Internship Report

on

Data Analytics on Shopping Mall Data Using Python & Power BI

(From 17-04-24 to 18-05-24)

Submitted By

Shashank Srivastava

Batch:24 Sep 2023

Under Guidance of

Guide

Priyanka Banik

Table of Contents :

1. Introduction

In this report, we will demonstrate a comprehensive data analytics workflow using Python. We will generate synthetic data using the Faker library, clean and prepare the data, and perform various analyses with visualizations. This project aims to showcase the practical application of Python in data analytics.

2. Data Creation Using Faker Library

First, we use the Faker library to create synthetic data for our analysis. The dataset includes customer information, orders, sales, and profits.

```
import faker
import pandas as pd
import random
import numpy as np
from datetime import datetime, timedelta

# Initialize faker
fake = faker.Faker()

# Define categories
categories = ['Electronics', 'Clothing', 'Books', 'Beauty', 'Home', 'Sports']

# Define function to generate synthetic e-commerce data
def generate_shopping_data(num_rows):
    data = {
        'order_id': [fake.random_int(min=1000, max=9999) for _ in range(num_rows)],
        'product_name': [fake.word() for _ in range(num_rows)],
        'price': [fake.random_number(digits=2) for _ in range(num_rows)],
        'quantity': [fake.random_int(min=1, max=10) for _ in range(num_rows)],
        'customer_name': [fake.name() for _ in range(num_rows)],
        'address': [fake.address() for _ in range(num_rows)],
        'email': [fake.email() for _ in range(num_rows)],
        'gender': [fake.random_element(elements=('Male', 'Female')) for _ in range(num_rows)],
        'date': [fake.date_between(start_date='-1y', end_date='today') for _ in range(num_rows)],
        'location': [fake.country() for _ in range(num_rows)],
        'category': [random.choice(categories) for _ in range(num_rows)]
    }
```

3. Data Cleaning and Preparation

Removing Blank Rows

We remove rows that have all elements as NaN using `dropna(how='all')`.

```
Location=r'C:\\Users\\DELL\\Documents\\.ipynb_checkpoints\\Note_new.csv'
Shopping_data.to_csv(Location,index=False)
```

```
Shopping_data = Shopping_data.dropna(how='all')
Shopping_data
```

Removing Duplicates from Order ID

We ensure each order ID is unique by removing duplicates.

```
Shopping_data.drop_duplicates(subset=['order_id'],inplace=True)
```

Checking for Null Values

We check for any remaining null values in the dataset.

```
Shopping_data.isnull().sum()
```

4. Data Transformation

Capitalizing Column Headings

We capitalize the column headings for better readability.

```
#Capitalized the All Heading of first coloum
Shopping_data.columns = Shopping_data.columns.str.capitalize()
Shopping_data.columns
```

```
Index(['Order_id', 'Product_name', 'Price', 'Quantity', 'Customer_name',
       'Address', 'Email', 'Gender', 'Date', 'Location', 'Category',
       'Sale_amount', 'Profit'],
      dtype='object')
```

Converting Date Column to DateTime Format

We convert the **'OrderDate'** column from an object to a datetime format.

```
pd.to_datetime(Shopping_data['Date'])
```

```
0     2024-02-26
1     2024-04-30
2     2024-01-09
3     2023-08-18
4     2024-03-16
         ...
994   2024-02-23
995   2023-11-26
996   2024-04-13
997   2024-01-20
998   2024-02-12
Name: Date, Length: 914, dtype: datetime64[ns]
```

