Gift This Course

A-Z Machine Learning using Azure Machine Learning (AzureML)

Hands on AzureML: From Azure Machine Learning Introduction to Advance Machine Learning Algorithms. No Coding Required.

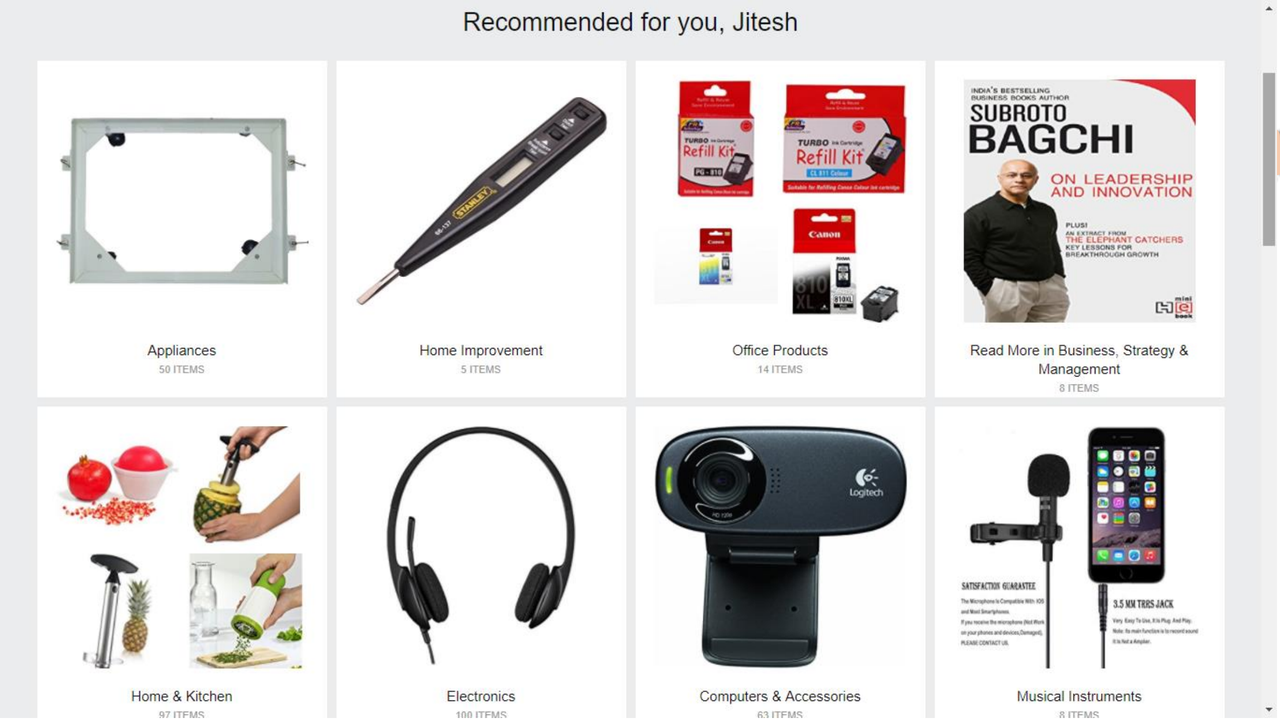
BEST SELLER 4.3 (215 ratings) 1,597 students enrolled

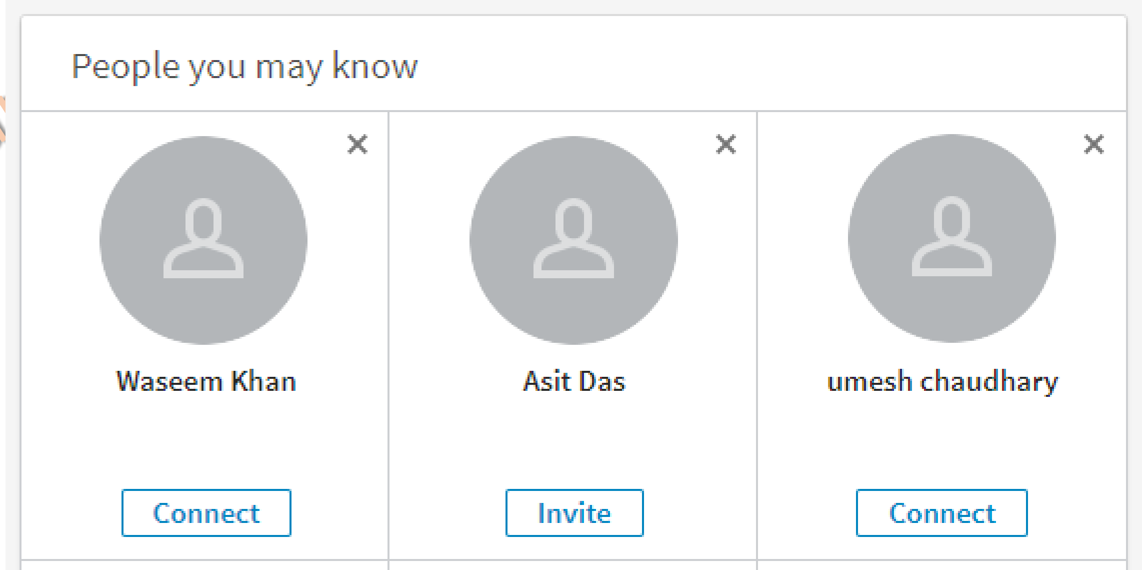
Created by Jitesh Khurkhuriya Last updated 3/2018 English English

AzureML Matchbox Recommender

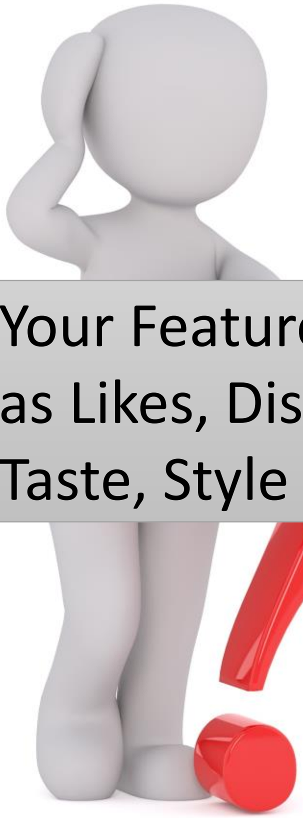
What is a Recommendation System?

“A recommender system or a recommendation system (platform or engine) seeks to predict the "rating" or "preference" that a user would give to an item.”





