

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**LUẬN VĂN TỐT NGHIỆP**

**Nghiên cứu, Phát triển công cụ  
Phân tích dữ liệu cho đại học  
và trực quan hóa**

**HỘI ĐỒNG:** **HỆ THỐNG & MẠNG 2**  
**GVHD:** PGS.TS. Thoại Nam  
**GVPB:** Lê Thanh Vân

**SINH VIÊN THỰC HIỆN:**

- |                  |         |
|------------------|---------|
| 1. Đỗ Thành Phát | 1512400 |
| 2. Mai Lê Thông  | 1513293 |

# LỜI CAM KẾT

Chúng tôi xin cam kết toàn bộ luận văn là kết quả nghiên cứu được thực hiện bởi chính nhóm chúng tôi - Đỗ Thành Phát và Mai Lê Thông, dưới sự hướng dẫn của PGS.TS Thoại Nam. Toàn bộ những kết quả này có được dựa trên thực tế nghiên cứu của nhóm chúng tôi, hoàn toàn không sao chép từ bất kỳ một nguồn nào khác. Ngoài ra, toàn bộ các tài liệu mà chúng tôi trích dẫn và tham khảo cũng được liệt kê đầy đủ ở phần tham khảo cuối luận văn. Chúng tôi khẳng định những điều nói trên là đúng sự thật và xin chịu trách nhiệm hoàn toàn với những cam kết trên.

**Nhóm sinh viên**

# LỜI CẢM ƠN

Luận văn này là kết quả của một quá trình nghiên cứu nghiêm túc và hết mình của nhóm chúng tôi. Bên cạnh đó là sự giúp đỡ nhiệt tình của các thầy/cô đã không quản thời gian và công sức để giúp đỡ cho chúng tôi, để luận văn này đạt được những mục tiêu đặt ra và hoàn thành đúng tiến độ.

Trước tiên, chúng tôi xin gửi lời cảm ơn chân thành đến **PGS.TS Thoại Nam** - giảng viên, nguyên trưởng Khoa Khoa học và Kỹ thuật máy tính trường Đại học Bách Khoa - Đại học Quốc gia Tp.HCM - là người trực tiếp phụ trách giám sát và hướng dẫn đề tài luận văn tốt nghiệp này. Nhờ sự giúp đỡ cũng như hướng dẫn của PGS.TS Thoại Nam, chúng tôi luôn định hướng được hướng đi để giải quyết đề tài và cách giải quyết các khó khăn gặp phải.

Chúng tôi xin chân thành cảm ơn thầy **Chung Thành Minh**, đã hỗ trợ và hướng dẫn cho chúng tôi trong suốt quá trình thực hiện đề tài luận văn. Chúng tôi cũng xin cảm ơn các thầy/cô, anh/chị của **Phòng thí nghiệm tính toán hiệu năng cao (HPCC Lab)** thuộc Khoa Khoa học và Kỹ thuật máy tính trường Đại học Bách Khoa - Đại học Quốc gia Tp.HCM đã hỗ trợ và tạo điều kiện cho chúng tôi trong quá trình hoàn thành luận văn.

Lời cuối cùng, chúng tôi xin cảm ơn toàn thể các thầy/cô của trường Đại học Bách Khoa - Đại học Quốc gia Tp.HCM nói chung và các thầy/cô khoa Khoa Khoa học và Kỹ thuật máy tính nói riêng, đã giảng dạy và đem lại cho chúng tôi những kiến thức nền tảng vững chắc, góp phần quan trọng để chúng tôi hoàn thành đề tài luận văn này.

**Nhóm sinh viên**

# TÓM TẮT LUẬN VĂN

Ngày nay, Education Data Mining (EDM) đóng vai trò quan trọng trong ngành giáo dục Việt Nam, đặc biệt là giáo dục trong môi trường đại học - nơi chuẩn bị các kiến thức chuyên môn cần thiết cho sinh viên trên con đường sự nghiệp sau này. Trong các mảng nghiên cứu của lĩnh vực EDM, Prediction (dự đoán) và Recommendation (gợi ý) là hai mảng nghiên cứu được quan tâm nhiều trong thời gian gần đây. Nhiều giải thuật đã được sử dụng để dự đoán điểm của sinh viên ở các môn học - các giải thuật này đa số được xây dựng dựa trên các giải thuật về Machine Learning (học máy) và Recommender System (hệ thống gợi ý). Kết quả của việc dự đoán có thể được dùng để xác định sớm các sinh viên có khả năng đạt kết quả không tốt trong môn học hoặc hỗ trợ sinh viên trong việc lựa chọn các môn học tiếp theo.

Trong luận văn này, chúng tôi xây dựng một hệ thống dự đoán điểm số của sinh viên đại học dựa trên kết quả của các môn học đã học trước đó của sinh viên. Hệ thống dự đoán này chạy trên môi trường phân tán thông qua việc sử dụng Spark Cluster. Sinh viên có thể giao tiếp với hệ thống này thông qua một hệ thống web. Trong hệ thống dự đoán điểm, các giải thuật được sử dụng để xây dựng mô hình dự đoán điểm bao gồm Collaborative Filtering, Matrix Factorization, Association Rule và Restricted Boltzmann Machine. Luận văn này cũng sẽ so sánh hiệu quả của các giải thuật sử dụng dựa trên tập dữ liệu được cung cấp bởi Trường Đại Học Bách Khoa Thành phố Hồ Chí Minh với thông tin của hơn 60000 sinh viên từ 14 khoa trong các năm học 2006 đến 2017.

Kết quả cho thấy giữa các khoa có sự chênh lệch về mức nền trong độ chính xác của việc dự đoán - trong đó khoa *Bảo dưỡng công nghiệp* có độ chính xác dự đoán thấp nhất với  $RMSE \approx 2.03$  và khoa *Môi trường và Tài nguyên* có độ chính xác dự đoán cao nhất với  $RMSE \approx 1.61$  đối với phương pháp nền. Trong các phương pháp được đề xuất trên thì phương pháp Matrix Factorization với ràng buộc không âm cho ra kết quả tốt nhất trong hầu hết các khoa. Số lượng điểm 0 của các khoa ảnh hưởng nhiều đến độ chính xác của việc dự đoán. Cụ thể khi loại bỏ điểm 0, độ chính xác nền và độ chính xác của tất cả các phương pháp khác đều tăng.

# Mục lục

<b>LỜI CAM KẾT</b>	<b>1</b>
<b>LỜI CẢM ƠN</b>	<b>2</b>
<b>TÓM TẮT LUẬN VĂN</b>	<b>3</b>
<b>Chương 1 TỔNG QUAN</b>	<b>9</b>
1.1 Giới thiệu đề tài . . . . .	9
1.2 Lý do chọn đề tài . . . . .	10
1.3 Ý nghĩa đề tài . . . . .	11
1.4 Phạm vi và đối tượng nghiên cứu của đề tài . . . . .	11
1.5 Mục tiêu của đề tài . . . . .	11
1.6 Bố cục của luận văn . . . . .	12
<b>Chương 2 CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN</b>	<b>13</b>
2.1 Tổng quan về Educational Data Mining và các ứng dụng thực tiễn . . . . .	13
2.2 Các nghiên cứu nổi bật . . . . .	14
<b>Chương 3 KIẾN THỨC NỀN TẢNG</b>	<b>16</b>
3.1 Collaborative Filtering . . . . .	16
3.2 Matrix factorization . . . . .	20
3.3 Restricted Boltzmann Machine . . . . .	23
3.4 Khai phá luật kết hợp với FP-Growth . . . . .	29
3.5 Các công nghệ sử dụng . . . . .	32
<b>Chương 4 PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỀ XUẤT</b>	<b>39</b>
4.1 Đặc tả bài toán . . . . .	39
4.2 Tập dữ liệu đại học . . . . .	41
4.3 Mô hình giải quyết bài toán . . . . .	45
<b>Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ</b>	<b>58</b>
5.1 Môi trường thực nghiệm . . . . .	58
5.2 Kết quả . . . . .	61
<b>Chương 6 TỔNG KẾT</b>	<b>78</b>
6.1 Tổng kết . . . . .	78
6.2 Hướng phát triển trong tương lai . . . . .	79
<b>PHỤ LỤC</b>	<b>80</b>
<b>Tài liệu tham khảo</b>	<b>81</b>

# Danh mục hình ảnh

3.1	User-Based và Item-Based Collaborative Filtering [1] . . . . .	17
3.2	Kiến trúc của một RBM . . . . .	25
3.3	Gibbs Sampling . . . . .	26
3.4	RBM for Collaborative Filtering . . . . .	28
3.5	FP-Tree được xây dựng từ các transaction [2] . . . . .	31
3.6	Kiến trúc của Apache Spark . . . . .	34
3.7	RPC Flow . . . . .	37
3.8	Thrift Compiler . . . . .	38
3.9	Thrift network protocol stack . . . . .	38
4.1	Ví dụ về chương trình đào tạo của Khoa Khoa học và Kỹ thuật Máy tính - Đại học Bách Khoa . . . . .	40
4.2	Mô hình hóa công cụ phân tích dữ liệu đại học . . . . .	40
4.3	Hai module chính của công cụ phân tích dữ liệu đại học . . . . .	41
4.4	Trực quan về số lượng sinh viên giữa các khoa trong tập dữ liệu . . . . .	42
4.5	Phân bố điểm số trong tập dữ liệu . . . . .	45
4.6	Phân bố điểm khoa MT . . . . .	46
4.7	Phân bố điểm khoa BD . . . . .	46
4.8	Phân bố điểm khoa CK . . . . .	46
4.9	Phân bố điểm khoa DC . . . . .	46
4.10	Phân bố điểm khoa DD . . . . .	46
4.11	Phân bố điểm khoa GT . . . . .	46
4.12	Phân bố điểm khoa HC . . . . .	47
4.13	Phân bố điểm khoa MO . . . . .	47
4.14	Phân bố điểm khoa QL . . . . .	47
4.15	Phân bố điểm khoa UD . . . . .	47
4.16	Phân bố điểm khoa VL . . . . .	47
4.17	Phân bố điểm khoa VP . . . . .	47
4.18	Phân bố điểm khoa XD . . . . .	48
4.19	Phân bố điểm khoa PD . . . . .	48
4.20	Số lượng môn học trước và sau khi thực hiện mapping . . . . .	48
4.21	Item-based Collaborative Filtering on Item Factor Matrix of Matrix Factorization	54
4.22	Recommendation module . . . . .	56
5.1	Mô hình spark cluster được sử dụng để xây dựng công cụ phân tích dữ liệu đại học	59
5.2	Kết quả trường hợp Locality khoa MT . . . . .	63
5.3	Kết quả trường hợp Global khoa MT . . . . .	63
5.4	Kết quả trường hợp Locality khoa BD . . . . .	63
5.5	Kết quả trường hợp Global khoa BD . . . . .	63
5.6	Kết quả trường hợp Locality K12 khoa MT . . . . .	63
5.7	Kết quả trường hợp Locality K12 khoa BD . . . . .	63
5.8	Kết quả trường hợp Locality khoa CK . . . . .	64

5.9	Kết quả trường hợp Global khoa CK . . . . .	64
5.10	Kết quả trường hợp Locality khoa DC . . . . .	64
5.11	Kết quả trường hợp Global khoa DC . . . . .	64
5.12	Kết quả trường hợp Locality K12 khoa CK . . . . .	64
5.13	Kết quả trường hợp Locality K12 khoa DC . . . . .	64
5.14	Kết quả trường hợp Locality khoa DD . . . . .	65
5.15	Kết quả trường hợp Global khoa DD . . . . .	65
5.16	Kết quả trường hợp Locality khoa GT . . . . .	65
5.17	Kết quả trường hợp Global khoa GT . . . . .	65
5.18	Kết quả trường hợp Locality K12 khoa DD . . . . .	65
5.19	Kết quả trường hợp Locality K12 khoa GT . . . . .	65
5.20	Kết quả trường hợp Locality khoa HC . . . . .	66
5.21	Kết quả trường hợp Global khoa HC . . . . .	66
5.22	Kết quả trường hợp Locality khoa MO . . . . .	66
5.23	Kết quả trường hợp Global khoa MO . . . . .	66
5.24	Kết quả trường hợp Locality K12 khoa HC . . . . .	66
5.25	Kết quả trường hợp Locality K12 khoa MO . . . . .	66
5.26	Kết quả trường hợp Locality khoa QL . . . . .	67
5.27	Kết quả trường hợp Global khoa QL . . . . .	67
5.28	Kết quả trường hợp Locality khoa UD . . . . .	67
5.29	Kết quả trường hợp Global khoa UD . . . . .	67
5.30	Kết quả trường hợp Locality K12 khoa QL . . . . .	67
5.31	Kết quả trường hợp Locality K12 khoa UD . . . . .	67
5.32	Kết quả trường hợp Locality khoa VL . . . . .	68
5.33	Kết quả trường hợp Global khoa VL . . . . .	68
5.34	Kết quả trường hợp Locality khoa VP . . . . .	68
5.35	Kết quả trường hợp Global khoa VP . . . . .	68
5.36	Kết quả trường hợp Locality K12 khoa VL . . . . .	68
5.37	Kết quả trường hợp Locality K12 khoa VP . . . . .	68
5.38	Kết quả trường hợp Locality khoa XD . . . . .	69
5.39	Kết quả trường hợp Global khoa XD . . . . .	69
5.40	Kết quả trường hợp Locality khoa PD . . . . .	69
5.41	Kết quả trường hợp Global khoa PD . . . . .	69
5.42	Kết quả trường hợp Locality K12 khoa XD . . . . .	69
5.43	Kết quả trường hợp Locality K12 khoa PD . . . . .	69
5.44	Flow hệ thống Web . . . . .	76
5.45	Giao diện nhập dữ liệu đầu vào . . . . .	77
5.46	Giao diện kết quả dự đoán . . . . .	77

# Danh mục bảng biểu

3.1	Ma trận Utility Matrix . . . . .	21
3.2	Ma trận User Matrix . . . . .	21
3.3	Ma trận Item Matrix . . . . .	21
3.4	Các transaction được lưu trong cơ sở dữ liệu . . . . .	31
3.5	Các Frequent itemset được khai phá . . . . .	32
4.1	Dữ liệu được dùng làm đầu vào cho mô hình dự đoán điểm . . . . .	41
4.2	Thông kê tổng quan về tập dữ liệu đại học trường Đại học Bách Khoa . . . . .	41
4.3	Dữ liệu ban đầu của bài toán . . . . .	42
4.4	Thông kê chi tiết dữ liệu của 14 Khoa thuộc đại học Bách Khoa . . . . .	43
4.5	Các điểm số lớn hơn 10 và điểm bằng chữ ở trường điểm tổng kết môn học . . . . .	44
4.6	Các môn học với nhiều mã được mapping về một mã thống nhất . . . . .	44
4.7	Một mẫu ví dụ trong tập dữ liệu . . . . .	52
4.8	Biểu diễn phân bố điểm của sinh viên dưới dạng ma trận . . . . .	53
4.9	Ví dụ ma trận dự đoán . . . . .	53
4.10	Similarity Score between student 1511000 and other students . . . . .	55
4.11	Similarity Score between course 2 and other courses . . . . .	55
5.1	Thông số cấu hình của các node trong spark cluster . . . . .	58
5.2	Software Specification on the testing environment . . . . .	59
5.3	Thông tin của các giải thuật được sử dụng cho thực nghiệm . . . . .	60
5.4	Số vòng lặp của mỗi CD step . . . . .	61
5.5	Tham số của các giải thuật Matrix Factorization . . . . .	62
5.6	Tham số của các giải thuật Collaborative Filtering . . . . .	62
5.7	Tham số của giải thuật Collaborative Filtering trên Item Factor của Matrix Factorization . . . . .	70
5.8	Kết quả trường hợp Locality . . . . .	71
5.9	Kết quả trường hợp Global . . . . .	72
5.10	Kết quả trường hợp dữ liệu từ khóa 2012 trở lên . . . . .	73
5.11	Kết quả trường hợp Locality khi loại bỏ điểm 0 . . . . .	74
5.12	Kết quả RBM trường hợp Locality . . . . .	75

# Danh mục từ viết tắt

AI	Artificial Intelligence
ALS	Alternative Least Square
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CD	Contrastive Divergence
EDM	Educational Data Mining
FP-Growth	Frequent Pattern Growth
FP-Tree	Frequent Pattern Tree
GFS	Google File System
HPCC	High Performance Computing and Communications
HQL	Historical Query Language
IBCF	Item-based Collaborative Filtering
ID	Identification
IDL	Interface Definition Language
IEEE	Institute of Electrical and Electronics Engineers
KL divergence	Kullback-Leibler divergence
MAE	Mean absolute error
MLlib	Machine Learning library
MSE	Mean square error
NMF	Non-negative Matrix Factorization
RBM	Restricted Boltzmann Machine
RDD	Resilient Distributed Dataset
RMSE	Root mean square error
RPC	Remote Procedure Call
SQL	Structured Query Language
SVD	Single Value Decomposition
Tp.HCM	Thành phố Hồ Chí Minh
UBCF	User-based Collaborative Filtering
XML	Extensible Markup Language

# Chương 1

## TỔNG QUAN

Ở chương mở đầu này, chúng tôi sẽ giới thiệu về đề tài của luận văn, những động lực cũng như lý do nhóm chúng tôi chọn thực hiện đề tài, ý nghĩa mà đề tài mang lại trên phương diện khoa học và thực tiễn. Bên cạnh đó, chúng tôi cũng sẽ nêu cụ thể phạm vi, đối tượng và mục tiêu của đề tài hướng đến. Ở phần cuối của chương, chúng tôi sẽ trình bày về bối cảnh tổ chức của luận văn này để tiện cho việc truyền tải nội dung và kết quả của đề tài.

### 1.1 Giới thiệu đề tài

Dữ liệu luôn là một tài sản quý giá đối với tất cả mọi tổ chức và cá nhân. Ngày nay, công nghệ thông tin phát triển mạnh mẽ với nhiều thành tựu và đột phá. Một trong những lĩnh vực có bước tiến lớn của công nghệ thông tin là trí tuệ nhân tạo.

Những năm gần đây, với sự bùng nổ của trí tuệ nhân tạo - *Artificial Intelligence* (AI) nói chung và machine learning cũng như deep learning nói riêng đã tạo nên một cuộc cách mạng về công nghệ, biến máy tính trở dần hoàn thiện, có thể lập luận và hành động giống như con người. AI ngày nay xuất hiện trong hầu hết các lĩnh vực của đời sống. Cùng với sự bùng nổ về trí tuệ nhân tạo, chúng ta ngày càng có nhu cầu lớn hơn về khai phá và rút trích thông tin từ một lượng dữ liệu khổng lồ (Big Data). Với sự hỗ trợ của trí tuệ nhân tạo, hầu hết mọi lĩnh vực của đời sống, việc phân tích và khai thác thông tin từ dữ liệu trở nên hiệu quả hơn. Và lĩnh vực giáo dục, đặc biệt là giáo dục đại học cũng không phải là ngoại lệ.

Đối với lĩnh vực giáo dục, nhất là giáo dục ở bậc đại học, việc phân tích và rút trích thông tin cần thiết từ dữ liệu học tập của sinh viên để phục vụ cho quá trình học tập và giảng dạy đạt hiệu quả cao nhất chính là một trong những yêu cầu rất thiết thực và hữu ích. Việc phân tích dữ liệu giáo dục đại học ra sao để có được những thông tin cần thiết phục vụ cho quá trình đào tạo là một lĩnh vực tuy không mới nhưng luôn cần thiết và có nhiều ý nghĩa.

Xuất phát từ nhu cầu thực tiễn, việc "Nghiên cứu và phát triển công cụ phân tích dữ liệu cho đại học và trực quan hóa" nhằm mục đích phân tích các đặc trưng cơ bản của tập dữ liệu đại học, xây dựng hệ thống đề xuất môn học với các mô hình dự đoán điểm số của sinh viên trong các môn học để thông qua đó có những đề xuất về môn học phù hợp với năng lực cho sinh viên để nâng cao chất lượng học tập.

Giáo dục ở bậc đại học có những đặc điểm rất khác biệt so với các cấp giáo dục khác. Một trong số những đặc biệt nổi bật nhất chính là việc tự học, tự rèn luyện của sinh viên được xem là yếu tố cốt lõi. Các trường đại học không quản lý sinh viên suýt xao như ở các cấp đào tạo dưới, mà chủ yếu chú trọng khả năng tự học, tự sáng tạo của sinh viên. Do vậy, có 2 câu hỏi

được đặt ra ở đây :

1. **Đối với nhà trường, làm sao để sinh viên phát huy được tối đa nỗ lực của bản thân nhưng vẫn đảm bảo được chất lượng giảng dạy?** Đây là một yêu cầu hết sức quan trọng.
2. **Đối với sinh viên, làm sao để biết chính xác điểm mạnh, điểm yếu của bản thân để cải thiện chất lượng học tập?**

Trong xu thế phát triển hiện nay, kỹ thuật phân tích dữ liệu và trực quan hóa có nhiều ứng dụng trong thực tế và cũng chiếm một tỉ trọng rất lớn trong các ứng dụng thực tế. Việc nghiên cứu và phát triển công cụ phân tích dữ liệu cho đại học và trực quan hóa là cần thiết vì nó sẽ đáp ứng được những yêu cầu đặt ra từ hai câu hỏi trên:

- Việc phân tích dữ liệu học tập của sinh viên sẽ giúp cho các trường đại học nắm được tình hình học tập của sinh viên, có cái nhìn chi tiết về quá trình học tập của sinh viên, sự so sánh giữa sinh viên các Khoa với nhau để có những điều chỉnh trong giảng dạy cho phù hợp.
- Phân tích dữ liệu đại học và trực quan hóa cũng giúp cho nhà trường sớm phát hiện những bất thường trong việc học tập của sinh viên, ví dụ như phát hiện khả năng không đạt môn học của sinh viên, từ đó có những sự hỗ trợ cần thiết đối với sinh viên để nâng cao chất lượng giảng dạy.
- Đối với sinh viên, việc nghiên cứu và phát triển công cụ phân tích dữ liệu cho đại học và trực quan hóa giúp các bạn có được công cụ hỗ trợ các bạn phân tích quá trình học tập của mình, có cái nhìn chính xác những lĩnh vực mình mạnh và yếu để có chiến lược học tập phù hợp.
- Áp dụng kỹ thuật machine learning trong phân tích dữ liệu đại học cho kết quả dự đoán về điểm số đạt được của mỗi môn học, gợi ý những môn học nên học tiếp theo trong tương lai giúp sinh viên lựa chọn môn học phù hợp với khả năng, cũng như nhận biết những môn có khả năng không đạt để có sự quan tâm cần thiết.

## **1.2 Lý do chọn đề tài**

Xuất phát từ thực tiễn việc học tập trong môi trường đại học nói chung và trong Trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh nói riêng, nhóm chúng tôi nhận thấy sự cần thiết của việc có một hệ thống trợ giúp việc lựa chọn cũng như dự đoán kết quả các môn học để nâng cao chất lượng học tập và chọn lựa được các môn học phù hợp với năng lực bản thân, bên cạnh đó là mong muốn ứng dụng những kiến thức chúng tôi đã được học trong suốt quá trình 4 năm đại học, kết hợp với đó là xu thế chung của việc ứng dụng trí tuệ nhân tạo nói chung và học máy nói riêng trong các lĩnh vực của đời sống, nhóm chúng tôi quyết định chọn đề tài "Nghiên cứu và phát triển công cụ phân tích dữ liệu cho đại học và trực quan hóa" để có thể thực hiện được những mong muốn nói trên và có thể ứng dụng những gì đã được học để đem lại những kết quả thiết thực trên thực tế.

## 1.3 Ý nghĩa đề tài

### 1.3.1 Ý nghĩa khoa học

- Tìm hiểu về cơ sở lý thuyết và nền tảng của trí tuệ nhân tạo nói chung và các kỹ thuật học máy nói riêng, đặc biệt là các kỹ thuật được ứng dụng để xây dựng nên một hệ thống đề xuất (Recommendation System).
- Đề xuất được các mô hình dự đoán điểm cũng các kỹ thuật tương ứng được sử dụng cho tập dữ liệu đại học cụ thể.
- Có sự so sánh để đánh giá và lựa chọn các mô hình phù hợp để xây dựng hệ thống đề xuất với tập dữ liệu đại học thực tế.
- Vận dụng được lý thuyết kết hợp với các đề xuất trong việc xây dựng hệ thống đề xuất môn học cho sinh viên trên thực tế.

### 1.3.2 Ý nghĩa thực tiễn

- Xây dựng được công cụ giúp cho việc phân tích và trực quan hóa việc học tập của sinh viên, giúp sinh viên có sự lựa chọn phù hợp các môn học phù hợp với năng lực bản thân.
- Giúp cho nhà trường có những cái nhìn sớm về chất lượng và tình hình các lớp khi sinh viên đăng ký môn học để có những biện pháp hỗ trợ kịp thời.

## 1.4 Phạm vi và đối tượng nghiên cứu của đề tài

### 1.4.1 Phạm vi nghiên cứu của đề tài

Phạm vi nghiên cứu của đề tài luận văn này nằm trong lĩnh vực giáo dục ở bậc đại học. Đề tài sẽ nghiên cứu xoay quanh các yêu cầu đặt ra về áp dụng các kỹ thuật học máy trong vấn đề khai thác thông tin từ dữ liệu học tập của sinh viên ở trường đại học. Cụ thể ở đề tài này, chúng tôi nghiên cứu và phân tích dữ liệu về trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh.

### 1.4.2 Đối tượng nghiên cứu của đề tài

Đối tượng nghiên cứu của đề tài là dữ liệu đại học, ở đây là dữ liệu đại học của trường Đại học Bách Khoa - Đại học Quốc gia thành phố Hồ Chí Minh. Dữ liệu này là thông tin về quá trình học tập của sinh viên bao gồm điểm số của các môn học mà sinh viên đã học qua các học kỳ tương ứng.

## 1.5 Mục tiêu của đề tài

Đề tài này hướng đến xây dựng một bộ công cụ giúp phân tích dữ liệu sinh viên của một trường đại học và trực quan hóa kết quả trên màn hình lớn. Việc nghiên cứu và phát triển công

cụ phân tích dữ liệu cho đại học và trực quan hóa có những mục tiêu về kết quả cần đạt được cụ thể như sau:

- Phân tích, thống kê để đưa ra những thông tin cơ bản phục vụ cho việc nắm bắt các đặc điểm chính của bộ dữ liệu đại học.
- Ứng dụng được các kỹ thuật học máy vào việc xây dựng mô hình dự đoán điểm số các môn học cho sinh viên dựa trên bộ dữ liệu này.
- Từ kết quả phân tích và dự đoán, đưa ra được những gợi ý về môn học tiếp theo cho sinh viên.
- Trực quan hóa các kết quả phân tích và dự đoán, để giúp cho việc học tập cho sinh viên cũng như giúp cho nhà trường có những cái nhìn tổng quan về tình hình học tập của sinh viên.
- Tích hợp và triển khai thành một công cụ cụ thể để phục vụ cho việc sử dụng trên thực tế. Cụ thể là xây dựng một trang web để sinh viên có thể trực tiếp trải nghiệm và sử dụng.

## **1.6 Bố cục của luận văn**

Trong luận văn này, chúng tôi sẽ trình bày về quá trình nghiên cứu và phát triển công cụ phục vụ cho việc phân tích dữ liệu đại học và trực quan hóa thông qua việc ứng dụng các kỹ thuật machine learning vào việc xây dựng các mô hình dự đoán. Luận văn được cấu trúc bao gồm 6 chương, cụ thể như sau:

**Chương 1. TỔNG QUAN.** sẽ giới thiệu cái nhìn tổng quan về đề tài, lý do chúng tôi chọn đề tài, sự cần thiết của việc nghiên cứu đề tài, đối tượng và phạm vi nghiên cứu cũng như mục tiêu của đề tài.

**Chương 2. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN.** Trình bày những công trình nghiên cứu liên quan đến đề tài mà các nhà nghiên cứu trong và ngoài nước đã thực hiện trước đó.

**Chương 3. KIẾN THỨC NỀN TẢNG.** Trình bày một số kiến thức nền tảng về các kỹ thuật machine learning như Collaborative Filtering, Matrix Factorization, FP-Growth, ... và các công cụ được sử dụng trong đề tài.

**Chương 4. PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỀ XUẤT.** Trình bày chi tiết bài toán đặt ra trong đề tài, cung cấp cái nhìn tổng quan về bộ dữ liệu đại học và các mô hình mà chúng tôi đề xuất để xây dựng bộ công cụ phân tích dữ liệu đại học, cụ thể ở đây là hệ thống hỗ trợ dự đoán kết quả các môn học.

**Chương 5. THỰC NGHIỆM VÀ KẾT QUẢ.** Trình bày về môi trường thực nghiệm mà chúng tôi sử dụng để xây dựng hệ thống và các kết quả đạt được của luận văn này.

**Chương 6. TỔNG KẾT.** Trình bày một số kết luận, nhận định của chúng tôi về đề tài cũng như tổng kết lại đề tài luận văn.

## Chương 2

# CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Ở chương này, chúng tôi sẽ trình bày tổng quan về khai phá dữ liệu trong lĩnh vực giáo dục, đi kèm với đó là sơ lược về các công trình nghiên cứu đã được tiến hành, phương pháp sử dụng và kết quả của các công trình nghiên cứu đó. Các công trình nghiên cứu này liên quan đến đề tài luận văn mà nhóm chúng tôi thực hiện.

## 2.1 Tổng quan về Educational Data Mining và các ứng dụng thực tiễn

Nhiều nghiên cứu tập trung vào hệ thống lại tất cả các nghiên cứu và phương pháp được áp dụng trong EDM. Một trong những bài báo nổi tiếng nhất trong lĩnh vực này là [3] và [4]. [4] xem xét 300 bài báo được phát hành cho tới năm 2009 phân loại theo các lĩnh vực gồm:

- Analysis & Visualization:** Lĩnh vực tìm ra các thông tin quan trọng và có ích ảnh hưởng nhiều đến việc ra quyết định và dự đoán.
- Providing Feedback:** Lĩnh vực đưa ra các phản hồi cho giáo viên, quản trị viên trong việc đưa ra các quyết định (về cách để nâng cao năng lực sinh viên, tổ chức lại tài liệu môn học cho hiệu quả) để giúp họ chủ động khắc phục sự cố.
- Recommendation:** đưa ra các gợi ý cho sinh viên về các hoạt động, đường dẫn đến trang web, môn học tiếp theo tùy thuộc vào khả năng, cá tính của sinh viên đó.
- Predicting Performance:** Dự đoán những kết quả chưa có của sinh viên bao gồm điểm số, năng lực học tập, độ hiểu biết.
- Student Modeling:** Mô hình hóa khả năng nhận thức của con người, bao gồm mô hình hóa các kĩ năng, trí tuệ, mục tiêu của con người,.
- Detecting Behavior:** Tìm ra những hành vi bất thường của các sinh viên như: mục tiêu, chơi game, gian lận, bỏ học, ... .
- Grouping Students:** Phân loại sinh viên dựa trên những thông tin cá nhân, tính cách nhằm nâng cao năng lực học nhóm
- Social Network Analysis:** Tìm hiểu về mối quan hệ giữa các cá nhân.

9. **Developing Concept Map:** Hỗ trợ giảng viên/ giáo viên xây dựng bản đồ tư duy biểu thị mối quan hệ giữa các kiến thức, khái niệm
10. **Planning & Scheduling:** Nâng cao khả năng quản lý, lập kế hoạch cho các môn học tiếp theo, lập kế hoạch phân bổ thời gian học tập cho sinh viên, giúp nhà trường lập thời khóa biểu hợp lý.
11. **Constructing Courseware:** Xây dựng các hệ thống hỗ trợ cho việc học tập của sinh viên.

