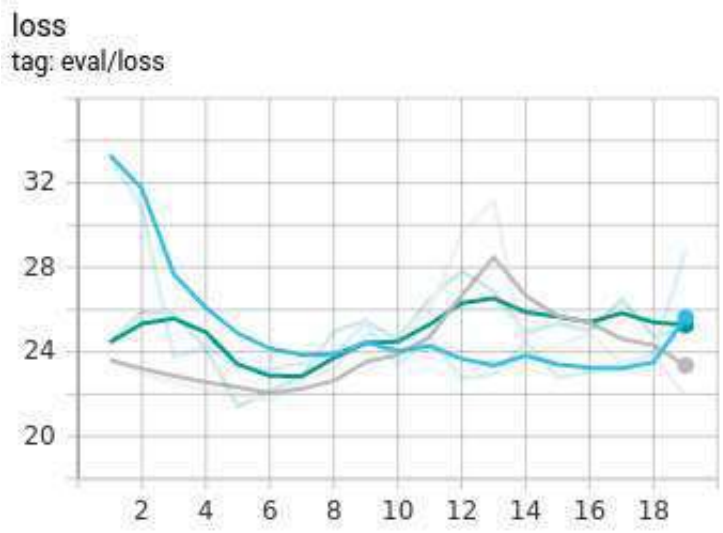


FIGURA 4.18 – Evolução do loss ao longo dos epochs para a base de validação utilizando o modelo final. Em azul os valores para o grupo "economia", em cinza "turismo"e em verde "aéreo"

Fonte: Autor

Percebe-se, a partir da figura 4.17, que durante a etapa de treino, as bases "turismo" e "aéreo" tiveram performance muito semelhantes e ambas apresentaram melhores resultados que a base "economia". Já para a base de validação, nota-se que a base "turismo" teve resultado superior às outras duas. A tabela 4.1 resume, para os 3 grupos da base de dados, a principal métrica de desempenho do modelo, o erro quadrático médio, aqui medido em pontos percentuais. A escolha de pontos percentuais como unidade de medida foi tomada para facilitar a compreensão dos resultados, já que o objeto de previsão, variação de demanda por transporte aéreo ano a ano, é dado em porcentagem.

TABELA 4.1 – Erro quadrático médio para os 3 grupos da base de dados utilizando o modelo final

	Erro quadrático médio (pp^2)	Raiz do erro quadrático médio (pp)
Economia	28,83	5,37
Turismo	21,92	4,68
Aéreo	25,08	5,01

Fonte: Autor

4.3 Discurssão

Para avaliar a qualidade do modelo, pode-se compará-lo a um previsor baseado em série histórica, ou seja, um modelo que prevê os valores futuros de demanda por transporte aéreo baseando-se na demanda passada. Neste caso, um modelo de previsão baseado em séries históricas, para o mesmo intervalo de tempo do modelo criado, teria um erro médio

de 5,81 pontos percentuais. Nota-se a partir da tabela 4.1 que todos os 3 grupos da base de dados performaram melhor que o modelo baseado na média dos valores históricos, o que indica uma correlação significativamente positiva entre as variáveis explicativas, no caso as notícias relacionadas a transporte aéreo, com a variável explicada, demanda por transporte aéreo. De acordo com os resultados obtidos na fase de testes, essa correlação positiva já era esperada, porém, um desempenho melhor que a média dos valores anteriores confirma o potencial de previsão que as notícias possuem.

Ao separar a base de dados nos 3 grupos escolhidos, buscava-se identificar qual esfera de influência tem mais impacto sobre a demanda por transporte aéreo no Brasil. A primeira esfera, "aéreo", que contém notícias específicas da operação em aeroportos e companhias aéreas teve um desempenho mediano, enquanto que a esfera mais geral sobre o setor aéreo, turismo e viagens, "turismo", teve o melhor desempenho dentre as 3. Por fim, o grupo "economia", contendo notícias do cenário econômico geral brasileiro, teve o pior desempenho. O desempenho mediano do grupo "aéreo" pode ser explicado pela mistura de notícias sobre a operação em aeroportos, como atrasos e cancelamentos que não têm um impacto significativo na demanda aérea futura, com notícias relacionadas às companhias aéreas, criação de novos aeroportos e novas rotas que, por sua vez, têm um impacto bastante significativo. O baixo desempenho do grupo "economia"pode ser explicado pela alta generalidade das notícias; por mais que o cenário econômico nacional tenha alto impacto na demanda por transporte aéreo, quando se pega notícias gerais com a tag "economia" muitas notícias sem relação com transporte aéreo são incluídas, o que acaba comprometendo o foco do algoritmo. Por fim, o melhor desempenho do grupo "turismo"pode indicar que essa seja a esfera com informações mais relevantes para o objeto de previsão, ou seja, pode indicar que identificando tendências de turismo e viagens pode-se prever com precisão a demanda por transporte aéreo brasileiro, indicando assim um caminho a ser seguido em pesquisas futuras.

