**AI Project:**

***Product Recommendation System Uniform cost search using Uninformed Search***

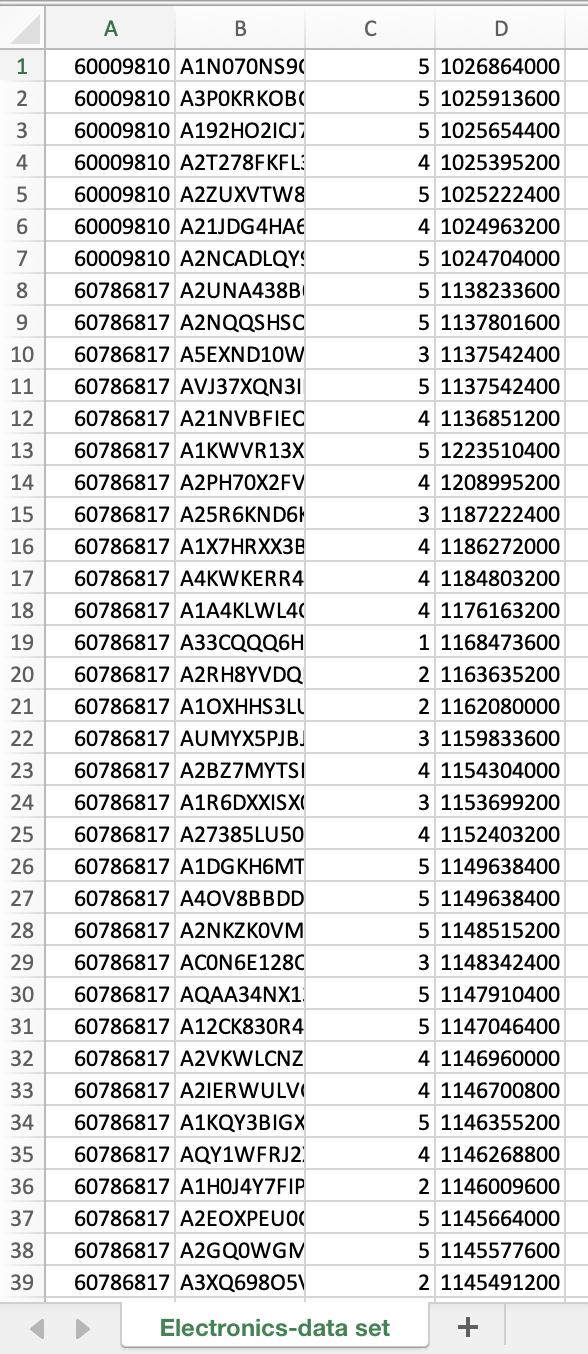
**Motivation:**

This system automatically recommends us products to the users on amazon using Item-based system, product popularity-based system, model-based system. With this system it will be easier for the users and creates a better shopping experience.

**Data:**

**Source:** <https://nijianmo.github.io/amazon/index.html>

**Content:** The data comes from Amazon Review Data. It was extracted in 2018 and it contains 200,953 records about different electronics.



**Objective:**

The main objective of the project is to recommend products based on the user experience, by using Item-based system, product popularity-based system, model-based system also a graph is made after the scatter plot which helps the admin.

* Electronics (Product Recommendation System)
* Better/Smother shopping experience

**Implementation:**

**Import libraries:**

import matplotlib  
from IPython.core.interactiveshell import InteractiveShell  
from jedi.api.refactoring import inline  
  
InteractiveShell.ast\_node\_interactivity = "all"  
from numpy import count\_nonzero, diag, arange, dot  
import pandas as pd  
import seaborn as sns  
from sklearn.model\_selection import train\_test\_split  
from scipy.sparse.linalg import svds  
import warnings

