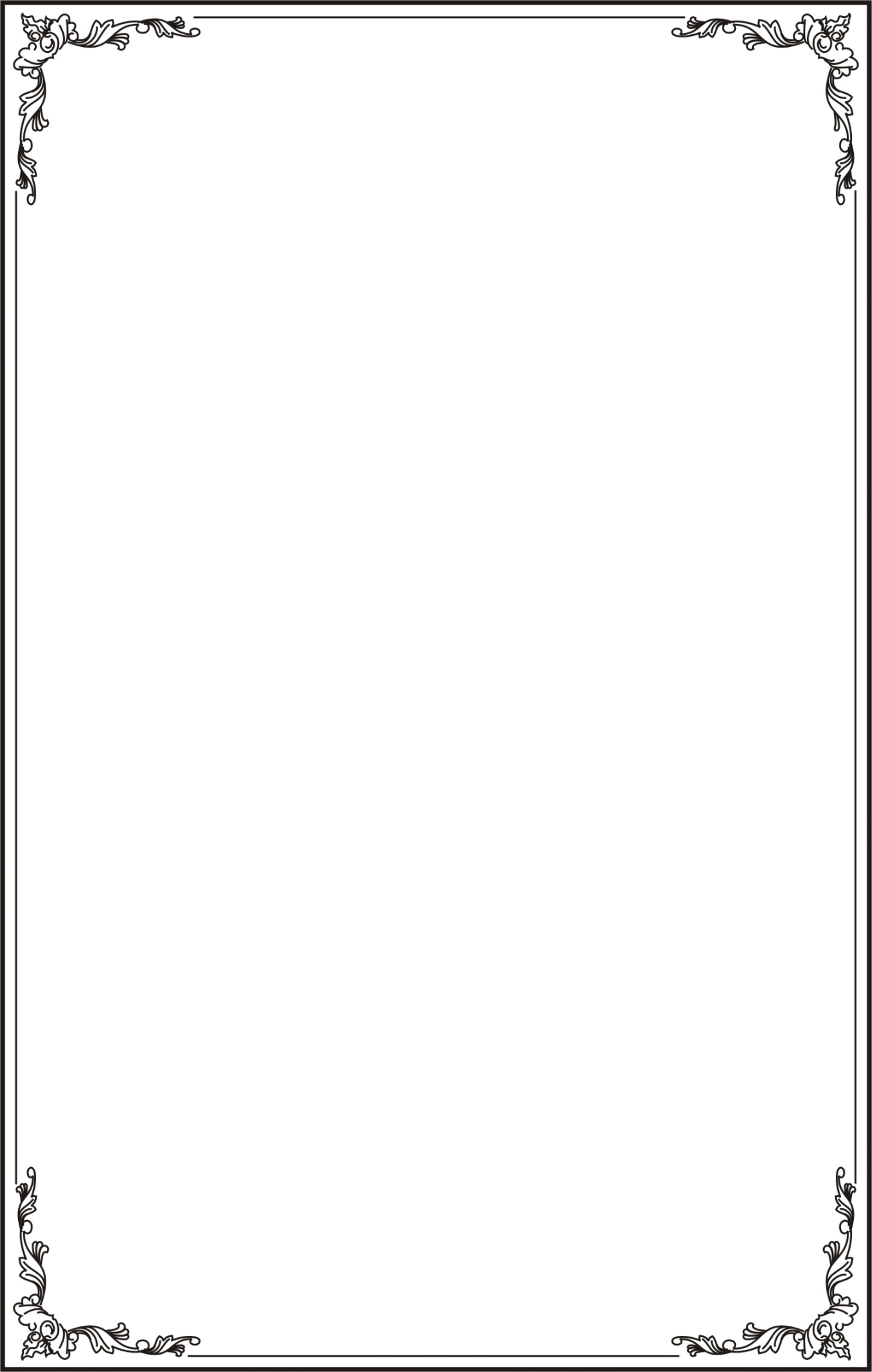
**TRƯỜNG ĐẠI HỌC KHOA HỌC HUẾ**

**KHOA CÔNG NGHỆ THÔNG TIN**

**- - - \*\*\* - - -**



**BÁO CÁO MÔN HỌC**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**ĐỀ TÀI: PHÂN LOẠI VĂN BẢN TIẾNG VIỆT THEO CHỦ ĐỀ**

**Nhóm 5:**

**Thành viên:**

22T1020289 - Nguyễn Bá Nhật

22T1020019 - Lê Ngọc Ánh

20T1020648 - Trần Gia Vũ

Giáo viên hướng dẫn:

**TS. Đoàn Thị Hồng Phước**

# MỤC LỤC

[**MỤC LỤC 1**](#_t2qgewi42921)

[**LỜI CẢM ƠN 1**](#_73zw0anxa3hq)

[**TÓM TẮT 5**](#_94b556kvynez)

[**Chương I. MỞ ĐẦU 6**](#_23d2p66f84nt)

[1.1 Lý do chọn đề tài: 6](#_gfvg4ct53lp9)

[1.2 Mục tiêu nghiên cứu: 7](#_6vvuczprg1jc)

[1.3 Đối tượng và phạm vi nghiên cứu: 8](#_kacxffarrjx1)

[1.3.1. Đối tượng nghiên cứu: 8](#_nv4d1ykv6v94)

[1.3.2. Phạm vi nghiên cứu: 8](#_bw4659cukd9h)

[1.4. Phương pháp nghiên cứu: 9](#_qgzo16wbeat0)

[**Chương II. CƠ SỞ LÝ THUYẾT 11**](#_g4k6xceuxov6)

[2.1. Tổng quan về bài toán phân loại văn bản: 11](#_rbz7mlhe5pa0)

[2.2. Tiền xử lý văn bản tiếng Việt: 12](#_4ndgbuwh7mgi)

[2.2.1. Đặc thù của tiếng Việt trong xử lý ngôn ngữ tự nhiên 12](#_6len59at43gg)

[2.2.2. Chuẩn hóa văn bản 12](#_remusrtpyik0)

[2.2.3. Loại bỏ stopword 13](#_es7ppiarasq9)

[2.2.4. Tách từ (Tokenization) 13](#_kacyff3eq6lr)

[2.3. Biểu diễn và trích xuất đặc trưng văn bản: 14](#_6ern5i331ccg)

[2.3.1. Khái niệm trích xuất đặc trưng 14](#_bjbw8iza218b)

[2.3.2. Bag of Words và Count Vector 15](#_6jtk9fhlga4o)

[2.3.3. Mã hóa TF-IDF 15](#_v6somwf8m7hc)

[2.3.4. N-gram 18](#_v3tkj4rkat00)

[2.3.5. Static Embedding – Word2Vec 19](#_35xfutsxuwxj)

[2.3.6. Contextual Embedding – Transformer 21](#_byco4vwbaz82)

[2.3.7. Giảm chiều dữ liệu (SVD) 24](#_c0o42ragbaqk)

[2.4. Các mô hình học máy (Machine learning): 26](#_3cdoeehdreye)

[2.4.1. Thuật toán Logistic Regression: 26](#_gehctwtoqp1h)

[2.4.2. Thuật toán Support vector machine: 28](#_j9cm4ijkzcdr)

[2.4.3. Thuật toán XGBoost: 29](#_vj6y4bk5iw1s)

[2.5. Các mô hình học sâu (Deep learning): 31](#_4dewi1vduznw)

[2.5.1. Thuật toán Deep Neural Network (DNN): 31](#_ga7uuxlgdkoq)

[2.5.2. Thuật toán Long Short-Term Memory (LSTM): 33](#_8nwwzgos3gr)

[2.6. Các chỉ số đánh giá hiệu suất mô hình: 34](#_pv9o2fv2hrj6)

[2.7. Xử lý mất cân bằng lớp: 38](#_m8rolmdcv47c)

[2.7.1. So sánh các phương pháp xử lý mất cân bằng lớp: 38](#_bjogy1411gc8)

[2.7.2. Lý do lựa chọn: 40](#_ww5ls5aabz9v)

[**Chương III. TRIỂN KHAI VÀ ĐÁNH GIÁ MÔ HÌNH 42**](#_jzauxlldb40u)

[3.1. Bộ dữ liệu: 42](#_6btw322ioa32)

[3.2. Tiền xử lý: 45](#_zdiwdj5gsuju)

[3.2.1. Tiền xử lý dữ liệu tổng thể 45](#_ln0ayvjc89g1)

[3.2.2. Tiền xử lý dữ liệu văn bản 46](#_wft5q7w20aay)

[3.2.3. Vai trò của tiền xử lý trong dự án 47](#_u8y9p9b4nuo4)

[3.3. Đánh giá các mô hình học máy: 47](#_kzekcfoyqzmk)

[3.3.1. Mô hình SVM kết hợp với đặc trưng TF-IDF: 47](#_gd8ucpeah3am)

[3.3.2. Mô hình SVM kết hợp với TF-IDF và giảm chiều dữ liệu bằng SVD: 48](#_jodp7fot8cpk)

[3.3.3. Mô hình SVM kết hợp với đặc trưng N-gram và SVD: 50](#_3d81aom6bxwf)

[3.3.4. Mô hình Logistic Regression kết hợp với đặc trưng TF-IDF: 51](#_i6yh2hsd7c0f)

[3.3.5. Mô hình Logistic Regression kết hợp với TF-IDF và SVD: 52](#_5c0qpyrswbjv)

[3.3.6. Mô hình Logistic Regression kết hợp với đặc trưng N-gram và SVD: 53](#_8e9xy09m4auc)

[3.3.7. Mô hình XGBoost kết hợp với TF-IDF và SVD: 54](#_ejhovfnn7pda)

[3.3.8. Mô hình XGBoost kết hợp với đặc trưng N-gram và SVD: 55](#_7x93ujqiljys)

[3.3.9. Mô hình Logistic Regression kết hợp với Word2Vec tiền huấn luyện: 56](#_puwcebstfr4w)

[3.3.10. Mô hình SVM kết hợp với Word2Vec tiền huấn luyện: 57](#_o9bxfkoqh8ex)

[3.3.11. Mô hình DNN kết hợp với Word2Vec tiền huấn luyện: 58](#_wtpnk6ndjv6p)

[3.3.12. Mô hình LSTM kết hợp với Word2Vec tiền huấn luyện: 59](#_i6l5t6jp4msw)

[3.3.13. Mô hình DNN kết hợp với TF-IDF và SVD: 60](#_w7feuwdu9mrv)

[3.3.14. Mô hình DNN kết hợp với đặc trưng N-gram và SVD: 61](#_ix261prtxen4)

[3.3.15. Mô hình LSTM kết hợp với TF-IDF và SVD: 63](#_t5rgm8uvk24l)

[3.4. Tổng hợp kết quả: 64](#_94srcttfkkcq)

[3.5. Đánh giá ma trận nhầm lẫn: 65](#_e0k1i2wto2yc)

[**Chương IV. KẾT LUẬN 68**](#_71ewwiwmvebr)

[**ĐÓNG GÓP CỦA CÁC THÀNH VIÊN 68**](#_3e0heosktt8o)

[**TÀI LIỆU THAM KHẢO 69**](#_qs6gk7wngi46)

# 

# LỜI CẢM ƠN

Nhóm em xin gửi đến quý thầy cô ở Khoa Công nghệ thông tin – Trường Đại Học Khoa Học Huế lời biết ơn sâu sắc nhất, những người đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong thời gian học tập tại trường.

Chúng em xin chân thành cảm ơn TS. Cô Đoàn Thị Hồng Phước đã tận tâm hướng dẫn, giúp đỡ trong quá trình định hướng, nghiên cứu và hoàn thành tiểu luận một cách tốt nhất. Chúng em cũng xin gửi lời cảm ơn chân thành đến gia đình, bạn bè, đã luôn là nguồn động viên giúp chúng em vượt qua những khó khăn trong suốt quá trình học tập và thực hiện viết niên luận.

Mặc dù đã rất cố gắng hoàn thành tiểu luận với tất cả sự nỗ lực, nhưng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự quan tâm, cảm thông và đóng góp của cô để khóa niên luận ngày càng được hoàn thiện hơn.

Một lần nữa, chúng em xin chân thành cảm ơn và luôn mong nhận được sự đóng góp của thầy cô. Xin kính chúc các thầy cô trong Khoa Công nghệ thông tin dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chúng em xin chân thành cảm ơn!

