Trabalho de Fundamentos de Ciência de Dados
Os Impactos das Fake News na População Mundial

Grupo de Trabalho:

Monica Vanessa Macedo Rodrigues Novellino
Sírius Thadeu Ferreira da Silva
Vinicius Deodoro da Silva

1. Detalhamento e descrição textual da definição do problema a ser explorado pela equipe:

Através das mídias sociais, os cidadãos estão em contato constante com as histórias, fatos e notícias políticas e sociais. Isso é resultado da revolução tecnológica que, com *tablets* e *smartphones*, coloca o mundo inteiro nas mãos das pessoas e tem, como consequência, uma revolução ao nível do *marketing* político-social e na forma como os políticos se comunicam com seus eleitores. Devido a esse fenômeno, na última década tem se observado um consequente crescimento do uso político-social das mídias sociais. As vantagens políticas do *marketing* digital, quando comparado ao *marketing* "tradicional", é que ele mostrou-se muito mais eficaz, atingindo um grande número de pessoas em um curto espaço de tempo, e a medição de seu impacto é imediata, principalmente com o *marketing* efetuado nas redes sociais.

O crescente uso político-social das mídias sociais tem ocasionado vários efeitos adversos que impactam tanto diretamente quanto indiretamente a população mundial: acentuação da polarização política, formação das bolhas sociais, disseminação e uso político de *fake news*, manipulação político-social da opinião pública, interferências diretas nos resultados de eleições, ameaças ao sistema eleitoral e à democracia e, mais recentemente, até mesmo agravando sérios problemas de saúde pública.

Portanto, o presente projeto se propõe a estudar os impactos das *fake news* na população mundial, através de análises qualitativas e quantitativas das notícias falsas e suas principais características. Pesquisas no sentido de estudar os problemas gerados pela disseminação de *fake news* são importantes para conscientizar os cidadãos sobre todos os malefícios causados pela desinformação. Ao realizarmos um estudo de percepção social sobre a disseminação de *fake news*, será possível identificar os principais fatores e indicadores responsáveis pelos impactos gerados na população mundial.

Neste projeto focaremos em estudar determinados aspectos envolvendo as *fake news*, tais como: principais propagadores; países mais afetados e línguas mais utilizadas para sua

disseminação; termos mais usados; identificar seus potenciais alvos; popularidade, potencialidade de disseminação e maiores períodos de atividade; capacidade de atingir e atrair o público em geral, bem como seu alcance; estatísticas gerais dos tipos atualmente disseminados na internet.

2. Descrever tecnicamente o dataset e sua fonte, o quê pretendem fazer e o quê vão e como extrair (plano dos experimentos):

O *dataset* (https://www.kaggle.com/mrisdal/fake-news) é um pequeno passo em direção à primeira etapa para entender e resolver o problema das *fake news* e como fazer para detectá-las e impedir seu compartilhamento viral. Ele contém texto e metadados extraídos de 244 *sites* marcados como *"bullshit"* pela extensão de navegadores (Chrome / Firefox) **"BS Detector"** (https://awesomeopensource.com/project/selfagency/bs-detector / https://gitlab.com/bs-detector/bs-detector) de Daniel Sieradski (https://github.com/selfagency).

O *dataset* contém texto e metadados de 244 *sites* e apresenta 12.999 postagens no total coletadas em um período de 30 dias (25/10/2016 a 25/11/2016). Os dados foram extraídos usando a API **webhose.io** (https://webhose.io/) e por estar vindo desse *crawler*, nem todos os *sites* identificados pelo **BS Detector** estão presentes no *dataset*. Cada *site* foi rotulado de acordo com os parâmetros de análise do **BS Detector**. As fontes de dados que não tinham um rótulo (segundo a classificação do **BS Detector**) foram simplesmente rotuladas como "bs" (*bullshit*). Não há (aparentemente) nenhuma fonte de notícias genuína, confiável ou verdadeira representada no *dataset* (até agora), então não há como acreditar em nada que está escrito nestes artigos.

