TESTE 1 -

1. Exercício 1

Suponha que você possui uma base de dados rotulada com 10 classes não balanceadas, essa base é formada por 40 features de metadados e mais 3 de dados textuais abertos.

Para todos os itens: Informe as bibliotecas usadas, se necessário, o motivo de cada decisão, explore as possibilidades.

* 1. Descreva como faria a modelagem dessas classes.

Inicialmente, usaria algoritmos para fazer seleção das features relevantes. Usar de input todas as 40 features de metadados poderia deixar a performance do modelo ruim. Para isso, provavelmente usaria ferramentas de seleção de features supervisionadas (já que há 10 classes), como árvores de decisão e métodos estatísticos. Boas funções para isso estão na biblioteca SKLearn, como a SelectKBest() ou RandomForestClassifier(). Lembrando que é difícil determinar com certeza a melhor abordagem sem saber se os dados são discretos, contínuos, numéricos, etc.   
A exploração dos dados textuais depende muito de como eles se organizam. Caso seja poucas palavras e com pouca variablidade, é possível transformá-las em categorias/features. É possível também utilzar algoritmos de repetição de palavras como Naive Bayes. Caso sejam textos longos, provavelmente usaria algum tipo de redução de dimensionalidade, utilizando Tokenizers do huggingface para vetorizar o texto, e depois algum embbeding para redução de tamanho.   
Após isso, é possível usar modelos supervisionados de classificação, seja modelos mais simples como RandomForests e SVMs, até modelos mais complexos, como redes neurais. Depende muito da complexidade dos dados.   
Em relação a datasets desbalanceados, várias técnicas podem ser utilizadas, desde as mais simples como undersampling e oversampling, até as mais complexas, como utilização de data augmentation em variáveis possíveis. Novamente, depende dos dados e da característica do problema.

* 1. Ao finalizar essa modelagem, como iria apresentar essa modelagem para a área contratante?

A partir da métrica com dados de testes nunca vistos antes no modelo, podemos apresentar para a área contratante, utilizando visualização de dados e dashboards para o mesmo (matplotlib, pyplot, PowerBI/LookerStudio). Uma apresentação clara do problema e dos ganhos práticos do modelo na rotina são os mais importantes para a gestão, e não tanto a técnica utilizada.

* 1. Como faria a validação desse modelo?

Utilizaria métricas com base em uma base de dados de teste. Se forem dados que são coletados todos os dias, uma separação para treino/validação/teste a partir do tempo que não tenha grandes viéses (por exemplo, dados coletados na pandemia em algumas empresas podem ter viéses grandes) é uma possibilidade. Outra possibilidade é utilizar algoritmos para separar os dados sem levar em consideração o tempo, provavelmente utilizando algum seletor para deixar a mesma proporção de classes entre input e output.

* 1. Supondo que esses dados são recebidos diariamente, como iria trabalhar com esse desafio?

Acredito que na resposta acima já tenha uma noção, utilizaria amostragem por tempo para fazer train/test split do dataset. Além disso, faria um deploy em batch ou real time, dependendo do problema. A utilização de Data Monitors em dashboards como PowerBI e LookerStudio também são muito úteis, para visualização em tempo real da performance do modelo e de coleta de possíveis drift nos dados de input e output.

* 1. Como levaria esse projeto para um ambiente produtivo?

Me adaptaria aos padrões da companhia. Já trabalhei muito com o uso de fluxos em pipeline de dados como o Airflow e acredito que seja uma boa escolha para trabalhar com batch contínuo de dados. Se o servidor for on-premise, é possível fazer o Airflow chamar scripts nas máquinas do servidor, baixar os dados, rodar e enviar os arquivos para a Cloud utilizada. Se o servidor for cloud, é possível usar o airflow para subir VMs que tenham conexão com a base de dados na Cloud, realizem a inferência dos modelos, e sejam fechadas logo em seguida, economizando custo de utilização.

EXTRA - Existe mais algo que gostaria de relatar sobre esse caso?

1. Exercício 2:

Suponha que você tenha uma base de dados de vendas de uma loja de varejo que inclui informações sobre produtos, clientes, datas de compra e valores das vendas. A base de dados possui, em média, 10.000 registros diários.

Para todos os itens: Informe as bibliotecas usadas, se necessário, o motivo de cada decisão, explore as possibilidades.