**Sinh Viên**

Lê Ngọc Ánh

Nguyễn Bá Nhật

Trần Gia Vũ

**DANH MỤC HÌNH VẼ**

[**Chương II. CƠ SỞ LÝ THUYẾT 11**](#_g4k6xceuxov6)

[Hình 2.1. pipeline của bài toán phân loại văn bản 11](#_xdoa1xlojh1b)

[Hình 2.2. Minh họa embedding space và mối quan hệ ngữ nghĩa giữa các từ 19](#_1psgxgfyxv8e)

[Hình 2.3. Minh họa kiến trúc Word2Vec gồm mô hình CBOW và Skip-gram 20](#_jxkbjk2u9dmb)

[Hình 2.4. Minh họa việc Word2Vec gán mỗi từ một vector cố định 21](#_c23niciivum2)

[Hình 2.5. khác biệt giữa embedding tĩnh và embedding ngữ cảnh 22](#_mdsm8lf7yjz7)

[Hình 2.6. Minh họa cơ chế Positional Encoding trong kiến trúc Transformer 23](#_8yokvs92356w)

[Hình 2.7. Minh họa cơ chế Multihead attention trong kiến trúc Transformer 23](#_3eb52qobygw)

[Hình 2.8. Giảm chiều dữ liệu (Dimensionality Reduction) 25](#_merwnidn4nn4)

[Hình 2.9. Nguyên lý phân rã ma trận trong SVD 26](#_qdc037nehx2w)

[Hình 2.10. Hồi quy Logistic: biến đổi đầu ra tuyến tính thành xác suất bằng hàm sigmoid. 27](#_wpkgnj8o35en)

[Hình 2.11. Siêu phẳng phân tách và khoảng cách biên (margin) trong SVM 28](#_62dtl7ouib9u)

[Hình 2.12. Kiến trúc XGBoost với chiến lược phát triển theo chiều sâu. 30](#_kaer47j1th76)

[Hình 2.13. Cấu trúc đơn giản hóa của mô hình XGBoost 31](#_ary41jfdsdm8)

[Hình 2.14. Kiến trúc mạng neural sâu (Deep Neural Network – DNN) 32](#_hqn8ugqw4hcu)

[Hình 2.15. Kiến trúc ô nhớ LSTM (Long Short-Term Memory). 33](#_d7jmh8llqlyo)

[**Chương III. TRIỂN KHAI VÀ ĐÁNH GIÁ MÔ HÌNH 42**](#_jzauxlldb40u)

[Hình 3.1. Tỷ lệ phân bố dữ liệu theo từng chủ đề 43](#_49a34l47dh6w)

[Hình 3.2. Nguyên lý hoạt động của Stratified Fold Cross-Validation 44](#_ubpiukxz6d3o)

[Hình 3.3. Nguyên lý kỹ thuật Early Stopping dựa trên validation loss. 45](#_7yp7ghzh1621)

[Hình 3.4. Chỉ số đánh giá của mô hình SVM với đặc trưng TF-IDF. 48](#_6w7j08tt9we6)

[Hình 3.5. Chỉ số đánh giá của mô hình SVM kết hợp với TF-IDF và giảm](#_y3qvplcj8zjj)

[Hình 3.6. Chỉ số đánh giá của mô hình SVM kết hợp với N-gram và giảm chiều SVD 50](#_ef2w15glk6yg)

[Hình 3.7. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với TF-IDF 51](#_9t9s8ww6z6q8)

[Hình 3.8. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với TF-IDF và SVD 52](#_9hdku9484tny)

[Hình 3.9. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với N-gram và SVD 53](#_w9rwn6sb3ei4)

[Hình 3.10. Chỉ số đánh giá của mô hình XGBoost kết hợp với TF-IDF và SVD 54](#_246n5lj4oy2p)

[Hình 3.11. Chỉ số đánh giá của mô hình XGBoost kết hợp với đặc trưng N-gram và SVD 55](#_e7wntxuj88vr)

[Hình 3.12. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với Word2Vec tiền huấn luyện. 56](#_3rv690kl8xp0)

[Hình 3.13. Chỉ số đánh giá của mô hình SVM kết hợp với Word2Vec tiền huấn luyện. 58](#_otrdi0rsttve)

[Hình 3.14. Chỉ số đánh giá của mô hình Mô hình Mạng nơ-ron (DNN) kết hợp với Word2Vec tiền huấn luyện. 59](#_9wakxr6615n)

[Hình 3.15. Chỉ số đánh giá của mô hình LSTM kết hợp với Word2Vec tiền huấn luyện. 60](#_tgfpc7bousan)

[Hình 3.16. Chỉ số đánh giá của mô hình Mạng nơ-ron sâu DNN kết hợp với TF-IDF và SVD. 61](#_lo0vcm6kk4py)

[Hình 3.17. Chỉ số đánh giá của mô hình DNN kết hợp với đặc trưng N-gram và SVD. 62](#_vagjq9jwtd41)

[Hình 3.18. Chỉ số đánh giá của mô hình Mạng nơ-ron hồi tiếp LSTM kết hợp](#_bx3dzwm5npsb)

[Hình 3.19. Ma trận nhầm lẫn của mô hình SVM kết hợp TF-IDF. 66](#_nf2elts9ebfr)

**DANH MỤC BẢNG**

[Bảng 2.1. Giá trị TF của các văn bản trong bộ dữ liệu 18](#_qocq98vjwdi4)

[Bảng 2.2. Giá trị IDF của các từ trong bộ từ điển 18](#_tzpgw8y0gm41)

[Bảng 2.4. Ma trận nhầm lẫn mô tả kết quả dự đoán của mô hình so với nhãn thực tế. 36](#_i4533ztpb9ob)

[Bảng 3.1. Phân bố số lượng văn bản theo từng chủ đề trong bộ dữ liệu VNTC 43](#_kex4umke054f)

# TÓM TẮT

Sự phát triển mạnh mẽ của Internet và các nền tảng số đã làm cho dữ liệu văn bản tiếng Việt ngày càng gia tăng cả về số lượng lẫn nội dung. Việc tổ chức và xử lý khối lượng lớn dữ liệu văn bản này đặt ra nhiều thách thức, đặc biệt do đặc thù ngôn ngữ tiếng Việt như từ ghép, đa nghĩa, cách viết không thống nhất. Trong bối cảnh đó, bài toán phân loại văn bản tiếng Việt theo chủ đề đóng vai trò quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, nhằm tự động gán nhãn chủ đề cho văn bản và hỗ trợ hiệu quả cho các bài toán khai thác thông tin. Đề tài “Phân loại văn bản tiếng Việt theo chủ đề” được thực hiện với mục tiêu xây dựng một hệ thống phân loại văn bản tự động dựa trên các phương pháp học máy và học sâu. Nội dung nghiên cứu tập trung xây dựng một quy trình xử lý hoàn chỉnh, bao gồm các bước tiền xử lý văn bản, trích xuất đặc trưng, huấn luyện mô hình và đánh giá kết quả trên tập dữ liệu văn bản tiếng Việt thực tế. Trong giai đoạn tiền xử lý, văn bản được làm sạch và chuẩn hóa, bao gồm xử lý teencode, tách từ tiếng Việt và loại bỏ các từ dừng nhằm nâng cao chất lượng dữ liệu đầu vào. Sau đó, văn bản được biểu diễn dưới dạng đặc trưng số thông qua các phương pháp như Bag-of-Words, TF-IDF, giảm chiều dữ liệu bằng SVD và mô hình Word2Vec huấn luyện sẵn (pretrained), cho phép khai thác thông tin ngữ nghĩa của từ dựa trên các vector đã được huấn luyện trước trên tập dữ liệu lớn tiếng Việt. Trên cơ sở các đặc trưng đã trích xuất, đề tài tiến hành huấn luyện và đánh giá nhiều mô hình khác nhau, bao gồm các mô hình học máy truyền thống như Logistic Regression, Support Vector Machine, XGBoost và các mô hình học sâu như DNN và LSTM. Kết quả thực nghiệm cho thấy hệ thống có khả năng phân loại văn bản theo chủ đề với độ chính xác tốt, góp phần giảm thời gian và công sức so với phương pháp xử lý thủ công, đồng thời tạo nền tảng cho việc phát triển các ứng dụng xử lý và khai thác văn bản tiếng Việt trong thực tế.

# 

# Chương I. MỞ ĐẦU

## 1.1 Lý do chọn đề tài:

Ngày nay, sự phát triển vượt bậc của công nghệ thông tin, đặc biệt là sự bùng nổ của mạng Internet, đã làm cho lượng thông tin được số hóa và đưa lên mạng ngày càng nhiều. Internet trở thành một kho tri thức khổng lồ bao phủ hầu hết các lĩnh vực của đời sống xã hội. Do đó, số lượng văn bản xuất hiện trên Internet cũng tăng lên nhanh chóng với tốc độ chóng mặt. Hiện nay, số lượng trang web được các công cụ tìm kiếm lập chỉ mục đã lên đến hàng tỷ trang, chưa kể đến các văn bản được lưu trữ và cập nhật thường xuyên trên các nền tảng số. Ngoài ra, nhiều nghiên cứu cũng chỉ ra rằng nội dung văn bản trên Internet thay đổi rất nhanh, với tốc độ tăng trưởng và cập nhật liên tục theo thời gian. Với lượng thông tin văn bản khổng lồ như vậy, một vấn đề đặt ra là làm thế nào để tổ chức, quản lý và khai thác hiệu quả nguồn dữ liệu này. Hơn nữa, trong thực tế sử dụng, người dùng thường có nhu cầu phân loại và nhóm các văn bản theo những chủ đề cụ thể nhằm phục vụ cho các mục đích như tra cứu, thống kê, phân tích và xử lý thông tin. Điều này đặt ra thách thức lớn đối với các phương pháp xử lý thủ công, đồng thời thúc đẩy sự phát triển của các phương pháp tự động trong lĩnh vực xử lý ngôn ngữ tự nhiên. Từ đó, bài toán phân loại văn bản theo chủ đề trở thành một hướng tiếp cận phù hợp và cần thiết để giải quyết nhu cầu trên.

Xuất phát từ những yêu cầu thực tiễn đó, nhóm quyết định lựa chọn đề tài *“Phân loại văn bản tiếng Việt theo chủ đề”*, với mục tiêu xây dựng và đánh giá một hệ thống có khả năng tự động phân loại văn bản tiếng Việt theo các chủ đề định sẵn. Ứng dụng này góp phần giảm thời gian và công sức của con người so với các phương pháp xử lý thủ công, đồng thời nâng cao hiệu quả khai thác và quản lý dữ liệu văn bản tiếng Việt trong thực tế.

## 1.2 [Mục tiêu nghiên cứu:](https://docs.google.com/document/d/1c-SVzs2IfORvZcwA-_CFpO6cT7RwcWMykZ-LzOb_DiI/edit?tab=t.0#heading=h.6vvuczprg1jc)

Trong bối cảnh số lượng văn bản Tiếng Việt ngày càng gia tăng mạnh mẽ trên các nền tảng số như báo điện tử, mạng xã hội và hệ thống quản lý tài liệu, việc tổ chức và khai thác thông tin hiệu quả trở thành một yêu cầu cấp thiết. Tuy nhiên, phần lớn các văn bản hiện nay tồn tại dưới dạng dữ liệu phi cấu trúc, gây nhiều khó khăn cho quá trình tìm kiếm, phân loại và phân tích nội dung. Xuất phát từ thực tế đó, đề tài *“Phân loại văn bản Tiếng Việt theo chủ đề”* được thực hiện nhằm nghiên cứu bài toán phân loại văn bản trong lĩnh vực xử lý ngôn ngữ tự nhiên và xây dựng hệ thống phân loại dựa trên các kỹ thuật học máy và học sâu.

Mục tiêu của đề tài là giúp người nghiên cứu hiểu rõ bản chất của bài toán phân loại văn bản, bao gồm cách biểu diễn văn bản Tiếng Việt dưới dạng dữ liệu số và cách các thuật toán học máy xử lý những biểu diễn này để nhận diện chủ đề của văn bản. Đồng thời, đề tài tập trung nghiên cứu và áp dụng các kỹ thuật tiền xử lý văn bản Tiếng Việt nhằm nâng cao chất lượng dữ liệu đầu vào, khảo sát và so sánh các phương pháp biểu diễn văn bản như Bag-of-Words, TF-IDF và Word Embedding, cũng như xây dựng, huấn luyện và đánh giá các mô hình học máy, học sâu thông qua các chỉ số đánh giá phù hợp. Trên cơ sở đó, đề tài hướng đến việc đề xuất cách tiếp cận có khả năng ứng dụng trong các hệ thống phân loại tin tức, quản lý tài liệu số và hỗ trợ tìm kiếm thông tin trong các ứng dụng xử lý ngôn ngữ Tiếng Việt.

## 

## 1.3 Đối tượng và phạm vi nghiên cứu:

### ***1.3.1. Đối tượng nghiên cứu:***

Đối tượng nghiên cứu chính của đề tài là các văn bản được viết bằng Tiếng Việt được thu thập từ nhiều trang tin uy tín như VNExpress, Tuổi Trẻ, Thanh Niên và Người Lao Động, phản ánh nhiều lĩnh vực và chủ đề khác nhau trong đời sống xã hội. Những văn bản này có thể là bài viết tin tức, bài đăng trực tuyến hoặc dữ liệu văn bản được xây dựng nhằm phục vụ mục đích nghiên cứu. Việc lựa chọn văn bản Tiếng Việt làm đối tượng nghiên cứu xuất phát từ đặc thù ngôn ngữ có cấu trúc phức tạp, nhiều biến thể ngữ nghĩa và hình thức biểu đạt, đặt ra không ít thách thức trong bài toán xử lý và phân loại tự động.

Bên cạnh dữ liệu văn bản, đề tài còn tập trung nghiên cứu các phương pháp và kỹ thuật cốt lõi trong lĩnh vực xử lý ngôn ngữ tự nhiên, bao gồm các bước tiền xử lý văn bản Tiếng Việt, các phương pháp biểu diễn nội dung văn bản dưới dạng số và các mô hình học máy được sử dụng để xác định chủ đề của văn bản. Đây là những thành phần quyết định khả năng nhận diện và phân loại chính xác nội dung văn bản theo từng nhóm chủ đề khác nhau.

### ***1.3.2. Phạm vi nghiên cứu:***

Về nội dung nghiên cứu, đề tài tập trung vào bài toán “*phân loại văn bản Tiếng Việt theo chủ đề*”, trong đó mỗi văn bản chỉ thuộc về một chủ đề duy nhất. Nghiên cứu không mở rộng sang các hướng tiếp cận nâng cao như phân loại đa nhãn, phân tích cảm xúc, trích xuất thực thể hay sinh văn bản tự động, nhằm đảm bảo tính tập trung và phù hợp với mục tiêu của học phần.

