Product Demand Prediction with Machine Learnings

**Phase 3: Development Part 1**

**Problem:**

Begin building the product demand prediction model by loading and preprocessing the dataset.

Collect and preprocess the historical sales data and external factors for analysis.

Program:

***# Data Manipulation***

**import numpy as** **np**

**import pandas as** **pd**

***# Data Visualization***

**import matplotlib.pyplot as** **plt**

**%matplotlib inline**

**import seaborn as** **sns**

**import plotly.express as** **px**

**import** **bokeh**

***# For Analysis and Forecasting***

**from scipy import** **stats**

***# Others***

**import** **datetime**

**import** **os**

**import** **pickle**

**import** **requests**

***# Loading the Data***

**df = pd.read\_csv(r"/kaggle/input/productdemandforecasting/Historical Product Demand.csv")**

**df.head(5)**

***# Description of the data***

**df.describe()**

**df.info()**

***# finding the percentage of missing value***

**print("Number of attributes with null vaules: ", df.isnull().any().sum())**

**print("Percentage of missing values: ",df.isnull().any(axis=1).sum()/len(df)\*100)**

***# Dropping the missing values***

**df.dropna(axis=0, how="any", inplace=True)**

**In [7]:**

***#Changing the datatype to datetime***

**df["Date"] = pd.to\_datetime(df['Date'])**

**df["Order\_Demand"] = df["Order\_Demand"].str.replace("(", "")**

**df["Order\_Demand"] = df["Order\_Demand"].str.replace(")", "")**

***#Changing the datatype to float***

**df["Order\_Demand"] = df["Order\_Demand"].astype(float)**

**In [8]:**

**linkcode**

**df = df.sort\_values(by=['Date', 'Product\_Code'])**

**df = df.set\_index('Date')**

**df.head()**

**category\_yearly\_demand = df.groupby([df.index.year, 'Product\_Category'])['Order\_Demand'].mean()**

**In [10]:**

**linkcode**

**padded\_category\_data = {}**

**for** **category, category\_data in category\_yearly\_demand.groupby(level = 'Product\_Category'):**

***# print(f"Category: {category}")***

**padded\_category\_data[category] = [0 for \_ in range(7)]**

**for** **year, total\_demand in category\_data.items():**

**index = ((year[0] - 2010) % 7) - 1**

**padded\_category\_data[category][index] = total\_demand**

**fig = plt.figure(figsize=(12, 25))**

**rows, cols = 11, 3**

**x = [2011, 2012, 2013, 2014, 2015, 2016, 2017]**

**for** **title, data in padded\_category\_data.items():**

***# Create subplots in the grid***

**ax = fig.add\_subplot(rows, cols, int(title[-2:]))**

***# Plotting data on the current subplot***

**ax.plot(x, data)**

**ax.set\_title(title)**

**plt.tight\_layout()**

**fig.suptitle("Yearly Average Demand for all the Product Categories", y=1.02)**

**padded\_yearly\_categories = {}**

**for** **year, year\_data in category\_yearly\_demand.groupby(level = 'Date'):**

***# print(f"Category: {category}")***

**padded\_yearly\_categories[year] = [0 for \_ in range(33)]**

**for** **category, total\_demand in year\_data.items():**

***# print(category)***

**index = (int(category[1][-2:]) % 33) - 1**

**padded\_yearly\_categories[year][index] = total\_demand**

**x = [i+1 for i in range(33)]**

**rows = len(padded\_yearly\_categories)**

**cols = 1**

**fig, axes = plt.subplots(nrows=rows, ncols=cols, figsize=(10, 20))**

**colors = plt.cm.viridis(np.linspace(0, 1, len(x)))**

**for** **i, (year, data) in enumerate(padded\_yearly\_categories.items()):**

***# Calculate the row and column indices for the subplot***

***# Create a bar plot in the current subplot***

**bars = axes[i].bar(x, data, color=colors)**

**axes[i].bar\_label(bars, labels=x, fontsize = 8)**

***# Set the category title as the subplot title***

**axes[i].set\_title(year)**

***# Hide only the y-axis scales (ticks)***

***# axes[row\_idx, col\_idx].get\_yaxis().set\_visible(False)***

**plt.tight\_layout()**

**fig.suptitle("Yearwise Average Demand of all Product Categories", y=1.01)**

**plt.show()**

warehouse\_yearly\_demand = df.groupby([df.index.year, 'Warehouse'])['Order\_Demand'].mean()