* 1. Como você iria explorar os dados para obter insights sobre o desempenho das vendas.

Inicialmente exploraria o número de dados nulos na base e daria uma olhada nos dados e possíveis incongruências e duplicações.  
O básico já pode trazer grandes insights, então dar uma olhada nos principais produtos vendidos e maiores receitas já podem dar uma noção ao negócio do que está funcionando ou não.   
Depois, análises como faixas etárias e informações extras dos clientes da base podem trazer insight do real público-alvo do produto, que possui mais interesse e acaba gastando mais.  
As datas das compras e os valores dos produtos que podem mudar ao longo do tempo também podem trazer insight da correlação entre preço e demanda, sabendo até onde a margem pode ser aumentada sem perder muita receita, a ponto do produto trazer o maior lucro possível.

SQL e utilização de dashboards como LookerStudio/PowerBI com as informações tendo updates diários em queries scheduladas para a gestão tomar decisões acredito que traria os maiores ganhos para o negócio.

* 1. Como você responderia as seguintes questões:
  2. Qual é o desempenho de vendas ao longo do tempo?

Criaria dashboards interativos, plotando gráficos de linha temporais, indicando a venda total dos produtos ao longo do tempo (tanto de receita quanto de número, além de uma linha para a margem), provavelmente em meses ou semanas, dependendo da demanda da empresa. A interação permitiria escolhar o range de datas e cada um dos produtos para análise.

* 1. Quais são os produtos mais vendidos?

Criaria outra aba apenas para os piores e melhores desempenhos, para encontrar outliers de vendas e algumas informações sobre os mesmos, para poder extrair insights para o negócio. A base do dashboard poderia ser atualizada todo dia via query schedulada.

* 1. Como as vendas variam por categoria de produtos?

Criar uma aba no dashboard apenas para análise das categorias dos produtos e seus desempenhos. Provavelmente permitindo tanto a análise global, podendo ver quais categorias desempenham melhor, quanto análises específicas, selecionando as categorias e visualizando os melhores desempenhos.

* 1. Qual é a distribuição dos valores de venda?

A distribuição pode ser calculada utilizando plot de histograma com base nos valores de venda. Caso seja uma variabilidade muito grandes de valores que tornem o gráfico com muitas barras, é possível traçar faixas de preços para cada barra.

* 1. Como os preços dos produtos afetam as vendas?

Esse tipo de cálculo pode ser feito utilizando correlações como regressões logísticas ou lineares. Para isso é possível usar a biblioteca Scikit-Learn e o pandas para utilizar os dados baixados das tabelas como dataframes.

* 1. Qual é o perfil dos principais clientes em termos de compras?

É possível utilizar análises básicas como faixas etárias mais frequentes, sexo e outras características dos perfis dos clientes. Para análise mais complexas, é possível utilizar o cálculo de RFM e clusterizações.

* 1. Como você faria para identificar grupos de clientes nessa base de dados?

Depende muito do tipo de colunas que há para as informações dos clientes, mas geralmente informações como localização, renda, faixa etária e sexo já podem dar uma noção breve de grupos de clientes e qual o público alvo mais adequado.

* 1. Qual teste estatístico você usaria para provar uma hipótese referente aos segmentos de clientes? e como iria aplicá-lo?

Se houver muitas informações valiosas dos clientes, é possível traçar perfis agrupando, por exemplo, tipos de clientes para cada categoria de produto, ou então utilizar modelos estatísticos para extrair features que correlacionam os clientes que mais trazem receita para a companhia em um determinado período de tempo (1 ano, por exemplo).

Extra - Pensando nos dados acima, seria possível fazer mais algum tipo de análise?

Acredito que análise que consiga traçar o maior ganho de margem a partir do preço seja o ideal, e isso pode ser estimado estatisticamente com base na variação dos preços ao longo do tempo e a receita do produto, excluindo, claro, vieses como o ano de 2020 com COVID. Dessa forma, é possível traçar um preço que não perca muito share, mantenha os clientes, mas que também não sejam muito descontados e aumentem a margem da companhia.

1. Exercício 3

Suponha que você tenha uma base de dados contendo textos jurídicos, como decisões judiciais, petições e documentos legais. A base de dados inclui informações sobre o conteúdo do texto, data, jurisdição e outras informações relevantes. Seu objetivo é criar um sistema de recomendação que sugira textos jurídicos semelhantes a um texto de referência.