Về mặt phương pháp, đề tài giới hạn trong việc khảo sát và áp dụng một số phương pháp tiêu biểu trong xử lý ngôn ngữ tự nhiên và học máy. Các phương pháp được lựa chọn vừa đảm bảo tính đại diện cho lĩnh vực nghiên cứu, vừa phù hợp với điều kiện thực nghiệm và mục tiêu học tập. Nghiên cứu không đi sâu vào việc tối ưu hóa thuật toán cho dữ liệu cực lớn hoặc các hệ thống yêu cầu xử lý thời gian thực.

Về dữ liệu và môi trường thực nghiệm, đề tài được thực hiện trên tập dữ liệu văn bản Tiếng Việt có quy mô vừa phải, đủ để đánh giá và so sánh hiệu quả của các phương pháp phân loại. Kết quả nghiên cứu chủ yếu mang tính học thuật và minh họa, chưa hướng đến triển khai trong các hệ thống ứng dụng thực tế quy mô lớn.

## [1.4. Phương pháp nghiên cứu:](https://docs.google.com/document/d/1c-SVzs2IfORvZcwA-_CFpO6cT7RwcWMykZ-LzOb_DiI/edit?tab=t.0#heading=h.94r8h2fg97tb)

Để đạt được các mục tiêu đã đề ra, đề tài *“Phân loại văn bản Tiếng Việt theo chủ đề”* được thực hiện dựa trên sự kết hợp giữa nghiên cứu lý thuyết và nghiên cứu thực nghiệm. Cách tiếp cận này cho phép vừa nắm vững nền tảng học thuật của bài toán, vừa đánh giá được hiệu quả của các phương pháp thông qua kết quả thực tế.

Trước hết, đề tài tiến hành nghiên cứu tài liệu nhằm tìm hiểu cơ sở lý thuyết liên quan đến xử lý ngôn ngữ tự nhiên và phân loại văn bản. Nội dung nghiên cứu bao gồm các khái niệm cơ bản về văn bản, đặc điểm của ngôn ngữ Tiếng Việt, các kỹ thuật tiền xử lý văn bản cũng như nguyên lý hoạt động của các phương pháp biểu diễn văn bản, mô hình học máy và học sâu. Việc nghiên cứu này đóng vai trò nền tảng, giúp định hướng cho quá trình xây dựng và triển khai mô hình.

Tiếp theo, đề tài sử dụng phương pháp thực nghiệm để xây dựng hệ thống phân loại văn bản Tiếng Việt theo chủ đề. Quá trình thực nghiệm được thực hiện theo một quy trình gồm nhiều bước liên tiếp: thu thập và chuẩn bị dữ liệu, tiền xử lý văn bản Tiếng Việt, trích xuất đặc trưng, huấn luyện mô hình và đánh giá kết quả. Mỗi bước trong quy trình đều được thực hiện có hệ thống nhằm đảm bảo tính nhất quán và độ tin cậy của kết quả nghiên cứu. Bên cạnh đó, đề tài áp dụng phương pháp so sánh và đánh giá để phân tích hiệu quả của các mô hình phân loại. Các mô hình được đánh giá dựa trên các chỉ số đo lường phổ biến trong bài toán phân loại văn bản, qua đó làm rõ ưu điểm, hạn chế và mức độ phù hợp của từng phương pháp đối với dữ liệu văn bản Tiếng Việt.

Cuối cùng, thông qua việc tổng hợp kết quả thực nghiệm và phân tích đánh giá, đề tài sử dụng phương pháp phân tích và tổng hợp để rút ra nhận xét chung, từ đó đề xuất hướng tiếp cận phù hợp cho bài toán phân loại văn bản Tiếng Việt theo chủ đề. Những kết luận này không chỉ phản ánh kết quả đạt được của đề tài mà còn là cơ sở cho các nghiên cứu mở rộng trong tương lai.

# 

# Chương II. CƠ SỞ LÝ THUYẾT

## 2.1. Tổng quan về bài toán phân loại văn bản:

Phân loại văn bản (Text Classification) là một trong những bài toán cơ bản và quan trọng trong lĩnh vực Xử lý ngôn ngữ tự nhiên. Mục tiêu của bài toán này là tự động gán nhãn cho văn bản dựa trên nội dung mà văn bản truyền tải. Trong phạm vi báo cáo này, bài toán được xem xét dưới dạng phân loại văn bản tiếng Việt theo chủ đề, trong đó mỗi văn bản thuộc về một chủ đề xác định. Phân loại văn bản được ứng dụng rộng rãi trong nhiều lĩnh vực như phân loại tin tức, lọc thư rác, phân loại ý kiến người dùng, phân tích dư luận xã hội và hỗ trợ tìm kiếm thông tin. Việc tự động hóa quá trình phân loại giúp giảm thiểu công sức xử lý thủ công, đồng thời nâng cao hiệu quả và độ chính xác trong việc quản lý và khai thác dữ liệu văn bản. Tuy nhiên, đối với tiếng Việt, bài toán phân loại văn bản gặp nhiều thách thức do đặc thù ngôn ngữ, bao gồm hiện tượng đa nghĩa, từ ghép đa âm tiết và việc thiếu dấu phân cách từ rõ ràng. Những đặc điểm này khiến quá trình xử lý và biểu diễn văn bản trở nên phức tạp hơn so với các ngôn ngữ như tiếng Anh, từ đó đặt ra yêu cầu cao hơn đối với các phương pháp tiền xử lý, trích xuất đặc trưng và xây dựng mô hình phân loại.

## 

## Hình 2.1 : pipeline của bài toán phân loại văn bản

## 2.2. Tiền xử lý văn bản tiếng Việt:

### ***2.2.1. Đặc thù của tiếng Việt trong xử lý ngôn ngữ tự nhiên***

Tiếng Việt là một ngôn ngữ đơn lập, trong đó một từ có thể bao gồm nhiều âm tiết được viết tách rời bằng dấu cách. Đặc điểm này gây ra nhiều khó khăn cho các bước xử lý văn bản, đặc biệt là quá trình tách từ và trích xuất đặc trưng, do ranh giới giữa các từ không được xác định rõ ràng như trong một số ngôn ngữ khác. Bên cạnh đó, văn bản tiếng Việt trên môi trường Internet thường chứa nhiều yếu tố nhiễu như ký tự đặc biệt, từ viết tắt, teencode và các từ không mang nhiều ý nghĩa ngữ nghĩa. Những yếu tố này làm gia tăng độ phức tạp của dữ liệu đầu vào và có thể ảnh hưởng tiêu cực đến hiệu quả của các phương pháp trích xuất đặc trưng cũng như các mô hình phân loại văn bản. Vì những lý do trên, tiền xử lý văn bản đóng vai trò quan trọng trong việc chuẩn hóa dữ liệu đầu vào, giúp giảm nhiễu và làm nổi bật các thông tin mang tính nội dung. Qua đó, tiền xử lý góp phần nâng cao hiệu quả của các phương pháp trích xuất đặc trưng và cải thiện độ chính xác của các mô hình phân loại văn bản tiếng Việt.

### ***2.2.2. Chuẩn hóa văn bản***

Chuẩn hóa văn bản là bước tiền xử lý nhằm đưa dữ liệu văn bản về một dạng thống nhất trước khi tiến hành các bước xử lý tiếp theo trong pipeline phân loại văn bản. Mục tiêu của bước này là giảm nhiễu, hạn chế sự khác biệt không cần thiết giữa các mẫu dữ liệu và tạo điều kiện thuận lợi cho quá trình trích xuất đặc trưng. Quá trình chuẩn hóa văn bản trong bài toán phân loại văn bản tiếng Việt thường bao gồm các thao tác chính sau:

* Chuyển đổi chữ hoa thành chữ thường, nhằm tránh việc cùng một từ nhưng được biểu diễn dưới nhiều dạng khác nhau, từ đó làm tăng kích thước không gian từ vựng.
* Loại bỏ các ký tự đặc biệt và dấu câu không cần thiết, giúp giảm nhiễu trong dữ liệu, đặc biệt đối với văn bản thu thập từ Internet hoặc mạng xã hội.
* Chuẩn hóa các từ viết tắt và teencode, chuyển chúng về dạng từ đầy đủ để đảm bảo tính nhất quán về mặt ngữ nghĩa giữa các văn bản.

Việc thực hiện chuẩn hóa văn bản giúp giảm số lượng từ vựng không cần thiết, hạn chế sự phân tán của đặc trưng trong không gian biểu diễn và góp phần nâng cao hiệu quả của các phương pháp trích xuất đặc trưng. Qua đó, bước chuẩn hóa đóng vai trò quan trọng trong việc cải thiện độ chính xác và khả năng khái quát của các mô hình phân loại văn bản.

### ***2.2.3. Loại bỏ stopword***

Stopword là những từ xuất hiện với tần suất cao trong văn bản nhưng mang ít hoặc không mang ý nghĩa phân biệt về mặt chủ đề, chẳng hạn như các từ nối, trợ từ hoặc từ chức năng trong tiếng Việt như “và”, “là”, “của”, “đã”,… Những từ này thường không đóng vai trò quan trọng trong việc xác định nội dung chính của văn bản. Việc loại bỏ stopword nhằm giảm nhiễu trong dữ liệu, giúp mô hình tập trung hơn vào các từ mang nhiều thông tin ngữ nghĩa và có khả năng phân biệt chủ đề tốt hơn. Qua đó, quá trình trích xuất đặc trưng trở nên hiệu quả hơn và không gian biểu diễn được thu gọn đáng kể. Trong xử lý văn bản tiếng Việt, danh sách stopword có thể được xây dựng thủ công dựa trên đặc điểm dữ liệu hoặc sử dụng từ các nguồn có sẵn. Sau khi được xác định, danh sách này được áp dụng để lọc bỏ các từ không cần thiết trong văn bản trước khi tiến hành các bước trích xuất đặc trưng và xây dựng mô hình phân loại.

### ***2.2.4. Tách từ (Tokenization)***

Do đặc thù của tiếng Việt không có dấu phân cách từ rõ ràng, trong đó một từ có thể bao gồm nhiều âm tiết được viết tách rời bằng dấu cách, tách từ (tokenization) được xem là một bước bắt buộc trong quá trình tiền xử lý văn bản. Việc tách từ nhằm chuyển đổi chuỗi văn bản thô thành một danh sách các từ hoặc token có ý nghĩa, tạo cơ sở cho các bước trích xuất đặc trưng và xây dựng mô hình phân loại ở các giai đoạn tiếp theo. Tách từ chính xác giúp đảm bảo các từ trong văn bản được nhận diện đúng ranh giới, hạn chế tình trạng một từ bị chia tách sai hoặc nhiều từ bị gộp lại không chính xác. Điều này đặc biệt quan trọng trong bài toán phân loại văn bản, bởi chất lượng của bước tách từ ảnh hưởng trực tiếp đến hiệu quả của các phương pháp biểu diễn văn bản như Bag-of-Words, TF-IDF hay Word2Vec.

Trong xử lý văn bản tiếng Việt, nhiều công cụ tách từ đã được đề xuất và sử dụng rộng rãi, trong đó các công cụ như ViTokenizer thường được áp dụng nhằm đảm bảo độ chính xác trong việc xác định ranh giới từ. Việc lựa chọn công cụ tách từ phù hợp góp phần nâng cao chất lượng dữ liệu đầu vào và cải thiện hiệu quả của các mô hình phân loại văn bản tiếng Việt.

## 2.3. Biểu diễn và trích xuất đặc trưng văn bản:

### ***2.3.1. Khái niệm trích xuất đặc trưng***

Trích xuất đặc trưng (Feature Extraction) là quá trình chuyển đổi dữ liệu văn bản từ dạng ký tự sang dạng số nhằm tạo ra các biểu diễn phù hợp để các mô hình học máy và học sâu có thể xử lý. Trong bài toán phân loại văn bản, trích xuất đặc trưng đóng vai trò then chốt, bởi chất lượng của đặc trưng ảnh hưởng trực tiếp đến khả năng học và hiệu quả dự đoán của mô hình phân loại. Thông qua quá trình trích xuất đặc trưng, nội dung văn bản được biểu diễn dưới dạng các vector số, phản ánh thông tin ngữ nghĩa và thống kê của văn bản. Việc lựa chọn phương pháp trích xuất đặc trưng phù hợp giúp mô hình nắm bắt tốt hơn các đặc điểm quan trọng của văn bản, đồng thời nâng cao độ chính xác và khả năng khái quát của hệ thống phân loại.

Các phương pháp trích xuất đặc trưng văn bản có thể được chia thành hai nhóm chính gồm biểu diễn truyền thống và biểu diễn dựa trên embedding. Nhóm biểu diễn truyền thống thường dựa trên thống kê tần suất xuất hiện của từ, trong khi nhóm biểu diễn dựa trên embedding tập trung vào việc học các biểu diễn liên tục của từ hoặc văn bản trong không gian vector.

### ***2.3.2. Bag of Words và Count Vector***

Bag of Words (BoW) là một trong những phương pháp biểu diễn văn bản đơn giản và phổ biến trong xử lý ngôn ngữ tự nhiên. Theo phương pháp này, mỗi văn bản được xem như một tập hợp các từ xuất hiện trong văn bản đó, không xét đến thứ tự hay mối quan hệ giữa các từ. Văn bản được biểu diễn dưới dạng một vector số, trong đó mỗi chiều tương ứng với một từ trong từ điển và giá trị của chiều đó thể hiện số lần xuất hiện của từ tương ứng trong văn bản.