**Imports:**

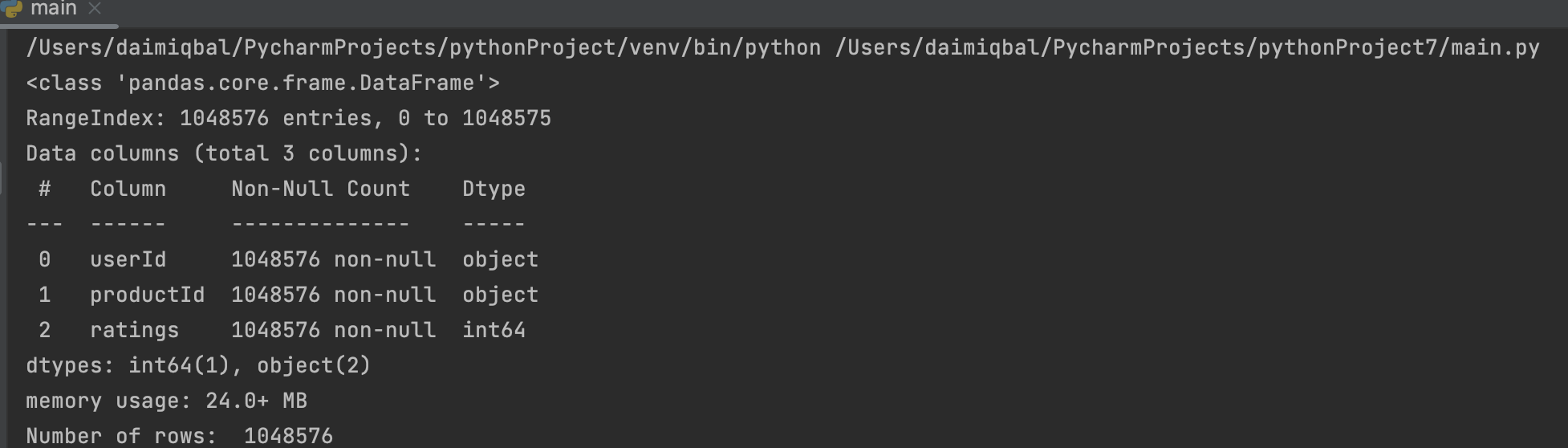
# Import the dataset and give the column names  
columns = ['userId', 'productId', 'ratings', 'timestamp']  
electronics\_df = pd.read\_csv('Electronics-data set.csv', names=columns)  
electronics\_df.head()  
electronics\_df.drop('timestamp', axis=1, inplace=True)  
electronics\_df.info()

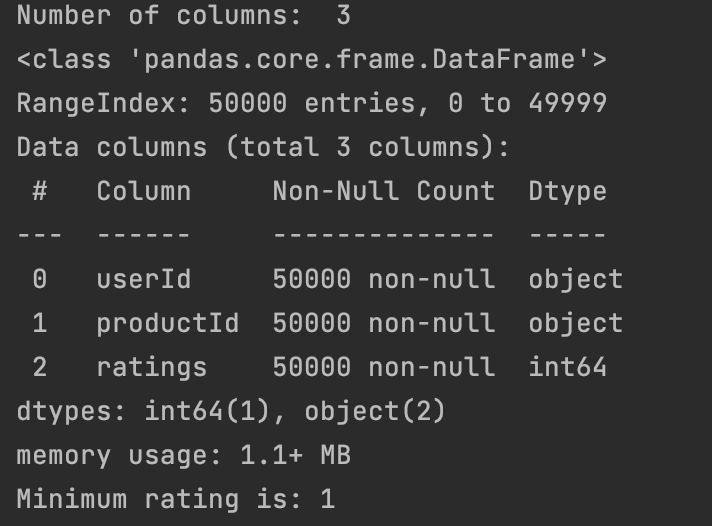
**Functions:**

def recommend(user\_id):  
 user\_recommendations = popularity\_recommendations  
  
 # Add user\_id column for which the recommendations are being generated  
 user\_recommendations['userId'] = user\_id  
  
 # Bring user\_id column to the front  
 cols = user\_recommendations.columns.tolist()  
 cols = cols[-1:] + cols[:-1]  
 user\_recommendations = user\_recommendations[cols]  
  
