Summary Of Task 1 (Data Cleaning And Preparation)

■ Software Used = Jupyter Notebook

Step 1: Load the Dataset

- Used pandas to load the CSV file into a DataFrame.
- Previewed the top rows to understand the structure.

Step 2: Understand the Dataset

- Checked the shape (9800 rows × 18 columns).
- Inspected column names, data types, and summary statistics.
- Identified the presence of missing values and data types (numeric, text, dates, etc.).

Step 3: Handle Missing Values

- Found that only postal_code had missing values.
- Filled missing values in postal code with the most frequent value (mode).
- Verified there were no missing values left using .isnull().sum().

Step 4: Remove Duplicates

- Checked for duplicate rows using df.duplicated().sum().
- Removed all duplicates with drop duplicates().

Step 5: Clean Text Columns

- Replaced spaces with underscores in string columns for consistency.
- Cleaned column headers: made lowercase and replaced spaces with underscores.

Step 6: Convert Data Types

- Converted order_date and ship_date to datetime format.
- Converted postal_code to integer after filling missing values.
- Converted categorical columns like segment, category, region, etc. to category dtype for better performance.

Step 7: Handle Outliers

- Detected outliers in the sales column using the IQR method.
- Removed rows with sales values outside the range of $1.5 \times IQR$.
- Created a new cleaned dataset df_cleaned.

Step 8: Encode Categorical Variables

- Identified text-based columns and applied One-Hot Encoding using pd.get dummies().
- This converted categorical columns to numeric format suitable for ML models.

Step 9: Normalize Numerical Columns

- Selected numeric columns and applied Standard Scaling using StandardScaler.
- This gave all numeric columns a mean of 0 and standard deviation of 1.

Final Result:

- Cleaned and prepared dataset with:
- No missing or duplicate values
- All columns correctly typed
- No outliers in sales
- All categorical data encoded
- All numerical data scaled

CODE

```
## Step 1: Import Required Libraries
```python
import pandas as pd
Step 2: Load the Dataset
df = pd.read_csv('path/to/train.csv')
df.head() # Preview the data
Step 3: Basic
Exploration # Shape of the
dataset print("Shape:",
df.shape)
Data types and column info
df.info()
Summary statistics
df.describe(include='all')
Step 4: Missing Values Analysis
missing = df.isnull().sum()
missing percent = (missing / len(df)) * 100
missing data = pd.DataFrame({'Missing Values': missing, 'Percent (%)': missing percent})
missing_data = missing_data[missing_data["Missing Values"] > 0]
missing_data.sort_values(by='Percent (%)', ascending=False)
```

```
Step 5: Fill Missing Values in 'postal code'
mode_postal = df['Postal Code'].mode()[0]
df['Postal Code'] = df['Postal Code'].fillna(mode postal)
Step 6: Remove Duplicate Rows
df.drop duplicates(inplace=True)
Step 7: Clean Text Columns and Column Headers
Replace spaces in string values with underscores
text cols = df.select dtypes(include='object').columns
for col in text cols:
 df[col] = df[col].str.replace(" ", "_", regex=False)
Clean column names
df.columns = df.columns.str.lower().str.strip().str.replace(' ', ' ')
Step 8: Convert Data Types
Convert date columns
df['order date'] = pd.to datetime(df['order date'], errors='coerce') df['ship date'] =
pd.to datetime(df['ship date'], errors='coerce')
Convert postal code to integer df['postal code']
= df['postal code'].astype(int)
Convert selected columns to category dtype
cat cols = ['ship mode', 'segment', 'country', 'city', 'state', 'region', 'category', 'sub-category'] for
col in cat cols:
 df[col] = df[col].astype('category')
Step 9: Handle Outliers in 'sales'
Q1 = df['sales'].quantile(0.25)
Q3 = df['sales'].quantile(0.75)
```

. .

# Remove outliers

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQRupper\_bound = Q3 + 1.5 \* IQR

df\_cleaned = df[(df['sales'] >= lower\_bound) & (df['sales'] <= upper\_bound)]

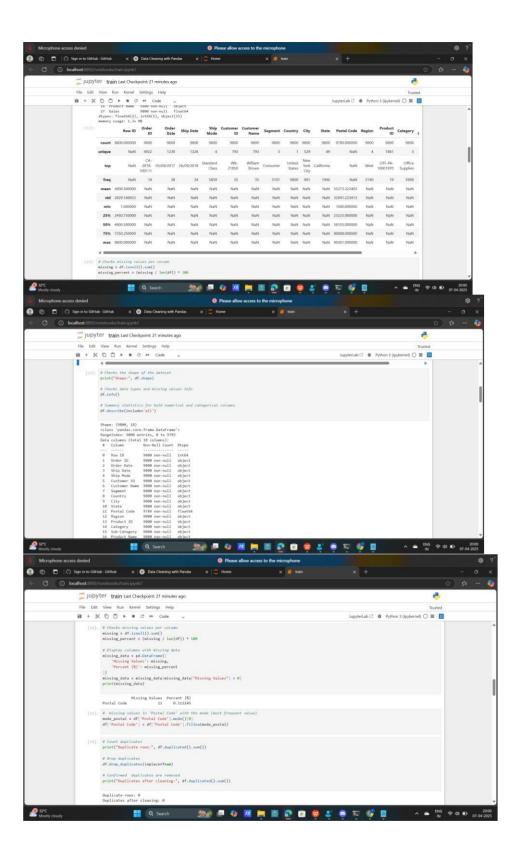
```
Step 10: Encode Categorical Variables (One-Hot Encoding)
categorical_cols = df_cleaned.select_dtypes(include='object').columns
df_encoded = pd.get_dummies(df_cleaned, columns=categorical_cols, drop_first=True)

