ARNALDO PEDROSO

FAKE NEWS – ANALISANDO TEXTOS E DESCOBRINDO VERDADES

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Governança e Inovação de Tecnologias digitais com sustentabilidade.

São Paulo
2019

ARNALDO PEDROSO

FAKE NEWS – ANALISANDO TEXTOS E DESCOBRINDO VERDADES

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologias digitais com sustentabilidade.

Área de Concentração: MBA

Orientador: Profa. Rosangela de Fátima Pereira Marquesone

São Paulo
2019

FICHA CATALOGRÁFICA

Pedroso, Arnaldo
FAKE NEWS – ANALISANDO TEXTOS E DESCOBRINDO VERDADES / A. Pedroso -- São Paulo, 2019.
64 p.

Monografia (MBA em MBA em Tecnologias digitais com sustentabilidade) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.Algoritmos 2.Padrões de Textos I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

DEDICATÓRIA

Dedico esse trabalho aos meus pais, que me ensinaram a importância dos estudos, a minha esposa e amor da minha vida, que sempre esteve ao meu lado me apoiando nas horas mais difíceis e me incentivando a superar limites. Por fim, mas não menos importante, às minhas filhas, por me fazerem sorrir. E ao José Matheus Silvério Júnior (in memoriam), grande programador que me ajudou com parte importante do código.

AGRADECIMENTOS

À Universidade de São Paulo – USP que me proporcionou um ambiente adequado aos estudos.

À Escola Politécnica da Universidade de São Paulo – EPUSP que através de seu corpo docente, me desafiou a encontrar meus limites e estabelecer novos parâmetros.

Ao PECE – Programa de Educação Continuada em Engenharia que possibilitou a realização desse curso de MBA, com uma qualidade ímpar e de tema tão importante para o futuro do planeta.

Aos meus pais que, desde muito cedo, me mostraram a importância do saber. Sem os incentivos que eles me deram, minha história seria diferente e provavelmente este trabalho não existiria.

RESUMO

Notícias e boatos falsos sempre existiram, entretanto, a internet nos tornou expostos a muito mais informações e numa velocidade de consumo tão fugaz, que por vezes, se torna inviável analisar todas as fontes para verificar a veracidade dos dados. Desde que as eleições presidenciais estadunidenses de 2016, que supostamente foram alteradas devido as *fake news*, ou notícias falsas, que foram espalhadas*,* o termo está presente nos noticiários ao redor do mundo. Diariamente encontramos matérias de jornais falando sobre *fake news* que prejudicaram alguma empresa ou pessoa. Por se tratar de um problema latente, o objetivo desse trabalho é conceituar o que é *fake news*, traçando uma linha histórica de como e por que elas têm sido espalhadas no decorrer dos anos, além de apresentar um algoritmo, com oitenta por cento de eficácia, que analisa textos e através de aprendizado de máquina, faz a classificação e predição de veracidade dos mesmos.

Palavras-chave: Veracidade, Fake News, Riscos, Algoritmos, Padrões de Textos.

ABSTRACT

False news and rumors have always existed; however, the internet has made us exposed to much more information and at such a fleeting speed of consumption that it is often unfeasible to analyze all sources to check the truth of the data we read. Since the US presidential elections of 2016 were supposedly altered due to fake News, which have been scattered, the term is present in the news around the world. Every day we find newspaper articles talking about fake news that hurt some company or person. Because it is a latent problem, the objective of this work is to conceptualize what is fake news, drawing a historical line of how and why they have been spread over the year. Besides that, an algorithm will be showed up. It has eighty percent of efficiency, and using a machine learning, it can analyze texts and predict if the text analyzed is true or not.

Keywords: Veracity, Fake News, Scratches, Algorithms, Text Patterns.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
IFCN	*International Fact-Checking Network*
JSON	*JavaScript Object Notation*
SGBD	*Sistema Gerenciador de Banco de Dados*
SMB	*Server Message Block*
URL	*Uniform Resource Locator*

SUMÁRIO

.

1. INTRODUÇÃO

Desde que seres humanos vivem em grupos, rumores e notícias falsas sempre se propagaram, antes mesmo da prensa ser inventada (BURKHARDT, 2017). É impossível evitá-las. O que podemos fazer é buscar minimizar os impactos que elas causam (HABER, 2017). Embora seja impossível traçar quando o termo foi cunhado, podemos dizer que, atualmente, ele passou a ser mais difundido depois das eleições presidenciais norte-americanas de 2016 (JANG e KIM, 2017).

Curioso notar que partidários do atual presidente americano Donald Trump, reclamaram da postura da rede de notícias CNN, afirmando que eles publicaram notícias falsas para prejudicar o candidato republicano, já os democratas disseram que as eleições foram muito influenciadas por notícias falsas de outras fontes (JANG e KIM, 2017). Os dois lados se acusaram mutualmente de propagarem notícias falsas, há uma dificuldade enorme no processo de verificação dos fatos e consequentemente apurar quem está com a razão.

O problema das *fake news* não se encontra apenas no âmbito da apuração da veracidade da informação, hoje com as mídias sociais e pessoas cada vez mais conectadas, as mensagens são espalhadas muito rapidamente. Estudos recentes mostram que 62% da população americana obtém notícias de mídias sociais (JANG e KIM, 2017), outro estudo mostra que os *baby boomers* são os mais afetados por essas notícias (GIBBONS, 2018), muitas vezes utilizadas apenas para validarem uma posição partidária que elas já possuem.

As *fake news*, ou notícias falsas, não ficam somente na esfera política, em janeiro de 2017 vários canais de tecnologia, veicularam informação sobre vulnerabilidade no protocolo SMB (*server message block*) e os riscos envolvidos. Tratava-se do *wanna cry*, um *ransonware* que se beneficiava de um bug da versão 1 do protocolo SMB (HABER, 2017). Embora a notícia não fosse totalmente falsa, pessoas mais experientes poderiam perceber que a informação se tratava de uma nota que perdurava há décadas e os riscos poderiam ser mitigados de diferentes maneiras. O frenesi que a notícia teve, levou várias empresas a buscar uma solução rápida para um problema que provavelmente elas nem tinham, ou que já havia sido mitigada, aumentando seus custos operacionais e elevando o risco de uma interrupção massiva de seus serviços (HABER, 2017).

1.1. Motivação

Não é de hoje que podemos afirmar que a tecnologia da informação aumentou exponencialmente as formas de interações humanas. Estudos de 2012 já mostravam como a internet tornou-se o mais poderoso meio de comunicação do século XXI (FERREIRA MENDES DE SOUZA, FERREIRA BORGES, *et al.*, 2012).

Há 20 anos, empresas do setor automobilístico ocupavam o topo das maiores empresas do mundo, hoje as empresas de alta tecnologia ocupam esse papel, o crescimento dessas companhias, surgiu da necessidade do ser humano de se comunicar. “Comunicação, networking e compartilhamento de informações estão entre as principais razões para usar mídias sociais” (KOOHIKAMALI e SIDOROVA, 2017).

As mídias sociais acabaram se tornando uma fonte de combate à censura e uma grande arma para a liberdade de expressão ao redor do mundo. Entretanto, outro problema apareceu. “A democratização da comunicação traz à tona um fator preocupante: web-atores difundindo informações com alcance global – e muitas vezes sem nexo com a autoria” (RAMONET, 2012).

As notícias passaram a se propagar sem qualquer filtro ou compromisso com a veracidade das informações. Difamações se tornaram constantes nos jornais e outras mídias. Algumas pessoas parecem não se preocupar com os impactos que essas notícias falsas podem causar, e não é incomum ouvirmos o ditado popular: “quem não deve não teme”, porém a intimidade, vida privada, honra e a imagem das pessoas são invioláveis, e legalmente a constituição brasileira dá direito a indenização pelo dano material ou moral decorrente de sua violação (COELHO LOBO DE CARVALHO e KANFFER, 2018).

Além disso, a disseminação de *fake news* atrapalha a distinção do que é verdadeiro e falso e representa uma amaça a jornais e à própria democracia (SPINELLI e SANTOS, 2018). Pesquisas direcionadas para o público com acesso à internet, mostram que 56% da população confia nas organizações de mídia, e a porcentagem cai para 54% quando se refere aos profissionais (SPINELLI e SANTOS, 2018). Esse quadro alarmante mostra o descrédito da população nos meios de comunicação e na qualidade das informações que estão sendo veiculadas.

Ao observar a confiança que as pessoas pesquisadas depositam nos jornalistas, o número piora ainda mais. Sessenta e quatro por cento do público entrevistado acredita que os jornalistas recebem pressões políticas sob suas publicações e 65% creem que eles não estão livres de pressões por parte de interesses econômicos (SPINELLI e SANTOS, 2018). Com esses dados é possível afirmar que as pessoas culpam o jornalismo pela proliferação de notícias inverídicas, colocando em xeque o bem mais precioso que um jornalista deveria ter, credibilidade. A população continuará a procurar informação sobre o que está acontecendo à sua volta, sejam notícias do cotidiano da sua cidade, país, ou assuntos mais abrangentes. Ainda que o acontecimento esteja geograficamente longe do leitor, esse tentará trazer o significado do acontecimento para seu círculo pessoal, procurando fazer da notícia um significado à sua noção de mundo (MANJOO, 2008).

Com o descrédito nos meios de comunicação tradicionais, é natural imaginar que notícias enviadas por um amigo ou pessoas comuns, que fazem parte de um mesmo círculo de interesse, serve ao leitor como fonte de informações verídicas, que ele busca encontrar nos textos que recebe. Assim sendo, faz-se necessário algum mecanismo que ajude a população a ter uma ideia acerca da veracidade de uma informação. Para Rishika Sadam, um mecanismo de “alfabetização de notícias” é primordial para ajudar os usuários a decidirem quais fontes são confiáveis (SADAM, 2017).

1.2. Objetivo

O objetivo desse trabalho é apresentar, a partir da revisão da literatura, a definição do que é *fake news,* buscando identificar como ela se propaga e quem se beneficia com a veiculação delas.

Além disso, esse trabalho tem como objetivo desenvolver e disponibilizar um algoritmo capaz de predizer se um texto analisado é verdadeiro ou falso, através da utilização de algoritmos de *machine learning* para classificação do texto analisado e facilitar a identificação de fontes de informação confiáveis.

2. FAKE NEWS - POR QUE PRECISAMOS DE CLAREZA NA DEFINIÇÃO?

A tradução literal da palavra *fake news* para o português é: “notícias falsas”. Embora a transcrição da palavra pareça clara e concisa, não é trivial entender o que as notícias falsas realmente são, já que elas podem ser aplicadas em diversos contextos diferentes.

Sob a ótica das eleições para presidente dos Estados Unidos em 2016, Allcott e Gentzkow descreveram *fake news* como “notícias intencionalmente e comprovadamente falsas, podendo enganar os leitores” (ALLCOTT e GENTZKOW, 2017). É importante destacarmos “intencionalmente”, para excluirmos erros editoriais, não intencionais, e sátiras (SPINELLI e SANTOS, 2018), já que esse tipo de abordagem não foi feita, de forma explícita, com o intuito de enganar um público ou buscando algum tipo de favorecimento.

Otavio Frias se aprofunda mais no tema, para ele “o termo vem sendo utilizado para efeitos de esgrima retórica, ou seja, para desqualificar versões diferentes daquela abraçada por quem o emprega. Nesse sentido mais permissivo, *fake news* passam a ser tudo aquilo que me desagrada, não apenas fatos que contemplo de maneira diferente da exposta, mas interpretações das quais discordo com veemência e opiniões que me parecem abomináveis. O que é *fake news* para um fanático é verdade cristalina para o fanático da seita oposta.” (FILHO, 2018).

Podemos perceber que o enfoque dado por Allcott e Gentzkow está incluído na afirmação de Otavio Frias, entretanto é importante percebermos como Frias acerta ao afirmar que o termo foi banalizado, e hoje é utilizado sempre que há lados conflitantes tentando desqualificar os argumentos de seus opositores (FILHO, 2018).

Farhad Manjoo está alinhado com a ideia de Otavio Frias ao afirmar que “A mente humana tende a escolher informações que estejam alinhadas às suas crenças, atitudes e comportamentos, rejeitando o que é contraditório.” (MANJOO, 2008). Nicholas Jankowski também parece concordar com Frias ao afirmar que “o termo adquiriu status como um rótulo pejorativo para os meios de comunicação liberais e perdeu o significado comumente aceito” (JANKOWSKI, 2018).

Essas visões complementares são de vital importância para montarmos um algoritmo capaz de verificar a veracidade de um texto. Deve-se tomar cuidado para não chamar de *fake news*, notícias que não estão alinhadas aos próprios vieses

políticos e religiosos, e assim cometer o mesmo erro de quem propaga informações em nome de suas próprias convicções.

Thomas Jefferson escreveu que o preço a pagar pelos benefícios da liberdade de imprensa é ter de tolerar a existência de maus jornais (FEA, 2017). De acordo com Frias (FILHO, 2018, p. 39) "ao chamarmos de *fake,* textos imprecisos ou desalinhados com nossa visão de mundo, estamos cometendo um ato covarde de censura". Para Daniela Spudeit, a avaliação da informação pauta-se na avaliação dos conteúdos recuperados com base em critérios tais como a veracidade, a credibilidade, a confiabilidade e a qualidade da informação bem como a autoria (DANIELA, 2017).

Nem todas as dimensões descritas por Spudeit são fáceis de serem atestadas e, consequentemente, passíveis de serem usadas nesse trabalho, para classificar um texto como verdadeiro ou falso. Mas a pedra fundamental desse estudo baseia-se na qualidade da informação, como será explicado no Capítulo 3.

2.1. Quem se beneficia com as fake News?

Em um estudo conduzido pelo BuzzFeed (BATHKE, 2017)**,** chegou-se à conclusão que mais de 60 sites, identificados como divulgadores de informações falsas, tiveram receita utilizando o serviço do *Google AdSense* e outras importantes redes de anúncios (SPINELLI e SANTOS, 2018).

Se imaginarmos que o *core business* de empresas como Google e Facebook é gerar *clicks* para seus anunciantes, podemos nos perguntar: qual a real responsabilidade dessas empresas, com a veracidade do conteúdo que será veiculado em seus canais?

