ANO

2025

UNINTER

ATIVIDADE PRÁTICA

NLP

Roteiro Elaborado por:

Prof. MSc. Guilherme Ditzel Patriota

INTRODUÇÃO

Olá a todos.

Sejam todos muito bem-vindos!

Esta avaliação foi planejada e preparada para a disciplina de Natural Language Processing dos Cursos de Tecnologia do Centro Universitário Internacional Uninter.

O objetivo desta atividade é fazer com que você, aluno, desenvolva os conhecimentos teóricos aprendidos na rota de aprendizagem, de maneira prática e aplicável no mercado de trabalho. Para tanto, será necessário o uso de um Python Notebook com as bibliotecas NLTK, scikit-learn, pandas, numpy, matplotlib, wordcloud e quais quer outras bibliotecas que você julgar necessário.

Você poderá criar seu notebook na plataforma do Google Colab ou em outra plataforma qualquer, desde que possa ser exportado um arquivo em formato .ipynb. Esta prática é baseada nas aulas teórica e prática 5, sobre a biblioteca NLTK.

Ao longo desse roteiro serão passadas as orientações gerais para realização da avaliação bem como os seus critérios de correção. Na sequência, apresenta-se um exemplo comentado de como se deve ser entregue uma questão. Seguindo o roteiro estarão as práticas a serem realizadas, cada uma delas possui uma explicação de como deve ser feita, como será cobrada e algumas dicas. Por fim, apresento uma seção as com as respostas das dúvidas mais frequentes realizadas por vocês. Bons estudos!

No mais, desejo-lhe boa atividade prática em nome dos professores da disciplina de NLP.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

ORIENTAÇÕES GERAIS

FORMATO DE ENTREGA

A entrega desta atividade prática deverá ser realizada pela área de “Trabalhos”, em formato PDF com o caderno de resolução da atividade prática e o link para seu arquivo Python Notebook (.ipynb) com toda a sua resolução do exercício de forma funcional. O link poderá ser de seu Github ou de seu Google Colab.

Em seu caderno de resolução deverão constar as imagens das duas nuvens de palavras referentes às duas questões, com o código usado para geração do modelo, o resultado da questão e mais o seu RU digitado em algum lugar da imagem da nuvem de palavras e do código (sua IP = Identificação Pessoal = seu RU).

O formato de entrega desejável dos prints das práticas desse roteiro, deve estar de acordo com o que é visto na seção “EXEMPLO DE APRESENTAÇÃO DE PRÁTICA”.

Recomenda-se que os trabalhos sejam enviados no formato .pdf. Uma vez que formatos .doc ou .docx podem apresentar falhas do tipo na codificação, carregamento ou apresentação de imagens. Sendo assim, fica **por conta e risco do estudante** se houver problemas com o documento enviados no formato doc ou docx ou outro formato editável.

Trabalhos feitos em outra forma que não seja utilizando Python com a biblioteca NLTK não serão aceitos!

CRITÉRIOS DE AVALIAÇÃO

Os critérios de avaliação desse trabalho visam deixar a avaliação o mais justa e transparente possível. Nessa avaliação, cada questão valerá 50,00 pontos, sendo um total de 100 pontos de trabalho.

Cada questão será composta por print do código, nuvem de palavras com o resultado e resposta da questão. As questões serão avaliadas e corrigidas individualmente conforme a seguinte equação:

$$N = (FE).(IP)\frac{COD + IMG + RESP}{6}$$

Em que:

N $(Nota\ da\ Questão)$: Nota total da questão, podendo variar de 0 até 50.

FE $(Formato\ da\ Entrega)$: Nota do Formato de Entrega, podendo variar de 0 até 1.

IP $(Identificação\ Pessoal)$: Nota Identificação Pessoal, podendo variar de 0 até 1.

COD $(Código)$: Nota do Código usado, podendo variar de 0 até 100.

$IMG(Imagens)$: Nota da Imagem com resultado correto (nuvem de palavras), podendo variar de 0 até 100.

$RESP(Resposta)$: Nota da Resposta com resultado correto, podendo ser 0 ou 100.

