



## **PROJECT REPORT**

**On**

### **Recommendation Systems Using Python and Data Science**

**Submitted in fulfilment for the  
Internship for B.Tech II Year**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Under the esteemed guidance of**

**Mrs. Paritala. Jhansi Rani**

**Ms. S. Harika Lakshmi**

**Submitted by**

Ms. P. DEEPTHI SREE	190031289
Ms. P. SAI ANISHA	190032023
Mr. C. YASHWANTH	190030311
Mr. M. SRIRAM	190030979



### DECLARATION

We hereby declare that this project-based lab report entitled **“Recommendation Systems Using Python and Data Science”** has been prepared by us from **COMPUTER SCIENCE AND ENGINEERING** during the Even Semester of the academic year 2020- 2021. We also declare that this project-based lab report is of our own effort, and it has not been shared with any others

**Mrs. Paritala Jhansi Rani**

**(Project Guide)**

**Ms. S. Harika Lakshmi**  
**(Project Guide)**

**Mr. V. Hari Kiran**  
**(Head of the Department)**

## ACKNOWLEDGEMENT

Our sincere thanks to our project guides **Mrs. P. Jhansi Rani & Ms. S. Harika Lakshmi** in the Lab for their outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Dr. Anjali Mathur** for coordinating best internship to prove our talent related to our courses in the Computer Science and Engineering Department for providing us with adequate planning and support and means by which we can complete this project-based Lab.

We express our gratitude to **Mr. V. Hari Kiran**, Head of the Department for computer science and Engineering for providing us with adequate facilities, ways and means by which we can complete this project-based Lab.

We would like to place on record the deep sense of gratitude to the honorable Vice Chancellor, Koneru Lakshmaiah University for providing the necessary facilities to carry the project-based Lab.

Last but not the least, we would also like to thank all Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our project-based Lab.



## Students' Declaration

We hereby declare that this Project entitled ON “**Recommendation Systems Using Python and Data Science**” is a record of bonafide work by our team submitted in partial fulfilment for the award of Internship for second year-B. Tech in Computer Science and Engineering in Koneru Lakshmaiah Education Foundation (Deemed to be University). The results embodied in this report have not been copied from any other means.

### Team Member 1

Name: **Ms. P. DEEPTHI SREE**

Register number: 190031289

Signature

### Team Member 2

Name: **Ms. P. SAI ANISHA**

Register number: 190032023

Signature

### Team Member 3

Name: **Mr. C. YASHWANTH**

Register number: 190030311

Signature

### Team Member 4

Name: **Mr. M. SRIRAM**

Register number: 190030979

Signature

## Steps to create VPC

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, click VPC Dashboard. If you do not already have any VPC resources, locate the Your Virtual Private Cloud area of the dashboard and click Get started creating a VPC. Otherwise, click Start VPC Wizard.

Select the second option, VPC with a Single Public Subnet, and then click Select.

Enter the following information into the wizard and click Create VPC

IP CIDR block

10.0.0.0/16

VPC name

ADS VPC

Public subnet

10.0.0.0/24

Availability Zone

No Preference

Subnet name

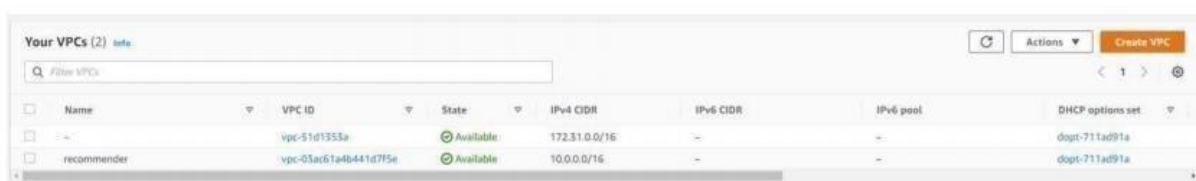
ADS Subnet 1

Enable DNS hostnames

Leave default selection

Hardware tenancy

default



The screenshot shows the 'Your VPCs (2)' section of the Amazon VPC console. It features a search bar and a table with columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, IPv6 pool, and DHCP options set. Two VPCs are listed: 'recommender' and 'vpc-51d1353a', both in an 'Available' state.

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	IPv6 pool	DHCP options set
<input type="checkbox"/>	recommender	vpc-03ac61a4b441d7f5e	Available	10.0.0.0/16	-	-	dhcp-711ad91a
<input type="checkbox"/>	-	vpc-51d1353a	Available	172.31.0.0/16	-	-	dhcp-711ad91a

## Launch subnet

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, select Subnets, select the subnet with the name ADS Subnet 1, and select the Summary tab at the bottom of the page. Make a note of the Availability Zone of this subnet.

Click Create Subnet and enter the following information in the Create Subnet dialog box and click Yes, Create.

