Course Recommender System

Aarti Veernala

May 15, 2024

OUTLINE

- Executive Summary
  - Objective
  - Observations
- Introduction
- Upshot
- Methodology
- Project Workflow
  - **EDA**
  - **Content Based Filtering**
  - **User Based Collaborative Filtering**
- Results & Summary
- Conclusion
- Insights

EXECUTIVE SUMMARY

What? The Objective

This **capstone project** was undertaken as the final step towards obtaining the IBM Machine Learning Certification. The primary objective of this project was to use various **machine learning (ML) methods to build a recommender system** that can **recommend unseen courses** of interest to the **prospective users** .

I further extended the scope of this project by extracting the **Top 10 Recommended Courses** for **both** filtering methods (content and user based). The top 10 recommended courses across 6 different models were compared.

The interesting insight from these top 10 recommendations across models is that our recommender system is model dependent

EXECUTIVE SUMMARY

What? The Objective

How? The Method

The recommender system was implemented based on two independent methods, namely, Content/Item based Filtering and User Based Collaborative Filtering. 8 different models were implemented to predict the rating for the unseen courses for each user and course recommendations were made based on these learned ratings.

Of the 8 different models, 3 models were based of content based filtering and the remaining 5 were based on user based collaborative filtering method. 6 out of these 8 models used machine learning techniques to make predictions on course ratings. The machine learning techniques were unsupervised K-Means clustering, K-Nearest Neighbors (KNN) classification, Non Matrix Factorizaton (NMF), Neural Network (NN) for latent space embedding, Classificatoin models like (Logistic regression, Random Forest, Bagging) and Regression Models (like linear regression and Ridge regression).

Programming Packags used : sklearn, tensorflow, keras, surprise

Plotting packages used : Matplotlib, PyPlots, Seaborn

EXECUTIVE SUMMARY

Why? The Reason

Recommender systems (RS) play a very impactful role in our everyday lives. Netflix, YouTube etc use these recommender systems to bring to us (the users) relevant video content that we might like. RSs can also be used to suggest restaurants to meet and eat, merchendise to shop, music to listen to and as in our case, courses that one can study and learn.

In today's world of endless choices and varieties, RSs prove to be extremely useful and effective in bring to the users new and unseen content, while still 'bearing in mind' the users' likes and dislikes. This saves the users the frustration of having to deal with content that they dislike or are uninterested in. RS systems, therefore, also go a long way in enhancing user satisfaction and experience.

RS also saves the users time by filtering through large amounts of data in a comparatively shorter span of time. RSs not only suggests content based on the user' own interests but it can also take into account the what is trending among other users of similar interests.

Executive Summary

The Observations

The Observations

The top 10 recommended courses across all users was found to be deeply influenced by the method implemented for recommendation. More specifically, there was a **significant difference in the top 10 recommend courses** between the all the **various models implemented.**

Linear Regression machine learning model was found to be ineffective for course recommendations. This is in fact, not surprising, because the course recommendations was based on predicting a rating for each unseen course. The target variable for the various ML models is the **course rating scale** which is **discrete!** Regression models like linear regression, Ridge etc perform well on predictive models that have a continuously varying target variable, UNLIKE the one we have at hand.

The top 10 course recommendations from each of the models, content based and user based collaborative filtering, was found to be model dependent!

Introduction

Overview

As mentioned in the executive summary, the goal of this capstone project is to build a recommender system that can:

- Predict course ratings for the courses
- Evaluate the prediction accuracy via metrics such as root mean square error (rmse)
- Make course recommendations from a list of UNseen courses to each user.
- Compile a list of top 10 most popular recommended courses across all users

Datasets

- Course description dataset comprising of Course ID, Title and Description
- Course genre dataset comprising of Course ID and a list of Course Genres
- User Profile dataset comprising of User IDs and a list of Course Genres
- User Rating dataset comprising of User IDs, Course IDs and Course Rating

Introduction

Overview

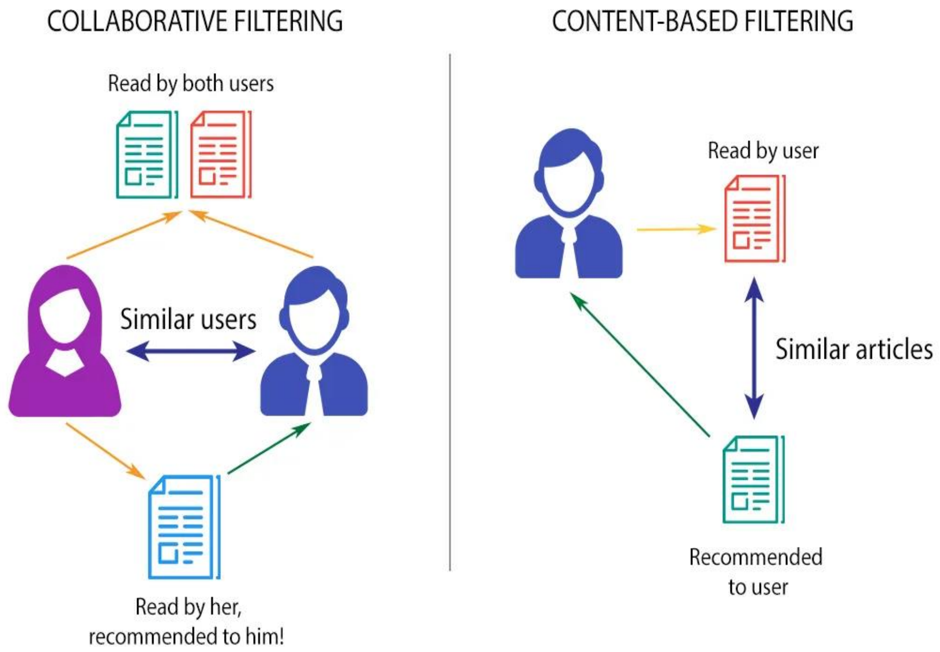
As mentioned in the executive summary, the goal of this capstone project is to build a recommender system (RS) that can:

- Predict course ratings for the courses
- Evaluate the prediction accuracy via metrics such as root mean square error (rmse)
- Make course recommendations from a list of UNseen courses to each user.
- Compile a list of top 10 most popular recommended courses across all users

Method

Two different methodologies have been implemented to study the RS:

- Content Based Filtering: Course predictions are made based on the similarity between items/courses rated by the user.
- User Based Collaborative Filtering Method: Course predictions are made based on similarities between users'.



UPSHOT

Content Based Recommender System

- User-course profiles
- Similarity Matrix
- Kmeans Clustering

Course Recommendations for Each User

TOP 10 Recommendations across all Users

User Based Recommender System (Collaborative filtering, CF)

- K-Nearest Neighbors
- Non-Matrix Factorization
- Neural Network Embedding
- Model Evaluation

Overall Results & Insights

1. **EDA**
   - The course enrollment data suggests that number of courses recommended to a user must be no greatrer than 20.
   - The most popular topics/genres from enrolled courses are : Data Science and Data related.

2. **Computation Time & RMSE**
   - Content Based Filtering models are **much faster** compared to User Based Collaborative Filtering models.
   - Models within Content Based Filtering showed no significant difference in run time. However, among the CF models, the Neural Network model with latent space embeddings took the least amount of run time.
   - The NN model also had the lowest RSME compared to the two other CF model, namely KNN and NMF.

3. **Average number of recommendation per user**
   - All models except NMF were able to restrict the average number of recommendations to below 20 with a reasonable and modest threshold.

4. **Top 10 recommendations across all users**
   - Interestingly, the top 10 recommended courses across all users seems to be heavily model-dependent.

Project Workflow

EDA

- Word cloud
- Popular Genres
- Enrollment
- BoW for Feature Extraction

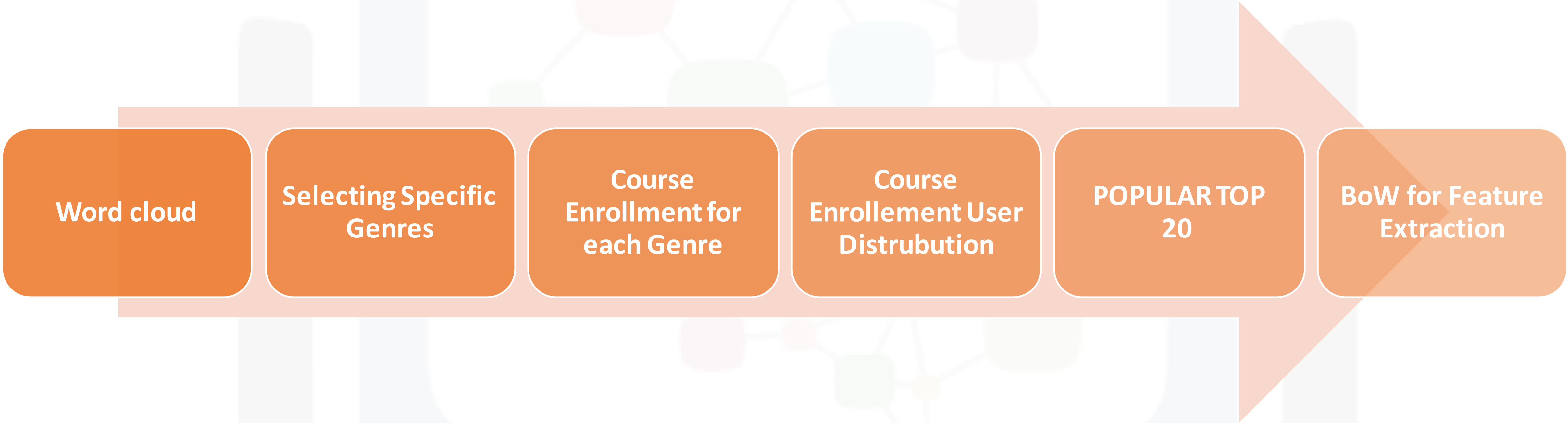
Content Based Filtering

- User profile and Course Genres
- Similarity Matrix
- K-means clustering

User Based Collaborative Filtering (CF)

- K-Nearest Neighbors
- Non-Matrix Factorization
- NN
- Evaluation based on NN Embedding

EDA/Feature Extraction:
Workflow

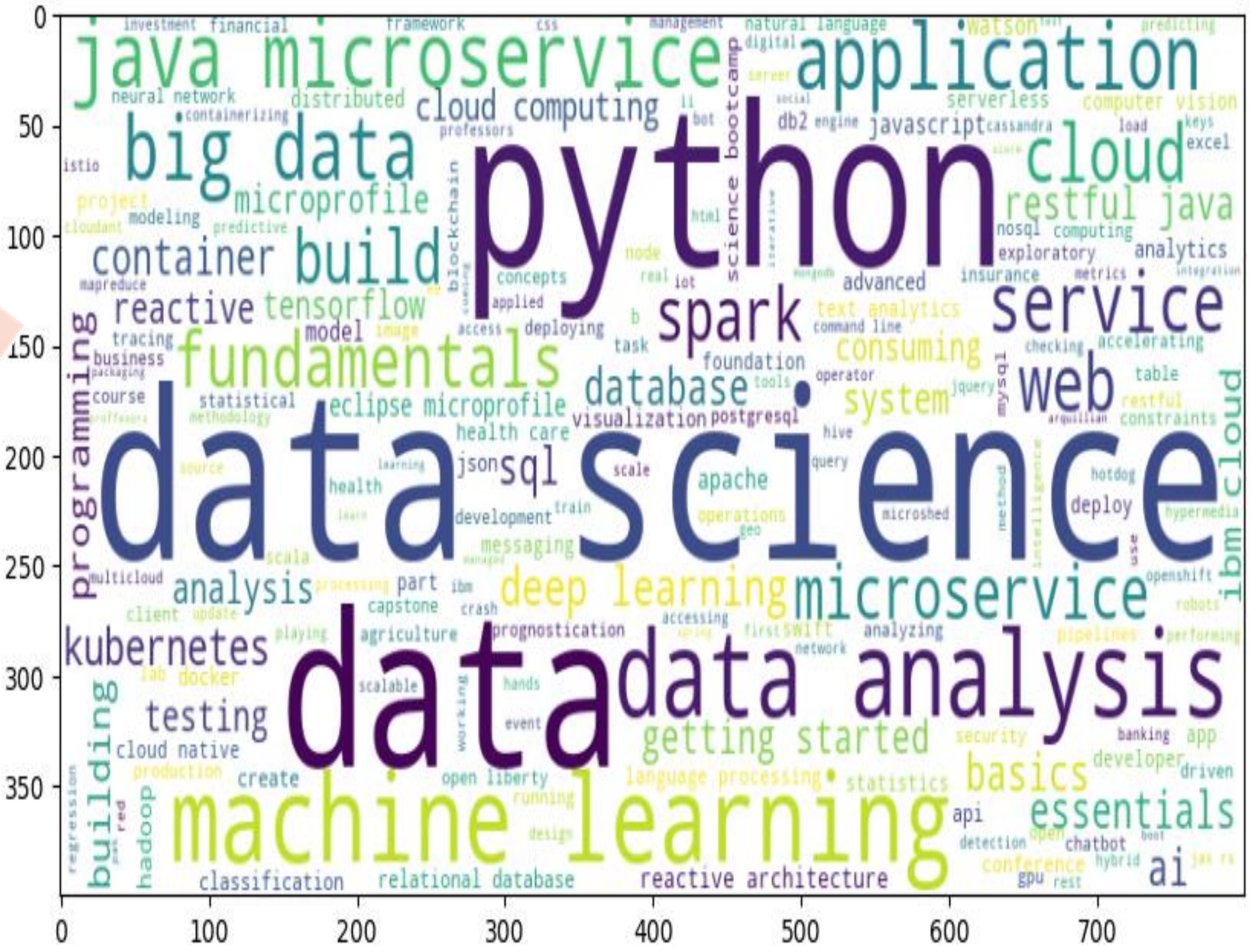


EDA: Discussion

Wordcloud graphically displays each word in a text based on its frequency of occurance.