 return user\_recommendations

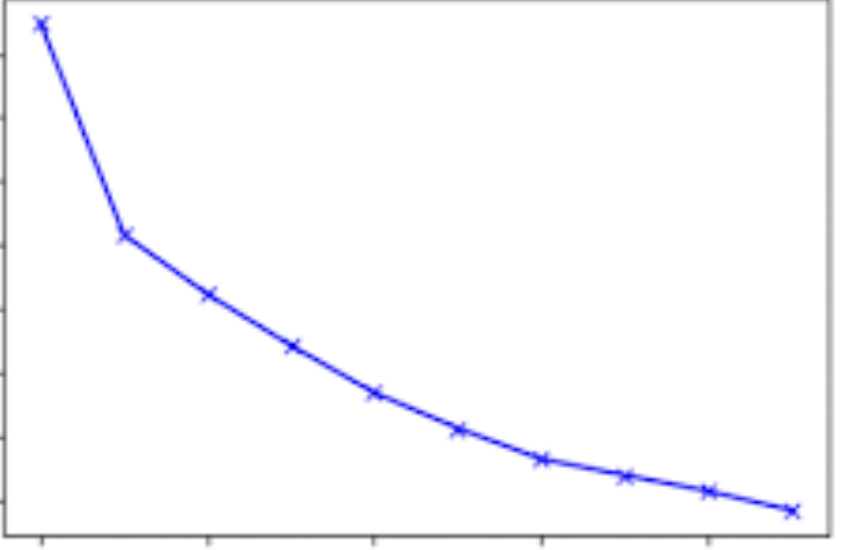
def recommend\_items(userID, pivot\_df, preds\_df, num\_recommendations):  
 # index starts at 0  
 user\_idx = userID - 1  
 # Get and sort the user's ratings  
 sorted\_user\_ratings = pivot\_df.iloc[user\_idx].sort\_values(ascending=False)  
 # sorted\_user\_ratings  
 sorted\_user\_predictions = preds\_df.iloc[user\_idx].sort\_values(ascending=False)  
 # sorted\_user\_predictions  
 temp = pd.concat([sorted\_user\_ratings, sorted\_user\_predictions], axis=1)  
 temp.index.name = 'Recommended Items'  
 temp.columns = ['user\_ratings', 'user\_predictions']  
 temp = temp.loc[temp.user\_ratings == 0]  
 temp = temp.sort\_values('user\_predictions', ascending=False)  
 print('\nBelow are the recommended items for user(user\_id = {}):\n'.format(userID))  
 print(temp.head(num\_recommendations))

