

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

MATHEUS NASCIMENTO SOARES MARQUES DE LIMA

**PROCESSAMENTO, TRANSCRIÇÃO E CLASSIFICAÇÃO
AUTOMATIZADA DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO
SETOR ELÉTRICO APLICANDO TÉCNICAS DE APRENDIZADO DE
MÁQUINA E PROCESSAMENTO DE LINGUAGEM NATURAL**

FLORIANÓPOLIS, 2021.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

MATHEUS NASCIMENTO SOARES MARQUES DE LIMA

**PROCESSAMENTO, TRANSCRIÇÃO E CLASSIFICAÇÃO
AUTOMATIZADA DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO
SETOR ELÉTRICO APLICANDO TÉCNICAS DE APRENDIZADO DE
MÁQUINA E PROCESSAMENTO DE LINGUAGEM NATURAL**

Trabalho de Conclusão de Curso submetido
ao Instituto Federal de Educação, Ciência e
Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
Engenheiro Eletricista.

Orientador:
Prof. Fabrício Y. K. Takigawa, Dr. Eng.

FLORIANÓPOLIS, 2021.

Ficha de identificação da obra elaborada pelo autor.

Lima, Matheus Nascimento Soares Marques de
PROCESSAMENTO, TRANSCRIÇÃO E CLASSIFICAÇÃO AUTOMATIZADA
DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO SETOR ELÉTRICO APLICANDO
TÉCNICAS DE APRENDIZADO DE MÁQUINA E PROCESSAMENTO
DE LINGUAGEM NATURAL / Matheus Nascimento
Soares Marques de Lima; orientação de Fabricio Yutaka
Kuwabata Takigawa. - Florianópolis, SC, 2021.
87 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Elétrica. Departamento Acadêmico
de Eletrotécnica.
Inclui Referências.

1. Operação do Setor Elétrico. 2. Transcrição de
Arquivos de Áudio. 3. Processamento de Linguagem Natural.
4. Aprendizado de Máquina. 5. Classificação Multiclasse.
I. Takigawa, Fabricio Yutaka Kuwabata. II. Instituto
Federal de Santa Catarina. III. PROCESSAMENTO,
TRANSCRIÇÃO E CLASSIFICAÇÃO AUTOMATIZADA DA COMUNICAÇÃO
VERBAL NA OPERAÇÃO DO SETOR ELÉTRICO APLICANDO TÉCNICAS
DE APRENDIZADO DE MÁQUINA E PROCESSAMENTO DE LINGUAGEM
NATURAL

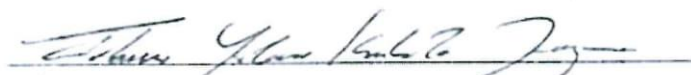
**PROCESSAMENTO, TRANSCRIÇÃO E CLASSIFICAÇÃO
AUTOMATIZADA DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO
SETOR ELÉTRICO APLICANDO TÉCNICAS DE APRENDIZADO DE
MÁQUINA E PROCESSAMENTO DE LINGUAGEM NATURAL**

MATHEUS NASCIMENTO SOARES MARQUES DE LIMA

Este trabalho foi julgado adequado para obtenção do título de Engenheiro Eletricista em 2021 e aprovado na sua forma final pela banca examinadora do Curso de Graduação em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 22 de Setembro de 2021.

Banca Examinadora:



Prof. Fabrício Yutaka Kuwabata Takigawa, D. Eng.



Narciso Ferreira Barbosa, Eng.



Prof. Carlos Ernani da Veiga, M. Eng.



Assinado de forma digital
por Rubiapiara Fernandes
Dados: 2021.10.08
08:59:46 -03'00'

Prof. Rubiapiara Cavalcante Fernandes, D. Eng.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais Alessandra e Valdison pelo constante apoio psicológico e financeiro para que eu sempre tenha acesso a uma educação de qualidade.

A minha namorada Leticia, por ser minha companheira e refúgio estando do meu lado durante esses anos me dando suporte, energia e incentivo, sem ela a graduação com certeza seria muito mais difícil e desgastante. A Flávia e André que se mostrarem muito zelosos nesse período.

Ao meu orientador Fabricio, quem me guiou desde o início da minha graduação e sempre me impulsionou a ir além, muito obrigado pela confiança, aprendizado e amizade durante esses anos.

Aos meus colegas de turma, em especial: Luiz, Rafael (Tomi), Bruno, Marcos, João, Felipe, Fernando e Vitor, por compartilhar conhecimento e por tornar a vida acadêmica mais feliz.

Aos meus parceiros de pesquisa Rodrick, Allon, Werik e Bruna, por me ensinarem a trabalhar em equipe e proporcionarem muito aprendizado.

E as instituições IFSC, CNPq e GESE por possibilitarem a minha participação e desenvolvimento de três projetos de pesquisa ao longo da graduação.

Ao ONS pelo estágio e pela disponibilização dos dados para o estudo.

RESUMO

A operação centralizada do Sistema Interligado Nacional (SIN) é uma tarefa complexa desempenhada pelo Operador Nacional do Sistema Elétrico (ONS). O SIN ao longo do território brasileiro possui diversos equipamentos e tipos de agentes. De maneira geral, os centros regionais de operação se comunicam com os agentes por meio de ligações telefônicas. Toda comunicação efetuada nas salas de operação é gravada (armazenada em um servidor) e, posteriormente, é disponibilizada para o setor de pós-operação, para ser consultada quando for necessário. O setor de pós-operação do ONS efetua a ponderação das ações tomadas pelos operadores, assim como a confirmação de informações da operação realizada (validação de dados horários, montantes, tipo de serviço, tipo de ocorrências, entre outros). Normalmente, esse procedimento é realizado por meio de oitivas das ligações efetuadas, ou seja, consultando os dados por meio da escuta de áudios. Pode-se afirmar que esse procedimento é um aspecto rotineiro e ilustra uma lacuna que pode ser aprimorada, visto que um dos principais problemas é o tempo gasto na realização, que dependendo da ocorrência sistêmica podem ser necessárias diversas oitivas. Neste sentido, o presente trabalho propõe uma metodologia para o processamento e análise dos arquivos de áudios e de texto. Tem-se como resultados uma transcrição que consiga preservar as principais informações das oitivas e a análise de classificação multiclasse da comunicação verbal na operação. Para isso explora-se técnicas de Processamento de Linguagem Natural (NLP) e algoritmos de Aprendizado de Máquina (ML) tendo como principal ferramenta a linguagem de programação Python. Adicionalmente foi desenvolvido um Produto Minimamente Viável (MVP) capaz de classificar dados de texto ou áudio seguindo a metodologia desenvolvida. Para o trabalho foi explorado um *dataset* de arquivos de áudio da comunicação verbal do ONS com os agentes do setor elétrico. Obtendo resultados promissores com modelos atingindo até 85% de acerto na classificação do áudio.

Palavras-chave: Operação do Setor Elétrico. Transcrição de Arquivos de Áudio. Processamento de Linguagem Natural. Aprendizado de Máquina. Classificação Multiclasse.

ABSTRACT

A centralized operation of the National Interconnected System (SIN) is a complex task performed by the National Electric System Operator (ONS). The SIN throughout the Brazilian territory has several equipment and types of agents. In general, regional operating centers communicate with agents via telephone calls. All communication carried out in the operation is recorded (stored on a server) and, later, is available to the post-operation sector, to be consulted when necessary. The ONS post-operation sector perform an overview the operator's actions and confirms information of the operation (validation of hourly data, amounts, type of service, type of occurrences, among others). Usually, this procedure is carried out by listening to audios. It can be said that this procedure is a routine aspect and illustrates a gap that can be improved, since one of the main problems in this procedure is the time spent on the performance, depending of the systemic occurrence, several wiretaps listening's may be required. In this sense, the present work proposes a methodology for the processing and analysis of audio and text files. The results are a transcript that manages to preserve the main information from the hearings and the multiclass classification analysis of verbal communication in the operation. For this, Natural Language Processing techniques (NLP) and Machine Learning (ML) algorithms are explored, having as main tool the Python programming language. Additionally, a Minimally Viable Product (MVP) was developed, capable of classifying text or audio data following the developed methodology. For this work, a dataset of audio files of verbal communication between ONS and agents in the electricity sector was explored. Obtaining promising results with models reaching up to 85% of accuracy.

Keywords: Operation of Electric Sector. Audio File Transcription. Natural Language Processing. Machine Learning. Multiclass Classification.

LISTA DE FIGURAS

Figura 1 – Áreas estudadas para o procedimento proposto.....	16
Figura 2 – Etapas para a comunicação verbal na operação	17
Figura 3 – Tendência das linguagens de programação	18
Figura 4 – Exemplo de designação de variáveis	21
Figura 5 – Composição do conjunto de dados	24
Figura 6 – Processo geral relacionado a NLP	24
Figura 7 – Exemplo de <i>stemming</i> e <i>lemmatization</i>	25
Figura 8 – Processo de vetorização hipotético.....	27
Figura 9 – Detalhe do processo de vetorização.	27
Figura 10 – Exemplo de aplicação da técnica N-Grams	28
Figura 11 – Divisão dos dados	32
Figura 12 – Matriz de confusão para um problema binário	33
Figura 13 – Metodologia para a classificação dos áudios da operação do setor elétrico.....	37
Figura 14 – Características do <i>dataset</i> explorado.....	38
Figura 15 – Características dos dados rotulados	41
Figura 16 – Processo de tratamento do áudio para transcrição	42
Figura 17 – Processo de preparação dos dados textuais.....	44
Figura 18 – Tela de agrupamento do <i>software</i> OpenRefine.....	45
Figura 19 – Imagem de humor sobre dados desbalanceados.....	46
Figura 20 – Legenda de extração de características.....	46
Figura 21 – Extração de características para cada modelo.....	47
Figura 22 – Resultados do desempenho das amostras de teste.....	50
Figura 23 – Resultados do desempenho das amostras de teste por classe	51
Figura 24 – Curva de aprendizado dos modelos.....	52
Figura 25 – Matriz de Confusão dos modelos	54
Figura 26 – Fluxo de navegação da aplicação	57
Figura 27 – Tela inicial da aplicação	57
Figura 28 – Tela para entrada de texto.	58
Figura 29 – Tela para entrada de áudio.	58

LISTA DE TABELAS

Tabela 1 – Lista de bibliotecas em Python para a realização do estudo	18
Tabela 2 – Lista de serviços estudados	19
Tabela 3 – Limitações dos planos gratuitos	20
Tabela 4 – Modelos de ML utilizados para classificação	29
Tabela 5 – Lista de classes e quantidade de amostras	40
Tabela 6 – Valores médios do F1 ponderado das dobras de teste	48
Tabela 7 – Quantidade de colunas da matriz numérica para cada alcance de <i>ngram</i>	48
Tabela 8 – Melhores parâmetros usados para a criação dos modelos	49
Tabela 9 – Valores da métrica MCC juntamente com o F1 ponderado para cada modelo	56

LISTA DE ABREVIATURAS E SIGLAS

API – Interface de Programação de Aplicações (*Application Programming Interface*)

CBOW – Saco Contínuo de Palavras (*Continuous Bag of Words*)

FN – Falso Negativo

FP – Falso Positivo

IDE – Ambiente de Desenvolvimento Integrado (*Integrated Development Environment*)

IDF – Inverso da Frequência nos Documentos (*Inverse Document Frequency*)

KER – Taxa de Erro de Palavras-Chave (*Keyword Error Rate*)

MCC – *Matthews Correlation Coefficient*

MER – Taxa de Erro de Correspondência (*Match Error Rate*)

ML – Aprendizado de Máquina (*Machine Learning*)

MVP – Produto Minimamente Viável (*Minimum Viable Product*)

NLP – Processamento de Linguagem Natural (*Natural Language Processing*)

ONS – Operador Nacional do Sistema Elétrico

PR – Procedimentos de Rede

SIG – Sistema de Gerenciamento de Intervenções

SIN – Sistema Interligado Nacional

SRS – Sistemas de Reconhecimento de Voz (*Speech Recognition Systems*)

STT – Fala para Texto (*Speech to Text*)

SVM – Máquina de vetores de suporte (*Support Vector Machine*)

TF – Frequência do Termo (*Term Frequency*)

VF – Verdadeiro Falso

VP – Verdadeiro Positivo

WER – Taxa de Erro de Palavras (*Word Error Rate*)

WIL – *Word Information Lost*

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Justificativa	11
1.2	Definição do Problema	13
1.3	Objetivos	14
1.3.1	Objetivo Geral	14
1.3.2	Objetivos Específicos	14
1.4	Estrutura do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Comunicação Verbal na Operação	16
2.2	Linguagem de Programação Python	17
2.3	Sistemas de Reconhecimento de Voz	19
2.3.1	Limitações dos planos gratuitos	20
2.3.2	Métricas de Desempenho de Sistemas de Reconhecimento de Voz	20
2.3.2.1	<i>Taxa de Erro de Palavras (Word Error Rate - WER)</i>	21
2.3.2.2	<i>Taxa de Erro de Correspondência (Match Error Rate - MER)</i>	22
2.3.2.3	<i>Word Information Lost - WIL</i>	22
2.3.2.4	<i>Taxa de Erro de Palavras-Chave (Keyword Error Rate - KER)</i>	22
2.4	Processamento de Linguagem Natural (NLP)	23
2.4.1	Processamento de Texto	25
2.4.2	Relevância das Palavras.....	26
2.4.3	N-Grams.....	28
2.5	Aprendizado de Máquina (ML)	28
2.5.1	Complement Naive Bayes.....	29
2.5.2	Linear Support Vector	30
2.5.3	Stochastic Gradient Descent (SGD).....	30
2.5.4	K-Nearest Neighbors.....	31
2.5.5	Multi-layer Perceptron	31
2.5.6	Random Forest Classifier.....	31
2.5.7	Divisão dos Dados e Validação Cruzada	32
2.5.8	Métricas de Desempenho para ML	33
2.5.8.1	<i>Matriz de Confusão (Confusion Matrix)</i>	33
2.5.8.2	<i>Acurácia (Accuracy)</i>	34

2.5.8.3	<i>Precisão (Precision)</i>	34
2.5.8.4	<i>Sensibilidade (Recall)</i> :.....	34
2.5.8.5	<i>F1 Score</i>	34
2.5.8.6	<i>Curva de Aprendizado</i>	35
2.5.8.7	<i>Matthews Correlation Coefficient (MCC)</i>	35
2.5.9	Sobreajuste e Subajuste	35
3	DESCRIÇÃO DA METODOLOGIA	37
3.1	Caracterização dos dados	38
3.1.1	Rotulagem dos Dados.....	39
3.1.2	Caracterização dos Dados Rotulados.....	40
3.2	Pré-Processamento (Áudios)	42
3.3	Pós-Processamento (Textos)	43
3.4	Classificação	45
3.4.1	Resultados	49
3.4.1.1	<i>Métricas Gerais</i>	50
3.4.1.2	<i>Curva de Aprendizado</i>	52
3.4.1.3	<i>Matriz de Confusão</i>	53
3.4.1.4	<i>Matthews Correlation Coefficient (MCC)</i>	55
3.5	Produto Minimamente Viável (MVP)	56
4	CONCLUSÃO	60
	REFERÊNCIAS	62
	APÊNDICE A – ARTIGO FERRAMENTAS E RECURSOS PARA RECONHECIMENTO DE FALA EM PORTUGUÊS BRASILEIRO	67
	APÊNDICE B – COMPARAÇÃO ENTRE BIBLIOTECAS E PARÂMETROS RELACIONADOS A SÉGMENTAÇÃO DE ÁUDIOS	73
	APÊNDICE C – DICIONÁRIO EM PYTHON PARA PROCESSAMENTO DAS TRANSCRIÇÕES	76
	APÊNDICE D – TUTORIAL PARA USO DO SISTEMA PARA CLASSIFICAÇÃO DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO SETOR ELÉTRICO	82

1 INTRODUÇÃO

O Brasil possui o Sistema Interligado Nacional (SIN) com centenas de milhares de quilômetros de linhas de transmissão e diversas fontes de geração espalhadas ao longo de seu imenso território. Esse sistema é operado de forma centralizada pelo Operador Nacional do Sistema Elétrico (ONS), que possibilita otimizar os recursos energéticos, reduzindo os custos operacionais no presente e no futuro (ONS, 2020a).

O ONS é composto por diversas gerências, cabe citar a Gerência Executiva de Operação Nacional do Sistema e a Gerência Executiva de Apuração, Análise e Custos da Operação. Para uma eficiente operação do sistema é fundamental que as gerências citadas possuam uma boa comunicação entre si e entre os agentes prestadores de serviços de energia elétrica (ONS, 2020b).

Atualmente o principal meio de comunicação dos operadores do ONS com os agentes do setor elétrico é por ligações telefônicas. É por meio delas que o SIN é operado e diversas atividades são realizadas e acompanhadas. Tais como mudanças de estado operativo de unidades geradoras, controle da transmissão, manutenção de equipamentos, informações hidrológicas, comprovação de disponibilidade, entre outros.

Nesse sentido, surge uma oportunidade de transformar gravações brutas em inteligência operacional por meio de uma metodologia capaz de processar, transcrever e analisar a comunicação verbal do setor elétrico. Tendo potencial para gerar produtos que assistem os agentes do setor. Podendo gerar de indicadores de desempenho até automatizar tarefas e ações.

1.1 Justificativa

Toda comunicação verbal realizada pelos operadores do ONS é gravada e armazenada em um servidor. Os operadores do setor elétrico seguem as diretrizes dos Procedimentos de Rede (PR), nelas existe a rotina de operação 4.7.1 do Submódulo 10.22 que padroniza a Comunicação Verbal na Operação, logo os diálogos realizados entre os operadores seguem obrigatoriamente um padrão de apresentação, comando, repetição do comando e confirmação (ONS, 2020c).

As atividades de apuração que necessitam de realização de oitivas da comunicação verbal são listadas a seguir (ONS, 2009):

- a) Sistema Integrado de Perturbações, que apura os desligamentos forçados de equipamentos;
- b) Sistema de Apuração da Transmissão, responsável pela computação de ações realizadas em linhas de transmissão;
- c) Apuração das Mudanças de Estados Operativos de UGs, encarregado de registrar as mudanças de estados operativos de unidades geradoras.

Já no âmbito de análise poderia consultar as transcrições para os seguintes processos:

- a) Avaliação da Comunicação Verbal da Operação, que verifica se às equipes de tempo real estão respeitando as regras, os procedimentos básicos e a fraseologia padrão.
- b) Análise Diária da Operação, responsável pela análise geral da operação, observando carregamentos de transformadores e linhas de transmissão;
- c) Triagem de Ocorrências e Perturbações, compreende a seleção de eventos conforme critérios preestabelecidos, para análise e emissão de relatórios específicos;
- d) Análise de Ocorrências, tem o objetivo de atribuir responsabilidades nos processos de análise de ocorrências e perturbações e de estabelecer as diretrizes básicas, a metodologia e os critérios para a análise dos aspectos técnicos das ocorrências e perturbações na rede de operação do SIN;
- e) Análise das Perturbações no Sistema, avalia os processos operativos envolvidos, principalmente o de recomposição do sistema;
- f) Análise da Operação do Sistema, estabelece as premissas, diretrizes, critérios e responsabilidades para a análise detalhada da operação do sistema e dos processos operativos de tempo real, bem como para o

acompanhamento e gerenciamento das recomendações registradas nos relatórios de análise, visando o aprimoramento da operação.

Apesar dos processos da pós-operação estarem fortemente atrelados a realização de oitivas da comunicação entre os operadores, os arquivos de áudio são armazenados em um servidor e somente são acessados para a realização de consultas de informações. Nesse sentido, nota-se uma oportunidade de aplicação de técnicas de análise de dados e aprendizado de máquina para evidenciar certos padrões de comportamento na operação da região sul do SIN.

Por meio da análise de dados é possível levantar estudos de processamento de áudio e texto, buscando uma transcrição dos áudios da operação, servindo como insumo para aprendizado de máquina viabilizando uma forma de efetuar uma classificação automatizada da comunicação verbal da operação.

1.2 Definição do Problema

O SIN é o sistema de produção e de transmissão de energia elétrica do Brasil e sua operação é realizada pelo ONS. Como citado anteriormente, o principal meio de comunicação dos operadores do ONS com os agentes do setor elétrico é por ligações telefônicas. Neste sentido, uma comunicação de qualidade entre os operadores é de fundamental importância para a operação eficiente do SIN, conforme está definido no submódulo 13.2, que define os requisitos mínimos de telecomunicações (ONS, 2020d).

O setor da pós-operação realiza rotineiramente escutas de ligações realizadas pelos operadores para apurar e confirmar as ações adotadas pela operação e uma das dificuldades encontradas pelo setor da pós-operação é a demanda de tempo de oitivas da comunicação entre ONS e agentes, para validação de informações. Pois no sistema utilizado somente é possível buscar os arquivos de áudio por meio da hora da ligação e do agente, logo, minutos são gastos somente para encontrar uma ocorrência específica. Outros aspectos da ligação como assunto ou informações como hora informada, subestação e equipamento tratado não são expostos.

Nesse sentido, surgem alguns questionamentos: existe uma forma mais atrativa de disponibilizar os áudios para o setor da pós-operação? Qual a melhor forma de transcrever e analisar o banco de áudios? Qual o melhor serviço de transcrição dos arquivos de áudio da operação? Quais são as melhores técnicas de análise de dados para a classificação e extração de informações relevantes das escutas? A classificação e identificação de entidades pode agregar maior eficiência para as atividades da pós-operação? Um Produto Minimamente Viável (*Minimum Viable Product - MVP*) capaz de transcrever e analisar os arquivos de áudio da operação seria algo atrativo para os agentes do setor elétrico?

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver uma metodologia para análise de áudios da operação do setor elétrico, processando-os, transcrevendo-os e classificando-os através de algoritmos de Aprendizado de Máquina (*Machine Learning - ML*) e técnicas de Processamento de Linguagem Natural (*Natural Language Processing - NLP*).

1.3.2 Objetivos Específicos

Os itens listados a seguir mostram os objetivos específicos que o estudo busca atingir ao longo de seu desenvolvimento:

- a) Organizar o banco de áudios (*dataset*) da comunicação verbal entre os operadores do setor elétrico¹;
- b) Estudar serviços de transcrição de voz e selecionar os mais atrativos considerando suas limitações e métricas de desempenho de sistema de reconhecimento de voz;
- c) Desenvolver os algoritmos para pré-processamento dos arquivos de áudios e para consumo de serviços de transcrição de voz selecionados;

¹ *Dataset* cedido pelo ONS pela gerência da pós operação em 27/08/2020.

- d) Desenvolver uma metodologia para rotulagem dos textos transcritos, classificando manualmente uma determinada quantidade de transcrições;
- e) Estudar algoritmos de ML e selecionar os mais viáveis para o problema de classificação multiclasse;
- f) Estudar técnicas de NLP para aplicação dos algoritmos de ML;
- g) Aplicar os algoritmos de ML selecionados no *dataset*;
- h) Avaliar o desempenho apresentado pelos algoritmos;
- i) Desenvolver um MVP que agrega a metodologia desenvolvida.

1.4 Estrutura do Trabalho

O documento está organizado em quatro capítulos principais. Inicialmente foi apresentada a introdução do tema, em que são definidos a justificativa, a definição do problema e os objetivos propostos. No Capítulo 2 é apresentada a fundamentação teórica estudada para a realização do trabalho, contendo os principais conceitos das áreas de estudo a serem exploradas. Posteriormente, no Capítulo 3 é descrita a metodologia para a transcrição e classificação multiclasse dos dados. E, por fim, no Capítulo 4 são apresentadas as conclusões e recomendações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Inicialmente, analisa-se a comunicação verbal na operação do setor elétrico, sendo necessário o estudo de sistemas de reconhecimento de voz. As transcrições, resultantes do sistema de reconhecimento de voz formam uma base de dados textuais que pode ser analisada por meio de técnicas de processamento de linguagem natural e, posteriormente, classificar a transcrição utilizando algoritmos de ML. A Figura 1 ilustra os principais campos de estudo para a execução do procedimento citado.

Figura 1 – Áreas estudadas para o procedimento proposto



Fonte: Elaboração própria (2021).

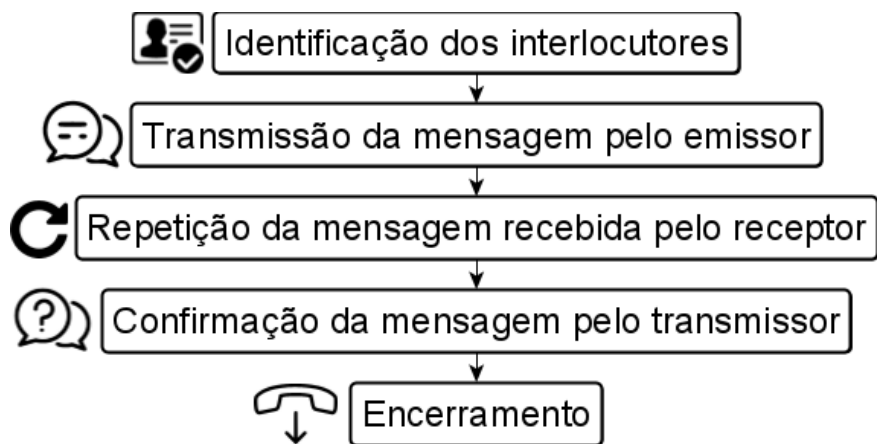
Pode-se observar que a primeira caixa se refere à comunicação verbal na operação do setor elétrico; a segunda, com a conversão do conjunto de dados dos áudios em texto; e, a terceira, refere-se ao processamento de linguagem natural juntamente com o aprendizado de máquina. Nas seções seguintes, será elucidada como a comunicação verbal ocorre, as ferramentas utilizadas, o levantamento efetuado dos sistemas de transcrição de voz e uma avaliação sobre a qualidade dos mesmos, técnicas aplicadas para o processamento textual e, por fim, algoritmos ML com as métricas para avaliação do aprendizado dos modelos treinados.

2.1 Comunicação Verbal na Operação

A operação centralizada do SIN é uma tarefa complexa desempenhada pelo ONS. O SIN ao longo do território brasileiro possui diversos equipamentos e tipos de agentes. De maneira geral, os centros regionais de operação se comunicam com os agentes por meio de ligações telefônicas (ONS, 2020c).

Para padronizar essa atividade, o ONS elaborou um documento específico para regularizar a comunicação no setor elétrico nacional. O Rotina de Operação RO-RO.BR.01, denominada Comunicação Verbal na Operação, que visa padronizar a comunicação entre os diversos agentes que operam o SIN (ONS, 2020c). A Figura 2 ilustra as principais etapas da padronização, que visa minimizar a ocorrência de possíveis equívocos.

Figura 2 – Etapas para a comunicação verbal na operação



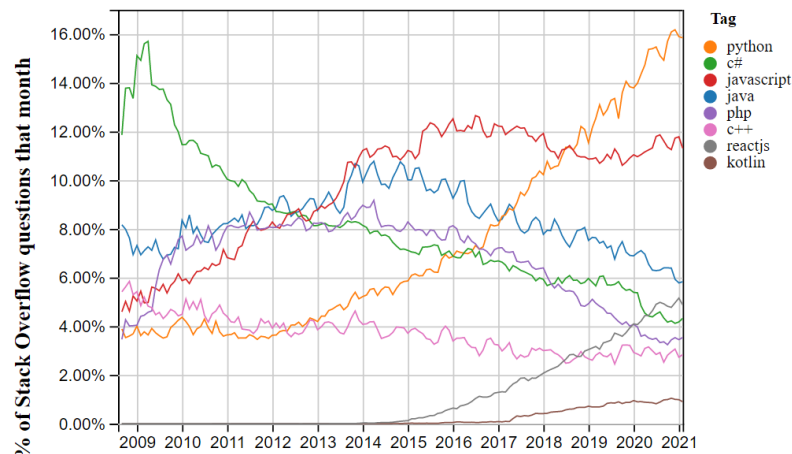
Fonte: Elaboração própria (2021).

2.2 Linguagem de Programação Python

A principal ferramenta para a execução do estudo foi a linguagem de programação Python (VAN ROSSUM GUIDO, DRAKE JR, 2009), que apresenta muitas vantagens, sendo as principais: *software* de código aberto (*open-source*), adaptação em diferentes sistemas operacionais, linguagem intuitiva (*user-friendly*), possibilidade de implementação de várias bibliotecas, gerenciamento de memória, operações em múltiplas *threads* e grande documentação básica (VIARBITSKAYA; DOBRUCKI, 2018).

