|  |
| --- |
| DEPI |
| Data Analysis |
| Final Project |

|  |
| --- |
| Hazem Hassan Ahmed Hassan  Mohamed Ahmed Mohamed Fetyan  Sayed Ahmed Sayed Elbaradei  Ebraam Girgis Shenouda Boules  Abdelshafi Saeed Abdelshafi Elsayyed  10-22-2024 |

# Outline

Table of Contents

[Outline 1](#_Toc180530596)

[Week 1: Data Cleaning and Preprocessing. 2](#_Toc180530597)

[Objective 2](#_Toc180530598)

[SQL Task: Explore the database schema 2](#_Toc180530599)

[Python task: Data cleaning & Transformation 3](#_Toc180530600)

[Week 2: Determine Data Analysis Questions 7](#_Toc180530601)

[Explore Possible Business Questions 7](#_Toc180530602)

[Formulate Analysis Questions for AdventureWorks 2019 7](#_Toc180530603)

[SQL Queries: 8](#_Toc180530604)

[Week 3: Sales Forecasting with SalesOrderHeader 14](#_Toc180530605)

[Objectives 14](#_Toc180530606)

[Week 4: Visualization 16](#_Toc180530607)

# Week 1: Data Cleaning and Preprocessing.

Objective**:**

Build a data model, clean, transform and preprocess the data.

SQL Task: Explore the database schema.

* Identify the key tables relevant to your analysis (e.g., Sales, Customers, Products).
* Use SELECT \* FROM to examine the structure and data of key tables.
* Identify relationships between tables.
* Create Diagram.

