Relatório do Projeto — Mineração de Dados

Lucas Monteiro de Oliveira, João Victor Fontinelle Consonni

08/12/2019

1 Introdução

Ao longo de toda a história da humanidade, o uso eficaz da linguagem teve um papel fundamental na construção da sociedade como a conhecemos, sendo fundamental em nossos relacionamentos interpessoais. Dominar uma linguagem significa conhecer um sistema formal de gestos, sinais, sons e símbolos usados para compor palavras que caracterizam um meio de comunicar o pensamento. As palavras consistem em sons (orais) e formas (escritas) que têm significados acordados com base em conceitos, ideias e memórias [TEL 2018]. Tão importante quanto a capacidade de exprimir estes conceitos é a habilidade de interpretá-los corretamente.

O reconhecimento da fala humana é um subcampo interdisciplinar da linguística computacional, que incorpora conhecimentos provenientes de pesquisas nas áreas de linguística, ciência da computação e engenharia elétrica, com o intuito de utilizar computadores para reconhecer e traduzir a linguagem falada em texto. Há um interesse comercial considerável neste campo de pesquisa devido a sua ampla gama de aplicações como em raciocínio automatizado, tradução automática, resposta a perguntas, coleta de notícias, categorização de texto, ativação por voz, arquivamento e análise de conteúdo em linguagem natural em larga escala [Juang and Rabiner 2005].

Para que o computador possa interpretar a linguagem falada, ele precisa ser ensinado a fazê-lo. Devido a complexidade envolvida em desenvolver um programa para identificar individualmente cada possibilidade fonética de uma dada linguagem, métodos que utilizam aprendizado de máquina tornam-se bastante atrativos, uma vez que automatizam a criação de modelos analíticos baseando-se na ideia de que os sistemas podem aprender com os dados, identificar padrões e tomar decisões com o mínimo de intervenção humana [SAS 2019].

A construção de modelos de aprendizado de máquina para a classificação de áudio geralmente envolve tarefas de modelagem em que os dados de entrada são amostras de áudio, comumente representadas como séries temporais ao longo do eixo horizontal, com a amplitude da forma de onda descrita no eixo vertical. A amplitude é geralmente medida em função da mudança de pressão do ar ao redor do microfone ou dispositivo receptor que captou o áudio originalmente. A menos que haja metadados associados às amostras de áudio, esses sinais de

séries temporais constituem os únicos dados de entrada para ajustar um modelo [Mendels 2018].

Neste projeto, buscando explorar conceitos relacionados à mineração de dados, ao aprendizado e máquina e ao processamento de linguagem natural, foi desenvolvido um classificador que recebe de entrada um arquivo de áudio no formato *.wav* contendo uma sequência de 4 caracteres arbitrários, dentre os caracteres *a, b, c, d, h, m, n, x, 6* e *7*, pronunciados separadamente, simulando um CAPTCHA de áudio. O classificador então particiona este arquivo de áudio em 4 segmentos, um para cada caractere, e faz a identificação de cada um deles individualmente. Por fim, o modelo retorna a sequencia de caracteres por ele identificados, referente ao arquivo de entrada.

Para a construção do modelo aqui descrito foi utilizada uma base de dados gerada pelos próprios alunos. Para facilitar e padronizar tal geração foi disponibilizado pelo professor um arquivo *.html* com alguns arquivos *JavaScript* da biblioteca *P5JS*. O *script* pode ser executado em qualquer navegador moderno e escolhe aleatoriamente os caracteres que devem ser gravados com o intuito de evitar a repetição exagerada de algum deles, o que poderia reduzir a variação nos dados. Cada gravação resultou em um arquivo de áudio nomeado com o caractere correspondente e tem duração fixa de 2 segundos. Cada aluno gravou cerca de 10 vezes cada caractere utilizando o próprio microfone do computador.

Os áudios com os caracteres foram então enviados para o professor, que os utilizou para gerar uma série de arquivos de áudio no formato de entrada esperado pelo modelo, conforme descrito acima. Cada arquivo foi rotulado com a própria sequência de caracteres. Finalmente, o conjunto de CAPTCHAS de áudio foi separado em bases de treino, validação e teste, sendo que somente as bases de treino e validação foram disponibilizadas para que os alunos desenvolvessem o classificador.

