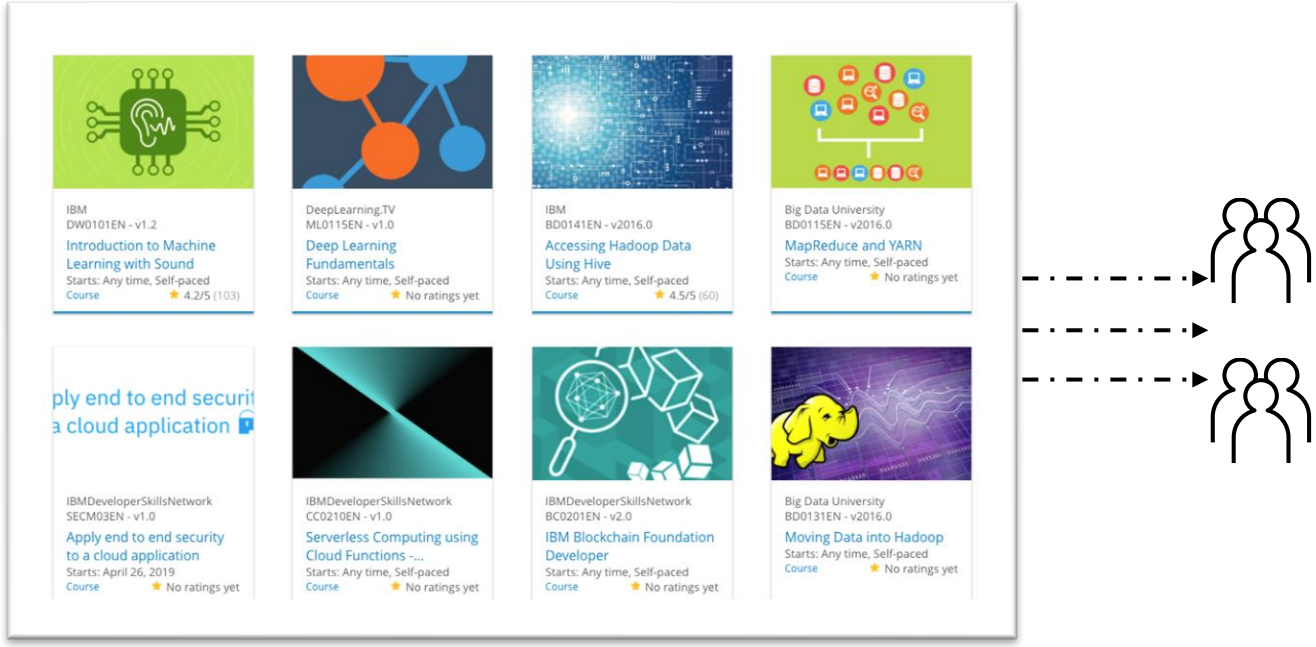
Build a Personalized Online Course Recommender System with Machine Learning

Muhammad Saad
25 January 2023



Outline

- Introduction
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

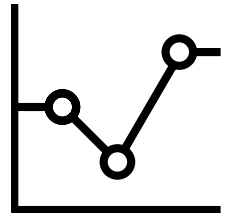
Introduction

- AI Training Room is a successful Massive Open Online Courses (MOOCS) startup.
- Learners across the world can learn leading technologies.
- Courses focus on Machine Learning, Data Science, Cloud, App Development etc.

Introduction

- As Company is growing rapidly, focus is to improve learning experience.
- This can be done by building a Recommender System Project.
- A Recommender System Project suggests or recommends additional products to customers.
- Will help learners to quickly find new interesting courses and better paving their learning paths.
- Highlighting Important Steps while Building a Recommender System.
- Observing Outcomes from different Machine Learning techniques.