[5] nghiên cứu những xu hướng của EDM . Họ tìm ra rằng 43% bài báo được xem xét trong [3] được phát hành từ 1995 đến 2005 đều xoay quanh các phương pháp Relationship Mining. Nhưng trong năm 2008 và 2009, Relationship Mining chỉ đứng thứ năm với 9% bài báo nghiên cứu. Mặc khác, chủ đề về dự đoán (Prediction) đứng thứ hai trong năm 1995 đến 2005 vượt lên đứng đầu trong năm 2008 và 2009.

## 2.2 Các nghiên cứu nổi bật

Trong EDM, một trong những hướng nghiên cứu phổ biến là trích xuất ra những thông tin cần thiết có thể được sử dụng để dự đoán khả năng học tập của sinh viên [5] [4]. Thông thường, những nghiên cứu này nhằm mục đích phân loại sinh viên, dự đoán những sinh viên có học lực kém để có thể can thiệp kịp thời. Để giải quyết vấn đề trên thì những giải thuật Machine Learning thường được áp dụng. Một số giải thuật được lấy ý tưởng từ những phương pháp thường được sử dụng trong các hệ thống Recommender System [6].

Romero et al. sử dụng các giải thuật phân loại như Decision Tree, Rule Induction, Neural Network, ... để dự đoán điểm tổng kết của sinh viên dựa trên các thông tin trên hệ thống Moodle [7]:

- Số lượng Assignment hoàn thành.
- Số lượng quiz tham gia.
- Số lượng quiz đạt.
- Số lượng quiz thất bại.
- Số lượng bài đăng trên forum.
- Số lượng bài đọc trên forum.
- ...

Kết quả dự đoán được rời rạc hóa bằng cách dự đoán các khoảng điểm với 4 loại nhãn:

- FAIL: điểm  $< 5$ .
- PASS: điểm  $\geq 5$  và  $< 7$ .
- GOOD: điểm  $\geq 7$  và  $< 9$
- EXCELLENT: điểm  $\geq 9$ .

Một nghiên cứu khác cũng dựa trên tập dữ liệu lấy được từ hệ thống Moodle [8]. Họ phân tích 17 môn học với 4989 sinh viên sử dụng 23 đặc trưng gồm số thời gian online, số lượng link xem, số lượng click, ... . Kết quả nghiên cứu thấy rằng có một sự tiến bộ lớn trong việc dự đoán nếu đặc trưng điểm giữa kì được thêm vào so với việc dự đoán vào thời gian ở các tuần đầu khi mà thông tin này chưa có. Vì thế, các dữ liệu của hệ thống Moodle đóng vai trò kém quan trọng hơn so với dữ liệu điểm giữa kì trong việc dự đoán đối với tập dữ liệu này.

Random Forest được sử dụng trong [9] để phân tích mối quan hệ giữa kết quả tốt nghiệp của sinh viên và những thành tựu đạt được của sinh viên trong đại học. Kết quả cho thấy những thành tích đạt được trong năm 3 có ảnh hưởng nhiều nhất để dự đoán kết quả tốt nghiệp của sinh viên.

Garcia et al. sử dụng Association Rule Mining để tìm ra những thông tin trong dữ liệu sinh viên dưới dạng các quy tắc gọi ý IF-THEN. Thông tin trên được sử dụng để xây dựng một hệ thống giúp giáo viên nâng cao chất lượng của môn học e-learning [10].

Nurjanah et al. đề xuất phương pháp để gợi ý các tài liệu học tập cho sinh viên bằng cách kết hợp content-based filtering và collaborative filtering trong [11], [12]. Trong phương pháp này, content-based filtering sẽ được áp dụng trước để lọc ra những tài liệu liên quan đến môn sinh viên đang học trước. Sau đó, Collaborative Filtering được sử dụng để chọn ra các sinh viên học tốt các môn học đó và gợi ý các tài liệu mà các sinh viên này đọc đến cho sinh viên cần gợi ý. Phương pháp này được đề xuất nhằm giảm thiểu hạn chế của phương pháp Collaborative Filtering thông thường chỉ tính đến độ giống nhau giữa các sinh viên mà không quan tâm đến khả năng học của các sinh viên. Mô hình dự đoán được xây dựng theo phương pháp trên đạt độ lỗi MAE 0.96 đối với thang điểm từ 1 đến 10 và 0.73 đối với thang điểm từ 1 đến 5.

Một phương pháp khác thường được sử dụng là Matrix Factorization. [13] sử dụng phương pháp này để dự đoán kết quả của sinh viên trong tập dữ liệu Knowledge Discovery and Data Mining Challenge 2010. Bài báo trên cho thấy Matrix Factorization có thể nâng cao độ chính xác của kết quả dự đoán so với các phương pháp hồi quy thông thường như hồi quy tuyến tính, hồi quy logistic. Trong bài báo tiếp theo [14], họ mở rộng nghiên cứu và sử dụng Tensor-based Factorization để thêm vào các hiệu ứng thay đổi theo thời gian cho việc dự đoán.

[15] áp dụng và đánh giá 3 phương pháp Collaborative Filtering, Matrix Factorization và Restricted Boltzmann Machine để dự đoán điểm cho sinh viên trong tập dữ liệu gồm 225 sinh viên và 24 môn học với 1736 điểm có sẵn và 3664 điểm chưa có (các điểm được tính trên thang điểm 0-4). Trong nghiên cứu của họ mô hình Restricted Boltzmann Machine đạt kết quả tốt nhất với độ lỗi RMSE là 0.3 tốt hơn gấp đôi so với mô hình hiệu quả thứ nhì là Non-negative Matrix factorization với độ lỗi RMSE 0.57.

# Chương 3

## KIẾN THỨC NỀN TẢNG

Trong chương này, chúng tôi sẽ trình bày những kiến thức nền tảng về các giải thuật được sử dụng cũng như các công nghệ, công cụ hỗ trợ để thực hiện việc xây dựng công cụ dự đoán điểm và đề xuất môn học cho sinh viên.

### 3.1 Collaborative Filtering

#### 3.1.1 Tổng quan

Các hệ thống đề xuất (Recommendation Systems) có 2 hướng tiếp cận lớn là *Collaborative Filtering* và *Content-based Filtering*. Content-based Filtering sử dụng các đặc trưng của item như thể loại, màu sắc, ... và sở thích của người dùng để gợi ý. Còn đối với Collaborative Filtering, cách tiếp cận này gợi ý các item cho người dùng dựa trên độ tương quan giữa các user và các item.

Collaborative Filtering có thể được chia thành 2 loại chính, đó là:

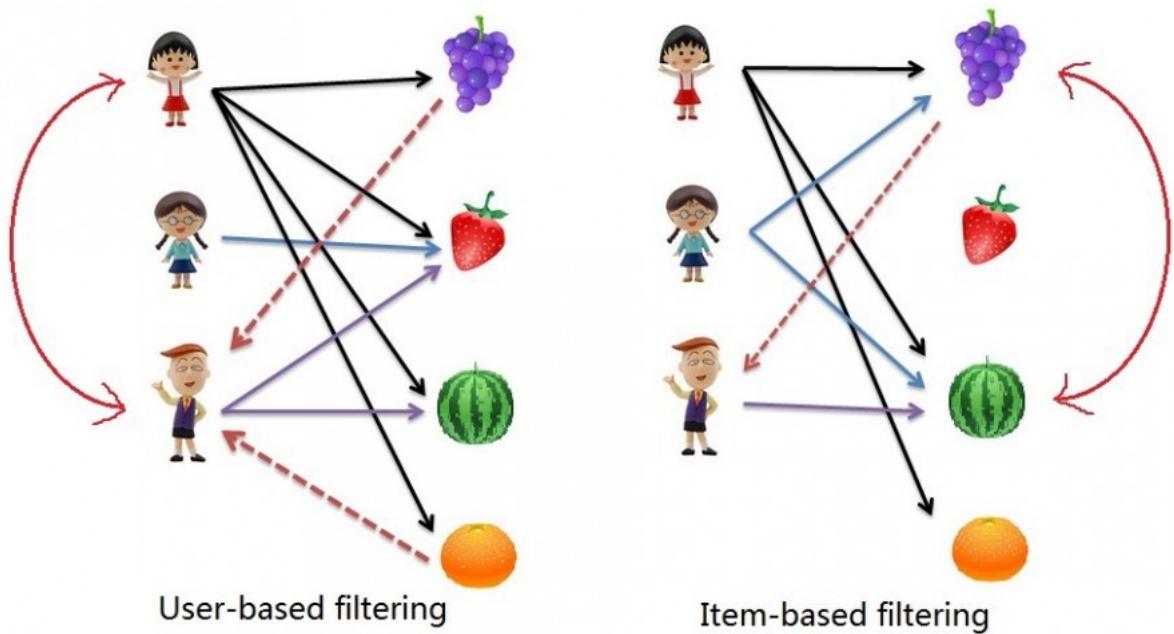
- **User-Based Collaborative Filtering (UBCF):** Phương pháp này sẽ tìm các user gần giống nhất đối với user đang cần đưa ra gợi ý và gợi ý các item cho user này dựa trên các user gần giống với user đó nhất.

Ví dụ: có 2 người (tạm gọi là A và B) đều đã ăn các món ăn giống nhau trong nhà hàng và đều đánh giá các món ăn đó với số điểm giống nhau. A chưa ăn món "Cơm chiên Dương Châu" nhưng B đã ăn món này rồi. Nếu B thích món "Cơm chiên Dương Châu" thì dựa trên cơ sở các đánh giá giống nhau cho cùng các món đã ăn, ta có thể dự đoán rằng A cũng thích món "Cơm chiên Dương Châu".

- **Item-Based Collaborative Filtering (IBCF):** Phương pháp này sẽ tìm các item gần giống với item hiện tại đang cần ra quyết định xem có nên gợi ý cho user không. Quyết định việc có nên đề xuất, gợi ý item hiện tại cho user hay không sẽ được đưa ra dựa trên sự đánh giá của user đối với các item gần giống với item đó.

Ví dụ: Trở lại với ví dụ về món ăn, ta có thể gợi ý cho user A món "Cơm chiên Dương Châu" nếu A đánh giá cao món "Cơm chiên trứng" và "Cơm chiên hải sản" vì các món ăn đó có độ tương quan với nhau cao, cụ thể đều là các món cơm và được chế biến bằng cách chiên.

Collaborative Filtering sẽ thực hiện việc gợi ý các item cho các user bằng cách xác định độ giống nhau giữa các user (hoặc item) dựa trên *Utility Matrix* bằng các *Similarity Function*. Chi



Hình 3.1: User-Based và Item-Based Collaborative Filtering [1]

tiết về *Utility Matrix*, các *Similarity Function* thường được sử dụng cũng như chi tiết cách thức vận hành của Collaborative Filtering sẽ được chúng tôi trình bày ở các phần ngay sau đây.

### 3.1.2 Utility Matrix

Gọi  $m$  là số lượng user,  $n$  là số lượng item. Ma trận  $M$  với kích thước  $m \times n$ , với  $M_{ij}$  là phần tử hàng thứ  $i$  cột thứ  $j$  của ma trận  $M$ , biểu thị độ thích/mức độ đánh giá của user thứ  $i$  đối với item thứ  $j$ .  $M_{ij}$  có thể không có giá trị nào, điều này biểu thị cho việc user thứ  $i$  chưa đánh giá cho item thứ  $j$ .

Ma trận  $M$  như trên được gọi là ***Utility Matrix***. Mục tiêu của Collaborative Filtering là làm sao điền vào các giá trị  $M_{ij}$  đang còn trống.

Utility matrix được tạo thành dựa trên các đánh giá của user cho các item. Ma trận này phục vụ cho việc đánh giá, tính toán mức độ giống nhau giữa các user hay giữa các item để đưa ra các giá trị dự đoán gần đúng nhất cho các item mà user chưa đánh giá.

### 3.1.3 Similarity Function

***Similarity Function*** là hàm được sử dụng để tính toán mức độ tương đồng (similarity) giữa user với user hoặc giữa item với item trong Collaborative Filtering.

Các hàm để tính toán mức độ tương đồng (similarity) thường được sử dụng là **Cosine**, **Pearson**, **Euclidean**.

- Cosine similarity:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.1)$$

Nếu sử dụng Cosine để tính toán similarity thì giá trị similarity sẽ nằm trong khoảng từ  $-1$  đến  $1$ . Giá trị similarity  $-1$  thể hiện sự đối lập tuyệt đối giữa A và B, giá trị  $0$  thể hiện A và B không có tương quan, giá trị  $1$  thể hiện sự tương quan tuyệt đối của A và B.

- Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.2)$$

Cũng giống như Cosine, giá trị similarity khi sử dụng Pearson sẽ nằm trong khoảng từ  $-1$  đến  $1$ . Giá trị similarity  $-1$  thể hiện sự đối lập tuyệt đối giữa A và B, giá trị  $0$  thể hiện A và B không có tương quan, giá trị  $1$  thể hiện sự tương quan tuyệt đối của A và B.

- Euclidean distance score:

Khoảng cách Euclidean giữa 2 điểm là chiều dài đoạn thẳng nối hai điểm đó. Ví dụ trong trường hợp tính độ tương quan giữa 2 item thì chiều dương 2 trục tọa độ biểu thị cho độ lớn điểm số đánh giá của 2 item đó và mỗi điểm trên không gian sẽ biểu thị cho điểm số đánh giá của 1 user đối với 2 item.

$$d = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad (3.3)$$

$$\text{similarity} = \frac{1}{(1+d)} \quad (3.4)$$

Đối với khoảng cách Euclidean thì nếu một user đánh giá các item cao hơn một user khác (user này thường cho item điểm cao hơn so với các user khác) thì khoảng cách Euclidean sẽ cho rằng user này không giống với các user khác mà không quan tâm đến độ tương quan giữa 2 user.

#### **Một số lưu ý khi lựa chọn similarity function:**

- Nên sử dụng Cosine similarity khi dữ liệu thưa.
- Nên sử dụng Pearson correlation coefficient khi thang đánh giá của các user là khác nhau.
- Nên sử dụng Euclidean khi độ lớn của điểm đánh giá là quan trọng.

### **3.1.4 Các bước thực hiện Collaborative Filtering**

#### **3.1.4.1 User-Based Collaborative Filtering**

Đối với User-Based Collaborative Filtering, ta thực hiện việc đề xuất các item cho user theo các bước như sau:

##### **1. Chuẩn hóa:**

Đối với các item chưa được user đánh giá, có nhiều cách để chúng ta có thể điền trước cho nó những giá trị mặc định.

- Điền trước vào các giá trị còn trống bằng giá trị trung bình cộng các điểm đánh giá tương ứng với từng user, hoặc

- Điền giá trị 0 vào các giá trị còn trống rồi chuẩn hóa các giá trị còn lại bằng cách trừ đi cho trung bình cộng điểm đánh giá của user tương ứng. Dùng cách này thay cho việc điền vào các giá trị còn trống là 0 hoặc trung bình của điểm cao nhất và thấp nhất để cân nhắc đến việc sự dễ tính của một user.

### 2. Tính ma trận similarity:

Đối với từng user, ta lần lượt tính độ tương đồng giữa user này với các user khác bằng một trong các similarity function. Kết quả thu được là một ma trận vuông  $S_{n \times n}$  đối xứng, với  $n$  là số lượng các user, giá trị trên đường chéo là 1. Giá trị  $S_{ij}$  là similarity giữa  $user_i$  và  $user_j$ .

### 3. Dự đoán:

Để đề xuất một item cho user, chọn ra  $k$  user với điều kiện có similarity cao nhất với user hiện tại dựa trên similarity đã tính toán ở bước trước đó và user được chọn đã đánh giá item đó. Việc tính toán giá trị dự đoán thường sử dụng công thức sau:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in N(u,i)} \bar{y}_{i,u_j} \text{similarity}(u, u_j)}{\sum_{u_j \in N(u,i)} |\text{similarity}(u, u_j)|} \quad (3.5)$$

Với  $\hat{y}_{i,u}$  là kết quả dự đoán điểm đánh giá của user  $u$  đối với item  $i$ ,  $N(u,i)$  là tập hợp  $k$  user có similarity cao nhất đối với user  $u$  đã đánh giá item  $i$ ,  $\bar{y}_{i,u_j}$  là điểm mà user  $u_j$  đánh giá item  $i$ .

Giá trị dự đoán tính được nói trên là giá trị dự đoán sau khi đã thực hiện chuẩn hóa ở bước 1. Vì vậy, sau khi tính được giá trị dự đoán đã chuẩn hóa, ta cộng thêm vào giá trị trung bình điểm đánh giá của mỗi user để được giá trị dự đoán ban đầu.

### 4. Recommend:

Thông thường, sẽ chọn ra các item có giá trị dự đoán cao nhất để gợi ý cho user. Việc chọn ra các giá trị dự đoán cao nhất để gợi ý này phù hợp trong ngữ cảnh đề xuất cho user những item mà họ có thể sẽ thích và phù hợp với họ. Đối với các ngữ cảnh đối lập, như dự đoán các item mà user sẽ không thích để không hiện lên quảng cáo, thì các item có giá trị dự đoán thấp nhất sẽ phù hợp trong ngữ cảnh này.

Do đó, việc chọn ra các item để đề xuất cho user là dựa trên các giá trị dự đoán đã tính được và phụ thuộc vào ngữ cảnh của hệ thống.

#### **Hạn chế của User-Based Collaborative Filtering:**

- Số lượng item trong thực tế thường ít hơn nhiều so với số lượng user rất nhiều và mỗi user đánh giá rất ít item. Do đó, nếu sử dụng UBCF, cần phải tốn chi phí lớn hơn IBCF cho việc tính toán similarity.
- Cold-start: Một user mới sẽ có rất ít thông tin để có thể so sánh với các user khác.
- Item mới: Một item mới sẽ có rất ít lượt đánh giá.

#### **3.1.4.2 Item-based Collaborative Filtering**

Các bước thực hiện tương tự như User-Based Collaborative filtering, trong trường hợp này ta có thể đảo ngược vai trò của item và user (item đánh giá user).

#### **Lợi thế của Item-Based Collaborative Filtering so với User-Based Collaborative Filtering:**

- Số lượng item thường nhỏ hơn số lượng user nên similarity matrix trong Item-Based Collaborative Filtering thường nhỏ hơn so với similarity matrix trong User-Based Collaborative Filtering.
- Một item có thể được nhiều người dự đoán hơn nên giá trị đánh giá trung bình của một item sẽ ít thay đổi hơn, vì thế nên việc cập nhật similarity matrix sẽ thực hiện ít hơn.

## 3.2 Matrix factorization

Matrix factorization là phương pháp phân tích một ma trận thành 2 hay nhiều ma trận khác. Ví dụ như LU Matrix Decomposition sẽ nhận vào một ma trận vuông phân tích ma trận vuông đó thành hai ma trận L và U với L là ma trận nửa tam giác dưới, U là ma trận nửa tam giác trên:

$$A = LU \quad (3.6)$$

Matrix Factorization là cơ sở cho một số latent factor model được hiện thực trong Recommender System mô hình hóa user/item thành các hidden feature. Latent factor model sẽ phân tích sinh viên và môn học dựa trên đặc trưng để giải thích phân bố điểm. Đối với môn học, các đặc trưng này có thể tương ứng cho số lượng tính toán, độ khó, hoặc những đặc trưng khác mà con người khó có thể hình dung. Đối với sinh viên, những đặc trưng này biểu thị cho khả năng thích ứng của sinh viên đối với các đặc trưng trên. Trong ngữ cảnh EDM, Matrix Factorization sẽ phân tích ma trận Utility Matrix biểu thị điểm đã biết của sinh viên thành hai hay nhiều ma trận.

### 3.2.1 Singular Value Decomposition

Singular Value Decomposition (SVD) là phương pháp phân tích ma trận Utility Matrix  $G$  thành hai ma trận  $U$  và  $V$ :

$$G \approx U \times V^T \quad (3.7)$$

Trong đó:

- $U$  là ma trận  $m \times r$ , với  $m$  là số lượng sinh viên,  $r$  là số lượng hidden features. Mỗi sinh viên  $u$  sẽ được biểu diễn bởi vector  $p_u$  có  $r$  chiều. Mỗi phần tử trong vector biểu diễn sự tương thích của sinh viên  $u$  đối với hidden feature tương ứng. Vector  $p_u$  là một dòng trong ma trận  $U$  trong đó  $U_{uk}$  biểu diễn độ tương thích của sinh viên  $u$  đối với hidden feature  $k$ .
- $V$  là ma trận  $r \times n$ , với  $n$  là số lượng môn học. Mỗi môn học  $i$  được biểu diễn bởi vector  $q_i$  với  $r$  chiều. Mỗi phần tử trong vector  $q_i$  biểu diễn độ tương thích của môn học  $i$  đối với hidden feature tương ứng. Vector  $q_i$  là một dòng trong ma trận  $V$  trong đó giá trị  $V_{ik}$  biểu diễn độ tương thích của môn học  $i$  đối với hidden feature  $k$ .

Điểm xấp xỉ được dự đoán của sinh viên  $u$  và môn học  $i$  là giá trị tích vô hướng  $p_u q_i^T$ :

$$\hat{g}_{ui} = p_u q_i^T \quad (3.8)$$

Để học ma trận  $U$  và  $V$ , ta sẽ tối thiểu hàm lỗi:

### Chương 3 KIẾN THÚC NỀN TẢNG

$$\sum_{u,i \in H} (r_{ui} - p_u \cdot q_i^T)^2 + \lambda (p_u^2 + q_i^2) \quad (3.9)$$

Trong đó:

- $H$  là tập hợp của các cặp  $(u, i)$  nếu  $g_{ui}$  nằm trong tập training;  $\lambda$  là regularization parameter.

Sử dụng gradient descent, đối với mỗi giá trị  $g_{ui}$  trong tập training, ta lần lượt cập nhật vector  $p_u$  và  $q_i$ :

$$\begin{aligned} p_u &= p_u + \gamma ((r_{ui} - p_u q_i^T) q_i - \lambda p_u) \\ q_i &= q_i + \gamma ((r_{ui} - p_u q_i^T) p_u - \lambda q_i) \end{aligned} \quad (3.10)$$

Trong đó:

- $\gamma$  là learning rate.

Ví dụ: Cho ma trận  $G$  sau:

Bảng 3.1: Ma trận Utility Matrix

	Courses 0	Courses 1	Courses 2	Courses 3
Student 1511000	9	8	?	?
Student 1512000	8	7	7.5	8.5
Student 1513000	7.5	8.5	7.5	?

Ma trận User Matrix  $U$  được khởi tạo ngẫu nhiên với các giá trị (số lượng hidden feature  $r$  được chọn là 3):

Bảng 3.2: Ma trận User Matrix

	Factor 0	Factor 1	Factor 2
Student 1511000	0.3	0.9	-0.1
Student 1512000	0.8	1	0.2
Student 1513000	-0.2	0.6	0.4

Ma trận Item Matrix  $V$  được khởi tạo ngẫu nhiên với các giá trị:

Bảng 3.3: Ma trận Item Matrix

	Course 0	Course 1	Course 2	Course 3
Factor 0	0.3	0.9	-0.1	0.7
Factor 1	0.8	1	0.2	-0.1
Factor 2	-0.2	0.6	0.4	0.6

Cho regularization parameter  $\lambda = 0$ , learning rate  $\gamma = 0.1$ .

Các bước thực hiện Singular Value Decomposition:

1. Khởi tạo ma trận  $U$  và  $V$ .

2. Tìm trong ma trận  $G$  một giá trị đã biết. Ví dụ giá trị 9 là điểm của sinh viên 151100 đối với môn học 0.
3. Lấy ra vector biểu diễn sinh viên 151100 trong ma trận  $U$ , trong ví dụ này là vector  $p = (0.3, 0.9, -0.1)$ . Lấy ra vector biểu diễn môn học 0 trong ma trận  $V$ , trong ví dụ này là vector  $q = (0.3, 0.8, -0.2)$ .
4. Tính tích vô hướng giữa hai vector để ra được điểm dự đoán:  $(0.3, 0.9, -0.1) \cdot (0.3, 0.8, -0.2) = 0.85$ .
5. Tính độ lỗi:  $error = (9 - 0.85) = 8.15$
6. Cập nhật  $p$ :  $p = (0.3, 0.9, -0.1) + 0.1 * 8.15 * (0.3, 0.8, -0.2)$   
 $= (0.5445, 1.552, -0.263)$ .
7. Cập nhật  $q$ :  $q = (0.3, 0.8, -0.2) + 0.1 * 8.15 * (0.5445, 1.552, -0.263)$   
 $= (0.7438, 2.0649, -0.4143)$
8. Cập nhật  $p$  và  $q$  vào ma trận  $U$  và  $V$ .
9. Lặp lại từ bước 2 cho đến hết giá trị trong  $G$  sẽ kết thúc một chu kỳ.

### 3.2.2 Alternative Least Square

Phương pháp Alternative Least Square (ALS) [16] là một tối ưu của phương pháp SVD. Trong **Công thức** (3.9) cả hai vector  $p_u$  và  $q_i$  đều chưa biết và được kết nối với nhau bởi tích vô hướng làm cho hàm lỗi này lỗi. Ý tưởng của giải thuật ALS là: Nếu ta cố định một trong hai biến  $p_u$  hoặc  $q_i$ , hàm lỗi sẽ trở thành bài toán quadratic problem. Trong mỗi vòng lặp, ALS sẽ cố định ma trận  $U$  (tất cả vector  $p_u$ ) trước và cập nhật  $V$ . Sau đó sẽ cố định  $V$  (tất cả vector  $p_i$ ) và cập nhật  $U$ . Quá trình này lặp lại cho tới khi hội tụ. Trong ALS, vì mỗi vector  $p_u$  độc lập với tất cả các vector  $p_{u' \neq u}$  còn lại và mỗi vector  $q_i$  độc lập với tất cả vector  $q_{i' \neq i}$  còn lại nên giải thuật này có thể được song song hóa.

### 3.2.3 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) là một trong những biến thể khác của Matrix Factorization trong đó ràng buộc không âm được thêm vào. Với một ma trận không âm  $G$  chứa tất cả các điểm của mọi sinh viên, ta tìm hai ma trận không âm  $W$  và  $H$  [17] sao cho:

$$G \approx W \times H \quad (3.11)$$

Trong đó:

- $G$  là ma trận không âm kích thước  $m \times n$ .
- $W$  là ma trận không âm kích thước  $m \times r$ .
- $H$  là ma trận không âm kích thước  $r \times n$

Đối với Matrix Factorization thông thường, ta có thể cho ra độ tương thích âm giữa sinh viên  $u$  và hidden feature  $k$  làm cho việc mô hình hóa trở nên khó khăn hơn để giải thích (ví dụ, hidden feature biểu diễn cho số lượng tính toán có thể âm). NMF có thể làm cho việc biểu diễn

các hidden feature này rõ ràng hơn bởi sự ràng buộc không âm, nhờ đó các môn học có thể được biểu diễn bởi 1 tập hợp các kiến thức nền tảng. Ví dụ, môn Đồ họa máy tính có thể được biểu diễn bằng tập hợp gồm 60% algebra +20% math +20% art +0% literature.

Trong NMF,  $W$  và  $H$  được cập nhật bằng công thức:

$$\begin{aligned} H &\leftarrow H \frac{W^T G}{W^T W H} \\ W &\leftarrow W \frac{G H^T}{W H H^T} \end{aligned} \quad (3.12)$$

Các công thức trên có thể được biểu diễn lại theo phương pháp gradient descent dưới dạng:

$$\begin{aligned} H &\leftarrow H - \eta (W^T W H - W^T G) \\ W &\leftarrow W - \lambda (W H H^T - G H^T) \end{aligned} \quad (3.13)$$

Trong đó nếu  $\eta$  được đặt là  $\frac{H}{W^T W H}$  và  $\lambda$  được đặt là  $\frac{W}{W H H^T}$  thì **công thức** (3.13) sẽ trở thành **công thức** (3.12).

### 3.3 Restricted Boltzmann Machine

#### 3.3.1 Kullback-Leibler divergence

Kullback-Leibler divergence (KL divergence) cho biết độ không giống nhau giữa phân bố  $P$  và phân bố  $Q$ . Một cách định nghĩa khác là Kullback-Leibler divergence cho biết độ hiệu quả của việc sử dụng phân bố xác suất (probability distribution)  $Q$  để xấp xỉ phân bố xác suất thực (true probability distribution)  $P$ .

$$\begin{aligned} D_{KL}(P||Q) &= H(P, Q) - H(P) \\ &= \mathbb{E}_{x \sim P} [\log \frac{P(x)}{Q(x)}] \end{aligned} \quad (3.14)$$

Trong đó:

$$\begin{aligned} H(P, Q) &= \mathbb{E}_{x \sim P} [-\log Q(x)] \\ H(P) &= \mathbb{E}_{x \sim P} [-\log P(x)] \end{aligned} \quad (3.15)$$

Với:

- $H(P, Q)$  là cross entropy của  $P$  và  $Q$ .
- $H(Q)$  là entropy của phân bố  $Q$ .

KL divergence có thể được viết dưới dạng rời rạc hoặc liên tục như sau:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (3.16)$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx \quad (3.17)$$

**Tính chất của KL divergence:**

- Không âm.
  - Khi  $P = Q$ , KL divergence có giá trị 0.
  - Khi  $P \neq Q$ , KL divergence có giá trị dương.
- Không đối xứng.

$$D_{KL}(P||Q) \neq D_{KL}(Q||P)$$

### Mối quan hệ giữa Minimize KL divergence và Maximize log likelihood

Trong Machine Learning, một trong những cách để học một cấu hình thông số để xấp xỉ một phân bố là maximize log likelihood. Một mối liên hệ đáng quan tâm là các thông số trong maximize log likelihood  $\theta_{MLE}$  chính là những thông số trong việc tối thiểu hóa KL divergence giữa phân bố  $p_{data}$  và phân bố của mô hình  $p_{model}$ .

$$\begin{aligned} \theta_{minKL} &= \underset{\theta}{\operatorname{argmin}} D_{KL}(p_{data}(x|\theta^*)||p_{model}(x|\theta)) \\ &= \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{x \sim p_{data}} [\log p_{data}(x|\theta^*) - \log p_{model}(x|\theta)] \\ &= \underset{\theta}{\operatorname{argmin}} -\mathbb{E}_{x \sim p_{data}} [\log p_{model}(x|\theta)] \\ &= \underset{\theta}{\operatorname{argmax}} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \log(p(x_i|\theta)) \\ &= \theta_{MLE} \end{aligned} \quad (3.18)$$

### 3.3.2 Restricted Boltzmann Machine

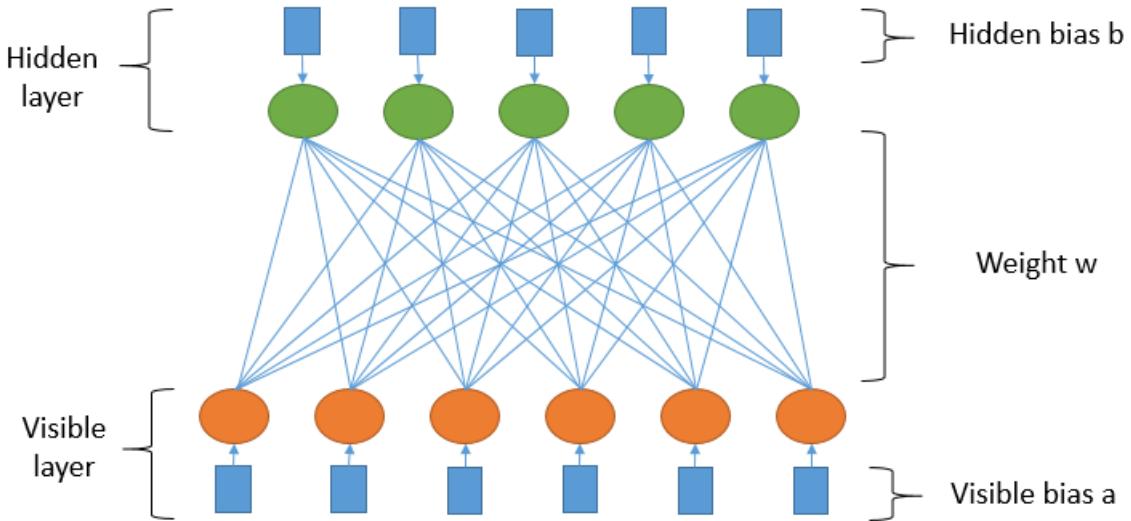
Restricted Boltzmann Machine (RBM) là một neural network gồm hai lớp: visible layer và hidden layer. Khác với Boltzmann Machine thông thường, RBM chỉ có liên kết giữa visible layer và hidden layer, các node trong visible layer không có liên kết với nhau và các node trong hidden layer cũng không có liên kết với nhau. **Hình 3.2** biểu diễn cấu trúc của một RBM đơn giản. Các giá trị của hidden node và visible node là binary (0 hoặc 1). Khi giá trị của một node là 1, ta gọi node này được kích hoạt và ngược lại, khi giá trị của node là 0, ta gọi node đó không được kích hoạt

Energy-based model liên kết cấu hình của các biến trong mô hình với một giá trị năng lượng. Giá trị năng lượng cao thể hiện sự tương thích không tốt. Vì thế các energy-based model sẽ cố tối thiểu một hàm năng lượng được định nghĩa trước. Trong RBM, hàm năng lượng được định nghĩa như sau:

$$E(v, h) = - \sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (3.19)$$

Trong đó:

- $a_i$ : giá trị bias ứng với visible node  $i$ .
- $v_i$ : trạng thái kích hoạt của visible node  $i$ .  $v_i = 1$  khi visible node  $i$  được kích hoạt,  $v_i = 0$  khi visible node  $i$  không được kích hoạt.
- $b_j$ : giá trị bias ứng với hidden node  $j$ .
- $h_j$ : trạng thái kích hoạt của hidden node  $j$ .  $h_j = 1$  khi hidden node  $j$  được kích hoạt,  $h_j = 0$  khi hidden node  $j$  không được kích hoạt.



