

FINAL PROJECT

PYTHON

Selviana

Data Analyst 14 - MySkill

INTRODUCTION

Pada kasus ini terdapat empat dataset seperti Customer_detail, Order_detail, Payment_detail, dan sku_detail.

Database berasal dari salah satu e-commerce Tokopaedi dengan periode waktu dari tahun 2021 sampai 2022.

Adapun penjelasan masing-masing dataset secara berurutan dari variabel class description sebagai berikut :

sku_detail:

<code>id</code>	<code>object</code>	angka unik dari produk (dapat digunakan untuk key saat join)
<code>sku_name</code>	<code>object</code>	nama dari produk
<code>base_price</code>	<code>float64</code>	harga barang yang tertera pada tagging harga / price
<code>cogs</code>	<code>int64</code>	cost of goods sold / total biaya untuk menjual 1 produk
<code>category</code>	<code>object</code>	kategori produk

INTRODUCTION

customer_detail:

id	object	angka unik dari pelanggan
registered_date	object	tanggal pelanggan mulai mendaftarkan diri sebagai anggota

payment_detail:

id	int64	angka unik dari metode pembayaran
payment_method	object	metode pembayaran yang digunakan

order_detail:

id	object	angka unik dari order / id_order
customer_id	object	angka unik dari pelanggan
order_date	object	tanggal saat dilakukan transaksi
sku_id	object	angka unik dari produk (sku adalah stock keeping unit)
price	int64	harga yang tertera pada tagging harga
qty_ordered	int64	jumlah barang yang dibeli oleh pelanggan
before_discount	float64	nilai harga total dari produk ($price * qty_ordered$)
discount_amount	float64	nilai diskon product total
after_discount	float64	nilai harga total produk ketika sudah dikurangi dengan diskon
is_gross	int64	menunjukkan pelanggan belum membayar pesanan
is_valid	int64	menunjukkan pelanggan sudah melakukan pembayaran
is_net	int64	menunjukkan transaksi sudah selesai
payment_id	int64	angka unik dari metode pembayaran

DATA PRE-PROCESSING

Import Modules

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pandas.tseries.offsets import BDay
```

Import dataset

```
#Sumber data yang digunakan
path_od = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/order_detail.csv"
path_pd = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/payment_detail.csv"
path_cd = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/customer_detail.csv"
path_sd = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/sku_detail.csv"
df_od = pd.read_csv(path_od)
df_pd = pd.read_csv(path_pd)
df_cd = pd.read_csv(path_cd)
df_sd = pd.read_csv(path_sd)
```

Connect sqlite3 to Run SQL in Colab

```
#Menjalankan SQL di Colab
from sqlite3 import connect
conn = connect(':memory:')
df_od.to_sql('order_detail', conn, index=False, if_exists='replace')
df_pd.to_sql('payment_detail', conn, index=False, if_exists='replace')
df_sd.to_sql('sku_detail', conn, index=False, if_exists='replace')
df_cd.to_sql('customer_detail', conn, index=False, if_exists='replace')
```

Joining Table Using SQL

```
#Query SQL untuk menggabungkan data
df = pd.read_sql("""
SELECT
    order_detail.*,
    payment_detail.payment_method,
    sku_detail.sku_name,
    sku_detail.base_price,
    sku_detail.cogs,
    sku_detail.category,
    customer_detail.registered_date
FROM order_detail
LEFT JOIN payment_detail
    on payment_detail.id = order_detail.payment_id
LEFT JOIN sku_detail
    on sku_detail.id = order_detail.sku_id
LEFT JOIN customer_detail
    on customer_detail.id = order_detail.customer_id
""", conn)
```

Changing Data Type for Proper processing

```
#Mengubah tipe data agar mudah dilakukan pengolahan data
df = df.astype({"before_discount":'int', "discount_amount":'int', "after_discount":'int','base_price':'int'})
df.dtypes
```

Changing Data Column

```
#Mengubah tipe kolom Date menjadi Datetime
df['order_date']= pd.to_datetime(df['order_date'])
df['registered_date']= pd.to_datetime(df['registered_date'])
df.dtypes
```

Result