Exploratory Data Analysis

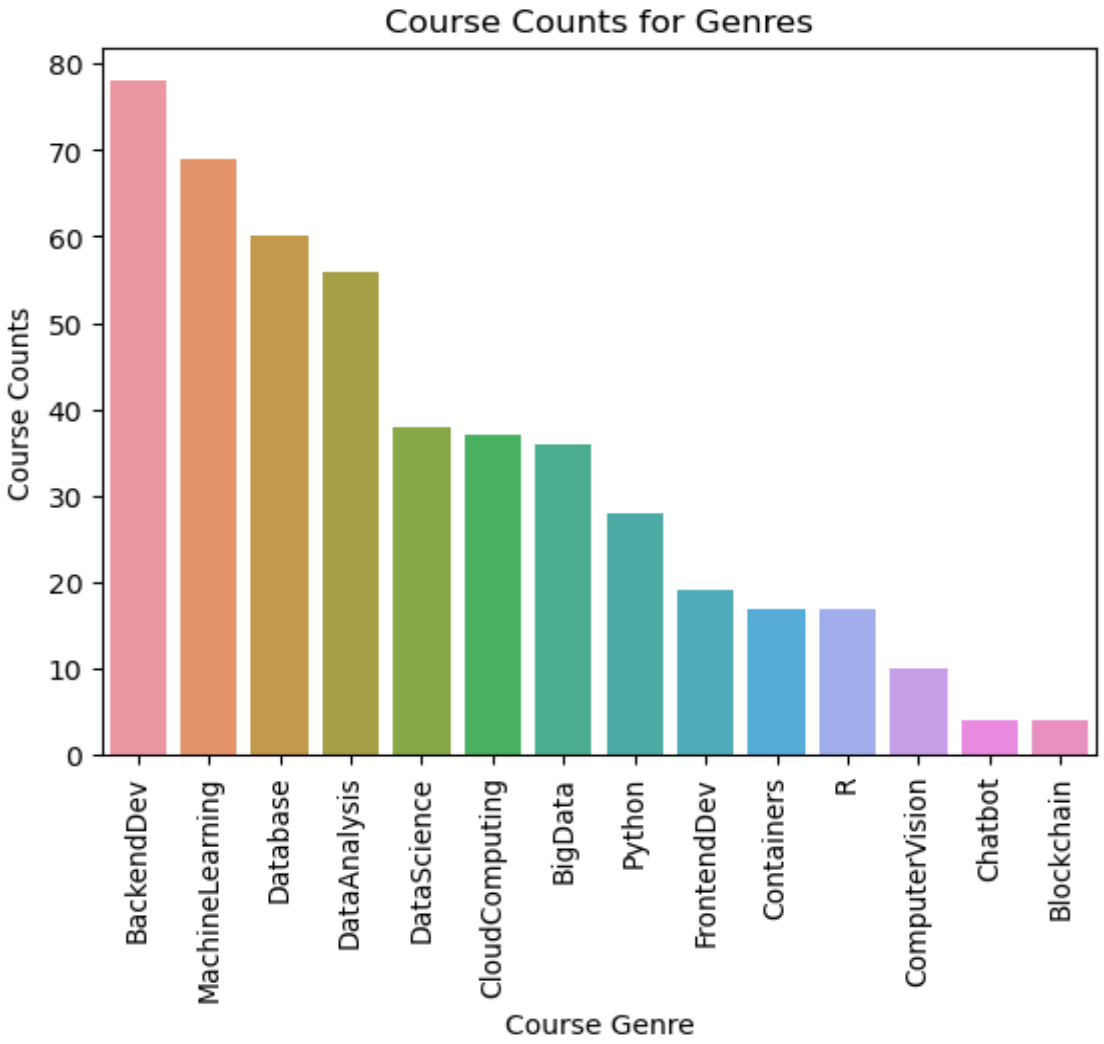


Exploratory Data Analysis

- Transforming Data to apply Machine Learning Techniques.
- To Draw insights such as:
  - Number of Course Genre's.
  - Number of Enrollments in Courses.
  - Most Popular Courses.
  - Word appearance in Course Titles to determine common genre's and topic.

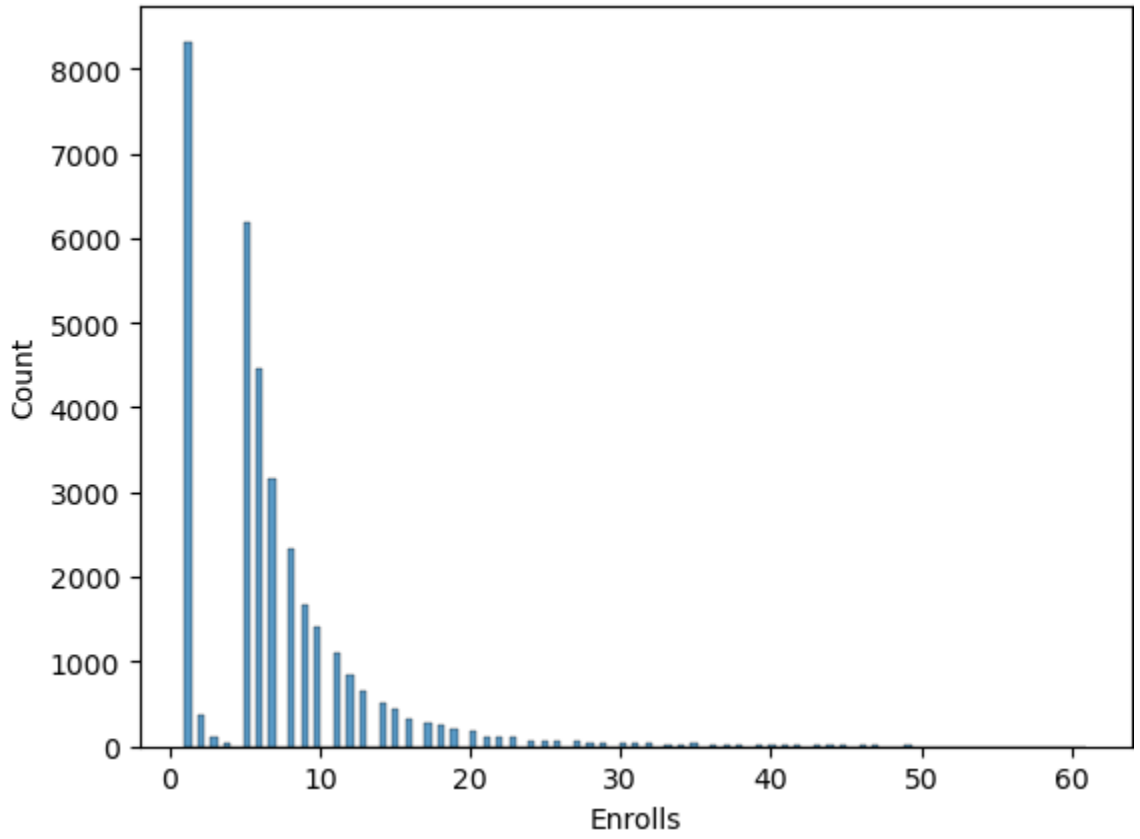
Course Counts Per Genre

- Total of 14 Course Genres or Topics.
- Visualization of Number of Courses for each genre.
- Backend Developer Genre has the most courses (78).
- Chatbot and Blockchain genres have the least courses (4 each)



Course Enrollment Distribution

- User Rating Counts Plot
- *Enrolls*: Users enrolled in course(s)
- *Count*: Total number of enrolled courses for each user.