Cada um dos itens/critérios que compõe a equação acima será detalhado nas subseções a seguir. **Se mesmo assim houver dúvidas, não hesite em perguntar. O desconhecimento dos critérios não será aceito como desculpa!**

FORMATO DA APRESENTAÇÃO

O **formato da apresentação** é um dos **critérios de avaliação**, pois um profissional **deve** ser capaz de seguir normas no momento de elaboração de relatórios técnicos, manuais e outros documentos afins, bem como ser capaz de apresentar seus dados de forma limpa e compreensível.

As possíveis notas desse critério são apresentadas na tabela a seguir:

Tabela 1: Possíveis notas no formato de apresentação

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
1,00	Formato da apresentação está correto	Está de acordo com o exemplo (ver a seção "EXEMPLO DE APRESENTAÇÃO DE PRÁTICA" para maiores detalhes)
0,70	Formato da apresentação está parcialmente correto	Está muito próximo do exemplo, mas apresenta alguns erros
0,50	Formato da apresentação está incorreto	Não seguiu o exemplo.

IDENTIFICAÇÃO PESSOAL

Todas as questões deverão apresentar um **identificador pessoal** nas seguintes partes:

- **No código deve haver ao menos uma variável cujo nome seja composto pelo seu RU (e.g. contadorxxxxxx – onde o "x" s deve ser substituído pelo seu RU), mesmo que esta variável não seja utilizada em nenhuma parte do código.**
- **Nas imagens/nuvem de palavras, onde deverá conter seu RU escrito em algum local.**

As possíveis notas para esse critério são apresentadas na tabela a seguir:

Tabela 2: Possíveis notas critério de Identificação Pessoal

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
1,00	Apresentou o identificador pessoal no código e nas imagens/fotos.	Está de acordo com o exemplo (ver a seção "EXEMPLO DE APRESENTAÇÃO DE QUESTÃO" para maiores detalhes).
0,80	Apresentou identificador pessoal na imagem, mas não no código.	Não apresentou um identificador no código (e.g. o RU como parte do nome de uma variável)
0,70	Apresentou o identificador pessoal no código, mas não nas imagens/prints.	Não apresentou um identificador na imagem.
0,50	Não apresentou identificador pessoal no código e nem nas imagens/prints.	Questão sem nenhuma identificação de autoria.
0,00	Apresentou o identificador de outra pessoa nas prints e/ou no código.	A questão veio com identificador pessoal de outra pessoa.

CÓDIGO

A apresentação dos códigos compõe um terço da nota total das questões. Este será avaliado conforme a tabela a seguir:

As possíveis notas para esse critério são apresentadas na tabela a seguir:

Tabela 3: Possíveis notas na apresentação do código

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
100	Código **coerente com a resposta encontrada** e apresentado no formato **imagem**.	Está de acordo com o exemplo (ver a seção "EXEMPLO DE APRESENTAÇÃO DE QUESTÃO" para maiores detalhes)
70	Código **coerente com a resposta encontrada** e apresentado no formato **texto.**	Acertou o código, mas copiou o texto do código ao invés de tirar *print*
60	Código **parcialmente** correto e apresentado no formato **imagem**.	Errou um pouco código, mas colocou no trabalho no formato imagem
40	Código **parcialmente** correto e apresentado no formato **texto**.	Errou um pouco código e copiou o texto do código ao invés de tirar *print*
0	Sem código ou com código **incorreto**	A questão não apresentou código ou o código estava errado.

OBS. 1: NÃO ESQUECER DO IDENTIFICADOR PESSOAL (Ex.: COLOCAR SEU RU NO NOME DE UMA VARIÁVEL DO PROGRAMA).

IMAGENS/NUVENS DE PALAVRAS

As imagens compõem um terço da nota total de cada questão. Essas, normalmente, são as nuvens de palavras geradas com base em seu código. Cada prática/questão dessa atividade prática virá com instruções de como devem ser essas imagens.

Entende-se que a **legenda faz parte de uma imagem**. Sendo assim, as **legendas serão avaliadas**.

