import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# Load the dataset (Replace 'your\_dataset.csv' with the actual file path)

df = pd.read\_csv("sales\_data.csv")

# Take a quick look at the data

print("First few rows of the dataset:")

print(df.head())

# --- DATA CLEANING ---

print("\nChecking for missing values...")

print(df.isnull().sum())

# Fill missing numerical values with the mean, and categorical values with the mode

df.fillna(df.mean(numeric\_only=True), inplace=True)

df.fillna(df.mode().iloc[0], inplace=True)

print("Missing values handled!")

# Remove any duplicate rows

df.drop\_duplicates(inplace=True)

print("Duplicates removed!")

# Convert Date column to a datetime format

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

print("Date column converted!")

# Detect and remove outliers using the IQR method

Q1 = df.quantile(0.25)

Q3 = df.quantile(0.75)

IQR = Q3 - Q1

outliers = ((df < (Q1 - 1.5 \* IQR)) | (df > (Q3 + 1.5 \* IQR)))

df = df[~outliers.any(axis=1)]

print("Outliers removed!")

# --- EXPLORATORY DATA ANALYSIS (EDA) ---

print("\nGenerating summary statistics...")

print(df.describe())

# Visualizing sales distribution

plt.figure(figsize=(8,5))

sns.histplot(df['Sales'], bins=30, kde=True)

plt.title('Sales Distribution')

plt.show()

# Checking for outliers with a boxplot

plt.figure(figsize=(8,5))

sns.boxplot(y=df['Sales'])

plt.title('Boxplot of Sales')

plt.show()

# Displaying correlation between features

plt.figure(figsize=(8,6))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title('Feature Correlation Heatmap')

plt.show()

# Relationship between Profit and Discount

plt.figure(figsize=(8,5))

sns.scatterplot(x=df['Discount'], y=df['Profit'])

plt.title('Profit vs Discount')

plt.show()

# Sales distribution by Region

plt.figure(figsize=(10,5))

sns.barplot(x=df['Region'], y=df['Sales'])

plt.xticks(rotation=45)

plt.title('Sales by Region')

plt.show()

# --- PREDICTIVE MODELING ---

print("\nPreparing data for the sales prediction model...")

X = df[['Profit', 'Discount']]

y = df['Sales']

# Splitting 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)

print("Data split into training and testing sets!")

# Train a Linear Regression Model

model = LinearRegression()

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

print("Model training complete!")

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate model performance

r2 = r2\_score(y\_test, y\_pred)

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

print(f'R² Score: {r2:.2f}')

print(f'Mean Squared Error: {mse:.2f}')

print("Model evaluation complete!")