National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Chethan Marigowda
StudentID:x21145377

School of Computing
National College of Ireland

Supervisor: Dr. Paul Stynes

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Chethan Marigowda
Student ID:	x21145377
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Paul Stynes
Submission Due Date:	15/12/2022
Project Title:	Configuration Manual
Word Count:	1309
Page Count:	15

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:	Chethan Marigowda
Date:	15th December 2022

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

Chethan Marigowda
x21145377

1 Introduction

This research experiment followed a specific implementation configuration, and the objective of this guide is to describe how that configuration was set up. This documentation provides comprehensive details of the software, hardware and library configurations utilized in the development of this project. In addition to this, it details the method of programming as well as the procedure that should be carried out in order to execute the code.

2 Local Computing Machine System Configuration

The Figure 1 shows system configuration used in this project.

Fig. 1. System Configuration

3 Software, Packages and Library.

Hybrid Machine Learning Framework to Recommend E-Commerce Products is implemented in open-source freeware python programming language on Google Colaboratory notebook. Following Fig 1 in the fig details important packages/libraries used for this project. In the first step, all the necessary libraries and modules are installed and imported in the notebook. Some of the major libraries that were imported include Keras, Pandas, NumPy among, genism and others. With help of these libraries, necessary functions to carry out data manipulation and machine learning related computations can be carried out

```
[ ]  #import libraries
     import json
     import numpy as np
     from collections import defaultdict
     import warnings
     import gensim
     import pandas as pd
     from gensim.models.doc2vec import Doc2Vec
     import numpy as np
     import keras
     import math
     import gzip
     from keras import backend as K
     from keras.models import Model
     from keras.layers import Embedding, Input, Dense, Flatten, Concatenate, Multiply, Lambda, Reshape
     from keras.layers.core import Dropout
     import scipy.sparse as sp
     import matplotlib.pyplot as plt
     from sklearn.metrics import mean_squared_error, mean_absolute_error
     from tensorflow.keras.callbacks import EarlyStopping , ModelCheckpoint
     import math as mt
```

Fig. 2. Libraries used

4 Notebook Execution Guidance.

The research was conducted using free software provided by Google called Colaboratory notebook, which does not require any static setup settings to be entered. The modest addition of this project is extensive code that enables it to execute by sourcing files directly from the source destination. This is the minor contribution of this research and useful in generating required product categories files.

The standard steps to be followed to execute the code from the fig 3 to 4 for the end-to-end execution.