Durante a implementação foi utilizada a linguagem de programação *Python* com as bibliotecas *LibROSA* para interpretar e realizar transformações nos áudios, *Pandas* para manipular os dados já interpretados e *Scikit-learn* para implementar os algoritmos de aprendizado de máquina.

2 Análise Exploratória

Antes mesmo de se iniciar a análise dos dados fornecidos para treinamento e validação do modelo, já era possível prever alguns possíveis problemas da base de dados. Primeiramente, tendo em vista que o desempenho de classificadores baseados em métodos de aprendizado de máquina é diretamente proporcional ao volume de objetos na base de treino, o volume da base de treinamento utilizada é relativamente pequeno, composto por 350 arquivos de áudio únicos, contendo 10 caracteres possíveis, o que implica em 10 classes possíveis para o classificador. É impossível prever qual seria um volume adequado para treinamento de uma aplicação como essa [Brownlee 2017], mas para efeito de comparação, a base de dados Urbansound8K contém 8732 trechos sonoros rotulados (¡= 4s) de sons urbanos de 10 classes [Salamon et al. 2014].

Outro ponto que deve ser levado em consideração é a heterogeneidade dos meios de captação dos sinais de áudio, dado que cada aluno gravou com um microfone diferente, com diferentes especificações quanto ao padrão polar, a faixa dinâmica e a resposta de frequência [Aldredge 2018], resultando na qualidade variável das amostras de aluno para aluno.

O padrão polar de um microfone é a sensibilidade ao som em relação à direção ou ângulo a partir do qual o som chega [Shure 2017]. Isso tem impacto direto na captação de áudio, pois, dependendo do microfone, os ruídos do ambiente, que são indesejados para este projeto, podem ser misturados à pronuncia do caractere de forma mais expressiva.

Já a faixa dinâmica tem relação com a largura do intervalo de amplitude que o sistema é capaz de captar [Rouse 2005]. A resposta de frequência é um conceito similar, porém diz respeito ao intervalo de frequências que o sistema consegue captar [FADGI 2017]. Geralmente, quanto melhor o microfone, maior é a cobertura desses dois indicadores. Quanto maior a largura de captura, tanto de amplitude quanto de frequência, menos informação é perdida [Aldredge 2018].

Logo, a heterogeneidade dos modelos de microfone faz com que algumas amostras possam apresentar níveis de ruído maiores do que outras, assim como outras amostras possam ter mais informação do caractere do que outras. Por conta disso, estabelecer uma regra para eliminação de ruído baseada em amplitude ou frequência pode ser problemático, uma vez que a remoção de ruído de um grupo de amostras pode resultar na eliminação de informação relevante em um outro grupo de amostras.

Outro fator que pode ser interpretado como um problema na base da dados são as diferenças consideráveis na pronúncia de um mesmo caractere entre amostras de alunos diferentes, dado que cada pessoa possui um tom de voz e até mesmo um sotaque diferente devido a eventuais regionalismos. Isso pode fazer com que um mesmo caractere apresente assinaturas consideravelmente distintas, causando eventuais *outliers* ou, eventualmente, barreiras pouco definidas entre objetos de grupos diferentes.

Dentre outros possíveis problemas estão a certeza de interferência no espectro de frequências causado por ruído estático e eventuais interferências causadas por sons ambientes indesejados, como possíveis cliques no notebook, carros passando na rua, pessoas falando ao fundo, etc.

Para treinar o modelo foi necessário segmentar cada amostra de áudio de 4 caracteres em 4 amostras de um caractere cada. Essa tarefa foi relativamente simples, uma vez que tínhamos a informação de que cada caractere ocupava um espaço de tempo constante de 2 segundos do áudio completo. Assim, a segmentação não exigiu um mecanismo muito complexo para perceber a separação entre caracteres.

