Use Case 1: Performance of sales representatives

Team members:

VTU24468	BELUGULA NIHARIKA
VTU24469	GADDAM SUSMITHA
VTU24470	KOPPERLA SRI PRAGNA
VTU24471	DHRUV P
VTU24572	UPLANCHWAR BALAPRASAD
VTU24598	MYNENI CHARAN KUMAR CHOWDARY
VTU24646	RAJAMAHENDRAVARAPU SAI PADMINI
VTU24693	PENMETSA VINAY VARMA
VTU24734	KURUVA ASHOK KUMAR
VTU24763	CHINTAPENTA VENKATA SAI SASI SEKHAR

Aim:

To analyze and predict the performance of sales representatives using historical sales data. We will use a **Linear Regression** model to predict the total sales achieved by each representative based on their experience, number of calls, number of clients, and region.

Algorithm:

Step 1: Import necessary libraries (pandas, numpy, matplotlib, sklearn).

Step 2: Load or create a dataset containing sales representatives' data.

Step 3: Perform data preprocessing — handle missing values, encode categorical variables.

Step 4: Split the dataset into training and testing sets.

Step 5: Train a **Linear Regression** model on the training data.

Step 6: Predict the sales performance on test data.

Step 7: Evaluate model performance using R^2 score and Mean Squared Error.

Step 8: Visualize actual vs predicted sales.

Step 9: Display final model performance and interpret the result.

Python Code :

```
# Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

# Step 1: Create sample dataset
data = {
    'Experience_Years': [1, 3, 5, 7, 2, 10, 4, 6, 8, 9],
    'Calls_Made': [50, 80, 100, 150, 60, 200, 110, 120, 180, 160],
    'Clients_Handled': [5, 8, 10, 14, 6, 18, 11, 12, 15, 17],
    'Region': ['East', 'West', 'North', 'South', 'East', 'South', 'North', 'West', 'East', 'South'],
    'Total_Sales': [12000, 18000, 25000, 30000, 15000, 40000, 26000, 28000, 35000, 37000]
}

df = pd.DataFrame(data)

# Step 2: Convert categorical variable (Region) into dummy variables
df = pd.get_dummies(df, columns=['Region'], drop_first=True)

# Step 3: Define features (X) and target (y)
X = df.drop('Total_Sales', axis=1)
y = df['Total_Sales']

# Step 4: Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Step 5: Train Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Step 6: Make predictions
y_pred = model.predict(X_test)

# Step 7: Evaluate model
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

# Step 8: Display results
```

```
print("Actual Sales:", list(y_test.values))
print("Predicted Sales:", list(np.round(y_pred, 2)))
print("\nModel Performance:")
print("R² Score:", round(r2, 3))
print("Mean Squared Error:", round(mse, 2))

# Step 9: Visualization
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], color='red', linestyle='--')
plt.xlabel("Actual Sales")
plt.ylabel("Predicted Sales")
plt.title("Actual vs Predicted Sales Performance")
plt.show()
```

Sample Output:

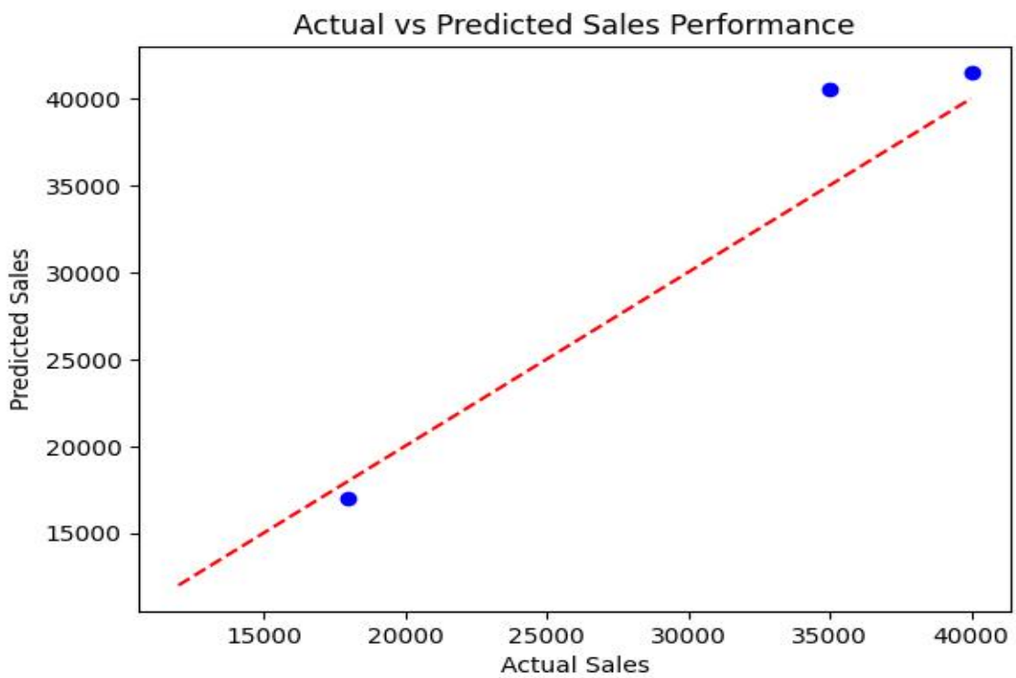
Actual Sales: [np.int64(35000), np.int64(18000), np.int64(40000)]