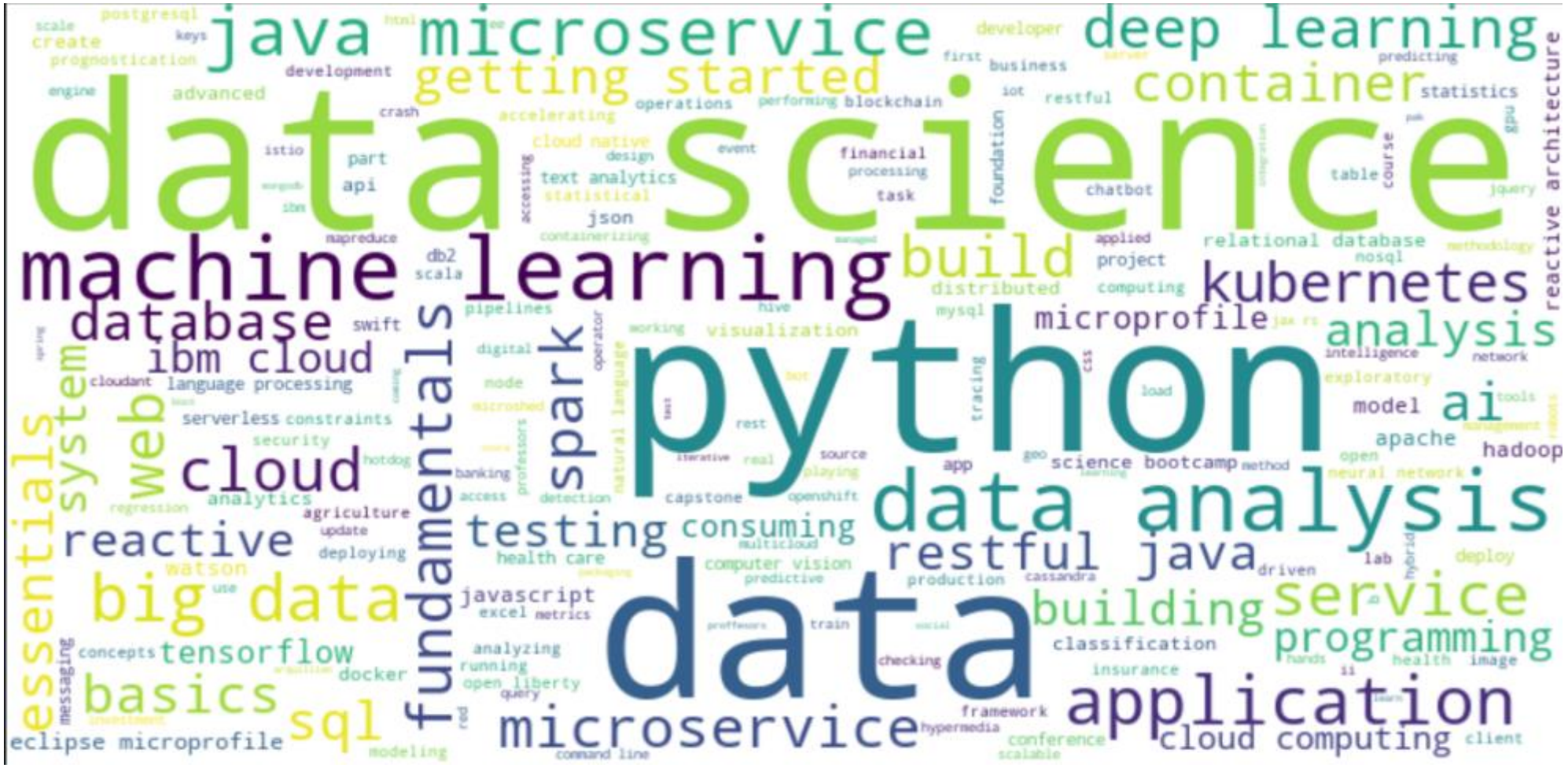
20 Most Popular Courses

- List of top 20 courses enrolled by users.
- These courses contribute to 63.3% of total users.
- Data Science and related tools and software are common among them.
- Most of the courses are fundamental or beginners level.

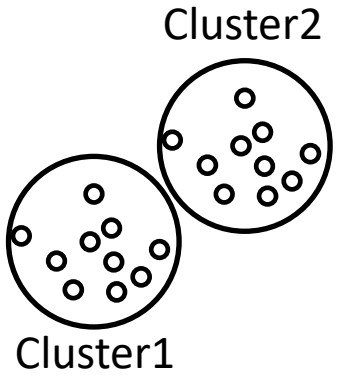
TITLE	Enrolls
python for data science	14936
introduction to data science	14477
big data 101	13291
hadoop 101	10599
data analysis with python	8303
data science methodology	7719
machine learning with python	7644
spark fundamentals i	7551
data science hands on with open source tools	7199
blockchain essentials	6719
data visualization with python	6709
deep learning 101	6323
build your own chatbot	5512
r for data science	5237
statistics 101	5015
introduction to cloud	4983
docker essentials a developer introduction	4480
sql and relational databases 101	3697
mapreduce and yarn	3670
data privacy fundamentals	3624

Word Cloud of Course Titles

- Graphical Representation of word frequency in all courses.
- Data Science, Machine Learning and Python appear the most in courses.



