Customer Acquisition Cost Analysis using Python

What is Customer Acquisition Cost?

Customer Acquisition Cost (CAC) is a fundamental metric in business, quantifying the financial investment needed to acquire a new customer. It covers expenses related to marketing, advertising, and sales efforts during a specific period, directly impacting a company's profitability. Analysing CAC is essential for evaluating the effectiveness of customer acquisition strategies, enabling informed resource allocation and determining the return on investment for each customer acquired. In the context of data analysis and Python programming, exploring CAC provides an opportunity to leverage technology for in-depth analysis, pattern recognition, and extracting actionable insights, empowering businesses to make strategic decisions for sustainable growth.

What is CAC Analysis?

CAC (Customer Acquisition Cost) Analysis involves evaluating the total expenses incurred by a business to acquire a new customer. It assesses the efficiency of customer acquisition strategies, aids in budget optimization, and informs decisions for sustainable growth and profitability.

Objective: The project aims to calculate and interpret CAC based on relevant data, providing valuable insights into the efficiency of customer acquisition strategies using Python Language.

Libraries:

- pandas
- plotly.express
- plotly.io
- plotly.graph_objects

Functions and Methods Used:

- pd.read_csv()
- print()
- info()
- head()
- px.bar()
- px.scatter()
- groupby()
- describe()
- add_trace()
- update_layout()
- show()

Code:

```
import pandas as pd
import plotly.express as px
import plotly.io as pio
import plotly.graph_objects as go


pio.templates.default = "plotly_white"

data = pd.read_csv("customer_acquisition_cost_dataset.csv")

print(data.head())

data.info()

data['CAC'] = data['Marketing_Spend'] / data['New_Customers']

print(data.head())

fig = px.bar(data, x='Marketing_Channel', y='CAC', title= 'CAC by Marketing
Channel')
fig.show()

fig2 = px.scatter(data, x='New_Customers', y='CAC',
                  color='Marketing_Channel',title= "New Customers vs CAC",
```

```
                  trendline='ols')
fig2.show()


summary_stats = data.groupby('Marketing_Channel')['CAC'].describe()

print(summary_stats)

data['Conversion_Rate'] = data['New_Customers'] / data['Marketing_Spend'] *
100
print(data.head())

data['Break_Even_Customers'] = data['Marketing_Spend'] / data['CAC']

fig3 = px.bar(data, x='Marketing_Channel', y='Break_Even_Customers',
             title='Break-Even Customers by Marketing Channel')

fig3.show()

fig = go.Figure()

#Actual Customers Acquired

fig.add_trace(go.Bar(x= data['Marketing_Channel'], y= data['New_Customers'],
                     name='Actual Customers Acquired',
marker_color='royalblue'))

# Break-Even Customers
fig.add_trace(go.Bar(x=data['Marketing_Channel'],
y=data['Break_Even_Customers'],
                     name='Break-Even Customers', marker_color='lightcoral'))

# Update the layout
fig.update_layout(barmode='group', title='Actual vs. Break-Even Customers by
Marketing Channel',
                  xaxis_title='Marketing Channel', yaxis_title='Number of
Customers')

# Show the chart
fig.show()
```

Output:

	Customer_ID	Marketing_Channel	Marketing_Spend	New_Customers
0	CUST0001	Email Marketing	3489.027844	16
1	CUST0002	Online Ads	1107.865808	33
2	CUST0003	Social Media	2576.081025	44
3	CUST0004	Online Ads	3257.567932	32
4	CUST0005	Email Marketing	1108.408185	13

