ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Hoàng Minh Nhật

HỆ THỐNG GỢI Ý KẾT BẠN DỰA TRÊN TÍNH CÁCH, XỬ LÝ TRỰC TIẾP TRÊN THIẾT BỊ ĐỂ BẢO VỆ QUYỀN RIÊNG TƯ

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ thông tin

TP. HCM - 2025

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Hoàng Minh Nhật

HỆ THỐNG GỢI Ý KẾT BẠN DỰA TRÊN TÍNH CÁCH, XỬ LÝ TRỰC TIẾP TRÊN THIẾT BỊ ĐỂ BẢO VỆ QUYỀN RIÊNG TƯ

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: Th.S Ngô Khánh Khoa

TP. HCM - 2025

LỜI CAM ĐOAN

Khóa luận tốt nghiệp này được thực hiện trực tiếp bởi em và dưới sự hướng dẫn của thầy Thạc Sĩ Ngô Khánh Khoa. Em xin cam đoan rằng mọi quá trình nghiên cứu, phát triển, triển khai và báo cáo được trình bày trong báo cáo này đề được chính em độc lập thực hiện mà không sao chép, đạo văn từ những nguồn khác mà không có sự cho phép. Nếu có vi phạm quy định về nội dung trí tuệ, em xin chịu trách nhiệm tất cả những truy cứu theo quy định của Trường Đại Học Công Nghệ Thông Tin - ĐHQGHCM.

TpHCM, ngày 26 tháng 12 năm 2024

Sinh viên

Hoàng Minh Nhật

LỜI CẢM ƠN

Lời đầu tiên cho phép em bày tỏ lòng biết ơn sâu sắc đến Khoa Công nghệ Thông tin - Trường Đại học Công nghệ, ĐHQGHN. Đây là nơi em đã có cơ hội tiếp cận với những tri thức mới mẻ, được học hỏi từ các thầy cô xuất sắc và kết nối với những người bạn, anh chị em đầy năng động và tài năng.

Em cũng xin gửi lời cảm ơn chân thành đến cô Hoàng Thị Điệp, người đã luôn là nguồn cảm hứng và sự hướng dẫn quý báu trong suốt thời gian em học tập tại trường. Sự tận tâm và hỗ trợ nhiệt tình của cô đã tiếp thêm động lực để em vượt qua những thử thách trong hành trình nghiên cứu và hoàn thiện khóa luận tốt nghiệp.

Ngoài ra, em xin gửi lời cảm ơn đến gia đình, bạn bè và những người đã luôn giúp đỡ, động viên, đồng hành cùng em suốt chặng đường học tập ở trường và khoảng thời gian thực hiện khóa luận.

Kính chúc tất cả mọi người luôn vui vẻ, hạnh phúc và gặt hái được nhiều thành công trong cuộc sống.

TÓM TẮT

Tóm tắt: Bài toán xây dựng cây bootstrap tiến hóa (Phylogenetic bootstrapping) là một phần quan trọng trong sinh học tiến hóa, nhằm tái tạo cây tiến hóa với số lượng thay đổi tối thiểu dựa trên tiêu chí tính tiết kiệm tối đa (maximum parsimony - MP) đồng thời tính độ tin cậy của các phân hoạch nhị phân trong cây này. MPBoot2 cải tiến so với phiên bản trước, MPBoot, bằng cách tích hợp kỹ thuật biến đổi cây Tree Bisection and Reconnection (TBR), cho phép khám phá không gian cây một cách toàn diện hơn so với Subtree Pruning and Regrafting (SPR). Hai phép biến đổi này bổ trợ lẫn nhau để tăng cường khả năng khám phá không gian cây. Để nâng cao hiệu suất hơn nữa, chúng tôi giới thiệu MPBoot-RL, áp dụng giải thuật tối ưu đàn kiến (ant colony optimization) nhằm kết hợp động giữa SPR và TBR, cải thiện cả độ chính xác và thời gian chạy. MPBoot2 còn bao gồm các tính năng tiên tiến như hệ thống checkpoint, hỗ trợ nhiều loại dữ liệu khác nhau, và quản lý bộ nhớ tối ưu, giúp nó phù hợp với các tập dữ liệu lớn và phức tạp. Kết quả thực nghiệm cho thấy hiệu suất vượt trội về cả độ chính xác lẫn tốc độ thực thi, khẳng định MPBoot2 và MPBoot-RL là những công cụ đa năng và mạnh mẽ cho phân tích phát sinh chủng loại dựa trên tiêu chí MP.

Từ khóa: *MPBoot*, *MPBoot2*, *MPBoot-RL*, *Phát sinh chủng loại học*

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

DANH MỤC GIẢI THUẬT

Chương 1

Giới thiệu

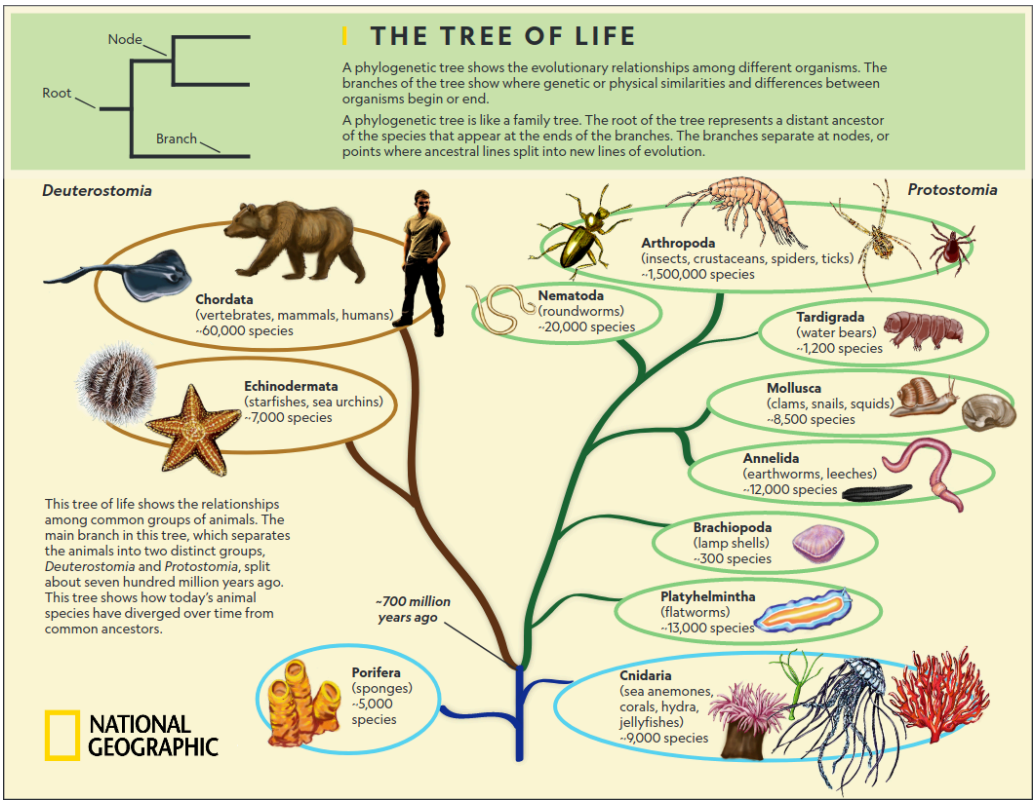
1.1. Bối cảnh và vấn đề

1.1.1. Mạng xã hội và nhu cầu kết nối theo tính cách

Twins là một ứng dụng mạng xã hội theo hướng bán khép kín, tập trung vào các cộng đồng nhỏ và chất lượng. Ứng dụng hướng tới việc tìm bạn có tính cách và sở thích tương đồng, lấy cảm hứng từ cơ chế lướt của Tinder và yếu tố kết nối thân mật của Locket. Khác với những nền tảng đại trà, Twins ưu tiên kết nối có chiều sâu thay vì số lượng tương tác. Mục tiêu này dẫn tới việc giảm bớt các tín hiệu bề mặt và tăng trọng số cho các yếu tố phản ánh đặc trưng cá nhân ổn định hơn. Về mặt trải nghiệm, người dùng được dẫn qua một chuỗi câu hỏi ngắn gọn để trích xuất tính cách, sau đó dùng kết quả này như một "dấu vân tính cách" phục vụ gợi ý và phân nhóm.

Các nền tảng mạng xã hội và ứng dụng kết nối hiện nay thường tối ưu cho tốc độ ghép cặp và số lượt tương tác, dựa trên yếu tố vị trí, sở thích bề mặt hoặc mạng bạn bè sẵn có. Cách tiếp cận này tạo ra nhiều kết quả, nhưng chưa chắc dẫn tới sự tương hợp lâu dài. Trong khi đó, các mô hình tính cách như Big Five được xem là khung tham chiếu ổn định, có khả năng giải thích xu hướng hành vi và mức độ phù hợp giữa các cá nhân [1,2].

Ở góc nhìn của đề tài, nhu cầu kết nối theo tính cách có ý nghĩa vì nó gắn với các đặc trưng ít thay đổi theo thời gian, nên phù hợp cho bài toán gợi ý dài hạn. Lựa chọn này cũng tránh việc phụ thuộc quá nhiều vào dữ liệu tương tác ngắn hạn, vốn dễ bị ảnh hưởng bởi bối cảnh, tâm trạng hoặc hiệu ứng thuật toán. Hình Hình 1.1 minh họa bối cảnh ứng dụng và mục tiêu kết nối theo tính cách.



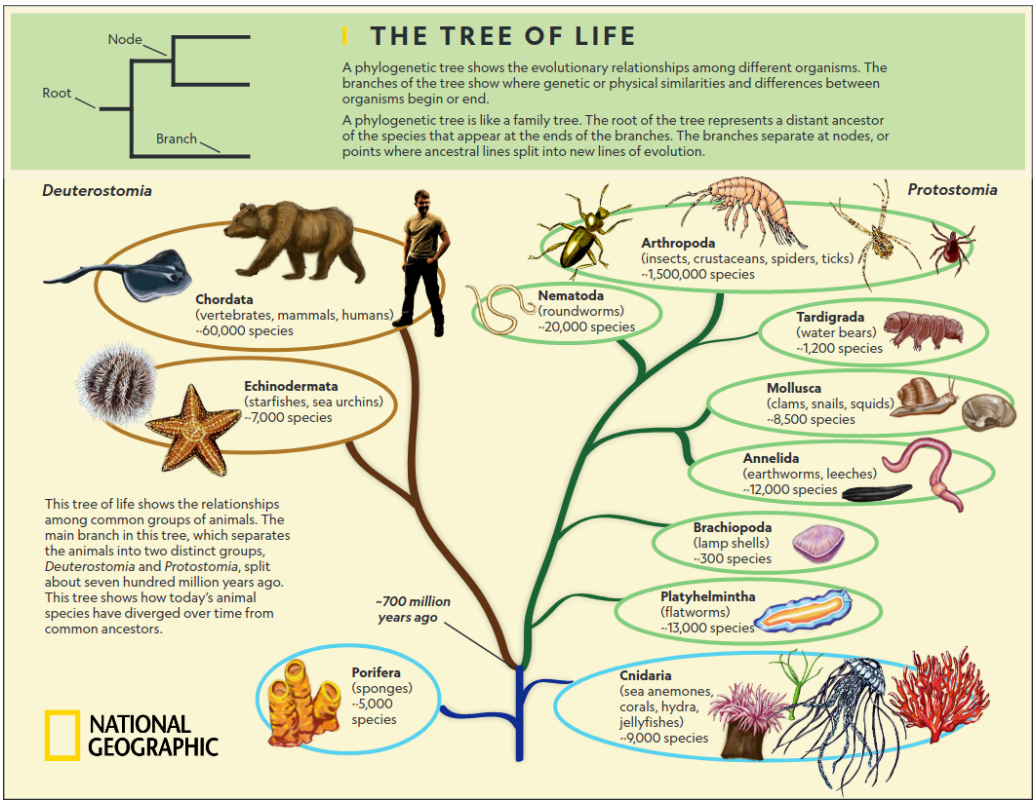
Hình 1.1 — Bối cảnh ứng dụng mạng xã hội và nhu cầu kết nối theo tính cách

Gợi ý hình: fig_context_social_apps.png

1.1.2. Rủi ro dữ liệu tính cách và yêu cầu bảo vệ

Dữ liệu tính cách có thể được suy diễn từ hành vi số hoặc từ bài trắc nghiệm, và thường được xem là dữ liệu nhạy cảm vì nó liên quan trực tiếp đến xu hướng tâm lý và hành vi của người dùng. Nhiều nghiên cứu chỉ ra rằng đặc điểm tính cách có thể dự đoán từ dữ liệu số và có mức độ ổn định cao [3]. Đồng thời, các đặc điểm này có thể bị khai thác để tác động đến hành vi, ví dụ trong các kịch bản thao túng nội dung hoặc quảng cáo cá nhân hóa quá mức [4]. Việc thu thập và lưu trữ tập trung vì thế cần được xem xét cẩn trọng về quyền riêng tư.

Trong bối cảnh đó, đề tài đặt ra yêu cầu bảo vệ dữ liệu tính cách ở mức tương tự như các loại dữ liệu nhạy cảm khác (tin nhắn, mật khẩu). Thay vì để dữ liệu gốc tồn tại dạng plaintext trên máy chủ, hệ thống cần có cơ chế chuyển đổi và mã hóa để giảm rủi ro rò rỉ. Hình Hình 1.2 mô tả các rủi ro chính khi xử lý dữ liệu tính cách theo mô hình tập trung.



Hình 1.2 — Rủi ro khi xử lý dữ liệu tính cách theo mô hình tập trung

Gợi ý hình: fig_privacy_risks.png

1.2. Mục tiêu và phạm vi

1.2.1. Mục tiêu chính

Mục tiêu của đề tài là xây dựng một pipeline chuyển đổi và bảo vệ dữ liệu tính cách, trong đó dữ liệu gốc được xử lý trên thiết bị, chuyển sang biểu diễn gọn hơn, và chỉ lưu trữ trên máy chủ dưới dạng mã hóa. Bên cạnh đó, hệ thống vẫn phải giữ khả năng so khớp và gợi ý người dùng một cách hiệu quả.

Các mục tiêu chính gồm:

- Xây dựng cơ chế chuyển đổi điểm Big Five sang không gian đặc trưng nhỏ gọn bằng PCA-4.
- Thiết kế cơ chế mã hóa AES-256-GCM để bảo vệ dữ liệu tính cách khi lưu trữ.
- Duy trì khả năng so khớp dựa trên cosine similarity để phục vụ pipeline gợi ý.

1.2.2. Phạm vi thực hiện

Đề tài tập trung vào khía cạnh chuyển đổi dữ liệu và bảo mật, không đi sâu vào triển khai giao diện hay tối ưu hóa trải nghiệm người dùng. Phạm vi hệ thống bao gồm:

- Thiết bị người dùng thực hiện chấm điểm Big Five và chuyển đổi PCA-4.
- Edge Function chịu trách nhiệm mã hóa và giải mã bằng AES-GCM.
- Cơ sở dữ liệu lưu trữ PCA vector và ciphertext thay vì dữ liệu thô.

Ngoài ra, từ các biểu diễn đã chuyển đổi này, hệ thống gợi ý sẽ khai thác thêm các nguồn dữ liệu đã được nhúng (embedding) từ sở thích và tương tác, nhằm tạo ra kết quả gợi ý có ý nghĩa thực tế nhưng vẫn giữ được nguyên tắc bảo mật thông tin cá nhân.

