Final Project Document

Project Topic

Alibaba Digital Marketing

Team Details

Team 5

Prathamesh Verlekar

Yash Pandya

Overview

Alibaba Group Holding Limited (also known as **Alibaba Group** and **Alibaba.com**) is a Chinese multinational technology company specializing in e-commerce, retail, Internet, and technology. On 19th September 2014, Alibaba's market value was US$231 billion. It is one of the top 10 most valuable and is the 59th biggest public company in the world by the Global 2000 list.

As of 2020, Alibaba has the 6th highest global brand valuation. Alibaba is the world's largest retailer and e-commerce company, is on the list of largest Interest companies and artificial intelligence companies, is one of the biggest venture capital firms, and one of the biggest investment corporations in the world. Its online sales and profits surpassed all US retailers (including Walmart, Amazon, and eBay) combined since 2015.

Goals

- Finding Recency, Frequency, and Monetary Value to find the company's best customers by using certain measures.
- To find the customers lost by calculating the churn rate and the customer's lifetime value.
- Build a recommendation system for suggesting products to the customers that they might also like.
- To Crawl, Index, and Rank data using the Search engine for better customer experience.
- Build a dashboard for business owners to better understand their sales and decide future marketing strategies.

➢ Create a web application for better user experience.

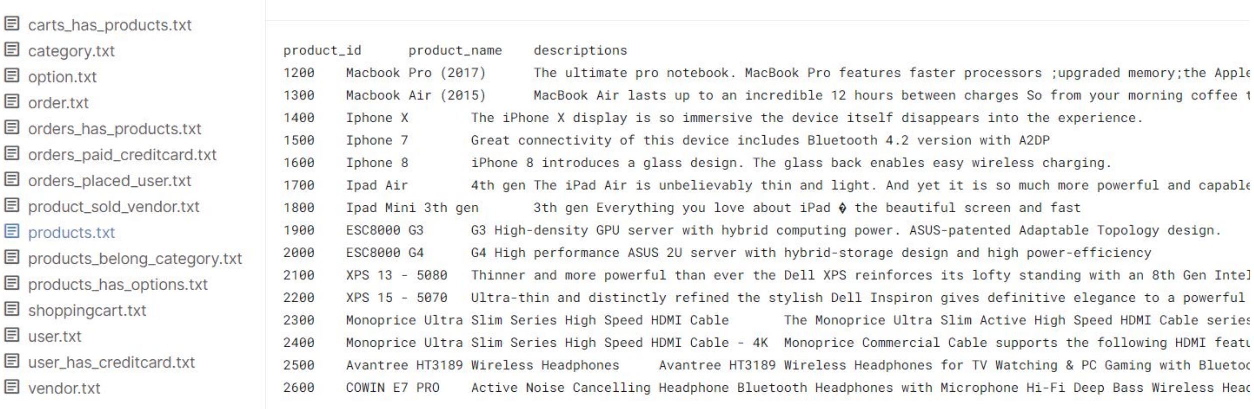
Dataset

➢ We will be using datasets available on Kaggle along with data available through different open sources and websites.

➢ Kaggle Dataset Link:https://www.kaggle.com/AppleEcomerceInfo/ecommerce-information?select=products.txt

➢ http://yongfeng.me/dataset/

Data Sample



- carts_has_products.txt
- category.txt
- option.txt
- order.txt
- orders_has_products.txt
- orders_paid_creditcard.txt
- orders_placed_user.txt
- product_sold_vendor.txt
- products.txt
- products_belong_category.txt
- products_has_options.txt
- shoppingcart.txt
- user.txt
- user_has_creditcard.txt
- vendor.txt

```
product_id      product_name    descriptions
1200    Macbook Pro (2017)      The ultimate pro notebook. MacBook Pro features faster processors ;upgraded memory;the Apple
1300    Macbook Air (2015)      MacBook Air lasts up to an incredible 12 hours between charges So from your morning coffee t
1400    Iphone X        The iPhone X display is so immersive the device itself disappears into the experience.
1500    Iphone 7        Great connectivity of this device includes Bluetooth 4.2 version with A2DP
1600    Iphone 8        iPhone 8 introduces a glass design. The glass back enables easy wireless charging.
1700    Ipad Air        4th gen The iPad Air is unbelievably thin and light. And yet it is so much more powerful and capable
1800    Ipad Mini 3th gen       3th gen Everything you love about iPad � the beautiful screen and fast
1900    ESC8000 G3      G3 High-density GPU server with hybrid computing power. ASUS-patented Adaptable Topology design.
2000    ESC8000 G4      G4 High performance ASUS 2U server with hybrid-storage design and high power-efficiency
2100    XPS 13 - 5080   Thinner and more powerful than ever the Dell XPS reinforces its lofty standing with an 8th Gen Intel
2200    XPS 15 - 5070   Ultra-thin and distinctly refined the stylish Dell Inspiron gives definitive elegance to a powerful
2300    Monoprice Ultra Slim Series High Speed HDMI Cable       The Monoprice Ultra Slim Active High Speed HDMI Cable series
2400    Monoprice Ultra Slim Series High Speed HDMI Cable - 4K  Monoprice Commercial Cable supports the following HDMI featu
2500    Avantree HT3189 Wireless Headphones     Avantree HT3189 Wireless Headphones for TV Watching & PC Gaming with Bluetoo
2600    COWIN E7 PRO    Active Noise Cancelling Headphone Bluetooth Headphones with Microphone Hi-Fi Deep Bass Wireless Head
```

Personas

Who – Alibaba’s technical teams, stakeholders, and sponsors for deciding their future marketing strategies.

What – Build a recommendation system for the company's customers and dashboards for stakeholders to analyze their sales and other insights.

When – It will be completed within 2 weeks of a given timeline.

Where – We will be working at our remote locations and then collaborate on our work.

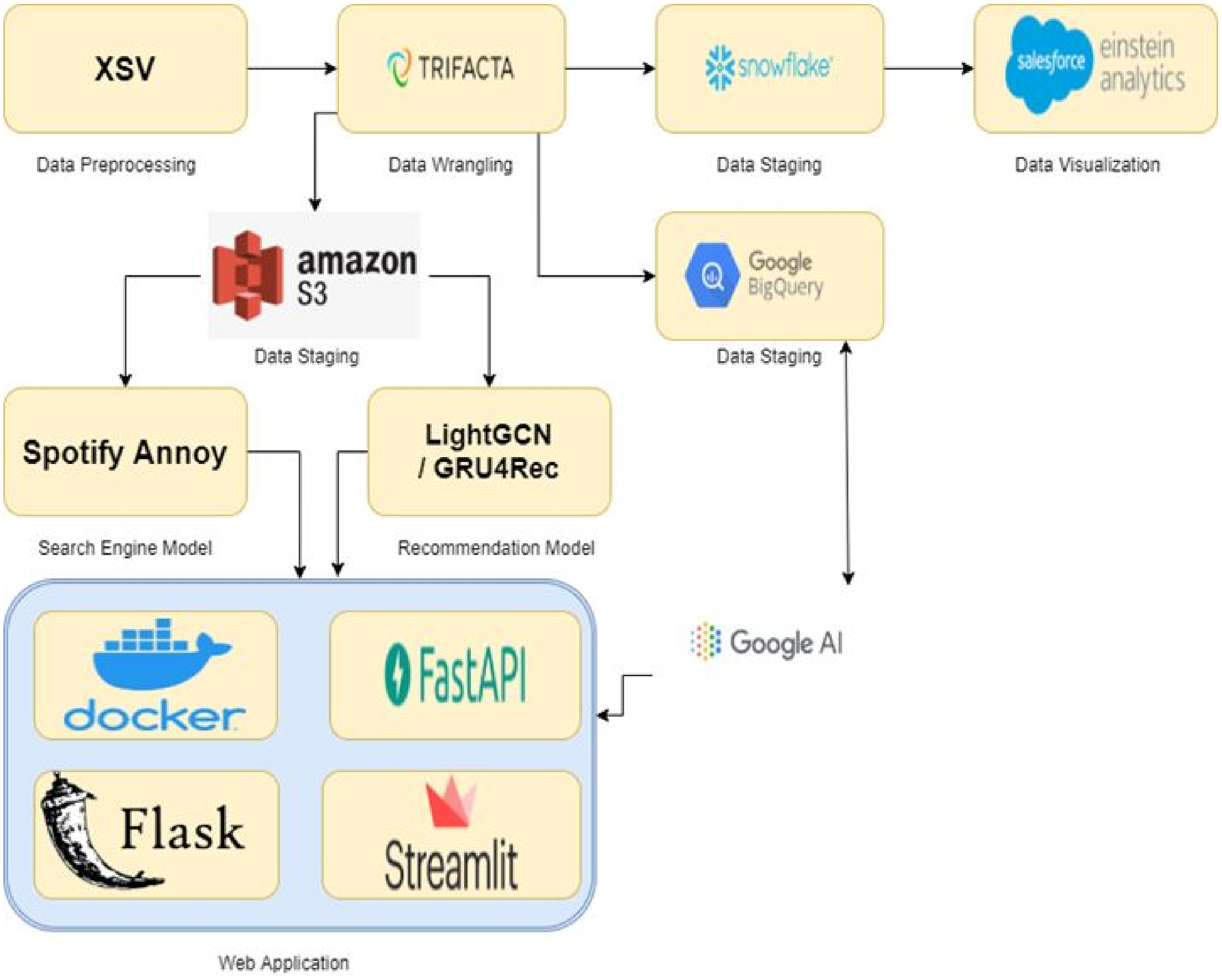
Why – To get data insights that will help the stakeholders decide their marketing strategies and to improve the customer experience by building a recommendation system that will recommend the customers product that may also like.

How – Using the tools and technologies learned in the course.

- ➢ **XSV**
- ➢ **Python**
- ➢ **Trifacta**
- ➢ **Salesforce Einstein Analytics**
- ➢ **Streamlit**
- ➢ **Flask**

- **JMeter**

Workflow



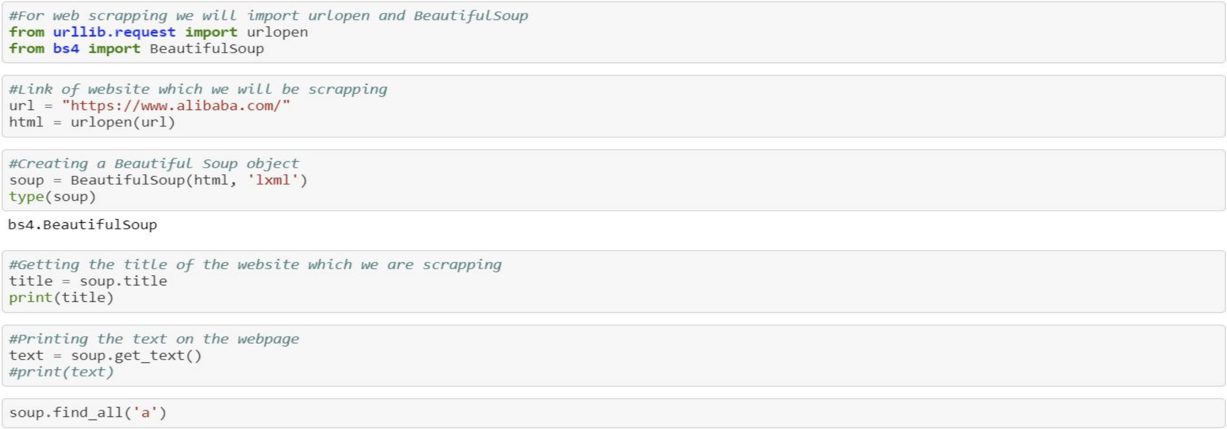
Use Cases

- **Making user shopping experience more efficient.**
- **Know the product trend to make future decisions based on it to increase profitability.**
- **Help stakeholders to make important decisions and changes.**

Web scraping

Web Scraping (also termed Screen Scraping, Web Data Extraction, Web Harvesting, etc.) is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format.