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Customer_ID       500 non-null    object
 1   Marketing_Channel 500 non-null    object
 2   Marketing_Spend   500 non-null    float64
 3   New_Customers     500 non-null    int64
dtypes: float64(1), int64(1), object(2)
memory usage: 15.8+ KB
```

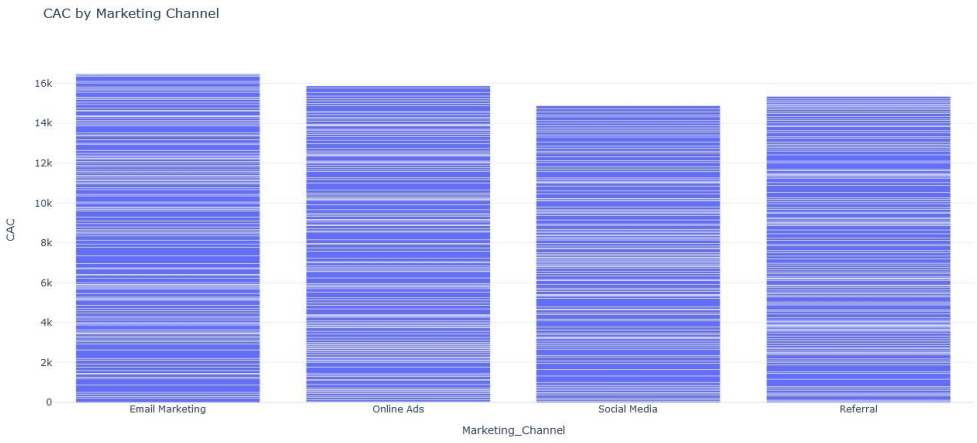
	Customer_ID	Marketing_Channel	Marketing_Spend	New_Customers	CAC
0	CUST0001	Email Marketing	3489.027844	16	218.064240
1	CUST0002	Online Ads	1107.865808	33	33.571691
2	CUST0003	Social Media	2576.081025	44	58.547296
3	CUST0004	Online Ads	3257.567932	32	101.798998
4	CUST0005	Email Marketing	1108.408185	13	85.262168

Marketing_Channel	count	mean	std	min	25%	50%	75%	max
Email Marketing	124.0	132.913758	89.597107	23.491784	68.226195	106.940622	177.441898	434.383446
Online Ads	130.0	122.135938	79.543793	24.784414	62.207753	97.736027	163.469540	386.751285
Referral	128.0	119.892174	74.101916	22.012364	71.347939	99.835688	137.577935	366.525209
Social Media	118.0	126.181913	77.498788	21.616453	75.633389	102.620356	167.354709	435.487346

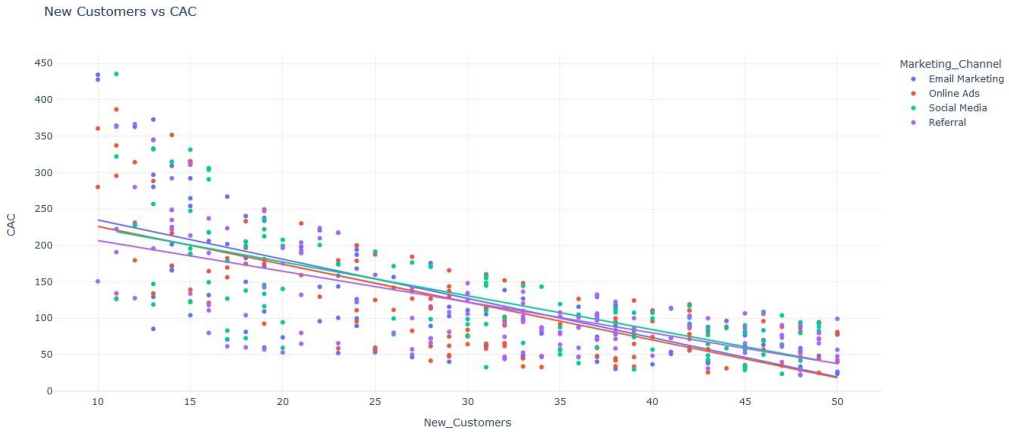
	Customer_ID	Marketing_Channel	Marketing_Spend	New_Customers	CAC	Conversion_Rate
0	CUST0001	Email Marketing	3489.027844	16	218.064240	0.458580
1	CUST0002	Online Ads	1107.865808	33	33.571691	2.978700
2	CUST0003	Social Media	2576.081025	44	58.547296	1.708021
3	CUST0004	Online Ads	3257.567932	32	101.798998	0.982328
4	CUST0005	Email Marketing	1108.408185	13	85.262168	1.172853

Graphs:

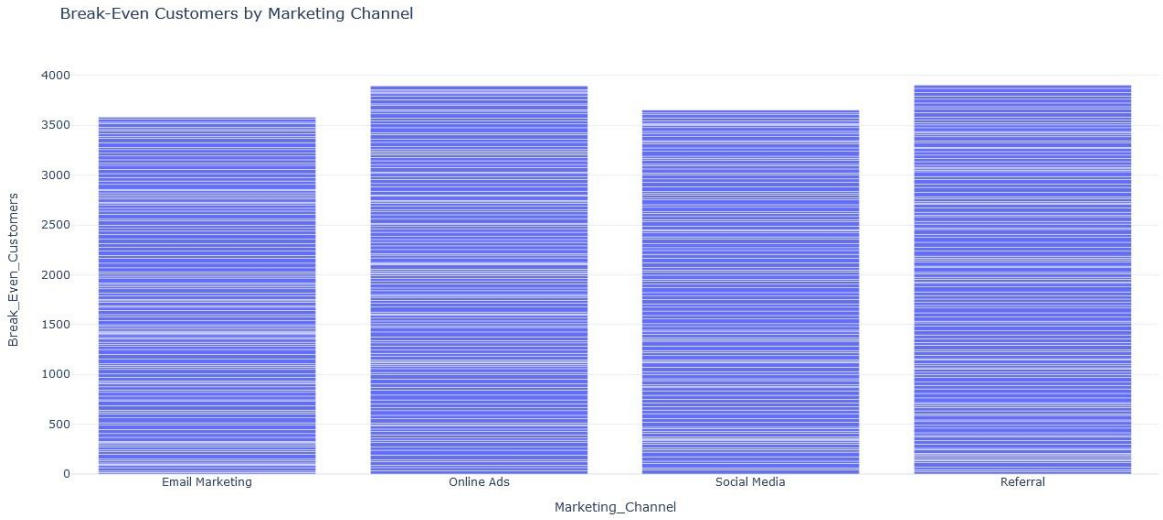
1) CAC by Marketing Channel



2) New Customers vs CAC



3) Break-Even Customers By Marketing Channel



4) Actual vs Break-Even Customers by Marketing Channel