1.3. Bài toán và cách tiếp cận

1.3.1. Bài toán chuyển đổi dữ liệu tính cách

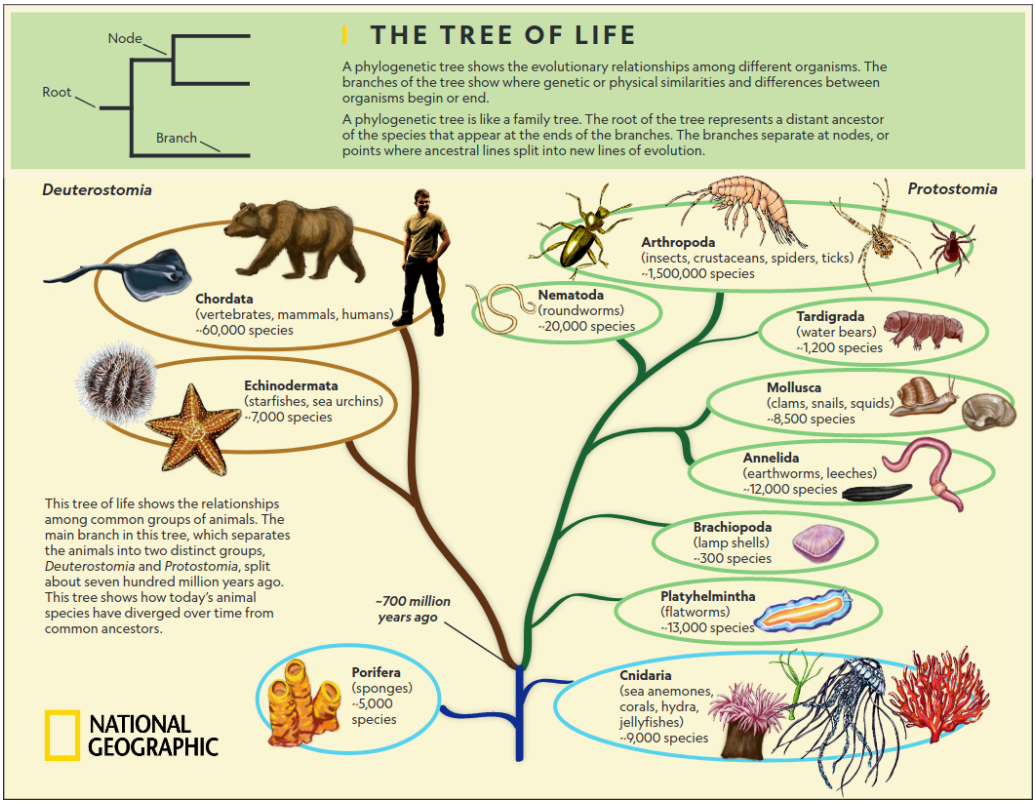
Bài toán đặt ra là chuyển đổi vector Big Five 5 chiều thành biểu diễn nhỏ gọn nhưng vẫn giữ được tính phân biệt đủ cao cho việc so khớp. Có nhiều hướng thay thế như dùng mô hình embedding ngữ nghĩa hoặc học sâu, nhưng các hướng này thường yêu cầu dữ liệu huấn luyện lớn hơn và khó giải thích.

Trong đề tài, PCA được chọn vì Big Five là mô hình tâm lý chuẩn hóa, đã có dữ liệu công khai quy mô lớn và ổn định theo quốc gia [1,2]. PCA cho phép giảm chiều mà vẫn giữ được phần lớn phương sai. Kết quả từ notebook thực nghiệm cho thấy PCA-4 giữ khoảng hơn 90% phương sai của dữ liệu gốc, trong khi PCA-2 hoặc PCA-3 mất đáng kể thông tin [5]. Hình Hình 1.3 mô tả pipeline chuyển đổi Big Five sang PCA-4.

Một điểm quan trọng là tính cách khác với ngôn ngữ tự nhiên. Đối với ngôn ngữ, việc nhúng văn bản thường dựa trên các semantic model lớn vì nội dung có tính mơ hồ, đa nghĩa và phụ thuộc ngữ cảnh. Trong khi đó, Big Five đã là một mô hình tâm lý chuẩn hóa, có cấu trúc dữ liệu rõ ràng và nguồn dữ liệu đủ lớn. Vì vậy PCA và cosine similarity phù hợp hơn cho phần tính cách, giúp giữ tính diễn giải và ổn định. Các mô hình semantic vẫn được sử dụng cho phần sở thích (hobbies), nơi dữ liệu là văn bản tự do và cần ánh xạ ngữ nghĩa.

Nói cách khác, đề tài không tìm cách "học lại" tính cách bằng mô hình ngôn ngữ, mà tận dụng một hệ đo đã có sẵn trong tâm lý học. PCA chỉ là bước nén và sắp xếp lại thông tin, không thay đổi ý nghĩa gốc của Big Five. Điều này giúp tránh lệch chuẩn khi dùng mô hình học sâu khó giải thích, đồng thời giảm phụ thuộc vào dữ liệu huấn luyện nội bộ. Tính cách vì thế được xử lý như một tín hiệu có cấu trúc, còn ngôn ngữ được xử lý như tín hiệu mở.

Ở cấp độ thu thập, hệ thống sử dụng bộ câu hỏi tính cách lớn hơn, sau đó chọn ngẫu nhiên 25 câu cho mỗi lượt làm bài. Mỗi 5 câu đại diện cho một nhóm trait, và điểm số được cộng hoặc trừ tùy theo hướng câu hỏi. Mô hình không phụ thuộc nội dung câu hỏi mà chỉ quan tâm đến hướng (key) và trait tương ứng. Cách tiếp cận này giúp duy trì tính nhất quán của thang đo trong khi giảm tải thời gian trả lời cho người dùng.



Hình 1.3 — Pipeline chuyển đổi Big Five sang vector PCA-4

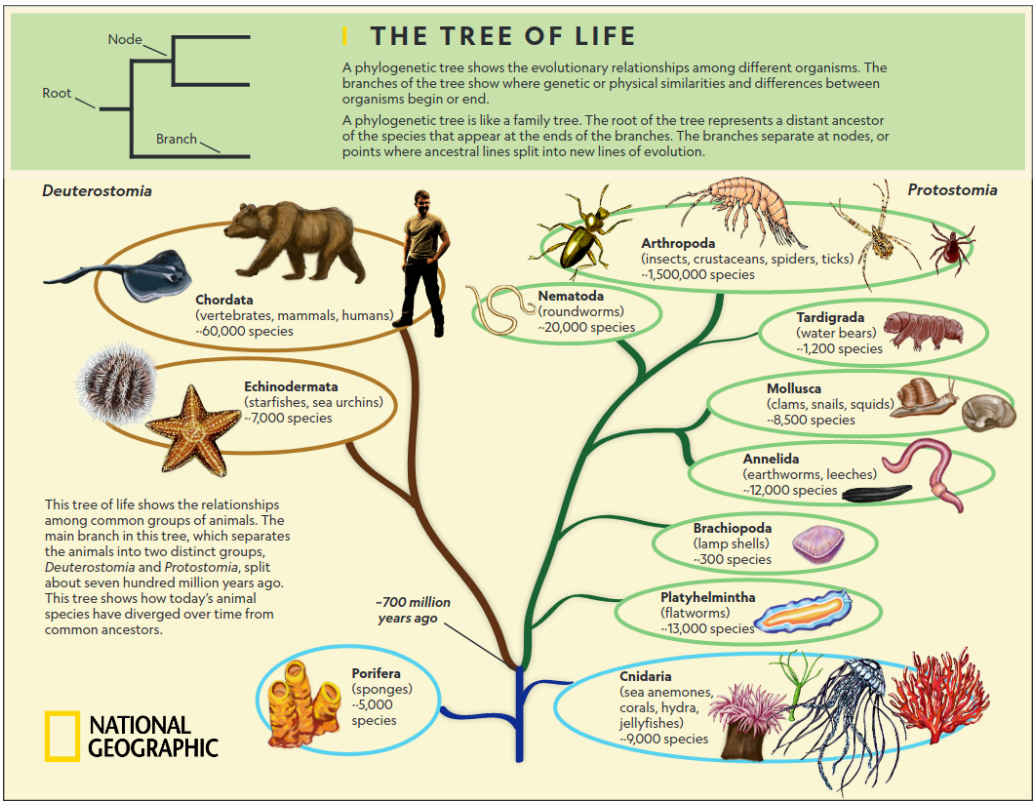
Gợi ý hình: fig_pca_pipeline.png

1.3.2. Bài toán bảo mật dữ liệu

PCA không phải cơ chế bảo mật. Các thành phần PCA có thể bị suy ngược gần đúng nếu biết tham số mô hình. Vì vậy, dữ liệu gốc vẫn cần được mã hóa.

Trong số các phương án, AES-256-GCM được chọn vì phù hợp với payload nhỏ, tốc độ cao và có tính toàn vẹn dữ liệu (integrity) nhờ GCM [6]. So với RSA hoặc Bcrypt, AES-GCM ít tốn tài nguyên hơn cho dữ liệu dạng JSON, và phù hợp với mô hình edge function.

Trong hệ thống, khóa AES chỉ nằm ở phía server (Edge Function). Thiết bị người dùng không giữ khóa, nhằm tránh nguy cơ bị trích xuất từ ứng dụng và vẫn cho phép khôi phục dữ liệu khi đăng nhập lại trên thiết bị khác. Hình Hình 1.4 mô tả luồng mã hóa và lưu trữ dữ liệu tính cách.



Hình 1.4 — Luồng mã hóa AES-GCM và lưu trữ dữ liệu tính cách

Gợi ý hình: fig_encrypt_flow.png

1.4. Đóng góp chính

1.4.1. Đóng góp về mô hình chuyển đổi

Đề tài xây dựng pipeline chuyển đổi Big Five sang PCA-4 chạy trên thiết bị, đảm bảo giảm kích thước dữ liệu nhưng vẫn giữ phần lớn thông tin. Hệ số PCA được huấn luyện trên tập dữ liệu công khai quy mô lớn, giúp kết quả có tính ổn định và tái lập.

1.4.2. Đóng góp về bảo mật

Đề tài đề xuất cơ chế mã hóa AES-256-GCM qua Edge Function, đảm bảo dữ liệu gốc không lưu plaintext trên cơ sở dữ liệu. Cách tiếp cận này cân bằng giữa khả năng so khớp và yêu cầu bảo mật dữ liệu nhạy cảm.

1.5. Cấu trúc của báo cáo

Phần còn lại của báo cáo được trình bày như sau:

- Chương 2: Trình bày pipeline tổng thể của hệ thống Twins, từ thu thập dữ liệu đến gợi ý.
- Chương 3: Phân tích chi tiết PCA-4, dữ liệu huấn luyện và cách chuyển đổi.
- (Dự kiến) Chương 4: Trình bày cơ chế bảo mật và luồng mã hóa/giải mã.
- (Dự kiến) Chương 5: Trình bày hệ gợi ý (PCA, ELO, hobbies) và cách tính trọng số.
- (Dự kiến) Chương 6: Thực nghiệm và đánh giá hệ thống.
- (Dự kiến) Chương 7: Kết luận và hướng phát triển.

Chương 2

Tổng quan pipeline hệ thống

2.1. Mục tiêu của chương

Chương này trình bày pipeline tổng thể của hệ thống Twins, theo thứ tự từ thu thập dữ liệu trên thiết bị, chuyển đổi và bảo mật, đến gợi ý người dùng. Mục tiêu là mô tả rõ các tác nhân tham gia, dữ liệu vào ra ở mỗi bước và cách các điểm số được kết hợp thành một điểm xếp hạng cuối cùng. Các chương sau sẽ đi sâu vào từng thành phần. Trong đó, Chương 4 tập trung vào bảo mật và mã hóa dữ liệu, còn Chương 5 trình bày chi tiết hệ gợi ý và các công thức xếp hạng.

2.2. Các nguồn dữ liệu đầu vào

2.2.1. Bộ câu hỏi Big Five và cách lấy mẫu

Hệ thống sử dụng tập câu hỏi Big Five lớn, được tổng hợp từ các bộ câu hỏi chuẩn như IPIP 50 và các biến thể đã được công bố rộng rãi [7]. Mỗi lượt làm bài chọn ngẫu nhiên 25 câu từ pool 150 câu, trong đó mỗi 5 câu đại diện cho một trait. Mỗi câu hỏi có hướng cộng hoặc trừ vào trait tương ứng, do đó mô hình không phụ thuộc nội dung câu hỏi mà chỉ phụ thuộc vào hướng (key) và trait của câu hỏi.

Cách lấy mẫu này giúp giảm thời gian làm bài, đồng thời vẫn giữ được cấu trúc cân bằng giữa các trait. Trên thực tế, hệ thống chỉ cần biết hai thông tin cho mỗi câu: thuộc tính trait nào và hướng tính điểm (cộng hay trừ). Nội dung câu hỏi được giữ để đảm bảo ngữ cảnh người dùng, nhưng không ảnh hưởng đến mô hình chuyển đổi. Trong Chương 3 sẽ trình bày chi tiết cách tính điểm từ thang Likert và quy trình chuẩn hóa.

2.2.2. Dữ liệu khảo sát công khai cho PCA

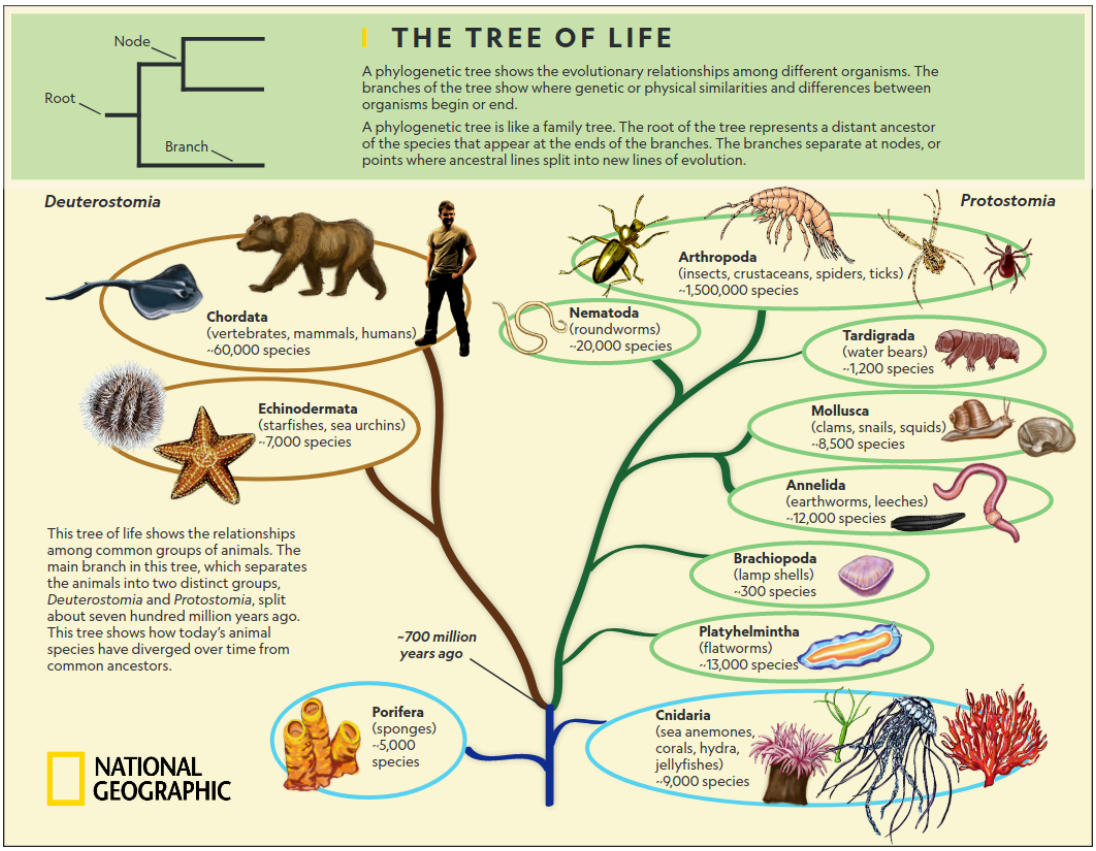
Để huấn luyện PCA, đề tài sử dụng tập dữ liệu Big Five công khai với hơn 300 nghìn mẫu từ nhiều quốc gia [5]. Dữ liệu đã được chuẩn hóa về thang 0-1 cho từng trait, phù hợp cho việc ước lượng các thành phần chính. Các kết quả giải thích phương sai sẽ được nêu ở Chương 3. Đây là lợi thế của Big Five: dữ liệu chuẩn hóa, quy mô lớn và đã được sử dụng rộng rãi trong nghiên cứu, nên PCA có thể học được cấu trúc phân bố ổn định.

2.2.3. Dữ liệu sở thích (hobbies)

Sở thích người dùng được nhập dưới dạng văn bản ngắn. Văn bản này không dùng để lưu trữ trực tiếp, mà được chuyển thành vector embedding 384 chiều thông qua mô hình semantic embedding từ Jina. Lý do dùng embedding là để so khớp nội dung sở thích theo ngữ nghĩa thay vì so khớp từ khóa đơn thuần. Cách làm này cho phép các sở thích có nghĩa gần nhau (ví dụ "chạy bộ" và "jogging") vẫn được đánh giá tương đồng. Chi tiết quy trình embedding và luồng mã hóa dữ liệu sở thích sẽ được mô tả ở Chương 5.