Code Sample:



```
#For web scrapping we will import urlopen and BeautifulSoup
from urllib.request import urlopen
from bs4 import BeautifulSoup
```

```
#Link of website which we will be scrapping
url = "https://www.alibaba.com/"
html = urlopen(url)
```

```
#Creating a Beautiful Soup object
soup = BeautifulSoup(html, 'lxml')
type(soup)
```

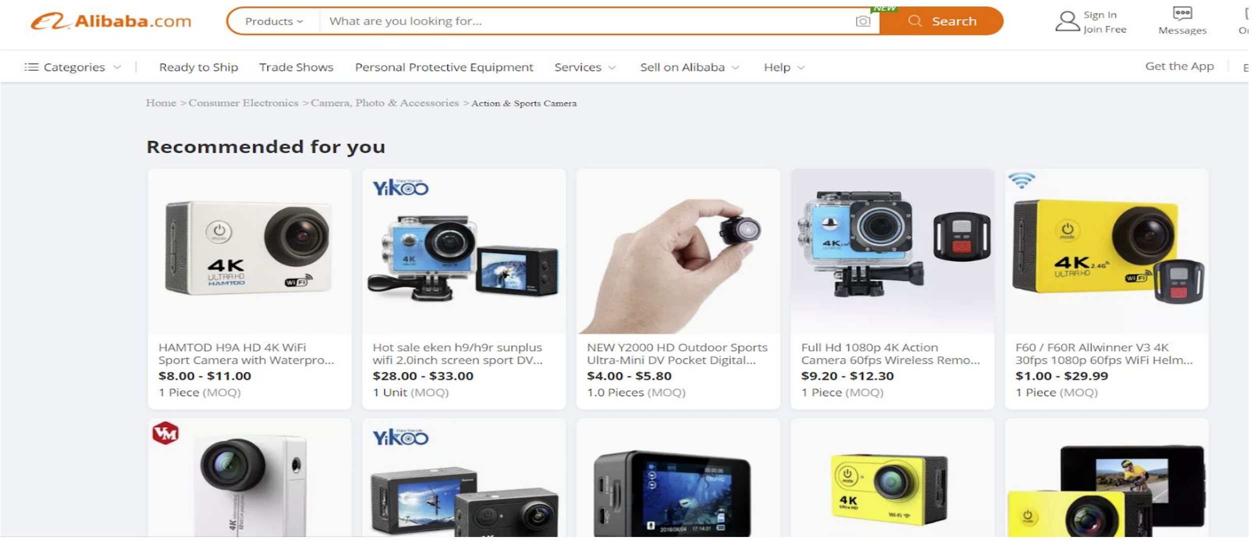
```
bs4.BeautifulSoup
```

```
#Getting the title of the website which we are scrapping
title = soup.title
print(title)
```

```
#Printing the text on the webpage
text = soup.get_text()
#print(text)
```

```
soup.find_all('a')
```

Website:



HAMTOD H9A HD 4K WiFi Sport Camera with Waterpro...
$8.00 - $11.00
1 Piece (MOQ)

Hot sale eken h9/h9r sunplus wifi 2.0inch screen sport DV...
$28.00 - $33.00
1 Unit (MOQ)

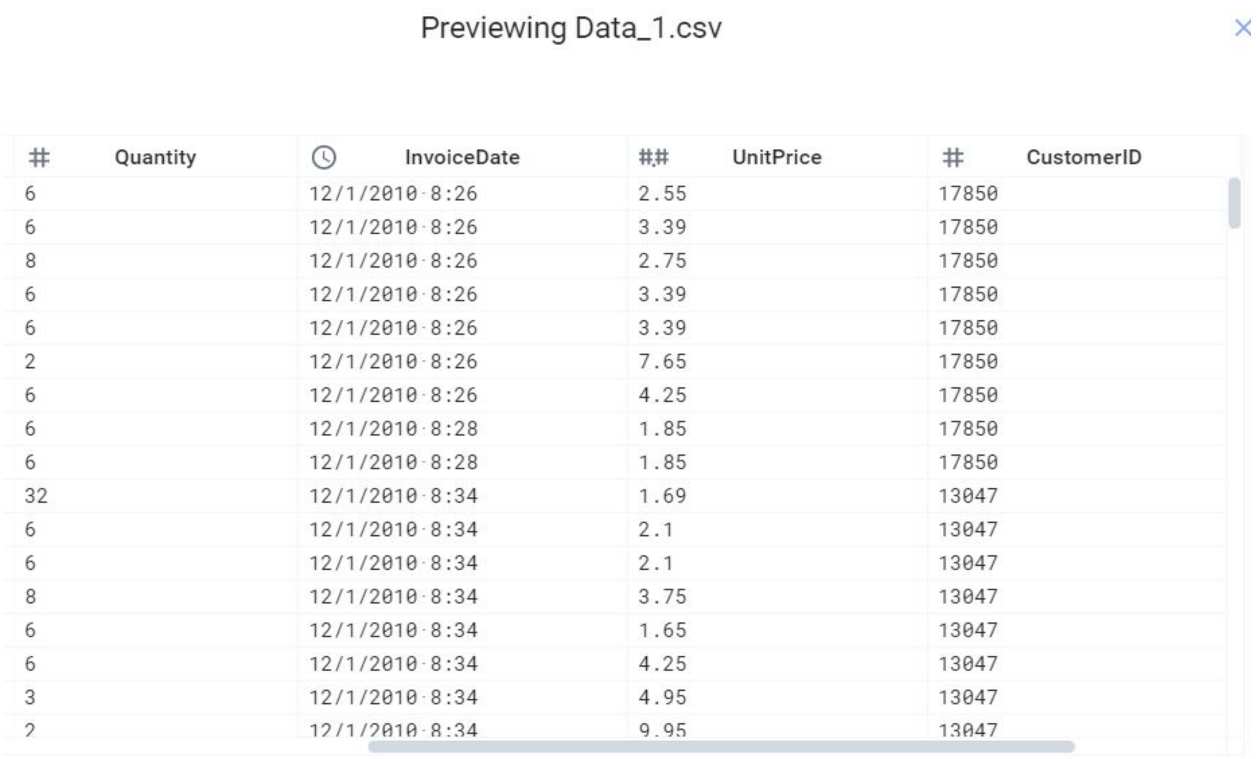
NEW Y2000 HD Outdoor Sports Ultra-Mini DV Pocket Digital...
$4.00 - $5.80
1.0 Pieces (MOQ)

Full Hd 1080p 4K Action Camera 60fps Wireless Remo...
$9.20 - $12.30
1 Piece (MOQ)

F60 / F60R Allwinner V3 4K 30fps 1080p 60fps WiFi Helm...
$1.00 - $29.99
1 Piece (MOQ)

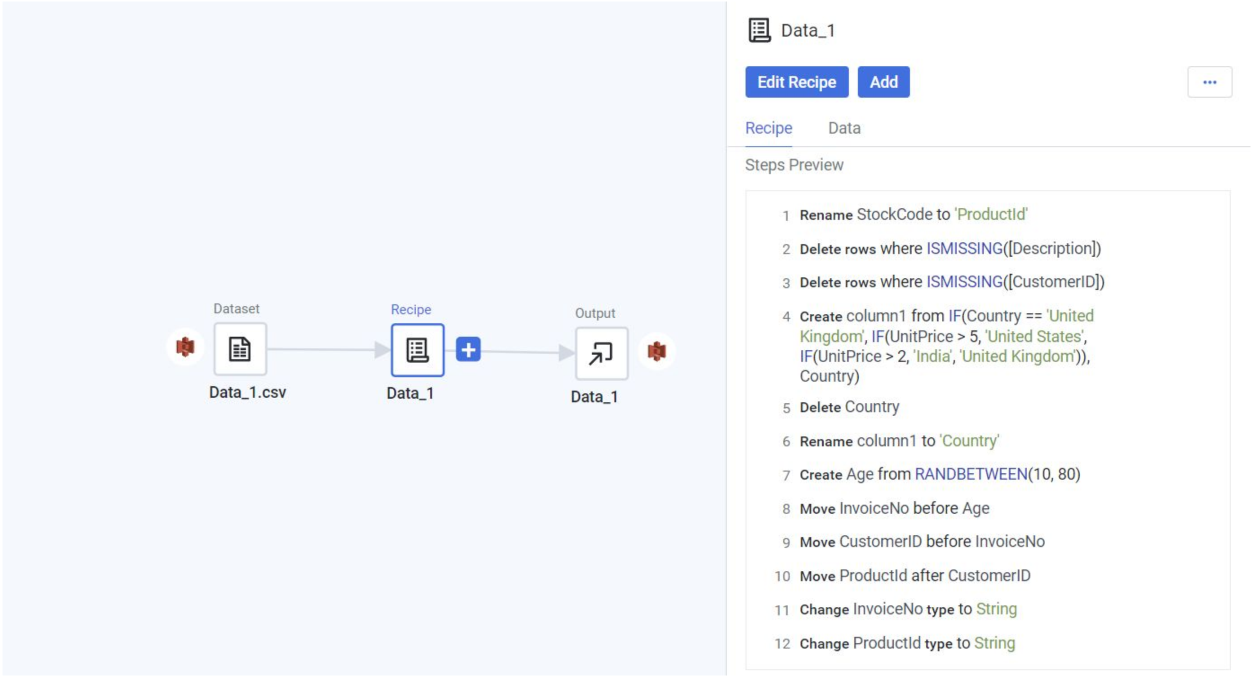
Trifacta (Data Wrangling)

Trifacta's data wrangling software allows you to prepare & visualize complex data in no time.



