Configuration Manual

MSc Research Project
Artificial Intelligence for Business

Erdal Ozcelik
Student ID: x22203184

School of Computing
National College of Ireland

Supervisor: VICTOR DEL ROSAL

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Erdal Ozcelik
Student ID:	x22203184
Programme:	Artificial Intelligence for Business
Year:	2023
Module:	MSc Research Project
Supervisor:	VICTOR DEL ROSAL
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	1223
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	27th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Erdal Ozcelik
x22203184

1 Introduction

This configuration manual guides replicating the product recommendation systems developed within the NCI research project. It also outlines the process for conducting and assessing a survey to evaluate these systems.

2 Software Requirement

The product recommendation systems are developed using the Python programming language. The implementation is carried out using Jupyter Notebook through the Anaconda Navigator. The survey, conducted to evaluate these systems, involved capturing videos using the ScreenRec program, with subsequent editing performed using Microsoft Clipchamp. The analysis of the survey data is conducted using IBM SPSS.

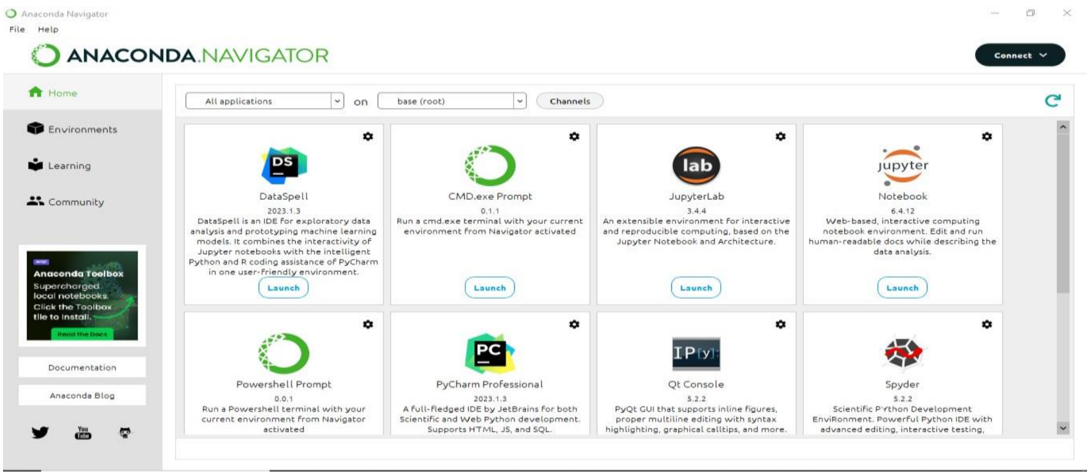


Figure 1: Anaconda Navigator

3 Package Requirements

The following Python packages are utilized for the development of the product recommendation systems. The packages are installed using both the pip command in Jupyter Notebook and Visual Studio Code.

- pandas
- re
- numpy
- Surprise
- PySimpleGUI
- os
- random
- string

1. Import the Libraries

In [229]:

```
# Before executing the code, ensure that all necessary packages are installed.
# The required packages can be installed using in a Jupyter Notebook or Visual Studio Code, with the following commands: pip

import pandas as pd
import re
from surprise import Dataset, Reader, SVD
from surprise.model_selection import train_test_split
from surprise.accuracy import rmse
import os
import PySimpleGUI as sg
from surprise import Dataset, Reader, SVD
import random
import string
```

Figure 2: Package Requirements and Installation

4 Dataset Description

The product recommendation systems are developed using the Amazon Sales Dataset, which is publicly available on Kaggle[1]. To facilitate ease of use, USER IDs are replaced with randomly assigned names and product names are shortened without altering their essence. The modified dataset is then placed on the local drive.

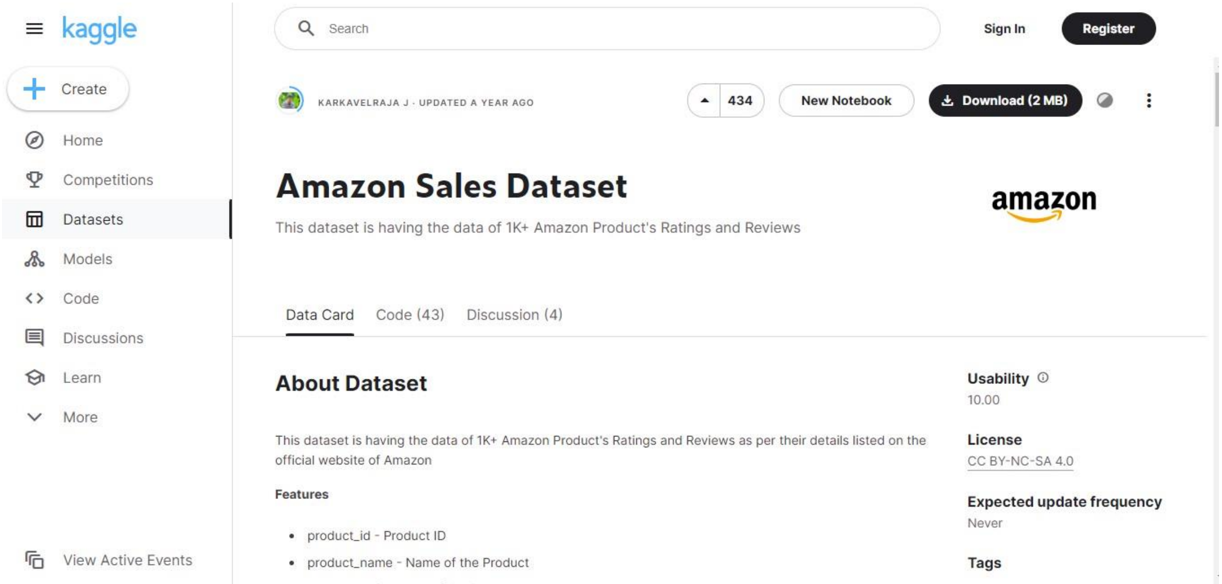


Figure 3: the Amazon Sales Dataset, which is publicly available on Kaggle.

