

## NM PROJECT PHASE-3

### **Product Demand Prediction (Case Study)**

A product company plans to offer discounts on its product during the upcoming holiday season. The company wants to find the price at which its product can be a better deal compared to its competitors. For this task, the company provided a dataset of past changes in sales based on price changes. You need to train a model that can predict the demand for the product in the market with different price segments.

1. Product\_Code
2. Warehouse
3. Product\_Category
4. Date
5. Order\_Demand

I hope you now understand what kind of problem statements you will get for the product demand prediction task. In the section below, I will walk you through predicting product demand with machine learning using Python.

### **Product Demand Prediction using Python**

Let's start by importing the necessary Python libraries and the dataset we need for the task of product demand prediction:

```
import pandas as pd
```

```
import numpy as np
```

```
import plotly.express as px
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
```

```
df = pd.read_csv('/content/drive/MyDrive/Historical Product Demand.csv')
```

```
df.head()
```

	Product_Code	Warehouse	Product_Category	Date	Order_Demand
0	Product_0993	Whse_J	Category_028	2012/7/27	100
1	Product_0979	Whse_J	Category_028	2012/1/19	500
2	Product_0979	Whse_J	Category_028	2012/2/3	500
3	Product_0979	Whse_J	Category_028	2012/2/9	500
4	Product_0979	Whse_J	Category_028	2012/3/2	500

```
df.shape
```

```
(1048575, 5)
```

```
df.columns
```

```
Index(['Product_Code', 'Warehouse', 'Product_Category', 'Date',  
      'Order_Demand'],  
      dtype='object')
```

```
df.Product_Code.unique()
```

```
array(['Product_0993', 'Product_0979', 'Product_1159', ...,  
      'Product_0237', 'Product_0644', 'Product_0853'], dtype=object)
```

```
df.Warehouse.unique()
```

```
array(['Whse_J', 'Whse_S', 'Whse_C', 'Whse_A'], dtype=object)
```

```
df.Product_Category.nunique()
```

```
33
```

```
df.dtypes
```

```
Product_Code      object  
Warehouse          object  
Product_Category  object  
Date              object  
Order_Demand      object  
dtype: object
```

```
def check_order_demand(x):
    try:
        int(x)
    except:
        return False
    return True
#Check where Order_demand is not an integer
df[~df.Order_Demand.apply(lambda x: check_order_demand(x))].head(6)
```

	Product_Code	Warehouse	Product_Category	Date	Order_Demand
112290	Product_2169	Whse_A	Category_024	2012/8/9	(1)
112307	Product_2132	Whse_A	Category_009	2012/11/1	(24)
112308	Product_2144	Whse_A	Category_009	2012/11/1	(24)
112356	Product_2118	Whse_A	Category_009	2012/3/7	(50)
112357	Product_2120	Whse_A	Category_009	2012/3/7	(100)
112360	Product_1794	Whse_A	Category_024	2012/6/28	(1)

```
def change_to_int(x):
    try:
        return int(x)
    except:
        return int(x[1:-1])
check = '(10)'
change_to_int(check)
```

10

```
df.Order_Demand = df.Order_Demand.apply(lambda x: change_to_int(x) )
```

```
df.describe()
```

Order_Demand	
count	1.048575e+06
mean	4.906977e+03
std	2.892678e+04
min	0.000000e+00
25%	2.000000e+01
50%	3.000000e+02
75%	2.000000e+03
max	4.000000e+06

```
df = df.rename(columns = {'Product_Code': 'Code',  
                          'Product_Category': 'Category',  
                          'Order_Demand': 'Demand'})  
  
df.head()
```

	Code	Warehouse	Category	Date	Demand
0	Product_0993	Whse_J	Category_028	2012/7/27	100
1	Product_0979	Whse_J	Category_028	2012/1/19	500
2	Product_0979	Whse_J	Category_028	2012/2/3	500
3	Product_0979	Whse_J	Category_028	2012/2/9	500
4	Product_0979	Whse_J	Category_028	2012/3/2	500

```
100 * df.isna().sum()[3]/ df.shape[0]
```

```
1.0718355863910545
```

```
df = df.dropna()  
df.isna().sum()
```

```
Code      0  
Warehouse 0  
Category  0  
Date      0  
Demand    0  
dtype: int64
```

```
df.Date.min(), df.Date.max()
```

```
('2011/1/8', '2017/1/9')
```