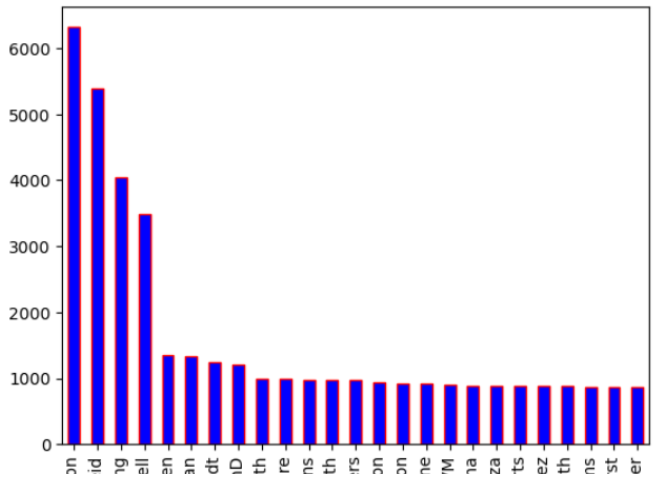
5. Data Analysis and Visualization

Top 10 Customer-wise Sales

We plot the top 10 customers by sales using a bar chart.

```
']: #top 15 customer who purchase most
    import matplotlib.pyplot as plt
    Shopping_data.groupby('Customer_name')['Sale_amount'].sum().sort_values(ascending = False).head(25).plot(kind='bar',color='blue'
```

```
']: <Axes: xlabel='Customer_name'>
```

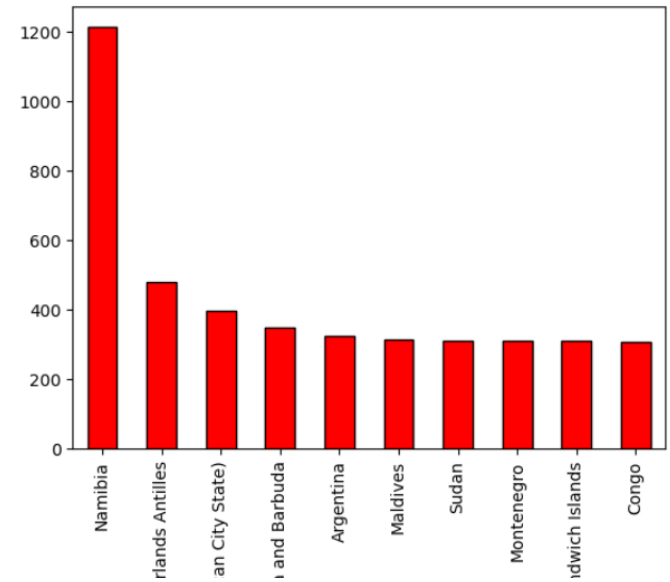


Location-wise Profit

We plot the profit by location using a histogram.

```
Shopping_data.groupby('Location')['Profit'].sum().sort_values( ascending =False).head(10).plot(kind='bar',color='red',edgecolor=
```

```
<Axes: xlabel='Location'>
```

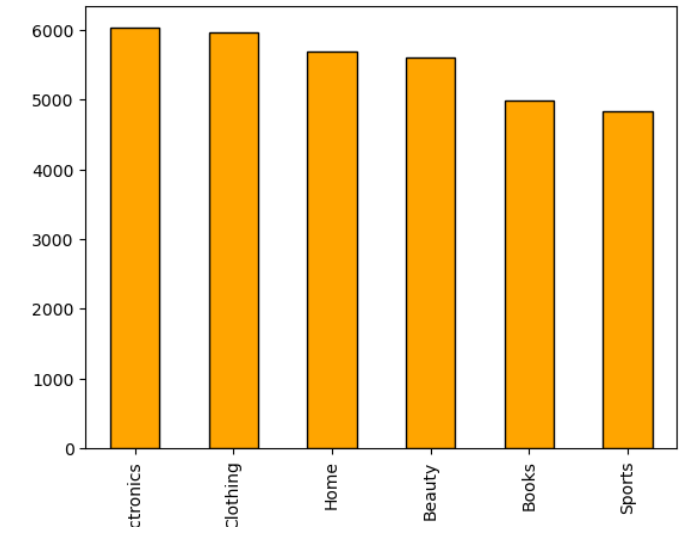


Category-wise Profit Sorted in Descending Order

We plot the profit by category sorted in descending order.

```
:
Shopping_data.groupby('Category')[ 'Profit'].sum().sort_values( ascending=False).plot(kind='bar',color='orange',edgecolor='black'
```

```
: <Axes: xlabel='Category'>
```