Name	VPC	Availability Zone	IPv4 CIDR block
public	vpc	select az	10.0.1.0/24
private	vpc	select az	10.0.2.0/24

\*\*\*\*Note\*\*\*\*

IPv4 CIDR block should be unique

All subnets default enter in default route table

## Create a custom route table

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose Route Tables.

Choose Create route table.

Name	VPC
PublicRT	Demovpc
PrivateRT	Demovpc

(Optional) Add or remove a tag.

[Add a tag] Choose Add tag and do the following:

For Key, enter the key name.

For Value, enter the key value.

[Remove a tag] Choose the Delete button ("X") to the right of the tag's Key and Value.

Choose Create

## Create and attach an internet gateway

After you create an internet gateway, attach it to your VPC.

To create an internet gateway and attach it to your VPC

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose Internet Gateways, and then choose Create internet gateway.

Name

demovpcig

Optionally add or remove a tag.

[Add a tag] Choose Add tag and do the following:

For Key, enter the key name.

For Value, enter the key value.

[Remove a tag] Choose Remove to the right of the tag's Key and Value.

Choose Create internet gateway.

Select the internet gateway that you just created, and then choose Actions, Attach to VPC.

Select <your VPC> from the list, and then choose Attach internet gateway.



## Determine the route table for a subnet

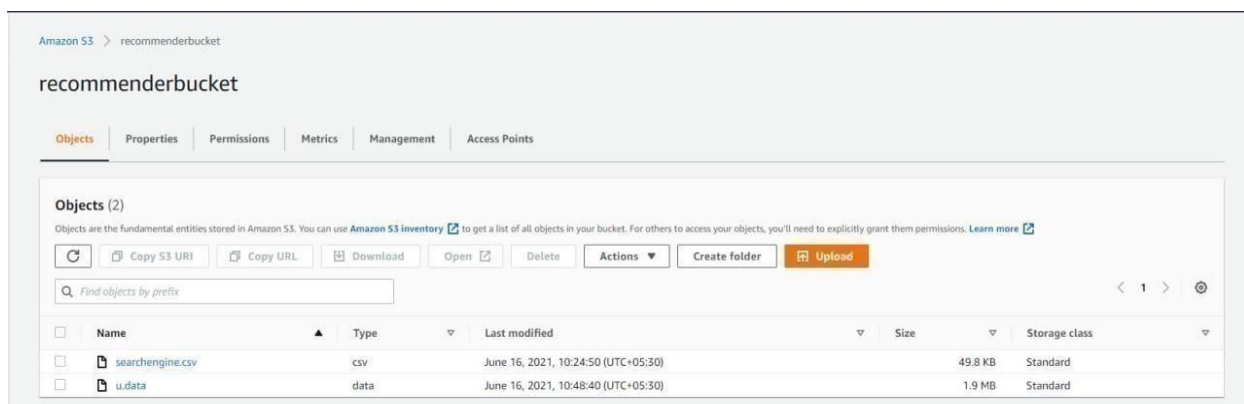
Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose Route Tables.

choose Action.

Select Edit Subnet Association

choose the subnet from the listed subnets and then click save



## Steps to associate Internet gateway to Route Tables

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose Route Tables.

choose PublicRT

Select Edit Routes

Provide the information as below:

Destination	Target
0.0.0.0/0	Select Internet Gateway (<your ig>)

click save

## Launch Instance in Subnet

Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

In the navigation pane, choose Launch Instance.

Select Amazon Linux AMI 2018.03.0

Click next

Provide Details as below:

Network	Subnet	Auto-assign Public IP
Demovpc	Public	Enable

Click Next

Keep storage as default

Select Tag

Choose name as PublicLinux

Click Launch

\*\*\*\*Note\*\*\*\*

In private subnet 'auto-assign public IP' should be disable

Filter by tags and attributes or search by keyword											
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring
publicec2	i-000db8d95bc5a85d3	t2.micro	ap-south-1a	running	2/2 checks ...	None		13.235.242.116	-	keypair	disabled
privateec2	i-012187687909478...	t2.micro	ap-south-1a	running	2/2 checks ...	None		-	-	keypair	disabled

## Steps to connect with Instance

Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

In the navigation pane, choose Launch Instance.

Select required ec2 instance

click connect

choose window ---> connect to PuttyGen

Enter command in terminal with right .pem file access

\*\*\*\*Note\*\*\*\*

make .pem file readable using command `chmod 400 <pemfile name>`

```
sai anisha@LAPTOP-UH18NAE7 MINGW64 /d/sem5/aws
$ ssh -i "keypair.pem" ec2-user@13.235.242.116
Last login: Tue Jun 15 05:13:04 2021 from 103.206.115.73

  _ | _ | _ )
  _ | ( _ /
  _ |\ _ | _ |
                Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-230 ~]$ aws s3 ls
2021-06-14 18:42:50 recommenderbucket
[ec2-user@ip-10-0-1-230 ~]$
```