Previewing Data_1.csv

# Quantity	InvoiceDate	## UnitPrice	# CustomerID
6	12/1/2010 8:26	2.55	17850
6	12/1/2010 8:26	3.39	17850
8	12/1/2010 8:26	2.75	17850
6	12/1/2010 8:26	3.39	17850
6	12/1/2010 8:26	3.39	17850
2	12/1/2010 8:26	7.65	17850
6	12/1/2010 8:26	4.25	17850
6	12/1/2010 8:28	1.85	17850
6	12/1/2010 8:28	1.85	17850
32	12/1/2010 8:34	1.69	13047
6	12/1/2010 8:34	2.1	13047
6	12/1/2010 8:34	2.1	13047
8	12/1/2010 8:34	3.75	13047
6	12/1/2010 8:34	1.65	13047
6	12/1/2010 8:34	4.25	13047
3	12/1/2010 8:34	4.95	13047
2	12/1/2010 8:34	9.95	13047



Data_1

Steps Preview

1 Rename StockCode to 'ProductId'

2 Delete rows where ISMISSING([Description])

3 Delete rows where ISMISSING([CustomerID])

4 Create column1 from IF(Country == 'United Kingdom', IF(UnitPrice > 5, 'United States', IF(UnitPrice > 2, 'India', 'United Kingdom')), Country)

5 Delete Country

6 Rename column1 to 'Country'

7 Create Age from RANDBETWEEN(10, 80)

8 Move InvoiceNo before Age

9 Move CustomerID before InvoiceNo

10 Move ProductId after CustomerID

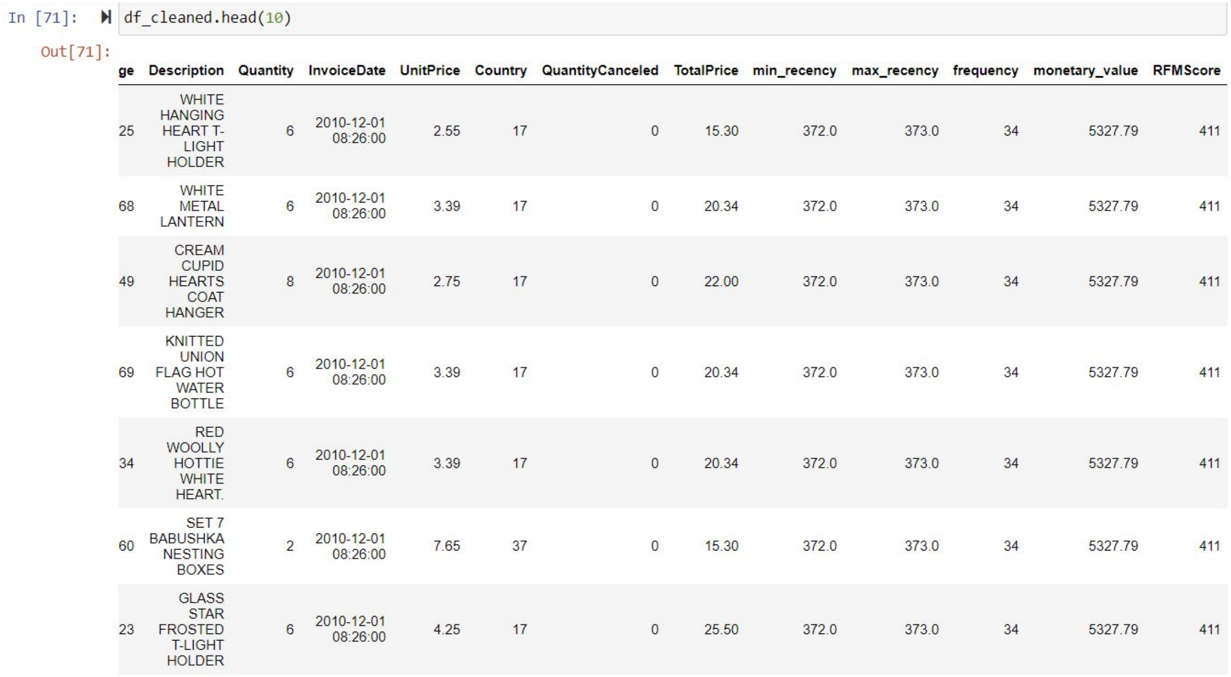
11 Change InvoiceNo type to String

12 Change ProductId type to String

RFM and Customer Segmentation

RFM stands for three dimensions:

- Recency – How recently did the customer purchase?
- Frequency – How often do they purchase?
- Monetary Value – How much do they spend?



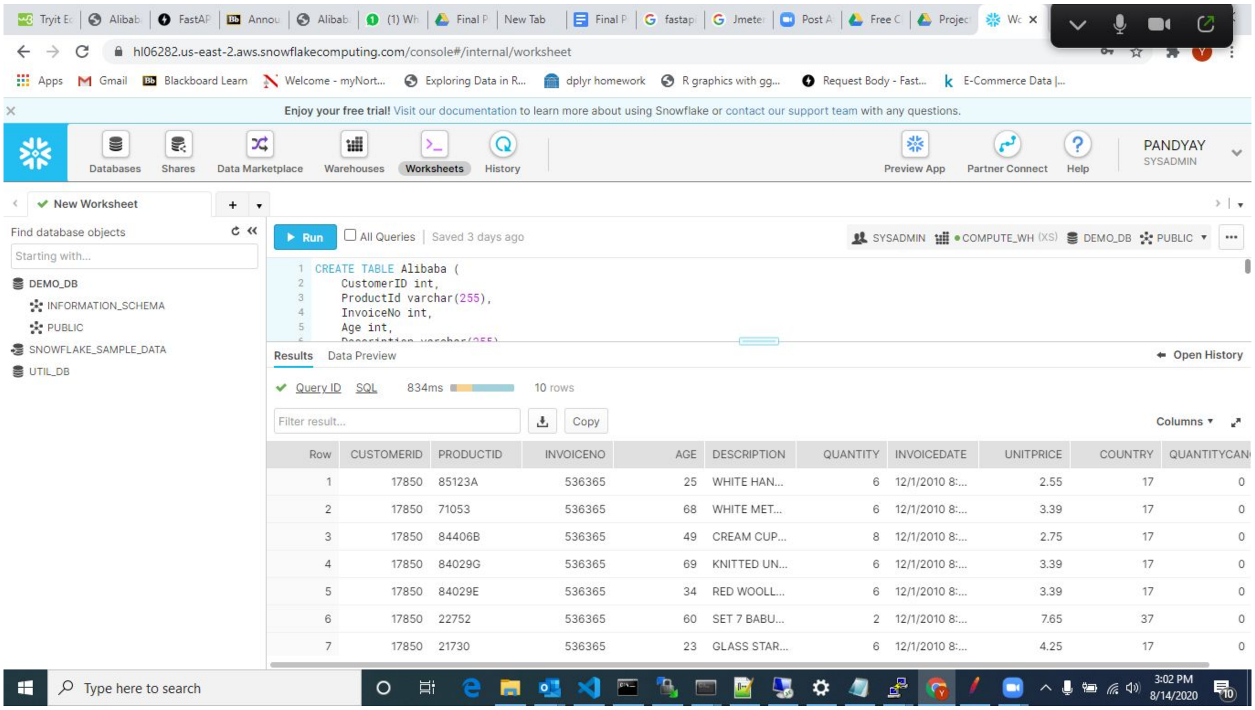
```
In [71]:  df_cleaned.head(10)
```

Out[71]:

ge	Description	Quantity	InvoiceDate	UnitPrice	Country	QuantityCanceled	TotalPrice	min_recency	max_recency	frequency	monetary_value	RFMScore
25	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17	0	15.30	372.0	373.0	34	5327.79	411
68	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17	0	20.34	372.0	373.0	34	5327.79	411
49	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17	0	22.00	372.0	373.0	34	5327.79	411
69	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17	0	20.34	372.0	373.0	34	5327.79	411
34	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17	0	20.34	372.0	373.0	34	5327.79	411
60	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	37	0	15.30	372.0	373.0	34	5327.79	411
23	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17	0	25.50	372.0	373.0	34	5327.79	411

SnowFlake

Snowflake is a cloud-based Data Warehouse solution provided as a Saas (Software-as-a-Service) with full support for ANSI SQL. It also has a unique architecture that enables users to just create tables and start querying data with very little administration or DBA activities needed.



Enjoy your free trial! Visit our documentation to learn more about using Snowflake or contact our support team with any questions.

```
CREATE TABLE Alibaba (
    CustomerID int,
    ProductId varchar(255),
    InvoiceNo int,
    Age int,
    Description varchar(255)
```

✓ 834ms 10 rows

Row	CUSTOMERID	PRODUCTID	INVOICENO	AGE	DESCRIPTION	QUANTITY	INVOICEDATE	UNITPRICE	COUNTRY	QUANTITYCAN
1	17850	85123A	536365	25	WHITE HAN...	6	12/1/2010 8:...	2.55	17	0
2	17850	71053	536365	68	WHITE MET...	6	12/1/2010 8:...	3.39	17	0
3	17850	84406B	536365	49	CREAM CUP...	8	12/1/2010 8:...	2.75	17	0
4	17850	84029G	536365	69	KNITTED UN...	6	12/1/2010 8:...	3.39	17	0
5	17850	84029E	536365	34	RED WOOLL...	6	12/1/2010 8:...	3.39	17	0
6	17850	22752	536365	60	SET 7 BABU...	2	12/1/2010 8:...	7.65	37	0
7	17850	21730	536365	23	GLASS STAR...	6	12/1/2010 8:...	4.25	17	0

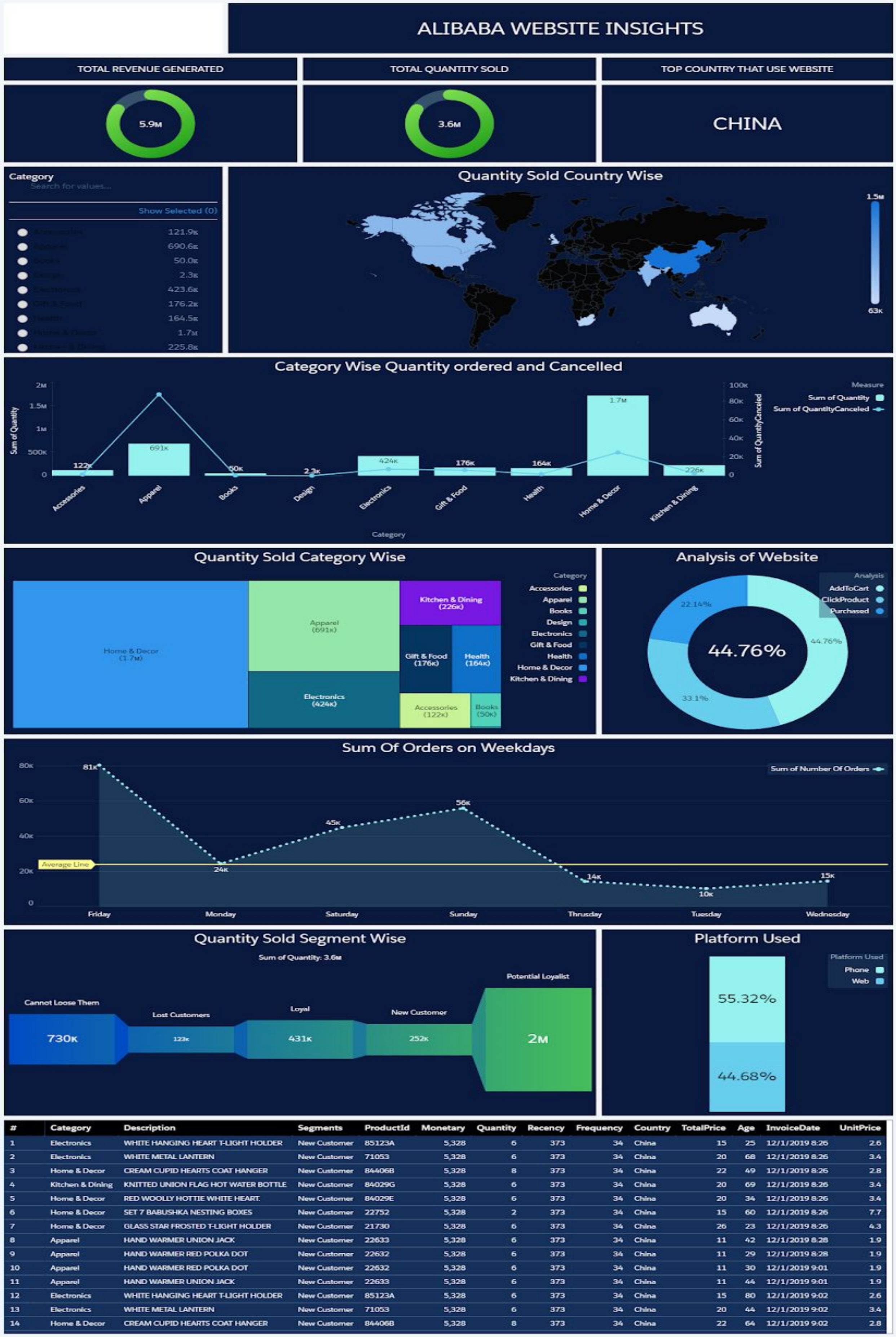
Salesforce Einstein Analytics Dashboard

Salesforce.com, inc. is an American cloud-based software company headquartered in San Francisco, California. It provides customer relationship management (CRM) service and also sells a complementary suite of enterprise applications focused on customer service, marketing automation, analytics, and application development.