5 Conclusão e Pesquisas Futuras

5.1 Conclusão e Pesquisas Futuras

Este trabalho apresentou um novo método para prever a demanda por transporte aéreo, baseado em redes profundas e processamento de linguagem natural. Para tal, as variáveis utilizadas foram as notícias de jornal relacionadas a transporte aéreo no Brasil. Conforme a estrutura proposta na seção 1.5, este estudo descreve os principais trabalhos da literatura que tratam de previsão de demanda por transporte aéreo utilizando diversos métodos, cumprindo assim com um dos objetivos específicos de revisar métodos tradicionais da literatura. O trabalho segue com a formação da base de dados utilizando Web scraping, criando assim uma base com 657719 notícias de jornais relacionadas a transporte aéreo, cumprindo assim com o segundo objetivo de criar e disponibilizar na literatura uma base de dados robusta para a literatura e futuros trabalhos. Em sequência, o modelo de aprendizado de máquina foi criado com as técnicas mais avançadas disponíveis em processamento de linguagem natural. Todos os códigos estão comentados e encontram-se no apêndice deste trabalho e o projeto está aberto e disponível na plataforma GitHub, cumprindo-se assim com o terceiro objetivo de disponibilizar um modelo replicável e escalável para a literatura.

Em resumo, pelos resultados apresentados na seção 4.2, pode-se afirmar que o presente trabalho cumpriu com o objetivo principal de criar uma ferramenta capaz de prever demanda por transporte aéreo com exatidão, a partir de notícias de jornal relacionadas a transporte aéreo. Entretanto, ao finalizar este estudo, identificam-se oportunidades de melhoria para trabalhos futuros que tratem do mesmo tema.

O primeiro ponto a ser estressado no estudo em busca de se obter diferentes resultados é a escolha de palavras-chave (tags) para a criação da base de dados. Neste estudo, a escolha das palavras foi feita objetivando-se extrair as melhoras notícias relacionadas a transporte aéreo, mas de forma subjetiva. O mesmo estudo pode ser realizado com palavras-chave diferentes chegando a diferentes resultados. O ideal seria determinar um método objetivo para a escolha das tags.

Neste trabalho todas as notícias em formato de vídeo, sem texto, foram excluídas

da base de dados. Em pesquisas futuras essas notícias podem ser incluídas por meio de transcrição do áudio dos vídeos.

Este trabalho concentrou-se no estudo da demanda agregada de todo o Brasil, todas as companhias aéreas e aeroportos. Em trabalhos futuros pode-se segmentar os diversos tipos de demanda a fim de prevê-las individualmente, possivelmente com resultados mais precisos do que prevendo a demanda geral de uma só vez. Também é possível segmentar por tipo de passagem, executiva ou econômica, uma vez que, de acordo com o referencial teórico, é possível que as notícias do grupo "turismo" tenham mais impacto na demanda por passagens da classe econômica do que da classe executiva.

Por fim, futuros trabalhos podem associar os dados usados neste estudo com outros dados relevantes para a demanda por transporte aéreo, como tamanho populacional próximo ao aeroporto estudado, ou na origem e destino da rota estudada, e até mesmo variáveis macroeconômicas clássicas como Yield e PIB.

Referências

ALAMMAR, J. **The Illustrated Transformer**. 2018. Disponível em: https://jalammar.github.io/illustrated-transformer/. Acesso em: 22 junho 2021.

ANAC, A. N. de A. C. **Demanda e Oferta do Transporte Aéreo**. 2021. Disponível em: https://app.powerbi.com/view?r=eyJrIjoiNGRmNTNjZTktOGU2OC00YTc4LWJkMjQtNWZjZWRhZjRiNTY3IiwidCI6ImI1NzQ4ZjZlLWI0YTQ Acesso em: 13 setembro 2021.

ARNOLD, B. C. Pareto distribution. **Wiley StatsRef: Statistics Reference Online**, Wiley Online Library, p. 1–10, 2014.

BATES, J. History of demand modelling. *In*: **Handbook of transport modelling**. [*S.l.*]: Emerald Group Publishing Limited, 2007.

BENDINELLI, W. E.; OLIVEIRA, A. V. Modelagem econométrica da demanda em aeroportos privatizados: estudo de caso do aeroporto internacional de confins, belo horizonte. **Journal of Transport Literature**, SciELO Brasil, v. 9, p. 20–24, 2015.

BRONS, M.; PELS, E.; NIJKAMP, P.; RIETVELD, P. **Price elasticities of demand for passenger air travel**. [*S.l.*], 2001.

CANTO, L. G. **Análise de Notícias do Mercado Financeiro Utilizando Processamento de Linguagem Natural e Aprendizado de Máquina para Decisões de Swing Trade**. Trabalho de Conclusão de Curso — UFRJ/ Escola Politécnica, Rio de Janeiro, 2020.

CARNEIRO, A. L. C. **Redes Neurais Convolucionais para processamento de linguagem natural**. 2020. Disponível em: https://medium.com/data-hackers/redes-neurais-convolucionais-para-processamento-de-linguagem-natural-935488d6901b. Acesso em: 22 junho 2021.

CONDÉ, M. Estudo e previsão de demanda aeroportuária para a cidade do rio de janeiro. **Journal of Transport Literature**, v. 5, n. 1, p. 161–183, 2011.

