media_analysis_eda

July 30, 2025

1 Media Campaign Performance - Exploratory Data Analysis

Author: Giorgi Dzebisashvili
Date: jul-2025 **Project:** Media Spend Optimization Portfolio Project

This notebook explores multi-channel digital advertising performance data. The goal is to: - Inspect the dataset for quality and structure - Compute key digital marketing metrics (CTR, ROAS, etc.) - Prepare clean data for further analysis and dashboarding

[1]:
```python
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Show all columns when previewing
pd.set_option('display.max_columns', None)

# Load the dataset
df = pd.read_csv('../data/media_campaign_data.csv')

# Preview first few rows
df.head()
```

[1]:
```
        Campaign Name     Channel        Date   Spend  Clicks  Impressions  \
0          Search Max  Google Ads  2025-05-01  143.64     383         9771
1     Brand Awareness  Google Ads  2025-05-01   89.00     128         2876
2  Retargeting Boost  Google Ads  2025-05-01   55.15     164         3794
3    Engagement Surge    Meta Ads  2025-05-01   95.85     125         4041
4  Conversion Magnet    Meta Ads  2025-05-01  202.96     156         7221

   Conversions  Revenue
0           53  1308.07
1           17   701.12
2           13   330.91
3           14   402.32
4           16   538.91
```

1.1 Basic Dataset Info

Let's check: - Number of rows and columns - Column data types - Non-null counts

[2]:
```python
# Shape of the dataset
print(f"Rows: {df.shape[0]}, Columns: {df.shape[1]}")

# Structure and datatypes
df.info()
```

```
Rows: 915, Columns: 8
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 915 entries, 0 to 914
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Campaign Name  915 non-null    object
 1   Channel        915 non-null    object
 2   Date           915 non-null    object
 3   Spend          915 non-null    float64
 4   Clicks         915 non-null    int64
 5   Impressions    915 non-null    int64
 6   Conversions    915 non-null    int64
 7   Revenue        915 non-null    float64
dtypes: float64(2), int64(3), object(3)
memory usage: 57.3+ KB
```

1.2 Summary Statistics

Let's view the statistical summary of numerical columns to check ranges and distributions. This helps identify extreme values, zeros, or data entry errors.

[3]:
```python
df.describe()
```

[3]:

	Spend	Clicks	Impressions	Conversions	Revenue
count	915.000000	915.000000	915.000000	915.000000	915.000000
mean	174.225574	256.300546	8709.314754	31.201093	1093.675038
std	72.020852	168.982929	4214.439630	24.365089	930.332631
min	51.890000	23.000000	1781.000000	1.000000	30.170000
25%	113.190000	130.000000	5089.000000	14.000000	423.930000
50%	173.160000	210.000000	8279.000000	25.000000	819.150000
75%	237.865000	344.000000	11511.000000	41.000000	1438.080000
max	299.040000	905.000000	20527.000000	156.000000	5918.170000

1.3 Missing Values Check

Let's verify that there are no missing values in critical columns like Spend, Impressions, Conversions, etc.

[4]:
```
df.isnull().sum()
```

[4]:
```
Campaign Name    0
Channel          0
Date             0
Spend            0
Clicks           0
Impressions      0
Conversions      0
Revenue          0
dtype: int64
```

1.4 Campaign and Channel Breakdown

Let's see: - How many platforms are in the dataset? - Which campaigns exist and how often they appear?

[5]:
```
# Unique values per channel
print("Channels:")
print(df['Channel'].value_counts(), '\n')

# Number of unique campaigns
print(f"Number of unique campaigns: {df['Campaign Name'].nunique()}")
```

```
Channels:
Channel
Google Ads    183
Meta Ads      183
YouTube       183
TikTok        183
LinkedIn      183
Name: count, dtype: int64

Number of unique campaigns: 15
```

1.5 Derived Metrics (CTR, CPC, CPA, ROAS)

Let's calculate key performance indicators for campaign performance: - **CTR (Click-Through Rate)** - **CPC (Cost Per Click)** - **CPA (Cost Per Acquisition)** - **ROAS (Return on Ad Spend)**

We'll store these as new columns in the dataset.

