UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Bernardo de Moraes Santana Júnior

Predição de popularidade de podcasts através de características textuais.

Recife

2019

Bernardo de Moraes Santana Júnior

Predição de popularidade de podcasts através de características textuais.

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Giordano Ribeiro Eulalio Cabral

Recife

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

J95p Júnior, Bernardo de Moraes Santana
Predição de popularidade de podcasts através de características textuais / Bernardo de Moraes Santana Júnior. - 2019.
42 f. : il.

Orientador: GIORDANO RIBEIRO EULALIO CABRAL.
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, , Recife, 2019.

1. Podcast. 2. Popularidade. 3. Processamento de linguagem natural. 4. Transcrição. 5. Chartable. I. CABRAL, GIORDANO RIBEIRO EULALIO, orient. II. Título

CDD

BERNARDO DE MORAES SANTANA JÚNIOR

Predição de popularidade de podcasts através de características textuais.

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 17 de Dezembro de 2019.

BANCA EXAMINADORA

Giordano Ribeiro Eulalio Cabral (Orientador)
Centro de Informática
Universidade Federal de Pernambuco

Gilberto Amado de Azevedo Cysneiros
Departamento de Estatísitca e Informática
Universidade Federal Rural de Pernambuco

À minha família que sempre meu deu suporte …

Agradecimentos

Agradeço a minha família por toda ajuda, mais específico minha avó Esmeralda por ter me dado a oportunidade de ter uma boa base de conhecimento acadêmico, base essa que me possibilitou de cursar este curso, agradeço novamente a ela por ter me ajudando tanto financeiramente quanto emocionalmente durante o percurso deste curso. Agradeço a minha mãe, pai e irmã pelo suporte e me aguentar durantes as épocas de entregas e provas, vocês são 10.

Agradeço aos colegas e amigos que fiz durantes os 5 anos que durou este curso por fazerem parte de um momento importante da minha vida. Em especial aqueles que estiveram mais próximos durante esse longo período, sei que posso contar com vocês e fiquem sempre a vontade par contar comigo. Um agradecimento especial para Victor Leuthier pelo companheirismo.

Agradeço aos mestres (professores e veteranos) que me guiaram durante esse processo de amadurecimento tanto pessoal quanto profissional, essa faze foi muito importante e sem seus conselhos e suporte seria extremamente difícil esse processo como um todo.

De coração... Muito obrigado.

“If you focus too far in front of you, you won’t see the shiny thing out the corner of your eye.”

(Tim Minchin)

Resumo

Com o tremendo crescimento dos *Podcasts* e profissionalização de seus criadores, ao ponto de redes de notícias chamarem esse momento como ”era de ouro”para os *Podcasts*, novas ferramentas surgiram para auxiliar esses produtores na construção e manutenção de seus canais. Nesse contexto encontrar características nos episódios produzidos que proporcionem um alcance maior ao público alvo é de grande valor tanto para os criadores quanto para os ouvintes, permitindo que canais permaneçam ativos por mais tempo e ofereçam uma melhor qualidade de conteúdo.

Assim, este trabalho propõe um estudo de análise de popularidade dos *Podcasts* nacionais, utilizando uma ferramenta de análise da audiência dos *Podcasts* em um dos agregadores de canais e episódios mais utilizados do mundo, o *iTunes*. Através de ferramentas de *Web Scraping* para a coleta das informações disponíveis e necessárias, de ferramentas para transcrições dos áudios dos episódios para a obtenção do que foi dito e o calculo de métricas para medir precisão do modelo gerado, assim fazer uma análise de quais informações são relevantes para a predição de popularidade de um canal.

Resultados obtidos foram favoráveis na correlação entre as categorias analisadas de forma individual e texto dos episódios nelas contidos, enquanto em uma análise em que categorias não são descriminadas há uma baixa relação entre texto e popularidade, demonstrando que a categoria de determinado canal tem um papel importante na análise de sua popularidade.

Palavras-chave: iTunes, Podcast, Popularidade, Processamento de linguagem natural, Correlações, Transcrição, Chartable.

Abstract

With the tremendous growth of Podcasts and the professionalization of its creators, to the point that news networks call this as Podcast's ”golden age”, new tools have emerged to assist its content producers in building and maintaining of their channels. In this context, finding features inside episodes that provide broader reach to the target audience is of great value to both creators and listeners, allowing channels to stay active longer and offer better content quality.

Thus, this paper proposes a study of popularity analysis of brazilian's podcasts using a podcast audience analysis tool in one of the most used channel and episode aggregators in the world, iTunes. By using Web Scraping tools to collect available and necessary information, also tools for transcriptions of the audios's episodes in order to obtain what has been said, and calculating metrics to measure the accuracy of the generated model, therefore making an analysis of which information is relevant or not to predicting a channel's popularity.

Results displayed were favorable in the correlation between the categories analyzed individually and the its text, whereas in an analysis in which categories are not discriminated there is a low relationship between text and popularity, demonstrating that the category of a given channel plays an important role in analyzing its popularity.

Keywords: iTunes, Podcast, Popularity, Natural Language Processing, Correlations, Transcription, Chartable.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

RSS	Really Simple Syndication
ALPR	Adversarial Learning-based Podcast Representation
API	Application Programming interface
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	Hypertext Markup Language
NLP	Natural Language Processing
STT	Speech to Text
VUI	Voice User Interface
LPC	Linear Predictive Coding
HHM	Hidden Markov Model
SVM	Support Vector machine
SaaS	Software as a Service
POS	Part-Of-Speech Tagging
NER	Named Entity Recognition
BoW	Bag-of-Words
Tf-Idf	Term frequency–Inverse document frequency
URL	Uniform Resource Locator

Sumário

1 Introdução

Podcast é um novo formato de mídia que vem chamando a atenção recentemente, caracterizado por canais de distribuição de arquivos de áudios, comumente no formato ”MP3”, e são disponibilizados em tempo real através de um *RSS Feed*, onde é disponibilizado todas as informações do canal e episódios. Esse formato permite a disponibilidade para qualquer usuário que acesse programas ou sites com o papel de agregadores, neles o usuário pode se inscrever nos canais que deseja para receber atualização de novos episódios.

Os holofotes voltaram para *Podcasts* tanto nacionalmente quanto internacionalmente depois que grandes *players* locais e internacionais iniciaram seu investimento nesse tipo de mídia. Com o mercado brasileiro sendo o segundo maior consumidor deste conteúdo no mundo[1], gigantes da comunicação como o Jornalismo Globo lançou em 2019 vários dos seus programas em formato de *Podcast*, identificando a tendência e investindo pesado nesse tipo de canal.[2] Além de eventos como o *Spotify For Podcasters Summit* que reuniu os principais canais para conversar sobre essa revolução[3]. Enquanto no maior mercado de *Podcast* do mundo (Estados Unidos) o número de pessoas que já ouviu a algum episódio chega a 51% da população.[4]

Mesmo com esse crescimento e o investimento de grandes agregadores como *Spotify* e *Deezer*, ainda é difícil ter acesso com precisão aos números relacionados como quantidade de usuários únicos que escutam determinado episódio ou quanto tempo de um episódio é em média escutado. Pesquisas e ferramentas de análise buscam encontrar valores aproximados desses e outros números para medir a popularidade dos canais entre os diversos agregadores disponíveis no mercado.

1.1 Motivação

O foco deste trabalho está na análise de como as características textuais podem auxiliar na predição de popularidade de *Podcasts*, que pode avançar na construção de um sistema de recomendação utilizando a popularidade como parâmetro para atacar o problema de *cold-start* em aplicações sem dados históricos do usuário, assim oferecer uma melhor experiência no primeiro uso. Essa recomendação pode utilizar os episódios para medir qual canal tem mais probabilidade de agradar um usuário novato da

1 https://powerpressPodcast.com/2019/02/01/Podcast-stats-soundbite-brazil-bloom/
2 https://globoplay.globo.com/v/7870430/
3 https://spotifyforPodcasterssummit.com.br/
4 https://www.edisonresearch.com/wp-content/uploads/2019/03/Infinite-Dial-2019-PDF-1.pdf

mídia ou dado a escolha de um canal oferecer o episódio com maior probabilidade de satisfazer o usuário. Uma outra usabilidade do resultado buscado neste trabalho está relacionada com a pesquisa de quais características textuais estão ligadas aos *Podcasts* mais populares, assim possibilitando a criação de um sistema capaz de auxiliar os produtores de conteúdo à entender quais características de seus episódios estão relacionadas aos *Podcasts* mais populares e quais não, permitindo que eles adaptem seus conteúdos.