user_id	product_name	user_name	review_id	review_title	review_content	rating_count	about_product
1	Wayona Fast Charging Lightning Cable (G...	Michael Johnson	R3HXWT0LRP0NMF,R2AJM3LF...	Satisfied,Charging is really fast,...	Looks durable Charging is fine t...	24269	High Compatibility : Compatible ...
2	Ambrane Fast Charging Type C Cable (Bla...	John White	RGIQEG07R9HS2,R1SMWZQ86...	A Good Braided Cable for Your T...	I ordered this cable to connect m...	43994	Compatible with all Type C enab...
3	Sounce iPhone Fast Charging Cable (White)	Sophia Taylor	R3J3EQQ9TZI5ZJ,R3E7WBGK7...	Good speed for earlier versions,...	Not quite durable and sturdy,http...	7928	【 Fast Charger& Data Sync】-...
4	boAt Deuce 2-in-1 Type-C & Micro USB Ca...	William Wilson	R3EEUZKKK9J36I,R3HJVYCLY...	Good product,Good one,Nice,Re...	Good product,long wire,Charges...	94363	The boAt Deuce USB 300 2 in 1 ...
5	Portronics Fast Charging 8 Pin USB Cable...	David Harris	R1BP4L2HH9TFUP,R16PVJEX...	As good as original,Decent,Goo...	Bought this instead of original a...	16905	[CHARGE & SYNC FUNCTION]- ...
6	pTron Solero Type-C Data & Charging Cab...	William Brown	R7S8ANNSDPR40,R3CLZFLHV...	It's pretty good,Average quality,ve...	It's a good product.,Like,Very go...	24871	Fast Charging & Data Sync: Sole...
7	boAt Micro USB Fast Charging Cable (Black)	Michael Brown	R8E73K2KWJRDS,RSD0JTIIW...	Long durable.,good,Does not ch...	Build quality is good and it is co...	15188	It Ensures High Speed Transmi...
8	MI Type-C Cable (Black)	Daniel Johnson	R2X090D1YHACKR,R32ZCIH9A...	Worth for money - suitable for An...	Worth for money - suitable for An...	30411	1m long Type-C USB Cable|Stur...
9	TP-Link USB WiFi Adapter for PC	Jane Brown	R1LW6NWSVTVZ2H,R3VR5WF...	Works on linux for me. Get the m...	I use this to connect an old PC t...	179691	USB WiFi Adapter —— Speedy ...
10	Ambrane Fast Charging Micro USB Cable (...	William Brown	RGIQEG07R9HS2,R1SMWZQ86...	A Good Braided Cable for Your T...	I ordered this cable to connect m...	43994	Universal Compatibility – It is co...
11	Portronics Type-C Fast Charging Cable (G...	Sophia Smith	R11MQS7WD9C3I0,R2AKH69X...	Good for fast charge but not for d...	The cable is efficient in fast char...	13391	[CHARGE & SYNC FUNCTION]- ...
12	boAt Rugged Micro USB Cable (Black)	Sophia Taylor	R3EEUZKKK9J36I,R3HJVYCLY...	Good product,Good one,Nice,Re...	Good product,long wire,Charges...	94363	The boAt rugged cable features ...
13	AmazonBasics HDMI Cable (Black)	Emily Harris	R1FKOKZ3HHKJBZ,R2WNMZI1...	It's quite good and value for mon...	I am using it for 14 days now. Th...	426973	Flexible, lightweight HDMI cable ...
14	Portronics Type-C to 8 Pin USB Cable (Gre...	Sophia Wilson	R1QETDIPRCX4S0,RARQYQ8P...	Works,Nice Product,Fast Chargi...	Definitely isn't as good as the ori...	2262	[20W PD FAST CHARGING]-It's ...
15	Portronics 8 Pin USB Cable (White)	Jane Johnson	R20XIOU25HEX80,R2X55FA2E...	Great but,Worked well for 6 six ...	Loosing charging cable of apple...	4768	[CHARGE & SYNC FUNCTION]- ...
16	MI Type-C Cable (Red)	Olivia Taylor	R2JPQNKCOE10UK,RQI80JG2...	Good product,using this product ...	I like it □□,Best charging pow...	18757	1M Long Cable. Usb 2.0 (Type A...
17	MI Smart Android LED TV (Black)	Daniel Brown	R13UTIA6KOF6QV,R2UGDZSG...	It is the best tv if you are getting it ...	Pros- xiomi 5a is best in budget-...	32840	Note : The brands, Mi and Xiaom...
18	Ambrane Type C to Type C Cable (Black)	John Brown	RGIQEG07R9HS2,R1SMWZQ86...	A Good Braided Cable for Your T...	I ordered this cable to connect m...	43994	Compatible with all Type C enab...
19	boAt Type C Cable (Black)	Emily Taylor	R2BP8Y5OJXKJLF,R218813TN...	Good for charging and Data tran...	Check for offera before buying,1...	13045	Type C A 325 Cable Is Designed...
20	LG Smart LED TV (Dark Iron Gray)	Sophia Clark	R2PNR69G0BQG2F,R31A0WW...	Sound quality,Very nice,Value for...	LG was always Good , correct d...	11976	Resolution: HD Ready (1366x76...
21	Duracell Lightning Apple Certified Cable (B...	Emily Lee	R12D1BZF9MU8TN,R32MNCW...	Good cable for car,Good substit...	I trust this product! Works well wi...	815	Supports Ios Devices With Max ...
22	tizum HDMI to VGA Adapter Cable (1080P)	Olivia Brown	R1GYK05NN6747O,R1J21BZ29...	Good product ; Average Finishin...	This connector has provided as ...	10962	Superior Stability: Built-in advanc...
23	Samsung Wondertainment Series LED S...	John Brown	R1SN0D4DFBKAZI,R1SX5L77L...	Good,Sound is very low another ...	Overall good.,TV picture ok smar...	16299	Resolution: HD Ready (1366x76...

Figure 4: The Modified Dataset

[1] https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset

4.1 Load Dataset

The modified dataset is loaded into Jupyter Notebook.

2. Load the dataset and Add Headers

```
In [230]: # Load CSV file from the local drive path
          # # Before executing the code, replace "D:\PRS\modified_amazon.csv" with the path to your local CSV file.

          file_path = r"D:\PRS\modified_amazon.csv"
          df = pd.read_csv(file_path,encoding='ISO-8859-1')
```

Figure 5: Loading Modified Dataset into Jupyter Notebook

4.2 Check for Missing Values

The dataset is examined for missing values.

Check for Missing Values

```
In [233]: # Check for missing values
          def check_missing_values(dataframe):
              return dataframe.isnull().sum()

          print(check_missing_values(df))
          df[df.rating_count.isnull()]
```

```
product_id             0
product_name           0
category               0
discounted_price       0
actual_price           0
discount_percentage    0
rating                 0
rating_count           2
about_product          0
user_id                0
user_name              0
review_id              0
review_title           0
review_content         0
img_link               0
product_link           0
dtype: int64
```

Figure 6: Visualization of Missing Values

4.3 Handle Missing Values

Steps are taken to handle any identified missing values.

Handle Missing Values

```
In [234]:  # Remove rows with missing values in the rating_count column
           df.dropna(subset=['rating_count'], inplace=True)
           print(check_missing_values(df))

           product_id             0
           product_name           0
           category               0
           discounted_price       0
           actual_price           0
           discount_percentage    0
           rating                 0
           rating_count           0
           about_product          0
           user_id                0
           user_name              0
           review_id              0
           review_title           0
           review_content         0
           img_link               0
           product_link           0
           dtype: int64
```

Figure 7: Missing Values Handling Process

4.4 Check for Duplicates Values

The dataset is examined for duplicate values.

Check for Duplicates Values

```
In [235]:  # Check for duplicates
           def check_duplicates(dataframe):
               return dataframe.duplicated().sum()

           print(check_duplicates(df))

           0
```

Figure 8: Detection of Duplicate Values

4.5 Data Types Conversion

The data types are converted for use in the implementation of the product recommendation system model.

Data Types Conversion

```
In [236]:  # Check data types
           def check_data_types(dataframe):
               return dataframe.dtypes

           print(check_data_types(df))

product_id              object
product_name            object
category                object
discounted_price        object
actual_price            object
discount_percentage     object
rating                 float64
rating_count           float64
about_product           object
user_id                  int64
user_name               object
review_id               object
review_title            object
review_content          object
img_link                object
product_link            object
dtype: object
```

Figure 9: Detection of Data Types

```
In [237]:  # Convert 'discounted_price' and 'actual_price' columns to strings
           df['discounted_price'] = df['discounted_price'].astype(str).apply(lambda x: re.sub('[^0-9.]', '', x)).astype(float)
           df['actual_price'] = df['actual_price'].astype(str).apply(lambda x: re.sub('[^0-9.]', '', x)).astype(float)

           # Convert 'discount_percentage' column to strings and then process it
           df['discount_percentage'] = df['discount_percentage'].astype(str).str.rstrip('%').astype(float) / 100

In [238]:  df['rating'] = df['rating'].astype(str).str.replace(',', '').astype(float)
           df['rating_count'] = df['rating_count'].astype(str).str.replace(',', '').astype(float)

In [239]:  # Change the 'user_id' column data type to object
           df['user_id'] = df['user_id'].astype(str)

In [240]:  print(check_data_types(df))

product_id              object
product_name            object
category                object
discounted_price       float64
actual_price           float64
discount_percentage    float64
rating                 float64
rating_count           float64
about_product           object
user_id                 object
user_name               object
review_id               object
review_title            object
review_content          object
img_link                object
```

Figure 10: Data Types Conversion Process

4.6 Feature Engineering

Feature engineering is applied to prepare the dataset for the implementation of the model used in the product recommendation system.

Feature Engineering

```
In [241]:  # Creating the column "rating_weighted"
           df['rating_weighted'] = df['rating'] * df['rating_count']

In [242]:  df['sub_category'] = df['category'].astype(str).str.split('|').str[-1]
           df['main_category'] = df['category'].astype(str).str.split('|').str[0]

In [243]:  df.columns

Out[243]:  Index(['product_id', 'product_name', 'category', 'discounted_price',
                  'actual_price', 'discount_percentage', 'rating', 'rating_count',
                  'about_product', 'user_id', 'user_name', 'review_id', 'review_title',
                  'review_content', 'img_link', 'product_link', 'rating_weighted',
                  'sub_category', 'main_category'],
                 dtype='object')

In [244]:  len(df)

Out[244]:  1463
```

Figure 11: Feature Engineering Techniques Applied

5 Model Preparation

5.1 Steps to Implement Collaborative Filtering Model

5.1.1 Data Preprocessing

Data Transformation The 'score' attribute is first converted into numerical data. Following this, the dataset is prepared to be compatible with the Surprise library.