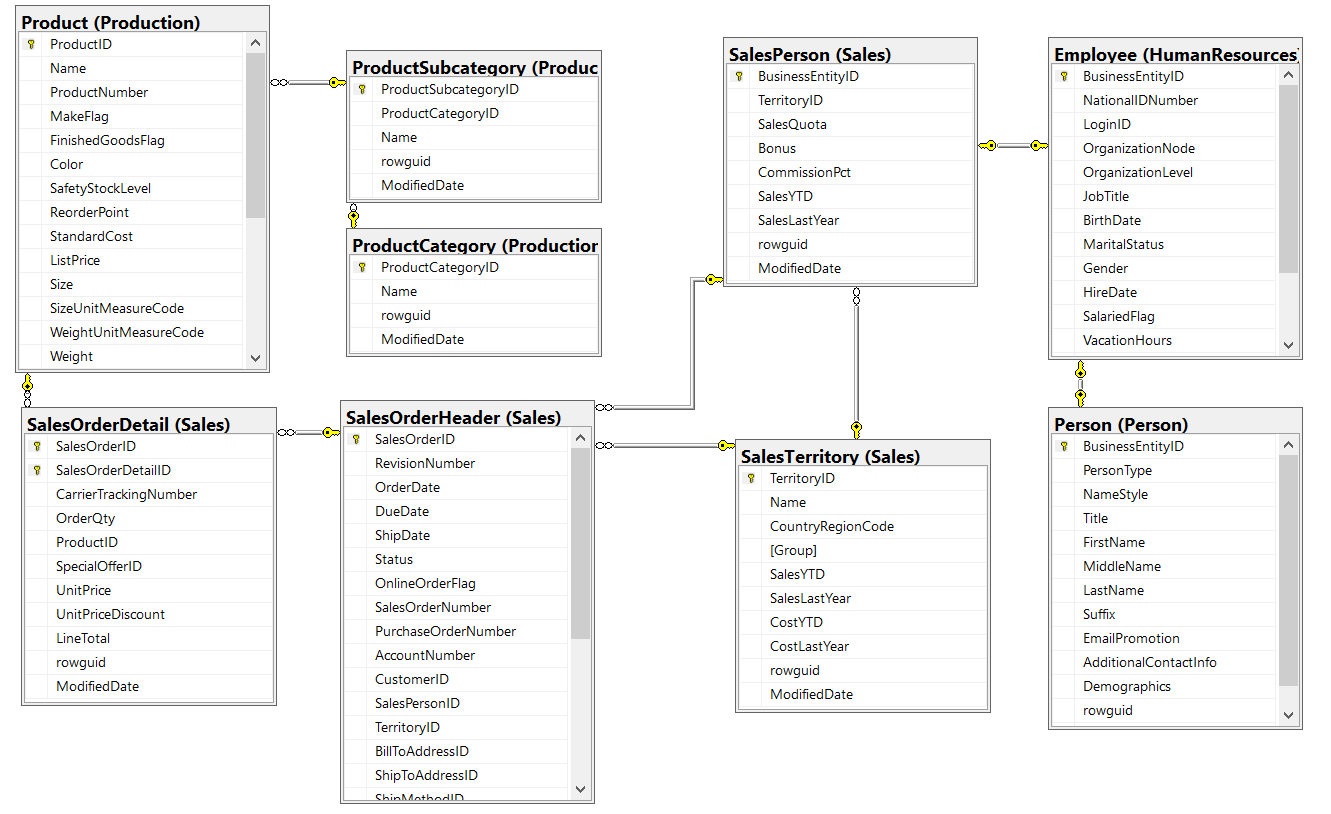


Figure -Project Diagram

* Relations Table:

|  |  |  |
| --- | --- | --- |
| Table 1 | Table 2 | Relation |
| Product | Product Subcategory | M:1 |
| Product Subcategory | Product Category | M:1 |
| Sales Order Detail | Product | M:1 |
| Sales Order Detail | Sales Order Header | M:1 |
| Sales Order Header | Sales Territory | M:1 |
| Sales Order Header | Sales Person | M:1 |
| Sales Person | Sales Territory | M:1 |
| Sales Person | Employee | 1:1 |
| Employee | Person | 1:1 |

Table - Relation Table

## Python task: Data cleaning & Transformation

1. Use pyodbc library to connect python with sql server, import pandas for data cleaning

|  |
| --- |
| import pyodbc  import pandas as pd |

1. Start connection

|  |
| --- |
| Cnxn\_str = ("Driver={ODBC Driver 17 for SQL Server};"  "Server=DESKTOP-CK5GPC9\MSSQLSERVER01;"  "Database=AdventureWorks2019;"  "Trusted\_Connection=yes;")  cnxn = pyodbc.connect(cnxn\_str) |

1. Load data to pandas data frame

|  |
| --- |
| SalesPerson = pd.read\_sql("Select \* From Sales.Salesperson", cnxn)  ProductCategory = pd.read\_sql("Select \* From Production.ProductCategory", cnxn)  ProductSubcategory = pd.read\_sql("Select \* From Production.ProductSubcategory", cnxn)  Product = pd.read\_sql("Select \* From Production.Product", cnxn)  SalesOrderDetail = pd.read\_sql("Select \* From Sales.SalesOrderDetail", cnxn)  SalesOrderHeader = pd.read\_sql("Select \* From Sales.SalesOrderHeader", cnxn)  SalesTerritory = pd.read\_sql("Select \* From Sales.SalesTerritory", cnxn)  Person = pd.read\_sql("Select \* From Person.Person", cnxn) |