O próximo passo foi extrair os atributos que precisaríamos para treinar o nosso modelo. Para fazer isso, poderíamos ter feito uso de espectrogramas, devido a sua utilidade em visualizar o espectro de frequências de um som e como elas variam durante um período muito curto de tempo. Contudo, acabamos por utilizar uma técnica semelhante, conhecida como Coeficientes Cepstrais de Frequência Mel (MFCC). A principal diferença entre essas duas estratégias é que

um espectrograma usa uma escala de frequência linear espaçada (para que cada compartimento de frequência seja espaçado com um número igual de Hertz), enquanto um MFCC usa uma escala de frequência espaçada quase-logarítmica, que é mais semelhante à maneira como o sistema auditivo humano processa sons. Pode-se observar através do espectrograma de Mel que caracteres semelhantes possuem espectros de frequência semelhantes [Smales 2019].

A utilização do MFCC resulta em 39 coeficientes para cada uma das diferentes bandas do espectro. Embora os coeficientes de ordem superior representem níveis crescentes de detalhes espectrais, dependendo da taxa de amostragem e do método de estimativa, 12 a 20 coeficientes cepstrais são tipicamente ideais para a análise da fala. A seleção de um grande número de coeficientes cepstrais resulta em uma maior complexidade nos modelos. Assim, foram selecionados 15 coeficientes relacionados à amplitude de frequências, o que nos fornece canais de frequência suficientes para analisar o áudio [Hui 2019].

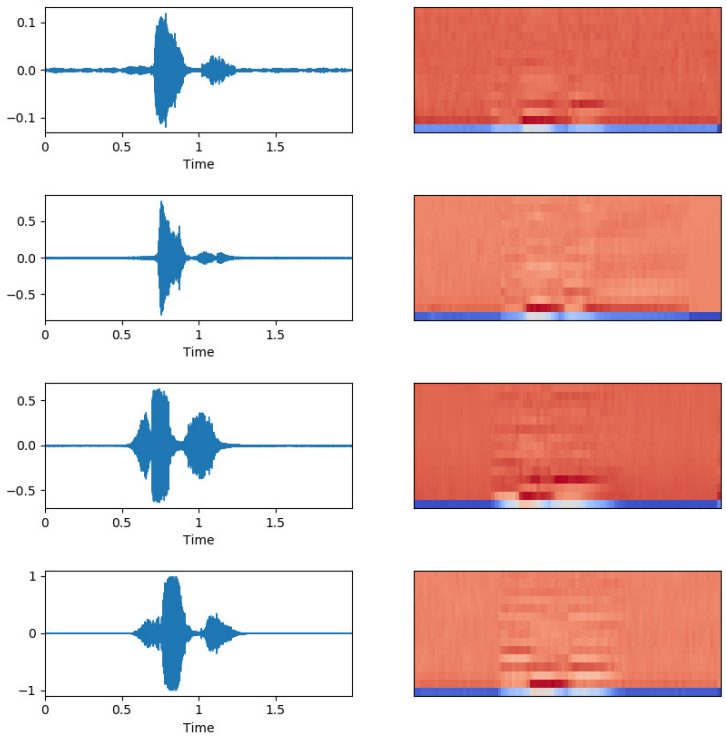

Figura 1: Amostras do caractere *7*

Na figura 1 são apresentadas 4 amostras do caractere 7. Na coluna da esquerda está a representação do sinal de áudio ao longo do tempo, enquanto na da direita está o espectrograma de Mel correspondente, sendo que cada linha do espectrograma é um dos 15 canais. É possível verificar que o MFCC ajuda a evidenciar as semelhanças entre caracteres similares, porém ainda existe a dependência quanto ao tempo. Uma análise do caractere m apresentada na figura 2 revela a presença de ruído na terceira amostra, o que acabou resultando em uma deformação do espectro quanto ao tempo, contudo, é possível verificar que a potencia em cada um dos canais ainda é semelhante. Esse tipo de informação nos fornece uma dica de que métricas independentes do tempo podem apresentar um bom resultado.