COSTA, J.; SANTOS, L.; YAMASHITA, Y. Vocação turística das cidades brasileiras: análise de modelos de previsão de demanda do transporte aéreo. **SIMPÓSIO DE TRANSPORTE AÉREO**, v. 7, 2008.

DAS, T. K.; KUMAR, P. M. Big data analytics: A framework for unstructured data analysis. **International Journal of Engineering Science & Technology**, Citeseer, v. 5, n. 1, p. 153, 2013.

DESAGULIER, G. **Word embeddings: the (very) basics**. 2018. Disponível em: `https://corpling.hypotheses.org/495`. Acesso em: 22 junho 2021.

DINIZ, R. R. Dimensionamento de ampliação do aeroporto de marabá com base em estudo de previsão de demanda aeroportuária. **Journal of Transport Literature**, SciELO Brasil, v. 7, p. 147–162, 2013.

DUAN, W.; GU, B.; WHINSTON, A. B. Do online reviews matter? — an empirical investigation of panel data. **Decision Support Systems**, v. 45, p. 1007–1016, 2008.

FALCÃO, V. A. Demanda aeroportuária de manaus e sua influência para o setor de turismo da região. **Journal of Transport Literature**, SciELO Brasil, v. 7, p. 127–146, 2013.

FELIPELEO1995. **felipeleo1995/Previsao-de-demanda-por-transporte-aereo-usando-deep-learning-e-processamento-de-linguagem-natural**. 2021. Available at: https://github.com/felipeleo1995/Previsao-de-demanda-por-transporte-aereo-usando-deep-learning-e-processamento-de-linguagem-natural.

FRAZÃO, J. A. F.; OLIVEIRA, A. V. Distribuição de renda e demanda por transporte aéreo: uma especificação de modelo econométrico para o mercado doméstico brasileiro. **TRANSPORTES**, v. 28, n. 3, p. 1–13, 2020.

GHOSE, A.; IPEIROTIS, P. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. **Knowledge and Data Engineering. IEEE Transactions on Knowledge and Data Engineering**, v. 23, p. 1498–1512, 2011.

GODOY, K. G. d. **A previsão do tráfego aéreo de passageiros**. Thesis (Doutorado), 1997.

GROSCHE, T.; ROTHLAUF, F.; HEINZL, A. Gravity models for airline passenger volume estimation. **Journal of Air Transport Management**, Elsevier, v. 13, n. 4, p. 175–183, 2007.

JORGE-CALDERÓN, J. A demand model for scheduled airline services on international european routes. **Journal of Air Transport Management**, Elsevier, v. 3, n. 1, p. 23–35, 1997.

JOSHI, M.; DAS, D.; GIMPEL, K.; SMITH, N. A. Movie reviews and revenues: An experiment in text regression. *In*: **Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics**. **Proceedings** [...]. [*S.l.: s.n.*], 2010. p. 293–296.

KATIUSKA, A.; DIAZ, R.; LIMA, A.; MARTINS, A.; FERNANDO, S.; DA, H.; COSTA, S.; LUIZ, J.; PAGNOSSIM, M.; PERES, S. **Uma análise comparativa das ferramentas de pré-processamento de dados textuais: NLTK, PreTexT e R**. 01 2018.

KO, C.-H. Exploring big data applied in the hotel guest experience. **Open Access Library Journal**, v. 5, n. e4877, 2018.

KUMAR, A. **Overfitting Underfitting Concepts Interview Questions**. 2020. Disponível em: `https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/`. Acesso em: 23 julho 2021.

LIMA, N. A. Análise econométrica aplicada ao planejamento de infraestrutura aeroportuária–estudo de caso do aeroporto de vitória. **Revista Tecnologia**, v. 34, n. 1/2, p. 104–112, 2013.

LOVATTI, H. Z. **TRANSPORTE AÉREO REGIONAL: ESTUDO DE DEMANDA DE PASSAGEIROS ENTRE LOCALIDADES DO SUL E SUDESTE**. Thesis (Doutorado) — Universidade Federal do Rio de Janeiro, 2018.

LUCIAN, B. **Overfitting: o que é e como evitar**. 2020. Disponível em: `https://www.dadosaleatorios.com.br/post/overfitting/`. Acesso em: 23 julho 2021.

MALDONADO, J. **Strategic planning–an approach to improving airport planning under uncertainty**. Thesis (Doutorado) — Massachusetts Institute of Technology, 1990.

NEUFVILLE, R. de; BARBER, J. Deregulation induced volatility of airport traffic. **Transportation Planning and Technology**, Taylor & Francis, v. 16, n. 2, p. 117–128, 1991.

NEURALMIND. **GitHub - neuralmind-ai/portuguese-bert: Portuguese pre-trained BERT models**. Feb 2021. Available at: https://github.com/neuralmind-ai/portuguese-bert.

OGNJANOVSKI, G. **Everything you need to know about Neural Networks and Backpropagation**. 2019. Disponível em: `https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-lear` Acesso em: 22 junho 2021.

