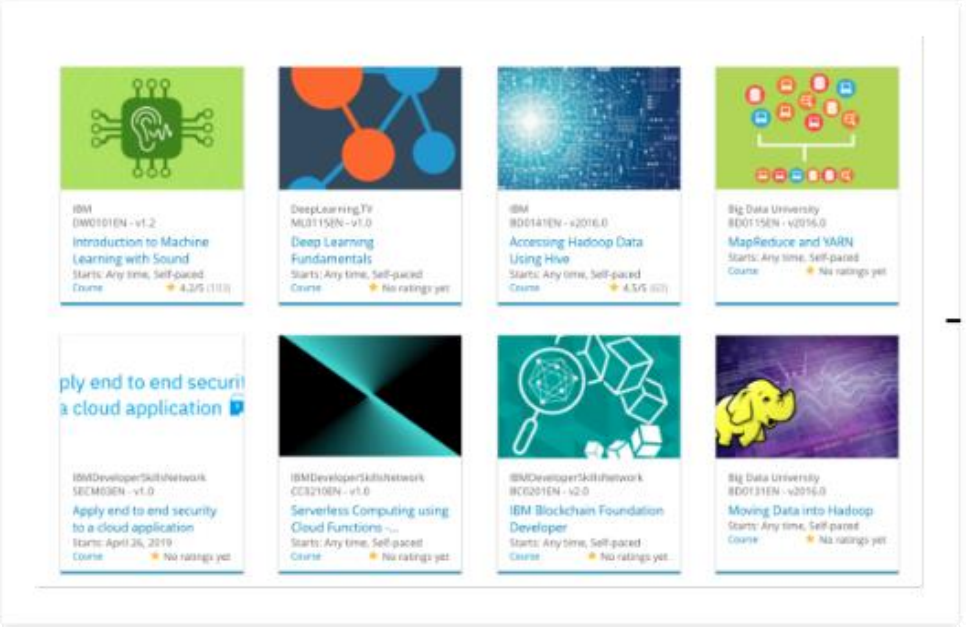
Build a Personalized Online Course Recommender System with Machine Learning

Soumyadeep Khandual

March 20, 2023



Outline

Introduction and Background

Exploratory Data Analysis

Content-based Recommender System using Unsupervised Learning

Collaborative-filtering based Recommender System using Supervised learning

Conclusion

Appendix

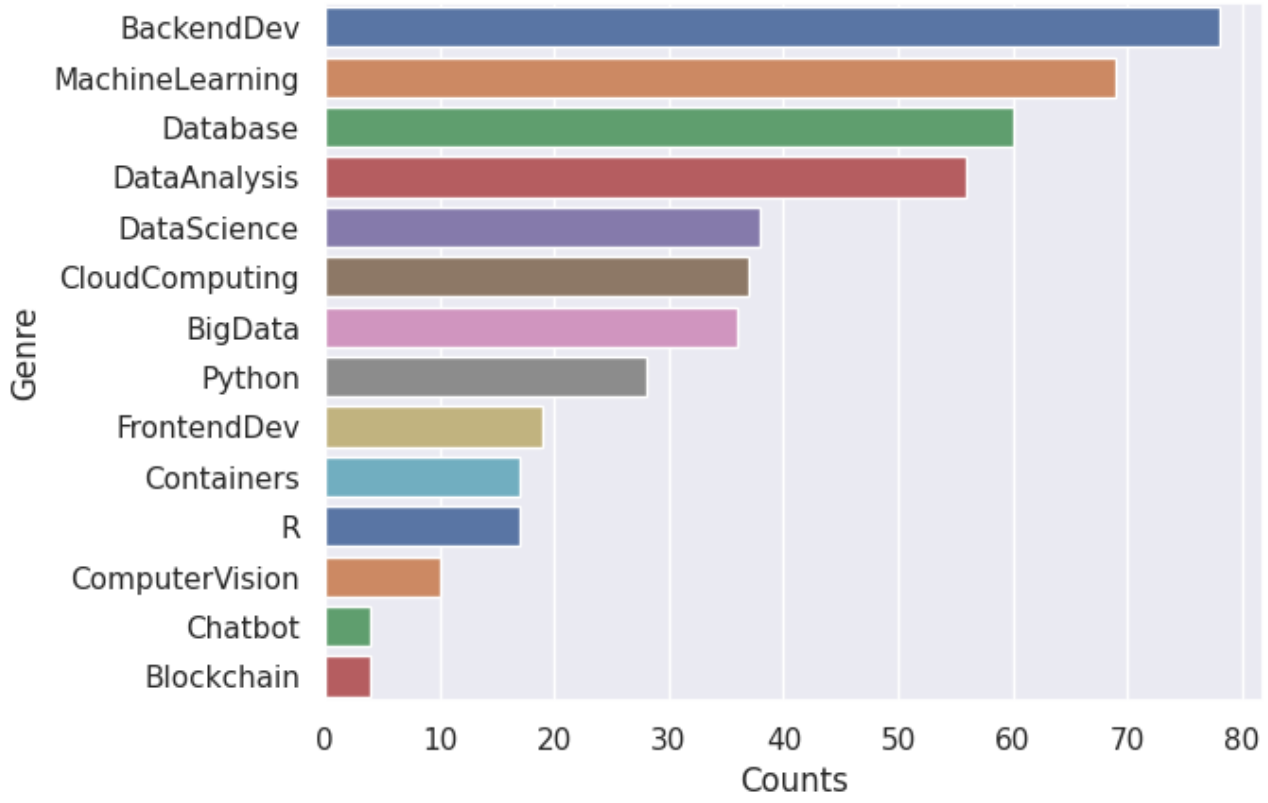
Introduction

- We aim to create a course recommender system to recommend users of courses based on various supervised and unsupervised machine learning models.
- Recommender systems works on two hypothesis:
    - **Content Based:** Similar products, services or content features will be rated similarly by the user.
    - **Collaborative filtering Based:** Users with similar interests will rate products similarly.