- Total Revenue generated , Total Quantity sold, Top country that use the website
- Which Country has purchased most items
- Category wise items ordered and Cancelled
- How many items after placing in cart where purchased
- Number of orders based on weekdays
- Which has platform(Phone and Web) used for Most

- Quantity sold based on segments (Loyal, potential Loyalist, Cannot lose them, new Customer and Lost customers)

- Tablewise segmentation

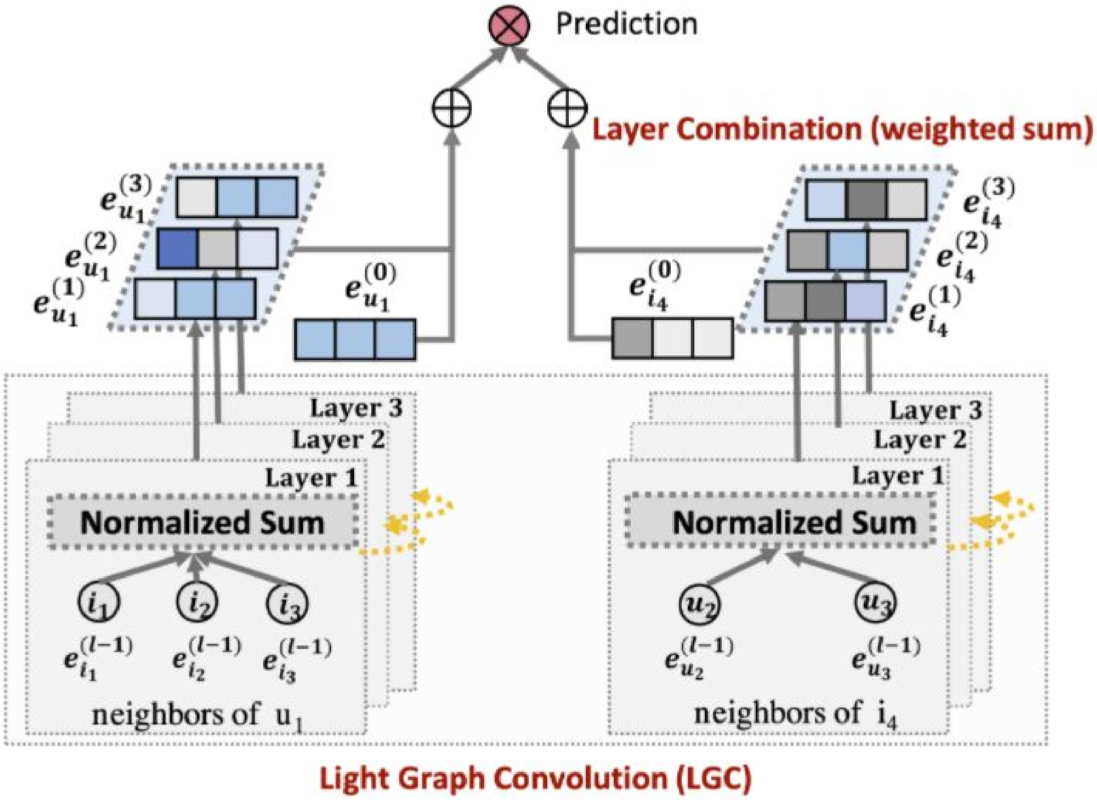


#	Category	Description	Segments	ProductId	Monetary	Quantity	Recency	Frequency	Country	TotalPrice	Age	InvoiceDate	UnitPrice
1	Electronics	WHITE HANGING HEART T-LIGHT HOLDER	New Customer	85123A	5,328	6	373	34	China	15	25	12/1/2019 8:26	2.6
2	Electronics	WHITE METAL LANTERN	New Customer	71053	5,328	6	373	34	China	20	68	12/1/2019 8:26	3.4
3	Home & Decor	CREAM CUPID HEARTS COAT HANGER	New Customer	84406B	5,328	8	373	34	China	22	49	12/1/2019 8:26	2.8
4	Kitchen & Dining	KNITTED UNION FLAG HOT WATER BOTTLE	New Customer	84029G	5,328	6	373	34	China	20	69	12/1/2019 8:26	3.4
5	Home & Decor	RED WOOLLY HOTTIE WHITE HEART.	New Customer	84029E	5,328	6	373	34	China	20	34	12/1/2019 8:26	3.4
6	Home & Decor	SET 7 BABUSHKA NESTING BOXES	New Customer	22752	5,328	2	373	34	China	15	60	12/1/2019 8:26	7.7
7	Home & Decor	GLASS STAR FROSTED T-LIGHT HOLDER	New Customer	21730	5,328	6	373	34	China	26	23	12/1/2019 8:26	4.3
8	Apparel	HAND WARMER UNION JACK	New Customer	22633	5,328	6	373	34	China	11	42	12/1/2019 8:28	1.9
9	Apparel	HAND WARMER RED POLKA DOT	New Customer	22632	5,328	6	373	34	China	11	29	12/1/2019 8:28	1.9
10	Apparel	HAND WARMER RED POLKA DOT	New Customer	22632	5,328	6	373	34	China	11	30	12/1/2019 9:01	1.9
11	Apparel	HAND WARMER UNION JACK	New Customer	22633	5,328	6	373	34	China	11	44	12/1/2019 9:01	1.9
12	Electronics	WHITE HANGING HEART T-LIGHT HOLDER	New Customer	85123A	5,328	6	373	34	China	15	80	12/1/2019 9:02	2.6
13	Electronics	WHITE METAL LANTERN	New Customer	71053	5,328	6	373	34	China	20	44	12/1/2019 9:02	3.4
14	Home & Decor	CREAM CUPID HEARTS COAT HANGER	New Customer	84406B	5,328	8	373	34	China	22	64	12/1/2019 9:02	2.8

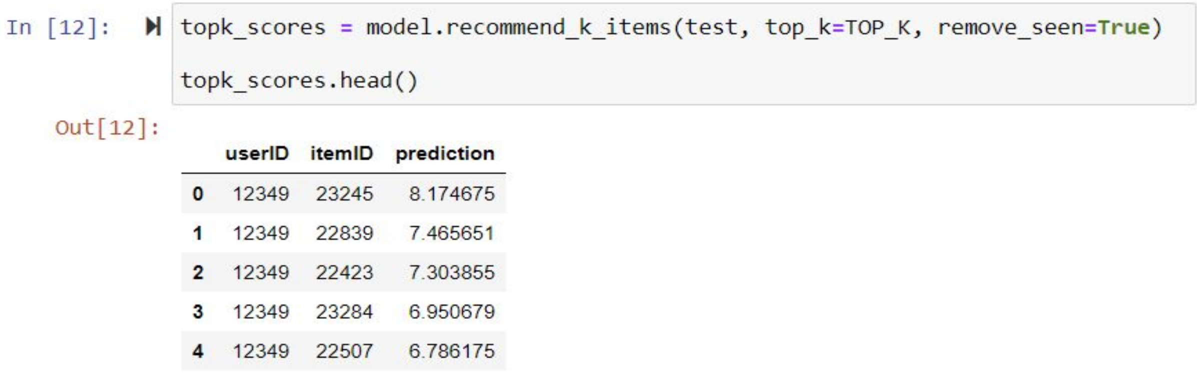
Recommendation Model

LightGCN

LightGCN is a simplified design of GCN to make it more concise and appropriate for a recommendation.



In LightGCN, we adopt the simple weighted sum aggregator and abandon the use of feature transformation and nonlinear activation.

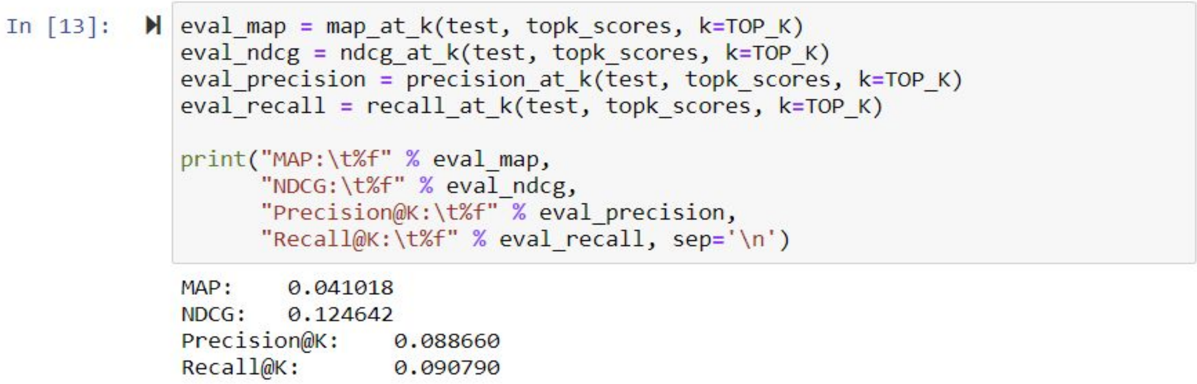


In [12]:

```
topk_scores = model.recommend_k_items(test, top_k=TOP_K, remove_seen=True)

topk_scores.head()
```

Out[12]:

	userID	itemID	prediction
0	12349	23245	8.174675
1	12349	22839	7.465651
2	12349	22423	7.303855
3	12349	23284	6.950679
4	12349	22507	6.786175