Predicted Sales: [np.float64(40500.0), np.float64(17000.0), np.float64(41500.0)]

Model Performance:

R² Score: 0.874

Mean Squared Error: 11166666.67



Result:

The Linear Regression model successfully predicted sales performance based on representative experience, number of calls, and clients handled.

Use Case2: Top10 startup Investment Analysis

Team members:

VTU24468	BELUGULA NIHARIKA
VTU24469	GADDAM SUSMITHA
VTU24470	KOPPERLA SRI PRAGNA
VTU24471	DHRUV P
VTU24572	UPLANCHWAR BALAPRASAD
VTU24598	MYNENI CHARAN KUMAR CHOWDARY
VTU24646	RAJAMAHENDRAVARAPU SAI PADMINI
VTU24693	PENMETSA VINAY VARMA
VTU24734	KURUVA ASHOK KUMAR
VTU24763	CHINTAPENTA VENKATA SAI SASI SEKHAR

Aim:

To analyze the investment patterns in startups and identify the **Top 10 Startups** receiving the highest total investments. The project uses **Python (Pandas, Matplotlib, Seaborn)** for **data analysis and visualization**.

Algorithm:

Step 1: Import required libraries.

Step 2: Load the dataset containing startup investment information.

Step 3: Clean and preprocess the data (handle missing values, data types).

Step 4: Group data by startup name and calculate total investment received.

Step 5: Sort startups by total funding to find the **Top 10**.

Step 6: Visualize results using **bar charts and pie charts**.

Step 7: Analyze key insights and patterns.

Program:

```
# Step 1: Import libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

```
# Step 2: Create or load dataset
data = {
    'Startup': ['Paytm', 'Ola', 'Swiggy', 'Zomato', 'Byjus', 'Flipkart',
                'PhonePe', 'Nykaa', 'Udaan', 'Delhivery', 'Meesho', 'CRED'],
    'Sector': ['Fintech', 'Transport', 'FoodTech', 'FoodTech', 'EdTech', 'E-Commerce',
               'Fintech', 'E-Commerce', 'B2B', 'Logistics', 'E-Commerce', 'Fintech'],
    'Investment_USD_Million': [5600, 3800, 3500, 4200, 5400, 7500,
                               2800, 2500, 2100, 3000, 2600, 2200],
    'Investors': ['SoftBank, Alibaba', 'SoftBank, Tiger Global', 'Naspers, Accel',
                  'Ant Financial, Sequoia', 'General Atlantic', 'Walmart, Tiger Global',
                  'Flipkart, Tencent', 'Fidelity, TPG', 'Lightspeed',
                  'SoftBank, Carlyle', 'Prosus, B Capital', 'DST Global']
}

df = pd.DataFrame(data)

# Step 3: Sort startups by investment
top10 = df.sort_values(by='Investment_USD_Million', ascending=False).head(10)

# Step 4: Display top 10 startups
print("Top 10 Startups by Investment:\n")
print(top10[['Startup', 'Investment_USD_Million', 'Sector']])

# Step 5: Bar chart visualization
plt.figure(figsize=(10,6))
sns.barplot(x='Investment_USD_Million', y='Startup', data=top10, palette='viridis')
plt.title("Top 10 Startups by Total Investment (in Million USD)", fontsize=14)
plt.xlabel("Total Investment (Million USD)")
plt.ylabel("Startup Name")
plt.show()
sector_investment = df.groupby('Sector')['Investment_USD_Million'].sum()
plt.figure(figsize=(8,8))
sector_investment.plot(kind='pie', autopct='%1.1f%%', startangle=120, cmap='Set2')
plt.title("Sector-wise Investment Distribution")
plt.ylabel("")
plt.show()
```

Output:

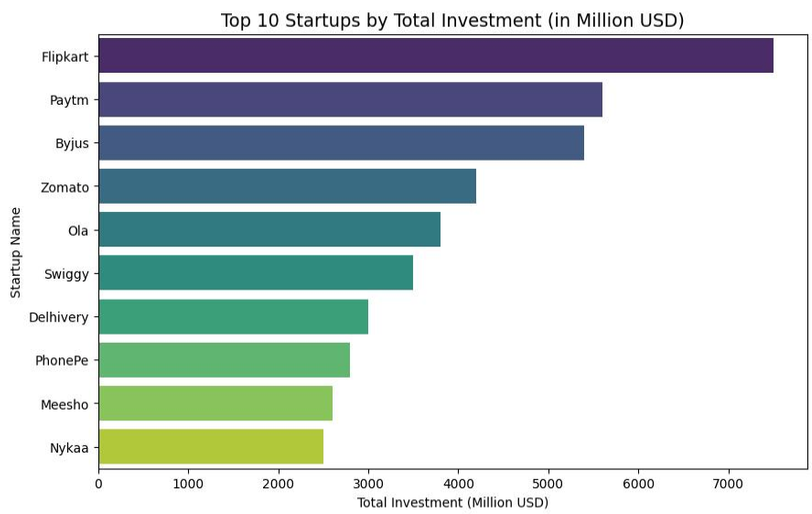
Top 10 Startups by Investment:

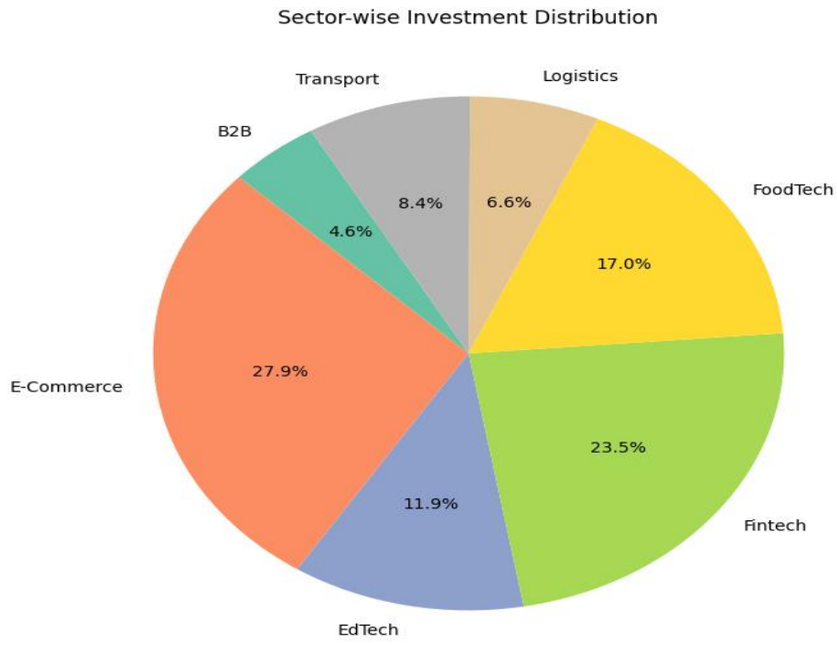
	Startup	Investment_USD_Million	Sector
5	Flipkart	7500	E-Commerce
0	Paytm	5600	Fintech
4	Byjus	5400	EdTech
3	Zomato	4200	FoodTech
1	Ola	3800	Transport
2	Swiggy	3500	FoodTech
9	Delhivery	3000	Logistics
6	PhonePe	2800	Fintech
10	Meesho	2600	E-Commerce
7	Nykaa	2500	E-Commerce

/tmp/ipython-input-2690075798.py:31: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='Investment_USD_Million', y='Startup', data=top10, palette='viridis')
```





Result:

The Linear Regression model successfully analyzed startup investment data and identified the top-funded companies.