```
id                      object
customer_id              object
order_date                datetime64[ns]
sku_id                    object
price                     int64
qty_ordered               int64
before_discount           int64
discount_amount           int64
after_discount            int64
is_gross                  int64
is_valid                  int64
is_net                    int64
payment_id                int64
payment_method             object
sku_name                  object
base_price                 int64
cogs                      int64
category                  object
registered_date           datetime64[ns]
dtype: object
```

STUDY CASE #1

Akhir tahun ini, perusahaan akan memberikan hadiah bagi pelanggan yang memenangkan kompetisi Festival Akhir Tahun. Tim Marketing membutuhkan bantuan untuk menentukan perkiraan hadiah yang akan diberikan pada pemenang kompetisi nantinya. Hadiah tersebut akan diambil dari TOP 5 Produk dari Kategori Mobiles & Tablets selama tahun 2022, dengan jumlah kuantitas penjualan (valid = 1) paling tinggi.

Mohon bantuan, untuk mengirimkan data tersebut sebelum akhir bulan ini ke Tim Marketing. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards
Tim Marketing

SYNTAX

```
#Memfilter data dengan is_valid = 1 dan category= mobile & tables tahun 2022
topfivproduct = df[(df['is_valid'] == 1)&(df['category'] == 'Mobiles & Tablets')&(df['order_date'].dt.year == 2022)]

#Menghitung total penjualan dan total after discount
topfivproduct = topfivproduct.groupby('sku_name')[['qty_ordered', 'after_discount']].sum(
    numeric_only= True).sort_values(
    by= 'qty_ordered', ascending= False)

topfivproduct.head(5)
```

INSIGHT

sku_name	qty_ordered	after_discount
IDROID_BALRX7-Gold	1000	518752000
IDROID_BALRX7-Jet black	31	17079202
Infinix Hot 4-Gold	15	10787130
samsung_Grand Prime Plus-Black	11	10207362
infinix_Zero 4-Grey	10	13339420

Dari output hasil di atas, dapat disimpulkan bahwa dari top 5 categori mobiles dan tables, produk yang paling banyak diminati pada tahun 2022 yaitu **IDROID_BALRX7-Gold**

STUDY CASE #2

Menindaklanjuti meeting gabungan Tim Werehouse dan Tim Marketing, kami menemukan bahwa ketersediaan stock produk dengan Kategori Others pada akhir 2022 kemarin masih banyak.

1. Kami mohon bantuan untuk melakukan pengecekan data penjualan kategori tersebut dengan tahun 2021 secara kuantitas penjualan. Dugaan sementara kami, telah terjadi penurunan kuantitas penjualan pada 2022 dibandingkan 2021. (Mohon juga menampilkan data ke-15 kategori)
2. Apabila memang terjadi penurunan kuantitas penjualan pada kategori Others, kami mohon bantuan untuk menyediakan data TOP 20 nama produk yang mengalami penurunan paling tinggi pada 2022 jika dibanding dengan 2021. Hal ini kami gunakan sebagai bahan diskusi pada meeting selanjutnya.

Mohon bantuan untuk mengirimkan data tersebut paling lambat 4 hari dari hari ini. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards

Tim Werehouse

SYNTAX 2.1

```
#2.1
df2021 = df[(df['is_valid']== 1) &\\
(df['order_date'].dt.year==2021)].groupby(
    by = ['category'])['qty_ordered'].sum()\\
.sort_values(ascending=[False])\\
.reset_index(name= "qty_2021")

df2022 = df[(df['is_valid']== 1) &\\
(df['order_date'].dt.year==2022)].groupby(
    by = ['category'])['qty_ordered'].sum()\\
.sort_values(ascending=[False])\\
.reset_index(name= "qty_2022")
|
df_combined = df2021.merge(df2022, how ='inner', left_on='category', right_on='category')

df_combined['changes']= df_combined['qty_2022']- df_combined['qty_2021']