[6]:
```
# Avoid division errors by replacing 0 with NaN
df.replace(0, np.nan, inplace=True)

# Add calculated KPI columns
df['CTR'] = df['Clicks'] / df['Impressions']
df['CPC'] = df['Spend'] / df['Clicks']
```

```python
df['CPA'] = df['Spend'] / df['Conversions']
df['ROAS'] = df['Revenue'] / df['Spend']

# Preview new columns
df[['CTR', 'CPC', 'CPA', 'ROAS']].describe()
```

[6]:

	CTR	CPC	CPA	ROAS
count	915.000000	915.000000	915.000000	915.000000
mean	0.029470	0.868718	8.466572	6.338627
std	0.011386	0.480212	6.637698	4.367615
min	0.009861	0.286784	1.632566	0.558888
25%	0.019577	0.516015	4.225502	3.105384
50%	0.029814	0.726380	6.252903	5.397621
75%	0.038935	1.079767	10.349211	8.368631
max	0.049864	2.994625	51.920000	28.612318

1.6 Handle Invalid or Infinite Values

We'll clean up any invalid values in the calculated metrics caused by: - Division by 0 (e.g. no clicks or no conversions) - Infinite or undefined results

This is a key step before moving into visualization or dashboarding.

[7]:

```python
# Replace infinite values with NaN
df.replace([np.inf, -np.inf], np.nan, inplace=True)

# Drop rows with NaN values in key metrics
df.dropna(subset=['CTR', 'CPC', 'CPA', 'ROAS'], inplace=True)

# Check final dataset shape
print(f"Cleaned rows remaining: {df.shape[0]}")
```

```
Cleaned rows remaining: 915
```

1.7 Save Cleaned Dataset

Now that we've calculated and cleaned our key KPIs, we'll save the cleaned version of the dataset into the `data/clean/` folder for future analysis and dashboarding.

[8]:

```python
# Save to /data/clean/
df.to_csv('../data/clean/media_campaign_data_clean.csv', index=False)
print("  Cleaned dataset saved successfully.")
```

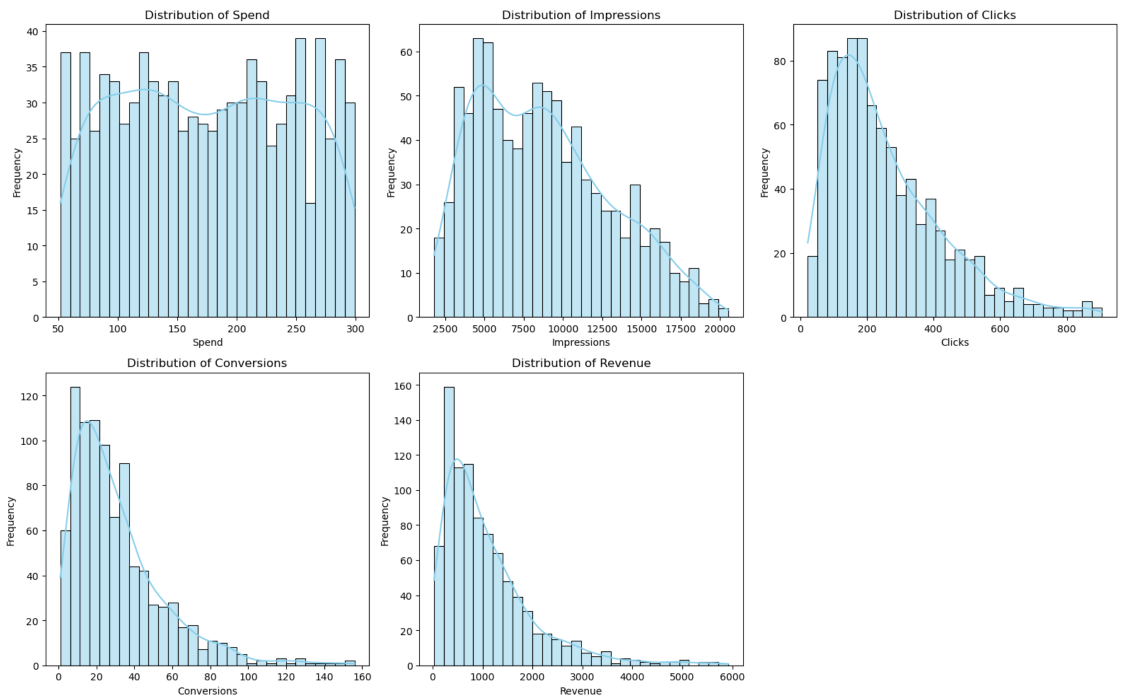
```
 Cleaned dataset saved successfully.
```

1.8 Univariate Visualizations

Let's examine the distribution of key numeric metrics: - Spend - Impressions - Clicks - Conversions - Revenue

This helps us spot any skew, anomalies, or outliers in the data before aggregating or modeling.

