Text

Description automatically generated

**Activity based**

**Project Report on**

**AI Business Intelligence**

**Project Phase - II**

**Submitted to Vishwakarma University, Pune**

**Under the Initiative of**

**Contemporary Curriculum, Pedagogy, and Practice (C2P2)**

**By**

**SAKSHI TOSNIWAL**

**SRN No : 202101398**

**Roll No : 30**

**Div : E**

**Third Year Engineering**

**Faculty In charge:- Prof. Moumita Pal**

**Date Of Project Phase 1:- 16 February 2024**

**Department of Computer Engineering**

**Faculty of Science and Technology**

**Academic Year**

**2023-2024 Term-II**

**Business Intelligence : Phase II**

**Project Name : TREND ANALYSIS IN RETAIL SALES**

### ****Introduction:****

### **In the dynamic landscape of retail sales, staying ahead of market trends is paramount for sustained success. Business Intelligence (BI) serves as a powerful tool in this endeavor, offering retailers a systematic approach to analyze and interpret data trends. Trend analysis within retail sales, a key component of BI, involves the examination of historical sales data to identify patterns, fluctuations, and emerging consumer behaviors. By leveraging advanced analytics and machine learning algorithms, retailers can uncover valuable insights into customer preferences, product performance, seasonal variations, and market trends. This proactive approach enables retailers to make data-driven decisions, optimize inventory management, enhance marketing strategies, and ultimately drive revenue growth. Moreover, trend analysis in retail sales empowers businesses to anticipate market shifts, mitigate risks, and capitalize on emerging opportunities in a competitive marketplace. In essence, incorporating trend analysis into the realm of business intelligence equips retailers with the foresight and agility necessary to thrive in an ever-evolving retail landscape.**

### Problem Statement

The rapid pace of change in consumer preferences, market dynamics, and technological advancements further complicates the landscape, requiring retailers to adapt quickly to stay competitive. Thus, there is a pressing need for robust BI solutions that can effectively aggregate, analyze, and visualize data to provide timely and accurate insights. Addressing these challenges will enable retailers to enhance their decision-making processes, optimize resource allocation, and ultimately drive business growth in an increasingly complex retail environment.

### Objective

**Identifying Consumer Preferences:** One primary objective of trend analysis in retail sales is to pinpoint evolving consumer preferences and behaviors. By analyzing historical sales data, retailers can detect patterns and trends in product demand, enabling them to anticipate future consumer needs. This insight empowers retailers to tailor their product offerings, marketing strategies, and inventory management to better align with customer expectations, ultimately driving customer satisfaction and loyalty.

### ****Existing Work Done:****

In this project, we undertake exploratory data analysis (EDA) on the Social Media Sentiment dataset. The process includes:

1. **Data Collection:** Collect social media data from various platforms, including customer reviews, comments, and mentions.

2. **Data Preprocessing:** Preprocess the social media data to handle noise, irrelevant information, and ensure data consistency. Clean text data, handle emotions, and address any data quality issues that may affect sentiment analysis.

3. **Feature Engineering:** Identify and engineer features that contribute to sentiment analysis. This may include sentiment scores, sentiment trends over time, and the identification of key topics or keywords associated with positive or negative sentiments.

4. **BI Dashboard Development**: Design and implement a user-friendly BI dashboard that visualizes key sentiment analysis metrics. Include components for monitoring overall sentiment trends, identifying sentiment influencers, and assessing the impact of marketing campaigns.

### ****Current Assessments:****

1. Implementation of Advanced Analytics: Invest in advanced analytical tools and technologies such as machine learning and predictive analytics to enhance the accuracy and effectiveness of trend analysis, enabling proactive decision-making and personalized customer experiences.
2. Integration of Big Data Sources: Expand data collection efforts to include a wide range of sources, including point-of-sale transactions, online interactions, social media data, and IoT devices, to gain comprehensive insights into consumer behavior and market trends.
3. Focus on Sustainability and Social Responsibility: Incorporate trend analysis into sustainability initiatives and ethical sourcing practices, leveraging insights to identify and capitalize on trends related to sustainable products and socially responsible practices, meeting consumer demands for eco-friendly and ethically sourced goods.
4. Investment in Talent and Training: Prioritize investment in talent development and training programs to equip employees with the necessary skills and expertise to effectively utilize trend analysis tools and technologies, fostering a data-driven culture within the organization.