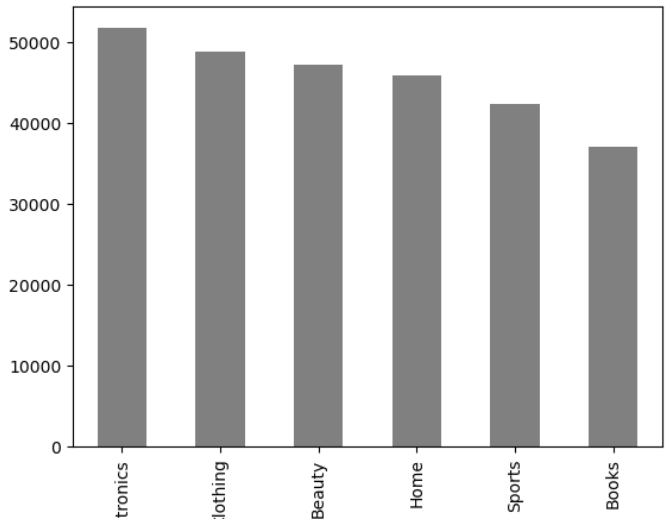


Category-wise Total Sales Sorted in Descending Order

We plot the total sales by category sorted in descending order.

```
]: Shopping_data.groupby('Category')['Sale_amount'].sum().sort_values(ascending=False).plot(kind='bar', color='Gray')
```

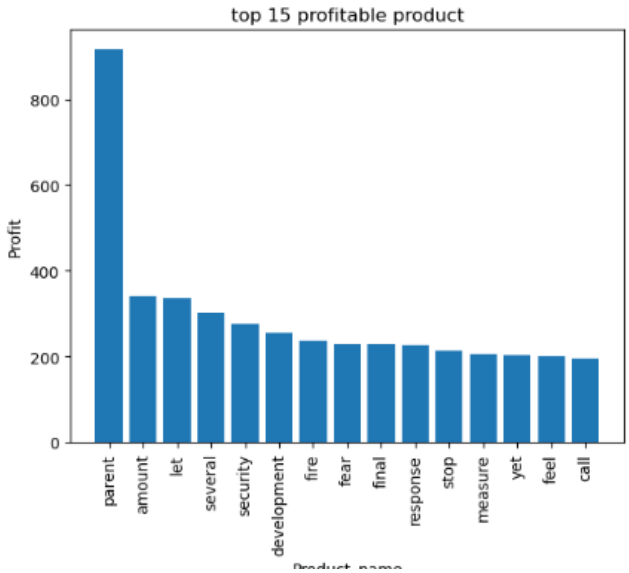
```
]: <Axes: xlabel='Category'>
```



Top 10 Most Profitable Products

We plot the top 10 most profitable products.

```
In [24]: New_Pr_1=Shopping_data.groupby('Product_name')['Profit'].sum().sort_values(ascending = False).head(15)
         plt.bar(New_Pr_1.index,New_Pr_1.values)
         plt.xticks(rotation=90)
         plt.xlabel('Product_name')
         plt.ylabel('Profit')
         plt.title('top 15 profitable product')
         plt.show()
```