Diretamente o Google e Facebook também estão se beneficiando com as *fake news* que são espalhadas através de seus produtos, porém, não só essas grandes corporações estão se beneficiando com essas notícias, notícias falsas também foram produzidas, propositadamente, por adolescentes dos Balcãs e empreendedores americanos, que procuravam uma maneira de fazer dinheiro com propagandas (MAHESHWARI, 2016).

Os casos acima citados, são mais fáceis de serem entendidos e percebidos pelo grande público, mas existem as situações onde uma notícia é veiculada com o intuito de causar controvérsias, de forma mais velada. A intenção desse método é iniciar uma discussão e, através do aumento da cobertura sobre um determinado tema, gerar um

efeito chamado *agenda setting*. O benefício da *agenda setting* é difamar uma pessoa física ou jurídica para obtenção de benefícios próprios (VARGO, GUO e AMAZEEN, 2017).

Esse tipo de comportamento não é novo, a manipulação da informação e o poder interpretativo sempre existiram. Ela já esteve nas mãos da Igreja durante a Idade Média, logo depois passou para as mãos da ciência moderna, em luta contra a Igreja (COELHO BEZERRA, CAPURRO e SCHNEIDER, 2017).

Para Bezerra, Capurro e Schneider (COELHO BEZERRA, CAPURRO e SCHNEIDER, 2017), na luta política e econômica, existe um importante papel na disseminação de “verdades” ou “meias verdades” ou “três quartos de verdade” ou simplesmente “mentiras”, que podem ser insultos pessoais, difamações ou ameaças.

Muitas vezes o que está em jogo, não é forçar uma pessoa a ter as mesmas ideias que a sua, mas simplesmente atentá-las a um determinado tema, gerando medo e desconfiança na sociedade. A hipótese da *agenda setting* sugere que, embora a mídia não determine o que as pessoas irão pensar, determina em grande medida sobre o que irão pensar (COELHO BEZERRA, CAPURRO e SCHNEIDER, 2017).

Com base nessas análises, deve-se fazer uma pergunta simples, para saber se uma notícia pode ou não estar enviesada: para quem e para que a notícia serve?

2.2. Como garantir que uma notícia é verdadeira?

Na literatura há linhas de estudos que propõe uma abordagem mais abrangente na aquisição de fontes de informações confiáveis. Segundo Brisola e Romeiro (BRISOLA e ROMEIRO, 2018), cabe aos profissionais de biblioteconomia (Bacharéis e Licenciadas(os)), serem os agentes de transformação no que consiste a mediação das informações para além da informação dada, exposta no ambiente *web* e consumida por usuários.

Nessa linha de raciocínio, o profissional de biblioteconomia não seria responsável por auditar e validar as informações, mas traria trabalhos complementares que fizessem o leitor a chegar às suas próprias conclusões sobre o que é útil ou o que pode ser descartado por ele. Parece fazer sentido quando se trata de pesquisas para projetos acadêmicos, mas para pesquisas diárias de informação, parece ser uma abordagem inviável pela quantidade de informações a que somos submetidos.

Miller (MILLER, 2007) relaciona a qualidade da informação ao grau em que as necessidades dos usuários da informação são abordadas e lista dez dimensões para a qualidade da informação: precisão, pontualidade, integridade, coerência, formato, acessibilidade, compatibilidade, segurança e validade.

Atualmente o órgão internacional, IFCN (*International Factchecking Network*), procura validar algumas dessas dimensões. No Brasil apenas três agências são certificadas pelo IFCN: Lupa7, Truco8 e Aos Fatos9.

Ainda que exista um órgão que procure validar os fatos veiculados pelos veículos de comunicação, a forma como essas agências atribuem uma classificação a informação é um editorial e não um método científico (LIM, 2017).

O algoritmo desenvolvido nesse trabalho, e que será apresentado no Capitulo 3, analisa dimensões de qualidades atribuídas empiricamente, para identificar a veracidade do texto e, assim, trazer um método mais científico para a essa análise, segundo a abordagem proposta por Miller (MILLER, 2007).

3. ALGORITIMO

3.1. Escolhas para o desenvolvimento

O algoritmo foi desenvolvido utilizando a linguagem de programação Python e não reaproveitou código de outros trabalhos, isto é, os programas apresentados no Capítulo 3.8 foram concebidos para esse projeto e são de autoria do autor dessa monografia.

Os únicos códigos reaproveitados, são bibliotecas amplamente conhecidas e utilizadas pela comunidade de desenvolvedores Python. Elas serviram de apoio ao algoritmo desenvolvido, para prover conectividade a banco de dados, analisar código *html*, etc.

A linguagem Python foi escolhida pela sua simplicidade com o manejo de *strings*, outro ponto importante, considerado para a escolha de Python, foi a facilidade em encontrar bibliotecas que trabalhem com *machine learning,* simplificando o desenvolvimento do algoritmo.

Para gerar as dimensões que serão utilizadas para treinarmos nosso modelo de dados, conforme descrito no Capítulo 3.4, foram utilizadas *APIs* providos pela plataforma de aplicativos e serviços da Microsoft chamado *Azure*. As *APIs* utilizadas fornecem resposta em arquivos *JSON* que são facilmente manipulados por bancos de dados orientados a documentos.

No armazenamento das *URLs* já analisadas, foi utilizado o banco de dados NoSQL MongoDB. Como esse Sistema de Gerenciamento de Banco de Dados (*SGBD)* é orientado a documentos, a interação dele com os registros providos pela *Azure,* foi facilitada. A escolha de um banco de dados para armazenar as *urls* e textos já classificados anteriormente, permite a criação de uma biblioteca com notícias verdadeiras e falsas, além da geração de mais dados para treinar o algoritmo a cada execução, já que sempre que uma informação é analisada, o algoritmo guarda o resultado para melhorar sua precisão em análises futuras.

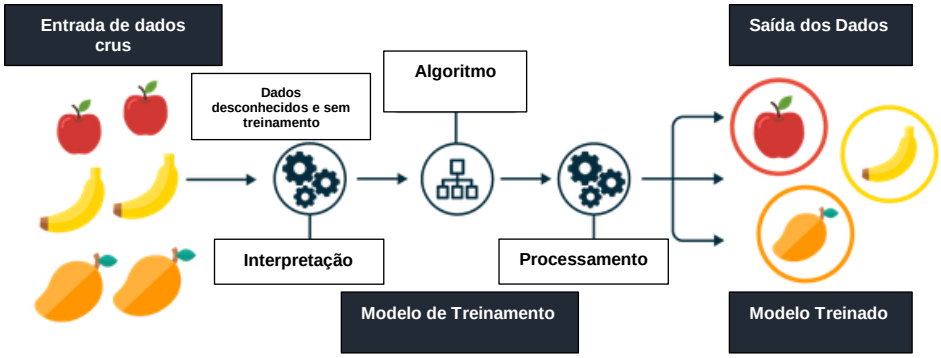
Para um melhor entendimento de como o algoritmo funciona, precisamos entrar em alguns conceitos de tecnologia que serão explicados detalhadamente nas seções a seguir.

3.2. O que é Machine Learning?

Machine Learning é um método de análise de dados que automatiza a construção de modelos analíticos. É um ramo da inteligência artificial baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana (SAS INSIGHTS, 2019).

Algoritmos de *machine learning* podem ser categorizados em dois tipos diferentes: supervisionados e não supervisionados (HASTIE, TIBSHIRANI e FRIEDMAN, 2008). Fundamentalmente, algoritmos supervisionados, classificam dados. Envia-se para o algoritmo, uma lista com itens já conhecidos, e treina-se a máquina de aprendizado. Assim, nas próximas iterações, os itens que serão inseridos como parâmetro, serão categorizados automaticamente pelo algoritmo.

Figura 1: Funcionamento de Algoritmos Supervisionados



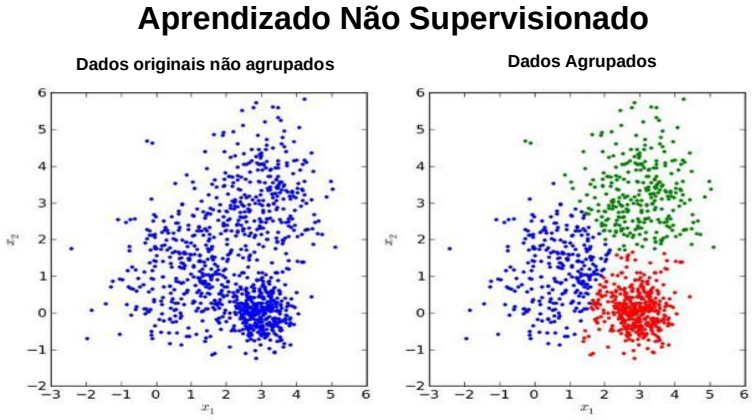
Fonte: (THAPLIYAL, 2018)

Na Figura 1, nota-se a entrada de dados que podem ser maçãs, bananas ou laranjas. Como os dados já são conhecidos o algoritmo cria um modelo que será utilizado nas próximas análises, assim numa próxima classificação o algoritmo conseguirá separar, adequadamente, as frutas que foram submetidas para serem classificadas (THAPLIYAL, 2018).

Um bom exemplo de algoritmo supervisionado é a identificação de spam e não spam numa lista de e-mails. Através de regras pré-definidas o sistema consegue separar corretamente as mensagens que serão disponibilizadas aos usuários.

Algoritmos não supervisionados funcionam de maneira diferente. São enviados conjuntos de dados para a máquina de aprendizagem, e a tarefa do algoritmo é procurar padrões, que muitas vezes não são identificados manualmente, e os agrupa em conjuntos diferentes (ABDULLAH, 2018).

Figura 2: Funcionamento de Algoritmos Não Supervisionados



Fonte: (ABDULLAH, 2018)

É notável a diferença entre algoritmos supervisionados e não supervisionados, conforme verificado na Figura 2, o algoritmo não supervisionado, não classifica os dados, mas busca padrões para agrupar os conjuntos em dados que parecem similares (ABDULLAH, 2018).

A diferença primordial entre aprendizado supervisionado e não supervisionado é que, no primeiro, utiliza-se dados já conhecidos para treinar o algoritmo. Já no aprendizado não supervisionado, busca-se achar uma representação mais informativa dos dados disponíveis. De acordo com Brownlee, classificação é sobre prever um rótulo e a regressão é sobre prever uma quantidade (BROWNLEE, 2017).

Geralmente os modelos criados em algoritmos supervisionados são mais simples que os modelos utilizados para condensar a informação em pontos mais relevantes, também conhecido como clusterização ou agrupamento de dados (HONDA, FACURE

e YAOHAO, 2017). Nesse trabalho, foi utilizado o método de classificação de dados, já que o intuito é relacionar as notícias analisadas de forma binária, isso é, catalogar as informações como “verdadeiras” ou “falsas”.

Dentro do método de classificação de dados, existem várias técnicas para separar esses, a saber: redes neurais artificiais, máquina de suporte vetorial (ou máquinas kernel), árvores de decisão, k-vizinhos mais próximos etc. Nesse trabalho foi utilizado árvore de decisão como metodologia de classificação da notícia.

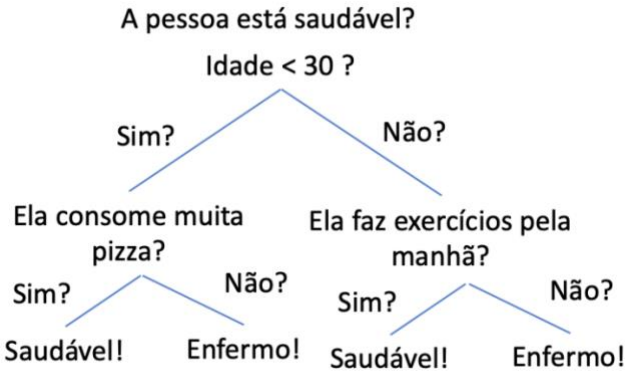
3.3. O que é Árvore de Decisão

Árvores de decisão são métodos de aprendizado de máquina supervisionado não-paramétricos, muito utilizados em tarefas de classificação e regressão (CAMPOS, 2017). O entendimento do funcionamento de uma árvore de decisão é simples. Cada nó da árvore armazena uma regra ou decisão a ser tomada.

No trabalho de Mara (DOTA, 2014) podemos ver que uma árvore é construída de cima para baixo e envolve três etapas principais: selecionar uma regra de divisão para cada nó interno; determinar os nós intermediários ou não terminais; atribuir rótulos de classe para os nós terminais.

Basicamente o método define as dimensões, ou regras, que serão utilizadas para o algoritmo classificar a informação analisada. Na Figura 3 temos um exemplo de como uma árvore de decisão funciona:

Figura 3: Funcionamento de Árvore de Decisão



Fonte: (KULKARNI, 2017)

Conforme ilustrado na Figura 3, esta árvore de decisão procura descobrir se uma pessoa está saudável. A primeira análise utilizada na tomada de decisão é a idade do sujeito analisado, segundo para a validade da idade, depois para a dieta alimentar. Se a pessoa examinada possui menos de trinta anos de idade e sua alimentação não é baseada em pizza, o algoritmo o classifica como saudável, por exemplo.

3.4. Dimensões utilizadas para a classificação da notícia

Conforme citado no Capítulo 2, foi escolhido as variáveis, ou dimensões analisadas, para se classificar uma notícia. Esses parâmetros serão utilizados no algoritmo de árvore de decisão para ajudar o *machine learning* a classificar os textos estudados.

Com o intuito de dar um enfoque mais científico e tentar evitar vieses, foram escolhidas seis dimensões:

- Percentual de identificação do texto com a língua portuguesa;
- Sentimento do texto;
- Percentual de palavras chaves encontradas no texto (em relação à quantidade total de palavras do texto) e relevância delas no motor de busca da Microsoft;
- Percentual de palavras chaves encontradas no texto (em relação à quantidade total de palavras do texto) e relevância delas na Wikipedia;
- Quantidade de palavras chaves encontradas no texto e listadas no motor de busca da Microsoft;
- Quantidade de palavras chaves encontradas no texto e listadas na Wikipedia.

Para a geração dessas dimensões foram utilizadas as *APIs* de busca da Microsoft, descritas na Tabela 1 (a *URL* principal para a chamada dessas *APIs* está descrita no Apêndice H). Importante salientar que, para o funcionamento do algoritmo proposto nesse trabalho, faz-se necessário a criação de uma conta na plataforma de nuvem *Azure* da Microsoft, bem como uma chave de acesso para utilização do recurso de reconhecimento cognitivo da Microsoft *Azure*.