Data Transformation

In [246]:
```python
# Load the dataset
file_path = r"D:\PRS\modified_amazon.csv"
data = pd.read_csv(file_path,encoding='ISO-8859-1')

# Check the unique values in the "rating" column
unique_ratings = data['rating'].unique()
print(unique_ratings)

# Filter out non-numeric ratings
numeric_ratings = pd.to_numeric(unique_ratings, errors='coerce')
valid_ratings = numeric_ratings[~pd.isna(numeric_ratings)]

# Display the valid numeric ratings
print(valid_ratings)

# Update the "rating" column to contain only numeric values
data['rating'] = pd.to_numeric(data['rating'], errors='coerce')

# Remove rows with NaN values in the "rating" column
data = data.dropna(subset=['rating'])
```

```
[4.2 4.  3.9 4.1 4.3 4.4 4.5 3.7 3.3 3.6 3.4 3.8 3.5 4.6 3.2 5.  4.7 3.
 2.8 3.1 4.8 2.3 2.  2.6 2.9]
[4.2 4.  3.9 4.1 4.3 4.4 4.5 3.7 3.3 3.6 3.4 3.8 3.5 4.6 3.2 5.  4.7 3.
 2.8 3.1 4.8 2.3 2.  2.6 2.9]
```

Figure 12: Data Transformation Illustration

Train-Test Split The dataset is then divided into two subsets: 80% of the data is used for training purposes, while the remaining 20% is reserved for testing.

Train-Test Split

In [247]:
```python
# Create a Surprise Reader object
reader = Reader(rating_scale=(1, 5))

# Load the data into the Surprise Dataset format
data = Dataset.load_from_df(df[['user_name', 'product_name', 'rating']], reader)

# Split the dataset into training and testing sets
trainset, testset = train_test_split(data, test_size=0.2, random_state=42)

# Train your model using SVD as an example
model = SVD()
model.fit(trainset)

# Make predictions
predictions = model.test(testset)

# Calculate RMSE
rmse_score = rmse(predictions)
print(f"RMSE: {rmse_score}")
```

```
RMSE: 0.2832
RMSE: 0.28324898243610824
```

Figure 13: Train-Test Split Illustration

5.1.2 Model Implementation

Surprise Library Integration The collaborative filtering model is implemented using the Surprise library, following the conversion of the 'score' attribute into numerical data and the preparation of the dataset. The SVD model is specifically utilized for this collaborative filtering process.

Model Evaluation The RMSE is calculated on the training data set using the SVD model.

4. Collaborative Filtering Model Implementation

In [279]:

```python
# Load your dataset
reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(df[['user_name', 'product_name', 'rating']], reader)

# Create and train the recommendation model
model = SVD()
trainset = data.build_full_trainset()
model.fit(trainset)

def suggest_products(df, n=20):
    all_product_names = df['product_name'].unique()
    suggested_products = random.sample(all_product_names.tolist(), n)
    return suggested_products

def get_recommendations(user_name, model, selected_products, df, top_n=5):
    recommendations = []

    # Generate recommendations for products that have not been selected
    for product_name in df['product_name'].unique():
        if product_name not in selected_products:
            prediction = model.predict(user_name, product_name)

            # Include products with a predicted rating above a certain threshold
            if prediction.est > 3.5:
                recommendations.append((product_name, prediction.est))

    # Sort recommendations by predicted rating and return top n
    recommendations.sort(key=lambda x: x[1], reverse=True)
    return recommendations[:top_n]
```

Figure 14: The Collaborative Filtering Model

Recommendation Set Creation Products that have a score exceeding 3.5 are given precedence for the recommendation. A set of recommendations is then formulated for users, considering their individual preferences.

Interface Development The PySimpleGUI library is employed to craft a user-friendly interface. Elements such as windows, buttons, and interactive components are designed to provide an intuitive user experience.

Recommendation Display In the initial display, 20 products are randomly presented to the user. Users are then allowed to select 5 products from which personalized recommendations are generated. In the subsequent display, users are prompted to input a product they are interested in, and 5 recommendations are provided based on their input.

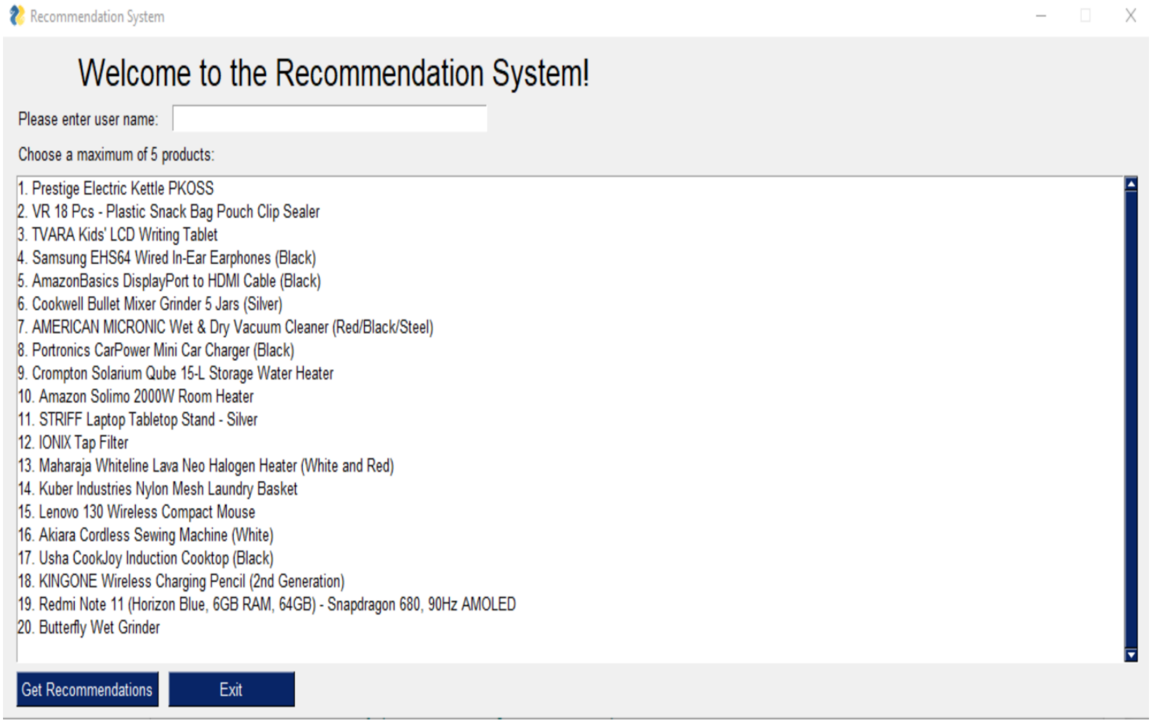


Figure 15: The Collaborative Filtering Model Display

Figure 16: The Collaborative Filtering Model Display

User Interaction and Feedback In the initial phase, a feedback mechanism is introduced with thumbs-up and thumbs-down buttons for capturing user sentiments. This allows users to play an active role in refining the recommendations.

