Active Product Sales Analysis using Matplotlib in Python

Last Updated : 10 Sep, 2024

Every modern company that engages in online sales or maintains a specialized e-commerce website now aims to maximize its throughput in order to determine what precisely their clients need in order to increase their chances of sales. The huge datasets handed to us can be properly analyzed to find out what time of day has the highest user activity in terms of transactions.

In this post, We will use Python Pandas and Matplotlib to analyze the insight of the dataset. We can use the column **Transaction Date,** in this case, to glean useful insights on the busiest time (hour) of the day. You can access the entire dataset here.

Stepwise Implementation

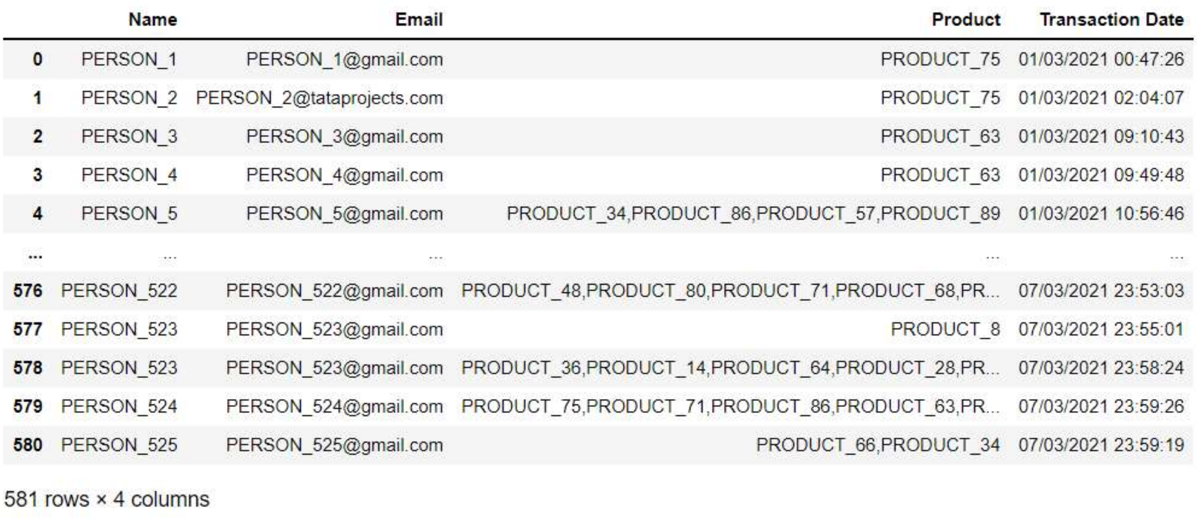
Step 1:

First, We need to create a Dataframe of the dataset, and even before that certain libraries have to be imported.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

Order_Details = pd.read_csv('Order_details(masked).csv')
```

Output:



	Name	Email	Product	Transaction Date
0	PERSON_1	PERSON_1@gmail.com	PRODUCT_75	01/03/2021 00:47:26
1	PERSON_2	PERSON_2@tataprojects.com	PRODUCT_75	01/03/2021 02:04:07
2	PERSON_3	PERSON_3@gmail.com	PRODUCT_63	01/03/2021 09:10:43
3	PERSON_4	PERSON_4@gmail.com	PRODUCT_63	01/03/2021 09:49:48
4	PERSON_5	PERSON_5@gmail.com	PRODUCT_34,PRODUCT_86,PRODUCT_57,PRODUCT_89	01/03/2021 10:56:46
...	...	...	...	...
576	PERSON_522	PERSON_522@gmail.com	PRODUCT_48,PRODUCT_80,PRODUCT_71,PRODUCT_68,PR...	07/03/2021 23:53:03
577	PERSON_523	PERSON_523@gmail.com	PRODUCT_8	07/03/2021 23:55:01
578	PERSON_523	PERSON_523@gmail.com	PRODUCT_36,PRODUCT_14,PRODUCT_64,PRODUCT_28,PR...	07/03/2021 23:58:24
579	PERSON_524	PERSON_524@gmail.com	PRODUCT_75,PRODUCT_71,PRODUCT_86,PRODUCT_63,PR...	07/03/2021 23:59:26
580	PERSON_525	PERSON_525@gmail.com	PRODUCT_66,PRODUCT_34	07/03/2021 23:59:19

581 rows × 4 columns

Step 2:

Create a new column called Time that has the DateTime format after converting the Transaction Date column into it. The DateTime format, which has the pattern **YYYY-MM-DD HH:MM:SS**, can be customized however you choose. Here we're more interested in obtaining hours, so we can have an Hour column by using an in-built function for the same:

```
# here we have taken Transaction
# date column
Order_Details['Time'] = pd.to_datetime(Order_Details['Transaction Date'])

# After that we extracted hour
# from Transaction date column
Order_Details['Hour'] = (Order_Details['Time']).dt.hour
```

Step 3:

We then require the "n" busiest hours. For that, we get the first "n" entries in a list containing the occurrence rates of the hours when the transaction took place. To further simplify the manipulation of the provided data in Python, we may utilize value counts for frequencies and tolist() to convert to list format. We are also compiling a list of the associated index values.

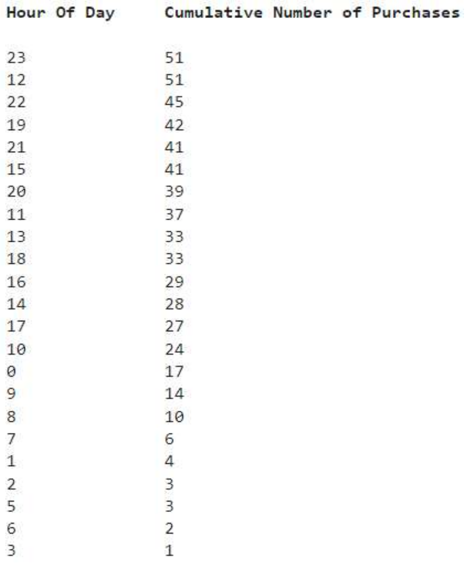
```
# n =24 in this case, can be modified
# as per need to see top 'n' busiest hours
timemost1 = Order_Details['Hour'].value_counts().index.tolist()[:24]
```

Step 4:

Finally, we stack the indices (hour) and frequencies together to yield the final result.