Adicionalmente é a linguagem mais discutida, atualmente, pela comunidade de programadores StackOverflow (STACK EXCHANGE, 2021). A Figura 3 exibe a tendência da discussão das linguagens de programação por mês. Pode-se concluir que a linguagem apresentada é a que exibe maior crescimento de discussões durante os últimos anos é o Python (desde 2019).

Figura 3 – Tendência das linguagens de programação



Fonte: Stack Exchange (2021).

Para a programação com Python, basicamente, é necessário ter definido três itens: o *Core*, o Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* - IDE) e as Bibliotecas. O *Core* é o interpretador da linguagem junto com algumas bibliotecas padrões. A IDE é a interface usada para o desenvolvimento, cita-se: Pyzo, PyCharm, Spider, Jupiter. Em relação às bibliotecas, a linguagem Python apresenta uma enorme quantidade para a realização de atividades que podem ser acessadas usando o site PyPI (2020). A Tabela 1 exibe as principais bibliotecas exploradas e suas descrições.

Tabela 1 – Lista de bibliotecas em Python para a realização do estudo

Biblioteca	Descrição	Referência
numpy	Pacote para Python suportar vetores e matrizes multidimensionais.	(NUMPY, 2005)
pandas	Similar ao numpy, usado para manipulação e análise de dados.	(PANDAS, 2021)
auditok	Biblioteca para detecção de atividade de áudio.	(SEHILI, 2021)
matplotlib	Biblioteca abrangente para a visualização de dados.	(HUNTER; DROETTBOOM, 2021)
SpeechRecognition	Biblioteca para realização de reconhecimento de fala, com suporte para diversas engines e APIs, online e offline.	(ZHANG, 2017)
wave	Fornece uma interface para o formato de som WAV.	(PYTHON, 2021)
jiwer	Pacote para medidas de similaridade para avaliação de sistemas de reconhecimento de voz.	(VAESSEN, 2004)
scikit-learn	Biblioteca de código aberto para ML.	(PEDREGOSA et al., 2011)
Gensin	Representa documentos como vetores semânticos.	(ŘEHŮŘEK, 2009)

Fonte: Elaboração própria (2021).

2.3 Sistemas de Reconhecimento de Voz

A necessidade por reconhecimento de fala é algo cada vez mais presente no cotidiano. O conceito da Internet das Coisas e da Indústria 4.0 eleva cada vez mais a proximidade do ser humano com a máquina. Nesse sentido, a crescente necessidade por Sistemas de Reconhecimento de Voz (*Speech Recognition Systems* - SRS) acarretou na abertura de um mercado de serviços *online* que realizam a tarefa de conversão de fala para texto (*Speech to Text* - STT). Existem muitos serviços de reconhecimento de voz no mercado. Os principais são listados na Tabela 2.

Tabela 2 – Lista de serviços estudados

Serviço	Descrição	Referência
Microsoft Azure	Reconhecimento de voz, permite a transcrição em tempo real e em lote de fluxo de áudio em texto.	(MICROSOFT, 2010)
Google Cloud	Usa tecnologia de reconhecimento de voz de código fechado com base em aprendizagem profunda.	(GOOGLE, 2016)
IBM Watson	Serviço aproveita o aprendizado de máquina para combinar o conhecimento da gramática, a estrutura do idioma e a composição de sinais de áudio e de voz e atualiza continuamente e refina sua transcrição à medida que recebe mais falas.	(IBM, 2004)
Amazon Transcribe	Usa um processo de aprendizagem profunda chamada reconhecimento automático de fala para converter fala em texto.	(AMAZON, 2017)
Wit.ai	Grátis até para uso comercial, suporta 130 línguas. Suporta as linguagens de programação Node, Python e Ruby.	(WIT.AI, 2013)
CMU Sphinx	Um SRS de código aberto offline para dispositivos mobile e servidores. Suporta as linguagens C, C++, C#, Python, Ruby, Java e Javascript.	(CARNEGIE MELLON UNIVERSITY, 2000)
Vosk	Biblioteca de código aberto derivada da CMU Sphinx. Suporta 17 línguas e dialetos, código-aberto e funciona offline.	(ALPHACEPHEI, 2019)

Fonte: Elaboração própria (2021).

Pode-se citar como exemplos as empresas *Microsoft*, *Google*, *IBM*, *Amazon* e *Wit.ai*. Entretanto, em paralelo a esse mercado, existem iniciativas de código aberto que também buscam desempenhar essa atividade de forma *offline*, como *CMU Sphinx* e *Vosk* (JUANG; RABINER, 2004).

Para consumir o SRS *online* é necessário o uso da Interface de Programação de Aplicações (*Application Programming Interface* - API) disponibilizada pela empresa. Para sistemas *offline* não é necessária a conexão com a internet e não apresenta restrições de uso.

2.3.1 Limitações dos planos gratuitos

Dentre as diversas opções apresentadas, seja de código aberto ou fechado, cada serviço pode possuir vantagens e desvantagens. Pode-se afirmar que a quantidade de opções disponíveis, dificulta a tomada de decisão do consumidor.

Para alguns serviços, existe a possibilidade de contratação de um plano pago com mais vantagens e menos limitações. A Tabela 3 mostra as limitações dos planos disponibilizados gratuitamente (LIMA; COELHO; TAKIGAWA, 2021).

Tabela 3 – Limitações dos planos gratuitos

	Azure	IBM	Google	Amazon	Wit.ai	Sphinx	Vosk
Online	Sim	Sim	Sim	Sim	Sim	Não	Não
Pagos	Sim	Sim	Sim	Sim	Não	Não	Não
Limite/mês	300 min	500 min	60 min	60 min	∞	∞	∞
Limite/requisição	1 min	100 MB	1 min	4 h ou 2 GB	20 seg	∞	∞
Personalização Grátis	Sim	Sim	Não	Sim	Não	Sim	Sim

Fonte: Elaboração própria (2021).

As opções *offline* não apresentam restrições de uso, entretanto, de acordo com os testes efetuados, as mesmas não têm uma performance tão boa quanto as soluções disponíveis no mercado. *Azure*, *IBM*, *Google* e *Amazon* são empresas consolidadas no ramo de SRS, mas fornecem os serviços gratuitos de maneira limitada. E, quando se busca o escalonamento de um produto é inviável consumir esses sistemas de maneira gratuita. Nesse sentido, o *Wit.ai* surge como um bom candidato a ser explorado por não apresentar limite mensal, apesar de impor o tempo pequeno de no máximo 20 segundos por requisição.

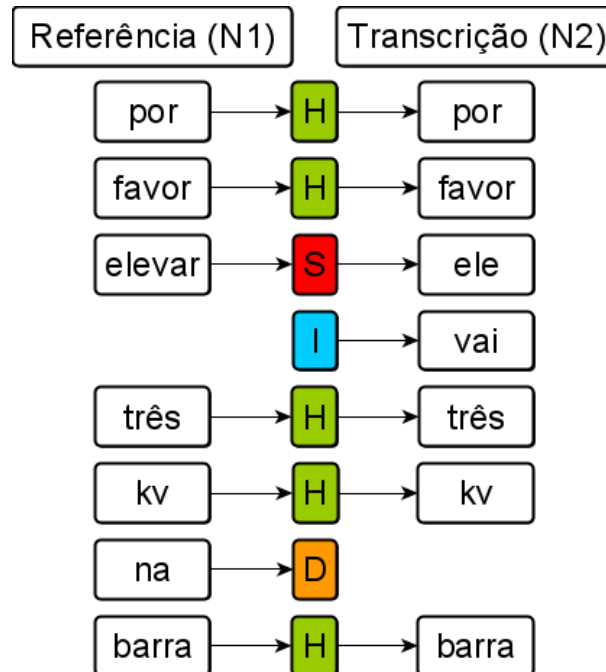
2.3.2 Métricas de Desempenho de Sistemas de Reconhecimento de Voz

Em meio as várias opções de serviços e reconhecimento de fala, surge a necessidade de métodos de avaliação dos SRS. Nesse sentido, como uma forma de auxílio para a tomada de decisão em relação a qual serviço escolher, cabe o uso de métricas para análise das transcrições (MORRIS; MAIER; GREEN, 2004; ERRATTAHI; HANNANI; OUAHMANE, 2018).

Antes do cálculo das métricas, cabe apresentar o sistema de variáveis das equações. Considera-se a referência do texto (N1), transcrição (N2), substituições (S),

deleções (D), inserções (I) e acertos (H). A Figura 4 exibe um exemplo seguindo esse sistema.

Figura 4 – Exemplo de designação de variáveis



Fonte: Elaboração própria (2021).

Tomando como exemplo a fala hipotética “por favor elevar três kv na barra” é possível supor uma transcrição “por favor ele vai três kv barra”. Nesse sentido, com o objetivo de exemplificar a aplicação de métricas para avaliação de SRS, é possível quantificar o número de acertos, substituições, inserções e deleções para o cálculo.

Na sequência, serão exemplificados os cálculos das métricas: Taxa de Erro de Palavras, Taxa de Erro de Correspondência, *Word Information Lost* e Taxa de Erro de Palavras-Chave.

2.3.2.1 Taxa de Erro de Palavras (Word Error Rate - WER)

Segundo Morris (2004), WER é a métrica mais comum para avaliação de SRS. A equação é descrita pela soma de substituições, deleções, inserções e divide pela soma de palavras da sentença correta ($N1$), conforme pode ser visto em (1).

$$WER = \frac{S + D + I}{H + S + D} = \frac{S + D + I}{N1} \quad (1)$$

Para o caso exemplo da Figura 4 a solução pode ser observada em (2).

$$WER = \frac{1 + 1 + 1}{7} = \frac{1 + 1 + 1}{5 + 1 + 1} = 0,43 = 43\% \quad (2)$$

2.3.2.2 Taxa de Erro de Correspondência (Match Error Rate - MER)

Uma segunda métrica recomendada por Morris (2004) é a MER, que é soma de *S*, *D* e *I* dividida pela soma dos *H*, *S*, *D* e *I*, evidenciada por (3).

$$MER = \frac{S + D + I}{H + S + D + I} \quad (3)$$

Para o caso exemplo da Figura 4 observa-se a solução em (4)

$$MER = \frac{1 + 1 + 1}{5 + 1 + 1 + 1} = 0,37 = 37\% \quad (4)$$

2.3.2.3 Word Information Lost - WIL

A última métrica proposta por Morris (2004) é a WIL.

$$WIL = 1 - \frac{H^2}{(H + S + D) \times (H + S + I)} = 1 - \frac{H^2}{N_1 N_2} \quad (5)$$

Para o caso exemplo da Figura 4 a solução calculada em (6)

$$WIL = 1 - \frac{5^2}{(5 + 1 + 1) \times (5 + 1 + 1)} = 0,49 = 49\% \quad (6)$$

2.3.2.4 Taxa de Erro de Palavras-Chave (Keyword Error Rate - KER)

No sentido de propor uma métrica diferente aos cálculos anteriores, é proposto neste trabalho a KER, que realiza a avaliação da transcrição levando em consideração as palavras-chave relacionadas ao tema do estudo. Neste sentido, para o caso da operação do setor elétrico, é possível observar a taxa de acerto de palavras relevantes isoladamente como nome de unidades de medida, equipamentos, ações, nomes de subestação, entre outros (PARK et al., 2008).

Cita-se por exemplo, as palavras “elevar”, “kv” e “barra” que são essenciais no contexto de controle de tensão pois somente com a aparição das mesmas em

conjunto é possível inferir que o objetivo da comunicação é elevar a tensão em uma barra (por exemplo, caso falte “kv” o operador não saberia o tipo grandeza em questão, podendo ajustar a potência equivocadamente).

$$KER = \frac{F + M}{N} \quad (7)$$

Onde:

F: Quantidade de palavras-chave falsamente identificadas.

M: Quantidade de palavras-chave não identificadas.

N: Quantidade de palavras-chave no texto de referência.

Logo, para o caso exposto, o valor da taxa de erro de palavras-chave é:

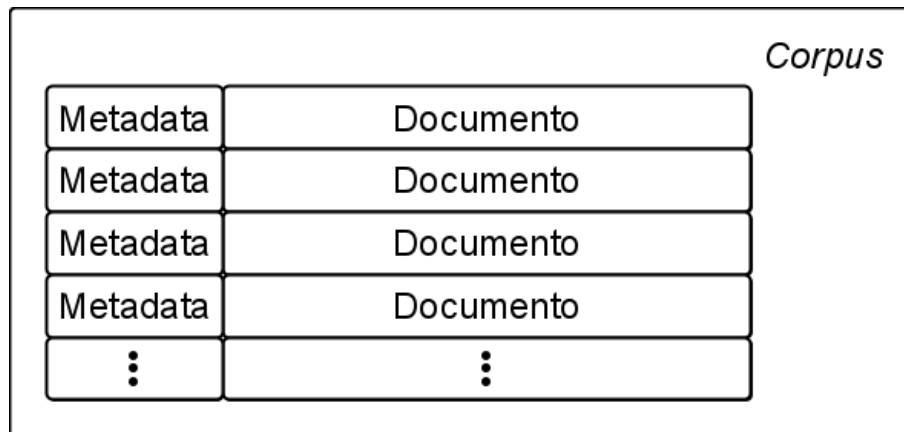
$$KER = \frac{1}{3} = 0,33 = 33\% \quad (8)$$

Observa-se que apesar de existir equívocos na transcrição proposta, de forma específica a taxa de erro de palavras-chave é menor que as outras métricas apresentadas. Nesse sentido, avaliar um SRS somente a partir das métricas anteriores pode não apresentar um resultado consistente dependendo do contexto estudado.

2.4 Processamento de Linguagem Natural (NLP)

As técnicas apresentadas pelo NLP trabalham com um conjunto de dados textual (*corpus*). Um *corpus* contém documentos e metadados associados com os documentos (genericamente chamados de *instâncias*). Os documentos são sequências de caracteres formando sentenças e o agrupamento de caracteres considerando o espaço em branco é chamado de *tokens*. Os metadados são dados auxiliares associados aos documentos. Podem ser identificadores, rótulos, horários, entre outros. A Figura 5 exhibe como o conjunto de dados é composto (MCMAHAN; RAO, 2019).

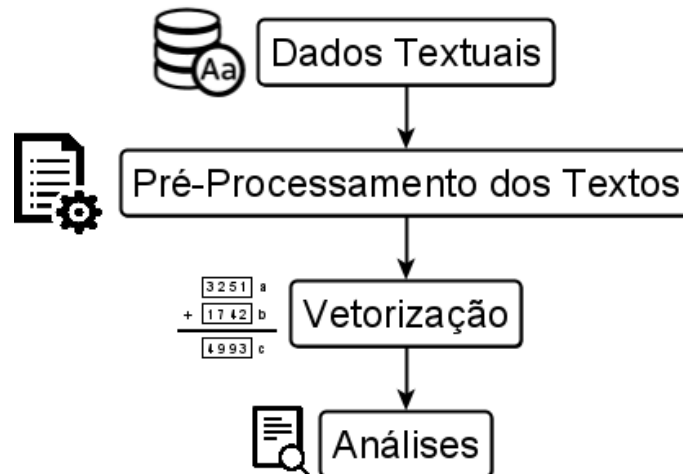
Figura 5 – Composição do conjunto de dados



Fonte: Elaboração própria (2021).

Tendo um *corpus* é possível a realização do processamento dele, para isso existem diferentes técnicas de NLP capazes de tratar os documentos para serem explorados analiticamente de forma eficiente. Após o pré-processamento dos documentos pode-se executar algum tipo de vetorização para então realizar as análises. A Figura 6 exibe o processo padrão para NLP.

Figura 6 – Processo geral relacionado a NLP



Fonte: Elaboração própria (2021).

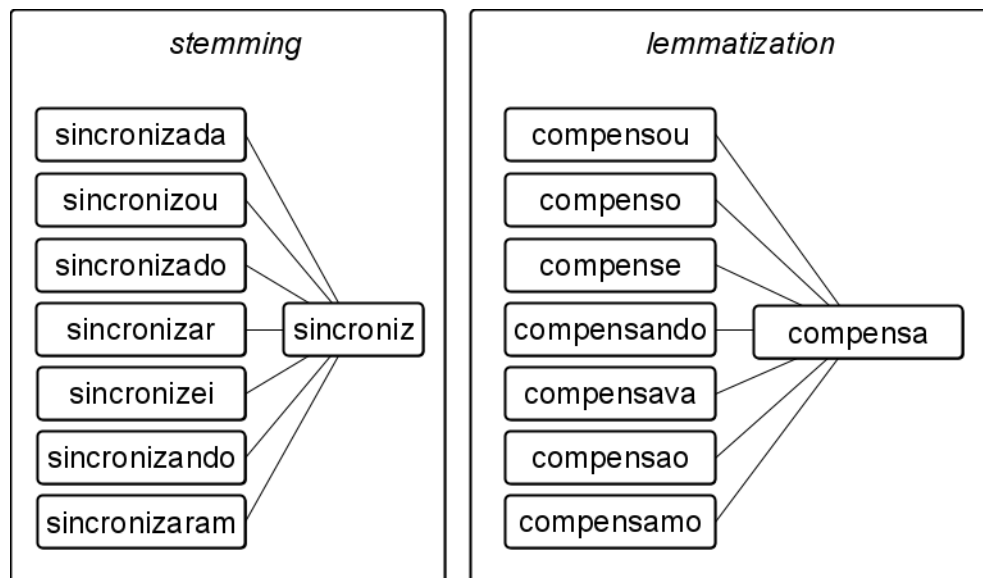
Quando se deseja analisar um *corpus* é inicialmente realizado um pré-processamento, nele é removido palavras desnecessárias, acentuações, sinais de pontuação, caracteres especiais, emoticons, entre outros. Após essa etapa ocorre a vetorização dos dados, como a máquina não tem capacidade de interpretar textos de forma direta como humanos, é necessária a conversão dos dados textuais em numéricos. Finalmente, a partir da vetorização, as máquinas passam a ser capazes de realizar análises em relação a frequência e ordem das palavras ou sentenças e

correlacionar os dados. Nessa etapa, é recomendada a padronização dos dados, para isso, pode-se explorar técnicas de normalização (JOSHI, 2017).

2.4.1 Processamento de Texto

De forma padrão, no processamento de texto, inicialmente realiza-se a conversão de todo *corpus* para minúsculo, então remove-se: acentuações, pontuações, palavras que não agregam significado a amostra (*stop words*), diminutivos (“inha” e “inho”), plural (“s”), palavras menores que três caracteres (exceto “kv” e “mw”), nomes próprios e nomes de agentes. Após a limpeza dos *tokens* realiza-se a correção de palavras transcritas equivocadamente. Outra etapa importante é a de flexionar palavras para o seu lexema (*lemmatization*) ou raiz (*stemming*), a Figura 7 exibe um exemplo para cada tipo de flexão (MCMAHAN; RAO, 2019; LANE; HOWARD; HAPKE, 2019).

Figura 7 – Exemplo de *stemming* e *lemmatization*



Fonte: Elaboração própria (2021).

Observando as técnicas *lemmatization* e *stemming*, nota-se que o *stemming* é a flexão da palavra para seu radical, que não precisa ser idêntico à raiz morfológica da palavra. Por outro lado, a realização de *lemmatization* é a flexão da palavra para a forma de dicionário de uma palavra, que deve ser uma palavra válida (STEVEN BIRD; LOPER, 2009).

2.4.2 Relevância das Palavras

Após a aplicação das técnicas *lemmatization* e *stemming* ocorre a vetorização dos dados. Para isso é necessária a conversão dos documentos de texto para uma matriz numérica de frequência das palavras, chamada de *bag of words*. Para a realização dessa tarefa o *scikit-learn* disponibiliza a função `CountVectorizer`. Após essa conversão, os dados estão aptos para o treinamento dos modelos de ML. Entretanto, esse método sozinho acaba ignorando palavras com menor ocorrência que poderiam melhorar o processo de treino dos modelos.

Nesse sentido é recomendado o uso da medida estatística Frequência do Termo (*Term Frequency* - TF) juntamente com o Inverso da Frequência nos Documentos (*Inverse Document Frequency* - IDF).

A medida TF é a frequência do termo dividido pelo número de termos do documento. Define-se $f_{p,d}$ como a frequência (número de ocorrências) do termo (palavra) p no documento d . Então TF pode ser descrito pela seguinte equação:

$$TF(p, d) = \frac{f_{p,d}}{\sum_{k \in d} f_{k,d}} \quad (9)$$

Por outro lado, o IDF penaliza termos comuns e recompensa termos raros. A IDF de um termo (palavra) p é definida em relação a um *corpus* D com documentos d e pode ser definida como:

$$IDF(p, C) = \log \frac{N}{|p \in D : p \in d|} \quad (10)$$

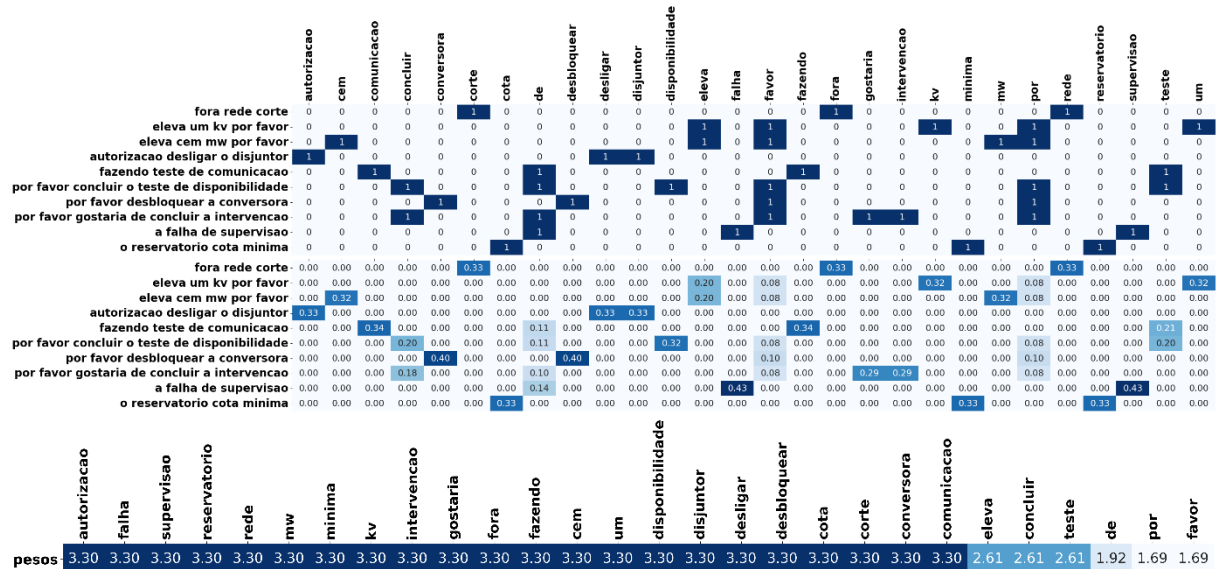
Em que, N representa o número total de documentos no *corpus*. E $|p \in D : p \in d|$ é o número de documentos em que o termo p aparece.

Dessa forma, cria-se a métrica TF-IDF é simplesmente TF multiplicado por IDF, sendo possível considerar o peso geral da palavra em relação ao documento como um todo. Os termos com os maiores valores de *TF-IDF* são os que melhor caracterizam o tópico do documento (LESKOVEC; RAJARAMAN; ULLMAN, 2010).

A função `TfidfVectorizer` (`CountVectorizer` + `TfidfTransformer`) do *scikit-learn* executa essa tarefa a partir da matriz numérica de frequência das palavras. (GÉRON, 2017). A Figura 8 exibe o processo de vetorização de um conjunto de dez

amostras hipotéticas, usando as funções do *scikit-learn*, realizando a normalização dos valores (soma de cada linha é um).

Figura 8 – Processo de vetorização hipotético

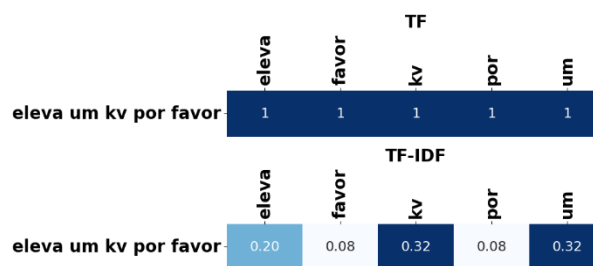


Fonte: Elaboração própria (2021).

Observado os valores resultantes da vetorização, nota-se que palavras que repetem em vários documentos possuem menores valores, nesse sentido ocorre a valorização de palavras que são ditas em contextos específicos e a desvalorização de palavras ditas com grande frequência. As palavras “de”, “por” e “favor” são as que detêm os menores valores, o que faz sentido visto que se repetem nas amostras e sozinhas não carregam nenhuma informação relevante.

Tomando como exemplo a amostra “eleva um KV por favor”, pode-se observar que os maiores valores estão nas palavras “eleva”, “um” e “KV”, fato que valida essa técnica de vetorização pois essas palavras são bem representativas, ou seja, as mesmas são importantes para a transmissão da mensagem. A Figura 9 exhibe em detalhe os valores da amostra em questão.

Figura 9 – Detalhe do processo de vetorização.



Fonte: Elaboração própria (2021).

Observa-se que os valores de TF-IDF são parametrizados de forma que os valores variam de 0 a 1, ou seja, é efetuada uma normalização unitária podendo ser do tipo L1 (mínimo desvio absoluto) ou L2 (mínimos quadrados) (JOSHI, 2017).

Importante destacar que dependendo do caso, palavras que se repetem muito podem ser importantes, é válido verificar se somente a matriz numérica de frequência das palavras (desconsiderando o TF-IDF) é mais atrativo para a análise. Nesse sentido, o modelo é treinado duas vezes (somente com TF e outra com TF-IDF) e verifica-se qual situação apresenta melhores valores de métricas.

2.4.3 N-Grams

Outra forma de agregar mais valor aos dados é com o agrupamento de uma e duas (ou mais) palavras ao mesmo tempo, técnica chamado de N-Grams. Em outras palavras é o agrupamento de termos considerando larguras fixas (n) de termos consecutivos em um texto (MCMAHAN; RAO, 2019). A Figura 10 exibe um exemplo de aplicação da técnica considerando a largura de um a três.

Figura 10 – Exemplo de aplicação da técnica N-Grams

N=1: Unigrams	por	favor	elegar	três	kv	barra
N=2: Bigrams	por favor	favor elegar	elegar três	três kv	kv barra	
N=3: Trigrams	por favor elegar	favor elegar três	elegar três kv	três kv barra		

Fonte: Elaboração própria (2021).

Dessa forma é possível levar em consideração a sequência das palavras em conjunto a palavras únicas, agregando maior confiabilidade ao aprendizado (WANG; MANNING, 2012). O uso dessa técnica pode ser ajustada pelo argumento “*ngram_range*” da função *TfidfVectorizer* da *scikit-learn* (GÉRON, 2017).

2.5 Aprendizado de Máquina (ML)

Algoritmos de ML podem ser divididos em duas categorias: não supervisionado ou supervisionado. No aprendizado não supervisionado as amostras não são rotuladas e o modelo deve aprender de forma autônoma. Como o modelo não conta com rótulos, esse tipo de aprendizado é mais desafiador. É usado quando

deseja-se realizar agrupamento (*clustering*) de dados. Entretanto, espera-se que os dados apresentem algum nível de organização e quanto menos esparsa, melhor o resultado (BISHOP, 2006).

O aprendizado supervisionado é quando cada amostra é rotulada e se tem como objetivo prever futuras respostas (predição) ou entender a relação entre a amostra e seu rótulo (inferência). Existem vários métodos que se especializaram nesse tipo de aprendizado, como exemplo é possível citar Regressão Linear (*Linear Regression*), Regressão Logística (*Logistic Regression*), Modelo Aditivo Generalizado (*Generalized Additive Model*), *Boosting* e Máquina de Vetores de Suporte (*Support Vector Machines*). Esse tipo de aprendizado é usado basicamente para regressão (*regression*) ou classificação (*classification*) (JAMES *et al.*, 2000).

Para o presente estudo, no contexto de análise de dados textuais relacionados a operação do setor elétrico, será adotado o escopo de classificação multiclasse (aprendizado supervisionado). A Tabela 4 exibe os modelos explorados para a realização dos estudos relacionados a ML neste trabalho (SCIKIT-LEARN, 2021; JOSHI, 2017).

Tabela 4 – Modelos de ML utilizados para classificação

Classificação	Abreviação
Complement Naive Bayes	ComplementNB
Linear Support Vector	LinearSVC
Stochastic Gradient Descent	SGDClassifier
K-Nearest Neighbors	KNeighborsClassifier
Multi-layer Perceptron	MLPClassifier
Random Forest Classifier	RandomForestClassifier

Fonte: Elaboração própria (2021).