Hình 3.2: Kiến trúc của một RBM

- $w_{ij}$  giá trị weight ứng với mỗi quan hệ giữa visible node  $i$  và hidden node  $j$ .

Tại một thời điểm, RBM sẽ có một trạng thái nhất định. Một trạng thái biểu thị cho giá trị của các node tại visible layer  $v$  và hidden layer  $h$  tương ứng. Xác suất để một trạng thái  $v$  và  $h$  xuất hiện là một phân bố sau:

$$p(v, h) = \frac{e^{-Z(v, h)}}{Z} \quad (3.20)$$

Giá trị  $Z$  được gọi là **partition function** được tính bằng **Công thức (3.21)**.

$$Z = \sum_{v, h} e^{-E(v, h)} \quad (3.21)$$

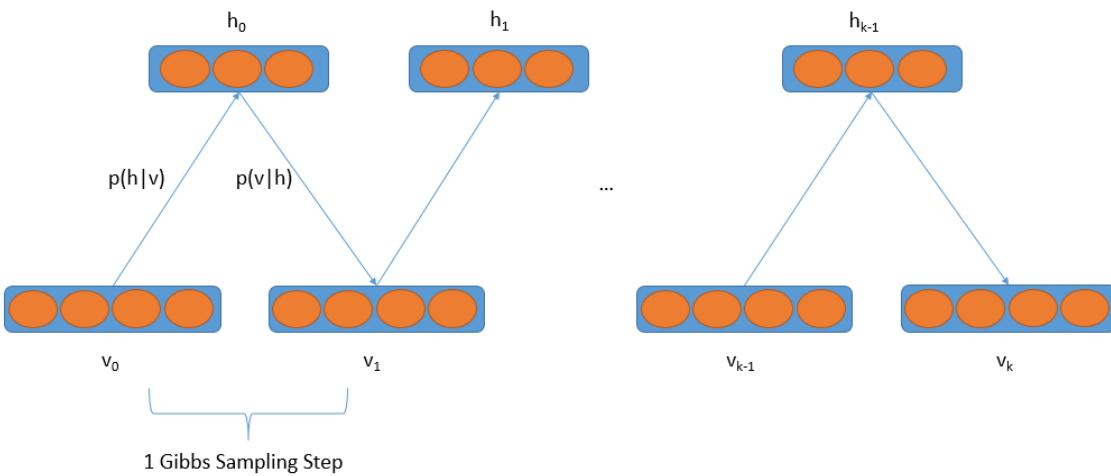
Vì các visible node độc lập với nhau và hidden node độc lập với nhau (các node trên cùng lớp không có liên kết với nhau) nên ta có:

$$\begin{aligned} p(h|v) &= \prod_i p(h_i|v) \\ p(v|h) &= \prod_i p(v_i|h) \\ p(h_j = 1|v) &= \sigma(b_j + \sum_i v_i w_{ij}) \\ p(v_i = 1|h) &= \sigma(a_i + \sum_j h_j w_{ij}) \end{aligned} \quad (3.22)$$

Trong đó  $\sigma$  là hàm Sigmoid:

$$\sigma(x) = \frac{1}{1 + e^x} \quad (3.23)$$

RBM được học bằng cách minimize negative log likelihood sử dụng gradient descend (hay



Hình 3.3: Gibbs Sampling

maximize log likelihood):

$$\begin{aligned}
 \frac{\partial \log p(v)}{\partial \theta} &= \frac{\partial}{\partial \theta} (\log \sum_h e^{-E(v,h)}) - \frac{\partial}{\partial \theta} (\log \sum_{v,h} e^{-E(v,h)}) \\
 &= - \sum_h p(h|v) \frac{\partial E(v,h)}{\partial \theta} + \sum_{v,h} p(v,h) \frac{\partial E(v,h)}{\partial \theta} \\
 &= -\mathbb{E}\left[\frac{\partial E(v,h)}{\partial \theta} | v\right] + \mathbb{E}\left[\frac{\partial E(v,h)}{\partial \theta}\right] \\
 &= \mathbb{E}\left[\frac{\partial E(v,h)}{\partial \theta}\right] - \mathbb{E}\left[\frac{\partial E(v,h)}{\partial \theta} | v\right]
 \end{aligned} \tag{3.24}$$

### 3.3.3 Training

#### Gibbs Sampling

Bước đầu của quá trình học trong RBM là Gibbs Sampling.

1. Với vector data ban đầu \$v\$ ta sử dụng công thức tính xác suất kích hoạt của các hidden node trong hidden layer \$p(h|v)\$ ở **Công thức (3.22)**.
2. Dựa trên xác suất kích hoạt của các hidden node trong hidden layer ta tiến hành sampling - một số hidden node sẽ được kích hoạt và một số không được kích hoạt.
3. Khi có giá trị kích hoạt của các hidden node, tính \$p(v|h)\$ theo **Công thức (3.22)**.
4. Dựa trên xác suất kích hoạt của visible node \$p(v|h)\$, tiến hành sampling sự kích hoạt của các visible node.
5. Quá trình trên được lặp lại \$k\$ lần, ta được một vector mới \$v\_k\$ được tạo từ vector input ban đầu \$v\_0\$.

Quá trình Gibbs Sampling được biểu diễn như **Hình 3.3**.

#### Contrastive Divergence

Để tránh việc tính toán log likelihood gradient, [18] đề xuất phương pháp Contrastive Divergence (CD) nhằm giảm thời gian huấn luyện. Phương pháp CD trở nên phổ biến trong việc

huấn luyện mô hình RBM. Phương pháp học của Machine Learning là tối thiểu KL divergence:

$$KL(p_0 || p_\infty) = \sum_x p_0(x) \log \frac{p_0(x)}{p_\infty(x)} \quad (3.25)$$

CD học theo gradient của hiệu hai KL divergence:

$$CD_n = KL(p_0 || p_\infty) - KL(p_n || p_\infty) \quad (3.26)$$

Với  $n$  là số bước Gibbs Sampling thực hiện. Công thức cập nhật các thông số cuối cùng sẽ là:

$$\begin{aligned} w &\leftarrow w + \lambda (v_0 \otimes p(h_0 | v_0) - v_k \otimes p(h_k | v_k)) \\ b &\leftarrow b + \lambda (p(h_0 | v_0) - p(h_k | v_k)) \\ a &\leftarrow a + \lambda (v_0 - v_k) \end{aligned} \quad (3.27)$$

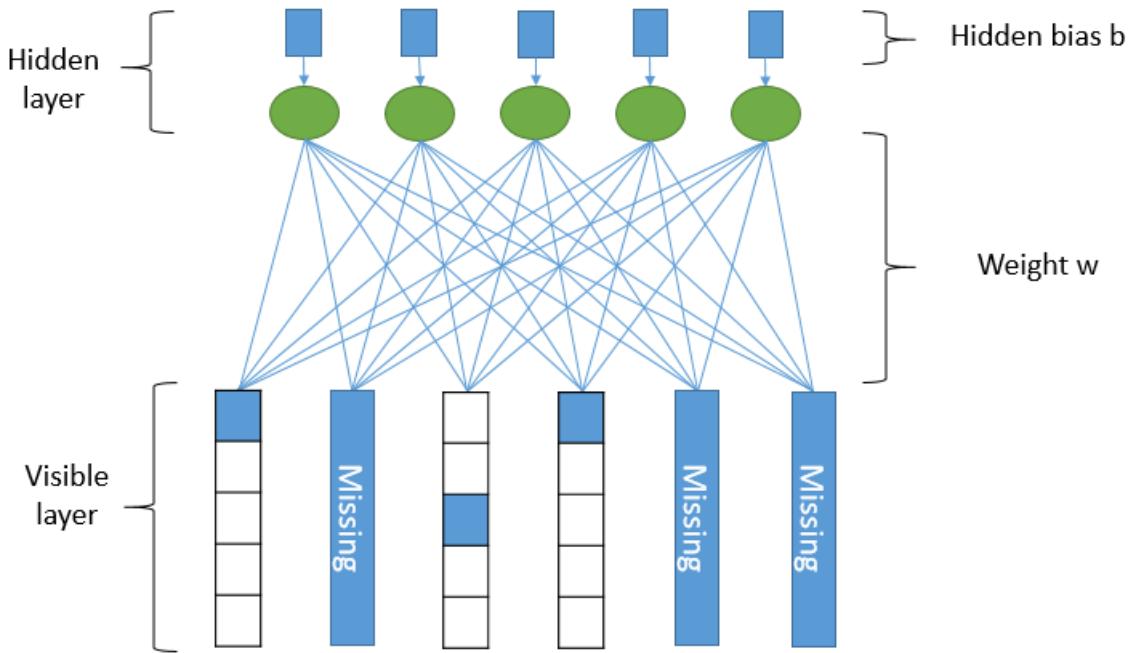
Trong đó:

- $w$  là ma trận weight kích thước  $F \times M$ .
- $b$  là vector  $F$  chiều với  $F$  là số lượng hidden node chứa bias của các hidden node.
- $a$  là vector  $M$  chiều với  $M$  là số lượng visible node chứa bias của các visible node.
- $v_0$  là vector là input đầu vào của RBM biểu thị trạng thái kích hoạt của các visible node có  $M$  chiều với các giá trị binary.
- $v_k$  là vector biểu thị trạng thái kích hoạt của các visible node sau  $k$  bước Gibbs Sampling.
- $h_0$  là vector biểu thị trạng thái kích hoạt của các hidden node với trạng thái kích hoạt  $v_0$  của các visible node tương ứng.
- $h_k$  là vector biểu thị trạng thái kích hoạt của các hidden node với trạng thái kích hoạt  $v_k$  sau  $k$  bước Gibbs Sampling của các visible node tương ứng.

### 3.3.4 Restricted Boltzmann Machine for Collaborative Filtering

Là một biến thể của RBM, nhằm hạn chế điểm yếu của RBM giá trị của các node chỉ có giá trị binary 0 hoặc 1 và giải quyết vấn đề input đầu vào bị thiếu (missing). Giả sử bài toán có  $M$  bộ phim (movies),  $N$  người dùng (users), và những giá trị đánh giá của người dùng cho các bộ phim với miền giá trị là các số nguyên từ 1 đến  $K$ . Cấu tạo của RBM này như sau:

- Mỗi người dùng sẽ là một training case với một RBM khác nhau.
- Các RBM đều có cùng số lượng hidden node ở hidden layer.
- Một visible node sẽ có  $K$  giá trị softmax.
- Các hidden node đều có liên kết đến từng giá trị softmax của visible layer.
- RBM chỉ chứa visible node cho bộ phim được đánh giá bởi người dùng đó. Nên một RBM sẽ có ít liên kết nếu người dùng đánh giá ít bộ phim.



Hình 3.4: RBM for Collaborative Filtering

- Các giá trị weight và bias của các RBM được chia sẻ cho nhau.

**Hình 3.4** biểu thị một RBM cho một người dùng. Giá trị của hidden node là một số binary, một visible node là nhiều softmax unit. Với mỗi user, RBM chỉ chứa softmax unit tương ứng với bộ phim mà user đã đánh giá. Mỗi hidden node một giá trị bias và mỗi softmax unit đều có một giá trị bias tương ứng.

*Công thức (3.22)* sẽ trở thành:

$$p(v_i^k = 1 | h) = \frac{\exp(b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)} \quad (3.28)$$

$$p(h_j = 1 | V) = \sigma(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k))$$

Với:

- $v_i^k$  bằng 1 khi nếu người dùng đánh giá bộ phim  $i$  là  $k$ , ngược lại là 0.
- $F$  là số lượng hidden node của RBM,  $h_j$  là giá trị binary của hidden node.
- $b_i^k$  là giá trị bias của điểm số  $k$  của bộ phim  $i$ .

Mô hình RBM được học bằng phương pháp học CD. Để dự đoán ta có thể tính xác suất kích hoạt của các hidden node, sau đó lấy giá trị xác suất này để tính xác suất kích hoạt của tất cả các softmax unit của mọi visible node (kể cả các visible node của những giá trị bị mất):

$$\hat{p}_j = p(h_j = 1 | V) = \sigma(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k) \quad (3.29)$$

$$p(v_i^q = 1 | \hat{p}) = \frac{\exp(b_q^k + \sum_{j=1}^F \hat{p}_j W_{qj}^k)}{\sum_{l=1}^K \exp(b_q^l + \sum_{j=1}^F h_j W_{qj}^l)}$$

Với:

- $q$  là bộ phim muôn dự đoán kết quả đánh giá của người dùng.

Chi tiết của các công thức trên và nhiều biến thể khác được đề cập trong [19].

## 3.4 Khai phá luật kết hợp với FP-Growth

### 3.4.1 Tổng quan về khai phá luật kết hợp

Khai phá luật kết hợp là một phần quan trọng trong khai phá dữ liệu, việc khai phá luật kết hợp nhằm mục đích tìm ra các item thường xuất hiện cùng nhau và dựa vào đó để xuất cho user những item phù hợp cho họ dựa trên những item mà user đang có.

**Ví dụ:** Trong ngữ cảnh bán hàng online, các nhà phân phối bán hàng thường dựa trên giỏ hàng của user để đề xuất cho họ những mặt hàng mà họ sẽ mua kèm với những mặt hàng hiện tại họ đang có trong giỏ. Chẳng hạn như một user A đang có trong giỏ hàng các mặt hàng: dao cạo râu, kem cạo râu và xịt khử mùi. Thông qua việc khai phá luật kết hợp dựa trên dữ liệu mua hàng của các khách hàng trước đó, ta có được tập các mặt hàng thường đi cùng nhau là dao cạo râu, lưỡi dao cạo râu, kem cạo râu và có được một suy luận (một luật) là nếu như user mua dao cạo râu và kem cạo râu thì họ cũng sẽ mua lưỡi dao cạo râu với một độ chính xác xác định, thì ta có thể đề xuất cho khách hàng A này mặt hàng là "dao cạo râu". Việc làm sao để tìm ra các phần tử thường xuyên xuất hiện cùng nhau (Frequent-item set) và các luật dựa trên các tập thường xuyên này sẽ được trình bày ở các phần ngay sau đây.

#### 3.4.1.1 Các khái niệm cơ bản trong khai phá luật kết hợp

- *Item (Phần tử):*

Là các phần tử, mẫu, đối tượng được quan tâm.

- *Itemset (Tập phần tử):*

Một tập hợp của các item. Đối với một tập hợp có k item, thì được gọi là k-itemset.

- *Transaction (Giao dịch):*

Là một lần thực hiện tương tác với hệ thống.

**Ví dụ:** một lần mua hàng trong ví dụ mua hàng. Các giao dịch này sẽ bao gồm một tập các phần tử trong giao dịch đó.

- *Association (Sự kết hợp):*

Các phần tử cùng xuất hiện với nhau trong một hay nhiều giao dịch. Sự kết hợp này thể hiện mối quan hệ của các phần tử trong dataset.

- *Association rule (Luật kết hợp):* Luật kết hợp thể hiện sự liên hệ có điều kiện giữa các tập phần tử.

**Ví dụ:** Luật  $A \rightarrow B$  thể hiện mối liên hệ có điều kiện giữa A và B, B chỉ xuất hiện khi A xuất hiện.

- *Support (Độ hỗ trợ):*

Độ đo đo tần số xuất hiện của các phần tử/tập phần tử. *Minimum support threshold (ngưỡng hỗ trợ tối thiểu)* là giá trị support nhỏ nhất được chỉ định.

- *Confidence (Độ tin cậy):*

Độ đo đo tần số xuất hiện của một tập phần tử trong điều kiện xuất hiện của một tập phần tử khác. *Minimum confidence threshold (ngưỡng tin cậy tối thiểu)* là giá trị confidence nhỏ nhất được chỉ định.

- *Frequent itemset (Tập phần tử phổ biến):* Tập phần tử có support thỏa minimum support threshold.

Cho A là một itemset. A là frequent itemset nếu và chỉ nếu  $\text{support}(A) \geq \text{minimum support}$ .

- *Strong association rule (Luật kết hợp mạnh):*

Luật kết hợp có support và confidence thỏa minimum support và minimum confidence.

Cho luật kết hợp  $A \rightarrow B$  giữa A và B, A và B là itemset.  $A \rightarrow B$  là luật kết hợp mạnh nếu và chỉ nếu:

- $\text{support}(A \rightarrow B) \geq \text{minimum support}$  và
- $\text{confidence}(A \rightarrow B) \geq \text{minimum confidence}$

### 3.4.2 Khai phá luật kết hợp với FP-Growth

Cũng giống như những giải thuật khai phá luật kết hợp khác, FP-Growth sẽ nhằm mục đích khai phá dữ liệu để tìm được các frequent itemset và dựa vào ngưỡng hỗ trợ tối thiểu (minimum support) và độ tin cậy tối thiểu (minimum confidence) được người dùng định nghĩa trước để xây dựng các luật kết hợp cho các tập itemset.

#### 3.4.2.1 Xây dựng FP-Tree

Cây FP-Tree được sử dụng để tìm ra tập các frequent itemset. Đối với mỗi transaction trong cơ sở dữ liệu, một đường đi tương ứng với một nhánh trên cấu trúc cây sẽ được tạo ra.

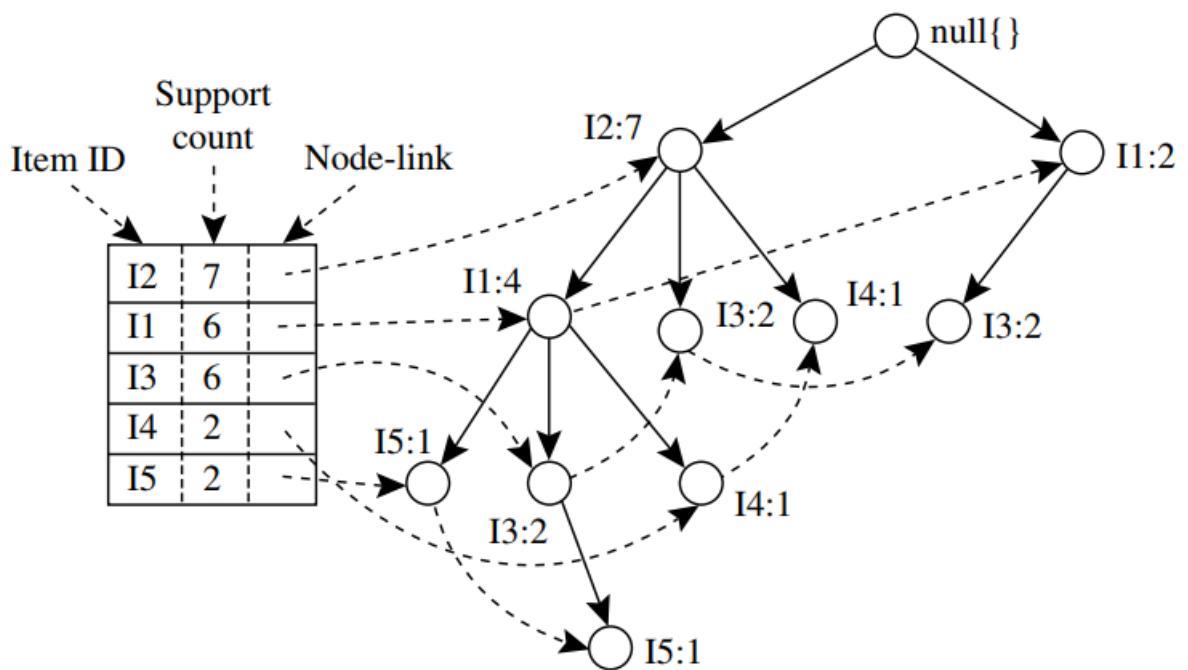
Việc xây dựng FP-Tree được thực hiện như sau:

1. Kiểm tra tập dữ liệu, tìm các frequent 1-itemsets.
2. Sắp xếp lại thứ tự các frequent 1-itemsets theo sự giảm dần của support count tương ứng.
3. Xây dựng FP-Tree:
  - Tạo root của FP-tree, được gán nhãn “null” {}.
  - Mỗi giao dịch tương ứng một nhánh của FP-tree.
  - Mỗi node trên một nhánh tương ứng một item của giao dịch. Các item của một giao dịch được sắp theo giảm dần. Mỗi node kết hợp với support count của item tương ứng.
  - Các giao dịch có chung item tạo thành các nhánh có prefix chung.

Với tập dữ liệu trong *Bảng 3.4*, cây FP-Tree được tạo thành như *Hình 3.5* (Ví dụ này được tham khảo từ ví dụ 6.5 thuộc chương 6, tài liệu tham khảo [2], trang 257 - 259).

Bảng 3.4: Các transaction được lưu trong cơ sở dữ liệu

Transaction ID	Tập các item
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Hình 3.5: FP-Tree được xây dựng từ các transaction [2]

### 3.4.2.2 Khai phá các tập frequent itemset

Từ FP-Tree đã xây dựng được, ta sẽ tiến hành khai phá các tập frequent itemset theo các bước như sau:

1. Tạo conditional pattern base cho mỗi node của FPtree.  
➤ Tích luỹ các đường đi (prefix paths) với tần số xuất hiện tương ứng của node đó.
2. Tạo conditional FP-tree từ mỗi conditional pattern base.  
➤ Tích lũy frequency cho mỗi item trong mỗi base.  
➤ Xây dựng conditional FP-tree cho frequent items của base đó.
3. Khám phá conditional FP-tree và phát triển frequent itemsets một cách đệ qui.

Bảng 3.5: Các Frequent itemset được khai phá

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent itemset được khai phá
I5	{ {I2,I1: 1}, {I2,I1,I3: 1} }	{I2: 2, I1: 2}	{I2,I5: 2}, {I1,I5: 2}, {I2,I1,I5: 2}
I4	{ {I2,I1: 1}, {I2:1} }	{I2: 2}	{I2,I4: 2}
I3	{ {I2,I1: 2}, {I2: 1} }	{I2: 4, I1: 2}	{I2,I3: 4}, {I1,I3: 4}, {I2,I1,I3: 2}
I1	{ {I2: 4} }	{I2: 4}	{I2,I1: 4}

➤ Nếu conditional FP-tree có một path đơn thì liệt kê tất cả các itemsets.

Với ví dụ trên, khi ràng buộc min support là 2 thì các tập frequent itemset được khai phá thể hiện như trong *Bảng 3.5*.

### 3.4.3 Khai phá luật kết hợp

Sau khi khai phá được các tập frequent itemset, các association rule sẽ được sinh ra bằng cách:

- Đối với mỗi một tập frequent itemset L, tạo các tập con không rỗng của nó.
- Đối với mỗi tập con không rỗng S của L, ta tạo ra luật " $S \rightarrow (L-S)$ " nếu  $\text{support}(L)/\text{support}(S) \geq \text{minimum confidence}$ .

*Ví dụ:* Ở ví dụ trên, nếu ta chọn minimum confidence là 50% thì với tập frequent itemset  $L = \{I1, I2, I5\}$  ta có các tập con không rỗng của nó là  $\{I1, I2\}$ ,  $\{I1, I5\}$ ,  $\{I2, I5\}$ ,  $\{I1\}$ ,  $\{I2\}$  và  $\{I5\}$ .

Các luật kết hợp được sinh ra là:

- $I1, I2 \rightarrow I5$  với confidence là 50%
- $I1, I5 \rightarrow I2$  với confidence là 100%
- $I2, I5 \rightarrow I1$  với confidence là 100%
- $I5 \rightarrow I1, I2$  với confidence là 100%

## 3.5 Các công nghệ sử dụng

### 3.5.1 Spark

#### 3.5.1.1 Tổng quan về Apache Spark

**Spark** là một công cụ được thiết kế tích hợp hầu như tất cả các công cụ cho việc giải quyết dữ liệu lớn (Big Data). Nếu như **Hadoop** - một framework nguồn mở viết bằng Java cho phép phát triển các ứng dụng phân tán có cường độ dữ liệu lớn một cách miễn phí dựa trên ý tưởng từ các công bố của Google về mô hình Map-Reduce và hệ thống file phân tán Google File System (GFS) - nổi tiếng với Map-Reduce thì Spark ngoài việc có sức mạnh không thua kém gì Hadoop Map-Reduce, nó còn mở rộng Hadoop Map-Reduce sang một cấp cao hơn.

Để sử dụng hiệu quả cho các loại dữ liệu khác nhau, chúng ta có các công cụ chuyên biệt, ví dụ:

- Batch processing - sử dụng Hadoop Map-Reduce.
- Stream processing - sử dụng Apache Storm/S4.
- Interactive processing - sử dụng Apache Impala hoặc Apache Tez.
- Graph processing - sử dụng Neo4j hoặc Apache Giraph...

Không có một công cụ mạnh mẽ nào có thể xử lý dữ liệu cả real-time (dữ liệu thời gian thực) và batch mode (dữ liệu bô). Vì vậy, nay sinh yêu cầu một công cụ có thể đáp ứng được các yêu cầu này và có thể xử lý được dữ liệu trong bộ nhớ (in-memory) để tăng tốc độ tính toán. Đó chính là lý do tại sao Apache Spark ra đời và trở thành một công cụ mạnh mẽ được sử dụng rộng rãi.

Apache Spark là một công cụ mã nguồn mở mạnh mẽ bởi vì nó hỗ trợ cả real-time processing, batch processing, interactive processing, graph processing, và được thực hiện in-memory. Nó có tốc độ rất nhanh, dễ sử dụng. Những điều này tạo nên sự khác biệt giữa Spark và Hadoop.

#### 3.5.1.2 Sơ lược về lịch sử phát triển của Spark

Đầu tiên, Spark được giới thiệu năm 2009 tại UC Berkeley R&D Lab. Đến năm 2010, Spark trở thành một công cụ mã nguồn mở dưới BSD License. Đến năm 2013, Spark được tài trợ bởi Apache Software Foundation và năm 2014, nó chính thức trở thành Apache Spark.

#### 3.5.1.3 Một số đặc điểm chính của Spark

Apache Spark có một số đặc điểm nổi bật như sau:

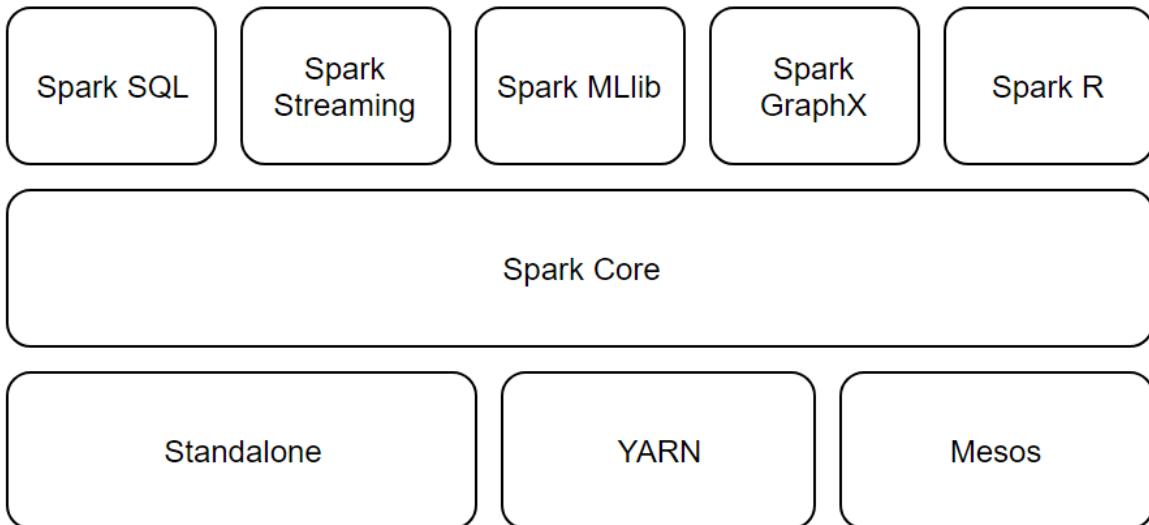
- Spark có một *key-feature* (tính năng nổi bật) đó là nó có khả năng thực hiện các tính toán trong bộ nhớ trên một cluster (in-memory cluster computation). Điều này giúp cho tốc độ tính toán được tăng lên và cũng đồng thời làm tăng tốc độ xử lý của ứng dụng.
- Spark được viết bằng Scala, nhưng nó hỗ trợ nhiều các ngôn ngữ như Scala, Java, Python và R.
- Quan trọng nhất, khi so sánh với Hadoop, Spark nhanh hơn khoảng 100 lần cho việc xử lý dữ liệu lớn so với Big Data Hadoop và nhanh hơn khoảng 10 lần về tốc độ truy cập dữ liệu từ đĩa.
- Spark có thể chạy ở chế độ Standalone hoặc trên hệ thống File System như Hadoop YARN hay Messo.

#### 3.5.1.4 Các thành phần chính của Spark

Apache Spark có khả năng xử lý dữ liệu nhanh hơn và dễ dàng phát triển hơn những công cụ khác. Điều này đến từ các thành phần cấu thành nên nó. Tất cả các thành phần của Apache Spark đã giải quyết các vấn đề gặp phải khi sử dụng Hadoop MapReduce.

Spark có 6 thành phần chính, đó là: Spark Core, Spark SQL, Spark Streaming, Spark MLlib, Spark GraphX và SparkR (Hình 3.6).

- **Spark Core** là trung tâm của Spark. Về cơ bản, Spark Core cung cấp một nền tảng thực thi cho toàn bộ các ứng dụng Spark. Hơn nữa, để hỗ trợ một loạt các ứng dụng, Spark Core còn cung cấp một nền tảng chung tổng quát (generalized platform).