2.3. Tổng quan pipeline và tác nhân

Hệ thống có ba tác nhân chính: thiết bị người dùng, Edge Function và cơ sở dữ liệu. Hình Hình 2.1 mô tả pipeline tổng thể từ thu thập dữ liệu đến gợi ý.



Hình 2.1 — Pipeline tổng thể của hệ thống Twins

Gợi ý hình: fig_pipeline_overview.png

Các bước chính gồm:

- Thiết bị người dùng trả lời 25 câu hỏi, chấm điểm Big Five và chuẩn hóa về thang 0-1.
- Thiết bị chuyển đổi PCA-4 bằng tham số đã huấn luyện sẵn.
- Thiết bị gửi dữ liệu Big Five gốc tới Edge Function để mã hóa AES-256-GCM.
- Cơ sở dữ liệu lưu trữ pca_dim1..4 và ciphertext (b5_cipher, b5_iv).
- Dữ liệu sở thích được embedding thành vector 384 chiều, mã hóa, và lưu trữ tương tự.
- Hệ gợi ý lấy PCA vector, ELO và embedding sở thích để tính điểm xếp hạng.

2.4. Mô hình điểm và trọng số trong gợi ý

2.4.1. Điểm tương đồng tính cách (PCA)

Vector PCA-4 được dùng để đo tương đồng giữa hai người dùng bằng cosine similarity. Cosine similarity phù hợp vì đo góc giữa hai vector, ít bị ảnh hưởng bởi độ lớn tuyệt đối và ổn định khi dữ liệu đã chuẩn hóa [8]. Công thức cosine similarity sẽ được trình bày chi tiết ở Chương 5.

2.4.2. ELO từ tương tác like/skip

Hệ thống dùng điểm ELO như một thước đo xã giao, phản ánh mức độ tương tác qua hành vi like và skip. Điểm ELO được cập nhật theo kỳ vọng thắng thua trong mô hình Elo gốc, nhưng được điều chỉnh để phù hợp với ngữ cảnh kết nối xã hội [9]. Trong hệ thống:

- Like: cả hai phía tăng nhẹ.
- Skip: chỉ người chủ động skip bị trừ.

Điểm ELO không phải thước đo hấp dẫn tuyệt đối, mà là tín hiệu phụ để gom nhóm người dùng có mức tương tác tương đồng. ELO trong Twins là hệ số ẩn, được cập nhật sau mỗi lần tương tác và bị giới hạn trong khoảng 800 đến 2000. Lưu ý rằng cách cập nhật này tạo xu hướng lạm phát điểm ELO theo thời gian, vì lượt "like" làm cả hai phía tăng điểm. Tuy vậy, mục đích chính không phải cạnh tranh, mà là đảm bảo người dùng có mức xã giao gần nhau được ưu tiên gặp nhau hơn.

Trong công thức gốc, kỳ vọng thắng được tính bởi:

$$E_a = \frac{1}{1 + 10^{\frac{R_b - R_a}{400}}} \tag{2.1}$$

Sau đó cập nhật theo $R_{a'} = R_a + K(S_a - E_a)$. Trong Twins, kết quả like được coi là một tín hiệu hợp tác nên cả hai phía tăng nhẹ, còn skip chỉ trừ phía chủ động. Cụ thể, với K=12 và clamp trong [800, 2000]:

- Like: $R_{a'} = \text{clamp}(R_a + K(1 - E_a))$, $R_{b'} = \text{clamp}(R_b + K(1 - E_b))$.
- Skip: $R_{a'} = \text{clamp}(R_a + K(0 - E_a))$, $R_{b'} = R_b$.

Bên cạnh đó, hệ gợi ý sử dụng hệ số gần nhau ELO để ưu tiên mức xã giao tương đồng:

$$p = \exp\left(-|\Delta R\frac{|}{\sigma}\right) \tag{2.2}$$

trong đó $\sigma = 400$.

2.4.3. Embedding sở thích và cosine similarity

Sở thích người dùng được chuyển thành vector 384 chiều thông qua mô hình semantic embedding. Cosine similarity được dùng để đo độ gần về sở thích, thay vì so khớp từ khóa. Cách làm này cho phép hai người dùng dùng từ khác nhau nhưng có ý nghĩa gần nhau vẫn được đánh giá cao hơn.

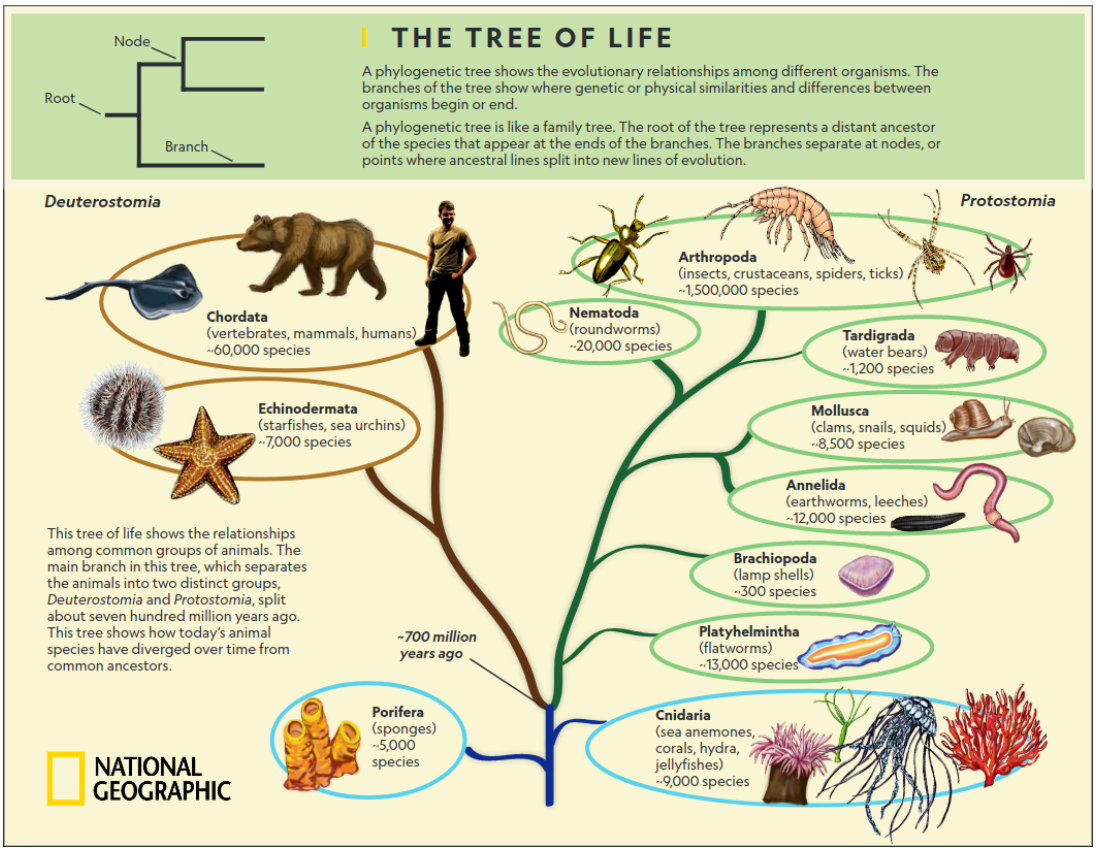
2.4.4. Trọng số tổng hợp

Điểm xếp hạng cuối cùng được tính theo trọng số của PCA, ELO và hobbies. Trong phiên bản hiện tại:

- Khi không dùng hobbies:
  - Nếu ELO bật: score = 0.8 **PCA + 0.2** ELO proximity.
  - Nếu ELO tắt: score = PCA.
- Khi dùng hobbies:
  - Nếu ELO bật: `score = 0.5 * PCA + 0.2 * ELO + 0.3 * Hobbies`.
  - Nếu ELO tắt: `score = 0.55 * PCA + 0.45 * Hobbies`.

Để minh họa, xét ba người dùng A, B, C khi A đang tìm gợi ý. Giả sử A có PCA tương đồng với B và C gần bằng nhau (ví dụ 0.90), nhưng B có sở thích gần hơn (hobbies 0.85) trong khi C có ELO gần hơn (proximity 1.0 so với 0.7). Trong cấu hình có ELO và hobbies, điểm cuối có thể làm B đứng trước nếu lợi thế sở thích lớn hơn lợi thế ELO. Trường hợp ngược lại, nếu B và C ngang nhau về hobbies, thì C sẽ được ưu tiên do proximity cao hơn. Ví dụ này thể hiện vai trò của từng trọng số trong việc phá vỡ tình huống hòa điểm.

Hình Hình 2.2 minh họa sơ đồ trọng số và các nhánh tính điểm.



Hình 2.2 — Sơ đồ trọng số tính điểm gợi ý

Gợi ý hình: fig_score_weights.png

2.5. Luồng dữ liệu chi tiết theo tác nhân

2.5.1. Thiết bị người dùng

Thiết bị thực hiện các bước sau:

- Thu thập câu trả lời và chấm điểm Big Five.
- Chuẩn hóa và chuyển đổi PCA-4.
- Gửi dữ liệu thô tới Edge Function để mã hóa.
- Gửi văn bản sở thích để tạo embedding, rồi lưu ciphertext và vector embedding.

2.5.2. Edge Function

Edge Function đảm nhận:

- Mã hóa/giải mã Big Five bằng AES-256-GCM.
- Gọi dịch vụ embedding để sinh vector sở thích.

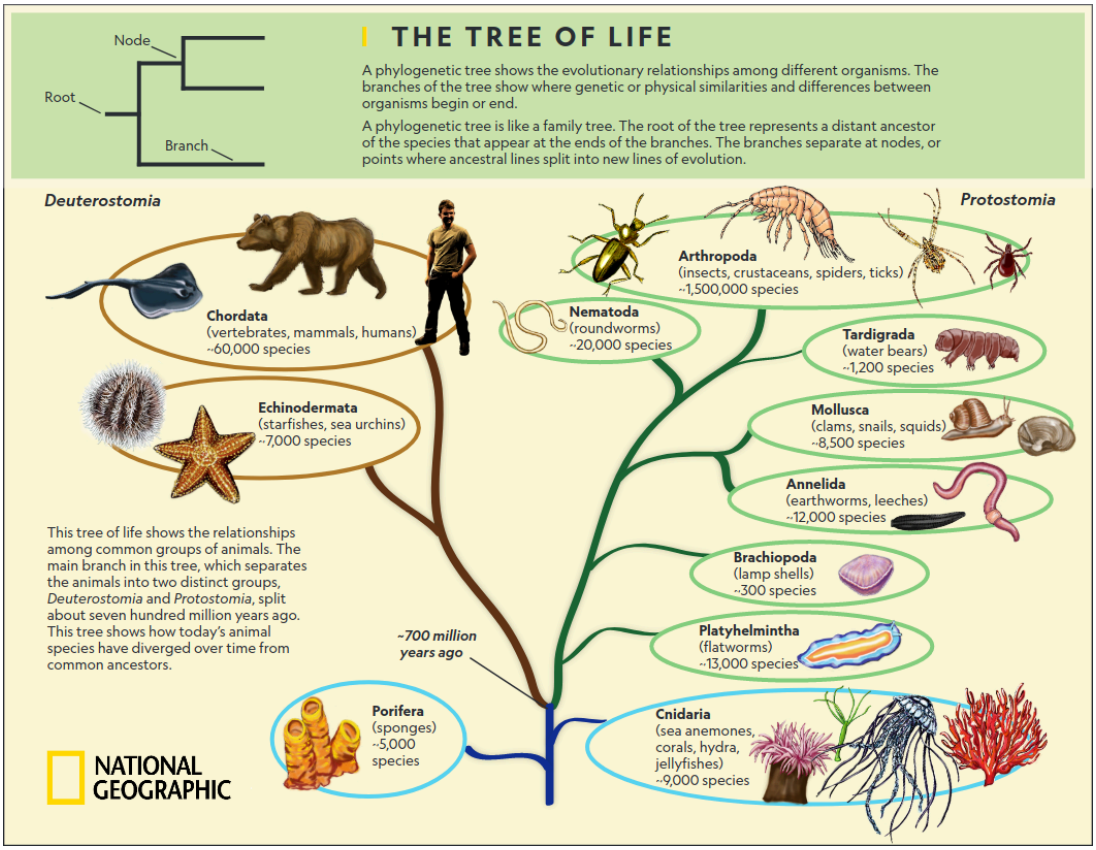
- Trả về ciphertext, iv và vector embedding cho thiết bị.

2.5.3. Cơ sở dữ liệu

Cơ sở dữ liệu lưu trữ:

- PCA vector (pca_dim1..4).
- Ciphertext và iv cho Big Five (b5_cipher, b5_iv).
- Ciphertext cho hobbies và vector embedding.

Hình Hình 2.3 trình bày luồng dữ liệu theo thứ tự tác nhân.



Hình 2.3 — Luồng dữ liệu giữa thiết bị, Edge Function và cơ sở dữ liệu

Gợi ý hình: fig_dataflow_sequence.png

Chương 3

Chuyển đổi dữ liệu tính cách (PCA-4)

3.1. Mục tiêu của chương

Chương này trình bày chi tiết quy trình chuyển đổi dữ liệu Big Five sang vector PCA-4, bao gồm cách chuẩn hóa điểm, cách huấn luyện PCA và cách triển khai trên thiết bị. Mục đích là làm rõ vì sao PCA-4 được chọn thay vì PCA-2/3 hoặc các mô hình embedding khác.

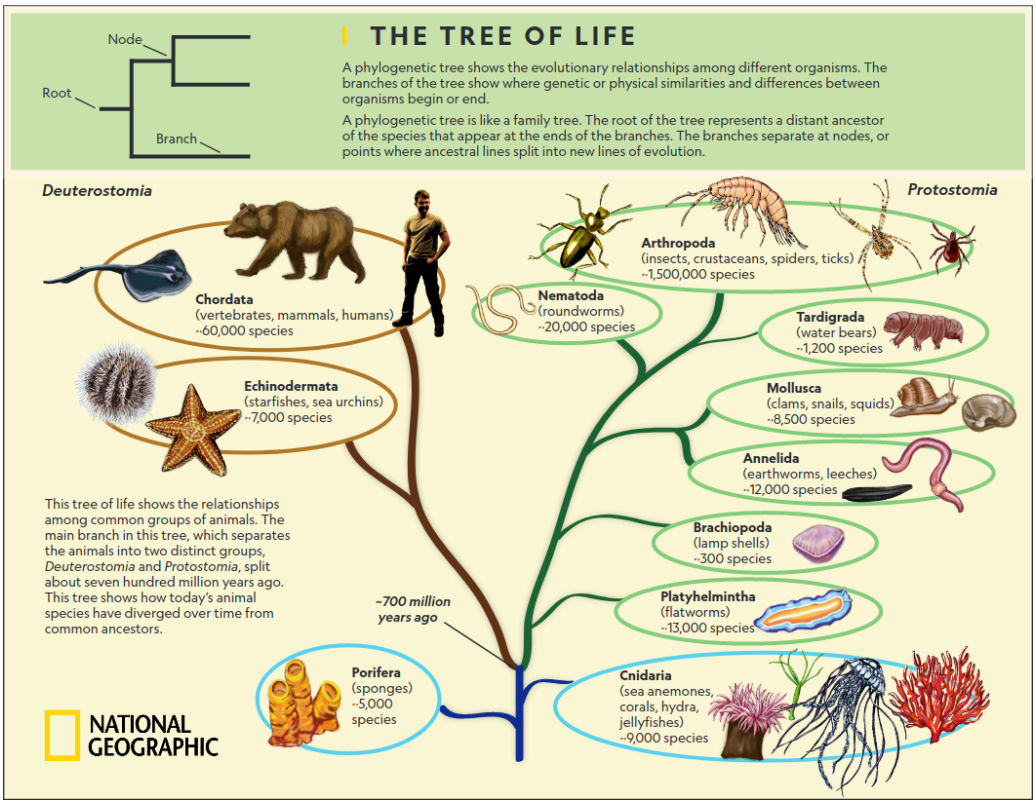
3.2. Big Five trong bối cảnh các mô hình tính cách

Trong tâm lý học có nhiều khung mô tả tính cách, không có mô hình nào tuyệt đối hoàn hảo. Big Five được sử dụng vì đã có lịch sử nghiên cứu dài, hệ thống câu hỏi chuẩn hóa và dữ liệu công khai phong phú. So với các mô hình khác như MBTI hoặc HEXACO, Big Five có ưu thế về tính tái lập và độ phủ dữ liệu, phù hợp cho bài toán chuyển đổi số liệu quy mô lớn [2,10]. Vì vậy, đề tài chấp nhận giới hạn của mô hình nhưng coi Big Five là lựa chọn thực tế nhất để làm nền cho pipeline chuyển đổi dữ liệu.