As possíveis notas para esse critério são apresentadas na tabela a seguir:

Tabela 4: Possíveis notas na apresentação das imagens/fotos

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
100	Imagens **corretas** e com legenda **adequada**.	Está de acordo com o exemplo (ver a seção "EXEMPLO DE APRESENTAÇÃO DE QUESTÃO" para maiores detalhes)
90	Imagens **correta,** mas com legenda **superficial.**	Ex. de legenda superficial: "Figura 1: Nuvem de palavras".
80	Imagens **corretas,** mas com legenda **precária.**	Ex. de legenda precária: "Figura 1: Imagem"
70	Imagens **correta,** mas **sem** legenda**.**	Apresentou imagens corretas, mas não colocou legenda.
60	Imagens **parcialmente** corretas, mas com legenda **adequada**.	Imagem que não consiga identificar o que esteja acontecendo ou a falta de uma das imagens se encaixam nesse grupo.
50	Imagens **parcialmente** correta**,** e com legenda **superficial.**	Similar ao segundo item de cima para baixo dessa tabela, mas com pelo menos uma das imagens com problemas.
40	Imagens **parcialmente** corretas**,** e com legenda **precária.**	Similar ao terceiro item de cima para baixo dessa tabela, mas com pelo menos uma das imagens com problemas.
30	Imagens **parcialmente** correta**,** e **sem** legenda**.**	Similar ao quarto item de cima para baixo dessa tabela, mas com pelo menos uma das imagens com problemas.
0	Sem imagens ou com imagens **incorretas**	A questão veio sem imagens ou com imagens erradas

OBS. 1: NÃO ESQUECER DO IDENTIFICADOR PESSOAL (Ex.: Colocar seu RU dentro da nuvem de palavras).

RESPOSTA

A apresentação da resposta correta será avaliada de forma booleana:

As possíveis notas para esse critério são 0 ou 100:

Tabela 5: Possíveis notas na apresentação das respostas

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
100	Resposta **correta**	Está de acordo com o exemplo (ver a seção "EXEMPLO DE APRESENTAÇÃO DE QUESTÃO" para maiores detalhes)
0	Resposta **incorreta**	A questão não apresentou a resposta correta para a pergunta.

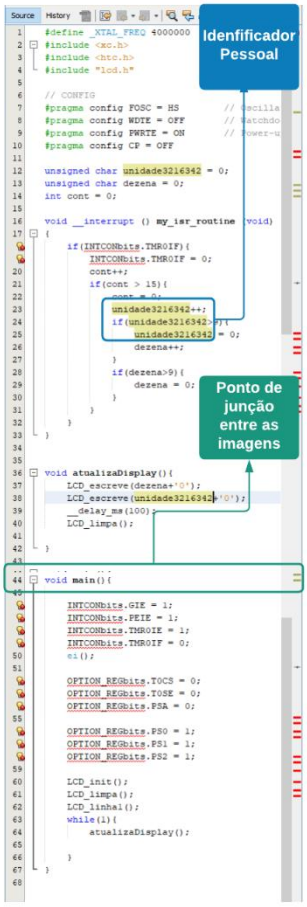
EXEMPLO DE APRESENTAÇÃO DE QUESTÃO

Prática XX – Título da prática

Questão XX – Enunciado curto da questão

Enunciado: Enunciado mais descritivo sobre o que fazer na questão.

I. Apresentação do Código (não esquecer do identificador pessoal):



II. Apresentação das Imagens/Prints (não esquecer do identificador pessoal):

a. Nuvem de palavras das palavras solicitadas:

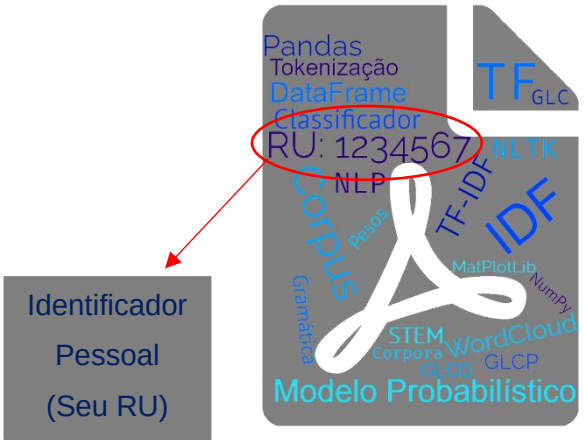


Figura 1: Resultado da nuvem de palavras geradas após tratamento com TF-IDF para sentimentos positivos.