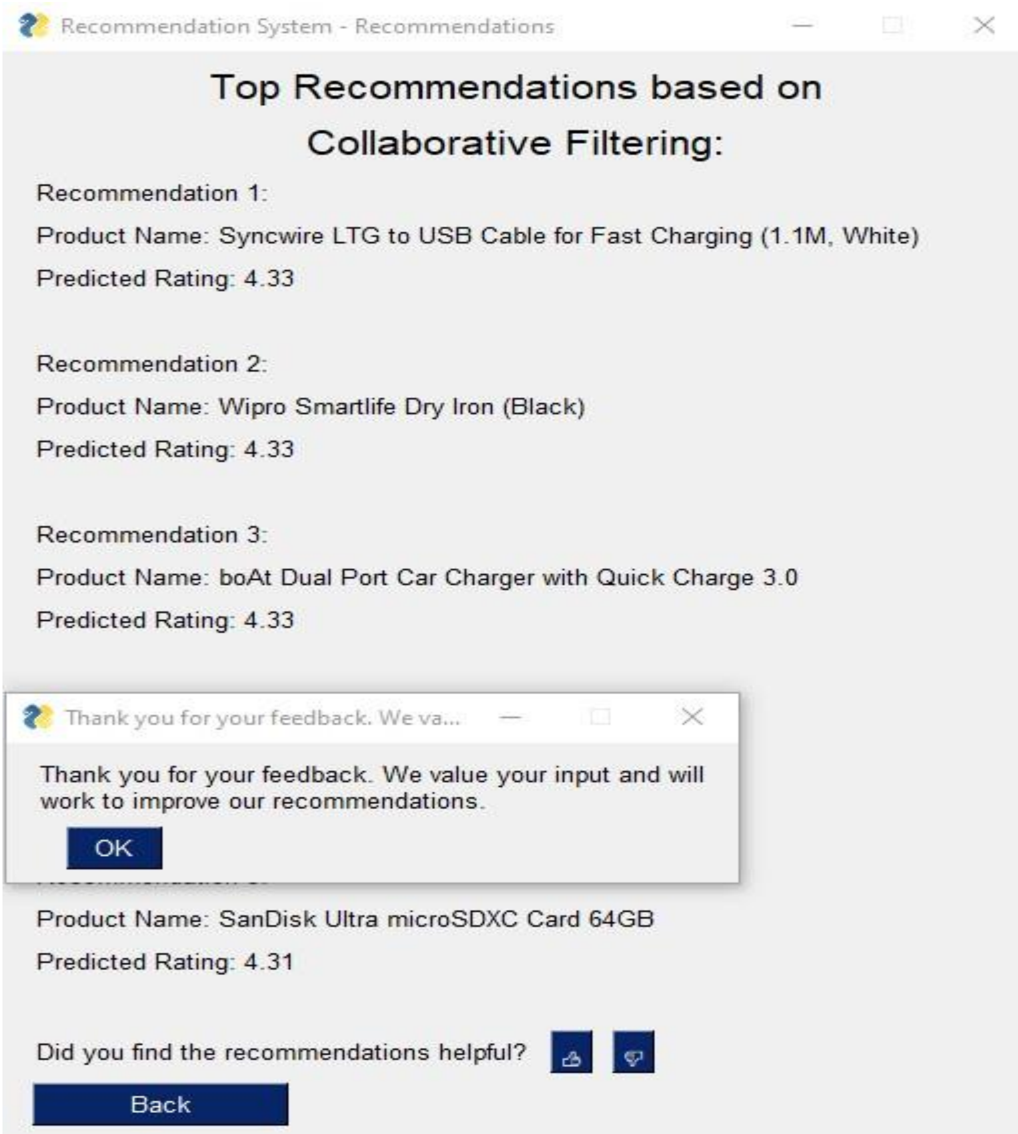


Figure 17: User Interaction and Feedback Process

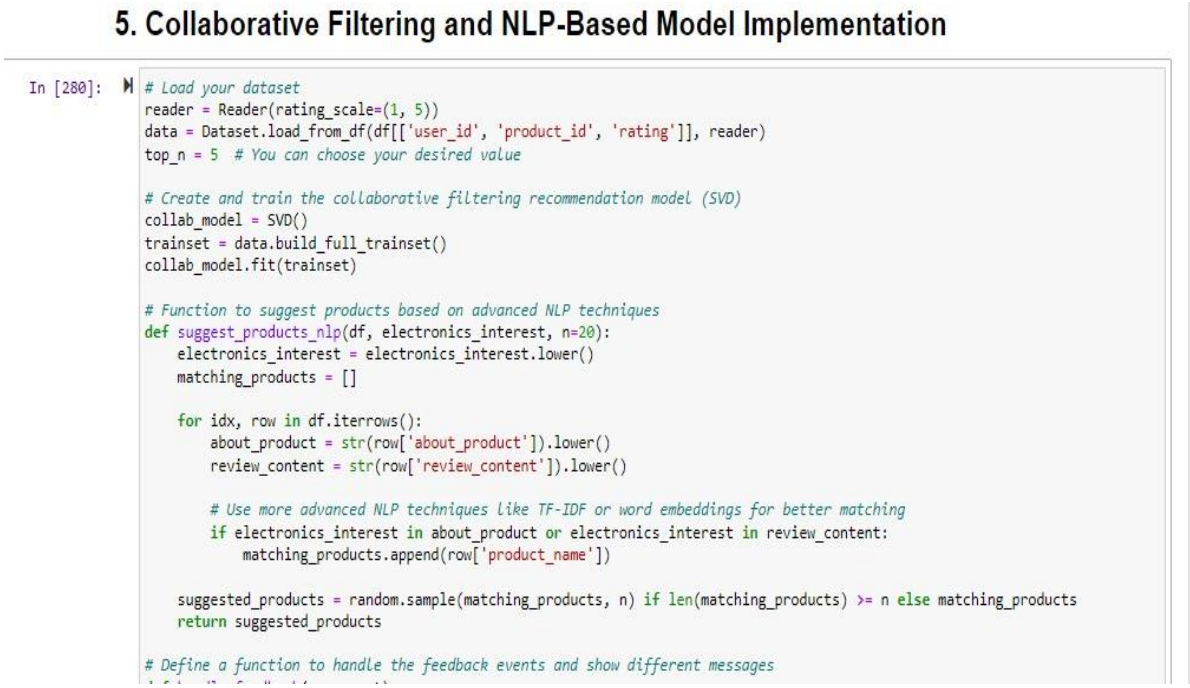
5.2 Steps to Implement the Collaborative Filtering and NLP-Based Model

5.2.1 Data Preprocessing

Text Mining and Similarity Score Calculation Through the application of natural language processing techniques, product reviews and descriptions are analysed. A calculation of similarity scores between products is performed.

5.2.2 Hybrid Model Implementation

Integration with Collaborative Filtering Similarity scores derived from text mining is integrated with collaborative filtering. The result is the formation of a hybrid recommendation model.



5. Collaborative Filtering and NLP-Based Model Implementation

In [280]:

```
# Load your dataset
reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(df[['user_id', 'product_id', 'rating']], reader)
top_n = 5  # You can choose your desired value

# Create and train the collaborative filtering recommendation model (SVD)
collab_model = SVD()
trainset = data.build_full_trainset()
collab_model.fit(trainset)

# Function to suggest products based on advanced NLP techniques
def suggest_products_nlp(df, electronics_interest, n=20):
    electronics_interest = electronics_interest.lower()
    matching_products = []

    for idx, row in df.iterrows():
        about_product = str(row['about_product']).lower()
        review_content = str(row['review_content']).lower()

        # Use more advanced NLP techniques like TF-IDF or word embeddings for better matching
        if electronics_interest in about_product or electronics_interest in review_content:
            matching_products.append(row['product_name'])

    suggested_products = random.sample(matching_products, n) if len(matching_products) >= n else matching_products
    return suggested_products

# Define a function to handle the feedback events and show different messages
```

Figure 18: The Collaborative Filtering and NLP-Based Model Display

Interface Development A scrolling user ID selection screen is displayed, with a prompt for users to 'Please select your user ID.' An input text area is incorporated on the first screen, instructing users to 'Please provide brief details of the electronic products under consideration, such as 'Tablet' or 'Phone.' The system proceeds to the second screen upon user selection and input. Five personalized product recommendations are generated and displayed by the system, based on the chosen User ID and provided product details.