Content-based Recommender System using Unsupervised Learning

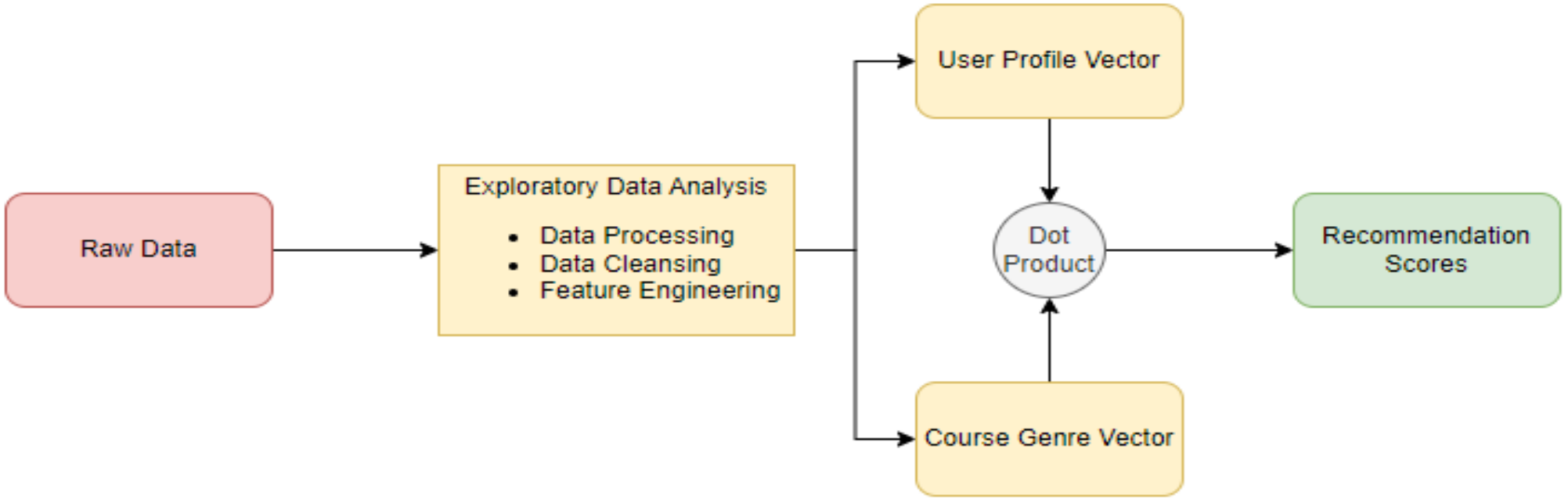


Content-Based Recommender System Using User Profile and Course Genres

- Most common type of Content-Based Recommender System.
- Based on:
  - User Profile
  - User's Preferences and Tastes
  - Clicks and likes on different items

Flowchart of Content-Based Recommender System using User Profile and Course Genres

- Features obtained after EDA.
- Constructing User Profile Vector and Course Genre Vector.
- Dot Product resulting in Recommendation Scores.



Evaluation Results of User Profile-Based Recommender System

- np.dot() from numpy Package in Python.
- Threshold = 10 to display recommended courses with high scores.
- For and if/else conditions to compare the scores with threshold.

On average, about 3 new courses have been recommended per user

Top-10 commonly recommended courses across all users:

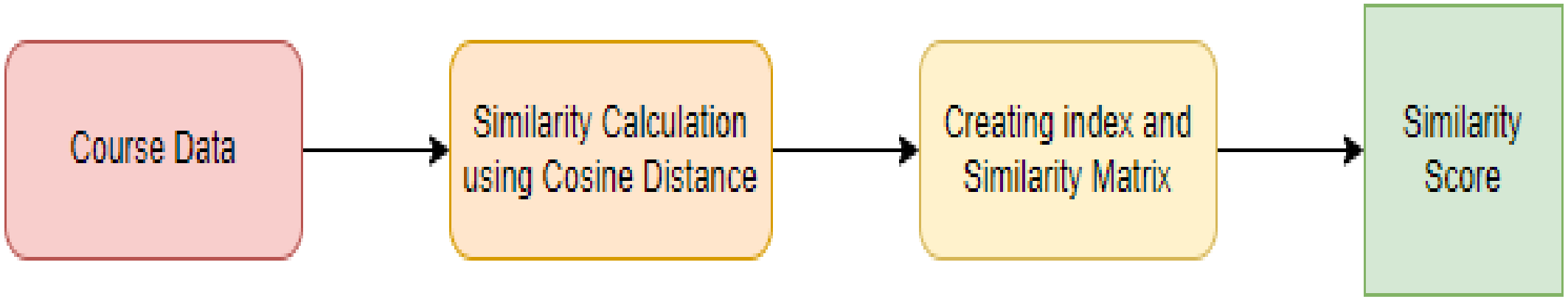
TITLE
python for data science
introduction to data science
big data 101
hadoop 101
data analysis with python
data science methodology
machine learning with python
spark fundamentals i
data science hands on with open source tools
blockchain essentials

Content-Based Recommender System Using Course Similarity

- Similarity between Courses from Similarity Matrix.
- Using Similarity Matrix to Recommend New Courses.

Flowchart of Content-Based Recommender System using Course Similarity

- Creating Similarity Data Frame.
- Using cosine distance.
- Index Mapping between dictionaries to calculate score.



Evaluation Results of Course Similarity Based Recommender System

- Cosine() from scipy Package for Similarity Distance .
- Threshold = 0.7 out of 1 to display recommended courses with high scores.
- Creating Function for Index to ID Mapping using “Bag of Words”.
- Sim_Matrix[#index 1][#index 2] returns Similarity Score.

On average, about 8 new courses have been recommended per user

Top-10 commonly recommended courses across all users:

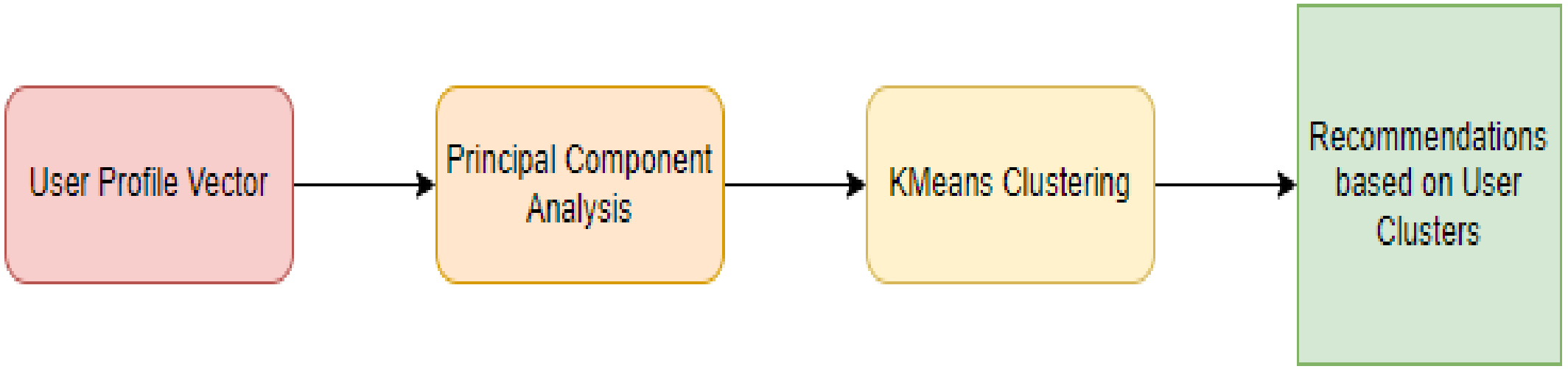
TITLE
python for data science
introduction to data science
big data 101
hadoop 101
data analysis with python
data science methodology
machine learning with python
spark fundamentals i
data science hands on with open source tools
blockchain essentials