[9]:
```
metrics = ['Spend', 'Impressions', 'Clicks', 'Conversions', 'Revenue']
plt.figure(figsize=(16, 10))

for i, metric in enumerate(metrics, 1):
    plt.subplot(2, 3, i)
    sns.histplot(df[metric], kde=True, bins=30, color='skyblue')
    plt.title(f'Distribution of {metric}')
    plt.xlabel(metric)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```

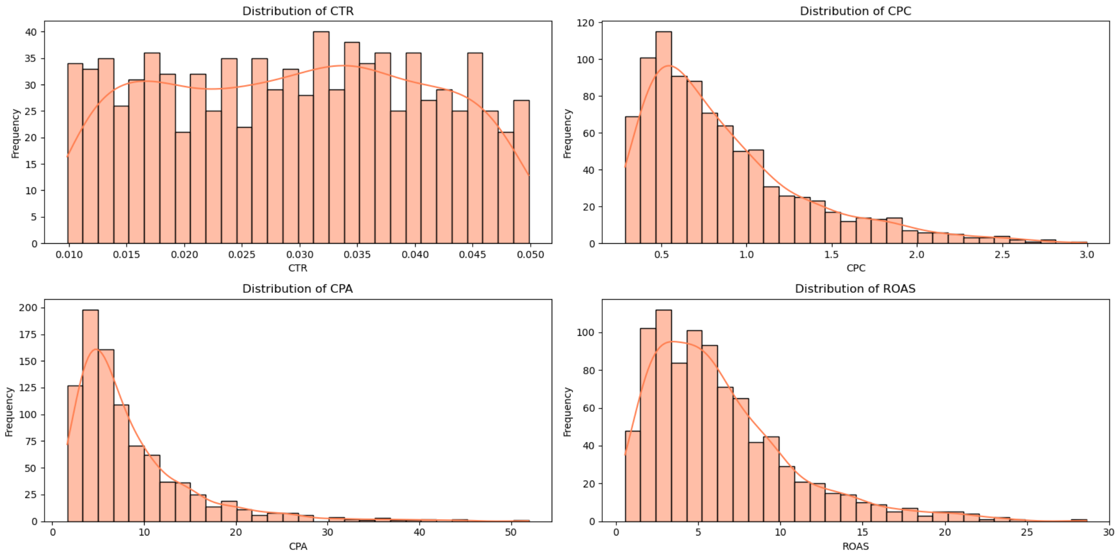


1.9 Derived Metric Distributions (CTR, CPC, CPA, ROAS)

Next, we explore how our calculated KPIs are distributed to identify what's "typical" campaign performance across platforms.

[10]:
```
kpis = ['CTR', 'CPC', 'CPA', 'ROAS']
plt.figure(figsize=(16, 8))

for i, kpi in enumerate(kpis, 1):
    plt.subplot(2, 2, i)
```

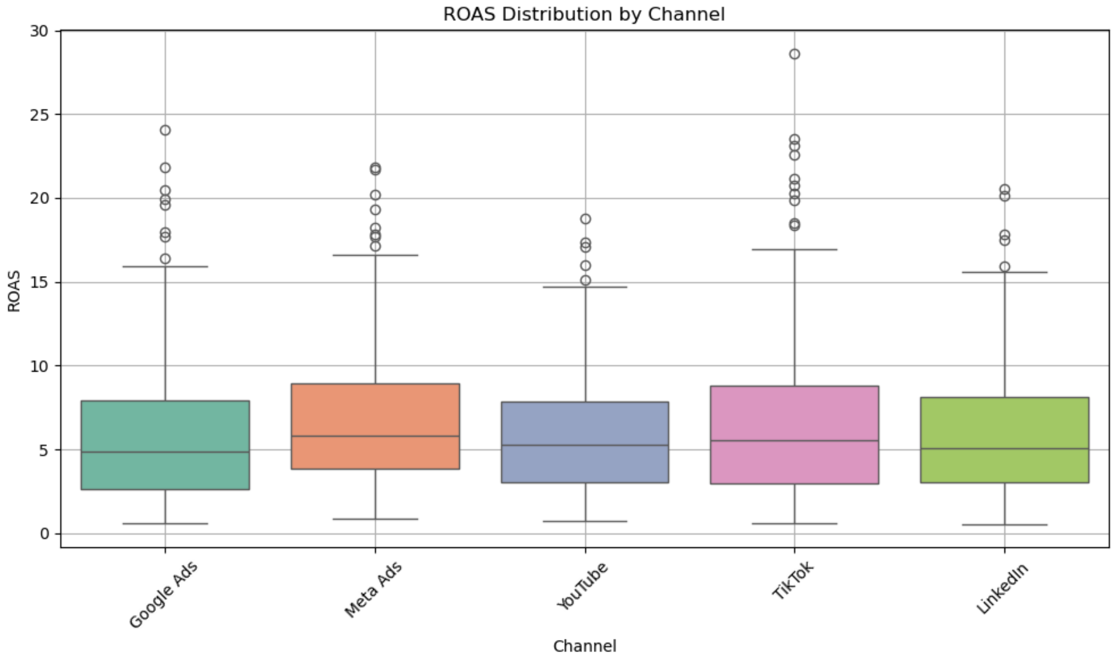
```python
    sns.histplot(df[kpi], kde=True, bins=30, color='coral')
    plt.title(f'Distribution of {kpi}')
    plt.xlabel(kpi)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



