

FT02
Curso: UFCD 10809
UFCD/Módulo/Temática: UFCD 10793 - Fundamentos de Python
Ação: 10809_1L Visualização de dados em Python
Formador/a: Sandra Liliana Meira de Oliveira
Data:
Nome do Formando/a:

1. Analisar dados – Uber Reviews Without Reviewid

Descrição do DataSet:

O DataFrame contém 12.000 linhas e 10 colunas relacionadas a análises de utilizadores sobre o Uber.

Informação Geral:

- O DataSet tem **12000 registos** e **10 colunas**.
- Algumas colunas apresentam valores ausentes:
 - **userImage:** Não possui valores preenchidos (0 non-null).
 - **replyContent** e **repliedAt:** Apenas 33 valores preenchidos (respostas a comentários).
 - **reviewCreatedVersion** e **appVersion:** Contêm valores ausentes (~1.740 registos sem preenchimento).

Colunas e Tipos de Dados:

1. **userName** (object): Nome dos utilizadores, com valores únicos.
2. **userImage** (float64): Não contém dados. Pode ser descartada.
3. **content** (object): Texto das análises dos utilizadores.
4. **score** (int64): Avaliação numérica (1 a 5).
5. **thumbsUpCount** (int64): Número de "gostos" que a análise recebeu.
6. **reviewCreatedVersion** (object): Versão da aplicação em que a análise foi criada.

7. **at** (object): Data e hora da análise.
8. **replyContent** (object): Resposta fornecida pelo Uber ao utilizador (apenas 33 análises).
9. **repliedAt** (object): Data da resposta (também com 33 valores).
10. **appVersion** (object): Versão da aplicação instalada no dispositivo.

Descrição Estatística:

Colunas Numéricas:

1. **score:**
 - Média: **3.93** (indicando análises globalmente positivas).
 - Mínimo: **1**; Máximo: **5**.
 - 50% das análises (mediana) são **5** (positivas).
2. **thumbsUpCount:**
 - Média: **0.52**, mas há outliers (máximo: **239**).
 - A maioria das análises não recebeu "gostos" (mediana: 0).

Colunas Categóricas e de Texto:

1. **userName:** Cada utilizador é único (12.000 valores únicos).
2. **content:** Texto altamente variado (8.172 valores únicos).
 - O comentário mais comum é "Good", com **985 ocorrências**.
3. **reviewCreatedVersion:** 10.260 análises especificam a versão da aplicação.
 - Versão mais frequente: **4.554.10001**, com **3.187 análises**.
4. **replyContent:** Apenas 33 análises receberam respostas personalizadas.

Colunas Temporais:

- **at:** 11.949 valores únicos (indicando quase todas as análises em momentos distintos).
- **repliedAt:** 33 respostas fornecidas em diferentes datas e horas.

Ações e Análise Potencial:

1. **Limpeza:**
 - **userImage:** Coluna sem dados. Pode ser descartada.

- **replyContent** e **repliedAt**: Apenas 33 valores preenchidos. Analisar sua relevância.
- **reviewCreatedVersion** e **appVersion**: Tratar os valores ausentes (e.g., preencher com "Desconhecida").

2. Exploração de Dados:

- Distribuição das avaliações (**score**): Analisar percentagem de 1 a 5 estrelas.
- Análise temporal: Identificar padrões nas datas das análises.
- Popularidade das análises: Estudo dos "gostos" (**thumbsUpCount**).

3. Correlação:

- Relação entre a avaliação (**score**) e o número de "gostos" (**thumbsUpCount**).
- Relação entre versões da aplicação e o tipo de avaliação.


4. Texto:

- Análise de frequência de palavras ou frases em **content**.
- Identificar sentimentos das análises (positivas, negativas).

1. Análise, tratamento e Limpeza dos dados do Dataset com recurso a pandas

1. Carregamento e Exploração Inicial

python

 Copy code

```
df = pd.read_csv(file_path)
print(df.info())
print(df.head())
```


Porquê?

- Antes de qualquer análise, é fundamental entender a estrutura do dataset: número de linhas, colunas, tipos de dados, e valores ausentes.
- `df.info()` mostra detalhes das colunas (número de entradas não nulas, tipo de dados).
- `df.head()` exibe as primeiras 5 linhas para visualizar os dados reais e identificar possíveis problemas ou padrões.