3.2.1. Mô hình MBTI (Myers-Briggs Type Indicator)

MBTI phân loại người dùng theo các cặp đối lập, tạo ra 16 nhóm tính cách. Cách biểu diễn này dễ truyền thông nhưng thiên về phân loại rời rạc, trong khi dữ liệu thực tế thường có phân bố liên tục. Với bài toán gợi ý cần đo mức độ gần nhau, dạng nhãn rời rạc làm giảm khả năng xếp hạng chi tiết và khó phản ánh mức độ "gần" giữa hai cá nhân. MBTI cũng có vấn đề về độ ổn định theo thời gian, nhiều người thay đổi nhóm khi làm lại bài test. Điều này làm cho dữ liệu khó tái lập và khó dùng cho pipeline so khớp dài hạn. Ngoài ra, MBTI ít có dữ liệu mở quy mô lớn theo chuẩn hóa số điểm, nên khó dùng cho chuyển đổi

PCA và huấn luyện ổn định. Ví dụ, hai người thuộc nhóm INFP và ENFP có thể khác nhau mạnh về hướng ngoại nhưng vẫn bị xem là hai nhãn rời rạc. Hình Hình 3.1 minh họa cách MBTI chia nhóm tính cách.

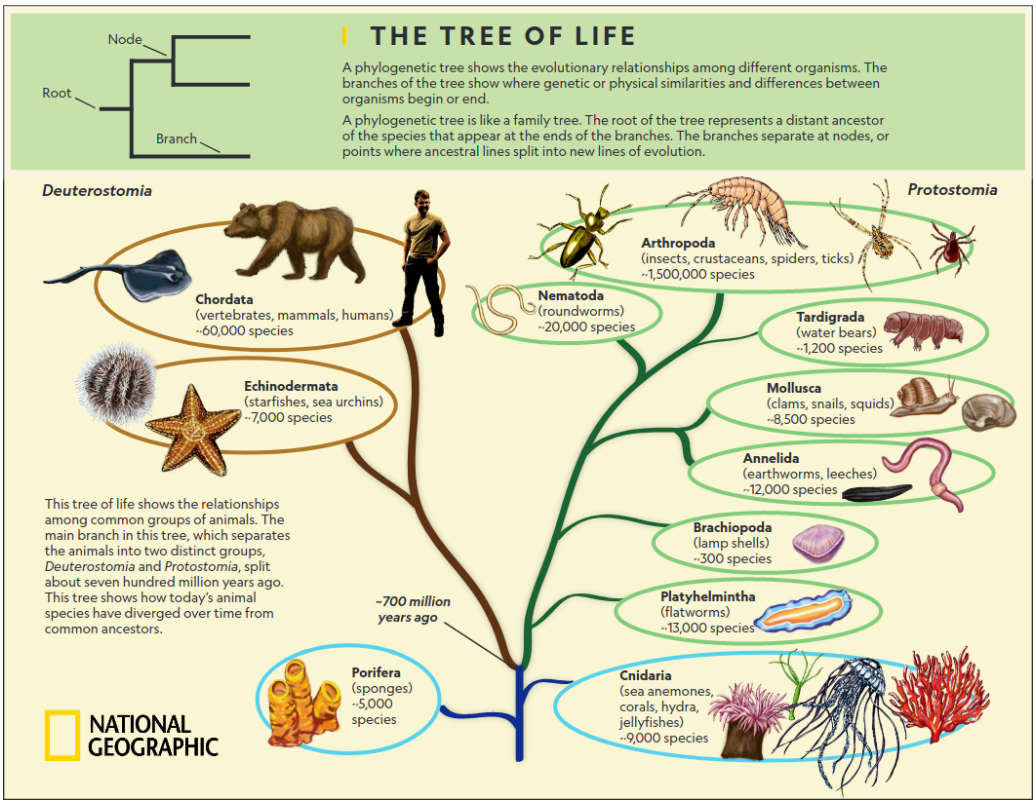


Hình 3.1 — Minh họa mô hình MBTI và cách phân nhóm tính cách

Gợi ý hình: fig_mbti_overview.png

3.2.2. Mô hình HEXACO

HEXACO mở rộng Big Five bằng cách thêm yếu tố Honesty-Humility. Mô hình này có giá trị về mặt học thuật, nhưng dữ liệu mở và bộ câu hỏi chuẩn hóa không phổ biến bằng Big Five. Việc thêm một trait thứ sáu làm tăng số câu hỏi cần thiết để giữ cân bằng độ tin cậy. Điều này gây áp lực lên trải nghiệm người dùng di động, vì thời gian trả lời dài hơn. Ngoài ra, chuyển đổi từ HEXACO sang dạng PCA sẽ cần dữ liệu huấn luyện riêng, trong khi dữ liệu chuẩn không nhiều bằng Big Five. Ví dụ, nếu chỉ dùng 25 câu, mỗi trait sẽ bị giảm số câu đánh giá, làm tăng nhiễu đo lường. Do đó HEXACO được xem là lựa chọn tham khảo hơn là lựa chọn chính cho đề tài. Hình Hình 3.2 minh họa cấu trúc HEXACO.



Hình 3.2 — Minh họa cấu trúc 6 yếu tố của HEXACO

Gợi ý hình: fig_hexaco_overview.png

3.3. Chuẩn hóa điểm Big Five

3.3.1. Thang đo và hướng câu hỏi

Mỗi câu trả lời được chấm theo thang Likert 1–5. Với câu hỏi hướng dương, điểm giữ nguyên thứ tự 1→5. Với câu hỏi hướng âm, điểm được đảo chiều. Sau đó các điểm trong cùng trait được cộng lại và chuẩn hóa về thang 0–1. Cách chuẩn hóa này giúp các trait có cùng thang đo, phù hợp cho PCA và so khớp cosine.

3.3.2. Ví dụ định dạng dữ liệu đầu vào

Sau bước chuẩn hóa, mỗi người dùng có một vector 5 chiều theo thứ tự trait cố định:

```
x = [Extraversion, Agreeableness, Conscientiousness, Emotional Stability,
Intellect]
```

Ví dụ một người dùng có thể có:

```
x = [0.68, 0.55, 0.72, 0.60, 0.47]
```

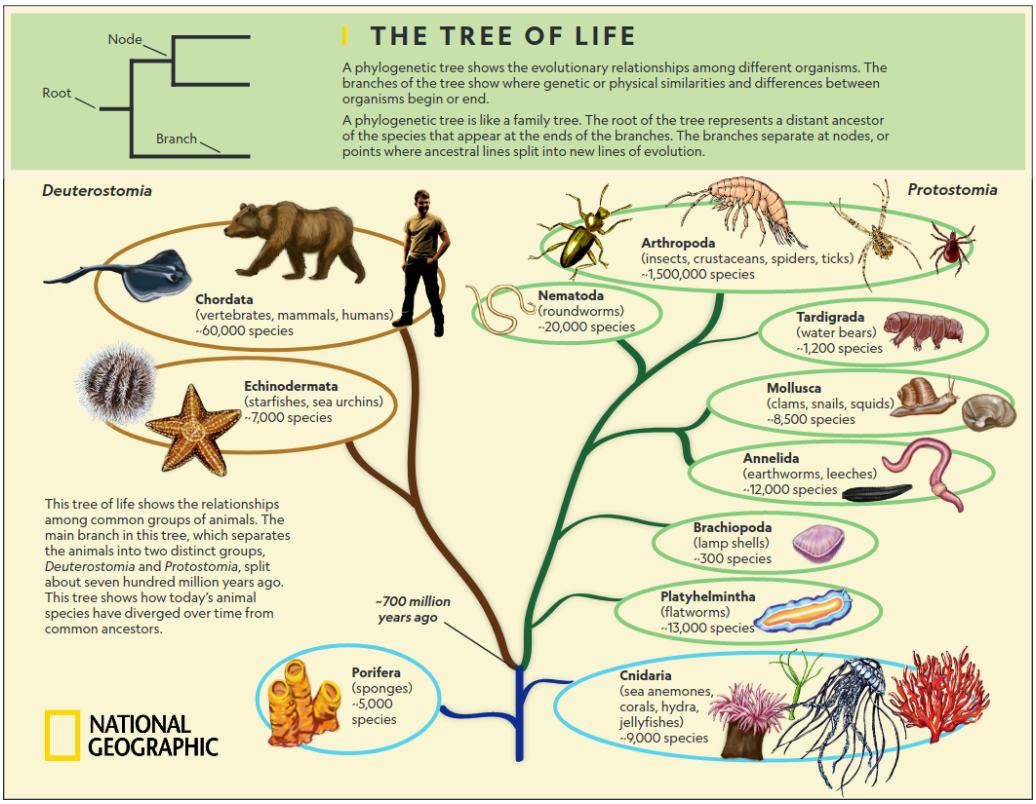
Đây là dạng dữ liệu đầu vào cho bước PCA.

3.3.3. Vì sao chọn PCA-4 sau khi chuẩn hóa

Chuẩn hóa đưa dữ liệu Big Five về cùng thang đo, giúp mỗi trait đóng góp cân bằng khi so khớp và khi học PCA. Tuy vậy, chuẩn hóa không giải quyết vấn đề dư thừa thông tin giữa các trait. PCA được dùng để rút gọn chiều và tách các trục phương sai lớn nhất. PCA-2 hoặc PCA-3 giảm nhiều hơn nhưng mất đáng kể thông tin, làm giảm khả năng phân biệt giữa các hồ sơ gần nhau. PCA-4 là điểm cân bằng: giảm chiều từ 5 xuống 4 nhưng vẫn giữ phần lớn phương sai, giúp hệ gợi ý hoạt động ổn định khi đo cosine similarity. Vì vậy, PCA-4 được chọn sau bước chuẩn hóa như một lớp chuyển đổi tối ưu cho dữ liệu tính cách.

3.4. Đề xuất PCA-4

Đề tài đề xuất PCA-4 như mức giảm chiều tối ưu cho Big Five trong bối cảnh gợi ý bạn bè. Giảm từ 5 xuống 4 chiều giúp tiết kiệm lưu trữ mà vẫn giữ phần lớn cấu trúc dữ liệu. PCA-4 cũng là dạng biểu diễn dễ triển khai trên thiết bị với phép nhân ma trận thuần. Mức giảm nhẹ này giúp hạn chế rủi ro mất thông tin so với PCA-2 hoặc PCA-3. Ngoài ra, PCA-4 giữ được tính diễn giải tương đối, phù hợp với việc so sánh cosine similarity ổn định. Hình Hình 3.3 gợi ý một minh họa quyết định chọn PCA-4 dựa trên phương sai.



Hình 3.3 — Minh họa tiêu chí lựa chọn PCA-4

Gợi ý hình: fig_pca_proposal.png

3.5. Huấn luyện PCA

3.5.1. Nguồn dữ liệu và quy mô

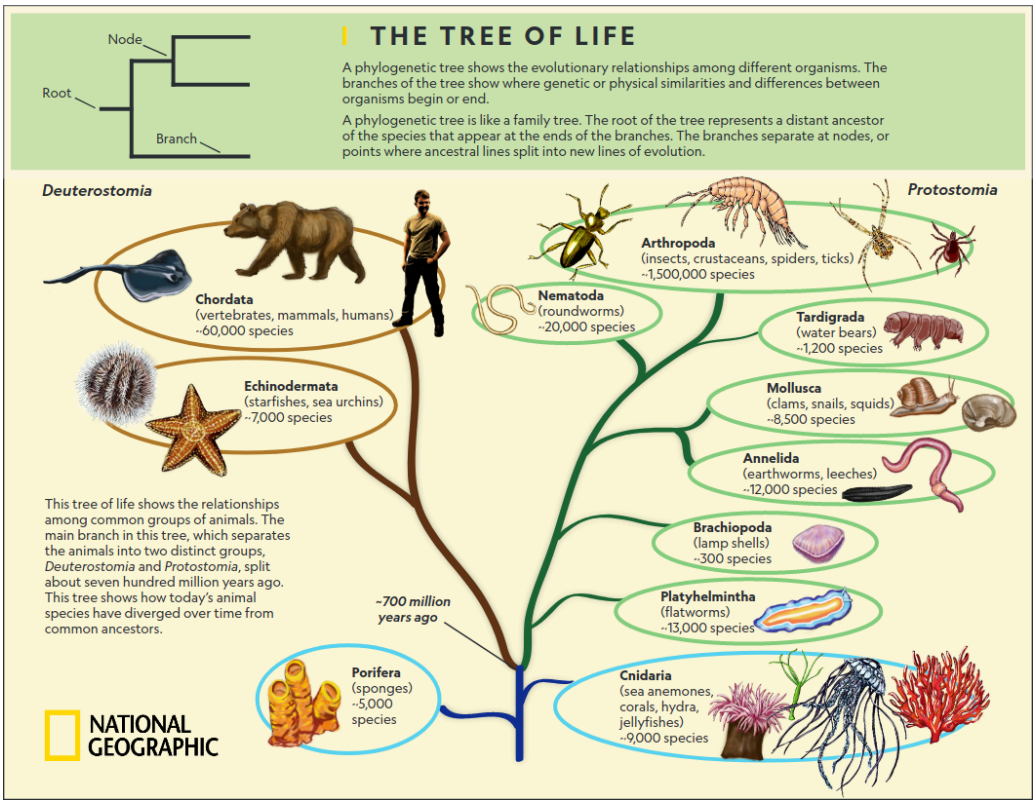
PCA được huấn luyện từ tập dữ liệu Big Five công khai quy mô lớn [5]. Tập này đã chuẩn hóa điểm về thang 0–1, giảm sai lệch do khác hệ đo, và có phân bố ổn theo nhiều quốc gia. Việc dùng dữ liệu lớn giúp các thành phần chính ổn định và dễ tái lập.

3.5.2. Công thức chiếu PCA

PCA thực hiện phép chiếu tuyến tính trên dữ liệu đã được trừ mean. Với vector đầu vào x (dài 5), ta có:

$$z = (x - \mu) \times W^T \tag{3.3}$$

trong đó μ là vector mean và W là ma trận thành phần chính [11]. Vector z là PCA-4 và được lưu dưới dạng 4 chiều. Hình Hình 3.4 mô tả phép chiếu và định dạng đầu ra.



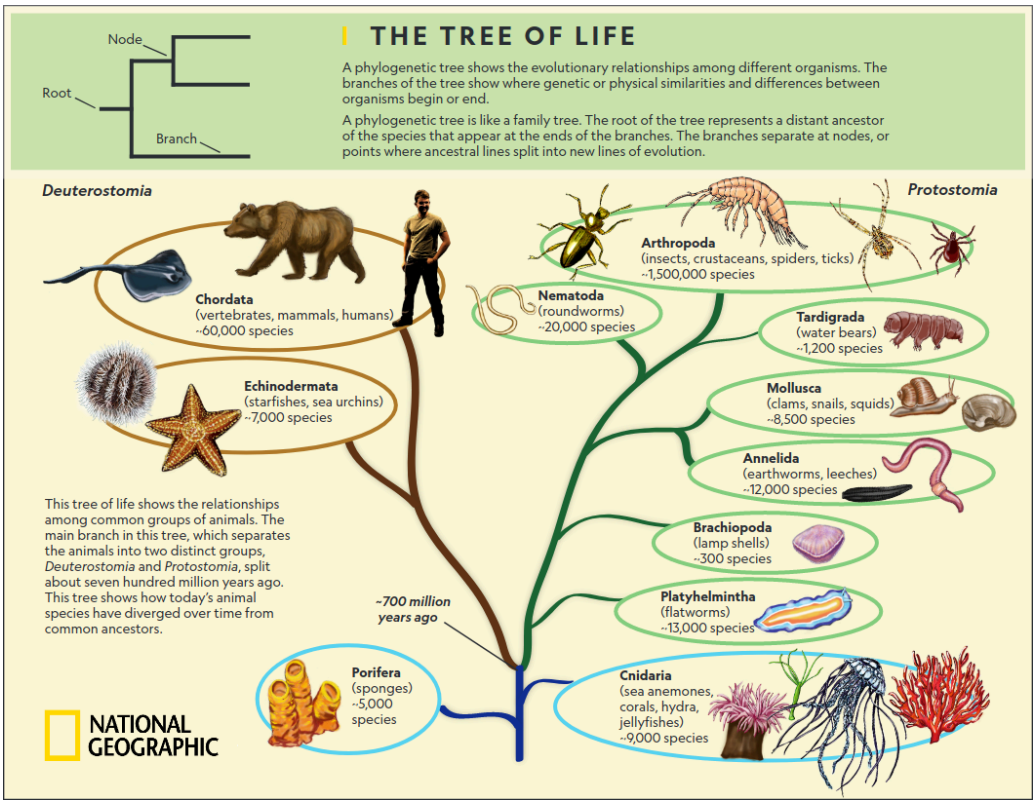
Hình 3.4 — Minh họa phép chiếu PCA và định dạng vector đầu ra

Gợi ý hình: fig_pca_math.png

3.5.3. So sánh PCA-2, PCA-3, PCA-4

Trong notebook thực nghiệm, PCA-2 chỉ giữ khoảng 63% phương sai, PCA-3 khoảng 80%, trong khi PCA-4 giữ hơn 90% phương sai dữ liệu gốc. Sự chênh lệch này ảnh hưởng trực tiếp đến khả năng phân biệt giữa các người dùng khi so khớp. Vì vậy PCA-4 được chọn để giảm mất thông tin mà vẫn đảm bảo kích thước nhỏ gọn.

