

4)  
Os

```
▶ import pandas as pd
import numpy as np
from datetime import datetime

def generate_synthetic_data(output_path: str = "data/synthetic_ads.csv"):
    np.random.seed(42)

    dates = pd.date_range(end=datetime.today(), periods=30)
    campaigns = ["Brand_US", "Brand_IN", "Remarketing", "New_Users"]
    channels = ["Google", "Meta", "Display"]
    countries = ["US", "IN"]

    rows = []

    for date in dates:
        for campaign in campaigns:
            impressions = np.random.randint(2000, 15000)
            ctr = np.random.uniform(0.01, 0.05)
            clicks = int(impressions * ctr)
            cpc = np.random.uniform(3, 20)
            spend = clicks * cpc
            conv_rate = np.random.uniform(0.02, 0.08)
            conversions = int(clicks * conv_rate)
            revenue_per_conv = np.random.uniform(150, 600)
            revenue = conversions * revenue_per_conv

            rows.append([
                date.date(),
```





V



Automated\_Report (1).pptx

712 KB · Done

RAM  
Disk

```
df = pd.merge(ads, ft, on="date", how="left")

total_spend = df["spend"].sum()
total_revenue = df["revenue"].sum()
total_impressions = df["impressions"].sum()
total_clicks = df["clicks"].sum()
total_conversions = df["conversions"].sum()

summary = {
    "total_spend": float(total_spend),
    "total_revenue": float(total_revenue),
    "total_impressions": int(total_impressions),
    "total_clicks": int(total_clicks),
    "total_conversions": int(total_conversions),
    "roas": float(total_revenue / total_spend) if total_spend > 0 else 0.0,
    "ctr": float(total_clicks / total_impressions) if total_impressions > 0
    "cvr": float(total_conversions / total_clicks) if total_clicks > 0 else
    "avg_foot_traffic": float(df["foot_traffic"].mean()),
    "spend_vs_foot_corr": float(df[["spend"]].corr(df[["foot_traffic"]]))
    if df[["foot_traffic"]].notna().any() else 0.0,
}

top_campaigns = (
    df.groupby("campaign")#[["spend", "revenue", "impressions", "clicks"]]
    .sum()
    .reset_index()
    .sort_values("spend", ascending=False)
    .head(5)
)
```

```
hands    + Code    + Text    ▶ Run all    ▾  
14] 0s  
▶  


```
        clicks,
        round(spend, 2),
        conversions,
        round(revenue, 2),
    ])

df = pd.DataFrame(
    rows,
    columns=[
        "date",
        "campaign",
        "channel",
        "country",
        "impressions",
        "clicks",
        "spend",
        "conversions",
        "revenue",
    ],
)
df.to_csv(output_path, index=False)
print(f"Synthetic data generated at {output_path}")

generate_synthetic_data()
import pandas as pd
import numpy as np

def generate_foot_traffic(path="data/foot_traffic.csv"):
```


```

```
[4]  def generate_foot_traffic(path="data/foot_traffic.csv"):
    ads = pd.read_csv("data/synthetic_ads.csv", parse_dates=[ "date"])
    unique_dates = ads[ "date" ].unique()

    rows = []
    for d in unique_dates:
        foot_traffic = np.random.randint(800, 5000)
        rows.append([d, foot_traffic])

    df = pd.DataFrame(rows, columns=[ "date", "foot_traffic"])
    df.to_csv(path, index=False)
    print(f"Foot traffic data generated at {path}")

generate_foot_traffic()
```

... Synthetic data generated at data/synthetic\_ads.csv  
Foot traffic data generated at data/foot\_traffic.csv

```
[5]  def analyze_data(ads_path: str):
    ads = pd.read_csv(ads_path, parse_dates=[ "date"])
    ft = pd.read_csv("data/foot_traffic.csv", parse_dates=[ "date"])

    # Merge digital + physical data
    df = pd.merge(ads, ft, on="date", how="left")
```

```
prs.save(output_path)
print(f"PPT generated at {output_path}")

build_ppt(insights_text, analytics_payload)
```

... PPT generated at output/Automated\_Report.pptx

▶ from google.colab import files

```
files.download("output/Automated_Report.pptx")
```

...

