project-main

February 6, 2025

[1]:
```
import pandas as pd
```

[2]:
```
import numpy as np
```

[ ]:
```

```

0.1 IMPORTING & READING THE DATASETS

[3]:
```
product_df = pd.read_csv("product_data.csv")
payment_df = pd.read_csv("payment_data.csv")
order_df = pd.read_csv("order_data.csv")
customer_df = pd.read_csv("customers_data.csv")
credit_card_df = pd.read_csv("credit_card_data.csv")
```

[ ]:
```

```

1 TASK

1.0.1 Problem Statement:

A company wants to cut down on its cost of production, producing only products that meet a revenue target of $1500 or a quantity target of 65 in the 1st, 2nd and 3rd quarter of 2024

[ ]:
```

```

2 SOLUTION

[ ]:
```

```

[4]:
```
# Converting order_date and payment_date to datetime as it allows for easier␣
  ↪filtering and comparison of dates later.
order_df['order_date'] = pd.to_datetime(order_df['order_date'])
payment_df['payment_date'] = pd.to_datetime(payment_df['payment_date'])

# Filter orders from Q1-Q3 of 2024
```

```
filtered_orders = order_df[(order_df['order_date'].dt.year == 2024) &␣
↪(order_df['order_date'].dt.quarter <= 3)]

# Merge orders with payment data to get revenue per order
order_payments = filtered_orders.merge(payment_df, on='order_id', how='left')

# Merge with product data to get price and product name
order_payments = order_payments.merge(product_df[['product_id', 'product_name',␣
↪'unit_price']], on='product_id', how='left')

# Calculate total revenue per product
order_payments['total_revenue'] = order_payments['quantity'] *␣
↪order_payments['unit_price']
revenue_per_product = order_payments.groupby(['product_id',␣
↪'product_name'])['total_revenue'].sum().reset_index()

# Aggregate quantity sold per product
product_sales = filtered_orders.groupby('product_id')['quantity'].sum().
↪reset_index()

# Merge revenue and quantity data
final_df = product_sales.merge(revenue_per_product, on='product_id', how='left')

# Filter products meeting either condition
profitable_products = final_df[(final_df['quantity'] >= 65) |␣
↪(final_df['total_revenue'] >= 1500)]

# Identify top customers contributing to revenue
customer_revenue = order_payments.groupby('customer_id')['total_revenue'].sum().
↪reset_index()
customer_revenue = customer_revenue.merge(customer_df[['customer_id',␣
↪'customer_name']], on='customer_id', how='left')
customer_revenue = customer_revenue.sort_values(by='total_revenue',␣
↪ascending=False)

# Display results
print("Profitable Products:")
print(profitable_products)
print("\nTop Customers:")
print(customer_revenue.head(10))
```

```
Profitable Products:
   product_id  quantity          product_name  total_revenue
0         P02         6          Acer Nitro 5       10199.94
1         P04         4             iPhone 12        3599.96
5         P09         2          HP Gaming 15        3199.98
6         P10         4                Dell G5        7200.00
```

```
7          P11          3                   Dell G7        6299.97
17         P23          2               Samsung S23        3599.98
21         P29          3          Macbook Air 13.6        2939.97
23         P33         10    Lenovo IdeaPad Flex 5i        2999.90
24         P36          4              ASUS ROG Z13        5596.00
25         P37          5         MSI Gaming Laptop        4395.00
26         P39          6   SAMSUNG Galaxy Z Fold 4        8699.94

Top Customers:
    customer_id  total_revenue     customer_name
36          170        8099.97       Bess Cotton
33          147        5400.00      Gladys Curry
3            14        5099.97   Angelo Castillo
12           55        4812.00      Karen Tanaka
41          186        4349.97      Rayford King
13           56        4299.95  Christopher Gwin
39          180        3879.97     Larry Barrera
25          114        3599.98      Tina Goodwin
0             2        3099.97     Virginia Read
42          191        2699.97      David Chafin
```

[5]:
```python
# Save results to CSV files
profitable_products.to_csv("Profitable_Products.csv", index=False)
customer_revenue.to_csv("Top_Customers.csv", index=False)

# Display results
print("Profitable Products saved to Profitable_Products.csv")
print("Top Customers saved to Top_Customers.csv")
```

```
Profitable Products saved to Profitable_Products.csv
Top Customers saved to Top_Customers.csv
```

[ ]:
```python

```

[ ]:
```python

```