Hình 3.6: Kiến trúc của Apache Spark

- **Spark SQL:** nằm ở top của Spark, Spark SQL cho phép người dùng có thể thực hiện các truy vấn SQL/HQL. Ngoài ra, Spark SQL còn hỗ trợ xử lý hiệu quả cho dữ liệu có cấu trúc, bán cấu trúc.
- **Spark Streaming:** thông qua các live streaming, Spark Streaming cho phép một ứng dụng tương tác mạnh mẽ và phân tích dữ liệu trên các streaming đó. Các live stream sẽ được chuyển đổi thành các micro-batches được thực thi trên top của Spark Core.
- **Spark MLlib** là thư viện Machine Learning của Spark, cung cấp cả sự hiệu quả và những giải thuật tốt. Một số giải thuật mà Spark MLlib cung cấp như: Matrix Factorization, các giải thuật gom cụm như K-Means, các giải thuật khai phá luật kết hợp như FP-Growth,... Spark MLlib là sự lựa chọn tốt và hiệu quả cho lĩnh vực khoa học dữ liệu. Vì Spark có khả năng thực hiện tính toán in-memory, do đó, nó cải thiện hiệu suất của các giải thuật lặp một cách đáng kể.
- **Spark GraphX:** Về cơ bản, Spark GraphX là một công cụ tính toán đồ thị (graph computing) được xây dựng để hỗ trợ cho việc xử lý các dữ liệu đồ thị. Nếu như trước đây, để xử lý với dữ liệu dạng đồ thị, cần phải sử dụng đến các công cụ chuyên dụng như Neo4j, thì Spark đã tích hợp cả Spark GraphX để hỗ trợ cho vấn đề xử lý trên dữ liệu đồ thị.
- **SparkR** để sử dụng Apache Spark từ R. Nó là một gói (package) R cho phép phân tích tập dữ liệu lớn. Ý tưởng chính đằng sau SparkR là khám phá những kỹ thuật khác để tích hợp khả năng sử dụng của R với khả năng mở rộng của Spark.

Sức mạnh của Spark đến từ các **Resilient Distributed Dataset (RDD)**.

**Resilient Distributed Dataset (RDD)** là một khái niệm trừu tượng của Spark, nó là đơn vị dữ liệu cơ bản trong Spark. Đó là một tập phân tán của các phần tử qua các node trên cluster của Spark. RDD thực hiện các thao tác, tác vụ song song. Hơn nữa, RDD là không đổi (immutable), vì vậy có thể tạo ra một RDD mới bằng cách biến đổi RDD sẵn có.

Có 3 cách để tạo ra Spark RDD:

- **Paralelled collections:** bằng cách gọi phương thức song song hóa (parallelize) trên chương trình điều khiển, có thể tạo ra các RDD.

- **External datasets:** tạo ra Spark RDD bằng cách gọi phương thức *textFile*. Vì thế, phương thức này lấy URL của file và đọc nó như một tập các dòng.
- **Existing RDD:** tạo ra các RDD mới từ các RDD sẵn có.

### 3.5.1.5 Các ưu điểm của Spark

Nhờ vào việc sử dụng các RDD, Spark có những ưu điểm như sau:

**Khả năng tính toán in-memory:** Trong khi lưu trữ các RDD, dữ liệu được lưu trữ trong bộ nhớ. Chính việc giữ dữ liệu trong bộ nhớ đã giúp cải thiện hiệu suất, hiệu suất được cải thiện tỉ lệ với độ lớn của dữ liệu, nghĩa là dữ liệu càng lớn thì hiệu suất cải thiện được thể hiện càng rõ ràng.

**Tính toán lười:** điều này có nghĩa dữ liệu trong RDD không được tính toán trong quá trình thực hiện trung gian. Chỉ sau khi cần thiết, tất cả các thay đổi hoặc tính toán mới được thực hiện. Điều này giúp hạn chế khối lượng công việc phải thực hiện và loại bỏ các tính toán không cần thiết, giúp cho thao tác trên dữ liệu trở nên nhanh chóng hơn.

**Khả năng phục hồi nếu có lỗi,** nếu có bất kỳ một node nào thực hiện tính toán hay xử lý thất bại, bằng cách sử dụng dòng hoạt động, Spark có thể tính toán lại vùng dữ liệu bị mất của RDD từ RDD gốc ban đầu. Vì thế, nó có thể phục hồi dữ liệu bị mất một cách dễ dàng.

**Khả năng không bị biến đổi** có nghĩa khi ta tạo 1 RDD, ta không thể thao tác trực tiếp để làm biến đổi nó, mà việc thao tác đó tạo ra một RDD kết quả tương ứng. Do đó, trong quá trình xử lý, dữ liệu đạt được tính nhất quán.

**Tính bền vững:** với việc lưu trữ các RDD thường xuyên được sử dụng trong bộ nhớ, khi cần truy xuất thì dữ liệu được truy xuất trực tiếp từ bộ nhớ mà không cần phải đi đọc đĩa, điều này giúp tăng tốc độ thực thi và có thể thực cùng lúc nhiều tác vụ trên cùng một dữ liệu.

**Tính phân hoạch:** RDD phân vùng các bản ghi (record) một cách luận lý. Phân tán dữ liệu trên nhiều node của cluster. Vì vậy, nó cung cấp khả năng song song hóa.

**Khả năng xử lý song song** nhờ vào việc dùng các RDD để phân tán dữ liệu trên các node của cluster.

**Hỗ trợ nhiều kiểu dữ liệu** khác nhau, có nhiều kiểu dữ liệu Spark RDD ví dụ như integer, long, string ...

**Không giới hạn RDD sử dụng,** số lượng RDD sử dụng bao nhiêu là phụ thuộc vào kích thước đĩa và bộ nhớ...

### 3.5.1.6 Các nhược điểm của Spark

Bên cạnh những ưu điểm nổi trội làm cho Spark trở thành công cụ được sử dụng rộng rãi, thì nó cũng có một số những hạn chế:

**Không hỗ trợ xử lý real-time hoàn toàn:** Spark chỉ hỗ trợ ở mức "gần" real-time. Nói cách khác, Spark không phải là công cụ xử lý hoàn toàn real-time.

**Gặp vấn đề với các tập tin dữ liệu nhỏ:** trong RDD, mỗi tập tin là một phân vùng nhỏ. Điều này có nghĩa là có một lượng lớn phân vùng nhỏ trong RDD. Vì vậy, nếu ta muốn hiệu quả trong xử lý, RDD nên được phân vùng lại thành nhiều định dạng có thể quản lý được. Việc này đòi hỏi yêu cầu xáo trộn rộng khắp trên cluster.

**Tốn chi phí về bộ nhớ:** spark đánh đổi giữa chi phí tính toán với chi phí về mặt lưu trữ.

**Không có hệ thống file system riêng:** vấn đề chính của Spark là nó không có hệ thống file system của riêng nó, mà chỉ yếu dựa trên những nền tảng khác như Hadoop hoặc cloud-based platform.

**Số lượng giải thuật hỗ trợ khá ít:** Spark MLlib có số lượng khá ít các giải thuật sẵn có.

**Phải tối ưu hóa thủ công,** cần phải đặc tả đầy đủ tập dataset. Hơn nữa, để việc phân vùng trong Spark được chính xác, bắt buộc phải điều khiển nó một cách thủ công...

### 3.5.2 Remote Procedure Call và Apache Thrift

Remote Procedure Call (RPC) là một phương thức giao tiếp để một chương trình (client) yêu cầu một service của một chương trình (server) khác trên một máy tính khác trong mạng.

Các bước hoạt động của RPC:

1. Client gọi 1 hàm cục bộ (client stub). Client stub thực hiện đóng gói các tham số truyền vào và tạo ra một hoặc nhiều gói tin để truyền qua mạng. Việc đóng gói này gọi là marshaling.
2. Gói tin được gửi sang cho server.
3. Server stub nhận gói tin và thực hiện đọc gói tin thành các tham số mà client truyền vào.
4. Server stub gọi hàm cục bộ xử lý với tham số đọc từ gói tin nhận được.
5. Khi server xử lý xong, kết quả sẽ được trả về cho server stub.
6. Server stub đóng gói kết quả trả về thành 1 hoặc nhiều gói tin truyền lại cho client.
7. Client stub nhận gói tin chứa kết quả, đọc gói tin và trả về kết quả.

Cách thức hoạt động trên làm cho client nhận được kết quả từ một lời gọi hàm mà không nhận thức được việc truyền gửi dữ liệu đã diễn ra.

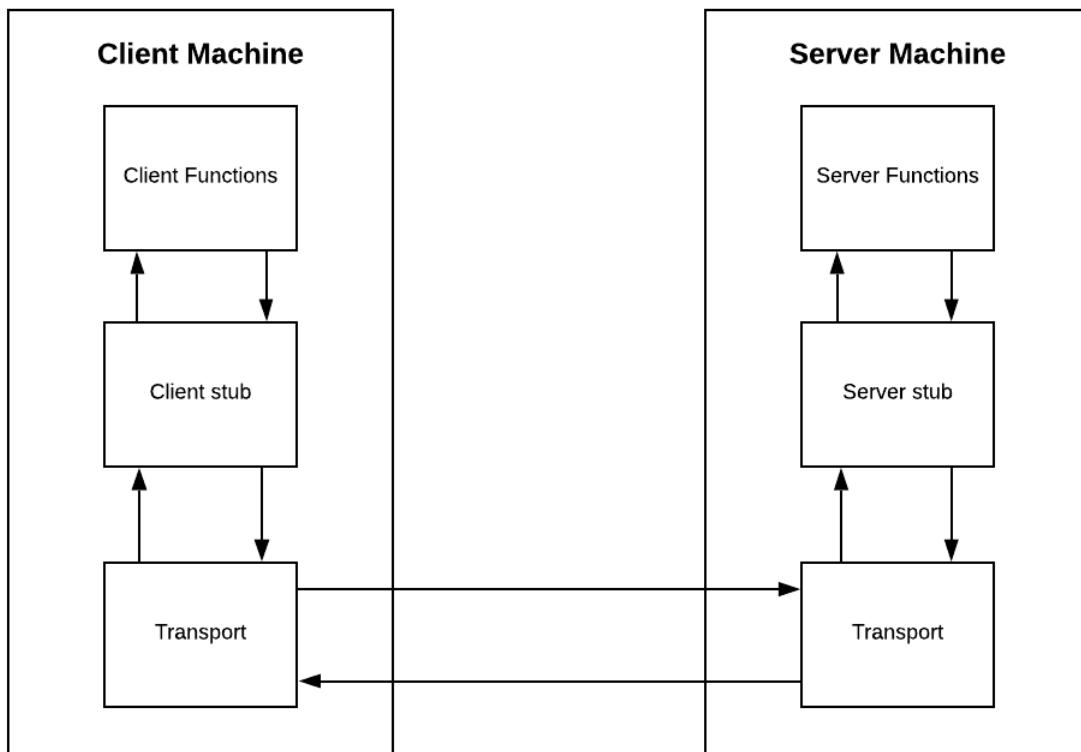
Lợi ích của RPC:

- Server và client độc lập.
- Người lập trình không cần quan tâm tới cấu trúc mạng và cách Server và client giao tiếp với nhau.
- Các nguyên lý truyền, gửi dữ liệu, định dạng dữ liệu ẩn đối với người lập trình.

Thrift là một thư viện và là công cụ tạo tự động code (code generation tool) được phát triển tại Facebook (sau này trở thành một open-source project của Apache Software Foundation) nhằm hỗ trợ cho việc hiện thực các dịch vụ backend một cách hiệu quả và nhanh chóng. Thrift cung cấp cho người lập trình khả năng định nghĩa kiểu dữ liệu, các dịch vụ bằng ngôn ngữ trung lập (source file định dạng .thrift). Thrift compiler sẽ build source file này và tạo ra các code cần thiết để xây dựng RPC client và server trên nhiều ngôn ngữ lập trình khác nhau như Java, C/C++, Python, ... (**Hình 3.8**).

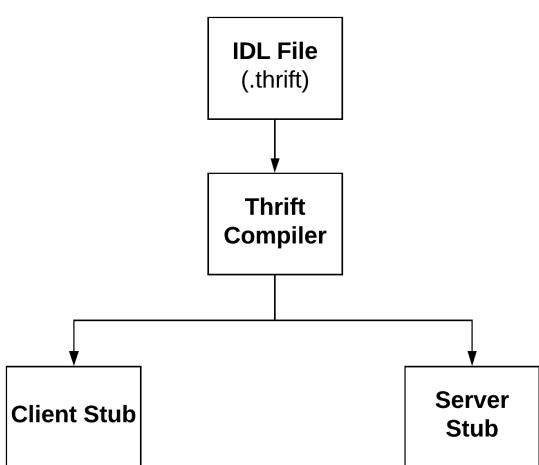
Các thành phần chính của Thrift (**Hình 3.9**):

- Type: Sử dụng Thrift IDL (Interface Definition Language) để định nghĩa kiểu dữ liệu truyền nhận giữa client-server và tạo ra các cấu trúc dữ liệu tương ứng trên các ngôn ngữ lập trình khác.

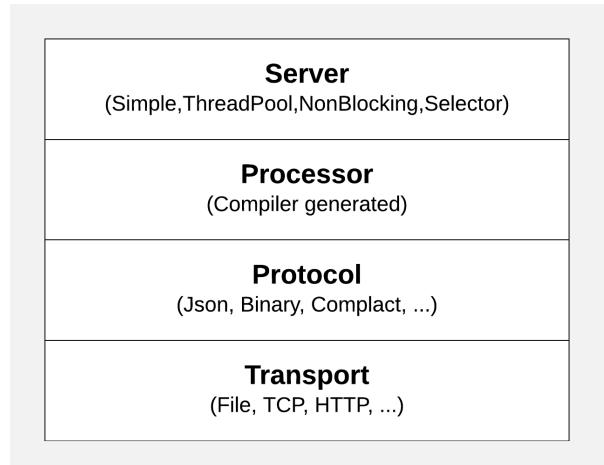


Hình 3.7: RPC Flow

- Transport: Giúp chuẩn hóa nguồn dữ liệu. Dữ liệu có thể truyền từ file, socket, memory, ... . Thrift transport interface gồm những hàm cơ bản như open(), close(), read(), write(), ....
- Protocol: Cung cấp giao thức chung để đọc dữ liệu truyền nhận giữa client-server bất kể dữ liệu được encode dưới dạng XML, binary, ASCII, ....
- Processor: Thành phần tự động tạo code cho client và server. Cung cấp interface để người lập trình hiện thực logic của service muốn định nghĩa.



Hình 3.8: Thrift Compiler



Hình 3.9: Thrift network protocol stack

# Chương 4

## PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỀ XUẤT

Nội dung cốt lõi của đề tài chính là việc xây dựng mô hình dự đoán điểm số các môn học cho sinh viên. Ở chương này, chúng tôi sẽ mô hình lại bài toán chính của đề tài, các đặc trưng của bộ dữ liệu đại học, các vấn đề tiền xử lý đối với bộ dữ liệu đại học cũng như đưa ra các giải pháp đề xuất để xây dựng mô hình giải quyết bài toán.

### 4.1 Đặc tả bài toán

Như đã đề cập ở các phần trước, đề tài "Nghiên cứu và phát triển công cụ phân tích dữ liệu cho đại học và trực quan hóa" hướng đến mục tiêu trọng tâm là xây dựng được một công cụ giúp cho việc dự đoán điểm số các môn học trong tương lai cho sinh viên dựa trên dữ liệu học tập hiện tại, từ đó đưa ra các kết quả cũng như đề xuất các môn học phù hợp với sinh viên. Có thể thấy, việc làm thế nào để dự đoán điểm số cho sinh viên và làm thế nào để đề xuất cho sinh viên những môn học phù hợp chính là bài toán cốt lõi của đề tài.

#### 4.1.1 Bài toán dự đoán điểm số và đề xuất môn học

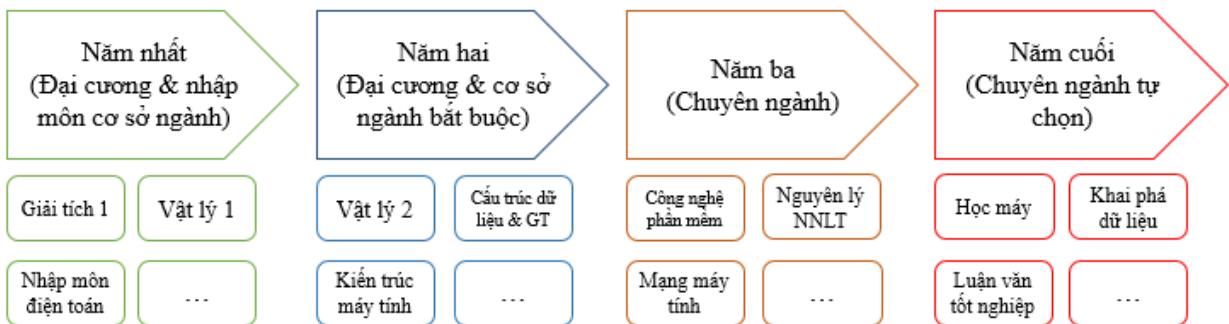
Ở bậc giáo dục đại học, chương trình học tập được chia thành nhiều năm (thông thường khoảng 4-5 năm), ở mỗi năm đào tạo sẽ được phân phối các môn học theo mức độ từ cơ sở, cơ bản đến chuyên ngành, nâng cao. **Hình 4.1** minh họa cho việc các môn học được phân phối, chia ra theo các năm học.

Bài toán dự đoán điểm số và đề xuất môn học cho sinh viên chính là việc dựa vào dữ liệu học tập trước đó của sinh viên để đưa ra dự đoán điểm số của sinh viên đó trong những môn học tiếp theo và dựa trên cơ sở những kết quả dự đoán đó để đưa ra đề xuất những môn học tiếp theo mà sinh viên nên học để có được kết quả tốt nhất hoặc phù hợp nhất với từng sinh viên.

Bài toán này có 2 nội dung chính cần giải quyết, đó là:

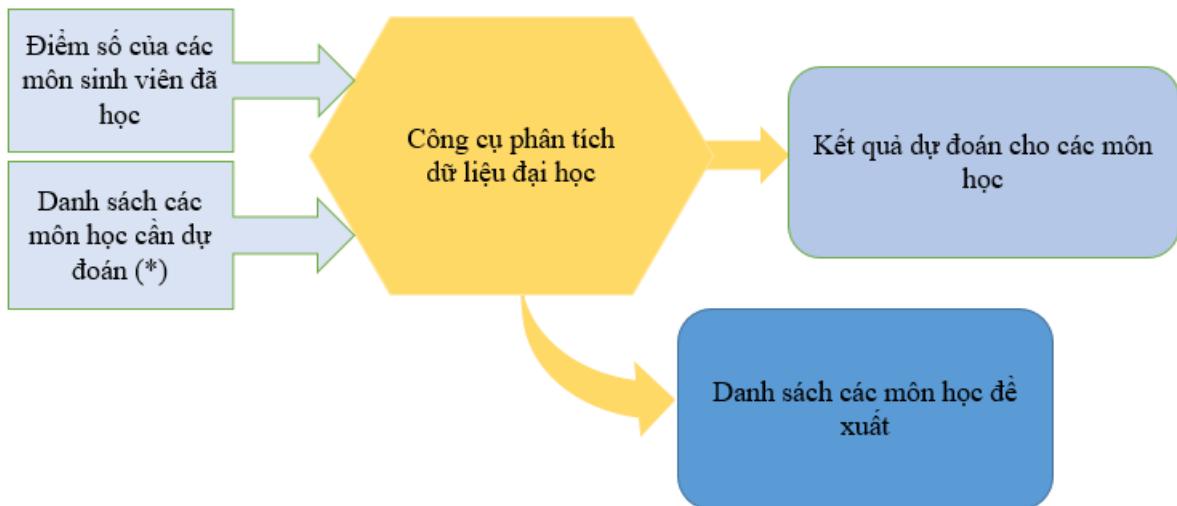
- **Thứ nhất**, đó là làm sao xây dựng công cụ dự đoán được điểm số các môn học trong tương lai.
- **Thứ hai**, là đưa ra những đề xuất môn học phù hợp cho sinh viên.

## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỀ XUẤT



Hình 4.1: Ví dụ về chương trình đào tạo của Khoa Khoa học và Kỹ thuật Máy tính - Đại học Bách Khoa

### 4.1.2 Mô hình hóa bài toán dự đoán điểm số và đề xuất môn học



Hình 4.2: Mô hình hóa công cụ phân tích dữ liệu đại học

Bài toán dự đoán điểm số và đề xuất môn học được mô hình như **Hình 4.2**, cụ thể:

- Input:** đầu vào của bài toán là dữ liệu điểm số các môn học mà sinh viên đã học. Khi sử dụng làm input cho các mô hình dự đoán điểm, dữ liệu được đưa vào gồm các trường chính là mã sinh viên, khoa, tên môn học, mã môn học và điểm tổng kết cho môn học đó. Dữ liệu được dùng làm input này được thể hiện minh họa như trong **Bảng 4.1**. Một input khác có thể đưa vào hoặc không đó là danh sách các môn học sinh viên muốn dự đoán điểm số. Nếu không đưa vào danh sách các môn học cần dự đoán, thì công cụ phân tích dữ liệu đại học sẽ dự đoán tất cả các môn còn lại mà sinh viên chưa học.
- Output:** kết quả đầu ra chính là điểm số dự đoán mà công cụ phân tích dữ liệu đại học đưa ra đối với danh sách các môn học cần dự đoán. Cùng với đó, công cụ phân tích dữ liệu đại học có thể đưa ra danh sách các môn học đề xuất cho sinh viên.

Công cụ phân tích dữ liệu đại học ở **Hình 4.2** bao gồm 2 module lớn đó là module dự đoán điểm số và module đề xuất môn học (**Hình 4.3**). Chi tiết về các giải pháp đề xuất để xây dựng công cụ phân tích dữ liệu đại học cũng như các module dự đoán điểm số và đề xuất môn học sẽ được trình bày chi tiết ở mục **mục 4.3 Mô hình giải quyết bài toán**.

## Chương 4 PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT

Bảng 4.1: Dữ liệu được dùng làm đầu vào cho mô hình dự đoán điểm

Mã sinh viên	Khoa	Mã môn học	Tên môn học	Điểm
29081892	MT	CO3029	Khai phá dữ liệu	7.5
28193782	HC	MT1003	Giải tích 1	9.0
32876719	CK	PE1003	Giáo dục thể chất 1	7.5
...	...	...	...	...



Hình 4.3: Hai module chính của công cụ phân tích dữ liệu đại học

## 4.2 Tập dữ liệu đại học

Tập dữ liệu đại học được sử dụng trong luận văn là tập dữ liệu đại học của trường đại học Bách Khoa - Đại học Quốc gia Tp.HCM. Tập dữ liệu bao gồm dữ liệu học tập của 61271 sinh viên đại học thuộc 14 khoa, với trên 2 triệu bản ghi về điểm số của sinh viên. Các thống kê tổng quan về bộ dữ liệu được thể hiện trong **Bảng 4.2**.

Bảng 4.2: Thống kê tổng quan về tập dữ liệu đại học trường Đại học Bách Khoa

Tổng số khoa	14
Tổng số môn học	2389
Tổng số sinh viên	61271
Tổng số bản ghi dữ liệu	2270045
Sparsity(Độ thừa của dữ liệu)	0.9845

Tập dữ liệu ban đầu bao gồm 33 trường sau: Năm học, học kỳ, mã học kỳ, mã môn học, tên môn học, đơn vị tín chỉ, phần trăm kiểm tra, phần trăm thi, mã nhóm, mã tổ, số thứ tự, mã khoa, tên lớp, khối, mã ngành, tên ngành, mã số sinh viên, điểm kiểm tra, tỉ lệ kiểm tra, điểm bài tập, tỉ lệ bài tập, điểm bài tập lớn, tỉ lệ bài tập lớn, điểm thí nghiệm, tỉ lệ thí nghiệm, điểm thi, tỉ lệ thi, điểm tổng kết, điểm tổng kết 1, điểm tổng kết 2, điểm tổng kết hệ 10, ghi chú và ghi chú khác, được thể hiện trong **Bảng 4.3**.

## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT

Bảng 4.3: Dữ liệu ban đầu của bài toán

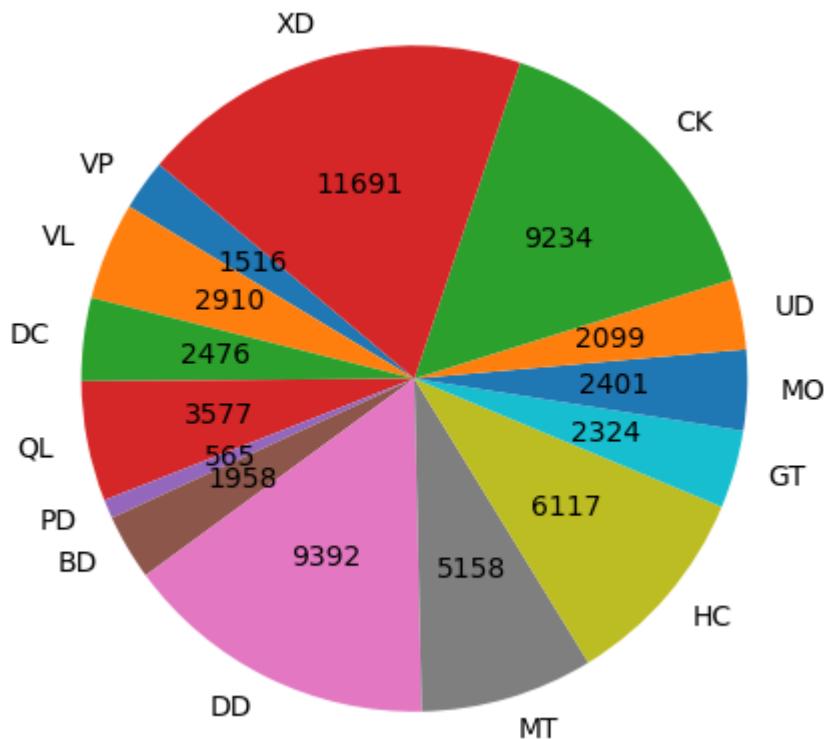
Năm học	Học kỳ	Môn học	Mã sinh viên	Khoa	...	Tổng kết
2014	CO3029	Khai phá dữ liệu	29081892	MT	...	8.5
2013	MT2001	Xác suất thống kê	28193782	XD	...	CT
2013	MT1009	Phương pháp tính	32876719	CK	...	3.5
...	...	...	...	...	...	...

Tập dữ liệu được thu thập từ dữ liệu học tập của sinh viên trường Đại học Bách Khoa từ năm 2006 đến năm 2017. Mỗi bản ghi thể hiện thông tin về sinh viên, về cách đánh giá của môn học và điểm số cụ thể của sinh viên cho môn học đó. Điểm số này được giới hạn trong khoảng từ 0 đến 10. Độ thưa của dữ liệu (Sparsity) của tập dữ liệu được tính bằng công thức sau:

$$S = 1 - \frac{G}{N \cdot C} \quad (4.1)$$

trong đó,  $G$ ,  $N$  và  $C$  lần lượt là tổng số bản ghi chứa điểm của sinh viên cho các môn học, tổng số sinh viên và tổng số môn học.

Số lượng sinh viên của mỗi khoa là không đồng đều, khoa có số lượng sinh viên đông nhất là khoa *Kỹ thuật Xây dựng* với số lượng sinh viên là 11691 (chiếm ≈ 19% tổng sinh viên toàn trường), khoa có số lượng sinh viên ít nhất là khoa *Chất lượng cao* với số lượng sinh viên là 565 (chiếm ≈ 0.9% tổng sinh viên toàn trường). Trực quan về tỷ lệ sinh viên các khoa được thể hiện như **Hình 4.4**.



Hình 4.4: Trực quan về số lượng sinh viên giữa các khoa trong tập dữ liệu

Chi tiết thông tin của dữ liệu 14 khoa đào tạo trong tập dữ liệu được thể hiện trong **Bảng 4.4**.

Định dạng ban đầu của những file dữ liệu này là định dạng excel (.xlsx). Trong tổng số 33 trường trên, không phải trường nào của dữ liệu cũng phục vụ cho việc phân tích và xây dựng mô

## Chương 4 PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT

Bảng 4.4: Thống kê chi tiết dữ liệu của 14 Khoa thuộc đại học Bách Khoa

Khoa	Ký hiệu khoa	Số lượng môn học	Số lượng sinh viên	Số lượng bản ghi	Sparsity
Khoa học và Kỹ thuật Máy tính	MT	168	5158	155574	0.8205
Bảo dưỡng công nghiệp	BD	116	1958	54976	0.7580
Cơ khí	CK	435	9233	351539	0.9125
Kỹ thuật địa chất và dầu khí	DC	207	2476	93516	0.8175
Điện - điện tử	DD	325	9391	360546	0.8819
Kỹ thuật giao thông	GT	230	2323	88510	0.8343
Kỹ thuật hóa học	HC	322	6117	222478	0.8870
Môi trường và tài nguyên	MO	177	2401	90633	0.7867
Quản lý công nghiệp	QL	137	3577	104514	0.7867
Khoa học ứng dụng	UD	192	2099	78986	0.8040
Kỹ thuật vật liệu	VL	183	2910	109612	0.7942
Việt Pháp	VP	309	1515	92040	0.8039
Kỹ thuật xây dựng	XD	445	11691	450209	0.9135
Chất lượng cao	PD	89	565	16912	0.6637

hình dự đoán. Đồng thời, dữ liệu trong các trường, ví dụ như trường *Tổng kết* không phải là dữ liệu chuẩn theo một định dạng nhất định, vì nó vừa có điểm là số, vừa có điểm là chữ. Do đó, cần phải thực hiện bước tiền xử lý dữ liệu để chuẩn hóa lại dữ liệu cho phù hợp với yêu cầu của bài toán.

**Xử lý dữ liệu đại học ở bước tiền xử lý bao gồm những công việc chính như sau:**

- Đối với trường điểm tổng kết môn học, có nhiều điểm số bằng chữ (Bảng ??):
  - Đối với các điểm theo quy định là bị tính như điểm 0, đó là điểm không đạt (KD), cầm thi (CT), vắng thi (VT) thì sẽ được quy về 0 điểm.
  - Đối với các điểm: Miễn học, miễn thi (MT), hoãn thi (HT), chưa có điểm (CH), rút môn (RT), đạt (DT), vắng thi có phép (VP) thì theo quy định như môn này không tính vào điểm trung bình tích lũy, cũng như không có nhiều ý nghĩa cho việc phân tích nên sẽ bị loại bỏ.

Sau khi chuẩn hóa, tập dữ liệu có phân bố điểm số như **Hình 4.5**, các **Hình 4.6 to 4.19** thể hiện phân bố điểm của từng khoa.

- Đối với các điểm số bằng số và lớn hơn 10, sẽ được quy đổi tương tự như điểm số bằng chữ trong bảng trên, và được giải quyết tương tự như đối với trường hợp các điểm số bằng chữ.
- Đối với trường hợp một sinh viên học một môn học nhiều lần, điểm sau cùng được ghi nhận là điểm số cao nhất trong các lần học, do vậy, chỉ lấy lần học có điểm số cao nhất, các lần học còn lại sẽ được loại bỏ.