4]  
Os

```
▶ import pandas as pd
import numpy as np
from datetime import datetime

def generate_synthetic_data(output_path: str = "data/synthetic_ads.csv"):
    np.random.seed(42)

    dates = pd.date_range(end=datetime.today(), periods=30)
    campaigns = ["Brand_US", "Brand_IN", "Remarketing", "New_Users"]
    channels = ["Google", "Meta", "Display"]
    countries = ["US", "IN"]

    rows = []

    for date in dates:
        for campaign in campaigns:
            impressions = np.random.randint(2000, 15000)
            ctr = np.random.uniform(0.01, 0.05)
            clicks = int(impressions * ctr)
            cpc = np.random.uniform(3, 20)
            spend = clicks * cpc
            conv_rate = np.random.uniform(0.02, 0.08)
            conversions = int(clicks * conv_rate)
            revenue_per_conv = np.random.uniform(150, 600)
            revenue = conversions * revenue_per_conv

            rows.append([
                date.date(),
```





V



Automated\_Report (1).pptx

712 KB · Done



RAM

Disk

```
df = pd.merge(ads, ft, on="date", how="left")

total_spend = df["spend"].sum()
total_revenue = df["revenue"].sum()
total_impressions = df["impressions"].sum()
total_clicks = df["clicks"].sum()
total_conversions = df["conversions"].sum()

summary = {
    "total_spend": float(total_spend),
    "total_revenue": float(total_revenue),
    "total_impressions": int(total_impressions),
    "total_clicks": int(total_clicks),
    "total_conversions": int(total_conversions),
    "roas": float(total_revenue / total_spend) if total_spend > 0 else 0.0,
    "ctr": float(total_clicks / total_impressions) if total_impressions > 0
    "cvr": float(total_conversions / total_clicks) if total_clicks > 0 else
    "avg_foot_traffic": float(df["foot_traffic"].mean()),
    "spend_vs_foot_corr": float(df[["spend"]].corr(df[["foot_traffic"]]))
    if df[["foot_traffic"]].notna().any() else 0.0,
}

top_campaigns = (
    df.groupby("campaign")#[["spend", "revenue", "impressions", "clicks"]]
    .sum()
    .reset_index()
    .sort_values("spend", ascending=False)
    .head(5)
)
```

```
14] 0s
  ⏎
  ⏎
    clicks,
    round(spend, 2),
    conversions,
    round(revenue, 2),
  ])
df = pd.DataFrame(
    rows,
    columns=[
        "date",
        "campaign",
        "channel",
        "country",
        "impressions",
        "clicks",
        "spend",
        "conversions",
        "revenue",
    ],
)
df.to_csv(output_path, index=False)
print(f"Synthetic data generated at {output_path}")

generate_synthetic_data()
import pandas as pd
import numpy as np

def generate_foot_traffic(path="data/foot_traffic.csv"):
```

```
[4]  def generate_foot_traffic(path="data/foot_traffic.csv"):
    ads = pd.read_csv("data/synthetic_ads.csv", parse_dates=[ "date"])
    unique_dates = ads[ "date" ].unique()

    rows = []
    for d in unique_dates:
        foot_traffic = np.random.randint(800, 5000)
        rows.append([d, foot_traffic])

    df = pd.DataFrame(rows, columns=[ "date", "foot_traffic"])
    df.to_csv(path, index=False)
    print(f"Foot traffic data generated at {path}")

generate_foot_traffic()
```

... Synthetic data generated at data/synthetic\_ads.csv  
Foot traffic data generated at data/foot\_traffic.csv

```
[5]  def analyze_data(ads_path: str):
    ads = pd.read_csv(ads_path, parse_dates=[ "date"])
    ft = pd.read_csv("data/foot_traffic.csv", parse_dates=[ "date"])

    # Merge digital + physical data
    df = pd.merge(ads, ft, on="date", how="left")
```

```
prs.save(output_path)
print(f"PPT generated at {output_path}")

build_ppt(insights_text, analytics_payload)
```

... PPT generated at output/Automated\_Report.pptx

▶ from google.colab import files

```
files.download("output/Automated_Report.pptx")
```

...

# Weekly Performance Report

Automated Insight Engine –  
GroundTruth Hackathon 2025

# Executive Insights

- Total spend was approximately 331224 with revenue of 572836, delivering a ROAS of 1.73.
- The account generated 990560 impressions and 29255 clicks, with a CTR of 2.95%.
- There were 1498 conversions, with a CVR of 5.12% across all campaigns.
- Spend is distributed across a small number of key campaigns, enabling focused optimization.

# Key Metrics Snapshot

- Total Spend: 331223.58
- Total Revenue: 572836.15
- ROAS: 1.73
- CTR: 2.95%
- CVR: 5.12%
- Impressions: 990560
- Clicks: 29255
- Conversions: 1498