III. Responda à pergunta: Qual a palavra ou termo que mais representou o sentimento?
Resposta: IDF.

OBS1: Nas suas imagens não precisa circular e apontar o identificador pessoal.

OBS2: Perceba que toda a atividade está contida numa só página.

OBS3: Optou-se por juntar as imagens do código. No entanto, não houve perda de clareza e organização.

OBS4: Este é apenas um exemplo e nem o código mostrado, nem a nuvem de palavras e nem a pergunta são referentes à atividade prática que você deverá entregar.

PRÁTICA: CRIAÇÃO DE MODELO DE CLASSIFICAÇÃO SUPERVISIONADO PARA ANÁLISE DE FAKE NEWS

Objetivo: Criar um modelo de classificação de notícias como verdadeiras ou falsas usando o corpus FakeBr.

As práticas desse roteiro utilizam notebooks Python em conjunto com as bibliotecas NLTK, Scikit-Learn, Pandas, NumPy e WordCloud para sua execução mínima, porém você poderá usar outras bibliotecas, caso deseje, porém a NLTK é obrigatória.

MOTIVAÇÃO DO TRABALHO

Com a disseminação de notícias falsas nas redes sociais, a criação de ferramentas automatizadas para verificar a veracidade dessas notícias se tornou essencial. Identificar se uma notícia é verdadeira ou falsa é desafiador até para profissionais, e a criação de um modelo automatizado pode ser uma solução prática e útil.

DESCRIÇÃO DO PROJETO

Você deverá utilizar o corpus FakeBr (https://github.com/roneysco/Fake.br-Corpus), que contém 7200 instâncias (3600 notícias falsas e 3600 verdadeiras), totalizando mais de 28 milhões de palavras. Seu objetivo é criar um modelo de classificação que identifique notícias falsas (FAKE) e verdadeiras (REAL), atingindo uma acurácia mínima de 85%.

Neste corpus existem 28.070.879 palavras no total, sendo 4.046.828 nos textos falsos e 24.024.047 nos textos verdadeiros, sendo assim, o início de seu pré-processamento será importar corretamente os textos para que nenhuma palavra fique com acentuação incorreta e você deverá garantir que em ambos os rótulos (FAKE e REAL) existam as mesmas quantidades de palavras, para que haja balanceamento no corpus e os modelos não sejam afetados por esta grande diferença de palavras. Além destas etapas, você poderá fazer qualquer outro pré-processamento que desejar, como retirar caracteres especiais, retirar símbolos desnecessários, retirar pontuações e outros processamentos que achar necessário.

Feito o pré-processamento você deverá separar seu corpus resultante em duas partes, uma para treinamento, com **75%** de todos os textos (divididos igualmente entre REAL e FAKE) e outro com **25%** de todos os textos, para testes.

Com seu corpus preparado, crie um modelo de classificação que possua **acurácia superior a 85%**, para que seu modelo seja aceito como resposta correta (acurácias superiores a 93% podem significar que você esqueceu de normalizar a quantidade de palavras nos textos).

A análise de acurácia deverá ser apresentada no print de seu código.

Para tomar como base, a acurácia padrão esperada para este tipo de projeto é de 88%, porém com utilização de técnicas mais robustas é possível atingir 97,83% (maior acurácia obtida até o momento entre os alunos).

OBJETIVO DO PROJETO

Nosso objetivo com este dataset é responder às duas questões que se encontram neste documento, criar o relatório da atividade prática, conforme modelo apresentado anteriormente e publicar em “Trabalhos”, em conjunto com de arquivo de python notebook.