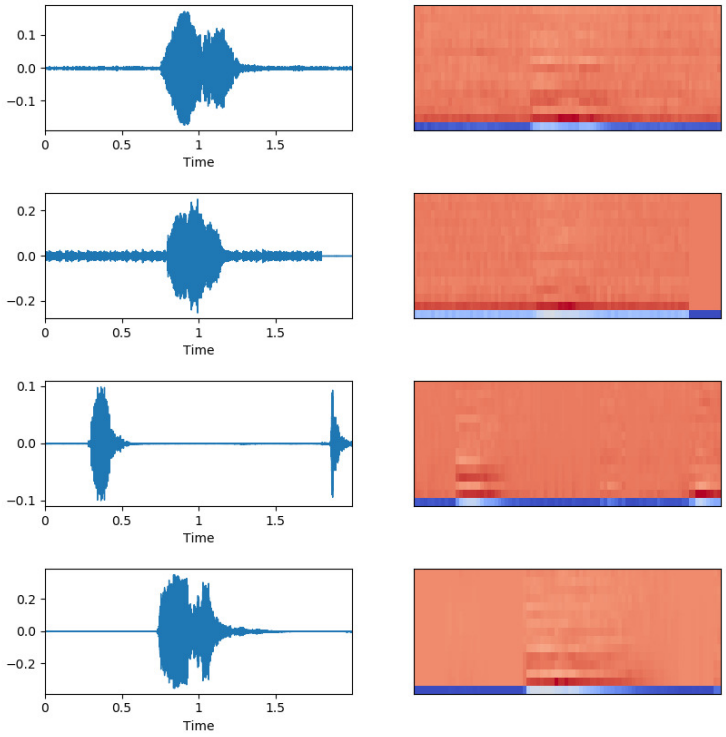

Figura 2: Amostras do caractere m

Outros atributos que foram considerados foram os coeficientes diferencial (primeira derivada do MFCC) e de aceleração (segunda derivada do MFCC). Os atributos do MFCC descrevem apenas o envelope espectral de potência de

um único quadro, porém a fala também possui informações na dinâmica, ou seja, nas trajetórias dos coeficientes MFCC ao longo do tempo. Calcular as trajetórias e aceleração do MFCC e anexá-las ao conjunto original de atributos aumenta bastante o desempenho do modelo de reconhecimento de áudio, porém também triplica o número de coeficientes, sendo 15 coeficientes MFCC, 15 coeficientes da primeira derivada, e 15 coeficientes da segunda derivada, totalizando 45 coeficientes [Lyons 2013].

Assim, sobre a base de dados de áudios já segmentados por caractere foram calculados os 15 coeficientes de Mel e suas primeira e segunda derivadas, que resultam em uma quantidade de atributos consideravelmente baixa para descrever uma série temporal complexa, porém distribuídos ao longo de um número muito alto de intervalos. Com o intuito de reduzir a dimensão da matriz de atributos e ainda tentar atenuar a presença de ruídos, como visto na figura 2, foram calculadas a média, a mediana, o desvio padrão, a variância, o valor máximo e o valor mínimo de cada um dos 45 coeficientes obtidos pela aplicação do MFCC, implicando numa matriz 45x6.

Além desses, foi considerado mais um atributo, conhecido como a taxa de *zero-crossing*, que representa a contagem de vezes que um sinal cruza o eixo cuja amplitude é zero em um determinado intervalo de tempo. Uma alta taxa de *zero-crossing* implica que não há oscilação dominante de baixa frequência, permitindo a distinção entre áudio com voz e sem voz [Jogy 2019].

Por fim, a base de treinamento para o modelo de classificação consiste em um conjunto de caracteres individuais rotulados e com os atributos acima descritos organizados em um vetor de uma linha e 271 colunas. As mesmas etapas de pré-processamento são conduzidas sobre os arquivos de entrada do modelo.