Larger the text, greater is the frequency with which the word appears in the entire text

Useful to estimate the more popular course genre in our **data**



EDA: Data Exploration

[MachineLearning == 1]   **First 5** courses that match the condition 'Maching Learning ==1

[55]:
```
# WRITE YOUR CODE HERE
idx = course_df['MachineLearning']==1
course_df[idx].head().T
```

[55]:

	1	15	19	21	24
COURSE_ID	ML0122EN	DAI101EN	HCC105EN	DS0132EN	BENTEST4
TITLE	accelerating deep learning with gpu	data ai essentials	ybrid cloud conference ai pipelines lab	data ai jumpstart your journey	ai for everyone master the basics
Database	0	0	0	0	0
Python	1	0	0	0	0
CloudComputing	0	0	0	0	0
DataAnalysis	0	0	0	0	0
Containers	0	0	0	0	0
MachineLearning	1	1	1	1	1
ComputerVision	0	0	0	0	0
DataScience	1	0	0	0	0
BigData	0	0	0	0	0
Chatbot	0	0	0	0	0
R	0	0	0	0	0
BackendDev	0	0	0	0	0
FrontendDev	0	0	0	0	0
Blockchain	0	0	0	0	0

EDA: Data Exploration

[MachineLearning == 1 and BigData==1]

[56]:

```
# WRITE YOUR CODE HERE
idx = course_df['MachineLearning']==1
MLeq1 = course_df[idx]

idx2=MLeq1['BigData']==1
MLeq1[idx2].T
```

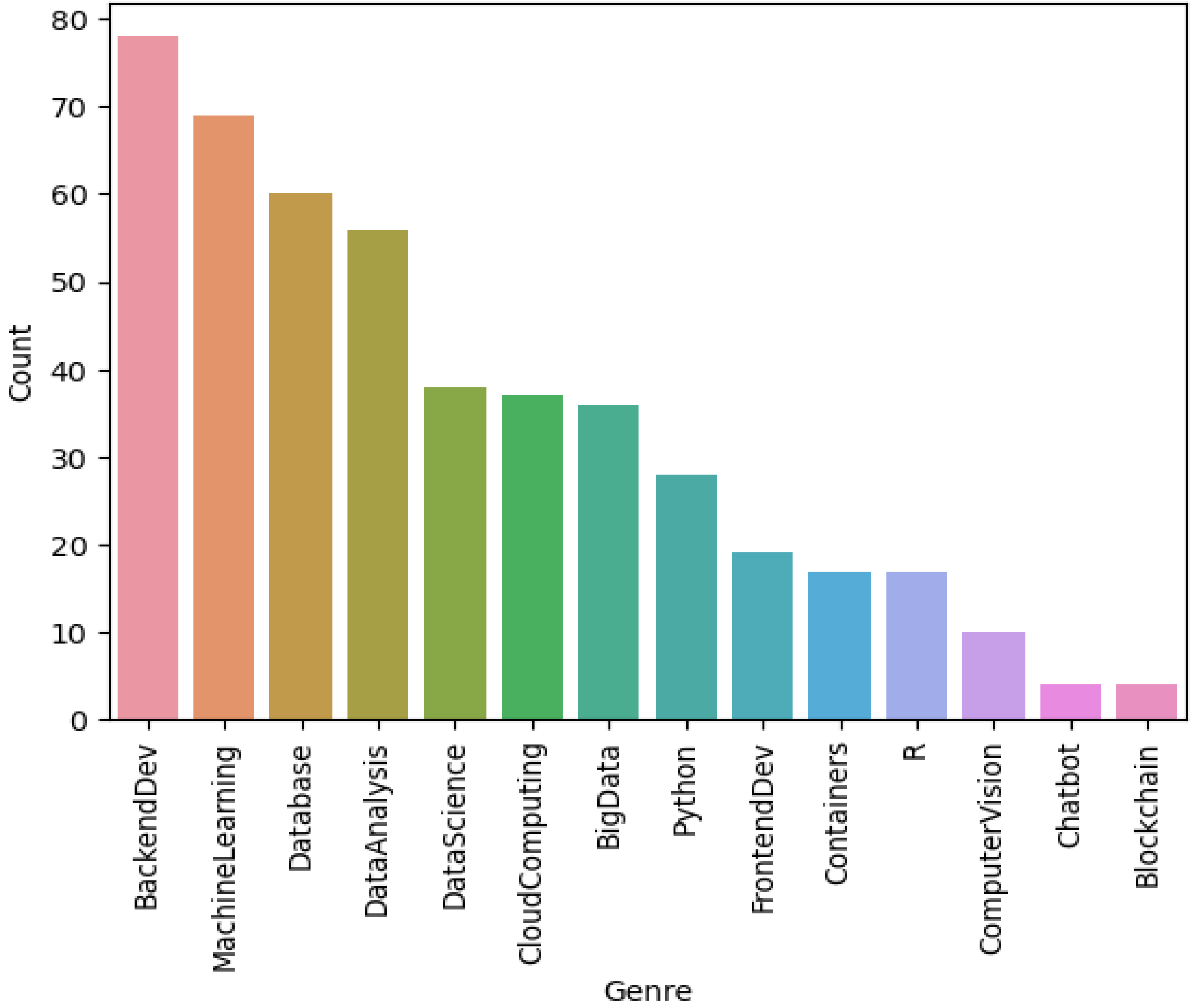
[56]:

	46	59	184	282
COURSE_ID	GPXX0BUBEN	TA0106EN	BD0221EN	excourse69
TITLE	insurance risk assessment with montecarlo meth...	text analytics at scale	spark mllib	machine learning with big data
Database	0	0	0	0
Python	0	0	0	0
CloudComputing	0	0	0	0
DataAnalysis	0	0	0	0
Containers	0	0	0	0
MachineLearning	1	1	1	1
ComputerVision	0	0	0	0
DataScience	0	1	0	0
BigData	1	1	1	1
Chatbot	0	0	0	0
R	0	0	0	0
BackendDev	0	0	0	0
FrontendDev	0	0	0	0
Blockchain	0	0	0	0

rnel) | Idle Mode: Command Ln 1, Col 1 lab_jupyter_eda.ipynb

EDA:
Popular Genres

	Genre	Count
0	BackendDev	78
1	MachineLearning	69
2	Database	60
3	DataAnalysis	56
4	DataScience	38
5	CloudComputing	37
6	BigData	36
7	Python	28
8	FrontendDev	19
9	Containers	17
10	R	17
11	ComputerVision	10
12	Chatbot	4
13	Blockchain	4



EDA:
Rating Count Aggregate

Total number of users after aggregation = 33901

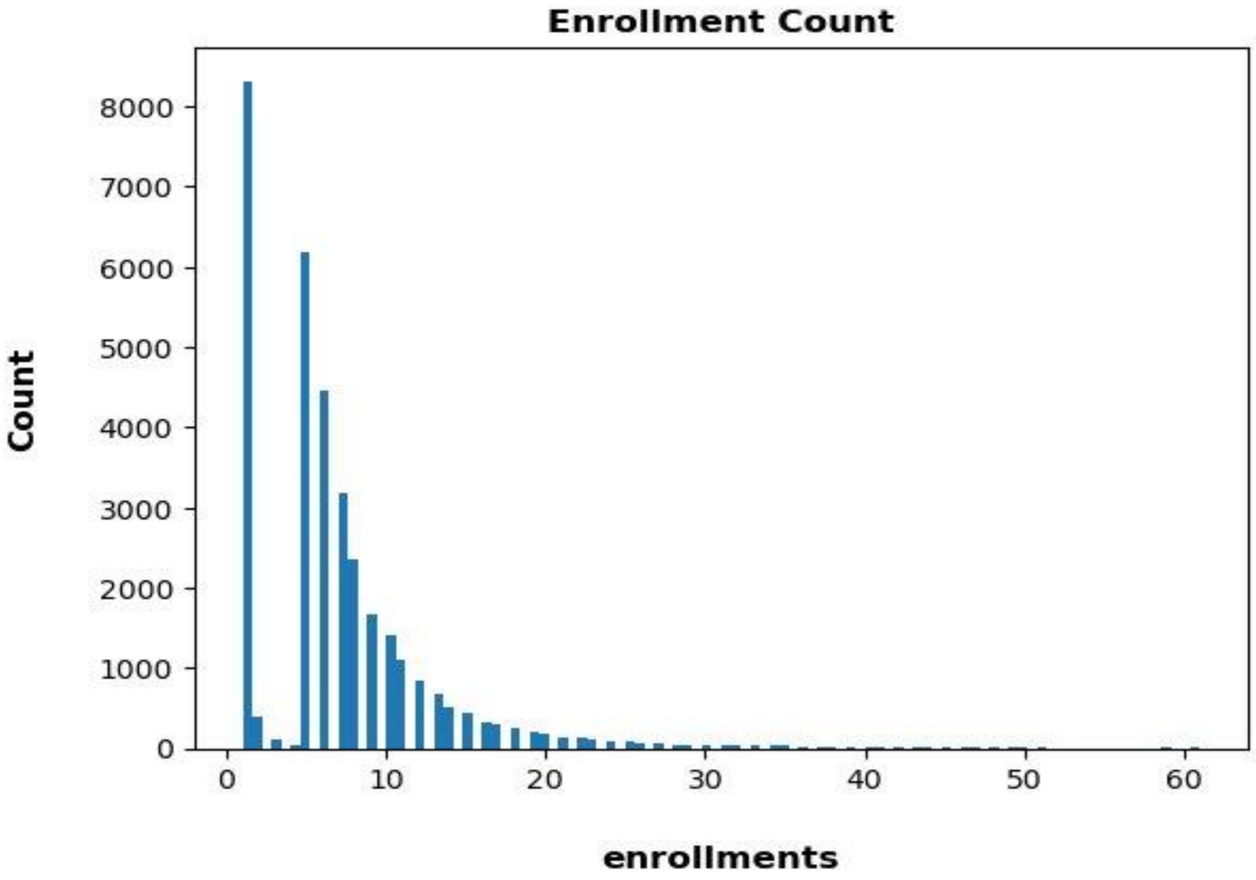
Rating Count Aggregate DataFrame

user	Aggregate Rating Count
2	61
5	59
924030	51
1653994	51
1761291	50
…	…
891557	1
892174	1
892298	1
892496	1
2103039	1

Statistical Description of Rating Count Aggregate Data

	Aggregate Rating Count
count	33901.000000
mean	6.881980
std	5.823548
min	1.000000
25%	2.000000
50%	6.000000
75%	9.000000
max	61.000000

EDA:
Enrollment Distribution



- This plot shows the distribution of the number of courses a user is enrolled into.
- It is clear from this plot that majority of the users have rated (aka been enrolled) in fewer than 10 courses.
- Less than 2000 users have enrolled themselves in between 10-20 courses.
- There are less than about 200 users with more than 20 course enrollments.
- **New Insight:**
  - Majority of the users have less than 20 courses enrollments
  - **Therefore, the number of new new course recommendations should be no more than 20.**

EDA:
Top 20 Courses with highest enrollment

Percentage of enrollement into the top 20 courses = 63 %

Total number of course enrollments = 233306
Total number of users enrolled in the top 20 courses =

	COURSE_ID	TITLE	No. of enrolls
0	PY0101EN	python for data science	14936
1	DS0101EN	introduction to data science	14477
2	BD0101EN	big data 101	13291
3	BD0111EN	hadoop 101	10599
4	DA0101EN	data analysis with python	8303
5	DS0103EN	data science methodology	7719
6	ML0101ENv3	machine learning with python	7644
7	BD0211EN	spark fundamentals i	7551
8	DS0105EN	data science hands on with open source tools	7199
9	BC0101EN	blockchain essentials	6719

	COURSE_ID	TITLE	No. of enrolls
10	DV0101EN	data visualization with python	6709
11	ML0115EN	deep learning 101	6323
12	CB0103EN	build your own chatbot	5512
13	RP0101EN	r for data science	5237
14	ST0101EN	statistics 101	5015
15	CC0101EN	introduction to cloud	4983
16	CO0101EN	docker essentials a developer introduction	4480
17	DB0101EN	sql and relational databases 101	3697
18	BD0115EN	mapreduce and yarn	3670
19	DS0301EN	data privacy fundamentals	3624

Feature Extraction: Bag of Words (BoW)

1. Bag of words (BoW) is a textual feature extraction method.
2. BoW is implemented to decompose the information about the each course into vector.
3. We use tokenize to convert the textual information (from the title and the course description) of each course into a vector.
4. This vector encapsulates the most significant words and their frequency of occurrence.
5. This table is an example of a BoW dataset, where each of the 306 courses are represented by a set of words describing the course and the number of times each of those words occur within the textual information regarding each course.

24253 rows × 4 columns

doc_index	doc_id	token	bow
0	ML0201EN	accelerate	1
0	ML0201EN	accelerated	5
0	ML0201EN	accelerating	1
0	ML0201EN	ai	2
0	ML0201EN	analyze	1
...	...	...	...
306	excourse93	understand	1
306	excourse93	unit	1
306	excourse93	uploading	1
306	excourse93	use	4
306	excourse93	videos	1

Content Based Filtering

Work Flow

User profile and Course Genres

- Threshold Vs No. of recommended courses
- Threshold Vs No. Of users
- TOP 10 recommended courses
- Avg rec courses per user

Similarity Matrix

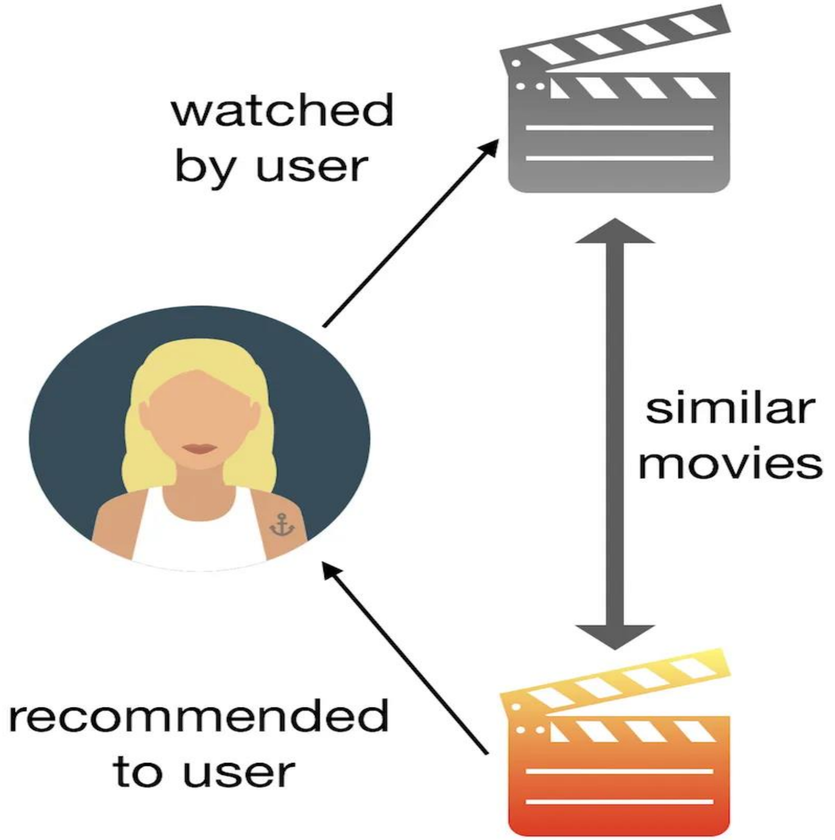
- Feature correlation
- My course recommendations
- TOP 10 recommended courses
- Avg rec courses per user

K-means clustering

- Choosing optimum K
- PCA analysis and choosing n_components for PCA
- Top 10 recommendation
- Average course recommendations per user

Content Based Filtering

General Concept



- Content based filtering relies on recommending courses to the users based on similar courses that the users liked or rated highly.
- This is similarity between well liked courses and the unseen courses are determined by various methods such as
  - Comparing user profiles and course genres
  - Looking at the similarity matrices between courses
  - Clustering models etc ..

