10 Đề tài nghiên cứu mới về AI, Machine Learning và Data Mining

1. Phát hiện thông tin sai lệch (Fake News) bằng Deep Learning

Hướng dẫn tổng quát:

1. Thu thập dữ liệu từ các nguồn tin tức có nhãn (thật/giả)
2. Tiền xử lý dữ liệu văn bản (làm sạch, chuẩn hóa)
3. Xây dựng mô hình transformer (như BERT hoặc RoBERTa) hoặc CNN-LSTM
4. Đánh giá và so sánh hiệu suất giữa các mô hình
5. Triển khai ứng dụng thực tế (như extension trình duyệt)

Tài liệu tham khảo:

- GitHub Kaggle Fake News Detection: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
- Bài báo "FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media": https://arxiv.org/abs/1809.01286

2. Phân tích cảm xúc đa phương tiện (Multimodal Sentiment Analysis)

Hướng dẫn tổng quát:

1. Thu thập dữ liệu đa phương tiện (văn bản, hình ảnh, âm thanh)
2. Trích xuất đặc trưng từ từng loại dữ liệu (NLP cho văn bản, CNN cho hình ảnh)
3. Thiết kế kiến trúc fusion kết hợp các đặc trưng
4. Huấn luyện mô hình end-to-end hoặc theo từng giai đoạn
5. Phân tích kết quả và đánh giá độ chính xác

Tài liệu tham khảo:

- Bộ dữ liệu CMU-MOSEI: http://multicomp.cs.cmu.edu/resources/cmu-mosei-dataset/

- Bài báo "Multimodal Sentiment Analysis: A Systematic Review of History, Datasets, Multimodal Fusion Methods, Applications, Challenges and Future Directions": https://arxiv.org/abs/2212.10250

3. Học liên tục (Continual Learning) cho các mô hình AI

Hướng dẫn tổng quát:

1. Nghiên cứu các phương pháp học liên tục hiện có (replay methods, regularization)
2. Thiết kế môi trường thử nghiệm với dữ liệu thay đổi theo thời gian
3. Triển khai mô hình cơ sở và các phương pháp học liên tục
4. So sánh với mô hình truyền thống (học lại từ đầu)
5. Phân tích quên thảm họa (catastrophic forgetting) và hiệu quả của các giải pháp

Tài liệu tham khảo:

- Continual-Learning-Benchmark: https://github.com/GT-RIPL/Continual-Learning-Benchmark
- Bài báo "Continual Learning in Neural Networks": https://arxiv.org/abs/1910.02718

4. Khai thác dữ liệu Y tế để dự đoán bệnh mãn tính

Hướng dẫn tổng quát:

1. Thu thập và làm sạch dữ liệu y tế (với sự chú ý đặc biệt về bảo mật và đạo đức)
2. Phân tích đặc trưng và lựa chọn đặc trưng quan trọng
3. Áp dụng các thuật toán ML khác nhau (Random Forest, XGBoost, Deep Learning)
4. Xây dựng hệ thống hỗ trợ quyết định lâm sàng
5. Đánh giá mô hình với các chỉ số y tế cụ thể (độ nhạy, độ đặc hiệu)

Tài liệu tham khảo:

- MIMIC-IV (Medical Information Mart for Intensive Care): https://physionet.org/content/mimiciv/
- Bài báo "Machine Learning for Clinical Predictive Analytics": https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6287645/

5. Hệ thống khuyến nghị dựa trên đồ thị (Graph-based Recommendation Systems)

Hướng dẫn tổng quát:

1. Xây dựng biểu diễn đồ thị từ dữ liệu người dùng-sản phẩm
2. Nghiên cứu các mô hình mạng nơ-ron đồ thị (GNN, GraphSAGE, GAT)

3. Thiết kế các phương pháp biểu diễn nút (node embeddings)
4. Huấn luyện và đánh giá hệ thống khuyến nghị
5. So sánh với các phương pháp truyền thống (collaborative filtering)

Tài liệu tham khảo:

- Thư viện PyTorch Geometric: https://github.com/pyg-team/pytorch_geometric
- Bài báo "Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions": https://arxiv.org/abs/2109.12843