Everyday Recommendations



Shopping

Restaurant

e

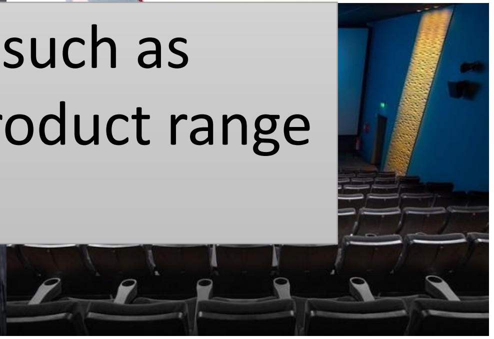
Books

Travel

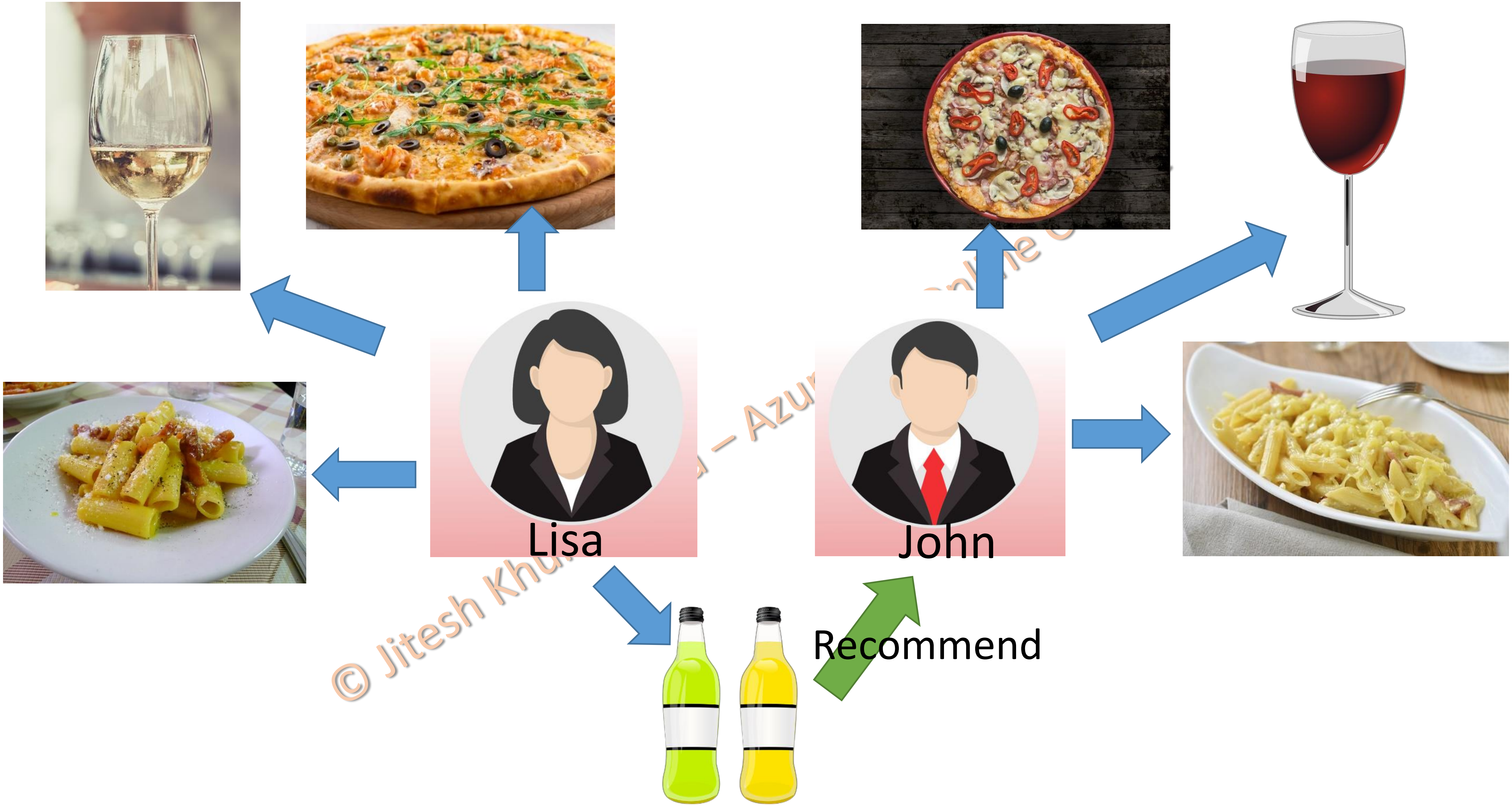
Your Features such as Likes, Dislikes, Taste, Style