**Run(Output):**



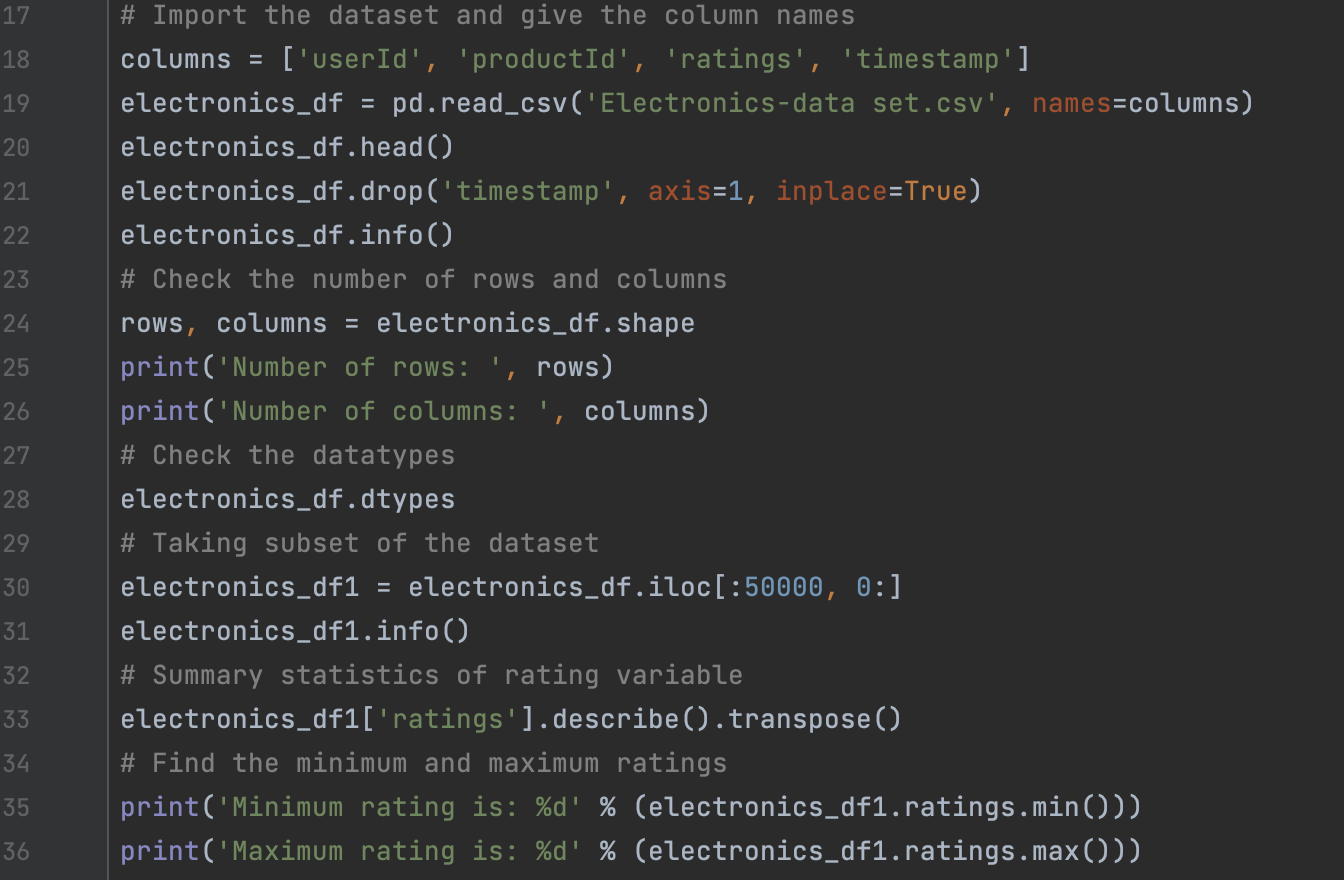


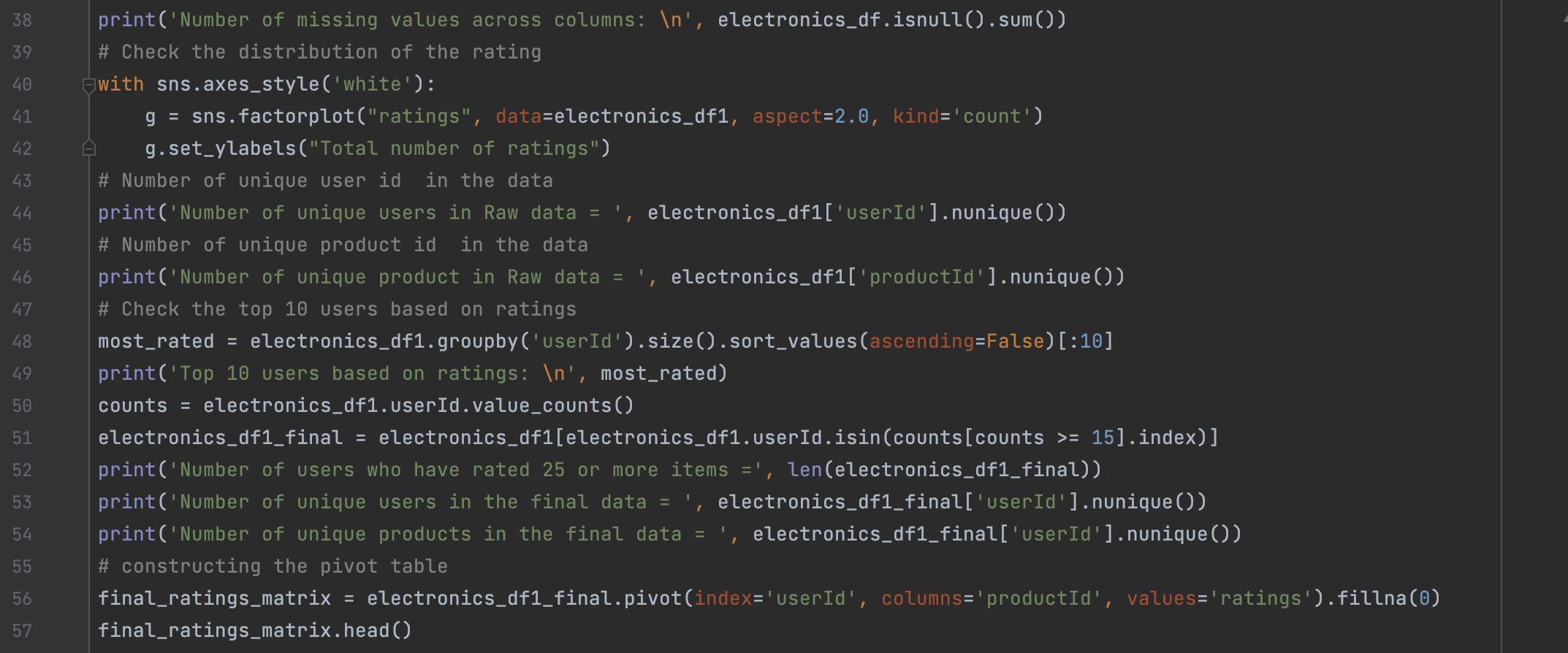