Para a classificação da comunicação foi levantada uma lista de estimadores capazes de resolver problemas do tipo classificação multiclasse.

2.5.1 Complement Naive Bayes

O modelo *Naive Bayes* é amplamente conhecido e aceito, visto que apresenta uma solução rápida e de qualidade para problemas de classificação textual (ZHANG, 2004). Nesse sentido, após sua popularização, surgiram variações buscando resultados mais assertivos. O *ComplementNB* surge como uma melhora no

tradicional *Multinomial Naive Bayes*, buscando competir com outros métodos como o *Support Vector Machines* (SVM) (RENNIE et al., 2003).

Complement Naive Bayes corrige suposições equivocadas aceitas pelo modelo padrão Naive Bayes Classifier, já que ele considera a não uniformidade dos dados, a normalização dos pesos, entre outros. Essas considerações adicionais elevam a qualidade do modelo (RENNIE et al., 2003).

2.5.2 Linear Support Vector

Esse modelo corresponde a biblioteca LIBLINEAR (FAN et al., 2008), implementada nos termos do *scikit-learn*. É uma derivação da biblioteca LIBSVM (CHANG; LIN, 2011). LIBLINEAR, em relação a LIBSVM, possui maior flexibilidade na escolha dos parâmetros de penalidade e de função de perda. A mesma é recomendada quando se deseja resolver problemas com grande quantidade de amostras.

LinearSVC resolve um problema de otimização com diferentes funções de perda. Como o método SVM nativamente suporta somente problemas binários, é implementado a estratégia um contra o resto (*one-vs-the-rest*) para problemas multiclasse no qual é efetuada várias classificações binárias, uma para cada classe (JAMES et al., 2000).

2.5.3 Stochastic Gradient Descent (SGD)

O modelo SGD (ROBBINS; MONRO, 1951), é um eficiente método de otimização que busca ajustar o aprendizado de problemas de classificação e de regressão, como o anteriormente citado SVM e *Logistic Regression*. O SGD pode ser amplamente aplicado em diversas abordagens (como otimizador em redes neurais) e é recomendado para contextos em que existe aprendizado em grande escala, ou seja, com grande número de amostras (BOTTOU; BOUSQUET, 2007).

O SGDClassifier é um classificador linear (semelhante ao LinearSVC) otimizado pelo método SGD que também segue a estratégia um contra o resto. Destaca-se que o modelo bem abrangente, em comparação com outros estimadores, pois permite a edição de mais parâmetros, tornando-o mais complexo de usar.

Dependendo da escolha da função de perda pode ser implementado: *Linear SVM*, *Logistic Regression* ou *Probabilistic Logistic Regression*.

2.5.4 K-Nearest Neighbors

O `KNeighborsClassifier` é diferente dos outros modelos pois não busca construir um modelo interno geral, o mesmo simplesmente guarda instâncias dos dados de treino e os dados de entrada são classificados de acordo com o número de vizinhos. Comparando esse estimador com o SVM, se os dados de treino são muito superiores ao número de atributos (*features*), `KNeighborsClassifier` pode ser considerado melhor que o SVM (TAUNK et al., 2019).

2.5.5 Multi-layer Perceptron

`MLPClassifier` é um algoritmo de treinamento supervisionado capaz de aprender baseado no número de dimensões do *input* e do *output* e o treino ocorre usando *backpropagation*. Dado um conjunto de atributos (*features*), ocorre o aprendizado por meio de um aproximador de função não linear para classificação ou regressão. Entre a camada de entrada e a de saída, pode haver uma ou mais camadas não lineares, chamadas de camadas ocultas (WAN, 1993).

Perceptron recebe *inputs*, multiplica os mesmos por pesos e então passa por uma função de ativação (*logistic*, *relu*, *tanh*, *identity*) para produzir um *output*. O `MLPClassifier` treina iterativamente já que cada passo de tempo derivadas parciais da função de perda em relação aos parâmetros do modelo são calculadas para atualizar os parâmetros (MURTAGH, 1991).

2.5.6 Random Forest Classifier

O `RandomForestClassifier` é um método baseado em agrupamento (*ensemble*), no qual adequa (*fits*) uma coleção de classificadores de árvore de decisão e usa a média para melhorar a predição preditiva e controlar o sobreajuste (*overfitting*) (BREIMAN, 2001). Por padrão, todo o conjunto de dados é usado para construir cada

árvore. Os métodos *ensemble* são frequentemente muito mais precisos do que os classificadores individuais que os compõem (DIETTERICH, 2000).

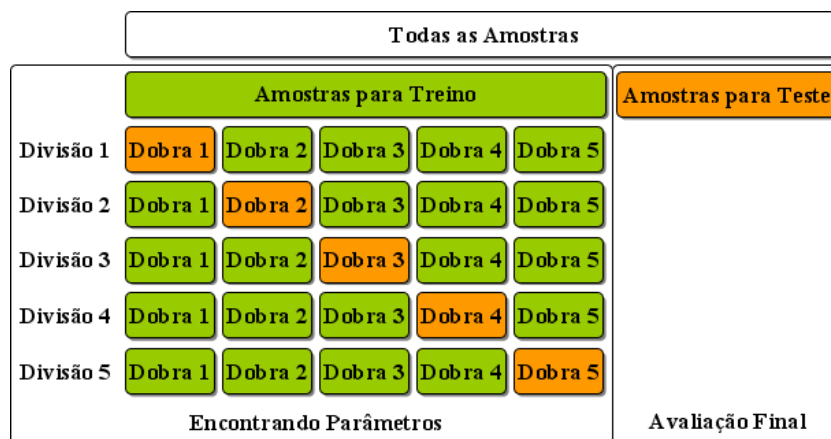
2.5.7 Divisão dos Dados e Validação Cruzada

Para a realização de estudos relacionados com ML (aprendizado supervisionado) é necessária a realização da divisão (normalmente randômica) do *dataset* para duas finalidades: treinamento e validação. A escolha da proporção da divisão do *dataset* é realizada pelo analisador, usualmente as seguintes proporções para treino e teste são adotadas: 50:50, 70:30, 67:33 ou 80:20 (IZBICKI; DOS SANTOS, 2020).

Entretanto, quando é necessário o trabalho com *datasets* limitados, para construir modelos com boa performance, deseja-se usar o máximo de amostras possíveis para treino. Nesse sentido, caso o conjunto de validação for pequeno (o presente estudo se encaixa nessa condição, somente 1500 amostras rotuladas), resultará em uma estimativa ruidosa de desempenho preditivo. Para solucionar isso, é possível explorar a metodologia de validação cruzada (*cross validation*) com os métodos Kfolds, ShuffleSplit ou StratifiedShuffleSplit, StratifiedKfolds (BISHOP, 2006; GÉRON, 2017).

Para exemplificar a metodologia usada para divisão dos dados observe a Figura 11, nela é exibida a divisão dos dados em teste e treinamento e a subdivisão do treinamento para a realização da validação cruzada.

Figura 11 – Divisão dos dados



Fonte: Elaboração própria (2021).

Na figura é explorado uso da validação cruzada KFold, em que os dados são divididos em dobras (*folds*) e, para cada interação, uma dobra é usada como teste e o restante é usada como treino. Quando se trata de *dataset* desbalanceado recomenda-se usar StratifiedKfolds pois dessa forma cada *fold* possui a mesma quantidade de amostras de cada classe. A validação cruzada é interessante para a definição dos hiperparâmetros dos estimadores e a extração de características do texto (BOUCKAERT; FRANK, 2004).

2.5.8 Métricas de Desempenho para ML

Após o aprendizado do modelo, é necessária a avaliação do mesmo para validar se o aprendizado foi realizado com sucesso, as métricas mais indicadas pela literatura são: Matriz de Confusão, Acurácia, Precisão, Sensibilidade, F1 Score, Curva de Aprendizado e *Matthews Correlation Coefficient* (MCC) (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006):

2.5.8.1 Matriz de Confusão (*Confusion Matrix*)

A matriz de confusão representa resumo geral das classificações, podendo ser observada para cada classe quantas amostras foram corretamente classificadas e quantas foram equivocadamente classificadas (TING, 2011). Observe na Figura 12 a matriz de confusão para um problema hipotético com duas classes (binário).

Figura 12 – Matriz de confusão para um problema binário

		Predição	
		P	N
Real	P	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	N	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Elaboração própria (2021).

Para exemplificar o cálculo das métricas pode-se observar a matriz de confusão para um problema binário. Para o presente estudo será realizada uma classificação multiclasse, ou seja, existem mais de duas classes a serem estimadas.

Quando se avalia problemas multiclasse, existe um valor de métrica para cada classe, olha-se para uma única classe e o restante é avaliado como outra. Nesse sentido, para avaliar um estimados de maneira geral é realizado um cálculo global ou uma média (normal ou ponderada) das métricas de cada classe individualmente.

2.5.8.2 Acurácia (Accuracy)

Acurácia (11) é fração de amostras classificadas corretamente.

$$\text{Acurácia} = \frac{VP + VN}{VN + VP + FN + FP} \quad ((11))$$

2.5.8.3 Precisão (Precision)

Precisão (12) responde o questionamento: do total de amostras classificadas para uma determinada classe, quantas estão corretas?

$$\text{Precisão} = \frac{VP}{VP + FP} \quad ((12))$$

2.5.8.4 Sensibilidade (Recall):

Sensibilidade (13) responde à pergunta: do total de amostras de uma determinada classe, quantas foram classificadas corretamente?

$$\text{Revocação} = \frac{VP}{VP + FN} \quad (13)$$

2.5.8.5 F1 Score

F1 Score (14) é interpretada como média harmônica entre precisão e recall.

$$\text{F1 Score} = \frac{2 \times \text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (14)$$

2.5.8.6 Curva de Aprendizado

A curva de aprendizado é um gráfico que apresenta a performance do modelo em relação a quantidade de amostras. É um bom indicativo para a estabilização do modelo em relação ao tamanho do *dataset*, com essa métrica é possível avaliar se o modelo extraiu completamente o aprendizado das características das amostras e se com a adição de mais dados existe a possibilidade melhora na performance. (ANZANELLO; FOGLIATTO, 2011).

2.5.8.7 Matthews Correlation Coefficient (MCC)

MCC (15) é um valor que varia entre -1 e 1, o limite superior representa uma classificação perfeita, o valor 0 representa uma classificação randômica e o limite inferior indica alta discordância entre a predição e as verdadeiras classes (GORODKIN, 2004).

$$MCC = \frac{(VP \times VN) - (FP \times FN)}{\sqrt{(VP + FP) \times (VP + FN) \times (VP + FP) \times (VN + FN)}} \quad (15)$$

2.5.9 Sobreajuste e Subajuste

Para efetuar o aprendizado de um estimador com sucesso é necessário levar em consideração o sobreajuste (*overfitting*). De forma geral alimenta-se o modelo com uma quantidade grande o suficiente para que o modelo aprenda a generalizar o aprendizado. Se existe uma quantidade baixa de dados, o algoritmo pode apresentar sobreajuste, ou seja, ele aprende muito bem baseado nos dados de treino, mas não é capaz de generalizar o aprendizado, tendo um desempenho baixo para os dados de teste. Logo, para solucionar esse problema é possível explorar uma técnica chamada de *data splitting*, em que os dados de treino são divididos em treino e validação. Uma forma de efetuar essa separação seria por meio da técnica chamada de validação cruzada comentada anteriormente (GÉRON, 2017).

Outra dificuldade encontrada durante o aprendizado seria quando o modelo não se adapta bem aos dados com os quais foi treinado, ou seja, as características

dos dados não são suficientes para realizar o aprendizado. Esse fenômeno é denominado subajuste (*underfitting*) (IZBICKI; DOS SANTOS, 2020).

3 DESCRIÇÃO DA METODOLOGIA

A Figura 13 ilustra as principais etapas da metodologia proposta. Iniciando com a comunicação verbal da operação do setor elétrico. A partir disso é possível criar uma base de áudios e transcrevê-los com um sistema de reconhecimento de voz. As transcrições formam uma base de dados textual que pode ser analisada usando técnicas de NLP e posteriormente ML. Tendo como principal resultado a classificação da comunicação verbal da operação do setor elétrico.

Figura 13 – Metodologia para a classificação dos áudios da operação do setor elétrico



Fonte: Elaboração própria (2021).

Pode-se observar na Figura 13 que a primeira etapa da metodologia está relacionada a operação do setor elétrico, especificamente com a comunicação verbal na mesma. Nela realiza-se um estudo relacionado aos equipamentos, ações, objetos, entidades, unidades de medida e números que são abordados durante a operação.

A partir da base de áudios da operação do setor elétrico é realizado um pré-processamento nos arquivos para a transcrição usando o serviço *Wit.ai*. As principais razões estão relacionadas ao serviço não apresentar restrições de requisições por mês e ser gratuito. Adicionalmente, foram efetuados testes preliminares de desempenho nos serviços e comparando os resultados com outros serviços (LIMA; COELHO; TAKIGAWA, 2021). A comparação do desempenho dos SRS gratuitos e disponíveis do mercado também está ilustrada no Apêndice A.

Com a transcrição é possível a aplicação de técnicas de NLP. Então, realiza-se a rotulação e classificação dos dados textuais. Para a classificação, por ser

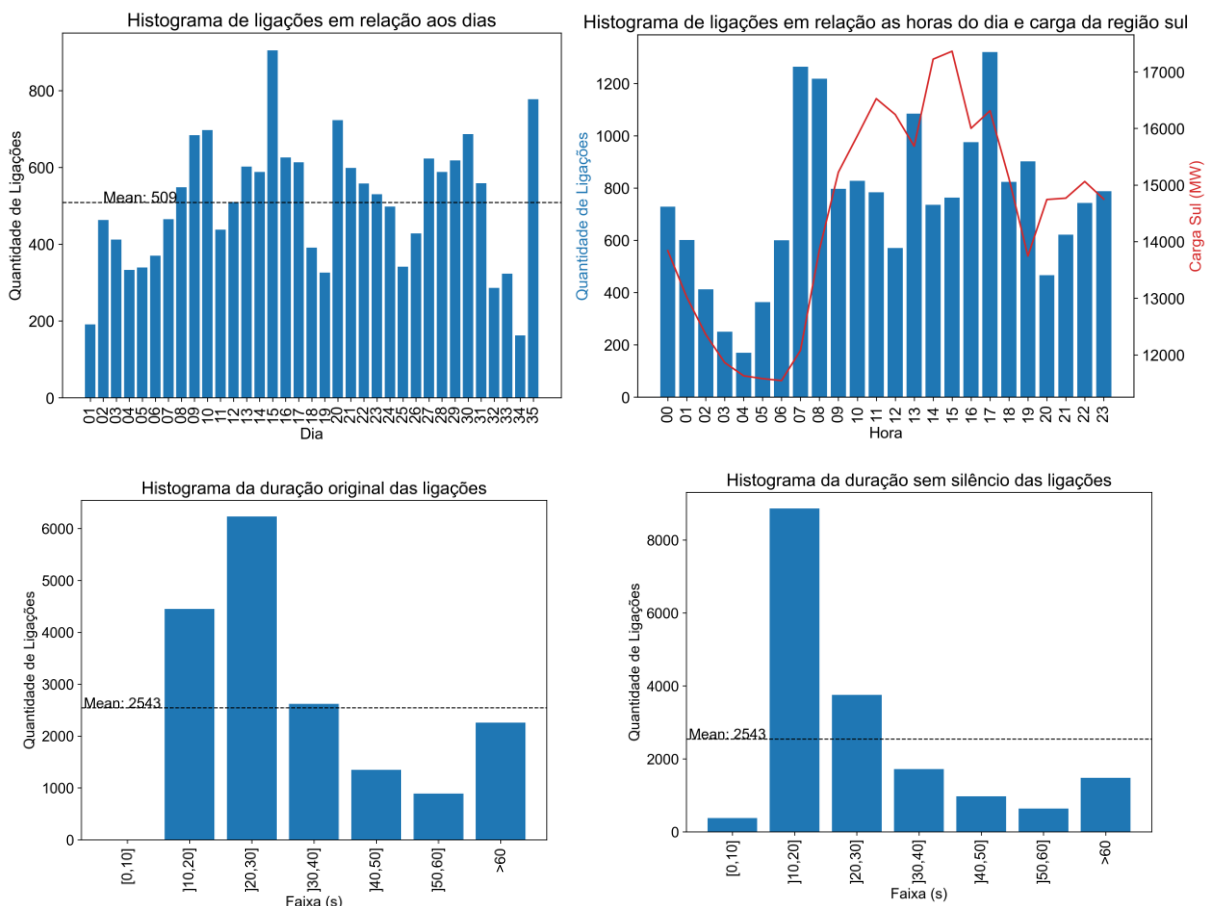
um aprendizado supervisionado, é necessário a rotulação das amostras. Nesse sentido, realiza-se uma metodologia para identificar classes existentes na comunicação verbal para a atribuição das amostras.

Nas seções seguintes, será evidenciada as características dos dados (rotulados e não rotulados), a lógica adotada para a rotulação das amostras, o pré-processamento dos áudios, o pós-processamento das transcrições e pôr fim a metodologias para a classificação das amostras.

3.1 Caracterização dos dados

O principal insumo explorado é um *dataset* de arquivos de áudio contendo a gravação de trinta e cinco dias da comunicação verbal realizada na sala de operação do Centro Regional de Operação Sul no ano de 2020. Totalizando 17.802 amostras. Histogramas relacionados às características do *dataset* são exibidos na Figura 14.

Figura 14 – Características do *dataset* explorado



Fonte: Elaboração própria (2021).

Observando as figuras relacionadas com a caracterização do *dataset* é possível observar que, em média, segundo o intervalo de dias explorado, o ONS realiza 509 ligações por dia. Em relação as horas do dia, a maior parte das ligações estão localizadas juntamente com a alta da carga da região sul. A curva de carga sobreposta com a quantidade média de ligações representa a soma dos três estados do sul do país no dia 15 de janeiro de 2020. Nota-se que a curva de carga possui três picos, um no período da manhã, outro no fim da tarde e outro por volta da meia noite. De manhã ocorre uma grande entrada de carga, logo muitas ligações sobre controle de geração e controle de tensão acontecem, adicionalmente os serviços de intervenção diários se iniciam. De tarde, na hora do almoço, a carga se reduz, entretanto volta a se elevar, demandado necessidade de controle de geração, tensão e transmissão. A partir das 17h a jornada de trabalho comercial finaliza e a carga tende a cair e os serviços de intervenção diários se encerram. Então um novo pico de carga se inicia, a carga sobe devido a rotina residencial. A partir das 22h, empresas que buscam uma redução de custos ficam ativas no período noturno (fora da ponta) e a iluminação pública é acionada, ao mesmo tempo a carga residencial cai. Por volta das 04h ocorre o aumento de carga juntamente com o aumento das ligações pois por volta das 06h a iluminação pública sai e se inicia as atividades diárias da população, comércio e indústria.

3.1.1 Rotulagem dos Dados

A partir do *dataset* principal, contendo todos os dados, foi possível realizar a rotulagem de uma parcela das amostras a fim de realizar a classificação por meio dos modelos de ML. A partir das 17.802 amostras foram realizadas 1.500 rotulações manuais, com 12 classes relacionadas aos aspectos da operação do setor elétrico, tanto no âmbito da geração, transmissão e distribuição, foram concebidas baseado na intuição dos autores, levando em consideração as atividades que o setor de pós-operação realiza. A Tabela 5 exhibe a quantidade de amostras e a descrição para cada classe.

Tabela 5 – Lista de classes e quantidade de amostras

Classe	Quantidade de Amostras	Descrição
Carga	51	Remete às ligações telefônicas relacionadas ao corte de carga, como montante cortado, motivo, hora de corte e retorno
Comprovação de Disponibilidade	52	Remete aos diálogos relacionados com a comprovação de disponibilidade ao retorno de unidades geradoras e consiste na geração plena da unidade por um período mínimo de 4 horas ininterruptas (ONS, 2019)
Controle de Geração	339	Relacionado aos assuntos de geração, requisição de alteração de estado e modulação de montantes de energia
Controle de Tensão	358	Tópico relacionado aos ajustes de tensão, acionamento de banco de capacitores ou reatores, mudança de <i>tap</i> de transformador, entre outros
Controle da Transmissão	59	Relacionado aos assuntos voltados para a transmissão, queda de torre, indisponibilidade de linha de transmissão, atuação de proteções entre outros
Conversora	54	Qualquer ligação que cita conversora. Na região sul existem cinco: Melo, Rivera, Uruguaiana, Garabi (I e II) e Acaray
Falha de Supervisão	53	Qualquer conversa que cita algum tipo de falha de supervisão
Hidrologia	70	Relacionada a dados hidrológicos reportados ao ONS, como chuva, nível de reservatório, afluência, entre outros
Horário	124	Quando existe o questionamento de horário, como a volta de um equipamento ou um desligamento
Sem Informação	118	Quando não existem dados relevantes no diálogo, ou seja, quando não se enquadra em nenhuma outra classe
Intervenção	106	Assuntos relacionados ao intervenções em equipamentos, atrelado ao Sistema de Gerenciamento de Intervenções (SGI) do ONS
Teste de Comunicação	116	Se enquadra quando ocorre o esporádico teste de comunicação entre o agente e o ONS
Total	1500	-

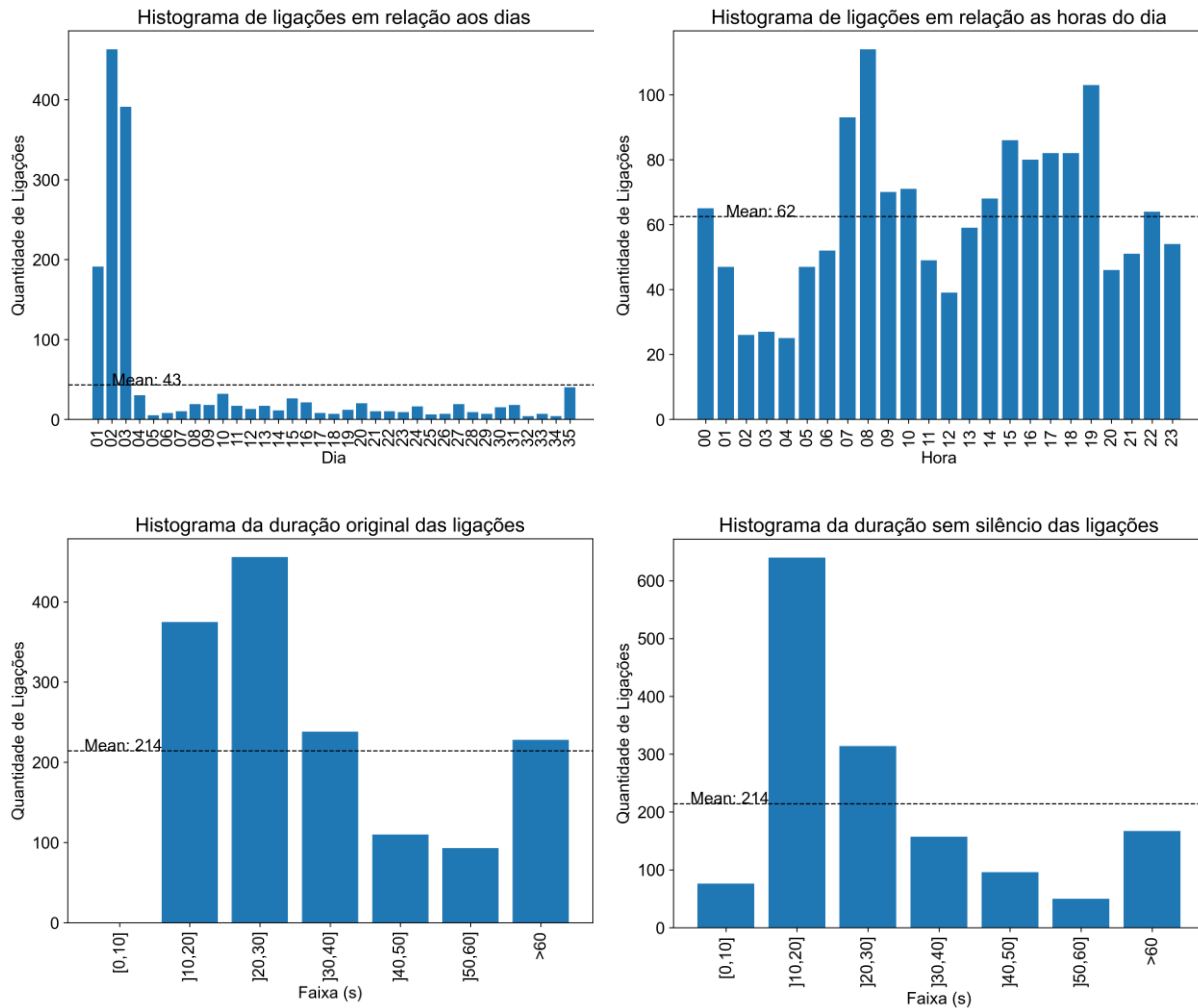
Fonte: Elaboração própria (2021).

3.1.2 Caracterização dos Dados Rotulados

Dos 1.500 arquivos de áudio que foram rotulados manualmente, a Figura 15 exibe respectivamente, a quantidade de ligações rotuladas em relação os dias, em relação as horas e em relação a faixa de duração dos áudios. É um *subset* do *dataset* completo exposto anteriormente. O valor de 1.500 amostra rotuladas foi selecionado

de forma a representar uma boa parcela dos dados, de forma geral três dias foram rotulados completamente e representam aproximadamente 8% do *dataset* inteiro.

Figura 15 – Características dos dados rotulados



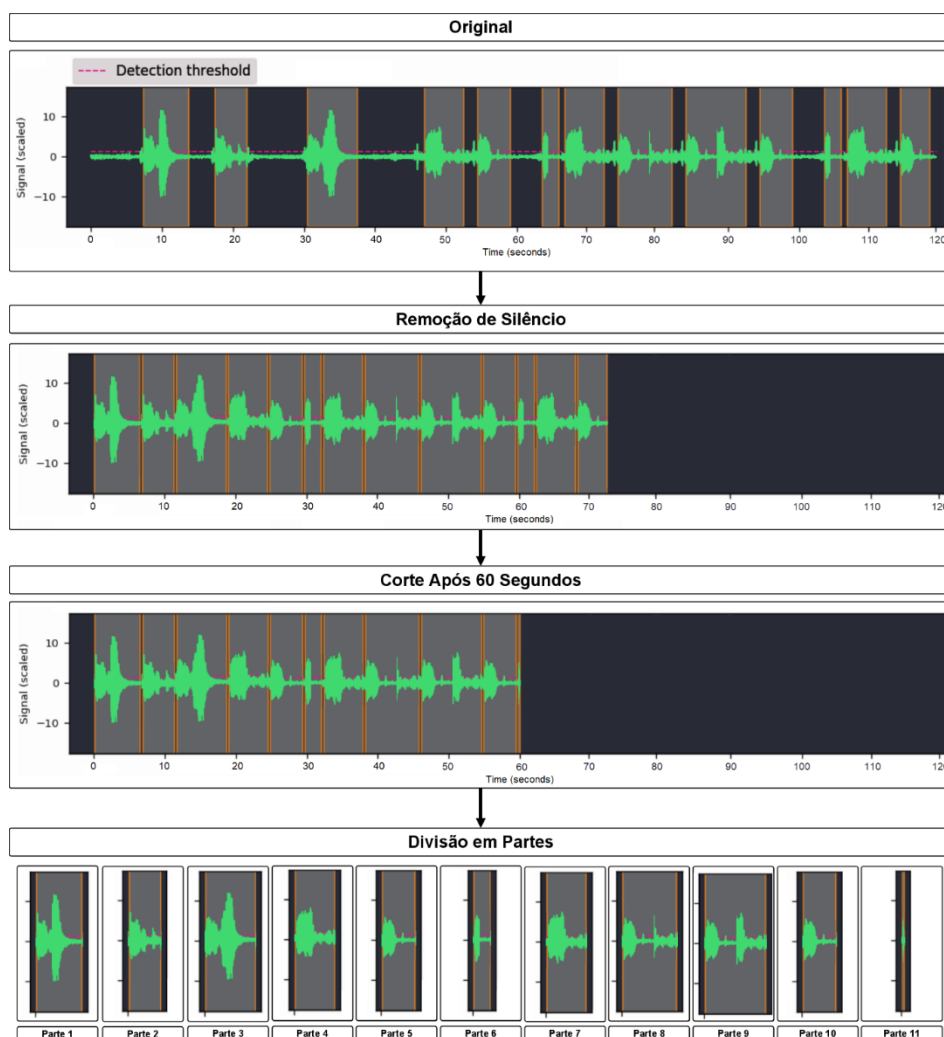
Fonte: Elaboração própria (2021).