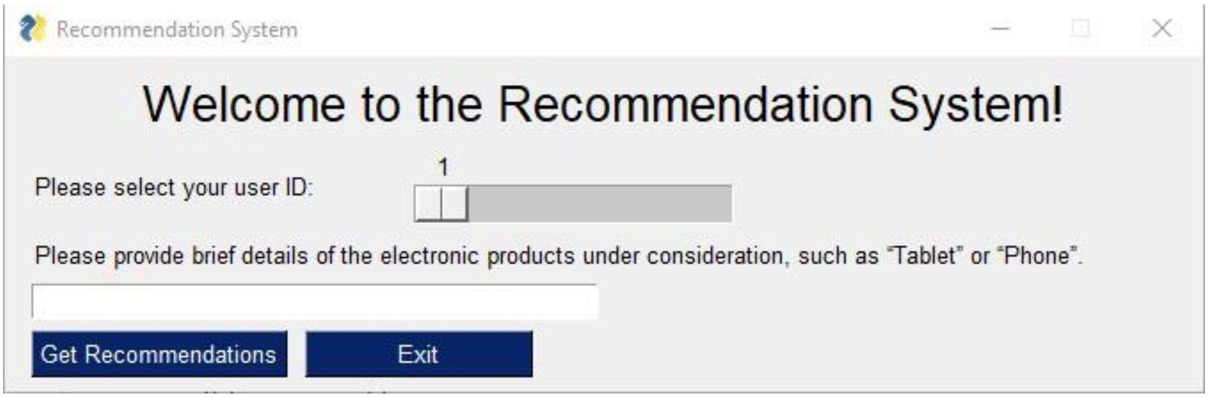


Figure 19: The Collaborative Filtering and NLP-Based Model Display

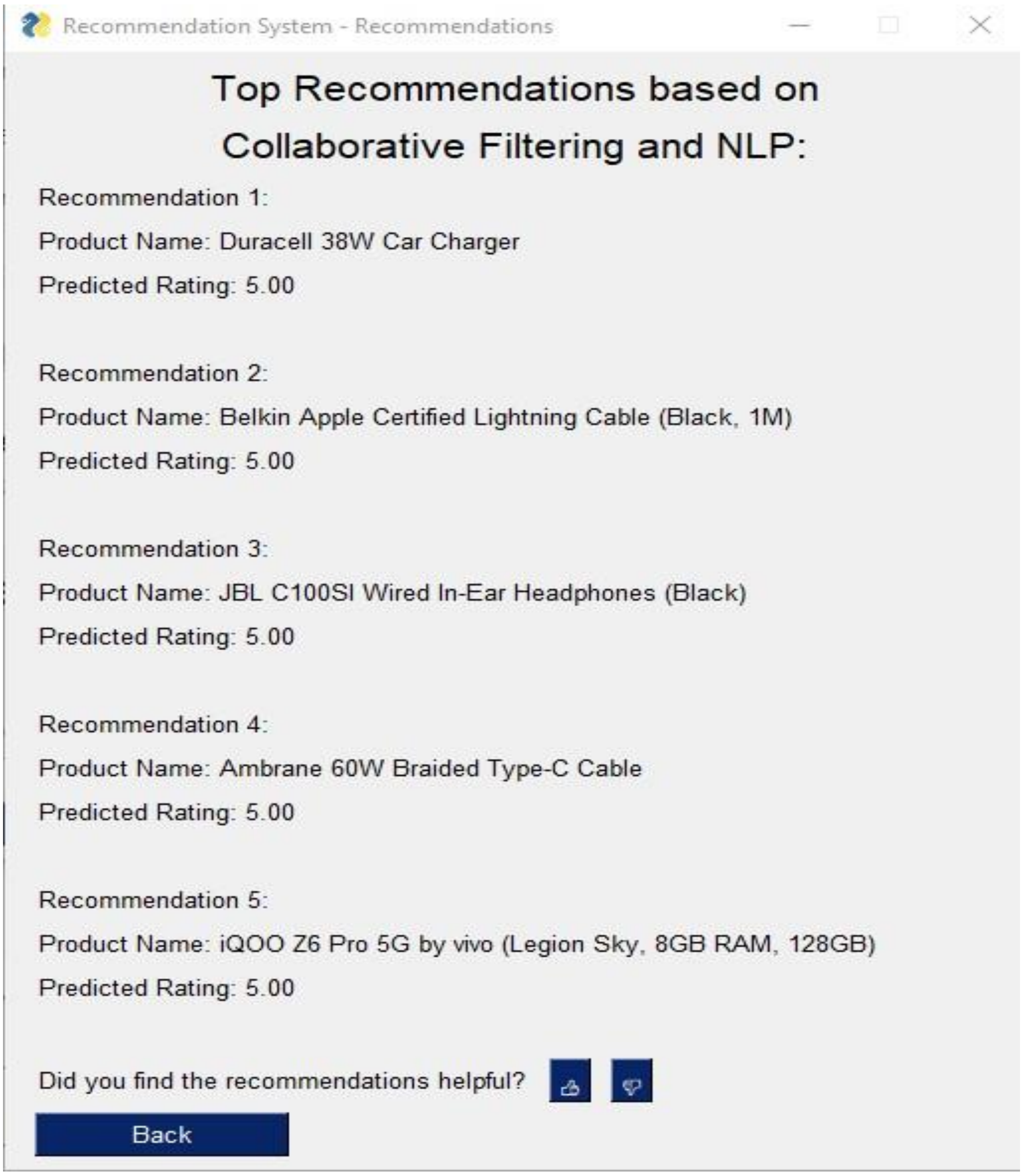


Figure 20: The Collaborative Filtering and NLP-Based Model Display

User Interaction and Feedback A feedback mechanism is established, featuring thumbs-up and thumbs-down buttons to capture user sentiments. This mechanism allows users to actively contribute to the refinement of recommendations.

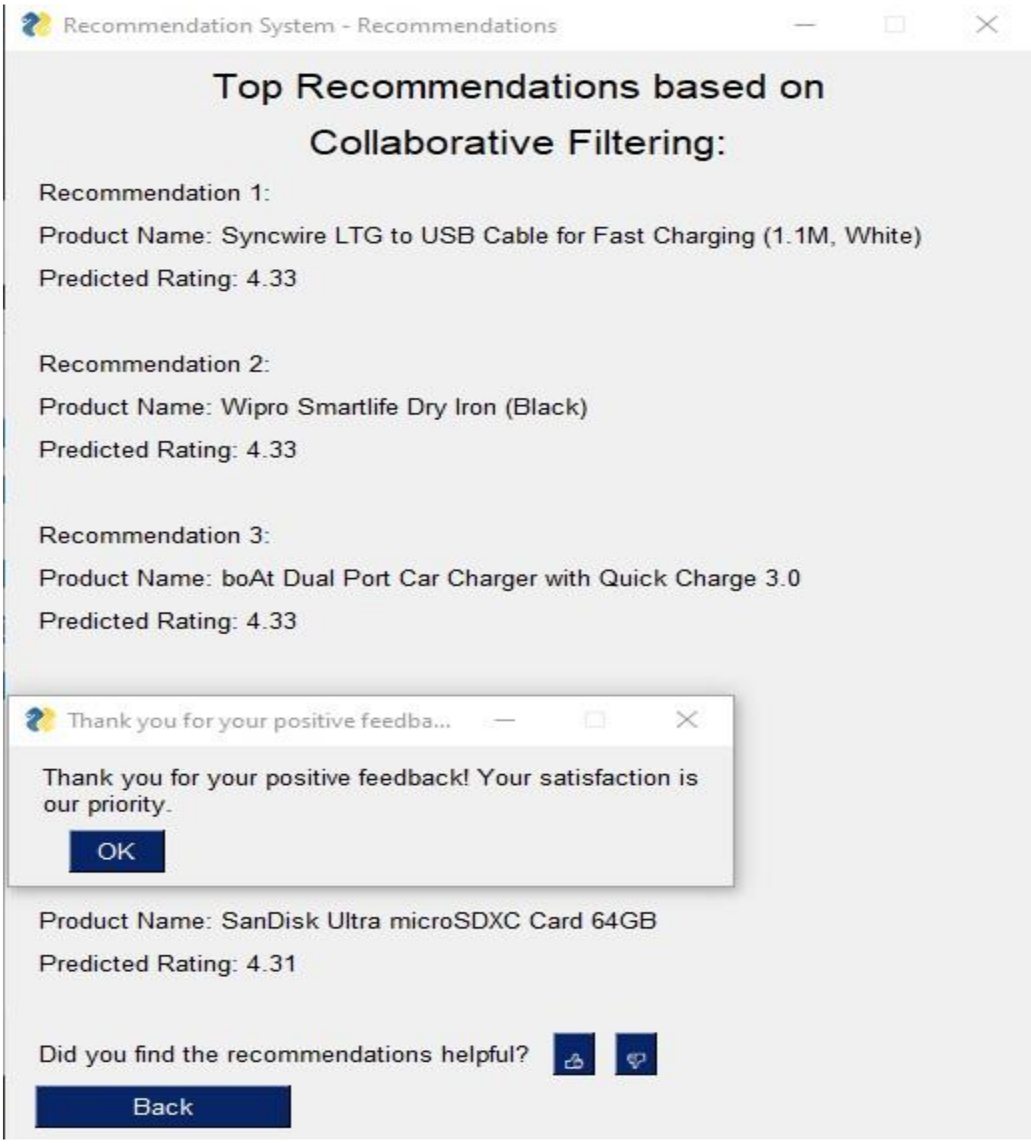


Figure 21: User Interaction and Feedback Process

6 Evaluation of Recommendation Systems

6.1 Survey Design and Distribution

6.1.1 Create a Google Forms Survey

A comprehensive survey, composed of 19 questions, is to be developed. This survey is to be divided into three sections: personal inquiries, an evaluation of the collaborative filtering model, and an assessment of the combined collaborative filtering and NLP model[2].

A random product generator is being prepared using the Python programming language and the PySimpleGUI library framework, utilizing the same dataset for the evaluation of the survey.

Four screen captures related to product recommendation systems usage are being recorded as videos using the ScreenRec program. The videos are then edited using the Microsoft Clipchamp program. Subsequently, the videos are uploaded to a YouTube account and the generated link is included in the survey.

