National College of Ireland

Optimizing Personalization in E-commerce Platforms using Artificial Intelligence and Machine Learning Techniques Configuration Manual

MSc Research Project
Data Analytics

Alper Bayram
Student ID: 22121773

School of Computing
National College of Ireland

Supervisor: Dr. Anu Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Alper Bayram
Student ID:	22121773
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Anu Sahni
Submission Due Date:	06/03/2024
Project Title:	Optimizing Personalization in E-commerce Platforms using Artificial Intelligence and Machine Learning Techniques Configuration Manual
Word Count:	XXX
Page Count:	5

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:	5th March 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):	

Optimizing Personalization in E-commerce Platforms using Artificial Intelligence and Machine Learning Techniques Configuration Manual

Alper Bayram
MSc Data Analytics
National College of Ireland

1 Introduction

Included in the setup handbook are libraries, software and hardware settings, and crucial code snippets for each implementation phase.

2 System Requirement

2.1 Hardware Requirement

Processor: Intel® Core™ i3 Processor (Minimum required processor)
RAM: 4GB

These configurations are minimum required configuration for hardware requirement.

2.2 Software Requirement

Python programming language has been used. Python is an interpreted programming language that is dynamic, high-level, and general-purpose Mane (2017). The Object-Oriented programming technique is supported for application development. It offers a plethora of high-level data structures and is both easy to learn and use.

An appealing language for Application Development, Python is easy-to-learn but powerful and flexible.

Google Colab tool has been used to code. Open source and freely available, Jupyter Notebook was developed by the Jupyter Project. A Jupyter notebook is similar to an interactive lab notebook in that it contains data, comments, and code to edit the data. You may run the code in the notebook and then save the results.

2.3 Library and Packages

pandas

matplotlib
seaborn
numpy
time
scikit-surprise-1.1.3
scipy-1.3.2

3 Research Implementation

3.1 Importing libraries

```
# import necessay linraries
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import time
from sklearn.model_selection import train_test_split
from surprise import Dataset
from surprise import Reader
from surprise import SVD, NMF, SlopeOne, KNNBasic, KNNWithMeans, KNNBaseline, CoClustering, BaselineOnly, NormalPredictor
from surprise.model_selection import cross_validate
from surprise.model_selection import KFold
from surprise.model_selection import GridSearchCV, RandomizedSearchCV
from surprise import accuracy, Dataset, Reader, SVD
from surprise.model_selection import PredefinedKFold
```

Figure 1: Importing Libraries

This section is associated with importing all the necessary libraries to perform the practical.

3.2 Loading data and displaying

```
[ ] # load data
    data_path = '/content/drive/MyDrive/Recommendation/Home-Data'

    train = pd.read_csv(f'{data_path}/train.csv', encoding = "ISO-8859-1" )
    test = pd.read_csv(f'{data_path}/test.csv', encoding = "ISO-8859-1")
```

```
[ ] print('Movie information shape: ', train.shape)
    print('number of unique product id: ', train['product_uid'].nunique())
    print('number of unique search term:', train['search_term'].nunique())
    train.head()
```

```
Movie information shape:  (74067, 5)
number of unique product id:  54667
number of unique search term: 11795
```

	id	product_uid	product_title	search_term	relevance
0	2	100001	Simpson Strong-Tie 12-Gauge Angle	angle bracket	3.00
1	3	100001	Simpson Strong-Tie 12-Gauge Angle	l bracket	2.50
2	9	100002	BEHR Premium Textured DeckOver 1-gal. #SC-141 ...	deck over	3.00
3	16	100005	Delta Vero 1-Handle Shower Only Faucet Trim Ki...	rain shower head	2.33
4	17	100005	Delta Vero 1-Handle Shower Only Faucet Trim Ki...	shower only faucet	2.67

Figure 2: Data Loading and displaying

The above section shows the data loading as well as data displaying.

3.3 Visualization

```
x = train.groupby('search_term')['product_uid'].count()
plt.hist(x , bins = 20)
plt.title("Distebution of product per search term");
```

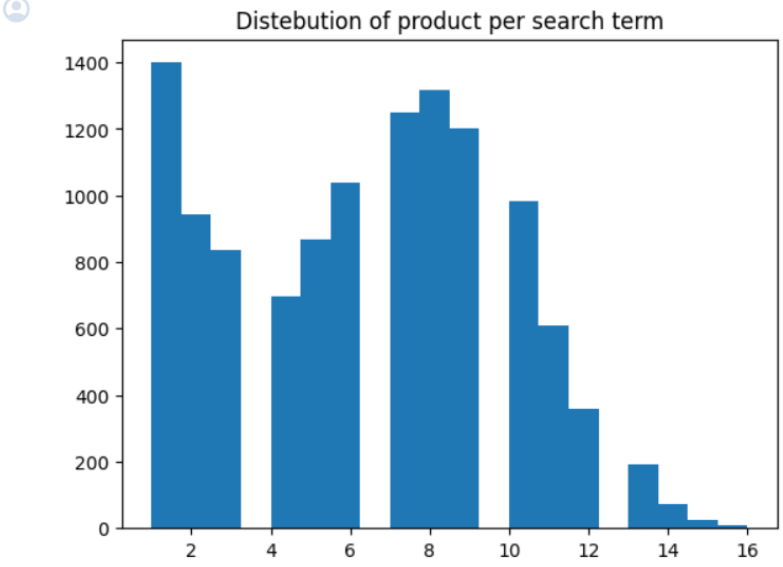

Figure 3: Distribution of product per search term

+ Code + Text

Preprocesing

```
# convert search term to numerical
from sklearn.preprocessing import LabelEncoder

# create label encoder.
encoder = LabelEncoder()
train['search_term_uid']= encoder.fit_transform(train['search_term'])
train.head()
```

	id	product_uid	product_title	search_term	relevance	search_term_uid
0	2	100001	Simpson Strong-Tie 12-Gauge Angle	angle bracket	3.00	2262
1	3	100001	Simpson Strong-Tie 12-Gauge Angle	l bracket	2.50	6560
2	9	100002	BEHR Premium Textured DeckOver 1-gal. #SC-141 ...	deck over	3.00	4001
3	16	100005	Delta Vero 1-Handle Shower Only Faucet Trim Ki...	rain shower head	2.33	8732
4	17	100005	Delta Vero 1-Handle Shower Only Faucet Trim Ki...	shower only faucet	2.67	9587

Figure 4: Preprocessing steps

The above section is shows about data pre-processing steps.