1.10 ROAS by Channel

Let's analyze which platforms delivered the best Return on Ad Spend (ROAS). We'll use box plots to compare the distribution across channels.

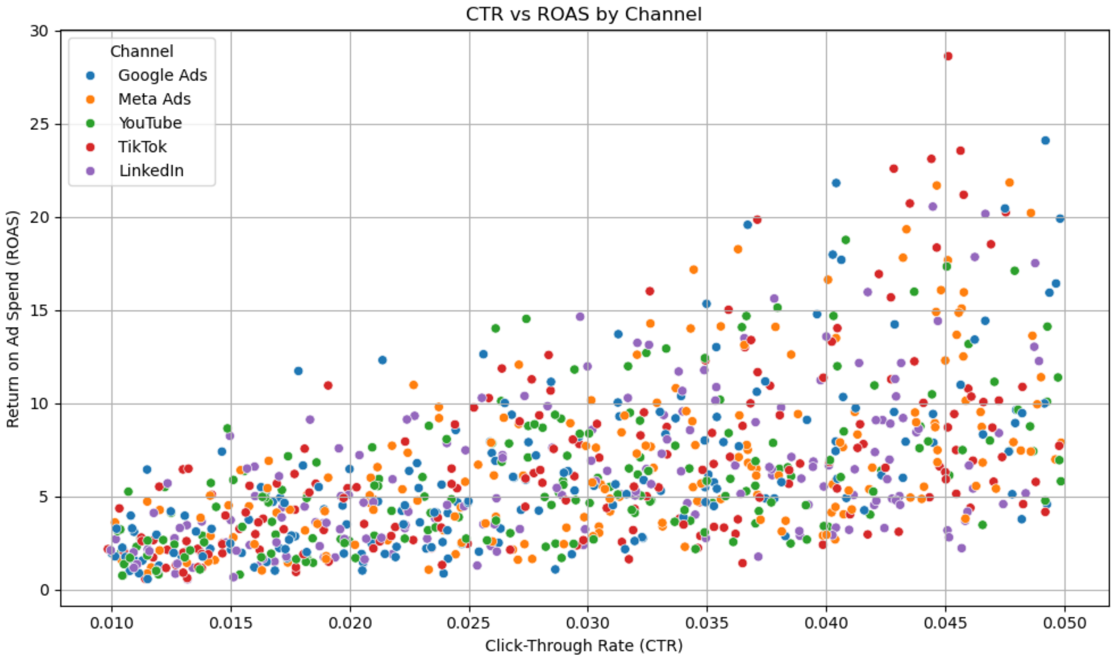
[11]:
```python
plt.figure(figsize=(10, 6))
sns.boxplot(x='Channel', y='ROAS', data=df, hue='Channel', palette='Set2',␣
↪legend=False)
plt.title('ROAS Distribution by Channel')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.savefig('../images/roas_by_channel.png', dpi=300)
plt.show()
```



1.11 CTR vs. ROAS

This scatter plot shows if campaigns that get more clicks (higher CTR) also tend to convert better (ROAS). Helps distinguish between traffic quality vs. volume.