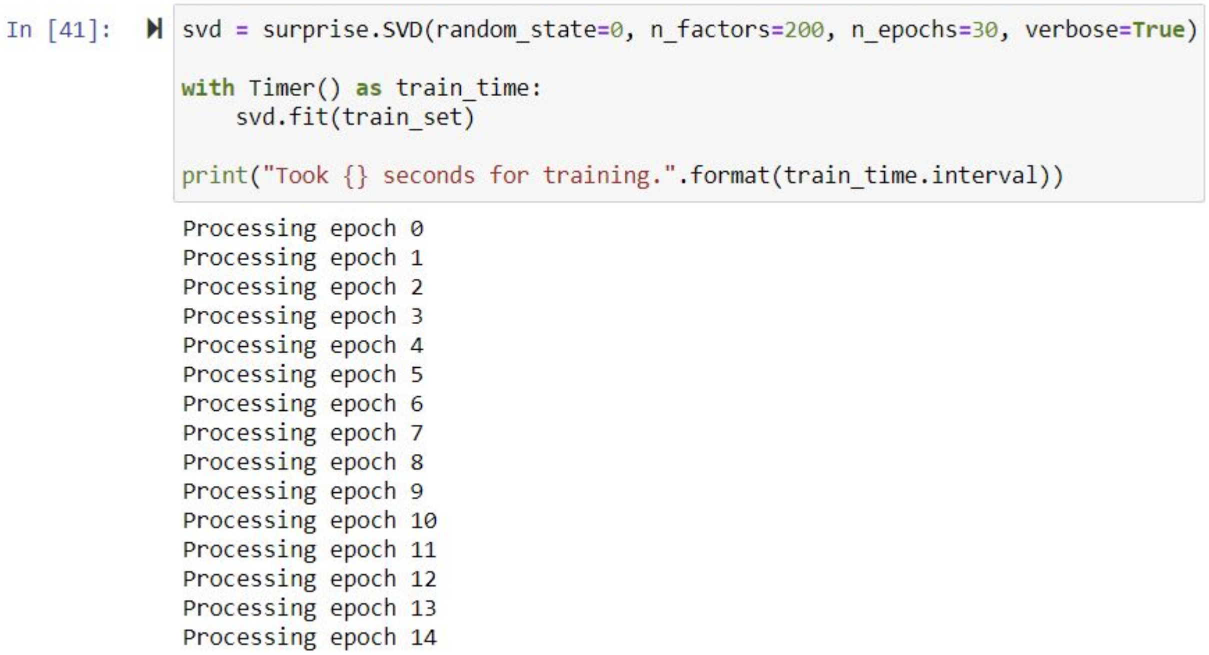
In [13]:

```
eval_map = map_at_k(test, topk_scores, k=TOP_K)
eval_ndcg = ndcg_at_k(test, topk_scores, k=TOP_K)
eval_precision = precision_at_k(test, topk_scores, k=TOP_K)
eval_recall = recall_at_k(test, topk_scores, k=TOP_K)

print("MAP:\t%f" % eval_map,
      "NDCG:\t%f" % eval_ndcg,
      "Precision@K:\t%f" % eval_precision,
      "Recall@K:\t%f" % eval_recall, sep='\n')
```

```
MAP:	0.041018
NDCG:	0.124642
Precision@K:	0.088660
Recall@K:	0.090790
```

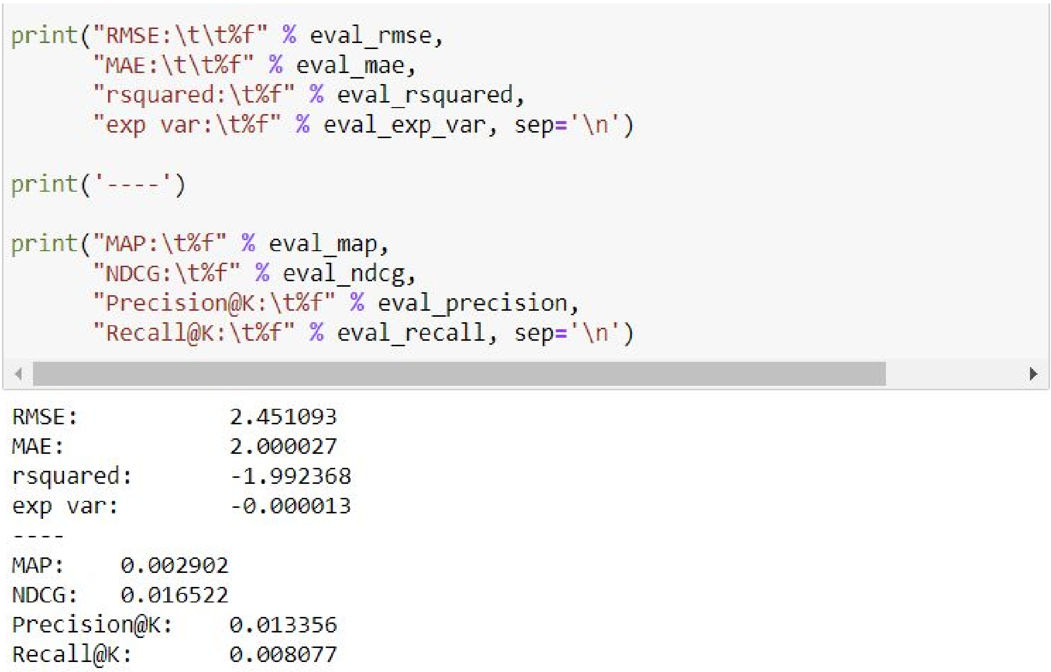
Surprise Singular Value Decomposition (SVD)

SVD introduces two new scalar variables: the user biases bu and item biases bi. The user biases are supposed to capture the tendency of some users to rate items higher (or lower) than the average. The same goes for items: some items are usually rated higher than some others.



```
In [41]:  svd = surprise.SVD(random_state=0, n_factors=200, n_epochs=30, verbose=True)

          with Timer() as train_time:
              svd.fit(train_set)

          print("Took {} seconds for training.".format(train_time.interval))
```

```
Processing epoch 0
Processing epoch 1
Processing epoch 2
Processing epoch 3
Processing epoch 4
Processing epoch 5
Processing epoch 6
Processing epoch 7
Processing epoch 8
Processing epoch 9
Processing epoch 10
Processing epoch 11
Processing epoch 12
Processing epoch 13
Processing epoch 14
```



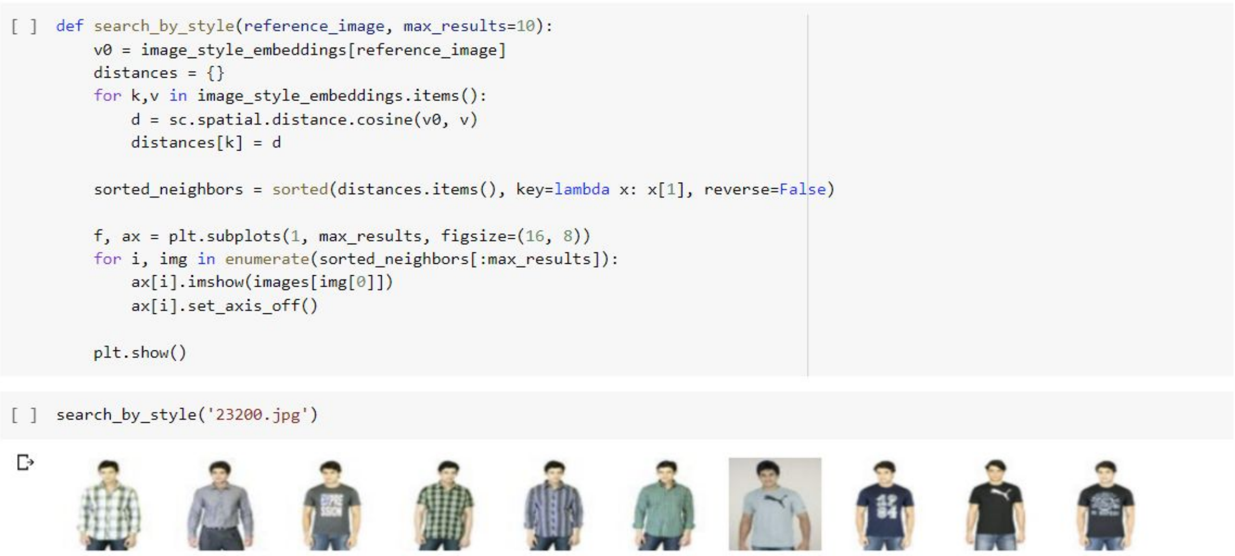
```
print("RMSE:\t\t%f" % eval_rmse,
      "MAE:\t\t%f" % eval_mae,
      "rsquared:\t%f" % eval_rsquared,
      "exp var:\t%f" % eval_exp_var, sep='\n')

print('----')

print("MAP:\t%f" % eval_map,
      "NDCG:\t%f" % eval_ndcg,
      "Precision@K:\t%f" % eval_precision,
      "Recall@K:\t%f" % eval_recall, sep='\n')
```

```
RMSE:           2.451093
MAE:            2.000027
rsquared:       -1.992368
exp var:        -0.000013
----
MAP:    0.002902
NDCG:   0.016522
Precision@K:    0.013356
Recall@K:       0.008077
```

Search Engine

Search By an Artistic Style

One typical business case is an eCommerce website that allows to search of a poster based on the example uploaded by the user. A user would usually expect to get results that are similar in terms of artistic style. A real search system would typically mix style similarity with other similarity scores such as image subject or category (landscape, still life, etc.) that can also be obtained using deep neural networks.



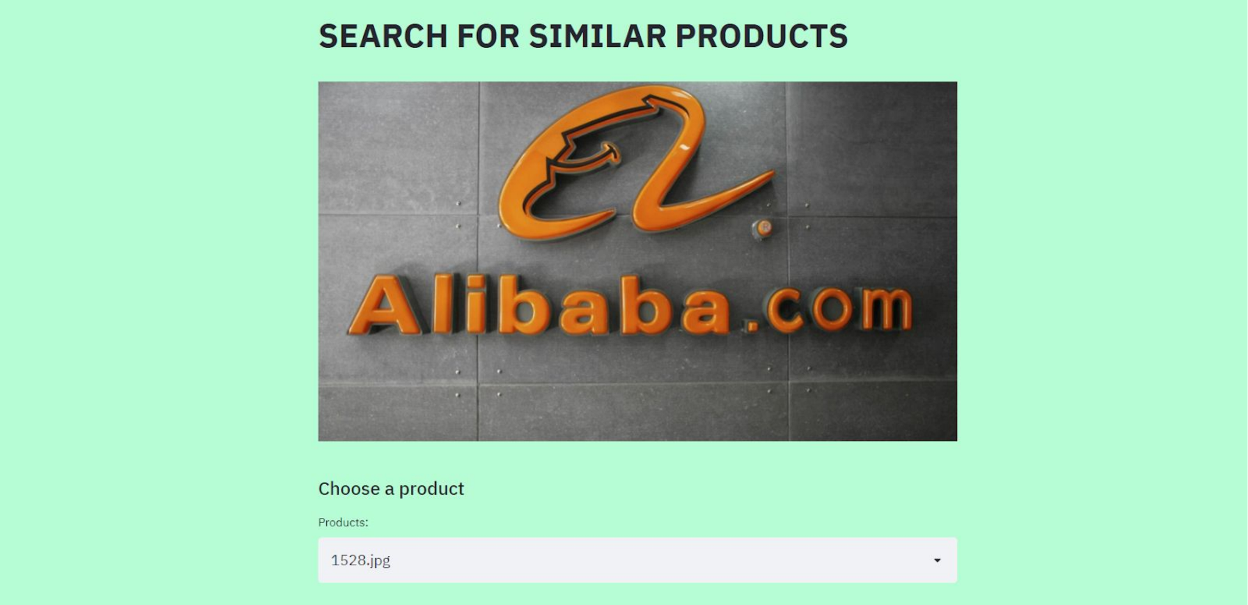
```
[ ]  def search_by_style(reference_image, max_results=10):
         v0 = image_style_embeddings[reference_image]
         distances = {}
         for k,v in image_style_embeddings.items():
             d = sc.spatial.distance.cosine(v0, v)
             distances[k] = d

         sorted_neighbors = sorted(distances.items(), key=lambda x: x[1], reverse=False)

         f, ax = plt.subplots(1, max_results, figsize=(16, 8))
         for i, img in enumerate(sorted_neighbors[:max_results]):
             ax[i].imshow(images[img[0]])
             ax[i].set_axis_off()

         plt.show()
```

```
[ ]  search_by_style('23200.jpg')
```

