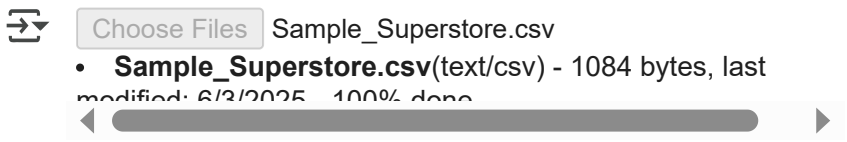
- TensorFlow with GPUs
- TPUs in Colab

Featured examples

- Retraining an Image Classifier: Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
- Text Classification: Classify IMDB movie reviews as either *positive* or *negative*.
- Style Transfer: Use deep learning to transfer style between images.
- Multilingual Universal Sentence Encoder Q&A: Use a machine learning model to answer questions from the SQuAD dataset.
- Video Interpolation: Predict what happened in a video between the first and the last frame.

```
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns

# Upload CSV
from google.colab import files
uploaded = files.upload()
```




```
df = pd.read_csv('Sample_Superstore.csv')
df.head()
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Custom Na
0	CA-2017-152156	2017-11-08	2017-11-11	Second Class	CG-12520	Cla Gu
1	CA-2017-152156	2017-11-08	2017-11-11	Second Class	CG-12520	Cla Gu
2	US-2016-108966	2016-10-11	2016-10-18	Standard Class	DV-13045	Dar Van H
3	US-2016-108966	2016-10-11	2016-10-18	Standard Class	DV-13045	Dar Van H

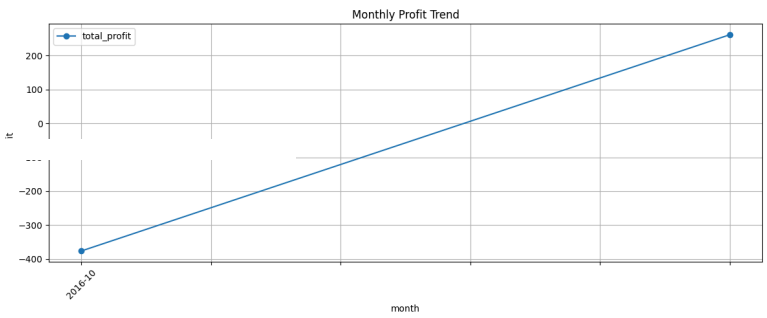
Next steps: Generate code with df View recommended plots

```
# Check for missing values
df.isnull().sum()

# Check column names
df.columns

# Clean column names
df.columns = df.columns.str.strip().str.lower().str.re



df['order_date'] = pd.to_datetime(df['order_date'])



conn = sqlite3.connect(':memory:')
df.to_sql('superstore', conn, index=False, if_exists='
```

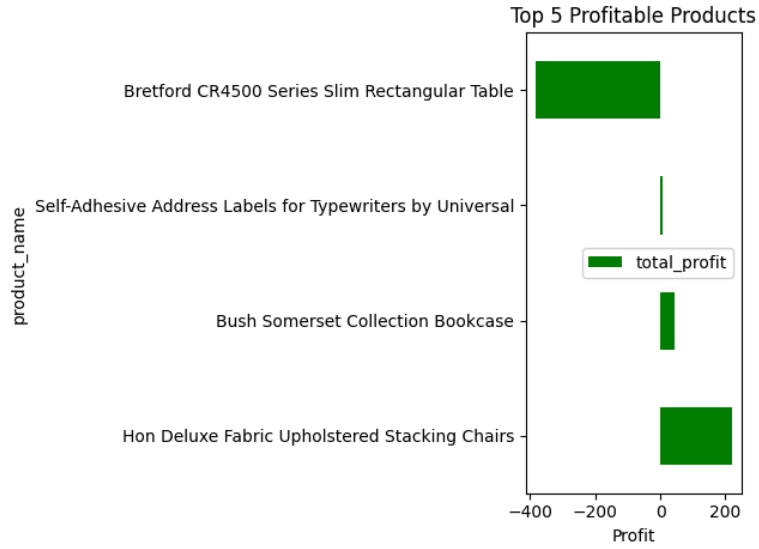
4