Clustering-Based Recommender System

- Dimentionality Reduction on Course Data Features:
  - Decreases Complexity
  - Increases Accuracy
- Creating User Clusters
- Recommending Courses according to Clusters

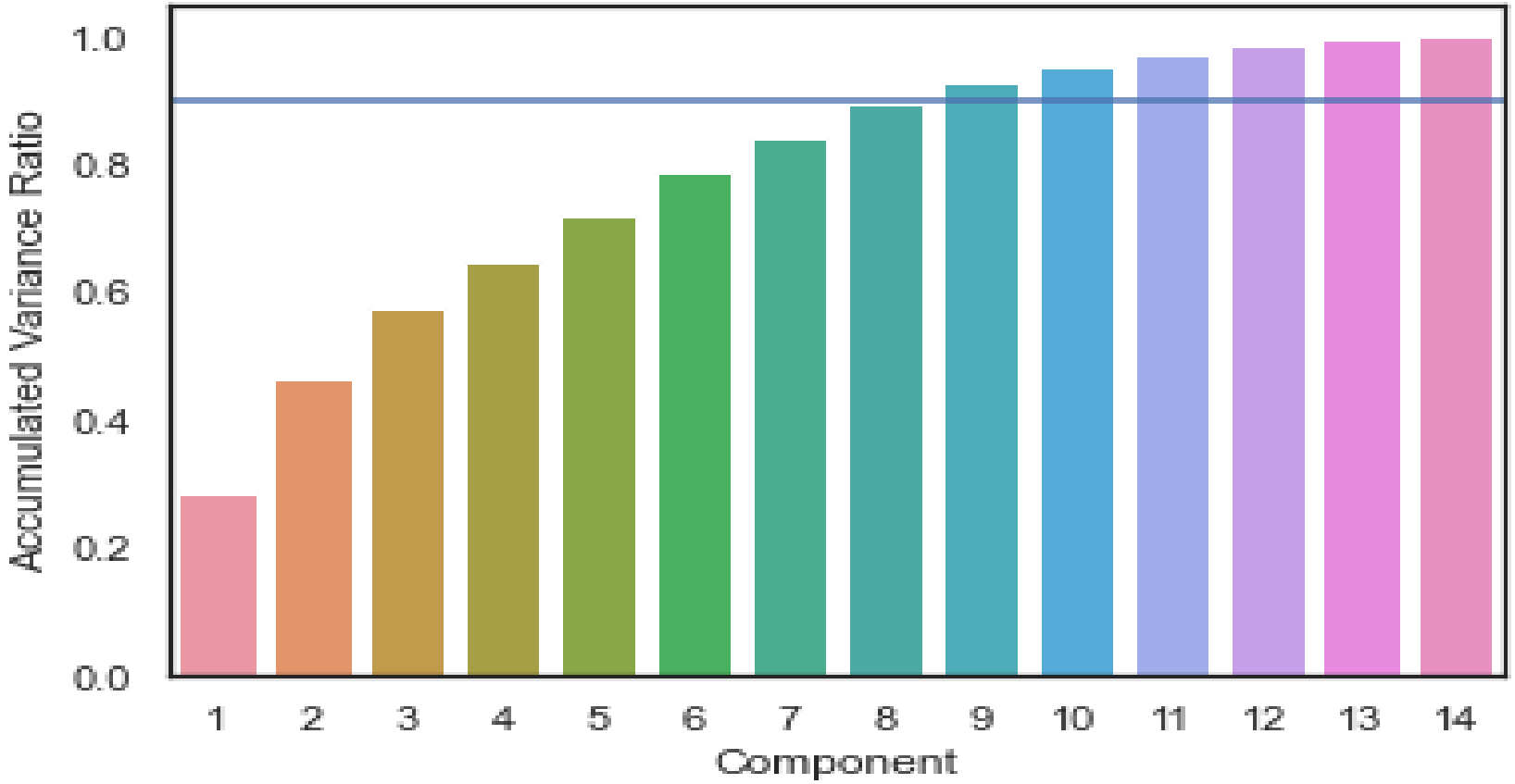
Flowchart of Clustering-Based Recommender System

- Dimentionality Reduction
- Clustering Technique
- Defining users in terms of Clusters