Exploratory Data Analysis

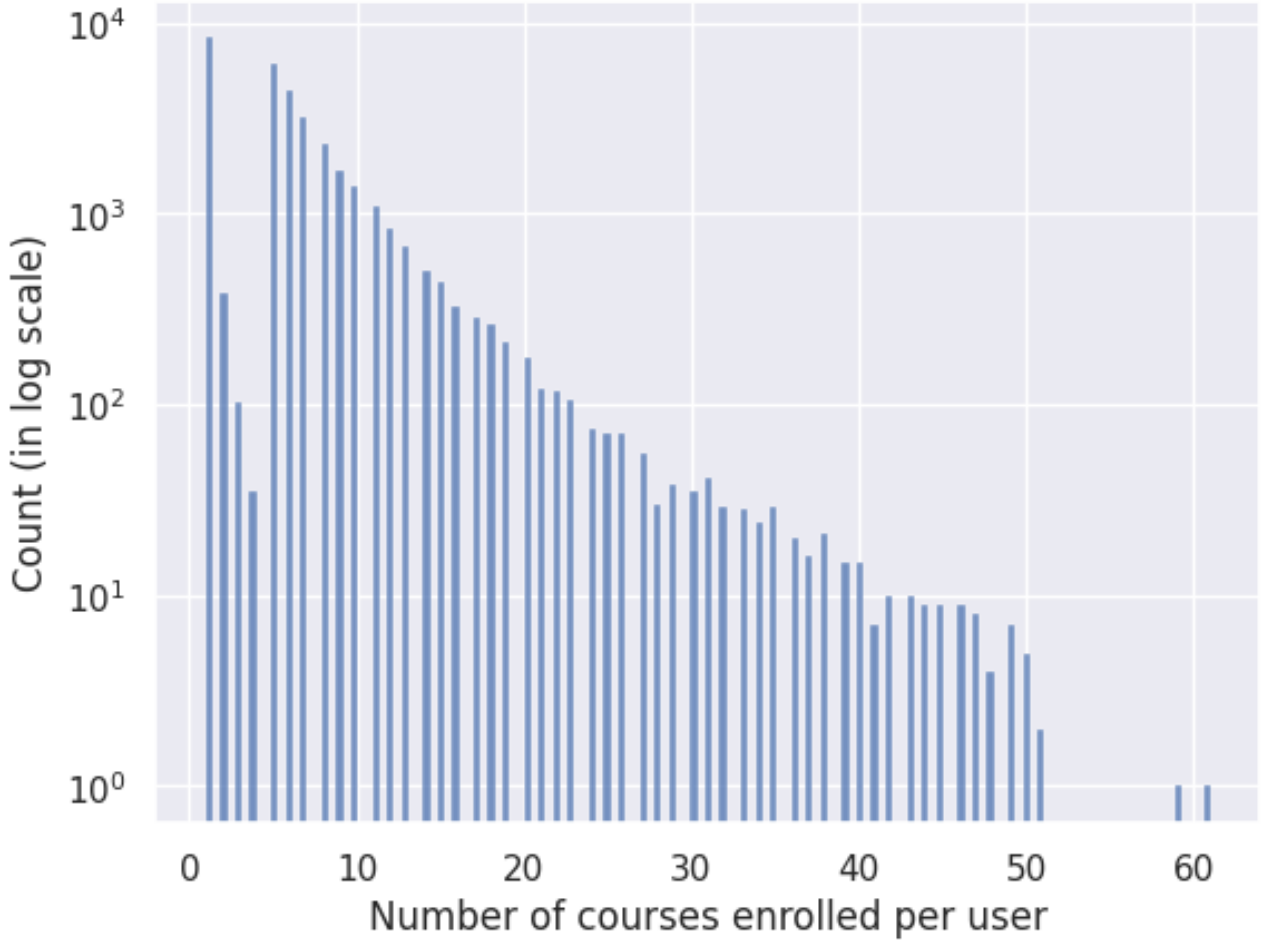
Course counts per genre

- The **top 3** genres are:
  - Backed Dev
  - Machine Learning
  - Database
- **Average** count per genre is 33.8
- **Median** count per genre is 32