Streamlit Application

Streamlit is an open-source app framework for Machine Learning and Data Science teams. Create beautiful data apps in hours, not weeks. All in pure Python.

https://finalproject-streamlit-app.herokuapp.com/

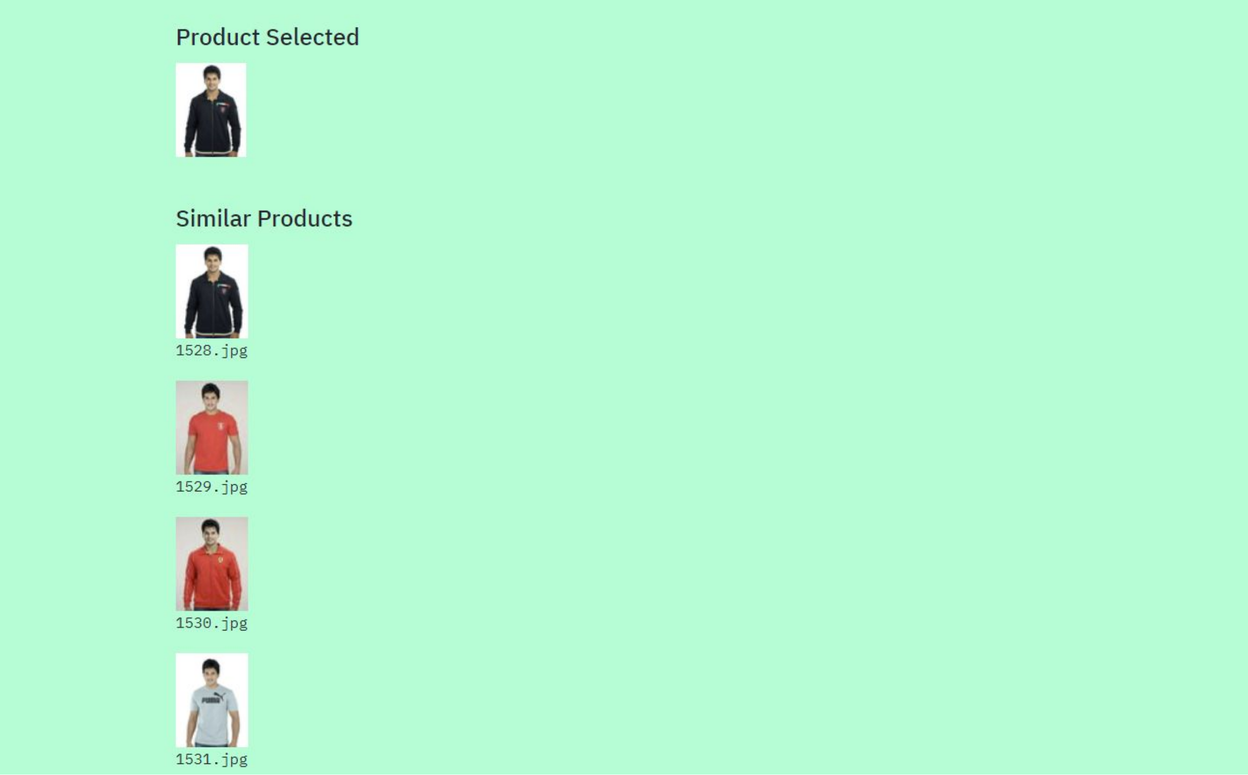


SEARCH FOR SIMILAR PRODUCTS

Choose a product

Products:

1528.jpg



Product Selected

Similar Products

1528.jpg

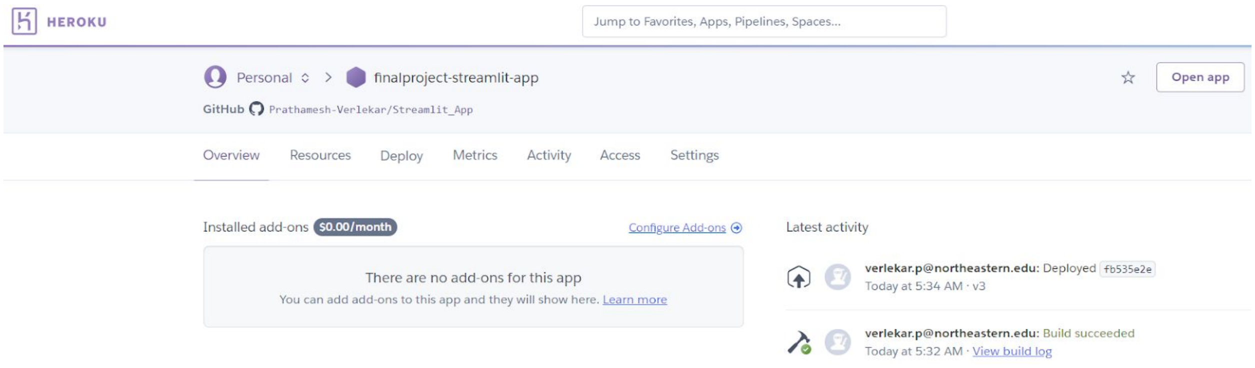
1529.jpg

1530.jpg

1531.jpg

Heroku Deployment

Heroku is a cloud platform as a service supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.



GitHub Prathamesh-Verlekar/Streamlit_App

Installed add-ons $0.00/month

Configure Add-ons

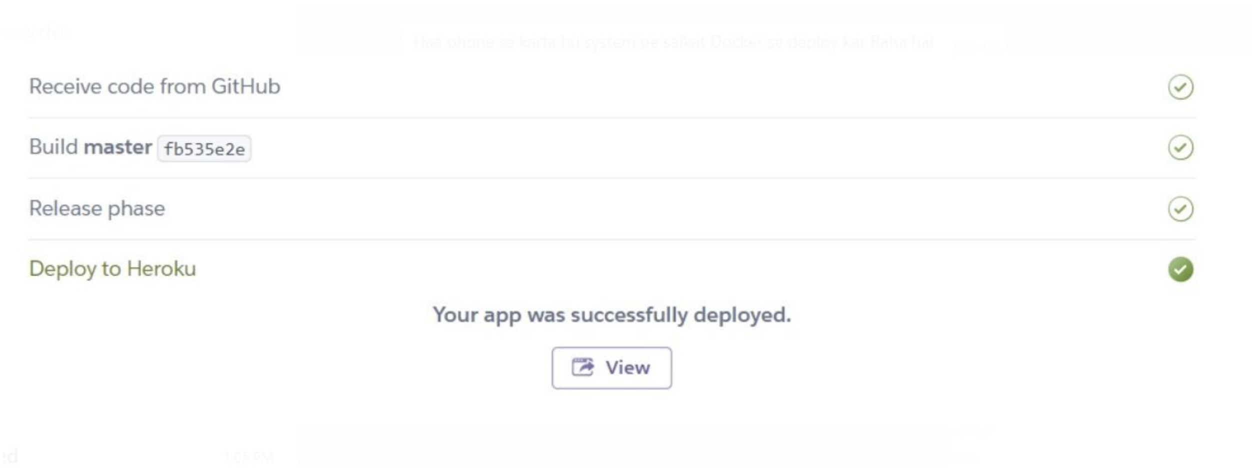
There are no add-ons for this app

You can add add-ons to this app and they will show here. Learn more

Latest activity

verlekar.p@northeastern.edu: Deployed fb535e2e
Today at 5:34 AM · v3

verlekar.p@northeastern.edu: Build succeeded
Today at 5:32 AM · View build log



Receive code from GitHub ✓

Build **master** fb535e2e ✓

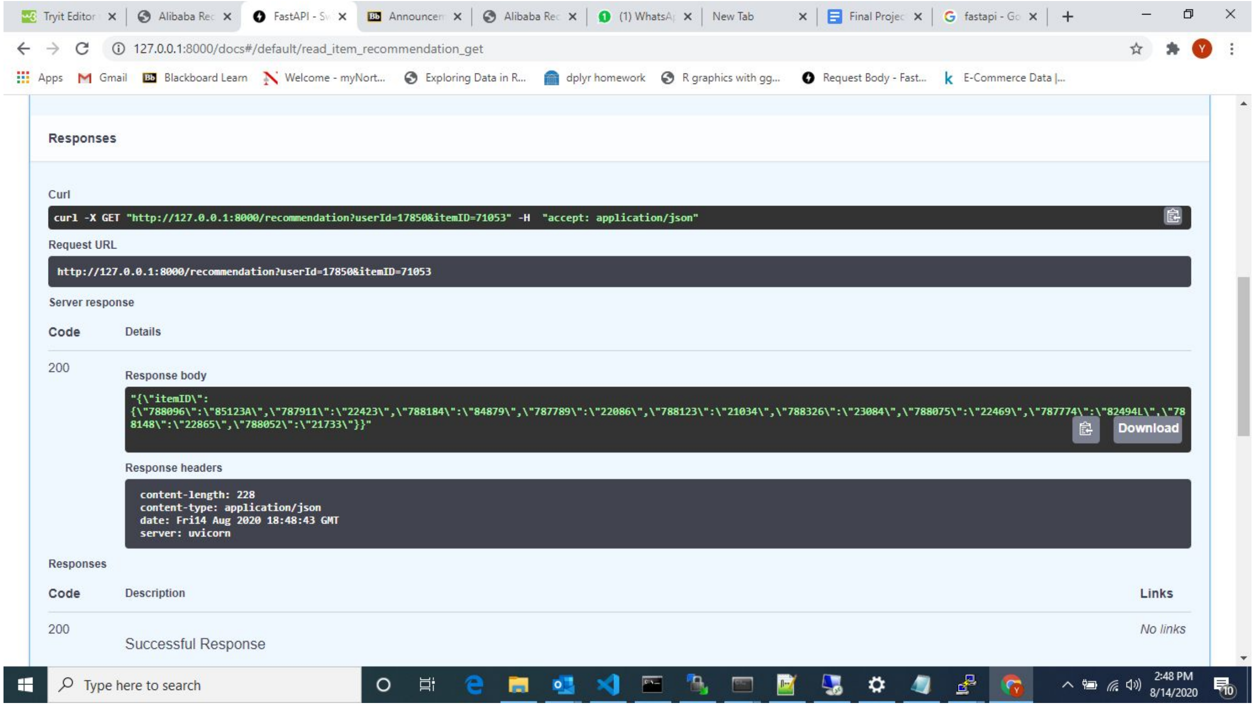
Release phase ✓

Deploy to Heroku ✓

Your app was successfully deployed.

FastAPI

FastAPI is a modern, fast (high-performance), a web framework for building APIs with Python 3.6+ based on standard Python type hints. The key features are: ... One of the fastest Python frameworks available. Fast to code: Increase the speed to develop features by about 200% to 300%.