Hình Hình 3.5 minh họa đồ thị phương sai giải thích theo số chiều.



Hình 3.5 — Đồ thị phương sai giải thích theo số chiều PCA

Gợi ý hình: fig_pca_variance.png

3.6. Triển khai PCA trên thiết bị

3.6.1. Cách triển khai

Thay vì chạy mô hình học sâu, PCA-4 được triển khai bằng phép nhân ma trận thuần trên thiết bị. Hệ số mean và components được trích từ notebook huấn luyện và lưu cố định trong ứng dụng. Cách này giảm phụ thuộc vào thư viện ML và hạn chế kích thước bundle.

3.6.2. Định dạng lưu trữ

Kết quả PCA-4 được lưu dưới dạng 4 trường số: pca_dim1..pca_dim4. Các giá trị này được lưu song song với ciphertext của Big Five. Việc lưu PCA dạng số thực giúp tính cosine similarity trực tiếp ở phía server khi gợi ý.

3.7. Thảo luận lựa chọn PCA

PCA là phép biến đổi tuyến tính, có thể giải thích và kiểm soát. Các lựa chọn thay thế như embedding học sâu hoặc semantic embedding không phù hợp vì dữ liệu tính cách đã có cấu trúc rõ ràng và ít phụ thuộc ngôn ngữ. Ngoài ra, PCA giúp duy trì tính ổn định giữa các phiên bản, tránh lệch kết quả do thay đổi mô hình.

Chương 4

Bảo mật và mã hóa dữ liệu

4.1. Mục tiêu của chương

Chương này trình bày cách dữ liệu được nhập từ góc độ người dùng, cách dữ liệu được chuyển đổi và mã hóa trước khi lưu trữ, cùng với lý do lựa chọn cơ chế AES-256-GCM. Trọng tâm là luồng dữ liệu và các tác nhân, không đi sâu vào mã nguồn.

4.2. Tổng quan về cơ chế AES-GCM

4.2.1. Nguyên lý cơ bản

AES là thuật toán mã hóa đối xứng khối, hoạt động trên các block cố định và cần một khóa chung cho cả mã hóa lẫn giải mã. GCM (Galois/Counter Mode) là chế độ hoạt động kết hợp giữa mã hóa dạng counter và xác thực dữ liệu. Nhờ đó, ngoài ciphertext, hệ thống còn có thể kiểm tra tính toàn vẹn của dữ liệu [6]. Trong ngữ cảnh dữ liệu tính cách, đây là điểm quan trọng vì tránh tình trạng ciphertext bị chỉnh sửa âm thầm.

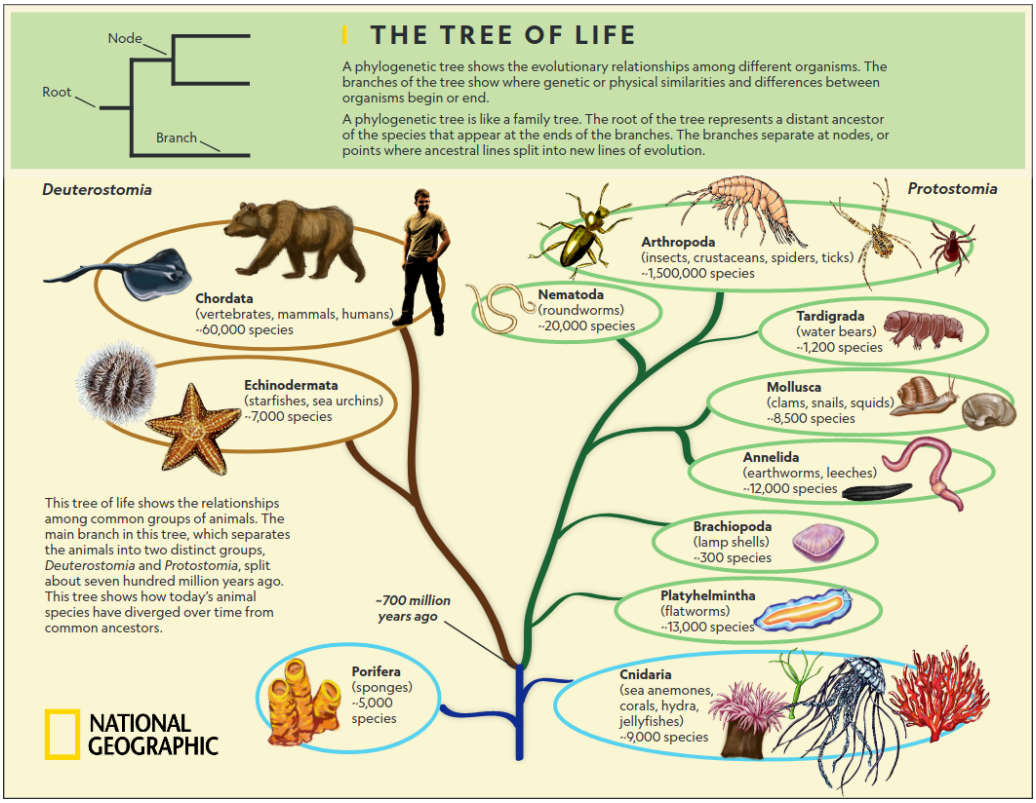
Một phiên AES-GCM tạo ra thêm authentication tag, giúp phát hiện việc thay đổi dữ liệu hoặc iv. Nếu tag không khớp, dữ liệu sẽ bị từ chối giải mã. Cơ chế này làm giảm nguy cơ người dùng nhận dữ liệu sai hoặc bị chỉnh sửa khi truyền qua mạng. Với dữ liệu nhạy cảm như tính cách và sở thích, việc đảm bảo tính toàn vẹn quan trọng không kém việc giữ bí mật. Vì vậy AES-GCM phù hợp hơn các chế độ chỉ mã hóa mà không xác thực.

4.2.2. Đầu vào và đầu ra của AES-GCM

Đầu vào gồm dữ liệu gốc (JSON điểm Big Five hoặc danh sách hobbies), khóa bí mật, và iv ngẫu nhiên. Đầu ra gồm ciphertext và iv. Trong triển khai của

đề tài, iv được lưu riêng trong cơ sở dữ liệu để phục vụ giải mã. Hình Hình 4.1 mô tả cấu trúc đầu vào và đầu ra của AES-GCM.

Ngoài ciphertext, AES-GCM còn sinh authentication tag. Tag được lưu kèm ciphertext để khi giải mã có thể kiểm tra tính toàn vẹn. Nếu tag không khớp, hệ thống từ chối giải mã và ghi log lỗi để tránh trả dữ liệu sai. Cách lưu trữ này giúp dữ liệu cá nhân không bị thay đổi âm thầm ở cấp độ cơ sở dữ liệu hoặc trong quá trình truyền tải.



Hình 4.1 — Định dạng đầu vào/đầu ra của AES-GCM

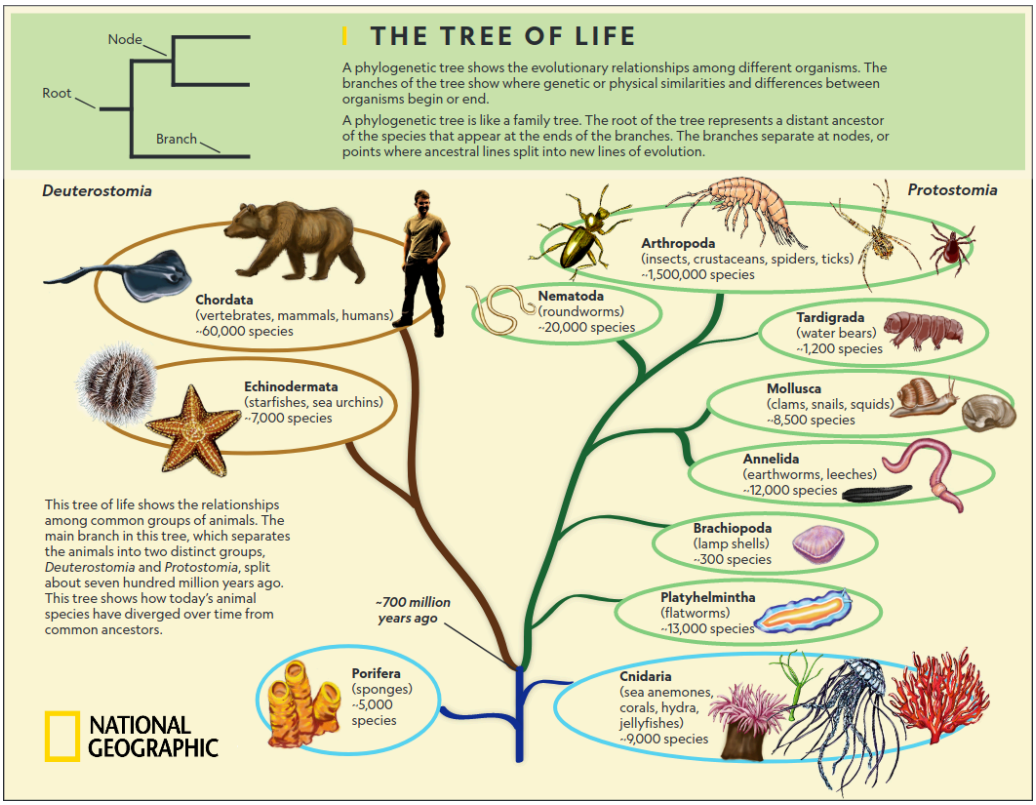
Gợi ý hình: fig_aes_io.png

4.3. Dữ liệu đầu vào từ góc nhìn người dùng

4.3.1. Trải nghiệm nhập liệu và ranh giới dữ liệu nhạy cảm

Người dùng đi qua bộ câu hỏi tính cách với 25 câu trên một lượt làm bài. Các câu trả lời này là dữ liệu nhạy cảm vì có thể suy diễn đặc trưng tâm lý. Ngay khi người dùng hoàn tất bài trả lời, hệ thống chỉ lưu lại các điểm đã tổng hợp theo Big Five, không lưu câu trả lời gốc. Việc này giảm bớt rủi ro rò rỉ dữ liệu thô và hạn chế các điểm nhận dạng gián tiếp.

Hình Hình 4.2 gợi ý bố trí UI và vị trí bước tổng hợp điểm trong luồng ứng dụng.



Hình 4.2 — Luồng UI và vị trí tổng hợp điểm Big Five

Gợi ý hình: fig_ui_quiz_flow.png

4.3.2. Chuyển đổi trên thiết bị

Sau khi tổng hợp, điểm Big Five được chuẩn hóa và chuyển đổi PCA-4 ngay trên thiết bị. Kết quả PCA là dữ liệu đã giảm chiều, đủ cho so khớp nhưng không thay thế được dữ liệu thô. Tuy vậy, PCA vẫn là phép biến đổi tuyến tính có thể suy ngược gần đúng nếu biết tham số. Vì vậy, dữ liệu gốc vẫn cần mã hóa trước khi lưu trữ.

4.4. Mã hóa dữ liệu bằng AES-256-GCM

4.4.1. Đề xuất AES-GCM

Đề xuất của đề tài là sử dụng AES-256-GCM làm cơ chế mã hóa chính cho dữ liệu tính cách và sở thích. Lý do là dữ liệu có kích thước nhỏ, cần mã hóa nhanh và phải giải mã được để hiển thị trên UI. AES-GCM đáp ứng được ba yêu cầu: tốc độ, xác thực và dễ triển khai trên Edge Function. Cơ chế này cũng

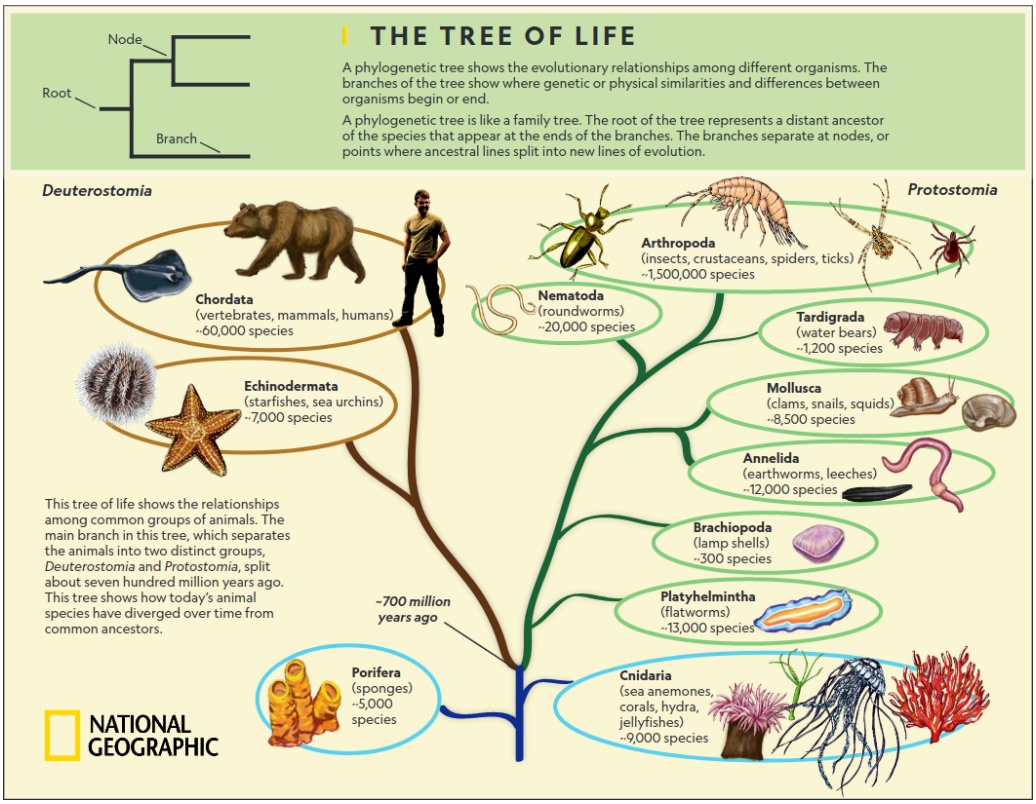
cho phép lưu trữ iv riêng để tái tạo dữ liệu khi người dùng đăng nhập lại. Trong phạm vi đồ án, AES-GCM là lựa chọn thực tế nhất để cân bằng bảo mật và khả năng vận hành.

4.4.2. Lý do chọn AES-GCM

AES-GCM được chọn vì phù hợp với payload nhỏ, tốc độ tốt, và có cơ chế xác thực dữ liệu (integrity) cùng lúc với mã hóa [6]. So với RSA hoặc Bcrypt, AES-GCM ít tốn tài nguyên hơn khi mã hóa dữ liệu JSON ngắn, và dễ tích hợp trong môi trường Edge Function.