Para mais detalhes sobre a proveniência do *dataset*, verificar a **seção 7**. Os dados coletados são dispostos em 20 colunas, das quais 17 podem ser utilizadas para analisá-los:

- **ord_in_thread:** O número de sequência da postagem no contexto do tópico, onde 0 significa a primeira postagem – Inteiro;
- **author:** Uma lista de nomes de usuários / autores / atores que escreveram a postagem – Array[String];
- **published:** A data / hora em que a página ou postagem foi publicada (se detectada) – Data (Formato yyyy-MM-dd'T'HH:mm:ss.SSSXXX);
- **title:** O título da página ou postagem – String;
- **text:** O texto na página ou postagem – String;
- **language:** A linguagem do texto – String;
- **site_url:** O *link* para a página – String;
- **country:** O país do *site*, determinado automaticamente pelo idioma do site, IP e TLD (Lista completa de códigos dos países em https://docs.webhose.io/reference#countries-codes) – String;
- **domain_rank:** Um *rank* que especifica a popularidade de um domínio (por tráfego mensal) – Inteiro;

- **thread_title:** O título do tópico – String;
- **spam_score:** Um valor de pontuação flutuante entre 0,0 e 1,0 que indica o nível *"spam"* do texto do tópico – Real;
- **replies_count:** O número de respostas à postagem principal – Inteiro;
- **participants_count:** O número de pessoas diferentes que participaram da discussão – Inteiro;
- **likes:** O número total de curtidas que o artigo recebeu – Inteiro;
- **comments:** O número total de comentários que o artigo recebeu – Inteiro;
- **shares:** O número total de compartilhamentos que o artigo recebeu – Inteiro;
- **type:** Classificação do artigo de acordo com o *BS Detector* (Lista completa das *tags* em https://web.archive.org/web/20170207020542/http://www.opensources.co/) – String.

Para fins de nosso experimento, iremos analisar o conteúdo dessas 17 colunas, correlacionando-as para entender a sistemática das *fake news*. Essas colunas contêm informações importantes que podem apontar respostas para as questões levantadas neste trabalho, tais como:

- As colunas de autor e *site* podem nos informar quais são os maiores propagadores de *fake news* no período coletado;
- As colunas de linguagem e país podem indicar os países e línguas que foram os principais responsáveis pela disseminação das *fake news* nesse período;
- Através dos campos de títulos e texto poderemos detectar os termos mais usados nessas *fake news*, fazer uma análise de sentimento, identificar os potenciais alvos dessas *fake news*, etc.;
- A coluna de *ranking* de domínio nos informará sobre a popularidade desses *sites* propagadores de *fake news*;
- A coluna de *score* de *spam* é capaz de indicar a potencialidade de *spamming* dessas *fake news*;
- Analisando a coluna de data de publicação poderemos verificar os maiores períodos de atividade das *fake news* nesses *sites*;
- As colunas de interação social com as postagens, quando analisadas em conjunto, poderão nos informar importantes dados sobre a capacidade de atingir e atrair o público em geral para essas *fake news*, popularidade dos tópicos discutidos, propensão de compartilhamento dessas *fake news*, alcance das *fake news*, etc.;
- Por fim, a coluna de tipos de *fake news* nos ajudará a entender mais profundamente as classificações dessas *fake news* e sua popularidade, alcance, espalhamento, etc.

Através da linguagem de programação **python** e suas principais bibliotecas para ciência de dados (como **NumPy**, **Pandas**, **NLTK**, **Matplotlib**, **Seaborn**, etc.) pretendemos explorar os

dados anteriormente mencionados, realizando os tratamentos necessários para extrairmos as informações almejadas por esta pesquisa. Como estamos lidando basicamente com três tipos de dados (Inteiro, Real e String) distribuídos nessas 17 colunas, cada coluna (ou conjunto de coluna) deverá passar por processos de extração e tratamento de dados distintos a fim de coletarmos as informações necessárias. Mais detalhes sobre o plano do experimento podem ser encontrados na **seção 6** deste documento.

3. Apresentar os objetivos geral e específico do projeto de DS:

3.1. Objetivo Geral:

O presente projeto tem como objetivo geral estudar as principais características das *fake news*, analisando-as e correlacionando-as a fim de identificar os potenciais impactos das *fake news* na população mundial.

3.2. Objetivos Específicos:

Através da análise dos principais aspectos das *fake news*, pretendemos responder algumas questões, quais sejam:

- Quem são seus principais autores / propagadores?
- Quais países são mais afetados?
- Quais línguas são mais utilizadas para sua disseminação?
- Quais termos são mais usados?
- Quem são seus alvos potenciais?
- Quais tópicos são mais comentados?
- Quais tópicos são mais curtidos?
- Quais tópicos são mais compartilhados?
- Quais os maiores períodos de atividade?
- Quais tipos são disseminados?

4. Apresentar métodos de data cleaning / tratamento de dados que serão usados:

Em nosso projeto utilizaremos métodos de tratamento de dados voltados para cada tipo de dado (Inteiro, Real e String) encontrado e técnicas de processamento de texto (Atomização, Contagem de palavras, Divisão de frases, Radicalização, Normalização e Remoção das *"stop words"*) visando possibilitar uma análise textual automática, até mesmo realizando análise de sentimento quando possível.