Tabela 1: APIs de *Machine Learning* e Busca a Microsoft.

API	Descrição
languages	Essa API verifica o percentual identificado pelo machine learning do mecanismo de Busca da Microsoft, de identificação de um texto com uma determinada língua;
sentiment	Essa API verifica qual o percentual de "sentimento" de um determinado texto analisado. Por exemplo, um texto com viés "positivo" terá um percentual > 50 e tendendo a 100, já um texto com viés "negativo" terá um percentual < 50 e tendendo a 0;
Entities	Essa API identifica palavras chave no texto, utilizando machine learning de análise de texto, e verifica a relevância delas no mecanismo de busca da Microsoft e na Wikipedia.

Fonte: (MICROSOFT DOCS, 2019)

3.5. Dimensão: Reconhecimento do texto analisado com a língua portuguesa

Textos com mais erros de português ou utilização de muitas palavras de outros idiomas, receberão um percentual menor de identificação com a língua portuguesa, de acordo com o *machine learning* utilizado pelo motor de buscas da Microsoft.

A hipótese levantada por esse trabalho é de que muitos erros de português não são comuns em textos escritos por canais de comunicação mais confiáveis, já que existe um trabalho de revisão sobre o conteúdo escrito.

É natural imaginar que grandes mídias possuem uma preocupação maior com a veracidade da informação que elas estão propagando. Embora essa premissa não seja observada em todos os casos, ainda assim, grandes jornais ou portais de notícias da internet, possuem uma reputação maior a zelar se compararmos a escritores anônimos ou blogueiros. Por isso, supõem-se que, validar se um texto possui uma semelhança com a língua portuguesa, é importante para atestação da veracidade de uma publicação analisada pelo algoritmo proposto nesse trabalho.

Assim sendo, foi levantado a hipótese H1: textos escritos, utilizando a língua portuguesa com maior correção, isso é sem erros de ortografia e gramática, terão uma pontuação maior pela *machine learning* da Microsoft e terão uma tendência maior de serem notícias verdadeiras, já que os agentes disseminadores de *fake news,* possuem uma preocupação menor em produzir textos com qualidade gramatical mais precisa.

3.6. Dimensão: análise de sentimento do texto analisado

Outra hipótese levantada por esse trabalho está fundamentada no grau de imparcialidade observada no texto estudado. Textos com conteúdo mais jornalísticos, tendem a ser neutros (hipótese H2). Conforme descrito no conceito de *fake news,* o ser humano busca informação com base em suas crenças, assim, é plausível supor que, atores mal-intencionados, ao propagarem *fake news,* tendem a expor também suas convicções nos textos escritos tornado a publicação menos neutra.

3.7. Demais dimensões

As outras dimensões são desdobramentos da mesma observação. As palavras chaves, encontradas no texto analisado, também relevantes para o motor de busca da Microsoft ou Wikipedia, tendem a ser temas que já foram citadas em outras obras e com isso possuem uma chance menor de serem falsas.

A hipótese H3 está fundamentada na quantidade de palavras chaves criadas em cima do texto analisado, existe ainda uma quarta Hipótese (H4) que é a relevância deles na *Wikipedia*. Imagina-se que textos melhores elaborados e consequentemente com mais palavras chaves geradas pela *machine learning* da Microsoft, além de possuírem mais relevância, isso é, serem mencionados mais vezes na *Wikipedia,* terão uma tendência maior de serem verdadeiros.

3.8. Códigos criados para o funcionamento do algoritmo

Para analisar os textos, foram criados diversos programas em Python*,* tendo cada um com um propósito específico, conforme apresentado a seguir.

3.8.1. Arquivo - DatabasePool.py

A classe criada é utilizada para que os demais programas, que compõe esse sistema, acessem o banco de dados utilizado pelo algoritmo (Apêndice B).

3.8.2. Arquivo - myHtmlClass.py

Quando uma URL é transmitida via parâmetro para o algoritmo criado, é utilizado a classe myHtmlClass que é responsável pela limpeza do arquivo HTML analisado, assim, todas as marcações como *tags* de HTML, comentários, etc, são limpos, deixando um texto simples para análise (Apêndice C).

3.8.3. Arquivo - fakenews.py

O código desenvolvido e expresso nesse arquivo, é responsável por chamar a API da nuvem *Azure* da Microsoft. Através deste, são geradas as dimensões que serão analisadas para identificar as notícias que são falsas e as que são verdadeiras.

Este programa, também, treina o *machine learning.* A cada nova URL analisada, o algoritmo é retroalimentado, aumentando o conteúdo de sua base de dados, e melhorando a precisão da classificação que se torna mais acurada (Apêndice D).

3.8.4. Arquivo - algorithmFakeNews.py

Esse é o programa principal responsável por orquestrar os demais programas criados. As URLs analisadas, são lidas por esse programa através de um arquivo JSON chamado news.json, onde a URL que será analisada e um parâmetro (is_fake), que pode ser utilizado para treinar o algoritmo, ou deixar que ele faça a classificação, após prévio treinamento e aprendizado.

3.8.5. Arquivo de parâmetros – *urls* analisadas

A seguir é apresentado o exemplo de uma linha do arquivo news.json que é transmitido via parâmetro para o programa algorithmFakeNews.py:

{"url":"http://dicadeculinaria.com/native/tabua-degeladora/?utm_source=tab&utm_campaign=tabua&utm_source=tab&utm_campaign=tabua_desktop","is_fake":"0"}

O parâmetro *url* é a notícia a ser analisada, já o parâmetro *is_fake* deve ser preenchido da seguinte maneira:

“0” – quando queremos alimentar nossa base de dados com URLs cujo texto já sabemos que não é falso;

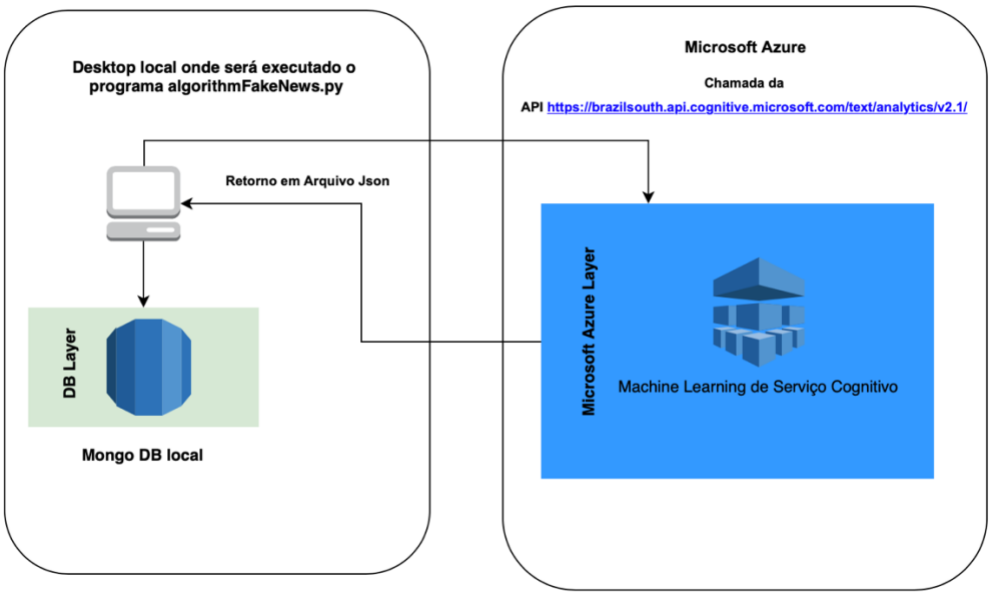
“1” – quando queremos alimentar nossa base de dados com URLs cuja informação sabemos que é falsa;

“” – Ao deixarmos esse parâmetro em branco, estamos sinalizando para o algoritmo que ele deve classificar a informação, apresentada na URL, como verdadeira ou falsa, baseado no modelo que passamos previamente para treiná-lo.

3.8.6. Chamadas internas e externas pelo algoritmo

Para o funcionamento do algoritmo é necessário ter um banco de dados mongodb instalado localmente na máquina onde será rodado o algoritmo, além disso é preciso uma subscrição na plataforma Microsoft *Azure* e uma chave de acesso ao recurso de cognição de textos, que é o motor utilizado pelo *Azure* para gerar as dimensões descritas do Capítulo 3.4 até o Capítulo 3.7. Na figura 4 abaixo citado, tem-se uma melhor compreensão da comunicação interna e externa do algoritmo.

Figura 4: Comunicação do algoritmo com banco de dados local e nuvem *Azure*

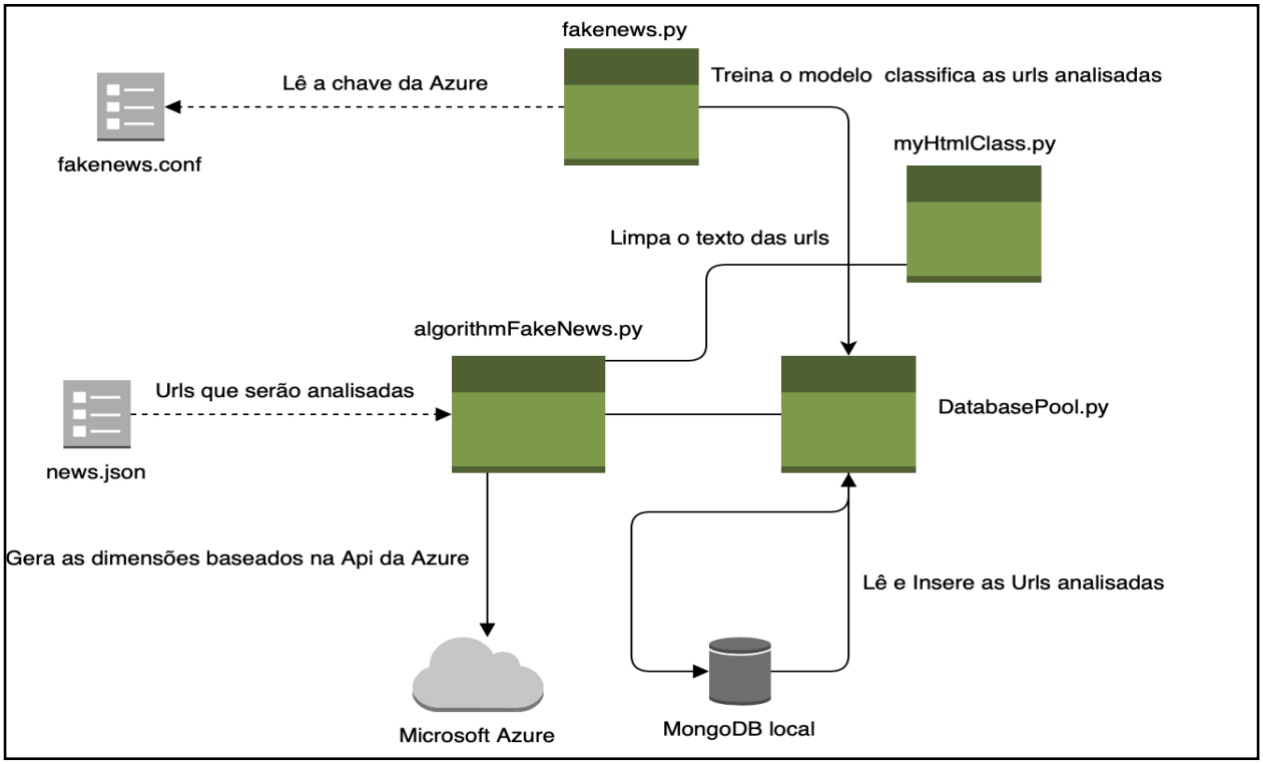


Fonte: Elaborado pelo autor, 2019

3.8.7. Comunicação entre os programas do algoritmo

A Figura 5 apresenta graficamente a comunicação entre os programas descritos anteriormente, além disso é possível verificar o propósito dos arquivos news.json e fakenews.conf, e como o algoritmo é retroalimentado para melhorar sua acurácia a cada execução.

Figura 5: Comunicação entre os programas do algoritmo



Fonte: Elaborado pelo autor, 2019

4. FONTES UTILIZADAS PELO MODELO

Para treinar o modelo com notícias verdadeiras, foram utilizados diferentes *websites* como UOL, Globo, iG, etc. Foram selecionadas notícias cotidianas ou com informações políticas que foram veiculadas por diferentes mídias, atestando assim sua veracidade.

Atualmente diversos portais possuem páginas especificas que analisam notícias falsas que estão sendo veiculadas. Essas páginas buscam informar o leitor que essas informações não passam de boatos. O site mais especializado nesse tipo de análise é o boatos.org.

Boa parte das notícias falsas que são desmentidas pelo boatos.org ou pelos demais portais, não são encontrados em sites de notícias ou *blogs*. Elas são espalhadas por *whatsapp* ou pelo *facebook* e ambos estão fora do escopo de análise do algoritmo apresentado.

O *whatsapp* está fora do escopo por não ser uma *url* que pode ser analisada. Já o *facebook* não é alcançado, por ser um site autenticado, impossibilitando o algoritmo de ler a informação veiculada dentro de uma página deste aplicativo.

Ainda assim as informações que puderam ser extraídas do site boatos.org, e encontradas em outros sites da internet, como sendo notícias verdadeiras, foram utilizadas pelo algoritmo. Outra fonte importante de notícias falsas foi o *twitter*. O restante dos dados falsos utilizados para montar o modelo de dados, foram extraídos de anúncios publicitários que prometiam tratamentos milagrosos para a cura de alguma doença crônica, como diabetes por exemplo.

A dificuldade para encontrar notícias, genuinamente falsas, em grandes portais ou jornais, influenciou negativamente o algoritmo, conforme explorado na Seção 5.3.