OLMEDO, E. Comparison of near neighbour and neural network in travel forecasting. **Journal of Forecasting**, Wiley Online Library, v. 35, n. 3, p. 217–223, 2016.

Oper Data. **Exemplo de sazonalidade**. [*S.l.*], 2019. Available at: https://operdata.com.br/blog/caracteristicas-das-series-temporais/. Accessed on: 20 jun. 2021.

PAMPLONA, D. A.; OLIVEIRA, A. V. M. de. Influence of airport demand in a shared airport between military and civil personnel: The case of salvador international airport. **International Journal of Science and Engineering Investigations**, v. 4, n. 37, p. 12–18, 2015.

PAUL, S.; PURKHYASTHA, B. shyam. An nlp tool for decoding the atc phraseology from english to bengali. *In*: IEEE. **2020 International Conference on Smart Electronics and Communication (ICOSEC)**. **Proceedings** [...]. [*S.l.: s.n.*], 2020. p. 264–270.

PLAKANDARAS, V.; PAPADIMITRIOU, T.; GOGAS, P. Forecasting transportation demand for the us market. **Transportation Research Part A: Policy and Practice**, Elsevier, v. 126, p. 195–214, 2019.

PRATEEK. **Statistics is Freaking Hard: WTF is Activation function**. 2017. Disponível em: `https://towardsdatascience.com/statistics-is-freaking-hard-wtf-is-activation-function-df8342cdf292`. Acesso em: 22 junho 2021.

PYTEXT Documentation. 2021. Available at: https://pytext.readthedocs.io/en/master/overview.html.

RIBEIRO, L. **O que é o modelo BERT?** 2020. Disponível em: `https://aprendizdofuturo.com.br/2020/07/02/o-que-e-o-modelo-bert/`. Acesso em: 22 junho 2021.

RINALDO, G.; NAGANO, H. Predicting voting outcomes in the brazilian congress. **Itaú-Unibanco**, 2019.

ROCHA, G. C. Ensaios sobre a demanda do transporte aéreo regional. **Journal of Transport Literature**, v. 4, n. 1, 2010.

SABO, I. C.; PONT, T. R. D.; WILTON, P. E. V.; ROVER, A. J.; HÜBNER, J. F. Clustering of brazilian legal judgments about failures in air transport service: an evaluation of different approaches. **Artificial Intelligence and Law**, Springer, p. 1–37, 2021.

SARMENTO. **Notícias de Jornal relacionadas a Transporte Aéreo**. 2021. Available at: https://www.kaggle.com/felipesarmento/notcias-de-jornal-relacionadas-a-transporte-areo?select=turismo.csv.

SCHUMAKER, R.-P.; CHEN, H. Textual analysis of stock market prediction using breaking financial news: The azf in text system. **ACM Transactions on Information Systems**, v. 27, p. 1–19, 2009.

SHANE. **Get Busy with Word Embeddings – An Introduction**. 2018. Disponível em: `https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/`. Acesso em: 22 junho 2021.

SIEGEL, E. **Predictive analytics: The power to predict who will click, buy, lie, or die**. [*S.l.*]: John Wiley & Sons, 2013.

TETLOCK, P.; SAAR-TSECHANSKY, M.; MACSKASSY, S. More than words: Quantifying language to measure firms' fundamentals. **The Journal of Finance**, v. 63, p. 1437–1467, 2008.

TULECHKI, N. **Natural language processing of incident and accident reports: application to risk management in civil aviation**. Thesis (Doutorado) — Université Toulouse le Mirail-Toulouse II, 2015.

VARELLA, R. R. Análise da capacidade de projeção de demanda: Estudo de caso do aeroporto de recife. 2016.

WAGNER, J.; WILKENS, R.; IDIART, M.; VILLAVICENCIO, A. The brwac corpus: A new open resource for brazilian portuguese. *In*: . **Proceedings** [...]. [*S.l.: s.n.*], 2018.

WEI, W.; HANSEN, M. An aggregate demand model for air passenger traffic in the hub-and-spoke network. **Transportation Research Part A: Policy and Practice**, Elsevier, v. 40, n. 10, p. 841–851, 2006.