1. Data Cleaning & Transformation

|  |
| --- |
| # Check Missing Values  SalesOrderDetail.isnull().sum()  SalesOrderHeader.isnull().sum()  Product.isnull().sum()  ProductCategory.isnull().sum()  ProductSubcategory.isnull().sum()  SalesTerritory.isnull().sum()  Person.isnull().sum()  SalesPerson.isnull().sum() |
| # Check Data types  SalesOrderDetail.info()  SalesOrderHeader.info()  Product.info()  ProductCategory.info()  ProductSubcategory.info()  SalesTerritory.info()  Person.info()  SalesPerson.info() |
| # Drop Columns  del SalesOrderDetail['CarrierTrackingNumber']  del SalesOrderDetail['rowguid']  del SalesOrderDetail['ModifiedDate']  del SalesOrderHeader['rowguid']  del SalesOrderHeader['ModifiedDate']  del Product['rowguid']  del Product['ModifiedDate']  del ProductCategory['rowguid']  del ProductCategory['ModifiedDate']  del ProductSubcategory['rowguid']  del ProductSubcategory['ModifiedDate']  del SalesTerritory['rowguid']  del SalesTerritory['ModifiedDate']  del Person['rowguid']  del Person['ModifiedDate']  del SalesPerson['rowguid']  del SalesPerson['ModifiedDate']  del SalesOrderHeader['Comment']  del Product['DiscontinuedDate']  SalesOrderHeader.drop(columns = ['CreditCardID','CreditCardApprovalCode'], inplace = True)  Product.drop(columns = ['Color','Size','SizeUnitMeasureCode','WeightUnitMeasureCode','Weight'], inplace = True) |
| # New Columns  Product['Profit'] = Product.ListPrice - Product.StandardCost  Product['Profit Margin'] = (Product.Profit/Product.ListPrice)\*100  SalesOrderHeader['DeliveryTime'] = (SalesOrderHeader['ShipDate'] - SalesOrderHeader['OrderDate']).dt.days |

1. Check outliers

|  |
| --- |
| Product.boxplot(column = ['ListPrice']) |

1. Data validation

|  |
| --- |
| SalesOrderDetail = SalesOrderDetail[SalesOrderDetail['OrderQty']>0]  SalesOrderDetail = SalesOrderDetail[SalesOrderDetail['UnitPrice']>0] |

1. Exporting Data

|  |
| --- |
| SalesOrderDetail.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\SalesOrderDetail.csv')  SalesOrderHeader.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\SalesOrderHeader.csv')  Product.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\Product.csv')  ProductCategory.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\ProductCategory.csv')  ProductSubcategory.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\ProductSubcategory.csv')  SalesTerritory.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\SalesTerritory.csv')  Person.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\Person.csv')  SalesPerson.to\_csv('D:\Data Analyst\Final Project\Cleaned Data\SalesPerson.csv') |

# Week 2: Determine Data Analysis Questions

**Objective**: Identify key analysis questions based on the dataset that can provide insights for decision-makers.

## Explore Possible Business Questions

Think about the kind of insights a business would want from the AdventureWorks 2019 data. Some common areas could be:

* **Sales Trends**: How have sales performed over time?
* **Customer Insights**: Who are the top customers, and what is their purchasing behavior?
* **Product Performance**: Which products or categories are generating the most revenue?
* **Salesperson Performance**: Which salespeople are performing well?
* **Geographical Insights**: How does sales performance vary across regions?

## Formulate Analysis Questions for AdventureWorks 2019

Based on your exploration of the dataset, start forming questions such as:

1. **Sales and Revenue Analysis**:

* What is the total revenue generated by different product categories?
* What is the trend of sales orders over the last year?
* What is the trend of sales orders over 2 years?
* Which months or quarters see the highest sales activity?
* Sales performance difference from online and instore.
* Discount or offers relation to sales.

1. **Customer Insights**:

* How does customer spending vary by region or country?
* What is the average revenue per customer?
* Type of Customers and their orders value and volume.

1. **Product Analysis**:

* Which products have the highest sales volume?
* What is the relationship between product price and sales quantity?
* Average order size for each category.
* Which products which people often purchase together?

1. **Salesperson Performance**:

* Who is the top-performing salespeople based on revenue?
* What is the average sales performance per region by sales team?

1. **Regional/Geographical Insights**:

* What are the sales figures for each region/country?