**Scatter plot:**

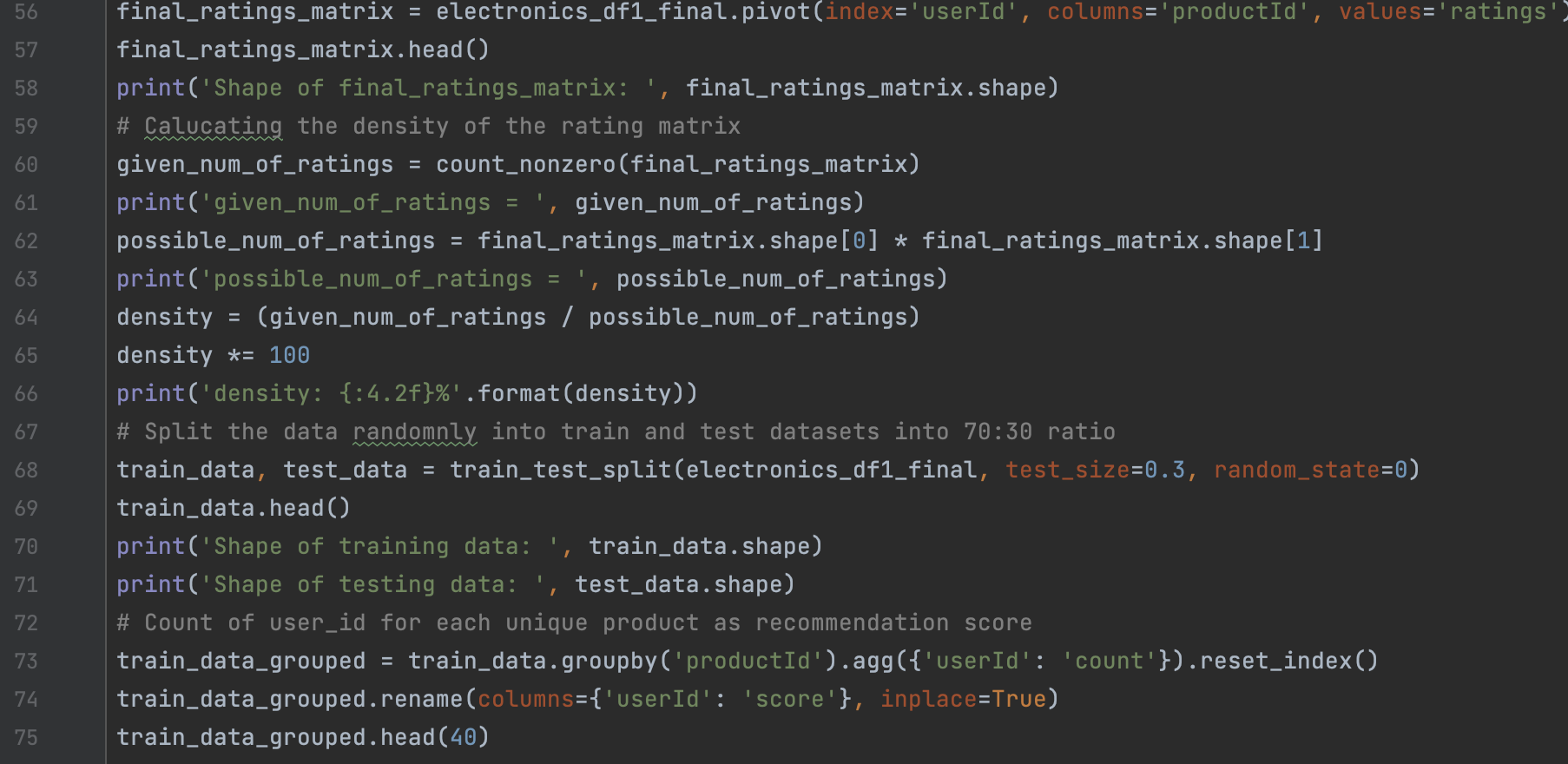
****

**Code:**