History of preferences or behaviour

Friends, Family, Colleagues, Professors



Item Features such as Food, taste, product range etc

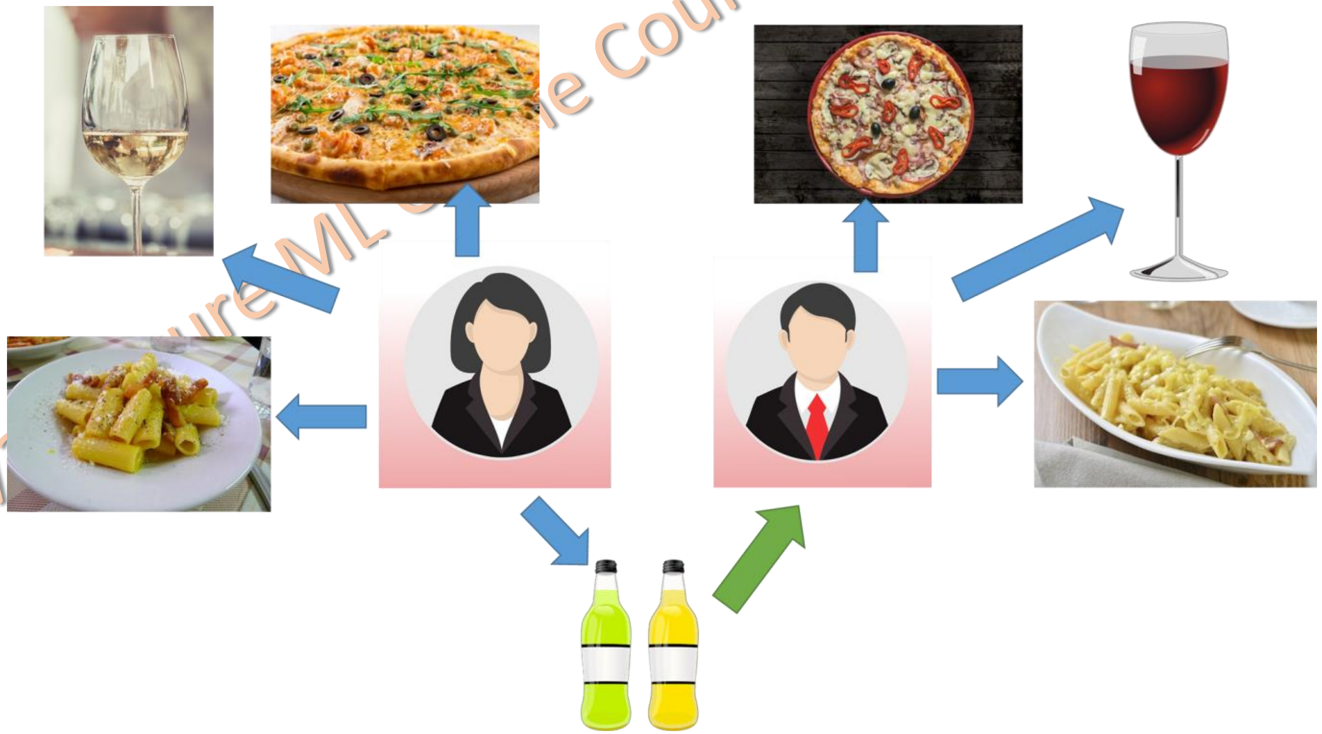

Lisa
John
Recommend

Types of Recommendation Systems

- Collaborative Filtering
- Content Based Filtering
- Hybrid – Combination of the Collaborative and Content Based
- Popularity based – Most bought, most watched, most downloaded, most heard

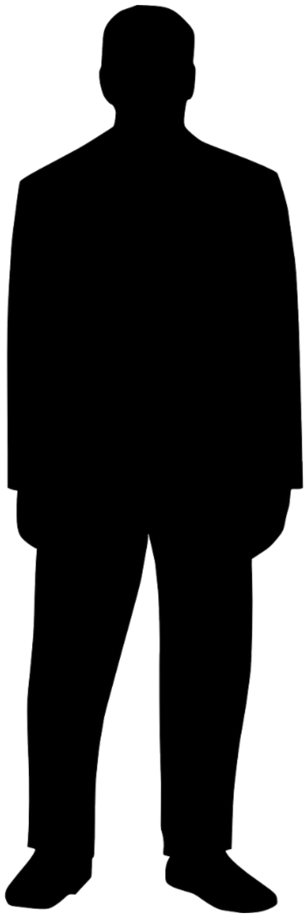
Collaborative Filtering

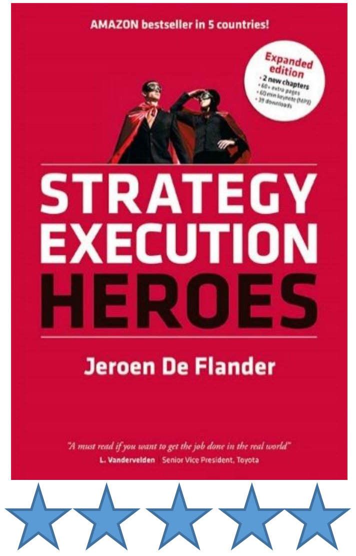
- Analyse User behaviour, Activities and preferences
- Recommend based on similarity to other user
- People who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.

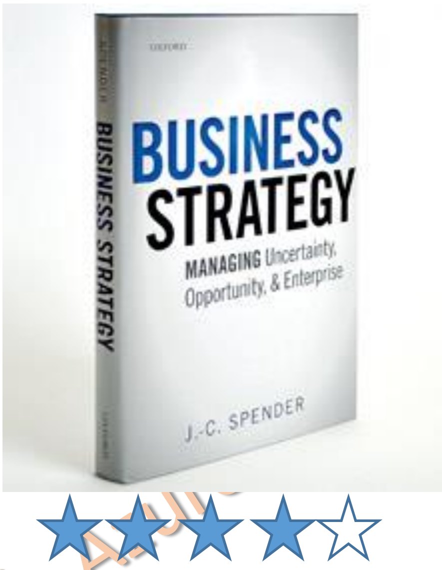


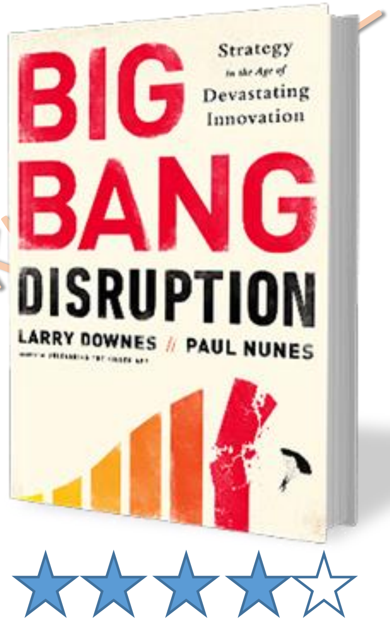
People who buy “x” also buy “y”.

Content Based Filtering

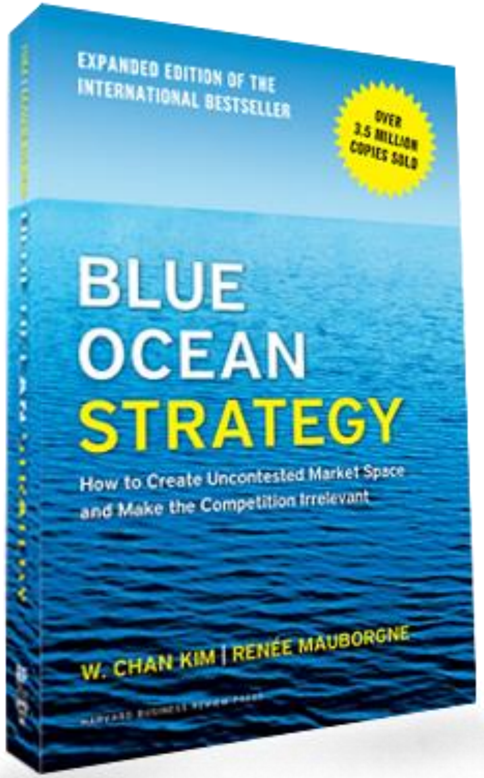








Recommend



Recommend Items with similar content to the items user purchased and liked in the past.

Business Problem

A travel booking website books thousands of hotel rooms every day.

With such a huge growth, the task at hand is to improve the booking per customer by showing the best hotels as per the taste and preferences of the user.