1.2 Objetivo

Para isso este trabalho tem como meta, a resolução de duas questões norteadoras. Pode-se prever a popularidade ou não de um determinado canal baseado no que é dito em seus episódios? Quais características são importantes nesta predição?

No percurso de construção desse sistema esse trabalho objetiva:

1. Construção de base de canais segregados por popularidade.
2. Download e transcrição dos episódios dessa base.
3. Encontrar correlação entre o conteúdo textual dos episódios e a popularidade dos canais.

1.3 Estrutura do trabalho

O presente trabalho está organizado na seguinte estrutura: Na seção 2 é apresentado os trabalhos relacionados a este, com um resumo do que foi proposto em cada um dos trabalhos, mostrando como problemas parecidos foram atacados por outros autores. Na seção 3 é apresentado o referencial teórico com os fundamentos necessários para a construção deste trabalho. Já na seção 4, a Metodologia, é apresentada como a base da dados foi construída para a análise que é realizada na seção 5. Na seção 6 uma avaliação dos resultados é apresentada bem como uma breve análise do que pode ser feito em trabalhos futuros.

2 Trabalhos relacionados

Não há uma grande variedade de trabalhos que utilize *Podcasts* para construir sistemas como classificação ou recomendação, a pouca variedade de estudos neste meio tem correlação com o pouco tempo em que os *Podcasts* são considerados uma mídia com grande número de ouvintes, a baixa quantidade de informações analíticas das aplicações utilizadas para ouvi-los, e a grande variabilidades de formatos em que o conteúdo é apresentado. Nesta seção é apresentado trabalhos que propuseram sistemas para sanar algumas das dificuldades citadas e trabalhos que propuseram análise das informações disponíveis de forma aberta sobre esse tipo de mídia.

Os trabalhos de (OGATA; GOTO, 2009) e (GOTO; OGATA, 2011) apresentam uma ferramenta chamada *PodCastle*, essa ferramenta busca atacar a dificuldade de gerar uma boa transcrição dos *Podcasts*. Para a transcrição foi utilizado um modelo construído com uma base com dados de músicas e dicionários abertos no idioma japonês, para construir um sistema que apresenta ao usuário, no momento em que um episódio está sendo escutado, as melhores opções de transcrição do que está sendo dito, assim criando uma forma de correção de erros colaborativa em que cada usuário seleciona as palavras que foram ditas dentro das opções disponíveis. Apresenta também os resultados e ganhos dessa abordagem, estatística de uso da ferramenta com objetivo de validar o modelo de negócio da ferramenta e de qualidade das transcrições obtidas antes e depois da utilização dos usuários.

Enquanto o trabalho de (MIZUNO; OGATA; GOTO, 2008) utilizou a plataforma *PodCastle* para propor um novo modelo para encontrar palavras chaves e para construir um sistema para encontrar episódios similares. Para isso esse modelo não utilizou somente a transcrição com maior probabilidade, então propôs um novo modelo que utiliza as transcrições mais prováveis de serem verdadeiras em uma rede de confusão para reduzir a interferência de erros de transcrição no processo. Inicialmente este modelo é testado com a comparação com modelos considerados estado da arte para a extração de palavras chaves, que mostra um ganho em performance na obtenção dessas palavras considerando erro na fase de transcrição do texto, ou seja, as palavras encontradas pelo modelos propostos tinham menor probabilidade de serem erro de transcrição. Em um segundo momento propõe a utilização deste modelo para a extração das 100 palavras chaves do texto e utilização disso para calcular a categoria de cada episódio, o resultado obtido foi favorável em sua maioria, somente não obtendo melhoria no caso em que a transcrição tinha uma baixa qualidade.

Já o trabalho de (YANG et al., 2019) propõe a utilização do ranking de *Podcasts*

por temas disponibilizados no site da *Chartable* (ferramenta de *analytics* de *Podcasts*), para a construção de um sistema que é capaz de prever a popularidade a partir das características obtidas no último episódio do canal, para isso esse trabalho assume que canais com maior popularidade em um determinado período podem ser entendidos através de seus episódios lançados nesse mesmo período. Apresenta, então, o ALPR (*Adversarial Learning-based Podcast Representation*) para capturar características não textuais (comuns para o cenário de música) dos *Podcast* como “energia” e “seriedade” com o objetivo de não somente utilizar características textuais no sistema de predição de popularidade proposto, que apresenta resultados melhores do que os modelos considerados estado da arte. Posteriormente este trabalho combina características textuais e não textuais na predição de popularidade testando a eficácia do modelo variando a quantidade de tempo dos episódios utilizada entre 1 e 10 minutos, com resultados favoráveis ao modelo combinado em relação aos demais. Enquanto (YANG et al., 2018) propôs uma ferramenta para recomendação de *Podcasts* utilizando linguagem natural como interface.

3 Referencial Teórico

3.1 *Podcast*

Termo inicialmente cunhado em 2004 como sendo um acrônimo de *IPod* (dispositivo portátil para reprodução de músicas desenvolvido pela *APPLE*[1]) e o termo *broadcast*. *Podcast* é um episódio de uma série que é disponibilizada para download na internet por meio de uma distribuição automática de arquivos (de conversa, música, vídeo, entre outros) por meio de uma inscrição na internet.(PRESS, 2004) O principal formato para disponibilização dos *Podcast* é arquivos MP3 e o *feed RSS*[*Really Simple Syndication*] que inclui informações de *metadata* dos episódios e do canal como: Nome; Descrição; Duração; Data de *upload*; Categorias; Idioma entre outras.(OGATA; GOTO, 2009)

O termo vem se popularizando com a distribuição de áudios em um formato similar aos programas de rádio e a entrada dos grandes *players* de *streaming* no segmento como *Spotify*, *Deezer* e *Google Podcasts*. Além do grupo Globo de comunicação ter lançado diversos *Podcast* em agosto de 2019 o que mostra uma tendência na utilização desse meio de consumo. O principal formato dos episódios são de pessoas conversando a respeito dos mais variados temas como política, futebol, saúde, idiomas, entre muitos outros. A duração desses episódios pode variar demais, mas é comum durar por mais de uma hora, com uma curiosidade o recorde mundial é de 52 horas.(RECORDS, 2012)

Segundo a *Podpesquisa*[2] realizada com mais de 22 mil pessoas, entre elas ouvintes, produtores e não ouvintes de *Podcast*, a quantidade média de horas gastas num dia escutando *Podcast* é de 2h52min onde 91.7% dos ouvintes escutam o *Podcast* inteiro. Segundo os entrevistados, 94.2% consideram o conteúdo do *Podcast* um aspecto importante e 75.4% disseram que a qualidade do áudio é um outro ponto relevante. A *Podpesquisa* ajudou a entender o cenário de *Podcast* no Brasil e o perfil dos ouvintes que acabou dando direcionamento sobre quais *Podcast* e *features* poderiam ser relevantes para explorarmos durante a pesquisa.

3.2 *Web Scraping*

Web Scraping é a prática de coletar dados de servidores da internet por qualquer meio que não seja uma API (*Application programming interface*), comumente rea-

[1] https://www.apple.com/
[2] http://www.abpod.com.br/media/docs/PodPesquisa-2018.pdf

lizada através da criação de programas automatizados que, através de requisições, obtém os dados e os analisa para extrair as informações necessárias.(MITCHELL, 2015)

Pode ser definido, também, como o processo de extração e combinação de conteúdo de interesse da Web de maneira sistemática. Em tais um processo, um programa, também conhecido como robô Web, imita a interação de navegação entre os servidores Web e o humano de uma maneira convencional.(GLEZ-PEÑA et al., 2013) Que ao invés de realizar a busca manual dessas informações, a criação de um robô permite de maneira automática e velozmente a obtenção de grande quantidade de dados e, possivelmente, a criação de novos serviços.

Os passos para realização desse processo podem ser sumarizados como:

1. Acesso à uma página Web.

A comunicação é criada através de uma requisição com o protocolo HTTP (*Hypertext Transfer Protocol*)[3], um protocolo que não armazena estado e é a base da internet e aquele utilizados pelos navegadores (Chrome, Safari, Firefox entre muitos outros), e mais comumente utiliza os métodos GET ou POST. A informação de origem é enviada contendo o meio que a requisição foi enviada, podendo ser um robô ou um ser humano utilizando um navegador.