Content Based Filtering

user profile and course genre - Concept Overview

User profile vector ***u***

	Python	Database	Machine Learning
user1	1.0	0	1.0

dot product

Course genre vector $\boldsymbol{i_1}$

	Genre
Python	1
Database	0
Machine Learning	1

→ Score: 2.0

dot product

Course genre vector $\boldsymbol{i_2}$

	Genre
Python	1
Database	1
Machine Learning	0

→ Score: 1.0

1. One way to implement content based filtering is by identifying similarities between the users' interest (in courses or specific genres) and the features of a given course (like topic, genres etc)
2. Given a user profile vector (data) - which comprises of the users' interest in certain topics – we can identify courses of interest to the user by combining the user profile vector with the course genre vector (which contains information about the various topics dicussed within the course)
3. The similarities between the user profile and course genres are score mathematically.
4. Predictions are made to the user if the similarity score is greater than the (pre-defined) threshold score.
5. The threshold score is set to 50

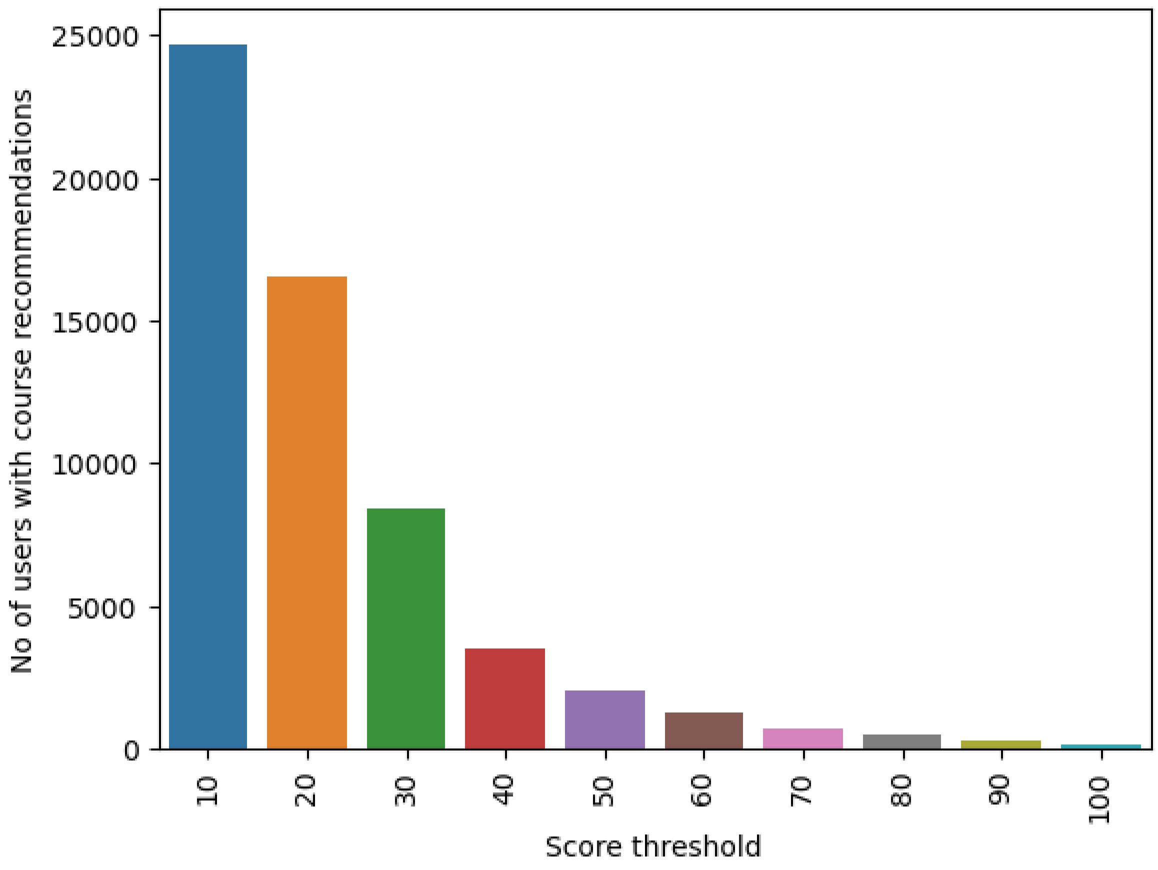
Content Based Filtering

user profile and course genres : Discussion

1. This is a plot that shows how the choice of an absolute score threshold impacts the number of users that given course recommendations.

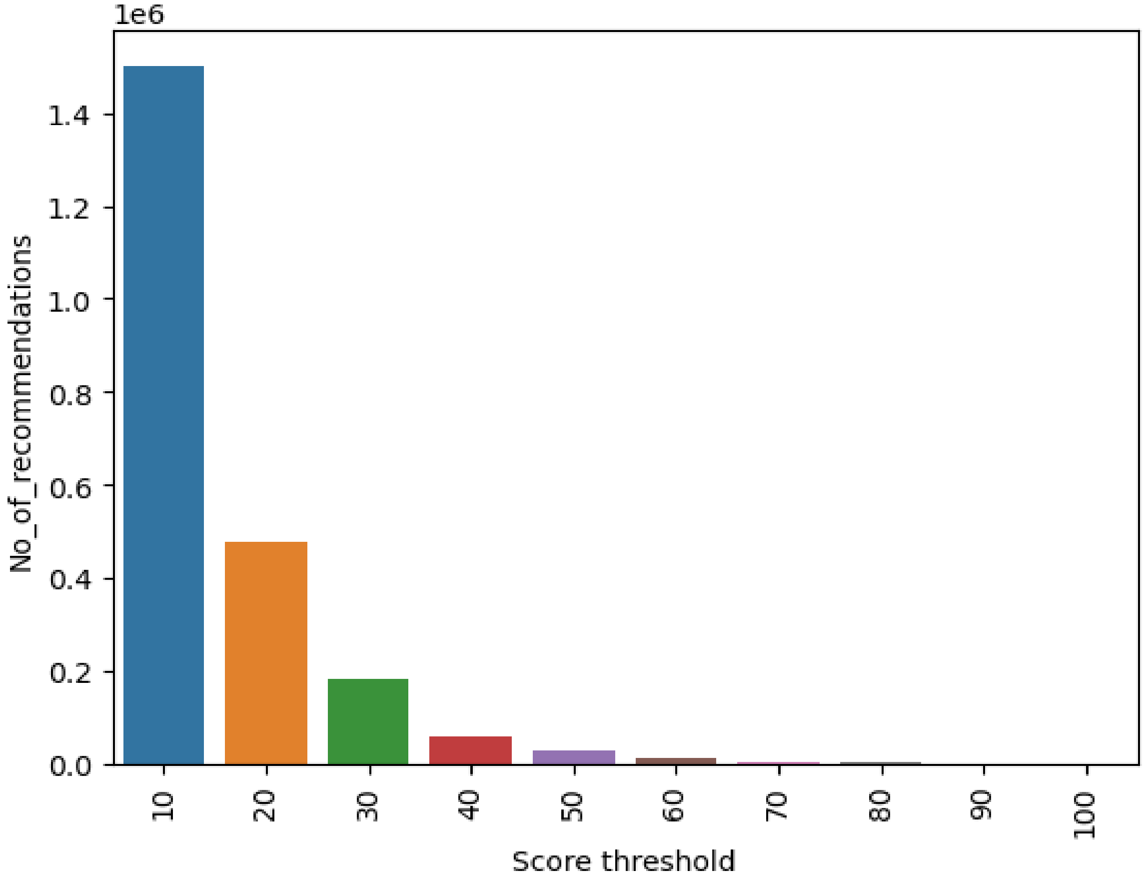
2. We see that fewer users are given course recommendations as the score threshold increases.

3. We must therefore pick a threshold that is large enough to pick up courses with a good similarity score but also small enough that considerable number of users are sent recommendations.



Content Based Filtering

user profile and course genres : Discussion



1. Here is a plot that displays the relationship between total number of courses recommended and the score threshold.

2. Greater the threshold, fewer courses recommended.

3. This matches our intuition.

Content Based Filtering

user profile and course genres : Discussion

Average number of courses recommended per user = 14

1. Implemented content based filtering using SIMILARITY matrix
2. Identified the UNSEEN courses withing each cluster for EACH by studying the course similarity from the similarity matrix
3. Predicted UNSEEN courses for EACH user when the similarity score (threshold) was > 0.6
4. Computed Average recommended courses per user and top 10 recommended courses

Course Id	**Top 10 recommended courses (across all users)**
RP0105EN	analyzing big data in r using apache spark
TMP0105EN	getting started with the data apache spark ma...
SC0103EN	spark overview for scala analytics
GPXX0M6UEN	using the cql shell to execute keyspace operat...
GPXX097UEN	performing table and crud operations with cass...
excourse05	\r\ndistributed computing with spark sql
excourse10	database architecture scale and nosql with e...
excourse31	cloud computing applications part 2 big data...
excourse72	foundations for big data analysis with sql
excourse73	analyzing big data with sql

Content Based Filtering

Similarity Matrix : Concept Overview

Course 1: “Machine Learning for Everyone”

	machine	learning	for	everyone	beginners
course1	1	1	1	1	0

Course 2: “Machine Learning for Beginners”

	machine	learning	for	everyone	beginners
course2	1	1	1	0	1

Similarity Calculation:
Cosine, Euclidean, Jaccard index, …

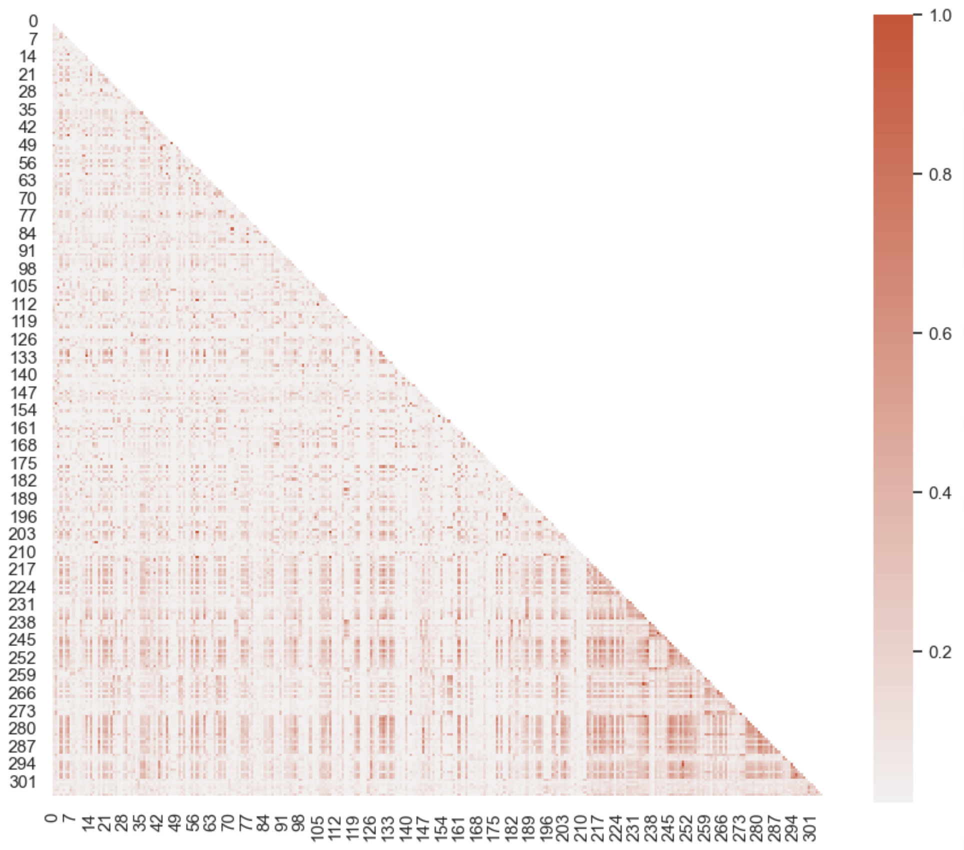
1. Content based filtering based on similarity matrix is a very intuitive method to make prediction for recommended courses to the users.

2. This method used the course-bow (bag of words) dataset that comprises of a list of all courses and the tokenized words that describe each of these courses.

3. Similarity between courses are then determined by extracting the respective similarity score from the pre-defined similarity matrix

Content Based Filtering

Similarity Matrix : Methodology



1. This plot shows the pairwise similarity between the various courses obtained by computing a similarity matrix between all these courses.

2. The similarity matrix used for this analysis is cosine similarity.

    a. The values of this matrix are real

    b. Range between 0 and 1

3. Unseen courses are identified for each user and a similarity score is obtained from the similarity matrix between the unseen course and the users' enrolled courses.

4. Predictions are made to the user if the similarity score is greater than the (pre-defined) threshold score.

5. The threshold score is set to 0.6.

Content Based Filtering

Similarity Matrix : Methodology

Courses Chosen by ME

COURSE_ID	TITLE
GPXX04XJEN	advanced machine deep learning for spam classification task
GPXX0ZMZEN	data science in health care advanced machine learning classification
excourse48	introduction to machine learning language processing
excourse60	introduction to tensorflow for artificial intelligence machine learning and deep learning

This table shows a list of a few courses chosen by me.

Recommended Courses based on course similarity

COURSE_ID	TITLE	Score
ML0115EN	**deep learning 101**	**0.615568**
excourse46	**machine learning**	**0.628894**
excourse47	**machine learning for all**	**0.882362**
excourse61	**convolutional neural networks in tensorflow**	**0.630767**

This table shows a list of courses **predicted by** the model based on learning course similarities

Content Based Filtering

Similarity Matrix : Methodology

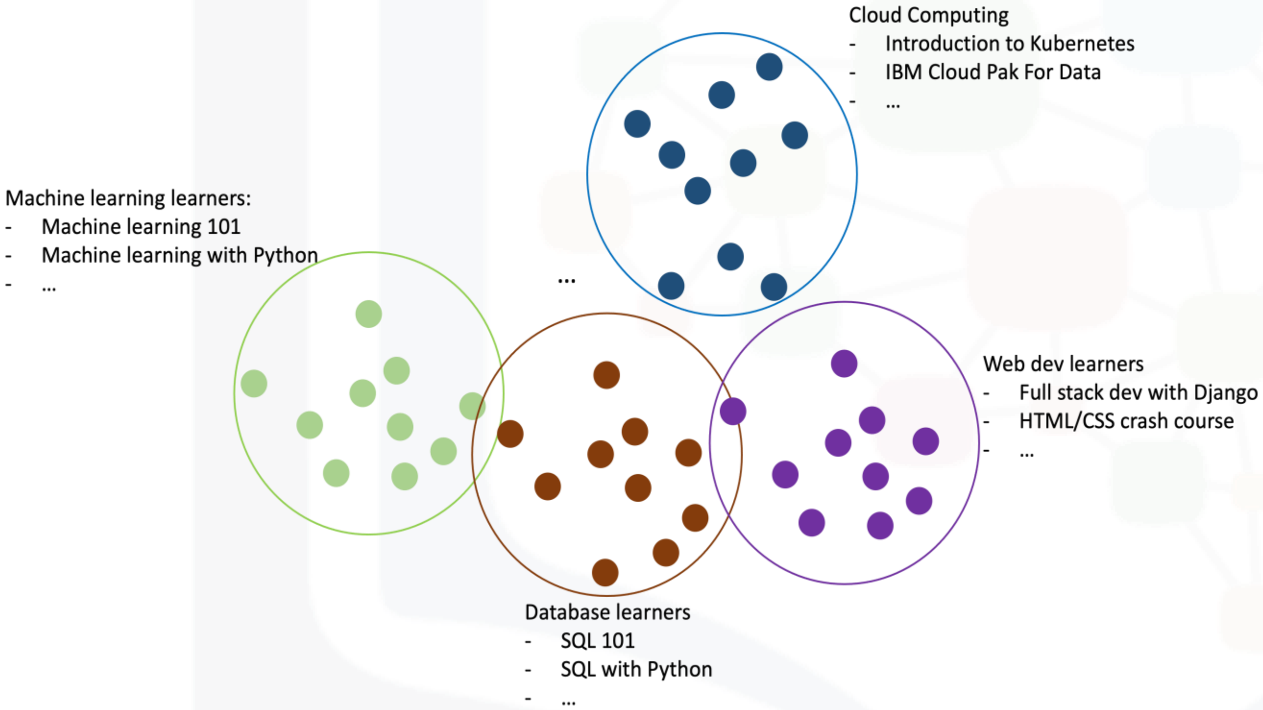
Average number of courses recommended per user = 9

- Implemented content based filtering using SIMILARITY matrix
- Identified the UNSEEN courses withing each cluster for EACH by studying the course similarity from the similarity matrix
- Predicted UNSEEN courses for EACH user when the similarity score (threshold) was > 0.6
- Computed Average recommended courses per user and top 10 recommended courses

COURSE_ID	Top 10 recommended courses (across all users)
DS0110EN	data science with open data
excourse22	introduction to data science in python
excourse62	introduction to data science in python
excourse63	a crash course in data science
excourse65	data science fundamentals for data analysts
excourse72	foundations for big data analysis with sql
excourse68	big data modeling and management systems
excourse67	introduction to big data
excourse74	fundamentals of big data
BD0145EN	sql access for hadoop