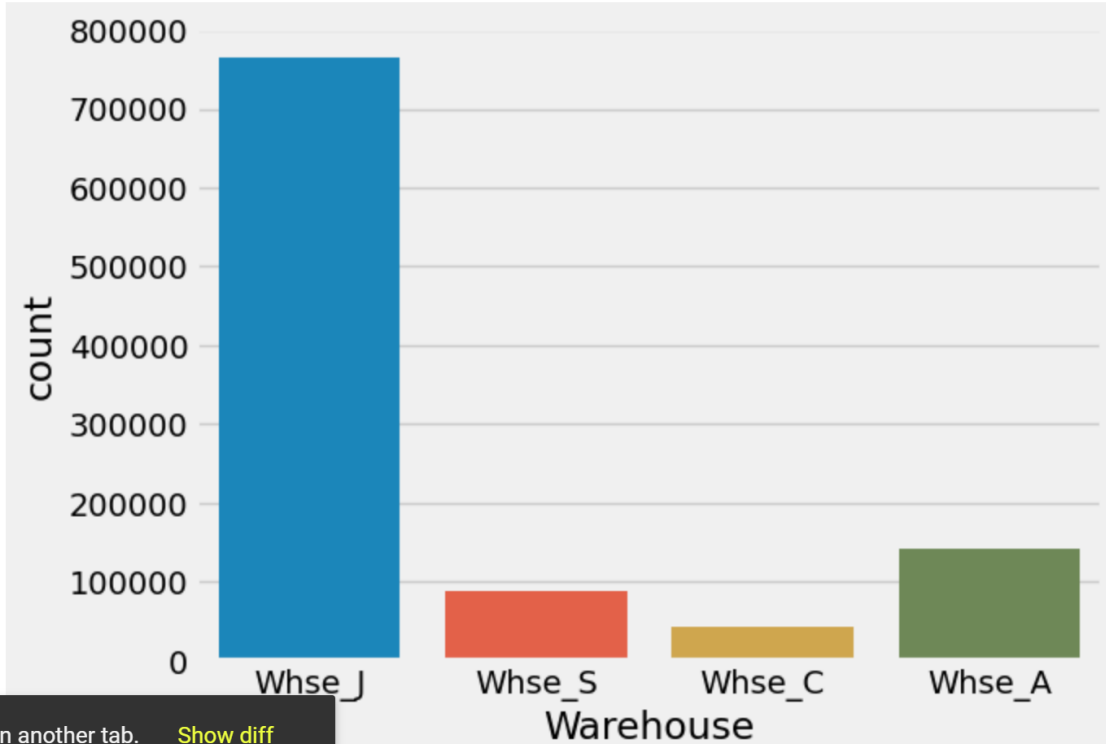
---

```
sns.countplot(x = 'Warehouse', data = df)
```

```
<Axes: xlabel='Warehouse', ylabel='count'>
```