O servidor Web pode conter um arquivos chamado “robot.txt” com diretrizes de quais dados podem ou não serem acessados por robôs. É importante, também, ter cuidado com o acesso simultâneo de muitas requisições em uma curta faixa de tempo, pois pode sobrecarregar o servidor em questão.

2. Carregamento e extração de informações da página Web.

Uma vez a página carregada, a aplicação pode extrair as informações desejadas. Existem algumas maneiras mais conhecidas de executar essa tarefa como seletores de CSS (Cascading Style Sheets)[4], bibliotecas que carregam o código html e transformam em objeto DOM (Document Object Model)[5], seletores XPath[6] e expressões regulares.

Com a página carrega e o método de extração de dados escolhido, a tarefa de extração das informações pode ser executada. Uma dificuldade que pode aparecer nesta etapa é a mudança no HTML das páginas requisitadas.

[3] https://developer.mozilla.org/en-US/docs/Web/HTTP
[4] https://www.w3schools.com/cssref/css_selectors.asp
[5] https://www.w3schools.com/js/js_htmldom.asp
[6] https://www.w3schools.com/xml/xpath_intro.asp

3. Pós-processamento.

Nessa etapa os dados já foram coletadas e precisam ser armazenado em algum lugar, algumas bibliotecas/ferramentas disponibilizam soluções como a gravação dessas informações em arquivos.

3.3 Processamento de linguagem natural

Linguagem natural se refere à forma de comunicação construída pelos seres humanos para transmissão de informações, textos escritos ou linguagem falada em algum idioma ou dialeto que possuem características peculiares e de difícil estruturação para o mundo binário dos computadores.

O processamento desta linguagem, também chamado em inglês de *Natural Language Processing* (NLP), utiliza conhecimentos de áreas como linguística, ciência da computação, estatística e inteligência artificial para possibilitar ao computador a capacidade de leitura, deciframento, entendimento e extração de informações que façam sentido e tragam valor na construção de produtos e soluções. Exemplos de aplicações nessa área são: conversão de áudios de fala em texto, de texto para voz, processamento textual, extração de características como emoção, tradução.

3.3.1 Reconhecimento automático da Fala

Reconhecimento automático da fala ou mais conhecido no inglês como ASR é a área da computação responsável pelo estudo da transcrição da voz humana para o texto, também chamada de *Speech to Text*(STT). Existem muitas aplicações comerciais para o uso dessa tecnologia como as VUI(*Voice User Interface*) que é a comunicação via fala entre ser humano e robô, automação de serviços telefones, etc. Atualmente a aplicação mais usual são as assistentes pessoais que processam a fala dos usuários convertendo para texto para realizar as atividades pré-programadas.(RUDNICKY; HAUPTMANN; LEE, 1994)

3.3.1.1 Uma breve história até os anos 2000

Os primeiros estudos para criar um sistema de reconhecimento automático de fala fizeram uso, principalmente da teoria fonética acústica, que descreve os elementos fonéticos da fala ou seja os sons mais básicos da língua, tentando explicar como são realizados acusticamente. Esses modelos naturais de ressonância são chamados de “formants” ou “formant frequencies” e manifestam-se pela concentração de energia no espectro de potência da Fala. Em 1952 Davis et al (DAVIS; BIDDULPH; BALASHEK,

1952) construíram um sistema de conhecimento isolado de reconhecimento de dígitos por uma única pessoa. na mesma época uma pesquisa dos laboratórios de RCA construíram um sistema capaz de reconhecer 10 sílabas por um único pessoa.[10]

Nos anos de 1970 as ideias de reconhecimentos de padrões fundamental para reconhecimento de fala, com base nos métodos “Linear Predictive Coding (LPC)” propostos por Atal e Itakura. Também durante esse período foi fundada a primeira empresa comercial com esses fins chamada de Threshold Technology Inc. onde desenvolveram o primeiro produto real chamado “VIP-100 System”. Outros sistemas desenvolvidos nessa época pela DARPA’s foram CMU’s Hearsay(-II) e BBN’s HWIM.

Já em 1980 com os esforços da IBM liderado por Fred Jelinek, tinha por objetivo criar uma máquina de escrever ativa por voz, onde a principal função era converter uma sentença falada em uma sequência de letras e palavras que poderiam ser mostradas num display ou escritas no papel. O sistema de reconhecimento foi chamado de Tangora, que foi um sistema dependente do locutor, ou seja, o sistema precisa ser treinado por cada usuário. Onde o foco técnico estava no tamanho do vocabulário criado e na gramática representada por estatísticas sintéticas que descrevem a probabilidade de uma sequência de símbolos do idioma(por exemplo fonemas ou palavras). Ainda esse ano as abordagens iniciadas pela IBM e AT&T Bell Laboratories sobre reconhecimento da fala obtiveram uma influência na evolução da comunicação por fala entre humano e computador nas próximas décadas. um tema comum entre esses esforços apesar da diferença, era o formalismo matemático e o rigor que começaram a surgir como aspectos distintos e importantes da fala. O rápido desenvolvimento de métodos estatísticos nesses anos principalmente a cadeia de markov escondida conhecida como (HHM), causou um grau de convergência onde hoje a maioria dos sistemas práticos são baseados nestas técnicas.(JUANG; RABINER, 2005)

Em 1990 iniciou-se uma consolidação dos algoritmos para processamento e ferramentas para desenvolvimento, temos nesses anos o avanço com as técnicas usando cadeias de markov escondidas e iniciando o uso de máquinas de vetor de suporte ou no inglês SVM, e o sucesso desses métodos estatísticos reacendeu o interesse pelo DARPA, levando ao desenvolvimento de novos sistemas alguns utilizados até hoje como o Sphinx um sistema da CMU (LEE, 1988) onde obteve resultados notáveis para reconhecimento de fala de amplo vocabulário.

Na figura é possível ver um gráfico com a avaliação da performance de diversos vocabulários medidos formalmente usando o DARPA e NIST. O dataset WSJ referi a transcrição de um conjunto de leituras de parágrafos do Wall Street Journal com o vocabulário maior que 60 mil palavras. A conclusão geral que pode-se extrair desse gráfico é que o dataset de conversação que não segue as restrições linguísticas, é significativamente mais difícil de reconhecer do que o discurso orientado a tarefas.

No final dos anos 90 e nos anos 2000 como um todo alcançamos um nível na área que ficou cada vez mais acessível para desenvolvedores sozinhos conseguem usufruir das técnicas de ASR e cada vez mais disponíveis para vários idiomas. E recentemente temos serviços disponibilizados que qualquer desenvolvedor pode utilizar sem grande processamento da sua máquina como as APIS da IBM speech to Text, Amazon transcribe, speech-to-text do Sistemas cognitivos da Azure e o Google speech recognition e outros serviços menores.

3.3.1.2 *Google Cloud Speech-to-Text*

O *Speech-to-text* é a abordagem do Google para a transcrição de áudios em texto. Aplicação localizada na área de ferramentas de aprendizado de máquina do sistema em nuvem da empresa e comercializada no formato SaaS(*Software as a Service)* e faturamento no modo *"pay as you go"*, que permite uma maior flexibilização e baixo custo para testar sua utilização.

Capaz de fazer transcrições em 120 idiomas e variantes[7], é uma ferramenta bastante consolidada no mercado e conhecida pela qualidade do resultado obtido em sua utilização. Para a transcrição dos áudios, lave-se, em média, a metade do tempo de duração do arquivo para completar o processo.

3.3.1.2.1 Modelos

Para a sua utilização a ferramenta oferece 4 modelos pré-treinados para diferentes utilizações:

1. command_and_search

Modelo recomendado para a transcrição de curta duração, como comandos de voz à assistentes conversacionais.

2. phone_callx

Modelo melhor adaptador para a transcrição de chamadas telefônicas, pela característica da baixa taxa de amostragem desse tipo de mídia.

3. vídeo

Modelo para transcrição de áudios originados de vídeos e quando há mais de um locutor. É considerado um modelo *Premium* e tem custo mais elevado dos demais.

[7] https://cloud.google.com/speech-to-text/

4. padrão

Modelo considerado padrão por ser o responsável pela transcrição dos áudios que não se encaixam nas outras categorias, como áudios longo com boa taxa de qualidade.