Content Based Filtering

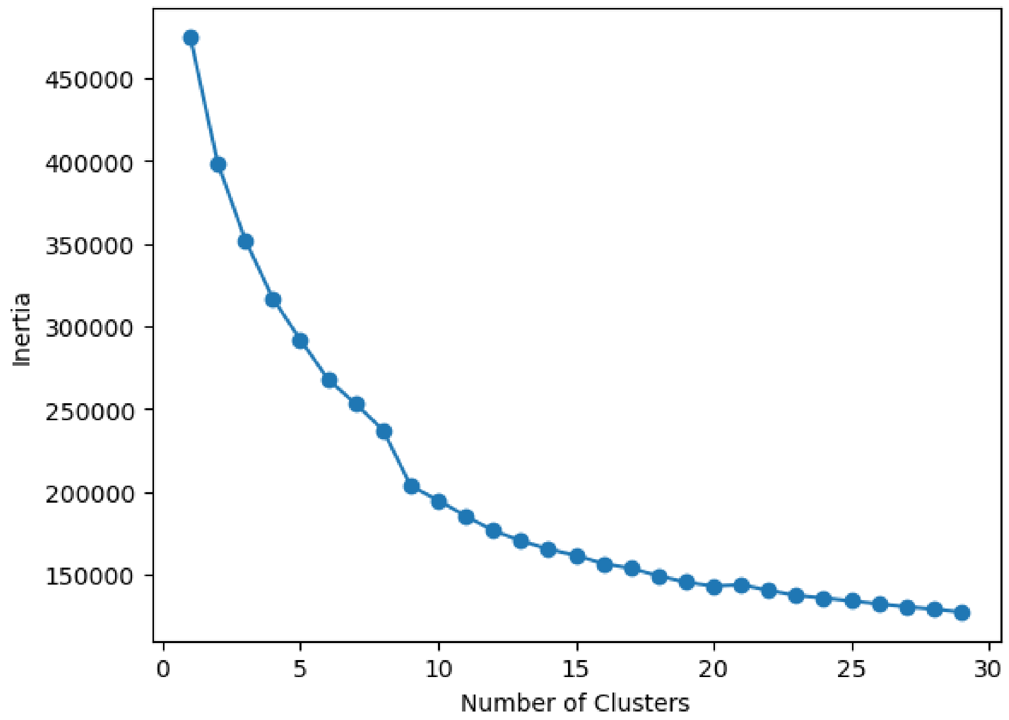
K-Means Clustering : Methodology



1. Clustering groups the users into different clusters based on their user profile data.
2. The user profile data comprises of the information of users' interest in various course genres.
3. Users with interest in similar couses will be grouped into a common cluster.
4. Unseen data (unseen courses) are then identified for each user within a group and course prediction are then made to each user.

Content Based Filtering

K-Means Clustering : Discussion



1. K-means clustering is performed for a specific number of pre-defined clusters.

2. It is therefore important to find the optimum number of clusters suitable for our model.

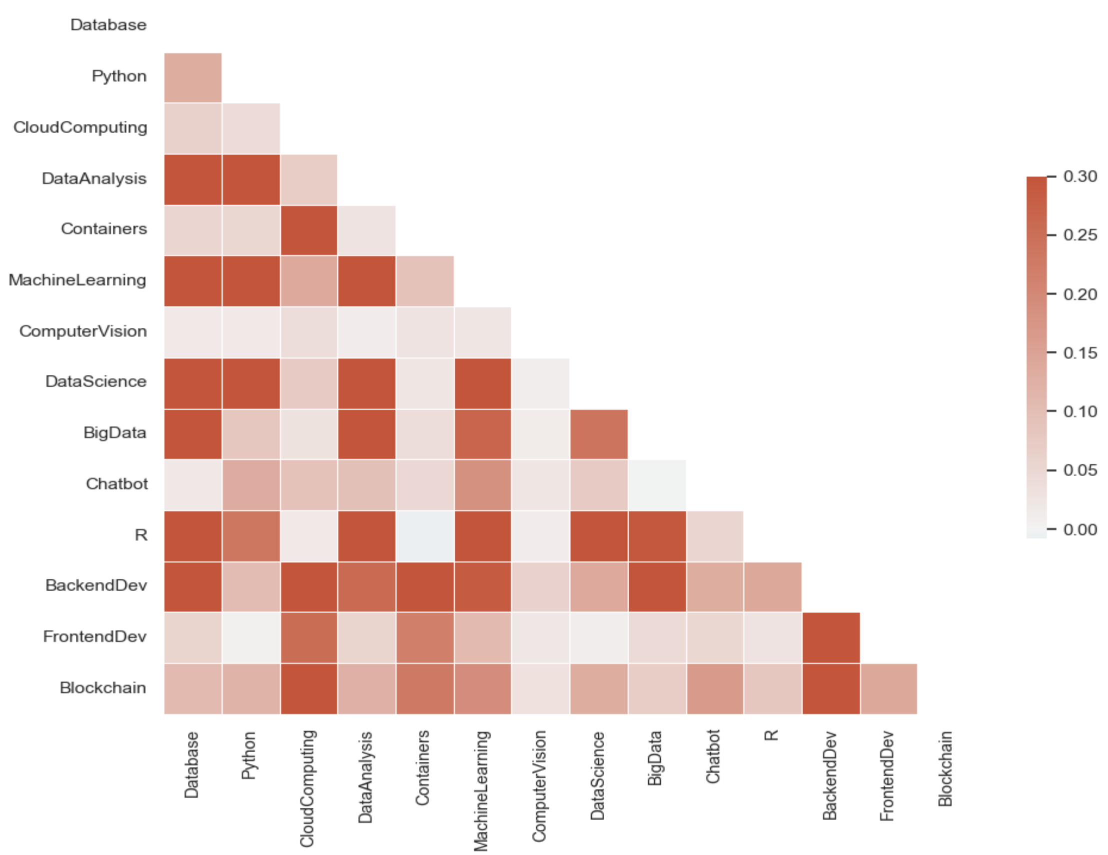
3. This figure shows the intertia for a large range of number of clusters.

4. The optimum value of n_clusters is found by picking out the value of n_clusters where the curve starts to plateau close to the lower end of the intertia values.

5. Based on this plot, we identify the optimum number of clusters to be, n_clusters = 20.

Content Based Filtering

K-Means Clustering : Discussion



1. This plot shows the pairwise correlation between the various features (course genres) of the course_genre data set.

2. The boxes with the darker shades indicate greater pairwise correlation than the ones with the lighter shades.

3. The presence of large number of highly correlated features (course genres) indicate a need to:

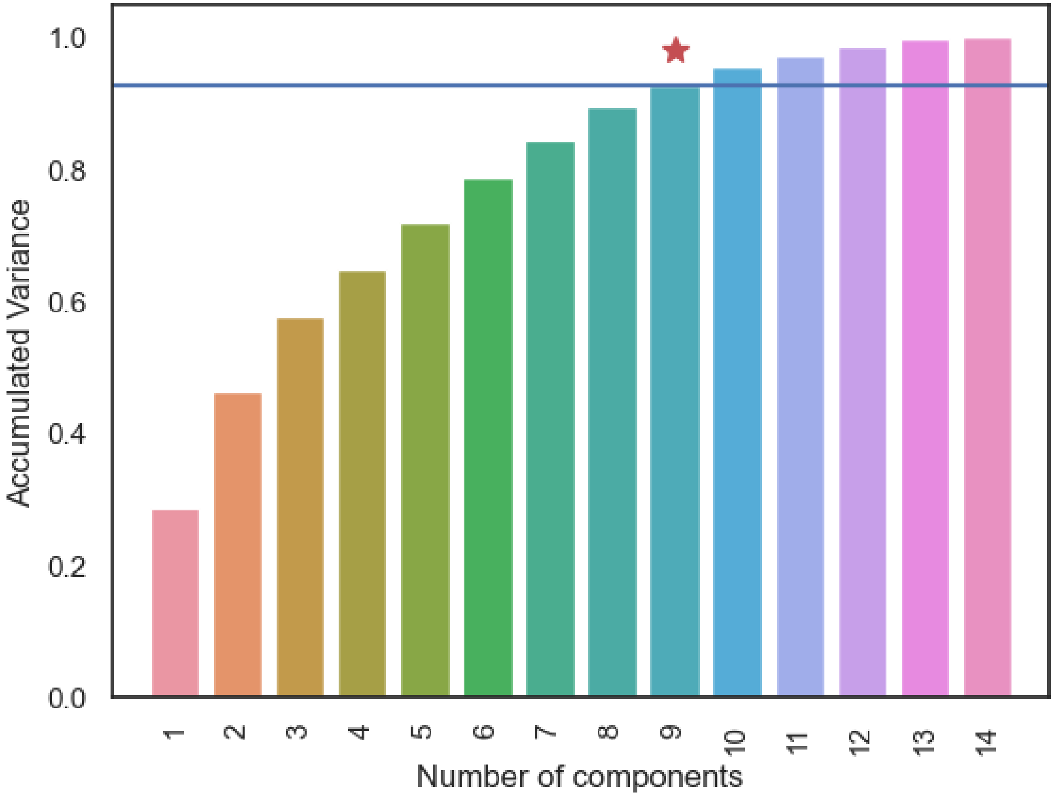
    a. Identify and use only those features that are significant to the analysis

    b. Reduce the dimensionality of the features space for more efficient computations.

4. Applying Principle Component Analysis (PCA) on this feature space will address the two points mentioned above. More on the next slide.

Content Based Filtering

K-Means Clustering : Discussion



1. PCA was performed on the user profile dataset for a set of PCA components (n_components) ranging from 1 to 14, in order to identify the minimum number of components that capture 90% of all the accumulated variances within the feature space.

2. This figure is a plot of accumulated variances over a range PCA components.

3. As the plot indicates, 90% (and more) of the accumulated variances are captured when the number of componanets if 9 and above.

4. PCA analysis at n_componenets = 9 is performed to capture the more impacting and significant variances within the feature space.

5. K-means is then applied to this PCA(n=9) feature dataset.

Content Based Filtering

K-Means Clustering : Discussion

1. K-Means clustering is applied to the user-profile dataset.
2. Every user belongs to a unique cluster.
3. This table displays:
   a. different courses within a particular cluster
   b. Number of users enrolled for each course within a particular cluster
4. UNSEEN courses are determined for user within a cluster
5. Course predictions among the set of unseen courses are made for each user based on the established threshold.

Cluster ID	Course Id	User Enrollment
0	AI0111EN	6
0	BC0101EN	137
0	BC0201EN	26
0	BC0202EN	9
0	BD0101EN	729
...	...	...
19	TA0106EN	2
19	TMP0105EN	151
19	TMP0106	27
19	WA0101EN	47
19	WA0103EN	7

2022 rows × 3 columns

Content Based Filtering

K-Means Clustering : Discussion

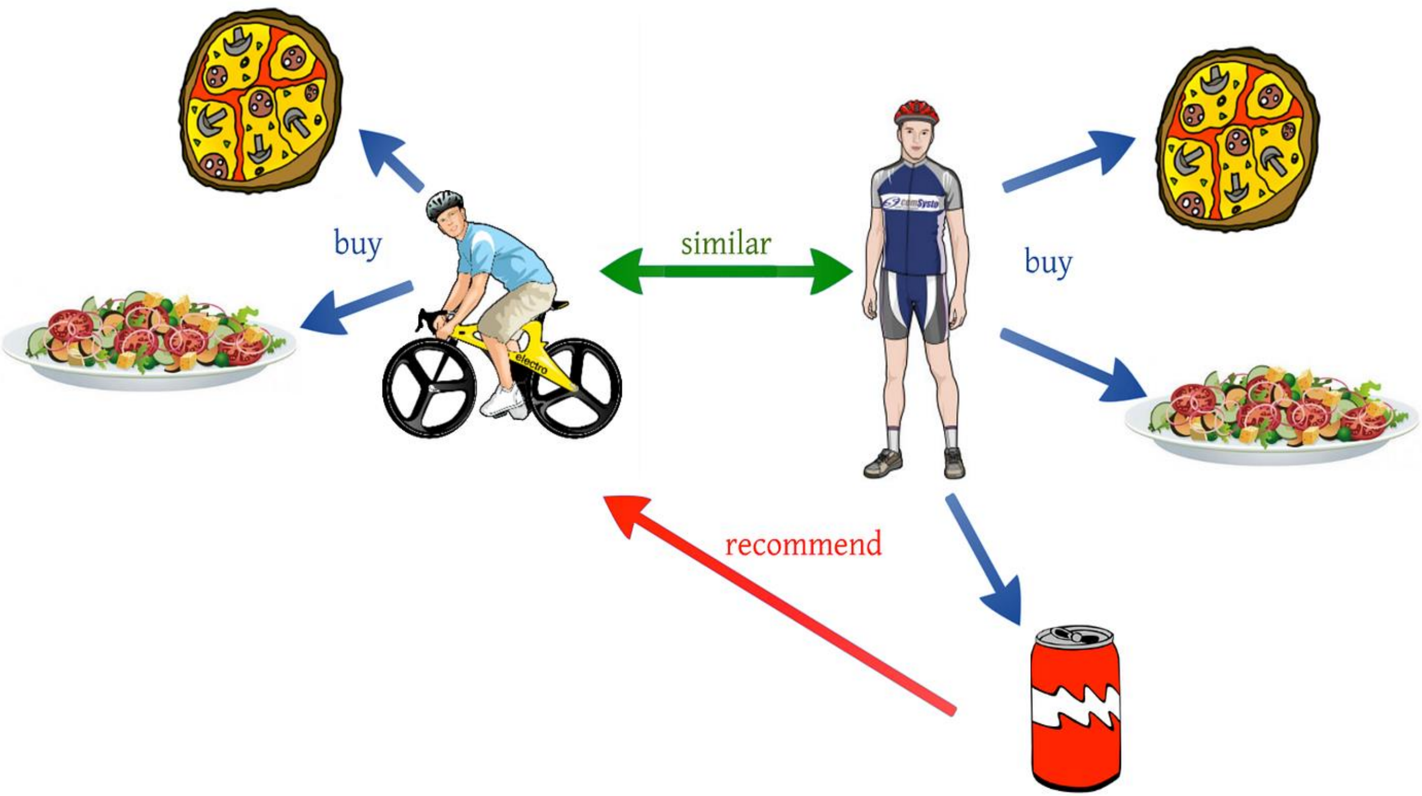
Average number of courses recommended per user = 19

- Implemented K-Means clustering model
- Obtained optimum value for K by looking at the elbow plot
- Applied PCA analysis to pick up only the relevant and important features.
- Identified the UNSEEN courses withing each cluster for EACH user belonging
- Computed Average recommended courses per user and top 10 recommended courses

Course ID	Top 10 recommended courses (across all users)
ST0101EN	statistics 101
ML0115EN	deep learning 101
DS0103EN	data science methodology
BD0111EN	hadoop 101
RP0101EN	r for data science
DB0101EN	sql and relational databases 101
BD0101EN	big data 101
CC0101EN	introduction to cloud
DS0301EN	data privacy fundamentals
CL0101EN	ibm cloud essentials

User Based Collaborative Filtering

General Concept



- User based collaborative filtering **looks for the data from similar users** to make a prediction to a user/customer.
- In this example, of the three item available to them, persons A (one on the bike) and person B (one standing) both share a common liking for pizza and salad.
- Given this similarity in interest in food choices, a prediction of a soda/pop is made to person/user A (one on the bike) based on his matching interest with person B (standing).