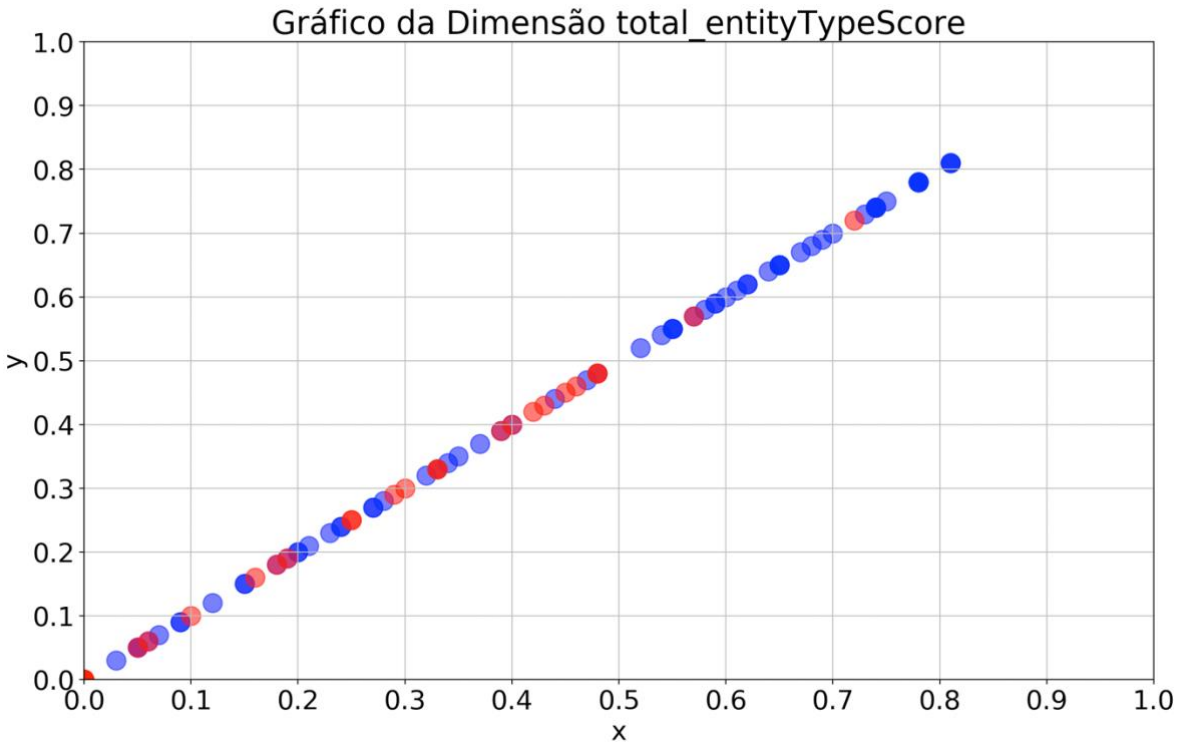
5. CONSIDERAÇÕES FINAIS

5.1. Análise dos textos analisados

Ao fazer uma análise exploratória dos dados à procura de padrões que expliquem a diferença entre os textos das notícias verdadeiras e falsas, nota-se padrões claros.

Conforme apresentado na Figura 6, ao observar a diferença entre o total de palavras chaves criadas pelo algoritmo de *machine learning* da *Microsoft*, notamos que as notícias falsas (em vermelho), possuem menos palavras chaves que as notícias verdadeiras (em azul). Essa análise confirma a Hipótese H3, onde imaginava-se que notícias verdadeiras seriam responsáveis por gerar mais palavras chaves que serão utilizadas pelos motores de busca da *Google* e *Microsoft*, por exemplo, e serão aplicadas pela *Wikipedia* como *tags* dentro da sua ferramenta de pesquisa de documentos e artigos.

Figura 6: Análise dos dados da Dimensão total_entityTypeScore

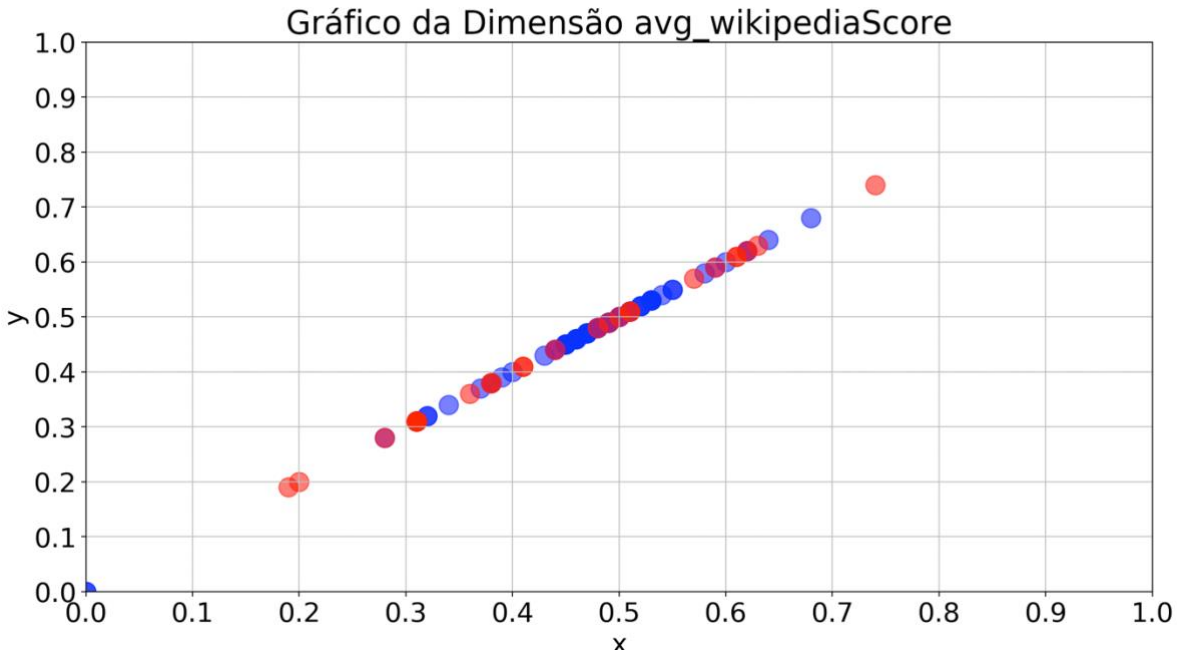


Fonte: Extraído através do programa graphs.py

O eixo x e y do gráfico apresentado, representam a quantidade de palavras chaves, geradas pelos textos analisados, divididos por cem. Essa representação foi feita para permitir um melhor agrupamento dos pontos e assim, expor graficamente, a distribuição de palavras chaves entre as notícias verdadeiras e falsas.

O motor de busca da Microsoft não encontrou diferença notória entre a média de palavras chaves geradas por textos falsos (em vermelho) ou verdadeiros (em azul), que são utilizadas dentro da *Wikipedia* para tag de seus artigos, conforme gráfico apresentado na Figura 7.

Figura 7: Análise dos dados da Dimensão avg_wikipediaScore

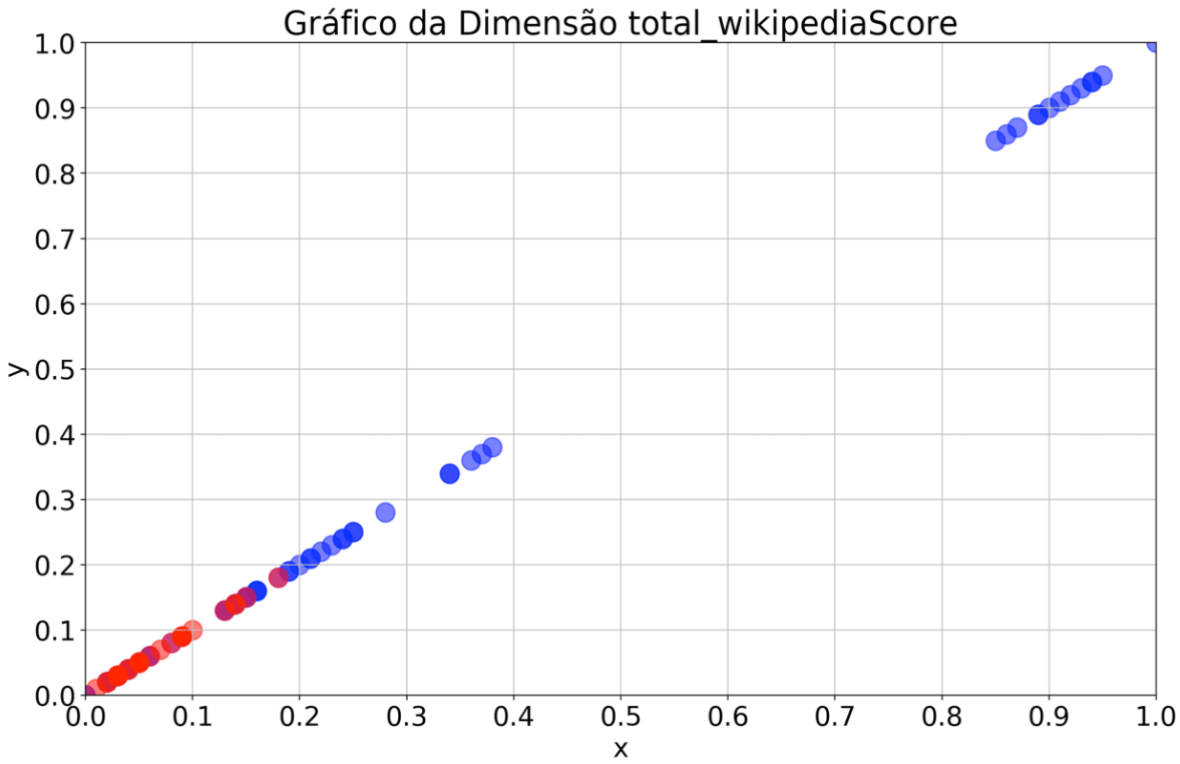


Fonte: Extraído através do programa graphs.py

A média de palavras chaves geradas por texto analisado, foi extraída através da divisão de palavras chaves geradas pela quantidade total de frases do texto analisado.

É importante notar que, embora notícias falsas (pontos em vermelho) e verdadeiras (pontos em azul) gerem médias parecidas de palavras chaves, a relevância delas dentro da *wikipedia* são diferentes. Isso significa que as notícias verdadeiras possuem termos que já foram mencionadas diversas vezes em outras páginas, aumentando a relevância do texto em motores de busca (vide Figura 8).

Figura 8: Análise dos dados da Dimensão total_wikipediaScore



Fonte: Extraído através do programa graphs.py

Análise essa que confirma a Hipótese H4, onde supunha-se que textos verdadeiros teriam mais referências em artigos da *Wikipedia*. Novamente para gerar um agrupamento mais visual, dividiu-se o total de termos citados na *Wikipedia* por cem, deixando os pontos mais juntos no gráfico. O resultado da divisão foi utilizado, igualmente, para popular os eixos x e y.

Curiosamente, uma das hipóteses levantada nesse trabalho, é que as notícias falsas teriam uma menor identificação com a língua portuguesa, ao serem analisadas pela *machine learning* da Microsoft, isto é, textos com erros ortográficos ou de sintaxe, ou ainda textos que utilizassem muitas palavras de outras línguas, teriam uma identificação menor com o português.

Para se ter uma melhor compreensão, deve-se analisar o gráfico da Figura 9, quanto mais próximo do número 1 no eixo x e y, mais a *machine learning* da *Azure* identificou o texto com português.

Figura 9: Análise dos dados da Dimensão language_score

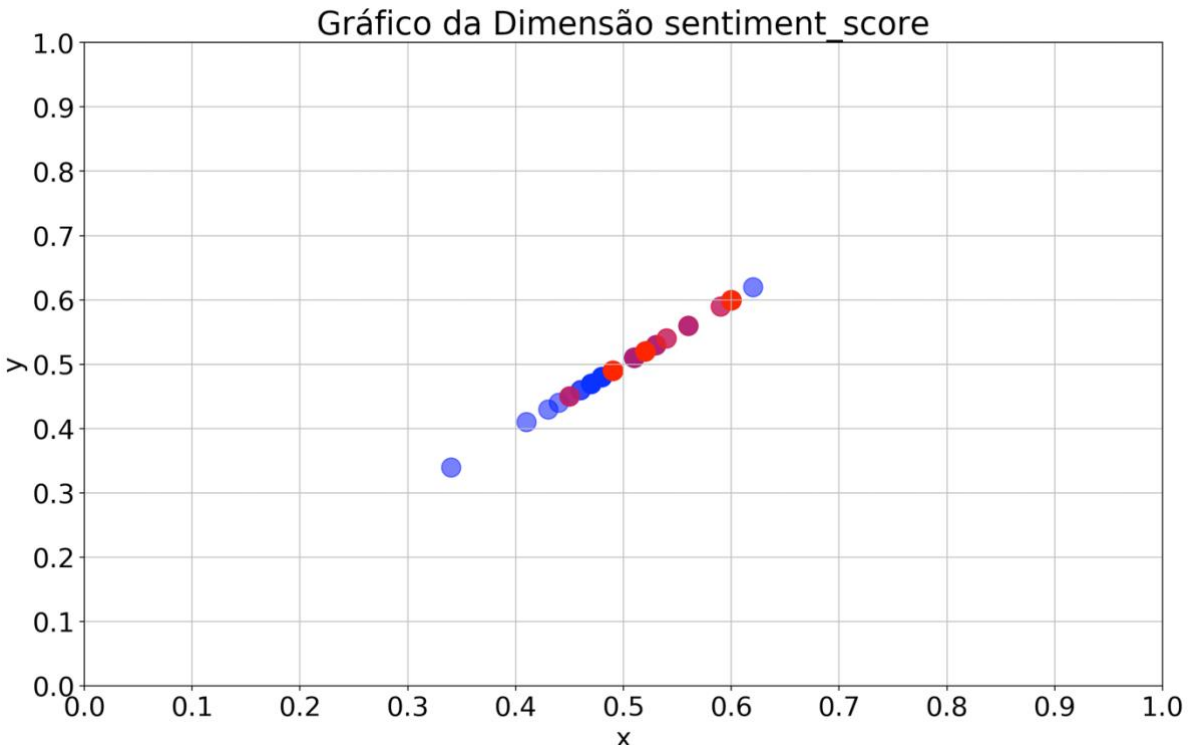
Fonte: Extraído através do programa graphs.py

A Hipótese H1 não foi comprovada, os textos falsos (pontos em vermelho) receberam uma classificação de identificação com a língua portuguesa maior que os textos verdadeiros. Fato esse que remete a uma nova hipótese, é possível que as informações falsas sejam mais diretas e curtas, consequentemente sem utilizarem siglas ou termos em outras línguas, aumentando assim sua pontuação, na *machine learning* da Azure e recebendo uma classificação de similaridade maior com a língua portuguesa.