Đề tài 6: Chatbot hỗ trợ khách hàng bằng AI

Mô tả: Xây dựng một chatbot sử dụng kỹ thuật **Xử lý Ngôn ngữ Tự nhiên (NLP)** để tự động tương tác và giải đáp thắc mắc cho người dùng. Chatbot là chương trình máy tính mô phỏng hội thoại của con người, có thể hoạt động 24/7 để hỗ trợ khách hàng mà không cần con người trực tiếp (Customer service chatbots: A buyer's guide for 2025). Đề tài này hướng đến việc tạo một chatbot đơn giản (ví dụ: chatbot trả lời câu hỏi thường gặp cho một website dịch vụ). Chatbot hiện đại thường gồm các thành phần như phân loại **ý định** người dùng, nhận dạng **thực thể** trong câu nói và sinh phản hồi thích hợp (Tổng quan về RASA Chatbot). Sinh viên sẽ học cách thu thập dữ liệu hội thoại, huấn luyện mô hình ngôn ngữ cơ bản và đánh giá chất lượng tương tác của chatbot.

Các bước thực hiện:

1. **Xác định mục tiêu chatbot:** Lựa chọn loại chatbot (hỗ trợ khách hàng, trả lời FAQ, trợ lý ảo, v.v.) và ngôn ngữ sử dụng. Xác định rõ phạm vi kiến thức hoặc dịch vụ mà bot sẽ cung cấp.
2. **Thu thập dữ liệu hội thoại:** Chuẩn bị bộ dữ liệu gồm các cặp **câu hỏi – câu trả lời** hoặc kịch bản hội thoại trong lĩnh vực đã chọn. Dữ liệu có thể lấy từ FAQ của công ty, nhật ký chat hỗ trợ khách hàng, hoặc tự tổng hợp tình huống giả định.
3. **Tiền xử lý dữ liệu:** Làm sạch và định dạng dữ liệu thoại. Với mô hình NLP, cần chuẩn hóa văn bản (ví dụ: chuyển về chữ thường, bỏ dấu câu thừa), gắn nhãn **ý định (intent)** cho câu hỏi và xác định các **thực thể (entity)** quan trọng (nếu cần). Có thể sử dụng các thư viện NLP cho tiếng Việt (như Underthesea) để tách từ và nhận diện thực thể.
4. **Xây dựng mô hình chatbot:** Lựa chọn phương pháp triển khai. Cách đơn giản là dùng các **framework có sẵn** như RASA để thiết kế chatbot theo dạng hướng kịch bản và máy học (Tổng quan về RASA Chatbot). Hoặc sinh viên có thể tự xây dựng mô hình **seq2seq** hoặc **Transformer** nhỏ để sinh câu trả lời từ câu hỏi (nâng cao hơn). Huấn luyện mô hình hoặc cấu hình bot với bộ dữ liệu đã chuẩn bị.
5. **Kiểm thử và đánh giá:** Thử nghiệm chatbot với các câu hỏi mẫu. Đánh giá độ chính xác của câu trả lời hoặc mức độ hài lòng của người dùng thử. Có thể phân loại phản hồi của bot thành đúng/chưa đúng ý và thống kê tỷ lệ. Nếu bot dùng mô hình học máy, có thể đo các chỉ số như độ chính xác intent, độ chính xác phản hồi dựa trên tập kiểm thử.
6. **Cải thiện và triển khai:** Dựa trên kết quả kiểm thử, điều chỉnh chatbot. Bổ sung dữ liệu huấn luyện cho những trường hợp bot trả lời chưa tốt. Tinh chỉnh các **luật hội thoại** hoặc

tham số mô hình. Sau khi hài lòng, triển khai chatbot lên một nền tảng giao tiếp (website, Facebook Messenger, Zalo, v.v.) để người dùng tương tác thực tế và thu thập phản hồi.

Tài liệu tham khảo:

- **Viblo:** *"Tổng quan về RASA Chatbot"* – giới thiệu kiến trúc và các thành phần của chatbot sử dụng RASA, bao gồm phân loại ý định và nhận dạng thực thể (Tổng quan về RASA Chatbot) (tiếng Việt).
- **IBM:** *"What is a chatbot?"* – định nghĩa chatbot và xu hướng chatbot dùng mô hình ngôn ngữ hiện đại (What Is a Chatbot? | IBM) (tiếng Anh).