Sample Set

- Hotel 1 – Good Gym, Small Pool
- Hotel 2 – Amazing Gym, No Pool
- Hotel 3 – Great Gym, Small Pool
- Hotel 4 – Basic Gym, Large Pool
- Hotel 5 – No Gym, Large and Beautifully designed pool

- John – Prefers Gym over swimming pool
- Kavin – Likes Gym more than pool
- Bill – Needs a pool with basic Gym
- Frans – A pool is a must compared to Gym

Item – Feature Matrix

Hotel	Gym	Pool
Hotel 1	0.8	0.2
Hotel 2	1	0
Hotel 3	0.9	0.1
Hotel 4	0.1	0.9
Hotel 5	0	1

User – Feature Matrix

User	Gym	Pool
John	0.9	0.1
Kavin	0.8	0.2
Bill	0.3	0.7
Frans	0	1

Creating Feature Vector

Hotel	Gym	Pool
Hotel 1	0.8	0.2
Hotel 2	1	0
Hotel 3	0.9	0.1
Hotel 4	0.1	0.9
Hotel 5	0	1

Hotel	Feature Vector
H1	[0.8, 0.2]
H2	[1, 0]
H3	[0.9, 0.1]
H4	[0.1, 0.9]
H5	[0, 1]

User	Gym	Pool
John	0.9	0.1
Kavin	0.8	0.2
Bill	0.3	0.7
Frans	0	1

User	Feature Vector
U1	[0.9, 0.1]
U2	[0.8, 0.2]
U3	[0.3, 0.7]
U4	[0, 1]

Content Based Recommendation

Hotel	Feature Vector
H1	[0.8, 0.2]
H2	[1, 0]
H3	[0.9, 0.1]
H4	[0.1, 0.9]
H5	[0, 1]

User	Feature Vector
U1	[0.9, 0.1]
U2	[0.8, 0.2]
U3	[0.3, 0.7]
U4	[0, 1]

- Which Hotels should be recommended to John (U1)?
- MAX (Uj * Hi)

MAX (U1*H1, U1*H2, U1*H3, U1*H4, U1*H5)

MAX (0.74, 0.9, 0.82, 0.18, 0.1)

Recommended Hotels in the order of preference

H2, H3, H1

Collaborative Filtering

Workflow of CF, (Wikipedia)

- A user expresses his or her preferences by rating items.
- Ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
- The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
- With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Which hotel to recommend to Kavin?

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Which hotel to recommend to John?

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Which hotel to recommend to Frans?

Recommendation Data Issues

- **Cold Start**

- **Same Scale Judgement**

Cold Start

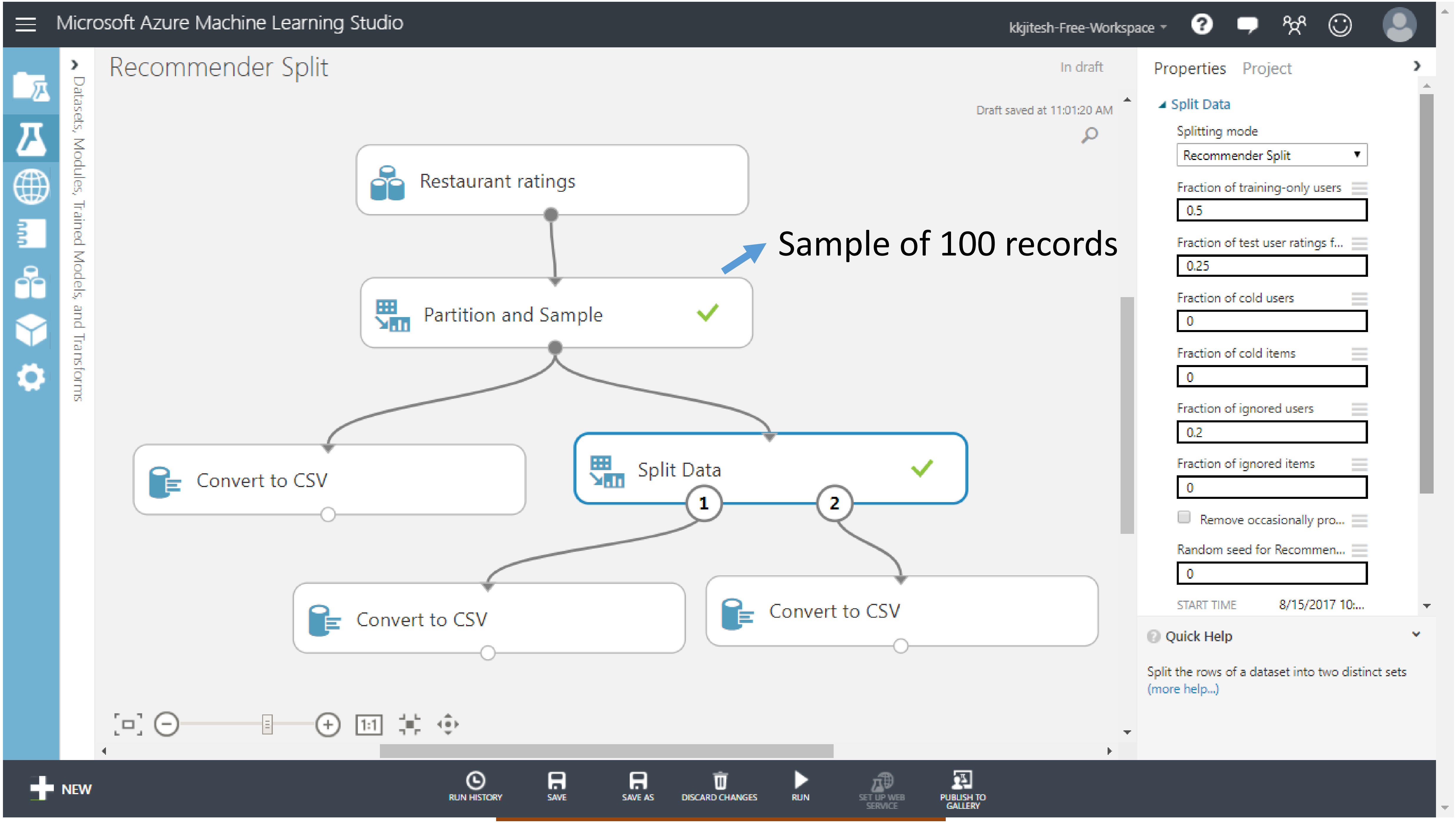
- Content based filtering constructs user profile before system can recommend
- Collaborative Filtering needs like-minded users for recommendations
- Completely new profile or item
- Can be tackled using Hybrid recommendation

Same Scale judgement

- A user rates all the items on a same scale
- An item is rated on the same scale by all users

Recommender Split

- Fraction of training only users – Fraction of users assigned only to the training dataset. The rows would never be used to test the model.
- Fraction of test user ratings for training – Portion of the user ratings that can be used for training.
- Fraction of cold users – Cold users are users that the system has not previously encountered.
- Fraction of cold items – Cold items are items that the system has not previously encountered.
- Fraction of ignored users – Specify the percentage of users that should be ignored.
- Fraction of ignored items – Specify the percentage of items to ignore.
- Remove occasionally produced cold items