3 Metodologia

O algoritmo de classificação utilizado foi o *Random Forest*. Como o próprio nome sugere, esse algoritmo consiste em um grande número de árvores de decisão individuais que funcionam como um conjunto. Cada árvore individual na floresta aleatória calcula uma previsão de classe e a classe com mais votos se torna a previsão do modelo. O conceito fundamental por trás da *Random Forest* é a de que um grande número de modelos (árvores) relativamente não correlacionados que operam como um comitê superará qualquer um dos modelos constituintes individuais. Esse efeito resulta do fato de que as árvores se protegem de seus erros individuais. Enquanto algumas árvores podem estar erradas, muitas outras estão certas, então, como um grupo, as árvores podem se mover na direção correta [Yiu 2019].

As árvores de decisão individuais, por sua vez, podem ser entendidas como modelos que, durante a sua construção, escolhem um atributo que permitirá dividir os objetos em grupos, de tal forma que os grupos resultantes sejam os mais diferentes possíveis e, concomitantemente, os membros de um mesmo subgrupo resultante sejam os mais parecidos possíveis. Uma árvore repete esse processo múltiplas vezes para cada subgrupo produzido até que os nós folha sejam tão

homogêneos ou pequenos quanto desejado [Yiu 2019]. Vale observar que uma árvore de decisão pode utilizar apenas uma fração dos atributos disponíveis para fazer a segregação dos objetos em grupos, pois alguns atributos são mais relevantes que outros do ponto de vista da diferenciação.

Como mencionado na seção anterior, após o pré-processamento da base de dados original, os objetos da base de dados de treinamento são descritos por 271 atributos, o que representa um número expressivamente alto de atributos. Isso não é desejado, pois atributos desnecessários, irrelevantes ou redundantes não contribuem para a precisão de um modelo preditivo e podem de fato diminuir a sua precisão. Portanto, selecionar quais atributos utilizar é uma tarefa que melhora o desempenho do preditor e fornece preditores mais rápidos e com melhor relação custo-benefício, além de proporcionar uma melhor compreensão do processo de classificação [Brownlee 2014].

A estratégia adotada neste projeto para a escolha dos melhores atributos utilizou o conceito de importância. A importância representa a pontuação de um dado atributo, de forma que quanto maior for essa pontuação, mais importante ou relevante é atributo. Assim, aqueles que tiverem uma importância superior a um dado limiar são mantidos, enquanto os demais são descartados [Shaikh 2018]. O limiar adotado neste projeto foi de 125% do valor da média para um dado atributo, conforme sugerido na documentação da função *SelectFromModel* da biblioteca do *Scikit-learn*.

O modelo considerou um floresta com 10,000 árvores de decisão independentes e foi treinado uma primeira vez com todos os atributos. Em seguida, a seleção de atributos por importância escolheu os atributos mais relevantes para o modelo. Com essa informação, o primeiro modelo foi descartado e um novo modelo foi treinado, porém considerando apenas os atributos calculados como relevantes. O desempenho deste modelo foi avaliado por meio de uma metodologia de avaliação que consistiu em aplicar o classificador resultante sobre a base de dados de validação e então comparar os resultados fornecidos pelo modelo com o valor esperado para cada amostra da base. Os resultados desta etapa são apresentados na seção seguinte.

Verificada a acurácia do modelo proposto foi criado um novo modelo a partir das bases de treinamento e validação combinadas, seguindo o mesmo procedimento para a criação do modelo anterior. O objetivo desta etapa é garantir que o modelo final tenha o maior volume de informação disponível na sua construção, o que tende a melhorar o desempenho do classificador [Brownlee 2017].

4 Resultados

O modelo obtido a partir da base de dados de treinamento obteve resultados bastante satisfatórios, atingindo 75% de acurácia na predição de caracteres e, consequentemente, 31% na predição da sequência de 4 caracteres. Considerando cada caractere o modelo produziu a matriz de confusão apresentada em 3.