Para todos os itens: Informe as bibliotecas usadas, se necessário, o motivo de cada decisão, explore as possibilidades.

* 1. Descreva como você desenvolveria o sistema de recomendação que recebe um texto de referência e sugere os textos mais semelhantes a ele na base de dados.

Diversas abordagens são possíveis. Nesse problema, o principal problema é dar match em textos que são parecidos.

Algumas abordagens mais simples utilizando algoritmos que transformam os textos em vetores, buscando as palavras mais frequentes do texto (como TF-IDF) ou então transformando as palavras em vetores, fazendo uma redução de dimensionalidade e usando funções de similaridade (distância angular) podem ser opções mais rápidas.

Entretanto, essas abordagens podem ter problemas (como pode-se observar no TESTE 3, em que utilizei TF-IDF), similaridade entre palavras não necessariamente levam em conta contexto, semântica, subjetividade e ordem das palavras. Para soluções mais complexas poderia utilizar embbedings de redes encoders como a BERT e utilizar o vetor dos textos para calcular essa similaridade, ou então, com abordagens mais recentes que explorei no TESTE 3, utilizar GenAIs para fazer a categorização, usando bibliotecas como a LangChain e utilizando agentes.

As bibliotecas aqui possíveis seriam scikit-learn, transformers (huggingface), pytorch e langchain.

* 1. Como você avaliaria esse sistema de recomendação?

Como a distância angular por si só já é uma métrica que reproduz a alta similaridade de texto, seria necessário uma avalição de usuário e com feedbacks para saber se o sistema está realizando boas recomendações. O mesmo vale para abordagens com GenAI ou Encoders de Transformers. Caso isso não fosse coerente, poderia olhar os casos e tentar entender os outliers e criar regras/contornos para isso.

1. Suponha que novos textos jurídicos sejam adicionados diariamente. Como você manteria o sistema de recomendação atualizado e garantiria que ele continue a fornece recomendações relevantes?

Seria interessante passar o TF-IDF em cima das bases atualizadas a cada certo período para que os cálculos de similaridade fossem coerentes conforme a base aumentasse a distribuição dos dados.  
No caso de uso de LLMs, a atualização de modelos mais recentes e poderosos deveriam tornar o trabalho melhor.

TESTE 2 –

1. Como funciona o teste de hipóteses e qual é a sua finalidade na análise estatística?

O teste de hipótese é um método estatístico tradicional, comum e muito poderoso. O teste de hipótese é muito utilizado para extrair informações verdadeiras a partir de dados extraídos do mundo real (amostras). Por exemplo, a hipótese em um artigo científico de que determinado medicamento funciona para determinada doença, pode ser testada por uma amostra de indivíduos que recebem o medicamento e outro de controle (placebo). O teste de hipótese que irá determinar a partir de um certo coeficiente (p) que irá determinar o nível de probabilidade da hipótese postulada ser rejeitada (ex: remédio funciona). Caso os dados demonstrem a partir da daus curvas de distribuição dos dados em que há uma diferença estatística relevante, utilizando o coeficiente de p como margem, é possível determinar se um medicamento funciona ou não. Por ser uma técnica de amostragem, sempre haverá riscos da hipótese nula (ex: remédio não funciona) existir, mas quanto maior a evidência em maior amostragem, menor pode ser o p, podendo satisfazer quase com certeza a hipótese postulada.

1. O que são redes generativas adversárias (GANs) e quais são os possíveis usos dessas redes?

GANs são redes neurais que servem para gerar informações. Muito utilizada em geração de imagens no passado (hoje praticamente substituídas por modelos de difusão), as redes generativas adversárias possuem um conceito bem interessante em relação a função de perda.