Đề tài 7: Hệ thống gợi ý sản phẩm cá nhân hóa

Mô tả: Phát triển một **hệ thống gợi ý (recommender system)** nhằm đề xuất sản phẩm hoặc nội dung phù hợp với sở thích của người dùng. Hệ thống gợi ý là công cụ hỗ trợ ra quyết định, cung cấp các đề xuất **cá nhân hóa** dựa trên hành vi và ưu tiên của người dùng (Giới thiệu về hệ thống gợi ý (Recommender systems hoặc Recommendation systems)). Ứng dụng phổ biến bao gồm: gợi ý phim cho người xem (Netflix), gợi ý sản phẩm cho khách hàng mua sắm (Amazon, Shopee) – thực tế, có thống kê cho thấy khoảng **35% doanh thu** của Amazon đến từ hệ thống gợi ý sản phẩm của họ. Với đề tài này, sinh viên có thể xây dựng hệ thống gợi ý phim hoặc sản phẩm đơn giản bằng cách áp dụng thuật toán lọc dữ liệu và học máy trên tập dữ liệu người dùng – sản phẩm cho trước.

Các bước thực hiện:

1. **Thu thập/chọn dữ liệu:** Lựa chọn một tập dữ liệu có chứa **tương tác của người dùng** với sản phẩm. Ví dụ: tập dữ liệu MovieLens (đánh giá phim của nhiều người dùng) hoặc một tập dữ liệu bán lẻ (danh sách sản phẩm mà khách hàng đã mua kèm đánh giá/xếp hạng). Dữ liệu thường ở dạng danh sách (user, item, rating) hoặc ma trận người dùng – sản phẩm.
2. **Tiền xử lý dữ liệu:** Làm sạch dữ liệu: loại bỏ những người dùng hoặc sản phẩm có quá ít tương tác (để tập trung vào các thực thể phổ biến). Chuyển đổi dữ liệu thành cấu trúc phù hợp – chẳng hạn tạo **ma trận utility** với hàng là người dùng, cột là sản phẩm, và ô là điểm đánh giá (có thể để trống nếu người dùng chưa đánh giá sản phẩm đó).
3. **Lựa chọn thuật toán gợi ý:** Có hai phương pháp chính: (a) **Lọc cộng tác** (Collaborative Filtering) dựa trên sự tương đồng giữa người dùng hoặc sản phẩm. Ví dụ: gợi ý dựa trên những người dùng có sở thích tương tự, hoặc dựa trên sản phẩm tương tự mà người dùng đã thích. (b) **Lọc nội dung** (Content-based Filtering) dựa trên đặc điểm của sản phẩm và hồ sơ sở thích của người dùng. Cũng có thể kết hợp hai phương pháp (hệ thống lai ghép).
4. **Xây dựng mô hình:** Triển khai thuật toán đã chọn. Với lọc cộng tác, có thể tính **độ tương đồng** giữa các vector người dùng (hoặc sản phẩm) và dùng công thức dự đoán điểm cho sản phẩm chưa được người dùng tương tác. Một cách phổ biến là sử dụng **phân rã ma trận** (matrix factorization) – tìm hai ma trận ẩn U và V để xấp xỉ ma trận utility, từ đó dự đoán mức độ ưa thích. Nhiều thư viện (Surprise, TensorRec) có thể hỗ trợ quá trình này.

5. **Đánh giá hệ thống gợi ý:** Sử dụng kỹ thuật **đánh giá chéo** trên tập dữ liệu: chia dữ liệu thành tập huấn luyện và kiểm thử. Đối với bài toán dự đoán rating, sử dụng RMSE hoặc MAE để đo sai số dự đoán. Đối với bài toán đề xuất top-N, sử dụng các chỉ số như **Precision@N, Recall@N** hoặc **Mean Average Precision** để đánh giá chất lượng danh sách gợi ý. Ngoài ra, kiểm tra thủ công một vài gợi ý xem có hợp lý không.
6. **Xây dựng ứng dụng mẫu:** Hiển thị kết quả gợi ý theo cách trực quan. Ví dụ: thiết kế một giao diện console hoặc web đơn giản, nhập vào một user và hiển thị danh sách phim/sản phẩm được đề xuất cho user đó. Điều này giúp kiểm tra hệ thống hoạt động đúng kỳ vọng.
7. **Cải thiện:** Thử điều chỉnh tham số mô hình (ví dụ số yếu tố ẩn trong phân rã ma trận, hoặc ngưỡng tương đồng). Bổ sung thông tin **ngữ cảnh** hoặc **thuộc tính** sản phẩm để cải thiện gợi ý (ví dụ kết hợp cả thể loại phim, giá sản phẩm). Nếu có thời gian, so sánh giữa các phương pháp gợi ý khác nhau và phân tích ưu nhược điểm (gợi ý dựa trên mô hình học sâu, mạng neural đa tầng cho recommender,...).