A Hipótese H2 é a mais relevante para todo o estudo, já que ao confrontar notícias verdadeiras e falsas, imagina-se que as notícias falsas devem possuir um viés maior, em outras palavras, elas devem estar repletas de “sentimento”. A *machine learning* da *Microsoft* faz uma análise do texto e através de observações próprias, informa se o texto transmitiu uma informação positiva, neutra ou negativa sobre o dado analisado. Imagina-se que jornais tendem a passar uma informação mais neutra da notícia retratada, sem fazer juízo de valor sobre o acontecimento descrito, em contrapartida notícias falsas deveriam apresentar uma neutralidade menor e estar nos eixos opostos, ou serem muito negativas ou positivas.

O gráfico da Figura 10 contradiz a hipótese levantada, notícias mais neutras deveriam estar próximas a 0.5, já as notícias abaixo desse valor seriam classificadas como negativas e os textos acima de 0.5 com uma positividade maior.

Figura 10: Análise dos dados da Dimensão sentiment_score



Fonte: Extraído através do programa graphs.py

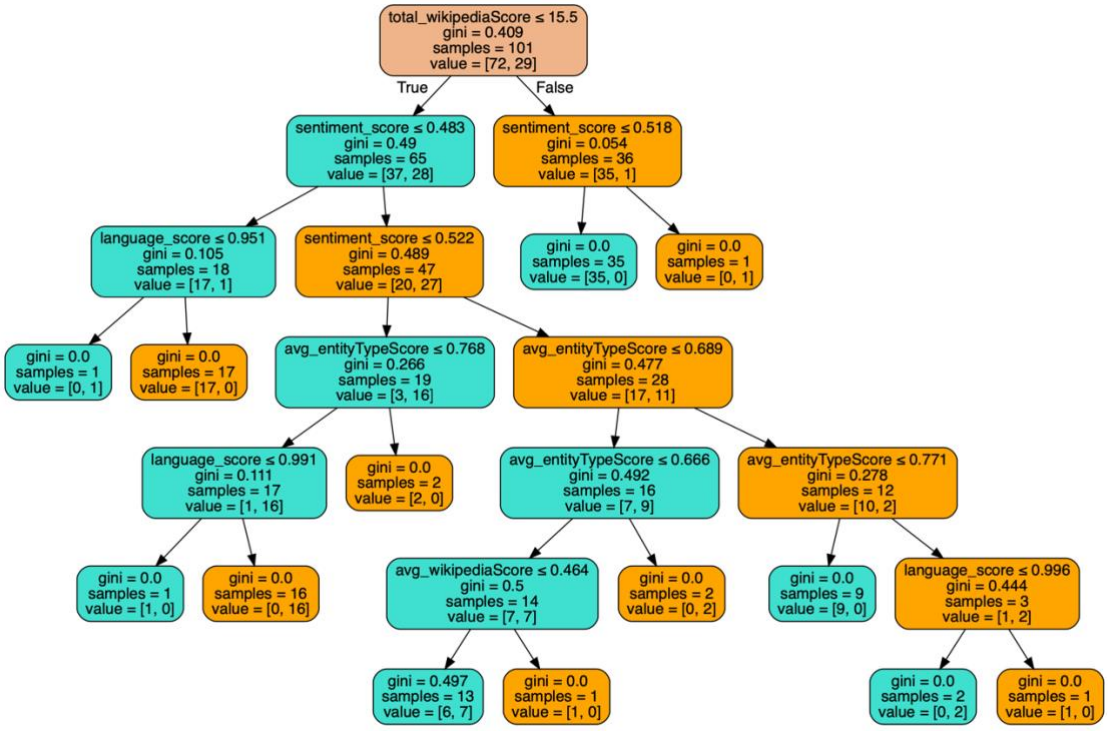
Novamente os pontos em vermelho representam as notícias falsas e os pontos em azuis as notícias verdadeiras. Nota-se como as notícias verdadeiras, treinadas pelo algoritmo, foram classificadas como sendo mais negativas que as notícias falsas.

Além dos fatos jornalísticos, que foram apresentados à *machine learning* desse trabalho, não serem mais neutras que as notícias falsas, o modelo gerado realçou o viés mais positivo das *fake news* ao descreverem o texto que elas se propuseram a apresentar aos seus leitores.

5.2. Árvore do modelo criado

Ao utilizarmos bibliotecas da própria linguagem de programação Python, é possível gerar a árvore criada pelo modelo treinado. Conforme ilustrado na Figura 11, percebe-se que algoritmo determinou que a maior diferença entre as notícias verdadeiras e falsas, é a relevância dos termos usados nos textos dentro da *Wikipedia.* A partir desse ponto, o algoritmo faz a classificação dos textos determinando as diferenças e similaridades entre cada um deles.

Figura 11: Árvore do Modelo Criado



Fonte: Extraído através do programa graphmodel.py

5.3. Conclusão

O algoritmo de *machine learning* apresentado nesse trabalho analisou 101 *URLs*, das quais 70% foram usadas para treinar o algoritmo e 30% foram utilizadas para verificar a taxa de acerto dele. Baseado nas notícias que foram enviadas para o algoritmo classificar, a taxa de acerto foi de 80%. Portanto, considera-se que o algoritmo funcionou adequadamente e conseguiu se mostrar útil como ferramenta de apoio para identificação de notícias verdadeiras e falsas que estão sendo veiculadas por sites da internet.

Por haver dificuldades em achar notícias comprovadamente falsas, com o intuito de treinar o algoritmo, o modelo criado foi populado com publicidades sensacionalistas, conforme descrito no Anexo II. Essa dificuldade para encontrar textos que, supostamente são sérios e ao mesmo tempo são falsos, pode ter causado um viés no algoritmo, o que vai na contramão da proposta desse trabalho.

Ainda que o algoritmo tenha se mostrado útil, é importante ressaltar que um código de programa computacional só classificará, o que de certa forma, foi inserido via parâmetro por um ser humano falível.

O algoritmo pode ser alterado, para mitigar o resíduo de viés na escolha do dado, ainda assim haverá um risco de erro não intencional na seleção do modelo utilizado para o treinamento do algoritmo, ou ainda o número de textos analisados, podem ser suficientes para um resultado mais preciso.

Portanto, mais importante que a classificação de um texto em falso ou verdadeiro, seja o encorajamento ao debate, através de apresentação de textos conflitantes com o original pesquisado, ou ainda a apresentação de outras fontes, levando o leitor a uma reflexão mais rica e profunda.

5.4. Trabalhos relacionados

Não foi encontrado na literatura muitos trabalhos relacionados a proposta do algoritmo. O que se assemelha mais ao algoritmo proposto nessa monografia foi criado por Sakeena M. Sirajudeen, Nur Fatihah A. Azmi e Adamu I. Abubakar (SIRAJUDEEN, FATIHAH e ABUBAKAR, 2017). Entretanto, os autores não

desenvolveram um algoritmo de *machine learning*, mas tentaram algumas abordagens diferentes como analisar a fonte, quem escreveu o texto, e a relevância dela através de outras notícias já veiculadas.

5.5. Trabalhos Futuros

Como trabalhos futuros, é proposto a modificação do algoritmo para, além de classificar textos, fazer um agrupamento das palavras, através de *machine learning* não supervisionado, para a identificação de possíveis padrões e análise se textos confiáveis e não confiáveis, se possuem um conjunto específico de palavras, ou semelhança na forma de escrita.

Além disso, pode-se procurar na literatura da área de psicologia, formas de identificação reconhecimento cognitivo de textos falsos e verdadeiros, incrementando as dimensões do algoritmo e o tornando mais preciso.

GLOSSÁRIO

BABY BOOMERS: Geração dos nascidos após Segunda Guerra Mundial até a metade da década de 1960. A designação vem da expressão "baby boom", que representa a explosão na taxa de natalidade nos Estados Unidos no pós-guerra.

SMB: Server Message Block, em português, “Bloco de Mensagem de Servidor”, é um protocolo utilizado para acesso compartilhado a arquivos e impressoras.

IFCN: International Factchecking Network – Rede internacional de checa dores de fatos.

Agenda-Setting: Teoria que explica quais os *trending topics* estão sendo veiculados e como eles afetam a opinião da pública.

Trending Topics: Tópicos de tendência pode ser visto como os assuntos do momento, aqueles que estão sendo mais divulgados nas mídias sociais.

WEB: Nome pelo qual a rede mundial de computadores internet se tornou conhecida a partir de 1991, quando se popularizou devido à criação de uma *interface gráfica* que facilitou o acesso e estendeu seu alcance ao público em geral.

AdSense: É o serviço de publicidade oferecido pelo Google inc. Os donos de websites podem inscrever-se no programa para exibir anúncios em texto, imagem.

Strings: Em computação, string é considerado uma cadeia de caracteres.

Machine Learning: Método de análise de dados que automatiza a construção de modelos analíticos. É um ramo da inteligência artificial baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana.

Core Business: Atividade central, aquilo que mais gera valor para uma empresa.

tag: Palavras chaves utilizadas para classificar e organizar arquivos.

html: Anacrônico de *HyperText Markup Language*, expressão inglesa que significa "Linguagem de Marcação de Hipertexto". Consiste em uma linguagem de marcação utilizada para produção de páginas na web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela internet.

url: O Uniform Resource Locator, é um termo técnico que foi traduzido para a língua portuguesa como "localizador uniforme de recursos". Um URL se refere ao endereço de rede no qual se encontra algum recurso informático, como por exemplo um arquivo de computador ou um dispositivo periférico.

nosql: Não SQL ou “não relacional”, posteriormente estendido para “Not Only SQL” (não somente SQL) é um termo genérico que representa os bancos de dados não relacionais.

SQL:	Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional.
json:	JavaScript Object Notation (Notação de Objetos JavaScript) é uma formatação leve de troca de dados que para seres humanos, é fácil de ler e escrever e para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação JavaScript.
Ransomware:	Ransomware é um tipo de software nocivo que restringe o acesso ao sistema infectado com uma espécie de bloqueio e cobra um resgate em criptomoedas para que o acesso possa ser restabelecido. Caso não ocorra o mesmo, arquivos podem ser perdidos e até mesmo publicados.
Website:	É um conjunto de páginas web, isto é, de hipertextos acessíveis geralmente pelo protoloco HTTP ou HTTPS na internet.
whatsapp:	Aplicativo multiplataforma de mensagens instantâneas e chamadas de voz para smartphones.
facebook:	Mídia social e rede social virtual utilizada por várias pessoas ao redor do mundo.
twitter:	É uma rede social e um servidor para microblogging, que permite aos usuários enviar e receber atualizações pessoais de outros contatos, por meio do website do serviço, por SMS e por softwares específicos de gerenciamento

Wikipedia:	A Wikipédia é um projeto de enciclopédia multilíngue de licença livre, baseado na web e escrito de maneira colaborativa
Blog:	Site eletrônico cuja estrutura permite a atualização rápida a partir de acréscimos dos chamados artigos, ou postagens e publicações.

REFERÊNCIAS

ABDULLAH, H. Medium. **Machine learning:** A strategy to learn and understand (Chapter 3) Part 3: Unsupervised Learning, 2018. Disponivel em: <https://medium.com/the-21st-century/machine-learning-a-strategy-to-learn-and-understand-chapter-3-9daaad4afc55>. Acesso em: 20 jan. 2019.

ALLCOTT, H.; GENTZKOW, M. Social Media and Fake News in the 2016 Election. **Journal of Economic Perspectives**, 2017. 211-236.

BATHKE, B. Como a publicidade incentiva "fake news". **Carta Capital**, maio 2017.

BRISOLA, A. C.; ROMEIRO, N. L. A Competência Crítica em Informação como Resistência: uma análise sobre o uso da informação na atualidade. **Revista Brasileira de Biblioteconomia e Informação**, maio 2018.

BROWNLEE, J. Diferença entre classificação e regressão na aprendizagem de máquinas. **iCrowd Newswire**, dezembro 2017.

BURKHARDT, J. M. **Combating Fake News in the Digital Age**. [S.l.]: Library Technology Reports, v. 1, 2017. 5-9 p.

CAMPOS, R. Árvores de Decisão. **Machine Learning Beyond Deep Learning**, 2017. Disponivel em: <https://medium.com/machine-learning-beyond-deep-learning/árvores-de-decisão-3f52f6420b69>. Acesso em: 4 dezembro 2018.

COELHO BEZERRA, A.; CAPURRO, R.; SCHNEIDER, M. Regimes de verdade e poder: dos tempos modernos à era digital. **Liinc em Revista**, novembro 2017. 371-380.

COELHO LOBO DE CARVALHO, G. A.; KANFFER, G. G.. **O Tratamento Jurídico das Notícias Falsas (fake news)**. [S.l.]: [s.n.]. 2018. p. 19.

DANIELA, S. Criação, implantação e avaliação de um programa de competência em informação em alunos do ensino fundamental. **RBBD. Revista Brasileira de Biblioteconomia e Documentação**, 2017. 885-906.

DOTA, M. A. Modelo para a classificação da qualidade da água contaminada por solo usando indução por árvore de decisão. **Tese (Doutorado em Sistemas Digitais) - Escola Politécnica, Universidade de São Paulo**, São Paulo, 2014.

FEA, J. The press was way more political in Jefferson's day - but he defended it anyway. **Penny Live**, 2017. Disponivel em:

<https://www.pennlive.com/opinion/2017/03/the_press_was_more_political_i.html>. Acesso em: 01 nov. 2019.

FERREIRA MENDES DE SOUZA, G. et al. Propagação de Mensagens na Internet: Teoria do Comportamento Planejado, junho 2012.

FILHO, O. F. O Que é Falso Sobre Fake News. **Revista USP**, março 2018. 39-44.

GIBBONS, K. Over-55s at risk from online fake news. **The Times**, Londres, p. 35, fevereiro 2018.

HABER, M. The Real Risks of Fake News. **Risk Management**, 2017. Disponivel em: <http://www.rmmagazine.com/2017/04/03/the-real-risks-of-fake-news/>. Acesso em: 20 marco 2018.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statical Learning:** Data Mining, Inference, and Prediction. New York: Springer, v. Segundo Volume, 2008.