Apêndice A - Extração dos links

extrai_links

```
from bs4 import BeautifulSoup
from selenium import webdriver
# from selenium.webdriver.firefox.options import Options
from selenium.webdriver.chrome.options import Options
from time import sleep
import pandas as pd
from datetime import datetime
from datetime import date
from selenium.webdriver.common.keys import Keys


# Rola a p gina para baixo 4 vezes para mostrar mais not cias
def rolar_pag_baixo():
    for c in range(0, 4):
        sleep(timer)
        navegador.execute_script("window.scrollTo(0, document.body.
            scrollHeight);")


# Clica em "Aplicar"
def clicar_aplicar():
    botao_aplicar = navegador.find_element_by_css_selector(
        'div > div > div > div > div > div > div > div > button')
    botao_aplicar.click()
    sleep(timer)


# Clica no bot o de "Ver Mais" at  o limite para mostrar todas as
    not cias
def clilcar_vermais():
    pagina = BeautifulSoup(navegador.page_source, 'html.parser')
    cards = pagina.findAll('div', attrs={'class': 'widget--info__text-
        container'})  # Pega todos os cards da p gina atual
    while (len(cards) < 600):  # 600    o limite de not cias por p gina
        try:
```

```
33              botao_ver_mais = navegador.find_element_by_css_selector('
                    section > div > div > div > a')
            except:
                break
            botao_ver_mais.click()
37          sleep(timer)
            pagina = BeautifulSoup(navegador.page_source, 'html.parser')
39          cards = pagina.findAll('div', attrs={'class': 'widget--info__text-
                container'})


41
# Extrai as informa  es das not cias da tela e armazena em csv
43  def extrai_infos():
        pagina = BeautifulSoup(navegador.page_source, 'html.parser')
45      cards = pagina.findAll('div',
                               attrs={'class': 'widget--info__text-container'})
                                    # Pega todos os cards da p gina atual
47      for card in cards:
            try:
49              titulo = card.find('div', attrs={'class': 'widget--info__title
                    product-color'}).text
            except:
51              titulo = ''
            data = card.find('div', attrs={'class': 'widget--info__meta'})
53          if data.text.split()[0] == 'h  ':
                data = date.today()
55          else:
                data = datetime.strptime(data.text.split()[0], '%d/%m/%Y').date
                    ()
57          link = 'https:' + card.find('a')['href']
            lista_noticias.append([data, titulo, link])
59


61  # Seleciona o primeiro dia do m s
    def clicar_dia1():
63      calendario = navegador.find_elements_by_tag_name('tbody')[1]  #
            Identifica o calend rio
        primeira_semana = calendario.find_elements_by_tag_name('tr')[0]  #
            Identifica a primeira semana
65      dias = primeira_semana.find_elements_by_tag_name(
            'td')  # Identifica a lista com todos os dias da primeira semana do
                m s
67      for dia in dias:  # Identifica o primeiro dia do m s
            if (dia.text == '1'):
69              sleep(1)
                dia.click()
71      sleep(timer)
```

```
73
# Seleciona o dia 8
75 def clicar_dia8():
    calendario = navegador.find_elements_by_tag_name('tbody')[1]
77     segunda_semana = calendario.find_elements_by_tag_name('tr')[1]
    dias = segunda_semana.find_elements_by_tag_name('td')
79     for dia in dias:
        if (dia.text == '8'):
81             sleep(1)
            dia.click()
83     sleep(timer)


85
# Seleciona o dia 15
87 def clicar_dia15():
    calendario = navegador.find_elements_by_tag_name('tbody')[1]
89     terceira_semana = calendario.find_elements_by_tag_name('tr')[2]
    dias = terceira_semana.find_elements_by_tag_name('td')
91     for dia in dias:
        if (dia.text == '15'):
93             sleep(1)
            dia.click()
95     sleep(timer)


97
# Seleciona o dia 15
99 def clicar_dia22():
    calendario = navegador.find_elements_by_tag_name('tbody')[1]
101     quarta_semana = calendario.find_elements_by_tag_name('tr')[3]
    dias = quarta_semana.find_elements_by_tag_name('td')
103     for dia in dias:
        if (dia.text == '22'):
105             sleep(1)
            dia.click()
107     sleep(timer)


109
# Seleciona o  ltimo  dia do m s
111 def clicar_ultimo_dia():
    calendario = navegador.find_elements_by_tag_name('tbody')[1]
113     ultima_semana = calendario.find_elements_by_tag_name('tr')[-1]
    dias = ultima_semana.find_elements_by_tag_name('td')
115     for dia in dias:
        if (dia.text != ''):
117             ultimo_dia = dia
    ultimo_dia.click()
```

```
119     sleep(timer)


121
# Pesquisa as palavras chave no campo de busca de not cias
123 def pesquisa(palavra):
        navegador.execute_script("window.scrollTo(0, -document.body.
            scrollHeight);")
125     try:
            campo = navegador.find_element_by_tag_name('fieldset')
127         input_place = campo.find_element_by_tag_name('input')
        except:
129         input_place = navegador.find_element_by_tag_name('input')
            caracteres_pesquisa = len(input_place.get_attribute('value'))
131         input_place.send_keys(caracteres_pesquisa * Keys.BACKSPACE)