## SQL Queries:

1. **Sales and Revenue Analysis**:

* What is the total revenue generated by different product categories?

|  |
| --- |
| SELECT PC.name AS productcategory, SUM(OrderQty\*UnitPrice) AS TotalRevenue  FROM Production.Product PD  INNER JOIN Production.ProductSubcategory PS ON PD.ProductSubcategoryID = PS.ProductSubcategoryID  INNER JOIN Production.ProductCategory PC ON PS.ProductCategoryID = PC.ProductCategoryID  INNER JOIN Sales.SalesOrderDetail SOD ON PD.ProductID = SOD.ProductID  GROUP BY PC.name  ORDER BY TotalRevenue desc  ; |

* What is the trend of sales orders over the last year?

|  |
| --- |
| SELECT YEAR(soh.OrderDate) AS Year, MONTH(soh.OrderDate) AS Month, COUNT(soh.SalesOrderID) AS TotalOrders  FROM Sales.SalesOrderHeader SOH  WHERE YEAR(soh.OrderDate) = 2014  GROUP BY YEAR(soh.OrderDate), MONTH(soh.OrderDate)  ORDER BY Year, Month  ; |

* What is the trend of sales orders over 2 years?

|  |
| --- |
| SELECT MIN(OrderDate), MAX(OrderDate)  FROM Sales.SalesOrderHeader  ;  WITH TopProducts AS (  SELECT TOP 3 SOD.ProductID, SUM(sod.LineTotal) AS TotalSales  FROM Sales.SalesOrderDetail SOD  JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID  WHERE YEAR(SOH.OrderDate) BETWEEN 2012 AND 2013  GROUP BY SOD.ProductID  ORDER BY TotalSales DESC  )  SELECT PD.Name AS ProductName, YEAR(SOH.OrderDate) AS SalesYear, MONTH(SOH.OrderDate) AS SalesMonth, SUM(SOD.LineTotal) AS MonthlySales  FROM Sales.SalesOrderDetail SOD  JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID  JOIN Production.Product PD ON SOD.ProductID = PD.ProductID  JOIN TopProducts TP ON SOD.ProductID = TP.ProductID  WHERE YEAR(SOH.OrderDate) BETWEEN 2012 AND 2013  GROUP BY PD.Name, YEAR(SOH.OrderDate), MONTH(SOH.OrderDate)  ORDER BY PD.Name, SalesYear, SalesMonth  ; |

* Which months or quarters see the highest sales activity?

|  |
| --- |
| SELECT  YEAR(soh.OrderDate) AS Year, DATEPART(QUARTER, soh.OrderDate) AS Quarter,  SUM(sod.LineTotal) AS TotalRevenue  FROM Sales.SalesOrderHeader soh  JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID  GROUP BY YEAR(soh.OrderDate), DATEPART(QUARTER, soh.OrderDate)  ORDER BY TotalRevenue DESC  ; |

* Sales performance difference from online and instore.

|  |
| --- |
| SELECT  CASE  WHEN OnlineOrderFlag = 1 THEN 'Online Store'  ELSE 'Physical Store'  END AS StoreType,  COUNT(SalesOrderID) AS TotalOrders, SUM(TotalDue) AS TotalSales, AVG(TotalDue) AS AverageOrderValue  FROM Sales.SalesOrderHeader  GROUP BY OnlineOrderFlag  ; |

* Discount or offers relation to sales.

|  |
| --- |
| SELECT  CASE  WHEN SOD.UnitPriceDiscount > 0 THEN 'Discounted'  ELSE 'Non-Discounted'  END AS DiscountType,  SUM(SOD.LineTotal) AS TotalSales, COUNT(SOH.SalesOrderID) AS TotalOrders, AVG(SOD.LineTotal) AS AverageOrderValue  FROM Sales.SalesOrderDetail SOD  JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID  GROUP BY CASE  WHEN SOD.UnitPriceDiscount > 0 THEN 'Discounted'  ELSE 'Non-Discounted'  END  ORDER BY DiscountType  ; |

1. **Customer Insights**:

* How does customer spending vary by region or country?

|  |
| --- |
| SELECT st.Name AS TerritoryName, SUM(sod.LineTotal) AS TotalSpent  FROM Sales.SalesOrderHeader soh  JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID  JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID  GROUP BY st.Name  ORDER BY TotalSpent DESC  ; |

* What is the average revenue per customer?

|  |
| --- |
| SELECT AVG(CustomerRevenue.TotalSpent) AS AvgRevenuePerCustomer  FROM (  SELECT soh.CustomerID, SUM(sod.LineTotal) AS TotalSpent  FROM Sales.SalesOrderHeader soh  JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID  GROUP BY soh.CustomerID  ) AS CustomerRevenue  ; |