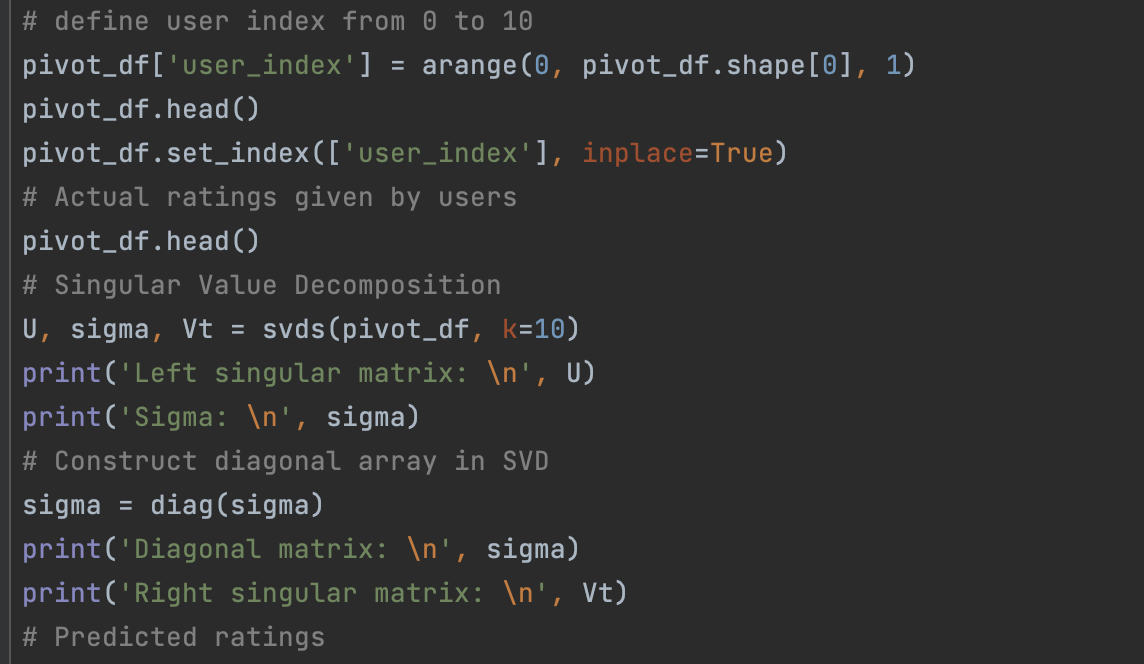
**Based on rating recommendation:**

****