        input_place.send_keys(palavra)
133     input_place.submit()
        sleep(timer)
135
# Tags que ser o pesquisadas
137 lista_de_palavras = ['economia', 'avi o', 'a reo', 'viagem', 'viajar', '
    turismo', 'linhas a reas', 'aeroporto']


139
# Intervalo de tempo para garantir que os elementos foram carregados na
    p gina antes de tentar acessar
141 timer = 2

143 # Inibe o Selenium de abrir o navegador real para mostrar o que t
    acontecendo
    options = Options()
145 options.add_argument('--headless')
    options.add_argument(('window-size=1920,1080'))
147
# Navegador do Selenium pra acessar a p gina inicial
149 navegador = webdriver.Chrome(options=options)
    navegador.get('https://g1.globo.com/')
151
# Clica no bot o do banner de cockies (necess rio para poder prosseguir)
153 botao_cokies = navegador.find_element_by_id('cookie-banner-lgpd')
    botao_cokies = botao_cokies.find_element_by_class_name('cookie-banner-
    lgpd_accept-button')
155 botao_cokies.click()


157
for palavra in lista_de_palavras:
159     for mes in range(259, 18, -1):
            # Lista onde ser o armazenadas as not cias e seus links
```

```
161         lista_noticias = []

163         # Pesquisa as palavras chave no  campo de busca de not cias e
                filtra pelas mais relevantes
            pesquisa(palavra)
165         barra_filtro = navegador.find_element_by_id('search-filter-
                component')
            barra_filtro = barra_filtro.find_element_by_css_selector('div > div
                > div> div')
167         botao_ordenar = barra_filtro.find_element_by_class_name('
                filters__container')
            botao_ordenar = botao_ordenar.find_elements_by_css_selector('div >
                a')[1]
169         botao_ordenar.click()
            sleep(timer)
171         botao_relevancia = barra_filtro.find_element_by_class_name('
                filters__container')
            botao_relevancia = botao_relevancia.find_elements_by_css_selector('
                div > ul > li')[8]
173         botao_relevancia.click()
            sleep(timer)
175
            # Clica no bot o de "filtrar por data"
177         barra_filtro = barra_filtro.find_element_by_class_name('
                filters__advanced-date-filter')
            botao_filtro = barra_filtro.find_element_by_class_name('
                filters__dropdown__link')
179         botao_filtro.click()
            sleep(timer)
181
            # Clica em "per odo personalizado"
183         botao_periodo = barra_filtro.find_element_by_class_name('
                filters__dropdown__list__right')
            botao_periodo = botao_periodo.find_element_by_class_name('
                filters__dropdown__list__range-date')
185         botao_periodo.click()
            sleep(timer)
187
            # Identifica o bot o de voltar 1 m s
189         botao_voltar = navegador.find_element_by_id('search-filter-
                component')
            botao_voltar = botao_voltar.find_element_by_class_name('range-date-
                filter-modal__container')
191         botao_voltar = botao_voltar.find_element_by_class_name('range-date-
                filters')
            botao_voltar = botao_voltar.find_element_by_css_selector('div > div
                > div > div > div > svg')
```

```
193
        # Clica no bot o "voltar" a quantidade de vezes necess ria para
            chegar na data correta
195     contador = 0
        while (contador < mes):
197         contador = contador + 1
            botao_voltar.click()
199         sleep(.3)

201     # Extrai os dados da primeira semana
        clicar_dia1()  # Seleciona o primeiro dia do m s
203     clicar_dia8()  # Seleciona o dia 8
        clicar_aplicar()  # Clica em "Aplicar"
205     rolar_pag_baixo()  # Rola a p gina para baixo para mostrar mais
            not cias
        clilcar_vermais()  # Clica no bot o de "Ver Mais" at  o limite
            para mostrar todas as not cias
207     extrai_infos()  # Extrai as informa  es das not cias da tela e
            armazena em csv

209     # Extrai os dados da segunda semana
        botao_filtro.click()
211     sleep(timer)
        botao_periodo.click()
213     sleep(timer)
        clicar_dia8()
215     clicar_dia15()
        clicar_aplicar()
217     rolar_pag_baixo()
        clilcar_vermais()
219     extrai_infos()

221     # Extrai os dados da terceira semana
        botao_filtro.click()
223     sleep(timer)
        botao_periodo.click()
225     sleep(timer)
        clicar_dia15()
227     clicar_dia22()
        clicar_aplicar()
229     rolar_pag_baixo()
        clilcar_vermais()
231     extrai_infos()

233     # Extrai os dados do resto do m s
        botao_filtro.click()
235     sleep(timer)
```

```
            botao_periodo.click()
237         sleep(timer)
            clicar_dia22()
239         clicar_ultimo_dia()
            clicar_aplicar()
241         rolar_pag_baixo()
            clilcar_vermais()
243         extrai_infos()

245         # Armazena as informa  es em csv
            noticias_df = pd.DataFrame(lista_noticias, columns=['Data', 'Titulo
                ', 'Link'])
247         noticias_df.to_csv('saida/noticias_' + palavra + '_' + str(mes) + '
                .csv', index=False)
```