## Chương 4 PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT

Bảng 4.5: Các điểm số lớn hơn 10 và điểm bằng chữ ở trường điểm tổng kết môn học

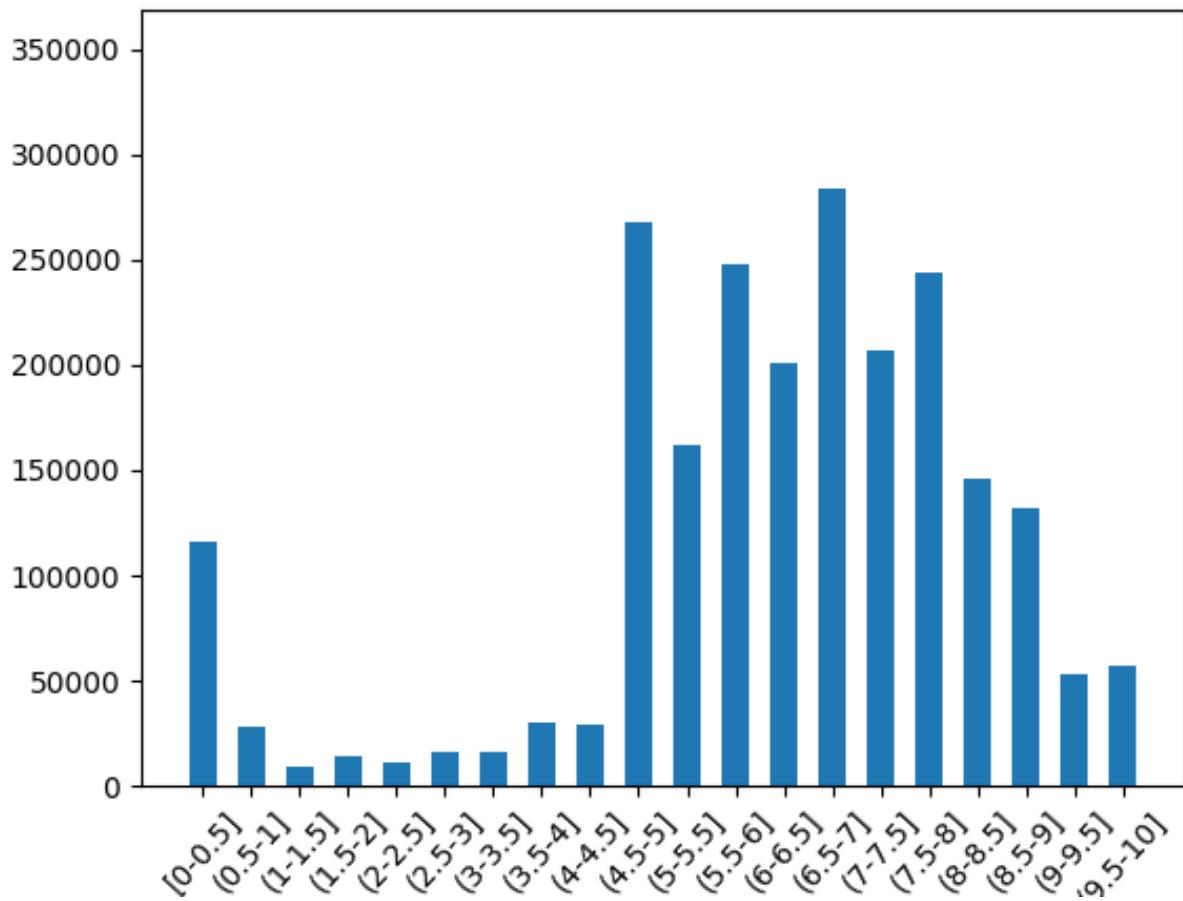
Loại điểm	Điểm số	Điểm chữ	Ý nghĩa
Cấm thi	11	CT	Được tính như điểm 0
Miễn học, miễn thi	12	MT	Đạt nhưng không tính vào điểm trung bình
Vắng thi	13	VT	Được tính như điểm 0
Hoãn thi, được phép thi sau	14	HT	Không đạt và không tính vào điểm trung bình
Chưa có điểm	15	CH	Chưa tính điểm
Rút môn học	17	RT	Không ghi vào bảng điểm
Không đạt	20	KD	Được tính như điểm 0
Đạt	21	DT	Đạt nhưng không tính vào điểm trung bình
Vắng thi có phép	22	VP	Không đạt và không tính vào điểm trung bình

Bảng 4.6: Các môn học với nhiều mã được mapping về một mã thông nhất

Tên môn học	Các mã môn học	Mã môn học thông nhất được mapping
Lập trình hướng đối tượng	CO2005, 502001	CO2005
Mô hình hóa hình học và mô phỏng	200013, 209009, ME3027	ME3027
Kỹ thuật cao áp	403003, 403113, EE3089	EE3089
Mạch điện tử thông tin	405403, 405003, EE3011	EE3011
Xác suất và thống kê	6018, 6048, 6818, 6718, 6805, MT2001	MT2001
Hóa đại cương	604045, 604001, 604040, 604601, CH1003	CH1003
Thực tập tốt nghiệp (Xây dựng)	801301, 804302, 810302, CI3313	CI3313
...	...	...

- Liên quan đến việc phân tích sau này, vì dữ liệu không có trường biểu thị sinh viên thuộc khóa nào, nên từ trường tên lớp sẽ tạo ra thêm một trường Khóa. Trường tên lớp có định dạng như ví dụ sau: MT15KHTN, CK13CK06, ... => 2 số phía sau 2 chữ cái bắt đầu sẽ cho biết sinh viên thuộc khóa nào.
- Những trường không phục vụ cho việc phân tích, dự đoán sẽ được loại bỏ.
- Sau khi lọc bỏ các trường không sử dụng và trích xuất thêm các trường thông tin cần thiết phục vụ cho việc giải quyết bài toán, do các môn học được lặp lại khá nhiều, cùng một môn học theo các khóa đào tạo khác nhau lại có những mã khác nhau, điều này được thể hiện minh họa trong **Bảng 4.6**, nên chúng tôi đã mapping các môn học có nhiều mã (nhưng thực chất là một môn) về một mã môn học chung thông nhất. Số lượng môn học trước và sau khi được mapping được thể hiện như **Hình 4.20**. Việc mapping môn học về những mã chung thông nhất giúp giảm bớt số lượng môn học, khi thực hiện các kỹ thuật dự đoán làm cho việc dự đoán được chính xác và hiệu quả hơn nhờ vào việc mỗi môn học có nhiều dữ liệu hơn.

Từ dữ liệu đầu vào là những file excel (xlsx) qua quá trình tiền xử lý dữ liệu sẽ tạo nên file



Hình 4.5: Phân bố điểm số trong tập dữ liệu

dữ liệu chuẩn phục vụ cho bài toán, được lưu dưới dạng csv, bởi vì:

- Định dạng csv được hỗ trợ trên các công cụ như spark tốt hơn định dạng excel.
- Đồng thời liên quan đến vấn đề big data, sử dụng csv sẽ mang nhiều ý nghĩa hơn excel.

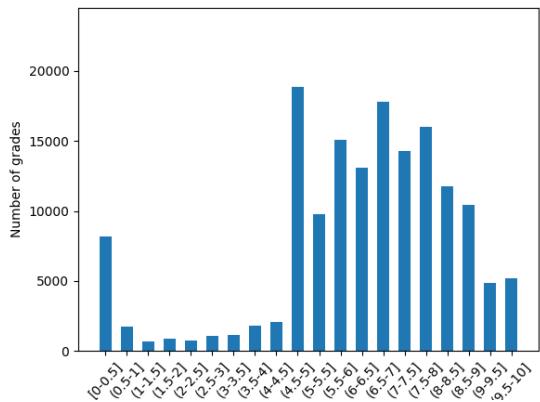
**DataFrame** chuẩn, được sử dụng làm input cho bài toán có những trường sau (11 trường): năm học, học kỳ, mã môn học, tên môn học, mã khoa, khóa, mã ngành, tên ngành, tên lớp, mã sinh viên, điểm tổng kết.

### 4.3 Mô hình giải quyết bài toán

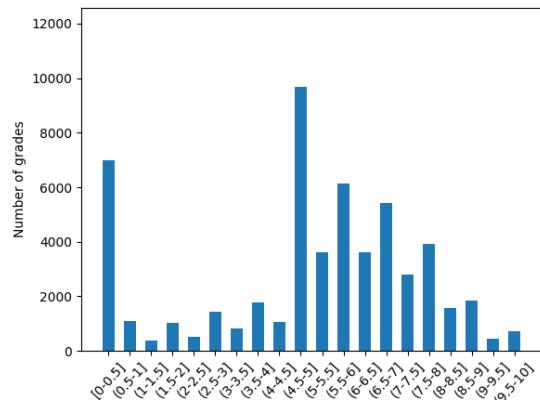
Hệ thống phân tích dữ liệu đại học do chúng tôi xây dựng bao gồm những thành phần sau:

1. **Web API Server:** Cung cấp REST API để cho Web Server gọi.
2. **Thrift Server:** Để Web API Server giao tiếp với máy chủ cài đặt Spark.
3. **Prediction Module (Spark):** Xử lý giải thuật dự đoán, đưa ra các kết quả dự đoán.
4. **Recommendation Module (Spark):** Xử lý giải thuật đề xuất (recommendation) và đưa ra đề xuất về môn học cho sinh viên.

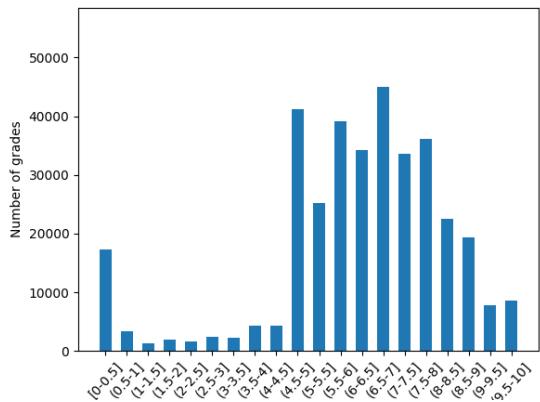
## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT



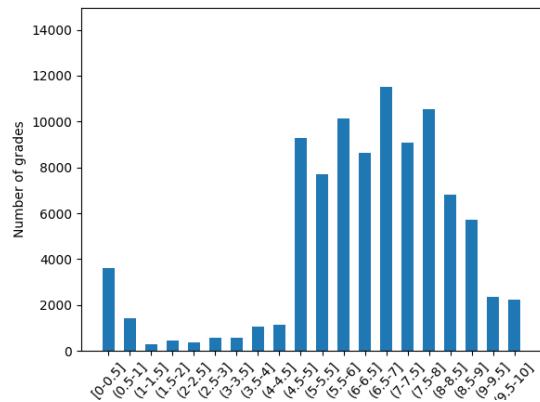
Hình 4.6: Phân bố điểm khoa MT



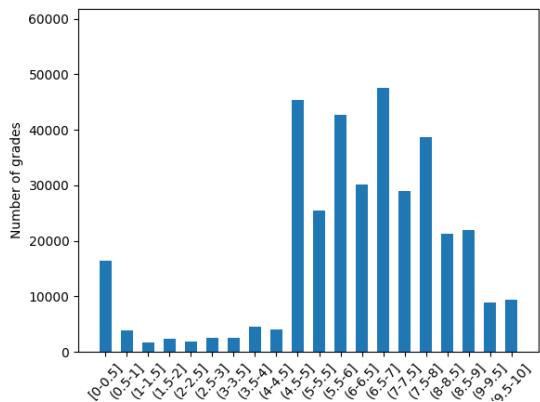
Hình 4.7: Phân bố điểm khoa BD



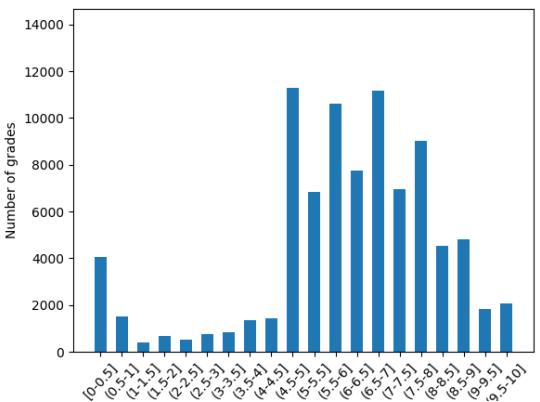
Hình 4.8: Phân bố điểm khoa CK



Hình 4.9: Phân bố điểm khoa DC

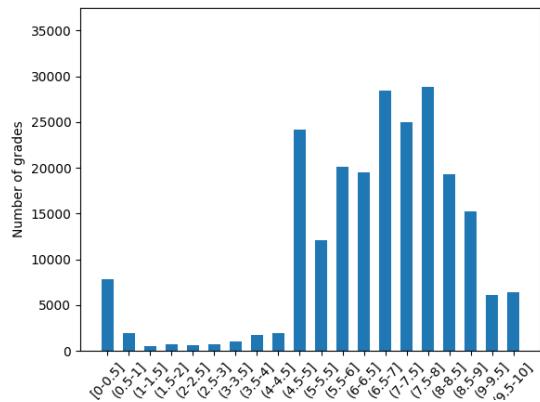


Hình 4.10: Phân bố điểm khoa DD

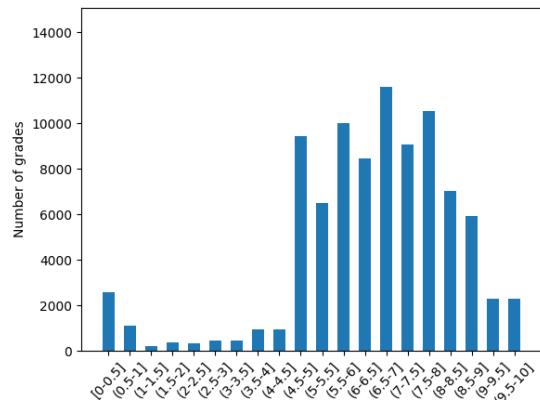


Hình 4.11: Phân bố điểm khoa GT

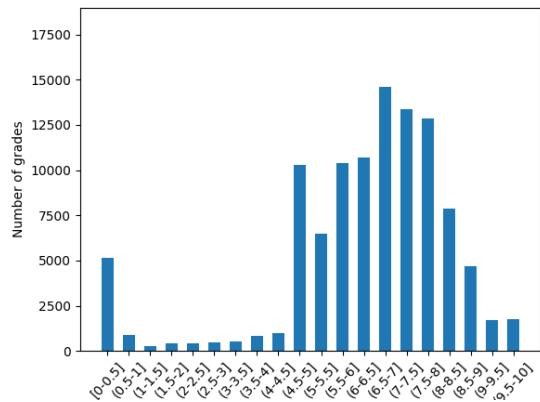
## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT



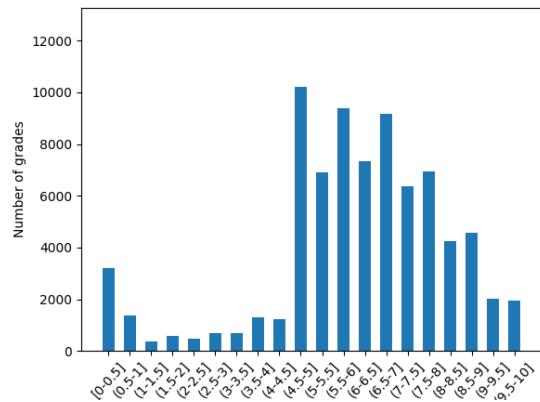
Hình 4.12: Phân bố điểm khoa HC



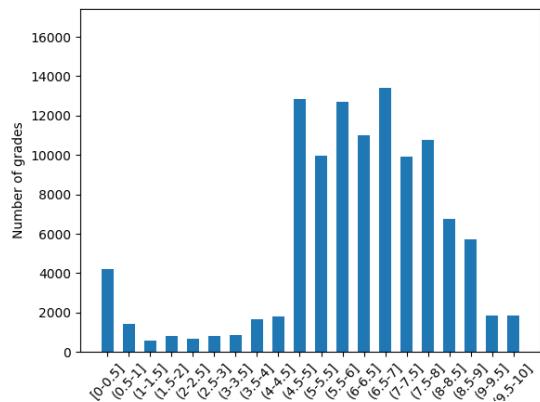
Hình 4.13: Phân bố điểm khoa MO



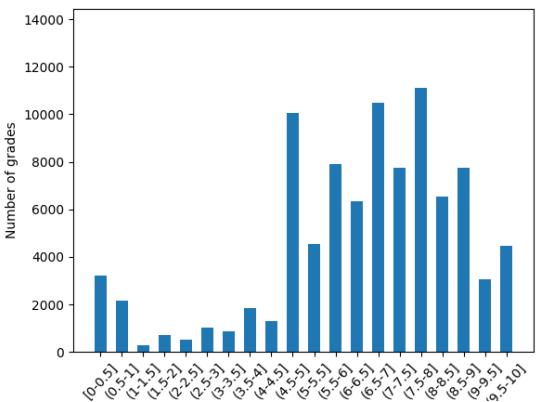
Hình 4.14: Phân bố điểm khoa QL



Hình 4.15: Phân bố điểm khoa UD

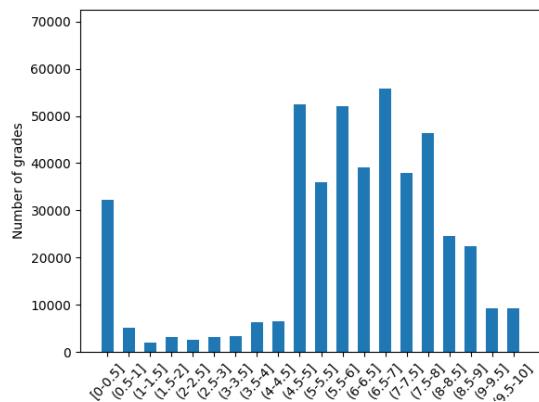


Hình 4.16: Phân bố điểm khoa VL

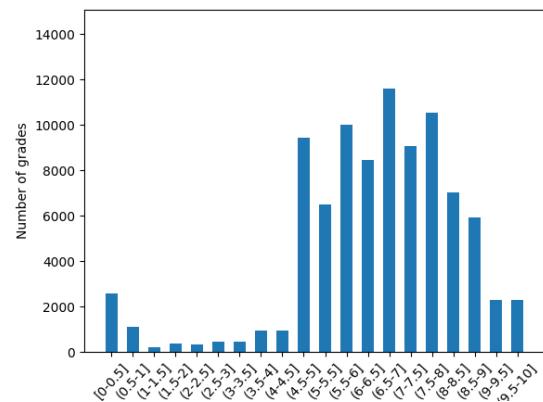


Hình 4.17: Phân bố điểm khoa VP

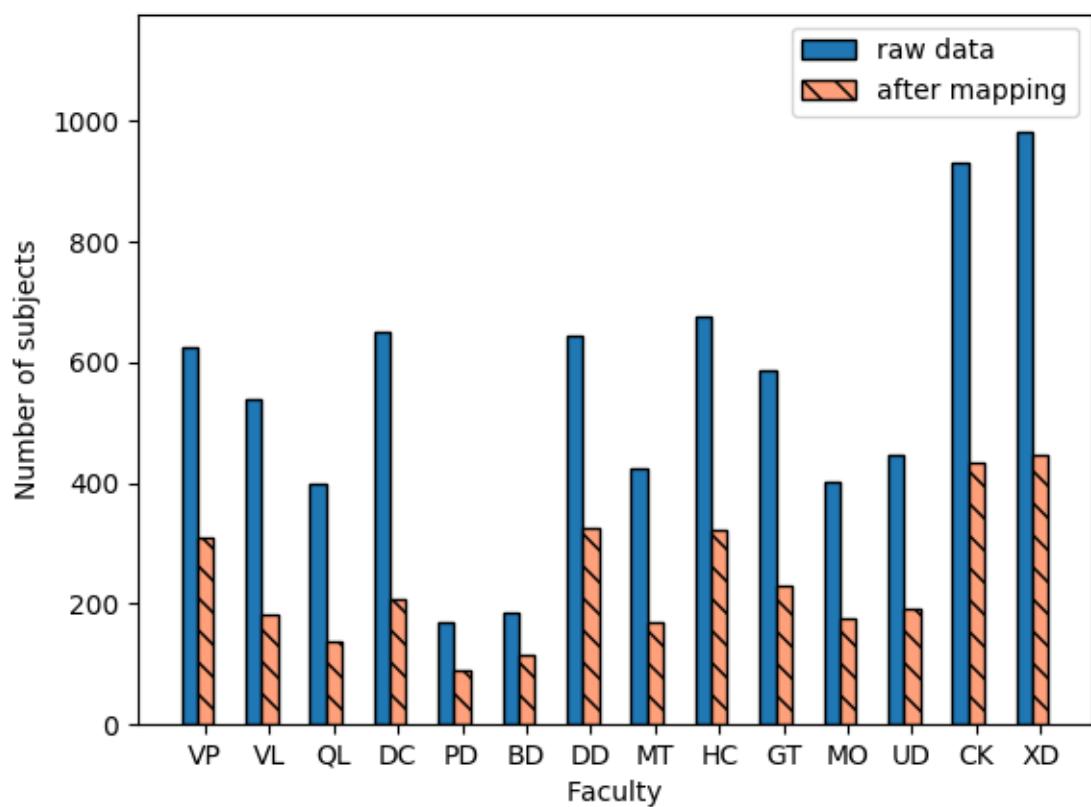
## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT



Hình 4.18: Phân bố điểm khoa XD



Hình 4.19: Phân bố điểm khoa PD



Hình 4.20: Số lượng môn học trước và sau khi thực hiện mapping

### 4.3.1 Web API Server

Được viết bằng Java sử dụng thư viện Jetty. Là nơi cung cấp REST API để Web Server giao tiếp với hệ thống dự đoán. Gồm 3 endpoint:

```
1 /predict  
2 /predict/top  
3 /recommend
```

#### Endpoint /predict

Chức năng: Dự đoán tất cả điểm của các môn học còn lại mà sinh viên chưa học.

Input:

```
1 {  
2     "algorithm": ① ,  
3     "data": {  
4         "user": ② ,  
5         "items": ③ ,  
6         "ratings": ④ ,  
7         "faculty": ⑤  
8     }  
9 }
```

Trong đó:

- ①: kiểu string cho biết giải thuật muốn sử dụng để dự đoán, các giá trị có thể có hiện tại gồm:
  1. als (Alternative Least Square).
  2. als\_nn (Non-negative Alternative Least Square).
  3. ibcf (Item-based Collaborative Filtering).
  4. ubcf (User-based Collaborative Filtering).
  5. als\_ibcf (Item-based Collaborative Filtering on Item Factor Matrix of Alternative Least Square).
  6. als\_nn\_ibcf (Item-based Collaborative Filtering on Item Factor Matrix of Non-negative Alternative Least Square).
- ②: kiểu string cho biết mã sinh viên của sinh viên muốn dự đoán.
- ③: một array các string cho biết các môn học đã học của sinh viên.
- ④: một array các double cho biết điểm của các môn đã học tương ứng.
- ⑤: kiểu string cho biết khoa của sinh viên.

Ví dụ:

```
1 {  
2     "algorithm": "als",  
3     "data": {  
4         "user": "1513293",  
5     }  
6 }
```

## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT

```
5         "items": ["C01011", "PE1003", "C01007"],  
6         "ratings": [9.0, 9.5, 9.3],  
7         "faculty": "MT"  
8     }  
9 }
```

Output:

```
1 {  
2     "user": ①,  
3     "items": ②,  
4     "ratings": ③,  
5     "status": ④  
6 }
```

Trong đó:

- ①: kiểu string cho biết mã sinh viên của sinh viên muốn dự đoán.
- ②: một array các string cho biết các môn học còn lại của sinh viên.
- ③: một array các double cho biết điểm của các môn học còn lại tương ứng.
- ④: kiểu integer cho biết trạng thái của việc dự đoán, có giá trị âm nếu xảy ra lỗi, 0 nếu không có lỗi.

Ví dụ:

```
1 {  
2     "user": "1513293",  
3     "items": ["C01013", "PE1005"],  
4     "ratings": [9.0, 9.1],  
5     "status": 0  
6 }
```

### Endpoint /predict/top

Chức năng: Lấy  $k$  môn học có điểm dự đoán cao nhất mà sinh viên chưa học.

Input:

```
1 {  
2     "algorithm": ①,  
3     "data": {  
4         "user": ②,  
5         "items": ③,  
6         "ratings": ④,  
7         "faculty": ⑤  
8     }  
9     "count": ⑥  
10 }
```

Trong đó:

- ①: kiểu string cho biết giải thuật muốn sử dụng để dự đoán.

## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT

- ②: kiểu string cho biết mã sinh viên của sinh viên muốn dự đoán.
- ③: một array các string cho biết các môn học đã học của sinh viên.
- ④: một array các double cho biết điểm của các môn đã học tương ứng.
- ⑤: kiểu string cho biết khoa của sinh viên.
- ⑥: kiểu integer cho biết số lượng môn học có điểm cao nhất muốn lấy.

Output: Như endpoint /predict.

### Endpoint /recommend

Chức năng: Sử dụng luật kết hợp để gợi ý môn học tiếp theo cho sinh viên và điểm dự đoán tương ứng.

Input:

```
1 {
2     "data": {
3         "user": ①,
4         "items": ②,
5         "ratings": ③,
6         "faculty": ④
7     }
8 }
```

Trong đó:

- ①: kiểu string cho biết mã sinh viên của sinh viên muốn dự đoán.
- ②: một array các string cho biết các môn học đã học của sinh viên.
- ③: một array các double cho biết điểm của các môn đã học tương ứng.
- ④: kiểu string cho biết khoa của sinh viên.

Output: Như endpoint /predict.

### 4.3.2 Thrift Server

Được dùng để Web API Server gọi hàm trên máy chủ master cài đặt Spark. Các bước chạy khi nhận được request gọi hàm:

1. Chuẩn bị Spark Job và các thông số cần truyền cho giải thuật như: mã sinh viên, môn học đã học, điểm tương ứng và khoa dưới dạng string format json.
2. Submit Job lên Spark cluster và lắng nghe sự kiện kết thúc job.
3. Đọc file và parse kết quả được ghi ra bởi Spark và trả về kết quả.

### 4.3.3 Prediction Module

#### Tổng quan:

Module chứa các giải thuật để dự đoán được viết theo Spark Framework hỗ trợ xử lý phân bố. Các giải thuật hỗ trợ hiện tại gồm có:

- Matrix Factorization. Gồm hai dạng:
  - Alternative Least Square.
  - Non-negative Alternative Least Square.
- Collaborative Filtering: gồm hai giải thuật:
  - User-based Collaborative Filtering.
  - Item-based Collaborative Filtering
- Item-based Collaborative Filtering on Item Factor Matrix of Matrix Factorization: Gồm hai dạng:
  - Item-based Collaborative Filtering on Item Factor Matrix of Matrix Factorization.
  - Item-based Collaborative Filtering on Item Factor Matrix of Non-negative Matrix Factorization

Bảng 4.7: Một mẫu ví dụ trong tập dữ liệu

Student ID	Course ID	Grade
1511000	0	9
1511000	1	8
1512000	1	7
1512000	0	8
1512000	2	7.5
1512000	3	8.5
1513000	0	7.5
1513000	2	7.5
1513000	1	8.5
<b>1511000</b>	<b>2</b>	<b>?</b>

**Matrix Factorization:** Được hỗ trợ sẵn trong thư viện ML của Spark. Được hiện thực bằng phương pháp Alternative Least Square nhằm hỗ trợ song song hóa, đồng thời cung cấp lựa chọn non-negative bằng cách thêm vào ràng buộc không âm trong lúc giải least squares problem. Trong ví dụ ở **Bảng 4.7**, phân bố điểm có thể được biểu diễn dưới dạng ma trận như **Bảng 4.8**. Ma trận này sẽ được factorized thành hai ma trận Student Factor (User Factor) và Course Factor (Item Factor). Tích của hai ma trận này sẽ tạo thành ma trận dự đoán, kết quả dự đoán của các giá trị chưa biết là giá trị của ma trận dự đoán tương ứng. Ví dụ giá trị cần dự đoán trong **Bảng 4.8** sẽ được dự đoán với giá trị **9.2** như trong **Bảng 4.9**.

**Collaborative Filtering:** Được hiện thực bằng cách sử dụng DataFrame API hỗ trợ bởi Spark. Collaborative Filtering ở đây được xây dựng theo cả 2 dạng là User-Based Collaborative Filtering và Item-Based Collaborative Filtering.

**Ví dụ:** Với dữ liệu điểm số sinh viên đã biết ở **Bảng 4.7**, được biểu diễn dưới dạng ma trận bao gồm các cột là các môn học, mỗi hàng biểu diễn cho tập điểm của một sinh viên ở các môn

Bảng 4.8: Biểu diễn phân bố điểm của sinh viên dưới dạng ma trận

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1511000</b>	9	8	?	?
<b>1512000</b>	8	7	7.5	8.5
<b>1513000</b>	7.5	8.5	7.5	?

Bảng 4.9: Ví dụ ma trận dự đoán

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1511000</b>	9.1	7.6	9.2	8.7
<b>1512000</b>	8	7.7	6.7	8.2
<b>1513000</b>	7.4	8.9	7.3	9

học tương ứng (*Bảng 4.9*), ta cần dự đoán điểm số của sinh viên có mã sinh viên 1511000 ở môn học số 2.

- **Với User-Based Collaborative Filtering:** để dự đoán điểm số của sinh viên 1511000 ở môn học 2, đầu tiên, độ tương đồng giữa sinh viên 1511000 với các sinh viên khác đã học môn học 2 sẽ được tính toán (sử dụng độ đo Cosine similarity). *Bảng 4.10* thể hiện các vector feature của sinh viên (user's feature), mỗi feature là điểm số của sinh viên ở lần lượt các môn: [môn 0, môn 1, môn 2, môn 3]. Dựa vào độ đo Consine similarity, độ tương đồng giữa sinh viên 1511000 với các sinh viên 1512000 và 1513000 được tính toán lần lượt là 0.68402 và 0.82787.

Với độ tương đồng vừa tính được, giải thuật đưa ra dự đoán bằng cách tính trung bình có trọng số độ tương đồng với điểm số tương ứng của sinh viên 1512000 và 1513000 để đưa ra dự đoán điểm số cho sinh viên 1511000 ở môn 2 là:

$$\text{Điểm số dự đoán} = \frac{0.68402 \cdot 7.5 + 0.82787 \cdot 7.5}{0.68402 + 0.82787} = 7.5$$

Như vậy với module dự đoán điểm số dùng User-Based Collaborative Filtering, điểm số dự đoán sẽ là 7.5.

- **Với Item-Based Collaborative Filtering:** để đưa ra dự đoán điểm cho sinh viên 1511000 ở môn học 2, thay vì tính độ tương đồng dựa vào user's feature như User-Based Collaborative Filtering, IBCF sẽ tính độ tương đồng dựa vào feature của các môn học (item's features). *Bảng 4.11* thể hiện feature của các môn học và độ tương đồng giữa môn học 2 với các môn học khác dựa vào feature này. Trong đó, mỗi feature là một vector điểm số tương ứng của các sinh viên [1511000, 1512000, 1513000] ở môn học tương ứng.