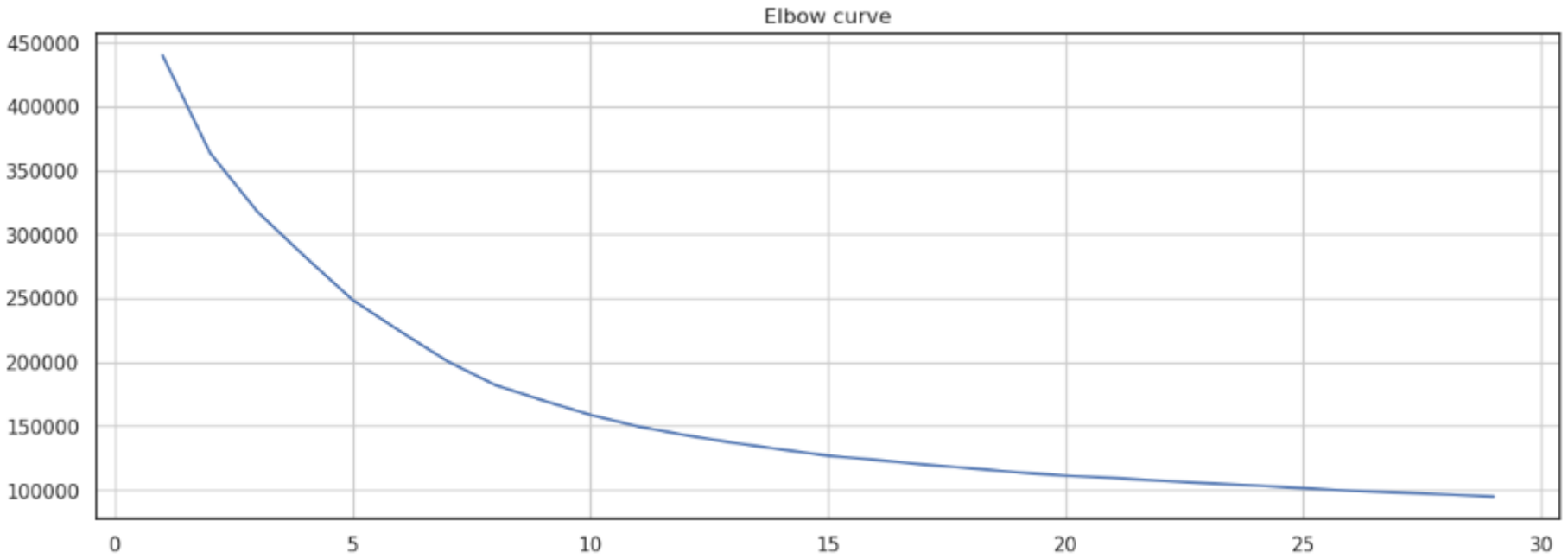
Principal Component Analysis

- Graph shows Variance Ratio of PCA components.
- Threshold = 0.9 to find appropriate number of components



KMeans Clustering

- Elbow Curve graph to find appropriate number of clusters.
- No change in graph after 25 clusters (Elbow).



Evaluation Results of Clustering-Based Recommender System

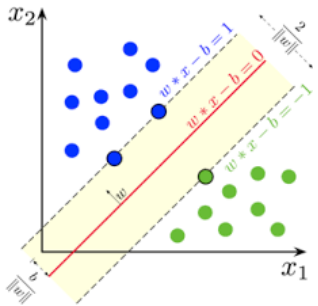
- PCA(n_components=9)
- KMeans(n_clusters = 25)

On average, about 4 new courses have been recommended per user

Top-10 commonly recommended courses across all users:

TITLE
python for data science
introduction to data science
big data 101
hadoop 101
data analysis with python
data science methodology
machine learning with python
spark fundamentals i
data science hands on with open source tools
blockchain essentials

Collaborative-filtering Recommender System using Supervised Learning

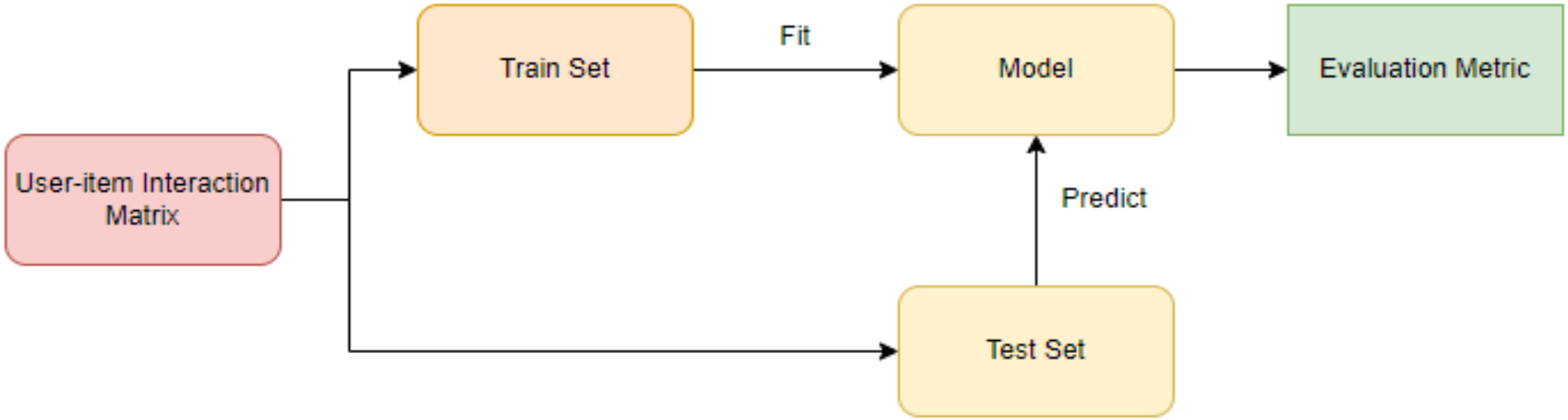


Collaborative-Filtering Recommender System

- Commonly used Recommendation Algorithm
- Based on similarity between users (neighborhood) or items

KNN-Based Recommender System

- Splitting into Training and Test Sets
- Fitting Training Set
- Prediction using Test Set
- Metric Score to Evaluate Performance



Functions & Hyper-Parameters for KNN-Based Recommender System

- Surprise Python Library
- Built for Recommendation Algorithms
- train_test_split(data, test_size=0.3) #Test Set is 30% and Train Set is 70% of Data
- KNNBasic( ) for KNN Technique
- Root Mean Square Error for Evaluating Prediction
- RMSE Score = *0.98*