Apêndice B - Limpeza dos links

limpa_links

```
import pandas as pd


def getCleanLink(link):
    # Retira da base as not cias sem texto (apenas v deo)
    if 'video' in link or 'globoplay' in link:
        linklimpo = None
    else:
        linklimpo = link
        print('Deu bom para o link:')
        print(link)
    return linklimpo

lista_de_palavras = ['economia', 'avi o', 'a reo', 'viagem', 'viajar', '
    turismo', 'linhas a reas', 'aeroporto']

for palavra in lista_de_palavras:
    links_final = pd.DataFrame()
    for mes in range(259, 18, -1):
        try:
            links = pd.read_csv('saida/noticias_' + palavra + '_' + str(mes
                ) + '.csv')
            links['LinkLimpo'] = links['Link'].apply(lambda x: getCleanLink
                (x))
            links = links[links.LinkLimpo.isnull() == False]
            links_final = links_final.append(links)
        except:
            print('Erro ao identificar o arquivo')
    try:
        links_final = links_final.sort_values('Data')
        links_final.drop(columns=['Link'], inplace=True)
        links_final.to_excel('limpos/Links_limpos_' + palavra + '.xlsx',
            index=False)
    except:
        print('Erro ao criar o arquivo')
```

Apêndice C - Separa os links por ano

separa_ano

```
import pandas as pd

def getAno(data):
    return data.year


tags = ['economia', 'avi o', 'a reo', 'viagem', 'viajar', 'turismo', '
    linhas a reas', 'aeroporto']

for tag in tags:
    df = pd.read_excel('limpos/Links_limpos_' + tag + '.xlsx')
    df['Data'] = pd.to_datetime(df['Data'].astype(str), format='%Y-%m-%d')
    df['Ano'] = df.apply(lambda row: getAno(row['Data']), axis = 1)
    for ano in range(2006, 2019):
        df_novo = df[df['Ano'] == ano]
        df_novo.to_excel('SaidaSeparadaAno/' + tag + '_' + str(ano) + '.
            xlsx', index=False, engine='xlsxwriter')
        print(df_novo)
```

Apêndice D - Captura o texto dos links

captura_texto

```
import requests
from bs4 import BeautifulSoup
import pandas as pd


def extraiTexto(link):
    response = requests.get(link)
    content = response.content
    site = BeautifulSoup(content, 'html.parser')
    link_real = site.find('script')
    link_real = str(link_real).split('"')
    link_real = link_real[1]

    response = requests.get(link_real)
    content = response.content
    site = BeautifulSoup(content, 'html.parser')
    try:
        noticia = site.find('div', attrs={'class': 'materia-conteudo entry-
            content clearfix'})
        return noticia.text
    except:
        pass
    try:
        noticia = site.find('div', attrs={'class': 'entry'})
        return noticia.text
    except:
        pass
    try:
        noticia = site.find('article')
        return noticia.text
    except:
        pass
    print('Erro')
    return ''
```

```
35 tags = ['economia', 'avi o', 'a reo', 'viagem', 'viajar', 'turismo', '
       linhas a reas', 'aeroporto']

37 for tag in tags:
       for ano in range(2010, 2013):
39         df = pd.read_excel('SaidaSeparadaAno/' + tag + '_' + str(ano) + '.
               xlsx')
           df['texto'] = df['LinkLimpo'].apply(lambda x: extraiTexto(x))
41         df.to_excel('saidaTexto/' + tag + str(ano) + '.xlsx', index=False,
               engine='xlsxwriter')
```

Apêndice E - Junta os textos no dataframe final

captura_texto

```
import pandas as pd


def Join(tags):
    df_final = pd.DataFrame()
    for tag in tags:
        df = pd.read_excel('saidaFinal/' + tag + '.xlsx')
        print(df)
        df_final = pd.concat([df_final, df])
    df_final = df_final.drop_duplicates()
    return df_final


tags = ['economia', 'avi o', 'a reo', 'viagem', 'viajar', 'turismo', '
    linhas aereas', 'aeroporto']

for tag in tags:
    df_final = pd.DataFrame()
    for ano in range(2007, 2019):
        df = pd.read_excel('saidaTexto/' + tag + str(ano) + '.xlsx')
        df.drop(['Data', 'LinkLimpo'], inplace=True, axis=1)
        df_final = pd.concat([df_final, df])
    df_final.to_excel('saidaFinal/' + tag + '.xlsx', index=False, engine='
        xlsxwriter')

tags_turismo = ['viagem', 'viajar', 'turismo']
tags_aereo = ['avi o', 'a reo', 'linhas aereas', 'aeroporto']
tags_economia = ['economia']

df_macro = Join(tags_macro)
df_macro.to_csv('Titulo/saidaJoin/turismo.csv', index=False)
df_micro = Join(tags_micro)
df_micro.to_csv('Titulo/saidaJoin/aereo.csv', index=False)
```

```
32 df_economia = Join(tags_economia)
   df_economia.to_csv('Titulo/saidaJoin/economia.csv', index=False)
```