User Based Collaborative Filtering

Workflow

K-Nearest Neighbors (KNN)

- Computing RMSE
- K Vs RMSE
- Top 10 recommendation
- Average recommended courses per user

Non-Matrix Factorization (NMF)

- Computing RMSE
- TOP 10 recommended courses
- Avg rec courses per user

Neural Network (NN) Embedding

- Computing RMSE
- Embedding Depth Vs RMSE
- TOP 10 recommended courses
- Avg rec courses per user

Model Evaluation with NN feature embeddings

- Classification models
- Regression models

User Based Collaborative Filtering

K-Nearest Neighbors : Concept Overview

User-Item interaction matrix

	Machine Learning With Python	Machine Learning 101	Machine Learning Capstone	SQL with Python	Python 101
...	...	...	...	...	
user2	3.0	3.0	3.0	3.0	3.0
user3	2.0	3.0	3.0	2.0	
user4	3.0	3.0	2.0	2.0	3.0
user5	2.0	3.0	3.0		
user6	3.0	3.0	?		3.0
...	...	...	...	...	

Similar users

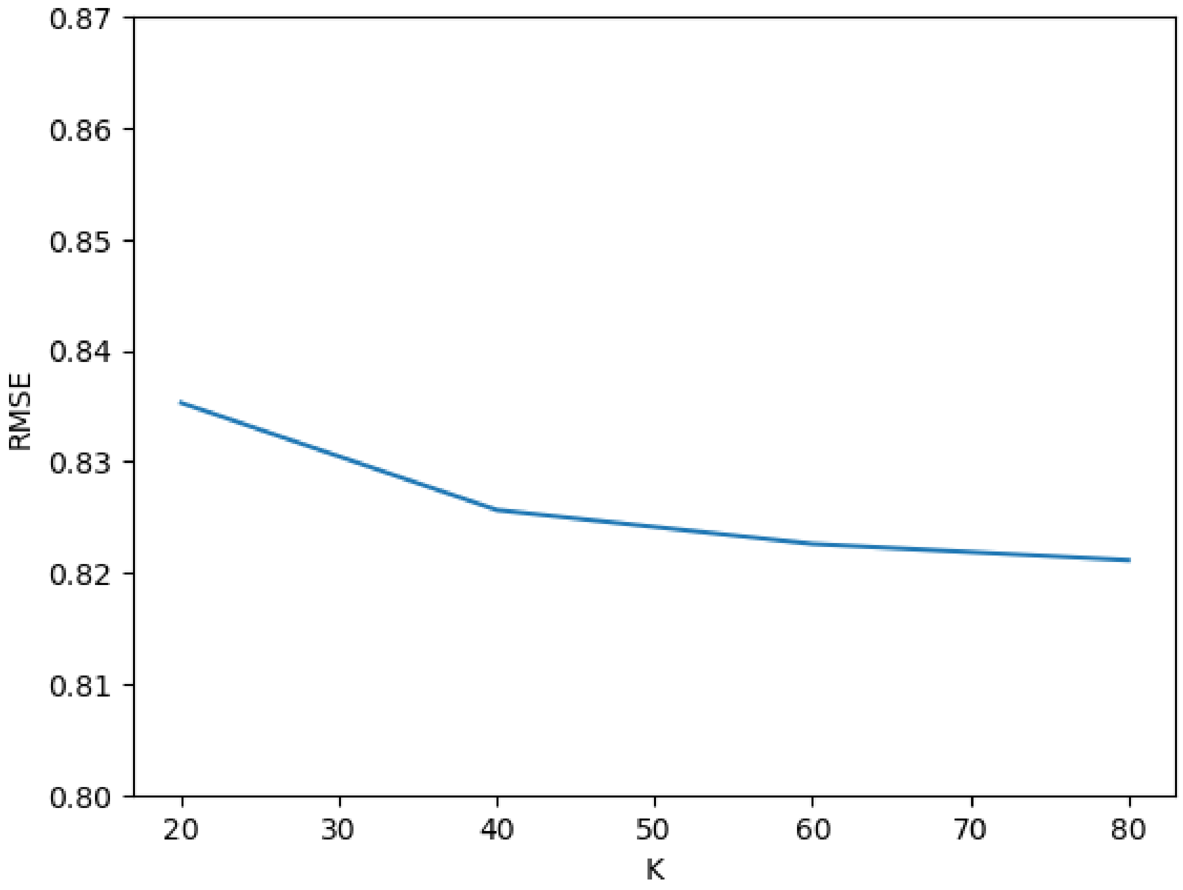
Predict the rating of user *user6* to item *Machine Learning Capstone*

- This large sparse matrix is our **user-item (item = course) rating data** of dimensions (33901, 127).
- User based collaborative filtering **looks for the data from similar users** to make a prediction for a particular course.

User Based Collaborative Filtering

KNN : Nearest neighbors, K vs RMSE

Number of Neighbors, K Vs Root Mean Square Error, RMSE



K tuning result

RMSE does not vary with the number of K-nearest neighbors !

User Based Collaborative Filtering

KNN : Observations

Average number of courses recommended per user = 5!

- Implemented NMF using surprise package
- Trained the model and computed rmse training error and prediction error on test data
- Identified UNSEEN courses for EACH user for predicted rating > 4.7
- Computed Average rating per user and top 10 recommended courses

COURSE_ID	Top 10 recommended courses (across all users)
DB0113EN	db2 fundamentals i
PA0103EN	predicting customer satisfaction
PA0107EN	predicting financial performance of a company
BD0151EN	text analytics 101
PY0101EN	python for data science
DS0101EN	introduction to data science
BD0101EN	big data 101
DS0132EN	data ai jumpstart your journey
BD0111EN	hadoop 101
DA0101EN	data analysis with python

User Based Collaborative Filtering

Non-Matrix Factorization (NMF) : Concept overview

- **NMF is a dimensionality reduction technique**, often used to decompose a large sparse matrix into smaller matrices

- As seen in the KNN concept overview slide, the user-item (item = course) rating data is a large sparse matrix of dimensions (33901, 127). We use NMF to decompose our user-item rating data into two smaller matrices, namely a user-interaction matrix of shape (33901, 16) and a item-interaction of shape (16, 127).

User-item interaction matrix: **A** 10000 x 100

	item1	...	item100
user1	...	...	
user2	3.0	3.0	3.0
user3	2.0	2.0	-
user4	3.0	2.0	3.0
user5	2.0	-	-
user6	3.0	-	3.0
...	...	...	

≈

User matrix: **U** 10000 x 16

	feature1	...	feature16
user1	...	...	...
user2	...	...	...
user3	...	...	...
user4	...	...	...
...	...	...	...
...	...	...	...
user6	...	...	...

X

Item matrix: **I** 16 x 100

	item1	...	item100
feature1	...	...	...
feature2	...	...	...
...	...	...	...
feature16	...	...	...

User Based Collaborative Filtering

Non-Matrix Factorization (NMF) : Discussion

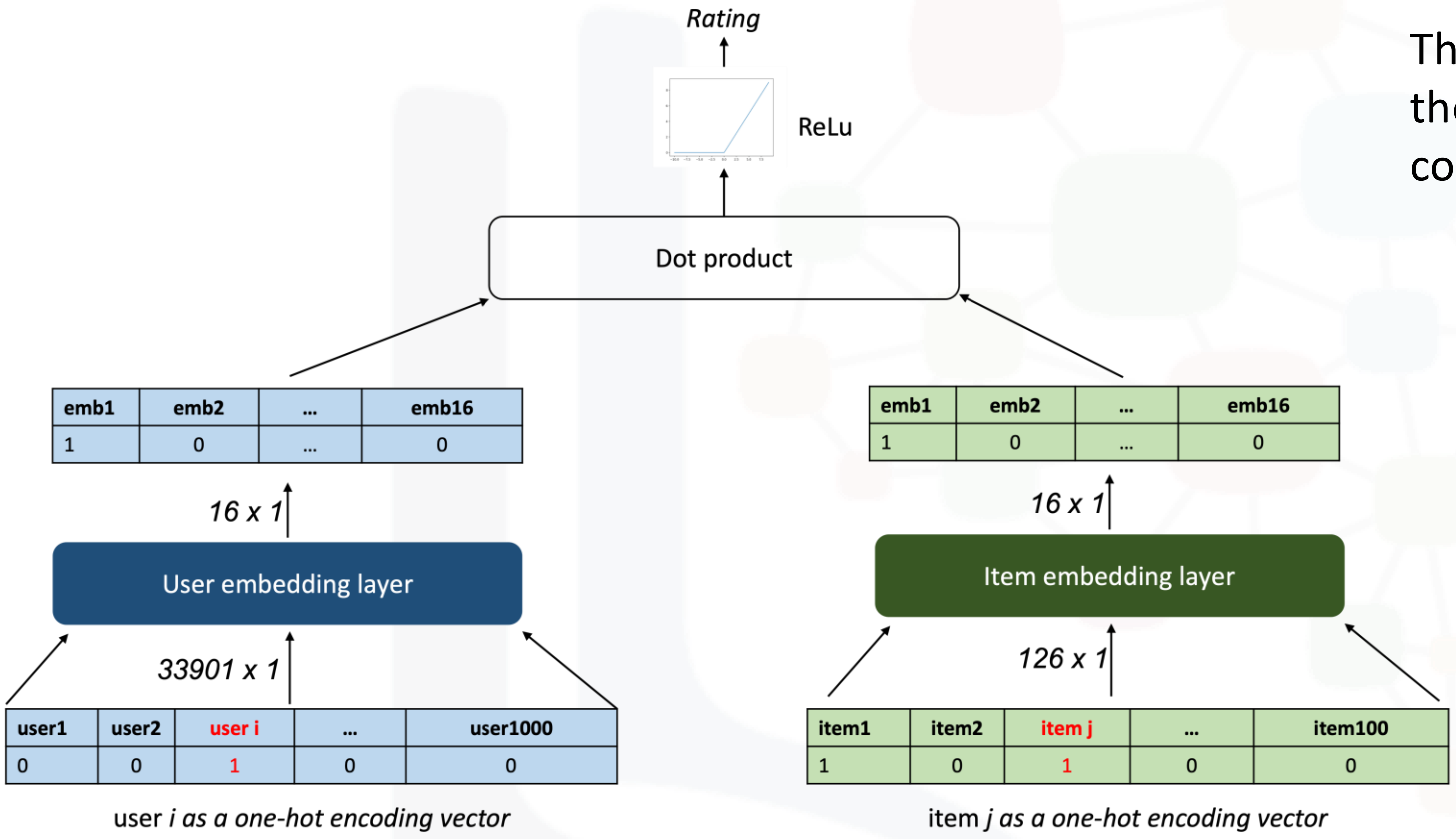
Average number of courses recommended per user = 88 !

1. Implemented NMF using surprise package
2. Trained the model and computed rmse training error and prediction error on test data
3. Identified UNSEEN courses for EACH user for predicted rating > 4.7
4. Computed Average rating per user and top 10 recommended courses

COURSE_ID	Top 10 recommended courses (across all users)
RAVSCTEST1	scorm test 1
DX0107EN	data science bootcamp with python for university professors
BD0151EN	text analytics 101
DS0132EN	data ai jumpstart your journey
ML0122EN	accelerating deep learning with gpu
DS0107	data science career talks
ML0120ENv3	deep learning with tensorflow
HCC104EN	hybrid cloud conference serverless lab
PA0109EN	using clustering methods for investment portfolio analysis
DS0110EN	data science with open data

User Based Collaborative Filtering

Embedding using Neural Network (NN) : Overview



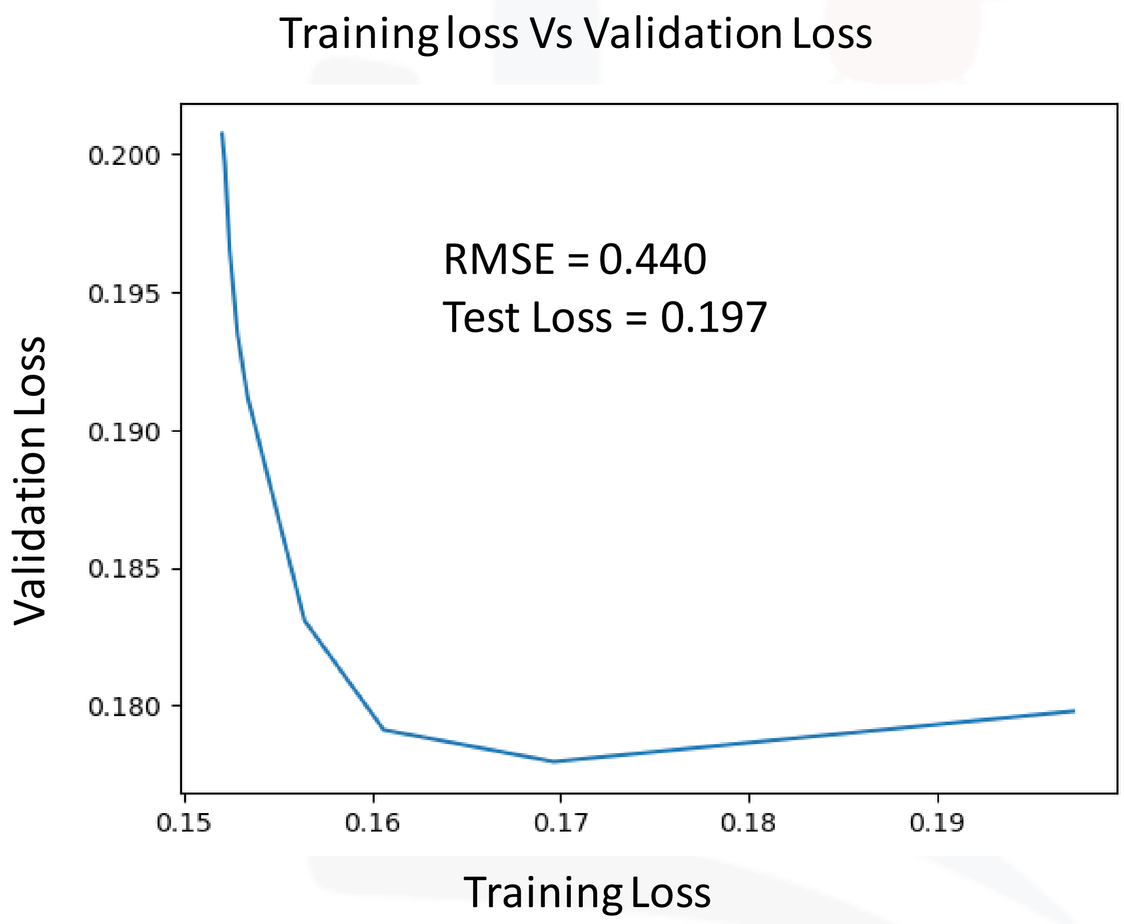
The ratings data frame comprises of three columns- the users, the courses and their ratings for the courses.

user	item	rating
1889878	CC0101EN	5
1342067	CL0101EN	3
1990814	ML0120ENv3	5
380098	BD0211EN	5
779563	DS0101EN	3

1. The user and the item (course id) columns are one-hot encoded respectively
2. These one-hot encoded vectors are fed into a neural network and trained.
3. The resulting 'embedded' vectors are combined via a dot product to finally predict the course ratings

User Based Collaborative Filtering

Embedding using Neural Network (NN) : Overview



- The model loss was computed by mean squared error
- Optimized used : Adam
- The model was evaluated based on the root mean square metric.