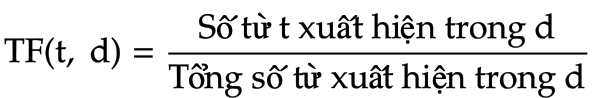
Count Vector là cách cài đặt cụ thể của mô hình Bag of Words, trong đó các phần tử của vector biểu diễn chính xác tần suất xuất hiện của các từ trong văn bản. Phương pháp này có ưu điểm là đơn giản, dễ triển khai và phù hợp với các mô hình học máy truyền thống. Tuy nhiên, Count Vector không phản ánh được mức độ quan trọng tương đối của các từ trong toàn bộ tập dữ liệu, do tất cả các từ đều được xem xét như nhau dựa trên tần suất xuất hiện. Do không xem xét ngữ cảnh và thứ tự từ, Bag of Words và Count Vector thường gặp hạn chế trong việc biểu diễn ý nghĩa sâu của văn bản. Mặc dù vậy, trong nhiều bài toán phân loại văn bản cơ bản, đặc biệt khi kết hợp với các mô hình học máy phù hợp, các phương pháp này vẫn cho kết quả chấp nhận được và đóng vai trò nền tảng cho các phương pháp trích xuất đặc trưng nâng cao hơn.

### ***2.3.3. Mã hóa TF-IDF***

Mặc dù các phương pháp biểu diễn văn bản dựa trên Bag of Words và Count Vector có ưu điểm là đơn giản, trực quan và dễ triển khai trong việc chuyển đổi văn bản sang dạng vector số, tuy nhiên các phương pháp này vẫn tồn tại một số hạn chế. Cụ thể, việc chỉ dựa trên tần suất xuất hiện của từ không phản ánh được mức độ quan trọng tương đối của từ trong từng văn bản cũng như trong toàn bộ tập dữ liệu. Trong bài toán phân loại văn bản theo chủ đề, nhiều từ có thể xuất hiện với tần suất cao ở nhiều chủ đề khác nhau nhưng lại không mang ý nghĩa phân biệt rõ ràng. Ví dụ, các từ như “thông tin”, “bài viết”, “nội dung” có thể xuất hiện phổ biến trong nhiều loại văn bản, khiến mô hình khó xác định được chủ đề đặc trưng nếu chỉ dựa vào tần suất từ.

Để khắc phục nhược điểm đó, một kỹ thuật biểu diễn văn bản tiên tiến hơn đã ra đời: TF–IDF (Term Frequency – Inverse Document Frequency xác định độ hiếm của từ ). Đây là phương pháp kết hợp giữa tần suất xuất hiện của từ trong một văn bản (TF) và độ hiếm của từ trong toàn bộ tập tài liệu (IDF). TF–IDF giúp đo lường mức độ quan trọng của một từ đối với một văn bản cụ thể trong ngữ cảnh của cả bộ dữ liệu. Được truyền cảm hứng từ bài báo của Karen Spärck Jones năm 1972 – người đầu tiên giới thiệu khái niệm IDF – phương pháp TF–IDF đã trở nên phổ biến và được ứng dụng rộng rãi trong nhiều bài toán xử lý ngôn ngữ tự nhiên (NLP), trong đó có bài toán phân loại văn bản tiếng việt. TF – IDF đo lường độ mức độ quan trọng của một từ đối với một mẫu văn bản trong bộ dữ liệu với hai thành phần chính:

TF(Term Frequency) hay tần suất thuật ngữ đo lường độ phổ biến của một từ *t* trong một văn bản *d* thông qua công thức:



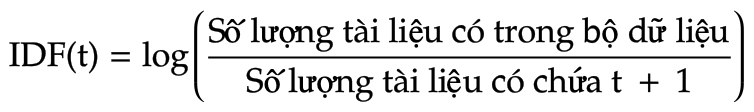
Từ đây ta có thể thấy rằng nếu t xuất hiện nhiều lần trong d thì TF(t, d) sẽ lớn.

Ví dụ, bộ dữ liệu như sau:

| 𝑡 (từ) | con | mèo | ngồi | trên | thảm | tôi | nhìn | bầu | trời | ghế |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0,25 | 0,25 | 0,25 | 0,25 | 0 |
|  | 0 | 0 | 0,25 | 0,25 | 0 | 0,25 | 0 | 0 | 0 | 0,25 |

#### **Bảng 2.1.** Giá trị TF của các văn bản trong bộ dữ liệu

IDF(Inverse Document Frequency) đo lường độ hiếm của một từ t trong bộ dữ liệu được tính bởi công thức:



Như vậy ta chỉ cần tính IDF của các từ có trong bộ từ điển của chúng ta:

| 𝑡 (từ) | con | mèo | ngồi | trên | thảm | tôi | nhìn | bầu | trời | ghế |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| df(t) | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 1 |
|  | 0,18 | 0,18 | 0 | 0 | 0,18 | 0 | 0,18 | 0,18 | 0,18 | 0,18 |

#### **Bảng 2.2.** Giá trị IDF của các từ trong bộ từ điển

Và rồi, ta chỉ cần lấy tích của và 𝐼𝐷𝐹(𝑡) thì sẽ được của các đặc trưng ở vector tương ứng:

| 𝑡 (từ) | con | mèo | ngồi | trên | thảm | tôi | nhìn | bầu | trời | ghế |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0,04 | 0,04 | 0 | 0 | 0,04 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0,05 | 0,05 | 0,05 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,05 |

### **Bảng 2.3.** Mã hóa TF-IDF của bộ dữ liệu

Từ bảng trên có thể nhận thấy rằng, những từ xuất hiện phổ biến trong nhiều văn bản sẽ có giá trị IDF thấp, dẫn đến giá trị TF-IDF tương ứng cũng thấp; ngược lại, các từ chỉ xuất hiện trong một số ít văn bản sẽ có giá trị IDF cao hơn và do đó có TF-IDF lớn hơn. Cơ chế này giúp giảm ảnh hưởng của các từ mang tính phổ biến chung trong tập dữ liệu, đồng thời làm nổi bật các từ đặc trưng cho từng chủ đề văn bản. Nhờ vậy, phương pháp TF-IDF góp phần cải thiện khả năng phân biệt giữa các chủ đề khác nhau, hỗ trợ hiệu quả cho các mô hình học máy trong bài toán phân loại văn bản tiếng Việt theo chủ đề.

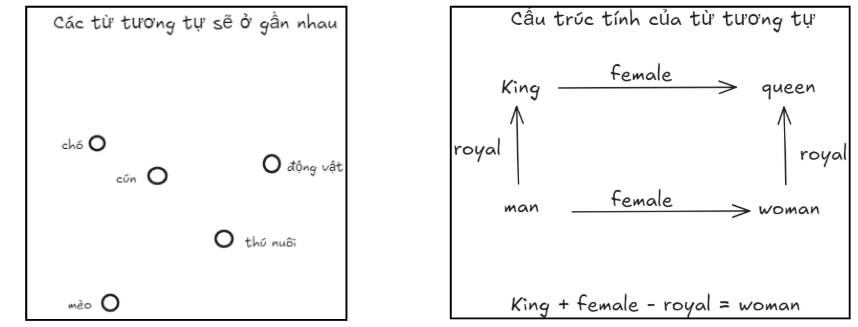
### ***2.3.4. N-gram***

Trong các phương pháp mã hóa văn bản như TF-IDF, mỗi văn bản thường được biểu diễn dưới dạng vector dựa trên tần suất và mức độ quan trọng của các từ đơn lẻ (unigram). Cách tiếp cận này giúp mô hình nắm bắt được sự xuất hiện của từng từ trong văn bản, tuy nhiên lại không xét đến thứ tự và mối quan hệ giữa các từ. Do đó, trong một số trường hợp, các văn bản có cùng tập từ nhưng khác nhau về cấu trúc và ngữ cảnh có thể được biểu diễn tương tự nhau. Ví dụ, hai câu tiếng Việt “học máy ứng dụng” và “ứng dụng học máy” đều chứa cùng các từ “học”, “máy”, “ứng dụng”, nhưng ý nghĩa và trọng tâm biểu đạt có sự khác biệt. Khi sử dụng unigram, sự khác biệt này khó được phản ánh đầy đủ trong quá trình biểu diễn đặc trưng. Để khắc phục hạn chế trên, N-gram được sử dụng như một kỹ thuật mở rộng trong biểu diễn văn bản. N-gram biểu diễn văn bản bằng các chuỗi gồm n từ liên tiếp xuất hiện trong văn bản.Với n=2 (bigram), câu “học máy ứng dụng” sẽ được biểu diễn bởi các cụm từ như “học máy” và “máy ứng dụng”. Việc xét đến các cụm từ liên tiếp giúp mô hình khai thác tốt hơn thông tin về thứ tự từ và ngữ cảnh cục bộ.

Trong báo cáo này, N-gram được kết hợp với phương pháp TF-IDF nhằm tăng cường khả năng biểu diễn các cụm từ mang tính chủ đề rõ ràng, từ đó cải thiện hiệu quả của các mô hình học máy trong bài toán phân loại văn bản tiếng Việt theo chủ đề.

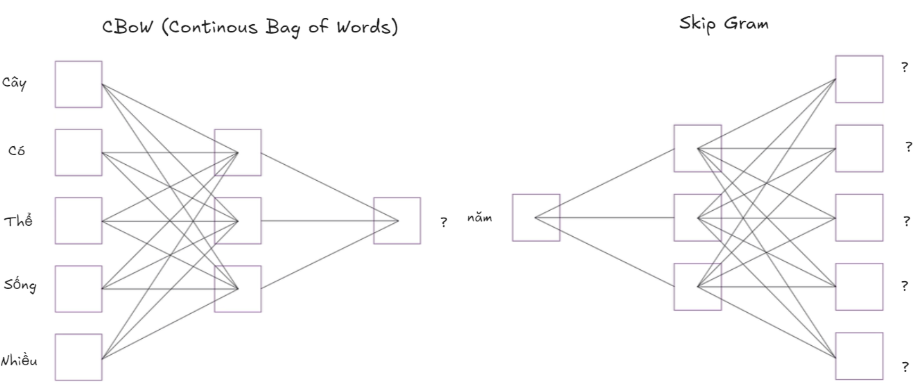
### ***2.3.5. Static Embedding – Word2Vec***

Word2Vec là một phương pháp biểu diễn từ dưới dạng vector liên tục trong không gian nhiều chiều, trong đó các từ có ngữ nghĩa tương tự sẽ được ánh xạ tới các vector có khoảng cách gần nhau. Phương pháp này được đề xuất bởi Mikolov và cộng sự, dựa trên việc sử dụng mạng nơ-ron nông để học mối quan hệ ngữ cảnh giữa các từ xuất hiện trong tập dữ liệu huấn luyện. Thay vì dựa vào tần suất xuất hiện của từ như các phương pháp truyền thống, Word2Vec tập trung khai thác thông tin ngữ nghĩa thông qua bối cảnh xuất hiện của từ trong văn bản.



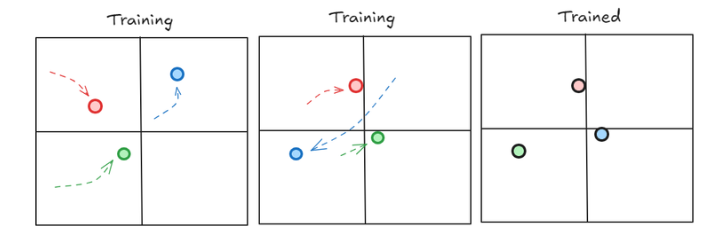
## Hình 2.2. Minh họa embedding space và mối quan hệ ngữ nghĩa giữa các từ

Word2Vec bao gồm hai kiến trúc phổ biến là Continuous Bag of Words CBOW và Skip-gram. Trong đó, CBOW dự đoán một từ dựa trên các từ xung quanh, còn Skip-gram dự đoán các từ ngữ cảnh dựa trên từ trung tâm. Nhờ cơ chế học này, Word2Vec có khả năng nắm bắt được các mối quan hệ ngữ nghĩa và cú pháp giữa các từ, chẳng hạn như các từ đồng nghĩa hoặc các từ thường xuất hiện trong cùng một ngữ cảnh. Trong bài toán phân loại văn bản, vector biểu diễn cho một văn bản có thể được xây dựng bằng cách tổng hợp các vector Word2Vec của các từ xuất hiện trong văn bản, phổ biến nhất là phép lấy trung bình các vector từ. Cách tiếp cận này cho phép biểu diễn toàn bộ văn bản dưới dạng một vector cố định chiều, thuận tiện cho việc đưa vào các mô hình học máy truyền thống như SVM, Logistic Regression.



## Hình 2.3. Minh họa kiến trúc Word2Vec gồm mô hình CBOW và Skip-gram

So với các phương pháp biểu diễn dựa trên tần suất như Bag of Words hoặc TF-IDF, Word2Vec có ưu điểm trong việc nắm bắt thông tin ngữ nghĩa của từ và giảm ảnh hưởng của sự khác biệt về hình thức từ vựng. Tuy nhiên, do là phương pháp embedding tĩnh, mỗi từ chỉ có một vector biểu diễn duy nhất, Word2Vec không phản ánh được sự thay đổi ngữ nghĩa của từ trong các ngữ cảnh khác nhau.