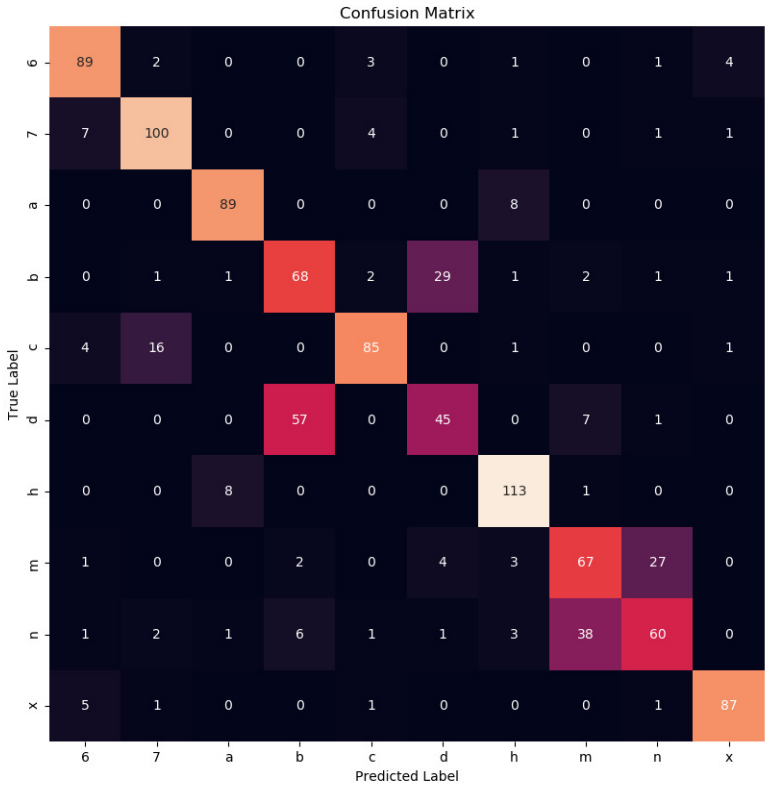

Figura 3: Matriz de Confusão

5 Comentários Finais

Apesar de satisfatório, o resultado obtido está muito aquém das ferramentas mais avançadas de reconhecimento de voz e áudio. Para alcançar o estado da arte em aplicações deste tipo é necessária uma vasta base de dados, assim como o esforço de tornar essa base livre de ruídos e o esforço de identificar os atributos que resultem na maior coesão entre objetos de uma mesma classe e maior heterogeneidade entre objetos de classes distintas.

Conforme evidenciado na matriz de confusão apresentada na figura 3, o modelo resultante tem bastante dificuldade em discernir entre as letras b e d, assim como entre as letras m e n, o que não é surpresa, dado que essas letras possuem uma pronúncia similar entre si e são frequentemente confundidas até

por humanos. Em menor escala, a letra *c* é confundida pelo número 7, enquanto a recíproca não é verdadeira.

Durante o desenvolvimento do projeto ficou evidente a importância da etapa de pré-processamento, principalmente levando em conta que o próprio processo de captação dos sinais resulta em perda de parte da informação, assim como na introdução de ruídos. O fato de cada objeto ser caracterizado por uma série temporal resultante da discretização de um sinal analógico tornou a análise ainda mais complexa, pois requereu a geração de atributos que o representassem de forma resumida e mais fácil de ser processada. Essa compressão conduz a uma perda potencialmente significativa de informação, mas ao mesmo tempo viabiliza a construção do modelo em tempo hábil.

Algumas ideias para melhorar a qualidade dos dados foram consideradas, mas não obtiveram o resultado esperado, efetivamente reduzindo a acurácia do modelo. Entre elas, estão a utilização de um filtro passa-baixa para tentar eliminar ruídos de fundo, e a aplicação de um filtro que considerasse apenas frequências dentro do espectro da voz humana. Aparentemente, a implementação de tais filtros resultou na perda de informação relevante para o classificador.

Considerou-se também a aplicação de uma função de *trim* para podar as partes inicial e final do arquivo de áudio, deixando a série temporal de cada caractere limitada ao intervalo no qual ocorreu a pronúncia do mesmo. No quesito de escolha de atributos, ponderou-se a utilização de um critério similar ao *zero-crossing*, porém para diferentes limiares, como 25%, 50% e 75% da amplitude do sinal. Contudo, nenhuma dessas abordagens não chegou a ser testada por falta de tempo para a implementação.