Tài liệu tham khảo:

- **Viblo:** *"Giới thiệu về hệ thống gợi ý"* – định nghĩa và phân loại các phương pháp gợi ý (lọc nội dung, lọc cộng tác, lai ghép) kèm ví dụ ứng dụng thực tế (Giới thiệu về hệ thống gợi ý (Recommender systems hoặc Recommendation systems)) (tiếng Việt).
- **Nghiên cứu tại UCSD:** *"Amazon Recommender System"* – phân tích hệ thống gợi ý của Amazon, có thống kê 35% doanh số Amazon đến từ gợi ý (tiếng Anh).

Đề tài 8: Phân tích cảm xúc từ dữ liệu mạng xã hội

Mô tả: Thực hiện **phân tích cảm xúc (sentiment analysis)** trên dữ liệu văn bản từ mạng xã hội (ví dụ: bài đăng Twitter, bình luận Facebook) để xác định xu hướng tình cảm của người viết (tích cực, tiêu cực hoặc trung tính). Phân tích cảm xúc, đôi khi gọi là *khai thác ý kiến (opinion mining)*, là quá trình sử dụng kỹ thuật NLP để phát hiện thái độ hoặc cảm xúc mà văn bản thể hiện (SCV | Con đường cày cuốc). Đây là một bài toán phổ biến trong xử lý ngôn ngữ tự nhiên và học máy, với nhiều ứng dụng thực tiễn như lắng nghe ý kiến khách hàng về sản phẩm, theo dõi phản hồi của cộng đồng về một sự kiện, hoặc hỗ trợ hệ thống **chatbot** hiểu cảm xúc người dùng. Về bản chất kỹ thuật, bài toán này là một dạng **phân loại văn bản**: đầu vào là câu hoặc đoạn văn, đầu ra là nhãn cảm xúc tương ứng (SCV | Con đường cày cuốc). Đề tài phù hợp cho sinh viên là xây dựng một mô hình phân tích cảm xúc cơ bản cho tiếng Việt (hoặc song ngữ) và áp dụng nó trên tập dữ liệu mạng xã hội thu thập được, sau đó rút ra những phân tích thú vị.

Các bước thực hiện:

1. **Thu thập dữ liệu:** Chọn nguồn dữ liệu mạng xã hội và chủ đề sẽ phân tích. Ví dụ: thu thập các tweet có hashtag cụ thể (thông qua API của Twitter), hoặc thu thập bình luận về một sản phẩm/dịch vụ trên Facebook, YouTube. Dữ liệu cần được gắn nhãn cảm xúc để huấn luyện mô hình – sinh viên có thể sử dụng các **bộ dữ liệu có sẵn** (vd: SA-VD hoặc UIT-VSMEC cho tiếng Việt) hoặc tự gán nhãn thủ công một tập nhỏ (phân loại thủ công thành tích cực/tiêu cực).