HOLMES, M. The Multiple Dimensions of Information Quality. **Information Systems Management**, outubro 2007. 79-82.

HONDA, H.; FACURE, M.; YAOHAO, P. Os Três Tipos de Aprendizado de Máquina. **lamfo-unb.github.io**, 2017. Disponivel em: <https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/>. Acesso em: 26 maio 2019.

JANG, S. M.; KIM, J. K. Third person effects of fake news: Fake news regulation and media literacy interventions, Columbia, 23 agosto 2017. 295-302.

JANKOWSKI, N. W. Researching Fake News: A Selective Examination of Empirical Studie. **Javnost - The Public**, fevereiro 2018. 248-255.

KOOHIKAMALI, M.; SIDOROVA, A. Information Re-Sharing on Social Network Sites in the Age of Fake News. **Informing Science: the International Journal of an Emerging Transdicipline**, maio 2017.

KULKARNI, M. Decision Trees for Classification: A Machine Learning Algorithm. **THE XORIANT BLOG**, 2017. Disponivel em: <https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html>. Acesso em: 15 novembro 2018. il. col.

LIM, C. Checking How Fact-checkers Check. **Stanford University**, maio 2017.

MAHESHWARI, S. How fake news goes viral: a case study. **The New York Times**, novembro 2016.

MANJOO, F. True Enough: Learning to live in a post-fat society. **John Wiley & Sons**, New Jersey, 2008.

MICROSOFT DOCS. Cognitive Services And Machine Learning. **https:** //docs.microsoft.com, 2019. Disponivel em: <https://docs.microsoft.com/en-us/azure/cognitive-services/cognitive-services-and-machine-learning>. Acesso em: 5 dez. 2019.

MILLER, H. The multiple dimensions of information quality. **Systems Management**, outubro 2007. 79-82.

RAMONET, I. A explosão do jornalismo: das mídias de massa à massa de mídia, São Paulo, 2012.

SADAM, R. Facebook Launches 'Journalism Project' to Improve Media Ties. **Time**, novembro 2017.

SANTOS, H. G. D. Comparação de Performance de Algoritmos de Machine Learning para Análise Preditiva em Saúde Pública e Medicina, agosto 2018.

SAS INSIGHTS. Machine Learning - O que é e qual sua importância? **SAS The Power to Know**, 2019. Disponivel em: <https://www.sas.com/pt_br/insights/analytics/machine-learning.html>. Acesso em: 01 jun. 2019.

SIRAJUDEEN, S. M.; FATIHAH, N.; ABUBAKAR, A. Online fake news detection algorithm. **Department of Computer Science International Islamic University Malaysia**, Kuala Lumpur, 2017.

SPINELLI, E. M.; SANTOS, J. D. A. Jornalismo na Era da Pós-Verdade: fact-checking como ferramenta de combate às fake news. **Revista Observatório**, maio 2018.

THAPLIYAL, M. Machine Learning Basics: Supervised Learning Theory part-1. **Medium**, 2018. Disponivel em: <https://medium.com/mlrecipies/machine-learning-basics-supervised-learning-theory-part-1-d910b96d56fc>. Acesso em: 20 jan. 2019.

VARGO, C.; GUO, L.; AMAZEEN, M. The agenda-setting power of fake news: A big data analysis of the online media landscape from 2014 to 2016, junho 2017.

APÊNDICE A – Criação do banco de dados em mongodb utilizado no algoritmo

Em mongodb não existe um comando como create database ou create collection, então pode-se utilizar uma GUI para fazer essas criações, ou ainda deixar que o próprio algoritmo faça essa criação.

APÊNDICE B – Arquivo DatabasePool.py – utilizado para estabelecer a comunicação com a base de dados

```
import mysql.connector
from pymongo import MongoClient

class DatabasePool():
    """Class created to optmize connection with database"""

    def __init__(self, host_name="localhost", database_name="fakenews", user_name="root",
password="<ALTERAR PARA SUA SENHA>"):
        self.host_name = host_name
        self.database_name = database_name
        self.user_name = user_name
        self.password = password

    def open_connection(self):
        return mysql.connector.connect(user=self.user_name, password=self.password,
                                       host=self.host_name, database=self.database_name)

    def connect_mongo(self):
        return MongoClient(self.host_name, 27017)
```

APÊNDICE C – Arquivo myHtmlClass.py – utilizado para fazer o parse do texto da url analisada

```
from urllib.request import Request, urlopen
from bs4 import BeautifulSoup
import re


class myHtmlClass:
    """This class is used to parse a html file and clean it converting to text."""

    def html_cleaner(self, url_to_be_analized):

        header = {
            'Content-Type': 'application/html; charset=UTF-8',
            'User-Agent': 'Mozilla/5.0'
        }

        try:
            html = Request(url_to_be_analized, headers=header)
            web_byte = urlopen(html).read()
            soup = BeautifulSoup(web_byte, "html.parser")

            # print(soup.body)

        except Exception as e:
            print(f"Error ----> {e}")
            exit(1)

        for script in soup.html(["script", "style"]):
            script.extract()  # rip it out

        text = soup.get_text()
        # print(f"{text}")

        # break into lines and remove leading and trailing space on each
        lines = (line.strip() for line in text.splitlines())

        # break multi-headlines into a line each
        chunks = (phrase.strip() for line in lines for phrase in line.split("    "))

        # drop blank lines
        text = '\n'.join(chunk for chunk in chunks if chunk)

        # limit text lenght
        # text = text[:5120]

        return text

    def prepare_text(self, text):

        h = myHtmlClass()
        hc = h.html_cleaner(text)

        pattern1 = '^[\w|\W]+\B[A-Z]+'
        pattern2 = '[a-z0-9á][A-Z]+[\w]+'

        final_list = []
```

```
for i in hc.split('\n'):
    a = i.split()
    if len(a) > 35:
        l = []
        for z in i.split(' '):
            if z != '':
                z = z.replace("!", "")
                l.append(z)
                a = re.findall(pattern1, z)
                # print(f"a--->{a} z--->{z}")
                if (len(a) > 0) \
                        and (str(a[0]).replace("(", "").replace(")", "").replace("+", "") != z.replace("(",
                                                                                                       "").replace(
                    ")", "").replace("+", "")) \
                        and (z != "HQs"):
                    w = re.findall(pattern1, z)
                    y = re.findall(pattern2, z)

                    if len(y) == 0:
                        continue

                    w = w[0][:-1]
                    w = str(w)
                    # print(f"w--->{w}")
                    # print(f"y--->{y}")
                    y = y[0][1:]

                    l.remove(z)
                    l.append(w + str('.'))
                    l.append(y)
        for z in l:
            final_list.append(z)

final_string = ' '.join(final_list)
return final_string
```

APÊNDICE D – Arquivo fakenews.py – utilizado para criar a matriz (dimensões) que serão analisadas arquivo principal do algoritmo

```
import requests
from sklearn import tree
from DatabasePool import DatabasePool


class fakenews:
    """This class analyze an url and says if the document is fakenews or not"""

    def __init__(self,
                 analitics_base_url="https://brazilsouth.api.cognitive.microsoft.com/text/analytics/v2.1/",
                 api="languages"):

        f = open('/Users/arnaldo.pedroso/git/Monografia/fakenews.conf', "r")
        subscription_key = f.read()

        self.analitics_base_url = analitics_base_url
        self.api = api
        self.subscrition_key = subscription_key
        self.headers = {"Ocp-Apim-Subscription-Key": subscription_key}

    ''' kill all scripts and style elements'''

    def check_language(self, text):

        if self.api != "languages":
            api = "languages"
        else:
            api = self.api

        language_api_url = self.analitics_base_url + api
        documents = {"documents": [{"id": "1", "text": text}]}

        response = requests.post(language_api_url, headers=self.headers, json=documents)
        languages = response.json()

        return languages

    def check_sentiment(self, text):

        if self.api != "sentiment":
            api = "sentiment"
        else:
            api = self.api

        language_api_url = self.analitics_base_url + api

        documents = {"documents": [{"id": "1", "language": "pt", "text": text}]}

        response = requests.post(language_api_url, headers=self.headers, json=documents)
        sentiment = response.json()

        return sentiment

    def check_key_words(self, text):
```

```
        if self.api != "entities":
            api = "entities"
        else:
            api = self.api

        language_api_url = self.analitics_base_url + api

        documents = {"documents": [{"id": "1", "text": text}]}

        response = requests.post(language_api_url, headers=self.headers, json=documents)
        key_words = response.json()

        return key_words

    def load_tree(self):
        list_dimension = []
        list_classification = []

        d = DatabasePool()
        c = d.connect_mongo()

        database = c['fakenews']
        url_collection = database['url']

        dict_colletion = url_collection.find({})
        for i in dict_colletion:
            ld = [i["avg_entityTypeScore"], i["avg_wikipediaScore"], i["language_score"],
i["sentiment_score"],
                  i["total_entityTypeScore"], i["total_wikipediaScore"]]
            list_dimension.append(ld[:])
            list_classification.append(i["is_fake"])

        return list_dimension, list_classification

    def classify_url(self, dimension):
        ld, lc = self.load_tree()

        clf = tree.DecisionTreeClassifier()
        clf = clf.fit(ld, lc)

        return clf.predict([dimension])
```

APÊNDICE E – Arquivo algorithmFakeNews.py – arquivo principal com toda inteligência da machine learning criada

```
from DatabasePool import DatabasePool
from fakenews import fakenews
from myHtmlClass import myHtmlClass
from bson.objectid import ObjectId
import json

def get_dimension(paramenter_url):
    try:
        f = fakenews()
        m = myHtmlClass()

        dimension_text = m.prepare_text(paramenter_url)
        dimension_text = dimension_text[:5120]
        # print(f"--->{dimension_text}<---")
        dimension_language = f.check_language(dimension_text)
        dimension_sentiment = f.check_sentiment(dimension_text)
        dimension_key_words = f.check_key_words(dimension_text)
        return dimension_language, dimension_sentiment, dimension_key_words
    except:
        return 0


def open_database():
    try:
        d = DatabasePool()
        c = d.connect_mongo()
    except:
        return 0
    return c


def language_score(url_id):
    global new_list
    x = data_collection.find({"url_id": url_id})
    nl = []
    for i in x:
        ll = i["language"]["documents"]
        new_list = [x["detectedLanguages"] for x in ll]
    nl.append([x["score"] for x in new_list[0]])
    return nl[0][0]


def sentiment_score(url_id):
    x = data_collection.find({"url_id": url_id})
    nl = []
    for i in x:
        ll = i["sentiment"]["documents"]
        nl = [x["score"] for x in ll]
    return nl[0]


def key_word_score(url_id):
    #print("url_id ---->", url_id)
    x = data_collection.find({"url_id": url_id})
    entityTypeScore = []
```

```
    wikipediaScore = []

    for i in x:
        ll = i["key_word"]["documents"]
        new_list = [x["entities"] for x in ll]

    lkeyword = new_list[0]
    for i in range(len(lkeyword)):
        l = lkeyword[i]
        lm = dict(l["matches"][0])

        try:
            wikipediaScore.append(lm["wikipediaScore"])
        except:
            entityTypeScore.append(lm["entityTypeScore"])

    return len(entityTypeScore), entityTypeScore, len(wikipediaScore), wikipediaScore


def log_classification(p_is_fake):
    if p_is_fake == 0:
        print("Document is true")
        print("***"*30)
        print("")
        print("")
    else:
        print("FAKE NEWS!!!")
        print("***"*30)
        print("")
        print("")


# Set fakenews objectx
f = fakenews()

# Set database
d = open_database()
database = d['fakenews']

#Documents to be analyzed
# if you need to training a text, inform is_fake parameter with the knowledge value
# if you need to classify the text, pass is_fake parameter with blank value
# 0 = not fake
# 1 = is fake

#documents = [
#    {"url": "https://noticia-tv.com/entrevista-helen-sbt/?utm_source=taboola&utm_medium=PHYTO-2B-
DESK-CP1&utm_campaign=editoraabril-exame", "is_fake":"1"},
#]


json_file = open("/Users/arnaldo.pedroso/git/Monografia/news.json", "r", encoding="utf-8")

news = json.load(json_file)

for doc in news:
    url = doc["url"]
    print("==*"*30)
    print(f"Working with url {url}")
    print("==*"*30)
```

```
    v_is_fake = doc["is_fake"]

    # Define dictionary to be searched
    dict_url = {"url": url}

    # Set collections
    url_collection = database['url']
    data_collection = database['data_to_be_analyzed']

    # Reset variables
    vlanguage_score = 0.0
    vsentiment_score = 0.0

    d_url = url_collection.find(dict_url)
    if v_is_fake is not "":
        if url_collection.count_documents(dict_url) >0:
            try:
                return_is_fake = [x["is_fake"] for x in d_url]
            except:
                print("Error - ignoring this url")
                url_collection.delete_one({"url": url})
                continue

            log_classification(int(return_is_fake[0]))
            continue
        else:
            # Insert url
            url_id = url_collection.insert_one(dict_url).inserted_id

            # Get dimensions

            try:
                language, sentiment, key_word = get_dimension(url)
            except:
                url_collection.delete_one({"url": url})
                continue


            # Prepare data to be recorded
            dict_data = {"url_id": url_id, "language": language, "sentiment": sentiment, "key_word":
key_word}

            # Inserting data_collection
            data_to_be_analyzed_id = data_collection.insert_one(dict_data).inserted_id

            # Determine score values
            vlanguage_score = float(language_score(url_id))
            vsentiment_score = float(sentiment_score(url_id))
            total_entityTypeScore, entityTypeScore, total_wikipediaScore, wikipediaScore =
key_word_score(url_id)

            sum_entityTypeScore = float(sum(entityTypeScore))

            try:
                avg_entityTypeScore = float(sum_entityTypeScore / total_entityTypeScore)
            except ZeroDivisionError:
                avg_entityTypeScore = 0

            sum_wikipediaScore = float(sum(wikipediaScore))
```

```
            try:
                avg_wikipediaScore = float(sum_wikipediaScore / total_wikipediaScore)
            except ZeroDivisionError:
                avg_wikipediaScore = 0

            url_collection.update_one({"url": url}, {"$set":
                                                {"language_score": vlanguage_score,
                                                 "sentiment_score": vsentiment_score,
                                                 "total_entityTypeScore": total_entityTypeScore,
                                                 "total_wikipediaScore": total_wikipediaScore,
                                                 "avg_entityTypeScore": avg_entityTypeScore,
                                                 "avg_wikipediaScore": avg_wikipediaScore,
                                                 "is_fake": v_is_fake
                                                 }
                                            })

            print(f"URL: {url} has been inserted.")
            print(f"URL id# is {url_id}.")
            print(f"Data id# is {data_to_be_analyzed_id}.")
            print("--"*30)
        elif url_collection.count_documents(dict_url) > 0:
            try:
                return_is_fake = [x["is_fake"] for x in d_url]
            except:
                print("Error - ignoring this url")
                url_collection.delete_one({"url": url})
                continue
            log_classification(int(return_is_fake[0]))
            continue
        else:
            url_swap_collection = database['url_swap']

            #Insert url_swap
            url_id = url_swap_collection.insert_one(dict_url).inserted_id

            #Get dimensions
            language, sentiment, key_word = get_dimension(url)

            #Prepare data to be recorded
            dict_data = {"url_id": url_id, "language": language, "sentiment": sentiment, "key_word": key_word}

            #Insert data_collection_swap
            data_to_be_analyzed_id = data_collection.insert_one(dict_data).inserted_id

            #Determine score values
            vlanguage_score = float(language_score(url_id))

            vsentiment_score = float(sentiment_score(url_id))
            total_entityTypeScore, entityTypeScore, total_wikipediaScore, wikipediaScore =
    key_word_score(url_id)

            sum_entityTypeScore = float(sum(entityTypeScore))
            avg_entityTypeScore = float(sum_entityTypeScore / total_entityTypeScore)

            sum_wikipediaScore = float(sum(wikipediaScore))
            avg_wikipediaScore = float(sum_wikipediaScore / total_wikipediaScore)

            #Verify if the url is fake or not
```

```
    is_fake = f.classify_url([avg_entityTypeScore, avg_wikipediaScore, vlanguage_score,
vsentiment_score, sum_entityTypeScore, sum_wikipediaScore])

    #Print classification of the url
    log_classification(int(is_fake[0]))

    #Remove swap registers
    url_swap_collection.delete_one({'_id': ObjectId(url_id)})
    data_collection.delete_one({'_id': ObjectId(data_to_be_analyzed_id)})

    #Inserting correct values
    url_id = url_collection.insert_one(dict_url).inserted_id
    url_collection.update_one({"url": url}, {"$set":
                                    {"language_score": vlanguage_score,
                                     "sentiment_score": vsentiment_score,
                                     "total_entityTypeScore": total_entityTypeScore,
                                     "total_wikipediaScore": total_wikipediaScore,
                                     "avg_entityTypeScore": avg_entityTypeScore,
                                     "avg_wikipediaScore": avg_wikipediaScore,
                                     "is_fake": is_fake[0]
                                    }
                                })

    # Inserting data collection
    dict_data = {"url_id": url_id, "language": language, "sentiment": sentiment, "key_word": key_word}
    data_to_be_analyzed_id = data_collection.insert_one(dict_data).inserted_id
exit(1)
```