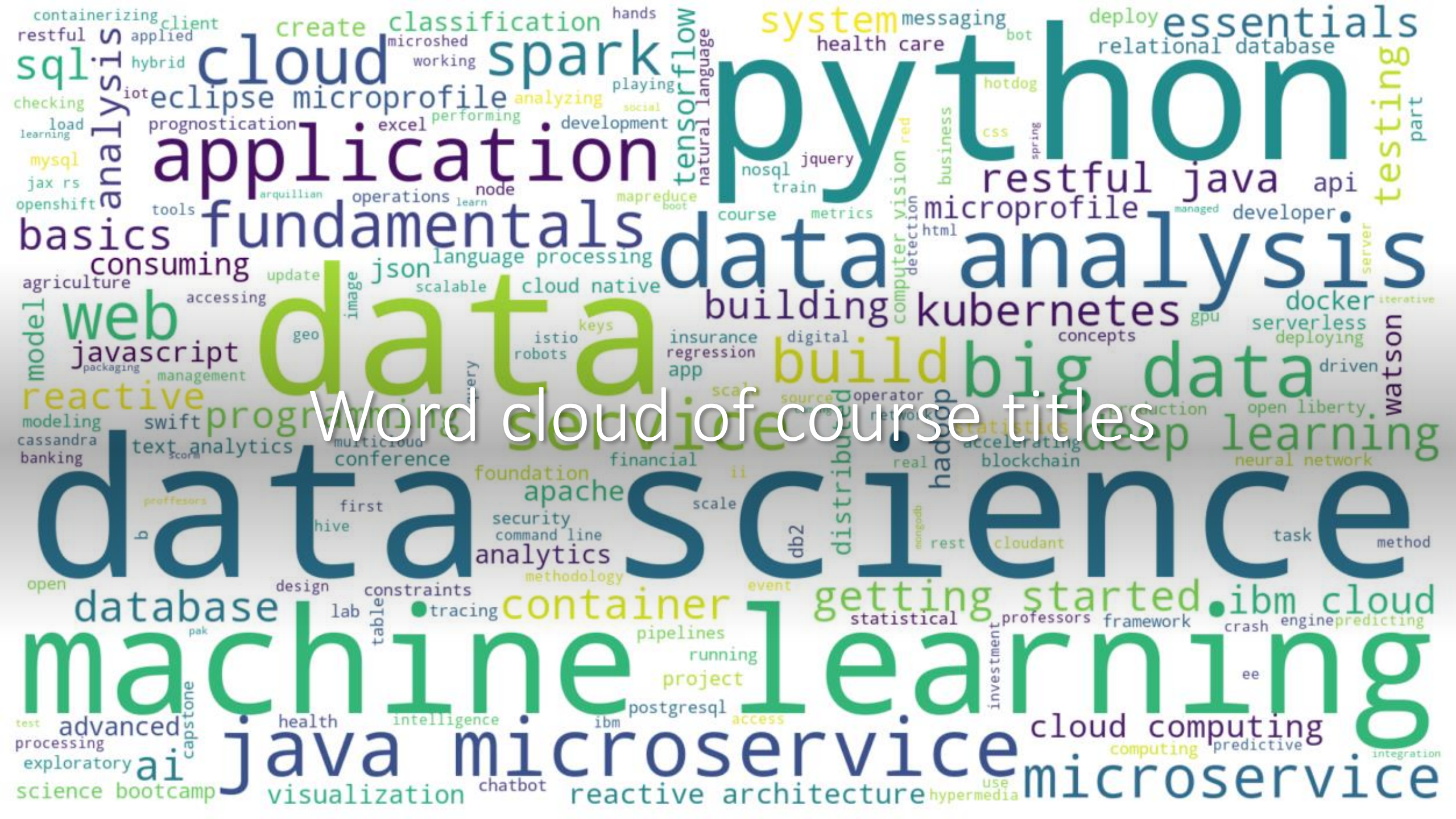
Course enrollment distribution

- The **top 3** number of courses per user are:
  - 1 course (8320)
  - 5 courses (6179)
  - 6 courses (4455)
- **Average** number of courses enrolled per users is 6.88
- **Median** number of courses enrolled per users is 6



20 most popular courses

COURSE_ID	TITLE	Ratings
PY0101EN	python for data science	14936
DS0101EN	introduction to data science	14477
BD0101EN	big data 101	13291
BD0111EN	hadoop 101	10599
DA0101EN	data analysis with python	8303
DS0103EN	data science methodology	7719
ML0101ENv3	machine learning with python	7644
BD0211EN	spark fundamentals i	7551
DS0105EN	data science hands on with open source tools	7199
BC0101EN	blockchain essentials	6719
DV0101EN	data visualization with python	6709
ML0115EN	deep learning 101	6323
CB0103EN	build your own chatbot	5512
RP0101EN	r for data science	5237
ST0101EN	statistics 101	5015
CC0101EN	introduction to cloud	4983
CO0101EN	docker essentials a developer introduction	4480
DB0101EN	sql and relational databases 101	3697
BD0115EN	mapreduce and yarn	3670
DS0301EN	data privacy fundamentals	3624