4 Model Evaluation

Algorithms use SVD, SlopeOne, NMF, KNNBaseline

```
# try different models on train data to find the best model
Results = pd.DataFrame(columns = ['Model','RMSE','MAE','Duration'])
models = (SVD, SlopeOne, NMF, KNNBaseline)
models_name = ['SVD', 'SlopeOne', 'NMF', 'KNNBaseline']
# Set number of itteration while doing cross validation
kf = KFold(4, random_state=0)
# itterate in models list and run all the model and calculate the perormance of each model
for model, name in zip(models, models_name):
    t0 = time.time()
    output = cross_validate(model(), train_data, ['rmse', 'mae'], kf)
    mean_rmse = '%.3f' % np.mean(output['test_rmse'])
    mean_mae = '%.3f' % np.mean(output['test_mae'])
    duration = round((time.time() - t0 ) / 60, 2)
    Results = Results.append({'Model': name, 'RMSE': mean_rmse, 'MAE': mean_mae, 'Duration': duration}, ignore_index=True)
# print the results
print("\nThe results of models:\n")
Results
```

```
The results of models:

<ipython-input-13-e045c524e536>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version.
  Results = Results.append({'Model': name, 'RMSE': mean_rmse, 'MAE': mean_mae, 'Duration': duration}, ignore_index=True)
```

	Model	RMSE	MAE	Duration
0	SVD	0.506	0.414	0.03
1	SlopeOne	0.585	0.452	0.21
2	NMF	0.655	0.529	0.13
3	KNNBaseline	0.542	0.435	0.10

As per results the best model with lower MAE is SVD then we tune the model

Figure 5: Result Comparison of SVD, SlopeOne, NMF, KNNBaseline

The above image is showing the result comparison of SVD, SlopeOne, NMF, KNNBaseline Wang (2021). The parameters like RMSE, MAE, and duration has been compared.

Prediction

```
def get_recommendation(search_term_uid , n):
    '''
    This function use to get the recommendation search term for a specific products
    :param
      search term id: the identification code of search term
      n : number of product to recommend
    return: list of n product that system recommend
    '''
    # filter the product-searchterm selected before
    search_term_product_list = train_cleaned[train_cleaned['search_term_uid']== search_term_uid]

    # load the user data in surprise format
    reader = Reader(rating_scale=(1.0,3.0))
    data = Dataset.load_from_df(train_cleaned[['search_term_uid' , 'product_uid', 'relevance']], reader)

    # train the model

    model = SVD(n_factors = 50, n_epochs= 40, lr_all = 0.005)

    train_data = data.build_full_trainset()
    model.fit(train_data)
```

Figure 6: Steps for prediction

```
# get top n recommendation
products_uid = train_cleaned['product_uid'].unique().tolist()
for product_uid in search_term_product_list['product_uid'].to_list():
  if product_uid in products_uid :
    products_uid.remove(product_uid)

# predict rating for user
predictions = []
for product_uid in products_uid:
  predictions.append((product_uid, model.predict(int(search_term_uid), int(search_term_uid)).est))

#predictions = model.test(test_data)
# prepare recommendation output
top_n = sorted(predictions, key = lambda x: x[1], reverse = True)
result = pd.DataFrame(top_n, columns = ['product_uid', 'relevance'])
top_n_products = [x[0] for x in top_n][:n]
top_n_products_name = []
for id in top_n_products:
  top_n_products_name.append(train_cleaned[train_cleaned['product_uid']== id]['product_title'].values)
return top_n_products_name
```

Figure 7: Steps for prediction

```
[ ] # user interface
search_term = input('please enter search term: ')
n = int(input('please enter number on product recommendation: '))

recommend_products_list = get_recommendation(train_cleaned[train_cleaned['search_term'] == search_term]['search_term_uid'].values[0], n)
print(f'\nRecommended products for search term: {search_term}')
for name in  recommend_products_list:
    print("- ", name[0])
```

```
please enter search term: angle bracket
please enter number on product recommendation: 5

Recommended products for search term: angle bracket
-  Delta Vero 1-Handle Shower Only Faucet Trim Kit in Chrome (Valve Not Included)
-  Whirlpool 1.9 cu. ft. Over the Range Convection Microwave in Stainless Steel with Sensor Cooking
-  Toro Personal Pace Recycler 22 in. Variable Speed Self-Propelled Gas Lawn Mower with Briggs &amp;amp; Stratton Engine
-  MD Building Products 36 in. x 36 in. Cloverleaf Aluminum Sheet, Silver
-  House of Fara 8 Linear ft. MDF Overlapping Wainscot Interior Paneling Kit
```

Figure 8: Steps for user interface

The above image is for user interface. It is designed to take input from user and provide the output to user as well after processing through ML model.

References

Mane, P. (2017). Predictive accuracy of recommendation algorithms, *Doctoral dissertation, Purdue University* .

Wang, Y. (2021). Research on recommendation algorithm based on collaborative filtering of fusion model., *In Journal of Physics: Conference Series* **1774**(1): p. 012058.