User Based Collaborative Filtering

Embedding using Neural Network (NN) : Observations

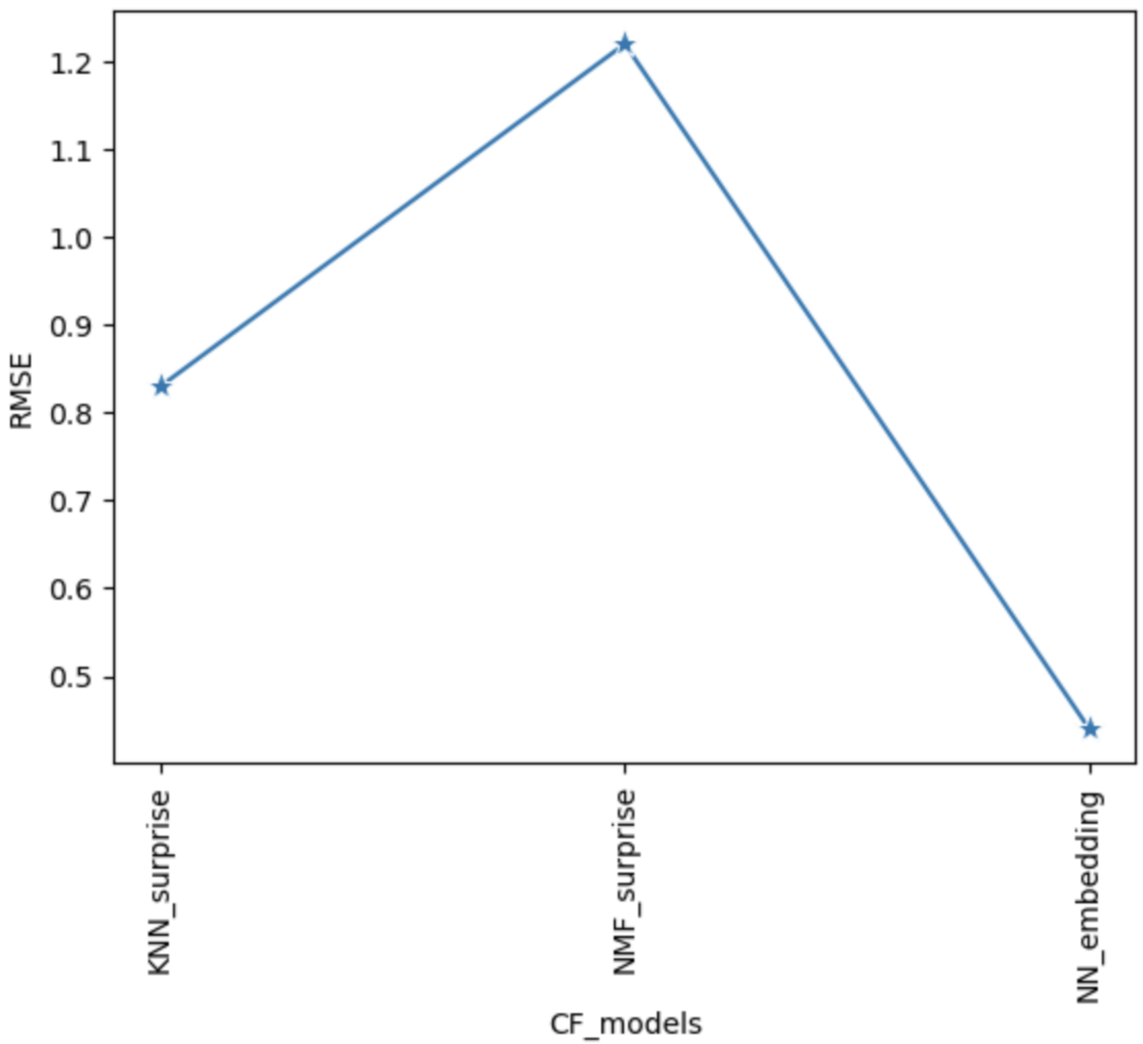
Average number of courses recommended per user = 8

- Implemented KNN using surprise package
- Performed K vs RMSE analysis to find optimum K
- Trained the model and computed prediction error on test data
- Identified UNSEEN courses for each user and chose the unseen courses that had a predicted rating > 4.4
- Predicted course ratings for UNSEEN courses for each user

Course Id	Top 10 recommended courses (across all users)
BD0151EN	text analytics 101
CB0105ENv1	node red basics to bots
DS0201EN	end to end data science on cloudpak for data
DS0132EN	data ai jumpstart your journey
ML0151EN	machine learning with r
RAVSCTEST1	scorm test 1
CNSC02EN	cloud native security conference data security
DV0151EN	data visualization with r
ML0201EN	robots are coming build iot apps with watson swift and node red
LB0105ENv1	reactive architecture reactive microservices

User Based Collaborative Filtering

Results : CF Models Vs RMSE



- This is a plot of the RMSE obtained from training the CF models using the training data.
- The three models are – KNN, NMF and NN
- The RMSE is the lowest for the NN model.

User Based Collaborative Filtering

Rating Scale Prediction : Methodology

Using a neural network to compute user and course embeddings, we evaluate the model for both classification and regression

Regression Models

- Basic Linear Regression
- Ridge Regression

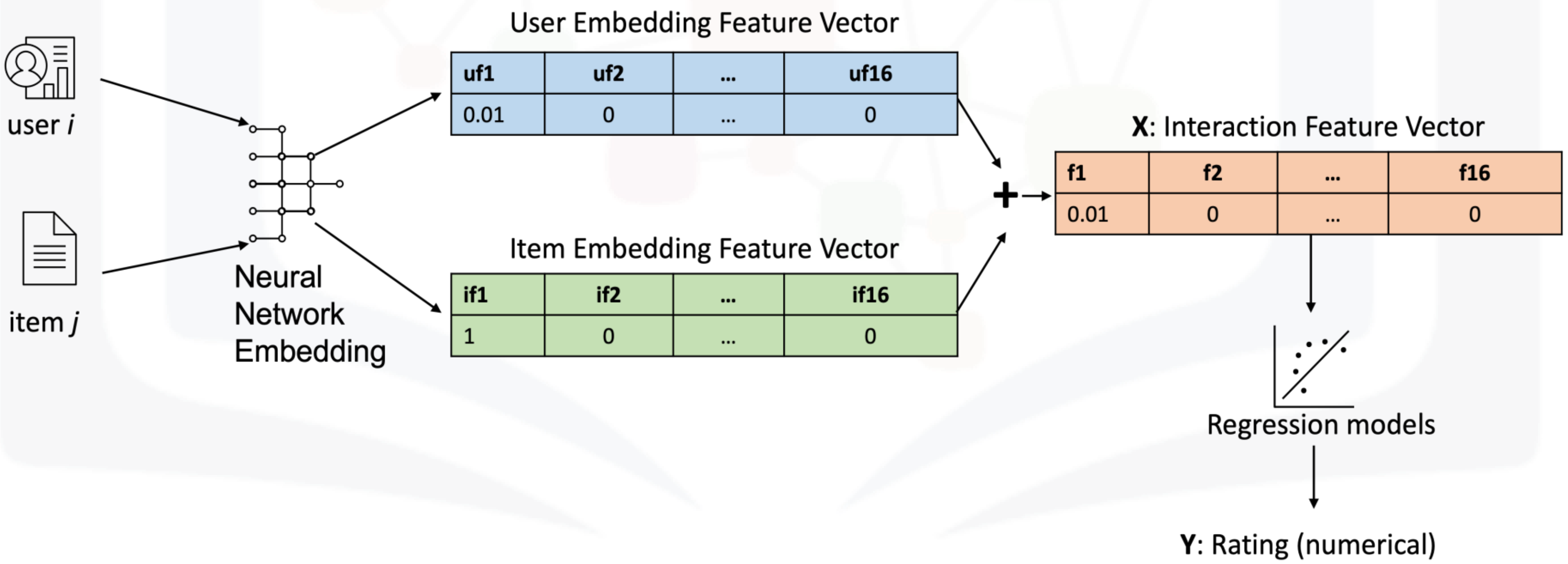
Classification Models

- Logistic Regression
- Random Forest Classification
- Bagging Classification

User Based Collaborative Filtering

Linear Regression : Overview

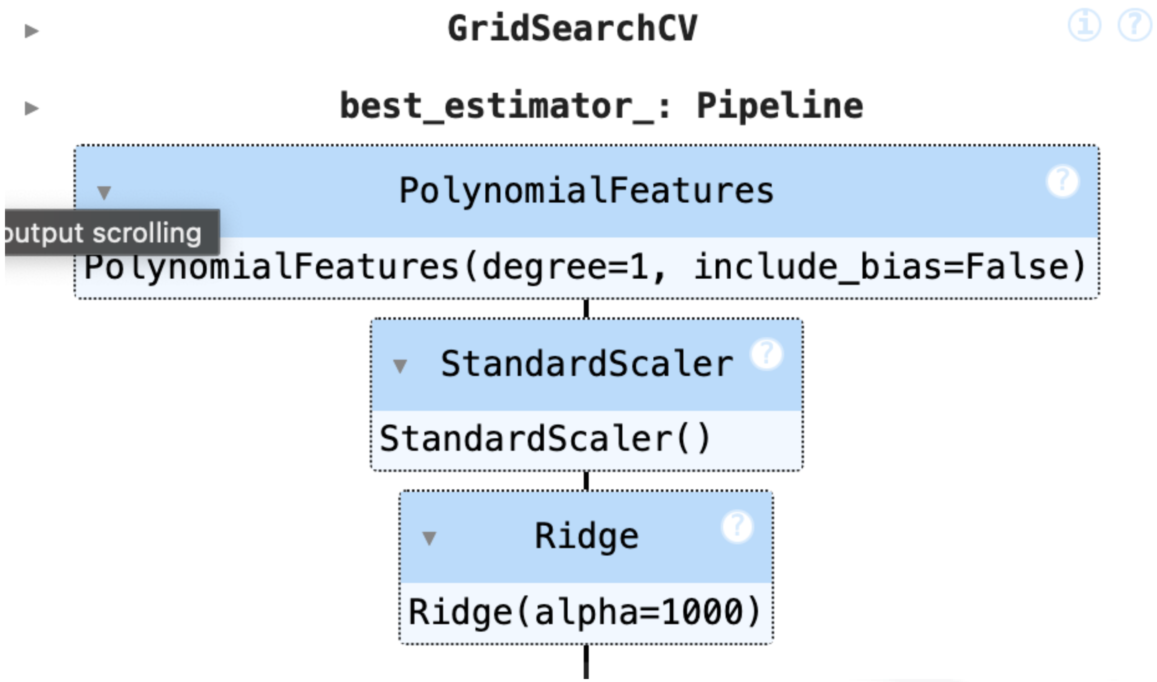
- We build a linear regression model to predict course rating
- User-course matrix is decomposed into two embedding feature vectors using neural network embedding
- These embedded features vectors are then combined to form a interaction feature network that ultimately feed into various regression models.



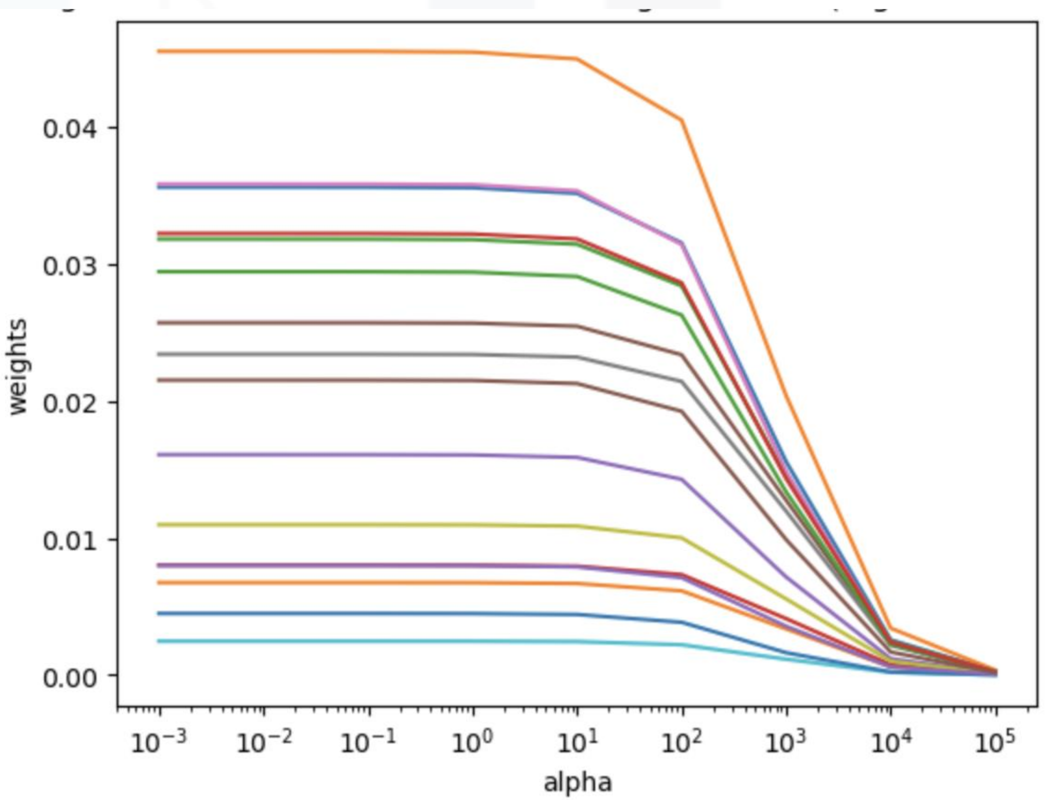
User Based Collaborative Filtering

Ridge Regression : Hyper-parameter Tuning

Hyper parameter Tuning and Cross validation done performed using GridSearchCV



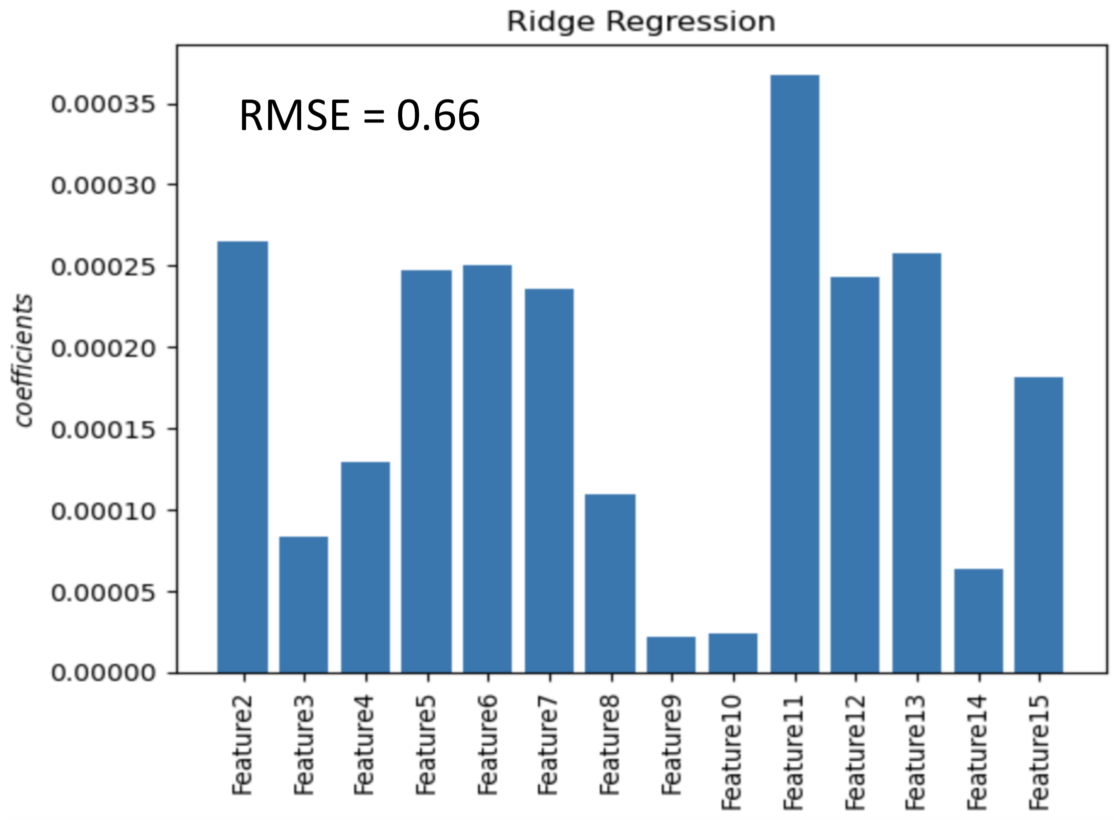
Optimum alpha found to be very large!
Alpha = 10000 !



