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.

My Solution - Q1

Day 2 Python Challenge

```
# Note: pandas and numpy are already imported as pd and np
# The following tables are loaded as pandas DataFrames with th
e same names: fct_ad_performance, dim_product
# Please print your final result or dataframe

joined_df = pd.merge(fct_ad_performance, dim_product, on = 'pr
oduct_id', how = 'inner')

october_electronics = joined_df[
  (joined_df['recorded_date'] >= '2024-10-01') &
  (joined_df['recorded_date'] < '2024-11-01') &
  (joined_df['product_category'].str.contains('Electronics'))
]

ctr_avg = (october_electronics.groupby('product_category').app
ly(lambda x : x['clicks'].sum()/x['impressions'].sum()).reset_
index(name='avg_ctr'))
print(ctr_avg)
```

InterviewMaster.AI

My Solution - Q2

Day 2 Python Challenge

```
merged_data = pd.merge(fct_ad_performance, dim_product, on='pr
oduct_id', how='inner')
oct_electronics = merged_data[
  (merged_data['recorded_date'] >= '2024-10-01')&
  (merged_data['recorded_date'] < '2024-11-01')
  ]

oct_electronics['ad_ctr'] = oct_electronics['clicks']/oct_elec
tronics['impressions']
overall_ctr = oct_electronics['ad_ctr'].mean()
print(overall_ctr)
ctr_category = oct_electronics.groupby('product_category')['ad
_ctr'].mean().reset_index(name='avg_ctr')
print(ctr_category)
high_performance = ctr_category[ctr_category['avg_ctr']>overal
l_ctr]
print(high_performance)
```

InterviewMaster.AI

My Solution - Q3

Day 2 Python Challenge

```
merged_data = pd.merge(fct_ad_performance, dim_product,
                       on='product_id', how = 'inner')

oct_data = merged_data[
  (merged_data['recorded_date']>='2024-10-01') &
  (merged_data['recorded_date']<'2024-11-01')]

oct_data['ad_ctr'] = oct_data['clicks']/oct_data['impression
s']
overall_ctr = oct_data['ad_ctr'].mean()

ads_ctr = oct_data.groupby('product_category')['ad_ctr'].mean
().reset_index(name='avg_ctr')
ads_ctr = ads_ctr[ads_ctr['avg_ctr']>overall_ctr]
ads_ctr['ctr_diff'] = (ads_ctr['avg_ctr'] - overall_ctr) / ove
rall_ctr * 100

print(ads_ctr)
```

Ready for your own challenge?

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

www.interviewmaster.ai/python-party

InterviewMaster.AI