De forma resumida, você deverá cumprir os seguintes passos mínimos, para finalizar esta tarefa:

1. AQUISIÇÃO: Fazer o download do dataset no link fornecido.
2. ESTRUTURAÇÃO: Carregar o dataset em um Pandas DataFrame. Certifique-se de que seus dados foram importados corretamente, sem nenhum erro de acentuação. Este dataset em particular pode gerar diversos erros de importação por conta de a língua portuguesa conter diversos símbolos menos comuns em inglês, como vírgulas e acentuações.
3. PRÉ-PROCESSAMENTO: Informações desnecessárias, espaços duplicados e demais artefatos irrelevantes ao projeto deverão ser filtrados, para obtenção do dado corretamente analisado. Outras técnicas poderão ser necessárias nesta etapa, para um resultado mais preciso. Aqui você também deverá separar seus dados em 70%, 75% ou 80% dos textos para treinamento e 30%, 25% ou 20% para testes e análise de acurácia (as proporções entre corpus de treinamento e de testes fica a seu critério e devem seguir as orientações recebidas nas aulas teóricas).
4. MINERAÇÃO: Agrupar os dados de forma a obter estatísticas desejadas das palavras. A sugestão aqui é o uso de TF-IDF em conjunto com bigramas. Utilize a biblioteca Scikit-Learn para a geração do TF-IDF.
5. MODELAGEM: Criação do modelo de classificação que será treinado e depois testado. Este modelo deverá resultar em mais de 85% de acurácia para ser aceito como resposta correta.
6. REPRESENTAÇÃO: Criar uma forma visual de apresentar os tokens (palavras) usados em cada classe (FAKE e REAL). Aqui você deverá usar a biblioteca wordcloud para geração de duas nuvens de palavras, uma para cada rótulo (ou usar a função gerar_nuvem_palavras do arquivo funcoes_auxiliares.py fornecido no repositório github do trabalho).
7. REFINAMENTO: Não conseguiu atingir a acurácia desejada? Agora é o momento de voltar no seu processo e melhorá-lo. Lembre-se que acurácias muito altas também podem ser um sinal de “over training”, que significa que seu modelo foi treinado em excesso e resultará em erros com outros tipos de textos que não pertençam ao corpus original. Para o caso deste dataset, acurácias acima de 93% geralmente sinalizam a utilização dos textos completos e não normalizados, pois os textos marcados como REAL possuem muito mais palavras do que os textos marcados como FAKE, causando uma análise por tamanho e não por contexto.
8. INTERAÇÃO: Rode seu notebook do zero novamente, para garantir que todos os comandos estão em pleno funcionamento. Garanta que suas imagens de nuvens de palavras estão com seu RU e estão apresentando os termos corretamente. Monte o relatório e faça a sua entrega.

PRÁTICA 01- CRIAÇÃO DE MODELO DE CLASSIFICAÇÃO SUPERVISIONADO PARA ANÁLISE DE FAKE NEWS.

Sua missão nesta prática será criar um modelo de classificação de notícias em verdadeiro ou falso. Para que o treinamento de sua inteligência artificial possa ser realizado, você deverá utilizar o corpus FakeBr (podendo ser o arquivo pre-processed.csv), disponível no github em https://github.com/roneysco/Fake.br-Corpus.

Para realizar a missão, você deverá criar um script com a linguagem Python em um Python Notebook com a utilização de, no mínimo, a biblioteca NLTK.

Sugerimos também a utilização das bibliotecas Pandas, Scikit Learn e WordCloud, mas você poderá usar quaisquer outras que achar necessário.

A missão será considerada bem-sucedida caso seu modelo atinja uma acurácia superior à 85% e tenha sido criado seguindo todas as condições impostas anteriormente, como a normalização da quantidade de palavras nos textos verdadeiros com os textos falsos, como a utilização das bibliotecas NLTK para parte do seu pré-processamento e/ou processamento, uso do TF-IDF para geração dos dados estatísticos das palavras e da WordCloud para apresentação das palavras consideradas no seu modelo após o pós-processamento.