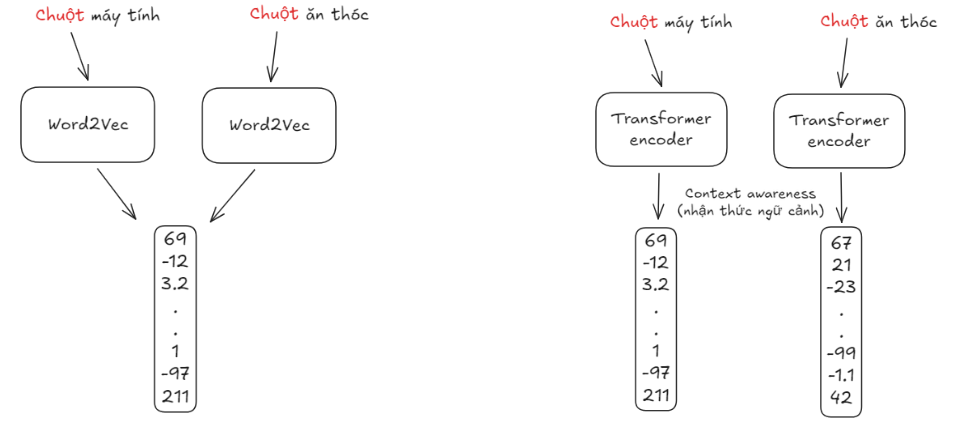
## Hình 2.4. Minh họa việc Word2Vec gán mỗi từ một vector cố định

Do đó, trong các trường hợp một từ xuất hiện trong nhiều ngữ cảnh khác nhau, Word2Vec không phản ánh được sự thay đổi về mặt ngữ nghĩa của từ theo từng ngữ cảnh cụ thể. Đây là một trong những hạn chế của phương pháp embedding tĩnh, ảnh hưởng đến khả năng biểu diễn ngữ nghĩa trong các bài toán xử lý ngôn ngữ tự nhiên phức tạp.

Trong dự án phân loại văn bản tiếng Việt theo chủ đề, Word2Vec được sử dụng dưới dạng pretrained, tức là mô hình embedding đã được huấn luyện sẵn trên tập dữ liệu tiếng Việt có quy mô lớn. Việc sử dụng Word2Vec pretrained giúp tận dụng tri thức ngữ nghĩa đã học được từ một corpus lớn, từ đó cải thiện chất lượng biểu diễn từ so với việc huấn luyện Word2Vec trực tiếp trên tập dữ liệu giới hạn. Đây là một lựa chọn phù hợp trong bối cảnh dữ liệu huấn luyện của dự án không đủ lớn để xây dựng embedding chất lượng cao từ đầu.

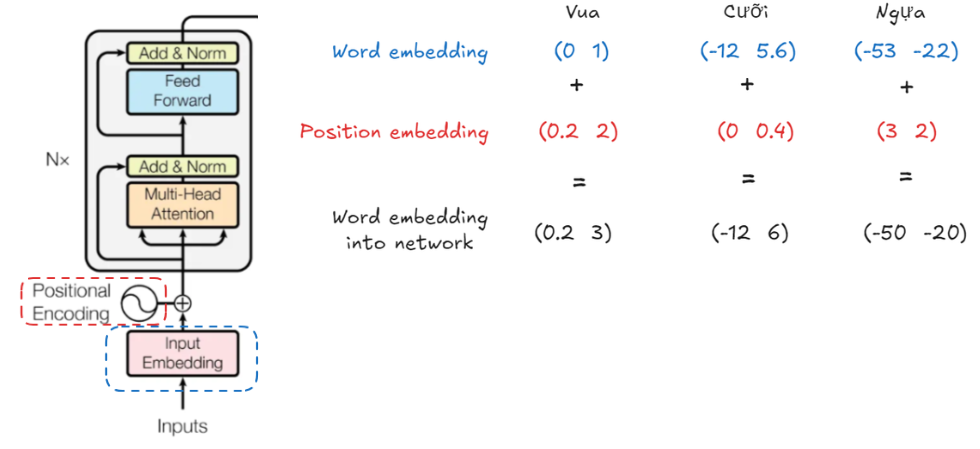
### ***2.3.6. Contextual Embedding – Transformer***

Khác với các phương pháp embedding tĩnh như Word2Vec, contextual embedding tạo ra vector biểu diễn của từ phụ thuộc trực tiếp vào ngữ cảnh xuất hiện trong câu. Điều này có nghĩa là cùng một từ nhưng xuất hiện trong các ngữ cảnh khác nhau sẽ được biểu diễn bằng các vector khác nhau, giúp mô hình nắm bắt chính xác hơn ý nghĩa thực sự của từ trong từng trường hợp cụ thể.



## Hình 2.5. khác biệt giữa embedding tĩnh và embedding ngữ cảnh

Các phương pháp contextual embedding hiện đại chủ yếu dựa trên kiến trúc Transformer, trong đó cơ chế self-attention và multi-head attention cho phép mô hình học được mối quan hệ ngữ cảnh giữa các từ trong toàn bộ câu, thay vì chỉ dựa trên cửa sổ ngữ cảnh cục bộ như các mô hình truyền thống. Nhờ cơ chế attention song song, Transformer có khả năng tập trung vào nhiều khía cạnh ngữ nghĩa khác nhau của câu, từ đó khai thác đồng thời thông tin về ngữ nghĩa, cú pháp và mối quan hệ dài hạn giữa các từ trong văn bản. Bên cạnh đó, do Transformer không sử dụng cấu trúc tuần tự như RNN, Positional Encoding được đưa vào nhằm bổ sung thông tin về vị trí của từ trong câu. Vector biểu diễn đầu vào của mỗi từ được tạo thành bằng cách kết hợp giữa word embedding và positional encoding, giúp mô hình vừa hiểu được nội dung từ vựng vừa nhận biết được thứ tự các từ trong chuỗi văn bản. Trong lĩnh vực xử lý ngôn ngữ tự nhiên, contextual embedding đã chứng minh hiệu quả vượt trội trong nhiều bài toán như phân loại văn bản, phân tích cảm xúc, trích xuất thực thể và hỏi đáp. Đối với bài toán phân loại văn bản tiếng Việt theo chủ đề, việc sử dụng contextual embedding giúp mô hình phân biệt tốt hơn các văn bản có nội dung và từ vựng tương đồng nhưng khác biệt về ngữ cảnh và chủ đề, đặc biệt trong các trường hợp từ ngữ mang nhiều nghĩa hoặc phụ thuộc mạnh vào ngữ cảnh sử dụng.



## Hình 2.6. Minh họa cơ chế Positional Encoding trong kiến trúc Transformer



## Hình 2.7. Minh họa cơ chế Multihead attention trong kiến trúc Transformer

Tuy nhiên, bên cạnh những ưu điểm về mặt biểu diễn ngữ nghĩa, các mô hình contextual embedding thường đòi hỏi tài nguyên tính toán lớn, thời gian huấn luyện dài và quá trình triển khai phức tạp hơn so với các phương pháp truyền thống. Do đó, trong thực tế, việc lựa chọn giữa contextual embedding và các phương pháp biểu diễn đơn giản hơn cần cân nhắc đến yêu cầu về độ chính xác, tài nguyên phần cứng và mục tiêu ứng dụng cụ thể của bài toán.

### ***2.3.7. Giảm chiều dữ liệu (SVD)***

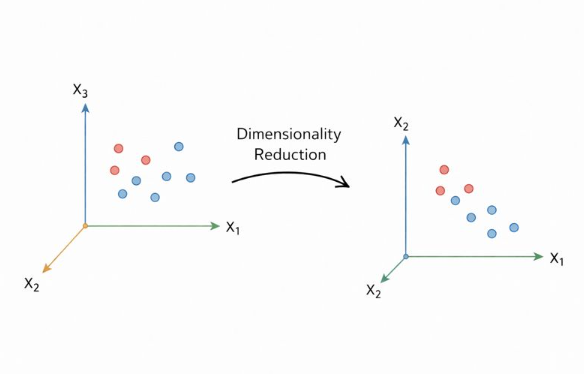
Sau bước trích xuất đặc trưng, dữ liệu văn bản được biểu diễn dưới dạng các ma trận đặc trưng và được sử dụng làm đầu vào cho các mô hình phân loại. Trong các phương pháp biểu diễn văn bản dựa trên tần suất từ như Bag of Words (BoW) và TF-IDF, dữ liệu thường có số chiều rất lớn do phụ thuộc vào kích thước từ điển. Khi đó, mỗi văn bản được biểu diễn dưới dạng một vector trong không gian đặc trưng có số chiều cao, dẫn đến chi phí tính toán lớn và có thể ảnh hưởng tiêu cực đến hiệu quả huấn luyện mô hình.

Trong trường hợp tổng quát, dữ liệu đặc trưng có thể được biểu diễn dưới dạng một ma trận:

Trong đó:

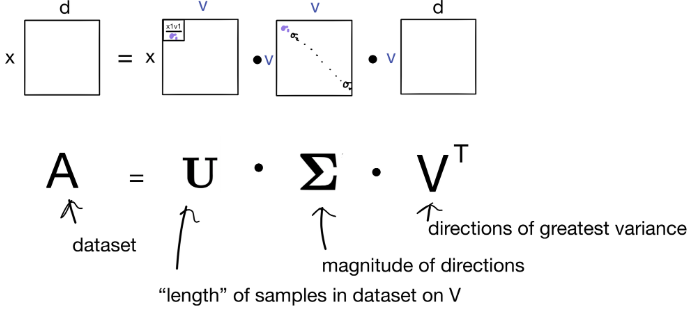
* là số lượng văn bản trong tập dữ liệu,
* là số lượng đặc trưng, tương ứng với số từ duy nhất trong toàn bộ tập dữ liệu hoặc kích thước của từ điển.

Mỗi hàng của ma trận tương ứng với một văn bản và được biểu diễn bằng một vector đặc trưng . Trong thực tế, giá trị m có thể lên tới hàng chục nghìn hoặc hơn khi xử lý các tập dữ liệu văn bản lớn, gây ra hiện tượng “lời nguyền số chiều” (curse of dimensionality). Điều này không chỉ làm tăng chi phí lưu trữ và tính toán mà còn có thể làm giảm khả năng tổng quát hóa của các mô hình học máy.



## Hình 2.8. Giảm chiều dữ liệu (Dimensionality Reduction)

Để khắc phục vấn đề trên, giảm chiều dữ liệu là một bước quan trọng nhằm giảm số chiều của không gian đặc trưng trong khi vẫn giữ lại phần lớn thông tin quan trọng của dữ liệu ban đầu. Một số kỹ thuật giảm chiều phổ biến bao gồm Principal Component Analysis (PCA), Latent Semantic Analysis (LSA) và t-distributed Stochastic Neighbor Embedding (t-SNE). Trong đó, PCA và LSA là các phương pháp giảm chiều tuyến tính, còn t-SNE chủ yếu được sử dụng cho mục đích trực quan hóa dữ liệu trong không gian có số chiều thấp. Đối với bài toán phân loại văn bản, đặc biệt khi sử dụng các ma trận đặc trưng thưa như TF-IDF, phương pháp LSA – thường được triển khai thông qua Truncated Singular Value Decomposition (SVD) – được xem là phù hợp và hiệu quả. Truncated SVD cho phép phân rã ma trận đặc trưng ban đầu thành các thành phần có ý nghĩa ngữ nghĩa tiềm ẩn, từ đó giảm số chiều của dữ liệu trong khi vẫn bảo toàn được các thông tin quan trọng liên quan đến nội dung văn bản. Trong bài toán này, kỹ thuật Truncated SVD được áp dụng để giảm chiều cho ma trận đặc trưng TF-IDF trước khi đưa vào huấn luyện các mô hình học máy truyền thống. Việc giảm chiều dữ liệu giúp giảm đáng kể độ phức tạp tính toán, tăng tốc độ huấn luyện và góp phần cải thiện khả năng tổng quát của mô hình phân loại.



## Hình 2.9. Nguyên lý phân rã ma trận trong SVD

Đối với các phương pháp biểu diễn văn bản dựa trên embedding liên tục như Word2Vec, mỗi từ đã được biểu diễn dưới dạng vector có số chiều cố định và tương đối nhỏ. Do đó, trong nghiên cứu này, bước giảm chiều dữ liệu không được áp dụng cho các vector Word2Vec nhằm bảo toàn đầy đủ thông tin ngữ nghĩa của văn bản.

Tóm lại, giảm chiều dữ liệu đóng vai trò quan trọng trong việc xử lý các đặc trưng văn bản có số chiều lớn, đặc biệt là đối với các phương pháp biểu diễn dựa trên TF-IDF. Việc lựa chọn kỹ thuật giảm chiều phù hợp giúp cân bằng giữa hiệu quả tính toán và khả năng biểu diễn ngữ nghĩa, từ đó nâng cao hiệu quả của các mô hình phân loại văn bản.

## 2.4. Các mô hình học máy (Machine learning):

### ***2.4.1. Thuật toán Logistic Regression:***

Logistic Regression là một thuật toán học máy có giám sát (supervised learning) được nhà thống kê học David Cox giới thiệu và phát triển vào năm 1958, được sử dụng rộng rãi trong các bài toán phân loại, đặc biệt là phân loại nhị phân. Mặc dù có tên gọi là “regression”, Logistic Regression thực chất là một mô hình phân loại, trong đó đầu ra của mô hình biểu diễn xác suất để một mẫu dữ liệu thuộc về một lớp nhất định. Cơ chế hoạt động của Logistic Regression dựa trên việc kết hợp tuyến tính các đặc trưng đầu vào, sau đó đưa giá trị này qua hàm kích hoạt sigmoid nhằm ánh xạ kết quả về khoảng [0,1]. Giá trị thu được có thể được hiểu là xác suất để văn bản thuộc về một lớp cụ thể. Dựa trên xác suất này và một ngưỡng phân loại (thường được lựa chọn là 0.5), mô hình sẽ quyết định gán nhãn cho văn bản.