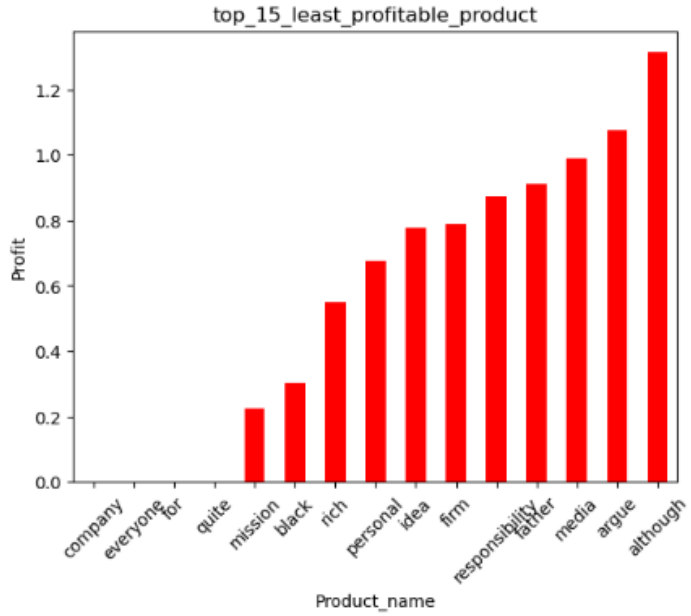


Top 15 Least Sold Products for Liquidation

We plot the top 15 least sold products for potential liquidation.

```
Shopping_data.groupby('Product_name')['Profit'].sum().sort_values(ascending = True).head(15).plot(kind='bar',color='red')
plt.xticks(rotation=45)
plt.xlabel('Product_name')
plt.ylabel('Profit')
plt.title('top_15_least_profitable_product')
```

```
Text(0.5, 1.0, 'top_15_least_profitable_product')
```



Gender-wise Sales Analysis

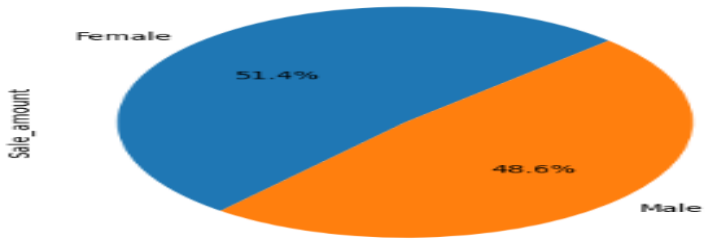
We analyze and plot sales by gender.

```
Shopping_data.groupby('Gender')['Sale_amount'].sum()
```

```
Gender
Female     140298.012395
Male       132568.949688
Name: Sale_amount, dtype: float64
```

```
Pie_chart=Shopping_data.groupby('Gender')['Sale_amount'].sum()
Pie_chart.plot(kind='pie',autopct='%1.1f%%', startangle=45)
```

```
<Axes: ylabel='Sale_amount'>
```



Data Analytics Using Power BI

Total Orders Card

1. **Add a Card Visual**:

- Go to the "Visualizations" pane and select "Card".
- Drag the "Order ID" field into the "Values" section.
- Set the aggregation to "Count (Distinct)" to show the total number of unique orders.

Total Sales Card

1. **Add a Card Visual**:

- Select "Card" from the "Visualizations" pane.
- Drag the "Sale Amount" field into the "Values" section.
- Set the aggregation to "Sum" to show the total sales amount.

Total Profit Card

1. **Add a Card Visual**:

- Select "Card" from the "Visualizations" pane.
- Drag the "Profit" field into the "Values" section.
- Set the aggregation to "Sum" to show the total profit amount.

Total Quantity Sold Card

1. **Add a Card Visual**:

- Select "Card" from the "Visualizations" pane.
- Drag the "Quantity Sold" field into the "Values" section.
- Set the aggregation to "Sum" to show the total quantity sold.

Date Slicer

1. **Add a Slicer Visual**:

- Select "Slicer" from the "Visualizations" pane.
- Drag the "Order Date" field into the "Field" section.
- Set the slicer to filter by date.

Category-wise Sales Bar Chart

1. **Add a Pie Chart Visual**:

- Select "Pie Chart" from the "Visualizations" pane.
- Drag the "Category" field into the "Axis" section.
- Drag the "Sale Amount" field into the "Values" section.
- Set the aggregation to "Sum".