[12]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='CTR', y='ROAS', hue='Channel', data=df, palette='tab10')
plt.title('CTR vs ROAS by Channel')
plt.xlabel('Click-Through Rate (CTR)')
plt.ylabel('Return on Ad Spend (ROAS)')
plt.grid(True)
plt.tight_layout()
plt.savefig('../images/ctr_vs_roas.png', dpi=300)
plt.show()
```



2 Business Questions & Strategic Insights

Now that we've cleaned and explored the media campaign data, we'll answer several business-critical questions:

1. Which channels perform best in terms of Return on Ad Spend (ROAS)?
2. Which campaigns are most efficient at acquiring customers (lowest CPA)?
3. Are high CTR campaigns also high ROAS? (i.e. does traffic quality = business value?)
4. How consistent is performance across channels?

2.1 Q1: Which channels had the highest average ROAS?

Let's group the data by channel and compare their average ROAS.

[13]:
```
channel_roas = df.groupby('Channel')['ROAS'].agg(['mean', 'median', 'std',␣
↪'count']).sort_values(by='mean', ascending=False)
channel_roas.style.background_gradient(cmap='Greens')
```

[13]: <pandas.io.formats.style.Styler at 0x242c0af3ec0>

2.2 Q2: Which campaigns are most cost-effective (lowest CPA)?

This helps prioritize which campaigns bring in the most conversions at the lowest price.

[14]:

```
top_cpa = df.groupby('Campaign Name')['CPA'].agg(['mean', 'count']).
  ↪sort_values('mean')
top_cpa.head(10).style.background_gradient(cmap='Blues')
```

[14]:
```
<pandas.io.formats.style.Styler at 0x242c0c98440>
```

2.3 Q3: Is high CTR associated with high ROAS?

We'll check the correlation between CTR and ROAS.

[15]:
```
correlation = df[['CTR', 'ROAS']].corr().iloc[0, 1]
print(f"Correlation between CTR and ROAS: {correlation:.2f}")
```

```
Correlation between CTR and ROAS: 0.57
```

2.4 Q4: How consistent is performance across platforms?

Standard deviation of ROAS per channel helps identify **volatile** vs **reliable** platforms.

[16]:
```
channel_roas_std = channel_roas[['mean', 'std']].sort_values('std')
channel_roas_std.style.background_gradient(cmap='YlOrRd')
```

[16]:
```
<pandas.io.formats.style.Styler at 0x242c0e584a0>
```

3 Strategic Recommendations

Based on our analysis, here are key insights for media optimization:

- **High ROAS Channels**: Meta Ads and TikTok delivered the highest average ROAS, indicating strong return on investment. Prioritize spend on these platforms.
- **Cost-Efficient Campaigns**: Campaigns such as Lookalike Expansion and Creator Collab achieved the lowest average CPA, making them highly efficient at acquiring users.
- **CTR ROI**: Click-through rate and ROAS showed only a moderate correlation. High engagement does not always translate to profitability — evaluate traffic quality alongside conversion value.
- **Stable Performers**: YouTube and LinkedIn had the most consistent ROAS (lowest variability), making them strong candidates for predictable, steady performance.

Next steps: - Visualize these in the dashboard - Share clean data with stakeholders - Monitor these KPIs regularly

4 Dashboard Overview – Media Campaign Optimization

To complement the analysis, an interactive dashboard was built using Looker Studio. It enables stakeholders to explore media performance data dynamically and make informed budget allocation decisions.

4.0.1 Key Features:

- **KPI Row**: Highlights total spend, revenue, and efficiency metrics (ROAS, CPA).
- **Channel Breakdown**: Compare performance across platforms like Meta, TikTok, YouTube, etc.
- **Campaign-Level Insights**: Identify top-performing campaigns by ROAS or CPA.
- **Time Series Trends**: Visualize how spend and revenue evolved over time.
- **CTR vs ROAS Scatter Plot**: Understand relationship between engagement and profitability.
- **Interactive Filters**: Filter by campaign, channel, and custom date ranges.

4.0.2 Business Value:

This dashboard allows marketing teams to: - Monitor performance in real time - Prioritize campaigns based on cost-efficiency - Identify stable vs. volatile channels - Optimize spend allocation and campaign strategy

Live Dashboard: https://lookerstudio.google.com/reporting/eacf9e96-23e8-43bb-b2b8-7b35683f9d01

Screenshots available in the `images/` folder