df_combined.sort_values(by=['changes'], ascending=[True])
```

INSIGHT 2.1

	category	qty_2021	qty_2022	changes
1	Others	426	263	-163
0	Soghaat	759	612	-147
3	Men Fashion	237	175	-62
9	Beauty & Grooming	168	153	-15
7	Books	171	195	24
11	Appliances	124	148	24
6	Health & Sports	173	200	27
12	Computing	109	153	44
5	School & Education	184	237	53
4	Home & Living	193	250	57
8	Kids & Baby	170	227	57
14	Entertainment	77	150	73
2	Superstore	327	536	209
10	Women Fashion	140	489	349
13	Mobiles & Tablets	107	1154	1047

Berdasarkan hasil disamping, dapat disimpulkan bahwa terjadi penurunan kuantitas penjualan tahun 2022 yaitu penurunan tertinggi terjadi pada category Others dengan penurunan sebesar 163 penjualan.

SYNTAX 2.2

```
#2.2
df2021_others = df[(df['is_valid']== 1) &\\
                    (df['category']=='Others') &\\
                    (df['order_date'].dt.year == 2021)]\\
                    .groupby(by=['sku_name'])['qty_ordered'].sum()\\
                    .sort_values(ascending=[False])\\
                    .reset_index(name ='qty_2021')

df2022_others = df[(df['is_valid']== 1) &\\
                    (df['category']=='Others') &\\
                    (df['order_date'].dt.year == 2022)]\\
                    .groupby(by=['sku_name'])['qty_ordered'].sum()\\
                    .sort_values(ascending=[False])\\
                    .reset_index(name ='qty_2022')

df_combined_others = df2021_others.merge(df2022_others, how ='outer', left_on='sku_name', right_on='sku_name')

df_combined_others['changes']= df_combined_others['qty_2022']- df_combined_others['qty_2021']

df_combined_others.fillna(0,inplace = True)

df_combined_others.sort_values(by=['changes'],ascending=[True]).head(20)
```

INSIGHT 2.2

	sku_name	qty_2021	qty_2022	changes
0	RB_Dettol Germ Busting Kit-bf	200.0	45.0	-155.0
1	Telemall_MM-DR-HB-L	23.0	2.0	-21.0
5	kansai_NeverWet	10.0	1.0	-9.0
7	emart_00-1	7.0	1.0	-6.0
13	MEGUIAR_G12711	4.0	1.0	-3.0
30	aw_Ultra Shine Wash & Wax-64oz./1893ml	2.0	1.0	-1.0
15	Aladdin_bike_cover	3.0	2.0	-1.0
26	Tmall_MM-DR-PAD	2.0	1.0	-1.0
25	EntertainerAsia_Vouch 365-2017 Mobile App Lahore	2.0	1.0	-1.0
119	MEGUIAR_G3200	0.0	1.0	0.0
108	clickbuscollection_nwc-36	0.0	1.0	0.0
109	clickbuscollection_nwc-32	0.0	1.0	0.0

Dari output hasil di atas, terlihat bahwa kategori others mengalami penurunan pada masing-masing produknya. Penurunan terbesar terjadi pada produk **RB_Dettol Gem Busting Kit-bf** sebesar 155 penjualan

STUDY CASE #3

Terkait ulang tahun perusahaan pada 2 bulan mendatang, Tim Digital Marketing akan memberikan informasi promo bagi pelanggan pada akhir bulan ini. Kriteria pelanggan yang akan kami butuhkan adalah mereka yang sudah melakukan check-out namun belum melakukan pembayaran (`is_gross = 1`) selama tahun 2022. Data yang kami butuhkan adalah ID Customer dan Registered Date.

Mohon bantuan, untuk mengirimkan data tersebut sebelum akhir bulan ini ke Tim Digital Marketing. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards
Tim Digital Marketing

SYNTAX

