

**FIAP – FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA**

BRUNO BIANCCHI – RM 84351

LUIS HENRIQUE CALDAS ALTERO – RM 88670

PEDRO GUILHERME POLLONI BARRETO - RM 88964

VITOR LAMPRECHT – RM 86691

**Estudo e Aplicação de Algoritmos de Machine Learning**

**São Paulo**

**2021**

Quando falamos criamos ondas sonoras que são captadas por meio de ACD (Conversor Analógico Digital) e são transformadas em dados que o computador possa processar. Para realizar tal conversão, o ADC digitaliza o som fazendo medições precisas nas vibrações com intervalos de frequência. O algoritmo usado filtra o som eliminando ruídos indesejados, faz o ajuste da velocidade de cada fala, adequa o volume e caso necessite separa o som em diferentes faixas de frequência.

Após essa etapa de ajustes, o sinal é fragmentado em segmentos curtos de centésimos de segundos ou milésimos de segundo para o algoritmo combinar esses segmentos em fonemas do idioma escolhido. Sendo que o Fonema é o menor elemento de uma linguagem, uma representação dos sons que produzimos para formar palavras. Agora na última etapa da conversão, o algoritmo compara os fonemas capturados com bibliotecas de palavras e frases conhecidas, determinando o que o usuário estava dizendo e assim realiza a transmissão em texto. Para esse estudo, fizemos a aquisição das seguintes bibliotecas, *PySpeech* e *Speech Recognition* no Python, no caso do Challenge usaremos o *Speech Recognition*.

Para realizar a análise de sentimentos é necessário usar a linguagem natural computadorizada (NPL) para extrair, identificar e analisar a polaridade dos dados. Existem dois tipos de abordagem, o lexical que é por voz e precisa de bibliotecas que analisam o tom de voz, e por Machine Learning. A abordagem baseada em ML precisa de uma coleção de documentos com sentimentos marcados; esta é uma coleção na qual cada documento foi avaliado manualmente e rotulado em termos de sentimento. Após o pré-processamento, o algoritmo supervisionado por ML é treinado para reconhecer o sentimento em cada texto.

Na primeira etapa, é necessário importar todas as bibliotecas que irá usar, como *Scikit Learn* que possui algoritmos de ML, *NumPy* para manipular os dados em arrays, *SciPy* para trabalhar com os arrays, *Natural Language Toolkit (NLTK)* contendo pacotes que permite que o computador entenda a linguagem humana e por final o *Pandas* para auxiliar na parte gráfica. Segunda etapa é trazer os dados que serão analisados, podendo estar em uma lista dentro do *Python* ou em algum arquivo “.txt”. Caso o arquivo esteja em “.txt” será necessário realizar a importação e com o `read()` e colocá-los dentro de uma lista.

Feito isso, o próximo passo é o tratamento desses dados, necessário retirar as vírgulas, os caracteres indesejados e as palavras que são definidas como “Stopwords” (as, os, um, uma, com, da, de, para etc.). Próxima etapa é fazer o “Stemming”, procedimento que remove os sufixos e prefixos de uma palavra, por exemplo (fazer, fazendo e fazia), se tornariam apenas “faz” ou no caso do inglês “Reading” seria necessário tirar o “ing” e deixar apenas “read”. Para mostrar as palavras mais faladas, usariamos a função “most\_common” do *NLTK*. Após todos esses tratamentos, seriam aplicados os algoritmos de probabilidades de *Bayes*, as tabelas seriam montadas e por fim a análise do modelo.

## REFERÊNCIAS

<https://www.lambda3.com.br/2019/04/inteligencia-artificial-como-fazer-analise-de-sentimentos-com-python-parte-1/>

<https://blog.novatics.com.br/como-construir-um-identificador-de-sentimentos-em-python-e2afbaade023>

<https://minerandodados.com.br/analise-de-sentimentos-utilizando-dados-do-twitter/>

<https://paulovasconcellos.com.br/aprenda-a-fazer-um-analisador-de-sentimentos-do-twitter-em-python-3979454f2d0d>

<https://letscode.com.br/blog/speech-recognition-com-python>