Word cloud of course titles
containerizing
restful
client
applied
create
classification
hands
system
messaging
deploy
essentials
sql
hybrid
cloud
microshed
working
spark
health care
bot
relational database
analysis
iot
eclipse microprofile
playing
tensorflow
natural language
python
checking
analyzing
social
hotdog
load
learning
prognostication
excel
performing
development
testing
part
application
mysql
jax rs
nosql
jquery
train
business
css
spring
restful java
api
openshift
tools
arquillian
operations
node
learn
mapreduce
boot
course
metrics
computer vision
detection
red
microprofile
developer
basics
fundamentals
data analysis
consuming
language processing
managed
agriculture
update
json
scalable
cloud native
image
accessing
model
web
building
kubernetes
docker
iterative
gpu
serverless
geo
keys
istio
insurance
digital
concepts
deploying
javascript
robots
regression
app
build
big data
driven
packaging
management
reactive
swift
operator
open liberty
modeling
cassandra
text analytics
multicloud
conference
financial
blockchain
deep learning
banking
scorm
foundation
neural network
apache
proffesors
first
scale
security
hive
command line
hadoop
distributed
db2
mongodb
rest
cloudant
task
method
analytics
open
design
constraints
methodology
event
database
lab
tracing
container
getting started
ibm cloud
table
pak
pipelines
statistical
professors
framework
crash
engine
predicting
machine learning
running
project
investment
ee
capstone
test
health
intelligence
ibm
postgresql
access
advanced
cloud computing
processing
computing
predictive
integration
exploratory
ai
java microservice
microservice
science bootcamp
visualization
chatbot
reactive architecture
hypermedia
use

Content based Recommender System using Unsupervised Learning

Flowchart of content-based recommender system using user profile and course genres

01

Extract user profiles vectors (u_i)

02

Extract course genre vectors (v_j)

03

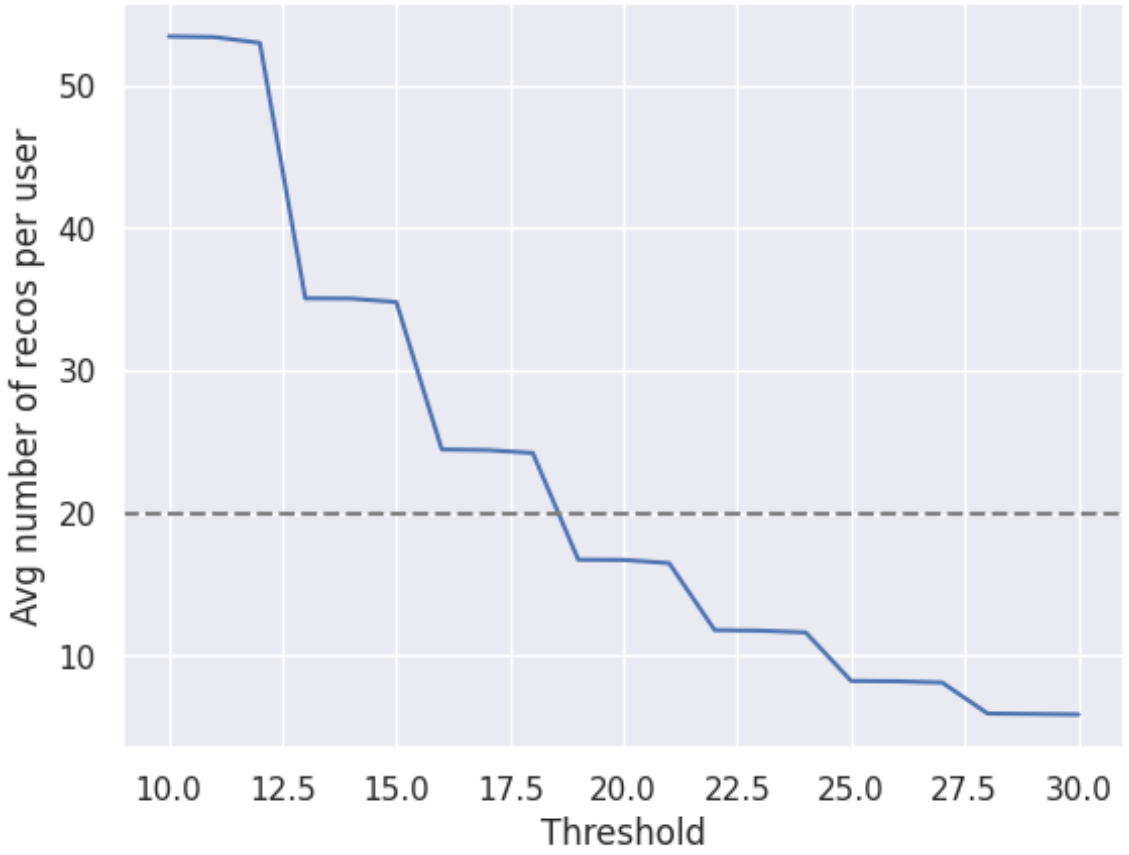
Dot product (u_i . v_j) to get similarity score

04

Recommend course (v_j) if score greater than threshold and not already enrolled by user.

Evaluation results of user profile-based recommender system

Average number of recommendations vs similarity score threshold



Top-10 commonly recommended courses (threshold = 16)

COURSE_ID	COUNT
TA0106EN	367
excourse21	364
excourse22	364
ML0122EN	361
RP0105EN	360
excourse73	360
excourse72	360
TMP0105EN	356
excourse31	354
SC0103EN	329

Flowchart of content-based recommender system using course similarity

01

Extract Bag of Words (BoW) features for each course.

02

Generate course similarity matrix based on cosine distance.