NMF-Based Recommender System

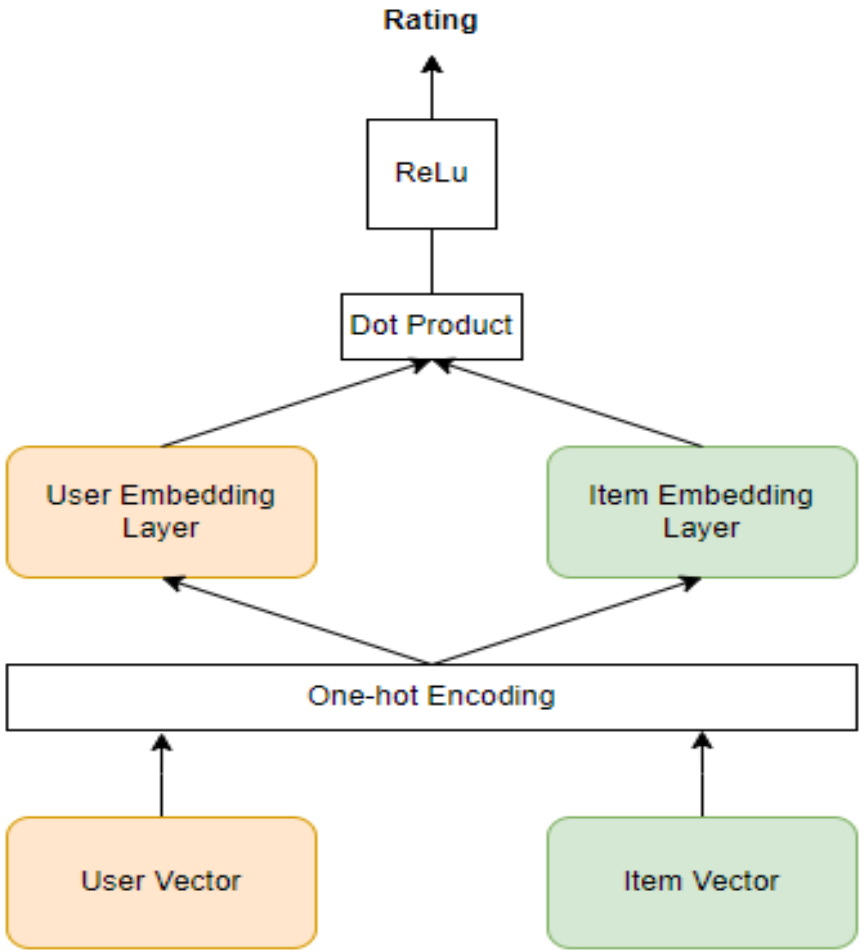
- KNN is memory-based
- Hard to load KNN Matrix in RAM
- Non-negative Matrix Factorization: A Dimensionality Reduction Technique
- Decompose big User-Item Matric into smaller User and Item Matrices

Functions & Hyper-Parameters for NMF-Based Recommender System

- Surprise Python Library
- NMF( ) for KNN Technique
- RMSE Score = *0.20*

Neural Network Embedding Based Recommender System

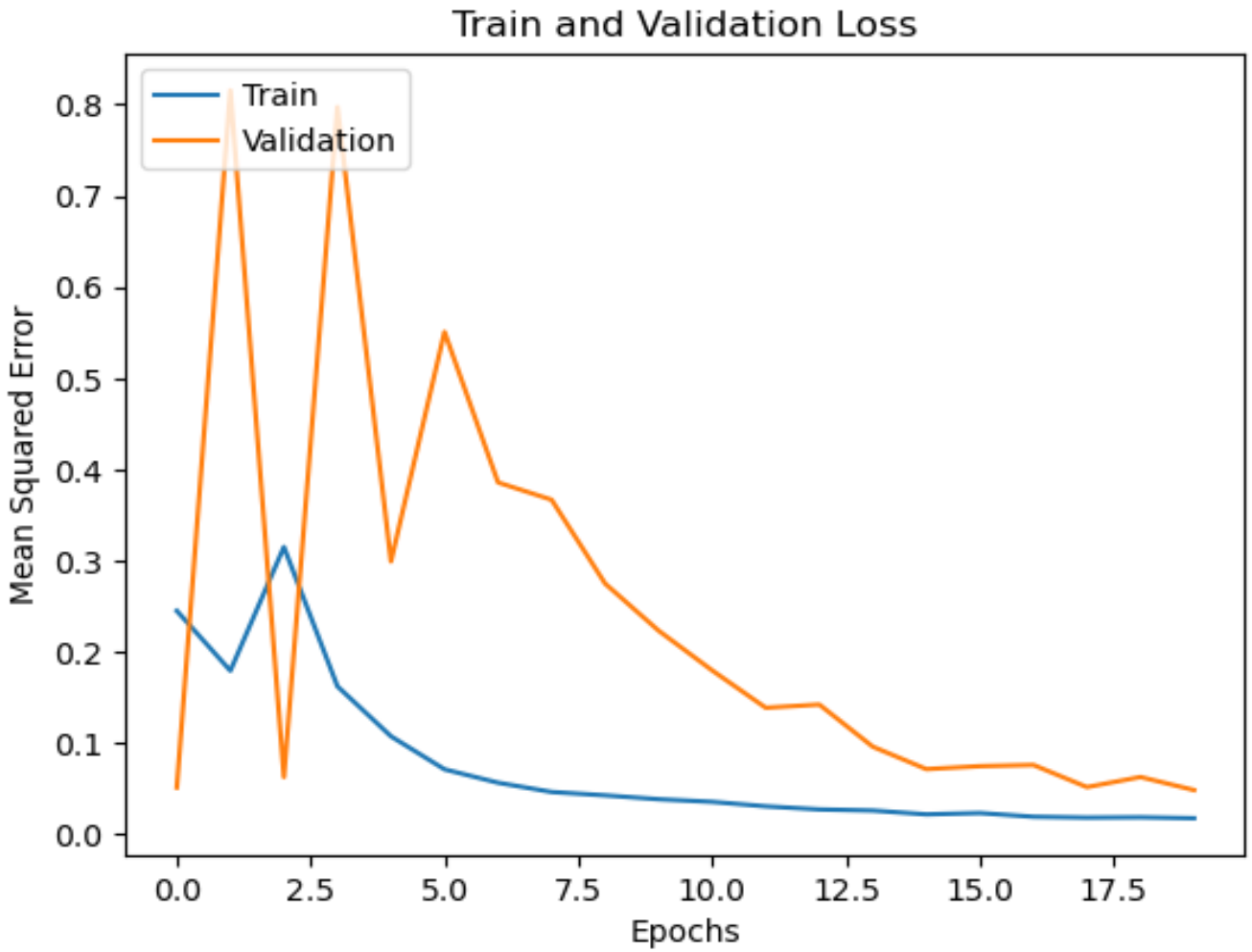
- Previous Techniques required building feature vectors
- Time-Consuming
- Neural Network Embedding after NMF
- Good at learning patterns and extract latent features