No caso das colunas de tipos numéricos (Inteiro e Real), verificaremos a existência de dados nulos ou **NaN** (*Not a Number*). Em caso positivo, de acordo com as regras descritivas da coluna em questão, poderemos atribuir o valor padrão 0 (para as colunas onde a ocorrência desse valor não é possível naturalmente) ou o valor padrão negativo -1 (para as colunas onde a ocorrência de valores negativos não são possíveis naturalmente).

Já no caso das colunas tipo texto (String), verificaremos a existência de dados nulos e os substituiremos pela String vazia. Além disso, utilizaremos técnicas de processamento de texto para normalizar esses dados (uniformização das letras maiúsculas e minúsculas, remoção de acentuação, pontuação, caracteres inválidos, etc.).

Também poderemos utilizar métodos de detecção de *outliers* baseados em estatística, distância ou modelo, cada um sendo aplicado nos casos em que forem julgados mais aptos. Após análise inicial do *dataset*, foram encontrados 8297 registros com algum tipo de dado nulo. Somente 7 das 20 colunas possuem dados nulos. Dessas, seis são as colunas que nos interessam e iremos trabalhar: **author**, **title**, **text**, **country**, **thread_title** (colunas do tipo String) e **domain_rank** (Inteiro).

5. Apresentar a proposta de modelo de extração de conhecimento e visualização de dados que será adotado:

Em nosso projeto, poderemos utilizar alguns dos principais modelos de extração de conhecimento, quais sejam:

- Modelos estatísticos para descobrir padrões e construir modelos preditivos;
- Modelos de clusterização para agrupamento de dados, identificando dados semelhantes e não semelhantes entre si;
- Árvore de decisão, um modelo preditivo que forma um desenho que lembra uma árvore, cujas ramificações do modelo funcionam como métodos de classificação. A árvore de decisão é uma técnica que permite a fácil interpretação dos dados e mostra o caminho a ser percorrido para alcançar determinado objetivo;
- Regras de associação, uma técnica que ajuda a encontrar associações entre dois ou mais itens. Ao definir relações entre as variáveis do *dataset* é possível descobrir padrões escondidos;
- Redes neurais, utilizadas com mais frequência nos estágios iniciais do processo de mineração de dados, servem para modelar relações entre os dados que entram e que saem do processo de mineração;
- Classificação, uma técnica que ajuda a obter informações importantes sobre dados e metadados. Está intimamente relacionada com a técnica de clusterização e utiliza a árvore de decisão ou rede neural;
- Visualização, técnicas utilizadas no início do processo de mineração de dados como um primeiro passo para descobrir padrões ocultos em um grande grupo de dados.

Quanto às técnicas de visualização de dados que poderão ser utilizadas em nossa pesquisa, histogramas e gráficos QQ permitem examinar a distribuição de uma única variável e gráficos de dispersão mostram como duas variáveis estão relacionadas. Além disso, *Boxplots* são uma ótima ferramenta para visualizar valores fora de um determinado intervalo. Eles

podem ser estendidos com *boxplots* agrupados ou gráficos de violino que nos permitem comparar as distribuições entre subconjuntos de dados.

Gráficos de dispersão nos permitem obter uma noção das relações multivariadas, dentre outras coisas. Os histogramas também podem, às vezes, oferecer uma visão adicional se exibirmos vários histogramas em um gráfico ou criarmos um histograma empilhado. Um mapa de calor pode ser gerado com base em uma matriz de correlação para nos auxiliar na análise do *dataset*.

Por fim, diversas outras técnicas de visualização de dados nos permite trabalhar de formas distintas com nosso *dataset*, cada uma com suas próprias vantagens:

- Histogramas permitem examinar a distribuição de variáveis contínuas;
- *Boxplots* permitem identificar *outliers* para variáveis contínuas;
- *Boxplots* Agrupados permitem descobrir valores inesperados em um determinado grupo;
- Gráficos de Violino permitem examinar a forma de distribuição e *outliers*;
- Gráficos de Dispersão permitem visualizar relações bivariadas;
- Gráficos de Linha permitem examinar tendências em variáveis contínuas.

6. Plano do experimento a ser executado:

Em nosso projeto usamos o Google Colab (https://colab.research.google.com/). O Google Colaboratory ou "Colab" permite escrever código Python diretamente no navegador, com fácil compartilhamento e acesso gratuito a GPUs (e demais *hardwares* e *softwares* necessários). Nenhuma configuração extra é necessária para começar a trabalhar com essa ferramenta, exceto a instalação de algumas bibliotecas específicas do Python que porventura não estejam já instaladas no ambiente (e seja preciso utilizar durante o projeto), facilitando assim o trabalho de um cientista de dados.