### ****Further Assessments:****

### ****Reporting : Creating a Report on the Overall Project****

### Execution Part:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import LabelEncoder

# Step 1: Read the dataset

data = pd.read\_excel('data.xlsx', sheet\_name='Data')

# Step 2: Preprocess the 'Order Date' column

data['Order Date'] = pd.to\_datetime(data['Order Date'])

data['Order\_Year'] = data['Order Date'].dt.year

data['Order\_Month'] = data['Order Date'].dt.month

data['Order\_Day'] = data['Order Date'].dt.day

data['Order\_DayOfWeek'] = data['Order Date'].dt.dayofweek

# Step 3: Encode categorical columns

encoder = LabelEncoder()

data['Category\_Encoded'] = encoder.fit\_transform(data['Category'])

data['Sub-Category\_Encoded'] = encoder.fit\_transform(data['Sub-Category'])

# Step 4: One-hot encode the 'Product ID' column

data = pd.get\_dummies(data, columns=['Product ID'], prefix='ProductID')

# Step 5: Select relevant features and target variable

features = ['Year', 'Order\_Year', 'Order\_Month', 'Order\_Day', 'Order\_DayOfWeek', 'Category\_Encoded', 'Sub-Category\_Encoded', 'Sales', 'Quantity', 'Discount']

X = data[features]  # Features

y = data['Sales']  # Target variable

# Step 6: 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, random\_state=42)

# Step 7: Train the regression model

model = RandomForestRegressor()

model.fit(X\_train, y\_train)

# Step 8: Make predictions on the testing set

predictions = model.predict(X\_test)

# Step 9: Evaluate the model

mse = mean\_squared\_error(y\_test, predictions)

print(f'Mean Squared Error: {mse}')

import matplotlib.pyplot as plt

import seaborn as sns

# Plot the distribution of the target variable 'Sales'

plt.figure(figsize=(10, 6))

sns.histplot(data['Sales'], bins=30, kde=True, color='skyblue')

plt.title('Distribution of Sales')

plt.xlabel('Sales')

plt.ylabel('Frequency')

plt.show()

### 

1. Data Loading and Preprocessing:

* The dataset is read from an Excel file named 'data.xlsx' and a specific sheet named 'Data'.
* The 'Order Date' column is converted to datetime format, and additional features like 'Order\_Year', 'Order\_Month', 'Order\_Day', and 'Order\_DayOfWeek' are extracted from it.
* Categorical columns like 'Category' and 'Sub-Category' are encoded using LabelEncoder.
* The 'Product ID' column is one-hot encoded.

1. Feature Selection:

* Relevant features including 'Year', 'Order\_Year', 'Order\_Month', 'Order\_Day', 'Order\_DayOfWeek', 'Category\_Encoded', 'Sub-Category\_Encoded', 'Sales', 'Quantity', and 'Discount' are selected.

1. Data Splitting:

* The data is split into training and testing sets using train\_test\_split function from sklearn.

1. Model Training:

* A RandomForestRegressor model is initialized and trained on the training data.

1. Model Evaluation:

* Predictions are made on the testing set, and mean squared error (MSE) is calculated to evaluate the model's performance.

1. Visualization:

* A histogram is plotted to visualize the distribution of the target variable 'Sales'.

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import LabelEncoder

# Step 1: Read the dataset

data = pd.read\_excel('data.xlsx', sheet\_name='Data')

# Step 2: Preprocess the 'Order Date' column

data['Order Date'] = pd.to\_datetime(data['Order Date'])

data['Order\_Year'] = data['Order Date'].dt.year

data['Order\_Month'] = data['Order Date'].dt.month

data['Order\_Day'] = data['Order Date'].dt.day

data['Order\_DayOfWeek'] = data['Order Date'].dt.dayofweek

# Step 3: Encode categorical columns

encoder = LabelEncoder()

data['Category\_Encoded'] = encoder.fit\_transform(data['Category'])

data['Sub-Category\_Encoded'] = encoder.fit\_transform(data['Sub-Category'])

# Step 4: One-hot encode the 'Product ID' column

data = pd.get\_dummies(data, columns=['Product ID'], prefix='ProductID')

# Step 5: Select relevant features and target variable

features = ['Year', 'Order\_Year', 'Order\_Month', 'Order\_Day', 'Order\_DayOfWeek', 'Category\_Encoded', 'Sub-Category\_Encoded', 'Sales', 'Quantity', 'Discount']

X = data[features]  # Features

y = data['Sales']  # Target variable

# Step 6: 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, random\_state=42)

# Step 7: Train the regression model

model = RandomForestRegressor()

model.fit(X\_train, y\_train)

# Step 8: Make predictions on the testing set

predictions = model.predict(X\_test)

# Step 9: Evaluate the model

mse = mean\_squared\_error(y\_test, predictions)

print(f'Mean Squared Error: {mse}')

import matplotlib.pyplot as plt

import seaborn as sns

# Plot the distribution of the target variable 'Sales'

plt.figure(figsize=(10, 6))

