Name : Tanish Dogra  
 Reg Number : 12113391

Task 3

To : ALLO HEALTH

Report on : Google Ads & Facebook ADs

Date of submission: 27th May 2024

Task 3

industry\_benchmarks\_google = {

    'ctr\_min': 0.08,

    'ctr\_max': 0.10,

    'traffic\_to\_lead\_min': 0.07,

    'traffic\_to\_lead\_max': 0.08,

    'lead\_to\_call\_min': 0.25,

    'lead\_to\_call\_max': 0.30

}

industry\_benchmarks\_facebook = {

    'ctr\_min': 0.02,

    'ctr\_max': 0.03,

    'traffic\_to\_lead\_min': 0.12,

    'traffic\_to\_lead\_max': 0.15,

    'lead\_to\_call\_min': 0.12,

    'lead\_to\_call\_max': 0.15

}

revenue\_per\_call = {

    'offline': 2000,

    'online': 1200

}

offline\_consultation\_capacity = 100

After giving benchmark details   
  
We filtered the data according to benchmarks  
def calculate\_performance\_diff\_google(df):

    df['ctr\_diff'] = df['ctr'].apply(lambda x: x - industry\_benchmarks\_google['ctr\_min'] if x < industry\_benchmarks\_google['ctr\_min'] else (x - industry\_benchmarks\_google['ctr\_max'] if x > industry\_benchmarks\_google['ctr\_max'] else 0))

    df['traffic\_to\_lead\_diff'] = df['traffic\_to\_lead'].apply(lambda x: x - industry\_benchmarks\_google['traffic\_to\_lead\_min'] if x < industry\_benchmarks\_google['traffic\_to\_lead\_min'] else (x - industry\_benchmarks\_google['traffic\_to\_lead\_max'] if x > industry\_benchmarks\_google['traffic\_to\_lead\_max'] else 0))

    df['lead\_to\_call\_diff'] = df['lead\_to\_call'].apply(lambda x: x - industry\_benchmarks\_google['lead\_to\_call\_min'] if x < industry\_benchmarks\_google['lead\_to\_call\_min'] else (x - industry\_benchmarks\_google['lead\_to\_call\_max'] if x > industry\_benchmarks\_google['lead\_to\_call\_max'] else 0))

    return df

def calculate\_performance\_diff\_facebook(df):

    df['ctr\_diff'] = df['ctr'].apply(lambda x: x - industry\_benchmarks\_facebook['ctr\_min'] if x < industry\_benchmarks\_facebook['ctr\_min'] else (x - industry\_benchmarks\_facebook['ctr\_max'] if x > industry\_benchmarks\_facebook['ctr\_max'] else 0))

    df['traffic\_to\_lead\_diff'] = df['traffic\_to\_lead'].apply(lambda x: x - industry\_benchmarks\_facebook['traffic\_to\_lead\_min'] if x < industry\_benchmarks\_facebook['traffic\_to\_lead\_min'] else (x - industry\_benchmarks\_facebook['traffic\_to\_lead\_max'] if x > industry\_benchmarks\_facebook['traffic\_to\_lead\_max'] else 0))

    df['lead\_to\_call\_diff'] = df['lead\_to\_call'].apply(lambda x: x - industry\_benchmarks\_facebook['lead\_to\_call\_min'] if x < industry\_benchmarks\_facebook['lead\_to\_call\_min'] else (x - industry\_benchmarks\_facebook['lead\_to\_call\_max'] if x > industry\_benchmarks\_facebook['lead\_to\_call\_max'] else 0))

    return df

And calculated the difference   
  
google\_ads\_campaigns = ['Campaign A', 'Campaign B']

facebook\_ads\_campaigns = ['Retargeting', 'OpenAudience', 'LookalikeAudience']

google\_ads\_df = aligned\_df[aligned\_df['campaign\_name'].isin(google\_ads\_campaigns)]

facebook\_ads\_df = aligned\_df[aligned\_df['campaign\_name'].isin(facebook\_ads\_campaigns)]

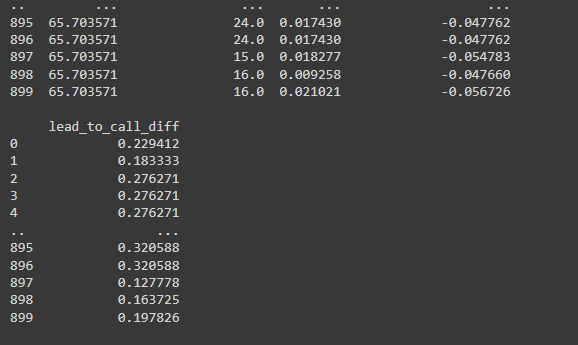
google\_comparison\_df = calculate\_performance\_diff\_google(google\_ads\_df)

facebook\_comparison\_df = calculate\_performance\_diff\_facebook(facebook\_ads\_df)

comparison\_df = pd.concat([google\_comparison\_df, facebook\_comparison\_df], ignore\_index=True)

print(comparison\_df)

output :



Summary

A screenshot of a computer

Description automatically generated

Now we will do visuilzation:

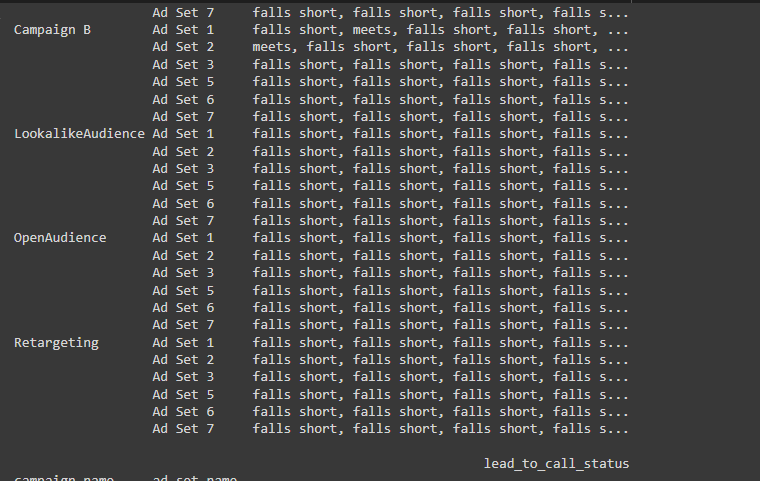
A graph of different colored bars

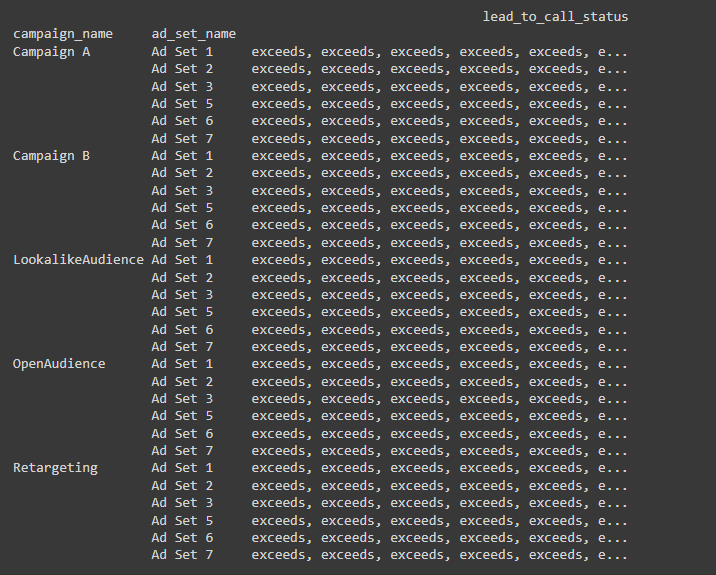
Description automatically generated with medium confidence

A graph showing different colored bars

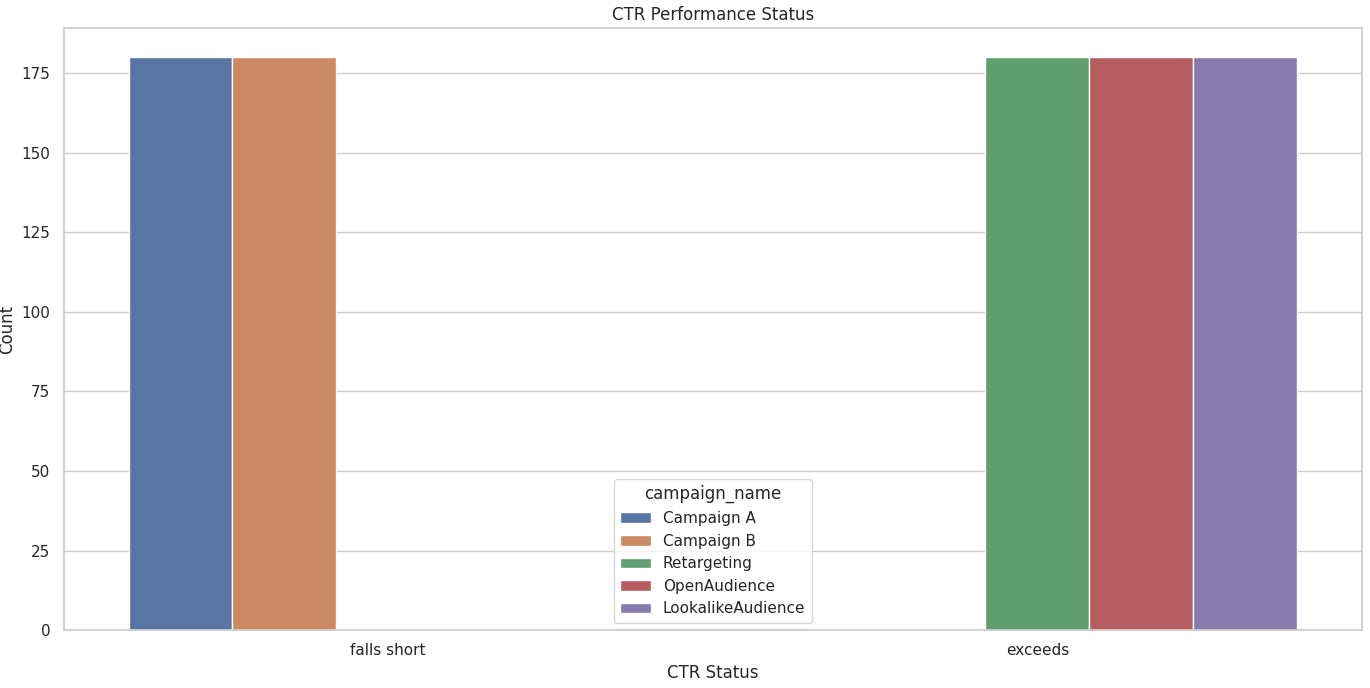
Description automatically generated with medium confidenceA graph of different colored bars

Description automatically generated

2nd   




Performance visuilization



A graph with different colored bars

Description automatically generated