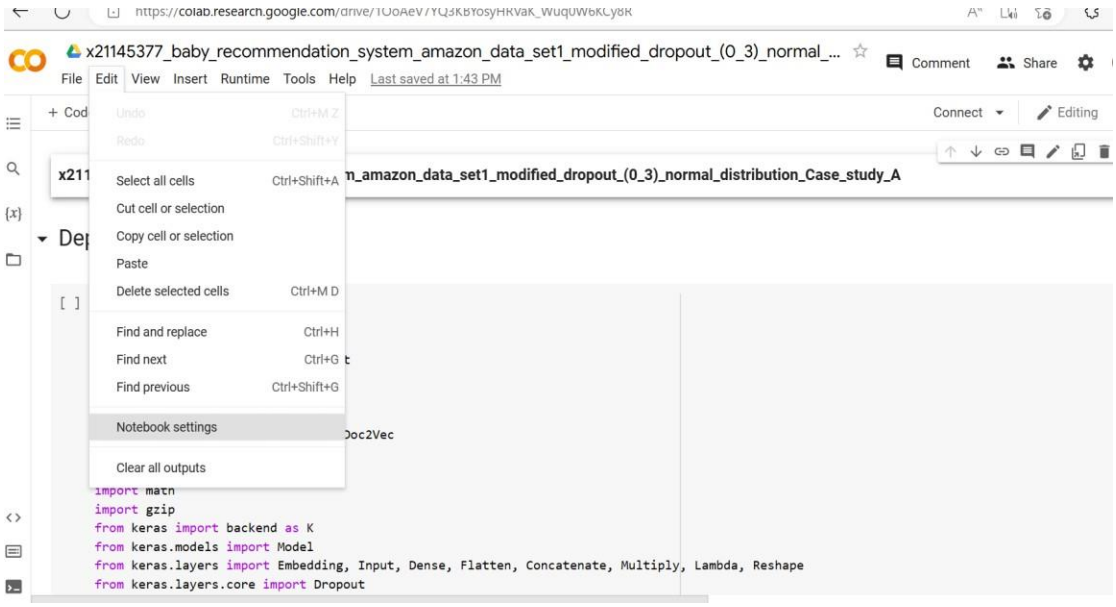


Fig. 3. Execution step1

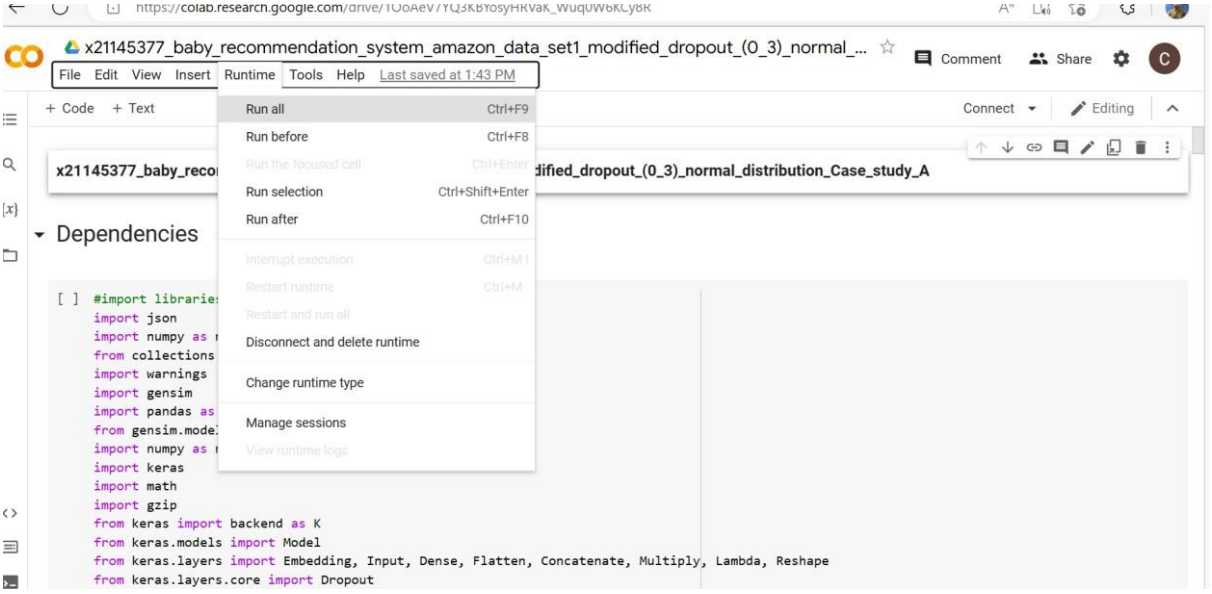


Fig. 4. Execution step2

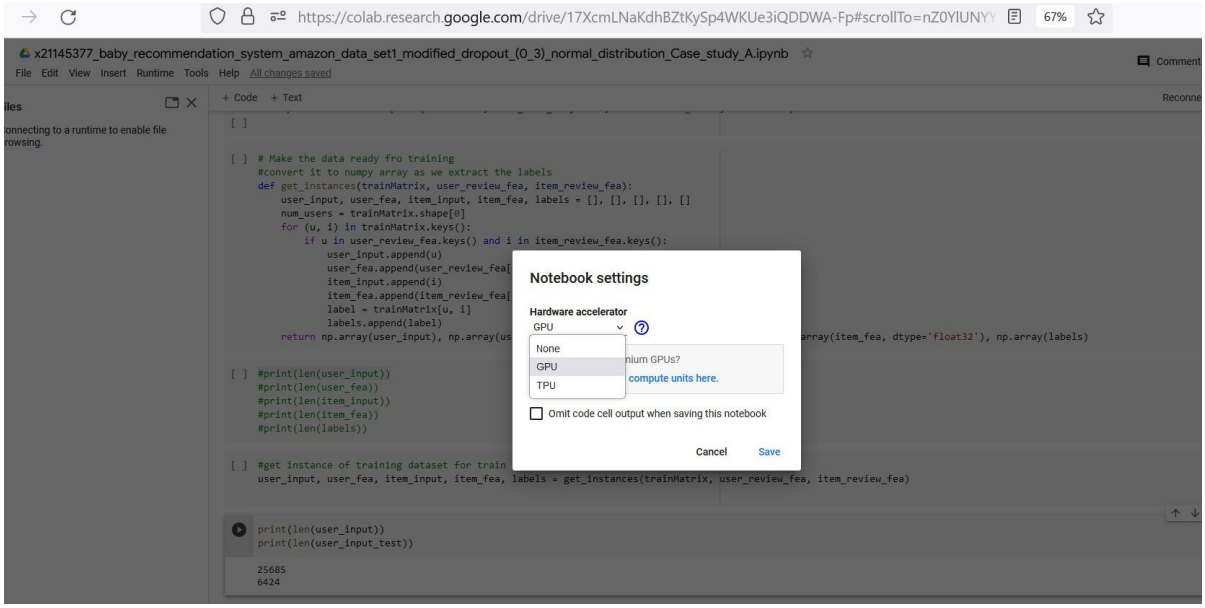


Fig. 5. Execution step3

5 Data Acquisition

The dataset collected from public open-source web server, after decompression contains 915445 consumer reviews and metadata of various baby products up to 71316 in semi structured json format for Data set 1[1]. Data set 2[2] contains 836005 consumer reviews and metadata of various digital music products up to 279898 in semi structured json format. Wget is a networking command-line tool that lets you download files and interact with REST APIs.

Display the dataset

```
[ ]  ## Convert data from gz to DataFrame
     def parse(path):
         g = gzip.open(path, 'rb')
         for l in g:
             yield eval(l)

     def getDF(path):
         i = 0
         df = {}
         for d in parse(path):
             df[i] = d
             i += 1
         return pd.DataFrame.from_dict(df, orient='index')
```

Fig. 6. Data acquisition illustration1

Download The Review and Meta Data

```
[ ]  # Download Review Data
     !wget http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Baby.json.gz

     --2022-12-14 18:05:23--  http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Baby.json.gz
     Resolving snap.stanford.edu (snap.stanford.edu)... 171.64.75.80
     Connecting to snap.stanford.edu (snap.stanford.edu)|171.64.75.80|:80... connected.
     HTTP request sent, awaiting response... 200 OK
     Length: 191601185 (183M) [application/x-gzip]
     Saving to: ‘reviews_Baby.json.gz.3’

     reviews_Baby.json.g 100%[===================>] 182.72M  4.60MB/s    in 34s

     2022-12-14 18:05:57 (5.42 MB/s) - ‘reviews_Baby.json.gz.3’ saved [191601185/191601185]
```