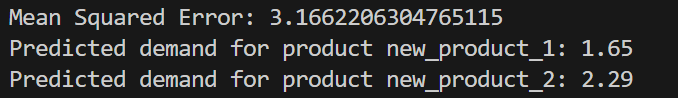
sns.histplot(data['Sales'], bins=30, kde=True, color='skyblue')

plt.title('Distribution of Sales')

plt.xlabel('Sales')

plt.ylabel('Frequency')

plt.show()



1. Data Loading: Reads the dataset from an Excel file named 'data.xlsx' and selects a specific sheet named 'Data'.
2. Data Preprocessing:

* Converts the 'Order Date' column to datetime format.
* Extracts additional features like 'Order\_Year', 'Order\_Month', 'Order\_Day', and 'Order\_DayOfWeek' from the 'Order Date' column.
* Encodes categorical columns ('Category' and 'Sub-Category') using LabelEncoder.
* One-hot encodes the 'Product ID' column.

1. Feature Selection: Selects relevant features including 'Year', 'Order\_Year', 'Order\_Month', 'Order\_Day', 'Order\_DayOfWeek', 'Category\_Encoded', 'Sub-Category\_Encoded', 'Sales', 'Quantity', and 'Discount'.
2. Data Splitting: Splits the dataset into training and testing sets using a 80-20 ratio.
3. Model Training: Initializes and trains a RandomForestRegressor model using the training data.
4. Model Evaluation: Makes predictions on the testing set and calculates the mean squared error (MSE) to evaluate the model's performance.
5. Visualization: Plots a histogram to visualize the distribution of the target variable 'Sales'.

Product

import pandas as pd

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

# Step 1: Read the dataset

data = pd.read\_excel('data.xlsx', sheet\_name='Data')

# Step 2: Select relevant features and target variable

features = ['Product ID', 'Category', 'Sub-Category', 'Sales']

X = data[features]  # Features

y = data['Quantity']  # Target variable

# Step 3: Preprocess categorical variables (if needed)

X\_encoded = pd.get\_dummies(X)  # One-hot encoding for categorical variables

# Step 4: Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_encoded, y, test\_size=0.2, random\_state=42)

# Step 5: Train the model

model = RandomForestRegressor()

model.fit(X\_train, y\_train)

# Step 6: Make predictions on the testing set

predictions = model.predict(X\_test)

# Step 7: Evaluate the model

mse = mean\_squared\_error(y\_test, predictions)

print(f'Mean Squared Error: {mse}')

# Step 8: Predict future demand

# Assuming you have new data for future sales

new\_data = pd.DataFrame({

    'Product ID': ['new\_product\_1', 'new\_product\_2'],  # Example product IDs for new products

    'Category': ['New Category', 'New Category'],  # Example category for new products

    'Sub-Category': ['New Sub-Category', 'New Sub-Category'],  # Example sub-category for new products

    'Sales': [100, 200]  # Example sales data for new products

})

# Preprocess the new data similarly to the training data

new\_data\_encoded = pd.get\_dummies(new\_data)  # One-hot encoding for categorical variables

# Align the columns of new\_data\_encoded with X\_encoded

new\_data\_encoded\_aligned = new\_data\_encoded.reindex(columns=X\_encoded.columns, fill\_value=0)

# Predict the future demand using the trained model

predicted\_demand = model.predict(new\_data\_encoded\_aligned)

# Print the predicted demand for each new product

for i, demand in enumerate(predicted\_demand):

    print(f"Predicted demand for product {new\_data['Product ID'][i]}: {demand}")

1. Data Loading: Reads the dataset from an Excel file named 'data.xlsx' and selects a specific sheet named 'Data'.
2. Feature Selection: Selects relevant features including 'Product ID', 'Category', 'Sub-Category', and 'Sales' as features, and 'Quantity' as the target variable.
3. Data Preprocessing:
4. One-hot encodes categorical variables in the feature set using pandas' get\_dummies function.
5. Data Splitting: Splits the dataset into training and testing sets using a 80-20 ratio.
6. Model Training: Initializes and trains a RandomForestRegressor model using the training data.
7. Model Evaluation: Makes predictions on the testing set and calculates the mean squared error (MSE) to evaluate the model's performance.
8. Future Demand Prediction:

* Creates new data for future sales, including product IDs, categories, sub-categories, and sales data.
* Preprocesses the new data similarly to the training data by one-hot encoding categorical variables.
* Aligns the columns of the new data with the training data to ensure consistency.
* Predicts the future demand for each new product using the trained model.
* Prints the predicted demand for each new product.