Microsoft Azure Machine Learning Studio
kkjitesh-Free-Workspace
Datasets, Modules, Trained Models, and Transforms
Recommender Split
In draft
Draft saved at 11:01:20 AM
Restaurant ratings
Partition and Sample
Sample of 100 records
Convert to CSV
Split Data
1
2
Convert to CSV
Convert to CSV
Properties
Project
Split Data
Splitting mode
Recommender Split
Fraction of training-only users
0.5
Fraction of test user ratings f...
0.25
Fraction of cold users
0
Fraction of cold items
0
Fraction of ignored users
0.2
Fraction of ignored items
0
Remove occasionally pro...
Random seed for Recommen...
0
START TIME
8/15/2017 10:...
Quick Help
Split the rows of a dataset into two distinct sets
(more help...)
NEW
RUN HISTORY
SAVE
SAVE AS
DISCARD CHANGES
RUN
SET UP WEB SERVICE
PUBLISH TO GALLERY

Sample Data Result

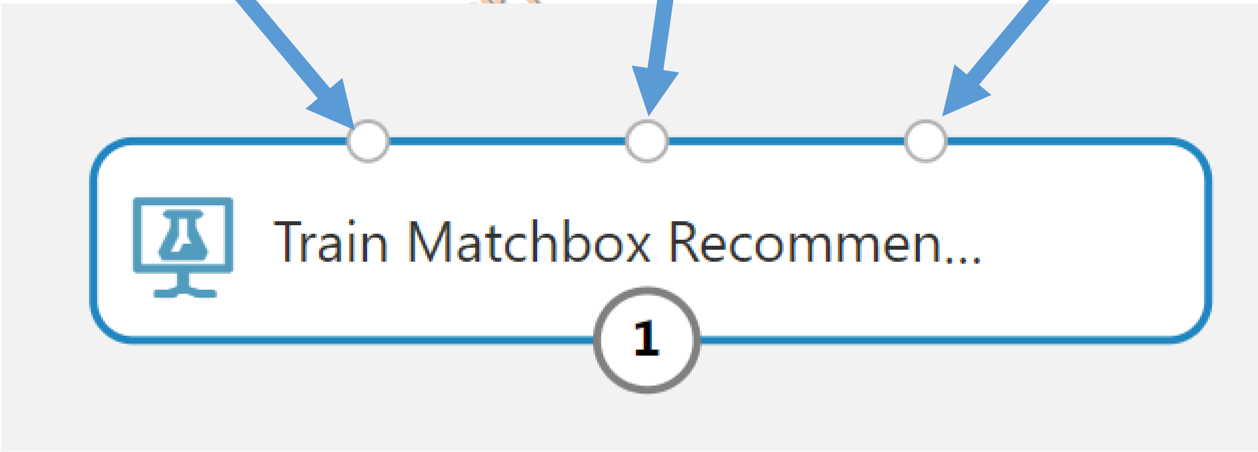
Original (100)			Split 1 (57)			Split 2 (24)		
userID	placeID	rating	userID	placeID	rating	userID	placeID	rating
U1031	132663	0						
U1031	132665	0						
U1031	132668	0						
U1068	132630	1						
U1068	132663	1						
U1068	132732	0						
U1068	132740	0						
U1068	135104	1						
U1093	132767	2						
U1093	134976	2						
U1093	134996	2						
U1093	134999	2						
U1093	135001	2						
U1093	135011	1						
U1093	135013	2						
U1093	135019	2						
U1107	132584	2						
U1107	132660	2						
U1107	132733	2						

What is a Matchbox Recommendation System?

- Recommendation system developed by Microsoft Research
- Based on a the Hybrid Approach of Content and Collaborative filtering
- Uses Train Matchbox Recommender and Score Matchbox Recommender Modules
- Train Matchbox Recommender reads a dataset of **user-item-rating** triples and, optionally, some user and item features.

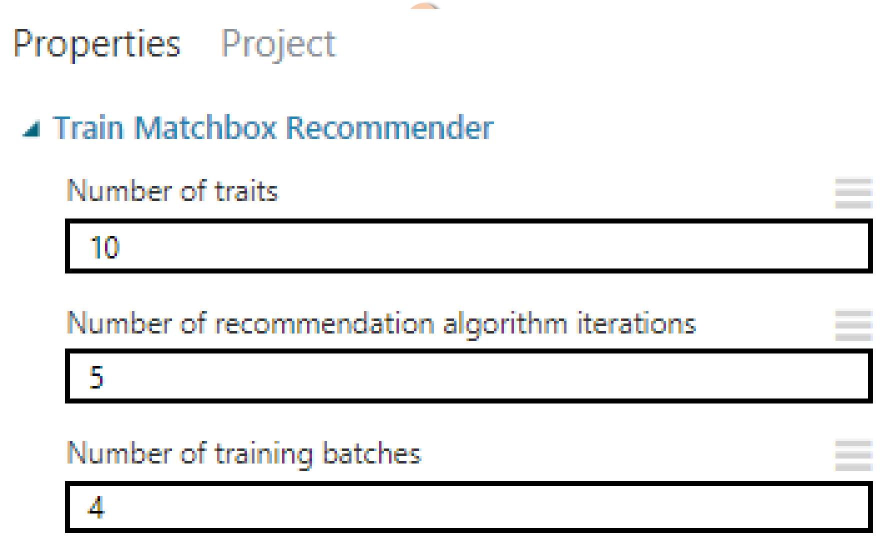
Configuring Train Matchbox Recommender

- User-Item-Rating Data - Prepare the data used for training, containing **user-item-rating** triples
  - First Column – User Identifier
  - Second Column – Item Identifier
  - Third Column – Rating for the user-item pair
- User-Feature – userID and the userfeatures
- Item-Feature – itemID and the Itemfeatures