GANs funcionam com duas redes neurais, onde a função de perda (loss function) das duas possui uma correlação. De forma resumida e usando exemplo com imagens, a GAN possui uma rede neural que gera a imagem (gerador), recebendo de input uma entrada qualquer (geralmente um matriz com ruído aleatório, também chamado de vetor de espaço latente) e dando como saída uma imagem também (gerada). Uma segunda rede neural (classificador) determina se a imagem criada é falsa ou não, recebendo como label todas as imagens da base de dados real como a classe verdadeira, e todas as imagens geradas pela rede generativa como falsas. A loss function de ambas é correlacionada, a ponto de que o classificador tenta sempre acertar (minimizar o erro) e a generativa quer sempre fazer o classificador errar (maximizar o erro). É considerado um treino satisfatório quando o classificador não consegue determinar se a imagem gerada pela rede generativa é verdadeira ou não. Depois do processo de treinamento, apenas a rede generativa é usada, e consegue gerar imagens a partir do dataset muito satisfatórias. O exemplo dado é para o conceito básico de GANs, que assim como os de difusão, foram introduzidas sem a correlação com inputs de texto, apenas gerando imagens parecidas com o dataset (por exemplo, gerar imagens de rostos, como no site <https://thispersondoesnotexist.com/>)

1. O que são modelos de linguagem? Qual a diferença entre LLMs e modelos de linguagem tradicionais?

Modelos de linguagem são modelos de IA que manipulam linguagens naturais, podendo gerar textos e extrair informações dos mesmos. LLMs significa Large Language Models e basicamente foi introduzido após o grande impacto do uso de Transformers em modelos de linguagens tradicionais. Inicialmente os modelos de redes neurais utilizados eram RNNs, que são redes neurais recorrentes que passam pelo texto atualizando os pesos, palavra após palavra e possuiam um grande problema de memória curta em textos um pouco longos, mas funcionavam razoavelmente bem em alguns problemas de classificação. As LSTMs tentaram resolver o problema de memória das RNNs passando pra frente informações de palavras anteriores para as funções dos nodes da rede, mas ainda sofriam de memória em textos muito longos. Com isso, surgiu o Transformer (final de 2017), uma técnica de multi head attention que criava um mapa de atenção entre todas as palavras e recebia de input todo o texto. Com isso os Transformers resolveram dois problemas das redes anteriores: não possuia problema de memória pois todos os tokens de entrada estavam sendo correlacionados com todas as palavras do texto de input e podia ser paralelizada durante o treino, não necessitando passar a rede por cada palavra para treino/inferência. Entretanto, Transformers possuem um limite de hardware em relação a entrada, já que o processamento de diversos tokens de entradas que correlacionam todos entre si diversas vezes para cada head attention em diversas camadas, e com isso tornaram-se redes muito pesadas em termos de processamento, principalmente quando o objetivo era ser mais genérico com o maior número de dados possíveis, como as LLMs generativas de chat. Dessa formar entrou esse conceito de LLMs, modelos de linguagem grandes.

1. Suponha que você tenha um conjunto de dados com três ou mais grupos para comparar e deseja determinar se há diferenças significativas entre eles. Descreva como você escolheria entre o teste ou outras técnicas estatísticas

Depende muito do problema, se há covariâncias diferentes ou não, se a distribuição dos dados é normal ou não. Dessa forma, primeiramente seria necessário traçar essa distrbuição com um notebook EDA, e assim tentar escolher as técnicas mais apropriadas. É possível utilizar testes estatísticos como o ANOVA para teste de hipótese com 3 ou mais classes, o ANCOVA para covariâncias diferentes. Caso não haja distribuição normal nos dados pode ser utilizado Kruskal-Wallis. Ainda existem outras técnicas, como regressões e friendman, mas necessitaria olhar primeiro o problema e a distribuição dos dados.

1. Qual é a importância do pré-processamento de texto em tarefas de NLP? Quais são as etapas comuns no pré-processamento de texto?

O pré-processamento de texto pode ser crucial para diversas tarefas de NLP. Para inteligências simples como uso de regras regex, o pré-processamento do texto pode mudar totalmente a facilidade em criar regras, diminuindo certa variabilidade. Para modelos de linguagens, o pré-processamento também pode facilitar muito o tamanho do vocabulário da rede, e portanto, a variabilidade do dataset, diminuindo o tamanho do problema e provavelmente aumentando a eficiência. Algumas etapas comuns de pré-processamento são a transformação para um único case (UPPERCASE para letras maiúsculas ou lowercase para minúsculas), diminuindo a variabilidade de caracteres, já que o significado das palavras se mantés, a remoção de stopwords (essa com um pouco mais de cuidado, já que pode mudar o significado de textos para LLMs, mas ajuda muito em algumas regras regex), e a remoção de caracteres especiais.

1. Descreva o processo de vetorização de texto e como modelos de linguagem como o Word2Vec ou o TF-IDF podem ser usados para representar palavras e documentos.