APÊNDICE F – Arquivo graphs.py – programa utilizado para gerar os gráficos para análise dos dados de cada dimensão

```
import matplotlib.pyplot as plt
import DatabasePool as db

ldimension = ["avg_entityTypeScore", "avg_wikipediaScore", "language_score", "sentiment_score",
"total_entityTypeScore", "total_wikipediaScore"]
ldimension_fake = []
ldimension_notfake = []

def read_dimension_data(dimension_name):
    dimension1 = []
    dimension2 = []

    ### Creating database connection
    dtb = db.DatabasePool()
    c = dtb.connect_mongo()
    database = c['fakenews']
    url_collection = database['url']

    for vis_fake in range(0, 2):
        c_url = url_collection.find({"is_fake": str(vis_fake)})
        if vis_fake == 0:
            for i in c_url:
                n = round(float(i[dimension_name]), 2)
                if n >= 1:
                    n = n/100
                dimension1.append(n)
        else:
            for i in c_url:
                n = round(float(i[dimension_name]), 2)
                if n >= 1:
                    n = n/100
                dimension2.append(n)

    ## closing database connection
    c.close()

    ## Analyzing data
    print('Dimension --->', dimension_name)
    print('Is not Fake --->', dimension1)
    print('Is fake --->', dimension2)

    return dimension1, dimension2


def format_graph(datagraph1, datagraph2, dimension):
    strdimension = 'Gráfico da Dimensão ' + dimension
    plt.rcParams.update({'font.size': 20})
    plt.figure(figsize=(8, 5))
    plt.scatter(datagraph1, datagraph1, color='b', s=200, alpha=0.60)
    plt.scatter(datagraph2, datagraph2, color='r', s=200, alpha=0.60)
    plt.xlim(0, 1)
    plt.ylim(0, 1)
    plt.xticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
    plt.yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
    plt.xlabel('x')
    plt.ylabel('y')
```

```
    plt.title(strdimension)
    plt.grid(True)
    plt.show()


# Dimensions
# "avg_entityTypeScore"
# "avg_wikipediaScore"
# "language_score"
# "sentiment_score"
# "total_entityTypeScore"
# "total_wikipediaScore"

for d in ldimension:
    ldimension_fake, ldimension_notfake = read_dimension_data(d)
    format_graph(ldimension_fake, ldimension_notfake, d)
    del ldimension_fake[:]
    del ldimension_notfake[:]
```

APÊNDICE G – Arquivo graphmodel.py – programa utilizado para gerar a figura com o modelo de dados criado pelo algoritmo

```
import DatabasePool as db
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from IPython.display import Image
import pydotplus
from io import StringIO
import collections

list_dimension = []
list_classification = []

### Creating database connection
dtb = db.DatabasePool()
d = dtb.connect_mongo()
database = d['fakenews']
url_collection = database['url']

# Load data
dict_colletion = url_collection.find({})
for i in dict_colletion:
    ld = [i["avg_entityTypeScore"], i["avg_wikipediaScore"], i["language_score"], i["sentiment_score"],
          i["total_entityTypeScore"], i["total_wikipediaScore"]]
    list_dimension.append(ld[:])
    list_classification.append(i["is_fake"])

data_feature_names = [ 'avg_entityTypeScore', 'avg_wikipediaScore', 'language_score',
'sentiment_score', 'total_entityTypeScore', 'total_wikipediaScore' ]

dtree=DecisionTreeClassifier()
dtree.fit(list_dimension, list_classification)

dot_data = StringIO()
export_graphviz(dtree,
                feature_names=data_feature_names,
                out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True)

graph = pydotplus.graph_from_dot_data(dot_data.getvalue())

colors = ('turquoise', 'orange')
edges = collections.defaultdict(list)

for edge in graph.get_edge_list():
    edges[edge.get_source()].append(int(edge.get_destination()))

for edge in edges:
    edges[edge].sort()
    for i in range(2):
        dest = graph.get_node(str(edges[edge][i]))[0]
        dest.set_fillcolor(colors[i])
Image(graph.create_png())

# Create PDF
graph.write_pdf("/Users/arnaldo.pedroso/git/Monografia/fakenews_model.pdf")
# Create PNG
graph.write_png("/Users/arnaldo.pedroso/git/Monografia/fakenews_model.png")
```

APÊNDICE H – urls da Microsoft *Azure* para geração das dimensões utilizadas pela *machine learning*

É utilizado a API de análise de textos do Microsoft Azure:

https://brazilsouth.api.cognitive.microsoft.com/text/analytics/v2.1/

Onde é enviado por parâmetro a chave de acesso para a utilização da *API.* A chave de acesso é provida pela Microsoft *Azure* após a criação de conta de usuário na plataforma descrita, e subscrição no serviço de análise cognitiva de texto.

No arquivo fakenews.py, na linha doze o código abaixo citado, representa o arquivo de configuração onde será passado a chave de acesso ao recurso descrito:

```
f = open('/Users/arnaldo.pedroso/git/Monografia/fakenews.conf', "r")
```

No exemplo acima citado, o arquivo fakenews.conf possui o valor da chave que foi gerado pela Microsoft *Azure*. Por se tratar de uma chave pessoal e intransferível, a chave utilizada para os testes desse trabalho, não foi listada em capítulo algum, e não será encontrado no repositório de código descrito no Apêndice I.

APÊNDICE I – versionamento dos arquivos utilizados

Todo o código utilizado para o desenvolvimento desse algoritmo pode ser baixado em: https://github.com/pedrosoarnaldo/python_programs/tree/master/FakeNews

ANEXO I - URLs de notícias verdadeiras utilizadas no algoritmo

http://dicadeculinaria.com/native/tabua-degeladora/?utm_source=tab&utm_campaign=tabua&utm_source=tab&utm_campaign=tabua_desktop
http://www.criacionismo.com.br/2019/06/como-olhamos-para-natureza.html
https://agora.folha.uol.com.br/sao-paulo/2019/06/governo-doria-anuncia-ampliacao-da-linha-2-verde-ate-a-penha.shtml
https://blogdomenon.blogosfera.uol.com.br/2019/06/03/tite-deixa-claro-que-neymar-e-apenas-mais-um/
https://carros.uol.com.br/noticias/redacao/2019/06/04/bolsonaro-quer-fim-de-exame-de-drogas-e-multa-menor-por-rodar-sem-capacete.htm
https://delas.ig.com.br/amoresexo/2019-06-03/sem-dor-na-consciencia-87-dos-adulteros-nao-se-arrependem-de-traicao.html
https://economia.ig.com.br/empresas/2019-06-03/governo-desiste-de-fundir-agencias-de-transporte-e-avalia-futuro-da-valec.html
https://economia.ig.com.br/previdencia/reforma-urgente/2019-06-04/sem-a-reforma-nao-teremos-como-manter-os-gastos-publicos-alerta-economista.html
https://economia.uol.com.br/noticias/estadao-conteudo/2019/06/03/relator-pretende-apresentar-relatorio-da-reforma-entre-quinta-e-segunda-feira.htm
https://economia.uol.com.br/noticias/redacao/2019/06/03/pente-fino-inss-senado.htm
https://economia.uol.com.br/noticias/redacao/2019/06/11/amazon-google-apple-pesquisa-kantar-marcas-mais-valiosas-mundo.htm
https://educacao.uol.com.br/noticias/2019/06/12/trf-1-derruba-liminar-que-suspendia-bloqueio-orcamentario-nas-universidades.htm
https://entretenimento.uol.com.br/noticias/redacao/2019/06/02/morre-mc-reaca-que-compos-musicas-a-favor-de-bolsonaro-presidente-lamenta.htm
https://esporte.ig.com.br/futebol/2019-06-02/mensagens-divulgadas-por-neymar-nao-comprovam-sua-inocencia.html
https://esporte.ig.com.br/futebol/2019-06-02/neymar-vai-ser-investigado-por-mostrar-fotos-de-mulher-que-o-acusou-de-estupro.html
https://esporte.uol.com.br/futebol/ultimas-noticias/2019/06/02/spfc-x-cruzeiro.htm
https://esporte.uol.com.br/futebol/ultimas-noticias/2019/06/03/ex-advogado-de-mulher-que-acusa-neymar-diz-que-ela-mentiu-atual-nega.htm

https://esporte.uol.com.br/futebol/ultimas-noticias/2019/06/04/flamengo-x-corinthians.htm
https://esporte.uol.com.br/futebol/ultimas-noticias/2019/06/04/vice-da-cbf-aposta-que-neymar-pede-dispensa-tem-novo-video-vindo.htm
https://extra.globo.com/noticias/brasil/casal-de-idosos-morre-com-diferenca-de-cinco-minutos-no-rio-grande-do-sul-23731920.html
https://g1.globo.com/am/amazonas/noticia/2019/06/10/nao-tem-nenhuma-orientacao-ali-naquelas-mensagens-diz-moro.ghtml
https://g1.globo.com/jornal-nacional/noticia/2019/06/03/maioria-e-contra-flexibilizacao-de-posse-e-porte-de-armas-diz-ibope.ghtml
https://g1.globo.com/mt/mato-grosso/noticia/2019/06/03/motociclista-atropela-anta-vai-para-casa-reclama-de-dores-e-morre-3-horas-depois-em-hospital-em-mt.ghtml
https://g1.globo.com/politica/noticia/2019/06/01/bolsonaro-diz-que-quer-manter-estados-e-municipios-na-reforma-da-previdencia.ghtml
https://g1.globo.com/politica/noticia/2019/06/03/senado-aprova-mp-que-cria-programas-de-combate-a-fraudes-previdenciarias.ghtml
https://g1.globo.com/politica/noticia/2019/06/11/nao-se-combate-corrupcao-a-ferro-e-fogo-diz-marco-aurelio-para-gilmar-prova-ilegal-pode-valer.ghtml
https://g1.globo.com/rj/rio-de-janeiro/noticia/2019/06/03/jovem-encontrada-morta-em-nova-iguacu-foi-morta-por-tiro-na-cabeca.ghtml
https://g1.globo.com/sp/santos-regiao/noticia/2019/06/02/mae-e-filha-sao-agredidas-a-marretadas-apos-marido-reclamar-de-post-na-web.ghtml
https://g1.globo.com/sp/santos-regiao/noticia/2019/06/03/mulher-e-presa-apos-abandonar-a-filha-sozinha-e-ir-para-a-balada-em-sp.ghtml
https://g1.globo.com/sp/sao-paulo/noticia/2019/06/03/ex-advogado-de-mulher-que-acusa-neymar-diz-que-ela-havia-relatado-agressao-e-nao-estupro.ghtml
https://g1.globo.com/sp/sao-paulo/noticia/2019/06/10/prefeitura-de-sp-anuncia-inicio-da-reurbanizacao-do-anhangabau-e-diz-que-pretende-conceder-area-a-iniciativa-privada.ghtml
https://g1.globo.com/sp/vale-do-paraiba-regiao/noticia/2019/06/10/onibus-desceu-desenfreado-diz-testemunha-de-acidente-com-10-mortos-em-rodovia-de-sp.ghtml
https://gente.ig.com.br/celebridades/2019-06-02/correcao-paulo-pagni-baterista-da-rpm-esta-vivo-e-internado-em-estado-grave.html