## Steps to connect Public instance with Private instance

Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Select required ec2 instance

we need .pem file of Private instance

We will copy .pem file and create one new file in public instance to access private instance

First connect with public instance

`sudo su -`

`vi new.pem`

press esc

enter `:wq!`

provide permission : `ssh -i 'new.pem' ec2-user@<private instance>`

```
[ec2-user@ip-10-0-1-230 ~]$ sudo su -  
Last login: Tue Jun 15 05:13:15 UTC 2021 on pts/1  
[root@ip-10-0-1-230 ~]# ssh -i "keypair.pem" ec2-user@10.0.2.17  
Warning: Identity file keypair.pem not accessible: No such file or directory.  
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).  
[root@ip-10-0-1-230 ~]#
```

 root@ip-10-0-1-230:~-----BEGIN RSA PRIVATE KEY-----  
MIIEpAIBAAKCAQEAyAIIkHDAgS9tvfBidi73gqg/y3/z1YrZYio3Ik0QF8yVvps1  
TRAvhpvokL38IO0RzfmwQLQaHb34UCxgmCqGCDEB503LHsthZqkrmIU4TOcQRiNe  
rkLcneabadHWz1HDo1vkSb3EjGuciZM5fBBQ9I1zXuKX7LgS7MGmYnUr1NT029GO  
ho9GOGEOG9aPuij3CBX4ORGOKbTEepY59k1Sxb4KeyhUw/K55HZcfYzifkk017r  
Cu1zi+KB90eULmBgN8ozWwNPXr31a20gGjIxsp8QXR+mYs0L9HWLZN6cPA5QMVsi  
LoECEbN47PSJJiWcGp+cOeod894cBqwJG9tWjwIDAQABaoIBAF1EXXWfTB6keFdn  
XnLgKf1avg0GiM0ep4pYR0eOih+e/1Dw11LhYi7YS1Yb0hQEmRXk+oUht1oBOWch  
mpw61q/BMrHv1ZWxn6UA5FyGwx+5bGRYOBFBcQhZzKuu963L4h+3Hdiq7pnNmMYq  
y1apQScXZZ5sqg3wNJsFHRkXExR2ukjD1WmMy1D1jZMIxCuUDanZxLSXz5CZvo3C  
LVJvwkGnd1Jmpil+GH/jK05S2PIwinvckEriJehSeoLUKQiv8uS8Lf9YxBX85uL5  
qBiOAgMK5cojGhST5V90vyzK8Dfbs2WLBgZsLqfZCTmz62CVK3pawQuoTVCg7eUg  
Xs8SB8pECgYEA9dUJBtne7xkUqqS+d6sFFs3PqbvnFgK5sHYpBAEio72CGGmc2Kst  
JMoig9gph+mSB6wN8Sw2is1eK1Uisip11SokK6TXffSvqFeOK6aabHGIXSVqrNVJt  
mVCDfYeyayCuSbndngoPNuhxsYQ0H4iLBHoEPmFNspHvI2XgUi axJkCgYEA0Efl  
N6afw0+IcH9tEuNCyu0J7UyviB+IA1ssud1NbbgUdwdn1Uh0mqxo3jc0uvFBjvZ1  
Jjhtxnz4u+nUQYifhqnKarr/cPT7Qbpqwy7aRTaTFZXCFX6pf2YVJScUFPrSv  
AT57pFDRh6R0c/tmG/pbfTm7VXnEW0i+k/oBRwcCgYAGfUMpvSKXaTAUerjWC9us  
ig5M6rrk59mDAPuAMY93yD7g9ogUB/TSHcvids+zTmfjnkX/rTJ4TX1KuAp191Yc  
QJqbIWLu6PPa9BC8rjs4uWYn947Bb5WZEhKFaUjtbwVqyM9IjU9ym9Gfb6cN+eZp  
B2f62rtbgod34Lz5GrOLMQKBgQCR+x/2G3Tp3062WrmqWVsBNUYA6pTFBCshBZBt  
nsyd0D2wpclklc9yHZFK3sq41Evs1dT60ubGdd5jyyCIvaetXE3tCeyf4rv6kzQm  
N80RSx7AmtI38Zn0mUehAAiECfFJJ/7ZEmA6pOKW/a8Zc4T36xugw4wHttbnXC8n  
LD0j1QKBgQCA3xprds7+Iwvcl6ETI7qxc8UdclKoq+aiOZ9CDSTndg0SiJ9p6Tvy  
batU1/CVVmWZ3xy23VdYLqIPXawf77YgVFjcyU7qs9t0dIct17CV9jBIYKDqb2xI  
gzhNKX+idTFxBU3Cv+xN8xmr03B3ilKrvs2hDE70ZouUh/kCyoMnPg==

```
-----END RSA PRIVATE KEY-----
```

```
"new.pem" [readonly] 27L, 1679C
```

```
[root@ip-10-0-1-230 ~]# vi new.pem
[root@ip-10-0-1-230 ~]# ssh -i "new.pem" ec2-user@10.0.2.17
Last login: Tue Jun 15 05:14:22 2021 from 10.0.1.230
```

```

_ | _ | _ )
_| ( _ /  Amazon Linux 2 AMI
_ | \ _ | _ |