2. **Tiền xử lý văn bản:** Làm sạch dữ liệu thô. Loại bỏ các yếu tố gây nhiễu như URL, thẻ hashtag, mention (@user), emoji hoặc ký tự đặc biệt (tuỳ mức độ cần thiết; cũng có thể chuyển emoji thành từ mô tả nếu muốn giữ lại cảm xúc từ emoji). Chuẩn hóa tiếng Việt: chuyển chữ viết tắt, tiếng lóng về dạng phổ thông; có thể bỏ dấu tiếng Việt hoặc giữ nguyên tuỳ phương pháp. Tách câu thành danh sách từ (word segmentation) bằng các công cụ NLP cho tiếng Việt. Loại bỏ **từ dừng** (những từ không mang nhiều ý nghĩa như "là", "và", "nhưng").
3. **Biểu diễn đặc trưng:** Chọn cách biểu diễn văn bản dưới dạng vector số. Cách đơn giản: dùng **Bag of Words** hoặc **TF-IDF** để biến mỗi câu thành vector độ dài bằng kích thước từ vựng. Cách hiện đại hơn: sử dụng **word embedding** (ví dụ pre-trained Word2Vec hoặc FastText tiếng Việt) để nhận vector cho từng từ, rồi ghép thành vector câu (thông qua trung bình cộng hoặc mô hình RNN). Thậm chí có thể dùng **BERT đa ngôn ngữ** để thu được embedding của cả câu văn.
4. **Xây dựng mô hình phân loại cảm xúc:** Lựa chọn mô hình ML thích hợp. Với dữ liệu vừa phải, có thể thử các mô hình truyền thống như **Naive Bayes** hoặc **SVM** – những mô hình này thường hiệu quả với đặc trưng TF-IDF. Nếu có kiến thức về deep learning, có thể xây dựng một mạng **LSTM** hoặc **CNN** đơn giản cho phân loại văn bản, hoặc fine-tune mô hình **BERT** để đạt kết quả tốt hơn (Cài đặt mô hình phân loại cảm xúc tiếng Việt). Tách dữ liệu thành tập huấn luyện và kiểm thử, huấn luyện mô hình trên tập huấn luyện.
5. **Đánh giá mô hình:** Áp dụng mô hình lên tập kiểm thử (hoặc dùng cross-validation) và tính các chỉ số đánh giá. Các chỉ số quan trọng gồm **Accuracy** (độ chính xác tổng quát), **Precision, Recall, F1-score** cho từng lớp (tích cực/tiêu cực). Đặc biệt chú ý những mẫu mà mô hình dự đoán sai để phân tích nguyên nhân (câu chứa từ lóng, câu mỉa mai có thể gây khó khăn). Nếu có nhãn trung tính, có thể thêm vào phân tích nhưng lưu ý thường lớp trung tính dễ bị mô hình nhầm lẫn.
6. **Phân tích kết quả:** Tổng hợp kết quả phân loại để rút ra thông tin. Ví dụ: nếu phân tích cảm xúc về một thương hiệu trên mạng xã hội, có thể tính tỷ lệ % bài viết tích cực và tiêu cực, xác định những **chủ đề** phổ biến trong các phản hồi tiêu cực (bằng cách xem từ khóa xuất hiện nhiều trong nhóm này). Sinh viên có thể trực quan hóa kết quả bằng biểu đồ hoặc word cloud để báo cáo.
7. **Cải thiện:** Thử các phương pháp nâng cao nếu kết quả ban đầu chưa tốt: thu thập thêm dữ liệu huấn luyện (có thể mở rộng sang dữ liệu tiếng Anh rồi dịch sang tiếng Việt để tăng kích thước tập), tinh chỉnh tham số mô hình (độ dài vector, kiến trúc mạng). Ngoài ra, có thể tích hợp **phân tích ngữ nghĩa** hoặc **phân tích theo khía cạnh** (aspect-based sentiment analysis) để hiểu rõ hơn điều gì làm người dùng thích hoặc không thích.

Tài liệu tham khảo:

- **StreetcodeVN:** *"Phân tích cảm xúc trong Tiếng Việt"* – giới thiệu khái niệm, các phương pháp và mức độ khó dễ của bài toán phân tích cảm xúc văn bản tiếng Việt (SCV | Con đường cày cuốc) (SCV | Con đường cày cuốc) (tiếng Việt).
- **Cambridge Univ. Press:** *"Sentiment Analysis"* – sách chuyên khảo của Bing Liu, trong đó định nghĩa sentiment analysis là nghiên cứu tính toán về ý kiến, cảm xúc của con người trong văn bản (Sentiment Analysis - Cambridge University Press) (tiếng Anh).