Fig. 7. Data acquisition illustration 2

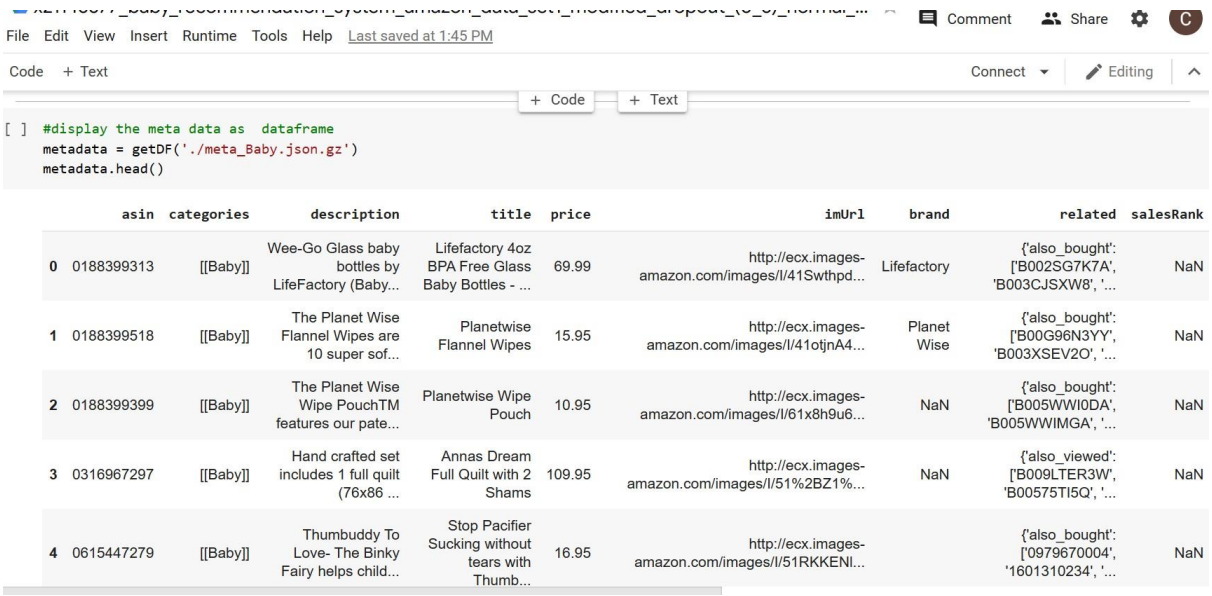
```
[ ]  # Download Meta Data
     !wget http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/meta_Baby.json.gz

     --2022-12-14 18:05:57--  http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/meta_Baby.json.gz
     Resolving snap.stanford.edu (snap.stanford.edu)... 171.64.75.80
     Connecting to snap.stanford.edu (snap.stanford.edu)|171.64.75.80|:80... connected.
     HTTP request sent, awaiting response... 200 OK
     Length: 31328004 (30M) [application/x-gzip]
     Saving to: ‘meta_Baby.json.gz.3’

     meta_Baby.json.gz.3 100%[===================>]  29.88M  2.11MB/s    in 13s

     2022-12-14 18:06:10 (2.30 MB/s) - ‘meta_Baby.json.gz.3’ saved [31328004/31328004]
```

Fig. 8. Data acquisition illustration 3



	asin	categories	description	title	price	imUrl	brand	related	salesRank
0	0188399313	[[Baby]]	Wee-Go Glass baby bottles by LifeFactory (Baby...	Lifefactory 4oz BPA Free Glass Baby Bottles - ...	69.99	http://ecx.images-amazon.com/images/I/41Swthpd...	Lifefactory	{'also_bought': ['B002SG7K7A', 'B003CJSXW8', '...	NaN
1	0188399518	[[Baby]]	The Planet Wise Flannel Wipes are 10 super sof...	Planetwise Flannel Wipes	15.95	http://ecx.images-amazon.com/images/I/41otjnA4...	Planet Wise	{'also_bought': ['B00G96N3YY', 'B003XSEV2O', '...	NaN
2	0188399399	[[Baby]]	The Planet Wise Wipe PouchTM features our pate...	Planetwise Wipe Pouch	10.95	http://ecx.images-amazon.com/images/I/61x8h9u6...	NaN	{'also_bought': ['B005WWI0DA', 'B005WWIMGA', '...	NaN
3	0316967297	[[Baby]]	Hand crafted set includes 1 full quilt (76x86 ...	Annas Dream Full Quilt with 2 Shams	109.95	http://ecx.images-amazon.com/images/I/51%2BZ1%...	NaN	{'also_viewed': ['B009LTER3W', 'B00575TI5Q', '...	NaN
4	0615447279	[[Baby]]	Thumbuddy To Love- The Binky Fairy helps child...	Stop Pacifier Sucking without tears with Thumb...	16.95	http://ecx.images-amazon.com/images/I/51RKKENl...		{'also_bought': ['0979670004', '1601310234', '...	NaN

Fig. 9. Data acquisition illustration 4

```
[ ] #display the review data as  dataframe
    reviews = getDF('./reviews_Baby.json.gz')
    reviews.head()
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A28O3NP6WR5517	0188399313	Jennifer gymer	[0, 0]	They work very well. Easy to clean, we wash th...	5.0	These bottles are great!	1369612800	05 27, 2013
1	AX0M1Z6ZWO52J	0188399399	Ash M.	[1, 1]	it came early and was not disappointed. i love...	5.0	perfect	1365465600	04 9, 2013
2	A1KD7N84L7NIUT	0188399518	Buffy Fan	[0, 0]	I ended up with a variety of different brands ...	4.0	Good cotton wipes	1392336000	02 14, 2014
3	A29CUDEIF4X1UO	0188399518	J. D. Solbach	[1, 1]	These flannel wipes are OK, but in my opinion ...	3.0	Sending them back.	1373241600	07 8, 2013
4	A32592TYN6C9EM	0316967297	Amazon Customer	[0, 0]	Cute quilt, the colors are perfect and my litt...	4.0	Very cute	1378425600	09 6, 2013

Fig. 10. Data acquisition illustration 5

6 Symbols and Notations

The important symbols used in the coding are shown in Table 1. For the convenience, consumer and product are used in programming as user and item for easier notation respectively and same is commented in code as shown in fig 11.

Table 1: Parameter conventions used with their function

Symbols	Meaning
vector_size	vector size of review text and item description
epoch_num	number of epochs to train for paragraph vectors.
num_ui_link	Min. no. of reviews required for a reviewer to be filtered
num_iu_link	Min. no. of reviews required for a product
max_num_item	max number of item
max_num_user	max number of users
split_ratio	80-20 split of train and test data
dataname	Name of the product eg: baby, digital music

```
#Parameters
dataName =  "Baby"# name of your product
vector_size = 100 # vector size of reviewtext and item description
epoch_num = 100 # number of vector
reviews_data = [] # review list to insert review data inside
meta_data = [] # meta list to insert meta data inside
num_ui_link = 20  # the number of each user link to items
num_iu_link = 0  # the number of each item link to user

# user -> item  ->  for text
ui_dict = defaultdict(list)
#item -> user - >  for text
iu_dict = defaultdict(list)
#rating training data list
reviews_train_data = []
#rating testing data list
reviews_test_data = []
# max number of item
max_num_item = 1
# max number of users
max_num_user = 1
#item -> user -> for rating
iu_dict2 = defaultdict(list)
```

Fig 11: Example of symbols

2 Implementation framework Specs

The input data pre-processing algorithm is described in Algorithm 1. The algorithm is capable of being trained in the same manner as other keras-based deep learning networks. On top of the recommendation model, this research add a regression layer, whose output serves as our loss function. This permits us to accomplish our rating prediction target. Figure 4 depicts the training procedure using the Python keras package. The dense network of consumer sentiment analysis model and product content model is shown by Algorithm

2. The Rating Predictor is shown by Algorithm 3 in Fig. The Model Summary is depicted in the fig 13

Algorithm 1 Preprocessing of Amazon Review Data

Input: Reviewers data set
Output: Preprocessed data for data transformation

```
Preprocess()
Initialize hyperparameters:
                vector_size ← 100
                epoch_num ← 100
                num_ui_link ← 20
                num_iu_link ← 0
                max_num_item ← 1
                max_num_user ← 1
                split_ratio ← 4
                Batch_size ← 64
for each data_row, i ∈ review_data do
                user_id ← i["reviewerID"]
            item_id ← i["asin"]
            ui_dict[user_id] ← append(item_id)
             iu_dict[item_id] ← append(user_id)
                  If ((num_ui_link < 20) || (num_iu_link < 1))
                        delete i
end for
Covert review_data.json to review_data.txt
for each data_row, i ∈ meta_data do
                  item_id ← i["asin"]
                  if (iu_num != len(iu_dict[item_id]))
                        if (iu_num < num_iu_link)
                              delete i
```

```
25: end for
26: Covert meta_data.json to meta_data.txt
27: Gensim_doc2vec()
28: Use pretrained model from Gensim with parameters:
29:             min_count←1
30:             window←3
31:             vector_size←vector_size
32:             sample←10-3
33:             negative←5
34:             workers←4
35:Convert data.txt to vector format
36: Extract_rating()
37:for each data_row, i ∈ review_data do
38:         user_id ←i["reviewerID"]
39:         item_id ←i["asin"]
40:         if (user_id ! ∈ ui_dict)
41:                 ui_dict[user_id] ← user_id
42:                     if (user_id > max_num_user)
43:                             max_num_user ← user_id
44:         ui_dict[user_id] ← item_id
46:         if (item_id ! ∈ iu_dict2)
47:                 iu_dict[item_id] ← item_id
48:                     if (item_id > max_num_item)
49:                             max_num_item ← item_id
50:         iu_dict[item_id] ← user_id
51:
52: Test_Train_Split()
53:         num ←  0
54: for each data_row, i ∈ reviews_data do
55:         if (num % (split_ratio + 1) < split_ratio)
56:                 reviews_train_data.append(i)
57:                 num ← num + 1
58:         else
59:                 reviews_test_data.append(i)
60: num ← num+1
```

Algorithm 2 User Sentiment and Item Content Dense Network

Input: User_Data, User_Latent, Item_vector, item_latent

Output: Users_Sentiment, Item_Content

```
1: User_Sentiment(User_data, User_Latent)
2: Initialize hyperparameters:
3:                 input ← User_data
4:                 activation() ← relu()
5:                 Lamda() ← softmax()
6: User_Sentiment ← Multiply(User_data, Lamda(activation(User_Latent), input)
7: Item_Content(Item_vector, item_Latent)
8: Initialize hyperparameters:
9:                 input ← Item_vector
10:                activation ← relu
11:                Lamda ← softmax()
12:  Item_Content ← Multiply(Item_vector, Lamda(activation(item_latent), input)
```

Algorithm 3 Recommender Model

Input: User_Sentiment, Item_Content

Output: Prediction, Results

```
1: Recommender(User_Sentiment, Item_Content)
2: Initialize hyperparameters:
3:                  optimizer() ← adam
4:                  loss() ← MAE()
5:                  metrics() ← RMSE
6: model() ← Compile(optimizer(), loss(), metrics())
7: Prediction, Results ← Compute(model())
```

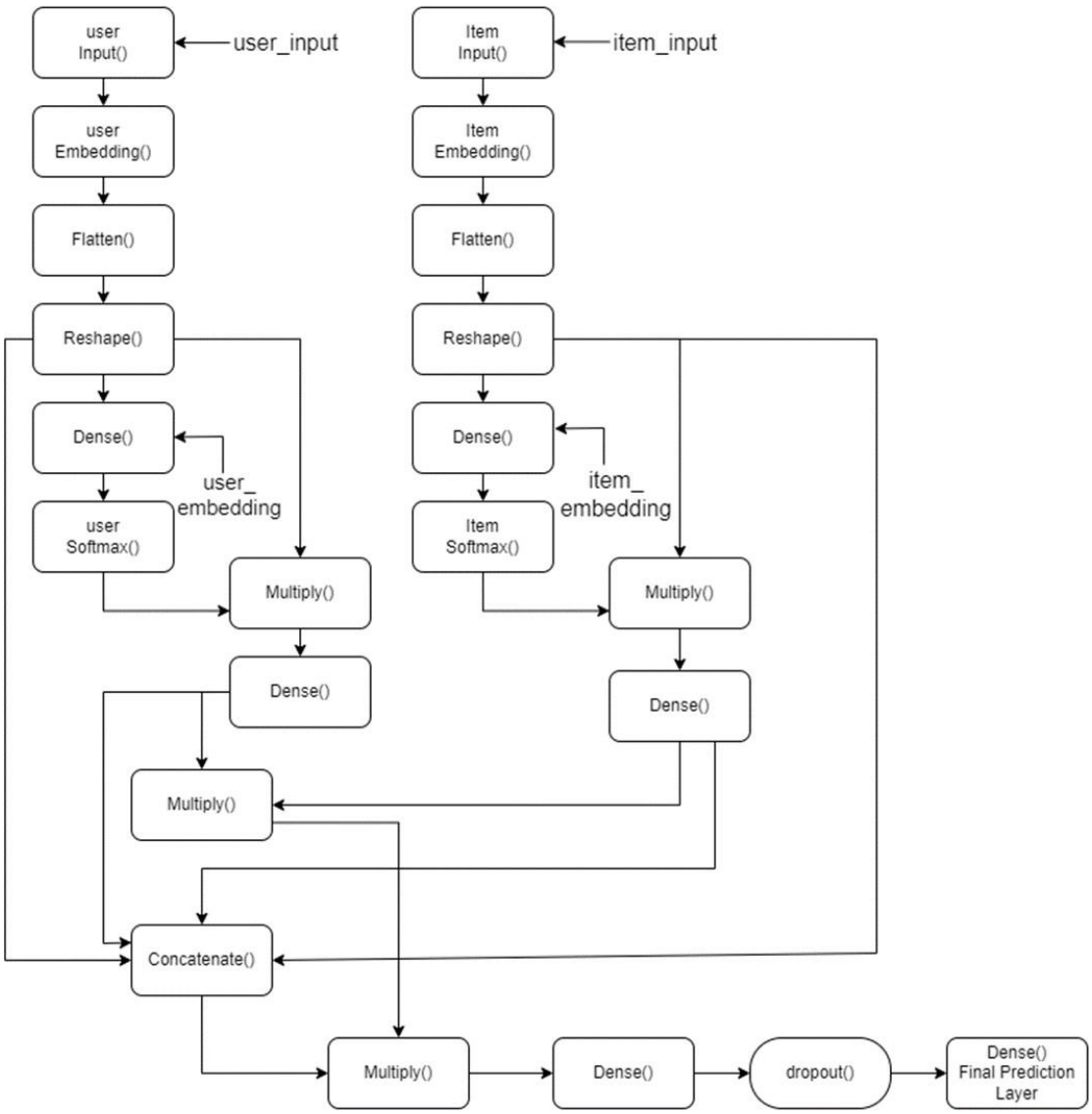


Fig. 12. Keras Training Flowchart

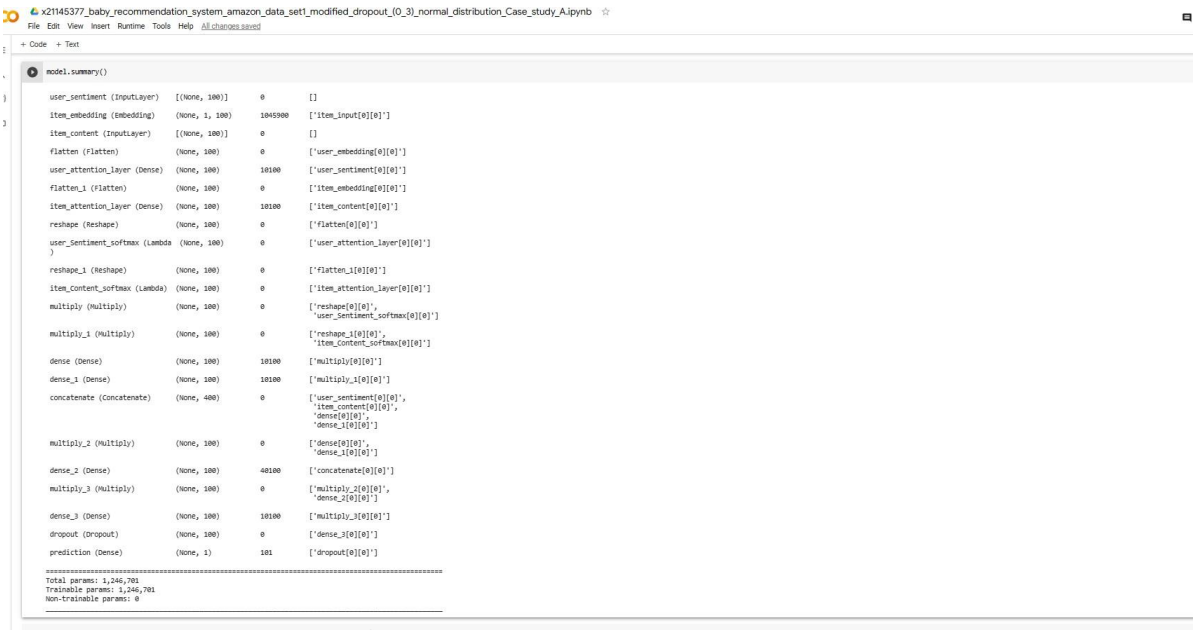


Fig. 13. Model Summary

3 Evaluation and Results

In all of our experiments and case studies, this research evaluated the performance of our proposed model by using the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). This is because the Mean Squared Error is the only metric that has been utilized for assessment in the majority of the relevant works [3].

Let's say that the total number of datapoints in the test set is denoted by the letter n. The MSE may be determined using the formulas below.

$$MAE = \frac{\sum_{k=0}^{n-1} |Y(k) - y(k)|}{n}$$

In a more similar manner, RMSE () is computed by:

$$RMSE = \sqrt{\frac{\sum_{k=0}^{n-1} (Y(k) - y(k))\ 2}{n}}$$

Where Y(k) and y(k) represent the actual and predicted value respectively

6.1 State of Art

State of Art-evaluation Metrics Capture are illustrated from the Fig14 to 15.

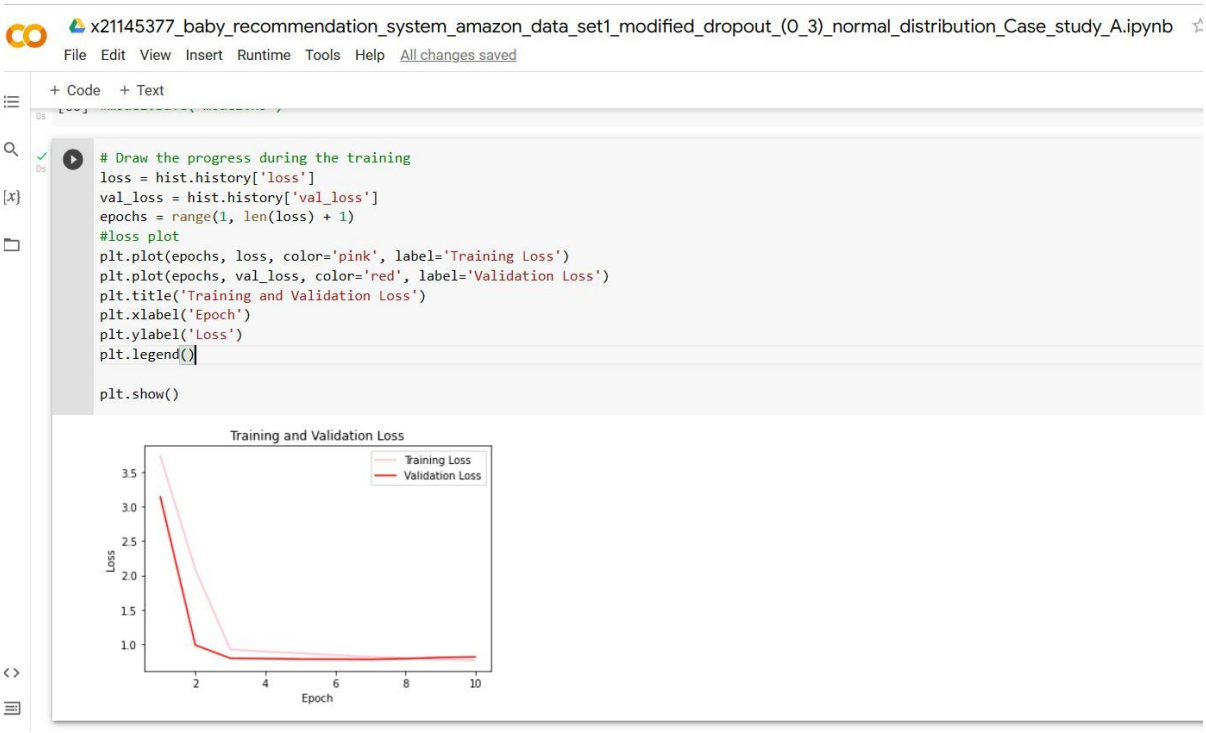


Fig. 14. Training and validation loss