QUESTÃO 01: Apresente aqui o código referente ao modelo gerado e a nuvem de palavras que foram usadas para identificar textos VERDADEIROS.

Após a criação de seu modelo, crie uma nuvem de palavras com os termos ou palavras que foram considerados para a classificação dos textos POSITIVOS.

Para isso, você deverá fazer todo o trabalho, encontrar a acurácia desejada e somente então criar a nuvem de palavras com a lista de palavras ou bigramas ou trigramas que foram usados no treinamento com o rótulo REAL.

Um exemplo de nuvem de palavras criado com a biblioteca wordcloud e a partir dos textos marcados como REAL deste mesmo dataset pode ser visto na Figura 2. Nesta figura, as palavras estão representadas com tamanhos relativos à sua frequência simples nos textos e não com a frequência gerada pelo processamento com TF-IDF.

Figura 2: Nuvem de palavras em formato de ícone de mão em formato de positivo, composta por palavras diversas em tons de verde e tamanhos diferentes. As palavras que se destacam são: de, que, da, para, a, o, não, em, um, e, é, foi. Estas palavras se destacam justamente por serem as que mais aparecem nos textos com rótulo igual a REAL. O RU do aluno consta na imagem em verde

Perceba que o exemplo mostrado na Figura 2 não possui muito significado léxico nem de sentido ou sentimento. Isto se dá, pois, esta nuvem foi gerada a partir de todas as palavras do corpus com entradas rotuladas como REAL, sem nenhum tipo de pré-processamento.

Além do print de seu código contendo seu RU (você poderá dividir seu código em duas partes e incluir uma parte na questão 1 e outra na questão 2) e da nuvem de palavras contendo seu RU, você deverá responder à pergunta:

Quantas palavras, bigramas e trigramas foram usados dos textos rotulados como REAL para a criação de seu modelo e qual a acurácia?

OBS1: Não é recomendado utilizar agregações maiores do que trigramas, pois o processamento poderá causar falhas e travar o computador em utilização, porém você é livre para testar.

OBS2: É possível que sua nuvem de palavras contenha termos pouco compreensíveis, por conta do seu pré-processamento. Caso isto ocorra, descreva um ou dois exemplos na legenda da figura sobre sua interpretação do significado dos termos.

OBS3: A nuvem de palavras deverá aparecer com cada palavra/termo com tamanho referente à frequência dada pela metodologia TF-IDF. Ainda, a nuvem poderá possuir o formato que você achar mais relevante para sua apresentação de dados.

QUESTÃO 02: Apresente aqui o código referente ao modelo gerado e a nuvem de palavras que foram usadas para identificar textos FALSOS.

Nesta questão você deverá seguir o mesmo modelo feito na anterior, porém para os textos usados com rótulo FAKE (textos NEGATIVOS).

Na Figura 3 é mostrado um exemplo da nuvem de palavras dos textos rotulados como FAKE do corpus FakeBr. Perceba que aqui as palavras também possuem pouco sentido semântico para identificarmos se são pertencentes à textos reais ou não, pois nenhum pré-processamento foi realizado para a geração desta nuvem de palavras.

Figura 3: Nuvem de palavras em formato de ícone de mão em formato de negativo, composta por palavras diversas em tons de vermelho e tamanhos diferentes. As palavras que se destacam são: de, que, da, para, a, o, não, em, um, e, é, foi. Estas palavras se destacam justamente por serem as que mais aparecem nos textos com rótulo igual a FAKE. O RU do aluno consta na imagem em vermelho.

Nesta questão você deverá incluir seu código de criação do modelo (você poderá dividir seu código em duas partes e incluir uma parte na questão 1 e outra na questão 2), a sua nuvem de palavras com os termos, bigramas e trigramas usados para treinamento do modelo que pertencem aos textos com o rótulo FAKE e responder à seguinte pergunta:

Quantas palavras, bigramas e trigramas foram usados dos textos rotulados como FAKE para a criação de seu modelo, quais foram as técnicas usadas em seu pré-processamento e qual tipo de modelo foi escolhido para este classificador?