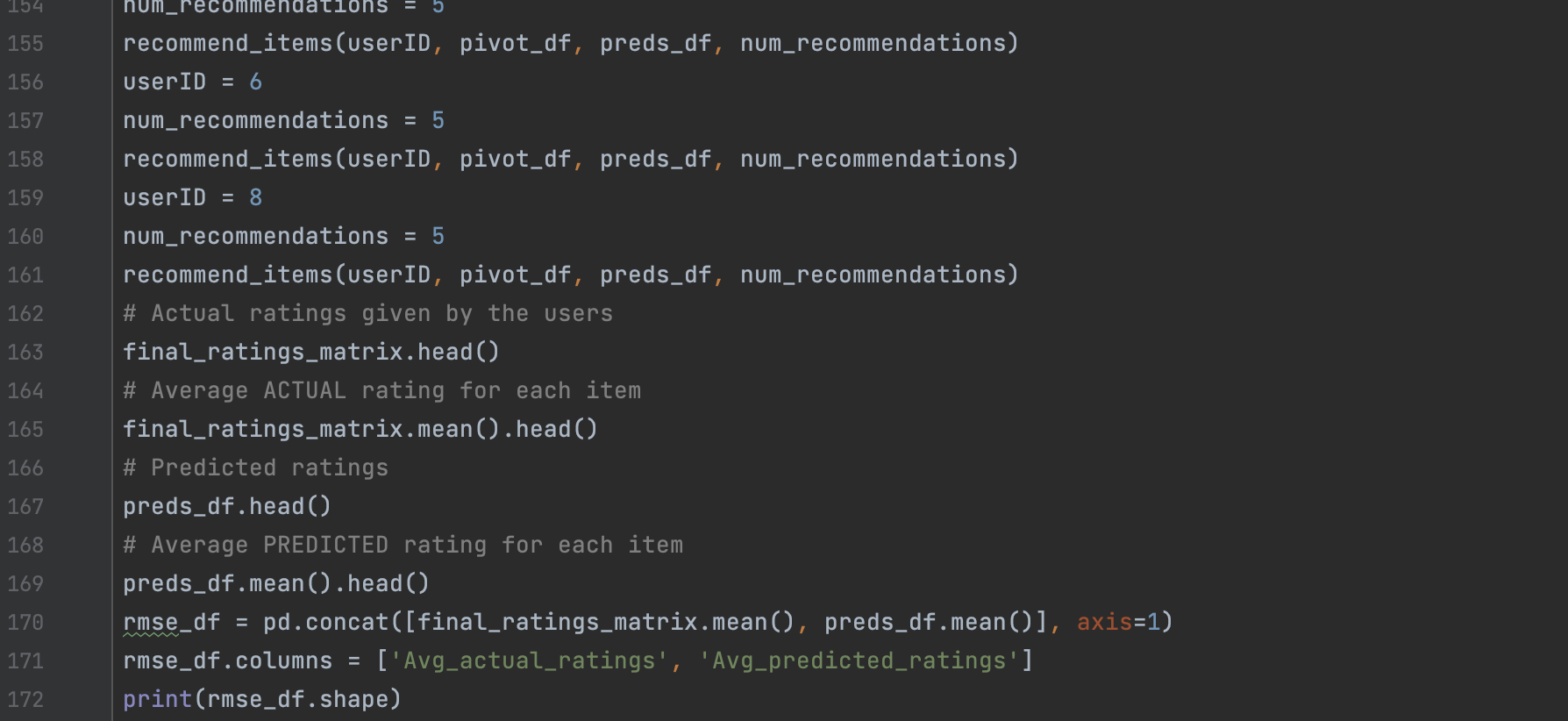
****

****

**Decomposition:**

****

**Average Predicted:**

****