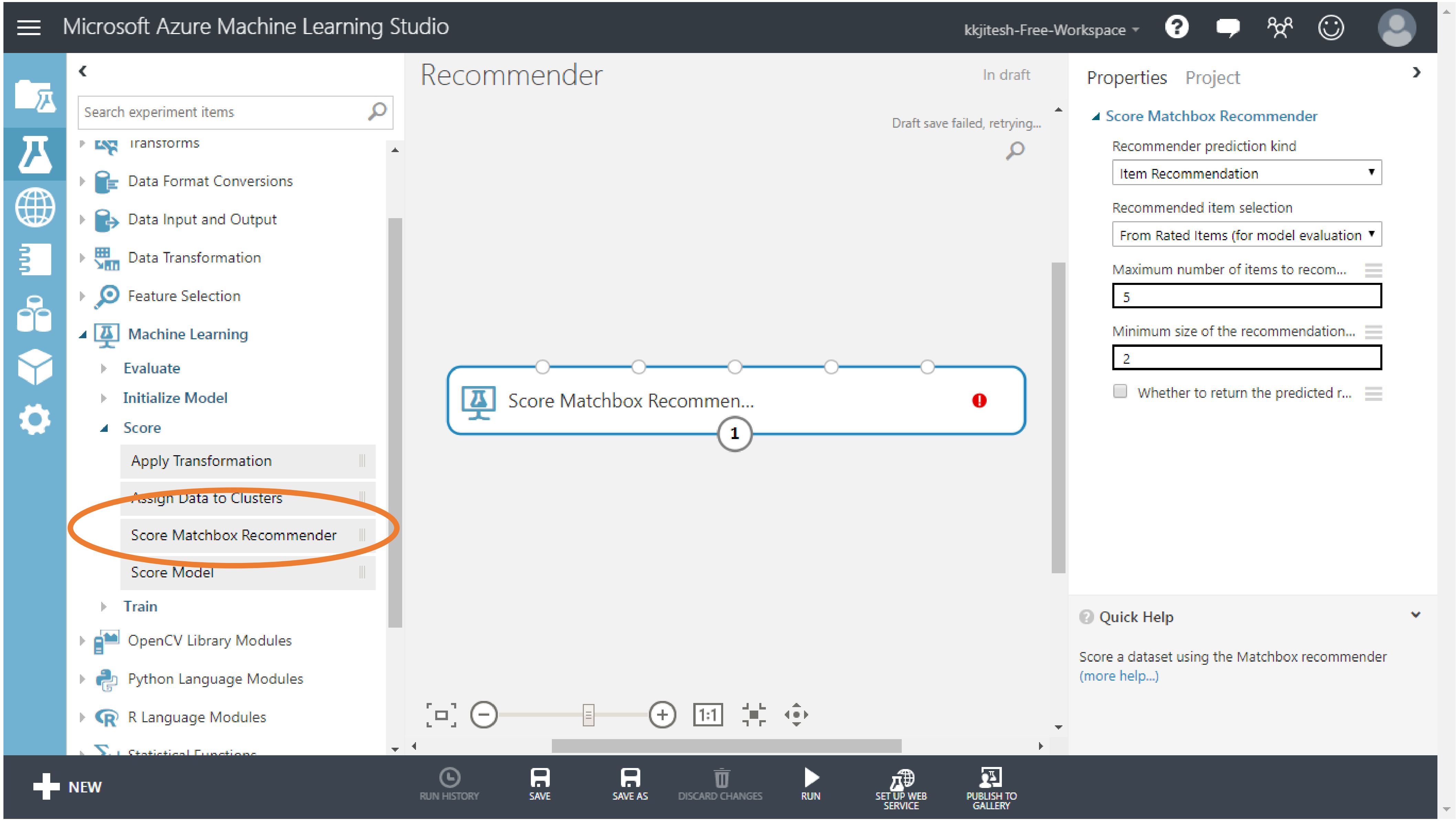


Parameters to Train Matchbox Recommender

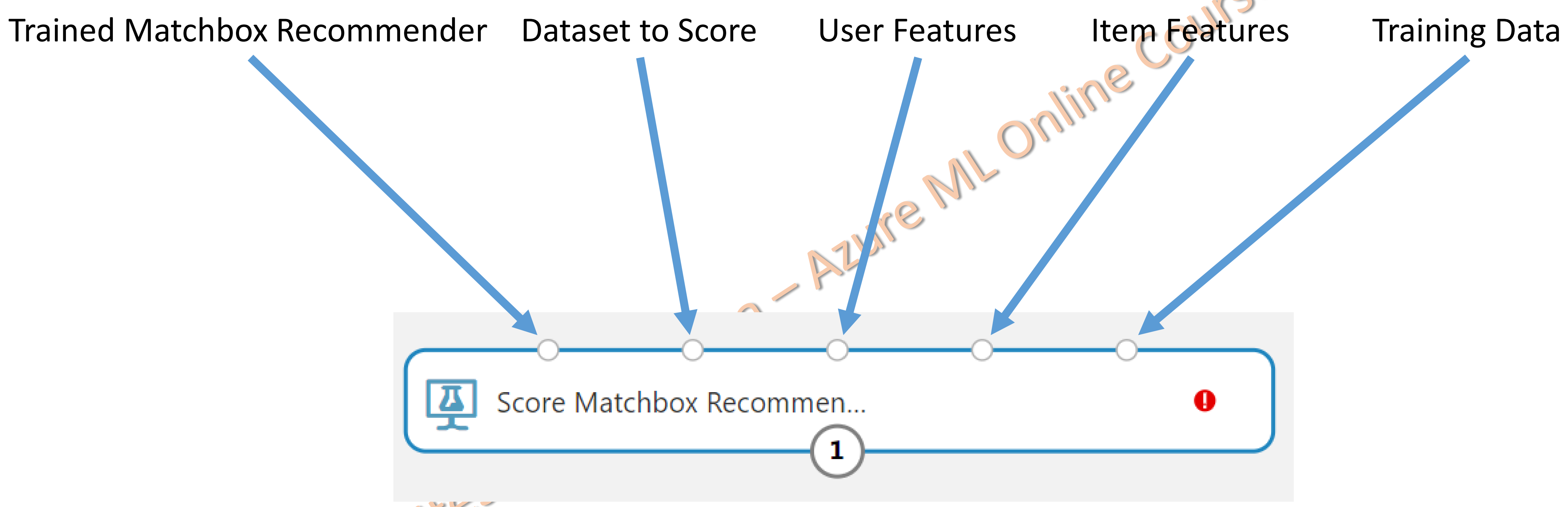
- **Number of Traits**
  - How many traits to learn for each user and item
  - Each feature is associated with a latent “trait” vector
- **Number of recommendation algorithm iterations**
  - how many times the algorithm should process the input data
- **Number of training batches**
  - Number of batches for dividing the data during training