Functions & Hyper-Parameters for Neural Network Embedding Based Recommender System

- TensorFlow
- Performed One-Hot Encoding
- Activation Function: ReLu
- Epochs = *20*
- Trained Model Mean Square Error (MSE): *0.26*

Evaluation of Neural Network Embedding Based Recommender System

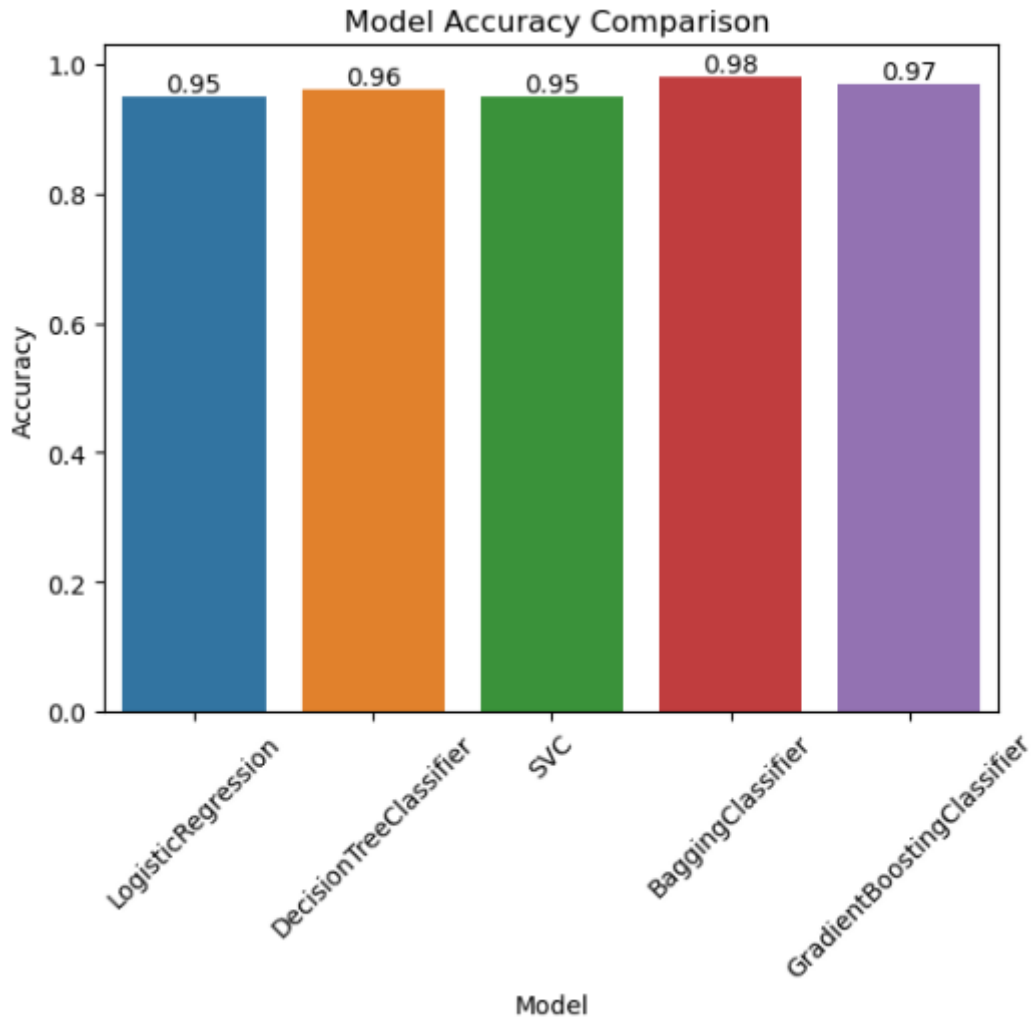


Collaborative-Filtering Recommendation System using Embedding Features

Classification Models:

- Logistic Regression
- Decision Tree Classifier
- SVC
- Bagging Classifier
- Gradient Boosting Classifier

Performance Comparison of Collaborative-Filtering Models



Conclusions

- Data Science and Machine Learning related courses contribute the most.
- High ratings of Beginner Level courses suggest that users either focus on gaining new skills or switch career path.
- User profile and course similarity helped determine recommended courses.
- Improved performance with NMF and embedding layers for Collaborative-Filtering.
- All Classification models performed nearly same. So any model is preferred.
- Performance can be improved further by experimenting hyperparameters for neural networks i.e. changing activation function or increasing number of epochs.

Appendix

- https://github.com/saadbinsohail2/portfolio