Đề tài 9: Khai phá luật kết hợp trong dữ liệu bán lẻ (Market Basket Analysis)

Mô tả: Thực hiện **phân tích giỏ hàng** nhằm khám phá các **luật kết hợp** giữa các sản phẩm trong dữ liệu giao dịch bán lẻ. Phân tích giỏ hàng (market basket analysis) là kỹ thuật **khai thác dữ liệu** phổ biến, dùng để tìm ra những mẫu hàng hóa thường được mua cùng nhau trong các giỏ hàng của khách (Phân tích giỏ hàng - Khai thác dữ liệu khách hàng trong thời 4.0 (P1)). Kết quả là các **quy tắc "Nếu – thì"** giữa các mặt hàng, ví dụ: "Nếu khách hàng mua A và B thì nhiều khả năng sẽ mua thêm C". Doanh nghiệp có thể dựa vào đó để đưa ra quyết định kinh doanh (bày sản phẩm gần nhau, bán chéo, khuyến mãi gói sản phẩm...). Một ví dụ kinh điển: phân tích chỉ ra khách hàng mua **bánh mì** và **sữa** thường mua thêm **bơ**, do đó siêu thị có thể sắp xếp ba sản phẩm này gần nhau để tăng doanh số (What is Association Rule Learning? | by Supriyo Ain | Medium). Đề tài này giúp sinh viên làm quen với thuật toán khai thác luật kết hợp (như Apriori) trên một tập dữ liệu giao dịch và rút ra các hiểu biết có ích.

Các bước thực hiện:

1. **Thu thập dữ liệu giao dịch:** Chuẩn bị một tập dữ liệu các giao dịch bán lẻ. Dữ liệu có thể ở dạng bảng, mỗi dòng là một hóa đơn với danh sách các sản phẩm được mua. Sinh viên có thể sử dụng dữ liệu mẫu (như *Online Retail Dataset* từ UCI hoặc *Groceries dataset* từ Kaggle) hoặc giả lập một tập giao dịch nhỏ dựa trên hiểu biết thực tế.
2. **Tiền xử lý dữ liệu:** Chuyển dữ liệu về dạng phù hợp cho thuật toán khai thác. Thường cần chuyển mỗi giao dịch thành một tập hợp các mã sản phẩm (itemset). Có thể cần mã hóa tên sản phẩm thành ID số. Loại bỏ các mục quá hiếm xuất hiện nếu chúng không đem lại thông tin (ví dụ những sản phẩm chỉ xuất hiện 1-2 lần). Đảm bảo dữ liệu được tổ chức thành danh sách các transaction, mỗi transaction là tập các item.
3. **Khai thác tập phổ biến:** Áp dụng **thuật toán Apriori** (hoặc FP-Growth) để tìm ra tất cả các **itemset phổ biến** (frequent itemset) thỏa mãn ngưỡng **hỗ trợ tối thiểu** đã định. *Support* của một itemset là tỷ lệ giao dịch chứa itemset đó. Ví dụ: đặt ngưỡng support = 5% nghĩa là chỉ quan tâm những nhóm sản phẩm xuất hiện trong ít nhất 5% tổng số hóa đơn. Thuật toán sẽ liệt kê các tập sản phẩm (kích thước 1, 2, 3, ...) có tần suất >= 5%.
4. **Khai thác luật kết hợp:** Từ mỗi itemset phổ biến (có từ 2 sản phẩm trở lên), liệt kê các luật dạng $X \rightarrow Y$ (trong đó X và Y là các tập con của itemset, $X \cup Y$ = itemset). Tính **độ tin cậy (confidence)** cho từng luật: tỷ lệ % giao dịch chứa X đồng thời cũng chứa Y. Lọc các luật có độ tin cậy >= ngưỡng cho trước (ví dụ 50%). Ngoài ra, tính thêm **lift** để đánh giá mức độ hữu ích của luật (lift > 1 cho thấy X làm tăng khả năng mua Y).
5. **Phân tích kết quả:** Sắp xếp các luật theo độ tin cậy hoặc lift giảm dần và xem xét các luật thú vị. Ghi lại một số luật điển hình, chẳng hạn: {"bánh mì", "sữa"} → {"bơ"} với confidence = 60%, lift = 1.5 (giả dụ). Diễn giải ý nghĩa: 60% những người mua bánh mì và sữa cũng mua bơ, và khả năng mua bơ tăng 1.5 lần khi có bánh mì và sữa. Xem xét tính hợp lý: luật này phù hợp trực giác vì bơ ăn kèm bánh mì sữa. Một luật khác có thể bất ngờ hơn, ví dụ {"bia"} → {"tã em bé"} chẳng hạn, cần kiểm chứng và tìm hiểu nguyên nhân (có thể do nhân khẩu học khách hàng).
6. **Ứng dụng kết quả:** Đề xuất các **chiến lược kinh doanh** dựa trên những luật kết hợp tìm được. Ví dụ: những sản phẩm được mua chung nhiều nên được đặt gần nhau trong cửa

hàng hoặc gợi ý "khách mua A thường mua thêm B" trên website. Thiết kế gói khuyến mãi cho nhóm sản phẩm hay đi cùng (combo giảm giá nếu mua cả bộ). Nếu phát hiện nhóm sản phẩm không ngờ tới, có thể nghiên cứu hành vi khách hàng tương ứng.