4.4.3. Lựa chọn thay thế: RSA

RSA là thuật toán bất đối xứng, thường dùng để trao đổi khóa hoặc ký số [12]. Trong bối cảnh dữ liệu tính cách, RSA không phù hợp để mã hóa payload trực tiếp vì chi phí tính toán lớn và giới hạn kích thước dữ liệu. Nếu dùng RSA cho mỗi lượt cập nhật, hệ thống sẽ tăng thời gian phản hồi và khó mở rộng trên thiết bị di động. Ngoài ra, RSA thường đi kèm cơ chế padding phức tạp, dễ phát sinh lỗi khi triển khai không cẩn thận. Vì vậy RSA được xem là lựa chọn thay thế, không phù hợp làm cơ chế mã hóa chính. Ví dụ, chỉ một payload JSON nhỏ cũng phải qua nhiều bước padding và tách khối, gây chậm trễ rõ rệt khi người dùng cập nhật hồ sơ liên tục.

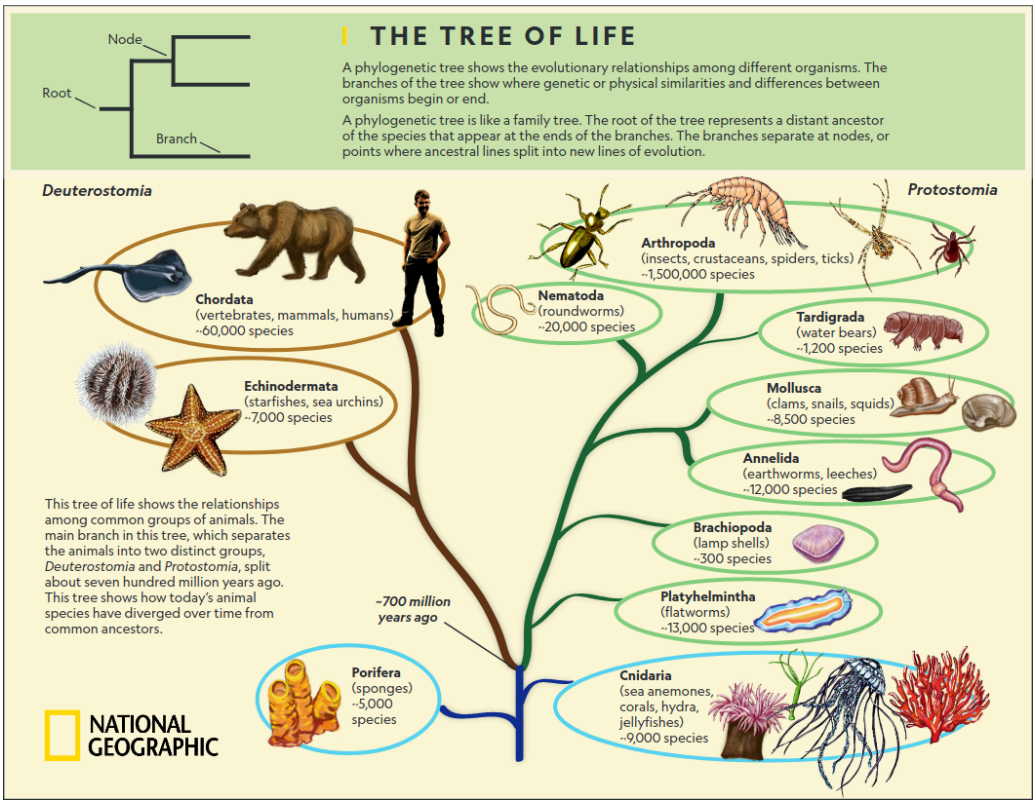


Hình 4.3 — Ví dụ chi phí tính toán khi dùng RSA cho payload nhỏ

Gợi ý hình: fig_rsa_alt.png Gợi ý hình: fig_rsa_alt.png

4.4.4. Lựa chọn thay thế: Bcrypt/Scrypt

Bcrypt và Scrypt là các hàm băm thiết kế cho mật khẩu [13]. Ưu điểm của chúng là làm chậm tấn công brute-force, nhưng điểm yếu là không thể giải mã. Trong hệ thống Twins, người dùng cần xem lại kết quả tính cách và sở thích nên cần giải mã dữ liệu. Nếu dùng bcrypt, hệ thống chỉ có thể so khớp băm, không thể trả dữ liệu gốc cho UI. Điều này đi ngược yêu cầu trải nghiệm và giới hạn chức năng. Vì vậy bcrypt/scrypt không phù hợp. Ví dụ, sở thích "chạy bộ" sau khi băm sẽ không thể khôi phục để hiển thị lại trong ứng dụng.

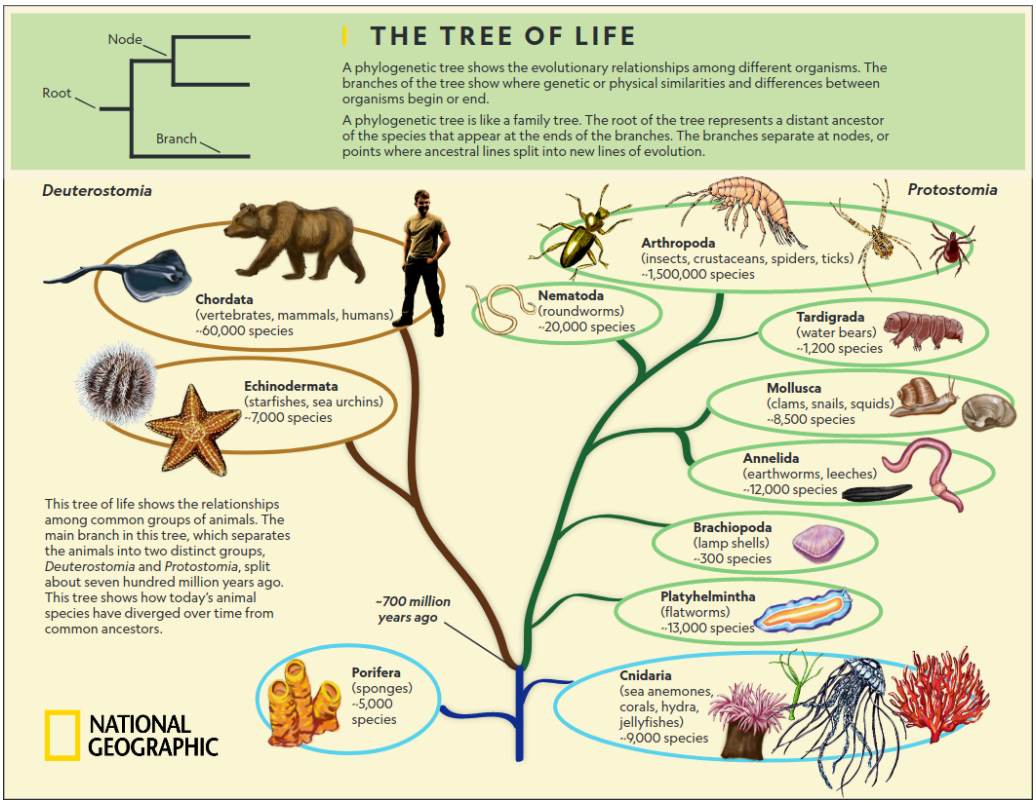


Hình 4.4 — So sánh dữ liệu băm và dữ liệu có thể giải mã

Gợi ý hình: fig_bcrypt_alt.png Gợi ý hình: fig_bcrypt_alt.png

4.4.5. Lựa chọn thay thế: Homomorphic encryption

Homomorphic encryption cho phép tính toán trực tiếp trên dữ liệu đã mã hóa [14]. Đây là hướng rất mạnh về bảo mật, nhưng chi phí tính toán cao và triển khai phức tạp. Với bài toán gợi ý cần phản hồi nhanh, việc dùng homomorphic encryption sẽ làm tăng độ trễ và đòi hỏi hạ tầng đặc biệt. Ngoài ra, mô hình này không cần thiết vì đề tài không tính trực tiếp trên ciphertext mà chỉ lưu trữ và giải mã khi cần. Do đó, homomorphic encryption vượt quá phạm vi thực tế của đề tài. Ví dụ, một phép so khớp cosine trên ciphertext có thể chậm hơn nhiều lần so với dữ liệu plaintext, gây cảm giác lag ở trải nghiệm di động.

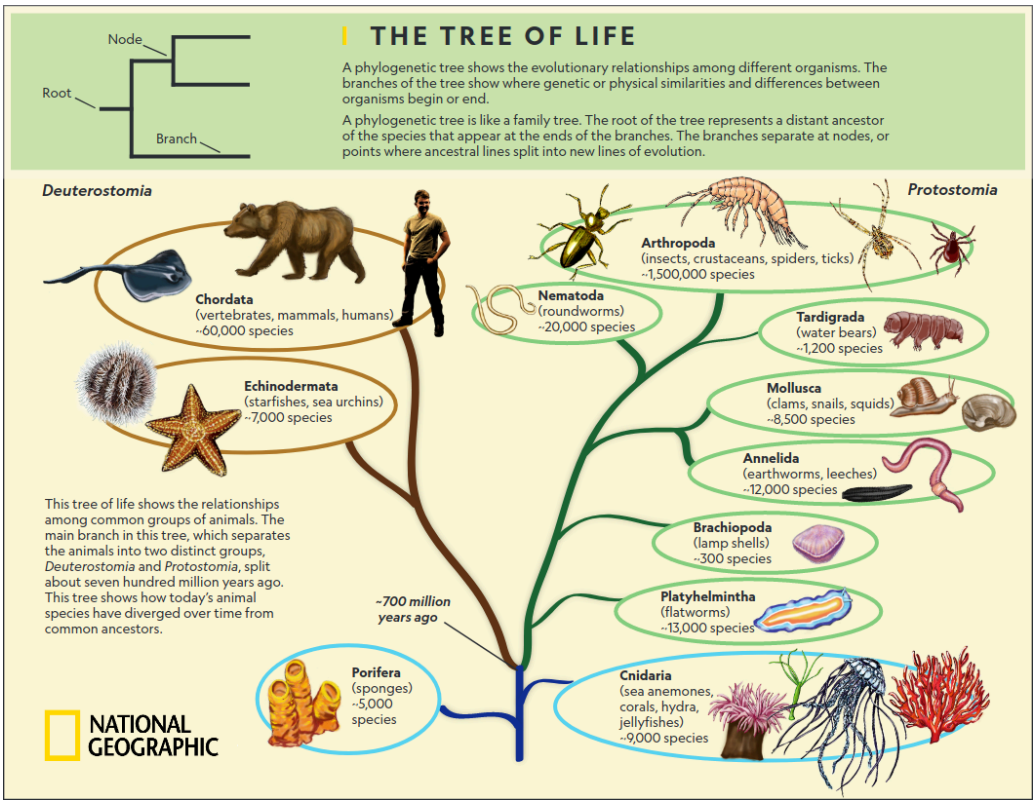


Hình 4.5 — Minh họa độ phức tạp của homomorphic encryption

Gợi ý hình: fig_homomorphic_alt.png Gợi ý hình: fig_homomorphic_alt.png

4.4.6. Lựa chọn thay thế: Differential privacy

Differential privacy tập trung vào ẩn danh khi công bố thống kê [15]. Phương pháp này phù hợp cho dữ liệu tổng hợp, nhưng không giải quyết bài toán lưu trữ và giải mã dữ liệu cá nhân. Nếu chỉ áp dụng differential privacy, người dùng vẫn cần truy cập dữ liệu gốc, dẫn tới vấn đề bảo mật ở cấp độ lưu trữ. Trong hệ thống Twins, yêu cầu là bảo vệ dữ liệu từng người nhưng vẫn cho phép họ xem lại nội dung. Vì vậy, differential privacy được coi như kỹ thuật bổ trợ chứ không thay thế AES-GCM. Ví dụ, nếu cộng nhiễu vào điểm Big Five để bảo vệ thống kê, kết quả gợi ý cá nhân sẽ giảm chính xác và khó giải thích cho người dùng.



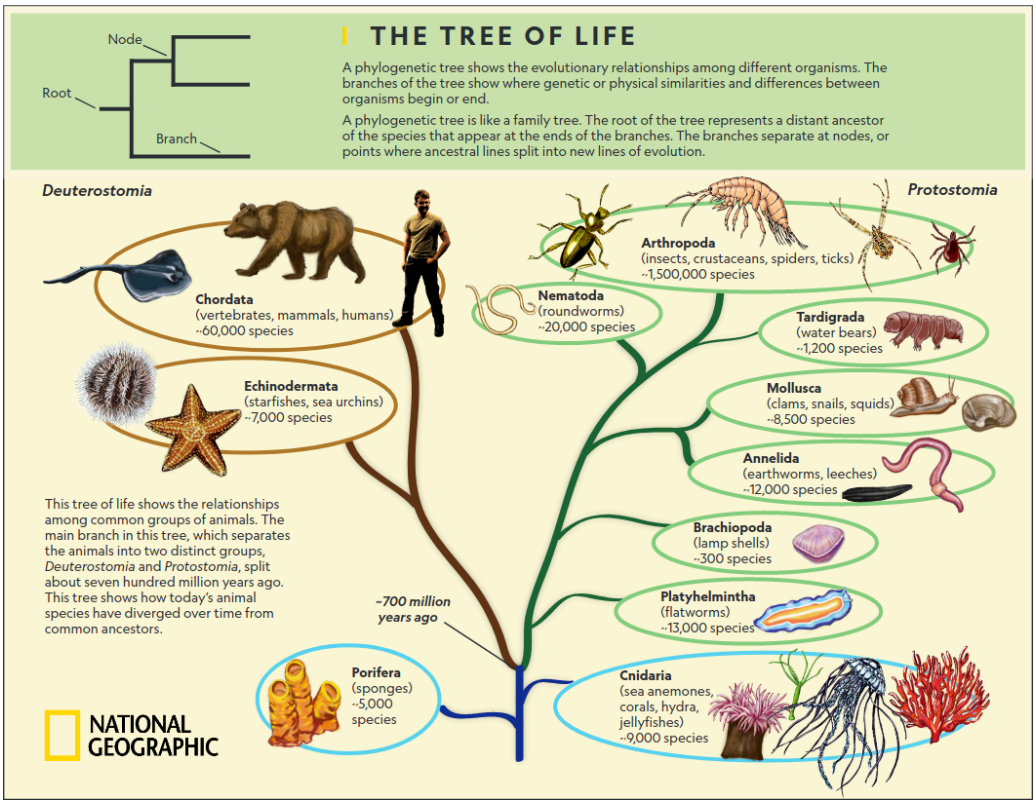
Hình 4.6 — So sánh differential privacy và mã hóa dữ liệu cá nhân

Gợi ý hình: fig_dp_alt.png Gợi ý hình: fig_dp_alt.png

4.4.7. Vai trò của Edge Function và khóa bí mật

Khóa AES chỉ tồn tại ở phía Edge Function. Thiết bị người dùng không giữ khóa, nhằm tránh bị trích xuất từ ứng dụng. Đồng thời, cách làm này cho phép người dùng phục hồi dữ liệu khi đăng nhập lại trên thiết bị khác. Đây là lựa chọn cân bằng giữa bảo mật và khả năng khôi phục.

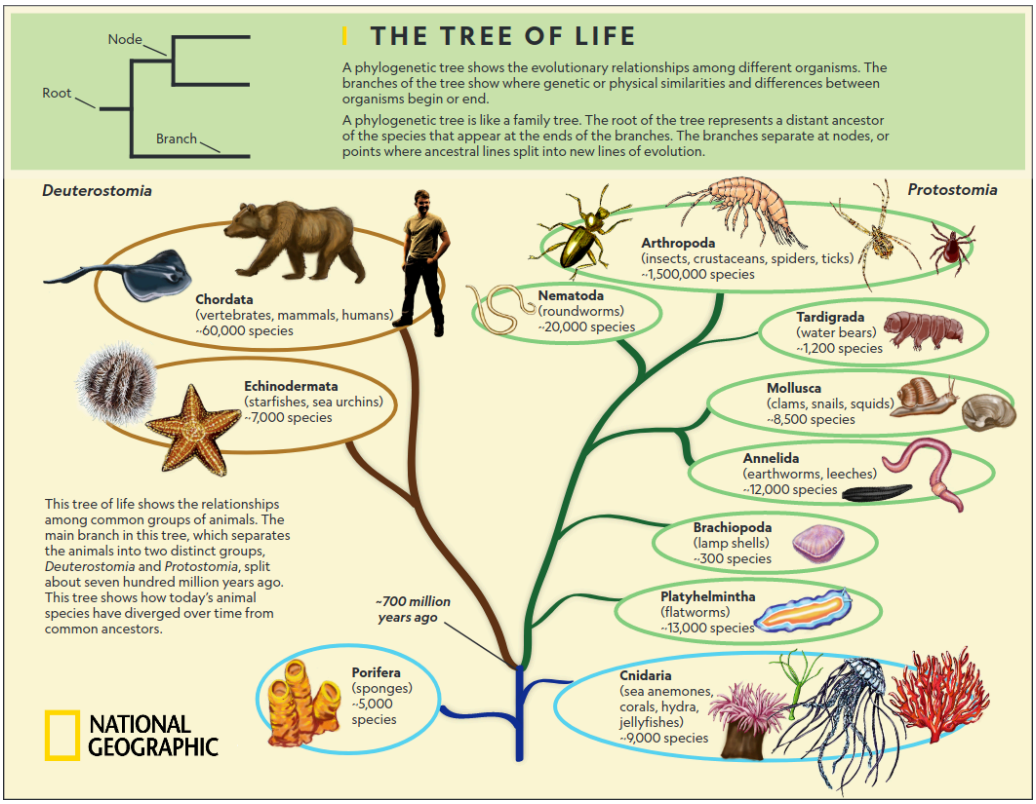
Hình Hình 4.7 mô tả luồng dữ liệu mã hóa và giải mã.