O processo de vetorização de texto consiste em transformar palavras, caracteres ou subwords (mais utilizado hoje em dia) em vetores numéricos. Isso pode ser feito de diversas formas, como utilizar redes neurais já treinadas e extraindo um vetor de alguma camada da rede, ou utilizando algumas técnicas como Word2Vec. O conceito do Word2Vec é justamente correlacionar palavras em certos clusters dentro do espaço multidimensional de um vetor, a ponto que palavras com conceitos e utilizadas em situações parecidas fiquem mais próximas. Dessa forma, para um computador o processamento desse tipo de informação é muito mais útil do que uma sequência de caracteres, por exemplo. Para o encontro de similaridade de palavras, a distância angular entre elas após o processamento de um Word2Vec pode dar uma noção rápida.

1. O que é a análise de sentimento em NLP e quais são os principais métodos para realizar essa tarefa? Como você avaliaria a eficácia de um modelo de análise de sentimento?

Análise de sentimento é uma técnica para classificar o sentimento humano com determinados textos. Exemplos famosos são de dizer se reviews de um produto qualquer é positivo ou negativo analisando textos em redes sociais (twitter, por exemplo). Portanto, a análise de sentimento costuma ser feita com um algoritmo classificador, desde regras regex sabendo palavras que estão mais correlacionados com certos sentimentos, até modelos estado-da-arte como um encoder de algum BERT e derivados para classificação. Importante notar que é sempre uma escolha de custo de processamento e ganho de métricas de acurácia, regex são super rápidos, mas não entenderiam textos irônicos/sarcásticos, por exemplo, o que um modelo mais complexo poderia entender a partir dos datasets de treino.   
Para avaliar a métrica desse modelo, cálculos de classificadores são eficazes, como acurácia, matrizes de confusão, F1-score (muito útil para datasets desbalanceados), etc.

1. Qual é a diferença entre a classificação de texto e o agrupamento (clustering) de texto em NLP? Em que situações cada um é mais apropriado?

A grande diferença está no tipo de técnica adotado. Enquanto classificadores são métodos supervisionados, necessitando de labels para treinamento, os de agrupamento são métodos não supervisionados. Métodos supervisionados são muito úteis para uma maior acurácia e para atender problemas específicos, enquanto métodos de clusterização podem ser úteis para agrupar certo grupos de textos principalmente quando a anotação necessária é muito difícil, mas pode ter um grande viés dependendo do algoritmo adotado, já que sem a label, o algoritmo poderá agrupar de maneiras distintas à requerida.

1. Explique o conceito de reconhecimento de entidades nomeadas (NER) em NLP e suas aplicações práticas.

O conceito de NER é a extração de informações específicas dentro do texto. Acredito que pode ser correlacionado com segmentações/detecção de objetos em imagens. Ou seja, a partir de um dado de input completo (texto inteiro ou imagem completa), não se é feita só uma classificação do todo, mas sim extraído informações de certas parte do dado para o problema requerido (ex: encontrar localizações em texto, ou encontrar carros numa imagem). Portanto, esse tipo de técnica é extremamente útil, já que extra informações valiosas de dados brutos, podendo aumentar muito eficiência em base de dados ou ser utilizado em produção para tarefas específicas dentro da empresa.

1. Como você lidaria com problemas de desequilíbrio de classe em tarefas de classificação de texto em NLP? Quais estratégias seriam eficazes?

O desequilíbrio de classes é muito comum no dia-a-dia de um cientista de dados. No projeto de NLP que fiz parte isso era rotineiro, pois trabalhei em projetos que extraiam informações de doenças como câncer em cima de todos os laudos de vários laboratórios da empresa. Em média, os achados que procurávamos era em cerca de 1 a 2% do total de laudos lidos. Dessa forma, para treinamento de LLMs era necessário fazer um undersampling das classes negativas (sem câncer, por ex), e utilizar as classes positivas na maior quantidade possível, para que o modelo tivesse uma boa ideia do que era câncer ou não. Para isso, utilizávamos regex para buscar certas palavras chaves que indicavam a classe positiva, e mais tarde estávamos usando GenAI (mais especificamente o MedPalm da Google) para classificar os casos positivos em cima de uma base grande. Com isso, extraíamos não uma amostra aleatória, mas sim uma amostra enviesada para o modelo poder ter uma noção de classe positiva dentro do dataset. Obviamente, a amostra aleatória era necessária para validação mais tarde. É importante também para datasets desbalanceados calcular métricas específicas das classes faltantes como precisão, e utilizar métricas genéricas como o F1-score.