Referências

[Aldredge 2018] Aldredge, J. (2018). **What's the Difference Between a Cheap Microphone and an Expensive One?** Disponível em https://www.premiumbeat.com/blog/cheap-vs-expensive-microphone/. Acessado em 07/12/2019.

[Brownlee 2014] Brownlee, J. (2014). **An Introduction to Feature Selection**. Disponível em https://machinelearningmastery.com/an-introduction-to-feature-selection/. Acessado em 08/12/2019.

[Brownlee 2017] Brownlee, J. (2017). **How Much Training Data is Required for Machine Learning?** Disponível em https://machinelearningmastery.com/much-training-data-required-machine-learning/. Acessado em 07/12/2019.

[FADGI 2017] FADGI (2017). **Frequency Response (audio)**. Disponível em http://www.digitizationguidelines.gov/term.php?term=frequencyresponseaudio. Acessado em 07/12/2019.

[Hui 2019] Hui, J. (2019). **Speech Recognition — Feature Extraction MFCC PLP**. Disponível em https://medium.com/@jonathan$_h$*ui*/*speech* − *recognition* − *feature* − *extraction* − *mfcc* − *plp* − 5455*f*5*a*69*dd*9. *Acessadoem*08/12/2019.

[Jogy 2019] Jogy, J. (2019). **What features to consider while training audio files?** Disponível em https://towardsdatascience.com/how-i-understood-what-features-to-consider-while-training-audio-files-eedfb6e9002b. Acessado em 08/12/2019.

[Juang and Rabiner 2005] Juang, B. and Rabiner, L. (2005). **Automatic Speech Recognition - A Brief History of the Technology Development**. Disponível em https://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354$_L$*ALI* − *ASRHistory* − *final* − 10 − 8.*pdf*. *Acessadoem*07/12/2019.

[Lyons 2013] Lyons, J. (2013). **Mel Frequency Cepstral Coefficient (MFCC) Tutorial**. Disponível em http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/. Acessado em 08/12/2019.

[Mendels 2018] Mendels, G. (2018). **How to apply machine learning and deep learning methods to audio analysis**. Disponível em https://towardsdatascience.com/how-to-apply-machine-learning-and-deep-learning-methods-to-audio-analysis-615e286fcbbc. Acessado em 07/12/2019.

[Rouse 2005] Rouse, M. (2005). **Dynamic Range**. Disponível em https://whatis.techtarget.com/definition/dynamic-range. Acessado em 07/12/2019.

[Salamon et al. 2014] Salamon, J., Jacoby, C., and Bello, J. P. (2014). **A Dataset and Taxonomy for Urban Sound Research**. *22nd ACM International Conference on Multimedia (ACM-MM'14)*, pages 1041–1044. Disponível em https://urbansounddataset.weebly.com/urbansound8k.html. Acessado em 07/12/2019.

[SAS 2019] SAS (2019). **Machine Learning**. Disponível em https://www.sas.com/en$_u$*s*/*insights*/*analytics*/*machine* − *learning.html*. *Acessadoem*07/12/2019.

[Shaikh 2018] Shaikh, R. (2018). **Feature Selection Techniques in Machine Learning with Python**. Disponível em https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e. Acessado em 08/12/2019.

[Shure 2017] Shure (2017). **Microphones: Polar Pattern/Directionality**. Disponível em https://www.shure.eu/musicians/discover/educational/polar-patterns. Acessado em 07/12/2019.

[Smales 2019] Smales, M. (2019). **Sound Classification Using Deep Learning**. Disponível em https://medium.com/@mikesmales/sound-classification-using-deep-learning-8bc2aa1990b7. Acessado em 07/12/2019.

[TEL 2018] TEL (2018). **History of Spoken Communication**. Disponível em https://tellibrary.org/lessons/history-of-spoken-communication/. Acessado em 07/12/2019.

[Yiu 2019] Yiu, T. (2019). **Understanding Random Forest**. Disponível em https://towardsdatascience.com/understanding-random-forest-58381e0602d2. Acessado em 08/12/2019.