3.3.1.2.2 Linguagens

Para a utilização deste serviço bibliotecas são disponibilizadas em várias linguagens de programação como: C#, Go, Java, Node.JS, PHP, Python e Ruby.[8] Neste trabalho a linguagem utilizada para consumir o serviço foi Python.

3.3.1.2.3 Codificação

Uma codificação de áudio refere-se à maneira como os dados de áudio são armazenados e transmitidos. As codificações aceitas pela API são: FLAC, LINEAR16, MULAW, AMR, AMR_WB, OGG_OPUS, SPEEX_WITH_HEADER_BYTE[9], mas é recomendado sempre que possível somente utilizar as codificações FLAC e LINEAR16.

Para a obtenção de uma boa resposta é indicado a conversão caso o arquivo não esteja na codificação com melhor performance, portanto arquivos de *Podcasts* que são documentos do tipo MP3 precisam serem convertidos para uma das codificações recomendadas para serem aceitos e obterem resultados melhores na API. Quanto a taxa de amostragem do áudio é recomendado que o arquivo tenho pelo menos 40.000 khz para um bom resultado.

3.3.1.2.4 Tipos de requisição

Existem 3 formas para requisitar a utilização do serviço, cada forma refere ao formato e duração em que o áudio vai ter, essas maneiras são:

1. Reconhecimento de fala síncrono

Também conhecido como reconhecimento de fala de áudios curtos (até 1 minuto de duração), esse método pode ser utilizado com arquivos armazenado localmente ou remotamente.[10]

2. Reconhecimento de fala assíncrono

[8] https://cloud.google.com/speech-to-text/docs/reference/libraries
[9] https://cloud.google.com/speech-to-text/docs/encoding
[10] https://cloud.google.com/speech-to-text/docs/sync-recognize

Esse método é recomendado para áudios longos (duração maior de 1 minuto). Para realizar a transcrição neste tipo de áudios é necessário que o arquivo esteja armazenado no *Google Cloud Storage*.[11]

3. Reconhecimento de fala em streaming

O reconhecimento de fala em streaming permite a transcrição de fala em tempo real conforme o áudio é processado, recomendados para casos em que se precisa da transcrição no momento em que o áudio é gerado. [12]

O *Google Speech-to-text* oferece uma plataforma bastante completa para transcrição de áudios para texto, a escolha desta em detrimento de concorrentes disponíveis no mercado foi pautada na familiaridade com a plataforma *Google Cloud* como um todo, a alta taxa de assertividade oferecida e a disponibilização de um total de 300 dólares em forma de crédito aos novos usuários, que possibilita a construção de produtos e, mais especificamente, uma parte deste trabalho.

Para a obtenção de uma boa transcrição, este trabalho utilizou o reconhecimento de fala assíncrono e modelo “padrão” por causa de características como grande duração média dos arquivos e alta taxa de amostragem, todos os arquivos com codificação LINEAR16 e formato WAV com somente um canal de áudio.

3.3.2 Técnicas de Processamento Textual

Processamento textual é uma área muito abrangente alguns exemplos de tarefas são apresentados neste texto, para realizar a análise do texto, é comum a transformações das sentenças em *tokens*, o processo chamado de *tokenization*, em que cada palavra de cada sentença é convertida para um objeto mais genérico chamado *token* que contém informações extraídas dessa palavra levando em consideração seu contexto utilizando modelos variados de extração da informação (regras, estatísticos ou aprendizado de máquina). Uma listagem de algumas das tarefas mais comuns em processamento de textos é apresentado a seguir.

- *Part-Of-Speech Tagging*

POS (*Part-Of-Speech Tagging*) é a tarefa de nomear cada palavra com uma marcação da usa função sintática na frase em que está inserida, nesta técnica as palavras são marcadas como verbo, advérbio, pronome, entre outros. (COLLOBERT et al., 2011)

[11] https://cloud.google.com/speech-to-text/docs/async-recognize
[12] https://cloud.google.com/speech-to-text/docs/streaming-recognize

- *Named Entity Recognition*

NER (*Named Entity Recognition*) é a tarefa de anotar determinadas palavras como sendo de algumas categorias como ”Pessoa”, ”Local”, ”Organização”, ”Moeda”, ”Data”, entre outras categorias (MANNING et al., 2014)

- *Chunking*

Também conhecido como *Parsing*, provem a anotação de sentenças de forma semântica como verbal ou nominal analisando a correlação das palavras nela contidas. (COLLOBERT et al., 2011)

- Análise de sentimento

Objetiva, comumente, a classificação do texto em Negativo ou Positivo (PANG; LEE et al., 2008). Neste caso pode ser utilizado o sentimento médio, valor de 0 a 1, de cada palavra que é obtido através de um processo de anotação manual de textos, no qual cada palavra é anotada com algum valor nessa escala, por final o algoritmo retira a média que essa palavra representa nas sentenças. Então é composto uma valor para a sentença inteira.

- *Lemmatization*

É a tarefa de conversão de uma palavra em sua forma ”base”(MANNING et al., 2014), assim evitando que algoritmos considerem palavras iguais, porém em tempos verbais diferentes, como sendo diferentes, dessa forma é possível uma redução na complexidade e aumento da eficácia de algoritmos.

- Remoção de *Stopwords*

Stopwords são palavras que aparecem com grande frequência nos textos e carregam pouco ou nenhum significado, comumente servem para compor uma função sintática no contexto mas não imbuem sentido quando presentes. A presença dessas palavras podem significar a efetividade dos métodos de extração de características com a grande repetição dessas palavras os textos ficam mais longos, causando maior complexidade no processamento.(EL-KHAIR, 2006) Cerca de 30% a 50% das palavras contidas em um texto são consideradas *stopwords*.(SCHÄUBLE, 2012)

- *Bag-of-Words*

Método que objetiva representar numericamente um conjunto de palavras de forma que cada palavra única é retratada com um número dentro de um sequências de palavras que compõe o texto em questão. Esse número é calculado pela quantidade de sua repetição durante o texto. Nesse método as características sintáticas do documento são perdidas. Esse método é comumente utilizados em algoritmos estatísticos de extração de informação.(IRIE; SCHLÜTER; NEY, 2015)

- *Term frequency–Inverse document frequency*

Medidas de frequências como a de uma BoW (*Bag-of-Words*) simples sofrem na busca de palavras que melhor definem o texto por enfatizas demasiadamente palavras com alta frequência de algumas palavras em detrimento das com menor frequência. Para melhorar esse processo o Tf-Idf (*Term frequency–Inverse document frequency*) adiciona mais um camada nesse cálculo. Esse termo pode ser divido em 2 partes:(CHOWDHURY, 2010)

1. *Term frequency*

Representa a simples frequência das palavras dentro de um texto ou documento.

$$tf(t,d) = \frac{t_d}{T_d}$$

Onde:

- t representa o termo (palavra).
- d representa o documento (texto).
- t_d representa a quantidade de vezes em que t aparece em d.
- T_d representa a quantidade de termos únicos presente em d.

2. *Inverse document frequency*

Calcula uma medida de quantidade de informação que uma determinada palavra representa no texto, gerando uma medida de ”peso”do quão importante essa palavra é para o texto. É representada pela função:

$$idf(t,D) = \log \frac{N}{n_t}$$

Onde:

- t representa o termo (palavra).
- D representa a coleção de documentos (textos).
- N representa a quantidade total de documentos (d) na coleção (D).
- n_t representa a quantidade de documentos (d) em que (t) está presente.

Por fim é produzida uma listagens de palavras e coeficientes que as define com a função que aglutina os dois conceitos:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

3.4 Aprendizado de máquina

Aprendizado de máquina é a área de conhecimento voltada em desenvolver maneiras de possibilitar através da computação a capacidade de aprendizado similar a do humano, com o objetivo de que problemas sejam solucionados sem a necessidade da configuração expressa de como agir a todas situações que o programa possa encontrar. Comumente utiliza dados históricos como aprendizado para assim ser capaz de predizer novos valores, dessa forma se demonstra um sistema mais flexível e assertivo em muitos casos.