In [13]:

linkcode

demand\_data = {}

warehouses = []

years = [2011 + i for i in range(7)]

for warehouse, warehouse\_data in warehouse\_yearly\_demand.groupby(level='Warehouse'):

warehouses.append(warehouse)

demand\_data[warehouse] = [0 for i in range(7)]

for year, year\_data in warehouse\_data.items():

index = ((year[0] - 2010) % 7) - 1

demand\_data[warehouse][index] = year\_data

*# print(demand\_data)*

*# Determine the number of warehouses and the number of years*

num\_warehouses = len(warehouses)

num\_years = len(years)

*# Set the width of the bars*

bar\_width = 0.15

*# Create a figure*

fig, ax = plt.subplots(figsize=(12, 8))

*# Define the index for the x-axis*

x = np.arange(num\_years)

*# Create a grouped bar chart*

for i, warehouse in enumerate(warehouses):

*# Offset the x-position for each warehouse*

x\_pos = x + i \* bar\_width

*# Plot the demand values for the current warehouse*

ax.bar(x\_pos, demand\_data[warehouse], width=bar\_width, label=warehouse)

*# Set x-axis labels and tick positions*

ax.set\_xticks(x + (num\_warehouses - 1) \* bar\_width / 2)

ax.set\_xticklabels(years)

*# Set labels and title*

ax.set\_xlabel('Year')

ax.set\_ylabel('Demand')

ax.set\_title('Yearly Average Demand by Warehouse')

*# Add a legend to distinguish the warehouses*

ax.legend()

*# Show the chart*

plt.show()

output:

| 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 |

| Product\_Code | Warehouse | Product\_Category | Date | Order\_Demand |
| --- | --- | --- | --- | --- |
| count | 1048575 | 1048575 | 1048575 | 1037336 | 1048575 |
| unique | 2160 | 4 | 33 | 1729 | 3828 |
| top | Product\_1359 | Whse\_J | Category\_019 | 2013/9/27 | 1000 |
| freq | 16936 | 764447 | 481099 | 2075 | 112682 |

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1048575 entries, 0 to 1048574

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Product\_Code 1048575 non-null object

1 Warehouse 1048575 non-null object

2 Product\_Category 1048575 non-null object

3 Date 1037336 non-null object

4 Order\_Demand 1048575 non-null object

dtypes: object(5)