```
df3 = df[(df['is_gross'] == 1)&(df['is_valid'] == 0)&(df['is_net']== 0)&(df['order_date'].dt.year == 2022)]  
df3[['customer_id','registered_date']].drop_duplicates()
```

INSIGHT

	customer_id	registered_date
9	C246762L	2022-05-08
18	C848774L	2021-11-07
19	C693415L	2022-04-12
21	C180595L	2022-04-22
22	C587425L	2022-03-22
...
5855	C653797L	2022-04-03
5856	C394076L	2021-10-12
5859	C248585L	2022-07-10
5865	C471304L	2022-05-13
5881	C265450L	2022-02-17

820 rows x 2 columns

Dari hasil output di samping, diperoleh bahwa pelanggan yang akan mendapatkan promo di akhir bulan dengan kriteria sudah melakukan checkout namun belum melakukan pembayaran yaitu sebanyak **820 orang**.

STUDY CASE #4

Pada bulan October hingga Desember 2022, kami melakukan campaign setiap hari Sabtu dan Minggu. Kami hendak menilai, apakah campaign tersebut cukup berdampak pada kenaikan penjualan (before_discount). Mohon bantuan untuk menampilkan data:

1. Rata-rata harian penjualan weekends (Sabtu dan Minggu) vs rata-rata harian penjualan weekdays (Senin-Jumat) per bulan tersebut. Apakah ada peningkatan penjualan pada masing-masing bulan tersebut.
2. Rata-rata harian penjualan weekends (Sabtu dan Minggu) vs rata-rata harian penjualan weekdays (Senin-Jumat) keseluruhan 3 bulan tersebut.

Mohon bantuan untuk mengirimkan data tersebut paling lambat minggu depan. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards
Tim Campaign

SYNTAX 4.1

```
df_od['order_date'] = pd.to_datetime(df_od['order_date'])

df_od['month'] = df_od['order_date'].dt.month
df_od['day'] = df_od['order_date'].dt.dayofweek

df_od['day_type'] = 'Weekdays'
df_od.loc[df_od['day'] >= 5, 'day_type'] = 'Weekends'

filtered_df = df_od[(df_od['order_date'] >= '2022-10-01') & (df_od['order_date'] <= '2022-12-31')]

average_sales = filtered_df.groupby(['month', 'day_type'])['before_discount'].mean().round()

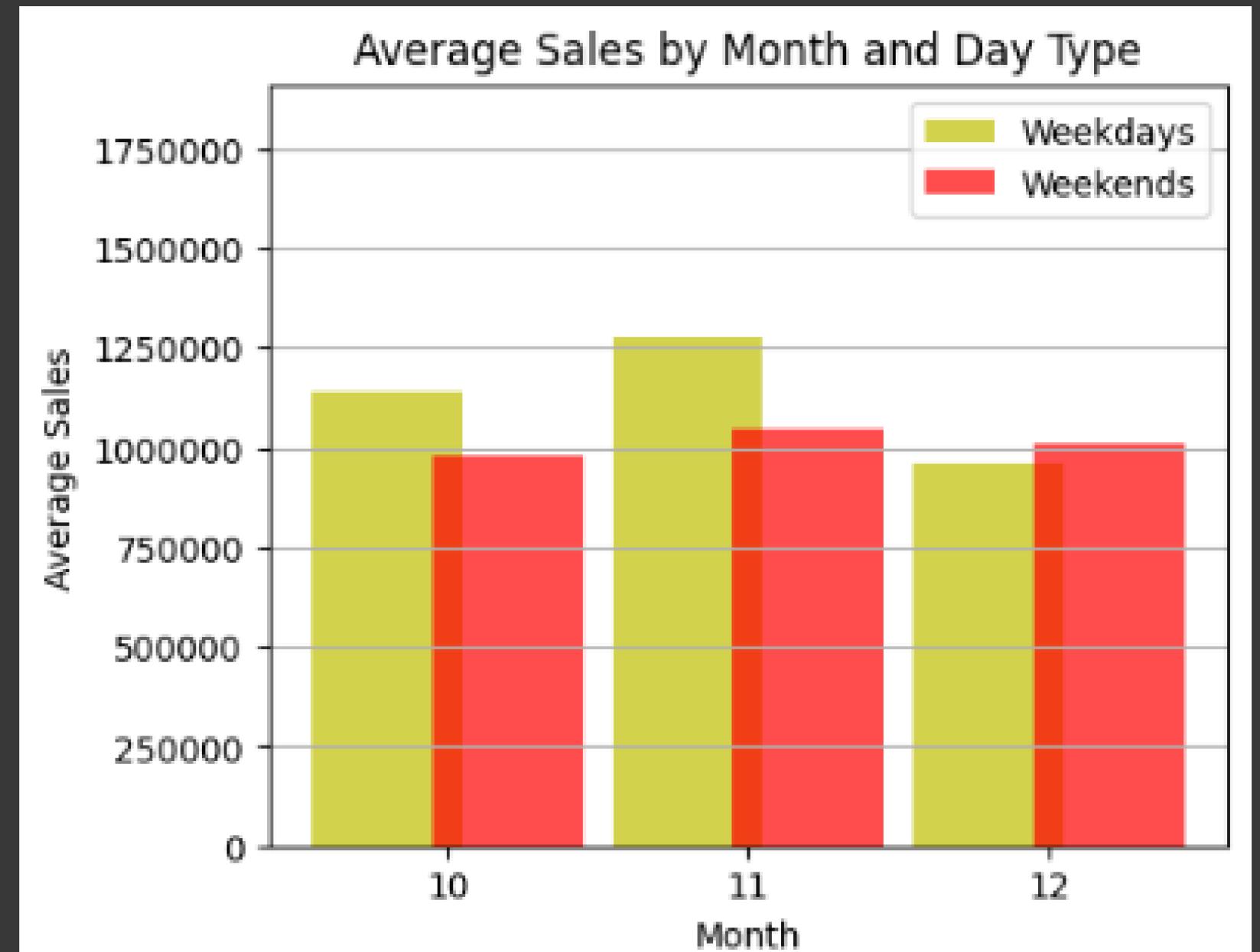
print(average_sales)