Aprendizado de máquina é comumente subdividido em:

1. Aprendizado supervisionado

Utiliza dados históricos anteriormente classificados para, a partir dos parâmetros e a classe de cada instância, para construir um modelo matemático genérico o suficiente para conseguir lidar com exemplos já apresentados e novos de maneira satisfatória.(KOTSIANTIS; ZAHARAKIS; PINTELAS, 2007)

2. Aprendizado não-supervisionado

O aprendizado não supervisionado se apresenta como solução para casos em que os dados não possuem uma classe que os definem unicamente. Para solucionar esse problema o aprendizado não supervisionado procura formas de agrupar os dados de forma a encontrar padrões que possam ser úteis.(KOTSIANTIS; ZAHARAKIS; PINTELAS, 2007) Pode também ser utilizados em dados classificados com o objetivo de encontrar relações não antes conhecidas.

3. Aprendizado semi-supervisionado

Neste caso somente uma parcela dos dados estão classificados enquanto a maior parte da base de dados não está classificada, no caso de dificuldade em que nomear manualmente a amplitude da base em questão a utilização dos dados não classificados representa um aumento na assertividade do modelo gerado.(ZHU, 2005)

4. Aprendizado por reforço

Diferentemente do aprendizado supervisionado que é focado na classificação de uma determinada instância a partir de seus parâmetros, o aprendizado por reforço, através de recompensa e punição, busca encontrar o melhor caso a longo termo dado o estado atual e estados passados.(GHAVAMZADEH et al., 2015)

Neste trabalho o aprendizado de máquina supervisionado foi utilizado dado que a completude da base levantada possui sua devida classe correspondente.

3.4.1 Aprendizado de Máquina Supervisionado

Os métodos de aprendizado de máquina supervisionado dependem do conhecimento de qual classe cada uma das instâncias treinadas pertencem. Existem muitos tipos de classificadores supervisionados na literatura. Nas subseções seguintes serão apresentados os classificadores que serão utilizados para neste artigo. Os modelos aqui apresentados, são todos supervisionados.

3.4.1.1 Naive Bayes

Classificador mais simples dos apresentados por este documento. É um classificador estatístico baseado no Teorema de Bayes, tem como premissa que os atributos de cada instâncias são independentes da classe respectiva (Independência condicional):(RISH et al., 2001)

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

(ZHANG, 2004)

- Tal que A e B são eventos e $P(B) \neq 0$.
- P(A) é a probabilidade do evento A ocorrer.
- P(B) é a probabilidade do evento B ocorrer.
- P($A \mid B$) é a probabilidade condicional do evento A ocorrer dado que B é verdade.
- P($B \mid A$) é a probabilidade condicional do evento B ocorrer dado que A é verdade.

Que calcula a probabilidade de determinada instancia ser de uma classe baseado na probabilidade de seus atributos serem da mesma classe.

3.4.2 Árvore de Decisão

Algoritmo que se baseia nos atributos das instâncias da base de treino para construir uma árvore (estrutura de dados), que dado determinado valor de determinado atributo, é escolhido qual caminho na árvore a instância deve “seguir”, as ”folhas”da árvore treinada representa alguma classe.

A escolha dos atributos a serem escolhidos pela árvore é feita através do cálculo de entropia:

$$H(X) = -\sum p(X) \log p(X)$$

- Onde X é uma amostra dos exemplos de treinamento.
- p é a proporção de determinada classe em X.

A construção de uma árvore de decisão é guiada pelo objetivo de minimizar a entropia, isto é, a dificuldade de previsão da variável em questão.

3.4.3 Rede Neural

Inspirada no cérebro humano, as redes neurais contam com neurônios que são uma unidade de processamento que dado a inputs e pesos previamente selecionados aplica uma função e retorna algum valor.

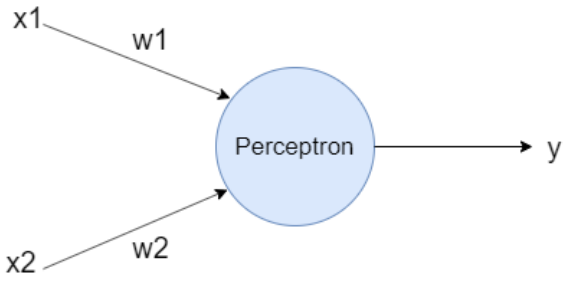

Figura 1 – Neurônio

Os neurônios que recebem as entradas do sistema são chamados de “Camada de entrada”, os que retornam o valor final do classificador de “Camada de saída” e todas as outras camadas são chamadas de “Camadas escondidas”. Cada neurônio soma cada valor de entrada (x) e atribui um peso(W) para cada um deles, também

chamado de “net”, que por sua vez é aplicada uma função continuamente diferenciável $f(net)$.

Os pesos de cada entrada encapsulam o conhecimento da rede. Assim, o treinamento consiste em encontrar pesos que minimizem a quantidade de erros da rede. Tal que a equação de peso é dada por:

$$E(W) = \frac{1}{2}(d - f)^2$$

- Onde ”d”é o valor desejado e ”f”é o valor obtido.

Para encontrar o menor erro, é necessário encontrar o inverso do vetor gradiente (visto que o mesmo encontrar o lado de maior subida) e é preciso atribuir um peso a este vetor (para o caso de não “pular” o ponto de mínimo local) que também é chamado de taxa de aprendizagem. Também chamado de Algoritmo Gradiente Descendente.

A arquitetura da rede utilizada é a *Feedfoward* na qual todos os neurônio de uma camada se conecta com todos os neurônios das camadas adjacentes. O método de correção do erro na rede é o *Backpropagation*, que significa que é o erro é calculado da camada de saída até a camada de entrada.

Quando um padrão é inicialmente apresentado à rede, ela produz uma saída. Após medir a distância entre a resposta atual e a desejada, são realizados os ajustes apropriados nos pesos das conexões de modo a reduzir esta distância.Este procedimento é conhecido como Regra Delta.

4 Metodologia

O trabalho iniciou com o objetivo de identificar a popularidade de cada canal com base nos episódios publicados, para isso foi necessário a criação de uma base para que análises pudessem ser realizadas. Neste capítulo é apresentado o processo de construção desta base com informações detalhadas de todo o processo.

Pelo fato de *Podcast* ter uma forma descentralizada de distribuição (podem ser acessados em agregadores, sites de hospedagem de mídia e diretamente nos sites dos canais) e não possuir informações consolidadas de todos esses canais de distribuição, obter o número certo de pessoas que escutam determinado episódio é uma missão complicada, ou mesmo informações sobre quais *Podcasts* são populares, fora isso, informações de uso das aplicações que distribuem esse tipo conteúdo simplesmente não são abertas ao público ou são informações indiretas como a quantidade de seguidores de determinado canal, o que dificulta ainda mais a obtenção destes dados.

Para medir o alcance de cada um dos canais, dados como quantidade de cliques em episódios de cada canal ou mesmo a quantidade de seguidores são utilizadas (quando a plataforma disponibiliza essas informações). No entanto, para obter informações mais precisas como a quantidade de usuários únicos que determinado *Podcast* obteve na última semana, essas informações não são de grande ajuda. Soluções como os *Smartlinks Chartable*[1] oferecem de maneira mais precisa informações como downloads únicos e cliques nas mais variadas ferramentas através de *cookie*[2] nas URL's de downloads dos episódios. Dessa forma é possível melhor mensurar o alcance que determinado *Podcasts* tem no mercado.

Para a construção da base dados a ser utilizada neste trabalho foi utilizado o serviço de ranqueamento disponibilizado abertamente pela Chartable, no qual é possível filtrar pelo país e algumas agregadores de *Podcast*.

4.1 Base de dados

Este trabalho utilizou dados de utilização do público brasileiro na plataforma iTunes para o levantamento da listagem de *Podcasts*, disponíveis em Cassificação Chartable.[3]

Para a construção da base de popularidade, as informações dos canais fo-

[1] https://chartable.com/
[2] https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
[3] https://chartable.com/charts/itunes/br

ram extraídas utilizando *Web scrapping* realizado no dia 20/11/2019, com a linguagem Python e a biblioteca Selenium, dos canais contidos em cada das categorias abaixo:

- *Arts*
- *Business*
- *Comedy*
- *Education*
- *Fiction*
- *Government*
- *Health & Fitness*
- *History*
- *Kids & Family*
- *Leisure*
- *Music*
- *News*
- *Religion & Spirituality*
- *Science*
- *Society & Culture*
- *Sports*
- *TV & Film*
- *Technology*
- *True Crime*

Cada uma dessas classificações contém até 3 páginas e um total de até 250 canais com o nome, URL da página do Chartable com informações do *Podcast* e a imagem de cada canal.