7. **Mở rộng và cải thiện:** Thử điều chỉnh các ngưỡng hỗ trợ và độ tin cậy để kiểm soát số lượng luật khai thác (ngưỡng thấp sẽ ra nhiều luật hơn nhưng dễ có luật nhiễu). Nếu dữ liệu lớn, tối ưu thuật toán (FP-Growth nhanh hơn Apriori với tập lớn). Ngoài ra, có thể phân tích sâu hơn: phân nhóm giao dịch theo **thời gian** (mùa sale, ngày lễ) hoặc theo **loại khách hàng**, sau đó chạy phân tích giỏ hàng riêng từng nhóm để thấy sự khác biệt. Điều này nâng cao hiểu biết và làm đề tài phong phú hơn.

Tài liệu tham khảo:

- **Blog Sapo:** *"Phân tích giỏ hàng – Market Basket Analysis là gì?"* – giải thích khái niệm phân tích giỏ hàng và lợi ích trong kinh doanh bán lẻ, thương mại điện tử (Phân tích giỏ hàng - Khai thác dữ liệu khách hàng trong thời 4.0 (P1)) (tiếng Việt).
- **Medium:** *"What is Association Rule Learning?"* – bài viết giới thiệu về thuật toán học luật kết hợp (Apriori), có ví dụ trực quan trong siêu thị với các sản phẩm như sữa, bánh mì, bơ (What is Association Rule Learning? | by Supriyo Ain | Medium) (tiếng Anh).

Đề tài 10: Nhận dạng biển báo giao thông bằng học sâu

Mô tả: Xây dựng mô hình **thị giác máy tính (computer vision)** để nhận dạng các **biển báo giao thông** trong ảnh. Đây là một bài toán phân loại hình ảnh nhiều lớp: đầu vào là ảnh chụp biển báo, đầu ra là loại biển báo (ví dụ: biển giới hạn tốc độ 50, biển cấm rẽ trái, biển dừng, v.v.) (Phân loại biển báo giao thông bằng Deep Learning (CNN) - Mì AI). Ứng dụng của hệ thống này là hỗ trợ lái xe hoặc xe tự hành nhận biết biển báo trên đường nhằm nâng cao an toàn giao thông. Với sự hỗ trợ của các mạng **Convolutional Neural Network (CNN)**, độ chính xác nhận dạng biển báo đã rất cao (có mô hình đạt ~95-99% trên tập dữ liệu chuẩn) (Traffic Signs Recognition using CNN and Keras in Python). Sinh viên sẽ thu thập bộ dữ liệu các hình ảnh biển báo, sau đó huấn luyện mô hình CNN để phân loại chúng. Đề tài giúp rèn luyện kỹ năng xử lý ảnh và ứng dụng học sâu cơ bản.

Các bước thực hiện:

1. **Thu thập dữ liệu hình ảnh:** Sử dụng bộ dữ liệu công khai về biển báo. Ví dụ phổ biến là **German Traffic Sign Recognition Benchmark (GTSRB)** – gồm ~40.000 ảnh biển báo thuộc 43 loại khác nhau (Phân loại biển báo giao thông bằng Deep Learning (CNN) - Mì AI). Ngoài ra có thể kết hợp với một số ảnh biển báo Việt Nam (biển báo tương tự châu Âu về hình dạng màu sắc, chỉ khác ngôn ngữ chữ viết). Chia dữ liệu thành các thư mục theo lớp biển báo và sẵn sàng cho bước huấn luyện.
2. **Tiền xử lý dữ liệu ảnh:** Thống nhất kích thước ảnh (ví dụ resize về 32x32 hoặc 64x64 pixel). Chuẩn hóa giá trị pixel (về khoảng [0,1] hoặc chuẩn hóa theo trung bình-độ lệch chuẩn). Có thể áp dụng **data augmentation** để tăng cường dữ liệu huấn luyện: xoay ảnh một góc nhỏ, thay đổi độ sáng/tương phản, dịch chuyển ảnh... giúp mô hình **chống overfitting** và hiểu được biến dạng của biển báo trong thực tế (do góc chụp, ánh sáng khác nhau).