Observando as figuras é possível notar que a maior parte das ligações rotuladas foram retiradas dos primeiros dias e as outras amostras rotuladas foram buscadas nos outros dias a fim de aumentar a quantidade de amostras das classes com menos recorrência. Pontua-se que os dados selecionados devem representar bem as amostras como um todo, ou seja, quanto maior e mais distribuídas forem as amostras rotuladas, melhor será o aprendizado.

3.2 Pré-Processamento (Áudios)

A fim de realizar a transcrição das amostras de áudio, foi efetuado um pré-processamento para tirar proveito do uso dos SRS de forma eficiente. A Figura 16 exibe o processo realizado para a transcrição de todo *dataset* disponibilizado pelo ONS. Inicialmente realiza-se a remoção de silêncio, então corta-se a amostra de áudios nos primeiros 60 segundos e finalmente segmenta-se o áudio em parte menores.

Figura 16 – Processo de tratamento do áudio para transcrição



Fonte: Elaboração própria (2021).

Nesse processo, é válido justificar dois pontos de decisão:

- A escolha do corte nos primeiros 60 segundos: considerou-se que as informações essenciais foram transmitidas no primeiro minuto da ligação, dessa maneira é possível reduzir a quantidade de

segmentações e conseqüentemente reduzir a quantidade de requisições de transcrições. Visto que, durante as oitivas efetuadas observou-se que o assunto principal é tratado logo no início do diálogo e as palavras ditas nesse intervalo são bem representativas em relação ao contexto geral da ligação.

- b) A divisão dos áudios em sub áudios: ação realizada para explorar a API do *Wit.ai* de forma livre, esse serviço de transcrição não apresenta restrições mensais e somente limita o tamanho do áudio por requisição em 20 segundos e 60 requisições por minuto. Realizando a separação de áudios grande em áudios menores é possível consumir a API com multiprocessamento sem grandes dificuldades.

Adicionalmente, após a divisão em partes, cada áudio segmentado é editado de forma que, no início e no fim do áudio é agregado um segundo de silêncio visto que, nos testes realizados, essa ação apresentou uma melhora na qualidade de transcrição dos áudios.

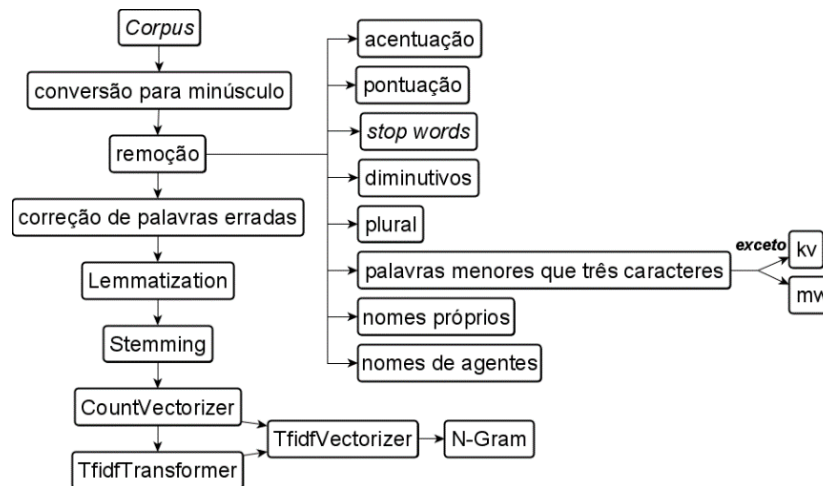
Em relação ao tempo de ligação, a média de duração das ligações originais é de 39,7 segundos enquanto a média de duração das ligações após a remoção do silêncio é de 31,6. Adicionalmente, grande parte das ligações está enquadrada na faixa de 10 a 20 segundos de duração (sem silêncio), como pode ser observado pela Figura 15. Dada a natureza do problema, existe uma quantidade considerável de silêncio nas ligações, visto que os operadores de forma corriqueira consultam informações durante as chamadas telefônicas.

Adicionalmente, foi realizada uma análise de bibliotecas e parâmetros relacionados a segmentação dos áudios buscando verificar qual a metodologia mais atrativa para processar os arquivos de áudio, essa análise pode ser observada no Apêndice B.

3.3 Pós-Processamento (Textos)

Realizadas as transcrições com o *Wit.ai*, segue-se para o processamento dos dados textuais. A Figura 17 exhibe o processo adotado para o tratamento dos textos transcritos.

Figura 17 – Processo de preparação dos dados textuais



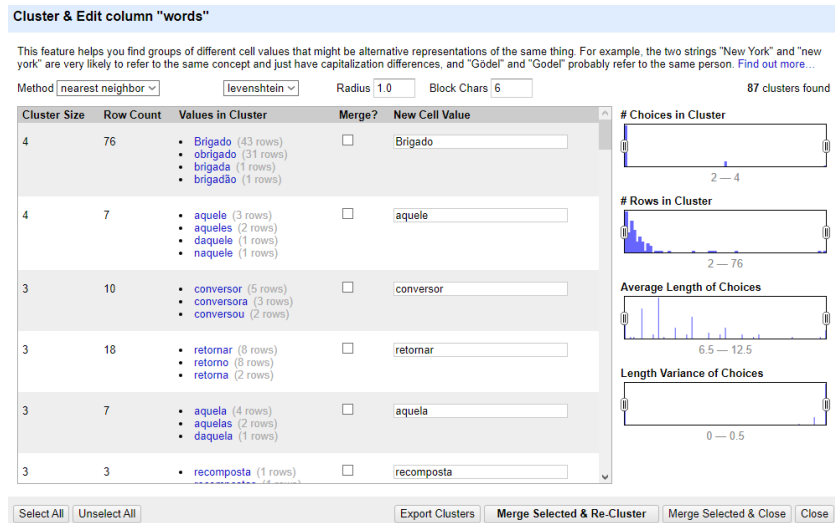
Fonte: Elaboração própria (2021).

A ação de *lemmatization* e *stemming* foram realizadas por meio da criação de um dicionário em Python, que pode ser consultado no Apêndice C. Esse dicionário foi concebido usando o *software* OpenRefine (HUYNH, 2012) e a função *most_similar* do Gensim (ŘEHŮŘEK, 2009). De forma prática, o dicionário representa uma lista de trocas.

O OpenRefine possibilita o agrupamento das palavras por similaridade e disponibiliza seis algoritmos para a realização dos *clusters*: Para a opção de *clustering*, é disponibilizado os seguintes algoritmos: *key collision* com as variações: *fingerprint*, *ngram-fingerprint*, *metaphone3*, *cologne-phonetic*, *Daitch-Mokotoff* e *Beider-Morse* ou *nearest neighbor* com as variações: *levenshtein* e *ppm* (OPEN REFINE, 2021).

A Figura 18 exibe a tela de agrupamento, nela é possível selecionar a opção da seleção do método e seus parâmetros e exibe os grupos (*clusters*) encontrados, possibilitando a combinação (*merge*) ou a exportação dos resultados no formato JSON.

Figura 18 – Tela de agrupamento do software OpenRefine



Fonte: Elaboração própria (2021).

Na abordagem com a biblioteca Gensin é efetuado o *Word Embeddings*, que representa documentos como vetores semânticos. Gensin foi desenvolvido para processar documentos de texto não estruturados, usando algoritmos de aprendizado não supervisionado como *Word2Vec*, *FastText*, *Latent Semantic Indexing*, *Latent Dirichlet Allocatio*, entre outros. Com base em um corpus (dados de texto) é possível identificar padrões nos documentos e correlacionar os dados (REHUREK; SOJKA, 2010). Para o presente estudo foi utilizado o algoritmo *Word2Vec*, que apresenta duas opções de algoritmo de treino *skip-gram* ou *continuous bag of words* (CBOW) (MIKOLOV; LE; SUTSKEVER, 2013).

3.4 Classificação

Depois de finalizar o processamento dos dados transcritos foi possível realizar o treinamento e teste dos modelos para a classificação multiclasse de dados textuais. Os cinco modelos selecionados foram:

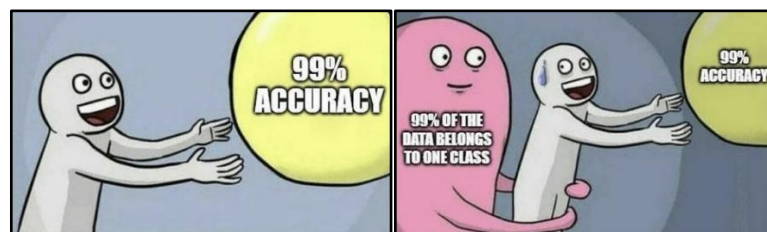
- Complement Naive Bayes (ComplementNB);
- Linear Support Vector Classification (LinearSVC);
- Stochastic Gradient Descent Classifier (SGDClassifier);
- K-Nearest Neighbors Classifier (KNeighborsClassifier);
- Multi-Layer Perceptron Classifier (MLPClassifier);

f) Random Forest Classifier (RandomForestClassifier).

Inicialmente efetua-se a busca pelos melhores parâmetros dos modelos usando o StratifiedKfolds junto com o GridSearchCV. Uma busca exaustiva é realizada para encontrar a melhor combinação de parâmetros para cada modelo usando a pontuação média do teste das dobras como métrica (BOUCKAERT; FRANK, 2004).

A métrica considerada para pontuação é o F1 ponderado (*weighted*), usando o peso de cada classe para levar em conta o desbalanceamento. O F1 *weighted* foi usado pois o mesmo consegue avaliar a classificação do modelo como um todo, mesmo com dados desbalanceados, diferentemente da acurácia. A Figura 19 exhibe a razão para que a *accuracy* não seja uma métrica representativa quando se trata de dados desbalanceados.

Figura 19 – Imagem de humor sobre dados desbalanceados

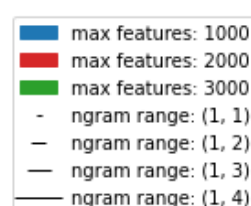


Fonte: (VALKLING, 2020)

Por outro lado, uma extração de características de texto foi realizada, principalmente, para descobrir qual combinação de tamanho de vocabulário (considerando as palavras mais frequentes) e intervalo de n-gramas traz o modelo mais interessante.

A Figura 21 mostra os valores médios ponderados de F1 para juntamente com um parâmetro relacionado ao modelo no eixo x a realização da extração das *features* de texto (por cor indicando faixa de n-grama e forma relacionada ao tamanho de vocabulário), conforme indicado pela Figura 20.

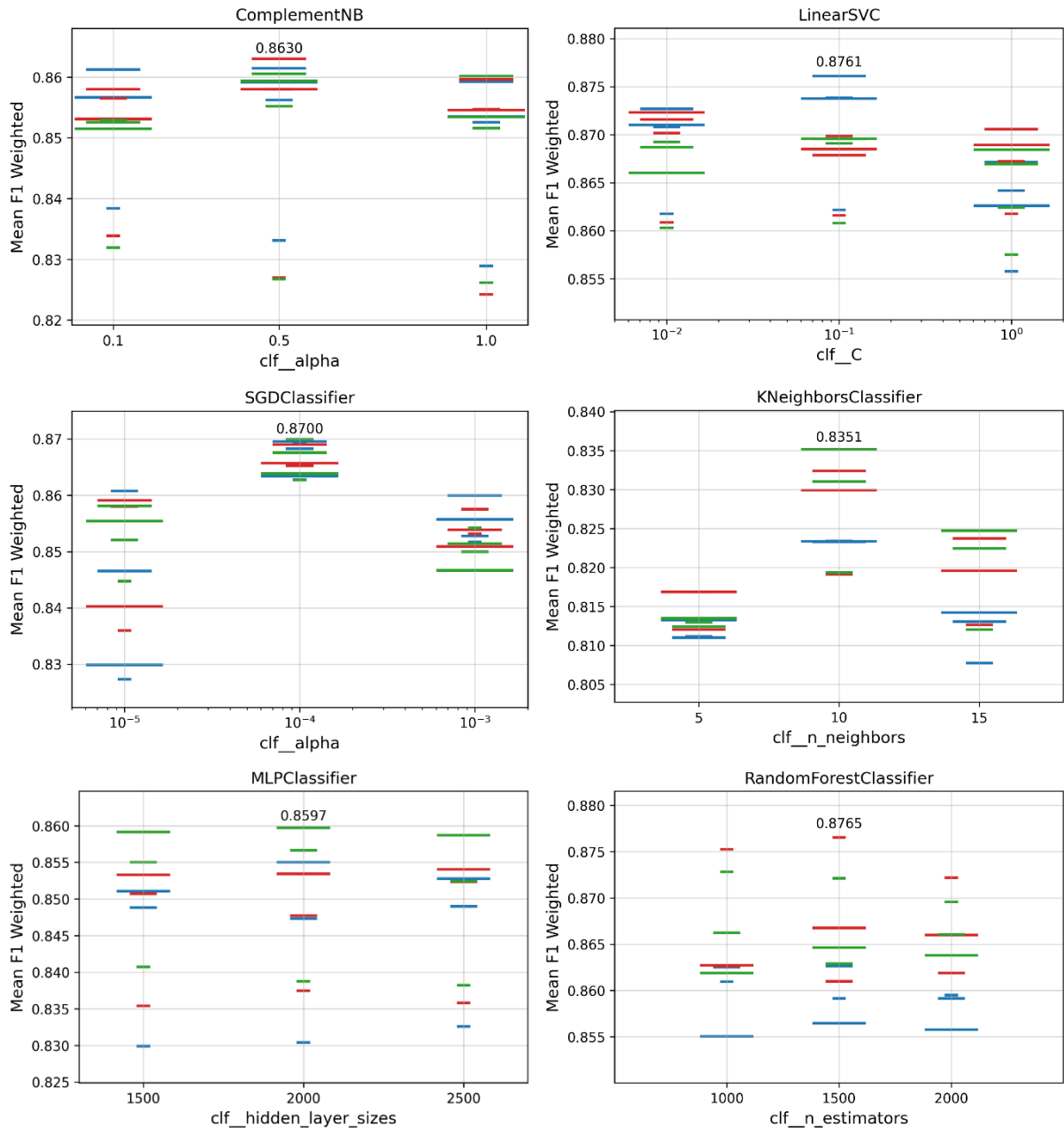
Figura 20 – Legenda de extração de características



Fonte: Elaboração própria (2021).

A extração da melhor combinação de parâmetros dos modelos é exibida pela Figura 21. Para cada modelo, o eixo x representa um parâmetro importante e varia de acordo com o tipo de modelo.

Figura 21 – Extração de características para cada modelo



Fonte: Elaboração própria (2021).

Observando a figura acima é possível concluir, em relação a extração das *features* relacionadas ao texto, que o melhor intervalo de *n-gram* para a metade dos modelos é de (1, 3), RandomForestClassifier com (1, 1), KNeighborsClassifier com (1, 4) e SGDClassifier com (1, 2).

Em relação ao tamanho do vocabulário, considerando as palavras ordenadas por frequência no *corpus*, um modelo teve melhor desempenho com 1000 (LinearSVC), dois com 2000 (ComplementNB e RandomForestClassifier) e três com 3000 (SGDClassifier, KNeighborsClassifier e MLPClassifier).

A configuração dos estimadores ocorreu seguindo o melhor valor de pontuação média do teste, buscando os melhores parâmetros relacionados ao recurso de texto e outros parâmetros específicos de cada modelo (Alfa, C, Número de Vizinhos, Tamanhos da Camada Oculta e Número de Estimadores).

A Tabela 6 mostra os melhores resultados de cada estimador, é evidenciada a média do F1 ponderado das dobras de teste, os três primeiros são RandomForestClassifier, LinearSVC e SGDClassifier, consecutivamente. No entanto, todos os modelos tiveram um desempenho semelhante, com o KNeighborsClassifier em último para esta análise.

Tabela 6 – Valores médios do F1 ponderado das dobras de teste

Modelo	Média do F1 Ponderado
ComplementNB	0,8630
LinearSVC	0,8761
SGDClassifier	0,8700
KNeighborsClassifier	0,8351
MLPClassifier	0,8597
RandomForestClassifier	0,8765

Fonte: Elaboração própria (2021).

O tamanho do vocabulário dependendo da escolha do parâmetro *ngram_range*. A Tabela 7 exhibe a dimensão da matriz numérica que os modelos usam para treino e teste variando a escolha do alcance do *ngram*.

Tabela 7 – Quantidade de colunas da matriz numérica para cada alcance de *ngram*.

<i>ngram_range</i>	Dimensão dos dados vetorizados
(1,1)	3261
(1,2)	20798
(1,3)	44310
(1,4)	68706

Fonte: Elaboração própria (2021).

Observa-se que os dados são esparsos, ou seja, possuem uma grande quantidade de elementos com valor zero. Limitando o tamanho do vocabulário (por

meio do parâmetro *max_features*) buscou-se reduzir a esparsidade, pois quanto maior o número de dados para computar maior a demanda computacional e maior o tempo para treino e busca entre os parâmetros. Interessante notar que cada modelo aprende melhor usando diferentes valores de alcance do *ngram* e diferentes tamanhos de vocabulário. A Tabela 8 exibe os melhores argumentos usados para cada modelo.

Tabela 8 – Melhores parâmetros usados para a criação dos modelos

	Função Classificadora	CountVectorizer	TfidfTransformer
a)	alpha: 0.5	binary:True max_features:2000 ngram_range:(1, 3)	sublinear_tf:True use_idf:False
b)	C: 0.1 class_weight: balanced max_iter: 10	binary:True max_features:1000 ngram_range:(1, 3)	sublinear_tf:True
c)	class_weight: balanced loss: log max_iter:5 penalty:l1	binary:True max_features:3000 ngram_range:(1, 2)	smooth_idf:False
d)	n_neighbors:10 weights:distance	binary:True max_features:3000 ngram_range:(1, 4)	smooth_idf:True sublinear_tf:True
e)	hidden_layer_sizes:(2000,) activation:identity alpha:0.1 max_iter:1000 solver:lbfgs	binary:True max_features:3000 ngram_range:(1, 3)	norm:l1
f)	n_estimators:1500 class_weight:balanced_subsample min_samples_split:25 max_depth:50 oob_score:True	max_features:2000 ngram_range:(1, 1)	smooth_idf:False sublinear_tf:True use_idf:False

Fonte: Elaboração própria (2021).

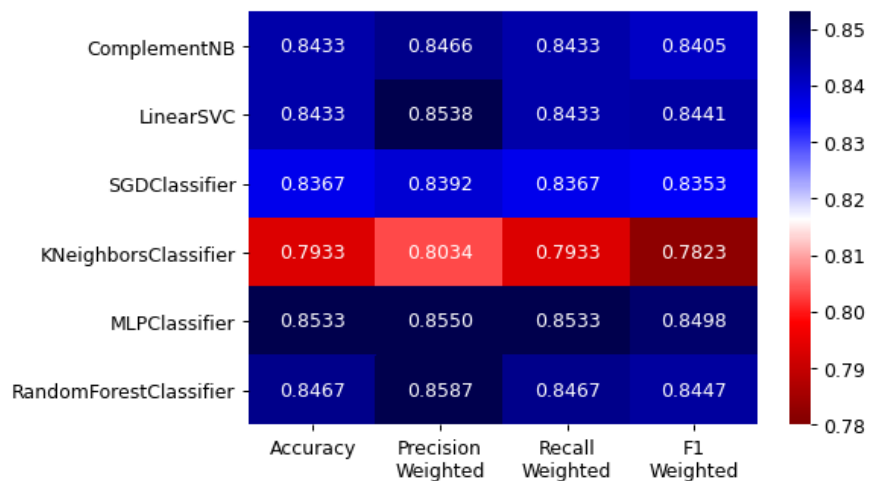
3.4.1 Resultados

Após a seleção dos melhores parâmetros é possível calcular as métricas para cada modelo de ML. Neste sentido, os resultados estão separados nas métricas gerais e, posteriormente, são ilustradas a curva de aprendizado, a matriz de confusão e a métrica MCC.

3.4.1.1 Métricas Gerais

Para evidenciar o aprendizado de cada modelo é interessante exibir as métricas: acurácia, precisão, sensibilidade e F1 (considerado a média ponderada de cada classe). A Figura 22 sumariza o valor das métricas para cada modelo para os dados de teste.

Figura 22 – Resultados do desempenho das amostras de teste

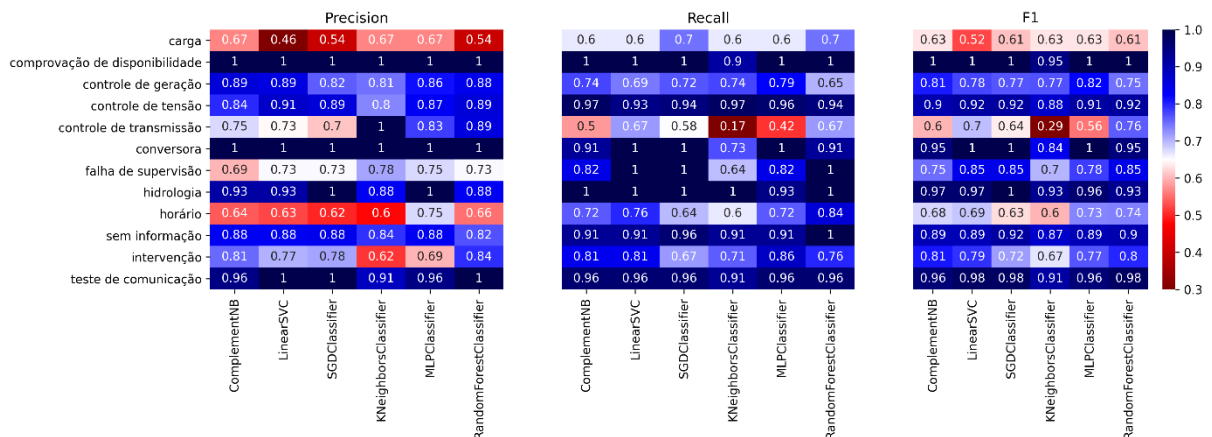


Fonte: Elaboração própria (2021).

Observando a figura é possível concluir que o modelo que apresentou a melhor performance foi o MLPClassifier (seguido pelo RandomForestClassifier) e o que apresentou o pior desempenho foi o modelo KNeighborsClassifier. Entretanto, em geral, os modelos apresentaram um desempenho satisfatório e similar, próximo de 80% de acerto geral e F1 ponderado.

Outra forma de exibir os resultados seria destacando o desempenho individual de cada classe e para cada modelo. Nesse sentido, por meio da função “*classification_report*” do *scikit-learn* a Figura 23 pode ser criada, em que no eixo horizontal são exibidos os modelos e no eixo vertical as classes.

Figura 23 – Resultados do desempenho das amostras de teste por classe



Fonte: Elaboração própria (2021).

Observando as métricas para cada classe e modelo é possível verificar que existem classes que apresentam maior dificuldade para a classificação, como “controle da transmissão”, “carga” e “horário”; e, as que apresentam maior acerto como: “comprovação de disponibilidade”, “hidrologia”, “teste de comunicação” e “conversora”.

Pode-se justificar o fato de as classes mais acertadas serem: “comprovação de disponibilidade”, “hidrologia”, “teste de comunicação” e “conversora”, pelo fato de que para essas classes existem palavras que são representativas, ou seja somente aparecem quando se trata da classe em questão, como “comprovação” para comprovação de disponibilidade ou “milímetro” para hidrologia.

Por outro lado, as classes que são mais problemáticas possuem compartilhamento de vocabulário, dificultando a classificação automatizada. Por exemplo, palavras que são ditas para controle de transmissão podem ser ditas quando se trata da carga. Adicionalmente, existe a possibilidade de existir mais de um assunto tratado por amostra, por exemplo o operador comunica um corte de carga e, ao mesmo tempo, reporta uma manobra realizada em uma linha de transmissão.

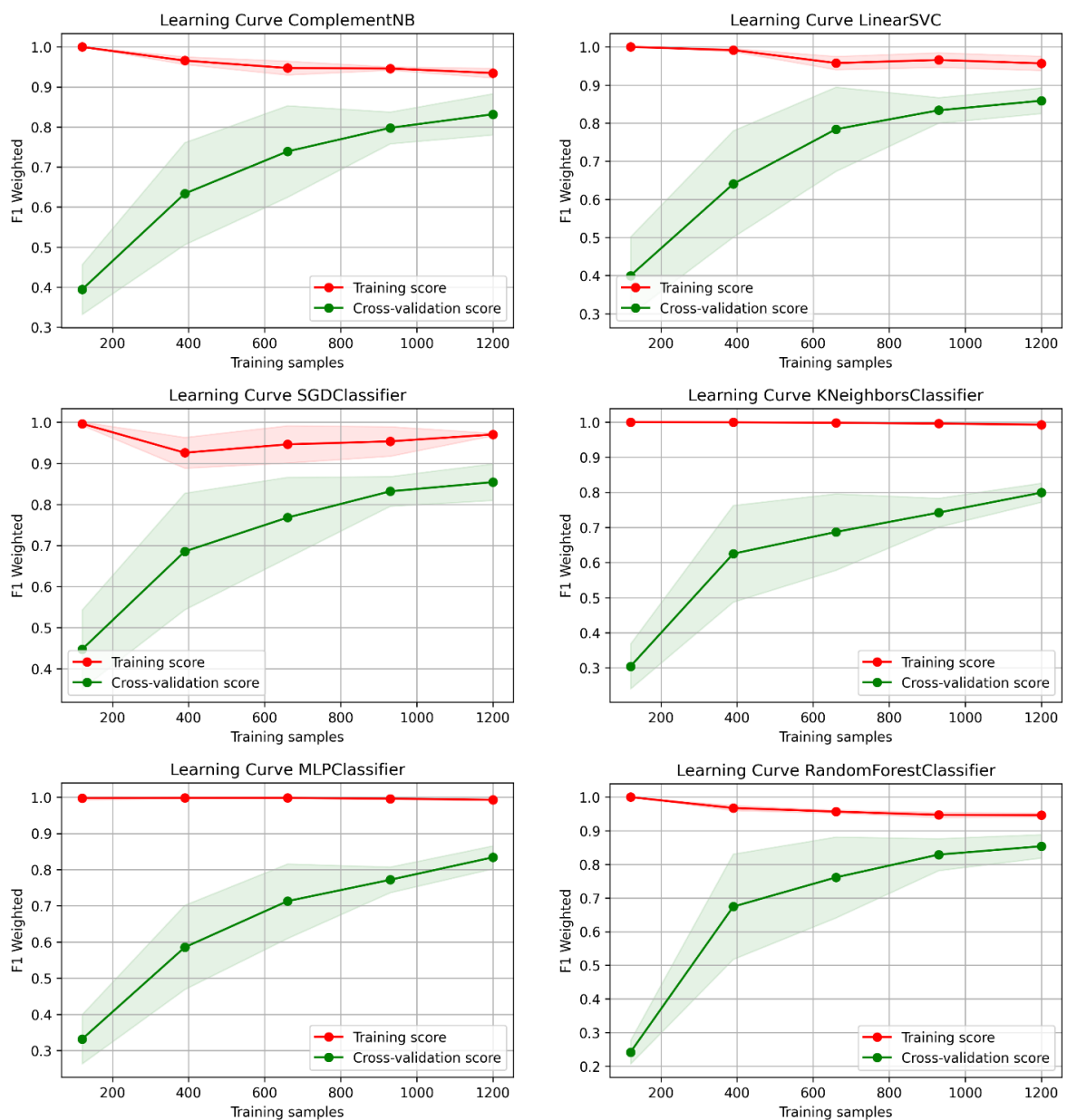
Outra dificuldade seria em relação a natureza do problema, que é desbalanceada. Das 1500 amostras rotuladas, 1/5 são separadas para o teste final dos modelos, logo, algumas classes são menos representativas (como pode ser observado pela Tabela 5): carga, comprovação de disponibilidade, controle de

transmissão possuem menos de 60 amostras, resultando em somente 12 amostras para avaliação dos modelos.

3.4.1.2 Curva de Aprendizado

Uma forma de evidenciar a maturidade dos modelos criados seria pelo uso da curva de aprendizado. Pode ser efetuado o treino e avaliação progressiva dos dados a fim de observar a evolução do modelo em relação a quantidade de amostras usadas. A Figura 24 exibe os gráficos de curva de aprendizado para cada modelo.

Figura 24 – Curva de aprendizado dos modelos



Fonte: Elaboração própria (2021).