No total, 3932 URL's foram coletadas e, para cada uma, uma requisição foi feita com o objetivo de coleta das seguintes informações de cada *Podcasts*:

- Nome

- *RSS Feed*
- Categoria (listagem acima)
- Classificação (entre 1 e 250)

Na coleta do arquivos xml com informações do *RSS Feed* necessárias para análise futuras deste trabalho, uma nova requisição para cada *Podcast* foi realizada para o download deste arquivo. Após essa etapa, um total de 3801 arquivos foram obtidos. Resultando 131 canais removidos pelo motivo de indisponibilidade dos servidores de distribuição dos arquivos.

Para carregar os arquivos baixados na etapa anterior, foi utilizada a biblioteca escrita na linguagem python chamada xmltodict, foram selecionados os canais com as seguintes características:

1. *Podcasts* no idioma português.

Segundo a Podpesquisa[4] 62% dos brasileiros somente escuta *Podcasts* em português, enquanto 29,5% escuta na maior parte do tempo somente os nacionais. Então aqueles produzidos em idiomas diferentes foram removidos da base.

Essa informação é obtida acessando a informação contida na *tag* <language> que está contida dentro da *tag* <channel> que contém as informações do canal. Nesta etapa foram selecionados 1777 canais que tinham o idioma em questão.

2. *Podcasts* ativos.

Com o objetivo de garantir a pontualidade dos rótulos de popularidade, foram utilizados apenas o último episódio de cada canal que foi publicado nas duas semanas mais recentes.

Essa informação é obtida pela *tag* <pubDate> do primeiro episódio que é representado pela *tag* <item> que, por sua vez está dentro da *tag* <canal>. Com o total de 1158 canais ativos resultantes.

3. Erros na obtenção das informações.

Os *Podcasts* que apresentaram erros para obtenção das suas informações (*tags* faltantes, falha no download do *feed*, entre outros) foram removidos da base por falta de dados para verificar os critérios necessários.

[4] http://abpod.com.br/podpesquisa/

Após selecionados 1158 *Podcasts*, o último episódio de cada canal foi baixado para que o texto de cada um fosse transcrito na próxima etapa. Nessa fase alguns arquivos apresentaram problemas com *codecs*, no qual a biblioteca utilizada para carregar os arquivos não foi capaz realizar essa tarefa, esses canais foram removidos da base.

Um total de 1037 arquivos no formato mp3 (um episódio por canal) foram baixados, totalizando 765 horas, 41 minutos e 49 segundos de áudios e 52.9 Gigabytes de armazenamento.[5]

4.2 Transcrição

Para o processo de transcrição foi utilizado o serviço *Google Speech to Text* pela qualidade da transcrição retornada e o valor em créditos dado pelo Google para experimentar o serviço, pelo tamanho característico de muitos *Podcasts* (mais de 1 hora de duração) foi possível testar o serviço sem custo para o trabalho, o que não foi possível em outras ferramentas.

Para realizar a transcrição alguns passos foram necessários.

1. Conversão dos arquivos para um formato menos compactado.

Para que não haja perdas de informações na transcrição do texto, somente os *codecs* “FLAC” e “LINEAR16” nos formatos “WAV” ou “FLAC” são recomendados pela plataforma, então, para obter uma boa qualidade no processo transcrição, todos os arquivos foram transformados para um formato com menor compressão (WAV). A biblioteca “pydub” foi utilizada para carregar todos os arquivos como um “AudioSegment” e, assim, exportar para o formato wav com o comando “export(nome_do_arquivo.wav, format=’wav’)”.

2. Conversão de todos os arquivos para mono.

Neste trabalho todos os áudios foram transcritos como tendo somente um falante, a transcrição de texto com separação por locutor não é objeto de estudo e tem um maior custo na ferramenta utilizada. Esse processo foi realizado através do comando “set_channels(parâmetro)”, com parâmetro igual a 1, da classe “AudioSegment”.

Os arquivos convertidos totalizaram 244.0 Gigabytes de armazenamento.

3. *Upload* dos arquivos convertidos para o serviço de armazenamento.

[5] O aumento na quantidade de episódio da base se tornou inviável pela falta de recursos (tempo e créditos na plataforma *Google Cloud*) durante a realização da pesquisa.

Para a transcrição de áudios longo, é necessário que o arquivo esteja disponível dentro do sistema de armazenamento *Cloud Storage* do Google, os arquivos foram enviados manualmente pela simplicidade do processo.

4. Requisição ao serviço de transcrição.

Devido a duração prolongada do episódios a requisição assíncrona é recomendada para transcrevê-los, para essa função a biblioteca “google-cloud-speech” que é disponibilizada pelo Google para esse fim foi utilizada através do método “long_running_recognize” da classe “SpeechClient”, onde os parâmetros ”configuração”e ”audio_url”são objetos do tipo “RecognitionConfig” com informações como a URL do arquivo dentro do *Cloud Storage*, o idioma de transcrição, no caso “pt-BR”, e o *encoding* (LINEAR16) dos arquivos. Um total de U$ 1200,00 em forma de crédito foram gastos para realizar a transcrição dos arquivos.

A distribuição final das categorias coletadas é apresentada em 2, é possível notar que algumas das classes tem uma relação destoante das demais classes, em que os canais considerados *Long-tail* (canais não populares) estão presentes em baixa quantidade em relação aos considerados populares. Isso ocorre nas categorias ”*Fiction*”e ”*Government*”. Enquanto a quantidade mediana de canais populares e *Long-tail* é, respectivamente, 14 e 46 episódios.

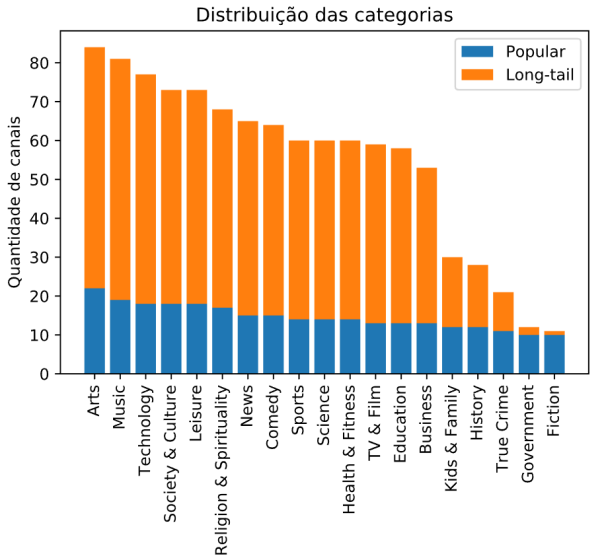

Figura 2 – Distribuição das categorias na base com separação por popularidade

5 Experimentos

Nesta seção é apresentado os desafios e achados deste trabalho a partir da base gerada no capítulo anterior, inicialmente é apresentado resultados da correlação entre termos presentes nos textos e a popularidade refente ao mesmo, posteriormente modelos de aprendizado de máquina são utilizados na tentativa de encontrar características e caminhos que possam ser uteis na predição de popularidade de canais a partir das informações contidas nos episódios.

5.1 Análise de correlação

Durante o processo de transcrição há uma perda de informação do texto resultante em relação ao áudio, seja por questões de ruídos e interferências no arquivo de áudio ou por erro do algoritmo de transcrição. A análise é possibilitada pela grande quantidade de palavras contidas nos textos, que, mesmo havendo erros, faz com que o efeitos desses erros sejam minimizados no resultado final.

Para a execução desta análise foi necessária uma etapa de pré-processamento da base de dados, as bibliotecas escritas na linguagem Python foram ”scikit-learn”e ”pandas”foram utilizadas para a execução do seguintes passos:

- Remoção de *Stop-words*
- *Lemmatization*

Após os textos serem pré-processados, aplicou-se o conceito de Tf-Idf e *Bag-of-Words* para a obtenção de palavras que identificassem melhor os textos. Os dados resultantes foram utilizados para o cálculo da correlação e do coeficiente de correlação de Pearson. Os resultados obtidos são demonstrados abaixo:

Aplicando o conceito para todas as categorias, tem-se:[1]

Os resultados encontrados em 1 e 2 demonstram uma baixa correlação entre palavras contidas em episódios e na popularidade quando considerando todas as categorias simultaneamente.