Month-wise Sales Trend

1. **Add a Bar Chart Visual**:

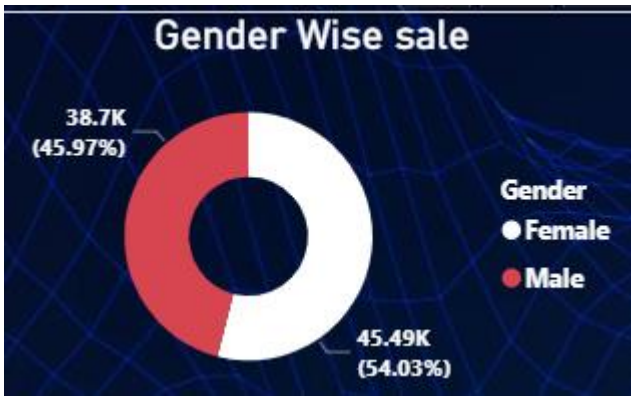
- Select "bar Chart" from the "Visualizations" pane.
- Drag the "Order Date" field into the "Axis" section.
- Drag the "Sale Amount" field into the "Values" section.
- Set the aggregation to "Sum".
- Format the "Order Date" field to show data by month.

Gender-wise Sales Analysis

1. **Add a Pie Chart Visual**:

- Select " Chart" from the "Visualizations" pane.
- Drag the "Gender" field into the "Axis" section.
- Drag the "Sale Amount" field into the "Values" section.

- Set the aggregation to "Sum".



Category-wise Profit Analysis

1. **Add a Do-nut Chart Visual**:

- Select "Do-nut Chart" from the "Visualizations" pane.
- Drag the "Category" field into the "Axis" section.
- Drag the "Profit" field into the "Values" section.
- Set the aggregation to "Sum".

Top 10 Location-wise Sales

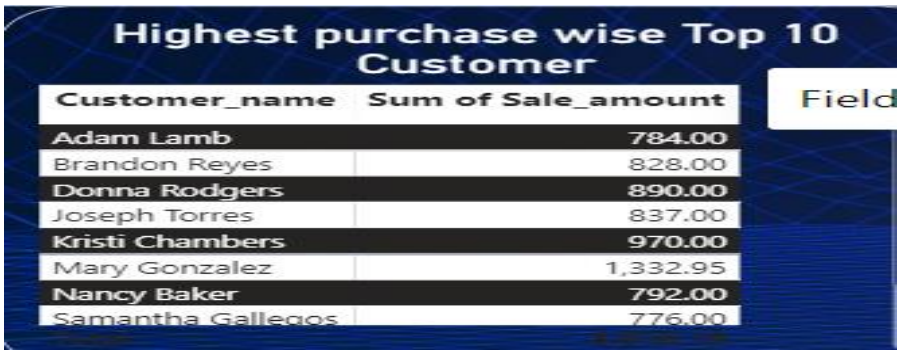
1. **Add a Line Visual**:

- Select "Line" from the "Visualizations" pane.
- Drag the "Location" field into the "Axis" section.
- Drag the "Sale Amount" field into the "Values" section.
- Set the aggregation to "Sum".
- Apply a Top N filter to show only the top 10 locations.

Top 10 Customers Who Purchase Most

1. **Add a Sheet Chart Visual**:

- Select "Sheet Chart" from the "Visualizations" pane.
- Drag the "Customer Name" field into the "Axis" section.
- Drag the "Sale Amount" field into the "Values" section.
- Set the aggregation to "Sum".
- Apply a Top N filter to show only the top 10 customers.



Conclusion

This Power BI dashboard provides a comprehensive analysis of the shopping mall dataset. By examining total orders, sales, profit, and quantity sold, as well as category and location-wise performance, businesses can gain valuable insights into their operations and customer preferences. The visualizations aid in identifying trends and making data-driven decisions for improved business performance.