```
monthly_profit = pd.read_sql_query("""
SELECT
    strftime('%Y-%m', order_date) as month,
    SUM(profit) as total_profit
```

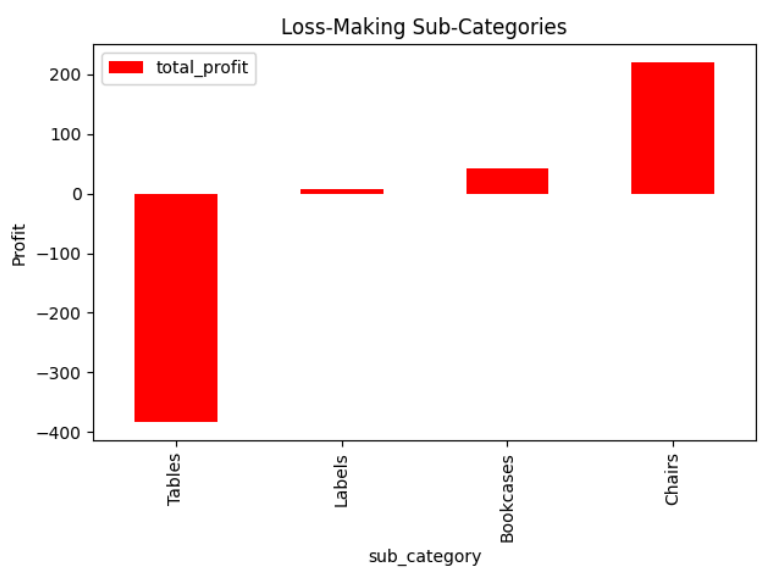
```
FROM superstore
GROUP BY month
ORDER BY month;
""", conn)

monthly_profit.plot(x='month', y='total_profit', kind=
plt.xticks(rotation=45)
plt.ylabel('Profit')
plt.grid(True)
plt.tight_layout()
plt.show()
```




```
top_products = pd.read_sql_query("""
SELECT
    product_name,
    ROUND(SUM(profit), 2) as total_profit
FROM superstore
GROUP BY product_name
ORDER BY total_profit DESC
LIMIT 5;
""", conn)

top_products.plot(x='product_name', y='total_profit',
plt.xlabel('Profit')
plt.tight_layout()
plt.show()
```

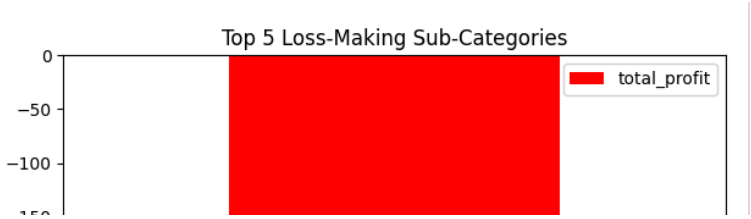



```
loss_subcats = pd.read_sql_query("""
SELECT
    "Sub-Category" as sub_category,
    ROUND(SUM(Profit), 2) as total_profit
FROM superstore
GROUP BY "Sub-Category"
ORDER BY total_profit
LIMIT 5;
""", conn)
loss_subcats.plot(x='sub_category', y='total_profit',
plt.ylabel('Profit')
plt.tight_layout()
plt.show()
```




```
loss_subcats = pd.read_sql_query("""
SELECT
    "Sub-Category" AS sub_category,
    ROUND(SUM(Profit), 2) AS total_profit
FROM superstore
GROUP BY "Sub-Category"
HAVING total_profit < 0
ORDER BY total_profit ASC
LIMIT 5;
""", conn)
```

```
loss_subcats.plot(x='sub_category', y='total_profit',
plt.ylabel('Total Profit')
plt.tight_layout()
plt.show()
```


Top 5 Loss-Making Sub-Categories
0
−50
−100
total_profit