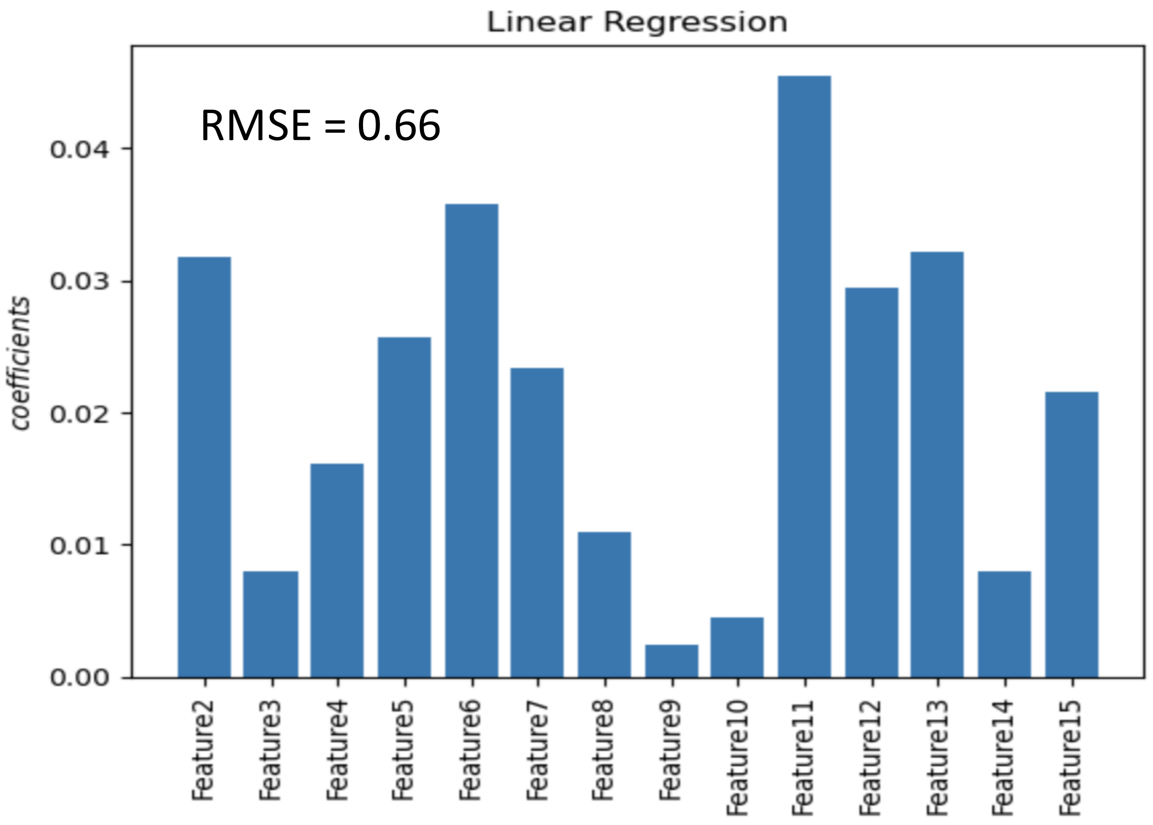
User Based Collaborative Filtering

Model Comparison : Basic Linear Regression & Ridge Regression

Result : Linear and Ridge Regression models have the same RMSE

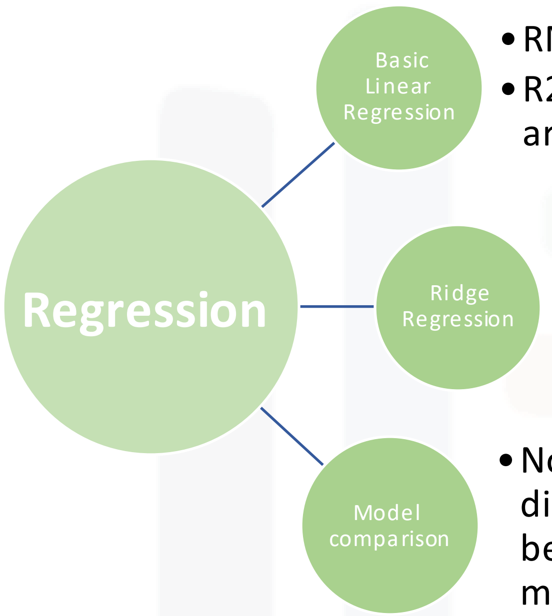


Result : Ridge Regression further shrinks all the coefficients to ~1e-4 !



User Based Collaborative Filtering

Regression : Observations



- RMSE =0.66
- R2 error around 1e-5

- RMSE = 0.66
- R2 error around 1e-5

- No significant difference between models

Regression Results :

1. Basic Linear Regression
   a. RMSE = 0.66
   b. R2 error ~ 1e-5 .. almost 0!
2. Ridge Regression
   a. RMSE = 0.66
   b. R2 error ~ 1e-5 .. almost 0!
3. Model Comparison
   a. NO change in RMSE
   b. Very low R2 error

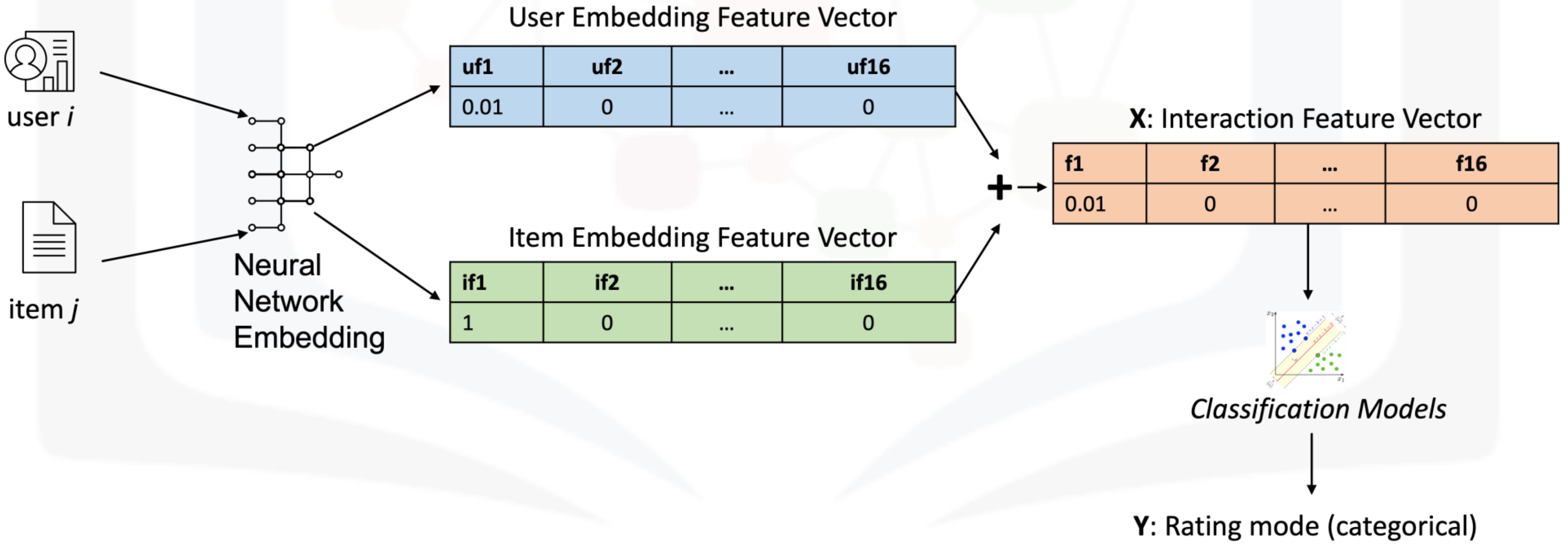
Inference :

Our target variable is discrete comprising of a rating of 3 (not good), 4(good) and 5(extremely good). Such a data is better suited to a clustering or a classification model and not a regression model which assumes a continuous target variable. This is why the regression models, linear, ridge and lasso (not discussed here), display poor R2 scores and imply no learning!

User Based Collaborative Filtering

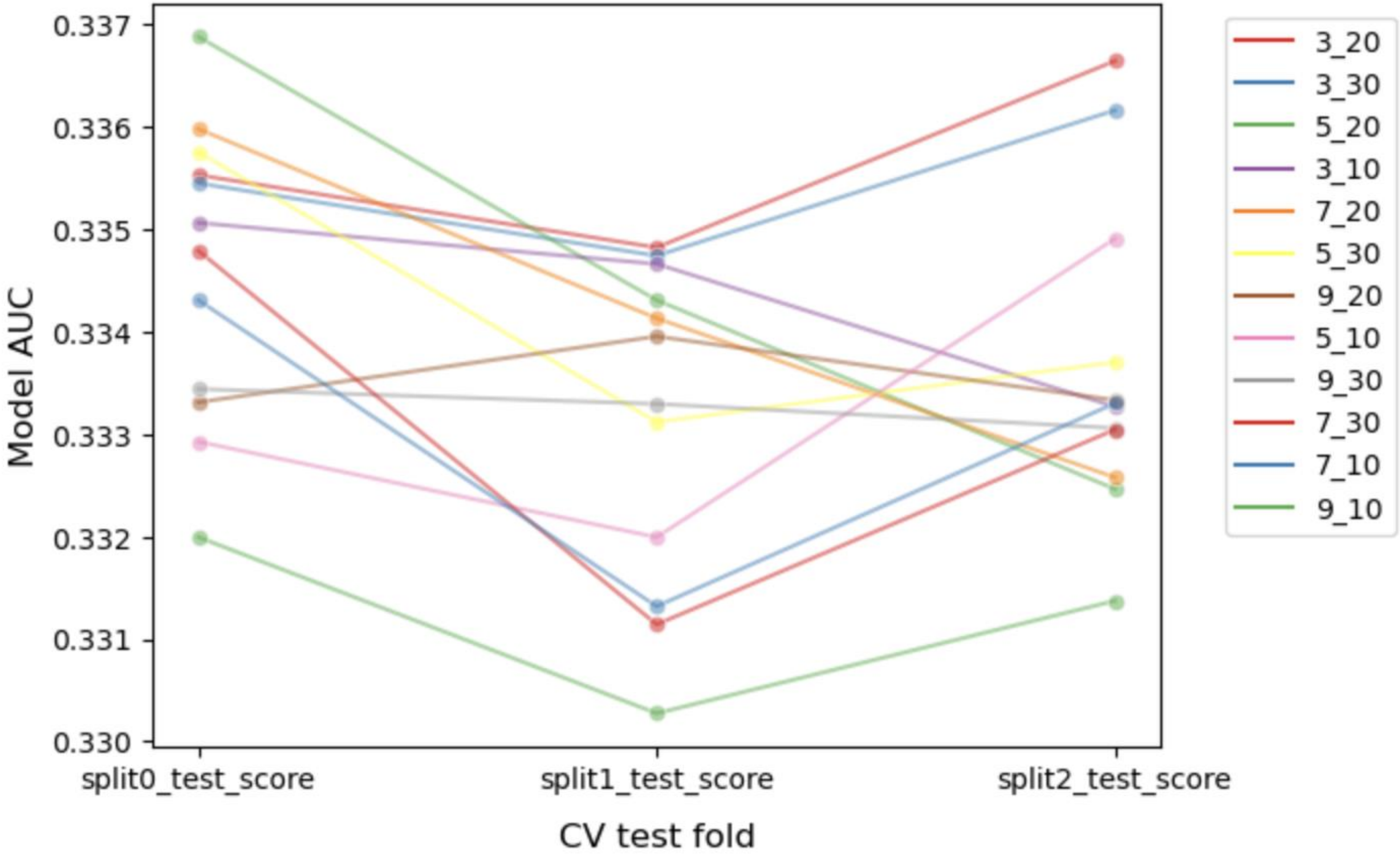
CLASSIFICATION : Overview

- We built 3 classification models to predict course rating- Logistic Regression, Random Forest & Bagging.
- The user-course matrix is decomposed into two embedding feature vectors using neural network embedding
- These embedded features vectors are then combined to form a interaction feature network that ultimately feed into various classification models.



User Based Collaborative Filtering

CLASSIFICATION : Random Forest Hyperparameter Tuning



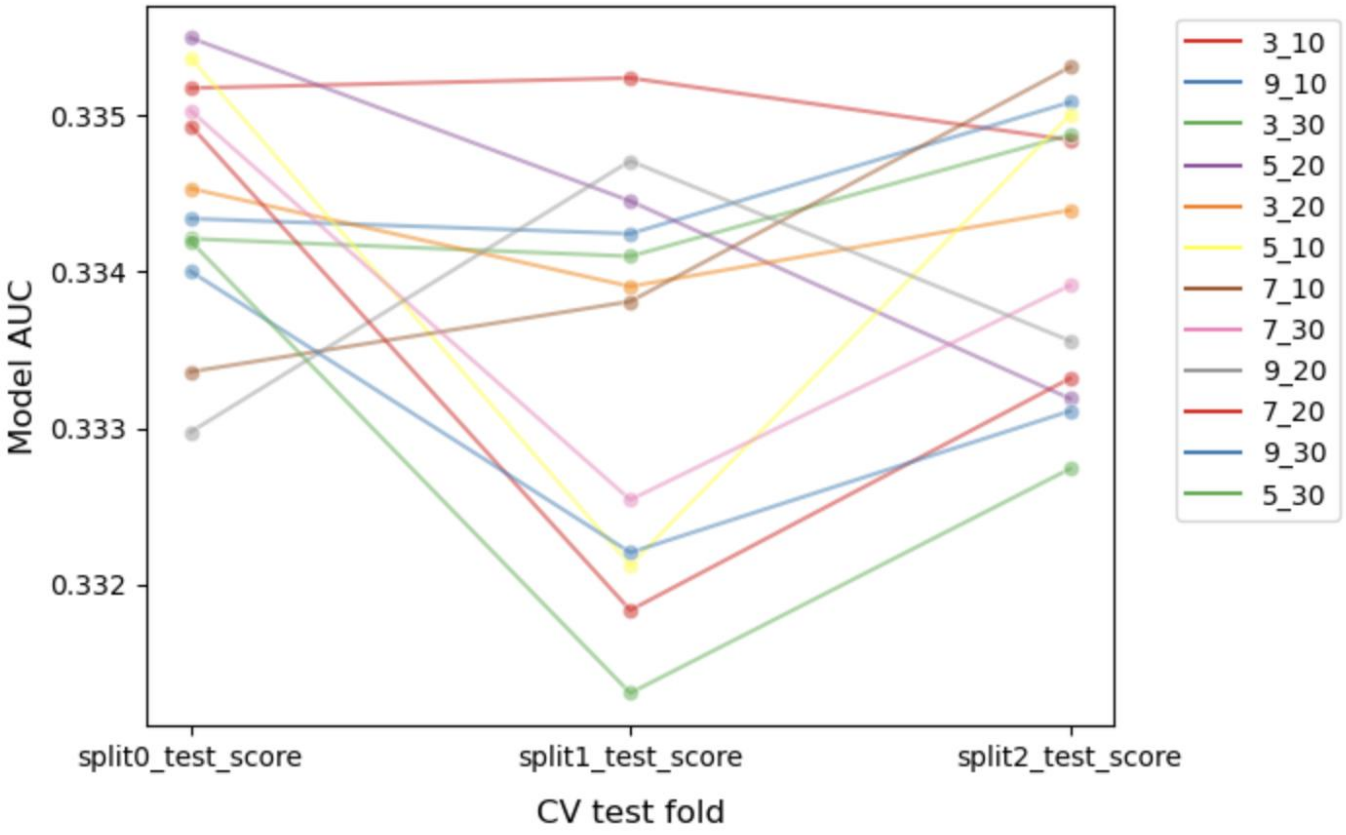
1. The following hyper parameters were studies via Grid Search Cross Validation (GridSearchCV) :
   - max_depth
   - n_estimators