OBS: As mesmas observações da questão valem aqui.

RESPOSTAS AS DÚVIDAS MAIS FREQUÊNTES

1. **Onde baixo os softwares criar os scripts das atividades?**
   R: Você poderá utilizar o Google Colab (https://colab.research.google.com/) de forma online ou instalar, caso você tenha o Windows, o Anaconda (https://www.anaconda.com/products/individual) para gerenciamento dos pacotes e ambientes virtuais do Python e uso do Python Notebook ou Jupyter Notebook. Para criação do script, qualquer editor de texto servirá, porém sugiro a utilização do Google Colab, Jupyter, VSCode, Spyder ou do PyCharm para esta tarefa.

2. **Estou terminando o curso, tem como fazer um questionário para atividade prática?**
   R: Não.

3. **Eu não possuo máquina para realizar esta atividade. Como devo proceder?**
   R: Você poderá usar um computador em seu polo. Nossas atividades são pensadas para execução em computador disponível nos polos, porém você pode optar por usar sua própria máquina.

4. **Além de minha máquina ou a do polo, tenho outra opção?**
   R: Você poderá usar o google colab para finalizar a tarefa.

5. **Posso usar o arquivo CSV (pre-processed.csv) contido no github do corpus para fazer este trabalho?**
   R: Sim.

6. **Professor, decidi não usar o arquivo pre-processed.csv e fiz o download dos arquivos do corpus original e ele veio dividido em um monte de arquivos pequenos. O que eu faço?**
   R: Neste caso você não está usando o arquivo pre-processed.csv e sim os arquivos originais do corpus. Neste caso alguns passos bem definidos devem ser executados. Vou descrevê-los aqui:

   Passo 1: Adquirir o banco de dados do GITHUB: https://github.com/roneysco/Fake.br-Corpus

Passo 2: Identificar todas as colunas que existirão em seu dataframe ou banco de dados (sugiro usar o pandas dataframe). Esta informação pode ser encontrada no próprio github dos dados, no arquivo README.md.

Para facilitar, os dados dos arquivos meta estão em linhas e cada linha possui a sequência de dados abaixo:

author
link
category
date of publication
number of tokens
number of words without punctuation
number of types
number of links inside the news
number of words in upper case
number of verbs
number of subjuntive and imperative verbs
number of nouns
number of adjectives
number of adverbs
number of modal verbs (mainly auxiliary verbs)
number of singular first and second personal pronouns
number of plural first personal pronouns
number of pronouns
pausality
number of characters
average sentence length
average word length
percentage of news with speeling errors
emotiveness
diversity

Passo 3: Unir todos os dados das pastas full_texts/fake, full_texts/fake-meta-information, full_texts/true e full_texts/true-meta-information em um único dataframe (supondo que você optou por usar dataframes pandas para isso).

Nesta etapa, você já iniciará o seu pré-processamento de dados.

O seu primeiro trabalho de pré-processamento é criar o dataframe com todos os textos e metadados de todas as 4 pastas.
Seu dataframe terá pelo menos as seguintes colunas:

Id - Vai identificar cada texto pelo número+FAKE ou número+REAL
Tag - Vai identificar a classificação do texto: FAKE ou REAL
full_text - Conterá todo o conteúdo dos textos (um arquivo das pastas true ou fake em cada linha)
demais metadados, um em cada coluna.

Quando terminar de unir todos os arquivos, seu dataframe deve se parecer com este exemplo da primeira linha do dataframe final:

Id	news_text_full	news_text_normalized	author	link	category	date_of_publication	number_of_tokens	number_of_words_without_punctuation	number_of_types	...	numbe
1-REAL	O Podemos decidiu expulsar o deputado federal...	O Podemos decidiu expulsar o deputado federal...	Naira Trindade	http://politica.estadao.com.br/blogs/coluna-do...	politica	13/12/2017	168	148	107	...	7

Minha sugestão é realizar a criação deste dataframe em partes.
Criar primeiro um dataframe para todos os textos (reais e falsos), depois um dataframe com todos os metadados e por fim, se achar interessante, já criar uma coluna com os textos normalizados, como eu mostrei no exemplo.
Unir os dataframes com o pandas.concat([dataframe1, dataframe2, dataframe3], axis=1).