https://globoesporte.globo.com/futebol/copa-america/noticia/apos-bom-desempenho-em-amistoso-do-uruguai-arrascaeta-ganha-destaque-e-vira-manchete.ghtml
https://globoesporte.globo.com/sc/futebol/times/chapecoense/noticia/chape-desiste-de-pedir-anulacao-de-jogo-contra-o-goias.ghtml
https://gq.globo.com/Musa/noticia/2018/03/gabriela-pugliesi-posa-com-o-bumbum-para-o-alto-e-topless.html
https://gq.globo.com/Musa/noticia/2019/04/paloma-bernardi-posa-de-biquini-de-oncinha-e-encanta-fas.html
https://gq.globo.com/Musa/noticia/2019/06/giovanna-ewbank-posa-para-bruno-gagliasso-em-viagem-para-dubai.html
https://inovacao-brasil.site/health-watch/?cep=OiU4xWWUuxpfGXQo8AJo9iBkCZzvtH5rMrQIjlZbYmaipbSxeCLoLIMVJko9eqRkE_Ld8IS6HDO4SmwLdOWh3KucZzTOZ02eStNSVfGDG8pBC5_QbbCqpMUvwQHwJIbAsGtdT-UFGDK9fUMy7JY8lEvc1r63hHKASulczsgfi26ABMxV-gSRpUSxLhaK2HcKmILF__BuxkCccbNfRRZ5xvHPkk8Zmx9EBtcPH4UVhdNZftfLt8SVomrul9urjS5cmLRZ2Kgt9TQUD7U9CWpLIGU9riSU6t7fR4HbNfveaVYTf60RMEIvVNbT3iNr-zJ_AuCU8vPmX7zk-w7fFCDon66j4sKIilpN5HGa6wQdwyKjO0zdsDK6z8-WxKTHkq0GphmhFG13YVPnJM1kv7umqsrS5ZB1FnYgabuNjeaPHMq6FJ9SniIHNQ4LzBlaDNCJpyrHCsH2zKFb7Kx4PptOiO3hdFil8Gc7-6yd4_HMQPpo4ei7EMKoxazX4CViEOa9ipqc9pQRMrT2xU6VT2Ov7x4FwO_kR6_D7xXePA6kMuQ1Hjy9-KIjZMAEeb-gG7vV8fD1tQolrABhELNU53ZnLI7rvoBHrsu1bXXszDkqsFJFfnzUSaLapMbf9x8-fffvMFAZP4h2expTU3cgU8r3P2NCcIrdLlybXjShNnf3t1W8RuLQ3ihzg9-7QE7vhYzJKwnRrWmtgdeu-hFuS-iPnpk2NbN2Kiuy10Tx38OIrxY&campaign_id=001539cc052ef02d21d729496d85881df7&campaign_id=001539cc052ef02d21d729496d85881df7&Ad_Id=0082c12d0bab40700d2c44bd228010bd69&Ad_Id=0082c12d0bab40700d2c44bd228010bd69&Ad_title=Rel%C3%B3gio%20de%20sa%C3%BAde%20vira%20febre%20entre%20os%20brasileiros&utm_term=MSN%20-%20PT-BR%20-%20NTP&utm_term=MSN%20-%20PT-BR%20-%20NTP&utm_content=MSN%20Brazil%20(MSN%20Intl)&utm_content=MSN%20Brazil%20(MSN%20Intl)&section_id=00274b318fb52dd0245d170690e6858798&section_id=00274b318fb52dd0245d170690e6858798&userid=8yzvL67rOHTJaOFbD5AC4YK4GTCkVnmMugGPUVltNSE7kvCw9T97UvGfyJl6zjsH&userid=8yzvL67rOHTJaOF

bD5AC4YK4GTCkVnmMugGPUVltNSE7kvCw9T97UvGfyJl6zjsH&publisher_id=00b8ea97a3e645a9480958879663c45ff1&publisher_id=00b8ea97a3e645a9480958879663c45ff1&ad_title=Rel%C3%B3gio%20de%20sa%C3%BAde%20vira%20febre%20entre%20os%20brasileiros
https://motorsport.uol.com.br/motogp/news/rossi-gp-da-italia-foi-um-dos-meus-piores-em-muito-tempo-na-motogp/4416271/
https://noticias.band.uol.com.br/mundo/noticias/100000960482/duas-criancas-brasileiras-morrem-em-deslizamento-de-pedras-no-chile.html
https://noticias.r7.com/minas-gerais/cpi-de-brumadinho-aprova-quebra-de-sigilo-de-ex-presidente-da-vale-04062019
https://noticias.r7.com/minas-gerais/mpt-mira-30-barragens-com-alto-potencial-de-dano-em-minas-04062019
https://noticias.uol.com.br/politica/ultimas-noticias/2019/06/04/lula-regime-semiaberto-procuradoria.htm
https://noticias.uol.com.br/politica/ultimas-noticias/2019/06/11/moro-bolsonaro-encontro-divulgacao-mensagens.htm
https://oglobo.globo.com/cultura/morre-atriz-flora-diegues-34-anos-filha-do-cineasta-caca-diegues-23712774
https://revistaglamour.globo.com/Celebridades/noticia/2019/06/marcos-dupla-de-belucci-manda-recado-para-paula-fernandes-e-entrega-furo-de-luan-santana.html
https://revistaquem.globo.com/QUEM-News/noticia/2019/06/carla-prata-fala-sobre-batalha-contra-doenca-rara-autoimune-em-2017-tive-minha-primeira-paralisia.html
https://talesfaria.blogosfera.uol.com.br/2019/06/10/obstrucao-contra-moro-atrasa-votacao-de-credito-suplementar-de-r-2489-bi/
https://twitter.com/_dersp/status/1138135737809981440
https://twitter.com/BombeirosPMESP/status/1138166011084058624
https://twitter.com/folha/status/1137874276713926656
https://twitter.com/folha/status/1138025512377376769
https://twitter.com/folha/status/1138162666575409155
https://twitter.com/folhapsol/status/1105163652347424773
https://twitter.com/iG/status/1138113593415745537
https://twitter.com/igormello/status/1138148913691709441
https://twitter.com/SouSaoPauloFC/status/1138167617196351489

https://ultimosegundo.ig.com.br/mundo/2019-06-02/ataque-de-misseis-de-israel-na-siria-deixa-ao-menos-10-mortos.html
https://ultimosegundo.ig.com.br/mundo/2019-06-03/trump-chega-a-londres-para-visita-oficial-e-critica-prefeito-e-um-perdedor.html
https://ultimosegundo.ig.com.br/politica/2019-06-03/maia-nega-pedidos-de-viagem-a-deputados-para-votar-reforma-da-previdencia.html
https://universa.uol.com.br/reportagens-especiais/universa-talks-um-encontro-de-mulheres-que-transformam-o-mundo/index.htm
https://www.bol.uol.com.br/entretenimento/2019/06/03/mulher-agredida-por-mc-reaca-esta-internada-e-devera-passar-por-cirurgia.htm
https://www.bol.uol.com.br/listas/estao-roubando-a-corrida-e-tom-de-advogado-as-frases-do-gp-do-canada.htm
https://www.bol.uol.com.br/noticias/2019/05/31/damares-defende-que-escolas-discutam-abstinencia-sexual-e-critica-popey.htm
https://www.bol.uol.com.br/noticias/2019/06/11/oposicao-tenta-barrar-decreto-que-esvazia-orgao-de-combate-a-tortura.htm
https://www.bol.uol.com.br/noticias/2019/06/12/somos-um-pais-de-perdedores-diz-dono-da-havan.htm
https://www.estrelando.com.br/nota/2019/06/02/sheila-mello-esta-internada-em-hospital-de-sao-paulo-238218
https://www.terra.com.br/noticias/tecnologia/walmart-comeca-a-entregar-compras-direto-na-geladeira-dos-clientes-nos-eua,15197004792e8d0bba957fcc21d0a3803ej9bdsx.html
https://www1.folha.uol.com.br/colunas/monicabergamo/2019/06/moro-vai-depor-no-congresso-no-proximo-dia-19.shtml
https://www1.folha.uol.com.br/mercado/2019/06/odebrecht-entra-com-maior-pedido-de-recuperacao-judicial-da-historia-do-pais.shtml
https://extra.globo.com/casos-de-policia/celular-amortece-bala-pm-ferido-de-raspao-em-nova-iguacu-23716246.html

ANEXO II - URLs de notícias falsas utilizadas no algoritmo

http://noticia-agora.com/ren31/blog/adv-4/
http://noticia-agora.com/kifina/blog/ex-bbb-priscila-pires/
http://mercadodigital.shop/xtradent/adv2/?src=outbrain_iG_desktop_adv2_c2_interest&utm_medium=5cf5cdd77614ec00014c9a21&utm_source=outbrain&utm_campaign=xtradent-c2-desktop_interest&d_adv=0
https://portal.horanoticia.com/tb12/ob/estimulante2/?src=outbrain&utm_source=outbrain&utm_medium=Casa+Vogue+-+Lazer+e+Cultura&utm_campaign=TBL14-ALL&utm_content=Estimulante+natural+ajuda+milhares+de+pessoas+e+esgota+no+Brasil.+Mas+&utm_term=Casa+Vogue+-+Lazer+e+Cultura&wt_token=afaa265e329f4a278246d3a7065fd76b&wt_campaign=008f53aa547f50767076f975ed421811c8&wt_ad=0002dc946a840e31eb75541d934676a8eb&wt_site=00c78b96442ada3338a091766f2e87bed0&wt_network=NativeAds
http://guiasaude.me/antiacne-aa9/
http://guiasaude.me/biocaps-06-m/
http://guiasaude.me/biocaps-02-male/
https://twitter.com/folha_sp/status/1137161474420420614
https://twitter.com/folha_sp/status/1137095703640064002
https://twitter.com/folha_sp/status/1136393147603460096
https://twitter.com/folhapsol/status/1136106218081140736
https://twitter.com/folhapsol/status/1121937086595121152
https://twitter.com/folhapsol/status/1102537172002000897
https://twitter.com/folhapsol/status/1111071634293702656
https://www.coolimba.com/view/groom-loves-other-girl-br-co/?src=outbrain&utm_source=outbrain&utm_medium=008ac1b956e8491d65933ffaeb42e5d97a&utm_campaign=00b3c65b24f234b51db855f8d049b2100a&utm_key=20&utm_content=00e8980719f5cceb939c9439992a436bd3&utm_term=CO_D_BR_groom-loves-other-girl-br-co_ian_a_75646
http://ofertas-top.com/maxtv-br-out/?utm_source=Outbrain&utm_medium=Discovery&utm_campaign=antena_br_desk&utm_content=Este+truque+para+obter+canais+em+HD+gratis+vira+febre+no+Brasil%21&utm_term=MSN+Brazil+%28MSN+Intl%29_MSN+-+PT-BR+-+Not%C3%ADcias&ad_id=00b8982560be3e4997e6e4d5d4c747afa4&outbrainClickI

d=v1-36a63d526e869489be26479cd7b7bc66-
00ceaa70af7f055bb0ea1d5ed25f2c298c-
g43wgzdbgnrdqllfhe4tcljummzdiljymqydalldmrrgmnbxmnsdomjzha
https://www.aceleradordaleitura.com.br/acelerador-da-leitura-d-t-01-c-
2/?utm_source=taboola&utm_medium=referral
https://www.japantech.net/enence_V5/br/002_mar/index.html?sxid=6a6peq6yyne4&e
xid=CjAzNzYzNjkyZi0zYjBiLTRkNjAtOGVjNC0yMTdhYzI5MWE1YzgtdHVjdDNjMTB
mNjgSDXlub3QtbXVhbWEtc2M&site=msn-
brazil&ci=2262551&cii=231466728&utm_source=taboola&utm_medium=referral&utm
_term=Esta+inven%C3%A7%C3%A3o+japonesa+permite+que+voc%C3%AA+fale+
43+idiomas&utm_content=http%3A%2F%2Fcdn.taboola.com%2Flibtrc%2Fstatic%2F
thumbnails%2F28c4d506a3583860ef0cd7c40cb9a609.jpg&wid=1023541&platform=
Desktop&name=EVENT_NAME
https://dailychasers.com/pt-br/uma-noiva-descobre-que-seu-noivo-esta-traindo-ela-
ela-se-vinga-no-altar-com-uma-
vinganca/?utm_source=taboola&utm_campaign=DAI_BRA_D_vreemdgaanbruiloft_v
1&utm_medium=msn-brazil&utm_content=227600366
http://boaforma.vip/n5/
https://materiais.euqueroinvestir.com/o-segredo-dos-grandes-
bancos/?utm_medium=cpc&utm_source=taboola&utm_campaign=o-segredo-dos-
bancos-desktop&utm_content=msn-edgedefaulthomepage-
brazil&utm_term=Investidor+saca+dinheiro+de+grandes+bancos+e+revela+motivos
https://bizcapital.com.br/emprestimo-para-
empresas/?utm_source=taboola&utm_medium=msn-edgedefaulthomepage-
brazil&utm_campaign=TB-20190425-IMG-WHITE-LIST-CPC
https://saudehoje.meiahora.info/ftp/tb/baixar-
acucar/?src=taboola&utm_source=taboola&utm_medium=msn-brazil-
home&utm_campaign=FTP14-
DT&utm_content=Isto+obriga+o+a%C3%A7%C3%BAcar+do+sangue+baixar+e+as+
dores+sumirem.+In%C3%A9dito+confira%21&utm_term=msn-brazil-
home&wt_token=d2fde0c2e6a64a71813bb02bd7fc0dff&wt_campaign=2248611&wt_
ad=224060549&wt_site=msn-brazil-home&wt_network=NativeAds
https://www1.folha.uol.com.br/mercado/2019/06/bolsonaro-diz-que-no-passado-
apenas-militares-foram-sacrificados-na-previdencia.shtml

https://noticia-brasil.com/alfa/?utm_source=taboola&utm_medium=PHYTO-PL-DESK-CP2&utm_campaign=msn-edgedefaulthomepage-brazil
https://twitter.com/folhapsol/status/1129945934857740289
https://melhor-saude.info/news/?utm_source=TBC20AD20&utm_source=taboola&utm_medium=referral
https://forum.cifraclub.com.br/forum/11/241434/
https://ipschneider.com.br/partido-diabolico-bandidos-do-pt-estao-abrindo-buracos-nas-estradas-federais-do-nordeste-para-conseguir-verbas-a-estrada-e-boa-de-se-andar-mas-a-mafia-petista-atraves-de-maquinario-estao-sabotando/