```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-17 ~]$
```

## Attach To Route Table

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose Route Tables.

Select Private Route Table

Choose Actions

Select Actions

Choose Edit Routes

Provide the following information:

Destination	Target
0.0.0.0/0	NAT

## To create a NAT gateway

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose NAT Gateways.

Choose NAT Gateway and provide the following information:

Subnet	ElasticIP
Public subnet	Generated EIPAttach To Route Table

Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

In the navigation pane, choose Route Tables.

Select Private Route Table

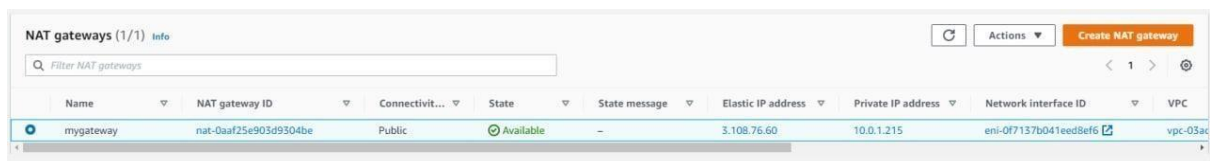
Choose Actions

Select Actions

Choose Edit Routes

Provide the following information:

Destination	Target
0.0.0.0/0	NA



Name	NAT gateway ID	Connectivity	State	State message	Elastic IP address	Private IP address	Network interface ID	VPC
mygateway	nat-0aaf25e903d9304be	Public	Available	–	3.108.76.60	10.0.1.215	eni-0f7137b041eed8ef6	vpc-03ac



## Testing of NAT

Connect private machine from public subnet

Try the command ping google.com

```
[ec2-user@ip-10-0-1-230 ~]$ ping google.com
PING google.com (142.250.67.206) 56(84) bytes of data:
64 bytes from bom12s08-in-f14.1e100.net (142.250.67.206): icmp_seq=1 ttl=111 time=2.03 ms
64 bytes from bom12s08-in-f14.1e100.net (142.250.67.206): icmp_seq=2 ttl=111 time=2.13 ms
64 bytes from bom12s08-in-f14.1e100.net (142.250.67.206): icmp_seq=3 ttl=111 time=2.14 ms
64 bytes from bom12s08-in-f14.1e100.net (142.250.67.206): icmp_seq=4 ttl=111 time=2.22 ms
64 bytes from bom12s08-in-f14.1e100.net (142.250.67.206): icmp_seq=5 ttl=111 time=3.24 ms
64 bytes from bom12s08-in-f14.1e100.net (142.250.67.206): icmp_seq=6 ttl=111 time=2.08 ms
```

## Running Jupyter Notebook on an EC2 Server

To run your Jupyter Notebook on your EC2 server, you are going to need to add a new security group. On the EC2 instance page go to the “Security Groups” section

click “Create Security Group” on the top of the page.

Use “Add Rule” to add new Security Group rules (SSH rule, HTTPS rule, Custom TCP rule)

Click Create

Go to instances tab

Select the private EC2 instance

Go to Actions, Networking

Click Change Security Groups

In this menu, select the security group that you just created and then click “Assign Security Group”

Connecting to your EC2 : `sudo chmod 400 /path/new.pem`

# Installing Jupyter Notebook

Run the command:

```
wget https://repo.anaconda.com/archive/Anaconda3-2019.03-Linux-x86_64.sh
```

```
bash Anaconda3-2019.03-Linux-x86_64.sh
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-230 ~]$ wget https://repo.continuum.io/archive/Anaconda2-4.1.1-Linux-x86_64.sh
--2021-06-16 07:21:17-- https://repo.continuum.io/archive/Anaconda2-4.1.1-Linux-x86_64.sh
Resolving repo.continuum.io (repo.continuum.io)... 104.18.200.79, 104.18.201.79, 2606:4700::6812:c94f, ...
Connecting to repo.continuum.io (repo.continuum.io)|104.18.200.79|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://repo.anaconda.com/archive/Anaconda2-4.1.1-Linux-x86_64.sh [following]
--2021-06-16 07:21:17-- https://repo.anaconda.com/archive/Anaconda2-4.1.1-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.130.3, 104.16.131.3, 2606:4700::6810:8303, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.130.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 419038579 (400M) [application/x-sh]
Saving to: 'Anaconda2-4.1.1-Linux-x86_64.sh'

100%[=====>] 419,038,579 9.37MB/s in 44s

2021-06-16 07:22:01 (9.17 MB/s) - 'Anaconda2-4.1.1-Linux-x86_64.sh' saved [419038579/419038579]