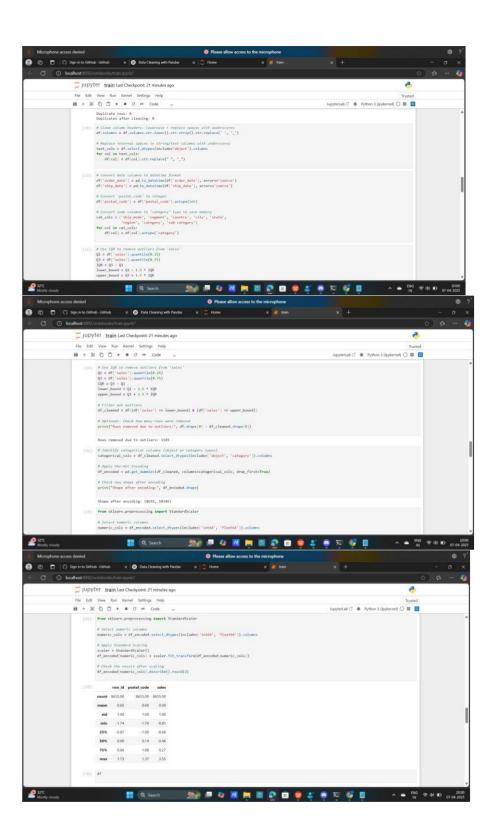
Step 11: Feature Scaling
from sklearn.preprocessing import StandardScaler

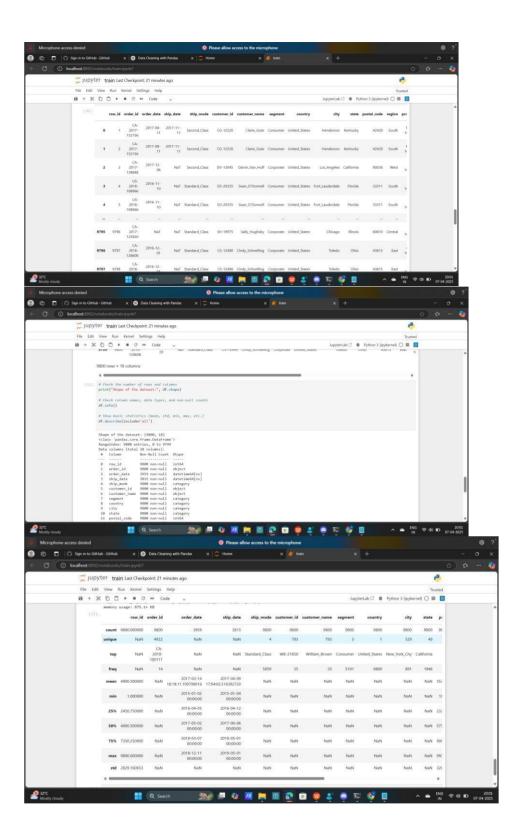
numeric_cols = df_encoded.select_dtypes(include=['float64', 'int64']).columns
scaler = StandardScaler()
```

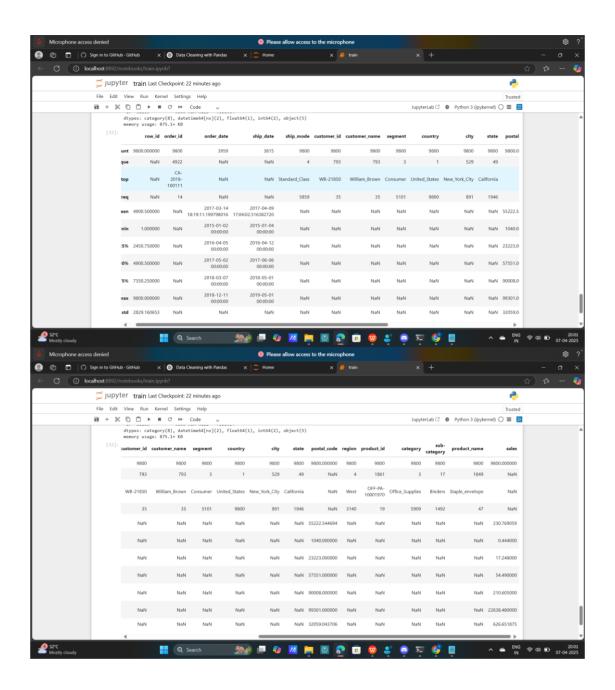
df encoded[numeric cols] = scaler.fit transform(df encoded[numeric cols])

,,,









# **END**