Observando as curvas de aprendizado, pode-se apontar que nenhum modelo alcançou estabilidade, ou seja, a quantidade de amostras usadas se mostra insuficiente. Principalmente, busca-se que a curva de teste (*Cross-validation score*) se apresente o maior possível e com pouca evolução. Nesse sentido, recomenda-se a expansão do *dataset* buscando a estabilidade do aprendizado.

Em relação ao *training score* é interessante que não ocorra grandes alterações, caso o mesmo permaneça no máximo é sinal que pode existir *overfitting*, ou seja, o modelo se torna muito especialista e não é capaz de generalizar o aprendizado, isso ocorreu para algumas classes e alguns modelos, entretanto não é um problema pois já era um comportamento esperado visto que existe palavras específicas que representam bem algumas classes. Caso o mesmo reduza muito é sinal de *underfitting*, ou seja, o modelo não está aprendendo com sucesso com base nas *features* em questão. Esse problema não foi notado em nenhum modelo treinado pois o *training score* se manteve acima de 0,9 durante todas as curvas, valor considerado alto e suficiente para descartar a existência do *underfitting*.

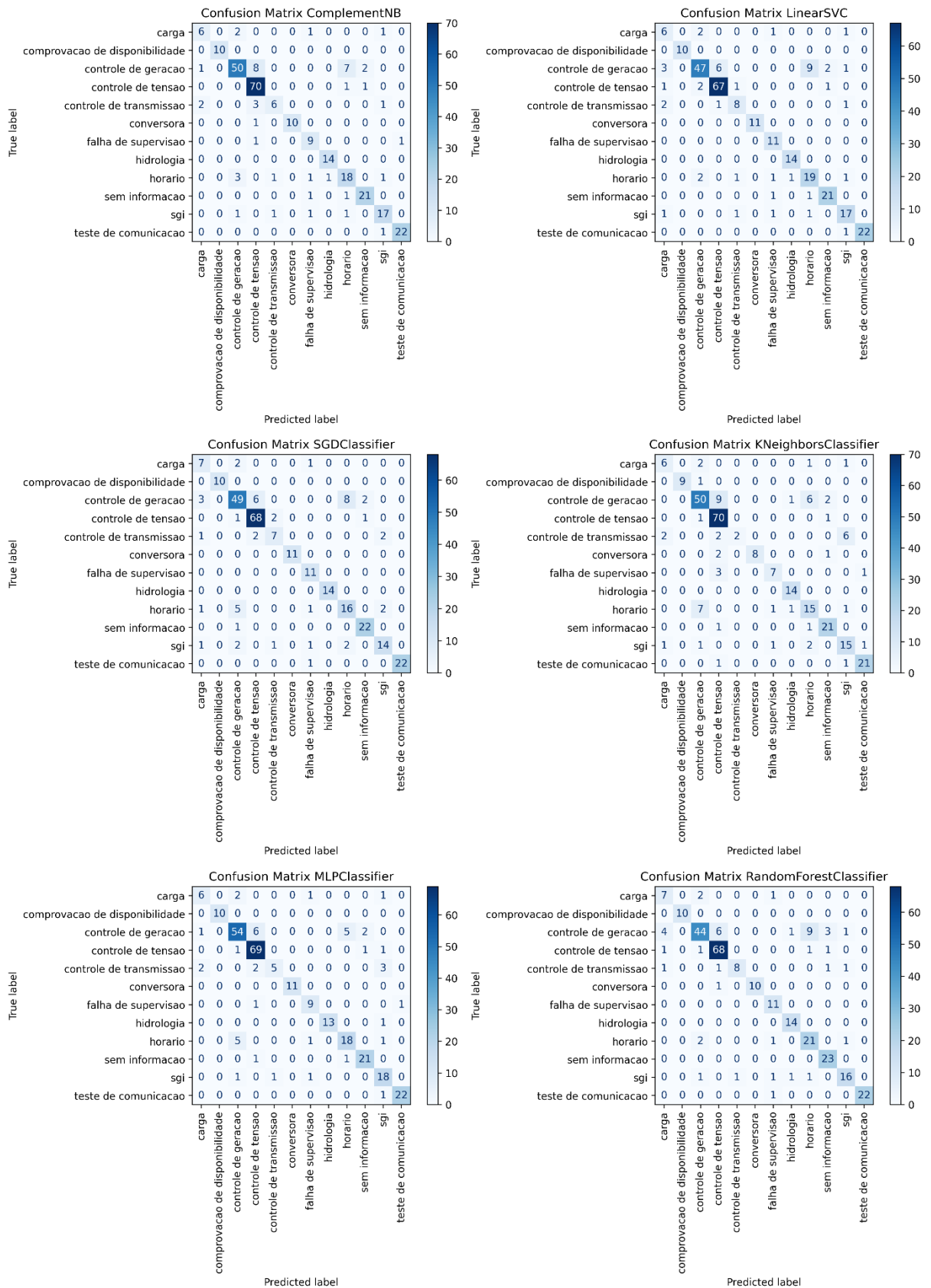
Quando se observa o valor da métrica de treino deve-se sempre considerar a natureza do problema, por exemplo, no caso da classe “hidrologia” é difícil evitar o *overfitting* pois o vocabulário usado é simples e fácil de ser identificado, ou seja, é esperado que o modelo aprenda bem e rapidamente a identificar essa classe.

3.4.1.3 Matriz de Confusão

A matriz de confusão serve para evidenciar a atribuição das amostras nas classes. A Figura 25 exibe a matriz de confusão das amostras de teste para cada modelo treinado.

Analisa-se uma matriz de confusão observando principalmente a diagonal secundária, nela está evidenciada a quantidade de amostras classificadas corretamente. Números que estão fora da diagonal secundária são classificações erradas, o verticalmente indica qual deveria ser a classe da amostra e horizontalmente qual a classe atribuída pelo estimador. Por exemplo, para o ComplementNB, 6 amostras foram classificadas corretamente como “carga”, 4 são da classe “carga” entanto estão classificadas como “controle de geração” (2), “falha de supervisão” (1) e “sgi” (1).

Figura 25 – Matriz de Confusão dos modelos



Fonte: Elaboração própria (2021).

Observando as matrizes de confusão dos modelos é possível afirmar que todos os modelos apresentam a diagonal secundária destacada, entretanto existem classes que apresentam maior erro, como observado anteriormente, “controle da transmissão”, “carga” e “confirmação de horário”.

Um apontamento interessante de ser realizado seria em relação a classe “horário” e “controle de geração”, de forma geral todos os modelos apresentaram equívocos para algumas amostras que eram pra ser classificadas como “controle de geração” e são estimadas como “horário”. Nesse sentido, durante um diálogo sobre “controle de geração” os operadores mencionam horas de forma rotineira.

A classe “carga” seria outra que foi estimada erroneamente pela maioria dos modelos, isso ocorre pois durante um corte de carga, ocorre uma intervenção (“sgi”) atrelada a esse evento e adicionalmente, a interrupção de carga pode ocorrer juntamente a uma falha de supervisão. A classe “carga” também é confundida com “controle de geração” pois a palavra “carga” faz parte de ambos os vocabulários.

De forma geral, recomenda-se uma nova rotulagem dos dados no formato de *multilabel*, visto que a classe “horário” pode ser observada ao mesmo tempo que outras. Por exemplo, supondo que exista o seguinte diálogo: “elevar 100 MW na usina. ok, vou elevar 100 MW na usina as dezoito e quarenta e seis”. Essa amostra hipotética seria classificada como controle de geração pois os autores rotularam as amostras considerando o primeiro assunto tratado. Entretanto, é informado um horário, logo, seria possível atribuir a classe “horário” para essa amostra juntamente com a classe “controle de geração”. Esse formato de rótulo, com mais de uma classe por amostra é chamado de *multilabel classification*.

3.4.1.4 Matthews Correlation Coefficient (MCC)

A métrica MCC se apresenta como uma alternativa em relação as métricas Acurácia e F1, que podem gerar resultados equivocados em conjuntos de dados desequilibrados. Com o MCC para obter uma pontuação boa (próxima a 1), o classificador deve fazer previsões corretas tanto na maioria dos casos negativos, como na maioria dos casos positivos, independentemente de suas proporções no conjunto de dados geral (CHICCO; JURMAN, 2020). A Tabela 9 exhibe os valores de MCC juntamente com o F1 ponderado para cada um dos modelos.

Tabela 9 – Valores da métrica MCC juntamente com o F1 ponderado para cada modelo

Modelo	MCC	F1 Weighted	Ordem
ComplementNB	0,8190	0,8405	4º
LinearSVC	0,8206	0,8441	3º
SGDClassifier	0,8110	0,8353	5º
KNeighborsClassifier	0,7593	0,7823	6º
MLPClassifier	0,8296	0,8498	1º
RandomForestClassifier	0,8254	0,8447	2º

Fonte: Elaboração própria (2021).

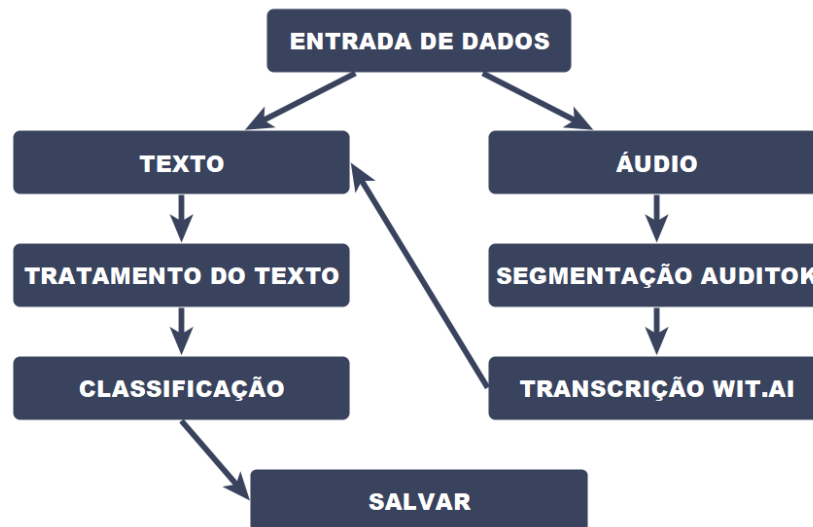
Usa-se o MCC como uma forma de validar constatações realizadas anteriormente. Pode-se comparar o F1 ponderado com o MCC e constatar que para ambas métricas, os modelos estão dispostos igualmente (de melhor para pior), MLPClassifier, RandomForestClassifier, LinearSVC, ComplementNB, SGDClassifier e KNeighborsClassifier.

3.5 Produto Minimamente Viável (MVP)

Após a criação dos modelos foi desenvolvido um MVP no formato de aplicação *web* para possibilitar que a comunidade realize classificações das próprias amostras. Sendo disponibilizado na página do Grupo de Estudos em Sistemas de Energia¹. A estrutura de navegação da aplicação segue o fluxograma exposto na Figura 26. Nele observa-se que existe a possibilidade de entrada de dados textual ou de áudio. Caso for um arquivo de áudio, ocorre o tratamento do áudio para a realização da transcrição para então efetuar o tratamento do texto e sua classificação. Por outro lado, caso o usuário opte por entrar dado textual o processo se inicia logo no tratamento do texto.

¹<https://gese.florianopolis.ifsc.edu.br/classificador-operacao/>

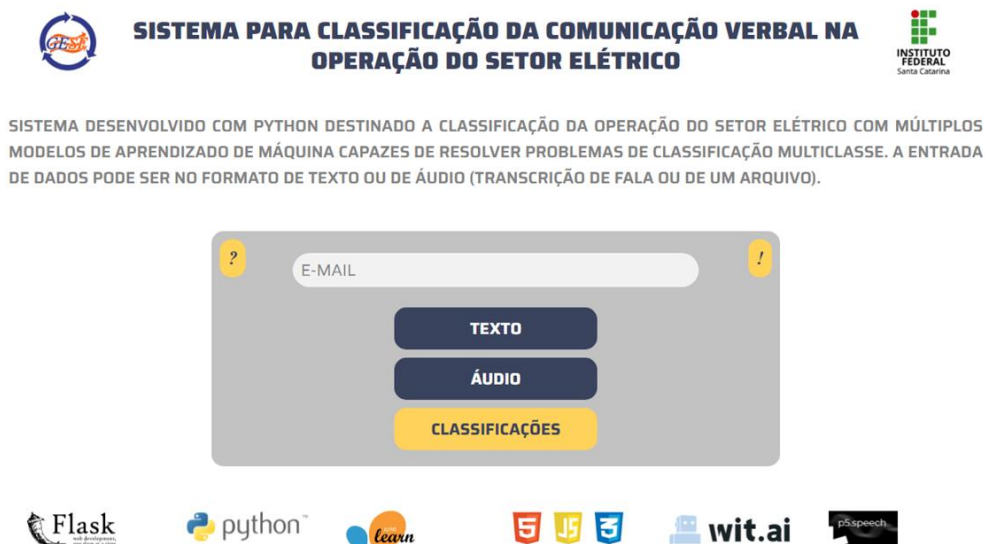
Figura 26 – Fluxo de navegação da aplicação



Fonte: Elaboração própria (2021).

A tela inicial da aplicação pode ser observada pela Figura 27, nela existe o campo para o usuário informar o e-mail para cadastro ou *login*, os botões para navegação na aplicação e as bibliotecas exploradas.

Figura 27 – Tela inicial da aplicação



Fonte: Elaboração própria (2021).

A partir da tela inicial é possível navegar para as seguintes telas:

- a) Tela de tutorial (?) de uso do sistema, exposto no Apêndice D;
- b) Tela de informação (!) da metodologia para a criação dos modelos de classificação juntamente com as métricas dos modelos para as amostras de teste;

- c) Tela para a entrada de texto, ilustrada pela Figura 28, nela o usuário pode informar o texto na caixa de texto ou pode realizar a transcrição em tempo real a partir do microfone;
- d) Tela para entrada de áudio onde o usuário pode realizar o *upload* do arquivo de áudio para a transcrição e posteriormente a classificação, conforme exibido pela Figura 29;
- e) Tela de classificações, em que existe uma tabela listando as classificações realizadas e adicionalmente existe a opção de realizar o *download* delas no formato .xlsx.

Figura 28 – Tela para entrada de texto.

CLASSIFICAÇÃO POR TEXTO

INFORME O TEXTO DA COMUNICAÇÃO VERBAL PARA A REALIZAÇÃO DA CLASSIFICAÇÃO. É POSSÍVEL IDENTIFICAR 12 CLASSES: [CARGA](#), [COMPROVAÇÃO DE DISPONIBILIDADE](#), [CONTROLE DE GERAÇÃO](#), [CONTROLE DE TENSÃO](#), [CONTROLE DE TRANSMISSÃO](#), [CONVERSORA](#), [FALHA DE SUPERVISÃO](#), [HIDROLOGIA](#), [HORÁRIO](#), [INTERVENÇÃO](#), [TESTE DE COMUNICAÇÃO](#) E [SEM INFORMAÇÃO](#).

▶

CLASSIFICAR

VOLTAR

Fonte: Elaboração própria (2021).

Figura 29 – Tela para entrada de áudio.

CLASSIFICAÇÃO POR ÁUDIO

INFORME O TEXTO DA COMUNICAÇÃO VERBAL PARA A REALIZAÇÃO DA CLASSIFICAÇÃO. É POSSÍVEL IDENTIFICAR 12 CLASSES: [CARGA](#), [COMPROVAÇÃO DE DISPONIBILIDADE](#), [CONTROLE DE GERAÇÃO](#), [CONTROLE DE TENSÃO](#), [CONTROLE DE TRANSMISSÃO](#), [CONVERSORA](#), [FALHA DE SUPERVISÃO](#), [HIDROLOGIA](#), [HORÁRIO](#), [INTERVENÇÃO](#), [TESTE DE COMUNICAÇÃO](#) E [SEM INFORMAÇÃO](#).

UPLOAD Nenhum arquivo selecionado

TRANSCREVER

VOLTAR

Fonte: Elaboração própria (2021).

Após informar a amostra dos dados ocorre a classificação, o sistema exibe o resultado da classificação para cada modelo e deve ser informado a classe verdadeira. Finalmente, após a classificação, ocorre o registro da classificação efetuada no banco de dados.

4 CONCLUSÃO

Com o presente trabalho foi possível desenvolver uma metodologia para o processamento de áudios relacionados com a comunicação verbal da operação do setor elétrico, assim como sua transcrição e pós processamento textual a fim de realizar a rotulagem e treinamento de diferentes algoritmos de classificação multiclasse.

Adicionalmente, conforme exposto pelo apêndice B, avaliou-se diferentes bibliotecas em *Python* de segmentação de áudio e por meio de uma busca entre parâmetros predeterminados encontrou-se a forma mais atrativa de segmentar áudios com o objetivo de consumir a API de transcrição do *Wit.ai*. Seguindo as métricas WER e KER, conclui-se que o *auditok* é a biblioteca mais atrativa com o parâmetro *energy_threshold* igual a 30 para a métrica WER (0,399) e para a métrica KER o *energy_threshold* que atingiu o menor valor é igual a 35 (0,236).

O trabalho trouxe métricas para validação da metodologia juntamente com a validação dos modelos criados. O WER e KER são métricas uteis para a validação do processamento e transcrição dos áudios. Em relação aos modelos de classificação, a acurácia, precisão, sensibilidade e *F1 Score* são as métricas básicas para avaliar os modelos treinados. Em contrapartida a curva de aprendizado, matriz de confusão e MCC são outras métricas válidas para avaliar o desempenho do aprendizado dos modelos.

Os resultados obtidos a partir das amostras de teste foram satisfatórios, sendo que o modelo *MLPClassifier* e *RandomForestClassifier* alcançaram os melhores valores seguindo a metodologia evidenciada, atingindo aproximadamente 0,85 de acurácia e de *F1*.

Adicionalmente foi desenvolvida uma aplicação *web* gratuita no formato de MVP, capaz de disponibilizar a metodologia criada para o público, por meio dela é possível a realização da classificação de amostras textuais ou arquivos de áudio por meio dos modelos desenvolvidos.

Apesar dos resultados satisfatórios obtidos, recomenda-se para trabalhos futuros, as seguintes análises e/ou aprimoramentos:

- a) Expandir a quantidade de amostras rotuladas e transcritas manualmente (*ground truth*);

- b) Expandir a análise usando dados de outros centros de operação regional, verificando e comparando suas especificidades;
- c) Explorar a metodologia de *Multilabel* ou *Multioutput Classification* (mais de uma classe por amostra). Para isso recomenda-se a seguinte metodologia: Dentre as seguintes classes principais: geração, transmissão, distribuição. É possível observar as seguintes ações (classes secundárias): supervisão, teste de comunicação, intervenção, teste de disponibilidade, importação, hidrologia, corte de carga, frequência, desligamento, disponibilização, autorização, retorno, proteção, problema, dúvida e sem classificação;
- d) Elaborar uma lista de palavras-chave para ser as *features* dos modelos (buscando evitar esparsidade);
- e) Hora e dia da semana como *features* para os modelos;
- f) Explorar modelos com *boost*;
- g) Uso do *scikit-optimize*;
- h) Uso de *Deep Learning* (Pythoch ou Tensorflow);
- i) Uso de *Word Embedding* como primeiro passo no *pipeline*;
- j) *Stemming* e *lemmatization* automático em português com alguma biblioteca;
- k) Explorar outras formas de transcrição;
- l) Estudo de viabilidade de aplicação de *Oversampling Techniques* (evitando desbalanceamento).

REFERÊNCIAS

- ALPHACEPHEI. **Vosk**. 2019. Disponível em: <https://alphacephei.com/vosk/>. Acesso em: 18 mar. 2021.
- AMAZON. **Amazon Transcribe**. 2017. Disponível em: <https://aws.amazon.com/pt/transcribe/>. Acesso em: 18 mar. 2021.
- ANZANELLO, Michel Jose; FOGLIATTO, Flavio Sanson. Learning curve models and applications: Literature review and research directions. **International Journal of Industrial Ergonomics**, [S. l.], v. 41, n. 5, p. 573–583, 2011. ISSN: 01698141. DOI: 10.1016/j.ergon.2011.05.001. Disponível em: <http://dx.doi.org/10.1016/j.ergon.2011.05.001>.
- BISHOP, Christopher M. **Pattern Recognition and Machine Learning**. Singapore: Springer International Publishing, 2006. ISSN: 9780387310732. ISBN: 0387310738.
- BOTTOU, Léon; BOUSQUET, Olivier. Learning using Large Datasets. **NATO ASI Mining Massive Data Sets for Security**, [S. l.], p. 15--26, 2007. DOI: 10.3233/978-1-58603-898-4-15. Disponível em: http://books.google.com/books?hl=en&lr=&id=4EjpmwhD7UC&oi=fnd&pg=PA15&dq=Learning+using+Large+Datasets&ots=kM1xsY_xc3&sig=Tn1yyf1dsbjy2_HY7DhtM3c2eik%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.8591&rep.
- BOUCKAERT, Remco R.; FRANK, Eibe. Evaluating the replicability of significance tests for comparing learning algorithms. **Lecture Notes in Computer Science**, [S. l.], v. 3056, p. 3–12, 2004. ISSN: 16113349. ISBN: 354022064X. DOI: 10.1007/978-3-540-24775-3_3.
- BREIMAN, LEO. Random forests. **Machine Learning**, [S. l.], v. 45, p. 5–32, 2001. ISBN: 9783110941975. DOI: 10.1201/9780429469275-8.
- CARNEGIE MELLON UNIVERSITY. **CMUSphinx**. 2000. Disponível em: <https://cmusphinx.github.io>. Acesso em: 13 mar. 2021.
- CHANG, Chih Chung; LIN, Chih Jen. LIBSVM: A Library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, [S. l.], v. 2, n. 3, p. 1–39, 2011. ISSN: 21576904. DOI: 10.1145/1961189.1961199.
- CHICCO, Davide; JURMAN, Giuseppe. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. **BMC Genomics**, [S. l.], v. 21, p. 1–13, 2020. ISSN: 14712164. DOI: 10.1186/s12864-019-6413-7.
- DIETTERICH, Thomas G. Ensemble Methods in Machine Learning. **International Workshop on Multiple Classifier Systems**, [S. l.], p. 1–15, 2000. DOI: 10.1007/3-540-45014-9_1.
- ERRATTAHI, Rahhal; HANNANI, Asmaa El; OUAHMANE, Hassan. Automatic speech recognition errors detection and correction: A review. *In*: PROCEDIA

COMPUTER SCIENCE 2018, Argel. **Anais [...]**. Argel: Elsevier B.V., 2018 p. 32–37. ISSN: 18770509. DOI: 10.1016/j.procs.2018.03.005. Disponível em: <https://doi.org/10.1016/j.procs.2018.03.005>.

FAN, Rong En; CHANG, Kai Wei; HSIEH, Cho Jui; WANG, Xiang Rui; LIN, Chih Jen. LIBLINEAR: A library for large linear classification. **Journal of Machine Learning Research**, [S. l.], v. 9, p. 1871–1874, 2008. ISSN: 15324435. DOI: 10.1145/1390681.1442794.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn e TensorFlow**. Sebastopol: O'Reilly, 2017. ISBN: 9781491962299.

GOOGLE. **Google Cloud - Speech-to-Text**. 2016. Disponível em: <https://cloud.google.com/speech-to-text>. Acesso em: 18 mar. 2021.

GORODKIN, J. Comparing two K-category assignments by a K-category correlation coefficient. **Computational Biology and Chemistry**, [S. l.], v. 28, n. 5–6, p. 367–374, 2004. ISSN: 14769271. DOI: 10.1016/j.compbiolchem.2004.09.006.

HUNTER, John D.; DROETTBOOM, Michael. **matplotlib 3.3.4**. 2021. Disponível em: <https://matplotlib.org>. Acesso em: 18 mar. 2021.

HUYNH, David. **OpenRefine**. 2012. Disponível em: <https://openrefine.org>.

IBM. **Watson Speech to Text**. 2004. Disponível em: <https://www.ibm.com/cloud/watson-speech-to-text>. Acesso em: 18 mar. 2021.

IZBICKI, Rafael; DOS SANTOS, Tiago Mendonça. **Aprendizado de Máquina: Uma abordagem estatística**. São Carlos: livro eletrônico, 2020. ISBN: 978-65-00-02410-4.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert. **An introduction to Statistical Learning**. New York: Springer International Publishing, 2000. v. 7 ISSN: 0929-8673. ISBN: 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7.

JOSHI, Prateek. **Artificial intelligence with Python**. Mumbai: Packt, 2017. ISBN: 9781786464392.

JUANG, B. H.; RABINER, Lawrence R. Automatic Speech Recognition – A Brief History of the Technology Development. **Elsevier Encyclopedia of Language and Linguistics**, [S. l.], v. 50, n. 2, p. 637–655, 2004.

LANE, Hobson; HOWARD, Cole; HAPKE, Hannes Max. **Natural Language Processing in Action**. Shelter Island: Manning Publications Co., 2019. ISBN: 9781617294631.

LESKOVEC, Jure; RAJARAMAN, Anand; ULLMAN, Jeffrey D. **Mining of Massive Datasets**. Palo Alto: Stanford, 2010. v. 9781107015 ISBN: 9781139058452. DOI: 10.1017/CBO9781139058452.

LIMA, Matheus N. S. M. De; COELHO, Bruna; TAKIGAWA, Fabrício. **Ferramentas e**

recursos disponíveis para reconhecimento de fala em Português Brasileiro. [s.l.] : Association for Computing Machinery, 2021. DOI: 10.14210/cotb.v12.p475-479.

MCMAHAN, Brian; RAO, Delip. **Natural Language Processing with PyTorch - Build Intelligent Language Applications Using Deep Learning.** Sebastopol: O'Reilly Media, 2019. ISSN: 00157120. ISBN: 9788578110796.

MICROSOFT. **Microsoft Azure - Conversão de Fala em Texto.** 2010. Disponível em: <https://azure.microsoft.com/pt-br/services/cognitive-services/speech-to-text/>. Acesso em: 18 mar. 2021.

MIKOLOV, Tomas; LE, Quoc V.; SUTSKEVER, Ilya. Exploiting Similarities among Languages for Machine Translation. [S. l.], 2013. Disponível em: <http://arxiv.org/abs/1309.4168>.

MORRIS, Andrew C.; MAIER, Viktoria; GREEN, Phil. From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition. *In*: 8TH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING, ICSLP 2004 2004, Korea. **Anais** [...]. Korea p. 2765–2768.

MURTAGH, Fionn. Multilayer perceptrons for classification and regression. **Neurocomputing**, [S. l.], v. 2, n. 5–6, p. 183–197, 1991. ISSN: 09252312. DOI: 10.1016/0925-2312(91)90023-5.

NUMPY. **The fundamental package for scientific computing with Python.** 2005. Disponível em: <https://numpy.org>. Acesso em: 18 mar. 2021.

ONS. **Submódulo 18.2 - Relação dos sistemas e modelos computacionais.** Operador Nacional do Sistema Elétrico, 2009.

ONS. **Submódulo 10.13 - Apuração da geração e de indisponibilidade de empreendimentos de geração.** Operador Nacional do Sistema Elétrico, 2019.

ONS. **Resultados da Apuração do PPR - Programa de Participação nos Resultados 2019.** Operador Nacional do Sistema Elétrico, 2020 a.

ONS. **PAR/PEL - Sumário Executivo.** Operador Nacional do Sistema Elétrico, 2020 b.

ONS. **Submódulo 5.13 - Comunicação Verbal na Operação.** Operador Nacional do Sistema Elétrico, 2020 c.

ONS. **Submódulo 13.2 Requisitos - Requisitos mínimos de telecomunicações.** Operador Nacional do Sistema Elétrico, 2020 d.

OPEN REFINE. Open Refine: Clustering Methods. [S. l.], 2021. Disponível em: <https://docs.openrefine.org/manual/cellediting#clustering-methods>.

PANDAS. **Pandas 1.2.3 documentation.** 2021. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 9 abr. 2021.

PARK, Youngja; PATWARDHAN, Siddharth; VISWESWARIAH, Karthik; GATES, Stephen C. An empirical analysis of word error rate and keyword error rate. **Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH**, [S. l.], n. September, p. 2070–2073, 2008. ISSN: 19909772.

PEDREGOSA, Fabian et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, [S. l.], v. 12, p. 2825–2830, 2011.

PYPI. **PyPI - The Python Package Index**. 2020. Disponível em: <https://pypi.org>. Acesso em: 18 mar. 2021.

PYTHON. **Python 3.9.2 documentation**. 2021. Disponível em: <https://docs.python.org/3/>. Acesso em: 18 mar. 2021.