2. Limpeza de Dados

2.1 Remover a coluna `userImage`

python

 Copy code


```
df_cleaned = df.drop(columns=['userImage'])
```

Porquê?

- A coluna `userImage` não contém valores úteis (todos são nulos). Manter colunas irrelevantes aumenta a complexidade da análise e consome memória desnecessariamente.

2.2 Preencher valores ausentes

python

 Copy code

```
df_cleaned.fillna({'reviewCreatedVersion': 'Desconhecida'}, inplace=True)  
df_cleaned.fillna({'appVersion': 'Desconhecida'}, inplace=True)
```


Porquê?

- Colunas como `reviewCreatedVersion` e `appVersion` têm valores ausentes (~1.740 registos). Preencher com `'Desconhecida'` permite manter a integridade do dataset, evitando erros em análises futuras.
- Preencher valores ausentes é uma prática comum para evitar perda de dados durante operações (e.g., agregações, filtragem).

3. Análise Descritiva

3.1 Distribuição de avaliações (`score`)

python

 Copy code


```
df_cleaned['score'].value_counts()
```

Porquê?

- Saber a frequência de cada pontuação (1 a 5 estrelas) ajuda a entender a perceção geral do serviço.
- Uma maior concentração em pontuações extremas (1 ou 5) pode indicar polarização nas opiniões.

3.2 Resumo de "thumbsUpCount" (gostos)

python

 Copy code


```
df_cleaned['thumbsUpCount'].describe()
```

Porquê?

- O resumo estatístico (`describe()`) fornece informações úteis:
 - **Média:** Indica a quantidade média de "gostos" recebidos.
 - **Máximo:** Identifica outliers (e.g., análises que atraíram muita atenção).
 - **Percentis (25%, 50%, 75%):** Mostram a distribuição geral dos "gostos".

4. Distribuição Temporal das Análises

python

 Copy code


```
df_cleaned['at'] = pd.to_datetime(df_cleaned['at'])
df_cleaned['at'].dt.date.value_counts().sort_index().head()
```

Porquê?

- Converter a coluna `at` para datetime permite análises temporais (e.g., padrões de análise por data).
- Contar o número de análises por data ajuda a identificar picos de atividade (lançamento de atualizações, eventos, ou problemas reportados em massa).

5. Resumo de Respostas

python

 Copy code


```
df_cleaned[['replyContent', 'repliedAt']].dropna().head()
```

Porquê?

- Apenas 33 análises têm respostas do Uber. Isolar estas análises permite entender quando e como a empresa responde.
- Isto pode ser útil para avaliar a proatividade da empresa no atendimento ao cliente.

6. Correlação entre `score` e `thumbsUpCount`

python

 Copy code


```
df_cleaned[['score', 'thumbsUpCount']].corr()
```

Porquê?

- Analisar a correlação entre a pontuação (`score`) e os "gostos" (`thumbsUpCount`) identifica relações entre a qualidade percebida (avaliação) e a interação da comunidade (gostos).
- Uma correlação positiva forte indicaria que análises bem avaliadas tendem a receber mais atenção. Neste caso, a correlação é fraca.

7. Visualização

python

 Copy code


```
df_cleaned['score'].value_counts().plot(kind='bar', title='Distribuição de Avaliações (Sco
```

Porquê?

- Visualizar a distribuição das avaliações em gráfico de barras facilita a interpretação e comunicação de resultados.
- Um gráfico é uma forma clara de identificar tendências, como polarização em avaliações ou dominância de uma única pontuação.

8. Exportação do DataFrame Limpo

python

 Copy code

```
df_cleaned.to_csv('/mnt/data/uber_reviews_cleaned.csv', index=False)
```

Porquê?

- Salvar o DataFrame limpo garante que futuras análises possam ser feitas sem repetir a etapa de limpeza.
- É uma boa prática salvar dados processados para evitar duplicação de esforço e manter consistência entre análises.

Os passos anteriores permitiram:

1. **Compreender a estrutura inicial do dataset.**
2. **Limpar dados irrelevantes ou ausentes para análises confiáveis.**
3. **Extrair padrões e insights a partir das métricas mais relevantes (score, gostos, respostas).**
4. **Fornecer uma base sólida para visualização e interpretações futuras.**

Exercício – Concretiza os passos anteriores no VS Code num ficheiro .py