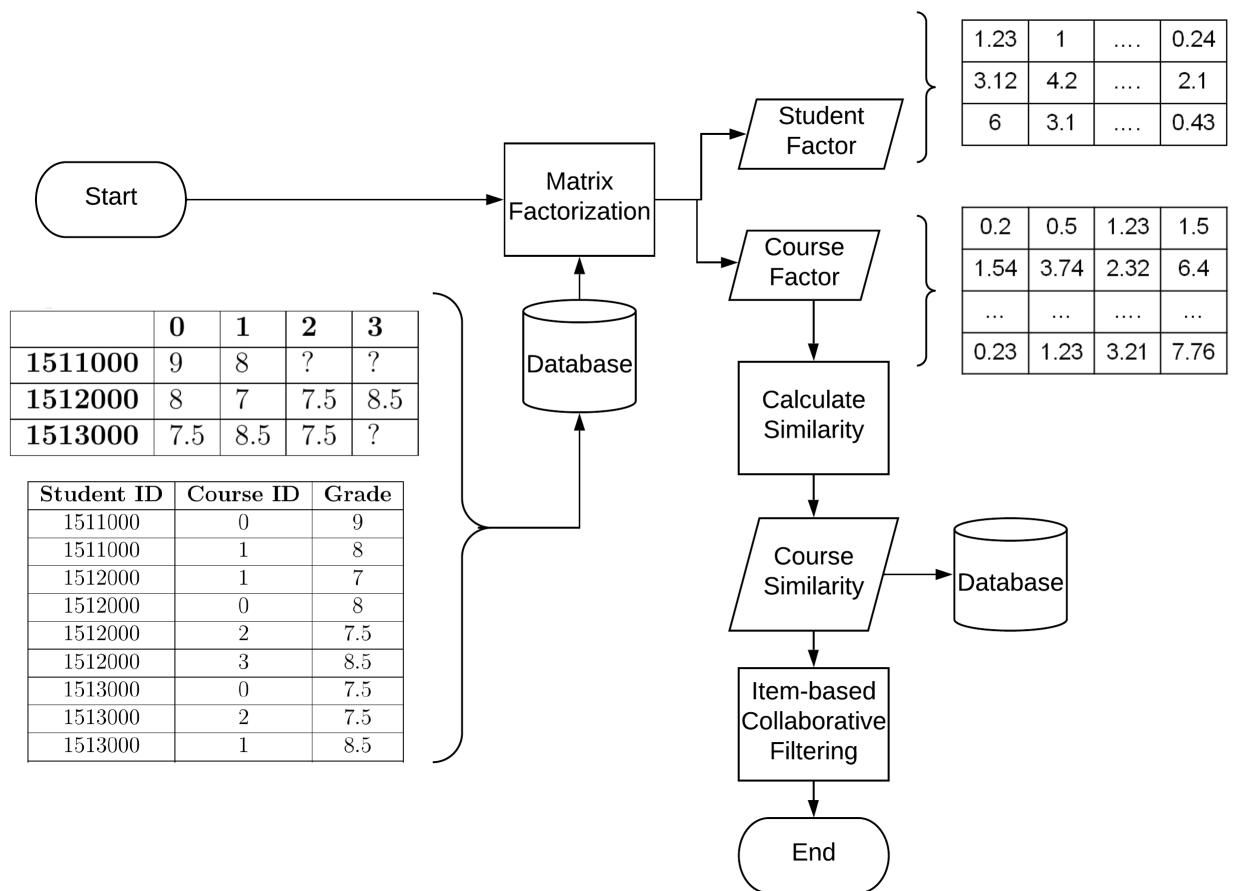
Kết quả dự đoán cũng sẽ được tính dựa vào trung bình có trọng số giữa độ tương đồng và điểm số của chính sinh viên đó trong các môn đã học:

$$\text{Điểm số dự đoán} = \frac{0.77259 \cdot 9 + 0.80526 \cdot 8}{0.77259 + 0.80526} = 8.48960$$

Với IBCF, điểm số dự đoán lúc này là  $\approx 8.5$ .

#### Item-based Collaborative Filtering on Item Factor Matrix of Matrix Factorization:

## Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT



Hình 4.21: Item-based Collaborative Filtering on Item Factor Matrix of Matrix Factorization

## Chương 4 PHÂN TÍCH VẤN ĐỀ VÀ CÁC GIẢI PHÁP ĐỀ XUẤT

Bảng 4.10: Similarity Score between student 1511000 and other students

1511000's feature	[9, 8, 0, 0]
1512000's feature	[8, 7, 7.5, 8.5]
1513000's feature	[7.5, 8.5, 7.5, 0]
Similarity(1511000, 1512000)	0.68402
Similarity(1511000, 1513000)	0.82787

Bảng 4.11: Similarity Score between course 2 and other courses

course 0's feature	[9, 8, 7.5]
course 1's feature	[8, 7, 8.5]
course 2's feature	[0, 7.5, 7.5]
Similarity(course 2, course 0)	0.77259
Similarity(course 2, course 1)	0.80526

Kết quả của phương pháp Alternative Least Square sẽ cho ra được ma trận Item Factor biểu thị độ tương thích của các môn học đối với các hidden features. Ta có thể áp dụng phương pháp Item-based Collaborative Filtering vào ma trận Item Factor này để tìm độ giống nhau giữa các môn học dựa trên các hidden features. Sau đó tiến hành dự đoán điểm như phương pháp Item-based Collaborative Filtering thông thường. Các bước thực hiện của phương pháp này được biểu diễn ở **Hình 4.21**.

### 4.3.4 Recommendation Module

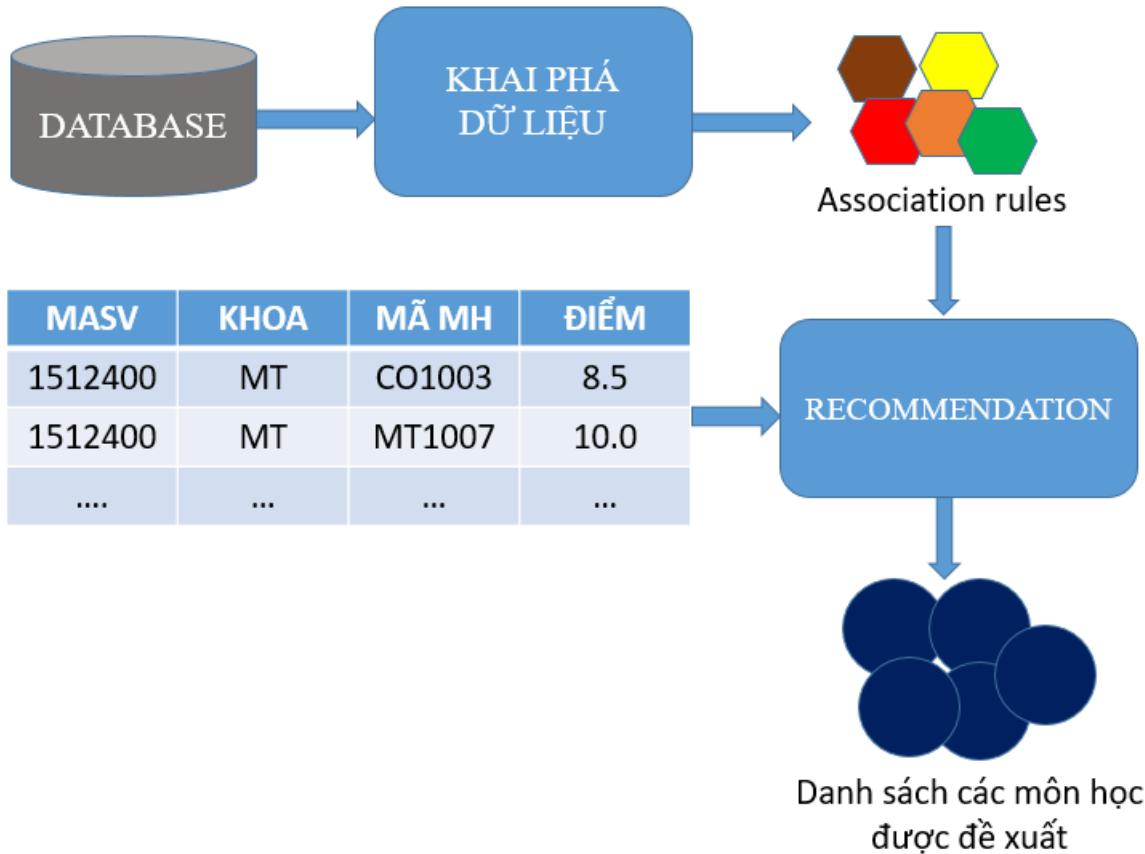
Recommendation Module có chức năng đưa ra các đề xuất về môn học tiếp theo nên học cho sinh viên. Tổng quan về recommendation module được thể hiện như **Hình 4.22**. Giải thuật được sử dụng để xây dựng recommendation module là FP-Growth. Theo đó, dựa trên cơ sở dữ liệu học tập của sinh viên trong tập dữ liệu, qua sử dụng giải thuật FP-Growth, chúng tôi có được các luật kết hợp (Association rules), những luật kết hợp này được lưu lại để sử dụng cho việc đề xuất. Khi thực hiện đề xuất các môn học cho sinh viên, Recommendation module sẽ kết hợp dữ liệu đầu vào của sinh viên và các luật kết hợp khai phá được để đề xuất các môn học thích hợp cho sinh viên.

Với recommendation module, input và output như sau:

- **Input:** Dữ liệu các môn học mà sinh viên đã học.
- **Output:** Các môn học đề xuất cho sinh viên.

Việc đề xuất các môn học cho sinh viên được thực hiện theo 3 phương pháp như sau:

- *Dựa vào những sinh viên khác trong tập dữ liệu:* việc đề xuất các môn học tiếp theo cho sinh viên sẽ dựa vào những sinh viên khác trong tập dữ liệu có những môn học giống như sinh viên hiện tại đã học. Phương pháp này đề xuất cho sinh viên những môn học tiếp theo chỉ dựa vào dữ liệu quá trình học tập của những sinh viên trong tập dữ liệu mà hoàn toàn không dựa vào điểm số.



Hình 4.22: Recommendation module

- *Dựa vào điểm số sau khi dự đoán:* với phương pháp này, module dự đoán sẽ dự đoán tất cả các môn học khác mà sinh viên chưa học, sau đó module đề xuất sẽ dựa vào kết quả dự đoán để chọn ra những môn học có kết quả dự đoán tốt nhất để đề xuất cho sinh viên.
- *Kết hợp 2 phương pháp trên:* với phương pháp này, dựa vào dữ liệu học tập của những sinh viên khác, recommendation module sẽ đề xuất các môn học tiếp theo nên học cho sinh viên, tiếp theo, những môn học được đề xuất này sẽ được đưa vào module dự đoán để dự đoán kết quả điểm số cụ thể. Cuối cùng, những môn học được dự đoán có số điểm cao nhất sẽ được đề xuất cho sinh viên.

### 4.3.5 Mở rộng

Một giải thuật khác có thể được sử dụng trong Prediction Module là phương pháp sử dụng RBM. Các bước thực hiện:

1. Điểm số đầu vào là các số thực sẽ được làm tròn thành các số nguyên gần nhất với giá trị 0 - 10.
2. Các số nguyên này được biểu diễn lại dưới dạng binary softmax units - một mảng gồm 11 giá trị binary từ 0 đến 10, điểm số của sinh viên là bao nhiêu thi giá trị tại index đó sẽ là 1, các giá trị còn lại sẽ là 0 → trong mảng binary softmax units chỉ có một giá trị là 1.
3. Dữ liệu ở dạng binary softmax unit sẽ được truyền vào làm input cho quá trình huấn luyện RBM. RBM được huấn luyện với giải thuật CD với các số lượng các bước Gibbs Sampling tăng dần.

## *Chương 4 PHÂN TÍCH VĂN ĐỀ VÀ CÁC GIẢI PHÁP ĐỂ XUẤT*

4. Kết quả được tái tạo lại của RBM là xác suất của các binary softmax units được kích hoạt. Các xác suất này có thể được cộng lại chia trung bình để ra kết quả dự đoán điểm hoặc lấy xác suất cao nhất hoặc sampling.

RBM chưa được hỗ trợ native trong thư viện ML của Spark nên giải thuật này được hiện thực lại bằng Dataframe API của Spark.

# Chương 5

## THỰC NGHIỆM VÀ KẾT QUẢ

Nội dung chính của chương này sẽ trình bày về thực nghiệm môi trường chúng tôi xây dựng hệ thống dự đoán điểm với Apache Spark, các độ đo dùng để đánh giá mức độ chính xác dựa của các giải thuật được áp dụng để xây dựng mô hình dự đoán và kết quả của từng giải thuật tương ứng với các độ đo này. Ở chương này, chúng tôi cũng sẽ trình bày về cách chúng tôi tiến hành các thực nghiệm và các so sánh, đánh giá về kết quả qua các thí nghiệm được tiến hành.

### 5.1 Môi trường thực nghiệm

#### 5.1.1 Môi trường

Các thực nghiệm sẽ được chạy trên hệ thống SuperNode-XP với 24 node tính toán. Mỗi node tính toán có cấu hình như sau:

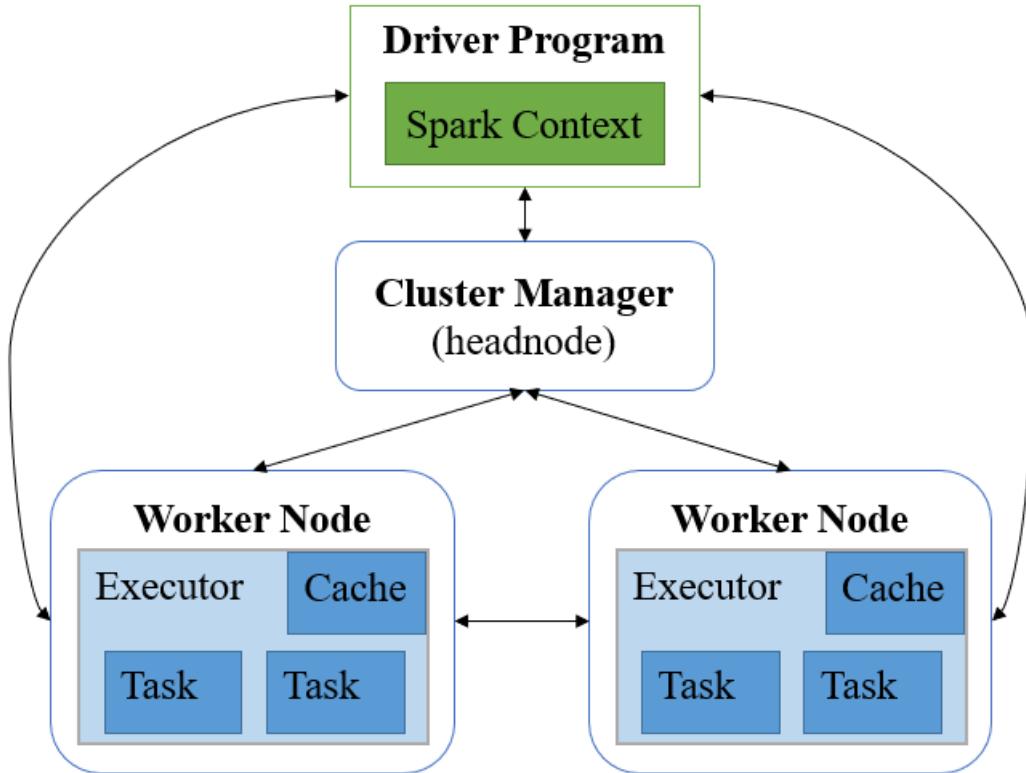
- 2 CPU socket - Intel Xeon E5 - 2680 v3 @ 2.70GHz.
- 2 Intel Xeon Phi 7120 (Knight Corners) cards.
- RAM tùy thuộc vào từng node.

Spark cluster chạy được xây dựng trên 4 node gồm 1 master và 3 worker (**Hình 5.1**). Trong đó, mỗi worker node có bộ nhớ RAM là 256GB và master node (headnode) có bộ nhớ RAM là 128GB, thông số chi tiết được thể hiện trong **Bảng 5.1**.

Ngoài ra các phần mềm sau được sử dụng để xây dựng nền hệ thống:

Bảng 5.1: Thông số cấu hình của các node trong spark cluster

No.	Loại node	Bộ nhớ RAM	Số core
1	Master node (headnode)	128 GB	48
2	Worker node (1)	256 GB	48
3	Worker node (2)	256 GB	48
4	Worker node (3)	256 GB	48



Hình 5.1: Mô hình spark cluster được sử dụng để xây dựng công cụ phân tích dữ liệu đại học

Bảng 5.2: Software Specification on the testing environment

No.	Phần mềm	Mô tả
1	Operating System	Red Hat Enterprise Linux 7.2
2	Spark	Apache Spark ver 2.4.0
3	Web Server	Jetty Version 9.4
4	Middleware	Thrift version 0.12

### 5.1.2 Thực nghiệm

Các sinh viên trong tập dữ liệu được chia thành hai loại:

- 80% sinh viên của mỗi khoa được dùng để huấn luyện mô hình. Tất cả thông tin về điểm của các sinh viên này sẽ làm đầu vào cho quá trình train.
- 20% sinh viên của mỗi khoa dùng để kiểm tra mô hình. Trong đó, 50% thông tin của các sinh viên này sẽ làm đầu vào cho các mô hình, kết quả của mô hình sẽ được so sánh với 50% thông tin còn lại của sinh viên để đánh giá mức độ chính xác.

Các thực nghiệm sẽ được tiến hành như sau:

1. Trường hợp Locality: Chỉ sử dụng dữ liệu của một khoa để huấn luyện và đánh giá kết quả dự đoán cho khoa đó (Trường hợp Locality).
2. Trường hợp Global: Sử dụng dữ liệu của tất cả các khoa để huấn luyện cho mô hình. Khi kiểm tra chỉ kiểm tra các sinh viên trong một khoa (Trường hợp Global).

## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ

3. Trường hợp Locality với dữ liệu K12: Chỉ sử dụng dữ liệu của sinh viên khóa 2012 trở lên. (Train và đánh giá theo trường hợp Locality)
4. Trường hợp Locality với dữ liệu loại bỏ điểm 0: Được thực hiện cho hai khoa MT và MO với dữ liệu huấn luyện và kiểm tra không chứa điểm 0.

Các phương pháp và mô hình được sử dụng cho các thực nghiệm được liệt kê trong **Bảng 5.3**.

Bảng 5.3: Thông tin của các giải thuật được sử dụng cho thực nghiệm

Tên	Giải thuật
Baseline	Lấy trung bình của tất cả các môn mà sinh viên đã học
IBCF	Item-based Collaborative Filtering
UBCF	User-based Collaborative Filtering
ALS	Alternative Least Square
ALS_NN	Alternative Least Square with non-negative constraint
ALS_NN_IBCF	Item-based Collaborative Filtering on Non-negative ALternative Least Square's Course Factor Matrix
ALS_IBCF	Item-based Collaborative Filtering on Normal ALternative Least Square's Course Factor Matrix

Phương pháp Baseline - Lấy trung bình của tất cả các môn mà sinh viên đã học để dự đoán cho các môn học còn lại sẽ là phương pháp cơ sở để các phương pháp khác so sánh độ chính xác. Ví dụ: với sinh viên 1511000 ở **Bảng 4.7** thì kết quả dự đoán sẽ là  $\frac{9+8}{2} = 8.5$ .

Đối với phương pháp UBCF khi được dùng cho một tập dữ liệu lớn gồm nhiều người dùng thì thời gian chạy giải thuật sẽ rất dài. Để giải quyết vấn đề này thì một giải pháp thường được sử dụng là gom cụm các người dùng và áp dụng giải thuật UBCF trên từng cụm. Khi sử dụng giải pháp trên thì trường hợp Global đối với phương pháp UBCF sẽ lại quay về trường hợp Locality. Vì thế nên trường hợp Global sẽ không được thực hiện cho UBCF.

Mỗi thực nghiệm được chạy 10 lần, với sinh viên dùng để huấn luyện và kiểm tra được chọn ngẫu nhiên lại sau mỗi lần chạy. Độ lỗi cuối cùng sẽ được tính trung bình độ lỗi của mỗi lần chạy cho mỗi giải thuật.

Tham số của các giải thuật Collaborative filtering, Matrix Factorization được thể hiện ở **Bảng 5.6** và **Bảng 5.5**:

- Đối với UBCF, rank là số lượng sinh viên gần giống với sinh viên cần dự đoán nhất cần tìm.
- Đối với IBCF, rank là số lượng môn học gần giống với môn học cần dự đoán nhất cần tìm.
- Đối với ALS và ALS\_NN, rank là số lượng hidden features.

**Bảng 5.7** thể hiện tham số của giải thuật kết hợp ALS\_IBCF và ALS\_NN\_IBCF, ý nghĩa của các tham số tương tự như trong giải thuật tương ứng.

Mô hình RBM được xây dựng và áp dụng thử trường hợp Locality vào một số khoa như khoa MO và khoa MT với số lượng hidden node là 40, mỗi visible node có 11 softmax unit ứng với điểm nguyên từ 0 – 10, hệ số học  $\lambda = 0.1$ . Contrastive Divergence step lần lượt là 1, 3, 5, 7 với số lượng vòng lặp như **Bảng 5.4**.

Bảng 5.4: Số vòng lặp của mỗi CD step

	1	3	5	7
Vòng lặp	100	50	50	50

### 5.1.3 Độ đo đánh giá

Các độ đo được sử dụng để đánh giá hiệu quả của mô hình:

- **RMSE:** Root mean square error:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (5.1)$$

- **MSE:** Mean square error:

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (5.2)$$

- **MAE:** Mean absolute error:

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (5.3)$$

Các độ đo lỗi RMSE, MSE, MAE đều có giá trị từ 0 đến  $\infty$  và giá trị càng nhỏ biểu thị mô hình càng tốt.

RMSE và MSE giá trị lỗi được bình phương trước khi được trung bình lại nên RMSE và MSE sẽ làm cho lỗi càng lớn thì kết quả RMSE và MSE càng tệ. Điều này làm cho nếu trong quá trình đánh giá dù chỉ có một số trường hợp lỗi quá cao nhưng cũng sẽ làm cho kết quả của độ đo cao dẫn đến mô hình càng tệ.

## 5.2 Kết quả

### 5.2.1 Kết quả thực nghiệm với trường hợp Locality và Global

Hai trường hợp này được thực hiện để xem xét sự liên hệ giữa các khoa, nhằm tìm xem dữ liệu của các khoa khác có ảnh hưởng nhiều đến kết quả dự đoán của một khoa cụ thể hay không. Kết quả chi tiết của hai trường hợp này được biểu diễn ở **Bảng 5.8** và **Bảng 5.9** kèm theo các biểu đồ cột so sánh các độ lỗi giữa các phương pháp trong từng khoa.

Độ lỗi của phương pháp nền Baseline có sự chênh lệch khá lớn giữa các khoa. Cụ thể phương pháp Baseline ở khoa BD có độ lỗi lớn nhất trong toàn trường với giá trị là 2.03. Ngược lại ở khoa MO và HC, RMSE của phương pháp Baseline là thấp nhất chỉ với giá trị 1.61 và 1.68 tương ứng. Mặc dù ở **Bảng 4.2** thì sparsity của khoa BD ( $\approx 75\%$ ) thấp hơn nhiều so với khoa MO ( $\approx 79\%$ ) và HC ( $\approx 89\%$ ).

Ở cả hai trường hợp, nhìn chung phương pháp ALS\_NN thường cho kết quả tốt nhất so với các phương pháp còn lại và mọi phương pháp đều có thể cho hiệu quả cao hơn phương pháp Baseline.

Giữa phương pháp có ràng buộc không âm và không có ràng buộc không âm thì phương pháp có ràng buộc không âm thường có độ lỗi thấp hơn, cụ thể:

## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ

Bảng 5.5: Tham số của các giải thuật Matrix Factorization

Khoa	Rank	
	ALS	ALS_NN
MT	1	2
BD	2	2
CK	1	3
DC	1	1
DD	3	3
GT	2	2
HC	1	3
MO	2	1
PD	1	1
QL	1	2
UD	3	3
VL	2	2
VP	2	2
XD	2	3

Khoa	Rank	
	ALS	ALS_NN
MT	2	2
BD	2	1
CK	3	2
DC	1	2
DD	3	3
GT	2	2
HC	2	2
MO	1	1
PD	2	1
QL	2	2
UD	2	2
VL	3	2
VP	2	7
XD	3	3

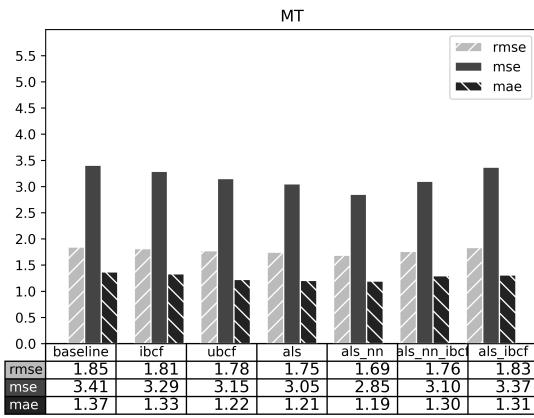
Bảng 5.6: Tham số của các giải thuật Collaborative Filtering

Khoa	Rank	
	IBCF	UBCF
MT	11	15
BD	11	23
CK	15	15
DC	13	7
DD	9	15
GT	17	7
HC	11	15
MO	17	5
PD	11	7
QL	9	15
UD	11	11
VL	13	13
VP	23	13
XD	11	25

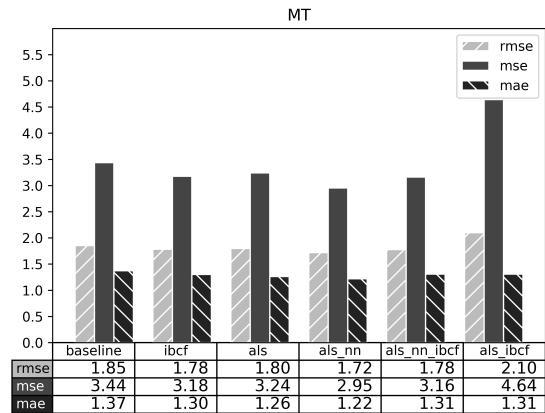
  

Khoa	Rank	
	IBCF	UBCF
MT	25	—
BD	25	—
CK	25	—
DC	23	—
DD	15	—
GT	25	—
HC	25	—
MO	20	—
PD	20	—
QL	17	—
UD	25	—
VL	25	—
VP	20	—
XD	11	—

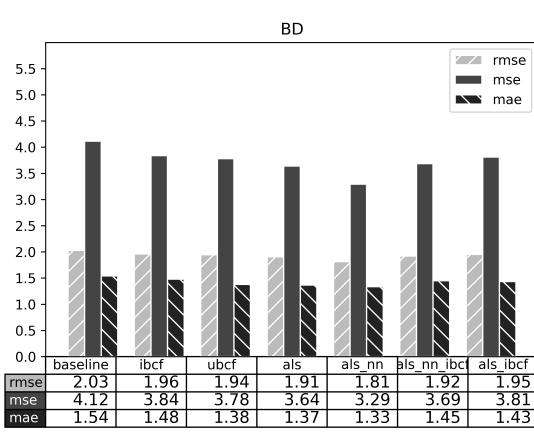
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



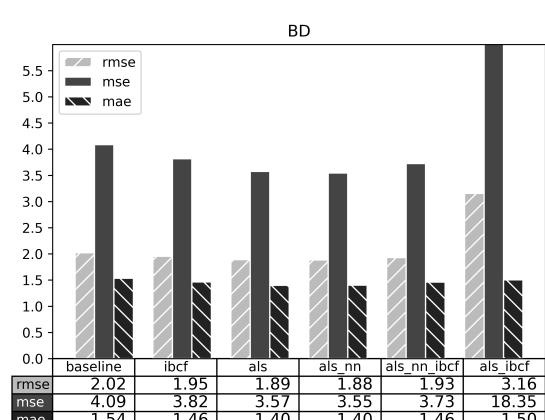
Hình 5.2: Kết quả trường hợp Locality khoa MT



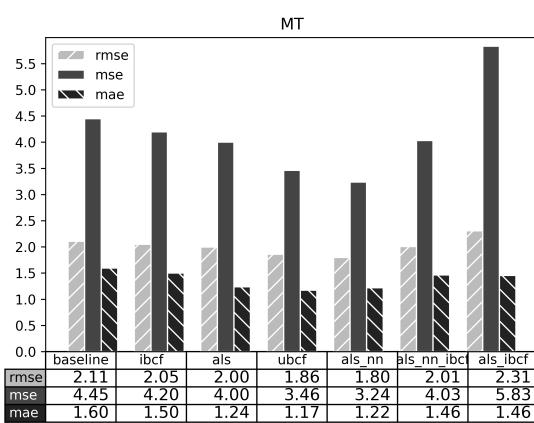
Hình 5.3: Kết quả trường hợp Global khoa MT



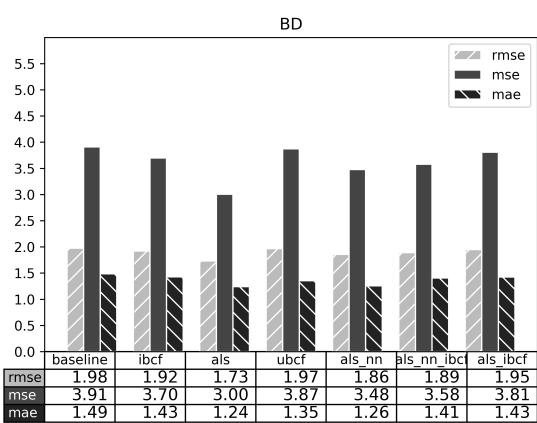
Hình 5.4: Kết quả trường hợp Locality khoa BD



Hình 5.5: Kết quả trường hợp Global khoa BD

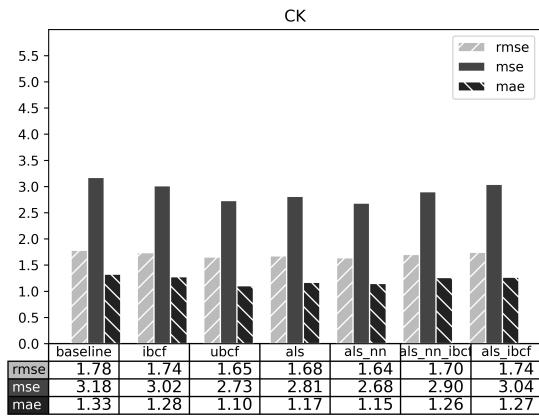


Hình 5.6: Kết quả trường hợp Locality K12 khoa MT

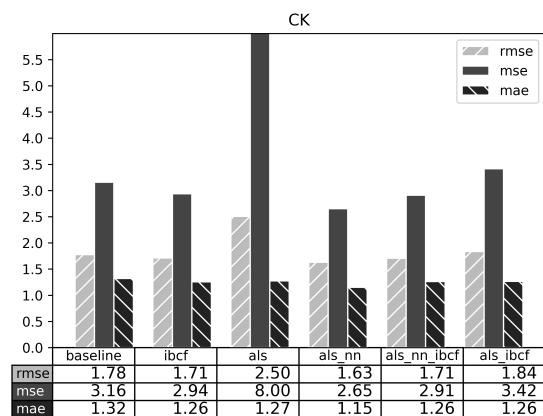


Hình 5.7: Kết quả trường hợp Locality K12 khoa BD

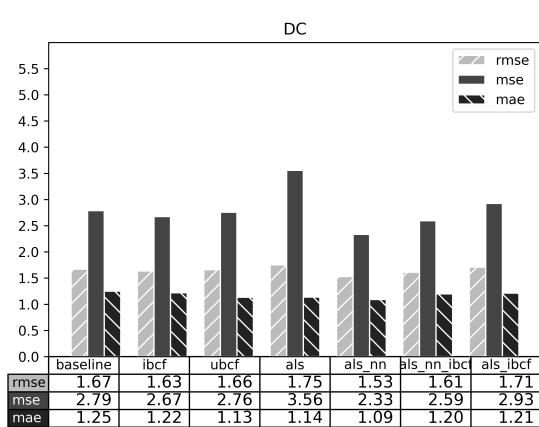
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



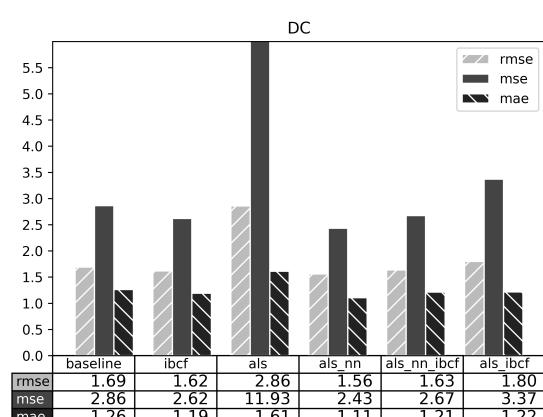
Hình 5.8: Kết quả trường hợp Locality khoa CK