ŘEHŮŘEK, Radim. **Gensim**. 2009. Disponível em: <https://radimrehurek.com/gensim/index.html>.

REHUREK, Radim; SOJKA, Petr. Software Framework for Topic Modelling with Large Corpora. **Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks**, [S. l.], p. 45–50, 2010. ISSN: 2951740867. ISBN: 2-9517408-6-7.

RENNIE, Jason D. M.; SHIH, Lawrence; TEEVAN, Jaime; KARGER, David. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. **Proceedings, Twentieth International Conference on Machine Learning**, [S. l.], v. 2, n. May 2013, p. 616–623, 2003. ISBN: 1577351894.

ROBBINS, Herbert; MONRO, Sutton. A Stochastic Approximation Method. **The Annals of Mathematical Statistics**, [S. l.], v. 22, n. 3, p. 400–407, 1951. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729586.

SCIKIT-LEARN. **Choosing the right estimator**. 2021. Disponível em: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html. Acesso em: 20 mar. 2021.

SEHILI, Amine. **auditok 0.2.0**. 2021. Disponível em: <https://auditok.readthedocs.io/en/v0.2.0/>. Acesso em: 18 mar. 2021.

SOKOLOVA, Marina; JAPKOWICZ, Nathalie; SZPAKOWICZ, Stan. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. **AAAI Workshop - Technical Report**, [S. l.], v. WS-06-06, p. 24–29, 2006. ISBN: 1577352882. DOI: 10.1007/11941439_114.

STACK EXCHANGE. **Stack Overflow Trends**. 2021. Disponível em: https://insights.stackoverflow.com/trends?tags=java%2Cpython%2Cc%23%2Cjavascript%2Cphp%2Cc%2B%2B%2Creactjs%2Ckotlin&utm_source=ActiveCampaign&utm_medium=email&utm_content=MIT%2C+empregos+e+Python&utm_campaign=%5BIMERSÔES%5D+%28Newsletter%29+%2307. Acesso em: 17 mar. 2021.

STEVEN BIRD, Ewan Klein; LOPER, Edward. **Natural Language Processing with Python**. [s.l.] : O'Reilly, 2009. ISBN: 9780596516499.

TAUNK, Kashvi; DE, Sanjukta; VERMA, Srishti; SWETAPADMA, Aleena. A brief review of nearest neighbor algorithm for learning and classification. **2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019**, [S. l.], n. Icccs, p. 1255–1260, 2019. ISBN: 9781538681138. DOI: 10.1109/ICCS45141.2019.9065747.

TING, Kai Ming. Confusion Matrix. **Lecture Notes in Computer Science**, [S. l.], v. 6820 LNAI, p. 17–18, 2011. ISSN: 03029743. ISBN: 9783642225888. DOI: 10.1007/978-3-642-22589-5_2.

VAESSEN, Nik. **jiwer 2.2.0**. 2004. Disponível em: <https://pypi.org/project/jiwer/>. Acesso em: 18 mar. 2021.

VALKLING. **99% Accuracy**. 2020. Disponível em: https://www.reddit.com/r/DataScienceMemes/comments/jfn5r3/99_accuracy/.

VAN ROSSUM GUIDO, DRAKE JR, Fred L. **The Python 3 Language Reference**. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697. Disponível em: <https://docs.python.org/3/download.html>.

VIARBITSKAYA, Tatsiana; DOBRUCKI, Andrzej. Audio processing with using Python language science libraries. *In*: SIGNAL PROCESSING - ALGORITHMS, ARCHITECTURES, ARRANGEMENTS, AND APPLICATIONS CONFERENCE PROCEEDINGS, SPA 2018, Poznan. **Anais** [...]. Poznan p. 350–354. ISSN: 23260319. ISBN: 9788362065318. DOI: 10.23919/SPA.2018.8563430.

WAN, Eric A. Discrete time neural networks. **Applied Intelligence**, [S. l.], v. 3, n. 1, p. 91–105, 1993. ISSN: 0924669X. DOI: 10.1007/BF00871724.

WANG, Sida; MANNING, Christopher D. Baselines and bigrams: Simple, good sentiment and topic classification. *In*: 50TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, ACL 2012 - PROCEEDINGS OF THE CONFERENCE 2012, Jeju Island. **Anais** [...]. Jeju Island p. 90–94. ISBN: 9781937284251.

WIT.AI. **Build Natural Language Experiences**. 2013. Disponível em: <https://wit.ai>. Acesso em: 18 mar. 2021.

ZHANG, Anthony. **SpeechRecognition 3.8.1**. 2017. Disponível em: <https://pypi.org/project/SpeechRecognition/>. Acesso em: 18 mar. 2021.

ZHANG, Harry. The optimality of Naive Bayes. **Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004**, [S. l.], v. 2, p. 562–567, 2004. ISBN: 1577352017.

**APÊNDICE A – ARTIGO FERRAMENTAS E RECURSOS PARA
RECONHECIMENTO DE FALA EM PORTUGUÊS BRASILEIRO**

Ferramentas e recursos disponíveis para reconhecimento de fala em Português Brasileiro

Matheus N. S. M. de Lima
matheus.marques_96@hotmail.com
IFSC - Instituto Federal de Santa
Catarina
Florianópolis, Santa Catarina, Brasil

Bruna H. Coelho
bhamess@icloud.com
IFSC - Instituto Federal de Santa
Catarina
Florianópolis, Santa Catarina, Brasil

Fabrcio Y. K. Takigawa
takigawa@ifsc.edu.br
IFSC - Instituto Federal de Santa
Catarina
Florianópolis, Santa Catarina, Brasil

RESUMO

Speech recognition allows natural communication between the humans and machines. With Industry 4.0 there is a great demand for systems that perform this task, since human-machine integrations are increasingly attractive. Currently, there are several tools and resources that perform this activity, with some companies providing their audio recognition services through the Application Programming Interface, such as Microsoft, Google, IBM and Wit. On the other hand, there are offline libraries and open source that can also be explored like Vosk. Each company has its business rule and its specificity, in this sense it is difficult to know which is the most interesting for each situation. Thus, a comparison was made between speech recognition services in terms of usability, limitation and precision. In the comparison, speech recognition performance metrics were used in a set of audios, using the programming language Python.

KEYWORDS

Conjunto de Dados, Transcrição de Áudio, Sistemas de Reconhecimento de Fala, Interface de Programação de Aplicações

1 INTRODUÇÃO

A necessidade por reconhecimento de fala é algo cada vez mais presente no cotidiano. O conceito da Internet das Coisas (IoT) e da Indústria 4.0 eleva cada vez mais a proximidade do ser humano com a máquina. Nesse sentido, a crescente necessidade por ferramentas e recursos acarretou na abertura de um mercado de serviços *online* que realizam a tarefa de reconhecimento de fala. Pode-se citar, como exemplos, as empresas Microsoft, Google, IBM e Wit. Entretanto, em paralelo a esse mercado, existem iniciativas de código aberto que também buscam desempenhar essa atividade de forma *offline*, como CMU Sphinx e Vosk [1].

Dentre essas diversas opções, seja de código aberto ou fechado, cada serviço pode possuir vantagens e desvantagens. Pode-se afirmar que a quantidade de opções disponíveis, dificulta a tomada de decisão do consumidor. Nesse sentido, neste artigo, inicialmente, foi realizado um levantamento das principais ferramentas e os recursos atuais disponíveis. E utilizando a linguagem de programação Python, foram realizados testes com um conjunto de áudios para analisar o desempenho de cada utilitário. A Figura 1 exibe o processo geral da execução e a metodologia utilizada para a análise dos serviços selecionados.

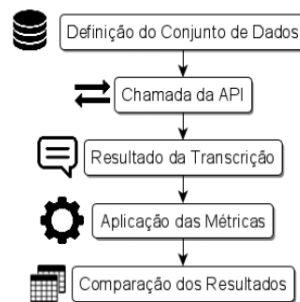


Figura 1: Processo geral da proposta de análise dos serviços de transcrição.

O conjunto de dados (*dataset*) é composto de trechos de cem áudios com duração inferior a vinte segundos. Os arquivos de áudios são gravações telefônicas de conversas entre operadores do setor elétrico. Nesse sentido, será evidenciado o desempenho geral dos serviços e, em paralelo, será avaliada a transcrição de palavras específicas do contexto técnico, como nome dos equipamentos, unidades de medidas, entre outros.

A principal forma de disponibilização dos serviços de transcrição de voz (*Speech to Text*) é por meio de Interface de Programação de Aplicações (*Application Programming Interface - API*), que se trata de um conjunto de tarefas que permite o envio de informações para um servidor, e a partir disso, obtém-se uma resposta (*request-response*). Com isso, as empresas prestadoras de serviço podem monitorar o uso de seus serviços e cobrar taxas caso for cabível.

O artigo está organizado da seguinte maneira: na Seção 2 estão ilustrados os requisitos necessários para o desenvolvimento da análise e quais os utilitários utilizados no artigo. Na Seção 3, os principais resultados são apresentados. Na Seção 4 são abordadas as conclusões; e por fim, na Seção 5, as considerações finais e os desenvolvimentos futuros são expostos.

2 MATERIAL E MÉTODOS

A principal ferramenta usada para a realização da transcrição do *dataset* foi a linguagem de programação Python. A utilização dessa ferramenta apresenta muitas vantagens, sendo as principais: *software* de código aberto (*open-source*), adaptação em diferentes sistemas operacionais, linguagem intuitiva e *user-friendly*, possibilidade de implementação de várias bibliotecas, gerenciamento de memória, operações em múltiplas *threads* e grande documentação básica [2].

Para a programação com Python, basicamente, é necessário ter definido três itens: o *Core*, o Ambiente de Desenvolvimento Integrado e as Bibliotecas. O *Core* é o interpretador da linguagem junto com algumas bibliotecas padrões. O Ambiente de Desenvolvimento Integrado (do inglês *Integrated Development Environment*) é a interface usada para o desenvolvimento, cita-se: Pyzo, PyCharm, Spider, Jupiter. E as bibliotecas são uma coleção de módulos de *script* que executam determinada atividade. No contexto atual, vale citar as seguintes bibliotecas e módulos:

- *numpy*: pacote para Python suportar *arrays* e matrizes multidimensionais.
- *glob*: módulo para encontrar todos os nomes de um determinado caminho.
- *pydub*: biblioteca para manipular áudio com uma interface simples e de alto nível.
- *audiotok*: biblioteca para detecção de atividade de áudio.
- *matplotlib*: biblioteca abrangente para a visualização de dados.
- *pathlib*: biblioteca para manipular caminhos de sistema de arquivos de maneira independente.
- *SpeechRecognition*: biblioteca para realização de reconhecimento de fala, com suporte para diversos *engines* e APIs, *online* e *offline*
- *wave*: fornece uma interface para o formato de som WAV.
- *jiwer*: pacote Python para cálculo *Word Error Rate (WER)*, *Match Error Rate (MER)*, *Word Information Lost (WIL)* e *Word Information Preserved (WIP)* de uma transcrição.

Em relação ao formato dos áudios, são trechos de ligações (formato WAV) com um único canal e taxa de amostragem de 8.000 kHz, contendo um diálogo específico relacionado a operação do setor elétrico. Os serviços que foram selecionados para análise e comparação de resultados por meio da transcrição do *dataset* são:

- Microsoft Azure: reconhecimento de voz, permite a transcrição em tempo real e em lote de *streams* de áudio em texto [3].
- Google Cloud: usa tecnologia de reconhecimento de voz de código fechado com base em aprendizagem profunda [4].
- Wit: grátis até para uso comercial, suporta 130 línguas. Suporta as linguagens de programação Node, Python, e Ruby [5].
- IBM Watson Speech to Text: serviço aproveita o aprendizado de máquina para combinar o conhecimento da gramática, a estrutura do idioma e a composição de sinais de áudio e de voz e atualiza continuamente e refina sua transcrição à medida que recebe mais falas [6].
- Vosk: Suporta 17 línguas e dialetos, código-aberto e funciona *offline* [7].

Vale destacar que os testes foram realizados usando o plano gratuito disponibilizado pelas empresas citadas. No entanto, para algumas, existe a possibilidade de contratação de um plano pago com mais vantagens e menos limitações. A Tabela 1 mostra as limitações dos planos disponibilizados gratuitamente.

Tabela 1: Limitações dos planos gratuitos.

	Azure	IBM	Google	Wit	Vosk
Online/Pagos	Sim	Sim	Sim	Sim	Não
Limite/mês	300 min	500 min	60 min	-	∞
Limite/request	1 min	100 MB	1 min	20 seg	∞
Personalização	Sim	Sim	Não	Não	Sim

Com relação as métricas para análise das transcrições, adotou-se os métodos recomendados pela literatura, *Word Error Rate (WER)*, *Match Error Rate (MER)*, *Word Information Lost (WIL)*. [8]. A métrica mais comum é a *Word Error Rate (WER)*, que usa a soma de substituições (S), deleções (D), inserções (I) e divide pela soma de palavras da sentença correta (N) [9].

$$WER = \frac{S + D + I}{N} \quad (1)$$

Uma segunda métrica recomendada por [10] é a *Match Error Rate (MER)*, que é soma de S, D e I dividida pela soma dos acertos (H), S, D e I.

$$MER = \frac{S + D + I}{H + S + D + I} = 1 - \frac{H}{N} \quad (2)$$

E a terceira métrica recomendada por [8] é a *Word Information Lost (WIL)*, que é um menos os acertos ao quadrado (H^2) dividido pela multiplicação da quantidade de palavras na entrada (N_1) pela quantidade de palavras na saída (N_2).

$$WIL = 1 - \left(\frac{H}{N_1 N_2} \right) \quad (3)$$

Para exemplificar o comportamento das métricas a Tabela 2 exhibe o resultado em porcentagem dos critérios para determinadas entradas e saídas. Em que, X, Y e Z são palavras arbitrárias, I remete a uma inserção e D a uma deleção [8].

Tabela 2: Comparação WER, MER e WIL [8].

Entrada	Saída	H	S	D	I	%WER	%MER	%WIL
X	X	1	0	0	0	0	0	0
Xiii	XXYY	1	0	0	3	300	75	75
XYX	XZd	1	1	1	0	67	67	83
X	Y	0	1	0	0	100	100	100
Xi	YZ	0	1	0	1	200	100	100

Para a avaliação geral dos serviços foi considerada a média das métricas. Neste sentido, quanto menor o valor, melhor será a precisão geral do serviço. Adicionalmente às métricas, foi avaliado o desempenho dos serviços em relação a palavras-chaves relacionadas ao contexto do *dataset*. Foi verificado o desempenho da transcrição de dez palavras específicas e suas variações, estas palavras são termos técnicos comuns utilizados na operação do setor elétrico.

A Tabela 3 exhibe as dez palavras-chaves selecionadas e a quantidade de ocorrências de cada uma no *dataset*.

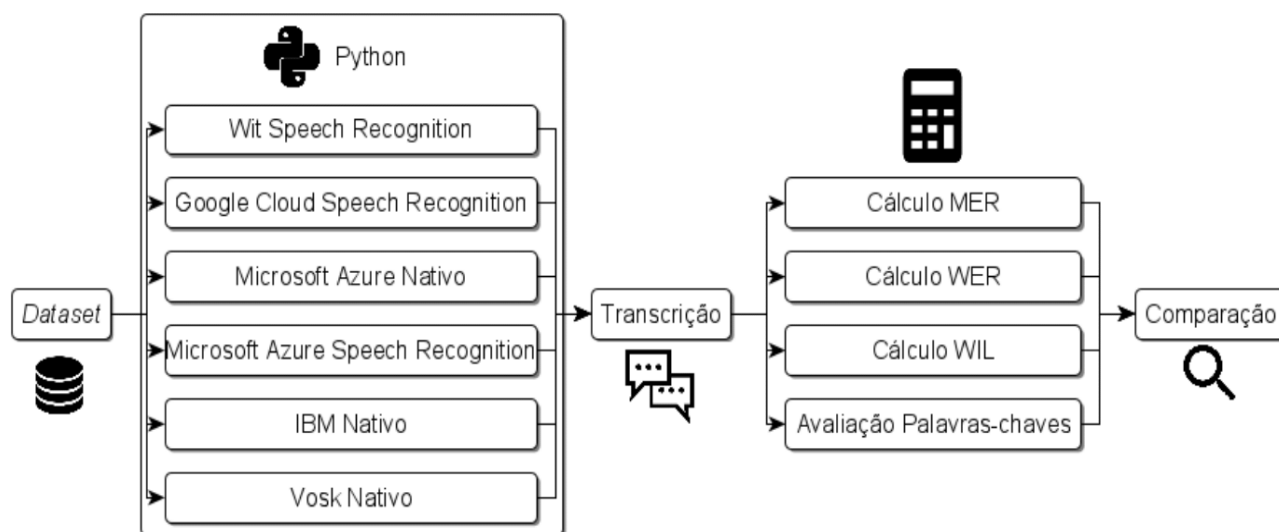


Figura 2: Processo de transcrição e análise.

Tabela 3: Listagem de palavras-chaves e suas ocorrências.

Ordem	Palavra-chave	nº de Ocorrências
i	kV	25
ii	megawatt	15
iii	geracao	23
iv	disjuntor	6
v	tap	7
vi	tensão	15
vii	barra	10
viii	máquina	7
ix	usina	9
x	compensador	6
	total	123

Para cada ocorrência de um termo específico foi verificado se ocorreu a transcrição correta. Nesse sentido, foi possível avaliar os serviços de forma geral e também de forma específica para o contexto proposto.

O processo adotado para a análise e comparação dos serviços é evidenciado na Figura 2. Conforme ilustrado, foi realizada a transcrição e análises usando 6 serviços:

- Microsoft Azure com a biblioteca Speech Recognition
- Microsoft Azure com a biblioteca nativa
- Google Cloud com a biblioteca Speech Recognition
- Wit com a biblioteca Speech Recognition
- IBM Watson com a biblioteca nativa
- Vosk com a biblioteca nativa

3 RESULTADOS

Por meio do uso da linguagem de programação Python em conjunto com a IDE Spyder, foi realizada a transcrição do *dataset* e análise

dos resultados de forma comparativa por meio das métricas propostas. A Tabela 4 sumariza os resultados das métricas para cada serviço testado.

Tabela 4: Resultados das métricas WER, MER e WIL para cada serviço em porcentagem.

Serviço	Biblioteca	WER	MER	WIL	Total
Azure	SpeechRecognition	50	46	67	54
Azure	Nativamente	49	46	65	53
Google	SpeechRecognition	50	49	67	55
Wit	SpeechRecognition	45	42	60	49
Vosk	Nativamente	94	94	97	95
IBM	Nativamente	72	68	88	76

A partir da Tabela 4 é possível constatar que o serviço que apresentou o melhor desempenho foi o Wit, seguido pela Microsoft Azure usando a biblioteca nativa da Microsoft. Vosk e IBM apresentaram os piores resultados, consecutivamente. A seguir é ilustrado um resumo do que foi observado dos serviços disponíveis, na ordem dos piores para os melhores, na métrica utilizada.

Vosk é uma biblioteca *offline* que funciona baseado em um modelo treinado. Para a execução das transcrições foi utilizado o modelo pt-BR disponibilizado no site gratuitamente. Esse modelo é relativamente pequeno e não foi treinado para executar transcrições de ligações telefônicas. Por isso a transcrição acabou não sendo efetiva (a biblioteca Vosk foi a única que não transcreveu todos os cem áudios do *dataset*, apenas 28 áudios), o que acarretou em uma perda significativa na sua avaliação por meio das métricas utilizadas (WER, MER e WIL).

Por outro lado, a IBM é um serviço consolidado no mercado. E, suspeita-se que o serviço gratuito não contempla um modelo voltado para transcrição de ligações telefônicas. Adicionalmente, o Google Cloud apresentou um desempenho mediano, mas que poderia ser melhorado com a contratação dos serviços específicos para

ter acesso a modelos de transcrição de áudios de ligações telefônicas.

O serviço da Microsoft Azure foi testado de duas formas, com o uso da biblioteca nativa da empresa e com a biblioteca SpeechRecognition. Nesse sentido, foi possível comparar o desempenho do serviço usando as bibliotecas. A Tabela 4 indica que o uso da biblioteca nativa apresentou melhor desempenho. Logo, entende-se que a biblioteca SpeechRecognition deve realizar algum tipo de tratamento nos dados antes de realizar a requisição.

O Wit, por sua vez, foi o que apresentou o melhor desempenho dos serviços analisados. Pode-se afirmar que o Wit obtém boas respostas na transcrição de comandos rápidos. Entretanto, o serviço apresenta limitação na requisição (20 segundos de áudio por requisição) e não possibilita a personalização do serviço.

Em relação as palavras-chaves selecionadas para avaliação, foram encontradas no total 123 ocorrências no conjunto de dados transcritos. A Tabela 5 evidencia em porcentagem a quantidade de acertos de cada palavra para cada serviço. Note que quando o serviço tiver a sigla SR remete que a requisição foi feita usando a biblioteca SpeechRecognition e quando tiver a sigla N remete a biblioteca nativa do serviço.

Tabela 5: Avaliação geral de acertos de palavras-chaves em porcentagem.

Serviço	i	ii	iii	iv	v	vi	vii	viii	ix	x
Azure SR	24	27	83	50	57	0	90	100	100	67
Azure N	28	20	83	67	43	0	70	86	100	67
Google SR	56	0	74	100	29	73	70	71	78	67
Wit SR	44	20	91	100	14	67	90	100	100	83
Vosk N	0	0	0	0	0	0	0	0	0	0
IBM N	0	0	74	33	0	0	0	0	78	17

A Tabela 6 exibe o resultado total de acertos, em porcentagem, para cada serviço.

Tabela 6: Avaliação total de acertos de palavras-chaves em porcentagem.

Serviço	Total
Wit SR	67
Google SR	59
Azure SR	53
Azure N	50
IBM N	22
Vosk N	0

Mantendo as constatações realizadas anteriormente, o Wit apresentou os melhores valores de acerto das palavras-chave. Entretanto, em segundo lugar, para essa métrica, está o Google Cloud. Para as métricas anteriores (WER, MER e WILL) o segundo melhor serviço tinha sido o Azure usando a biblioteca nativa. Entretanto ao observar somente as palavras do contexto técnico (voltadas à operação do setor elétrico), o segundo melhor recurso é o da empresa Google.

Outra análise feita foi através de uma média do percentual de acertos para cada um dos dez termos técnicos selecionados para a avaliação das palavras-chaves. Pode-se observar num contexto geral quais palavras obtiveram maior número de acertos, e em contrapartida, quais palavras apresentaram maiores dificuldades para serem transcritas. A Tabela 7 exibe a porcentagem de acertos médio para cada uma das palavras-chaves.

Tabela 7: Porcentagem de acertos médio referente a cada palavra-chave.

Palavras-chaves	Média percentual
Usina	76
Máquina	71
Geração	68
Barra	58
Disjuntor	58
Compensador	50
Tap	33
Tensão	27
kV	26
Megawatt	13

Comparando os resultados apresentados na Tabela 7, observa-se que as palavras com maiores dificuldades de transcrição foram: "megawatt", "kV" e "tensão", atingindo uma média percentual menor do que trinta por cento. As palavras com melhor desempenho não ultrapassaram os oitenta por cento, essa média pode ter sido prejudicada por conta dos serviços que apresentaram uma performance muito inferior.

Em termos gerais, deve-se levar em consideração que a análise está sendo feita a partir de áudios em gravações telefônicas. Portanto, trata-se de um ambiente não-controlado, isto é, um ambiente passível de interferências sonoras do local como vozes e ruídos. Na análise de palavras-chaves foram considerados termos técnicos, que talvez não sejam muito usuais na linguagem informal do português brasileiro, o que pode ter determinado resultados negativos se levar em consideração que os serviços utilizados não foram treinados para estas especificidades.

4 CONSIDERAÇÕES FINAIS

Todas as constatações realizadas são referentes a transcrição dos dados em questão, logo para análises mais abrangentes e gerais recomenda-se aumentar e diversificar o *dataset* explorado. O contexto atual se limita ao desempenho da transcrição dos serviços gratuitos em relação a comunicação verbal da operação do setor elétrico.[11].

Nos testes efetuados no artigo, pode-se observar que o Wit foi o que apresentou o melhor desempenho entre os serviços analisados. Entretanto, esse serviço é o que mais apresenta limitação por requisição, sendo que em cada requisição só pode ser enviado 20 segundos de áudio. Em relação a personalização dos modelos, as empresas Azure e IBM possibilitam a personalização e o Wit não. Apesar dessas limitações, deve-se levar em consideração que o objetivo do Wit é transcrever comandos rápidos e executar determinadas ações baseado na intenção do usuário.

Por outro lado, mesmo com desempenho insatisfatório, o Vosk é um serviço promissor e pode ser explorado com uma abordagem diferente ao utilizado (modelo disponível no site). É possível usar um programa chamado Kaldi e aprimorar um modelo existente para obter resultados mais atraentes. Outra possibilidade é o uso do Vosk Server (*online*) e consumir o serviço por meio do uso de um *WebSocket* (tecnologia que permite a comunicação entre o cliente e um servidor) com um modelo voltado para chamadas telefônicas. Entretanto, até o momento, esse serviço não estava disponível para o público, podendo ser testado somente mediante contato com a empresa desenvolvedora.

REFERÊNCIAS

- [1] Lawrence R. Rabiner B. H. Juang. Automatic speech recognition a brief history of the technology development, 2004.
- [2] T. Viarbitskaya and A. Dobrucki. Audio processing with using python language science libraries. In *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 350–354, 2018. doi: 10.23919/SPA.2018.8563430.
- [3] Microsoft. Speech service documentation, 2020. URL <https://docs.microsoft.com/en-us/azure/cognitive-services/Speech-Service/>.
- [4] Google. Speech-to-text, 2020. URL <https://cloud.google.com/speech-to-text>.
- [5] Wit. Build natural language experiences, 2013. URL <https://wit.ai>.
- [6] IBM. Speech to text, 2020. URL <https://cloud.ibm.com/catalog/services/speech-to-text>.
- [7] Alpha Cephei. Vosk, 2020. URL <https://alphacephei.com/vosk/>.
- [8] Phil Green Andrew C. Morris, Viktoria Maier. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *INTERSPEECH*, 2004.
- [9] Alex Acero Xuedong Huang, Raj Reddy. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 1nd. edition, 2001. ISBN 0-13-022616-5.
- [10] Elias Pimenidis Ilias Maglogiannis, Lazaros Iliadis. benchmarking of ibm, google and wit automatic speech recognition systems. *Artificial Intelligence Applications and Innovations*, 2020. doi: 10.1007/978-3-030-49161-1_7.
- [11] ONS. Comunicação verbal na operação, 2020. URL http://www.ons.org.br/%2FMPO%2FDocumento%20Normativo%2F4.%20Rotinas%20Operacionais%20-%20SM%2010.22%2F4.1.%20Rotinas%20Gerais%2F4.1.7.%20Relacionamento%20Operacional%2FRO-RO.BR.01_Rev.12.pdf.

APÊNDICE B – COMPARAÇÃO ENTRE BIBLIOTECAS E PARÂMETROS RELACIONADOS A SEGMENTAÇÃO DE ÁUDIOS

Buscando validar a escolha da biblioteca *auditok* foi elencada duas outras bibliotecas capazes de realizar a segmentação de áudios (*webrtcvad* e *pyAudioAnalysis*). Como critério de avaliação explorou-se as métricas WER e KER.

Ocorreu inicialmente a escolha das bibliotecas e da variação dos parâmetros, então o processamento dos áudios (segmentação), a transcrição deles com a API do *Wit.ai* e por fim o cálculo do WER e KER médio. A Tabela B1 exibe as bibliotecas, parâmetros, valores selecionados e descrição dos parâmetros para criar os *datasets*.

