ANÁLISE DE TEMPO DE TELA

Projeto de análise de tempo de tela usando python e suas bibliotecas (Pandas e Matplotlib) e SQL

1. INTRODUÇÃO

Como meu primeiro projeto de análise de dados, busco explorar e responder certas perguntas que ajudam na construção do relatório, apresentando resultados através dos dados analisados com Pandas e SQL, as ferramentas de análise de dados em que eu tenho mais experiência atualmente. Portanto, com base nos dados disponibilizados no site Kaggle, foi feito a extração do arquivo em CSV que contém a tabela que com os dados necessários para a análise, de acordo com o contexto dessa tabela.

Fonte dos dados:

https://www.kaggle.com/datasets/anandshaw2001/mobile-apps-screentime-analysis

2. OBJETIVO

A partir do auxílio da ferramenta Chat GPT, foi possível a criação de algumas perguntas para que a análise possa ser feita e disponibilizar uma visão sobre as informações que os dados entregam, e para que isso possa ser possível, foram utilizados nesse projeto ferramentas como o Jupyter Notebook para que os códigos em python possam ser executados, assim como suas bibliotecas (Pandas, Matplotlib e Seaborn), e o PostgreSQL, como sistema de gerenciamento de banco de dados usado para extrair os dados do arquivo CSV e executar as consultas em SQL. Desse modo, o objetivo desse projeto é responder as perguntas e gerar informações para a análise a partir dos dados fornecidos.

3. ESTRUTURA DOS DADOS

A tabela utilizada para a análise se chama “screentime_analysis”, e contém as seguintes colunas:

. **Date**: Data da coleta dos dados.

. **App**: Nomes dos aplicativos.

. **Usage_minutes**: Tempo de uso do aplicativo em minutos

. **Times_opened**: Quantidade de vezes que o aplicativo foi aberto

. **Notifications**: Quantidade de notificações recebidas no aplicativo no dia.

4. METODOLOGIA

A análise realizada foi feita com base nas 3 etapas que serão mostradas a seguir:

4.1. Limpeza e verificação dos dados

A limpeza de dados é um processo de identificar, corrigir e remover erros e dados incoerentes que estão na base de dados, como dados nulos, valores ausentes ou dados duplicados, para que a análise possa ser feita com dados relevantes e de forma precisa. Além disso, é preciso converter certos tipos de dados para que a análise possa ser feita do jeito correto.

4.2. Análise exploratória de dados (EDA)

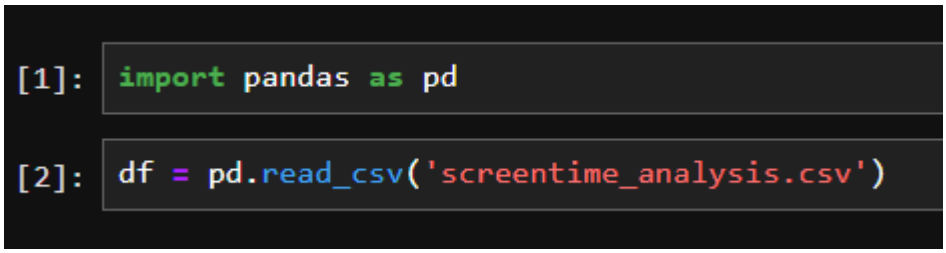
A análise exploratória de dados é feita para entender as características dos dados de forma estatística, o que permite descobrir padrões, tendências, relações entre variáveis e possíveis anomalias, tudo isso para que possa ser feito a extração de informações de maneira eficiente para responder as perguntas que interessam, formular hipóteses e suposições na análise. Além disso, exemplos de operações que podem ser feitas para esse tipo de análise, são cálculos estatísticos (como média, desvio padrão e distribuições), análise de

correlação entre variáveis, filtragem de dados e visualização de dados para facilitar a interpretação

4.3. Extração de Insights e conclusões

Após a análise exploratória de dados, em que são extraídas as informações necessárias para responder as perguntas, é preciso analisar todos esses resultados e transformar em um conhecimento útil, identificando padrões, tirando as conclusões para resolver o problema proposto e auxiliar na tomada de decisões.