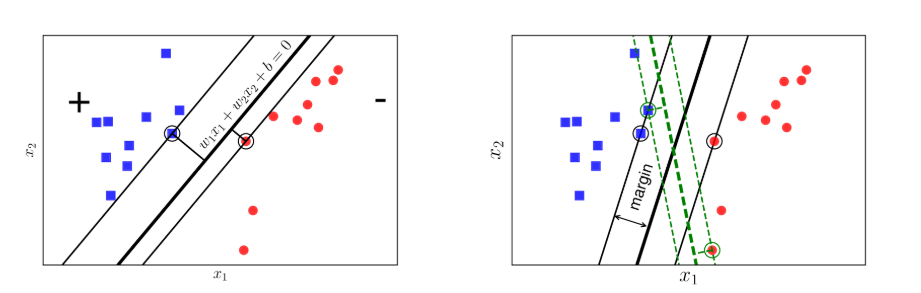


## Hình 2.10. Hồi quy Logistic: biến đổi đầu ra tuyến tính thành xác suất bằng hàm sigmoid.

Trong bài toán phân loại văn bản tiếng Việt theo chủ đề, Logistic Regression thường được kết hợp với các phương pháp trích xuất đặc trưng như Bag of Words hoặc TF-IDF, Word2Vec. Sau khi văn bản được biểu diễn dưới dạng vector, mô hình học cách phân biệt các chủ đề dựa trên sự phân bố và mức độ quan trọng của các từ trong từng văn bản. Do các vector đặc trưng văn bản thường có số chiều lớn và không tuân theo phân phối chuẩn, Logistic Regression tỏ ra phù hợp vì không yêu cầu giả định chặt chẽ về phân phối dữ liệu đầu vào. Ưu điểm của Logistic Regression là cấu trúc đơn giản, tốc độ huấn luyện nhanh, dễ triển khai và có khả năng mở rộng tốt với các tập dữ liệu lớn. Tuy nhiên, do là mô hình tuyến tính, Logistic Regression có hạn chế trong việc nắm bắt các mối quan hệ phi tuyến và ngữ cảnh phức tạp trong văn bản tiếng Việt. Vì vậy, trong nghiên cứu phân loại văn bản theo chủ đề, Logistic Regression thường được sử dụng như một mô hình nền (baseline) để so sánh với các phương pháp học máy và học sâu nâng cao hơn.

### ***2.4.2. Thuật toán Support vector machine:***

Support Vector Machine (SVM) là một trong những thuật toán học máy có giám sát mạnh mẽ và được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Thuật toán này được được phát triển bởi Vapnik và cộng sự từ những năm 1990 và đến nay vẫn giữ vai trò quan trọng trong nhiều hệ thống học máy nhờ hiệu quả và độ ổn định cao. Nguyên lý hoạt động của SVM dựa trên việc tìm kiếm một siêu phẳng phân tách (hyperplane) tối ưu trong không gian đặc trưng nhằm phân biệt các lớp dữ liệu khác nhau. Siêu phẳng tối ưu được xác định sao cho khoảng cách biên (margin) giữa siêu phẳng và các điểm dữ liệu gần nhất của mỗi lớp là lớn nhất. Những điểm dữ liệu nằm gần siêu phẳng và ảnh hưởng trực tiếp đến vị trí của nó được gọi là các vector hỗ trợ (support vectors).



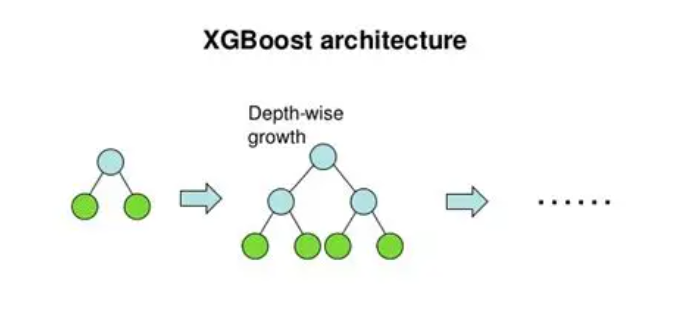
## Hình 2.11. Siêu phẳng phân tách và khoảng cách biên (margin) trong SVM

Trong bài toán phân loại văn bản tiếng Việt, SVM đặc biệt phù hợp khi dữ liệu văn bản đã được biểu diễn dưới dạng vector đặc trưng như TF-IDF. Các vector TF-IDF thường có số chiều rất lớn và phân bố thưa, gây khó khăn cho nhiều mô hình học máy truyền thống. Tuy nhiên, SVM vẫn hoạt động hiệu quả trong không gian nhiều chiều nhờ cơ chế tối ưu hóa dựa trên các vector hỗ trợ thay vì toàn bộ dữ liệu. Bên cạnh đó, thông qua việc sử dụng các hàm kernel, SVM có khả năng mở rộng sang các bài toán phân loại không tuyến tính, giúp mô hình học được các ranh giới phân tách phức tạp giữa các lớp văn bản. Nhờ những ưu điểm này, SVM thường cho kết quả tốt trong các bài toán phân loại văn bản theo chủ đề, đặc biệt khi số lượng đặc trưng lớn nhưng số lượng mẫu huấn luyện không quá nhiều.

Tuy nhiên, SVM cũng tồn tại một số hạn chế như chi phí tính toán cao khi huấn luyện trên tập dữ liệu rất lớn và việc lựa chọn kernel cùng các siêu tham số phù hợp có thể ảnh hưởng đáng kể đến hiệu quả của mô hình. SVM thường được kết hợp với các phương pháp trích xuất đặc trưng như TF-IDF để chuyển đổi văn bản sang dạng vector số. Việc kết hợp SVM với các đặc trưng này giúp mô hình khai thác hiệu quả thông tin từ vựng và phân biệt các chủ đề khác nhau trong không gian đặc trưng nhiều chiều.

### ***2.4.3. Thuật toán XGBoost:***

XGBoost (eXtreme Gradient Boosting) là một thuật toán học máy mạnh mẽ dựa trên kỹ thuật cây quyết định tăng cường (Gradient Boosting) được phát triển bởi Tianqi Chen và các cộng sự. Thuật toán này xây dựng mô hình bằng cách kết hợp nhiều cây quyết định yếu (weak learners) theo cách tuần tự, trong đó mỗi cây mới được huấn luyện nhằm khắc phục các sai sót của các cây trước đó. Nhờ cơ chế tối ưu hóa dựa trên gradient, XGBoost có khả năng học được các mối quan hệ phi tuyến phức tạp trong dữ liệu.



## Hình 2.12. Kiến trúc XGBoost với chiến lược phát triển theo chiều sâu.

Trong bài toán phân loại văn bản, XGBoost thường được áp dụng trên các vector đặc trưng thu được từ các phương pháp trích xuất đặc trưng như TF-IDF. Tuy nhiên, do đặc trưng TF-IDF thường có số chiều rất lớn và phân bố thưa, việc sử dụng XGBoost trực tiếp trên không gian đặc trưng đầy đủ có thể dẫn đến chi phí tính toán cao và nguy cơ quá khớp (overfitting). Vì vậy, XGBoost thường cho hiệu quả tốt hơn khi được kết hợp với các kỹ thuật giảm chiều dữ liệu, chẳng hạn như Truncated Singular Value Decomposition (SVD), nhằm giữ lại các thông tin quan trọng đồng thời giảm độ phức tạp của mô hình. Ưu điểm nổi bật của XGBoost là khả năng mô hình hóa tốt các mối quan hệ phi tuyến, khả năng tổng quát hóa cao và hiệu suất tốt trên nhiều loại dữ liệu khác nhau. Bên cạnh đó, XGBoost cung cấp nhiều cơ chế kiểm soát quá khớp như regularization và early stopping, giúp cải thiện độ ổn định của mô hình trong quá trình huấn luyện.



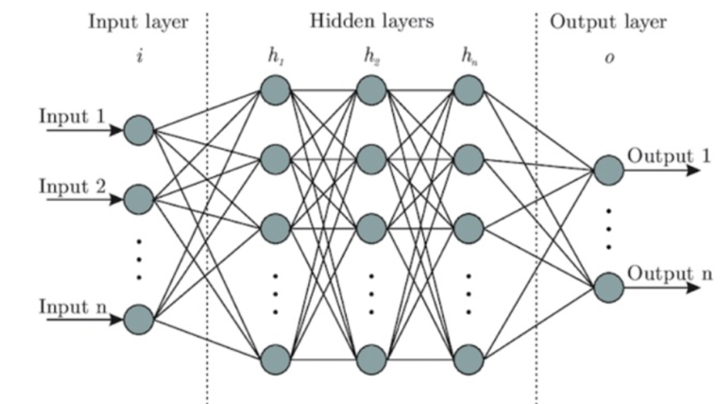
## Hình 2.13. Cấu trúc đơn giản hóa của mô hình XGBoost

Tuy nhiên, XGBoost cũng tồn tại một số hạn chế như yêu cầu tài nguyên tính toán cao hơn so với các mô hình tuyến tính và việc lựa chọn siêu tham số phù hợp có thể ảnh hưởng đáng kể đến hiệu quả của mô hình. Do đó, trong khóa luận này, XGBoost được sử dụng như một mô hình nâng cao nhằm đánh giá khả năng khai thác các đặc trưng văn bản đã được xử lý và so sánh với các phương pháp học máy khác trong bài toán phân loại văn bản tiếng Việt theo chủ đề.

## 2.5. Các mô hình học sâu (Deep learning):

### ***2.5.1. Thuật toán Deep Neural Network (DNN):***

Deep Neural Network (DNN) là một mô hình học sâu gồm nhiều lớp ẩn (hidden layers), có khả năng học và biểu diễn các mối quan hệ phức tạp trong dữ liệu thông qua các phép biến đổi phi tuyến. Không giống như các mô hình học máy truyền thống dựa trên giả định tuyến tính, DNN cho phép mô hình hóa các đặc trưng trừu tượng ở nhiều mức độ khác nhau, từ đó nâng cao khả năng phân biệt giữa các lớp dữ liệu.



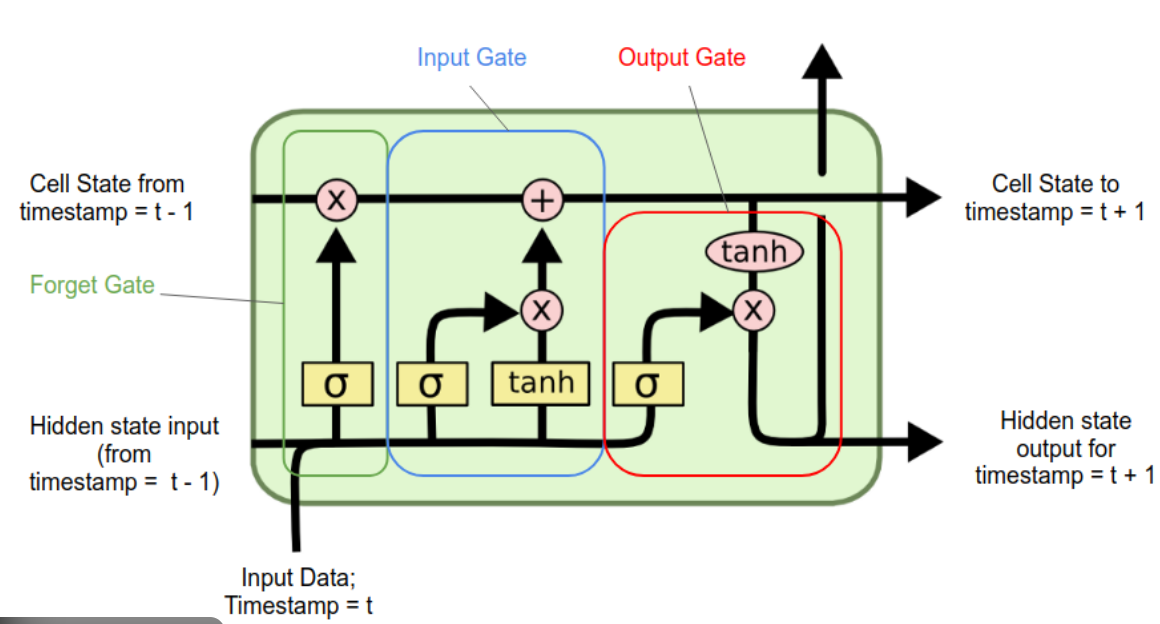
## Hình 2.14. Kiến trúc mạng neural sâu (Deep Neural Network – DNN)

Trong bài toán phân loại văn bản tiếng Việt, DNN thường được áp dụng sau khi văn bản đã được chuyển đổi sang dạng vector số thông qua các phương pháp trích xuất đặc trưng. Các vector đặc trưng phổ biến có thể kể đến như TF-IDF (thường kết hợp với kỹ thuật giảm chiều dữ liệu) và các embedding liên tục như Word2Vec. Những biểu diễn này đóng vai trò là đầu vào cho mạng nơ-ron, giúp DNN học được các mẫu đặc trưng mang tính ngữ nghĩa trong văn bản. Ưu điểm nổi bật của DNN là khả năng khai thác tốt các mối quan hệ phi tuyến và học được các biểu diễn đặc trưng giàu thông tin hơn so với các mô hình tuyến tính. Khi được cung cấp đủ dữ liệu và đặc trưng phù hợp, DNN có thể đạt hiệu quả cao trong các bài toán phân loại văn bản, đặc biệt là những bài toán có độ phức tạp lớn hoặc số lượng lớp nhiều.