* Type of Customers and their orders value and volume.

|  |
| --- |
| WITH CustomerOrders AS (  SELECT soh.CustomerID, COUNT(soh.SalesOrderID) AS OrderCount  FROM Sales.SalesOrderHeader soh  GROUP BY soh.CustomerID  )  SELECT  CASE  WHEN co.OrderCount = 1 THEN 'One-Time Customer'  WHEN co.OrderCount > 1 AND co.OrderCount<= 10 THEN 'Occasional Customer'  ELSE 'Frequent Customer'  END AS CustomerType,  COUNT(co.CustomerID) AS CustomerVolume, SUM(co.OrderCount) AS OrderVolume  FROM CustomerOrders co  GROUP BY CASE  WHEN co.OrderCount = 1 THEN 'One-Time Customer'  WHEN co.OrderCount > 1 AND co.OrderCount<= 10 THEN 'Occasional Customer'  ELSE 'Frequent Customer'  END  ORDER BY OrderVolume DESC  ; |

1. **Product Analysis**:

* Which products have the highest sales volume?

|  |
| --- |
| SELECT TOP 10 p.Name AS ProductName, SUM(sod.OrderQty) AS TotalQuantitySold  FROM Sales.SalesOrderDetail sod  JOIN Production.Product p ON sod.ProductID = p.ProductID  GROUP BY p.Name  ORDER BY TotalQuantitySold DESC  ; |

* What is the relationship between product price and sales quantity?

|  |
| --- |
| SELECT p.Name AS ProductName, p.ListPrice AS UnitPrice, SUM(sod.OrderQty) AS TotalQuantitySold  FROM Sales.SalesOrderDetail sod  JOIN Production.Product p ON sod.ProductID = p.ProductID  GROUP BY p.Name, p.ListPrice  ORDER BY UnitPrice; |

* Average order size for each category.

|  |
| --- |
| SELECT PC.Name, AVG(SOD.OrderQty) as Average\_Order\_Size  FROM Sales.SalesOrderDetail SOD  JOIN Production.Product P ON SOD.ProductID = P.ProductID  JOIN Production.ProductSubcategory PS ON P.ProductSubcategoryID = PS.ProductSubcategoryID  JOIN Production.ProductCategory PC ON PS.ProductCategoryID = PC.ProductCategoryID  GROUP BY Pc.Name  ORDER BY Average\_Order\_Size DESC  ; |

* Which products which people often purchase together?

|  |
| --- |
| WITH OrderProducts AS (  SELECT sod.SalesOrderID, sod.ProductID  FROM Sales.SalesOrderDetail sod  )  SELECT TOP 10 p1.Name AS Product1, p2.Name AS Product2, COUNT(\*) AS PairCount  FROM OrderProducts op1  JOIN OrderProducts op2 ON op1.SalesOrderID = op2.SalesOrderID  JOIN Production.Product p1 ON op1.ProductID = p1.ProductID  JOIN Production.Product p2 ON op2.ProductID = p2.ProductID  WHERE op1.ProductID < op2.ProductID -- Avoid duplicate and self-pairing  GROUP BY p1.Name, p2.Name  ORDER BY PairCount DESC  ; |

1. **Salesperson Performance**:

* Who are the top-performing salespeople based on revenue?

|  |
| --- |
| SELECT TOP 3 sp.BusinessEntityID AS SalespersonID, p.FirstName, p.LastName, SUM(sod.LineTotal) AS TotalRevenue  FROM Sales.SalesPerson sp  JOIN HumanResources.Employee e ON sp.BusinessEntityID = e.BusinessEntityID  JOIN Person.Person p ON e.BusinessEntityID = p.BusinessEntityID  JOIN Sales.SalesOrderHeader soh ON sp.BusinessEntityID = soh.SalesPersonID  JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID  GROUP BY sp.BusinessEntityID, p.FirstName, p.LastName  ORDER BY TotalRevenue DESC  ; |