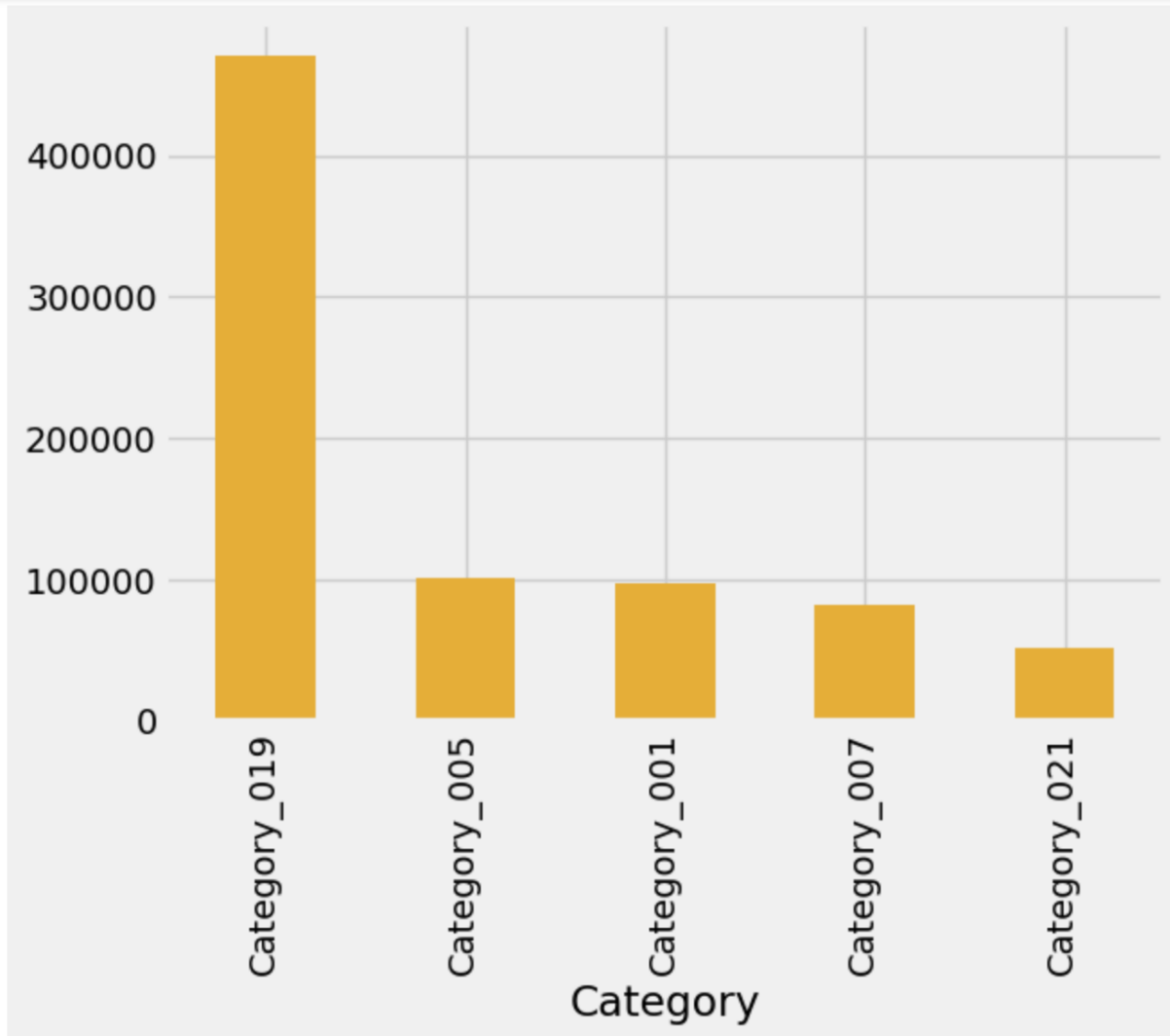
```
sns.countplot(x = 'Warehouse', data = df)
```

```
<Axes: xlabel='Warehouse', ylabel='count'>
```

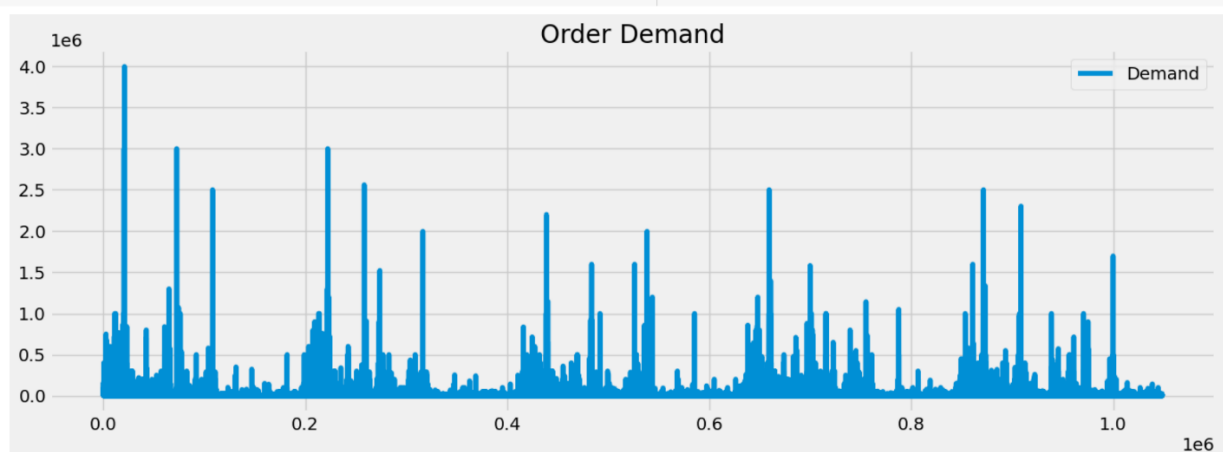


in another tab. [Show diff](#)

```
# Plot the 5 most popular category
df.Category.value_counts().head(5).plot(kind = 'bar', color = color_pal[2])
plt.xlabel('Category')
plt.show()
```



```
df.plot(kind = 'line',figsize=(15, 5),color = color_pal[0], title = 'Order Demand' )
plt.show()
```



```
df.Demand.skew()
```

```
31.432925049321977
```

```
# Total Demand by Warehouse
```

```
warehouse_Demand = df.groupby('Warehouse')['Demand'].sum()
```

```
warehouse_Demand
```

```
Warehouse
```

```
Whse_A    147877431
```

```
Whse_C    585071404
```