Tuy nhiên, DNN cũng tồn tại một số hạn chế, bao gồm yêu cầu tài nguyên tính toán cao, thời gian huấn luyện dài và nguy cơ quá khớp nếu dữ liệu huấn luyện không đủ lớn hoặc không đa dạng. Ngoài ra, việc lựa chọn kiến trúc mạng và các siêu tham số phù hợp có ảnh hưởng đáng kể đến hiệu quả của mô hình. Do đó, trong khóa luận này, DNN được sử dụng như một phương pháp học sâu nhằm đánh giá khả năng cải thiện hiệu quả phân loại văn bản tiếng Việt so với các mô hình học máy truyền thống.

### ***2.5.2. Thuật toán Long Short-Term Memory (LSTM):***

Long Short-Term Memory (LSTM) là một biến thể của mạng nơ-ron hồi tiếp (Recurrent Neural Network – RNN), phát triển bởi Sepp Hochreiter và Jürgen Schmidhuber, lần đầu công bố vào năm 1997, được thiết kế nhằm khắc phục hạn chế của RNN truyền thống trong việc ghi nhớ và khai thác các phụ thuộc dài hạn trong dữ liệu chuỗi. Thông qua cơ chế cổng (gates), LSTM có khả năng kiểm soát dòng chảy thông tin, từ đó giữ lại các thông tin quan trọng và loại bỏ các thông tin không cần thiết trong quá trình xử lý chuỗi.



## Hình 2.15. Kiến trúc ô nhớ LSTM (Long Short-Term Memory).

Trong bài toán xử lý ngôn ngữ tự nhiên, LSTM đặc biệt phù hợp để mô hình hóa dữ liệu văn bản, nơi mà thứ tự từ và ngữ cảnh đóng vai trò quan trọng trong việc xác định ý nghĩa. Mô hình này cho phép khai thác mối quan hệ tuần tự giữa các từ trong câu, giúp nắm bắt tốt hơn ngữ cảnh và cấu trúc ngôn ngữ so với các mô hình học máy truyền thống dựa trên giả định độc lập giữa các từ. Đối với bài toán phân loại văn bản tiếng Việt theo chủ đề, LSTM thường được sử dụng kết hợp với các biểu diễn từ liên tục như Word2Vec hoặc embedding được học trong quá trình huấn luyện. Các vector từ này đóng vai trò là đầu vào cho mạng LSTM, giúp mô hình học được các đặc trưng ngữ nghĩa và ngữ cảnh của văn bản theo trình tự thời gian. Ưu điểm nổi bật của LSTM là khả năng xử lý hiệu quả các phụ thuộc dài hạn và biểu diễn ngữ cảnh chuỗi, từ đó cải thiện độ chính xác trong các bài toán phân loại văn bản phức tạp.

Tuy nhiên, LSTM cũng đòi hỏi tài nguyên tính toán lớn, thời gian huấn luyện dài và cần tập dữ liệu đủ lớn để phát huy hiệu quả. Ngoài ra, việc lựa chọn kiến trúc mạng và các siêu tham số phù hợp có ảnh hưởng đáng kể đến kết quả của mô hình. Trong khuôn khổ báo cáo này, LSTM được xem xét như một phương pháp học sâu nhằm đánh giá khả năng khai thác thông tin ngữ cảnh chuỗi của văn bản tiếng Việt và so sánh hiệu quả với các mô hình học máy truyền thống cũng như các mô hình học sâu khác.

## 2.6. Các chỉ số đánh giá hiệu suất mô hình:

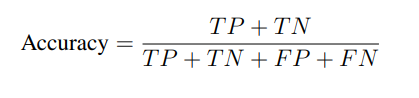
Để đánh giá hiệu quả của các mô hình học máy truyền thống và học sâu trong bài toán phân loại văn bản tiếng Việt theo chủ đề, nghiên cứu này sử dụng một tập hợp các chỉ số đánh giá tiêu chuẩn trong bài toán phân loại. Các chỉ số này cho phép đánh giá toàn diện khả năng của mô hình trong việc gán đúng nhãn chủ đề cho văn bản, đồng thời phản ánh mức độ cân bằng giữa các lớp trong bối cảnh dữ liệu có thể không phân bố đồng đều gây mất cân bằng. Việc sử dụng đồng thời nhiều chỉ số đánh giá giúp hạn chế những đánh giá phiến diện khi chỉ dựa vào một thước đo duy nhất, từ đó hỗ trợ lựa chọn mô hình có hiệu năng tổng thể tốt và khả năng tổng quát hóa cao.

**Ma trận nhầm lẫn (confusion matrix)** là công cụ cơ bản dùng để mô tả mối quan hệ giữa nhãn thực tế và nhãn dự đoán của mô hình trong bài toán phân loại. Thông qua ma trận này, có thể xác định số lượng các trường hợp dự đoán đúng và sai cho từng lớp, làm cơ sở để tính toán các chỉ số đánh giá hiệu suất.

| **Confusion Matrix** | | **Actual Positive** | |
| --- | --- | --- | --- |
| **Positive** | **Negative** |
| Predicted  Label | Positive | TP  True Positive | FP  False Positive |
| Negative | FN  False Negative | TN  True Negative |

#### **Bảng 2.4**. Ma trận nhầm lẫn mô tả kết quả dự đoán của mô hình so với nhãn thực tế.

**Độ chính xác (Accuracy):** Là tỷ lệ phần trăm các dự đoán đúng trên tổng số mẫu dữ liệu. Chỉ số này cho biết mô hình phân loại văn bản dự đoán chính xác bao nhiêu phần trăm văn bản trong toàn bộ tập dữ liệu, bao gồm các dự đoán đúng đối với tất cả các chủ đề khác nhau. Accuracy phản ánh mức độ chính xác tổng thể của mô hình trong việc gán nhãn chủ đề cho các văn bản tiếng Việt.

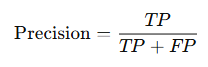


Trong đó:

* TP (True Positive): Số văn bản thuộc một chủ đề cụ thể và được mô hình dự đoán đúng là chủ đề đó.
* TN (True Negative): Số văn bản không thuộc chủ đề đang xét và được mô hình dự đoán đúng.
* FP (False Positive): Số văn bản không thuộc chủ đề đang xét nhưng bị mô hình dự đoán nhầm là thuộc chủ đề đó.
* FN (False Negative): Số văn bản thuộc chủ đề đang xét nhưng bị mô hình dự đoán nhầm sang chủ đề khác.

Mặc dù trực quan và dễ hiểu, Accuracy có thể đánh giá sai lệch khi dữ liệu giữa các chủ đề không cân bằng, ví dụ khi một số chủ đề có số lượng văn bản lớn hơn đáng kể so với các chủ đề còn lại. Trong trường hợp này, mô hình có thể đạt Accuracy cao nhưng vẫn phân loại kém đối với các chủ đề có ít dữ liệu, do đó cần kết hợp Accuracy với các chỉ số khác như Precision, Recall và F1-score để đánh giá toàn diện hiệu năng của mô hình.

**Độ chính xác dương tính (Precision):** Đo lường tỷ lệ trong số các văn bản mà mô hình dự đoán thuộc một chủ đề cụ thể, có bao nhiêu văn bản thực sự thuộc chủ đề đó. Chỉ số này phản ánh độ tin cậy của mô hình khi gán nhãn chủ đề cho văn bản:



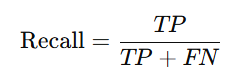
Trong đó:

* TP (True Positive): Số văn bản thuộc chủ đề đang xét và được mô hình dự đoán đúng là chủ đề đó.
* FP (False Positive): Số văn bản không thuộc chủ đề đang xét nhưng bị mô hình dự đoán nhầm là thuộc chủ đề đó.

Precision cao cho thấy khi mô hình dự đoán một văn bản thuộc một chủ đề nhất định, khả năng dự đoán đó là chính xác là cao, đồng thời hạn chế hiện tượng nhầm lẫn văn bản từ các chủ đề khác. Trong bài toán phân loại văn bản theo chủ đề, Precision đặc biệt quan trọng đối với các chủ đề có nội dung dễ chồng chéo, bởi việc gán nhãn sai có thể làm giảm độ tin cậy của hệ thống phân loại.

Tuy nhiên, nếu chỉ tối ưu Precision mà Recall thấp, mô hình có thể bỏ sót nhiều văn bản thực sự thuộc chủ đề đó (FN), làm suy giảm khả năng bao phủ của mô hình đối với tập dữ liệu. Do đó, Precision cần được xem xét đồng thời với Recall và F1-score để đánh giá toàn diện hiệu suất của mô hình.

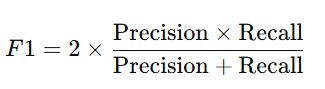
**Độ nhạy (Recall)** còn gọi là tỷ lệ phát hiện đúng, đo lường khả năng của mô hình trong việc phát hiện và phân loại đúng các văn bản thực sự thuộc một chủ đề cụ thể. Chỉ số này cho biết, trong số tất cả các văn bản thuộc một chủ đề, mô hình dự đoán đúng được bao nhiêu phần trăm:



Trong đó:

* TP (True Positive): Số văn bản thuộc chủ đề đang xét và được mô hình dự đoán đúng là chủ đề đó.
* FN (False Negative): Số văn bản thực sự thuộc chủ đề đang xét nhưng bị mô hình dự đoán nhầm sang một chủ đề khác.

**Điểm F1 (F1-Score):** là trung bình điều hòa giữa Precision và Recall, được sử dụng để đánh giá sự cân bằng giữa độ tin cậy của dự đoán và khả năng phát hiện đúng của mô hình đối với một chủ đề cụ thể. Chỉ số này đặc biệt hữu ích trong bài toán phân loại văn bản, nhất là khi phân bố dữ liệu giữa các chủ đề không đồng đều:



Trong đó:

* **Precision**: mức độ chính xác khi mô hình gán một văn bản vào một chủ đề.
* **Recall**: khả năng phát hiện đầy đủ các văn bản thực sự thuộc chủ đề đó.

Do đó, F1-Score phản ánh hiệu năng tổng hợp của mô hình khi cần vừa phân loại đúng, vừa hạn chế nhầm lẫn giữa các chủ đề.

Trong bài toán phân loại văn bản tiếng Việt theo chủ đề, F1-Score thường được sử dụng để so sánh hiệu năng giữa các mô hình học máy và học sâu, bởi chỉ số này không bị chi phối quá mạnh bởi các chủ đề có số lượng văn bản lớn. F1-Score cao cho thấy mô hình vừa nhận diện tốt các văn bản thuộc từng chủ đề (Recall cao), vừa duy trì độ tin cậy cao khi gán nhãn (Precision cao), qua đó thể hiện sự cân bằng hợp lý giữa khả năng phát hiện và độ chính xác của mô hình.

## 2.7. Xử lý mất cân bằng lớp:

Trong bài toán phân loại, dữ liệu huấn luyện thường tồn tại hiện tượng mất cân bằng lớp, trong đó một số chủ đề phổ biến có số lượng văn bản lớn, trong khi các chủ đề khác chỉ chiếm tỷ lệ nhỏ. Nếu không được xử lý phù hợp, hiện tượng này có thể khiến mô hình thiên lệch về các lớp chiếm ưu thế, dẫn đến việc phân loại kém đối với các lớp thiểu số và làm suy giảm khả năng tổng quát hóa của mô hình.

### ***2.7.1. So sánh các phương pháp xử lý mất cân bằng lớp:***

Trong học máy và học sâu, nhiều phương pháp đã được đề xuất để xử lý mất cân bằng dữ liệu:

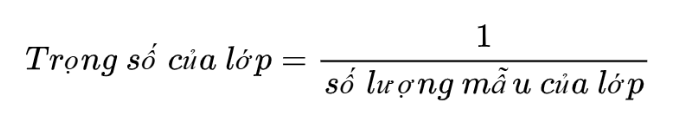
* **Phương pháp tái lấy mẫu dữ liệu (Resampling)** là một cách tiếp cận phổ biến, bao gồm oversampling các lớp thiểu số hoặc undersampling các lớp chiếm đa số. Oversampling nhằm tăng số lượng mẫu của lớp thiểu số bằng cách lặp lại các mẫu hiện có, trong khi undersampling làm giảm số lượng mẫu của lớp đa số để đạt được sự cân bằng. Tuy nhiên, đối với dữ liệu văn bản, oversampling thường dẫn đến hiện tượng overfitting do nội dung văn bản bị lặp lại nhiều lần, còn undersampling có thể gây mất mát thông tin quan trọng từ các lớp có nhiều dữ liệu. Do đó, phương pháp này không thực sự phù hợp với bài toán phân loại văn bản theo chủ đề.
* **Phương pháp SMOTE(Synthetic Minority Over-sampling Technique)** là một biến thể nâng cao của oversampling, trong đó các mẫu tổng hợp của lớp thiểu số được sinh ra bằng cách nội suy giữa các điểm dữ liệu lân cận trong không gian đặc trưng. SMOTE giúp giảm hiện tượng lặp dữ liệu so với oversampling truyền thống và đã được chứng minh hiệu quả trong nhiều bài toán phân loại dữ liệu số. Tuy nhiên, khi áp dụng cho dữ liệu văn bản sau khi vector hóa (như TF-IDF hoặc Word2Vec), các vector tổng hợp sinh ra có thể không phản ánh đúng ngữ nghĩa của văn bản, làm giảm tính diễn giải và có nguy cơ đưa nhiễu vào dữ liệu huấn luyện. Vì vậy, SMOTE không phải là lựa chọn tối ưu cho bài toán phân loại văn bản theo chủ đề.
* **Phương pháp điều chỉnh ngưỡng quyết định (Decision Threshold Adjustment)** tập trung vào việc thay đổi ngưỡng dự đoán của mô hình nhằm ưu tiên một số lớp nhất định. Phương pháp này thường được áp dụng hiệu quả trong các bài toán phân loại nhị phân, nơi đầu ra của mô hình là xác suất. Tuy nhiên, trong bài toán phân loại đa lớp theo chủ đề, việc xác định và tối ưu ngưỡng cho từng lớp là phức tạp và khó đạt hiệu quả ổn định, do đó phương pháp này ít được sử dụng trong thực tế.
* **Phương pháp gán trọng số cho lớp hoặc cho mẫu dữ liệu (Class Weight / Sample Weight)** là một cách tiếp cận hiệu quả và được sử dụng rộng rãi trong học máy và học sâu. Phương pháp này tác động trực tiếp vào hàm mất mát trong quá trình huấn luyện bằng cách tăng mức phạt đối với các lỗi dự đoán sai thuộc lớp thiểu số. Nhờ đó, mô hình được khuyến khích học tốt hơn các đặc trưng của những lớp có ít dữ liệu, đồng thời vẫn giữ nguyên phân bố dữ liệu gốc.

