Python Summer Party Challenge

by Interview Master

Day 2 of 15

Amazon

You are a Product Analyst on the Amazon Sponsored Advertising team investigating sponsored product ad engagement across electronics categories. Your team wants to understand CTR variations to optimize targeted advertising strategies.

Challenge Questions

Q1:

What is the average click-through rate (CTR) for sponsored product ads for each product category that contains the substring 'Electronics' in its name during October 2024? This analysis will help determine which electronics-related categories are performing optimally.

Q2:

Which product categories have a CTR greater than the aggregated overall average CTR for sponsored product ads during October 2024? This analysis will identify high-performing categories for further optimization. For this question, we want to calculate CTR for each ad, then get the average across ads by product category & overall.

Q3:

For the product categories identified in the previous question, what is the percentage difference between their CTR and the overall average CTR for October 2024? This analysis will quantify the performance gap to recommend specific categories for targeted advertising optimization.

Want to try this yourself?

Join the Challenge

Sign up for the Python Summer Party Challenge and solve 21 days of data science problems

www.interviewmaster.ai/python-party

Or keep scrolling to see my solutions

My Solution - Q1

Day 2 Python Challenge

```
# Merge fct_ad_performance with dim_product
merged_df = pd.merge(fct_ad_performance, dim_product, on='prod
uct_id', how='inner')

# Filter merged_df to get october ads only & Electronics produ
ct_category
filtered_df = merged_df[
  merged_df['product_category'].str.contains('Electronics') &
  (merged_df['recorded_date'] >= '2024-10-01') &
  (merged_df['recorded_date'] <= '2024-10-31')
]

# Calculate the average CTR
avg_ctr_by_category = (
    filtered_df.groupby('product_category')
    .apply(lambda x: x['clicks'].sum() / x['impressions'].sum
())
    .reset_index(name='avg_ctr')
)
print(avg_ctr_by_category)
```

InterviewMaster.AI

My Solution - Q2

Day 2 Python Challenge

```
# Merge fct_ad_performance with dim_product
merged_df = pd.merge(fct_ad_performance, dim_product, on='prod
uct_id', how='inner')

# Filter merged_df to get October ads only
october_ads = merged_df[
  (merged_df['recorded_date'] >= '2024-10-01') &
  (merged_df['recorded_date'] <= '2024-10-31')
]

# Calculate the CTR for each ad and get the overall average ct
r
october_ads['ad_ctr'] = october_ads['clicks'] / october_ads['i
mpressions']
overall_avg_ctr = october_ads['ad_ctr'].mean()

# Calculate the CTR for each product category
ctr_by_category = october_ads.groupby('product_category')['ad_
ctr'].mean().reset_index(name='avg_ctr')

# High performing categories
high_performing = ctr_by_category[ctr_by_category['avg_ctr'] >
overall_avg_ctr]
print(high_performing)
```

My Solution - Q3

Day 2 Python Challenge

```
# Merge fct_ad_performance with dim_product
merged_df = pd.merge(fct_ad_performance, dim_product, on='prod
uct_id', how='inner')

# Filter merged_df to get October ads only
october_ads = merged_df[
  (merged_df['recorded_date'] >= '2024-10-01') &
  (merged_df['recorded_date'] <= '2024-10-31')
]

# Calculate the CTR for each ad and get the overall average ct
r
october_ads['ad_ctr'] = october_ads['clicks'] / october_ads['i
mpressions']
overall_avg_ctr = october_ads['ad_ctr'].mean()

# Calculate the CTR for each product category
ctr_by_category = october_ads.groupby('product_category')['ad_
ctr'].mean().reset_index(name='avg_ctr')

# High performing categories
high_performing = ctr_by_category[ctr_by_category['avg_ctr'] >
overall_avg_ctr]

# Calculate the percentage difference
high_performing['percentage_diff'] = (high_performing['avg_ct
r'] - overall_avg_ctr) * 100 / overall_avg_ctr
print(high_performing)
```

Ready for your own challenge?

Try this yourself by signing up for the Python Summer Party challenge:

www.interviewmaster.ai/python-party

InterviewMaster.AI