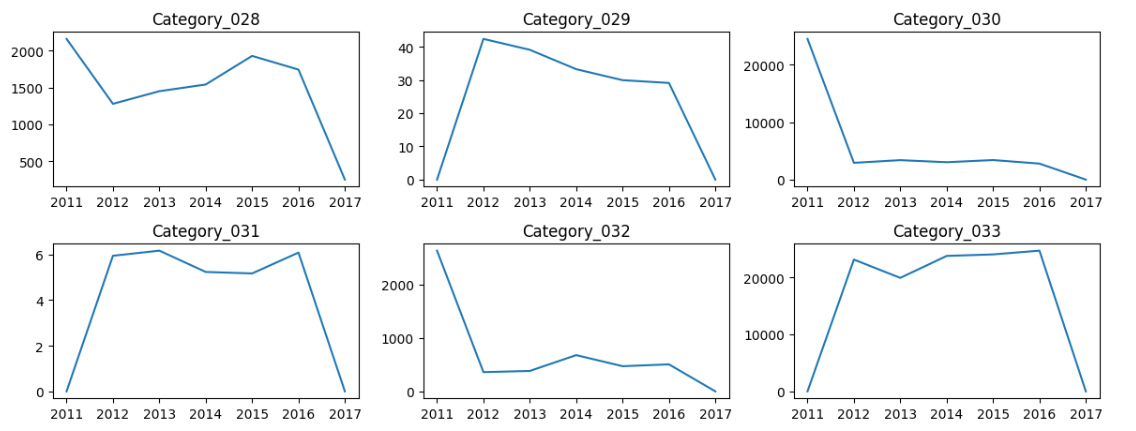
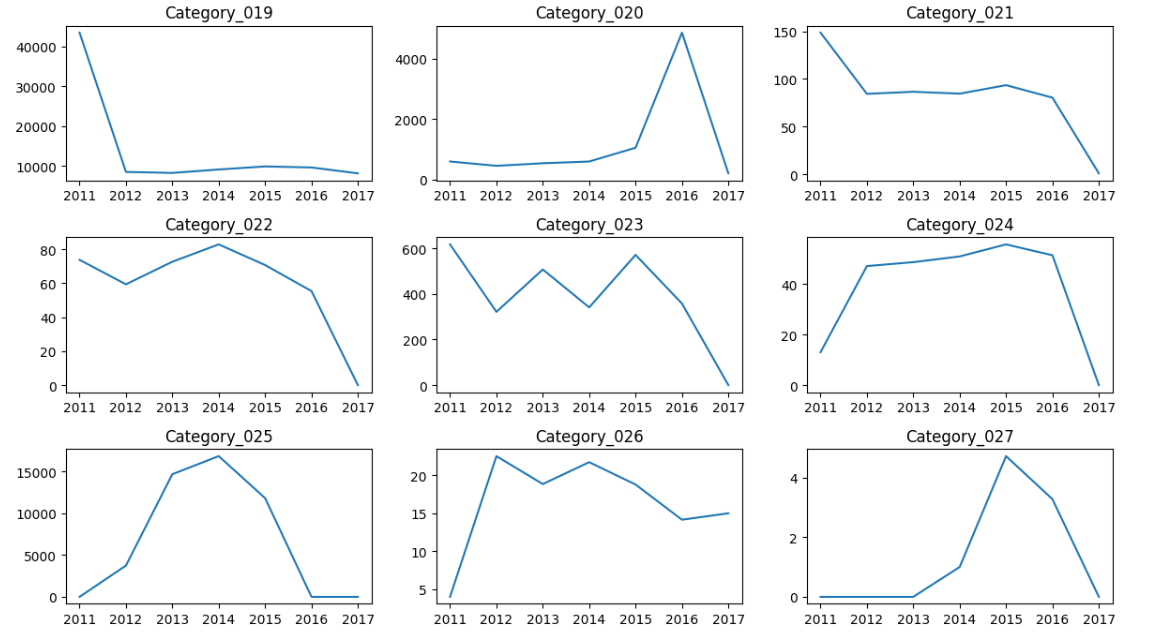
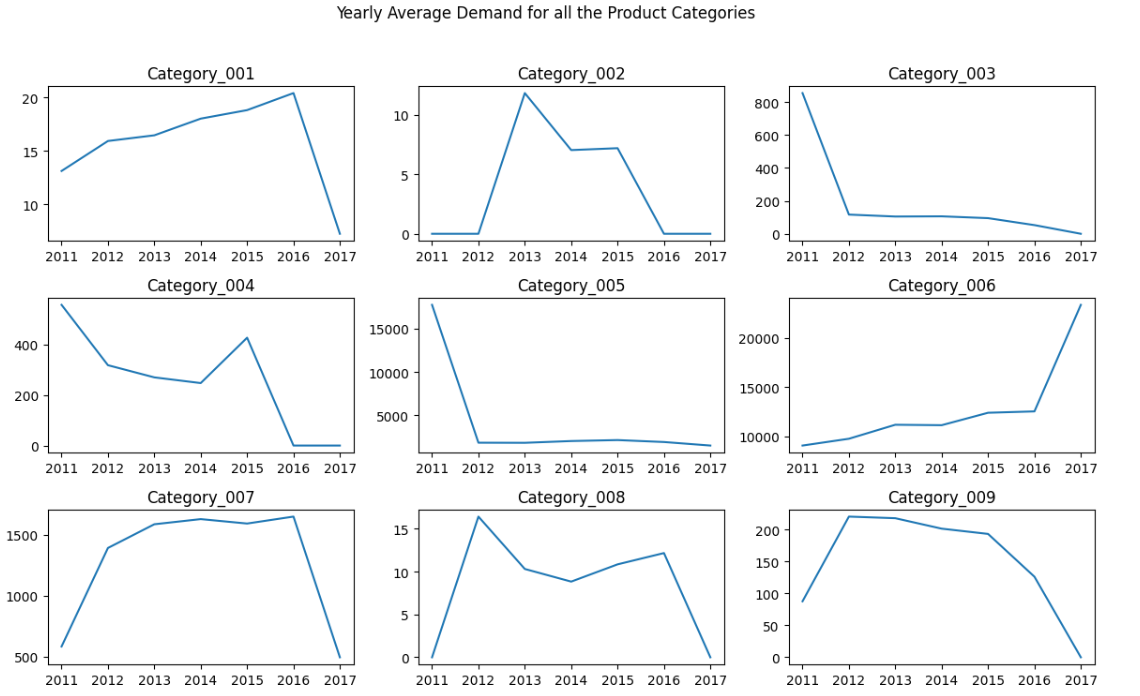
memory usage: 40.0+ MB

