

Task 2: Lookalike Model

1. Importing Required Libraries

```
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import MinMaxScaler
```

1. Loading Datasets

```
# Load datasets
customers = pd.read_csv("Customers.csv")
products = pd.read_csv("Products.csv")
transactions = pd.read_csv("Transactions.csv")
```

1. Merging the Datasets

```
# Merge datasets
merged_data = transactions.merge(products,
on="ProductID").merge(customers, on="CustomerID")
```

1. Feature Engineering

```
# Feature engineering
customer_features = merged_data.groupby("CustomerID").agg({
    "TotalValue": "sum",                # Total spend
    "Quantity": "sum",                  # Total quantity purchased
    "Price_x": "mean",                  # Average product price
    "TransactionDate": "count"          # Transaction count
}).rename(columns={
    "TotalValue": "Total_Spend",
    "Quantity": "Total_Quantity",
    "Price_x": "Avg_Product_Price",
    "TransactionDate": "Transaction_Count"
})
```

1. Adding Customer Demographics

```
# Add customer demographics
customer_features = customer_features.merge(customers[["CustomerID",
"Region", "SignupDate"]], on="CustomerID")
```

1. Converting Signup Date to Numerical (Days Since Signup)

```
# Convert signup date to numerical (days since signup)
customer_features["SignupDate"] =
pd.to_datetime(customer_features["SignupDate"])
customer_features["Days_Since_Signup"] = (pd.Timestamp.now() -
customer_features["SignupDate"]).dt.days
customer_features.drop(columns=["SignupDate"], inplace=True)
```

1. One-Hot Encoding Categorical Data

```
# One-hot encode categorical data (e.g., Region)
customer_features = pd.get_dummies(customer_features,
columns=["Region"])
```

1. Normalizing Numerical Features

```
# Normalize numerical features
scaler = MinMaxScaler()
numerical_cols = ["Total_Spend", "Total_Quantity",
"Avg_Product_Price", "Transaction_Count", "Days_Since_Signup"]
customer_features[numerical_cols] =
scaler.fit_transform(customer_features[numerical_cols])
```

1. Computing Similarity Matrix

```
# Compute similarity matrix
customer_ids = customer_features["CustomerID"]
feature_matrix = customer_features.drop(columns=["CustomerID"])
similarity_matrix = cosine_similarity(feature_matrix)
```

1. Generating Lookalike Recommendations

```
# Generate lookalike recommendations for first 20 customers
lookalikes = {}
for idx, customer_id in enumerate(customer_ids[:20]): # First 20
customers only
    similarity_scores = list(enumerate(similarity_matrix[idx]))
    similarity_scores = sorted(similarity_scores, key=lambda x: x[1],
reverse=True) # Sort by similarity
    top_lookalikes = [(customer_ids[i], round(score, 3)) for i, score
in similarity_scores if customer_ids[i] != customer_id[:3]]
    lookalikes[customer_id] = top_lookalikes
```

1. Saving Lookalike Recommendations to CSV

```
# Create Lookalike.csv
lookalike_df = pd.DataFrame({
    "CustomerID": lookalikes.keys(),
    "Lookalike_Customers": [str(val) for val in lookalikes.values()]
})
lookalike_df.to_csv("Lookalike.csv", index=False)

print("Lookalike recommendations saved to 'Lookalike.csv'")
Lookalike recommendations saved to 'Lookalike.csv'
```