Apêndice F - Prepara e pré-processa os textos

captura_texto

```
import pandas as pd
import re
import nltk
from unicodedata import normalize
import csv

nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer

STEMMER = SnowballStemmer('portuguese')
STOPWORDS = stopwords.words('portuguese')
REGEXP_REMOVE_SPECIAL = re.compile('[^a-zA-Z0-9 ]+')
TRAIN_PERCENTAGE_SIZE = 60 / 100
TEST_PERCENTAGE_SIZE = 20 / 100


# Limpa o texto, passa tudo para minscula, remove as stopwords, Stemiza todas as palavras e passa tudo para o padro ASCII
def getCleanText(text):
    finalTextArray = []
    lowerText = text.lower()
    teste = 0
    for word in lowerText.split():
        teste = teste + 1
        if word not in STOPWORDS:
            finalTextArray.append(STEMMER.stem(word))
    finalText = ' '.join(finalTextArray)
    finalText = normalize('NFKD', finalText).encode('ASCII', 'ignore').decode('ASCII')
    finalText = REGEXP_REMOVE_SPECIAL.sub('', finalText)
    finalText = re.sub(' +', ' ', finalText)
    return finalText
```

```
32
bases = ['economia', 'aereo', 'turismo']
34
for tag in bases:
36    # L   a base de dados bruta
      articles = pd.read_excel('saidaJoin/' + tag + '.xlsx')
38    demand = pd.read_excel('demanda.xlsx')

40    # Retira da base as not cias sem texto
      articles = articles[articles.texto.isnull() == False]
42
      # Limpa todos os textos das not cias usando a fun   o getCleanText
44    articles['texto'] = articles['texto'].apply(lambda x: getCleanText(x))

46    # Junta o dataframe de not cias com o dataframe de varia   o na
          demanda por transporte a reo usando a coluna 'date' como
          refer ncia
      articles = articles.merge(demand, how='inner', on='Ano')
48    articles.drop('Ano', inplace=True, axis=1)
      articles.columns = ['text', 'label']
50
      # Organiza a lista de not cias de forma rand mica
52    articles = articles.sample(frac=1)
      articles['label'] = pd.to_numeric(articles['label'])
54    cols = articles.columns.tolist()
      cols = cols[-1:] + cols[:-1]
56    articles = articles[cols]

58    # Contabiliza o n mero de not cias para a base de treino, teste e
          valida   o
      trainSize = round(len(articles) * (TRAIN_PERCENTAGE_SIZE))
60    testSize = round(len(articles) * (TEST_PERCENTAGE_SIZE))

62    # Separa as bases em arquivos diferentes
      data_train = articles.head(trainSize)
64    data_test = articles.iloc[trainSize:(trainSize + testSize)]
      data_validation = articles.iloc[(trainSize + testSize):]
66
      data_train.to_csv('saidaBaseFinal/data_train_' + tag + '.csv', index=
          False, header=False)
68    data_train.to_csv('saidaBaseFinal/data_train_' + tag + '_TSV.tsv', sep=
          '\t', quoting=csv.QUOTE_NONE, index=False, header=False)
      data_test.to_csv('saidaBaseFinal/data_test_' + tag + '.csv', index=
          False, header=False)
70    data_test.to_csv('saidaBaseFinal/data_test_' + tag + '_TSV.tsv', sep='\
          t', quoting=csv.QUOTE_NONE, index=False, header=False)
      data_validation.to_csv('saidaBaseFinal/data_validation_' + tag + '.csv'
```

```
        , index=False, header=False)
72    data_validation.to_csv('saidaBaseFinal/data_validation_' + tag + '_TSV.
        tsv', sep='\t', quoting=csv.QUOTE_NONE, index=False, header=False)
```

FOLHA DE REGISTRO DO DOCUMENTO

[1.] CLASSIFICAÇÃO/TIPO	[2.] DATA	[3.] REGISTRO N°	[4.] N° DE PÁGINAS
TC	24 de novembro de 2021	DCTA/ITA/TC-116/2021	90

[5.] TÍTULO E SUBTÍTULO:

Previsão de demanda por transporte aéreo baseado em processamento de linguagem natural e *deep learning*.

[6.] AUTOR:

Felipe Leonardo Sarmento da Silva

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica - ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Demanda por Transporte Aéreo; Deep Learning; Processamento de Linguagem Natural.

9.PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Transporte aéreo; Demanda (economia); Planejamento estratégico; Processamento da linguagem natural; Transpostes.

[10.] APRESENTAÇÃO: **(X) Nacional ( ) Internacional**

ITA, São José dos Campos. Curso de Graduação em Engenharia Civil-Aeronáutica. Orientador: Prof. Dr. Marcelo Xavier Guterres. Publicado em 2021.

[11.] RESUMO:

O transporte aéreo é um serviço essencial para o desenvolvimento do país e para a sociedade, logo é de suma importância que Estado e instituições privadas realizem um planejamento estratégico de forma a maximizar a eficiência do setor, para isso a previsão de demanda por transporte aéreo é peça fundamental. Existem alguns métodos na literatura para previsão de demanda, como o econométrico e o gravitacional, porém todos com as suas limitações. Este trabalho propõe um novo método para a tarefa, baseado em aprendizado profundo e processamento de linguagem natural. Para isso, criou-se uma base de dados com notícias de jornal relacionadas ao setor aéreo de 2006 a 2018 utilizando raspagem web. Essa base de dados foi associada com os dados da ANAC de demanda anual por passagens aéreas. A base associada serviu de entrada para a rede neural que, por meio de regressão, realiza as previsões de demanda. Os resultados mostraram que o método é eficaz para realizar a tarefa proposta e que as notícias de jornal relacionadas ao transporte aéreo tem conteúdo linguístico suficiente para prever com exatidão a demanda por transporte aéreo.

[12.] GRAU DE SIGILO:

(X) OSTENSIVO ( ) RESERVADO ( ) SECRETO