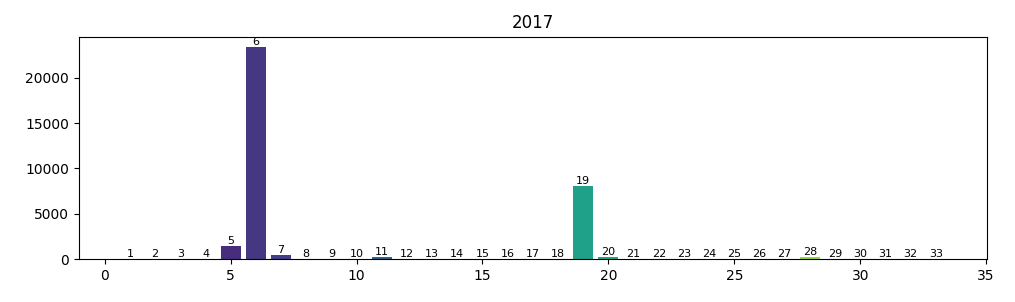
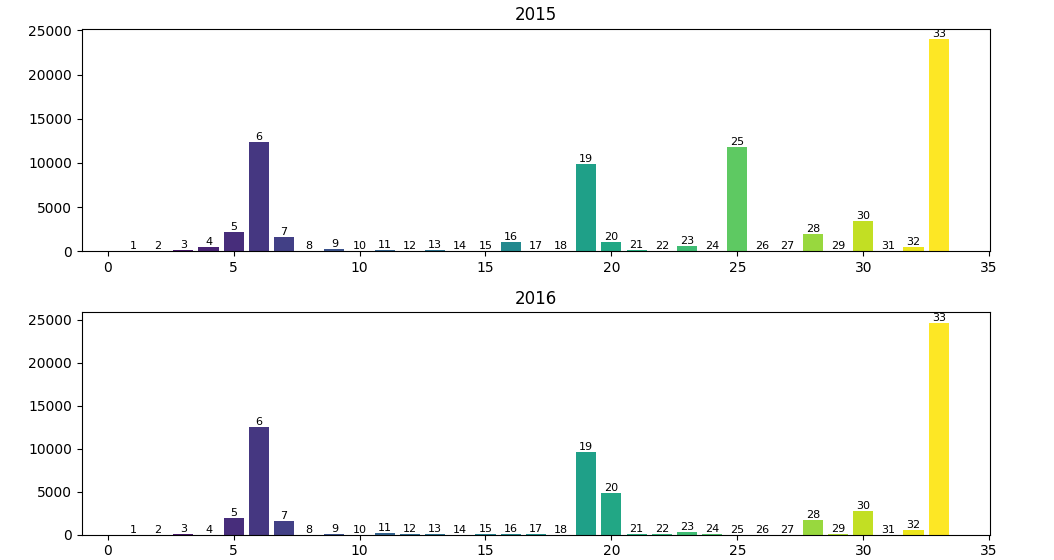
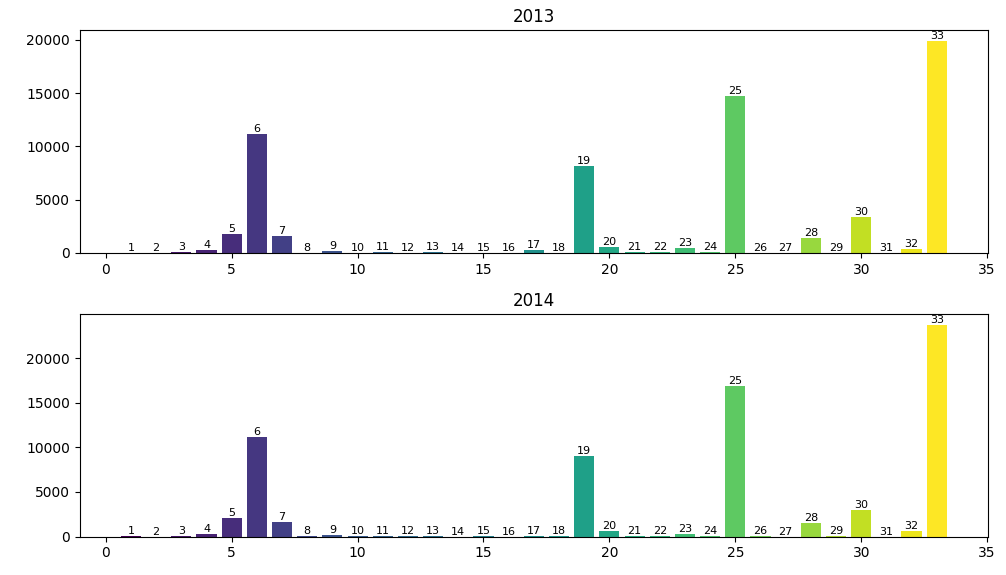