[ec2-user@ip-10-0-1-230 ~]$ bash what_Anaconda_you_downloaded_Linux_x86_64.sh
bash: what_Anaconda_you_downloaded_Linux_x86_64.sh: No such file or directory
[ec2-user@ip-10-0-1-230 ~]$ bash https://repo.continuum.io/archive/Anaconda2-4.1.1-Linux-x86_64.sh
bash: https://repo.continuum.io/archive/Anaconda2-4.1.1-Linux-x86_64.sh: No such file or directory
[ec2-user@ip-10-0-1-230 ~]$ ^C
[ec2-user@ip-10-0-1-230 ~]$ bash 'Anaconda2-4.1.1-Linux-x86_64.sh'
bash: 'Anaconda2-4.1.1-Linux-x86_64.sh: No such file or directory
[ec2-user@ip-10-0-1-230 ~]$ bash Anaconda2-4.1.1-Linux-x86_64.sh

welcome to Anaconda2 4.1.1 (by Continuum Analytics, Inc.)

In order to continue the installation process, please review the license agreement.
Please, press ENTER to continue
>>>
=====
Anaconda License
```

## Installing Jupyter Notebook

Run the command:

```
wget https://repo.anaconda.com/archive/Anaconda3-2019.03-Linux-x86_64.sh
```

```
bash Anaconda3-2019.03-Linux-x86_64.sh
```

## Configuring Jupyter Notebook settings

```
jupyter notebook --generate-config
```

Enter the IPython command line:

```
ipython
```

```
from IPython.lib import passwd
```

```
passwd()
```

You will be prompted to enter and re-enter your password. IPython will then generate a hash output, COPY THIS AND SAVE IT FOR LATER. We will need this for our configuration file.

Next go into your jupyter config file:

```
cd .jupyter
```

```
vim jupyter_notebook_config.py_
```

add the following code:

```
conf = get_config()
```

```
conf.NotebookApp.ip = '0.0.0.0'
```

```
conf.NotebookApp.password = u'YOUR PASSWORD HASH'
```

```
conf.NotebookApp.port = 8888
```

press “i” for insert mode

- press “esc” to escape

- press “shift+ z” to exit doc

Create a directory for your notebooks: `mkdir MyNotebooks`

## Configuring Jupyter Notebook settings

```
jupyter notebook --generate-config
```

Enter the IPython command line:

```
ipython
```

```
from IPython.lib import passwd
```

```
passwd()
```

You will be prompted to enter and re-enter your password. IPython will then generate a hash output, **COPY THIS AND SAVE IT FOR LATER**. We will need this for our configuration file.

Next go into your jupyter config file:

```
cd .jupyter
```

```
vim jupyter_notebook_config.py_
```

add the following code:

```
conf = get_config()
```

```
conf.NotebookApp.ip = '0.0.0.0'
```

```
conf.NotebookApp.password = u'YOUR PASSWORD HASH'
```

```
conf.NotebookApp.port = 8888
```

press “i” for insert mode

- press “esc” to escape
- press “shift+ z” to exit doc

Create a directory for your notebooks: `mkdir MyNotebooks`

```
In [1]: from IPython.lib import passwd  
  
In [2]: passwd()  
Enter password:  
Verify password:  
Out[2]: 'sha1:bfc470d205e3:03d34d2c808f1cb865dc495327d69a9e873a2e01'  
  
In [3]: ^C  
KeyboardInterrupt  
  
In [3]: cd .jupyter  
/home/ec2-user/.jupyter
```

## Connecting to your EC2 Jupyter Server

jupyter notebook

From there you should be able to access your server by going to:

[https://\(your AWS dns\):8888/](https://(your AWS dns):8888/)

Enter the password you created and you should be greeted with your full Jupyter Notebook system

```
In [6]: exit
[ec2-user@ip-10-0-1-230 ~]$ vim jupyter_notebook_config.py_
[ec2-user@ip-10-0-1-230 ~]$ mkdir MyNotebooks
[ec2-user@ip-10-0-1-230 ~]$ jupyter notebook
[W 07:41:12.242 NotebookApp] Unrecognized JSON config file version, assuming version 1
[I 07:41:12.641 NotebookApp] [nb_conda_kernels] enabled, 1 kernels found
[I 07:41:12.652 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook_cookie_secret
[I 07:41:13.240 NotebookApp] ✓ nbpresent HTML export ENABLED
[W 07:41:13.240 NotebookApp] ✗ nbpresent PDF export DISABLED: No module named nbpresent.pdf.exporters.pdf
[I 07:41:13.245 NotebookApp] [nb_conda] enabled
[I 07:41:13.388 NotebookApp] [nb_anacondacloud] enabled
[I 07:41:13.389 NotebookApp] Serving notebooks from local directory: /home/ec2-user
[I 07:41:13.390 NotebookApp] 0 active kernels
[I 07:41:13.390 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 07:41:13.390 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