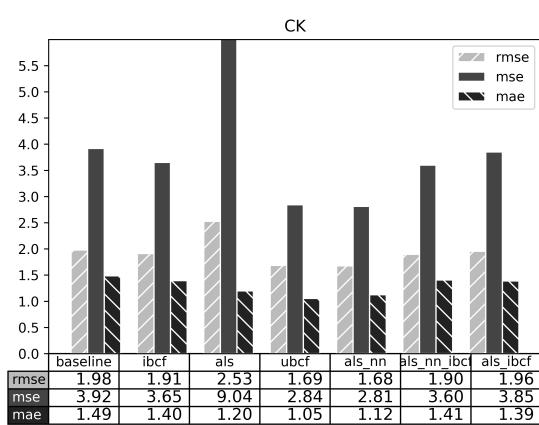
Hình 5.9: Kết quả trường hợp Global khoa CK



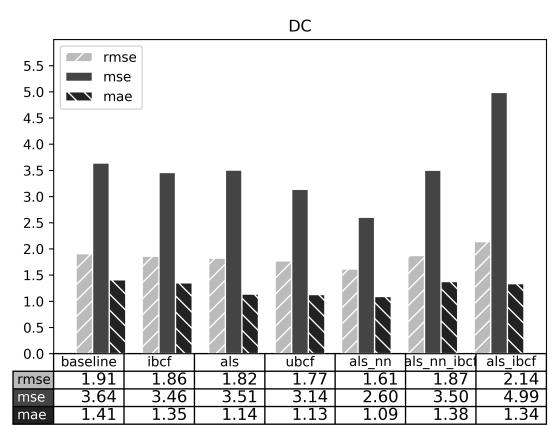
Hình 5.10: Kết quả trường hợp Locality khoa DC



Hình 5.11: Kết quả trường hợp Global khoa DC

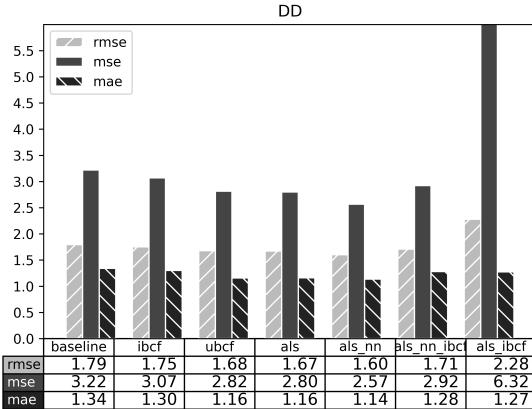


Hình 5.12: Kết quả trường hợp Locality K12 khoa CK

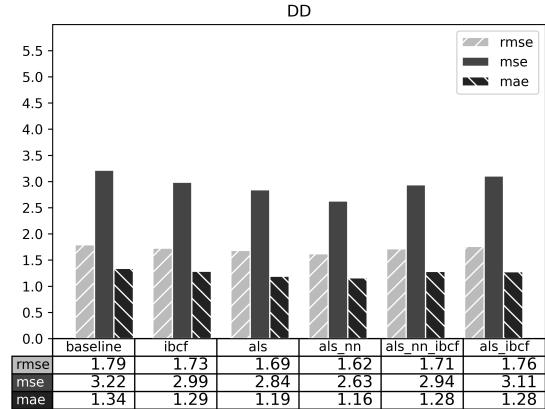


Hình 5.13: Kết quả trường hợp Locality K12 khoa DC

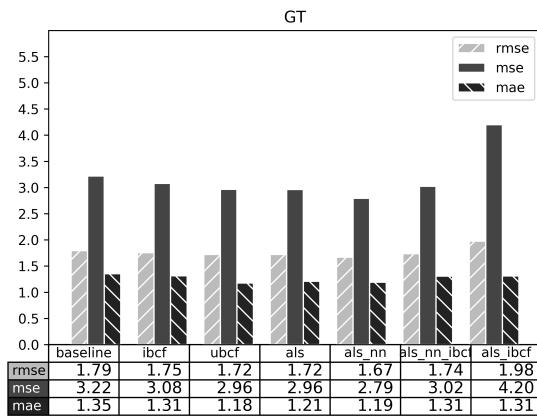
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



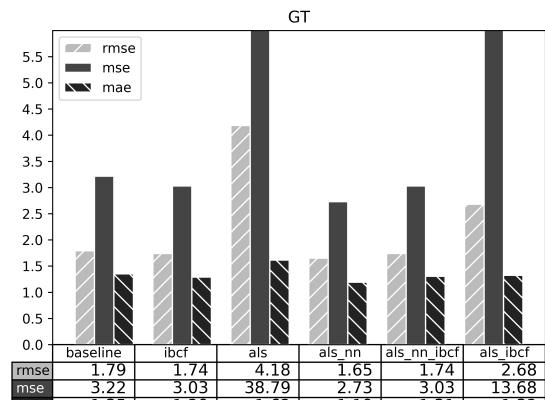
Hình 5.14: Kết quả trường hợp Locality khoa DD



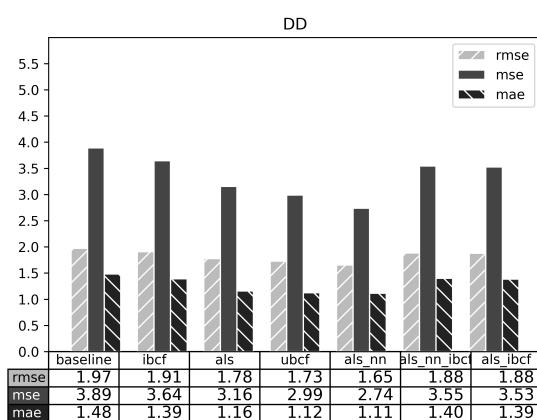
Hình 5.15: Kết quả trường hợp Global khoa DD



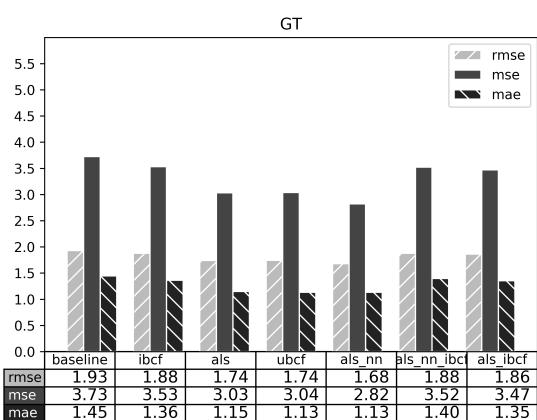
Hình 5.16: Kết quả trường hợp Locality khoa GT



Hình 5.17: Kết quả trường hợp Global khoa GT

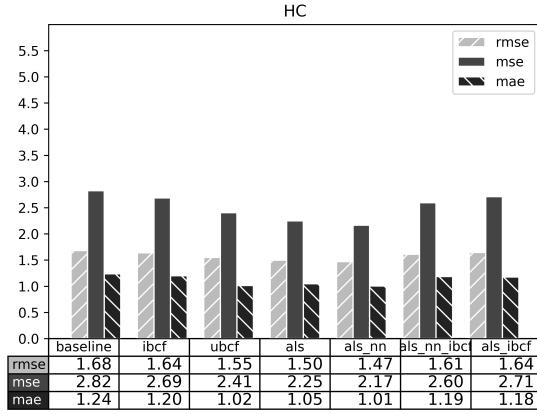


Hình 5.18: Kết quả trường hợp Locality K12 khoa DD

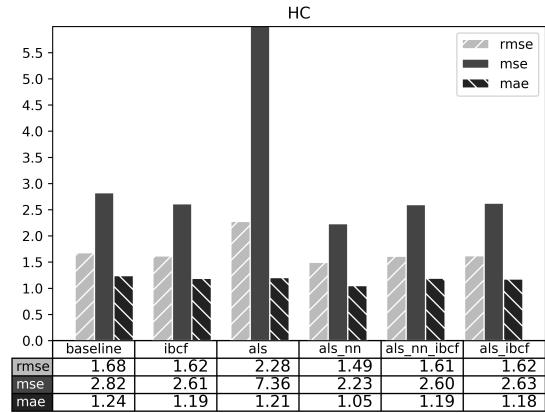


Hình 5.19: Kết quả trường hợp Locality K12 khoa GT

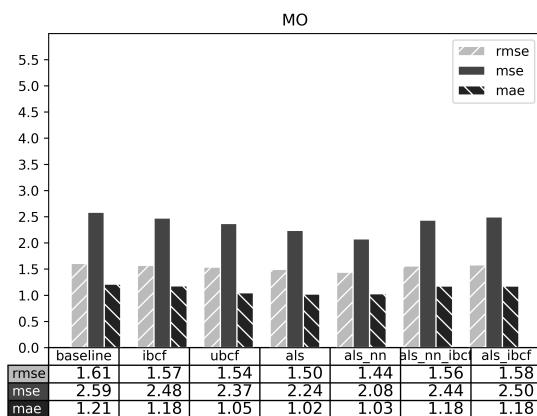
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



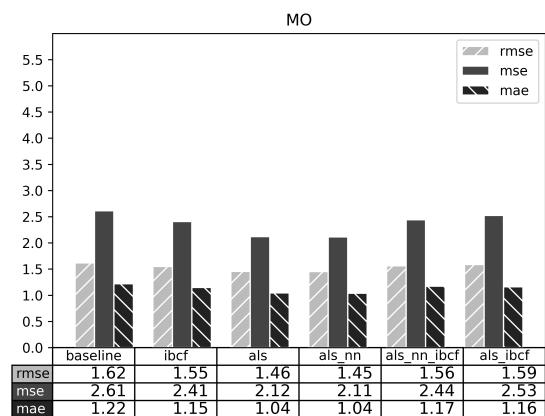
Hình 5.20: Kết quả trường hợp Locality khoa HC



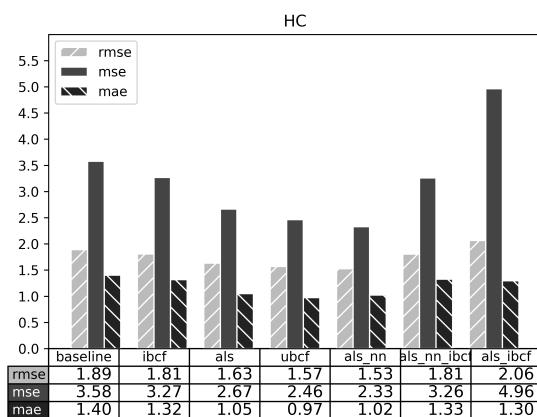
Hình 5.21: Kết quả trường hợp Global khoa HC



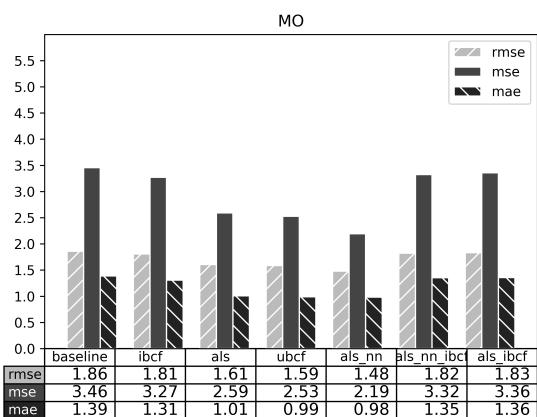
Hình 5.22: Kết quả trường hợp Locality khoa MO



Hình 5.23: Kết quả trường hợp Global khoa MO

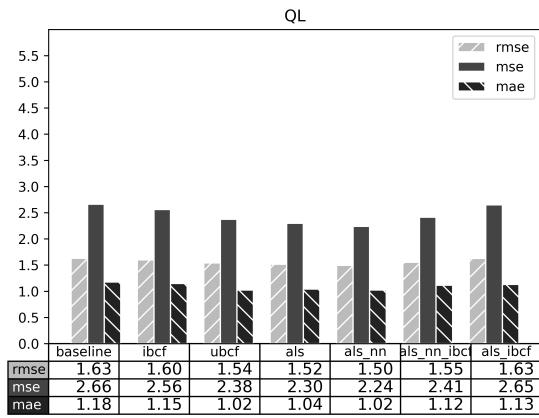


Hình 5.24: Kết quả trường hợp Locality K12 khoa HC

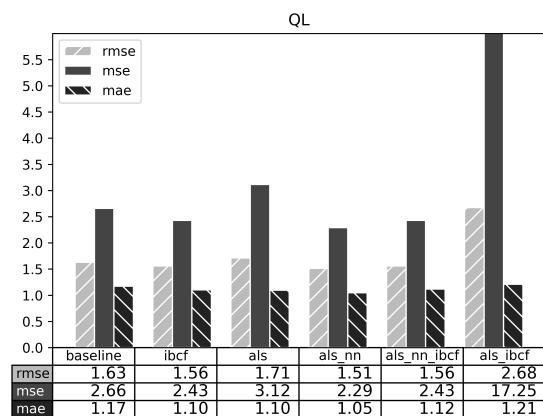


Hình 5.25: Kết quả trường hợp Locality K12 khoa MO

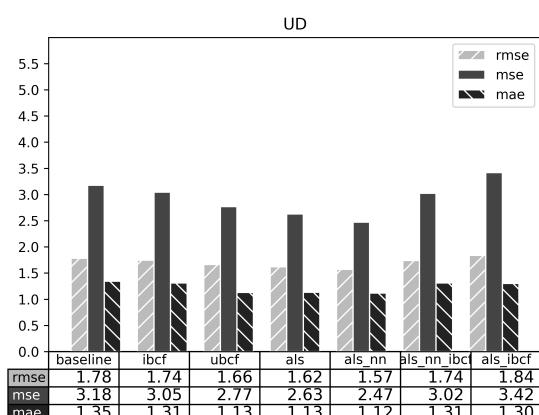
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



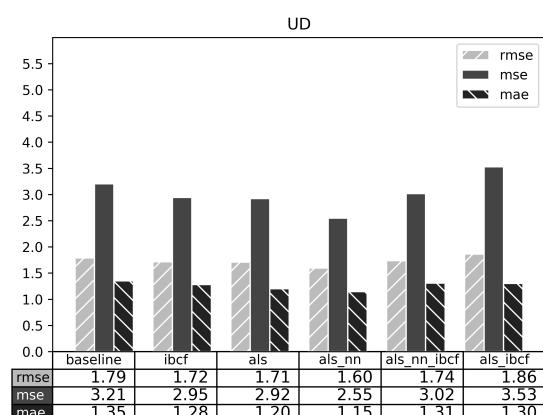
Hình 5.26: Kết quả trường hợp Locality khoa QL



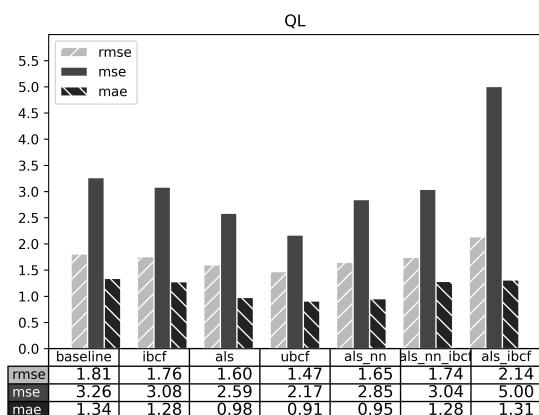
Hình 5.27: Kết quả trường hợp Global khoa QL



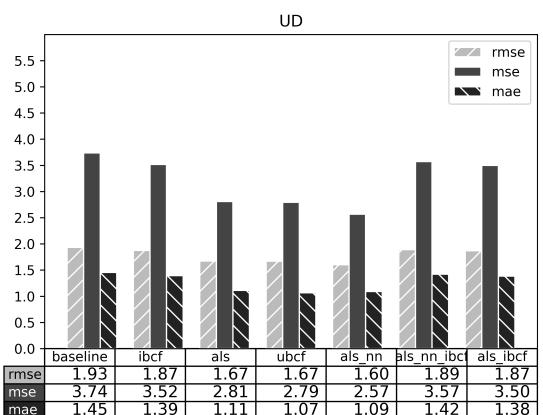
Hình 5.28: Kết quả trường hợp Locality khoa UD



Hình 5.29: Kết quả trường hợp Global khoa UD

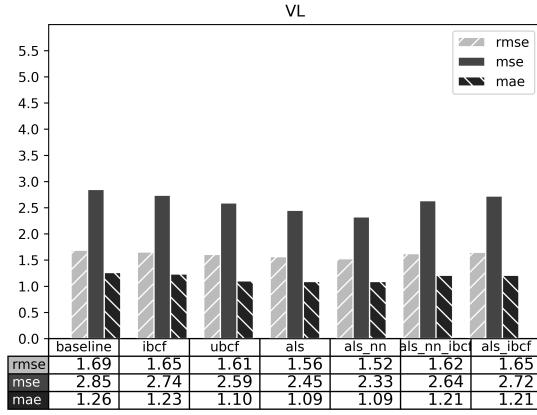


Hình 5.30: Kết quả trường hợp Locality K12 khoa QL

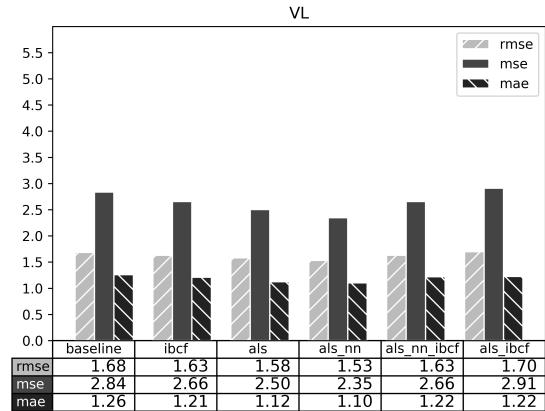


Hình 5.31: Kết quả trường hợp Locality K12 khoa UD

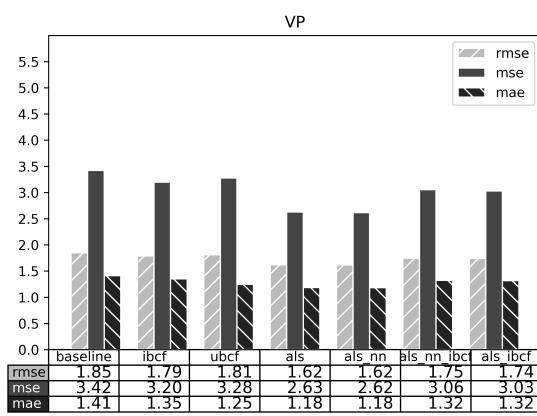
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



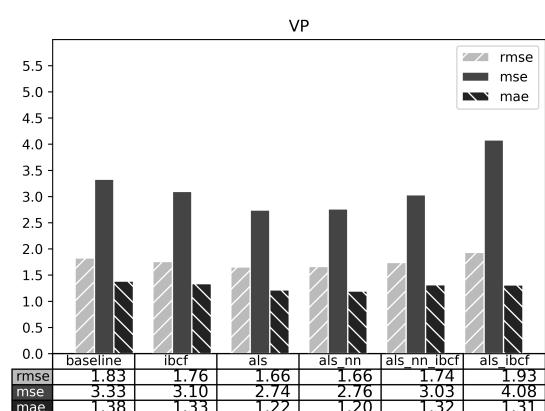
Hình 5.32: Kết quả trường hợp Locality khoa VL



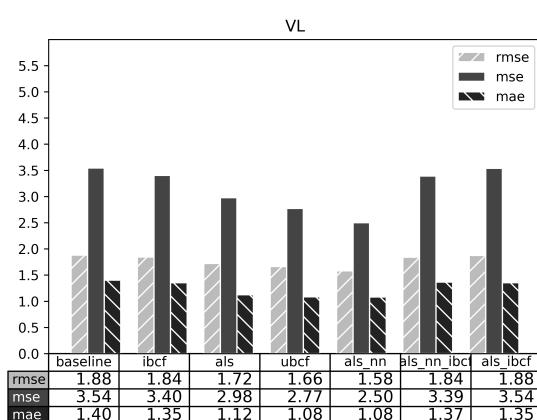
Hình 5.33: Kết quả trường hợp Global khoa VL



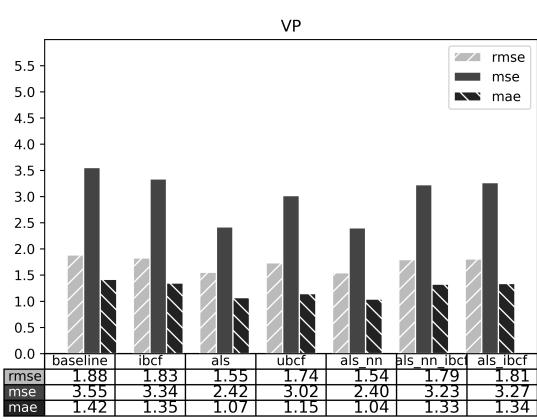
Hình 5.34: Kết quả trường hợp Locality khoa VP



Hình 5.35: Kết quả trường hợp Global khoa VP

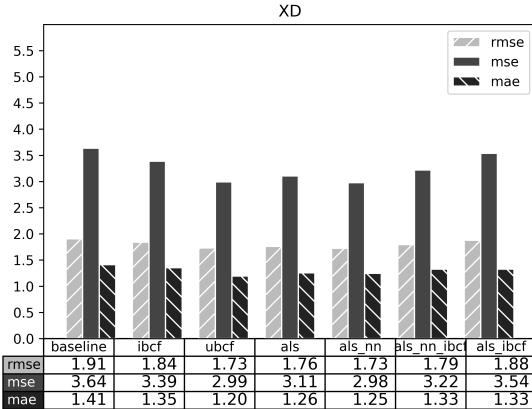


Hình 5.36: Kết quả trường hợp Locality K12 khoa VL

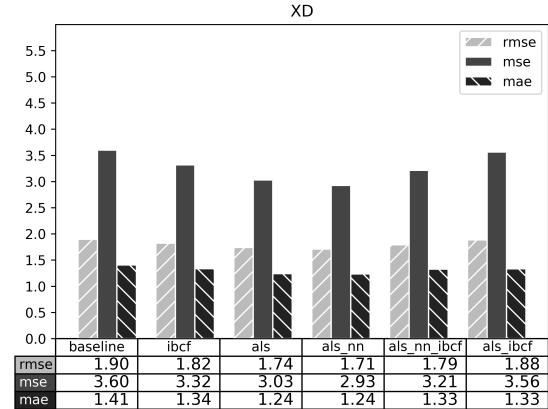


Hình 5.37: Kết quả trường hợp Locality K12 khoa VP

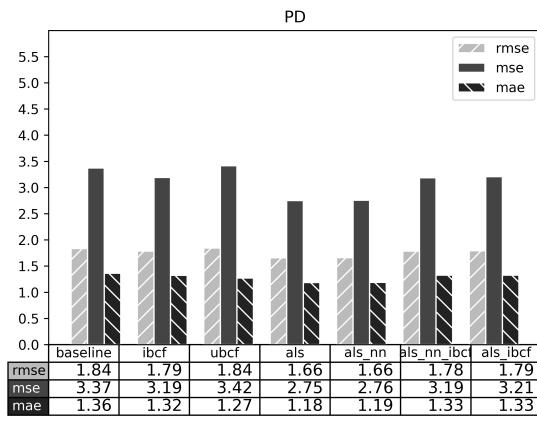
## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ



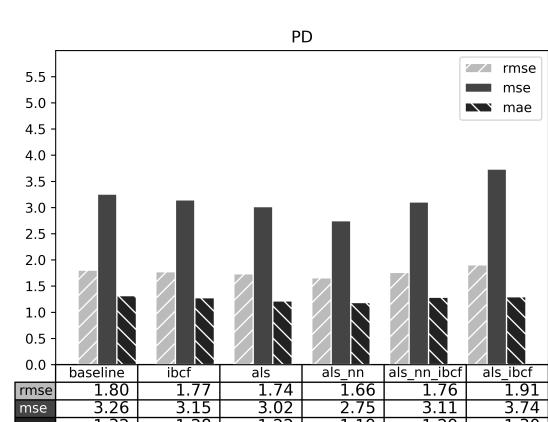
Hình 5.38: Kết quả trường hợp Locality khoa XD



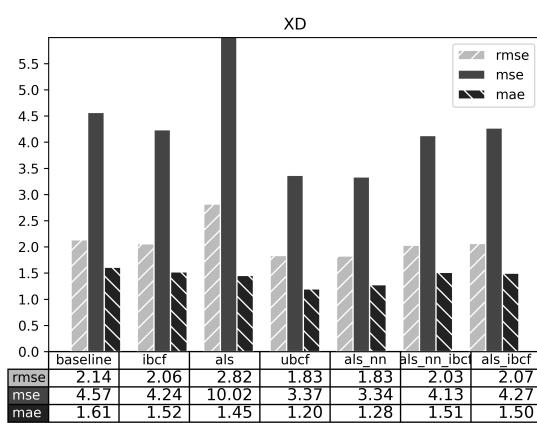
Hình 5.39: Kết quả trường hợp Global khoa XD



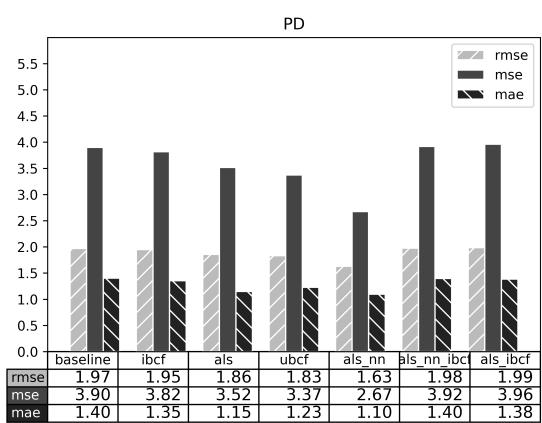
Hình 5.40: Kết quả trường hợp Locality khoa PD



Hình 5.41: Kết quả trường hợp Global khoa PD



Hình 5.42: Kết quả trường hợp Locality K12 khoa XD



Hình 5.43: Kết quả trường hợp Locality K12 khoa PD

## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ

Bảng 5.7: Tham số của giải thuật Collaborative Filtering trên Item Factor của Matrix Factorization

Khoa	(a) Trường hợp Locality		(b) Trường hợp Global	
	ALS	IBCF	ALS_NN	IBCF
MT	7	3	7	7
BD	7	5	7	9
CK	5	9	5	11
DC	5	9	7	9
DD	7	7	5	11
GT	5	7	5	9
HC	5	5	5	7
MO	7	11	5	13
PD	13	5	13	5
QL	5	5	7	5
UD	7	9	7	9
VL	7	7	5	9
VP	7	7	7	7
XD	7	7	5	9

- ALS\_NN thường tốt hơn ALS.
- ALS\_NN\_IBCF thường tốt hơn ALS\_IBCF.

Ví dụ trong khoa MT, giữa phương pháp ALS\_NN và ALS thì ALS\_NN cho ra kết quả tốt hơn ALS (với độ lỗi RMSE tương ứng là 1.69 và 1.75). Tương tự đối với phương pháp ALS\_NN\_IBCF và ALS\_IBCF thì ALS\_NN\_IBCF được lợi hơn  $\approx 4\%$  RMSE.

Sự chênh lệch về độ lỗi giữa cả hai trường hợp không quá lớn cho thấy mô hình được huấn luyện bằng dữ liệu toàn cục vẫn có khả năng dự đoán khá chính xác so với mô hình huấn luyện bằng dữ liệu cục bộ hơn.

Ở các phương pháp không có ràng buộc không âm (ALS và ALS\_IBCF) thì trong một số lần chạy ở cả hai trường hợp, độ lỗi của các phương pháp này cao đột ngột. Nhìn sâu hơn thì trong một số trường hợp này thì độ lỗi RMSE và MSE của các mô hình này cao hơn nhiều so với mô hình Baseline mặc dù độ lỗi MAE tốt hơn mô hình Baseline. Đeniable ở khoa GT, RMSE của mô hình ALS\_IBCF cao hơn nhiều so với mô hình Baseline (1.98 và 1.79 tương ứng) nhưng MAE của mô hình ALS\_IBCF (có giá trị 1.31 bằng với mô hình IBCF thông thường) lại tốt hơn mô hình Baseline (có giá trị 1.35). Một ví dụ khác ở trường hợp Global của phương pháp ALS ở khoa HC, độ lỗi MAE thường rất cố định giữa các lần chạy ( $\approx 1.21$ ) và tốt hơn so với Baseline ( $\approx 1.24$ ). Nhưng RMSE lại cao đột ngột trong hai lần chạy với giá trị lần lượt là 5.79 và 4.56. Điều này cho thấy một số sinh viên được dự đoán sai hơn rất nhiều so với các sinh viên còn lại, nguyên nhân có thể do dữ liệu được phân chia trong quá trình kiểm tra không được tốt.

Ở trường hợp Locality, phương pháp UBCF luôn cho ra độ lỗi tốt hơn phương pháp IBCF ở mọi khoa. Cụ thể hơn độ lỗi trung bình RMSE giữa mọi khoa của UBCF là 1.57 và của IBCF là 1.61. Ngoài ra mô hình ALS\_NN\_IBCF cũng có kết quả tốt hơn so với mô hình IBCF thông thường ở mọi khoa cho thấy mô hình IBCF xây dựng trên các hidden feature của phương pháp Matrix Factorization vẫn có thể đạt hiệu quả.