127.0.0.1:8000/docs#/default/read_item_recommendation_get

Responses

Curl

```
curl -X GET "http://127.0.0.1:8000/recommendation?userId=17850&itemID=71053" -H  "accept: application/json"
```

Request URL

```
http://127.0.0.1:8000/recommendation?userId=17850&itemID=71053
```

Server response

Code Details

200

Response body

```
"{\"itemID\":
{\"788096\":\"85123A\",\"787911\":\"22423\",\"788184\":\"84879\",\"787789\":\"22086\",\"788123\":\"21034\",\"788326\":\"23084\",\"788075\":\"22469\",\"787774\":\"82494L\",\"78
8148\":\"22865\",\"788052\":\"21733\"}}"
```

Response headers

```
content-length: 228
content-type: application/json
date: Fri14 Aug 2020 18:48:43 GMT
server: uvicorn
```

Responses

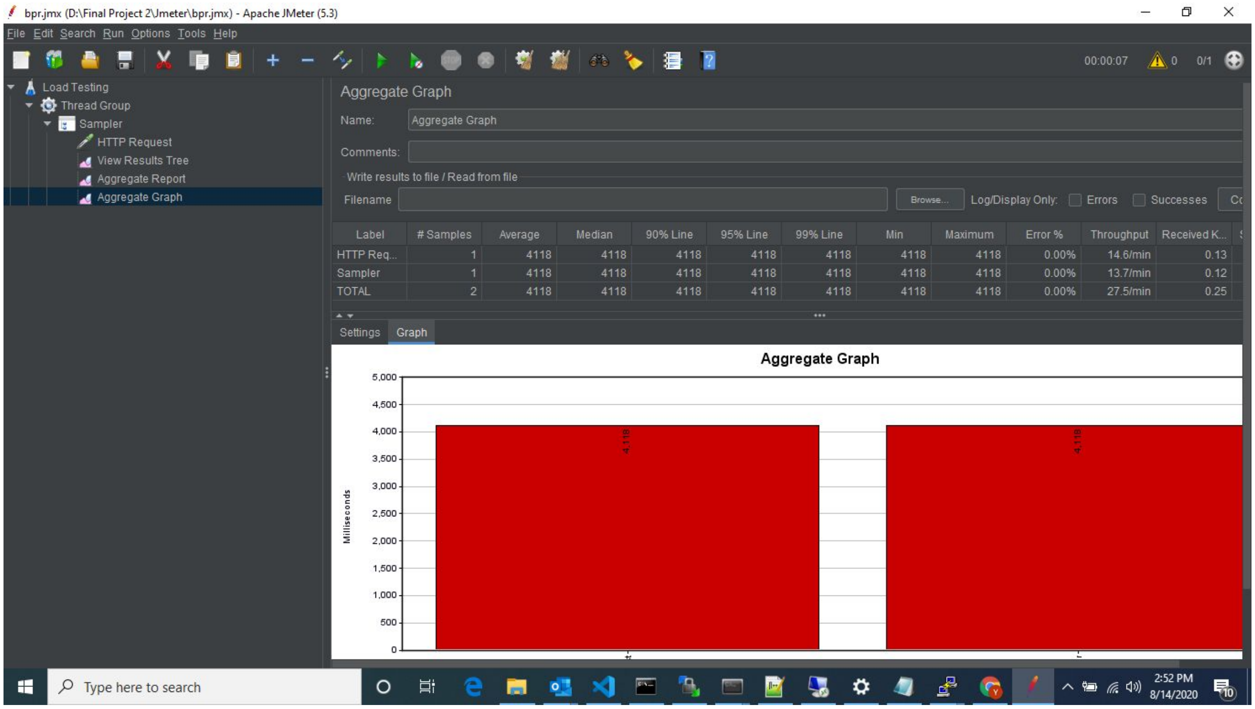
Code Description Links

200 Successful Response *No links*

JMeter

Apache **JMeter** is a **testing** tool used for analyzing and measuring the performance of different software services and products. It is a pure Java open source software used for **testing** Web Application or FTP application. It is used to execute performance **testing**, load **testing,** and functional **testing** of web applications.

We used Jmeter to test our three recommendation models and based on the throughput rate we were able to decide that LightGCN was much better and faster as compared to other models so we used LightGCN for building our recommendation web application.



Aggregate Graph

Name: Aggregate Graph

Comments:

Write results to file / Read from file

Filename [Browse...] Log/Display Only: □ Errors □ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...
HTTP Req...	1	4118	4118	4118	4118	4118	4118	4118	0.00%	14.6/min	0.13
Sampler	1	4118	4118	4118	4118	4118	4118	4118	0.00%	13.7/min	0.12
TOTAL	2	4118	4118	4118	4118	4118	4118	4118	0.00%	27.5/min	0.25

Flash Application Deployed on AWS

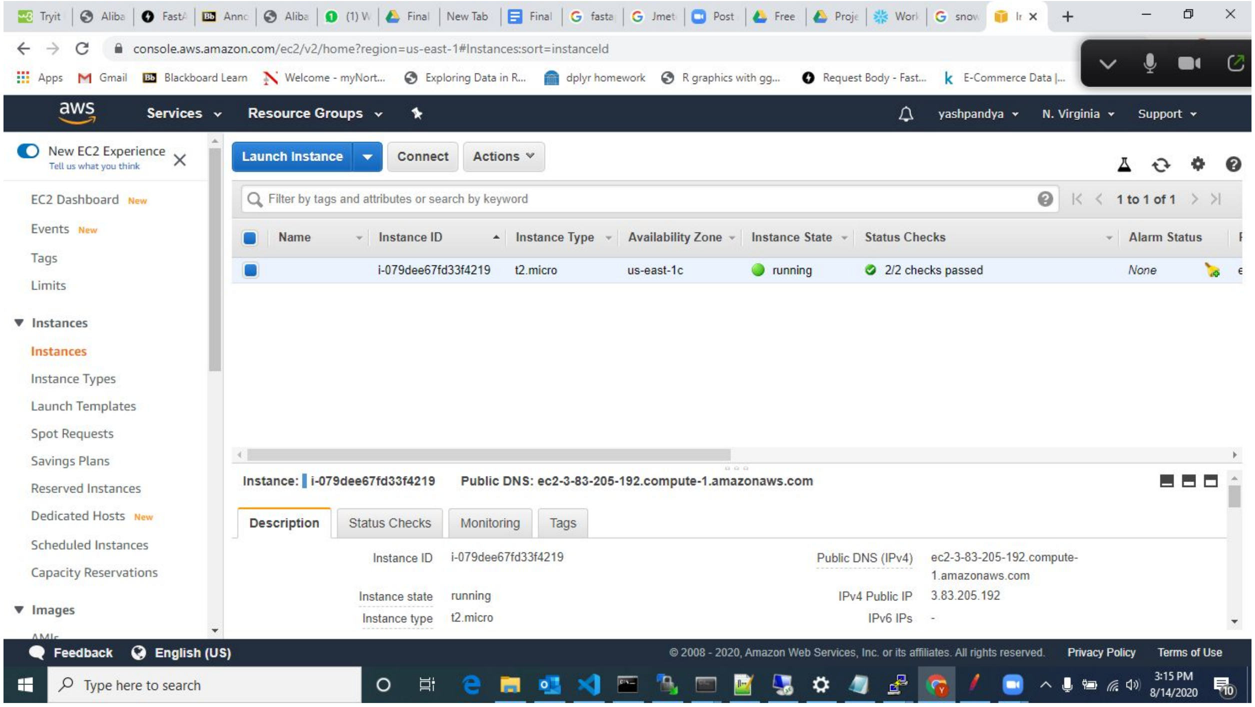
http://ec2-3-83-205-192.compute-1.amazonaws.com:5000/home.html

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

AWS provides on-demand access to scalable web and **application** servers, storage, databases, content delivery, cache, search, and other **application services** that make it easier to build and run apps that deliver a great customer experience.

Amazon EC2

1) AWS Instance created on AWS account



console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-079dee67fd33f4219	t2.micro	us-east-1c	running	2/2 checks passed	None

Instance: i-079dee67fd33f4219 Public DNS: ec2-3-83-205-192.compute-1.amazonaws.com