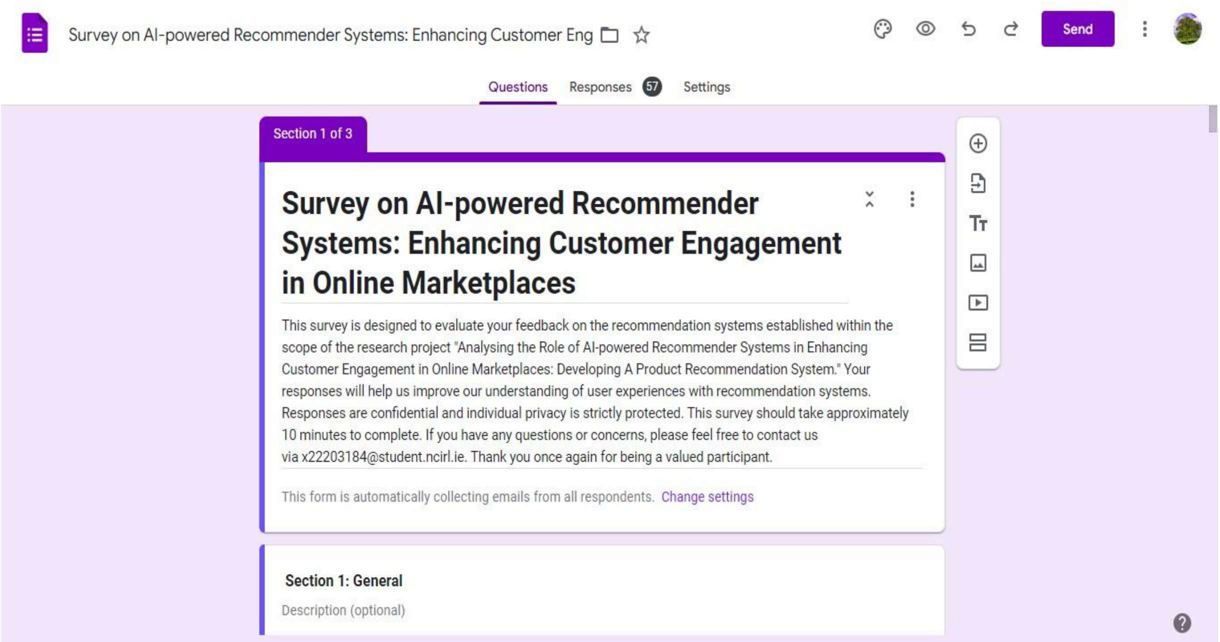


Figure 22: Survey Design and Distribution Process

2 https://forms.gle/81ctiXVPMC26Ytuw9

[Video 1: The Product Recommendation Systems](#) (watch on youtube in 1080 HD quality) (Questions 8-9 pertain directly to the content presented in the video.)

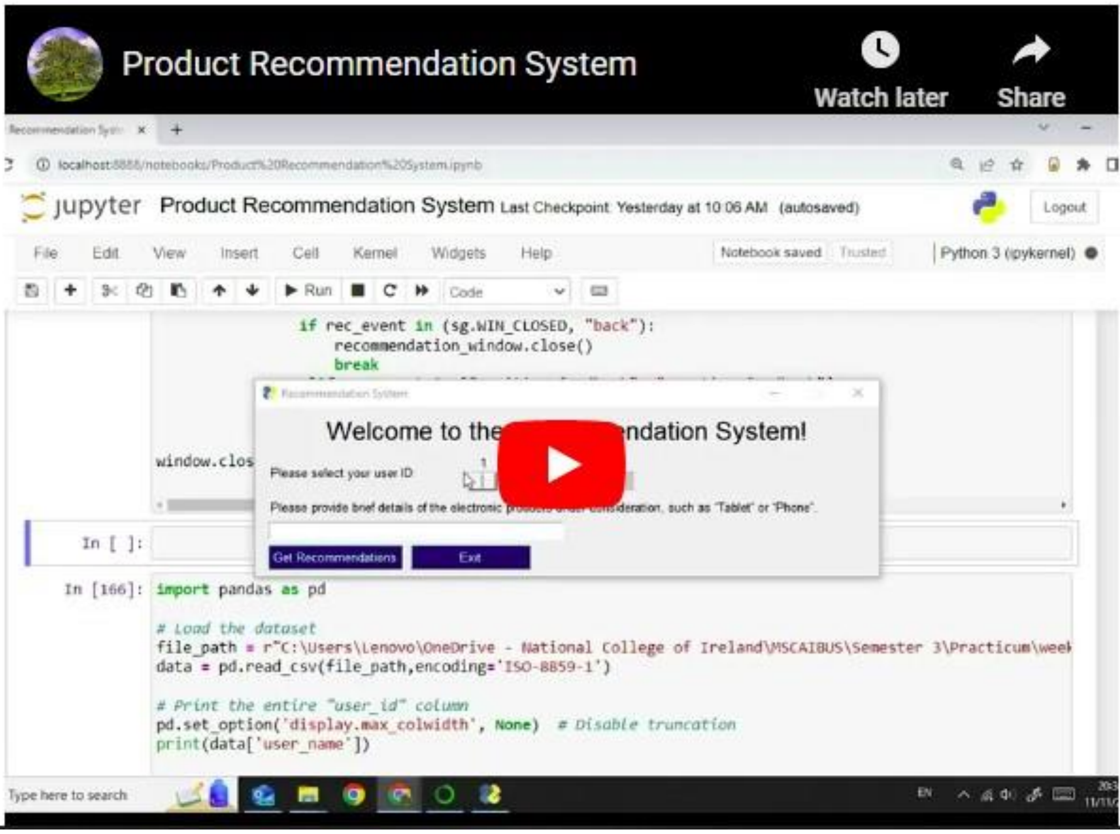


Figure 23: Video Editing and Uploading the Survey

6. Evaluation

The Random Product Generator

In [282]:

```
# The Random Product Generator is designed to be used in the survey that evaluates product recommendation systems.

def generate_random_product_names_from_dataset(data, n=5):
    random_products = random.sample(data['product_name'].tolist(), n)
    return random_products

# Define a function to generate a numbered list of products
def numbered_list(products):
    numbered_products = []
    for i, product in enumerate(products, start=1):
        numbered_products.append(f"{i}. {product}")
    return numbered_products

sg.theme("DefaultNoMoreNagging")

layout = [
    [sg.Text("Welcome to the Random Product Generator!", size=(40, 1), font=("Arial", 20), justification="center")],
    [sg.Text("Generated Product Names:", size=(40, 1), font=("Arial", 16), justification="center")],
    [sg.Listbox(values=[], size=(160, 21), key="selected_products", select_mode="multiple")],
    [sg.Button("Generate Random Products", key="generate_random", size=(25, 1)),
     sg.Button("Exit", key="exit", size=(15, 1))]
]

window = sg.Window("Random Product Generator", layout)

while True:
    event, values = window.read()
```

Figure 24: Random Product Generator Mechanism

6.1.2 Utilize the Snowball Sampling Technique

The survey is to be disseminated online via Google Forms. The snowball sampling technique is to be utilized to ensure a diverse range of participants.