Score Matchbox Recommender



Input to Score Matchbox Recommender



Prediction Types

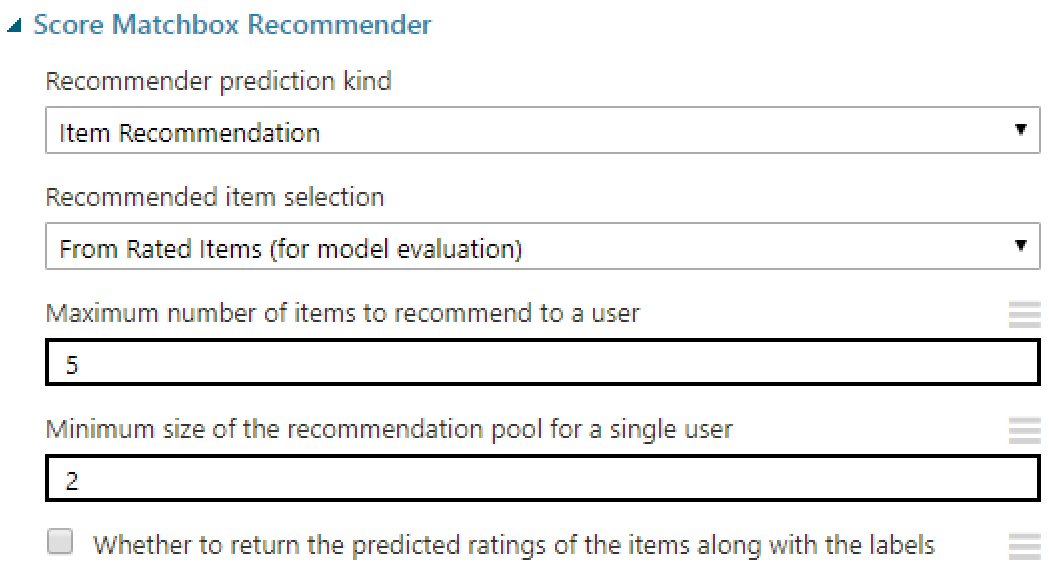
- Predict Ratings
- Item Recommendation
- Related Users
- Related Items

Predict Ratings

- Rating predictions by a user
- Input data must contain User and Item
- Does not require any parameter

Item Recommendation

- Users and Items as input
- Uses its knowledge about existing items and users
- Generates a list of items that will appeal to each user



Related Users and Items

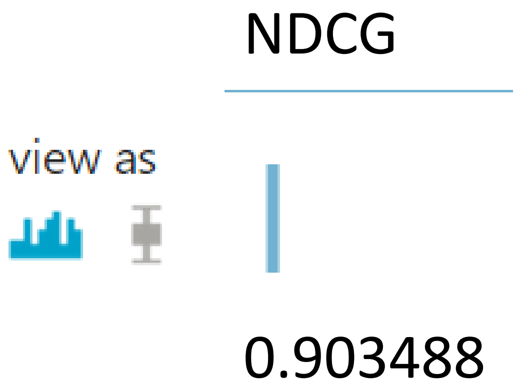
- Can be used for “People like you” predictions

- Generate recommendations for users based on items that have already been rated

Understanding the Recommender Result

- For Item Recommendations

- Normalized Discounted Cumulative Gain (NDCG)



Ranking Quality

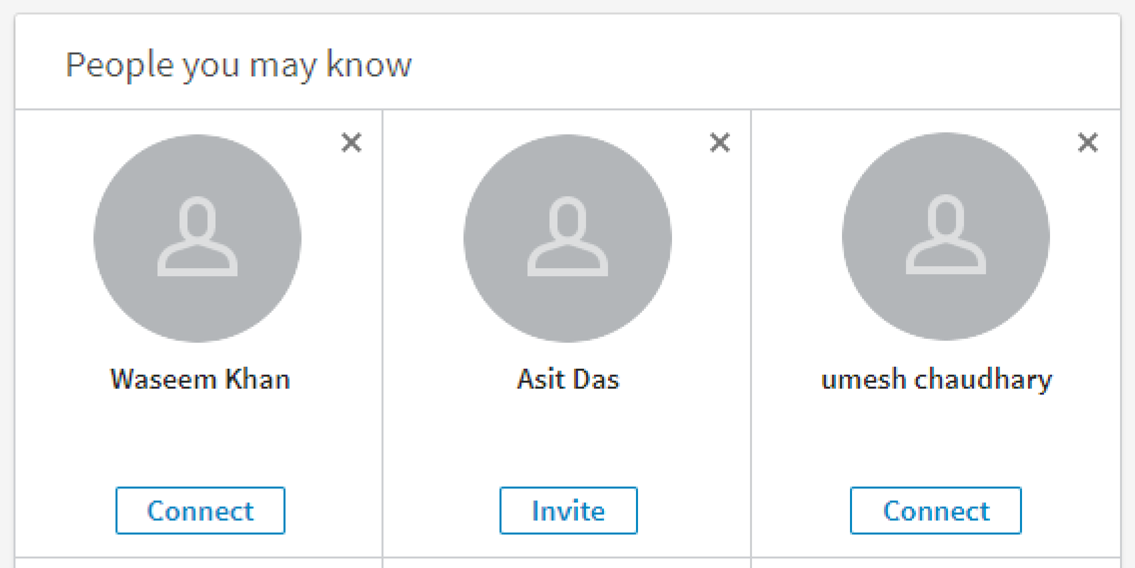
Search Result

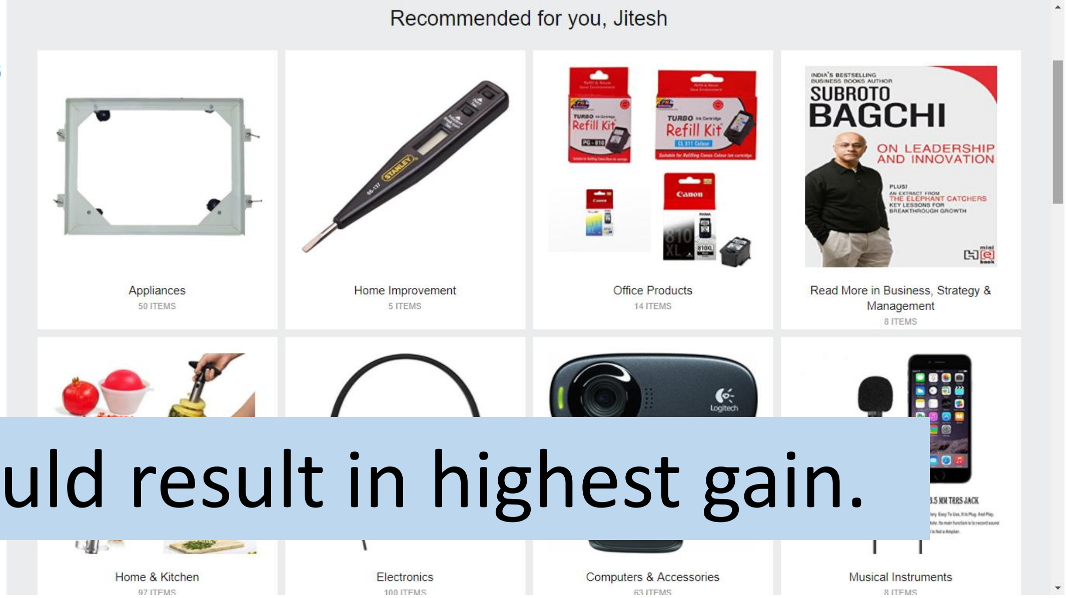
What is machine learning? - Definition from WhatIs.com
whatis.techtarget.com › Topics › AppDev › Programming ▼
Jun 24, 2017 - **Machine learning** is the field of study that gives computers the ability to learn without being explicitly programmed.