Instance ID i-079dee67fd33f4219

Instance state running

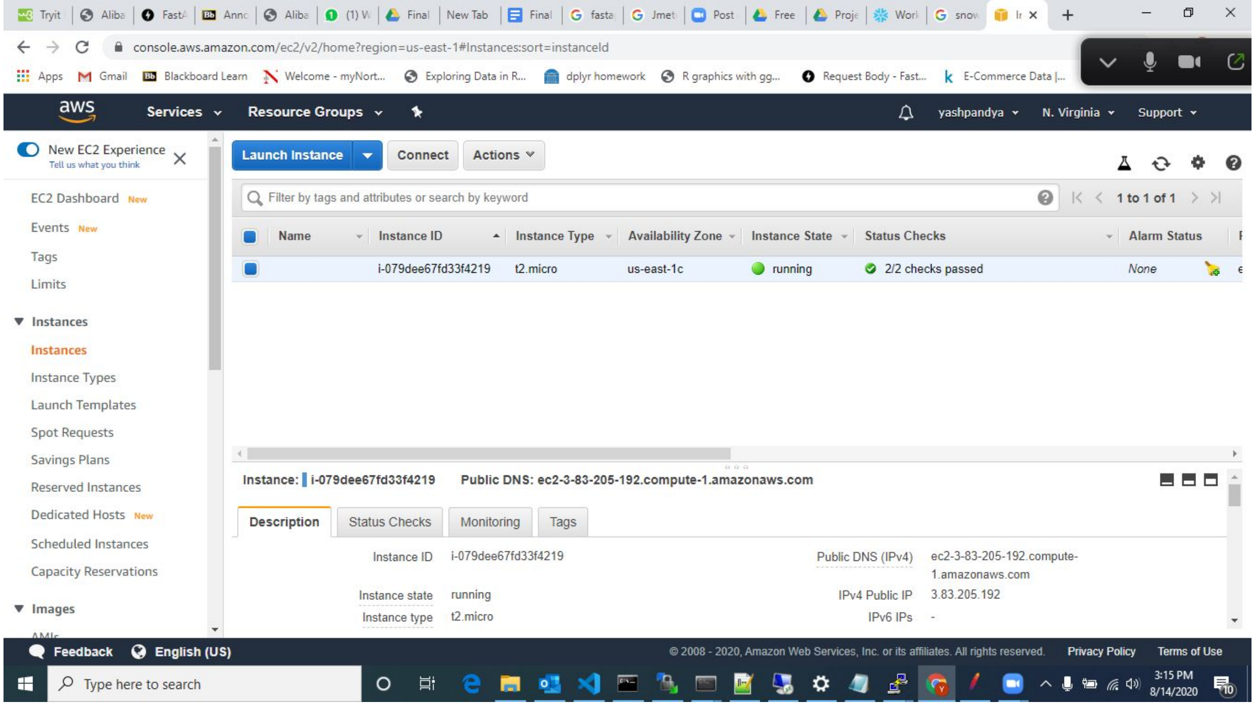
Instance type t2.micro

Public DNS (IPv4) ec2-3-83-205-192.compute-1.amazonaws.com

IPv4 Public IP 3.83.205.192

IPv6 IPs -

2) File Transferred using WinSCP

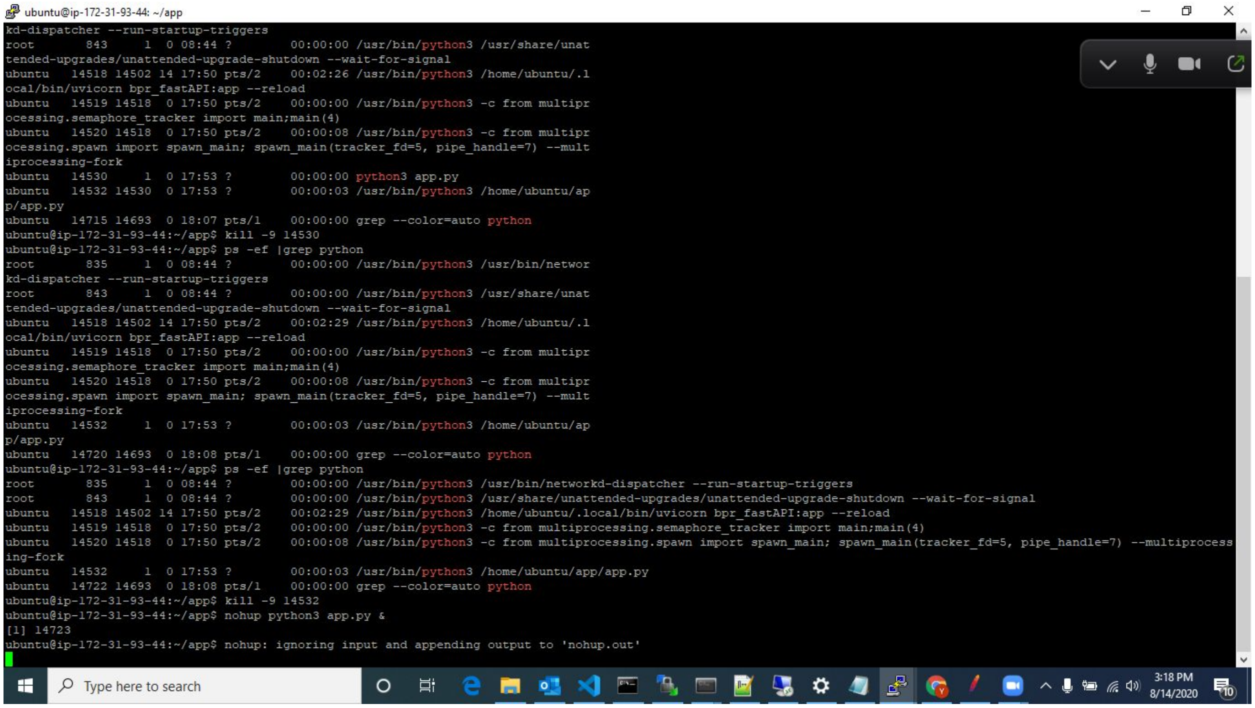


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-079dee67fd33f4219	t2.micro	us-east-1c	● running	✓ 2/2 checks passed	None

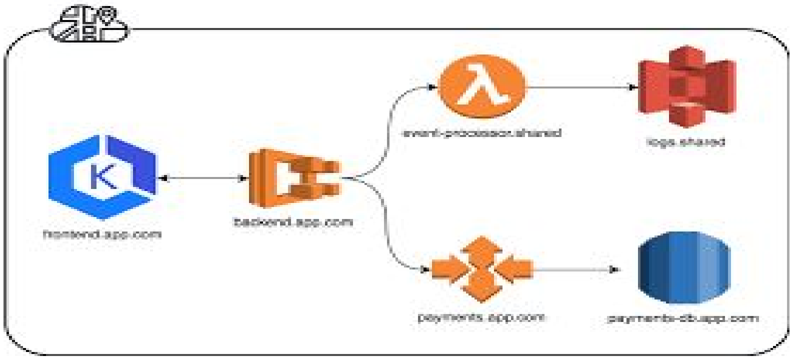
Instance: i-079dee67fd33f4219 Public DNS: ec2-3-83-205-192.compute-1.amazonaws.com

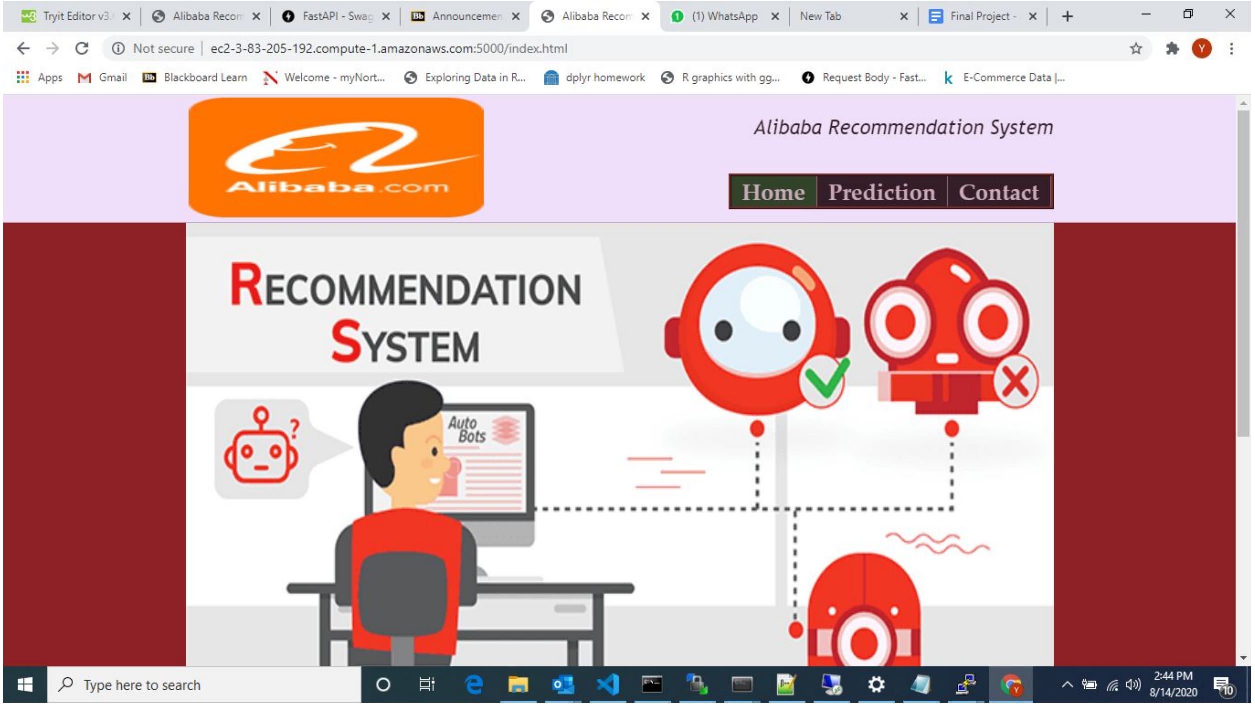
Instance ID	i-079dee67fd33f4219	Public DNS (IPv4)	ec2-3-83-205-192.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.83.205.192
Instance type	t2.micro	IPv6 IPs	-

3) Used putty for deployment



```
kd-dispatcher --run-startup-triggers
root       843     1  0 08:44 ?        00:00:00 /usr/bin/python3 /usr/share/unat
tended-upgrades/unattended-upgrade-shutdown --wait-for-signal
ubuntu   14518 14502 14 17:50 pts/2    00:02:26 /usr/bin/python3 /home/ubuntu/.l
ocal/bin/uvicorn bpr_fastAPI:app --reload
ubuntu   14519 14518  0 17:50 pts/2    00:00:00 /usr/bin/python3 -c from multipr
ocessing.semaphore_tracker import main;main(4)
ubuntu   14520 14518  0 17:50 pts/2    00:00:08 /usr/bin/python3 -c from multipr
ocessing.spawn import spawn_main; spawn_main(tracker_fd=5, pipe_handle=7) --mult
iprocessing-fork
ubuntu   14530     1  0 17:53 ?        00:00:00 python3 app.py
ubuntu   14532 14530  0 17:53 ?        00:00:03 /usr/bin/python3 /home/ubuntu/ap
p/app.py
ubuntu   14715 14693  0 18:07 pts/1    00:00:00 grep --color=auto python
ubuntu@ip-172-31-93-44:~/app$ kill -9 14530
ubuntu@ip-172-31-93-44:~/app$ ps -ef |grep python
root       835     1  0 08:44 ?        00:00:00 /usr/bin/python3 /usr/bin/networ
kd-dispatcher --run-startup-triggers
root       843     1  0 08:44 ?        00:00:00 /usr/bin/python3 /usr/share/unat
tended-upgrades/unattended-upgrade-shutdown --wait-for-signal
ubuntu   14518 14502 14 17:50 pts/2    00:02:29 /usr/bin/python3 /home/ubuntu/.l
ocal/bin/uvicorn bpr_fastAPI:app --reload
ubuntu   14519 14518  0 17:50 pts/2    00:00:00 /usr/bin/python3 -c from multipr
ocessing.semaphore_tracker import main;main(4)
ubuntu   14520 14518  0 17:50 pts/2    00:00:08 /usr/bin/python3 -c from multipr
ocessing.spawn import spawn_main; spawn_main(tracker_fd=5, pipe_handle=7) --mult
iprocessing-fork
ubuntu   14532     1  0 17:53 ?        00:00:03 /usr/bin/python3 /home/ubuntu/ap
p/app.py
ubuntu   14720 14693  0 18:08 pts/1    00:00:00 grep --color=auto python
ubuntu@ip-172-31-93-44:~/app$ ps -ef |grep python
root       835     1  0 08:44 ?        00:00:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
root       843     1  0 08:44 ?        00:00:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
ubuntu   14518 14502 14 17:50 pts/2    00:02:29 /usr/bin/python3 /home/ubuntu/.local/bin/uvicorn bpr_fastAPI:app --reload
ubuntu   14519 14518  0 17:50 pts/2    00:00:00 /usr/bin/python3 -c from multiprocessing.semaphore_tracker import main;main(4)
ubuntu   14520 14518  0 17:50 pts/2    00:00:08 /usr/bin/python3 -c from multiprocessing.spawn import spawn_main; spawn_main(tracker_fd=5, pipe_handle=7) --multiprocess
ing-fork
ubuntu   14532     1  0 17:53 ?        00:00:03 /usr/bin/python3 /home/ubuntu/app/app.py
ubuntu   14722 14693  0 18:08 pts/1    00:00:00 grep --color=auto python
ubuntu@ip-172-31-93-44:~/app$ kill -9 14532
ubuntu@ip-172-31-93-44:~/app$ nohup python3 app.py &
[1] 14723
ubuntu@ip-172-31-93-44:~/app$ nohup: ignoring input and appending output to 'nohup.out'
```


frontend.app.com
backend.app.com
event-processor.shared
logs.shared
payments.app.com
payments-db.app.com


Tryit Editor v3.
Alibaba Recom
FastAPI - Swag
Announcemen
Alibaba Recom
(1) WhatsApp
New Tab
Final Project -
Not secure | ec2-3-83-205-192.compute-1.amazonaws.com:5000/index.html
Apps
Gmail
Blackboard Learn
Welcome - myNort...
Exploring Data in R...
dplyr homework
R graphics with gg...
Request Body - Fast...
E-Commerce Data |...
Alibaba.com
Alibaba Recommendation System
Home
Prediction
Contact
RECOMMENDATION SYSTEM
Auto Bots
Type here to search
2:44 PM
8/14/2020

Conclusion

We were able to implement various tools we have learned in this course to help support Alibaba's Website analyze their sales and increase their profit.

References

1. https://github.com/microsoft/recommenders
2. https://fastapi.tiangolo.com/tutorial/body/
3. https://github.com/microsoft/recommenders/blob/master/examples/00_quick_start/sequential_recsys_amazondataset.ipynb
4. https://github.com/microsoft/recommenders/blob/master/examples/02_model_collaborative_filtering/lightgcn_deep_dive.ipynb
5. https://github.com/microsoft/recommenders/tree/master/tests
6. https://github.com/davidefiocco/streamlit-fastapi-model-serving