# Membuat plot dengan matplotlib
plt.figure(figsize=(5, 4))
weekdays_data = average_sales[average_sales.index.get_level_values('day_type') == 'Weekdays']
weekends_data = average_sales[average_sales.index.get_level_values('day_type') == 'Weekends']
x_weekdays = weekdays_data.index.get_level_values('month') - 0.2
x_weekends = weekends_data.index.get_level_values('month') + 0.2
plt.bar(x_weekdays, weekdays_data.values, width=0.5, label='Weekdays', color='y', alpha=0.7)
plt.bar(x_weekends, weekends_data.values, width=0.5, label='Weekends', color='r', alpha=0.7)
plt.ticks(label_format='plain', axis='y')
plt.ylim(0, max(average_sales.values) * 1.5)

plt.xlabel('Month')
plt.ylabel('Average Sales')
plt.title('Average Sales by Month and Day Type')
plt.xticks(range(10, 13))
plt.legend()
plt.grid(axis='y')
plt.show()
```

INSIGHT

```
month  day_type
10    Weekdays    1138896.0
      Weekends     976614.0
11    Weekdays    1274436.0
      Weekends    1046216.0
12    Weekdays    961183.0
      Weekends    1008588.0
Name: before_discount, dtype: float64
```



Dari hasil output di samping, diperoleh bahwa pada weekday memiliki rata-rata pendapatan tertinggi selama tiga bulan terakhir

SYNTAX 4.2

```
df_od['order_date'] = pd.to_datetime(df_od['order_date'])

df_od['month'] = df_od['order_date'].dt.month
df_od['day'] = df_od['order_date'].dt.dayofweek

df_od['day_type'] = 'Weekdays'
df_od.loc[df_od['day'] >= 5, 'day_type'] = 'Weekends'