3. **Xây dựng mô hình CNN:** Thiết kế một kiến trúc CNN phù hợp. Với bài toán này có thể bắt đầu bằng một mạng nhỏ: 2-3 lớp tích chập (convolutional layer) xen kẽ lớp pooling, sau đó là 1-2 lớp kết nối đầy đủ (fully-connected) và softmax đầu ra tương ứng số lớp biển báo. Sử dụng hàm kích hoạt ReLU cho các lớp ẩn. Ngoài ra, có thể thử phương pháp **transfer learning**: sử dụng các mô hình pretrained trên ImageNet như **ResNet50, MobileNet** rồi fine-tune lại trên dữ liệu biển báo để tận dụng đặc trưng có sẵn.
4. **Huấn luyện mô hình:** Chia dữ liệu thành tập huấn luyện, **validation** và **test**. Sử dụng tập huấn luyện để training CNN qua nhiều epoch, theo dõi độ chính xác và hàm lỗi trên tập validation để điều chỉnh (dừng huấn luyện khi model có dấu hiệu overfit hoặc lỗi không giảm). Dùng **optimizer** như Adam với learning rate phù hợp (ví dụ 0.001). Nếu mô hình không học tốt, thử thay đổi kiến trúc hoặc tham số (tăng số epoch, điều chỉnh batch size).
5. **Đánh giá mô hình:** Sau khi huấn luyện xong, chạy mô hình trên tập test (những ảnh chưa thấy trong quá trình train/val). Đánh giá **accuracy** trên tập test. Ngoài ra tính ma trận nhầm lẫn (confusion matrix) để xem mô hình hay nhầm lẫn giữa những biển báo nào – ví dụ có thể nhầm giữa các biển hạn chế tốc độ 30 và 50 do nhìn khá giống, v.v. Nếu độ chính xác đạt mức cao (~90-95% trở lên) là tín hiệu tốt cho mô hình. So sánh kết quả với các mô hình khác nếu có triển khai (transfer learning vs từ đầu).
6. **Trình bày kết quả:** Xây dựng một demo nhỏ hiển thị kết quả nhận dạng. Ví dụ: viết script Python cho phép chọn một ảnh biển báo bất kỳ, model sẽ dự đoán loại và hiển thị tên biển báo tương ứng. Đưa vào báo cáo một số ảnh mẫu và kết quả dự đoán để minh họa khả năng của mô hình.
7. **Mở rộng:** Nếu sinh viên hứng thú, có thể mở rộng từ nhận dạng **tĩnh** sang **nhận dạng biển báo theo thời gian thực**. Ví dụ: tích hợp mô hình vào một ứng dụng dùng camera (hoặc video) để vừa phát hiện vùng biển báo trong khung hình, vừa phân loại loại biển báo đó. Việc phát hiện có thể dùng phương pháp đơn giản dựa trên màu sắc hình dạng (vì biển báo có hình dạng đặc trưng) hoặc dùng mô hình object detection (như YOLO, SSD – phần này nâng cao). Mở rộng này giúp tiếp cận gần hơn với hệ thống trên xe tự lái thực tế.

Tài liệu tham khảo:

- **Mì AI blog:** *"Phân loại biển báo giao thông bằng Deep Learning (CNN)"* – hướng dẫn chi tiết (tiếng Việt) sử dụng mạng CNN để huấn luyện trên bộ dữ liệu GTSRB 43 loại biển báo, bao gồm cách chuẩn bị dữ liệu và kiến trúc mạng mẫu (Phân loại biển báo giao thông bằng Deep Learning (CNN) - Mì AI).
- **Analytics Vidhya:** *"Traffic Signs Recognition using CNN"* – bài tutorial (tiếng Anh) xây dựng mô hình CNN nhận dạng biển báo với độ chính xác ~95%, có phần trình bày cách thức hoạt động và ứng dụng của hệ thống nhận dạng biển báo trong hỗ trợ lái xe (Traffic Signs Recognition using CNN and Keras in Python).