03

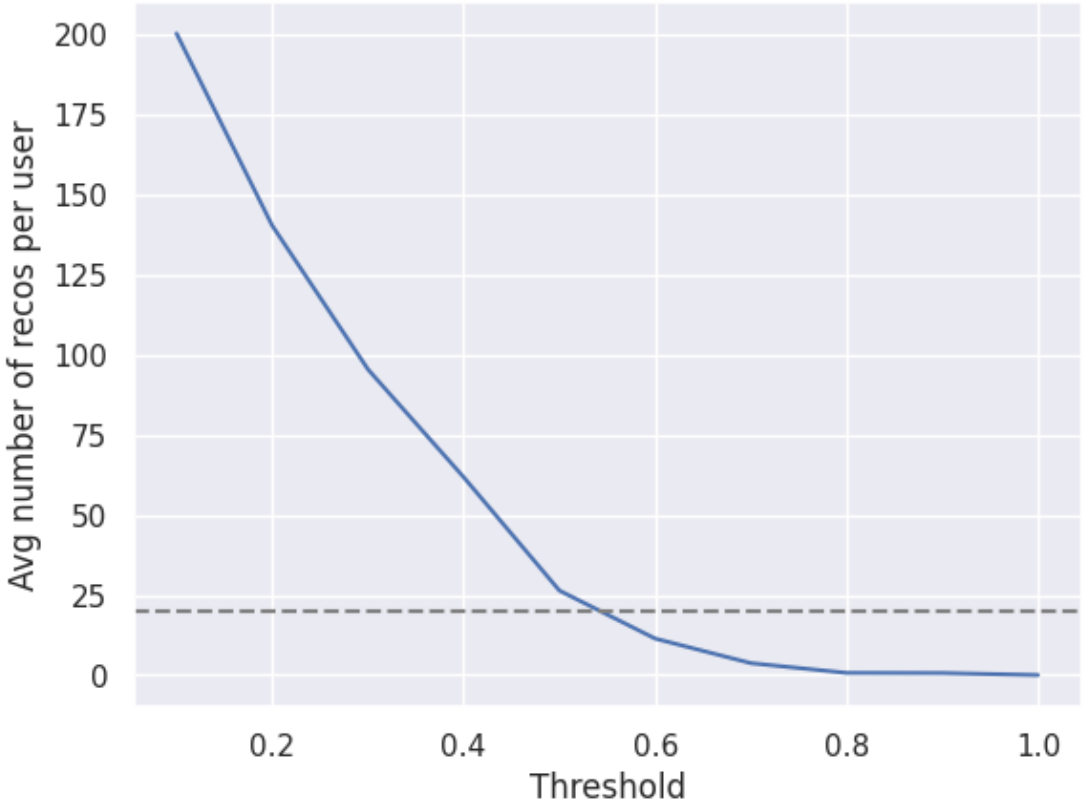
For every enrolled course we find the similarity with unenrolled course.

04

Recommend the unenrolled courses that are greater than threshold.

Evaluation results of course similarity based recommender system

Average number of recommendations vs similarity score threshold



Top-10 commonly recommended courses (threshold = 0.5)

COURSE_ID	COUNT
excourse68	887
excourse67	854
excourse32	828
excourse23	779
excourse36	779
TMP107	773
excourse74	728
DS0110EN	726
excourse65	715
excourse09	715

Flowchart of clustering-based recommender system

01

Extract user profile vectors (u_i)

02

Perform KMeans clustering on (u_i) vectors to cluster users based on profile.

03

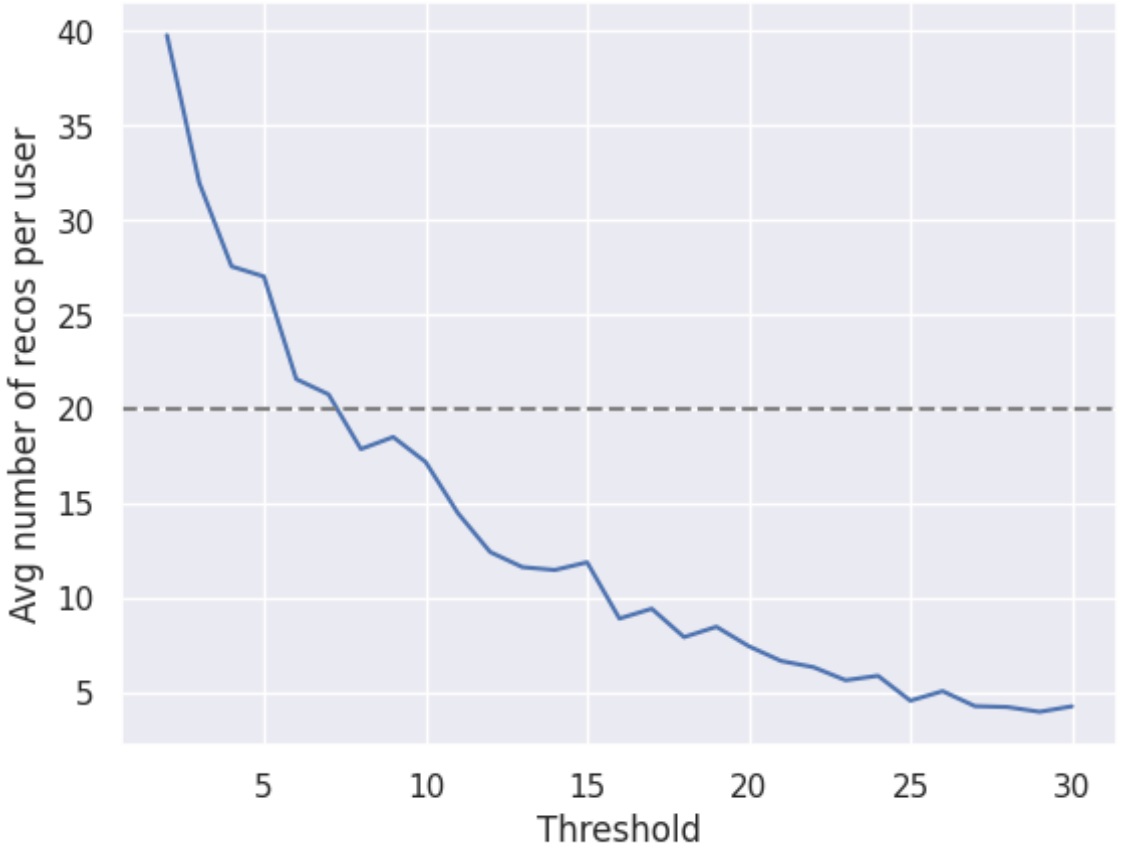
Find all the enrolled course frequencies for each cluster to find popular courses per cluster.