## Installing boto3 and Connecting to S3 Bucket

```
!pip install boto3
```

```
import boto3
```

```
client=boto3.client('s3')
```

```
path='s3://recommenderbucket/searchengine.csv'
```

```
path1='s3://recommenderbucket/u.data'
```

```
In [3]: !pip install boto3
```

```
Collecting boto3
  Using cached boto3-1.17.95-py2.py3-none-any.whl (131 kB)
Collecting jmespath<1.0.0,>=0.7.1
  Using cached jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting botocore<1.21.0,>=1.20.95
  Using cached botocore-1.20.95-py2.py3-none-any.whl (7.6 MB)
Collecting s3transfer<0.5.0,>=0.4.0
  Using cached s3transfer-0.4.2-py2.py3-none-any.whl (79 kB)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\sai anisha\anaconda3\lib\site-packages (from botocore<1.21.0,>=1.20.95->boto3) (2.8.1)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in c:\users\sai anisha\anaconda3\lib\site-packages (from botocore<1.21.0,>=1.20.95->boto3) (1.25.8)
Requirement already satisfied: six>=1.5 in c:\users\sai anisha\anaconda3\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.21.0,>=1.20.95->boto3) (1.14.0)
Installing collected packages: jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.17.95 botocore-1.20.95 jmespath-0.10.0 s3transfer-0.4.2
```

```
In [4]: import boto3
```

```
In [5]: client=boto3.client('s3')
```

```
In [14]: path='s3://recommenderbucket/searchengine.csv'
path1='s3://recommenderbucket/u.data'
```



## Importing necessary and Installing s3fs

```
import numpy as np
import pandas as pd
```

```
column_names = ['user_id', 'keyword_id', 'count', 'timestamp']
```

```
!pip install s3fs
```

```
df=pd.read_csv(path)
df.head()
```

```
In [7]: import numpy as np
import pandas as pd
```

```
In [8]: column_names = ['user_id', 'keyword_id', 'count', 'timestamp']
```

```
In [10]: !pip install s3fs
```

```
Collecting s3fs
  Downloading s3fs-2021.6.0-py3-none-any.whl (24 kB)
Collecting fsspec==2021.06.0
  Downloading fsspec-2021.6.0-py3-none-any.whl (114 kB)
Collecting aiobotocore>=1.0.1
  Downloading aiobotocore-1.3.1.tar.gz (48 kB)
Collecting botocore<1.20.50,>=1.20.49
  Downloading botocore-1.20.49-py2.py3-none-any.whl (7.4 MB)
Collecting aiohttp>=3.3.1
  Downloading aiohttp-3.7.4.post0-cp37-cp37m-win_amd64.whl (630 kB)
Requirement already satisfied: wrapt>=1.10.10 in c:\users\sai anisha\anaconda3\lib\site-packages (from aiobotocore>=1.0.1->s3fs) (1.11.2)
Collecting aioitertools>=0.5.1
  Downloading aioitertools-0.7.1-py3-none-any.whl (20 kB)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in c:\users\sai anisha\anaconda3\lib\site-packages (from botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs) (0.10.0)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in c:\users\sai anisha\anaconda3\lib\site-packages (from botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs) (1.25.8)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\sai anisha\anaconda3\lib\site-packages (from botocore<1.20.50,>=1.20.49->aiobotocore>=1.0.1->s3fs) (2.8.1)
```

```
In [12]: df=pd.read_csv(path)
df.head()
```

Out[12]:

	keyword_id	keyword
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

Reading the CSV file using the path variable

Displaying the top 5 values in the dataset

```
df = pd.read_csv(path1, sep='\t', names=column_names)
```

```
df.head()
```

```
titles = pd.read_csv(path)
```

```
titles.head()
```

```
In [12]: df=pd.read_csv(path)
df.head()
```

Out[12]:

	keywod_id	keyword
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

```
In [16]: df = pd.read_csv(path1, sep='\t', names=column_names)
```

```
In [17]: df.head()
```

Out[17]:

	user_id	keywod_id	count	timestamp
0	0	50	5	881250949
1	0	172	5	881250949
2	0	133	1	881250949
3	196	242	3	881250949
4	186	302	3	891717742

```
In [18]: titles = pd.read_csv(path)
titles.head()
```

Out[18]:

	keywod_id	keyword
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

## Sorting dataset using Pandas

```
df = pd.merge(df,titles,on='keywod_id')
df.head()
```

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('white')
%matplotlib inline
```

```
df.groupby('keyword')['count'].mean().sort_values(ascending=False).head()
```

```
df.groupby('keyword')['count'].count().sort_values(ascending=False).head()
```

```
ratings = pd.DataFrame(df.groupby('keyword')['count'].mean())
ratings.head()
```

```
In [20]: import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('white')
%matplotlib inline
```

```
In [21]: df.groupby('keyword')['count'].mean().sort_values(ascending=False).head()
```