Levando em consideração cada categoria de forma isolada e aplicando o mesmo critério, os seguintes resultados foram obtidos[2]:

[1] Valor de correlação arredondado na segunda casa decimal, enquanto *p* na quarta casa decimanl

[2] Como os resultados obtidos nesta análise utilizando tanto *bag-of-word* quanto Tf-Idf foram similares somente os resultados do segundo modelo são mostrados.

Maiores correlações	Correlação	p
primoroso	0.13	0.0000
salsicha	0.12	0.0001
leste	0.12	0.0002
sistematicamente	0.11	0.0003
defunto	0.11	0.0004
recomendável	0.11	0.0004
refletem	0.11	0.0004
bruxa	0.11	0.0005
pureza	0.10	0.0007
arrecadar	0.10	0.0009

Tabela 1 – Correlação entre palavras e popularidade considerando todas as categorias utilizando *Bag-of-Words*

Maiores correlações	Correlação	p
arrecadar	0.11	0.0003
primoroso	0.11	0.0003
salsicha	0.11	0.0005
vendido	0.11	0.0005
goulart	0.11	0.0006
assistência	0.10	0.0007
seguinte	0.10	0.0008
bruxa	0.10	0.0010
five	0.10	0.0010
comentário	0.10	0.0011

Tabela 2 – Correlação entre palavras e popularidade considerando todas as categorias utilizando Tf-Idf

	Correlação	p
canção	0.44	0.0001
copiar	0.43	0.0001
mimi	0.43	0.0001
prontamente	0.43	0.0001
telefonar	0.43	0.0001
principalmente	-0.23	0.0450
barrir	-0.23	0.0469
pago	-0.22	0.0521
importar	-0.22	0.0549
parceiro	-0.22	0.0552

Tabela 3 – Correlação entre palavras e popularidade da categoria *Arts* utilizando Tf-Idf

	Correlação	p
segue	0.46	0.0004
criar	0.44	0.0008
ótimo	0.44	0.0008
administrar	0.43	0.0010
pausar	0.43	0.0010
vivo	-0.28	0.0411
sensação	-0.27	0.0502
verdadeiro	-0.26	0.0529
finalizar	-0.26	0.0562
crescimento	-0.25	0.0691

Tabela 4 – Correlação entre palavras e popularidade da categoria *Business* utilizando Tf-Idf

	Correlação	p
claro	0.33	0.2976
terminar	0.30	0.3412
nenhum	0.30	0.3488
história	0.29	0.3564
contar	0.29	0.3630
abastecer	-1.0	0.0
abundante	-1.0	0.0
acolher	-1.0	0.0
alegrar	-1.0	0.0
amanhecer	-1.0	0.0

Tabela 5 – Correlação entre palavras e popularidade da categoria *Fiction* utilizando Tf-Idf

As tabelas 3 e 4 são exemplos da melhoria na correlação entre palavras contidas no texto e a classificação dos canais. O aumento na correlação combinado com baixo valor de p foi considerável e constante por todas as categorias da base de dados.

Devida a quantidade de canais coletados das categorias *Fiction* e *Government* serem bastantes desbalanceados, os resultados obtidos (5 e 6) foram de pouca ajuda nesta análise.

Com o objetivo de prever a popularidade de cada canal, foram utilizados alguns algoritmos de aprendizado de máquina e seus resultados são apresentados a seguir.

5.2 Aprendizado de máquina

Para a análise da predição foram utilizados os seguintes algoritmos e respectivos parâmetros:

	Correlação	p
preciso	0.45	0.0803
exatamente	0.34	0.1910
conversar	0.34	0.1961
brasil	0.34	0.1976
estar	0.34	0.2006
banca	-0.98	0.0000
digital	-0.98	0.0000
jornal	-0.96	0.0000
horizonte	-0.93	0.0000
solução	-0.89	0.0000

Tabela 6 – Correlação entre palavas e popularidade da categoria *Government* utilizando Tf-Idf

1. *Naive Bayes*
    - Não possui parâmetros
2. Ávore de decisão
    - Mínimo número de objetos: 1
    - Poda: Não
    - Critério: *Entropy*
3. Rede Neural
    - Número de neurônios na camada escondida: 100
    - Função de ativação: *Rectifier*
    - Taxa de aprendizado: 0.001
    - Épocas: até 200
    - Momentum: 0.9

Antes de inserir os dados nos algoritmos listados acima, a base de dados foi pré-processada para garantir um resultado mais confiável. Todos as categorias tiveram seus a seus dados equalizados em relação à popularidade, isto é, a quantidade de instâncias populares e não-populares foram tornadas iguais. O texto de cada episódio passou pelo mesmo tratamento aplicado na sessão de categoria, isto é, *Stop words* foram removidos e radicais foram extraídos.

Primeiramente tentou-se prever a popularidade utilizando todos os canais na base de dados e Tf-Idf na seleção de atributos, o resultado encontrado não foi muito favorável com valores próximos à 50% e pode ser visto na figura 3.

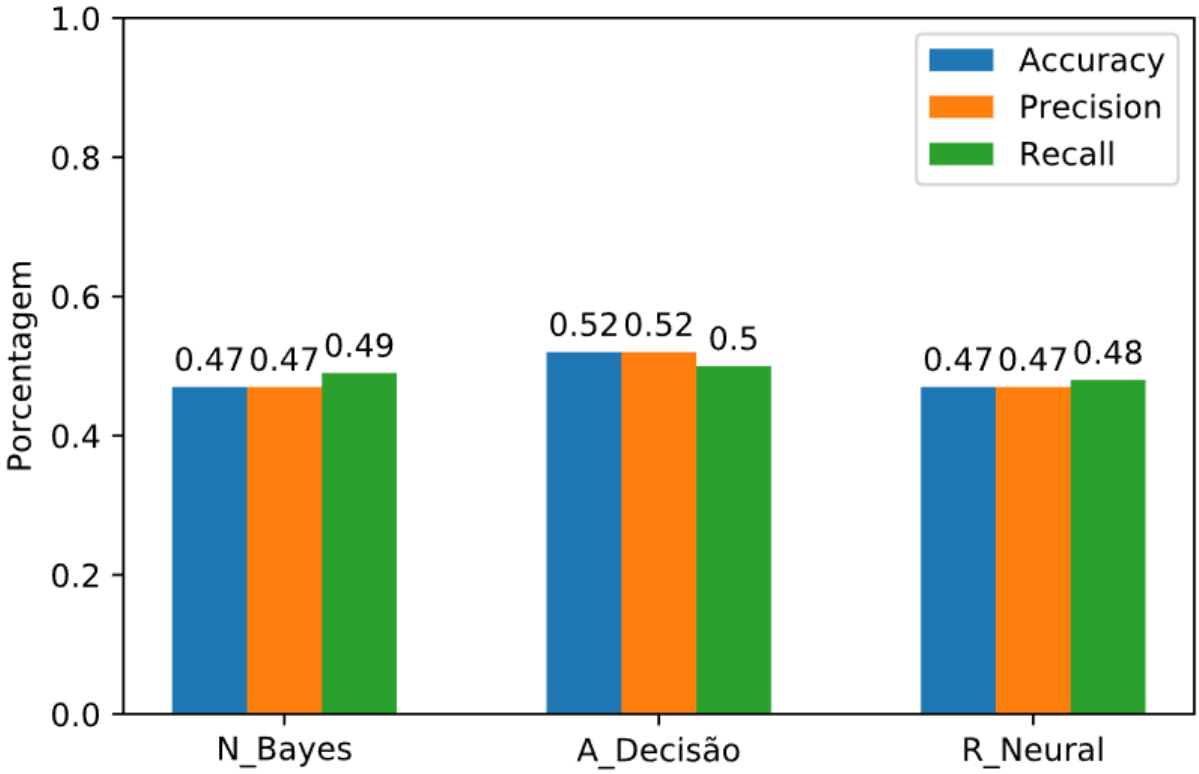

Figura 3 – Resultados obtidos atravéz dos modelo de aprendizado de máquina utilizando todas as classes da base de dados

Como os resultados de correlação foram favoráveis para uma análise por categoria, o mesmo método foi aplicado, porém reduzindo a base a aqueles que pertencerem à categoria em análise.

Nem todas as categorias reproduziram bons resultados na classificação, porém em algumas delas como *History*, resultados melhores foram obtidos como mostra a figura 4.