O Google Colaboratory é construído sobre o Jupyter Notebook e fornece um ambiente interativo chamado *notebook* Colab que permite escrever e executar código (*script* Python). Os *notebooks* do Colab permitem combinar código executável e *rich text* em um só documento, além de imagens, HTML, LaTeX e muito mais. Os *notebooks* do Colab são armazenados na conta do Google Drive. É possível compartilhar os *notebooks* do Colab facilmente com outras pessoas e permitir que elas façam comentários ou até editem o documento. O Colab fornece completamentos automáticos para explorar atributos de objetos Python, bem como para visualizar rapidamente sua documentação.

É uma das ferramentas ideais para se trabalhar com Ciência de Dados, pois com o Colab podemos aproveitar todo o potencial das conhecidas bibliotecas Python para analisar e ver dados (numpy, matplotlib, pandas, etc.). É possível importar para os *notebooks* do Colab os dados de uma conta do Google Drive, do GitHub e de muitas outras fontes. Além disso, o Colab é altamente integrável com o Google Drive, permitindo montar um *drive* virtual para leitura e armazenamento perpétuo dos dados gerados pelos *scripts* Python (uma vez que o ambiente de execução do Colab é temporário, sendo "zerado" após longos períodos de inatividade do

usuário – ou a pedido do próprio usuário). Os *notebooks* do Colab executam os códigos nos servidores em nuvem do Google, significando que podemos tirar proveito da potência de *hardware* do Google, como GPUs e TPUs, independentemente da potência do computador local, necessitando somente de um navegador (com acesso à internet).

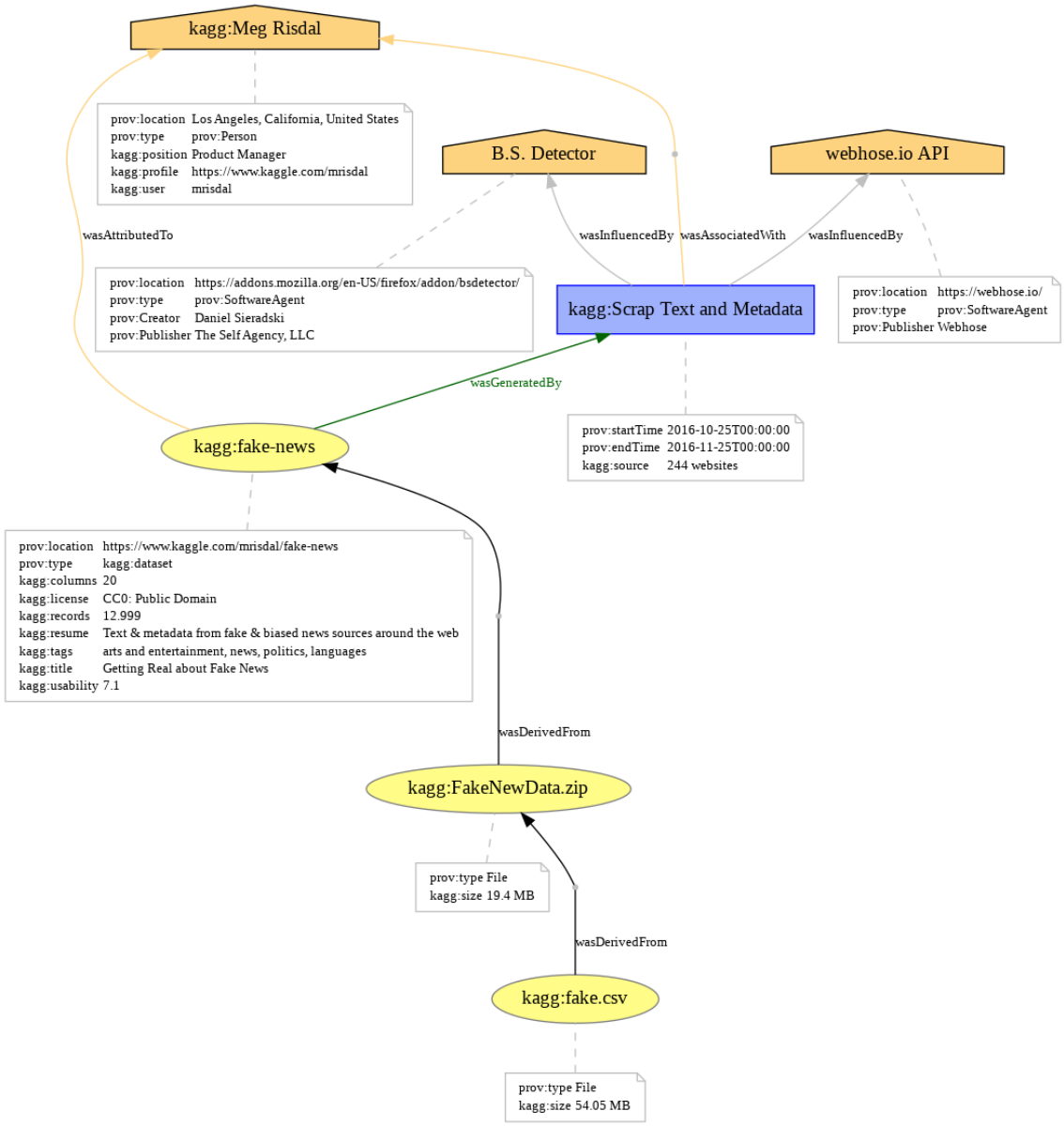
Além do planejamento prévio descrito na **seção 2** deste documento, pretendemos usar também uma biblioteca de análise exploratória de dados chamada Sweetviz (https://pypi.org/project/sweetviz/). Ela pega os *dataframes* do pandas e cria um relatório HTML autocontido que pode ser visualizado em um navegador ou integrado em *notebooks*. Além de criar visualizações perspicazes e bonitas com apenas duas linhas de código, ela fornece análises que levariam muito mais tempo para serem geradas manualmente, incluindo algumas que nenhuma outra biblioteca fornece tão rapidamente, como:

- **Análise de alvo:** mostra como um valor alvo (por exemplo, "Sobreviveu" no *dataset* do Titanic) se relaciona com outros recursos;
- **Comparações de *dataset*:** entre *datasets* (por exemplo, "Treino x Teste") e intra-conjunto (por exemplo, "Homem x Mulher");
- **Correlações / associações:** integração total de correlações e associações de dados numéricos e categóricos, tudo em um único gráfico e tabela.

Por fim, todos os *scripts* e artefatos desenvolvidos para este projeto estão disponíveis para acesso através do GitHub (https://github.com/ppgi-ufrj-data-science/FakeNews).

7. Projeto de coleta de metadados da proveniência dos experimentos:

Para coletar os metadados de proveniência do *dataset* escolhido (https://www.kaggle.com/mrisdal/fake-news), descrito na **seção 2** deste documento, e gerar o respectivo grafo de proveniência (mostrado abaixo) foi utilizada a biblioteca PROV do Python (https://pypi.org/project/prov/), em sua versão 2.0.0.



8. Projeto para tornar o experimento reprodutível:

Seguindo-se os preceitos dispostos em “Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions” (Fabien C. Y. Benureau e Nicolas P. Rougier), coletamos todas as informações necessárias de ambiente (Sistema Operacional e sua arquitetura; Linguagem de Programação e as bibliotecas utilizadas) para tornar o experimento reprodutível. Além disso, os *scripts* Python estão sendo criados com o maior zelo possível, visando documentar os passos executados durante todo o experimento e seguir as boas práticas de programação para tornar esses códigos reusáveis.

Também, durante toda a escrita do artigo será documentada a metodologia utilizada, bem como descritos os processos, algoritmos e modelos utilizados durante o projeto, visando

assim permitir que este experimento seja replicado por outros cientistas. A seguir, descrevemos as configurações de ambiente utilizadas durante este projeto.

Para o experimento, utilizamos a ferramenta Google Colab, que possui as seguintes configurações de ambiente de execução:

- Sistema Operacional Ubuntu 18.04.5 LTS (Bionic Beaver), com kernel Linux versão 4.19.112+ (Chromium OS versão 10.0.0) e arquitetura x86_64 (64-bit) com ordem de byte “Little Endian” e dois processadores Intel® Xeon® de 2.20GHz cada;
- Python na versão 3.7.10 [GCC 7.5.0] e as bibliotecas Pandas versão 1.1.5, Beautiful Soup versão 4.6.3, PROV versão 2.0.0 e Sweetviz versão 2.0.9.

Mais informações sobre as configurações do ambiente utilizado durante o experimento podem ser visualizadas através do arquivo **Environment.conf**, disponível no GitHub (https://github.com/ppgi-ufrj-data-science/FakeNews/tree/main/out).