```
Out[21]: keyword
Marlene Dietrich: Shadow and Light (1996)    5.0
Prefontaine (1997)                          5.0
Santa with Muscles (1996)                   5.0
Star Kid (1997)                             5.0
Someone Else's America (1995)               5.0
Name: count, dtype: float64
```

```
In [22]: df.groupby('keyword')['count'].count().sort_values(ascending=False).head()
```

```
Out[22]: keyword
Star Wars (1977)          584
Contact (1997)            509
 Fargo (1996)             508
Return of the Jedi (1983)  507
Liar Liar (1997)          485
Name: count, dtype: int64
```

```
In [23]: ratings = pd.DataFrame(df.groupby('keyword')['count'].mean())
ratings.head()
```

```
Out[23]:
```

	count
keyword	
Till There Was You (1997)	2.333333
1-900 (1994)	2.600000
101 Dalmatians (1996)	2.908257
12 Angry Men (1957)	4.344000
187 (1997)	3.024390

## Data Visualization using Matlab and Seaborn

```
ratings['num of ratings'] = pd.DataFrame(df.groupby('keyword')['count'].count())
ratings.head()
```

```
plt.figure(figsize=(10,4))
ratings['num of ratings'].hist(bins=60)
```

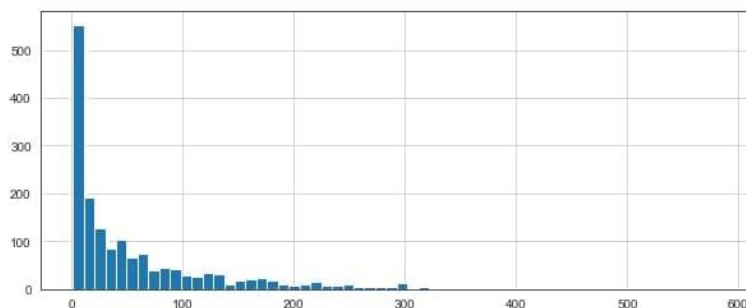
```
In [24]: ratings['num of ratings'] = pd.DataFrame(df.groupby('keyword')['count'].count())
ratings.head()
```

Out[24]:

	count	num of ratings
keyword		
'Til There Was You (1997)	2.333333	9
1-900 (1994)	2.600000	5
101 Dalmatians (1996)	2.908257	109
12 Angry Men (1957)	4.344000	125
187 (1997)	3.024390	41

```
In [25]: plt.figure(figsize=(10,4))
ratings['num of ratings'].hist(bins=60)
```

Out[25]: <matplotlib.axes\_subplots.AxesSubplot at 0x1b755105d88>

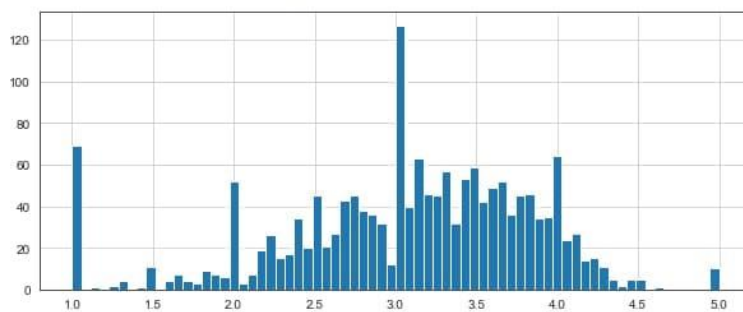


```
plt.figure(figsize=(10,4))
ratings['count'].hist(bins=70)
```

```
sns.jointplot(x='count',y='num of ratings',data=ratings,alpha=0.5)
```

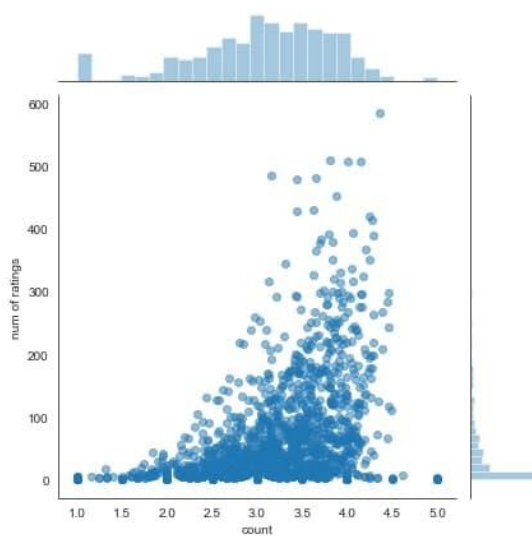
```
In [26]: plt.figure(figsize=(10,4))
ratings['count'].hist(bins=70)
```

Out[26]: <matplotlib.axes\_subplots.AxesSubplot at 0x1b75591c988>



```
In [27]: sns.jointplot(x='count',y='num of ratings',data=ratings,alpha=0.5)
```

Out[27]: <seaborn.axisgrid.JointGrid at 0x1b755aab248>



## Displaying the data