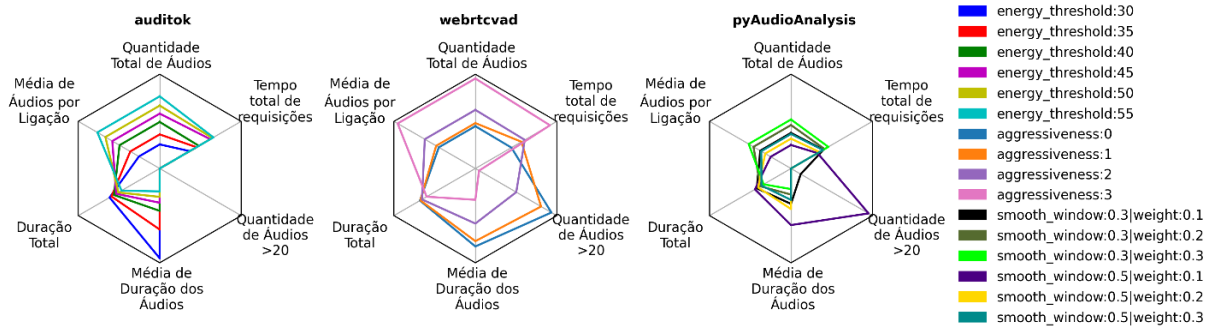
Tabela B1 – Bibliotecas e parâmetros para avaliação do processamento dos áudios

Biblioteca	Parâmetros	Valor	Descrição
<i>auditok</i>	<i>min_dur</i>	0.2	mínima duração dos áudios criados (em segundos)
	<i>max_dur</i>	20	máxima duração dos áudios criados (em segundos)
	<i>max_silence</i>	0.3	máxima duração de silêncio tolerado dentro dos áudios criados (em segundos)
	<i>energy_threshold</i>	30, 35, 40, 45, 50 e 55	limiar de energia para detecção de atividade de áudio
<i>webrtcvad</i>	<i>aggressiveness</i>	0, 1, 2 e 3	agressividade da remoção do silêncio
<i>pyAudioAnalysis</i>	<i>st_win</i> e <i>st_step</i>	0.05	tamanho e passo da janela de curto prazo
	<i>smooth_window</i>	0.3 e 0.5	janela (em segundos) usada para suavizar a sequência probabilística do algoritmo
	<i>weight</i>	0.1, 0.2 e 0.3	fator entre 0 e 1 que especifica quão rigoroso é o limiar
	<i>plot</i>	False	booleano associado ao plotagem dos resultados

Fonte: Elaboração própria (2021).

Após a seleção das bibliotecas e parâmetros, foi criado dezesseis *datasets* com características diferentes para a realização da transcrição e cálculo do WER e KER. A Figura B1 exibe os dados normalizados de cada *dataset* de áudios criado, separados por biblioteca usada.

Figura B1 – Características dos *datasets* criados com cada bibliotecas e variando os parâmetros

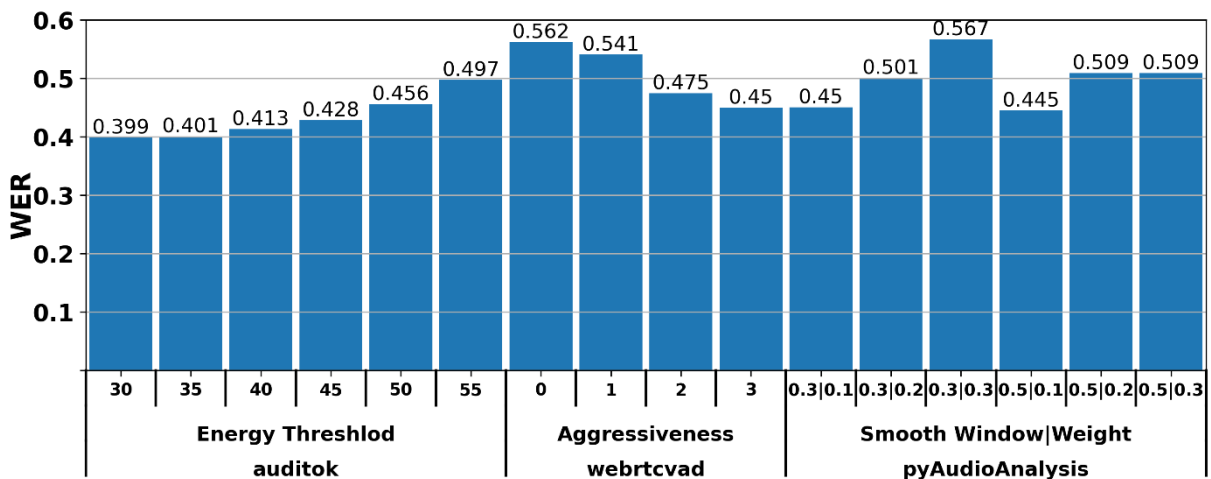


Fonte: Elaboração própria (2021).

Com a variação dos parâmetros, cada grupo de áudios criado tem características distintas. É válido pontuar que para a biblioteca *auditok*, graças ao parâmetro que limita o tempo do áudio segmentado, nenhum áudio ultrapassa o limite de 20 segundos imposto. Isso é atrativo para a análise realizada pois quando os áudios têm mais de 20 segundo a API de transcrição ignora o áudio, ou seja, essa característica representa uma perda de informação direta.

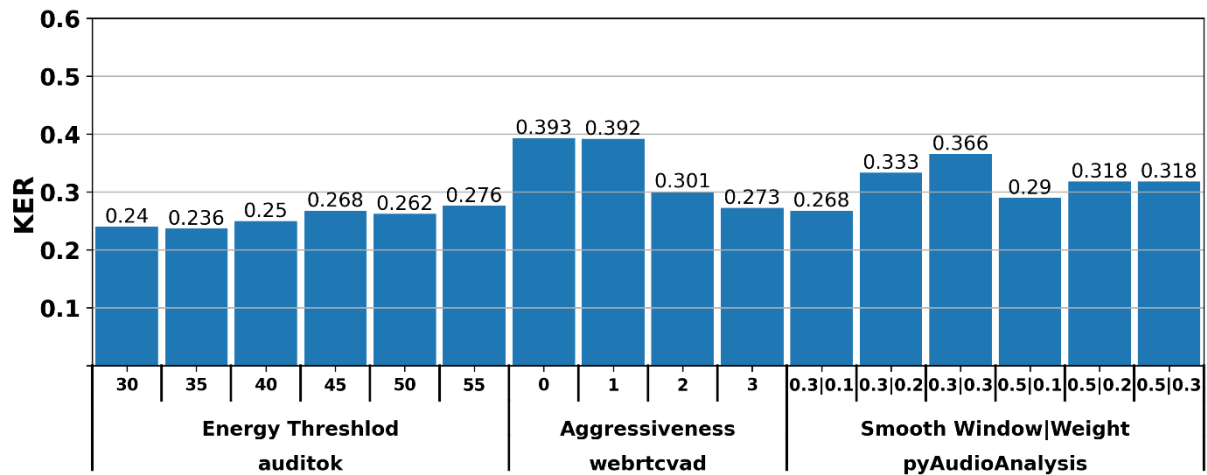
Após a transcrição dos áudios foi possível realizar o cálculo das métricas WER e KER para cada *dataset* criado. A Figura B2 e B3 exibem os valores de WER e KER, respectivamente, para cada parâmetro selecionado.

Figura B2 – WER médio para todas as bibliotecas e variações de parâmetros



Fonte: Elaboração própria (2021).

Figura B3 – KER médio para todas as bibliotecas e variações de parâmetros



Fonte: Elaboração própria (2021).

Observa-se que, dentre as segmentações realizadas, a mais atrativa é com a biblioteca *auditok* com o parâmetro *energy_threshold* igual a 30 para a métrica WER (0,399) e para a métrica KER o *energy_threshold* que atingiu o menor valor é igual a 35 (0,236). Nota-se que os valores de 30 e 35 são similares, logo supõem-se que a partir do *threshold* igual a 35 não existe grandes ganhos de informação.

Conclui-se que o *auditok* é uma biblioteca atrativa pois apresenta maior flexibilidade na escolha dos parâmetros. Para a metodologia proposta (transcrição com o *Wit.ai*) o tempo máximo de 20 segundos foi decisivo pois nenhuma informação foi perdida em relação a essa característica com o *auditok* e em geral quando menor o limiar de segmentação melhor, pois menos informação é perdida.

APÊNDICE C – DICIONÁRIO EM PYTHON PARA PROCESSAMENTO DAS TRANSCRIÇÕES

Para a realização do pós-processamento das transcrições foi desenvolvido o seguinte dicionário em Python:

```
{'nf':'ons','oms':'ons','vns':'ons','ams':'ons','on':'ons','ms':'ons','onf':'ons','onieci':'ons','ns':'ons','ans':'ons','om':'ons','n
sl':'ons','onf':'ons','am':'ons','nsc':'ons','onfl':'ons','isl':'ons','uns':'ons','ans':'ons','alegria':'alegrete','cia':'cag','ciage':'c
ag','caged':'cag','cg':'cag','ciag':'cag','idiota':'candiotia','iguacu':'biguacu','ibiguacu':'biguacu','piguacu':'biguacu','carn
eiro':'canet','canego':'canet','carne':'canet','canetal':'canet','carne':'canet','canetal':'canet','candeia':'canet','caneta':'
canet','kanessenca':'canet','canella':'canet','caneide':'canet','cania':'canet','itaberato':'ita','itaberate':'ita','itaberati':'ita
','itaipasso':'ita','itaipassu':'ita','imbirucu':'imbirussu','capacito':'capacitor','capacitore':'capacitor','carga':'carga','emc
erro':'cerro','sermo':'cerro','comecar':'comeca','comecamos':'comeca','comecam':'comeca','comecando':'comeca','co
mecado':'comeca','comecaram':'comeca','comecarem':'comeca','comecamos':'comeca','compensou':'compensa','c
ompense':'compensa','compenso':'compensa','pensa':'compensa','pensar':'compensa','pensando':'compensa','co
mpensando':'compensa','compensa':'compensa','compensar':'compensa','compensado':'compensa','compensamo
':'compensa','compensei':'compensa','compensava':'compensa','compensada':'compensa','compensabilidade':'co
mpensa','itacompensa':'compensa','compensacao':'compensa','compensao':'compensa','recompensa':'compensa',
'compensalidade':'compensa','recompensar':'compensa','compensaram':'compensa','compensam':'compensa','de
scompensar':'compensa','recompensando':'compensa','compensamento':'compensa','unicompensando':'compens
a','compensarem':'compensa','condensador':'compensador','condensadore':'compensador','compensadore':'comp
ensador','compensadora':'compensador','compressador':'compensador','compressadore':'compensador','recompe
nsador':'compensador','descompensadore':'compensador','compresso':'compensador','computador':'compensador
','pensador':'compensador','descompensador':'compensador','recomposta':'recomp','recomposicao':'recomp','reco
mposto':'recomp','recompor':'recomp','recompondo':'recomp','recompomo':'recomp','recompo':'recomp','recompu':'r
ecomp','recompa':'recomp','recompar':'recomp','computacao':'comuta','mutacao':'comuta','contar':'comutar','montar
':'comutar','convertido':'convert','converti':'convert','convertida':'convert','convertemo':'convert','converte':'convert','c
onverter':'convert','convertimento':'convert','converteu':'convert','convertendo':'convert','converso':'convert','convert
am':'convert','elevamos':'elev','levamos':'elev','levamos':'elev','elevamos':'elev','elevamos':'elev','leve':'elev','lev
a':'elev','levamos':'elev','levar':'elev','levou':'elev','levamos':'elev','elevamos':'elev','elevamos':'elev','elevamos':'elev','heleval':'
elev','estatica':'estatico','oracao':'geracao','kvs':'kv','cabeca':'kv','cafe':'kv','caver':'kv','caverna':'kv','caveira':'kv','cav
eia':'kv','caveio':'kv','caverna':'kv','cavinha':'kv','caverreiro':'kv','cadeira':'kv','caven':'kv','caminho':'kv','escavez':'kv',
'cadeia':'kv','cabeca':'kv','cavete':'kv','cabine':'kv','cave':'kv','cavalo':'kv','cabere':'kv','craver':'kv','caverno':'kv','caveira
o':'kv','gaveta':'kv','cavei':'kv','massambara':'macambara','matambara':'macambara','sambara':'macambara','camba
ra':'macambara','ambara':'macambara','tambara':'macambara','machada':'machadinho','macha':'machadinho','mac
hadoo':'machadinho','machambara':'machadinho','machadeiro':'machadinho','itamachado':'machadinho','machaba
ra':'machadinho','machado':'machadinho','marchadinho':'machadinho','achadinho':'machadinho','machadinha':'ma
chadinho','achadinha':'machadinho','fachadinha':'machadinho','rachadinha':'machadinho','mameleiro':'marmeleiro',
'maneleiro':'marmeleiro','maveleiro':'marmeleiro','paneleiro':'marmeleiro','mineiro':'marmeleiro','marmiteiro':'marmel
eiro','monjolo':'monjolinho','monjola':'monjolinho','monjole':'monjolinho','monjoli':'monjolinho','monjolina':'monjolinh
o','monjolino':'monjolinho','movimentacao':'movimentacao','mergulhote':'mw','negocio':'mw','negocio':'mw','lote':'m
w','meca':'mw','macho':'mw','megaway':'mw','megath':'mw','megathe':'mw','megawarra':'mw','melgote':'mw','megaw
att':'mw','megote':'mw','magote':'mw','meoria':'mw','megalota':'mw','megalote':'mw','negoto':'mw','megotico':'mw','m
egawattpri':'mw','megaware':'mw','megagote':'mw','megagotel':'mw','megagoto':'mw','megawayt':'mw','lainei':'ney','
```

deneý': 'ney', 'ozorio': 'osorio', 'ozonio': 'osorio', 'ovario': 'osorio', 'aosorio': 'osorio', 'posorio': 'osorio', 'tosorio': 'osorio', 'proso
 rio': 'osorio', 'pampacu': 'pampa', 'parigotto': 'parigot', 'parigot': 'parigot', 'parigor': 'parigot', 'parigo': 'parigot', 'parigoa': 'parig
 ot', 'parigoto': 'parigot', 'parigotso': 'parigot', 'parigu': 'parigot', 'parigut': 'parigot', 'parigua': 'parigot', 'parigor': 'parigot', 'parig
 o': 'parigot', 'parigoa': 'parigot', 'perigon': 'parigot', 'perigot': 'parigot', 'perigou': 'parigot', 'sideropoli': 'sideropolis', 'fideropoli'
 ': 'sideropolis', 'hideropoli': 'sideropolis', 'paco': 'passo', 'passa': 'passo', 'pilarzo': 'pilarzinho', 'pinhalzo': 'pinhalzinho', 'trator'
 ': 'reator', 'resto': 'reator', 'relator': 'reator', 'retou': 'reator', 'reto': 'reator', 'retor': 'reator', 'retro': 'reator', 'ator': 'reator', 'revestida'
 ': 'revertid', 'overtido': 'revertid', 'avertido': 'revertid', 'descarregada': 'descarregad', 'descarregado': 'descarregad', 'descarr
 egador': 'descarregad', 'estacionador': 'estacionad', 'estacionada': 'estacionad', 'estacionadore': 'estacionad', 'estaciona
 da': 'estacionad', 'rive': 'riveira', 'plantiago': 'santiago', 'antiago': 'santiago', 'sexadora': 'seccionador', 'sexadore': 'secciona
 dor', 'seccionadore': 'seccionador', 'teccionadora': 'seccionador', 'confeccionadora': 'seccionador', 'infeccionadora': 'sec
 cionador', 'infeccionadore': 'seccionador', 'flexionadora': 'seccionador', 'flexionador': 'seccionador', 'flexionadura': 'secci
 onador', 'flexionadore': 'seccionador', 'restabelecida': 'restabelecid', 'reestabelecida': 'restabelecid', 'restabelecer': 'resta
 belec', 'estabelecer': 'estabelec', 'estabeleceu': 'estabelec', 'fgi': 'sgi', 'sge': 'sgi', 'fg': 'sgi', 'sgei': 'sgi', 'gi': 'sgi', 'sg': 'sgi', 'ufgi': 's
 gi', 'usb': 'sgi', 'csg': 'sgi', 'usg': 'sgi', 'usgi': 'sgi', 'cgi': 'sgi', 'sci': 'sgi', 'ciclizar': 'sincroniz', 'secronizada': 'sincroniz', 'sicroniza
 da': 'sincroniz', 'sicronia': 'sincroniz', 'sincronizou': 'sincroniz', 'sincronizado': 'sincroniz', 'sincronizada': 'sincroniz', 'sincron
 izar': 'sincroniz', 'sincronizei': 'sincroniz', 'sincroniza': 'sincroniz', 'sincronizando': 'sincroniz', 'sincronizo': 'sincroniz', 'sincr
 onizaria': 'sincroniz', 'sincronizacao': 'sincroniz', 'sincronizarmo': 'sincroniz', 'sincronizamo': 'sincroniz', 'sincronizaram': 'si
 ncroniz', 'sincronizam': 'sincroniz', 'sincronizadora': 'sincroniz', 'sincronize': 'sincroniz', 'sincronia': 'sincroniz', 'sincrono': 's
 incroniz', 'sincroni': 'sincroniz', 'charlauw': 'charlau', 'charlal': 'charlau', 'charlall': 'charlau', 'sharlal': 'charlau', 'sharlau': 'charl
 au', 'charlauk': 'charlau', 'charlaun': 'charlau', 'charlau': 'charlau', 'charlal': 'charlau', 'charlall': 'charlau', 'charlao': 'charlau', 's
 harlao': 'charlau', 'sharla': 'charlau', 'dap': 'tap', 'tapinha': 'tap', 'tapa': 'tap', 'tata': 'tap', 'taxi': 'tap', 'tape': 'tap', 'top': 'tap', 'atenca
 o': 'tensao', 'intencao': 'tensao', 'pensao': 'tensao', 'som': 'tensao', 'topzinho': 'top', 'tr1': 'transf', 'trafico': 'transf', 'trafego': 'tran
 sf', 'transformadore': 'transf', 'transformadora': 'transf', 'transformador': 'transf', 'humanidade': 'unidade', 'umidade': 'unida
 de', 'im unidade': 'unidade', 'cadusina': 'usina', 'telhado': 'utla', 'telhar': 'utla', 'telha': 'utla', 'tla': 'utla', 'anauteliante': 'utla', 'gol':
 'volt', 'solto': 'volt', 'voo': 'volt', 'voto': 'volt', 'voto': 'volt', 'traduzir': 'reduz', 'reduzisse': 'reduz', 'reduziuse': 'reduz', 'pre reduzida':
 'reduz', 'pre reduzido': 'reduz', 'reduza': 'reduz', 'reduzam': 'reduz', 'reduzi': 'reduz', 'reduzir': 'reduz', 'reduziu': 'reduz', 'reduzi
 do': 'reduz', 'reduzida': 'reduz', 'reduziram': 'reduz', 'reduzimo': 'reduz', 'reduzia': 'reduz', 'reduziria': 'reduz', 'reduzilo': 'reduz
 ', 'reduzila': 'reduz', 'reduzita': 'reduz', 'seduzir': 'reduz', 'deduzir': 'reduz', 'eduzir': 'reduz', 'elear': 'elev', 'elevator': 'elev', 'rele
 var': 'elev', 'elevaria': 'elev', 'levaria': 'elev', 'elevarse': 'elev', 'elevaram': 'elev', 'elevarmo': 'elev', 'prelevar': 'elev', 'delevar': 'el
 ev', 'elevarem': 'elev', 'levarmo': 'elev', 'elevara': 'elev', 'alevar': 'elev', 'aumentar': 'aument', 'aumentarse': 'aument', 'aument
 ou': 'aument', 'aumenta': 'aument', 'aumentando': 'aument', 'aumento': 'aument', 'aumentaram': 'aument', 'aumentaria': 'au
 ment', 'aumente': 'aument', 'aumentacao': 'aument', 'aumentozo': 'aument', 'aumentado': 'aument', 'diminuida': 'diminui', 'di
 minuir': 'diminui', 'diminuindo': 'diminui', 'diminui': 'diminui', 'diminuiu': 'diminui', 'diminuiam': 'diminui', 'diminua': 'diminui', 'd
 iminuo': 'diminui', 'abaixa': 'abaix', 'abaixada': 'abaix', 'abaixadinho': 'abaix', 'abaixado': 'abaix', 'abaixando': 'abaix', 'abaixar
 ': 'abaix', 'abaixo': 'abaix', 'abaixou': 'abaix', 'rebaixado': 'reabaix', 'rebaixando': 'reabaix', 'solic': 'solicit', 'solicitar': 'solicit', 's
 olicitou': 'solicit', 'solicitado': 'solicit', 'solicita': 'solicit', 'solicitei': 'solicit', 'solicitada': 'solicit', 'solicito': 'solicit', 'solicitam': 'solic
 it', 'solicitamo': 'solicit', 'solicite': 'solicit', 'solicitalo': 'solicit', 'solicitase': 'solicit', 'solicitem': 'solicit', 'solicitar': 'solicit', 'nor
 matizacao': 'normaliz', 'normalizar': 'normaliz', 'normalizou': 'normaliz', 'normalizada': 'normaliz', 'formalizar': 'normaliz', 'fo
 rmalizando': 'normaliz', 'normalizando': 'normaliz', 'normalizado': 'normaliz', 'normalizaram': 'normaliz', 'anormalizado': 'n
 ormaliz', 'normalizacao': 'normaliz', 'normalizo': 'normaliz', 'normalize': 'normaliz', 'normaliza': 'normaliz', 'normalizei': 'nor
 maliz', 'formalizado': 'normaliz', 'mormalizou': 'normaliz', 'formalizou': 'normaliz', 'anormalizar': 'normaliz', 'normalizamo':
 'normaliz', 'normalizarmo': 'normaliz', 'restricao': 'retricao', 'reverter': 'reverte', 'reverteu': 'reverte', 'reverti': 'reverte', 'revert
 o': 'reverte', 'revertendo': 'reverte', 'severtendo': 'reverte', 'revertem': 'reverte', 'considerar': 'consider', 'considerou': 'conside
 r', 'considerado': 'consider', 'considerando': 'consider', 'consideramo': 'consider', 'considera': 'consider', 'considerarei': 'cons
 ider', 'considerada': 'consider', 'considero': 'consider', 'consideraram': 'consider', 'considere': 'consider', 'consideram': 'con

sider', 'considerador': 'consider', 'desconsiderando': 'desconsider', 'desconsiderou': 'desconsider', 'desconsiderar': 'desconsider', 'desconsidero': 'desconsider', 'desconsidera': 'desconsider', 'desconsiderarei': 'desconsider', 'normalizar': 'normaliz', 'normalizou': 'normaliz', 'normalizada': 'normaliz', 'normalizando': 'normaliz', 'normalizado': 'normaliz', 'normalizaram': 'normaliz', 'anormalizado': 'normaliz', 'normalizacao': 'normaliz', 'normalizo': 'normaliz', 'normalize': 'normaliz', 'normaliza': 'normaliz', 'normalizei': 'normaliz', 'anormalizar': 'normaliz', 'normalizamo': 'normaliz', 'normalizarmo': 'normaliz', 'normalizou': 'normaliz', 'absorvendo': 'absorve', 'absorver': 'absorve', 'absorvelo': 'absorve', 'absorva': 'absorve', 'absorve': 'absorve', 'absorveu': 'absorve', 'absorvi': 'absorve', 'gravata': 'gravatai', 'gravataista': 'gravatai', 'gravatar': 'gravatai', 'gravatei': 'gravatai', 'engravata': 'gravatai', 'gravato': 'gravatai', 'encerrado': 'encerra', 'encerrada': 'encerra', 'encerrando': 'encerra', 'encerrar': 'encerra', 'encerram': 'encerra', 'encerramento': 'encerra', 'encerraram': 'encerra', 'encerraria': 'encerra', 'encerravamse': 'encerra', 'encerrava': 'encerra', 'encerramo': 'encerra', 'encerrarem': 'encerra', 'encerrarmo': 'encerra', 'encerralo': 'encerra', 'antecipar': 'antecip', 'antecipando': 'antecip', 'antecipe': 'antecip', 'antecipado': 'antecip', 'antecipai': 'antecip', 'antecipadamente': 'antecip', 'anteciparam': 'antecip', 'antecipacao': 'antecip', 'antecipou': 'antecip', 'antecipada': 'antecip', 'programaram': 'program', 'programarem': 'program', 'programar': 'program', 'reprogramar': 'reprogram', 'reprograma': 'reprogram', 'reprograme': 'reprogram', 'prorrogar': 'prorrog', 'prorrogacao': 'prorrog', 'prorrogando': 'prorrog', 'prorrogaram': 'prorrog', 'prorrogado': 'prorrog', 'prorrogada': 'prorrog', 'prorrogalo': 'prorrog', 'prorrogaria': 'prorrog', 'prorroguei': 'prorrog', 'prorroga': 'prorrog', 'prorrogamo': 'prorrog', 'prorrogo': 'prorrog', 'prorrogavel': 'prorrog', 'jogar': 'prorrog', 'concluir': 'concl', 'conclusao': 'concl', 'concluida': 'concl', 'concluindo': 'concl', 'concluido': 'concl', 'concluiu': 'concl', 'conclu': 'concl', 'conclui': 'concl', 'concluirem': 'concl', 'concluirem': 'concl', 'concluimo': 'concl', 'concluo': 'concl', 'conclui': 'concl', 'desconcluir': 'concl', 'concluidamente': 'concl', 'terminando': 'termin', 'termino': 'termin', 'terminou': 'termin', 'terminal': 'termin', 'terminar': 'termin', 'terminaram': 'termin', 'termina': 'termin', 'terminei': 'termin', 'terminada': 'termin', 'terminado': 'termin', 'terminava': 'termin', 'terminai': 'termin', 'terminasse': 'termin', 'termine': 'termin', 'terminante': 'termin', 'finalizando': 'finaliz', 'finalizou': 'finaliz', 'finalizada': 'finaliz', 'finalizar': 'finaliz', 'finalizacao': 'finaliz', 'finalizo': 'finaliz', 'finalizamo': 'finaliz', 'finalizaram': 'finaliz', 'finalizado': 'finaliz', 'aprova': 'aprov', 'aprovar': 'aprov', 'aprovacao': 'aprov', 'aprovaram': 'aprov', 'aprovou': 'aprov', 'aprovando': 'aprov', 'aprovado': 'aprov', 'aprovarem': 'aprov', 'aprovada': 'aprov', 'aprove': 'aprov', 'aprovo': 'aprov', 'aprovamo': 'aprov', 'aprovei': 'aprov', 'aprovam': 'aprov', 'manobra': 'manobr', 'manobrinha': 'manobr', 'manobrando': 'manobr', 'manobrado': 'manobr', 'manobrar': 'manobr', 'manobrada': 'manobr', 'manobrava': 'manobr', 'manobram': 'manobr', 'manobrai': 'manobr', 'manobrei': 'manobr', 'manobrar': 'manobr', 'manobrava': 'manobr', 'manobram': 'manobr', 'manobrou': 'manobr', 'manobraram': 'manobr', 'banca': 'banco', 'pastore': 'capacitor', 'capacitador': 'capacitor', 'capacore': 'capacitor', 'capacote': 'capacitor', 'capacitou': 'capacitor', 'capacitada': 'capacitor', 'capacitar': 'capacitor', 'capacite': 'capacitor', 'capacitacao': 'capacitor', 'capacitadore': 'capacitor', 'capacitacao': 'capacitor', 'capacitado': 'capacitor', 'capacitando': 'capacitor', 'capacitora': 'capacitor', 'capacitoria': 'capacitor', 'capacitivo': 'capacitor', 'capacita': 'capacitor', 'capacitura': 'capacitor', 'capacitao': 'capacitor', 'capacitorzinho': 'capacitor', 'capacitante': 'capacitor', 'capacitone': 'capacitor', 'compastor': 'capacitor', 'reatore': 'reator', 'reatora': 'reator', 'reatorzinho': 'reator', 'reatou': 'reator', 'reatoruna': 'reator', 'reatorio': 'reator', 'reatoru': 'reator', 'reatoriamente': 'reator', 'reatonico': 'reator', 'ligarreator': 'reator', 'ligarreador': 'reator', 'isoladore': 'isolador', 'isoladora': 'isolador', 'conversora': 'conversor', 'conversore': 'conversor', 'reativo': 'reativ', 'reativa': 'reativ', 'reative': 'reativ', 'invertedor': 'vertedor', 'vertedora': 'vertedor', 'vertedoro': 'vertedor', 'invertedouro': 'vertedor', 'invertedora': 'vertedor', 'vertedouro': 'vertedor', 'devertedor': 'vertedor', 'invertedore': 'vertedor', 'vertedou': 'vertedor', 'revertedora': 'vertedor', 'advertedora': 'vertedor', 'divertedora': 'vertedor', 'divertedouro': 'vertedor', 'vertedo': 'vertedor', 'invertedo': 'vertedor', 'invertedor': 'vertedor', 'invertedouro': 'vertedor', 'invertedora': 'vertedor', 'invertedore': 'vertedor', 'vertida': 'vertid', 'avertida': 'vertid', 'vertelho': 'vertid', 'vertilha': 'vertid', 'vertilhar': 'vertid', 'reinvert': 'invert', 'invertiram': 'invert', 'invertendo': 'invert', 'invertiu': 'invert', 'invert': 'invert', 'inverte': 'invert', 'hidreletrica': 'hidroeletric', 'hidreletrico': 'hidroeletric', 'hidroeletrica': 'hidroeletric', 'hidroeletrico': 'hidroeletric', 'registrou': 'registr', 'registrar': 'registr', 'registrado': 'registr', 'registrando': 'registr', 'registro': 'registr', 'registraram': 'registr', 'registrada': 'registr', 'registre': 'registr', 'registra': 'registr', 'registrador': 'registr', 'registramo': 'registr', 'autorizado': 'autoriz', 'autorizacao': 'autoriz', 'autorizar': 'autoriz', 'autorizando': 'autoriz', 'autoriza': 'autoriz', 'autorizo': 'autoriz', 'autorizada': 'autoriz', 'autorize': 'autoriz', 'autorizou': 'autoriz', 'autorizei': 'autoriz', 'autorizaram': 'autoriz', 'auto