* What is the average sales performance per region by sales team?

|  |
| --- |
| SELECT st.Name AS TerritoryName, AVG(sod.LineTotal) AS AvgRevenuePerTerritory  FROM Sales.SalesOrderHeader soh  JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID  JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID  GROUP BY st.Name  ORDER BY AvgRevenuePerTerritory DESC  ; |

* Top performing sales person in each area.

|  |
| --- |
| SELECT TOP 5 SP.BusinessEntityID, P.FirstName, P.LastName, ST.Name AS Territory\_Name, SUM(sod.LineTotal) AS Total\_Sales  FROM Sales.SalesOrderHeader SOH  JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID = SOD.SalesOrderID  JOIN Sales.SalesPerson SP ON SP.BusinessEntityID = SOH.SalesPersonID  JOIN Sales.SalesTerritory ST ON SOH.TerritoryID = ST.TerritoryID  JOIN Person.Person p ON SP.BusinessEntityID = P.BusinessEntityID  GROUP BY SP.BusinessEntityID, P.FirstName, P.LastName, ST.Name  ORDER BY Total\_Sales DESC  ; |

1. **Regional/Geographical Insights**:

* What are the sales figures for each region/country?

|  |
| --- |
| SELECT st.Name AS TerritoryName, SUM(sod.LineTotal) AS TotalSales  FROM Sales.SalesOrderHeader soh  JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID  JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID  GROUP BY st.Name  ORDER BY TotalSales DESC  ; |

# Week 3: Sales Forecasting with SalesOrderHeader

Objectives:

* Extracting and preprocessing sales data from the SalesOrderHeader table.
* Grouping the data by date to analyze monthly or quarterly trends.
* Building a forecasting model using Python to predict future sales trends.

1. Extract Data from SalesOrderHeader Table

|  |
| --- |
| sales\_data = pd.read\_sql("Select OrderDate,TotalDue From Sales.SalesOrderHeader", cnxn) |

1. Preprocess the Data

|  |
| --- |
| sales\_data['OrderDate'] = pd.to\_datetime(sales\_data['OrderDate'])  sales\_data['Year']=sales\_data['OrderDate'].dt.year  sales\_data['month']=sales\_data['OrderDate'].dt.month  monthly\_sales = sales\_data.groupby([sales\_data['Year'], sales\_data['month']])['TotalDue'].sum().reset\_index()  monthly\_sales['Time'] = monthly\_sales['Year'] \* 12 + monthly\_sales['month'] |

1. Build a Forecasting Model

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LinearRegression  import matplotlib.pyplot as plt  # Features (Time) and target (TotalSales)  X = monthly\_sales[['Time']]  y = monthly\_sales['TotalSales']  # Split the data into training and testing sets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, shuffle=False)  # Create and train the linear regression model  model = LinearRegression()  model.fit(X\_train, y\_train) |
| # Forecast for the next 3 months  future\_time = pd.DataFrame({'Time': [X\_train['Time'].max() + i for i in range(1, 4)]})  future\_sales = model.predict(future\_time) |
| # Plot the historical sales and the forecasted sales  plt.plot(monthly\_sales['Time'], monthly\_sales['TotalSales'], label='Historical Sales')  plt.plot(future\_time['Time'], future\_sales, label='Forecasted Sales', linestyle='--', marker='o')  plt.xlabel('Time (Year-Month)')  plt.ylabel('Total Sales')  plt.legend()  plt.title('Sales Forecast for the Next 3 Months')  plt.show() |

1. Output

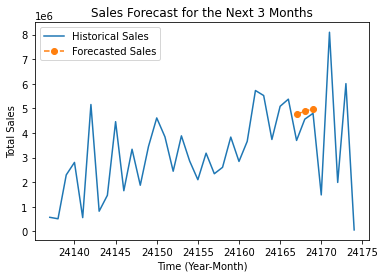


Figure - Sales Forecast for next 3 Months

# Week 4: Visualization

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generatedA screenshot of a computer screen

Description automatically generated