```
mat = df.pivot_table(index='user_id',columns='keyword',values='count')
mat.head()
```

```
ratings.sort_values('num of ratings',ascending=False).head(10)
```

```
ratings.head()
```

```
In [28]: mat = df.pivot_table(index='user_id',columns='keyword',values='count')
mat.head()
```

Out[28]:

keyword	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...	Yankee Zulu (1994)	Year of the Horse (1997)	You So Crazy (1994)	Young Frankenstein (1974)	Young Guns (1988)	Young Guns II (1990)	Poie Han The
user_id																		
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	2.0	5.0	NaN	NaN	3.0	4.0	NaN	NaN	...	NaN	NaN	NaN	5.0	3.0	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows x 1664 columns



```
In [29]: ratings.sort_values('num of ratings',ascending=False).head(10)
```

Out[29]:

	count	num of ratings
keyword		
Star Wars (1977)	4.359589	584
Contact (1997)	3.803536	509
Fargo (1996)	4.155512	508
Return of the Jedi (1983)	4.007890	507
Liar Liar (1997)	3.156701	485
English Patient, The (1996)	3.656965	481
Scream (1996)	3.441423	478
Toy Story (1995)	3.878319	452
Air Force One (1997)	3.631090	431
Independence Day (ID4) (1996)	3.438228	429

```
starwars_user_ratings = mat['Star Wars (1977)']
starwars_user_ratings.head()
```

```
similar_to_starwars = mat.corrwith(starwars_user_ratings)
```

```
corr_starwars = pd.DataFrame(similar_to_starwars, columns=['Correlation'])
corr_starwars.dropna(inplace=True)
corr_starwars.head()
```

```
In [30]: ratings.head()
```

```
Out[30]:
```

	count	num of ratings
keyword		
Til There Was You (1997)	2.333333	9
1-900 (1994)	2.600000	5
101 Dalmatians (1996)	2.908257	109
12 Angry Men (1957)	4.344000	125
187 (1997)	3.024390	41

```
In [31]: starwars_user_ratings = mat['Star Wars (1977)']
starwars_user_ratings.head()
```

```
Out[31]:
```

```
user_id
0  5.0
1  5.0
2  5.0
3  NaN
4  5.0
Name: Star Wars (1977), dtype: float64
```

```
In [32]: similar_to_starwars = mat.corrwith(starwars_user_ratings)
```

```
C:\Users\sai anisha\anaconda3\lib\site-packages\numpy\lib\function_base.py:2526: RuntimeWarning: Degrees of freedom <= 0 for slice
c = cov(x, y, rowvar)
C:\Users\sai anisha\anaconda3\lib\site-packages\numpy\lib\function_base.py:2455: RuntimeWarning: divide by zero encountered in true_divide
c *= np.true_divide(1, fact)
```

```
In [33]: corr_starwars = pd.DataFrame(similar_to_starwars, columns=['Correlation'])
corr_starwars.dropna(inplace=True)
corr_starwars.head()
```

```
Out[33]:
```

	Correlation
keyword	
Til There Was You (1997)	0.872872
1-900 (1994)	-0.645497
101 Dalmatians (1996)	0.211132
12 Angry Men (1957)	0.184289



## Displaying Recommendations

```
corr_starwars.sort_values('Correlation',ascending=False).head(10)
```

```
corr_starwars = corr_starwars.join(ratings['num of ratings'])  
corr_starwars.head()
```

```
corr_starwars[corr_starwars['num of ratings']>100].sort_values('Correlation',ascending=False).head(2)
```

```
In [34]: corr_starwars.sort_values('Correlation',ascending=False).head(10)
```

Out[34]:

		Correlation
keyword		
Commandments (1997)		1.0
Cosi (1996)		1.0
No Escape (1994)		1.0
Stripes (1981)		1.0
Man of the Year (1995)		1.0
Hollow Reed (1996)		1.0
Beans of Egypt, Maine, The (1994)		1.0
Good Man in Africa, A (1994)		1.0
Old Lady Who Walked in the Sea, The (Vieille qui marchait dans la mer, La) (1991)		1.0
Outlaw, The (1943)		1.0

```
In [35]: corr_starwars = corr_starwars.join(ratings['num of ratings'])  
corr_starwars.head()
```

Out[35]:

		Correlation	num of ratings
keyword			
'Til There Was You (1997)		0.872872	9
1-900 (1994)		-0.645497	5
101 Dalmatians (1996)		0.211132	109
12 Angry Men (1957)		0.184289	125
187 (1997)		0.027398	41

```
In [36]: corr_starwars[corr_starwars['num of ratings']>100].sort_values('Correlation',ascending=False).head(2)
```

Out[36]:

		Correlation	num of ratings
keyword			
Star Wars (1977)		1.000000	584
Empire Strikes Back, The (1980)		0.748353	368



# Thank You