rizamo:'autoriz','autorizarem':'autoriz','autorizam':'autoriz','autorizacoes':'autoriz','autorizala':'autoriz','autorizariam':
 'autoriz','disjuntou':'disjuntor','disjuntor':'disjuntor','disjuntore':'disjuntor','executar':'execut','executando':'execut','exec
 utado':'execut','executada':'execut','executo':'execut','executa':'execut','executante':'execut','executamo':'execut','e
 xecutou':'execut','executaram':'execut','executei':'execut','executivo':'execut','recomposta':'recomp','recomposicao':
 'recomp','recomposto':'recomp','recompor':'recomp','recompondo':'recomp','recompomo':'recomp','recompo':'reco
 mp','recompu':'recomp','recompa':'recomp','recompar':'recomp','transferir':'transfer','transferencia':'transfer','transfe
 rida':'transfer','transfer':'transfer','transferi':'transfer','transferido':'transfer','transferindo':'transfer','transferem':'transf
 er','transferimo':'transfer','transferiu':'transfer','transfere':'transfer','transferirem':'transfer','remanejando':'remanej','r
 emanejar':'remanej','remanejamento':'remanej','remanejamo':'remanej','remanejado':'remanej','remanejo':'remanej'
 ,'remanejada':'remanej','remaneja':'remanej','desligar':'deslig','desliga':'deslig','desligando':'deslig','desligamento':'d
 eslig','desligou':'deslig','desligado':'deslig','desligada':'deslig','desliguei':'deslig','desligaram':'deslig','desligo':'deslig'
 ,'desligue':'deslig','desligarquia':'deslig','desligam':'deslig','desligava':'deslig','desligamo':'deslig','desligarem':'deslig'
 ,'desligarei':'deslig','desligasse':'deslig','desligalo':'deslig','desligarte':'deslig','ligada':'lig','ligadao':'lig','ligadinha':'lig','
 ligado':'lig','ligador':'lig','ligadose':'lig','ligagem':'lig','ligalo':'lig','ligam':'lig','ligamento':'lig','ligamo':'lig','ligando':'lig','lig
 ante':'lig','ligar':'lig','ligaram':'lig','ligarem':'lig','ligaria':'lig','ligaru':'lig','ligasse':'lig','ligava':'lig','ligo':'lig','liga':'lig','ligoe':
 'lig','ligou':'lig','ligue':'lig','liguei':'lig','religa':'relig','religada':'relig','religado':'relig','religador':'relig','religadore':'relig','rel
 igala':'relig','religamento':'relig','religando':'relig','religar':'relig','religassem':'relig','religou':'relig','religui':'relig','interli
 ga':'interlig','interligacao':'interlig','interligada':'interlig','interligado':'interlig','interligador':'interlig','interligando':'interli
 g','interligar':'interlig','desinteligada':'desinterlig','desinteligar':'desinterlig','desinteligavel':'desinterlig','desinteligad
 a':'desinterlig','desinteligou':'desinterlig','desinterligada':'desinterlig','desinterligado':'desinterlig','cancelar':'cancel','c
 ANCELADA':'cancel','cancelado':'cancel','cancela':'cancel','cancelando':'cancel','cancelo':'cancel','cancelou':'cancel','
 descancelado':'cancel','cancelamento':'cancel','cancelador':'cancel','compartilhar':'compartilh','compartilha':'compa
 rtilh','compartilho':'compartilh','coordena':'coorden','coordenacao':'coorden','coordenada':'coorden','coordenado':'c
 oorden','coordenador':'coorden','coordenam':'coorden','coordenando':'coorden','coordenar':'coorden','coordenava':
 'coorden','coordenei':'coorden','coordeno':'coorden','coordenou':'coorden','descoordenacao':'descoorden','descoor
 denar':'descoorden','trezento':'trezent','trezenta':'trezent','trezentu':'trezent','seiscenta':'seiscent','seiscento':'seisce
 nt','seiscentu':'seiscent','distribuidora':'distribuidor','distribuidor':'distribuidor','distribuidore':'distribuidor','disponibiliz
 ando':'disponibiliz','disponibilizar':'disponibiliz','disponibilizacao':'disponibiliz','disponibilizado':'disponibiliz','disponib
 ilizou':'disponibiliz','disponibilizaram':'disponibiliz','disponibilizada':'disponibiliz','disponibiliza':'disponibiliz','disponibi
 lizasse':'disponibiliz','disponibilizei':'disponibiliz','disponibilizo':'disponibiliz','disponibilizava':'disponibiliz','disponibili
 zarem':'disponibiliz','disponibilizador':'disponibiliz','indisponibilizando':'indisponibiliz','indisponibilizar':'indisponibiliz',
 'indisponibilidade':'indisponibiliz','indisponivei':'indisponivel','restabelecida':'restabelecid','restabelecido':'restabelec
 id','conjunto':'conjunt','conjunta':'conjunt','maximizada':'maximizad','maximizado':'maximizad','atendendo':'atend','at
 ende':'atend','atender':'atend','atendia':'atend','atendida':'atend','atendelo':'atend','atendido':'atend','atenderam':'ate
 nd','atenderia':'atend','atendo':'atend','atendela':'atend','atendeu':'atend','atendimento':'atend','atendi':'atend','atend
 esse':'atend','atendese':'atend','atendemo':'atend','atendem':'atend','companheiro':'companheir','companheira':'co
 mpanheir','controle':'control','controlar':'control','controlando':'control','controlam':'control','controla':'control','control
 encia':'control','controlo':'control','atipica':'atipic','atipico':'atipic','servico':'servic','servica':'servic','habilitado':'habilita
 d','habilitada':'habilitad','sistemico':'sistemic','sistemica':'sistemic','retorno':'retorn','retornar':'retorn','retorna':'retorn',
 retornou':'retorn','retornamo':'retorn','retornando':'retorn','retornei':'retorn','retornasse':'retorn','retornaram':'retorn','r
 etornarmo':'retorn','retornarem':'retorn','retorne':'retorn','retorninho':'retorn','retornamente':'retorn','retornado':'retorn
 ','retornarei':'retorn','retornam':'retorn','retornaria':'retorn','retornem':'retorn','retornada':'retorn','nenhum':'nenhu','ne
 nhuma':'nenhu','abria':'abri','abrigando':'abri','abrigo':'abri','abril':'abri','abrilha':'abri','abrilho':'abri','abrimo':'abri','abrind
 o':'abri','abrir':'abri','abriram':'abri','abrirem':'abri','abriria':'abri','abriria-se':'abri','abrisse':'abri','abriu':'abri','fechada':'fe
 ch','fechadeira':'fech','fechado':'fech','fechadu':'fech','fechalo':'fech','fecham':'fech','fechamento':'fech','fechamo':'fec

h', 'fechando': 'fech', 'fechar': 'fech', 'fecharam': 'fech', 'fecharem': 'fech', 'fecharia': 'fech', 'fechasse': 'fech', 'fechava': 'fech', 'feche': 'fech', 'fechei': 'fech', 'fecho': 'fech', 'fecha': 'fech', 'fechou': 'fech', 'desenergizada': 'desenergizad', 'desenergizado': 'desenergizad', 'energizado': 'energizad', 'energizando': 'energizad', 'energizamo': 'energizad', 'acompanhar': 'acompanh', 'acompanha': 'acompanh', 'acompanhe': 'acompanh', 'acompanho': 'acompanh', 'acompanham': 'acompanh', 'confirmar': 'confirm', 'confirma': 'confirm', 'confirmo': 'confirm', 'confirme': 'confirm', 'confirmam': 'confirm', 'confirmaram': 'confirm', 'confirmarem': 'confirm', 'aguardo': 'aguard', 'aguardar': 'aguard', 'aguarda': 'aguard', 'aguarde': 'aguard', 'aguardam': 'aguard', 'desarme': 'desarm', 'desarma': 'desarm', 'desarmar': 'desarm', 'desarmo': 'desarm', 'desarmal': 'desarm', 'desarmado': 'desarm', 'desarmando': 'desarm', 'normativa': 'normativ', 'normativo': 'normativ', 'entendi': 'entend', 'entendeu': 'entend', 'entender': 'entend', 'entende': 'entend', 'entendo': 'entend', 'entendem': 'entend', 'permanece': 'permanec', 'permanecer': 'permanec', 'permanec': 'permanec', 'permaneceu': 'permanec', 'permanecem': 'permanec', 'permaneci': 'permanec', 'cadastrar': 'cadastr', 'cadastra': 'cadastr', 'cadastre': 'cadastr', 'cadastrado': 'cadastr', 'cadastrada': 'cadastr', 'cadastrando': 'cadastr', 'cadastramo': 'cadastr', 'cadastrai': 'cadastr', 'cadastrou': 'cadastr', 'interromper': 'interromp', 'interrompeu': 'interromp', 'interrompe': 'interromp', 'interrompi': 'interromp', 'interrompida': 'interromp', 'interrompeda': 'interromp', 'interrompela': 'interromp', 'desarme': 'desarm', 'desarmo': 'desarm', 'desarme': 'desarm', 'desarmen': 'desarm', 'maximizar': 'maximiz', 'maximizad': 'maximiz', 'maximiza': 'maximiz', 'maximize': 'maximiz', 'conectar': 'conect', 'conecta': 'conect', 'conectam': 'conect', 'conecte': 'conect', 'conectado': 'conect', 'conectando': 'conect', 'conectamo': 'conect', 'informa': 'inform', 'informar': 'inform', 'informa': 'inform', 'informo': 'inform', 'informe': 'inform', 'informei': 'inform', 'informaco': 'inform', 'informacao': 'inform', 'informar': 'inform', 'informalo': 'inform', 'informa': 'inform', 'informou': 'inform', 'informe': 'inform', 'informando': 'inform', 'informaram': 'inform', 'informarei': 'inform', 'informei': 'inform', 'informativo': 'inform', 'informo': 'inform', 'informasse': 'inform', 'informata': 'inform', 'informava': 'inform', 'informacaozia': 'inform', 'informalmente': 'inform', 'informado': 'inform', 'informada': 'inform', 'precisarem': 'precisa', 'precisaram': 'precisa', 'precisariam': 'precisa', 'precisavam': 'precisa', 'verificar': 'verific', 'verifica': 'verific', 'verifico': 'verific', 'verificam': 'verific', 'verificando': 'verific', 'verificado': 'verific', 'verificamo': 'verific', 'necessita': 'necessit', 'necessitar': 'necessit', 'necessitam': 'necessit', 'necessite': 'necessit', 'necessitado': 'necessit', 'necessitando': 'necessit', 'completar': 'complet', 'completa': 'complet', 'completo': 'complet', 'pergunta': 'pergunt', 'perguntar': 'pergunt', 'pergunto': 'pergunt', 'pergunte': 'pergunt', 'repassar': 'repass', 'repasso': 'repass', 'repassa': 'repass', 'repass': 'repass', 'providenciar': 'providenc', 'providencia': 'providenc', 'providencie': 'providenc', 'providencio': 'providenc', 'complementar': 'complement', 'complemento': 'complement', 'complemente': 'complement', 'complementa': 'complement', 'substituir': 'substit', 'substitui': 'substit', 'substituiu': 'substit', 'programa': 'program', 'programar': 'program', 'programo': 'program', 'deslocar': 'desloc', 'desloca': 'desloc', 'desloco': 'desloc', 'uruguaina': 'uruguaiiana', 'uruguaiano': 'uruguaiiana', 'uruguai': 'uruguaiiana', 'ultrapassar': 'ultrapas', 'ultrapassa': 'ultrapas', 'ultrapassam': 'ultrapas', 'j'eribatu': 'geribatu', 'geribatur': 'geribatu', 'j'eribatu': 'geribatu', 'acrescentar': 'acrescent', 'acrescenta': 'acrescent', 'acrescente': 'acrescent', 'declarando': 'declara', 'declarado': 'declara', 'declaramo': 'declara', 'horaria': 'horario', 'horariu': 'horario', 'reenviar': 'reenvi', 'reenvio': 'reenvi', 'reenvia': 'reenvi', 'colocando': 'coloca', 'colocado': 'coloca', 'colocamo': 'coloca', 'colocava': 'coloca', 'colocada': 'coloca', 'calculo': 'calcul', 'calcular': 'calcul', 'calcula': 'calcul', 'supervisor': 'supervis', 'supervisora': 'supervis', 'supervisor': 'supervis', 'supervisorio': 'supervis', 'geradora': 'gerador', 'geradore': 'gerador', 'comando': 'comand', 'comanda': 'comand', 'comandar': 'comand', 'efetivo': 'efetiv', 'efetiva': 'efetiv', 'efetivar': 'efetiv', 'realizando': 'realiz', 'realizado': 'realiz', 'realizamo': 'realiz', 'sinalizando': 'sinaliz', 'sinalizado': 'sinaliz', 'sinalizador': 'sinaliz', 'ajustando': 'ajust', 'ajustado': 'ajust', 'ajustamo': 'ajust', 'sincronismo': 'sincroni', 'sincronismo': 'sincroni', 'sincronismo': 'sincroni', 'sincronizar': 'sincroni', 'sincronizar': 'sincroni', 'comutado': 'comut', 'comutando': 'comut', 'comutador': 'comut', 'iniciado': 'inici', 'iniciada': 'inici', 'iniciala': 'inici', 'iniciar': 'inici', 'inicial': 'inici', 'inicia': 'inici', 'inicio': 'inici', 'iniciou': 'inici', 'iniciaram': 'inici', 'iniciarem': 'inici', 'iniciando': 'inici', 'neolica': 'eolica', 'teolica': 'eolica', 'perguntado': 'pergunt', 'perguntada': 'pergunt', 'perguntava': 'pergunt', 'anastacia': 'anastacio', 'anastasia': 'anastacio', 'excitada': 'excita', 'excitado': 'excita', 'excitava': 'excita', 'excitando': 'excita', 'desbloqueado': 'desbloque', 'desbloqueamo': 'desbloque', 'desbloqueando': 'desbloque', 'desbloqueada': 'desbloque', 'desbloqueio': 'desbloque', 'desbloqueia': 'desbloque', 'bloqueado': 'bloque', 'bloqueando': 'bloque', 'bloqueada': 'bloque', 'excluido': 'exclui', 'excluindo': 'exclui', 'excluir': 'exclui', 'liberando': 'libera', 'liberamo': 'libera', 'maquina': 'maquin', 'maquinas': 'm

aquin', 'maquinao': 'maquin', 'maquinal': 'maquin', 'desenergizar': 'desenergiz', 'desenergizad': 'desenergiz', 'desenergiz
 a': 'desenergiz', 'limitada': 'limita', 'limitado': 'limita', 'limitava': 'limita', 'ratificar': 'ratific', 'ratifica': 'ratific', 'retifica': 'retific', 'barr
 amento': 'barra', 'barramento': 'barra', 'barrado': 'barra', 'desabilitado': 'desabilita', 'desabilitando': 'desabilita', 'desabilitar':
 desabilita', 'consolidado': 'consolidad', 'consolidada': 'consolidad', 'bloqueio': 'bloqueei', 'bloqueia': 'bloqueei', 'aerogerado':
 aerogerador', 'desobstruir': 'desobstrui', 'desobstruiu': 'desobstrui', 'importa': 'import', 'importar': 'import', 'explodiu': 'explo
 di', 'explodir': 'explodi', 'explodido': 'explodi', 'explodiu': 'explodi', 'comuniquei': 'comunique', 'turbinando': 'turbin', 'turbinad
 o': 'turbin', 'paralelo': 'paralel', 'paralela': 'paralel', 'caldeia': 'caldeira', 'vinculada': 'vinculad', 'vinculado': 'vinculad', 'partida':
 partid', 'partido': 'partid', 'bartiguacu': 'iguacu', 'batiguacu': 'iguacu', 'melhorada': 'melhora', 'melhorava': 'melhora', 'validad
 o': 'valida', 'validando': 'valida', 'favorzao': 'favor', 'favorzo': 'favor', 'esgotado': 'esgota', 'esgotamo': 'esgota', 'equivoco': 'equ
 ivoc', 'equivoca': 'equivoc', 'subindo': 'subind', 'subinda': 'subind', 'energizou': 'energiz', 'energizo': 'energiz', 'desviou': 'desv
 io', 'sinaliza': 'sinaliz', 'sinalizo': 'sinaliz', 'sinalizar': 'sinaliz', 'isolado': 'isolad', 'isolada': 'isolad', 'subtencao': 'subtensao', 'pro
 gramazao': 'programacao', 'eletar': 'deletar', 'operativa': 'operativ', 'operativo': 'operativ', 'necessario': 'necessari', 'necess
 aria': 'necessari', 'telecomunica': 'telecomunicacao', 'telecomunicacaoe': 'telecomunicacao', 'fusivei': 'fusivel', 'normalmen
 te': 'normal', 'normalidade': 'anormal', 'anormalidade': 'anormal', 'comunicar': 'comunic', 'comunica': 'comunic', 'comunico':
 'comunic', 'comunicam': 'comunic', 'comunicando': 'comunic', 'comunicado': 'comunic', 'comunicalo': 'comunic', 'comunic
 ou': 'comunic', 'declarar': 'declar', 'declara': 'declar', 'declaro': 'declar', 'declare': 'declar', 'declarada': 'declar', 'declarou': 'dec
 lar', 'devolver': 'devolv', 'devolveu': 'devolv', 'devolve': 'devolv', 'devolvi': 'devolv', 'devolvido': 'devolv', 'devolvida': 'devolv',
 melhorar': 'melhor', 'melhore': 'melhor', 'melhora': 'melhor', 'melhoria': 'melhor', 'aproveitar': 'proveit', 'aproveita': 'proveit',
 'proveito': 'proveit', 'proveite': 'proveit', 'sobreavisa': 'sobreaviso', 'trifasico': 'trifasic', 'trifasica': 'trifasic', 'alimentar': 'al
 iment', 'alimenta': 'aliment', 'alimentam': 'aliment', 'falimenta': 'aliment', 'alimentador': 'aliment', 'alimentando': 'aliment', 'ali
 mentado': 'aliment', 'alimentadore': 'aliment', 'alimentada': 'aliment', 'eletrica': 'eletric', 'eletrico': 'eletric', 'forcado': 'forcad', 'f
 orcada': 'forcad', 'movimentada': 'movimentad', 'movimentado': 'movimentad', 'dequacao': 'equacao', 'adequacao': 'equa
 cao', 'inadequacao': 'equacao', 'inadequacao': 'equacao', 'desconectado': 'desconectad', 'desconectada': 'desconectad',
 confirma': 'confirm', 'confira': 'confirm', 'confirar': 'confirm', 'confiro': 'confirm', 'confir': 'confirm', 'confirmar': 'confirm', 'confir
 mamo': 'confirm', 'confirmam': 'confirm', 'confirmado': 'confirm', 'confirmando': 'confirm', 'confirmou': 'confirm', 'confirmo': 'c
 onfirm', 'confirmei': 'confirm', 'confirme': 'confirm', 'confirmada': 'confirm', 'confirmarei': 'confirm', 'confirando': 'confirm', 'co
 nfirmasse': 'confirm', 'confirmassei': 'confirm', 'preventiva': 'preventiv', 'preventivo': 'preventiv', 'programado': 'programad',
 'programada': 'programad', 'resultado': 'result', 'resultando': 'result', 'importado': 'importa', 'importando': 'importa', 'reajust
 ado': 'reajust', 'reajustando': 'reajust', 'reajustada': 'reajust', 'danificado': 'danific', 'danificando': 'danific', 'danificada': 'danifi
 c', 'planeja': 'planej', 'planejar': 'planej', 'otica': 'optica', 'digital': 'digita', 'digita': 'digita', 'digital': 'digital', 'digitai': 'digital', 'falhad
 a': 'falha', 'palhada': 'falha', 'talhada': 'falha', 'catorze': 'quatorze'}

APÊNDICE D – TUTORIAL PARA USO DO SISTEMA PARA CLASSIFICAÇÃO DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO SETOR ELÉTRICO



SISTEMA PARA CLASSIFICAÇÃO DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO SETOR ELÉTRICO



SISTEMA DESENVOLVIDO COM PYTHON DESTINADO A CLASSIFICAÇÃO DA OPERAÇÃO DO SETOR ELÉTRICO COM MÚLTIPLOS MODELOS DE APRENDIZADO DE MÁQUINA CAPAZES DE RESOLVER PROBLEMAS DE CLASSIFICAÇÃO MULTICLASSE. A ENTRADA DE DADOS PODE SER NO FORMATO DE TEXTO OU DE ÁUDIO (TRANSCRIÇÃO DE FALA OU DE UM ARQUIVO).



CLASSIFICAÇÃO POR ÁUDIO

Informe o texto da comunicação verbal para a realização da classificação. é possível identificar 12 classes: CARGA, COMPROVAÇÃO DE DISPONIBILIDADE, CONTROLE DE GERAÇÃO, CONTROLE DE TENSÃO, CONTROLE DE TRANSMISSÃO, CONVERSORA, FALHA DE SUPERVISÃO, HIDROLOGIA, HORÁRIO, INTERVENÇÃO, TESTE DE COMUNICAÇÃO e SEM INFORMAÇÃO.

Campo para o upload de um arquivo de áudio no formato .WAV ou .MP3, de duração máxima 3 minutos e mono

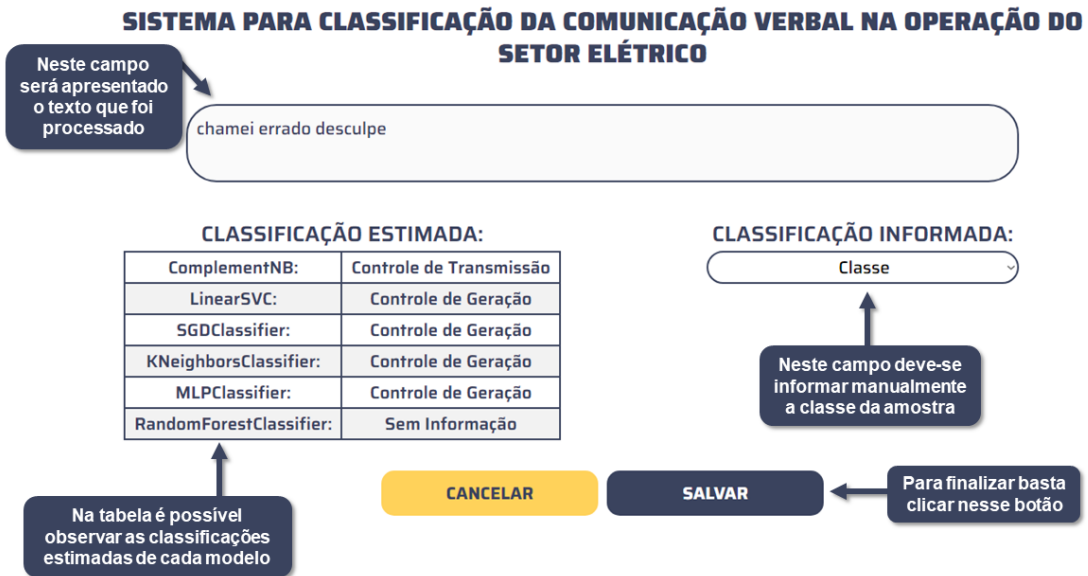
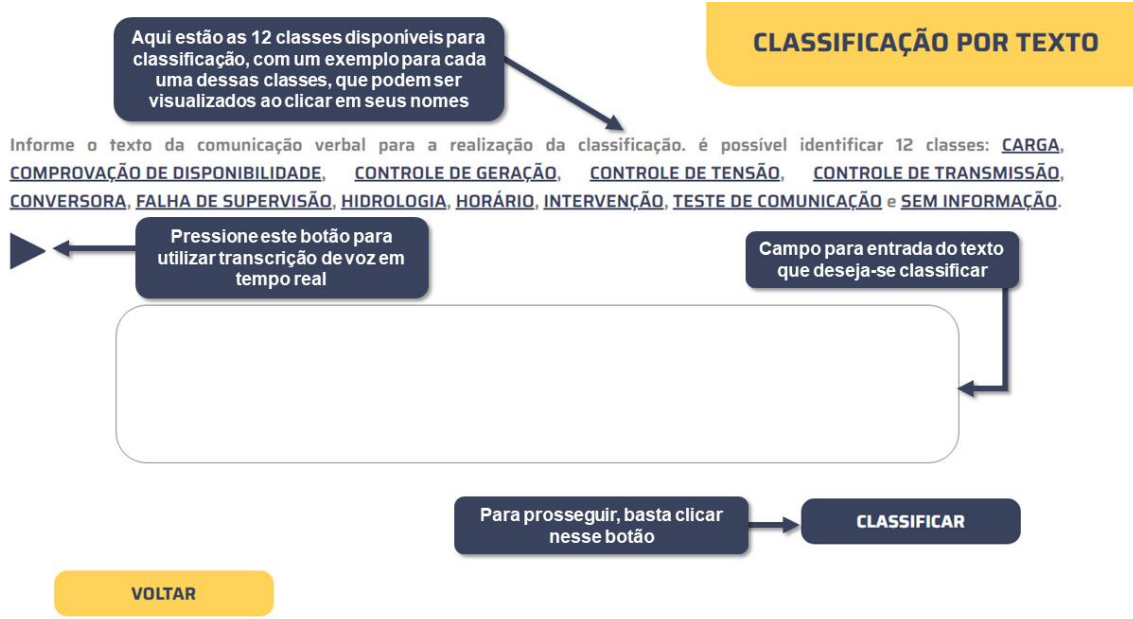
UPLOAD

Nenhum arquivo selecionado

Para prosseguir basta clicar nesse botão

TRANSCREVER

VOLTAR



SISTEMA PARA CLASSIFICAÇÃO DA COMUNICAÇÃO VERBAL NA OPERAÇÃO DO SETOR ELÉTRICO

Todos os testes realizados podem ser observados por essa tabela

Data e Hora	Amostra Original	Amostra Tratada	Categoria Informada	Categoria Complement NB	Categoria LinearSVC	Categoria SGDClassifier	Categoria KNeighborsClassifier	Categoria MLPClassifier	Categoria RandomForestClassifier
30/06/2021	Tinha sido pa	sido passado	Carga	Carga	Carga	Carga	Carga	Carga	Carga
30/06/2021	Na usina, a n	usina maquir	Comprovaçã	Comprovaçã	Comprovaçã	Comprovaçã	Comprovaçã	Comprovaçã	Comprovaçã
30/06/2021	Eleva trezent	elev trezent	Controle de C	Controle de C	Controle de C	Controle de C	Controle de C	Controle de C	Controle de C
30/06/2021	Dados de chu	dado chuva ç	Hidrologia	Hidrologia	Hidrologia	Hidrologia	Hidrologia	Hidrologia	Hidrologia
01/07/2021	Dados de chu	dado chuva ç	Hidrologia	Hidrologia	Hidrologia	Hidrologia	Hidrologia	Hidrologia	Hidrologia
01/07/2021	Só confirmar	confirm hora	Carga	Horário	Horário	Horário	Controle de C	Horário	Horário
01/07/2021	Chamei errac	chamei errad	Carga	Controle de l	Controle de C	Controle de C	Controle de C	Controle de C	Sem Informa
07/07/2021	a gente tem l	sony guaira t	Controle de l	Controle de l	Controle de l	Controle de l	Controle de l	Controle de l	Controle de l
08/07/2021	sudoeste de	nova porto p	Controle de l	Controle de l	Controle de l	Controle de l	Controle de l	Controle de l	Controle de l
12/08/2021	Tô fazendo h	fazendo test	Teste de Corr	Teste de Corr	Teste de Corr	Teste de Corr	Teste de Corr	Teste de Corr	Teste de Corr

Botão para voltar para o menu principal

VOLTAR

DOWNLOAD

Para baixar a tabela no formato .XLSX basta clicar nesse botão