Kết quả của hai trường hợp trên thể hiện việc sử dụng dữ liệu Global không ảnh hưởng nhiều

*Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ*

Bảng 5.8: Kết quả trường hợp Locality

Khoa	Độ đo	Baseline	IBCF	UBCF	ALS	ALS_NN	ALS_NN_IBCF	ALS_IBCF
MT	RMSE	1.85	1.81	1.78	1.75	<b>1.69</b>	1.76	1.83
	MSE	3.40	3.29	3.15	3.05	<b>2.85</b>	3.10	3.37
	MAE	1.37	1.33	1.22	1.21	<b>1.19</b>	1.31	1.30
BD	RMSE	2.03	1.96	1.94	1.91	<b>1.81</b>	1.91	1.95
	MSE	4.12	3.84	3.78	3.63	<b>3.29</b>	3.69	3.81
	MAE	1.54	1.48	1.38	1.37	<b>1.33</b>	1.45	1.43
CK	RMSE	1.78	1.74	1.65	1.68	<b>1.64</b>	1.70	1.74
	MSE	3.18	3.02	2.73	2.81	<b>2.68</b>	2.90	3.04
	MAE	1.33	1.28	<b>1.10</b>	1.17	1.15	1.26	1.27
DC	RMSE	1.67	1.63	1.66	1.75	<b>1.53</b>	1.61	1.71
	MSE	2.79	2.67	2.76	3.56	<b>2.33</b>	2.59	2.93
	MAE	1.25	1.22	1.13	1.14	<b>1.09</b>	1.20	1.21
DD	RMSE	1.79	1.75	1.68	1.67	<b>1.60</b>	1.71	2.28
	MSE	3.22	3.07	2.82	2.80	<b>2.57</b>	2.92	6.32
	MAE	1.34	1.30	1.16	1.16	<b>1.14</b>	1.28	1.27
GT	RMSE	1.79	1.75	1.72	1.72	<b>1.67</b>	1.74	1.98
	MSE	3.22	3.08	2.96	2.96	<b>2.79</b>	3.02	4.20
	MAE	1.35	1.31	<b>1.18</b>	1.21	1.19	1.31	1.31
HC	RMSE	1.68	1.64	1.55	1.50	<b>1.47</b>	1.61	1.64
	MSE	2.82	2.69	2.41	2.25	<b>2.17</b>	2.60	2.71
	MAE	1.24	1.20	1.02	1.05	<b>1.01</b>	1.19	1.18
MO	RMSE	1.61	1.57	1.54	1.50	<b>1.44</b>	1.56	1.58
	MSE	2.59	2.48	2.37	2.24	<b>2.08</b>	2.44	2.50
	MAE	1.21	1.18	1.05	<b>1.02</b>	1.03	1.18	1.18
QL	RMSE	1.63	1.60	1.54	1.52	<b>1.50</b>	1.55	1.63
	MSE	2.66	2.56	2.38	2.30	<b>2.24</b>	2.41	2.65
	MAE	1.18	1.15	<b>1.02</b>	1.04	<b>1.02</b>	1.12	1.13
UD	RMSE	1.78	1.74	1.66	1.62	<b>1.57</b>	1.74	1.84
	MSE	3.18	3.05	2.77	2.63	<b>2.47</b>	3.02	3.42
	MAE	1.35	1.31	1.13	1.13	<b>1.12</b>	1.31	1.30
VL	RMSE	1.69	1.65	1.61	1.56	<b>1.52</b>	1.62	1.65
	MSE	2.85	2.74	2.59	2.45	<b>2.33</b>	2.64	2.72
	MAE	1.26	1.23	1.10	<b>1.09</b>	<b>1.09</b>	1.21	1.21
VP	RMSE	1.85	1.79	1.81	<b>1.62</b>	<b>1.62</b>	1.75	1.74
	MSE	3.42	3.20	3.28	2.63	<b>2.61</b>	3.06	3.03
	MAE	1.41	1.35	1.25	<b>1.18</b>	<b>1.18</b>	1.32	1.32
XD	RMSE	1.91	1.84	<b>1.73</b>	1.76	<b>1.73</b>	1.79	1.88
	MSE	3.64	3.39	2.99	3.11	<b>2.98</b>	3.22	3.54
	MAE	1.41	1.35	<b>1.20</b>	1.26	1.25	1.33	1.33
PD	RMSE	1.84	1.79	1.84	<b>1.66</b>	<b>1.66</b>	1.78	1.79
	MSE	3.37	3.20	3.42	<b>2.75</b>	2.76	3.19	3.21
	MAE	1.36	1.32	1.27	<b>1.18</b>	1.19	1.32	1.33

*Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ*

Bảng 5.9: Kết quả trường hợp Global

Khoa	Độ đo	Baseline	IBCF	UBCF	ALS	ALS_NN	ALS_NN_IBCF	ALS_IBCF
MT	RMSE	1.85	1.78	—	1.80	<b>1.72</b>	1.78	2.10
	MSE	3.44	3.18	—	3.24	<b>2.95</b>	3.16	4.64
	MAE	1.37	1.30	—	1.26	<b>1.22</b>	1.31	1.31
BD	RMSE	2.02	1.95	—	1.89	<b>1.88</b>	1.93	3.16
	MSE	4.08	3.82	—	3.57	<b>3.55</b>	3.73	18.35
	MAE	1.54	1.46	—	1.40	<b>1.40</b>	1.46	1.51
CK	RMSE	1.78	1.71	—	2.50	<b>1.63</b>	1.71	1.84
	MSE	3.16	2.94	—	8.00	<b>2.65</b>	2.91	3.42
	MAE	1.32	1.26	—	1.27	<b>1.15</b>	1.26	1.26
DC	RMSE	1.69	1.62	—	2.86	<b>1.56</b>	1.63	1.80
	MSE	2.86	2.62	—	11.93	<b>2.43</b>	2.67	3.37
	MAE	1.26	1.19	—	1.61	<b>1.11</b>	1.21	1.22
DD	RMSE	1.79	1.73	—	1.69	<b>1.62</b>	1.71	1.76
	MSE	3.22	2.99	—	2.84	<b>2.63</b>	2.94	3.11
	MAE	1.34	1.29	—	1.19	<b>1.16</b>	1.28	1.28
GT	RMSE	1.79	1.74	—	4.18	<b>1.65</b>	1.74	2.68
	MSE	3.22	3.03	—	38.79	<b>2.73</b>	3.03	13.68
	MAE	1.35	1.29	—	1.61	<b>1.19</b>	1.29	1.32
HC	RMSE	1.68	1.62	—	2.28	<b>1.49</b>	1.61	1.62
	MSE	2.82	2.61	—	7.36	<b>2.23</b>	2.60	2.63
	MAE	1.24	1.19	—	1.21	<b>1.05</b>	1.19	1.18
MO	RMSE	1.62	1.55	—	1.46	<b>1.45</b>	1.56	1.59
	MSE	2.61	2.41	—	2.12	<b>2.11</b>	2.44	2.53
	MAE	1.22	1.15	—	1.04	<b>1.04</b>	1.17	1.16
QL	RMSE	1.63	1.56	—	1.71	<b>1.51</b>	1.56	2.68
	MSE	2.66	2.43	—	3.12	<b>2.29</b>	2.43	17.25
	MAE	1.17	1.10	—	1.10	<b>1.05</b>	1.12	1.21
UD	RMSE	1.79	1.72	—	1.71	<b>1.60</b>	1.74	1.86
	MSE	3.21	2.95	—	2.92	<b>2.55</b>	3.02	3.53
	MAE	1.35	1.28	—	1.20	<b>1.15</b>	1.31	1.30
VL	RMSE	1.68	1.63	—	1.58	<b>1.53</b>	1.63	1.70
	MSE	2.84	2.66	—	2.50	<b>2.35</b>	2.66	2.91
	MAE	1.26	1.21	—	1.12	<b>1.10</b>	1.22	1.22
VP	RMSE	1.83	1.76	—	<b>1.66</b>	<b>1.66</b>	1.74	1.93
	MSE	3.33	3.10	—	<b>2.74</b>	2.76	3.03	4.08
	MAE	1.38	1.33	—	1.22	<b>1.20</b>	1.31	1.31
XD	RMSE	1.90	1.82	—	1.74	<b>1.71</b>	1.79	1.88
	MSE	3.60	3.32	—	3.03	<b>2.93</b>	3.21	3.56
	MAE	1.41	1.34	—	1.24	<b>1.24</b>	1.33	1.33
PD	RMSE	1.80	1.77	—	1.74	<b>1.66</b>	1.76	1.91
	MSE	3.26	3.15	—	3.02	<b>2.75</b>	3.11	3.74
	MAE	1.32	1.28	—	1.22	<b>1.19</b>	1.29	1.30

Bảng 5.10: Kết quả trường hợp dữ liệu từ khóa 2012 trở lên

Khoa	Độ đo	Baseline	IBCF	UBCF	ALS	ALS_NN	ALS_NN_IBCF	ALS_IBCF
MT	RMSE	2.11	2.05	1.86	2.00	<b>1.80</b>	2.01	2.31
	MSE	4.45	4.20	3.46	4.00	<b>3.24</b>	4.03	5.83
	MAE	1.60	1.50	1.17	1.24	<b>1.22</b>	1.46	1.46
BD	RMSE	1.98	1.92	1.97	1.73	<b>1.86</b>	1.89	1.95
	MSE	3.91	3.70	3.87	3.00	<b>3.48</b>	3.58	3.81
	MAE	1.49	1.43	1.35	1.24	<b>1.26</b>	1.41	1.43
CK	RMSE	1.98	1.91	1.69	2.53	<b>1.68</b>	1.90	1.96
	MSE	3.92	3.65	2.84	9.04	<b>2.81</b>	3.60	3.85
	MAE	1.49	1.40	1.05	1.20	<b>1.12</b>	1.41	1.39
DC	RMSE	1.91	1.86	1.77	1.82	<b>1.61</b>	1.87	2.14
	MSE	3.64	3.46	3.14	3.51	<b>2.60</b>	3.50	4.99
	MAE	1.41	1.35	1.13	1.14	<b>1.09</b>	1.38	1.34
DD	RMSE	1.97	1.91	1.73	1.78	<b>1.65</b>	1.88	1.88
	MSE	3.89	3.64	2.99	3.16	<b>2.74</b>	3.55	3.53
	MAE	1.48	1.39	1.12	1.16	<b>1.11</b>	1.40	1.39
GT	RMSE	1.93	1.88	1.74	1.74	<b>1.68</b>	1.88	1.86
	MSE	3.73	3.53	3.04	3.03	<b>2.82</b>	3.52	3.47
	MAE	1.45	1.36	1.13	1.15	<b>1.13</b>	1.40	1.35
HC	RMSE	1.89	1.81	1.57	1.63	<b>1.53</b>	1.81	2.06
	MSE	3.58	3.27	2.46	2.67	<b>2.33</b>	3.26	4.96
	MAE	1.40	1.32	<b>0.97</b>	1.05	1.02	1.33	1.30
MO	RMSE	1.86	1.81	1.59	1.61	<b>1.48</b>	1.82	1.83
	MSE	3.46	3.27	2.53	2.59	<b>2.19</b>	3.32	3.35
	MAE	1.39	1.31	0.99	1.01	<b>0.98</b>	1.35	1.36
QL	RMSE	1.81	1.76	<b>1.47</b>	1.60	1.64	1.74	2.13
	MSE	3.26	3.08	<b>2.17</b>	2.59	2.85	3.04	5.00
	MAE	1.34	1.28	<b>0.91</b>	0.97	0.95	1.28	1.31
UD	RMSE	1.93	1.87	1.67	1.67	<b>1.60</b>	1.89	1.87
	MSE	3.74	3.52	2.79	2.81	<b>2.57</b>	3.57	3.50
	MAE	1.45	1.39	<b>1.07</b>	1.11	1.09	1.42	1.38
VL	RMSE	1.88	1.84	1.66	1.72	<b>1.58</b>	1.84	1.88
	MSE	3.54	3.40	2.77	2.98	<b>2.50</b>	3.39	3.54
	MAE	1.40	1.35	<b>1.08</b>	1.12	<b>1.08</b>	1.37	1.35
VP	RMSE	1.88	1.83	1.74	1.55	<b>1.54</b>	1.79	1.81
	MSE	3.55	3.34	3.02	2.42	<b>2.40</b>	3.23	3.27
	MAE	1.42	1.35	1.15	1.07	<b>1.04</b>	1.33	1.34
XD	RMSE	2.14	2.06	<b>1.83</b>	2.82	<b>1.83</b>	2.03	2.07
	MSE	4.57	4.24	3.37	10.12	<b>3.34</b>	4.13	4.27
	MAE	1.61	1.52	<b>1.20</b>	1.45	1.28	1.51	1.50
PD	RMSE	1.97	1.95	1.83	1.86	<b>1.63</b>	1.98	1.99
	MSE	3.90	3.82	3.37	3.52	<b>2.67</b>	3.92	3.96
	MAE	1.40	1.35	1.23	1.15	<b>1.10</b>	1.40	1.38

## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ

Bảng 5.11: Kết quả trường hợp Locality khi loại bỏ điểm 0

Khoa	Độ đo	Baseline	IBCF	UBCF	ALS	ALS_NN	ALS_NN_IBCF	ALS_IBCF	RBM
MT	RMSE	1.58	1.55	1.51	1.58	<b>1.45</b>	1.51	1.70	1.49
	MSE	2.51	2.39	2.27	2.57	<b>2.11</b>	2.28	2.98	2.24
	MAE	1.23	1.20	1.08	1.09	<b>1.06</b>	1.18	1.18	1.13
MO	RMSE	1.44	1.40	<b>1.23</b>	1.33	1.28	1.40	1.41	1.30
	MSE	2.06	1.97	<b>1.51</b>	1.76	1.64	1.96	1.98	1.70
	MAE	1.14	1.11	<b>0.89</b>	0.95	0.94	1.11	1.11	0.99

đến kết quả dự đoán của một khoa cụ thể. Vì thế nên việc sử dụng dữ liệu lớn để dự đoán cho một khoa là không cần thiết.

### 5.2.2 Kết quả thực nghiệm Locality với dữ liệu khóa 2012 trở lên

Vì chương trình học và nội dung của các môn học có thể có sự thay đổi qua các năm - vì thế dữ liệu của các năm cách xa năm hiện tại có thể phản ánh khác so với dữ liệu sinh viên của năm gần đây. Thực nghiệm này được thực hiện để xác định dữ liệu sinh viên ở các năm trước có nên dùng để dự đoán điểm của những năm gần đây hay không. Kết quả chi tiết của thực nghiệm này được biểu diễn ở **Bảng 5.10** và các biểu đồ so sánh các độ lỗi giữa các phương pháp trong từng khoa.

Trong trường hợp này, một số khoa như khoa MT có độ lỗi tăng cao ở mọi phương pháp. Tuy nhiên ở khoa QL, phương pháp ALS\_NN từ phương pháp có độ chính xác tốt nhất ở khoa này lại có độ lỗi RMSE tăng từ 1.50 lên 1.64 - trong khi đó độ lỗi của phương pháp UBCF giảm từ 1.54 xuống 1.47 trở thành phương pháp tốt nhất ở khoa QL. Phương pháp ALS\_NN ở thực nghiệm này vẫn là phương pháp tốt nhất với 9 khoa trong 14 khoa đạt độ lỗi thấp nhất.

Kết quả trên thể hiện dữ liệu của các năm cách xa năm hiện tại vẫn có ảnh hưởng đến phân bố điểm ở các khoa.

### 5.2.3 Kết quả thực nghiệm Locality loại bỏ điểm 0

Trong biểu đồ phân bố điểm thì số lượng điểm 0 chiếm một tỉ lệ lớn và có thể gây ảnh hưởng nhiều đến kết quả dự đoán vì rất nhiều trường hợp có thể bị 0 điểm - vắng thi, cầm thi, .... Thực nghiệm này được thực hiện để xem xét ảnh hưởng của số lượng điểm 0 trong các khoa vào kết quả dự đoán trong hai khoa MT (khoa có độ lỗi nền tương đối cao ở trường hợp Locality) và khoa MO (khoa có độ lỗi nền tương đối thấp ở trường hợp Locality).

**Bảng 5.11** thể hiện kết quả của trường hợp Locality khi loại bỏ điểm 0. Độ lỗi của các phương pháp được cải thiện hơn. Diễn hình ở khoa MT RMSE giảm từ 1.69 khi chưa loại bỏ điểm 0 xuống 1.45. Khoa MO dù độ lỗi tương đối thấp ở trường hợp Locality thì độ lỗi cũng giảm mạnh, cụ thể ở mô hình UBCF RMSE giảm từ 1.50 xuống 1.23. Độ lỗi nền của cả hai khoa cũng giảm hẳn tốt hơn so với khi chưa loại bỏ điểm 0. Quan sát kĩ hơn biểu đồ phân bố điểm ta có thể thấy được số lượng điểm 0 ảnh hưởng nhiều đến độ lỗi của các phương pháp trong trường hợp Locality - số lượng điểm 0 càng nhiều thì độ lỗi càng cao.

Vì thế điểm 0 ảnh hưởng tiêu cực đến kết quả dự đoán. Các phương pháp trên khi được sử dụng để dự đoán các điểm có điểm 0 thì không đạt được kết quả cao, việc dự đoán điểm 0 có thể cần sử dụng một phương pháp khác

### 5.2.4 Bổ sung mô hình RBM và thực nghiệm

Bảng 5.12: Kết quả RBM trường hợp Locality

	<b>MT</b>	<b>MO</b>
<b>rmse</b>	2.23	1.76
<b>mse</b>	4.95	3.11
<b>mae</b>	1.61	1.24

Mô hình RBM đã được sử dụng vào các bộ dữ liệu đại học trong các bài báo khác và cho kết quả khá tốt. Vì thế chúng tôi xây dựng mô hình RBM và áp dụng thử vào hai khoa MT và MO cho trường hợp Locality và trường hợp Locality loại bỏ điểm 0. **Bảng 5.12** và **Bảng 5.11** lần lượt thể hiện kết quả cho trường hợp Locality và trường hợp Locality khi loại bỏ điểm 0 của mô hình RBM

Ở trường hợp Locality, mô hình RBM cho kết quả không tốt, cao hơn nhiều so với phương pháp nền với kết quả RMSE ở khoa MT và MO tương ứng là 2.23 và 1.76 (so với 1.85 và 1.62). Xem kết quả ở mức độ sâu hơn thì RBM thường dự đoán các điểm từ 5 trở lên, rất ít khi dự đoán các điểm ở mức thấp hơn và vì số lượng điểm 0 chiếm một tỉ lệ tương đối lớn nên có thể đóng vai trò quan trọng trong kết quả dự đoán không tốt của RBM.

Ở trường hợp Locality khi loại bỏ điểm 0, RMSE của RBM cải thiện rõ rệt (giảm từ 2.23 xuống 1.49 ở khoa MT) và tốt hơn hẳn so với phương pháp baseline (1.58). Vì thế việc áp dụng phương pháp RBM vào dự đoán điểm là khả quan và có thể được nghiên cứu nhiều hơn.

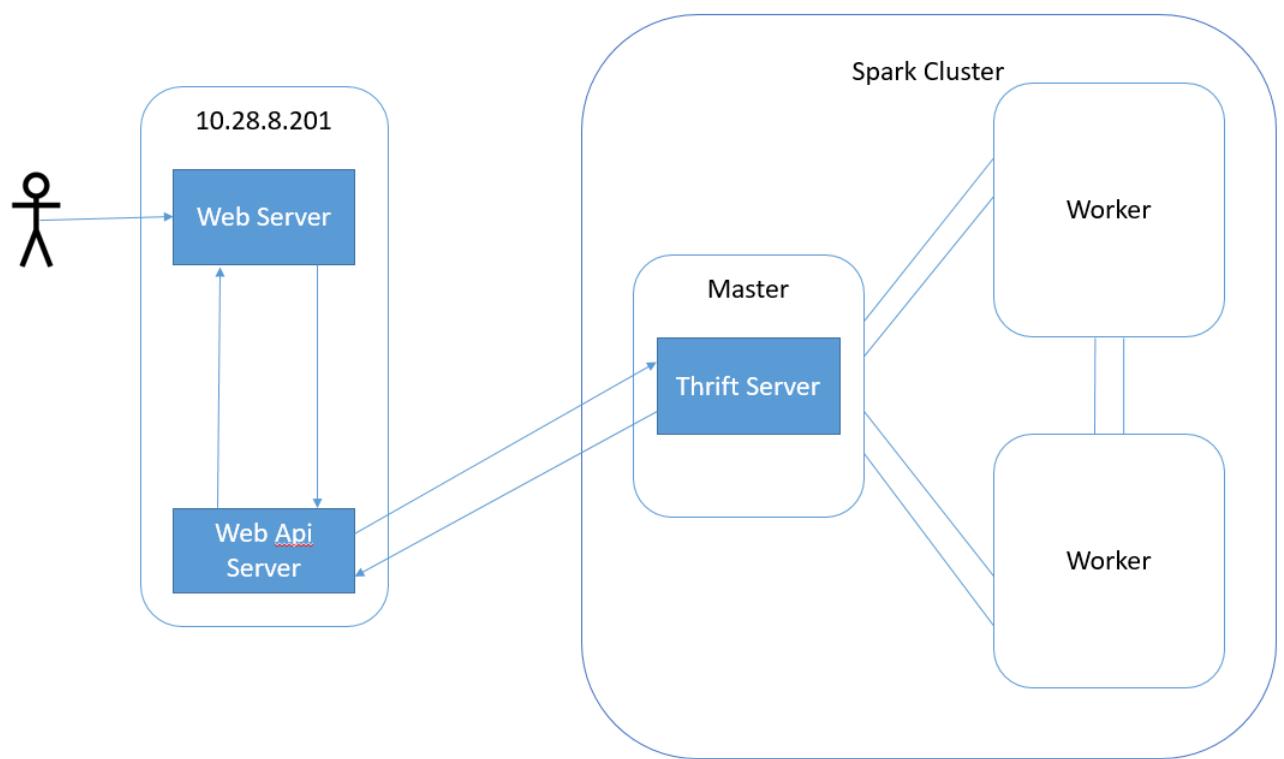
### 5.2.5 Hệ thống Web

Web Server sẽ giao tiếp với hệ thống dự đoán thông qua Web API Server. Cấu trúc json của dữ liệu được truyền lên được mô tả chi tiết ở **Mục 4.3.1**.

**Hình 5.44** biểu diễn flow của hệ thống khi post request được thực hiện. Cụ thể Flow của một lệnh Post như sau:

1. Web server gửi Post request lên endpoint tương ứng trên Web API Server.
2. Web API Server nhận request và gửi cho Thrift server đặt tại master node của Spark cluster.
3. Thrift server đóng gói job và submit lên Spark cluster.
4. Spark cluster xử lý job và ghi kết quả ra file.
5. Thrift server nhận event hoàn thành của Spark cluster và đọc file kết quả rồi gửi kết quả về cho Web API Server.
6. Web API Server nhận kết quả và gửi trả về cho Web Server.

Giao diện nhập dữ liệu đầu vào (điểm những môn mà sinh viên đã học) và kết quả dự đoán được biểu diễn ở **Hình 5.45** và **Hình 5.46**. Ở giao diện nhập dữ liệu, người dùng có thể chọn giải thuật dự đoán và điền điểm đã học, kết quả dự đoán bao gồm tất cả những môn mà sinh viên chưa học.



Hình 5.44: Flow hệ thống Web

## Chương 5 THỰC NGHIỆM VÀ KẾT QUẢ

The screenshot shows a web-based rating estimation form. At the top, there's a navigation bar with links for Home, Teams, Publications, Projects, Documentations, About us, Application, and Sign in. Below the navigation is a section titled "Estimate rating". It contains fields for "STUDENT ID" (1513293), "FACULTY" (Khoa học và kỹ thuật máy tính), and "ALGORITHM" (als\_nn). There are two rows for "SUBJECT CODE - RATING": one row for CO1011 with a rating of 9.5 and another for CO1007 with a rating of 8.0. Each row has a red "x" button to remove it. A green "Submit" button is at the bottom.

Hình 5.45: Giao diện nhập dữ liệu đầu vào

Subject Code	Name	Rating
CO2001	Kỹ năng chung nghiệp cho kỹ sư	9.75
CO2003	Cấu trúc dữ liệu & giải thuật	8.75
CO2005	Lập trình hướng đối tượng	9
CO2007	Kiến trúc máy tính	8.5
CO2009	Thiết kế logic với verilog	9.5
CO2011	Mô hình hóa toán học	8.75
CO2013	Hệ cơ sở dữ liệu	8.75
CO2015	Linh kiện & mạch điện tử	8.75
CO2017	Hệ điều hành	8.75
CO2019	Thực tập phần cứng máy tính	9.5

Hình 5.46: Giao diện kết quả dự đoán

# Chương 6

# TỔNG KẾT

Ở chương này, chúng tôi sẽ tổng kết lại đề tài, các giải thuật sử dụng cũng như kết quả của đề tài. Bên cạnh đó, ở phần cuối chương là hướng để phát triển đề tài trong tương lai dựa trên những kết quả hiện có.

## 6.1 Tổng kết

Khai phá, phân tích dữ liệu trong lĩnh vực giáo dục để đưa ra các công cụ có ích phục vụ cho công tác đào tạo là nhu cầu thiết thực và rất cần thiết ngày nay, nhất là trong thời đại công nghệ thông tin phát triển, các kỹ thuật học máy giúp đem lại nhiều thành tựu khoa học nổi bật trong nhiều lĩnh vực. Ở luận văn này, chúng tôi đã nghiên cứu, phân tích và phát triển công cụ phân tích dữ liệu đại học giúp cho việc dự đoán điểm số và đề xuất môn học cho sinh viên đại học dựa trên bộ dữ liệu đại học của trường Đại học Bách Khoa - Đại học Quốc gia Tp.HCM.

Trên cơ sở có bộ dữ liệu thực về quá trình học tập của sinh viên, chúng tôi đã đưa ra các đề xuất để xây dựng mô hình dự đoán điểm áp dụng các kỹ thuật học máy như Collaborative Filtering, Matrix Factorization, Item-based Collaborative Filtering on Item Factor Matrix of Alternative Least Square, Restricted Boltzmann Machine, ... và trình bày chi tiết việc áp dụng các giải thuật này vào xây dựng công cụ. Với việc áp dụng nhiều kỹ thuật học máy khác nhau để xây dựng mô hình dự đoán điểm cho bộ công cụ phân tích dữ liệu đại học, chúng tôi cũng đã rút ra được kết quả so sánh giữa các kỹ thuật với nhau. Kết quả, khi sử Non-negative Alternative Least Square (Matrix Factorization), kết quả thường tốt hơn các kỹ thuật khác trong đa số các trường hợp.

Ngoài việc so sánh các kỹ thuật khác nhau trong việc xây dựng mô hình dự đoán điểm, chúng tôi cũng tiến hành các thực nghiệm với tập dữ liệu khác nhau để tìm ra các giải pháp phù hợp nhất và tính chất của tập dữ liệu, đó là so sánh thực nghiệm:

1. Trường hợp Locality: Chỉ sử dụng dữ liệu của một khoa để huấn luyện và đánh giá kết quả dự đoán cho khoa đó (Trường hợp Locality).
2. Trường hợp Global: Sử dụng dữ liệu của tất cả các khoa để huấn luyện cho mô hình. Khi kiểm tra chỉ kiểm tra các sinh viên trong một khoa (Trường hợp Global).
3. Trường hợp Locality với dữ liệu K12: Chỉ sử dụng dữ liệu của sinh viên khóa 2012 trở lên. (Train và đánh giá theo trường hợp Locality)
4. Trường hợp Locality với dữ liệu loại bỏ điểm 0: Được thực hiện cho hai khoa MT và MO với dữ liệu huấn luyện và kiểm tra không chứa điểm 0.

Kết quả cho thấy:

- Giữa trường hợp Locality và Global thì kết quả đánh giá không có nhiều chênh lệch.
- Sự phân bố điểm của các khoa có ảnh hưởng nhiều đến kết quả dự đoán - điển hình là số lượng điểm 0. Vì thế việc dự đoán chỉ có thể dự đoán tốt cho các điểm từ 5 trở lên (điểm qua môn), còn việc dự đoán điểm 0 (hoặc các điểm rớt môn) cần một hướng tiếp cận khác.
- Trường hợp Locality với dữ liệu loại bỏ điểm 0: Điểm 0 có ảnh hưởng tiêu cực đối với mọi mô hình dự đoán. Các phương pháp trên không phù hợp cho việc dự đoán điểm 0.
- Trường hợp Locality với dữ liệu từ khóa 2012 trở lên cho thấy các dữ liệu ở các năm học cách xa những năm gần đây vẫn có ảnh hưởng đến kết quả dự đoán.

Chúng tôi cũng đã hiện thực bộ công cụ phân tích dữ liệu đại học với hai module chính là module dự đoán điểm số và module đề xuất môn học. Module đề xuất môn học được xây dựng bằng cách khai phá dữ liệu để tìm ra các luật kết hợp sử dụng giải thuật FP-Growth. Các đề xuất môn học cho sinh viên sẽ dựa trên kết quả của module dự đoán và module đề xuất.

Với công cụ phân tích dữ liệu đại học đã được xây dựng, chúng tôi cũng đã trực quan hóa lên web để sinh viên có thể sử dụng trong thực tế.

## **6.2 Hướng phát triển trong tương lai**

Với bộ công cụ phân tích dữ liệu đại học hiện tại, có thể tiếp tục hoàn thiện và phát triển trong tương lai như:

- Sử dụng kết quả dự đoán để phát triển thêm module phục vụ cho phòng đào tạo của trường, sau khi đăng ký môn học có thể nắm được kết quả dự đoán số sinh viên sẽ có kết quả không tốt trong môn học để có biện pháp hỗ trợ kịp thời.
- Thêm các module giúp cho việc trực quan hóa để người dùng dễ rút trích thông tin từ tập dữ liệu đại học.
- Kết hợp việc dự đoán với các dữ liệu cá nhân của sinh viên, kết quả thi đại học, dữ liệu học tập trên e-learning,...
- Dựa vào kết quả học tập của sinh viên, xây dựng module phân tích mức độ tương quan giữa trường trung học phổ thông, vùng miền với kết quả học tập sau khi vào trường để phục vụ cho công tác tuyển sinh....

Đề tài "*Nghiên cứu và phát triển công cụ phân tích dữ liệu đại học và trực quan hóa*" còn nhiều hướng thú vị để tiếp tục phát triển trong tương lai. Với kết quả hiện tại, chúng tôi hoàn toàn tin rằng công cụ phân tích dữ liệu đại học có thể được ứng dụng trên thực tế để phục vụ cho công tác giảng dạy tốt hơn và giúp sinh viên nâng cao chất lượng học tập.

# **PHỤ LỤC**

Kết quả hai thực nghiệm Locality và Global ở 4 khoa MT, MO, HC, XD của luận văn này đã được đưa vào paper (đã được chấp nhận) nộp tại hội nghị 21st IEEE International Conference on High Performance Computing and Communications (HPCC-2019) (accepted).

T.L.Mai, P.T.Do, M.T.Chung and N.Thoai, "An Apache Spark-based Platform for Predicting The Performance of Undergraduate Student", 21st IEEE International Conference on High Performance Computing and Communications, 2019 (accepted).

# Tài liệu tham khảo

- [1] C. Pinela. (2017, Nov.) Recommender system - user-based and item-based collaborative filtering. [Online]. Available: <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>
- [2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2012.
- [3] C. Romero and S. Ventura, “Educational data mining: A survey from 1995 to 2005,” *Expert Systems with Applications*, vol. 33, pp. 135–146, Jul. 2007.
- [4] ———, “Educational data mining: A review of the state of the art,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, pp. 601–618, Dec. 2010.
- [5] R. S. Baker and K. Yacef, “The state of educational data mining in 2009: A review and future visions,” *Journal of Educational Data Mining*, vol. 1, pp. 601–618, Dec. 2009.
- [6] P. Resnick and H. R. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, pp. 56–59, 1997.
- [7] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, “Data mining algorithms to classify students,” *1st International Conference on Educational Data Mining*, p. 8–17, Jun. 2008.
- [8] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, “Predicting student performance from lms data: A comparison of 17 blended courses using moodle lms,” *IEEE Transactions on Learning Technologies*, 2017.
- [9] J. Zimmermann, K. H. Brodersen, J.-P. Pellet, E. August, and J. Buhmann, “Predicting graduate-level performance from undergraduate achievements,” in *EDM*, Jul. 2011, pp. 357–358.
- [10] E. García, C. Romero, S. Ventura, and C. D. Castro, “An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering,” *User Modeling and User-Adapted Interaction*, vol. 19, no. 1-2, pp. 99–132, Feb. 2009.
- [11] D. Nurjanah, “Good and similar learners’ recommendation in adaptive learning systems,” *Conference on Computer Supported Education*, vol. 1, pp. 434–440, 2016.
- [12] R. Turnip, D. Nurjanah, and D. Kusumo, “Hybrid recommender system for learning material using content-based filtering and collaborative filtering with good learners’ rating,” in *2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, Nov. 2017, pp. 61–66.

## Chương 6 TÀI LIỆU THAM KHẢO

- [13] N. Thai-nghe, L. Drumond, A. Krohn-Grimberghe, and L. Schmidt-Thieme, “Recommender system for predicting student performance,” *Procedia Computer Science*, vol. 1, p. 2811–2819, 2010.
- [14] N. Thai-nghe, L. Drumond, R. Nanopoulos, and L. Schmidt-thieme, “Recommender system for predicting student performance,” in *In Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU)*, 2011.
- [15] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran, “Machine learning based student grade prediction: A case study,” *CoRR*, Aug. 2017.
- [16] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” in *Linear Algebra*. Springer, 1971, pp. 134–151.
- [17] D. Lee and H. Seung, “Algorithms for non-negative matrix factorization,” in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, vol. 13, Jan. 2000, pp. 535–541.
- [18] G. E Hinton, “Article training products of experts by minimizing contrastive divergence,” *Neural computation*, vol. 14, pp. 1771–800, 09 2002.
- [19] R. Salakhutdinov, A. Mnih, and G. E. Hinton, “Restricted boltzmann machines for collaborative filtering,” in *ICML*, vol. 227, 01 2007, pp. 791–798.