Hình 4.7 — Luồng mã hóa/giải mã dữ liệu Big Five qua Edge Function

Gợi ý hình: fig_crypto_flow.png

Hình Hình 4.8 gợi ý log của Edge Function cho quá trình mã hóa và giải mã.



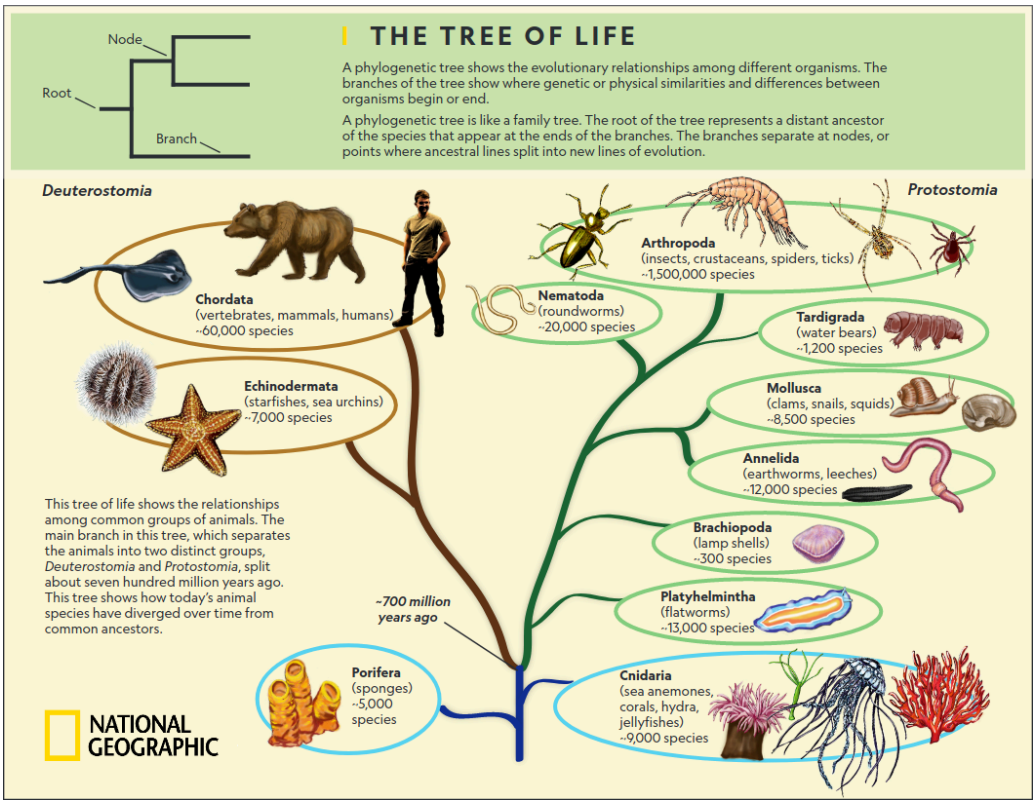
Hình 4.8 — Log Edge Function khi mã hóa và giải mã dữ liệu

Gợi ý hình: fig_edge_logs.png

4.4.8. Lưu trữ và giới hạn truy cập

Cơ sở dữ liệu chỉ lưu ciphertext và iv cho Big Five (b5_cipher, b5_iv). Điều này có nghĩa là quản trị viên cơ sở dữ liệu không thể đọc trực tiếp dữ liệu tính cách dạng thô. Dữ liệu chỉ được giải mã khi người dùng đã xác thực và gọi qua Edge Function. Cách làm này hạn chế nguy cơ mass surveillance từ bảng dữ liệu plaintext, đồng thời vẫn cho phép người dùng xem lại kết quả trong UI.

Hình Hình 4.9 minh họa mẫu dữ liệu ciphertext lưu trong cơ sở dữ liệu.



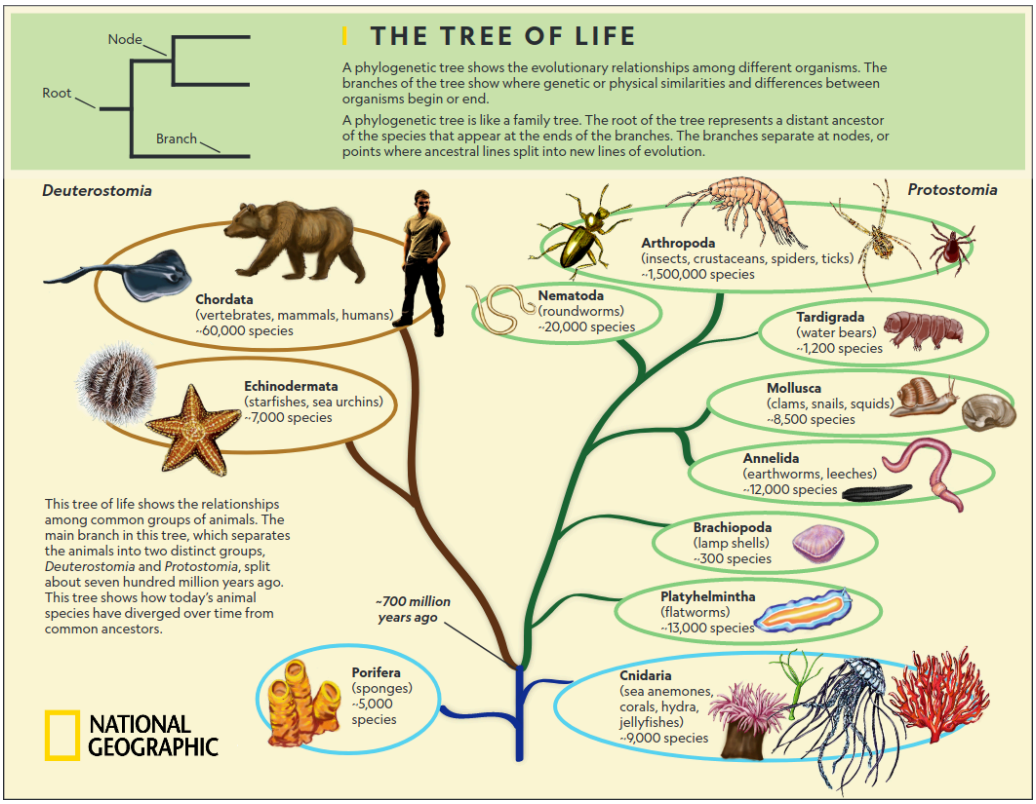
Hình 4.9 — Ví dụ ciphertext của Big Five trong bảng profiles

Gợi ý hình: fig_cipher_sample.png

4.5. Dữ liệu sở thích và mã hóa

Sở thích người dùng được nhập dưới dạng văn bản tự do, sau đó được nhúng thành vector 384 chiều. Dữ liệu này cũng được mã hóa theo cơ chế AES-GCM tương tự Big Five. Do đó, UI có thể hiển thị sở thích sau khi giải mã, nhưng cơ sở dữ liệu không lưu plaintext.

Hình Hình 4.10 mô tả luồng dữ liệu sở thích từ nhập liệu đến lưu trữ.



Hình 4.10 — Luồng mã hóa dữ liệu sở thích và lưu trữ vector embedding

Gợi ý hình: fig_hobby_encrypt.png

Chương 5

Hệ gợi ý và cơ chế xếp hạng

5.1. Mục tiêu của chương

Chương này mô tả cách hệ gợi ý kết hợp ba nguồn tín hiệu: tính cách (PCA), hành vi xã giao (ELO) và sở thích (embedding hobbies). Đồng thời, chương giải thích vì sao từng tín hiệu vẫn cần thiết, ngay cả khi người dùng đã khai báo sở thích.

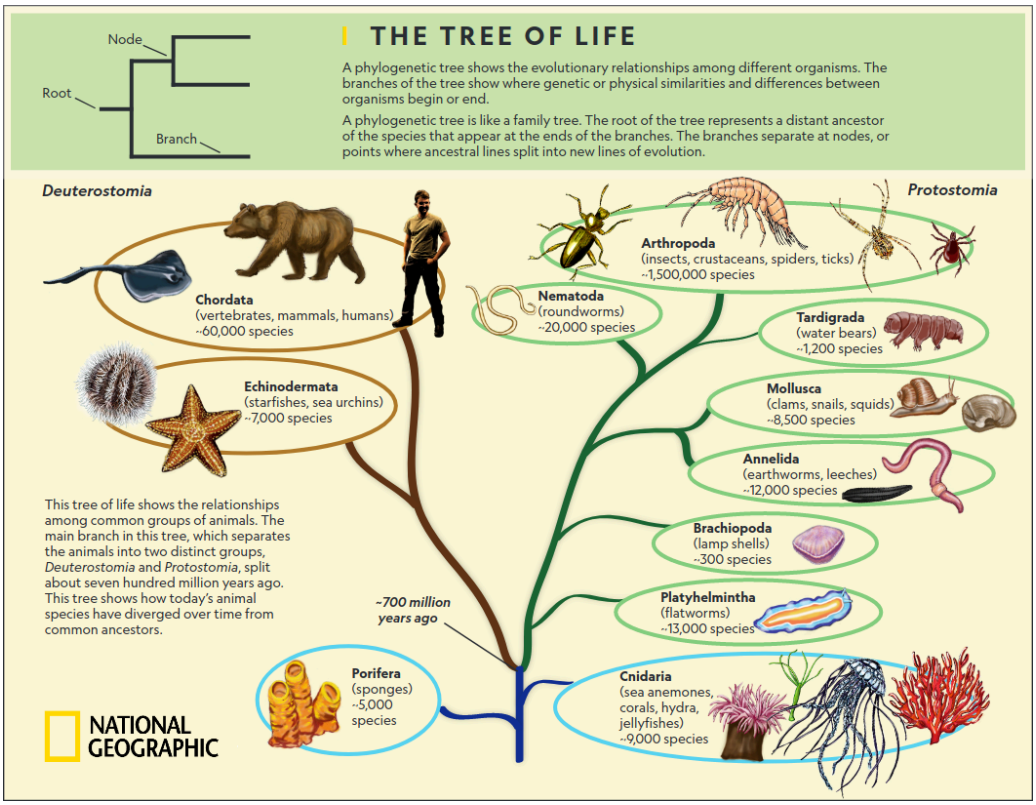
5.2. Vì sao vẫn cần tính cách khi đã có sở thích

Sở thích phản ánh các chủ đề người dùng quan tâm, nhưng không đủ để mô tả mức độ tương hợp về cách suy nghĩ và hành vi. Hai người cùng thích "chụp ảnh" có thể khác nhau rõ rệt về cách giao tiếp, nhịp sống và mức độ ổn định cảm xúc. Với các kết nối dài hạn, các khác biệt này thường quan trọng hơn sở thích bề mặt. Vì vậy, tính cách vẫn là trục chính để bảo đảm kết nối có chiều sâu, còn sở thích đóng vai trò bổ trợ. Hệ gợi ý dùng PCA như trục ổn định, còn sở thích giúp tinh chỉnh trong các trường hợp hòa điểm.

5.3. Đề xuất thuật toán ELO

Trong hệ thống, ELO được dùng như một tín hiệu hành vi ẩn. ELO không nói người dùng "tốt" hơn hay "xấu" hơn, mà phản ánh mức độ xã giao thể hiện qua lượt like/skip. Công thức cập nhật dựa trên kỳ vọng thắng thua gốc của Elo [9], được điều chỉnh để phù hợp với bối cảnh kết nối, nơi lượt like là một tín hiệu hợp tác. Cách cập nhật chi tiết đã được mô tả ở (2.1) và (2.2). Việc giới hạn điểm trong khoảng 800–2000 giúp tránh việc điểm bị trôi quá xa và làm giảm tác dụng phân nhóm hành vi.

Hình Hình 5.1 minh họa trực quan cách ELO phản ánh hành vi xã giao qua các chuỗi like/skip khác nhau.



Hình 5.1 — Ví dụ ELO phản ánh hành vi xã giao qua chuỗi tương tác

Gợi ý hình: fig_elo_behavior.png

5.4. Vai trò của ELO trong hành vi xã giao

Điểm ELO phản ánh mức độ like/skip trong thực tế. Đây là tín hiệu hành vi, không phải kết quả tự khai báo. Khi người dùng thường xuyên skip, điểm ELO giảm và hệ thống ưu tiên gợi ý những người có mức xã giao tương đồng. Điều này giúp giảm sai lệch giữa câu trả lời trắc nghiệm và hành vi thực tế.

ELO trong hệ thống là hệ số ẩn, được cập nhật sau mỗi tương tác và giới hạn trong khoảng 800–2000. Mặc dù cập nhật theo kiểu hợp tác dẫn tới lạm phát điểm, mục tiêu chính là gom nhóm hành vi thay vì xếp hạng cạnh tranh.

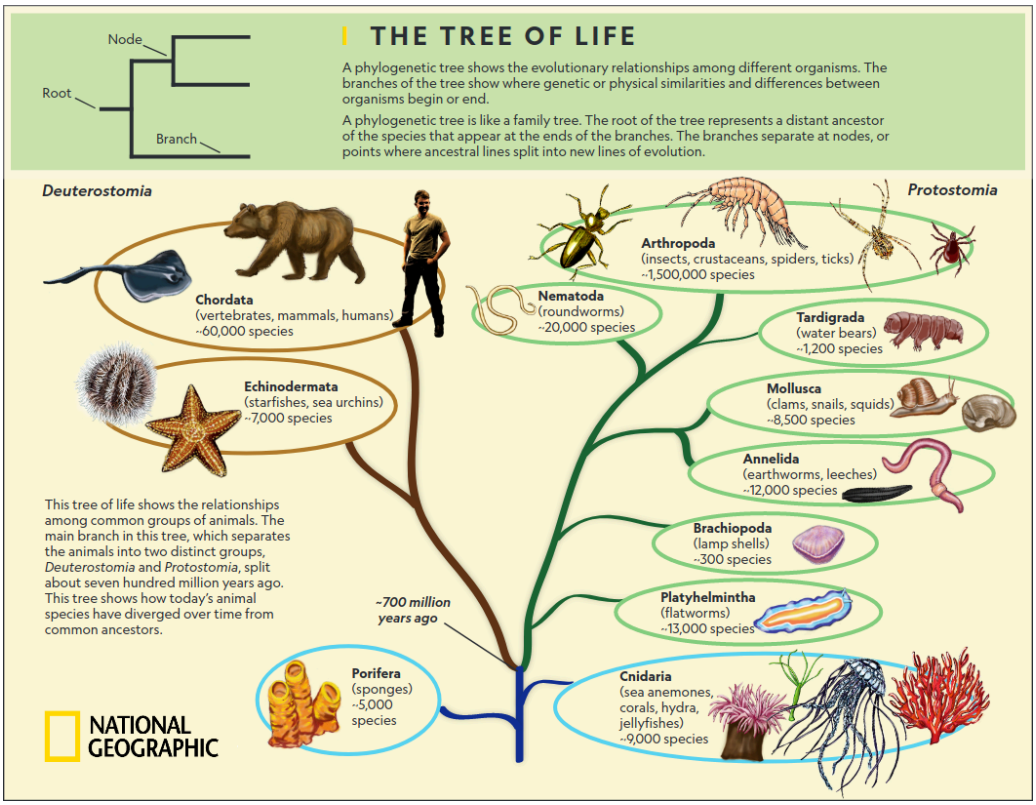
5.5. Ngưỡng sử dụng sở thích

Sở thích chỉ được dùng khi người dùng nhập đủ số lượng tối thiểu (ví dụ 3–5 mục). Điều này tránh việc dùng dữ liệu quá ít dẫn tới nhiễu hoặc thiên lệch do

một sở thích đơn lẻ. Khi đủ ngưỡng, vector embedding được tạo và dùng cosine similarity để tính điểm gần nhau về sở thích. Quy tắc ngưỡng này cũng giúp người dùng mới không bị bất lợi nếu chưa kịp khai báo đầy đủ sở thích.