```
tmost = np.column_stack((timemost1,timemost2))

print(" Hour Of Day" + "\t" + "Cumulative Number of Purchases \n")
print('\n'.join('\t\t'.join(map(str, row)) for row in tmost))
```



```
Hour Of Day     Cumulative Number of Purchases

23              51
12              51
22              45
19              42
21              41
15              41
20              39
11              37
13              33
18              33
16              29
14              28
17              27
10              24
0               17
9               14
8               10
7               6
1               4
2               3
5               3
6               2
3               1
```

Step 5:

Before we can create an appropriate data visualization, we must make the list slightly more customizable. To do so, we gather the hourly frequencies and perform the following tasks:

```
timemost = Order_Details['Hour'].value_counts()
timemost1 = []

for i in range(0,23):
    timemost1.append(i)
    
timemost2 = timemost.sort_index()
timemost2.tolist()
timemost2 = pd.DataFrame(timemost2)
```

Step 6:

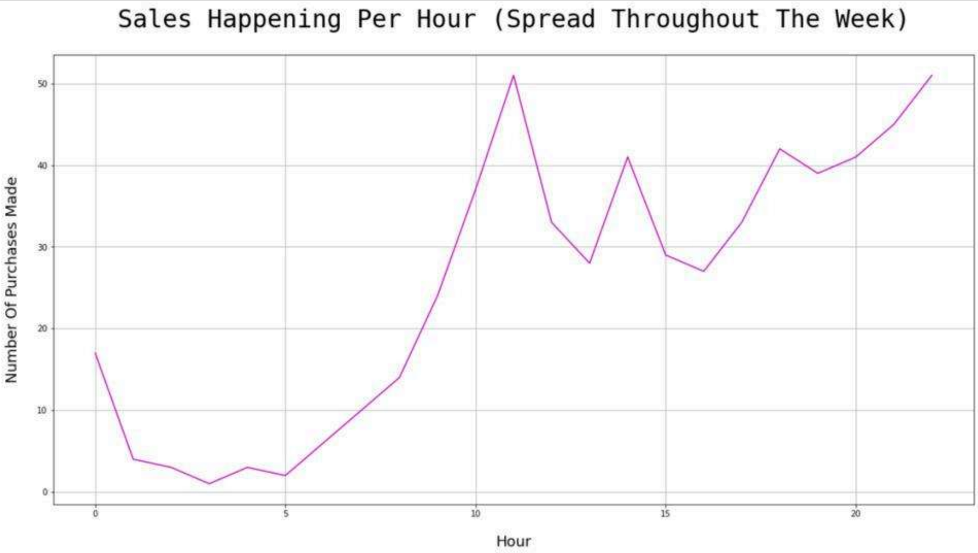
For data visualization, we will proceed with Matplotlib for better libraries. But, It is up to you to choose any of the pre-existing libraries like Matplotlib, Ggplot, Seaborn, etc., to plot the data graphically.

The commands written below are mainly to ensure that X-axis takes up the values of hours and Y-axis takes up the importance of the number of transactions affected, and also various other aspects of a line chart, including color, font, etc., to name a few.

```
plt.figure(figsize=(20, 10))

plt.title('Sales Happening Per Hour (Spread Throughout The Week)',
          fontdict={'fontname': 'monospace', 'fontsize': 30}, y=1.05)

plt.ylabel("Number Of Purchases Made", fontsize=18, labelpad=20)
plt.xlabel("Hour", fontsize=18, labelpad=20)
plt.plot(timemost1, timemost2, color='m')
plt.grid()
plt.show()
```



The results are indicative of how sales typically peak in late evening hours prominently, and this data can be incorporated into business decisions to promote a product during that time specifically.

Get the complete notebook link here

Colab Link : click here.

Dataset Link : click here.

Active Product Sales Analysis using Matplotlib in Python

Similar Reads

RFM Analysis Analysis Using Python

In business analytics one of the easiest ways to understand and categorize customers is through RFM analysis. RFM stands for Recency, Frequency and Monetary value which are three simple ways to look at...

4 min read

Draw Multiple Y-Axis Scales In Matplotlib

Why areMatplotlib is a powerful Python library, with the help of which we can draw bar graphs, charts, plots, scales, and more. In this article, we'll try to draw multiple Y-axis scales in Matplotlib. Why are...

6 min read

Medical Insurance Price Prediction using Machine Learning - Python

You must have heard some advertisements regarding medical insurance that promises to help financially in case of any medical emergency. One who purchases this type of insurance has to pay premiums monthly...

7 min read

Zomato Data Analysis Using Python

Understanding customer preferences and restaurant trends is important for making informed business decisions in food industry. In this article, we will analyze Zomatoâ€™s restaurant dataset using Python to...

3 min read

Loan Eligibility Prediction using Machine Learning Models in Python

Have you ever thought about the apps that can predict whether you will get your loan approved or not? In this article, we are going to develop one such model that can predict whether a person will get his/her...

5 min read

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:
K 061, Tower K, Gulshan Vivante

Company

About Us

Legal

Privacy Policy

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Python Web Scraping

OpenCV Tutorial

Python Interview Question

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects