

Tutorial #6

Python- PRODUCT RECOMMENDER

Dr. Atef Bader

TA : Prakhar Nag

Product Recommendation and types:

- A product recommendation is basically a filtering system that seeks to predict and show the items that a user would like to purchase.

Types:

a) Content based Filtering Technique

It focuses on the properties of the items. Similarity of items is determined by measuring similarities in their properties.

b) Collaborative Filtering Technique

It focuses on the relationship between users and items. Similarity of items is determined by the similarity of the ratings of those items by the users who have rated both the items.

Ref : Mining of Massive Datasets - Lescovec, Rajaraman and Ullman

Collaborative Filtering :

There are further 2 types of Collaborative Filtering :

a) **Memory-based CF**

This approach can be divided into two main sections:

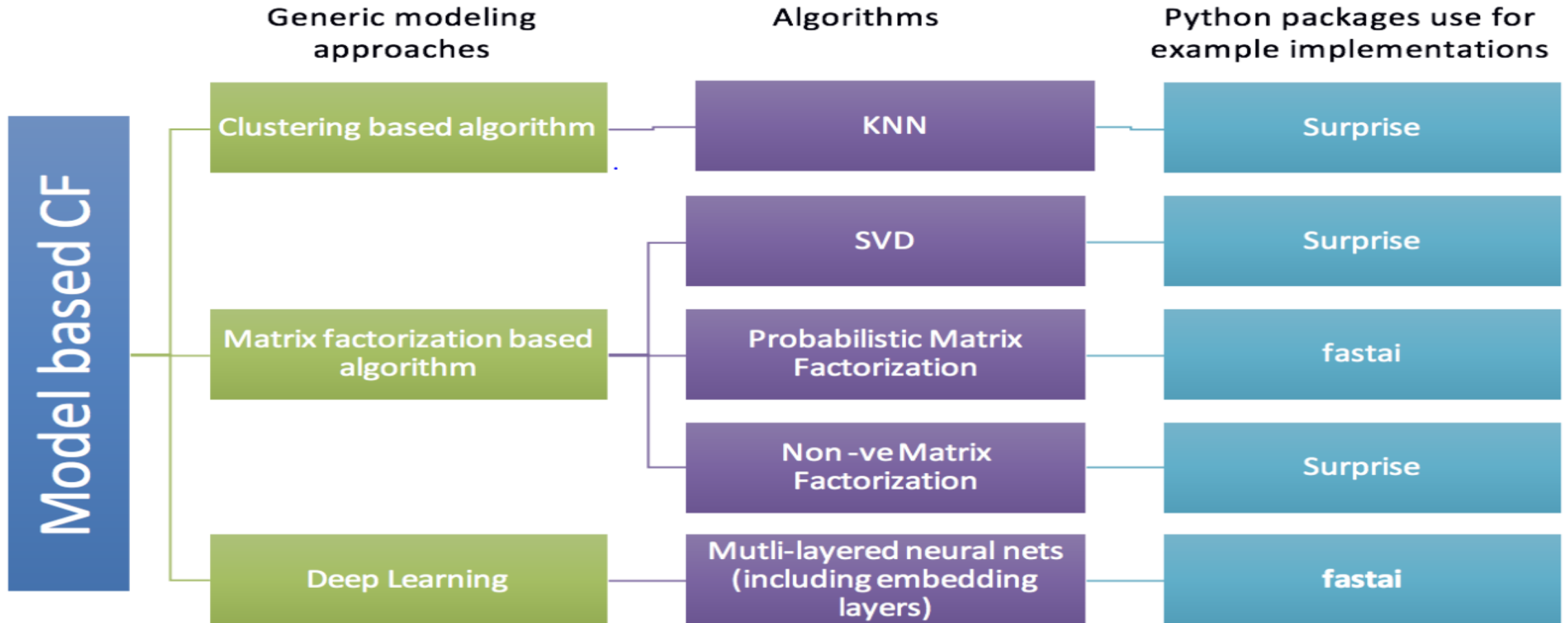
- A **user-item filtering** takes a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked.
- In contrast, **item-item filtering** will take an item, find users who liked that item, and find other items that those users or similar users liked. It takes items and, outputs other items as recommendations.

b) **Model-based CF**

In this approach, CF models are developed using machine learning algorithms to predict user's ratings of unrated items.

Ref: <https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0>

Figure below shows the further classification of Model based CF.



Ref: <https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0>

Our Approach to Implementing Recommender Systems :

- For this Tutorial implementation, we have used **Matrix Factorization technique** which will be using **Singular Value Decomposition (SVD)** algorithm that is already implemented in the **Surprise** package.

So, now let's learn more about Matrix Factorization technique.

Matrix Factorization : The Basics

- Consider below matrix of users as rows and items as columns.
- What matrix factorization does is to come up with two smaller matrices, one representing users and one representing items, which when multiplied together will roughly produce matrix of ratings, ignoring the 0 entries.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

A matrix of user/item ratings

Matrix Factorization : The Basics

- Let our matrix be $m \times n$, where m is the number of users and n is the number of items.
- So, we need $m \times d$ and a $d \times n$ matrix as our factors, where d is chosen to be small enough for the computation to be efficient and large enough to represent the number of dimensions along which interactions between users and items are likely to vary in some significant way.
- If we choose $d=2$, the predicted rating given by a user for a given item is then the dot product of the vector representing the user and the vector representing the item.
- We also need to add the bias for item i , user u and overall average rating for predicting the rating.

Matrix Factorization : The Basics

Theoretically, the predicted rating is given by below formula :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

Where, μ is the overall average rating, b_i is the bias for item i , b_u is the bias for user u , and $q_i^T p_u$ is the interaction between item i and user u .

Ref : <http://katbailey.github.io/post/matrix-factorization-with-tensorflow/>

Matrix-factorization : API docs

- The following class implements Matrix-factorization based prediction algorithms:

`class surprise.prediction_algorithms.matrix_factorization.SVD`

- The abstract class that defines the behavior of a prediction algorithm :

`class surprise.prediction_algorithms.algo_base.AlgoBase`

The famous SVD algorithm, as popularized by Simon Funk during the Netflix Prize(Competition for the best collaborative filtering algorithm) uses it.

Ref:

https://surprise.readthedocs.io/en/stable/matrix_factorization.html#surprise.prediction_algorithms.matrix_factorization.SVD

Surprise : Basic Note

- Surprise is a Python Scikit building and analyzing package for recommender systems.
- It alleviates the pain of Dataset handling. Users can use both built-in datasets (Movielens, Jester), and their own custom datasets.
- It provides ready-to-use prediction algorithms such as baseline algorithms, SVD, SVD++, neighborhood methods and many more.

Surprise – Installation

- Download Anaconda from the link → <https://www.anaconda.com/download>
- After downloading, Open “Anaconda Prompt”
- Type the command : ***conda install -c conda-forge scikit-surprise***
- Type ‘y’ when prompted, it should install the required packages inside the “Anaconda3” folder.

```
Proceed ([y]/n)? y

Downloading and Extracting Packages
ca-certificates-2018 | 170 KB      | ##### | 100%
openssl-1.0.2p       | 5.4 MB      | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\user>
```

- To know more about how conda install works :

<http://docs.anaconda.com/anaconda-cloud/user-guide/howto/#use-packages>

Jupyter notebook

- Type “jupyter notebook” in anaconda prompt and it will open the notebook and you can “upload” any .ipynb file to the notebook.

```
(base) C:\Users\user>jupyter notebook
[I 22:50:54.166 NotebookApp] [nb_conda_kernels] enabled, 3 kernels found
[I 22:51:07.866 NotebookApp] [nb_anacondacloud] enabled
[I 22:51:09.677 NotebookApp] \u2713 nbpresent HTML export ENABLED
[W 22:51:09.707 NotebookApp] \u2717 nbpresent PDF export DISABLED: No module named 'nbbrowserpdf'
[I 22:51:09.844 NotebookApp] [nb_conda] enabled
[I 22:51:10.557 NotebookApp] Serving notebooks from local directory: C:\Users\user
[I 22:51:10.557 NotebookApp] 0 active kernels
[I 22:51:10.572 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 22:51:10.576 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Jupyter notebook

- Following is the screenshot of the jupyter notebook server home screen where you can upload any **“.ipynb”** file and Select **“Cell”** from Menu and **“Run All”** to run the notebook file.



Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕ ↻

☐ 0 ▾

📁 /

Name ▾

Last Modified

File size

☐ 📁 [AndroidStudioProjects](#)

6 months ago

☐ 📁 [Applications](#)

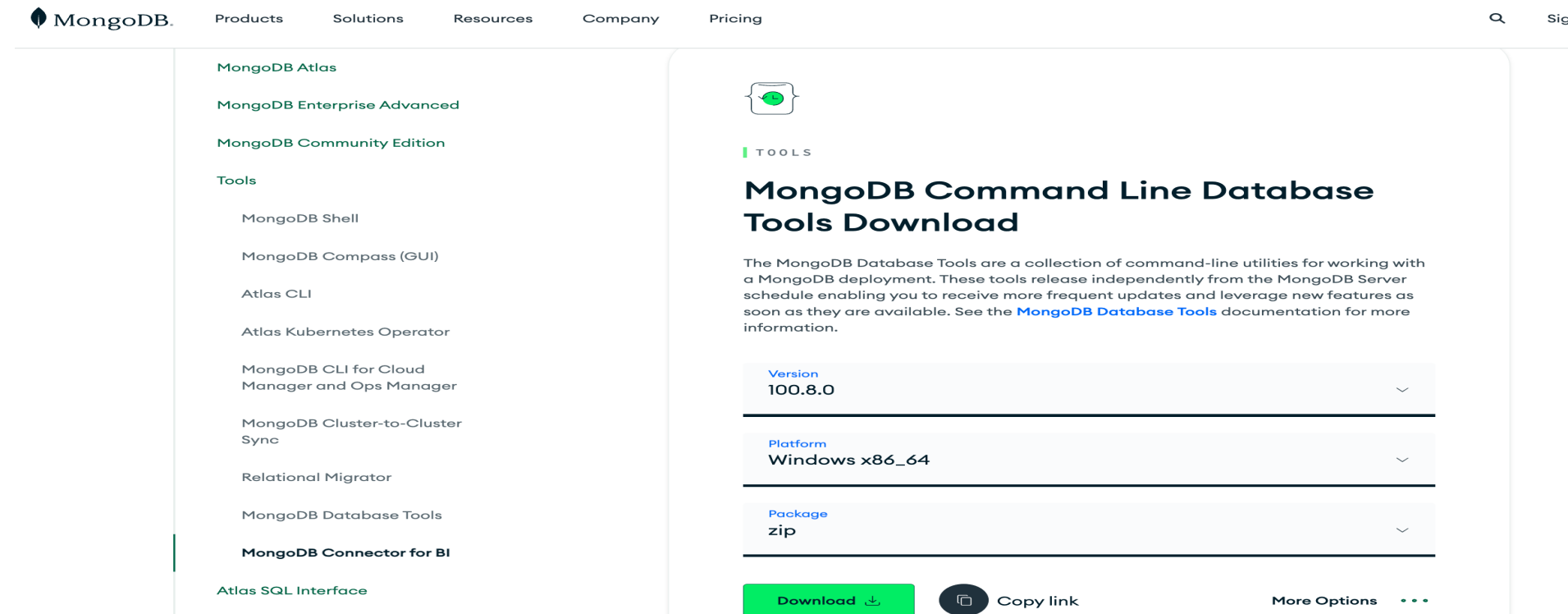
a month ago

Python code:

Reviews are stored inside MongoDB. So, we need to export the MongoDB data first using the command “mongoexport”.

Please note : If you are using the new version of MongoDB then it will not contain “mongoexport”. So, you will have to download it manually from this link →

<https://www.mongodb.com/try/download/database-tools>



The screenshot shows the MongoDB website's navigation bar with links for Products, Solutions, Resources, Company, and Pricing. The left sidebar lists various MongoDB products and tools, with 'MongoDB Database Tools' highlighted. The main content area is titled 'MongoDB Command Line Database Tools Download' and includes a description of the tools. Below the description are three dropdown menus for selecting the version (100.8.0), platform (Windows x86_64), and package (zip). At the bottom, there are buttons for 'Download', 'Copy link', and 'More Options'.

MongoDB. Products Solutions Resources Company Pricing

MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

Tools

- MongoDB Shell
- MongoDB Compass (GUI)
- Atlas CLI
- Atlas Kubernetes Operator
- MongoDB CLI for Cloud Manager and Ops Manager
- MongoDB Cluster-to-Cluster Sync
- Relational Migrator
- MongoDB Database Tools
- MongoDB Connector for BI**

Atlas SQL Interface

TOOLS

MongoDB Command Line Database Tools Download

The MongoDB Database Tools are a collection of command-line utilities for working with a MongoDB deployment. These tools release independently from the MongoDB Server schedule enabling you to receive more frequent updates and leverage new features as soon as they are available. See the [MongoDB Database Tools](#) documentation for more information.

Version
100.8.0

Platform
Windows x86_64

Package
zip

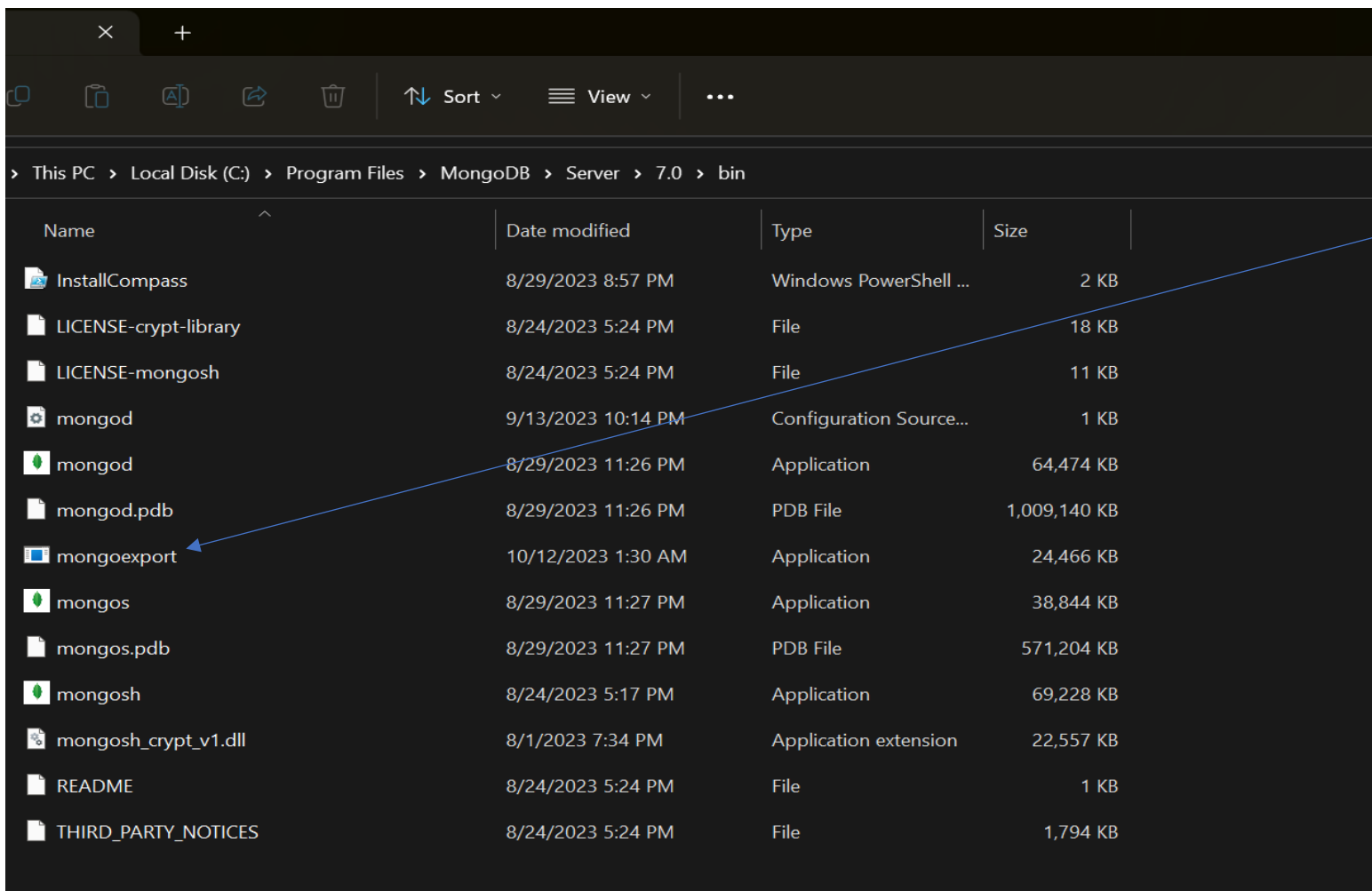
Download

Copy link

More Options

Python code:

Unzip the download folder and paste the “mongoexport” inside the MongoDB installation directory.



The screenshot shows a Windows File Explorer window with the address bar set to `This PC > Local Disk (C:) > Program Files > MongoDB > Server > 7.0 > bin`. The file list contains the following items:

Name	Date modified	Type	Size
InstallCompass	8/29/2023 8:57 PM	Windows PowerShell ...	2 KB
LICENSE-crypt-library	8/24/2023 5:24 PM	File	18 KB
LICENSE-mongosh	8/24/2023 5:24 PM	File	11 KB
mongod	9/13/2023 10:14 PM	Configuration Source...	1 KB
mongod	8/29/2023 11:26 PM	Application	64,474 KB
mongod.pdb	8/29/2023 11:26 PM	PDB File	1,009,140 KB
mongoexport	10/12/2023 1:30 AM	Application	24,466 KB
mongos	8/29/2023 11:27 PM	Application	38,844 KB
mongos.pdb	8/29/2023 11:27 PM	PDB File	571,204 KB
mongosh	8/24/2023 5:17 PM	Application	69,228 KB
mongosh_crypt_v1.dll	8/1/2023 7:34 PM	Application extension	22,557 KB
README	8/24/2023 5:24 PM	File	1 KB
THIRD_PARTY_NOTICES	8/24/2023 5:24 PM	File	1,794 KB

A blue arrow points from the **mongoexport** file in the list to a green box on the right containing the text **Mongoexport**.

Python code:

Link: <https://surprise.readthedocs.io/en/stable/FAQ.html>

Refer 'How to get the top-N recommendations for each user'

Code snippet which imports required packages :

```
In [7]: import os
import csv
from surprise import BaselineOnly
from surprise import Dataset
from surprise import Reader
from surprise import SVD
from surprise import accuracy
from surprise.model_selection import cross_validate
from surprise.model_selection import train_test_split
from collections import defaultdict
```

Python code:

```
pr_file_path="C:/apache-tomcat-9.0.52/webapps/Tutorial_6/"
os.chdir('C:/Program Files/MongoDB/Server/7.0/bin')
os.system(r'mongoexport --db CustomerReviews --collection myReviews --type=csv --fields userName,productName,reviewRating >'
          +pr_file_path+'mongodata_train.csv')
```

We read this file and generate a test set of it.

```
with open(pr_file_path+"/mongodata_train.csv", "r") as f:
    reader = csv.DictReader(f, delimiter=',')
    with open(pr_file_path+"/mongodata_test.csv", "w", newline='') as f_out:
        writer = csv.DictWriter(f_out, fieldnames=reader.fieldnames, delimiter=",")
        for row in reader:
            writer.writerow(row)
```

```
file_path = os.path.expanduser(pr_file_path+'mongodata_test.csv')
```

```
# As we're loading a custom dataset, we need to define a reader. In the
# movielens-100k dataset, each line has the following format:
# 'user item rating timestamp', separated by '\t' characters.
reader = Reader(line_format='user item rating', sep=',')

#data = Dataset.load_from_file(file_path, reader=reader)
```

Python code:

We then find top – N recommendation for each user and save them in an ‘matrixFactorizationBasedRecommendations.csv’ file.

```
def get_top_n(predictions, n=10):
    '''Return the top-N recommendation for each user from a set of predictions.

    Args:
        predictions(list of Prediction objects): The list of predictions, as
            returned by the test method of an algorithm.
        n(int): The number of recommendation to output for each user. Default
            is 10.

    Returns:
        A dict where keys are user (row) ids and values are lists of tuples:
            [(raw item id, rating estimation), ...] of size n.
        ...

    # First map the predictions to each user.
    top_n = defaultdict(list)
    for uid, iid, true_r, est, _ in predictions:
        top_n[uid].append((iid, est))

    # Then sort the predictions for each user and retrieve the k highest ones.
    for uid, user_ratings in top_n.items():
        user_ratings.sort(key=lambda x: x[1], reverse=True)
        top_n[uid] = user_ratings[:n]

    return top_n
```

Python code:

```
# First train an SVD algorithm on the movielens dataset.
data = Dataset.load_from_file(file_path, reader=reader)
trainset = data.build_full_trainset()
algo = SVD()
algo.fit(trainset)

# Than predict ratings for all pairs (u, i) that are NOT in the training set.
testset = trainset.build_anti_testset()
predictions = algo.test(testset)

top_n = get_top_n(predictions, n=2)

# Print the recommended items for each user
for uid, user_ratings in top_n.items():
    print(uid, [iid for (iid, _) in user_ratings])

out = open(pr_file_path+'/matrixFactorizationBasedRecommendations.csv', 'w', newline='')
output=csv.writer(out)

for uid, user_ratings in top_n.items():
    output.writerow([uid, [iid for (iid, _) in user_ratings]])

out.close()
```

```
sai ['xbox360']
john ['Surface4', 'ipad3']
customer ['Xbox', 'xbox360']
```

Python Code output :

Output is generated in 'matrixFactorizationBasedRecommendations.csv' file that consists of username and their top – n (n=2 in our case) recommendations based on the similar user ratings.

	A	B	C	D
1	sai	['xbox360']		
2	john	['Surface4', 'ipad3']		
3	customer	['Xbox', 'xbox360']		
4				
5				
6				
7				

Java Code:

ProductRecommenderUtility.java : method that reads output file

```
public HashMap<String,String> readOutputFile(){

    String TOMCAT_HOME = System.getProperty("catalina.home");
    BufferedReader br = null;
    String line = "";
    String cvsSplitBy = ",";
    HashMap<String,String> prodRecmMap = new HashMap<String,String>();
    try {

        br = new BufferedReader(new FileReader(new File(TOMCAT_HOME+"\\webapps\\Tutorial_6\\matrixFactorizationBasedRecommendations.csv")));
        while ((line = br.readLine()) != null) {

            // use comma as separator
            String[] prod_recm = line.split(cvsSplitBy,2);
            prodRecmMap.put(prod_recm[0],prod_recm[1]);

        }

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    return prodRecmMap;
}
```

Java Code:

ProductRecommenderUtility.java : method that retrieves product details from the database

```
public static Product getProduct(String product){
    Product prodObj = new Product();
    try
    {
        String msg = getConnection();
        String selectProd="select * from Productdetails where Id=?";
        PreparedStatement pst = conn.prepareStatement(selectProd);
        pst.setString(1,product);
        ResultSet rs = pst.executeQuery();

        while(rs.next())
        {
            prodObj = new Product(rs.getString("Id"),rs.getString("productName"),rs.getDouble("productPrice"),rs.getString("productImage"),rs.getString("productManufacturer"),
            rs.getString("productCondition"),rs.getString("ProductType"),rs.getDouble("productDiscount"));
        }
        rs.close();
        pst.close();
        conn.close();
    }
    catch(Exception e)
    {
    }
    return prodObj;
}
```

Java Code:

Display the products in the Carousel below Cart :

```
HashMap<String,String> prodRecmMap = new HashMap<String,String>();
prodRecmMap = prodRecUtility.readOutputFile();

int l =0;
for(String user: prodRecmMap.keySet())
{
    if(user.equals(utility.username()))
    {
        String products = prodRecmMap.get(user);
        products=products.replace("[", "");
        products=products.replace("]", "");
        products=products.replace("\\"", " ");
        ArrayList<String> productList = new ArrayList<String>(Arrays.asList(products.split(",")));

        myCarousel = "myCarousel"+l;

        sb.append("<div id='content'><div class='post'><h2 class='title meta'>");
        sb.append("<a style='font-size: 24px;'>"+""+" Recommended Products</a>");

        sb.append("</h2>");

        sb.append("<div class='container'>");
        /* Carousels require the use of an id (in this case id="myCarousel") for carousel controls to function properly.
        The .slide class adds a CSS transition and animation effect, which makes the items slide when showing a new item.
        Omit this class if you do not want this effect.
        The data-ride="carousel" attribute tells Bootstrap to begin animating the carousel immediately when the page loads.
        */
        sb.append("<div class='carousel slide' id='"+myCarousel+"' data-ride='carousel'>");
```

Java Code:

Display the products in the Carousel below Cart :

```
int k = 0;
for(String prod : productsList){
    prod= prod.replace("'", "");
    Product prodObj = new Product();
    prodObj = ProductRecommenderUtility.getProduct(prod.trim());
    if (k==0 )
    {
        sb.append("<div class='item active'><div class='col-md-6' style = 'background-color: #58acfa;border :1px solid #cfd1d3'>");
    }
    else
    {
        sb.append("<div class='item'><div class='col-md-6' style = 'background-color: #58acfa ;border :1px solid #cfd1d3' >");
    }
    sb.append("<div id='shop_item'>");
    sb.append("<h3>"+prodObj.getName()+"</h3>");
    sb.append("<strong>"+prodObj.getPrice()+"$</strong><ul>");
    sb.append("<li id='item'><img src='images/'"+prodObj.getType()+"/"+prodObj.getImage()+"' alt='' /></li>");
    sb.append("<li><form method='post' action='Cart'>"+
        "<input type='hidden' name='name' value='"+prod.trim()+"'>"+
        "<input type='hidden' name='type' value='"+prodObj.getType()+"'>"+
        "<input type='hidden' name='maker' value='"+prodObj.getRetailer()+"'>"+
        "<input type='hidden' name='access' value='"++" '+'>"+
        "<input type='submit' class='btnbuy' value='Buy Now'></form></li>");
    sb.append("<li><form method='post' action='WriteReview'>"+<input type='hidden' name='name' value='"+prodObj.getName()+"'>"+
        "<input type='hidden' name='type' value='"+prodObj.getType()+"'>"+
        "<input type='hidden' name='maker' value='"+prodObj.getRetailer()+"'>"+
        "<input type='hidden' name='access' value='"++" '+'>"+
        "<input type='hidden' name='price' value='"+prodObj.getPrice()+"'>"+
        "<input type='submit' value='WriteReview' class='btnreview'></form></li>");
    sb.append("<li><form method='post' action='ViewReview'>"+<input type='hidden' name='name' value='"+prodObj.getName()+"'>"+
        "<input type='hidden' name='type' value='"+prodObj.getType()+"'>"+
        "<input type='hidden' name='maker' value='"+prodObj.getRetailer()+"'>"+
        "<input type='hidden' name='access' value='"++" '+'>"+
        "<input type='submit' value='ViewReview' class='btnreview'></form></li>");

    sb.append("</ul></div></div>");
    sb.append("</div>");

    k++;
}
```


Output:

If I login as a user and add a product to the Cart, it shows me Recommended products in the carousel which can be added to the Cart or can be Reviewed as well.

The screenshot displays the 'Game Speed' website interface. The top navigation bar includes links for Home, Consoles, Games, Tablets, and Trending. A secondary navigation bar contains links for Addproduct, Updateproduct, Deleteproduct, Trending, DataAnalytics, Hello.Prakhar, Logout, and Cart(1). The left sidebar features four category lists: Consoles (Microsoft, Sony, Nintendo), Games (Electronic Arts, Activision, Take-Two Interactive), Tablets (Apple, Microsoft, Samsung), and Accessories (Microsoft Accessories, Sony Accessories, Nintendo Accessories). The main content area shows the 'Cart(1)' summary with a table listing items and a total, followed by a 'Recommended Products' carousel featuring Surface Pro 3 and PS4.

Game Speed

Search Product:
search here..

Home Consoles Games Tablets Trending

Addproduct Updateproduct Deleteproduct Trending DataAnalytics Hello.Prakhar Logout Cart(1)

Consoles

- Microsoft
- Sony
- Nintendo

Games

- Electronic Arts
- Activision
- Take-Two Interactive

Tablets

- Apple
- Microsoft
- Samsung

Accessories

- Microsoft Accessories
- Sony Accessories
- Nintendo Accessories

Cart(1)

1.	Samsung Tab 2	: 150.99
Total		150.99
		CheckOut

Recommended Products

Surface
299.99\$

Buy Now
WriteReview
ViewReview

PS4
349.99\$

Buy Now
WriteReview
ViewReview

Manually Checking - sample example:

Let say, users have given following reviews for the products :

	sai	john	customer
Samsung2	5		
Surface4	5		4
PS4	4	5	5
g1	1	5	5
WII	3		
Xbox	5	5	
xbox360		5	
ipad3			5

Correctness of the Example looking manually:

- Looking at the data clearly, we can say that user “customer” seems more likely to match with “john” and hence “customer” will be recommended ‘Xbox’ and ‘Xbox360’ which john has rated.

	sai	john	customer
Samsung2	5		
Surface4	5		4
PS4	4	5	5
g1	1	5	5
WII	3		
Xbox	5	5	
xbox360		5	
ipad3			5

- As a concluding remark, a machine learning algorithm is expected to have accuracy of more than 50% and not always 100%.

Questions?