5.6. Đề xuất mô hình ngữ nghĩa (semantic model)

Đề tài sử dụng mô hình ngữ nghĩa của Jina (semantic model) để chuyển đổi văn bản sở thích thành vector 384 chiều. Lý do chính là khả năng nắm bắt tương đồng ngữ nghĩa thay vì trùng từ khóa, phù hợp với cách người dùng mô tả sở thích bằng nhiều cách khác nhau. Mô hình kiểu sentence embedding cũng ổn định khi so khớp cosine similarity, dễ triển khai và ít tốn tài nguyên hơn so với các mô hình sinh lớn [16]. Hình Hình 5.2 mô tả luồng chuyển đổi từ văn bản sang vector và cách dùng cosine similarity trong gợi ý.



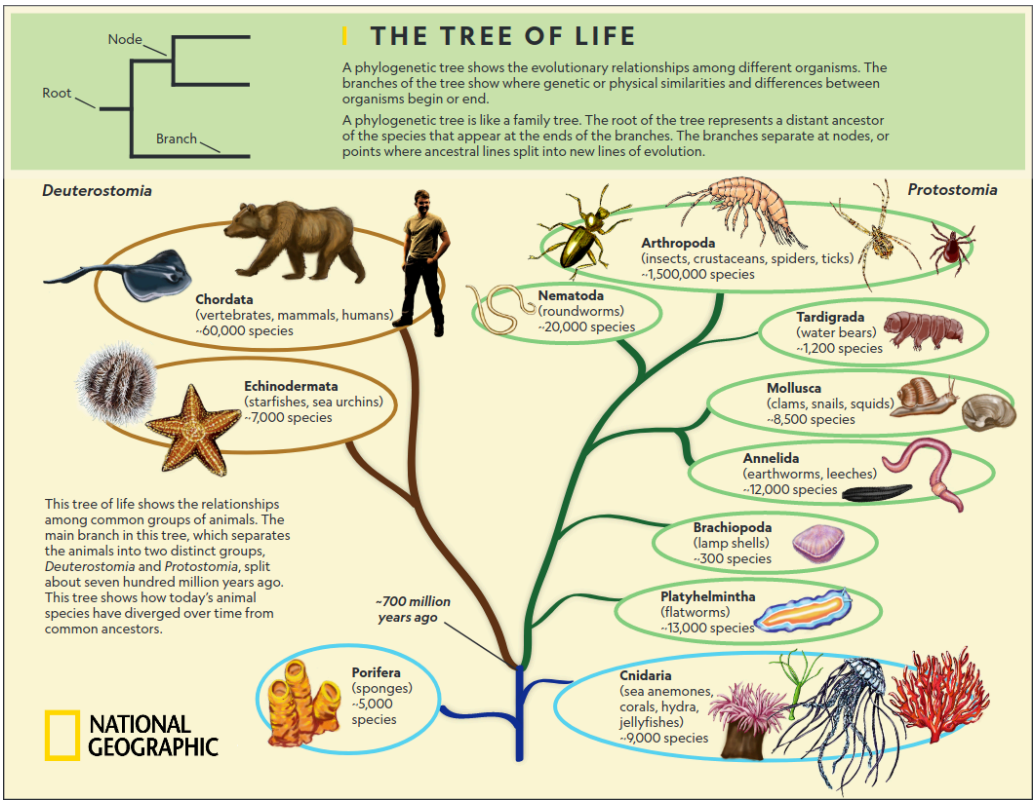
Hình 5.2 — Luồng tạo embedding sở thích bằng semantic model

Gợi ý hình: fig_semantic_model.png

5.6.1. Lựa chọn thay thế: TF-IDF

TF-IDF là cách biểu diễn văn bản theo trọng số từ khóa [8]. Điểm mạnh của TF-IDF là đơn giản, dễ giải thích, và chạy nhanh trên thiết bị. Tuy nhiên,

TF-IDF không hiểu ngữ nghĩa nên khó nhận biết các từ đồng nghĩa như “jogging” và “chạy bộ”. Ngoài ra, TF-IDF tạo vector thưa và kích thước lớn, làm tăng chi phí lưu trữ và so khớp khi số lượng từ vựng tăng. Trong bối cảnh sở thích ngắn và đa dạng, TF-IDF dễ bị nhiễu bởi các từ hiếm. Vì vậy, TF-IDF được coi là lựa chọn thay thế tham khảo chứ không phù hợp làm lõi gợi ý.

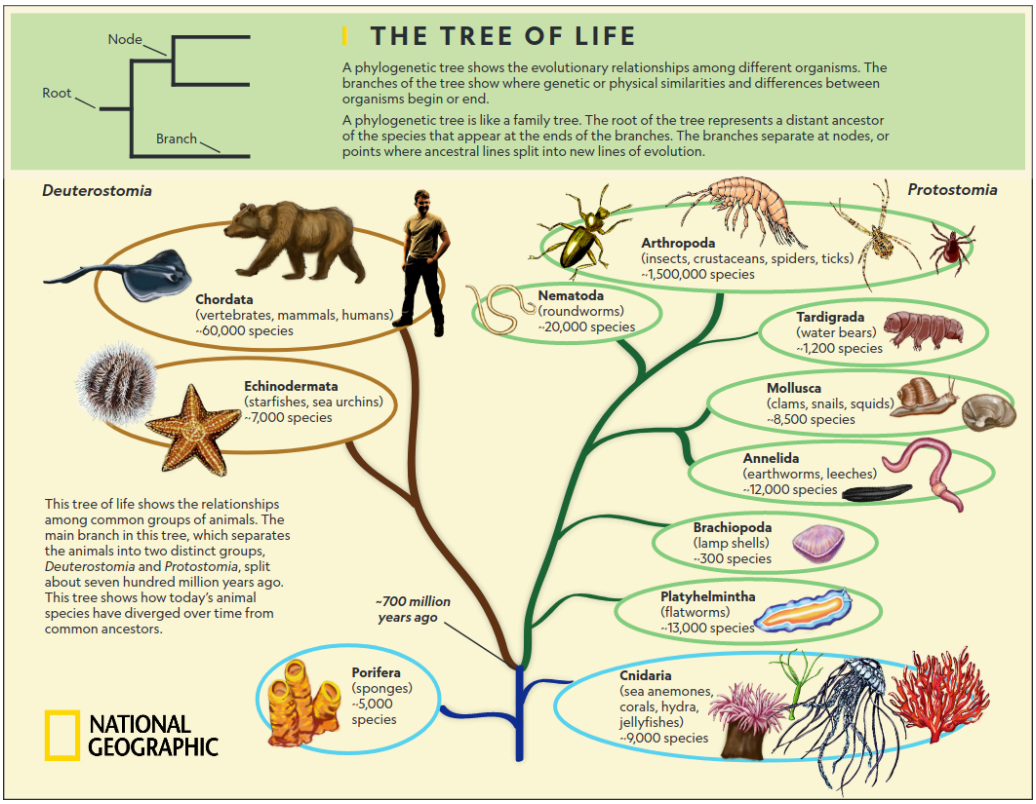


Hình 5.3 — Ví dụ hạn chế của TF-IDF khi so khớp sở thích

Gợi ý hình: fig_tfidf_alt.png

5.6.2. Lựa chọn thay thế: Word2Vec

Word2Vec tạo vector cho từng từ dựa trên ngữ cảnh [17]. Cách này nắm bắt được một phần quan hệ ngữ nghĩa, nhưng vẫn gặp khó khi chuyển sang mức câu hoặc cụm sở thích ngắn. Người dùng thường nhập cụm như “đi phượt cuối tuần” hoặc “nấu ăn healthy”, trong khi Word2Vec cần thêm bước gộp nhiều vector để đại diện cho cả cụm. Việc gộp thủ công làm mất sắc thái và không ổn định giữa các mẫu khác nhau. Do đó, các mô hình sentence embedding được ưu tiên vì xử lý trực tiếp cụm sở thích, ổn định hơn trong so khớp.



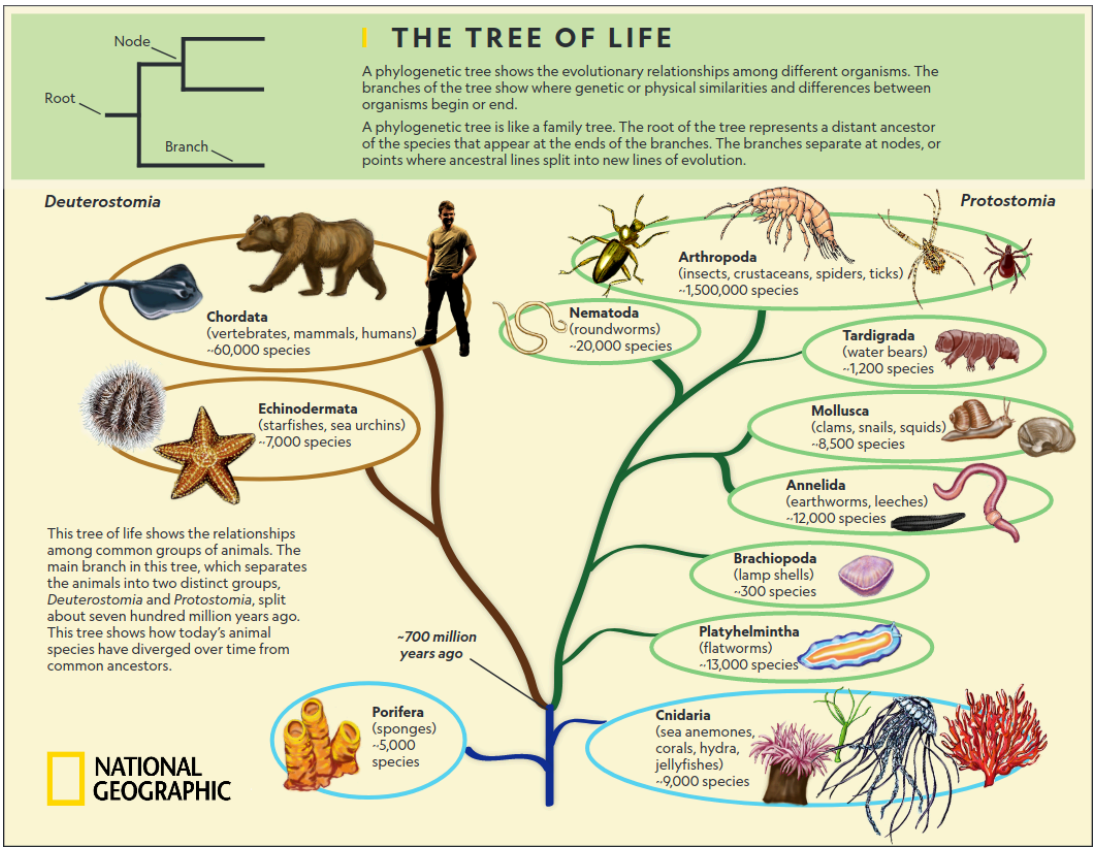
Hình 5.4 — So sánh Word2Vec và sentence embedding trên cụm sở thích

Gợi ý hình: fig_word2vec_alt.png

5.7. Công thức xếp hạng tổng hợp

Hệ thống tính điểm theo các trọng số đã nêu ở Chương 2. Về bản chất, PCA là trục chính, ELO là trục hành vi, và hobbies là trục ngữ nghĩa. Hình Hình 5.5 mô tả cây quyết định tính điểm và cách nhánh ELO/hobbies được bật tắt.

Việc đặt PCA làm trục chính giúp kết quả ổn định hơn theo thời gian, vì tính cách thay đổi chậm và ít bị ảnh hưởng bởi các biến động ngắn hạn. ELO chỉ đóng vai trò điều chỉnh, tránh trường hợp hai người có tính cách gần nhau nhưng hành vi xã giao quá khác biệt. Hobbies được dùng như một tín hiệu làm mượt, giúp hệ gợi ý nhận ra các chủ đề tương đồng mà tính cách không nắm bắt được. Cấu trúc này giảm rủi ro hệ thống chỉ dựa vào một nguồn dữ liệu duy nhất, vốn dễ gây thiên lệch hoặc thiếu đa dạng.



Hình 5.5 — Cây quyết định tính điểm gợi ý

Gợi ý hình: fig_rank_flow.png

5.8. Ví dụ minh họa xếp hạng

Xét người dùng A đang xem gợi ý, với ba ứng viên B và C. Giả sử:

- PCA similarity: A-B = 0.90, A-C = 0.90 (hòa nhau).
- Hobbies similarity: A-B = 0.85, A-C = 0.55.
- ELO proximity: A-B = 0.70, A-C = 1.00.

Trong cấu hình bật cả ELO và hobbies, điểm cuối của B sẽ tăng nhờ hobbies, còn C tăng nhờ ELO. Nếu trọng số hobbies lớn hơn phần chênh lệch ELO, B sẽ đứng trước. Nếu ngược lại, C sẽ đứng trước. Ví dụ này cho thấy các nguồn tín hiệu có thể phá vỡ thế hòa PCA theo các hướng khác nhau.

5.9. Bảo vệ dữ liệu sở thích và quyền riêng tư

Mặc dù UI có thể hiển thị sở thích đã giải mã, cơ sở dữ liệu không lưu plaintext. Điều này tránh việc quản trị viên có thể quét hàng loạt sở thích từ

bảng dữ liệu. Người dùng chỉ thấy sở thích khi đã được xác thực và giải mã thông qua Edge Function.

Ngoài ra, việc lưu ciphertext giúp giảm rủi ro lộ dữ liệu ở cấp độ hệ quản trị. Người dùng vẫn nhìn thấy sở thích trên UI vì dữ liệu được giải mã theo phiên đăng nhập hợp lệ, nhưng cơ sở dữ liệu không có điểm tập trung plaintext để khai thác hàng loạt. Đây là điểm khác biệt quan trọng so với cách lưu trữ sở thích truyền thống trong nhiều ứng dụng mạng xã hội.

Tài liệu tham khảo

[1] Tupes EC, Christal RE. Recurrent personality factors based on trait ratings. Journal of Personality 1961;30:563–80.

[2] John OP, Srivastava S. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. Handbook of personality: Theory and research 1999;2:102–38.

[3] Youyou W, Kosinski M, Stillwell D. Computer-based personality judgments are more accurate than those made by humans. Proceedings of the National Academy of Sciences 2015;112:1036–40.

[4] Meng KS, Leung L. Factors influencing TikTok engagement behaviors in China: An examination of gratifications sought, narcissism, and the Big Five personality traits. Telecommunications Policy 2021;45:102172.

[5] Automoto. Big five trait scores for 307,313 people from many different countries 2023.

[6] NIST. Galois/Counter Mode of Operation (GCM). NIST Special Publication 800-38D 2007.

[7] Goldberg LR. The development of markers for the Big-Five factor structure. Psychological Assessment 1992;4:26–42.

[8] Manning CD, Raghavan P, Sch"utze H. Introduction to Information Retrieval. Cambridge University Press; 2008.

[9] Elo AE. The Rating of Chessplayers, Past and Present. Arco Publishing; 1978.

[10] Ashton MC, Lee K. Empirical, theoretical, and practical advantages of the HEXACO model of personality structure. Personality and Social Psychology Review 2007;11:150–66.

[11] Jolliffe IT. Principal Component Analysis. Springer; 2002.

[12] Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 1978;21:120–6.

[13] Provos N, Mazieres D. A Future-Adaptable Password Scheme. trong:. Proceedings of the 1999 USENIX Annual Technical Conference, 1999, tr 81–92.

[14] Gentry C. Fully homomorphic encryption using ideal lattices. trong:. Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 2009, tr 169–78.

[15] Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. trong:. Theory of Cryptography Conference, 2006, tr 265–84.

[16] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. trong:. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019, tr 3982–92.

[17] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. trong:. International Conference on Learning Representations, 2013.