Fig. 15. Baby evaluation accuracy Metrics Capture

6.2 *Replicating state of Art with different product category data set- :*

The accuracy evaluation Metrics Capture is illustrated from the Fig16

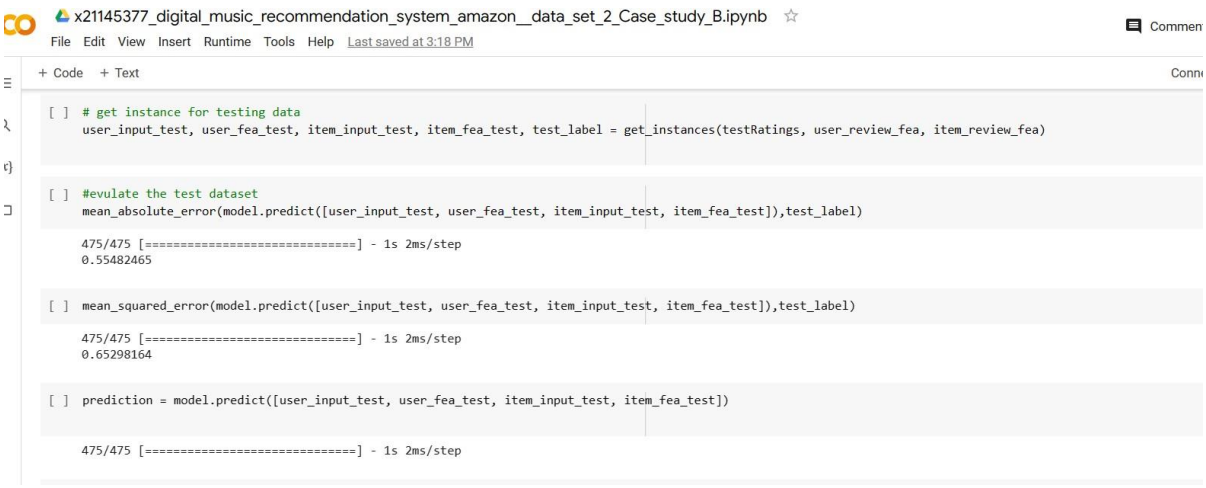


Fig. 16. Digital music dataset evaluation accuracy Metrics Capture

6.3 Sentiment analysis

Sentiment analysis accuracy evaluation Metrics Capture is illustrated from the Fig 17 to 19

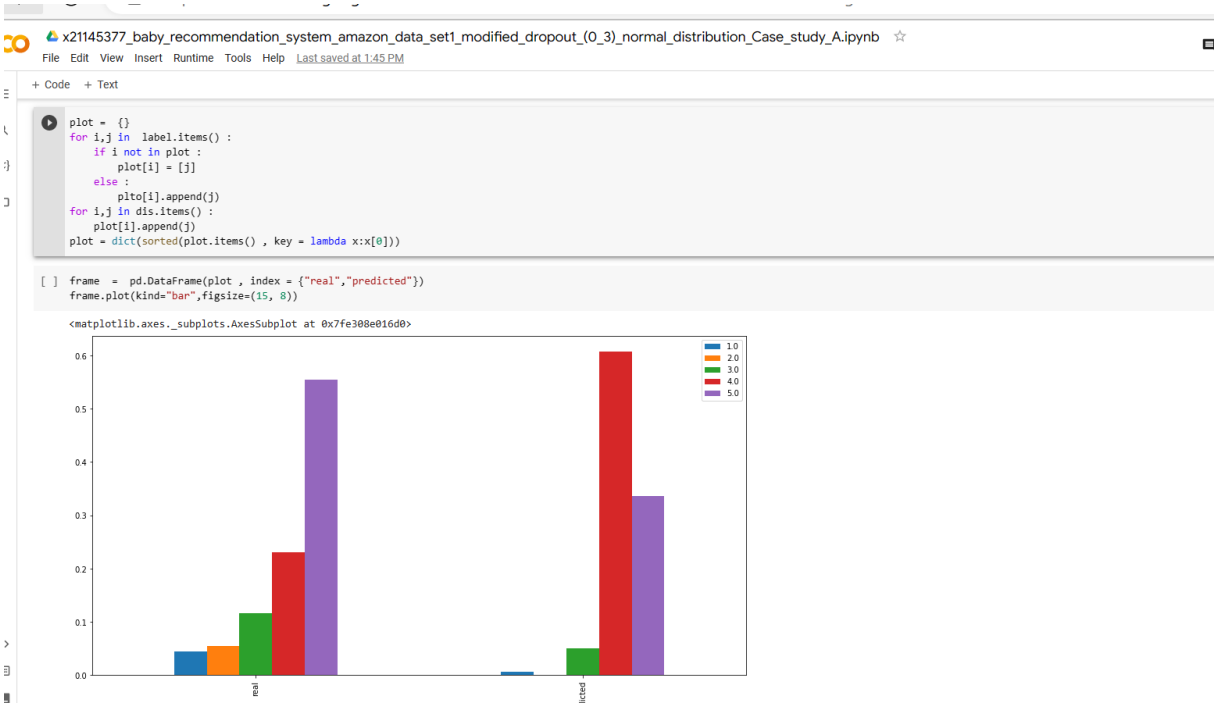


Fig17. Baby data set ratings distribution Metrics Capture

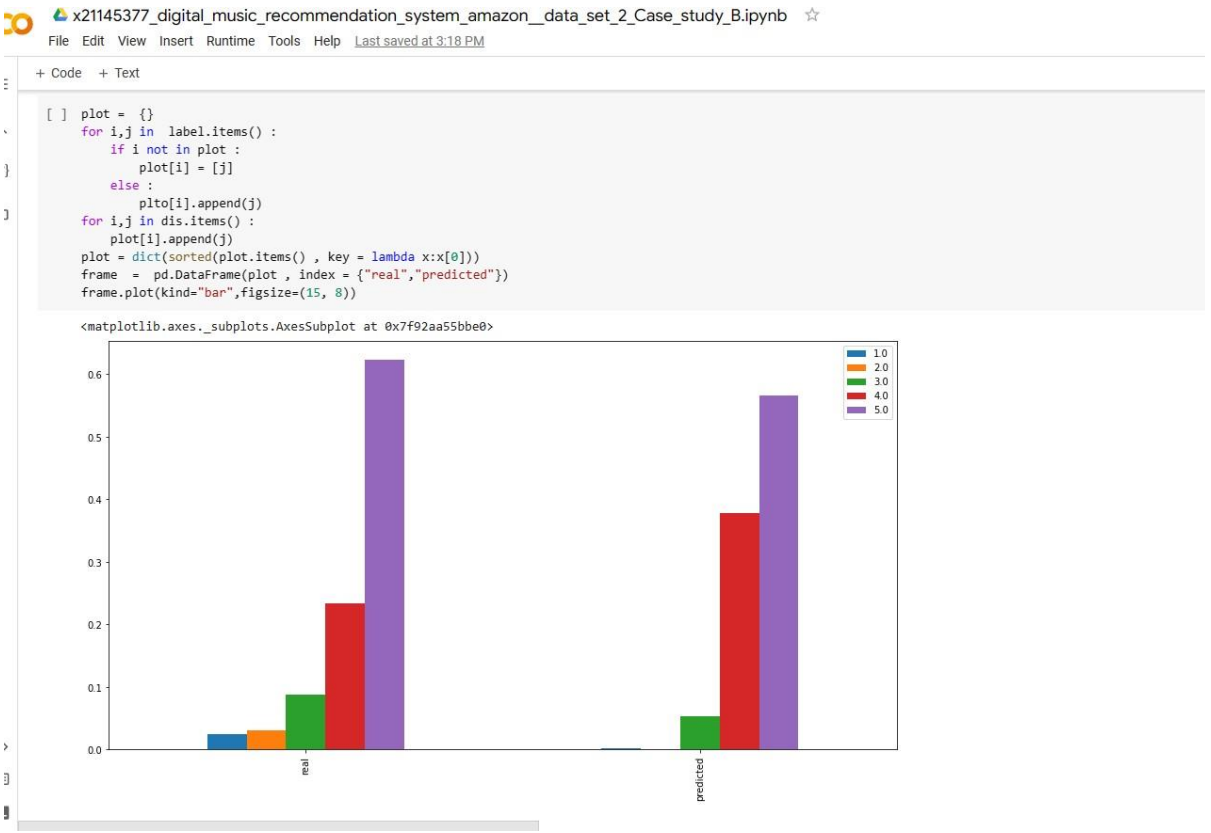


Fig. 18. Digital Music data set ratings distribution Metrics Capture