Outra sugestão é, primeiro colocar todos os dados de todos os arquivos em uma lista, para facilitar o código. Só depois de todos os arquivos lidos e adicionados à lista, realizar a criação dos dataframes. Isso agilizará o processo e ocupará menos memória.

Passo 4: Agora que você já possui todos os dados em um dataframe, sugiro salvar o dataframe em formato CSV, para garantir que você já terá tudo pronto, caso tenha problemas e perca algum dado.

Passo 5: Use a biblioteca wordcloud para criar uma nuvem de palavras de testes, com todas as palavras de todos os textos, para você aprender a usar a bibliteca e já deixar pronta uma função de plotagem do gráfico/desenho da nuvem de palavras, quando precisar, pois você vai usar bastante essa nuvem para te ajudar na limpeza dos dados.

Passo 6: Os textos verdadeiros e falsos possuem tamanhos muito diferentes. Os falsos são quase sempre curtos e os verdadeiros são mais longos, o que pode causar um erro de classificação dos modelos que você for testar. Para evitar que seus textos acabem sendo classificados por conta de tamanho e não de conteúdo, faça a normalização dos tamanhos (truncar os textos maiores para que todos os textos tenham mais ou menos a mesma quantidade de palavras).

Passos que poderão se repetir várias vezes nos testes e treinamentos:

Passo 7: Limpe seus dados retirando palavras com pouco sentido semântico, deixando tudo em letras minúsculas, retirando acentuação, realizando steam e demais processos que você achar pertinente. Use a nuvem de palavras para visualizar seus dados e tentar entender o que está sendo feito e para onde seria melhor prosseguir, como retirar palavras específicas, símbolos específicos ou outras partes. O artigo feito pela equipe do NILC, que criou o corpus FAKE.br pode dar uma luz em como começar.
Link do projeto: https://nilc-fakenews.herokuapp.com/about
Link do artigo: https://sites.icmc.usp.br/taspardo/PROPOR2018-MonteiroEtAl.pdf
Perceba que no artigo ele já apresenta diversas técnicas e suas acurácias. Verifique no roteiro da atividade quais os parâmetros mínimos necessários para a entrega do trabalho.

Passo 8: Separe seu corpus (todos os textos truncados) em 4 partes: 70% (sugestão de porcentagem mais usada atualmente, mas você pode escolher usar outra se desejar) dos textos verdadeiros, 70% (sugestão de porcentagem mais usada atualmente, mas você pode escolher usar outra se desejar) dos textos falsos, 30% (sugestão de porcentagem mais usada atualmente, mas você pode escolher usar outra se desejar) dos textos verdadeiros, 30% (sugestão de porcentagem mais usada atualmente, mas você pode escolher usar outra se desejar) dos textos falsos (conforme feito nas aulas práticas). Sugiro a utilização da biblioteca sklearn, módulo train_test_split para automatizar e randomizar esta separação:

from sklearn.model_selection import train_test_split

Com isso você terá seus dados para treinamento já classificado (70% dos textos verdadeiros e 70% dos falsos) e poderá usá-los para iniciar os treinamentos de seus algoritmos. Para testar a eficiência dos seus modelos, utilize os outros 30% dos textos que foram separados.

Passo 9: Treine seu modelo e com os 70% dos textos e verifique a acurácia com os 30% dos textos separados para este fim. Não treine seu modelo com 100% dos textos, pois isso causará erros no modelo que só serão percebidos com textos de fora do seu corpus.

Passo 10: Se atingiu mais de 85% de acurácia, finalize o trabalho e entregue. Se não, teste outro modelo (retorne para o passo 7).

Sendo assim, desejo a todos um ótimo aprendizado e nos vemos na tutoria.
Atenciosamente.
Professor Guilherme D Patriota.