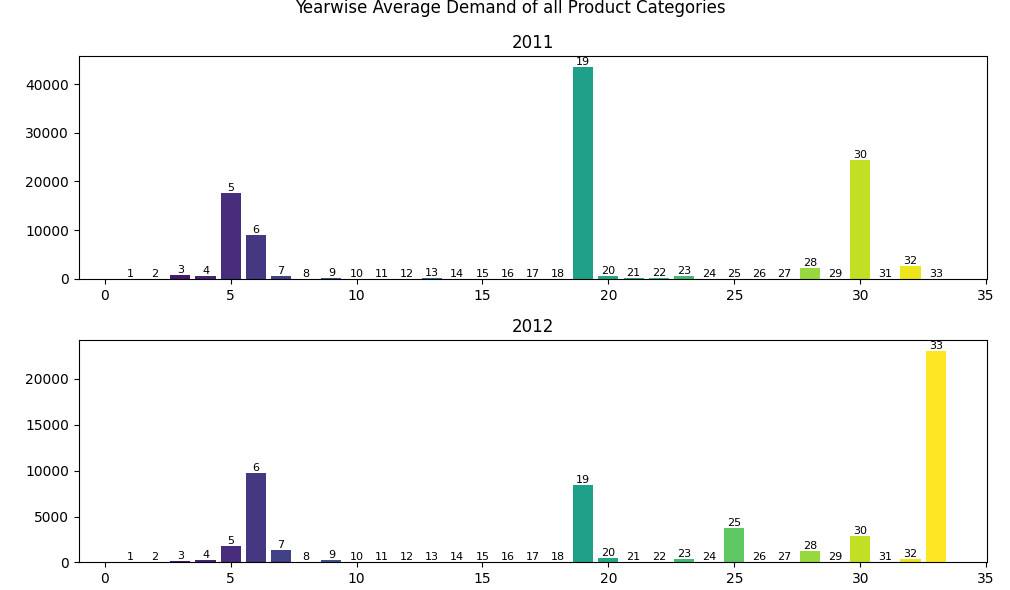
Number of attributes with null vaules: 1