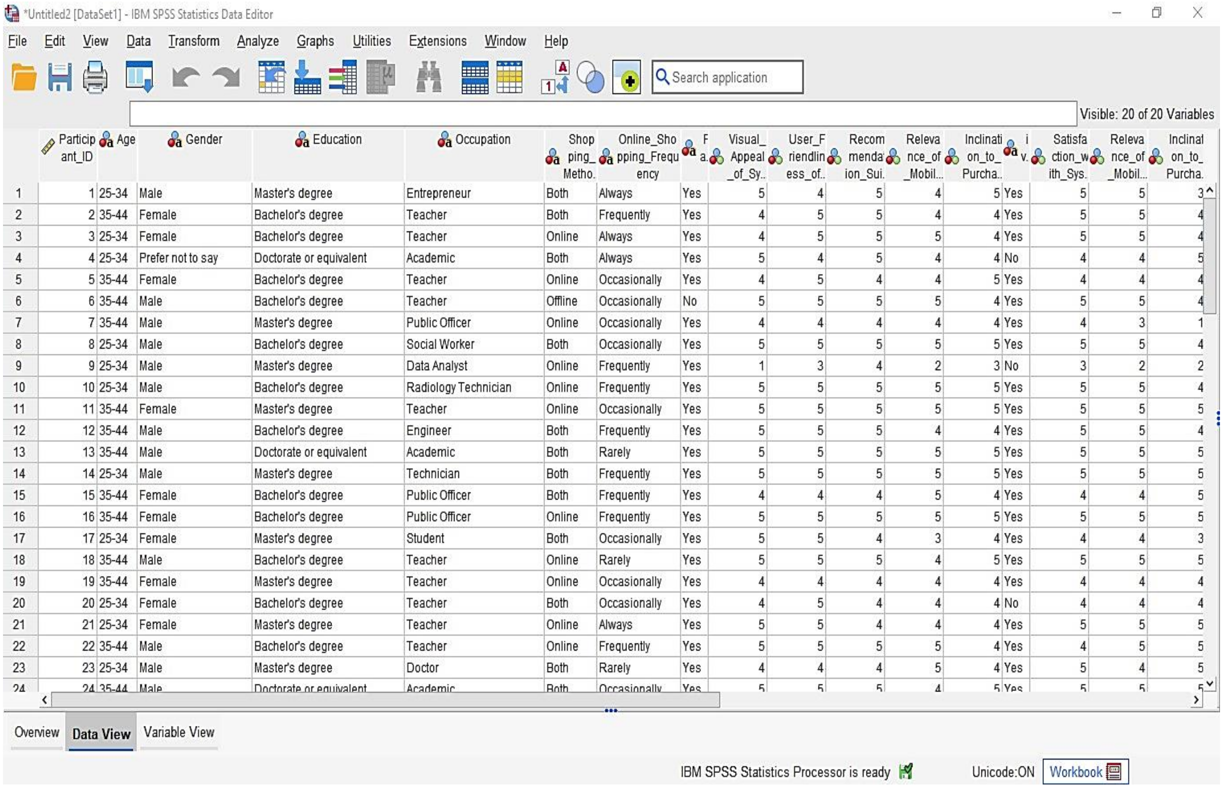
6.1.3 Collect Data from Participants

Responses are to be collected from 57 participants who have verified accounts.

6.2 Data Analysis with IBM SPSS

6.2.1 Prepare and Cleanse Data

IBM SPSS is to be employed for the preparation and cleansing of data. The handling of missing data and the standardization of English for occupational categories are to be carried out.



	Participant_ID	Age	Gender	Education	Occupation	Shopping_Metho.	Online_Shopping_Frequency	Fa.	Visual_Appeal_of_Sy..	User_Friendliness_of..	Recommendation_Sui.	Relevance_of_Mobil...	Inclination_to_Purcha..	iv.	Satisfaction_with_Sys.	Relevance_of_Mobil...	Inclination_to_Purcha.
1	1	25-34	Male	Master's degree	Entrepreneur	Both	Always	Yes	5	4	5	4	5	Yes	5	5	3
2	2	35-44	Female	Bachelor's degree	Teacher	Both	Frequently	Yes	4	5	5	4	4	Yes	5	5	4
3	3	25-34	Female	Bachelor's degree	Teacher	Online	Always	Yes	4	5	5	5	4	Yes	5	5	4
4	4	25-34	Prefer not to say	Doctorate or equivalent	Academic	Both	Always	Yes	5	4	5	4	4	No	4	4	5
5	5	35-44	Female	Bachelor's degree	Teacher	Online	Occasionally	Yes	4	5	4	4	5	Yes	4	4	4
6	6	35-44	Male	Bachelor's degree	Teacher	Offline	Occasionally	No	5	5	5	5	4	Yes	5	5	4
7	7	35-44	Male	Master's degree	Public Officer	Online	Occasionally	Yes	4	4	4	4	4	Yes	4	3	1
8	8	25-34	Male	Bachelor's degree	Social Worker	Both	Occasionally	Yes	5	5	5	5	5	Yes	5	5	4
9	9	25-34	Male	Master's degree	Data Analyst	Online	Frequently	Yes	1	3	4	2	3	No	3	2	2
10	10	25-34	Male	Bachelor's degree	Radiology Technician	Online	Frequently	Yes	5	5	5	5	5	Yes	5	5	4
11	11	35-44	Female	Master's degree	Teacher	Online	Occasionally	Yes	5	5	5	5	5	Yes	5	5	5
12	12	35-44	Male	Bachelor's degree	Engineer	Both	Frequently	Yes	5	5	5	4	4	Yes	5	5	4
13	13	35-44	Male	Doctorate or equivalent	Academic	Both	Rarely	Yes	5	5	5	5	5	Yes	5	5	5
14	14	25-34	Male	Master's degree	Technician	Both	Frequently	Yes	5	5	5	5	5	Yes	5	5	5
15	15	35-44	Female	Bachelor's degree	Public Officer	Both	Frequently	Yes	4	4	4	5	4	Yes	4	4	5
16	16	35-44	Female	Bachelor's degree	Public Officer	Online	Frequently	Yes	5	5	5	5	5	Yes	5	5	5
17	17	25-34	Female	Master's degree	Student	Both	Occasionally	Yes	5	5	4	3	4	Yes	4	4	3
18	18	35-44	Male	Bachelor's degree	Teacher	Online	Rarely	Yes	5	5	5	4	5	Yes	5	5	5
19	19	35-44	Female	Master's degree	Teacher	Online	Occasionally	Yes	4	4	4	4	4	Yes	4	4	4
20	20	25-34	Female	Bachelor's degree	Teacher	Both	Occasionally	Yes	4	5	4	4	4	No	4	4	4
21	21	25-34	Female	Master's degree	Teacher	Online	Always	Yes	5	5	4	4	4	Yes	5	5	5
22	22	35-44	Male	Bachelor's degree	Teacher	Online	Frequently	Yes	5	5	5	5	4	Yes	4	5	5
23	23	25-34	Male	Master's degree	Doctor	Both	Rarely	Yes	4	4	4	5	4	Yes	5	4	5
24	24	35-44	Male	Doctorate or equivalent	Academic	Both	Occasionally	Yes	5	5	5	4	5	Yes	5	5	5

Figure 25: Data Preparation and Cleansing Workflow using IBM SPSS

6.2.2 Descriptive Statistics and Frequency Distribution

The examination of descriptive statistics and the frequency distribution of survey responses is to be conducted.