The 10 Algorithms Machine Learning Engineers Need to Know
www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html ▼
Aug 8, 2016 - It is no doubt that the sub-field of **machine learning** / artificial intelligence has increasingly gained more popularity in the past couple of years.

Machine Learning | SAP - SAP.com
https://www.sap.com/india/trends/machine-learning.html ▼
Discover how AI, **machine learning**, and deep learning are powering a new breed of software that uses Big Data to drive radical changes to business.

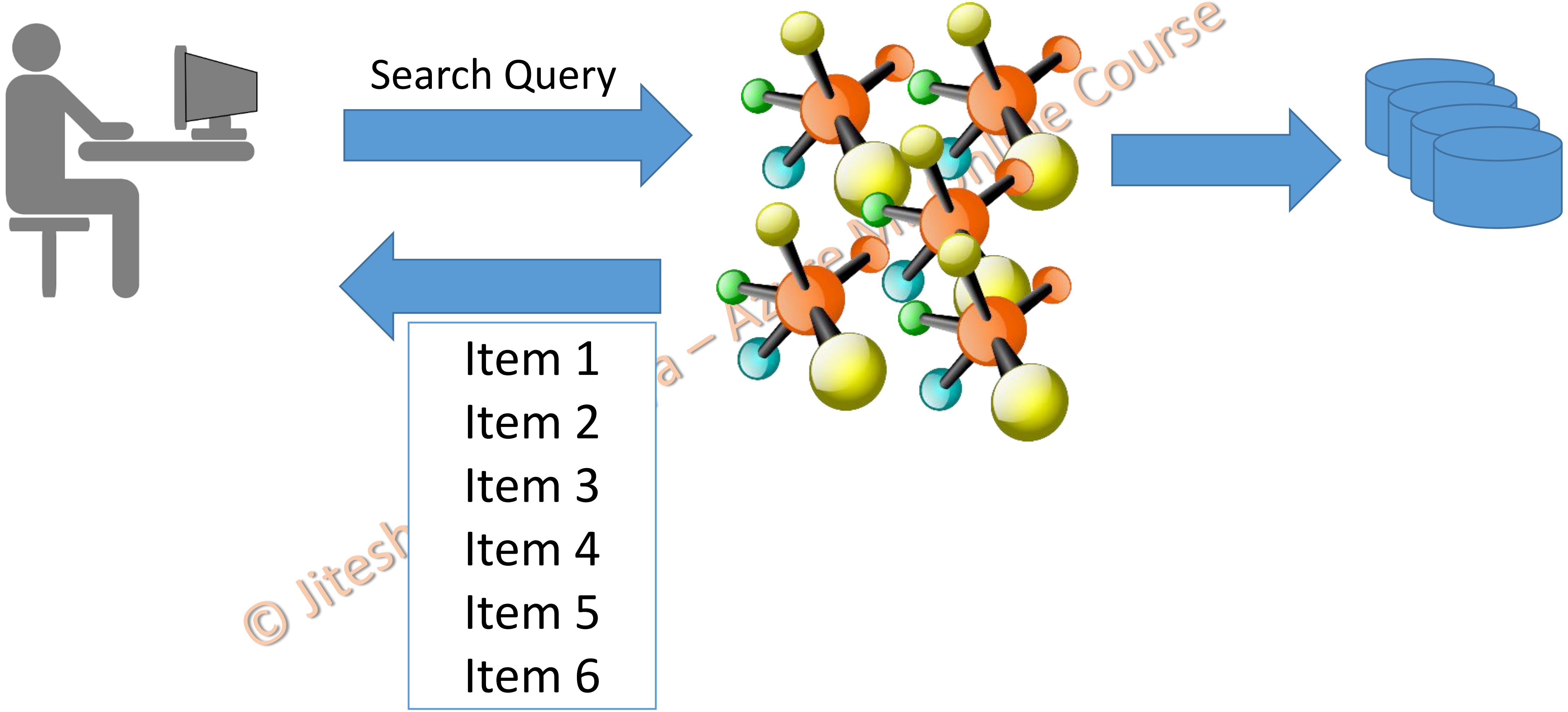
Machine Learning | edX
https://www.edx.org/course/machine-learning-columbiax-csmm-102x-0 ▼
Master the essentials of **machine learning** and algorithms to help improve learning from data without human intervention.





The highest ranked item should result in highest gain.

Ranking Quality



Best ranked items as per the algorithm

Discounted Cumulative Gain

Ranked items by the algorithms → I1, I2, I3, I4, I5, I6

Gain perceived by the user → 4, 3, 4, 0, 1, 2

$$CG = 4 + 3 + 4 + 0 + 1 + 2 = 14$$

i	rel_i	$\log_2 (i+1)$	$\dfrac{rel_i}{\log_2 (i+1)}$
1	4	1	4.00
2	3	1.585	1.89
3	4	2	2
4	0	2.322	0
5	1	2.585	0.39
6	2	2.81	0.71
		DCG	**8.99**

Ideal Discounted Cumulative Gain

Ideal Ranking by algorithm → I1, I3, I2, I6, I5, I4

Ideal Gain perceived by the user → 4, 4, 3, 2, 1, 0

$$CG = 4 + 4 + 3 + 2 + 1 + 0 = 14$$

i	rel_i	$\log_2(i+1)$	$\dfrac{rel_i}{\log_2(i+1)}$
1	4	1	4.00
2	4	1.585	2.52
3	3	2	1.5
4	2	2.322	0.86
5	1	2.585	0.38
6	0	2.81	0
		IDCG	**9.27**

Normalised DCG

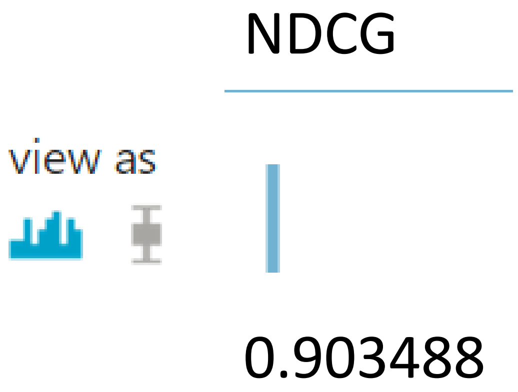
DCG = 8.99 IDCG = 9.27

$$NDCG = \frac{DCG}{IDCG} = \frac{8.99}{9.27} = 0.9697$$

Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than non-relevant documents

Understanding the Recommender Result

- Normalized Discounted Cumulative Gain (NDCG)



Thank You and Have a Great Time..!!