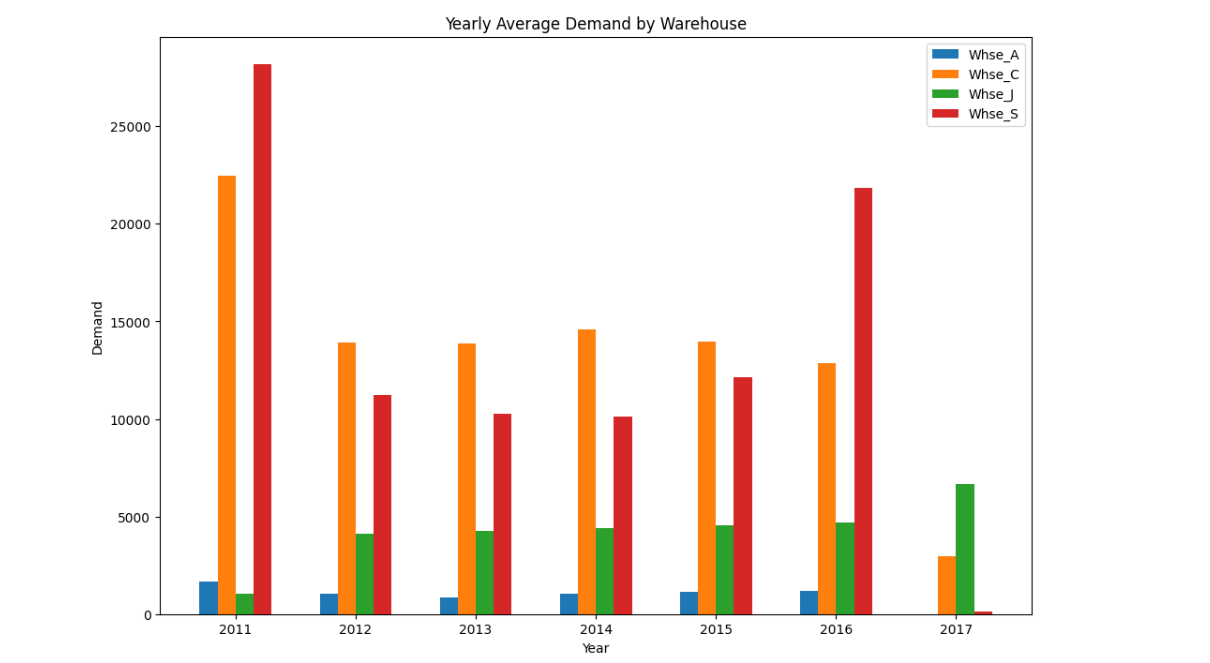
Percentage of missing values: 1.0718355863910547

| Product\_Code | Warehouse | Product\_Category | Order\_Demand |
| --- | --- | --- | --- |
| Date |  |  |  |  |
| 2011-01-08 | Product\_0965 | Whse\_A | Category\_006 | 2.0 |
| 2011-05-31 | Product\_1724 | Whse\_A | Category\_003 | 108.0 |
| 2011-06-24 | Product\_1521 | Whse\_S | Category\_019 | 85000.0 |
| 2011-06-24 | Product\_1521 | Whse\_S | Category\_019 | 7000.0 |
| 2011-09-02 | Product\_1507 | Whse\_C | Category\_019 | 1250.0 |

Text(0.5, 1.02, 'Yearly Average Demand for all the Product Categories')







Conclusion:

Data loading and preprocessing the data set done successfully.