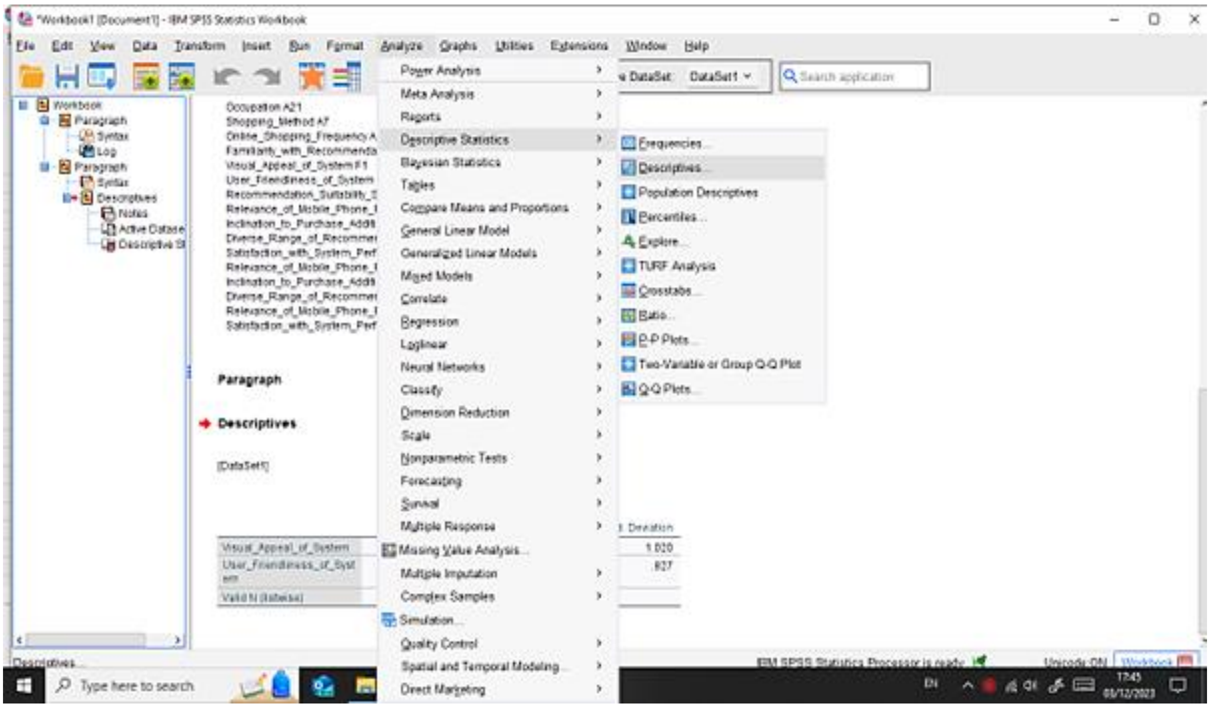


Figure 26: Descriptive Statistics

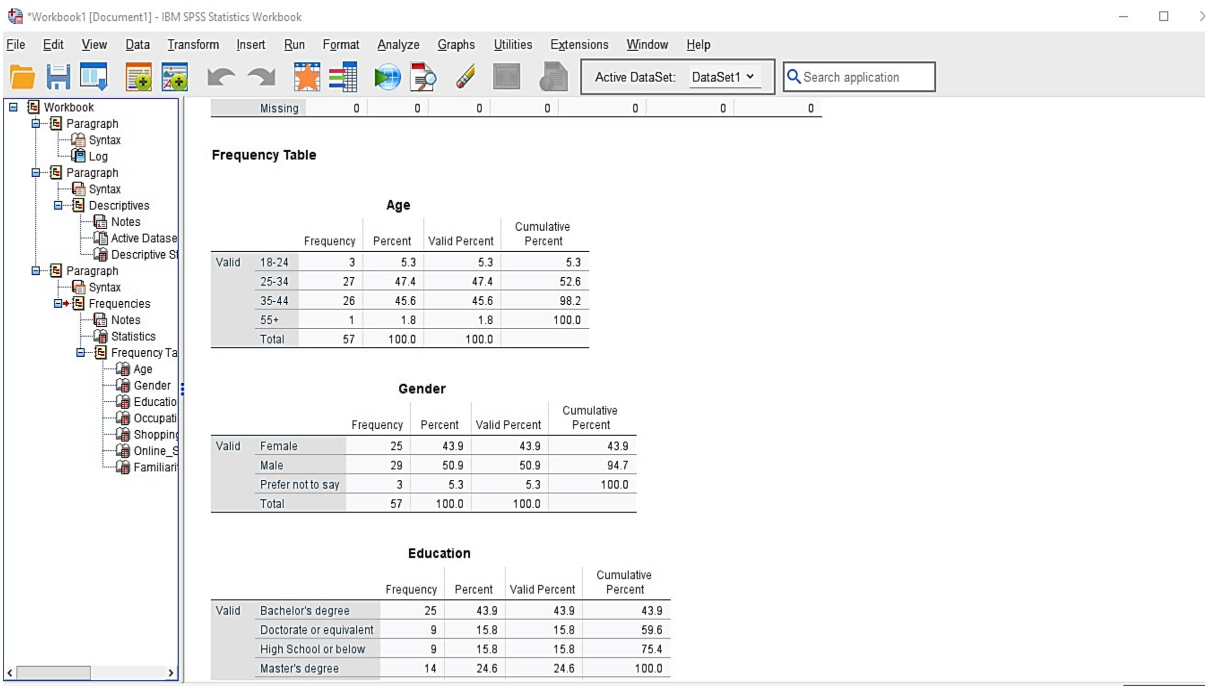


Figure 27: Frequency Distribution Analysis

6.2.3 Correlation Analysis and Paired Samples T-test Analysis

The correlation coefficients are utilized for the analysis of attribute correlations. A comparison of attributes between collaborative filtering and hybrid NLP-based models is conducted through a paired samples t-test.

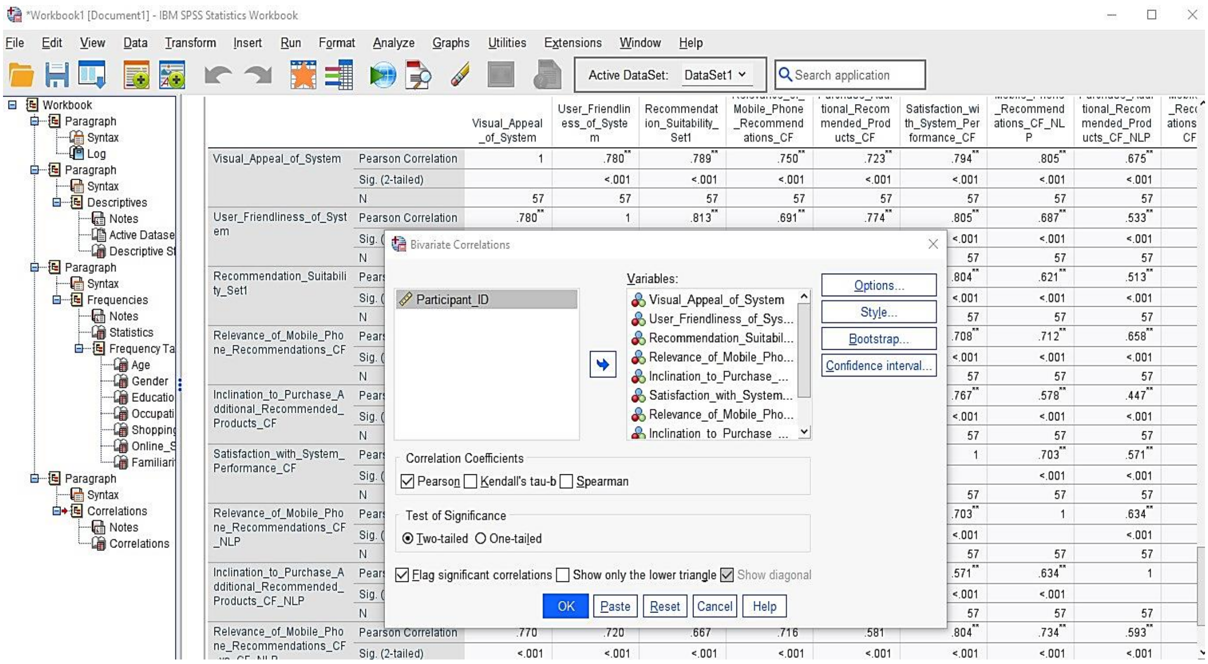


Figure 28: Correlation Analysis

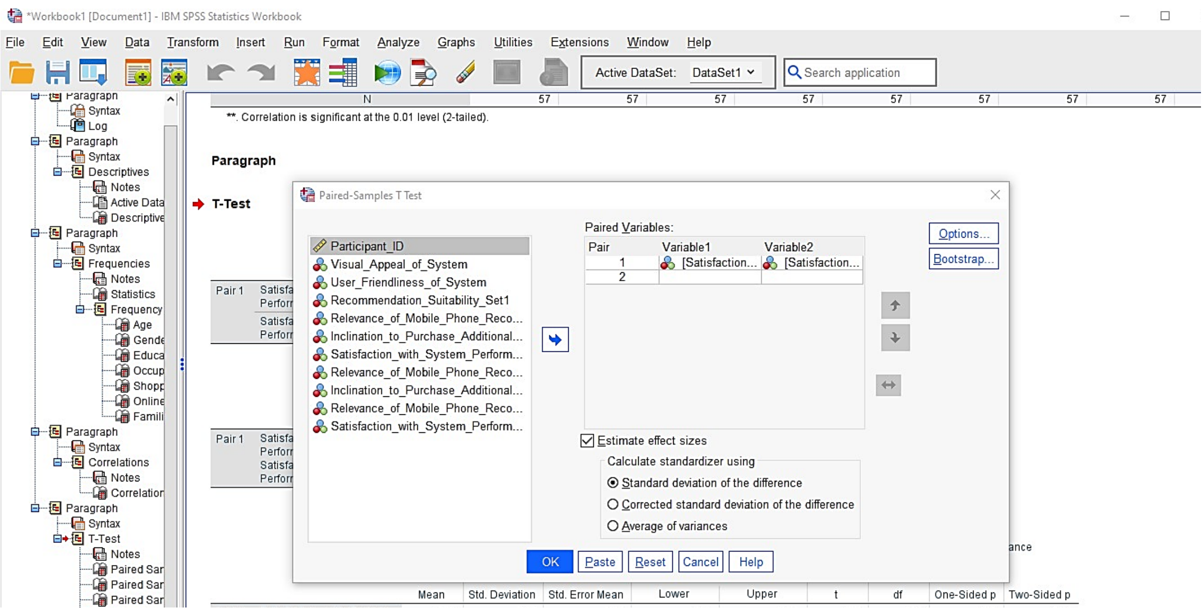


Figure 29: Paired Samples T-test Analysis

6.2.4 Interpret and Discuss Findings

The implications are discussed, and the statistical results are interpreted. Insights into the performance of collaborative filtering and hybrid models are provided.

7 References