Customer Segmentation using Data Science

TEAM MEMBER: Sakthibala A

NM ID : aut22leai03

Introduction:

- It is the process of grouping customers according to how and why they are buy products.
- The problem is to implement data science techniques to segment customers based on their behavior, preferences, and demographic attributes.
- The goal is to enable businesses to personalize marketing strategies and enhance customer satisfaction.
- This project involves data collection, data preprocessing, feature engineering, clustering algorithms, visualization, and interpretation of results.
- The main goal for the customer segmentation using data science is to divide the customer base into distinct groups based on similar characteristics.
- This segment will helpful for many Business purpose.

Project Phase 4:

In this section we have to continue building the project by performing different activities like feature engineering, model training, model evaluation(Mall Dataset).

Dataset:

Dataset link :(https://www.kaggle.com/datasets/akram24/mall-customers)

CustomerI	Genre	Age	Annual Inc	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72
11	Male	67	19	14
12	Female	35	19	99
13	Female	58	20	15
14	Female	24	20	77
15	Male	37	20	13
16	Male	22	20	79
17	Female	35	21	35
18	Male	20	21	66
19	Male	52	23	29
20	Female	35	23	98
21	Male	35	24	35
22	Male	25	24	73
23	Female	46	25	5
24	Male	31	25	73
25	Female	54	28	14
26	Male	29	28	82
27	Female	45	28	32
28	Male	35	28	61

Exploratory Data Analysis:

Exploratory Data Analysis (EDA) for customer segmentation in a mall dataset involves first importing and cleaning the data. Then, it requires summarizing key variables and employing data visualization techniques to understand customer characteristics. EDA helps identify patterns, correlations, and outliers, which are essential for informed decision-making. Subsequently, relevant variables are chosen for segmentation, and clustering techniques are applied to create customer segments. These segments are then interpreted to understand the distinct characteristics of each group. EDA facilitates insights into shopping behavior and

assists in tailoring marketing strategies and store layouts to better meet the diverse needs of mall customers, enhancing business outcomes.

Program :

```
# import required packages
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(context="notebook", palette="Spectral", style = 'darkgrid' ,font_scale = 1.5, color_codes=True)
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
from tensorflow import keras
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.metrics import mean_squared_error
```

```
# import dataset
df=pd.read_csv("C:/Users/ASUS/Downloads/archive/Mall_Customers.csv")
df.head()
df.tail()
```

In [86]:
```
df=pd.read_csv("C:/Users/ASUS/Downloads/archive/Mall_Customers.csv")
df.head()
```

Out[86]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

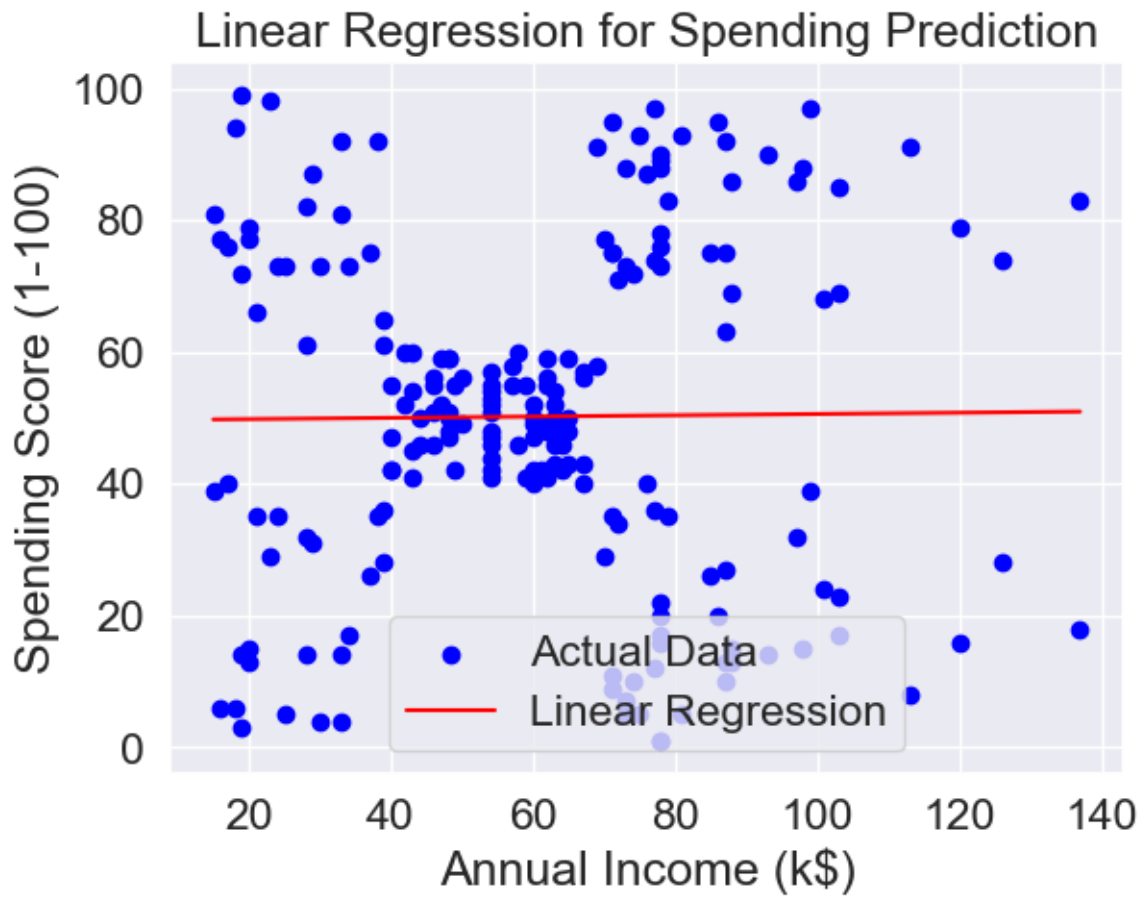
In [62]:
```
df.tail()
```

Out[62]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

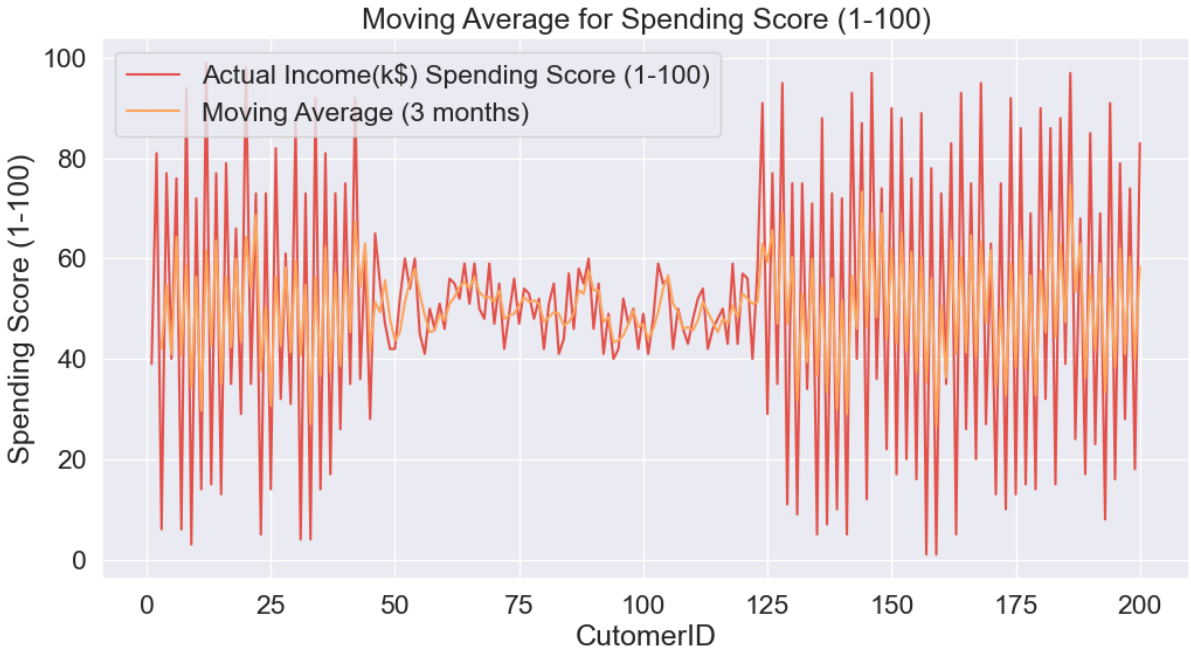
Perfoming Linear Regression:

```
X = df[['Annual Income (k$)']]
y = df['Spending Score (1-100)']
model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)
plt.scatter(X, y, color='blue', label='Actual Data')
plt.plot(X, y_pred, color='red', label='Linear Regression')
plt.title('Linear Regression for Spending Prediction')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



Performing Moving Average

```
feature = 'Spending Score (1-100)'

window_size = 3

df['Moving_Average'] = df[feature].rolling(window=window_size).mean()

plt.figure(figsize=(12, 6))

plt.plot(df['CustomerID'], df[feature], label='Actual Income(k$) ' + feature)

plt.plot(df['CustomerID'], df['Moving_Average'], label='Moving Average (' +
str(window_size) + ' months)')

plt.xlabel('CutomerID')

plt.ylabel(feature)

plt.title('Moving Average for ' + feature)

plt.legend()

plt.grid(True)

plt.show()
```



Performing LSTM

```
feature = 'Spending Score (1-100)'
data = df[[feature]]
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)
train_size = int(len(data) * 0.8)
train_data, test_data = data_scaled[:train_size], data_scaled[train_size:]
def create_sequences(data, sequence_length):
    X, y = [], []
    for i in range(len(data) - sequence_length):
        X.append(data[i:i+sequence_length])
        y.append(data[i+sequence_length])
    return np.array(X), np.array(y)


sequence_length = 10  # Adjust this as needed
X_train, y_train = create_sequences(train_data, sequence_length)
X_test, y_test = create_sequences(test_data, sequence_length)
model = Sequential()
```

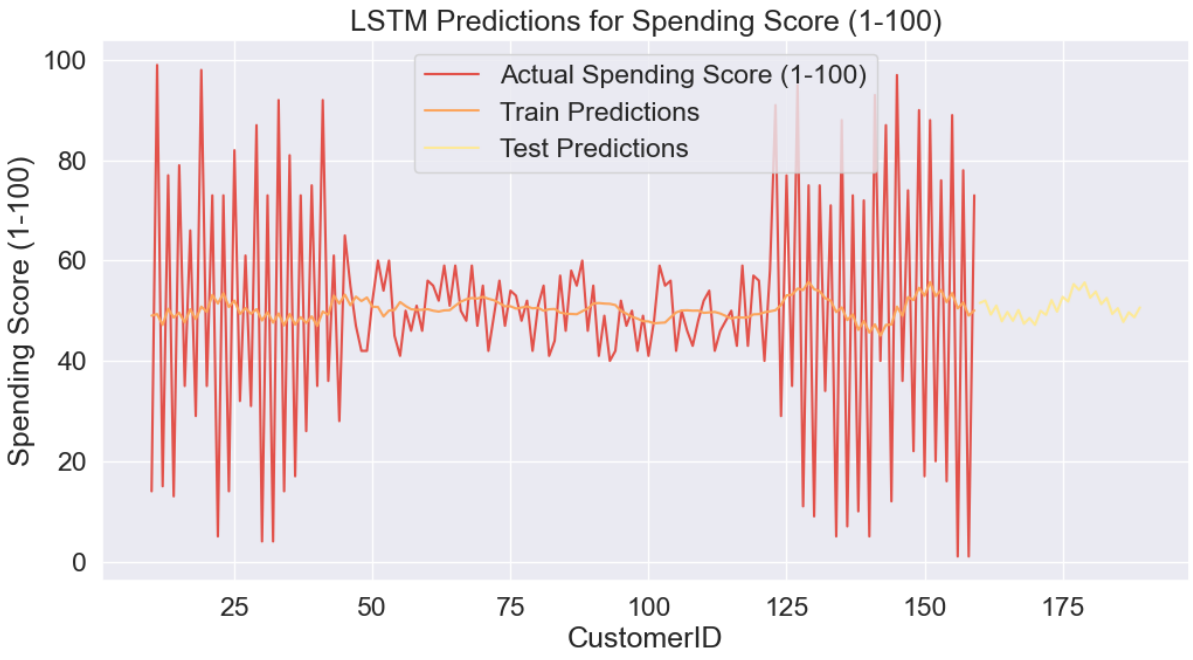
```
model.add(LSTM(50, input_shape=(sequence_length, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(X_train, y_train, epochs=15, batch_size=32, verbose=1)
```

```
Epoch 1/15
5/5 [==============================] - 2s 4ms/step - loss: 0.3066
Epoch 2/15
5/5 [==============================] - 0s 4ms/step - loss: 0.1686
Epoch 3/15
5/5 [==============================] - 0s 6ms/step - loss: 0.0828
Epoch 4/15
5/5 [==============================] - 0s 6ms/step - loss: 0.0557
Epoch 5/15
5/5 [==============================] - 0s 5ms/step - loss: 0.0671
Epoch 6/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0627
Epoch 7/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0549
Epoch 8/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0551
Epoch 9/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0567
Epoch 10/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0556
Epoch 11/15
5/5 [==============================] - 0s 5ms/step - loss: 0.0543
Epoch 12/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0541
Epoch 13/15
5/5 [==============================] - 0s 5ms/step - loss: 0.0540
Epoch 14/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0536
Epoch 15/15
5/5 [==============================] - 0s 4ms/step - loss: 0.0534
```

```
train_predictions = model.predict(X_train)
test_predictions = model.predict(X_test)
train_predictions = scaler.inverse_transform(train_predictions)
test_predictions = scaler.inverse_transform(test_predictions)
```

```
plt.figure(figsize=(12, 6))
```

```
plt.plot(df.index[sequence_length:sequence_length + len(train_predictions)], df[feature][sequence_length:sequence_length + len(train_predictions)], label='Actual ' + feature)

plt.plot(df.index[sequence_length:sequence_length + len(train_predictions)], train_predictions, label='Train Predictions')

test_index = df.index[sequence_length + len(train_predictions):sequence_length + len(train_predictions) + len(test_predictions)]

plt.plot(test_index, test_predictions, label='Test Predictions')


plt.xlabel('CustomerID')

plt.ylabel(feature)

plt.title('LSTM Predictions for ' + feature)

plt.legend()

plt.grid(True)

plt.show()
```



conclusion :

Training and Evaluating models in customer segmentation using mall dataset is like creating a tool to sort shoppers into groups based on their behavior. This helps malls better target customers, manage stock, and give personalized experiences. Checking how well this tool works ensures it's doing its job correctly, and using it can boost business success in a crowded mall setting.