Porém, antes dessa etapa, é preciso fazer a importação da biblioteca pandas para fazer as operações e ler o arquivo CSV:



Logo após a importação e a leitura da base de dados, serão feitas as operações para verificar as 5 primeiras linhas e as 5 últimas linhas, para uma verificação rápida e inicial do dataframe, além das suas informações:

```
[3]: df.head()
```

[3]:

	Date	App	Usage (minutes)	Notifications	Times Opened
0	2024-08-07	Instagram	81	24	57
1	2024-08-08	Instagram	90	30	53
2	2024-08-26	Instagram	112	33	17
3	2024-08-22	Instagram	82	11	38
4	2024-08-12	Instagram	59	47	16

```
[7]: df.tail()
```

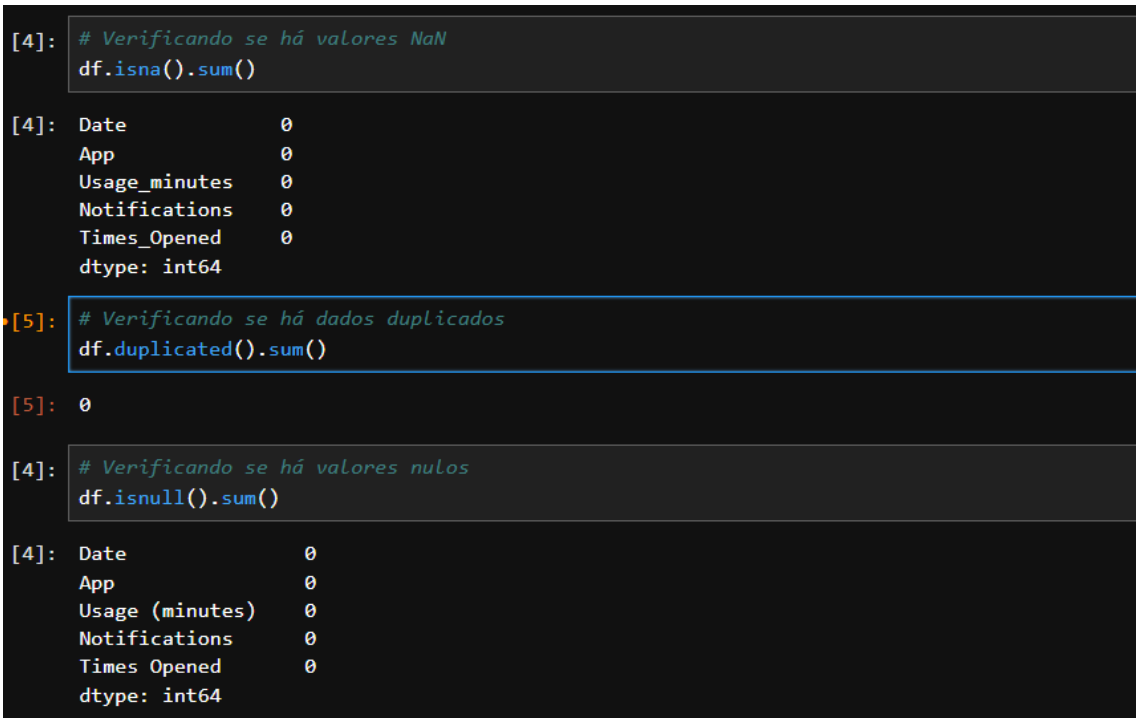
[7]:

	Date	App	Usage_minutes	Notifications	Times_Opened
195	2024-08-10	LinkedIn	22	12	5
196	2024-08-23	LinkedIn	5	7	1
197	2024-08-18	LinkedIn	19	2	5
198	2024-08-26	LinkedIn	21	14	1
199	2024-08-02	LinkedIn	13	4	1

```
[3]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            200 non-null    object
 1   App             200 non-null    object
 2   Usage_minutes   200 non-null    int64
 3   Notifications   200 non-null    int64
 4   Times_Opened    200 non-null    int64
dtypes: int64(3), object(2)
memory usage: 7.9+ KB
```

Após essa etapa, todas as operações foram feitas com sucesso, sem nenhum erro no dataframe, e a partir das informações, é possível concluir que esse dataframe possui duas colunas do tipo object, que é um conjunto de caracteres, que no caso são as colunas “Date” e “App”, e também possui três

colunas do tipo inteiro, que são as colunas “Usage_minutes”, “Notifications” e “Times_Opened”. Além disso, serão feitas as operações para verificar se existem dados incoerentes dentro do Dataframe:



Como é possível ver, foram feitas operações para verificar se há valores NaN ou indefinido, dados nulos e dados duplicados, e em todos os resultados, não existem erros ou inconsistências no dataframe, o que quer dizer que não é necessário que haja limpeza de dados.