TESTE 3 - CASE

Amostra: 

Contextualização:

O Base de dados canada\_amostra em formato CSV representa um conjunto de empresas do Canadá com a respectiva descrição de seus produtos, dados econômicos e localização.

Assim, podemos caracterizar cada variável:

name: nome da empresa;

description: descrição do produto da empresa;

employees: número de empregados da empresa;

total\_funding: Total de investimento já recebido pela empresa;

city: cidade;

subcountry: estado;

lat: latitude da cidade;

lng: Longitude da cidade.

1. Problema:

Deseja-se prospectar empresas que possuam soluções em \*\*tratamento de água\*\* , principalmente, elativas à : \*\*solutions on waste and water, Improve water quality and water efficiency use, water contamination, water for human consumption, water resources\*\* .

* 1. EXERCÍCIO 1 - Aplique um algoritmo de ML (ou um conjunto deles) capaz de selecionar as principais empresas indicadas para desenvolver a solução de acordo com seu alinhamento com o tema (Justifique a escolha do algoritmo).

Escolhi como resolução mais rápida e simples a vetorização usando o TF-IDF, e calculando a similaridade por distância angular entre os vetores dos textos. É importante ressaltar no entanto que essa abordagem acabou encontrando palavras como water, e não necessariamente era tratamento, ou então a palavra treatment e não necessariamente era com água (como por exemplo, tratamento de pessoas). Para limpar um pouco removi stopwords e caracteres especiais, retirei os cases. Outra abordagem foi utilizar uma média de similaridade entre todas as strings sugeridas no problema, e não só uma, dessa forma garanti que tratamento de pacientes não entrassem como similar, entretanto o threshold precisou ser baixo.

Para dar uma noção de ataque mais preciso (e também mais custoso computacionalmente) utilizei a biblioteca LangChain chamando uma API da OpenAI para classificar os textos, criando um agente que realizava essa ação.

Outras técnicas possíveis ainda seriam usar o Word2Vec para transformar as palavras em vetores, e um redutor de dimensionalidade para deixar tudo em mesma dimensão independente do tamanho do texto (como um PCA), ou então utilizar diretamente um Embedding de encoder de qualquer transformer e comparar os vetores também por distância angular.

* 1. EXERCÍCIO 2 - Faça uma análise exploratória dos resultados acrescentando as demais variáveis contidas no dataset. Quais insights você pode obter a partir desses dados? Quais são as principais cidades (pólos de desenvolvimento) para essa solução?

Fiz um notebook de EDA que está no repositório tentando usar o máximo das informações disponíveis no dataset. Fiz plot de mapas com as principais regiões, as maiores empresas e as principais cidades.

* 1. EXERCÍCIO 3 - EXTRA - Se você terminou o desafio de forma rápida, temos mais algumas perguntas para serem respondidas. Elas, como dito, não são obrigatórias, então sinta-se à vontade em não as responder ou até mesmo respondê-las parcialmente. Essa parte visa observar seu entendimento de um ambiente real de produção.

Não finalizei tão rápido. O dataset .xlsm estava bem sujo, com a tabulação errada e precisei primeiro transpor para csv da maneira correta, não sendo algo muito comum ao baixar de tabelas de banco de dados em cloud. Além disso, o LangChain teve alguns problemas de importações por ser um framework muito recente, então tem alguns problemas de bugs ainda e muita coisa tornando-se deprecated rápido.

Tenho experiência com documentação usando o MkDocs (usando docstrings) e também em páginas de confluence. Preferi deixar esse teste em jupyter notebooks para não ser necessário rodar tudo que fiz. Sempre utilizei Docker em ambientes de produção. Trabalhei pouco com testes automatizados, mas conheço algumas ferramentas, como o pytest. Subi o github.

* + 1. a) organize seus códigos em pacotes garantindo seu versionamento e documentação (bibliotecas auxiliares, etc.).
    2. b) construa testes automatizados para validação do seu pacote.
    3. c) crie uma imagem Docker capaz de executar suas análises em um ambiente de produção.
    4. d) crie um GitHub público e suba todo o código do Teste 3, e disponibilize para avaliação.