04

Recommend **the** unenrolled **courses from the popular course lists** that have frequencies greater than threshold.

Evaluation results of clustering-based recommender system

Average number of recommendations vs similarity score threshold



Top-10 commonly recommended courses (threshold = 8)

COURSE_ID	COUNT
ST0101EN	697
ML0115EN	678
DS0103EN	657
DB0101EN	655
BD0211EN	647
RP0101EN	635
DA0101EN	626
BD0111EN	561
CO0101EN	556
CC0101EN	550

Collaborative filtering Recommender System using Supervised Learning

Flowchart of KNN based recommender system

01

Extract User-Item interaction matrix and use rows (user based) for collaborative filtering

02

Calculate the similarities between row vectors in the interaction matrix.

03

Based on the similarity measurements, find the k nearest neighbor as the similar users.

04

Scikit-surprise library was used to fit the user vectors and predict for test users.

Flowchart of NMF based recommender system

01

Decompose **the original** sparse user-item **interaction matrix into two dense smaller user and item matrices** using NMF.

02

Estimate **the unknown ratings via the dot product of specific row in user matrix and specific column in item matrix**

03

Find the RMSE for the entire dataset and adjust hyperparameter (NMF hidden dimension) accordingly.

04

Scikit-surprise library was used to fit NMF using trainset and predict using testset.

Flowchart of Neural Network Embedding based recommender system

01

Create neural network with two embedding layers, one for one-hot user vectors and other for one-hot item vectors.

02

The embedding layer outputs two embedding vectors. Dot product the user and item embedding vector to output a rating estimation.

03

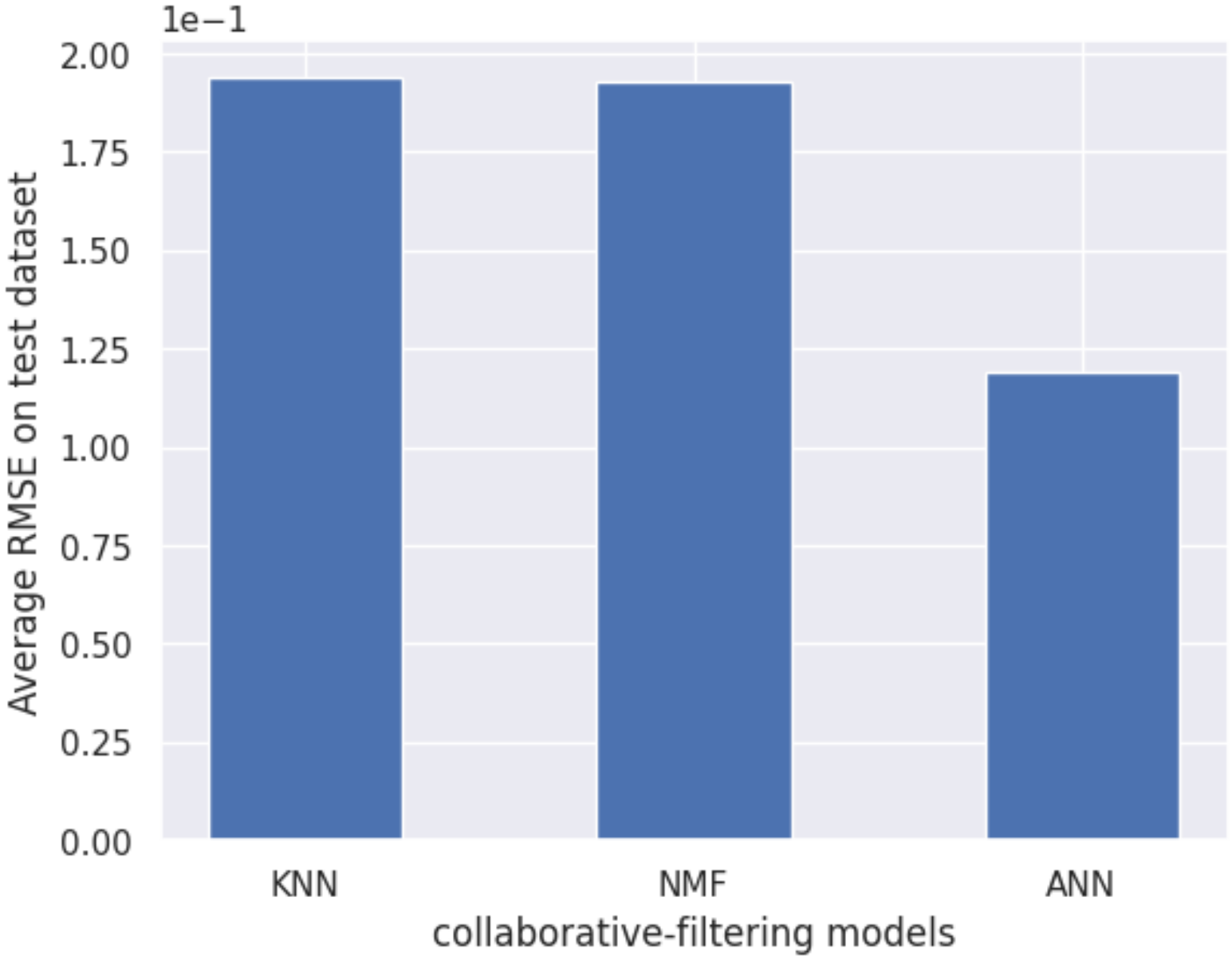
Train the neural network to optimize the weights and bias of embedding layer.