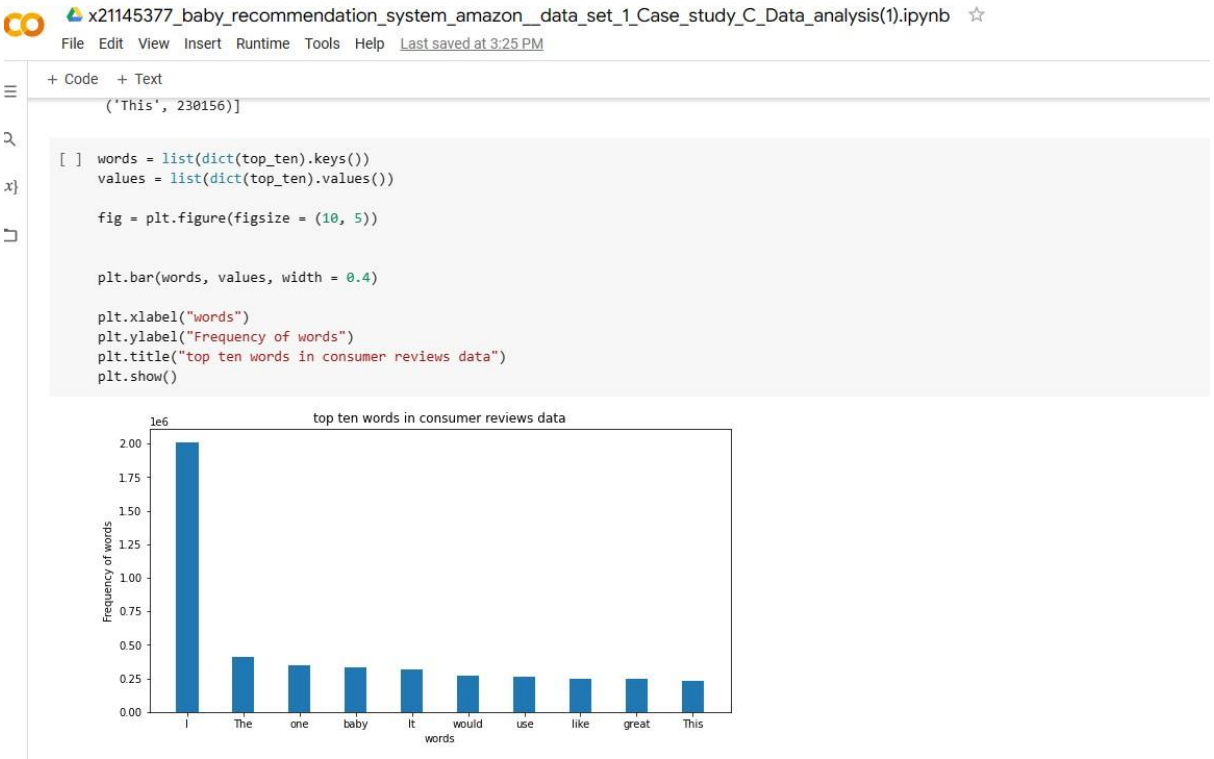


Fig. 19. Words distribution illustration

6.4 Cold Start Analysis

The Cold start accuracy Evaluation Metrics is Captured in Fig 19.

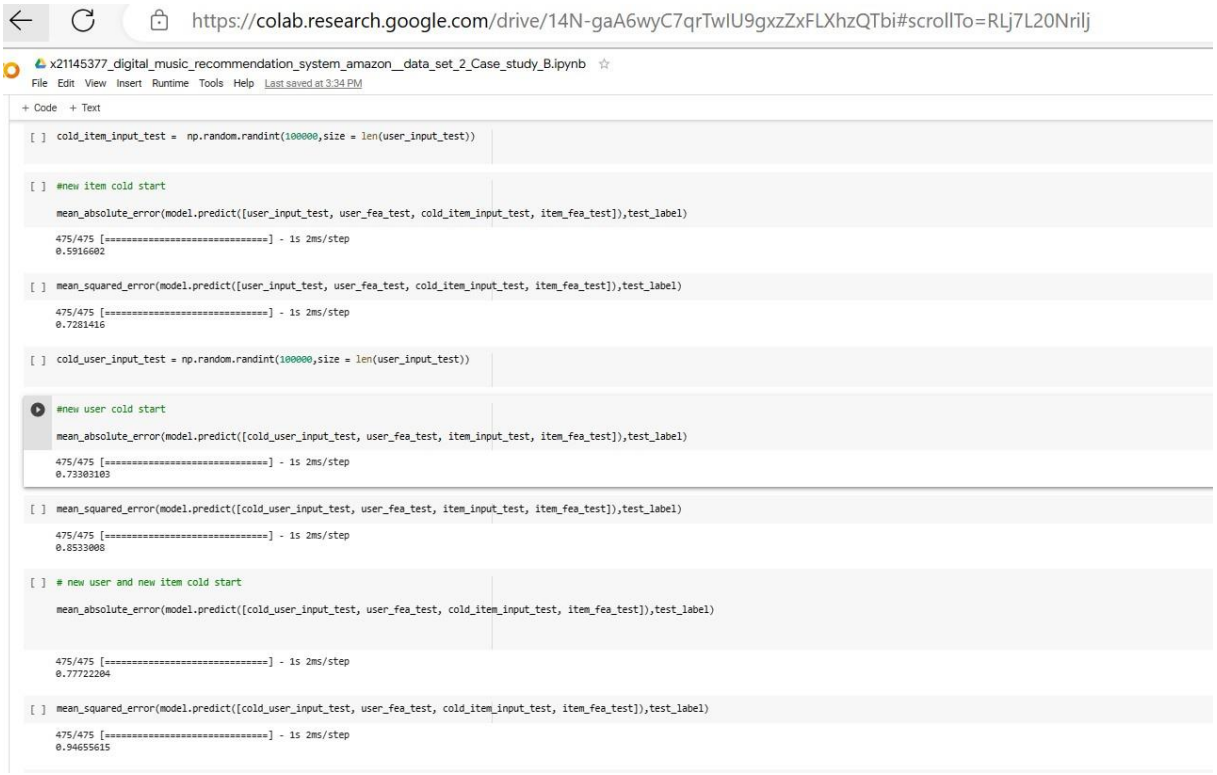


Fig. 20. Cold start Evaluation Metrics Capture

References

[1] Dataset1 (Baby Product Dataset) – Source: http://jmcauley.ucsd.edu/data/amazon/links.html,last accessed 2022/12/13.

[2] Dataset2 (Digital Music Product Dataset) – Source: http://jmcauley.ucsd.edu/data/amazon/links.html ,last accessed 2020/12/13.

[3] Dezfouli, P. A. B., Momtazi, S. and Dehghan, M. [2020]. Deep neural review text interaction for recommendation systems, Appl. Soft Comput. 100: 106985.