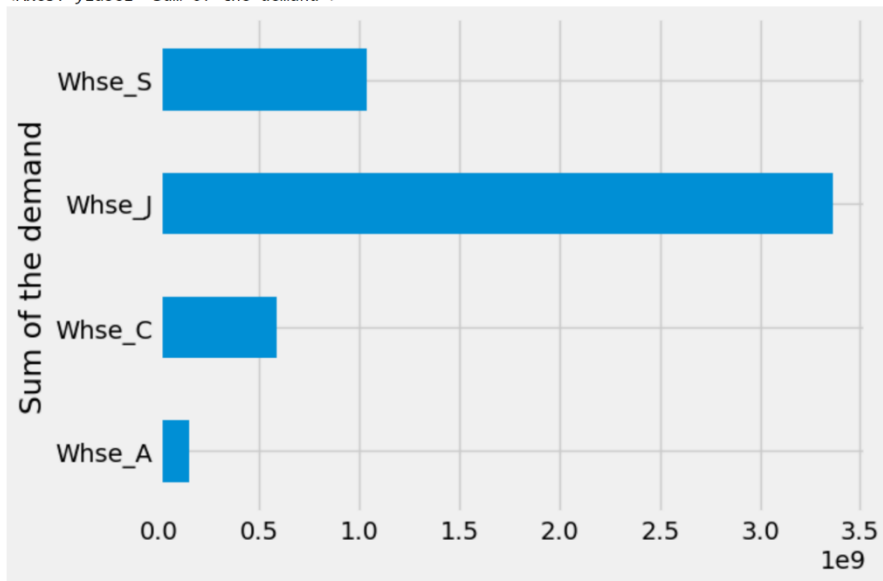
```
Whse_J    3363200396
```

```
Whse_S    1038024700
```

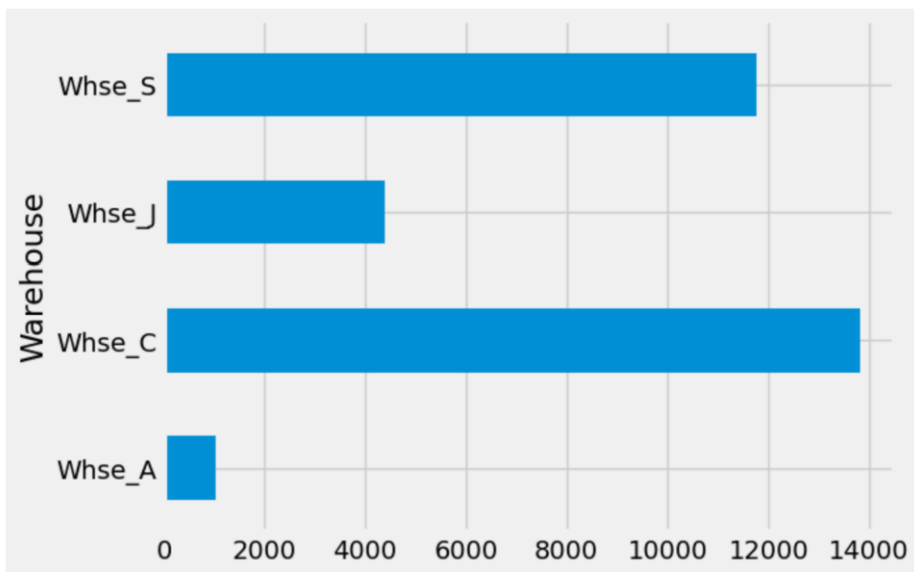
```
Name: Demand, dtype: int64
```

```
warehouse_Demand.plot(kind = 'barh', ylabel = 'Sum of the demand' )
```

```
<Axes: ylabel='Sum of the demand'>
```



```
df.groupby('Warehouse')['Demand'].mean().plot(kind = 'barh')
plt.show()
```



```
df.head()
```

	Code	Warehouse	Category	Date	Demand
0	Product_0993	Whse_J	Category_028	2012/7/27	100
1	Product_0979	Whse_J	Category_028	2012/1/19	500
2	Product_0979	Whse_J	Category_028	2012/2/3	500
3	Product_0979	Whse_J	Category_028	2012/2/9	500
4	Product_0979	Whse_J	Category_028	2012/3/2	500

```
# features, Target variable
Features = ['day_of_the_week', 'Quarter', 'Month', 'Year', 'Week']
target = ['Demand']
```