Online Retail Recommendation System

1. Introduction

In this project, we aim to develop a recommendation system for an online retail dataset using the Singular Value Decomposition (SVD) algorithm. Recommendation systems are essential in online retail as they enhance customer experience by providing personalized product recommendations. The SVD algorithm, a popular collaborative filtering method, helps predict user preferences for products based on past interactions. This project involves data preprocessing, model training, evaluation, and analysis of the results to ensure accurate and reliable recommendations.

2. Objective

The objective of this project is to build a recommendation system using the SVD algorithm that can accurately predict customer ratings for products. By leveraging historical sales data, we aim to identify patterns and preferences that allow us to recommend products effectively. The success of the recommendation system will be evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics, ensuring that the recommendations are precise and beneficial for users.

3. Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for analysis and model training. The following steps were undertaken for data preprocessing:

- **Loading the Dataset:** The dataset was loaded from an Excel file using the pandas library.
- **Handling Missing Values:** Missing values were removed to ensure data quality.
- **Removing Duplicates:** Duplicate entries were identified and removed to avoid redundancy.
- **Converting Data Types:** The InvoiceDate column was converted to a datetime format for better manipulation.
- **Feature Engineering:** A new feature, TotalAmount, was created by multiplying the Quantity and UnitPrice columns to represent the total amount spent on each transaction.
- **Scaling the Data:** The TotalAmount feature was scaled to a typical rating scale (0.5 to 5) using MinMaxScaler to match the format required by the SVD algorithm.

Here is a preview of the preprocessed data:

	CustomerID	StockCode	TotalAmount
0	17850.0	85123A	2.750204
1	17850.0	71053	2.750272
2	17850.0	84406B	2.750294
3	17850.0	84029G	2.750272
4	17850.0	84029E	2.750272

4. Model Training and Evaluation

The model training and evaluation process involves the following steps:

- **Loading Data into Surprise Library:** The preprocessed data was loaded into the Surprise library, which is designed for building and analyzing recommendation systems.
- **Train-Test Split:** The dataset was split into training and testing sets to evaluate the model's performance on unseen data.
- **Training the SVD Model:** The SVD algorithm was trained on the training set using cross-validation to ensure robust performance evaluation.
- **Evaluating the Model:** The model's performance was evaluated using RMSE and MAE metrics across multiple folds. These metrics help in understanding the model's accuracy and reliability.

5. Results

The evaluation of the SVD algorithm resulted in the following metrics:

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.0406  0.0403  0.0413  0.0414  0.0404  0.0408  0.0005
MAE (testset)     0.0264  0.0263  0.0262  0.0262  0.0264  0.0263  0.0001
Fit time          5.92    6.40    6.48    6.24    6.65    6.34    0.25
Test time         1.01    1.14    0.80    0.89    0.91    0.95    0.11
RMSE: 0.029201229413272325
```

These results demonstrate that the SVD algorithm effectively predicts customer ratings, providing a strong basis for reliable product recommendations.

6. Conclusion

In conclusion, this project successfully developed a recommendation system using the SVD algorithm for an online retail dataset. The data preprocessing steps ensured high-quality input data, while the SVD algorithm effectively captured the patterns and preferences in customer behavior. The evaluation metrics, RMSE and MAE, indicated that the model provides accurate and reliable recommendations. Future work can involve exploring other algorithms, incorporating additional features, and testing the system in a live environment to further enhance the recommendation system's performance.

7. Appendix: Code

The complete code used for data preprocessing, model training, and evaluation is provided below:

```
import pandas as pd

from surprise import Dataset, Reader, SVD

from surprise.model_selection import cross_validate, train_test_split

from sklearn.metrics import mean_squared_error

from sklearn.preprocessing import MinMaxScaler

file_path = r'C:\Users\hp\Downloads\OnlineRetail (1) (1).xlsx'


# Load the dataset

data = pd.read_excel(file_path)


# Data preprocessing

data.dropna(inplace=True)

data.drop_duplicates(inplace=True)

data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])

data['TotalAmount'] = data['Quantity'] * data['UnitPrice']


# Scaling the TotalAmount to a typical rating scale (0.5 to 5)

scaler = MinMaxScaler(feature_range=(0.5, 5))

data['TotalAmount'] = scaler.fit_transform(data[['TotalAmount']])
```

```
# Verify the data
print(data[['CustomerID', 'StockCode', 'TotalAmount']].head())


# Load data into Surprise library
reader = Reader(rating_scale=(0.5, 5))
data_surprise = Dataset.load_from_df(data[['CustomerID', 'StockCode', 'TotalAmount']], reader)


# Train-test split
trainset, testset = train_test_split(data_surprise, test_size=0.2)


# Train the SVD model
svd = SVD()
cross_validate(svd, data_surprise, measures=['RMSE', 'MAE'], cv=5, verbose=True)


# Fit on the whole training set
trainset = data_surprise.build_full_trainset()
svd.fit(trainset)


# Make predictions on the test set
predictions = svd.test(testset)
rmse = mean_squared_error([pred.r_ui for pred in predictions], [pred.est for pred in predictions], squared=False)
print(f'RMSE: {rmse}')
```