04

Tensorflow Keras library was used to fit the ANN using trainset and predict using testset.

Compare the performance of collaborative filtering models

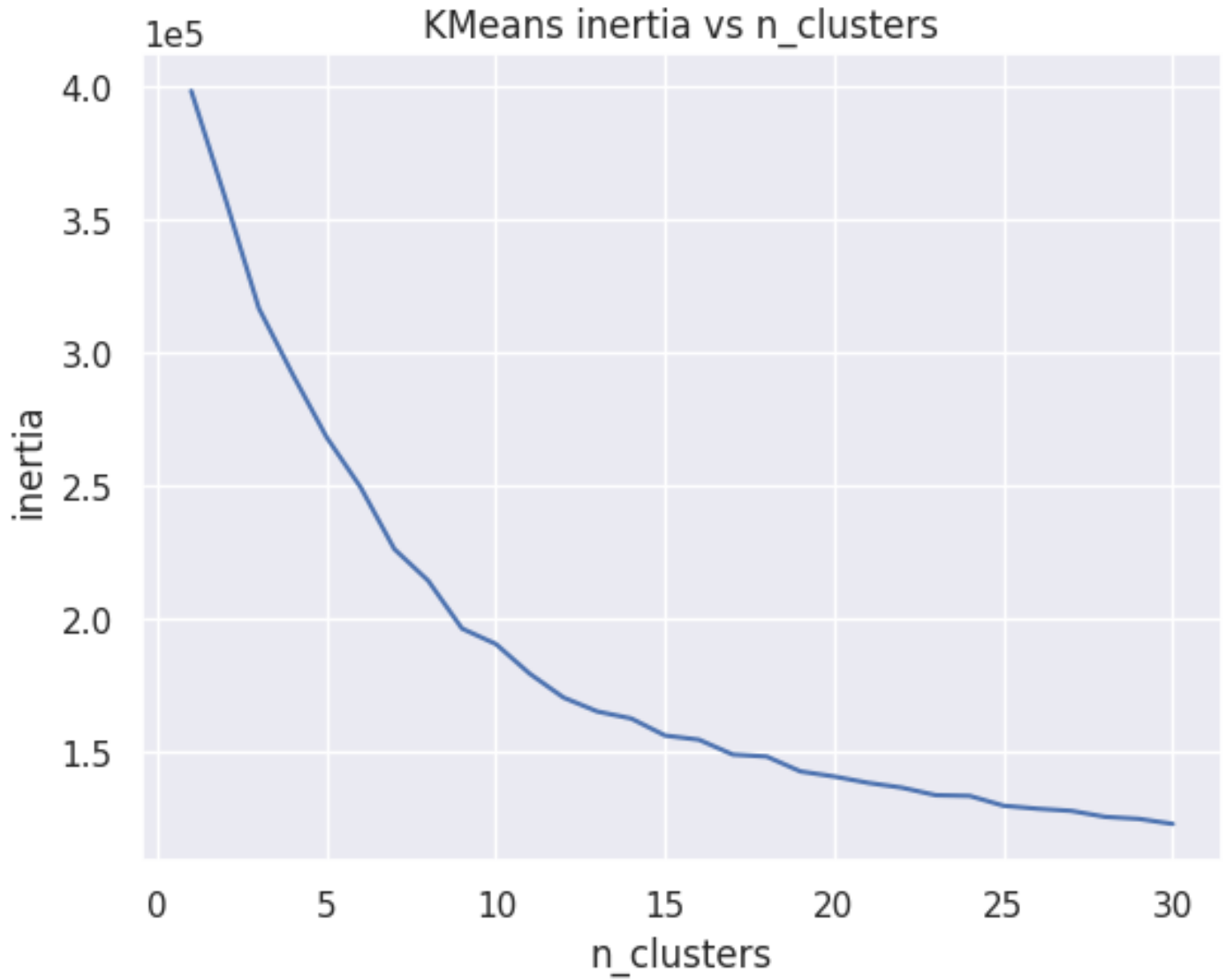
- The best performing model was found to be the Artificial Neural Network.