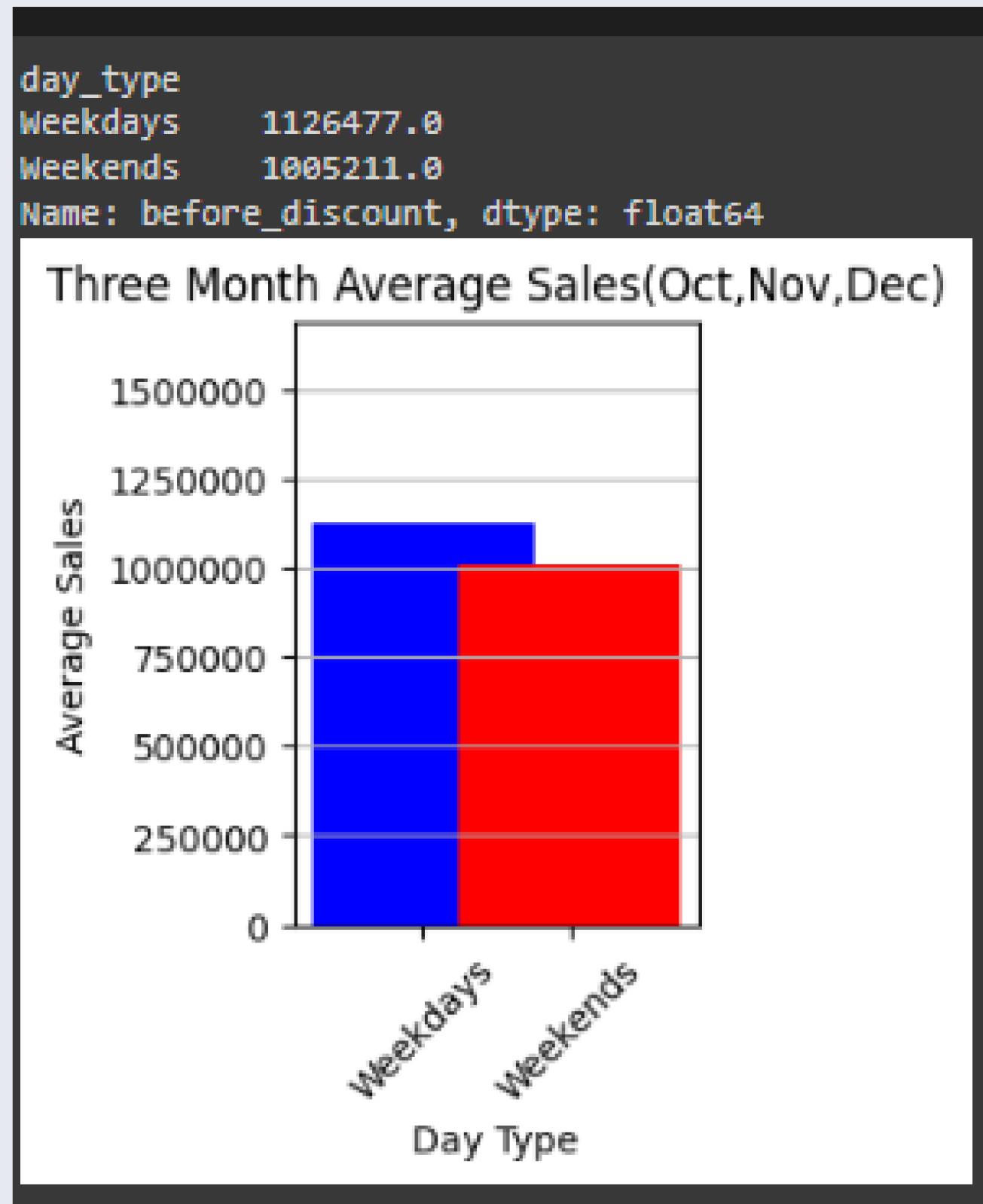
filtered_df = df_od[(df_od['order_date'] >= '2022-10-01') & (df_od['order_date'] <= '2022-12-31')]

average_sales = filtered_df.groupby('day_type')['before_discount'].mean().round()

print(average_sales)

plt.figure(figsize=(2, 3))
plt.ticklabel_format(style='plain', axis='y')
plt.ylim(0, max(average_sales.values) * 1.5)
width = 1.5
plt.bar(average_sales.index, average_sales.values, width, color=['blue', 'red'])
plt.title('Three Month Average Sales(Oct,Nov,Dec)')
plt.xlabel('Day Type')
plt.ylabel('Average Sales')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

INSIGHT



Dari hasil output di samping, diperoleh bahwa rata-rata nilai pendapatan penjualan gabungan terbesar pada saat weekdays

[source code in google colab](#)

THANKS YOU



selvianazainuddin1010@gmail.com



selvianazainuddin