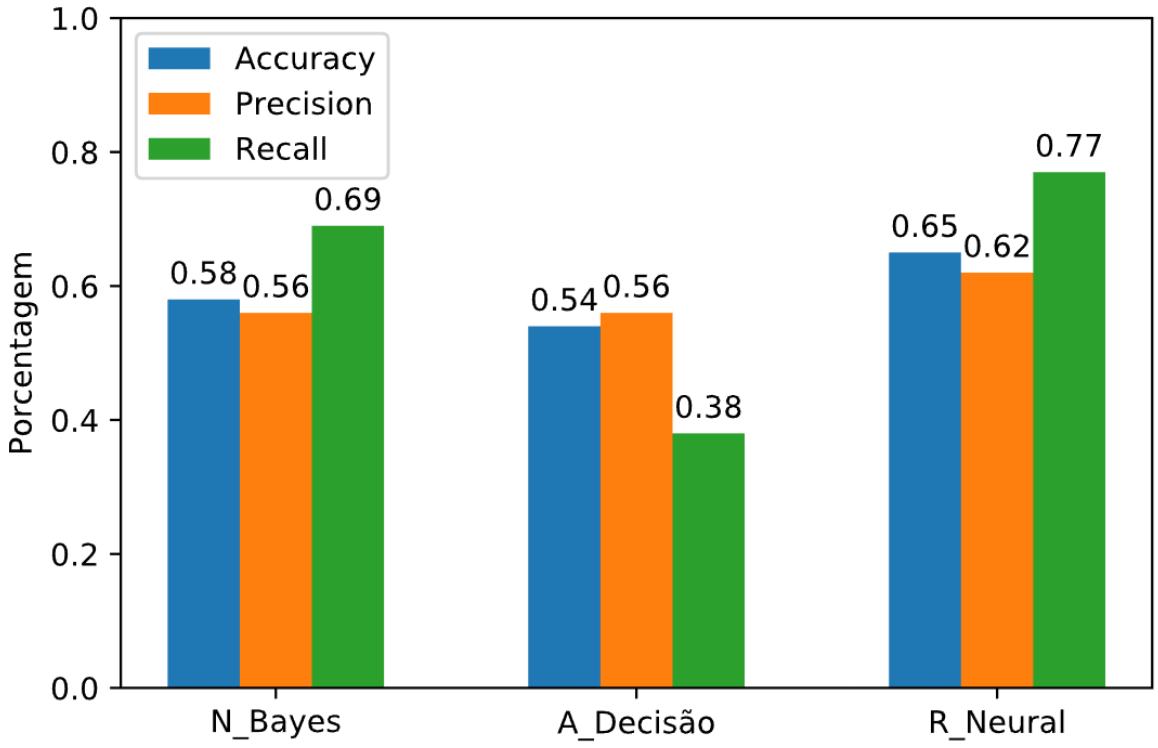

Figura 4 – Resultados obtidos atravéz dos modelo de aprendizado de máquina utilizando somente a classe *History*

6 Avaliação de resultados e Trabalhos futuros

Ainda existe muito a ser analisado para uma boa classificação dos canais, mas sem dúvida a análise textual representa uma característica importante para obtenção deste resultado. Através da construção da base de *Podcasts* pela transcrição dos episódios, foi possível notar correlações favoráveis entre suas palavras e popularidade, uma primeira análise utilizando algoritmos de aprendizado de máquina notou-se um resultado favorável em algumas categorias indicando que, dependendo da categoria, a análise textual pode ter um maior impacto na construção de serviços relacionados.

Outras informações textuais podem ser utilizadas para melhorar os resultados, informações como a descrição do canal, que pode representar um viés na argumentação dos locutores, informações como acontecimentos recentes podem resultar numa maior procura momentânea por determinado tópico, um processo de extração textual capaz de separar por locutor pode ser utilizado para melhor medir o que está sendo discutido, entre outras possíveis melhorias.

No entanto, *Podcast* não é só texto, (YANG et al., 2019) propõe a utilização de informações normalmente relacionadas à musicas na ajuda da classificação por popularidade, a verificação de ruídos, tempo de silêncio, frequência de atualização dos episódio e outras características podem ajudar nesta tarefa.

Referências

CHOWDHURY, G. G. *Introduction to modern information retrieval*. [S.l.]: Facet publishing, 2010. Citado na página 24.

COLLOBERT, R. et al. Natural language processing (almost) from scratch. *Journal of machine learning research*, v. 12, n. Aug, p. 2493–2537, 2011. Citado 2 vezes nas páginas 22 e 23.

DAVIS, K. H.; BIDDULPH, R.; BALASHEK, S. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, ASA, v. 24, n. 6, p. 637–642, 1952. Citado na página 19.

EL-KHAIR, I. A. Effects of stop words elimination for arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences*, v. 4, n. 3, p. 119–133, 2006. Citado na página 23.

GHAVAMZADEH, M. et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, Now Publishers, Inc., v. 8, n. 5-6, p. 359–483, 2015. Citado na página 26.

GLEZ-PEÑA, D. et al. Web scraping technologies in an api world. *Briefings in bioinformatics*, Oxford University Press, v. 15, n. 5, p. 788–797, 2013. Citado na página 17.

GOTO, M.; OGATA, J. Podcastle: Recent advances of a spoken document retrieval service improved by anonymous user contributions. In: *Twelfth Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2011. Citado na página 14.

IRIE, K.; SCHLÜTER, R.; NEY, H. Bag-of-words input for long history representation in neural network-based language models for speech recognition. In: *Sixteenth Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2015. Citado na página 24.

JUANG, B.-H.; RABINER, L. R. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, v. 1, p. 67, 2005. Citado na página 19.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, v. 160, p. 3–24, 2007. Citado na página 25.

LEE, K.-F. On large-vocabulary speaker-independent continuous speech recognition. *Speech communication*, Elsevier, v. 7, n. 4, p. 375–379, 1988. Citado na página 19.

MANNING, C. et al. The stanford corenlp natural language processing toolkit. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. [S.l.: s.n.], 2014. p. 55–60. Citado na página 23.

MITCHELL, R. *Web Scraping with Python: Collecting More Data from the Modern Web*. [S.l.]: ”O’Reilly Media, Inc.”, 2015. Citado na página 17.

MIZUNO, J.; OGATA, J.; GOTO, M. A similar content retrieval method for podcast episodes. In: IEEE. *2008 IEEE Spoken Language Technology Workshop*. [S.l.], 2008. p. 297–300. Citado na página 14.

OGATA, J.; GOTO, M. Podcastle: Collaborative training of acoustic models on the basis of wisdom of crowds for podcast transcription. In: *Tenth Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2009. Citado 2 vezes nas páginas 14 e 16.

PANG, B.; LEE, L. et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, Now Publishers, Inc., v. 2, n. 1–2, p. 1–135, 2008. Citado na página 23.

PRESS, O. U. *Podcasts definition*. 2004. Disponível em: <https://www.oed.com/viewdictionaryentry/Entry/273003>. Citado na página 16.

RECORDS, G. W. *Longest audio only live-stream*. 2012. Disponível em: <https://www.guinnessworldrecords.com.br/world-records/373222-longest-uninterrupted-live-webcast-audio-only>. Citado na página 16.

RISH, I. et al. An empirical study of the naive bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. [S.l.: s.n.], 2001. v. 3, n. 22, p. 41–46. Citado na página 26.

RUDNICKY, A. I.; HAUPTMANN, A. G.; LEE, K.-F. Survey of current speech technology. *Communications of the ACM*, Citeseer, v. 37, n. 3, p. 52–57, 1994. Citado na página 18.

SCHÄUBLE, P. *Multimedia information retrieval: content-based information retrieval from large text and audio databases*. [S.l.]: Springer Science & Business Media, 2012. v. 397. Citado na página 23.

YANG, L. et al. Understanding user interactions with podcast recommendations delivered via voice. In: ACM. *Proceedings of the 12th ACM Conference on Recommender Systems*. [S.l.], 2018. p. 190–194. Citado na página 15.

YANG, L. et al. More than just words: Modeling non-textual characteristics of podcasts. In: ACM. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. [S.l.], 2019. p. 276–284. Citado 2 vezes nas páginas 14 e 40.

ZHANG, H. The optimality of naive bayes. *AA*, v. 1, n. 2, p. 3, 2004. Citado na página 26.

ZHU, X. J. *Semi-supervised learning literature survey*. [S.l.], 2005. Citado na página 26.