2. The AUC curves here indicate that the highest score belongs to the set of hyperparameters
   - depth = 3
   - n_estimators = 20)

User Based Collaborative Filtering

CLASSIFICATION :Bagging Classifier Hyperparameter Tuning

1. The following hyper parameters were studies via Grid Search Cross Validation (GridSearchCV) :
    - max_depth
    - n_estimators



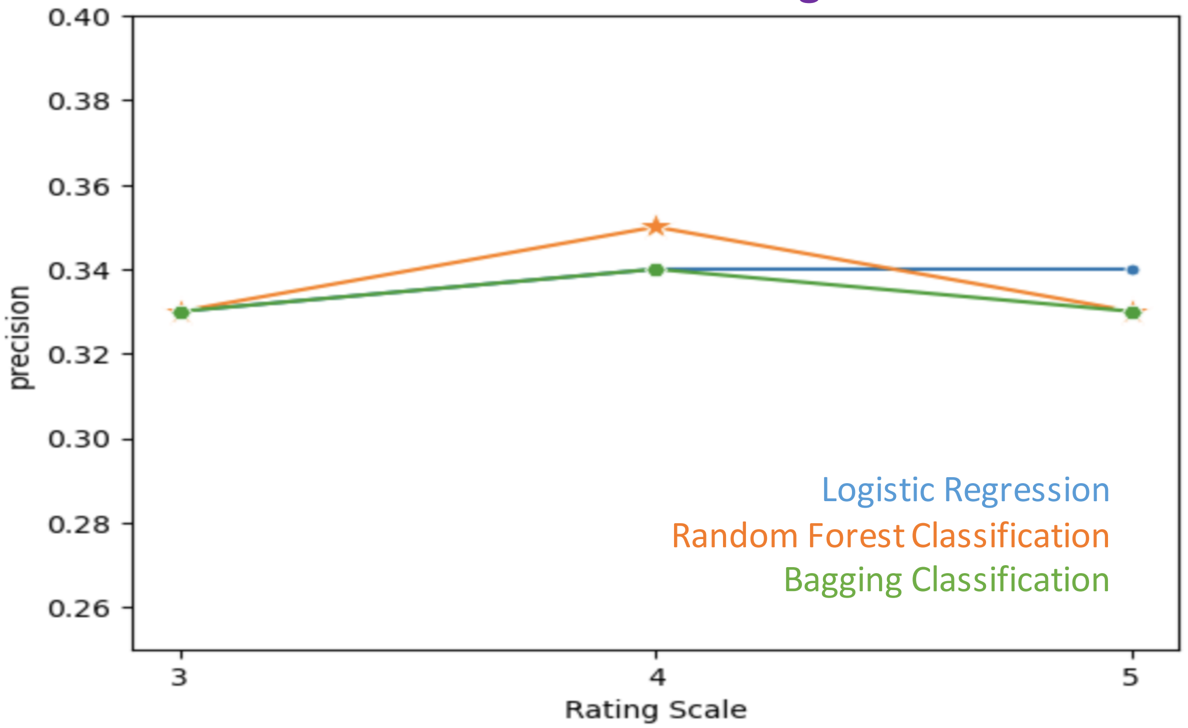
2. The AUC curves here indicate that the highest score belongs to the set of hyperparameters
    - depth = 3
    - n_estimators = 10)

User Based Collaborative Filtering

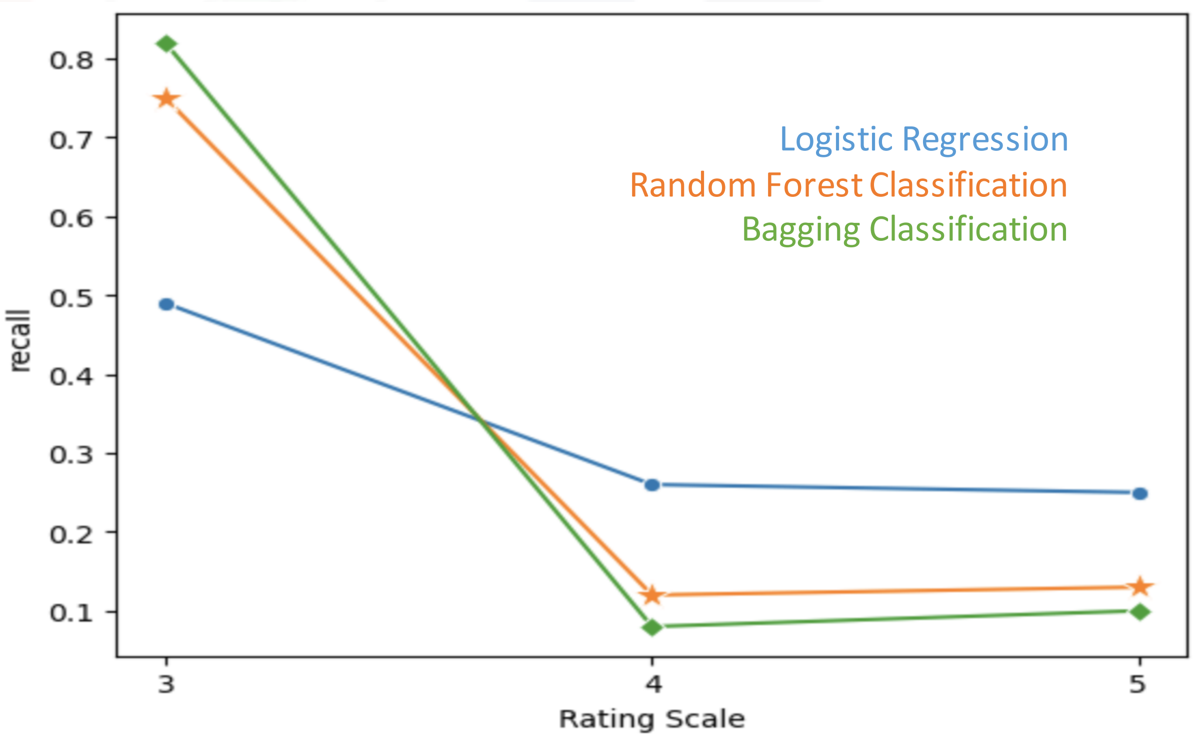
CLASSIFICATION : Precision& Recall Scores

1. Accuracy score is constant across all three rating scales and across all three models.
2. The ensemble models, Random Forest and Bagging, show better (higher) recall values for a rating scale of 3 but worse recall for rating scale of 4 and 5.
3. Model quality moderate

Precision Score Vs Rating Scale



Recall Score Vs Rating Scale



User Based Collaborative Filtering

CLASSIFICATION : Observations

Classification

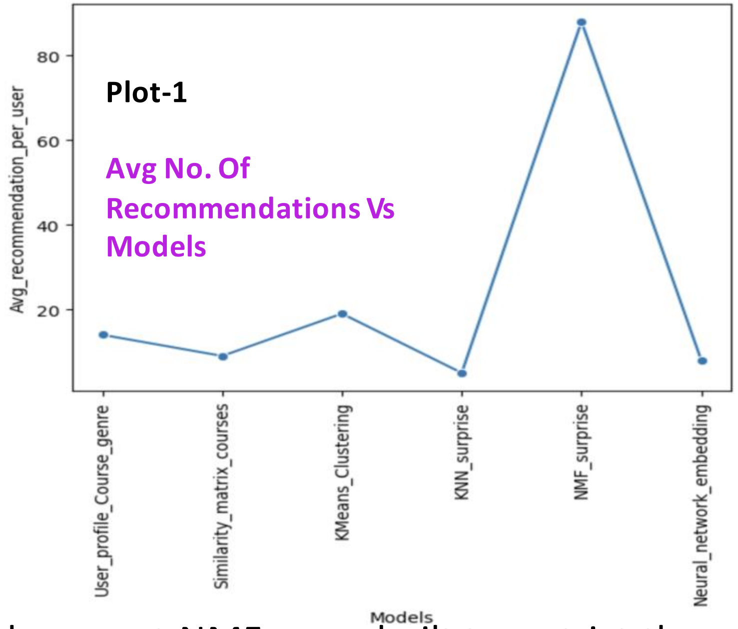
- Logistic Regression
  - accuracy=0.33
- Random Forest Classification
  - accuracy=0.33
- Bagging
  - accuracy=0.33
- Insights
  - No significant difference between models

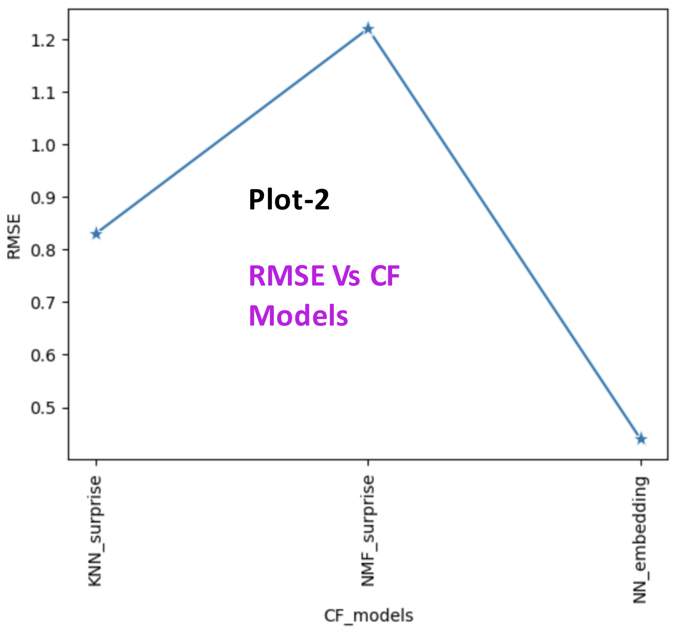
1. Precision value is constant across all three rating scales and across all three models.
2. The ensemble models, Random Forest and Bagging, show better (higher) recall values for a rating scale of 3 but worse recall for rating scale of 4 and 5.
3. Model quality moderate

Classification Model	Rating Scale	Precision	Recall
Logistic Regression	**3**	0.33	0.49
	4	0.34	0.26
	5	0.34	0.25
Random Forest	**3**	0.33	0.75
	4	0.35	0.12
	5	0.33	0.13
Bagging	**3**	0.33	0.77
	4	0.34	0.11
	5	0.33	0.12

Results & Summary

Observations





All models, except NMF, were built to restrict the average number of course recommendations to each user to less than 20 (see plot-1)

- Despite a very strict threshold, the NMF model still predicted extremely large number of courses to each user and was unable to get more specific.
- This is also consistent with the highest RMSE value for NMF compared to other CF (collaborative filtering) models (plot-2)

Results & Summary
Observations

1. **EDA**
   - The course enrollment data suggests that number of courses recommended to a user must be no greatrer than 20.
   - The most popular topics/genres from enrolled courses are : Data Science and Data related.

2. **Computation Time & RMSE**
   - Content Based Filtering models are **much faster** compared to User Based Collaborative Filtering models.
   - Models within Content Based Filtering showed no significant difference in run time. However, among the CF models, the Neural Network model with latent space embeddings took the least amount of run time.
   - The NN model also had the lowest RSME compared to the two other CF model, namely KNN and NMF.

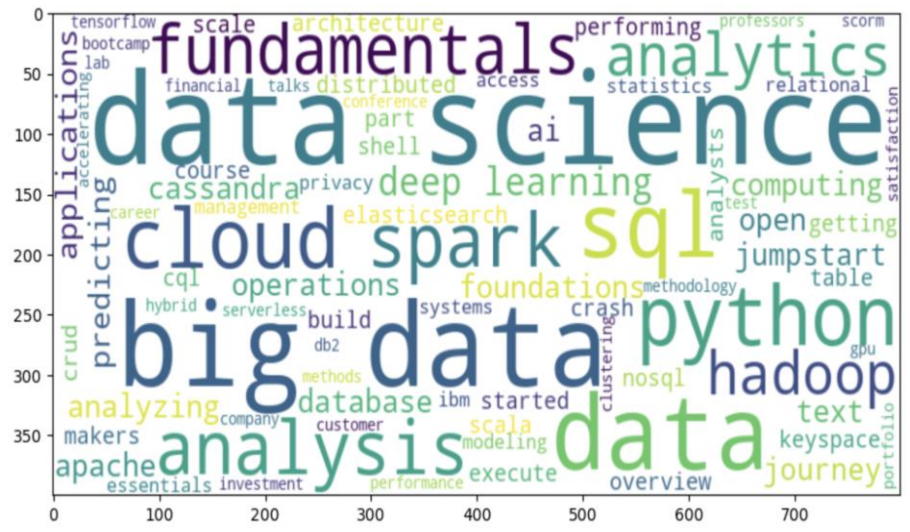
3. **Average number of recommendation per user**
   - All models except NMF were able to restrict the average number of recommendations to below 20 with a reasonable and modest threshold.

4. **Top 10 recommendations across all users**
   - Interestingly, the top 10 recommended courses across all users seems to be heavily model-dependent.

5. **Regression models (non-classification) performed very poorly** due to non-availability of a continuous set of ratget variable.

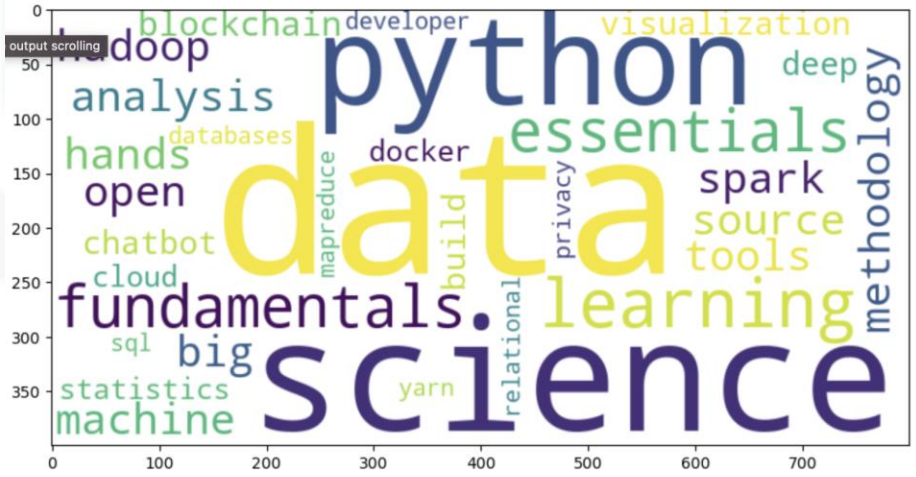
Innovative Insights

Recommended



The recommended courses seem to point to users' interest into more **specific topics** like big data, cloud, spark etc.

Enrolled



The enrolled courses seem to point to users' interest into more **general** and fundamental topics like data science, python etc

Conclusion

- The course recommendations are model dependent!
- More detailed models need to be built to get better and more consistent prections across all models.
- Regression models like linear regression and Ridge etc are not suitable for this course recommender system. The low R2 errors indicate no learning. This should not be surprising as non-classification regression models require a continuous target variable while our RS dataset as a discrete set of target variable (aka course ratings of 3, 4 or 5)