Statistics_Challenge_MDST.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

Table of contents

Section

+ Code + Text

RAM Disk

Editing

[ ]

Statistics Challenge (Optional) Use the orders.csv dataset in the same directory to complete this challenge.

Background:

There are exactly 100 sneaker shops on a sneaker retailing website, and each of these shops sells only one model of shoe. We want to do some analysis of the average order value (AOV). When we look at orders data over a 30 day window, we naively calculate an AOV of $3145.13. Given that we know these shops are selling sneakers, a relatively affordable item, something seems wrong with our analysis.

Questions:

What went wrong with this metric and our analysis?

Propose some new metrics that better represents the behavior of the stores' customers. Why are these metrics better? You can propose as many new metrics as you wish but quality heavily outweights quantity.

Find the values of your new metrics.

Report any other interesting findings.

Show all of your work in this notebook.

[1]
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]
```
df = pd.read_csv("https://raw.githubusercontent.com/tjamesbu/MDST-Tutorial-Redesign/main/Optional%20Challenges/S
df.head()
```

	order_id	shop_id	user_id	order_value	total_items	payment_method	created_at
0	1	53	746	224	2	cash	2017-03-13 12:36:56
1	2	92	925	90	1	cash	2017-03-03 17:38:52
2	3	44	861	144	1	cash	2017-03-14 4:23:56
3	4	18	935	156	1	credit_card	2017-03-26 12:43:37
4	5	18	883	156	1	credit_card	2017-03-01 4:35:11

[3]
```
df.shape
```

```
(5000, 7)
```

[25]
```
df_date_sort = df.sort_values(by="created_at")
df_date_sort.head(5)
```

	order_id	shop_id	user_id	order_value	total_items	payment_method	created_at
1862	1863	39	738	536	4	cash	2017-03-01 00:08:09
1741	1742	39	910	268	2	cash	2017-03-01 00:10:19
3228	3229	97	912	324	2	cash	2017-03-01 00:14:12
1267	1268	80	798	290	2	credit_card	2017-03-01 00:19:31
2689	2690	49	799	258	2	credit_card	2017-03-01 00:22:25

[26]
```
df_date_sort.tail(5)
```

	order_id	shop_id	user_id	order_value	total_items	payment_method	created_at
2630	2631	53	940	112	1	credit_card	2017-03-30 23:12:13
1685	1686	34	818	244	2	cash	2017-03-30 23:16:10
1474	1475	21	815	142	1	cash	2017-03-30 23:26:54
317	318	52	848	292	2	cash	2017-03-30 23:41:34
2457	2458	95	700	168	1	credit_card	2017-03-30 23:55:35

[27]
```
avg_order_value = df_date_sort['order_value'].mean()
avg_order_value
```

```
3145.128
```

[29]
```
df_shop_id_sort = df.groupby(['shop_id'])
df_shop_id_sort.head()
```

	order_id	shop_id	user_id	order_value	total_items	payment_method	created_at
1862	1863	39	738	536	4	cash	2017-03-01 00:08:09
1741	1742	39	910	268	2	cash	2017-03-01 00:10:19
3228	3229	97	912	324	2	cash	2017-03-01 00:14:12
1267	1268	80	798	290	2	credit_card	2017-03-01 00:19:31
2689	2690	49	799	258	2	credit_card	2017-03-01 00:22:25
...	...	...	...	...	...	...	...

1601	1602	56	944	117	1	credit_card	2017-03-06 01:49:41
2537	2538	38	739	380	2	credit_card	2017-03-06 23:11:48
2617	2618	48	721	468	4	debit	2017-03-07 11:35:25
794	795	48	806	351	3	credit_card	2017-03-08 15:03:28
168	169	48	797	351	3	credit_card	2017-03-08 06:12:47

500 rows × 7 columns

```
[33] print(len(df['shop_id'].unique()))
```

```
100
```

```
[51] avg_cost_per_shoe_pair = sum(df['order_value'])/sum(df['total_items'])
     round(avg_cost_per_shoe_pair)
```

```
358
```

```
[50] avg_of_avgs = df['order_value'].mean()/df['total_items'].mean()
     round(avg_of_avgs)
```

```
358
```

```
[38] payment_method_order_value_comparison = df.groupby(['payment_method'])['order_value'].sum()
     payment_method_order_value_comparison
```

```
payment_method
cash            1164183
credit_card    12945867
debit           1615590
Name: order_value, dtype: int64
```

```
[39] payment_method_total_items_comparison = df.groupby(['payment_method'])['total_items'].sum()
     payment_method_total_items_comparison
```

```
payment_method
cash            3130
credit_card    37415
debit           3391
Name: total_items, dtype: int64
```

```
[48] #total order value per payment method divided by total items per payment method
     cash_method = 1164183/3130
     credit_card_method = 12945867/37415
     debit_method = 1615590/3391
     print('cash =',round(cash_method), ',', 'credit_card =',round(credit_card_method), ',', 'debit =',round(debit_me
```

```
cash = 372 , credit_card = 346 , debit = 476
```

To calculate the Average Order Value (AOV), more calculations need to be performed than just determining the average of all the order values. Only calculating the average of all the order_value values does not account for the number of orders (total_items) and will not give you the average cost per pair of shoes. You could calculate the total of the order_value and the total of the total_items, and then divide the total order_value by the total total_items to get the average cost per pair of shoes....or you could calculate the average of all the order_value values and the average of all the total_items values, and divide those two averages (order_value avg / total_items avg) to get the average cost per pair of shoes. You also could evaluate the average amount spent per type of payment_method. After doing this it was observed that the average amount spent per type of payment method was slightly higher when using debit, and was close to the average cost of a pair of shoes when using cash or credit_card.

```
[ ]
```

```
[22]
```

```
(5000, 7)
```

Colab paid products - Cancel contracts here

0s completed at 11:40 PM