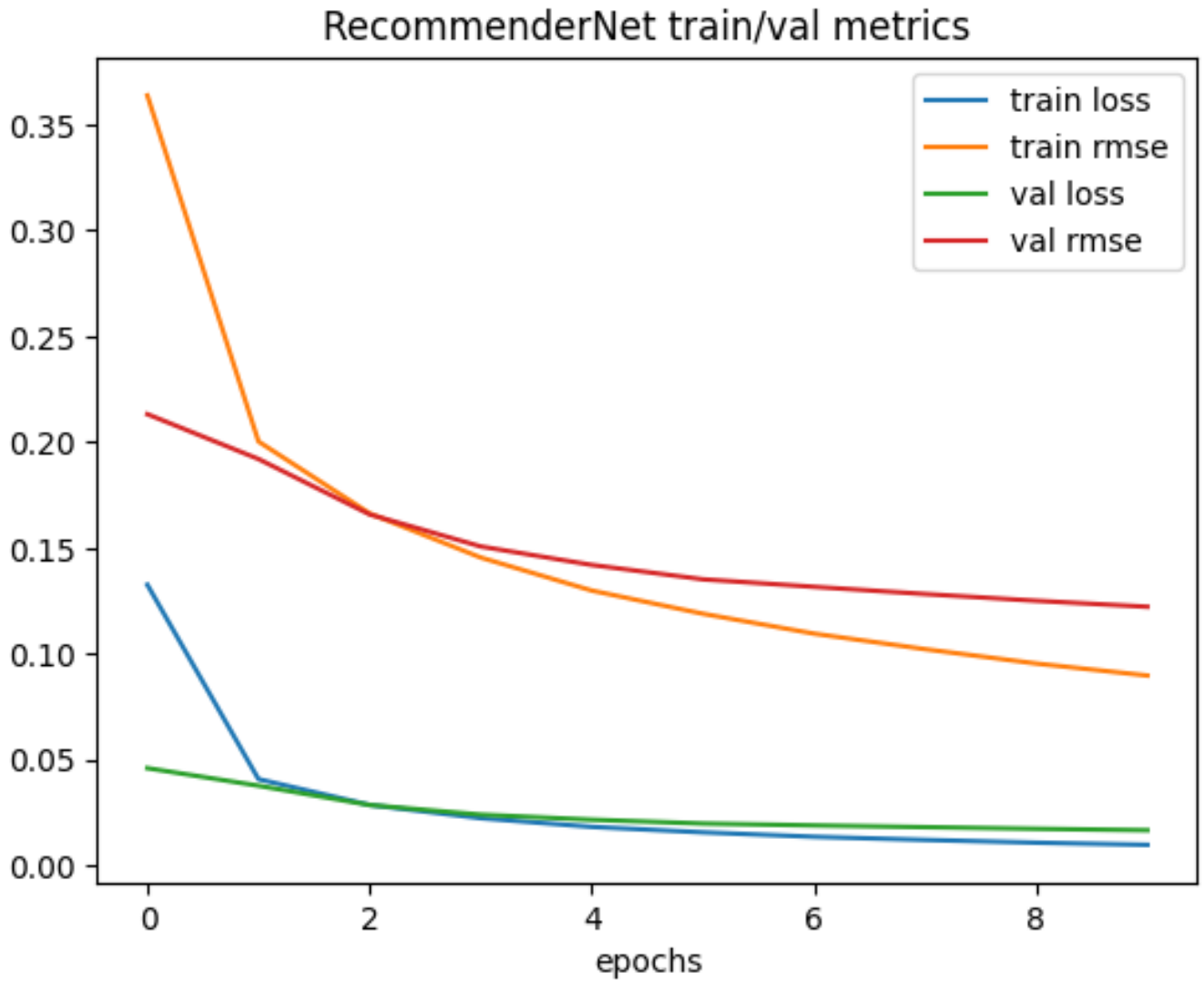
Conclusions

- We performed exploratory data analysis on the user item interaction datasets.
- We performed content based recommendation systems using unsupervised models
- We finally use supervised learning methods to perform collaborative filtering based recommender systems.
- For the content based systems, a good method to adjust optimize similarity threshold was to look at the average number of recommendations per user parameter. A average of around 20 recommendations per user was preferred.
- Of the collaborative filter based systems, ANN particularly performed the best.

Appendix I


KMeans inertia vs n_clusters
1e5
4.0
3.5
3.0
2.5
2.0
1.5
inertia
0
5
10
15
20
25
30
n_clusters

Appendix II


RecommenderNet train/val metrics
train loss
train rmse
val loss
val rmse
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
epochs