So với các phương pháp trên nhóm nhận thấy phương pháp **class weight và sample weight** là cách tiếp cận trực tiếp, hiệu quả và phù hợp hơn đối với dữ liệu văn bản có không gian đặc trưng chiều cao và phân bố lớp không đồng đều.

### ***2.7.2. Lý do lựa chọn:***

Trong nghiên cứu này, phương pháp gán trọng số (class weight và sample weight) được lựa chọn để xử lý mất cân bằng lớp vì các lý do sau:

* Phương pháp này không làm thay đổi phân bố dữ liệu gốc, do đó tránh được việc tạo ra dữ liệu nhân tạo hoặc loại bỏ dữ liệu quan trọng. Điều này đặc biệt phù hợp với bài toán phân loại văn bản, nơi mỗi văn bản mang ý nghĩa ngữ nghĩa riêng và khó có thể thay thế.
* Class weight tác động trực tiếp vào hàm mất mát, khiến các lỗi dự đoán sai ở lớp thiểu số bị “phạt” nặng hơn. Nhờ đó, mô hình được buộc phải học tốt hơn các đặc trưng của những lớp có ít dữ liệu, thay vì chỉ tối ưu cho các lớp chiếm đa số.
* Phương pháp này áp dụng thống nhất cho cả các mô hình học máy và học sâu, giúp xây dựng một pipeline huấn luyện nhất quán và dễ so sánh kết quả giữa các mô hình khác nhau.



***Ảnh hưởng đến các mô hình học máy:***

Đối với các mô hình học máy truyền thống như Logistic Regression, SVM và XGBoost, việc sử dụng class weight hoặc sample weight làm thay đổi mức độ đóng góp của từng mẫu dữ liệu vào hàm mục tiêu:

* Các lớp thiểu số có ảnh hưởng lớn hơn trong quá trình tối ưu, giúp mô hình giảm thiên lệch về các lớp chiếm ưu thế.
* Phương pháp này thường giúp cải thiện Recall và F1-score cho các lớp có ít dữ liệu, trong khi Accuracy tổng thể có thể chỉ thay đổi nhẹ phản ánh sự đánh đổi hợp lý giữa độ chính xác chung và khả năng bao phủ lớp.
* Trong XGBoost, sample weight cho phép điều chỉnh mức độ quan trọng của từng mẫu khi xây dựng cây quyết định, từ đó nâng cao khả năng phân biệt giữa các chủ đề.

***Ảnh hưởng đến các mô hình học sâu:***

Đối với các mô hình học sâu như DNN và LSTM, class weight được sử dụng để điều chỉnh hàm mất mát trong quá trình lan truyền ngược:

* Các lỗi dự đoán sai thuộc lớp thiểu số tạo ra gradient lớn hơn, khiến mô hình cập nhật trọng số mạnh hơn theo hướng cải thiện khả năng phân loại các lớp này. Điều này giúp mô hình học sâu khai thác tốt hơn các đặc trưng ngữ nghĩa của các chủ đề ít dữ liệu, đặc biệt quan trọng trong bài toán phân loại văn bản.
* Tuy nhiên, nếu chênh lệch trọng số giữa các lớp quá lớn, quá trình huấn luyện có thể kém ổn định. Do đó, class weight thường được sử dụng kết hợp với các kỹ thuật như early stopping hoặc điều chỉnh learning rate để đảm bảo sự hội tụ ổn định.

Từ việc so sánh các phương pháp xử lý mất cân bằng lớp, có thể thấy rằng class weight / sample weight là lựa chọn phù hợp cho bài toán phân loại văn bản tiếng Việt theo chủ đề. Phương pháp này vừa giữ nguyên dữ liệu gốc, vừa cải thiện khả năng học của các lớp thiểu số, đồng thời áp dụng hiệu quả cho cả các mô hình học máy và học sâu. Nhờ đó, các mô hình đạt được hiệu năng cân bằng hơn giữa các chủ đề và có khả năng tổng quát hóa tốt hơn trên dữ liệu kiểm tra.

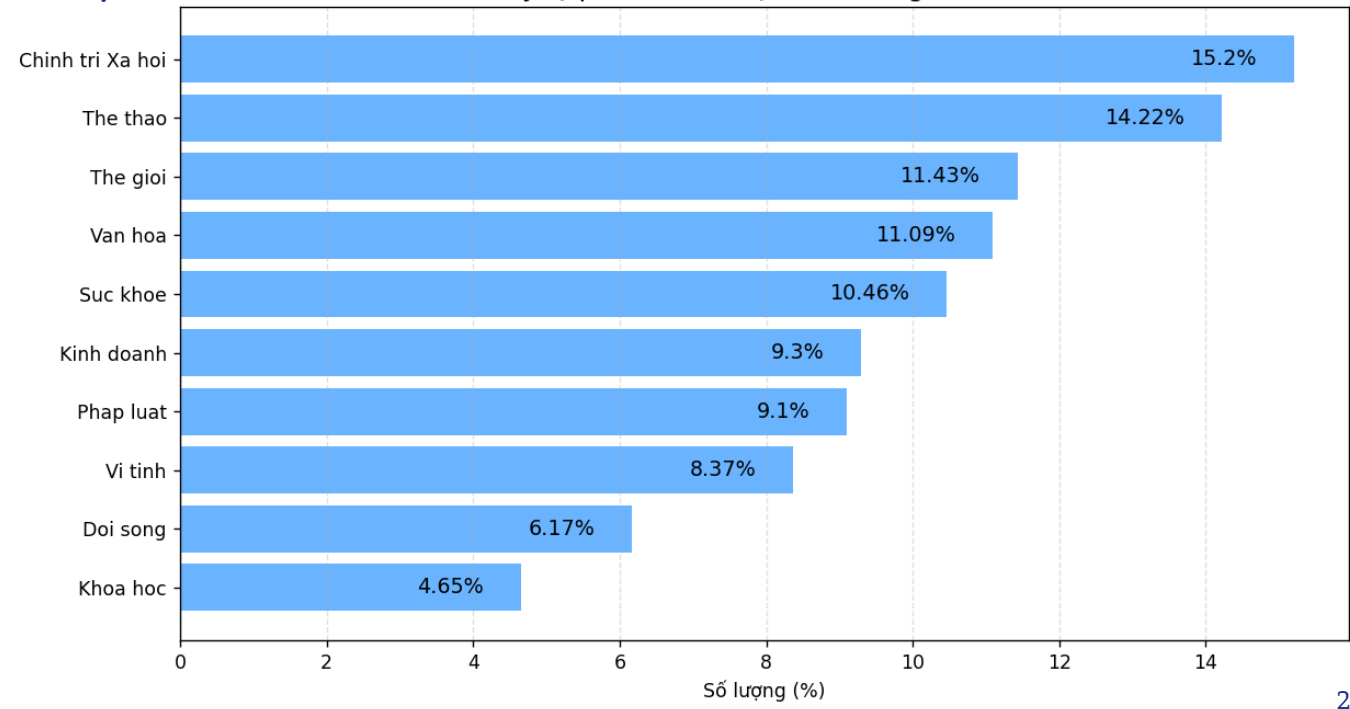
# Chương III. TRIỂN KHAI VÀ ĐÁNH GIÁ MÔ HÌNH

## 3.1. Bộ dữ liệu:

Trong nghiên cứu này, bộ dữ liệu được sử dụng là VNTC (Vietnamese News Text Classification) được lấy từ github của anh DuyVuLeo, một bộ dữ liệu chuẩn cho bài toán phân loại văn bản tiếng Việt theo chủ đề. Dữ liệu được xây dựng từ các bài báo điện tử thu thập từ nhiều nguồn tin chính thống và uy tín như VNExpress, Tuổi Trẻ, Thanh Niên và Người Lao Động, đảm bảo độ tin cậy về nội dung cũng như tính đa dạng về chủ đề. Mỗi mẫu dữ liệu trong bộ VNTC là nội dung một bài báo tiếng Việt ở dạng văn bản thuần (plain text) và được gán nhãn tương ứng với một trong mười chủ đề bao gồm: *Chính trị – Xã hội, Đời sống, Khoa học, Kinh doanh, Pháp luật, Sức khỏe, Thế giới, Thể thao, Văn hóa* và *Vi tính*. Do đó, bài toán được đặt ra trong nghiên cứu này là bài toán phân loại văn bản đa lớp, trong đó mỗi văn bản chỉ thuộc về một chủ đề duy nhất.

#### **Bảng 3.1.** Phân bố số lượng văn bản theo từng chủ đề trong bộ dữ liệu VNTC

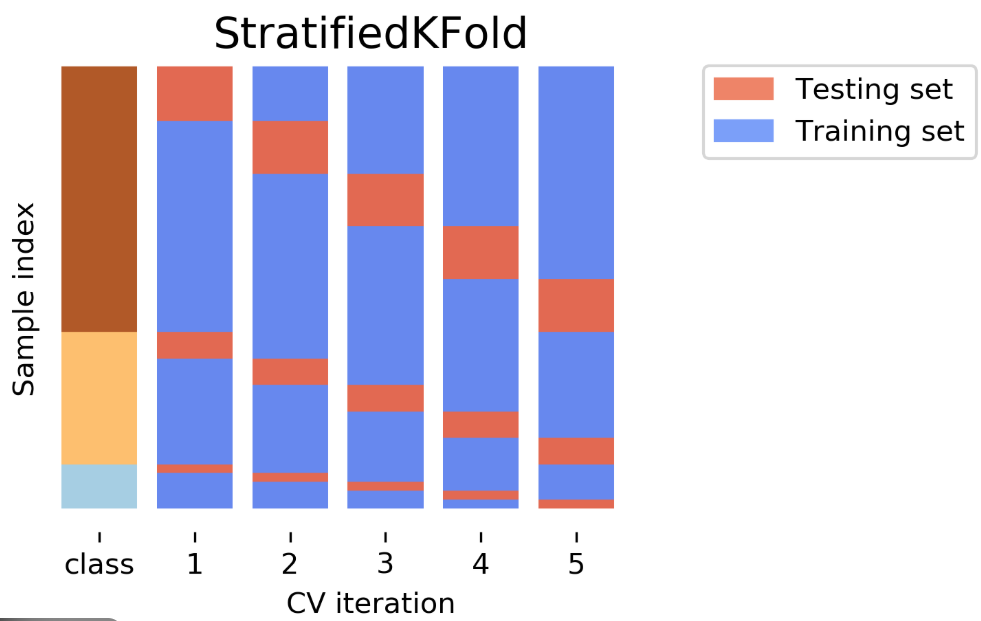
Theo cấu trúc gốc do bộ dữ liệu cung cấp, VNTC bao gồm tổng cộng 84.132 văn bản, được chia sẵn thành hai phần: 33.759 văn bản dùng cho mục đích huấn luyện và 50.373 văn bản dùng cho mục đích kiểm tra. Lưu ý rằng, đây là cách tổ chức ban đầu của bộ dữ liệu gốc, không phải là kết quả của quá trình chia dữ liệu trong nghiên cứu này. Trong quá trình thực nghiệm, toàn bộ dữ liệu được gộp lại và chia nhằm đảm bảo tính nhất quán trong huấn luyện và đánh giá mô hình. Cụ thể, toàn bộ dữ liệu được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ 80% – 20% bằng phương pháp Stratified Split. Phương pháp này được lựa chọn nhằm đảm bảo tỷ lệ phân bố các chủ đề trong tập huấn luyện và tập kiểm tra được giữ tương đồng với phân bố của toàn bộ dữ liệu, từ đó hạn chế sai lệch trong đánh giá, đặc biệt trong bối cảnh dữ liệu có hiện tượng mất cân bằng lớp. Sau khi chia dữ liệu, tập huấn luyện (80%) được sử dụng cho quá trình xây dựng mô hình, trong khi tập kiểm tra (20%) được giữ cố định và chỉ sử dụng cho đánh giá cuối cùng, nhằm đảm bảo tính khách quan của kết quả thực nghiệm.



## Hình 3.1. Tỷ lệ phân bố dữ liệu theo từng chủ đề

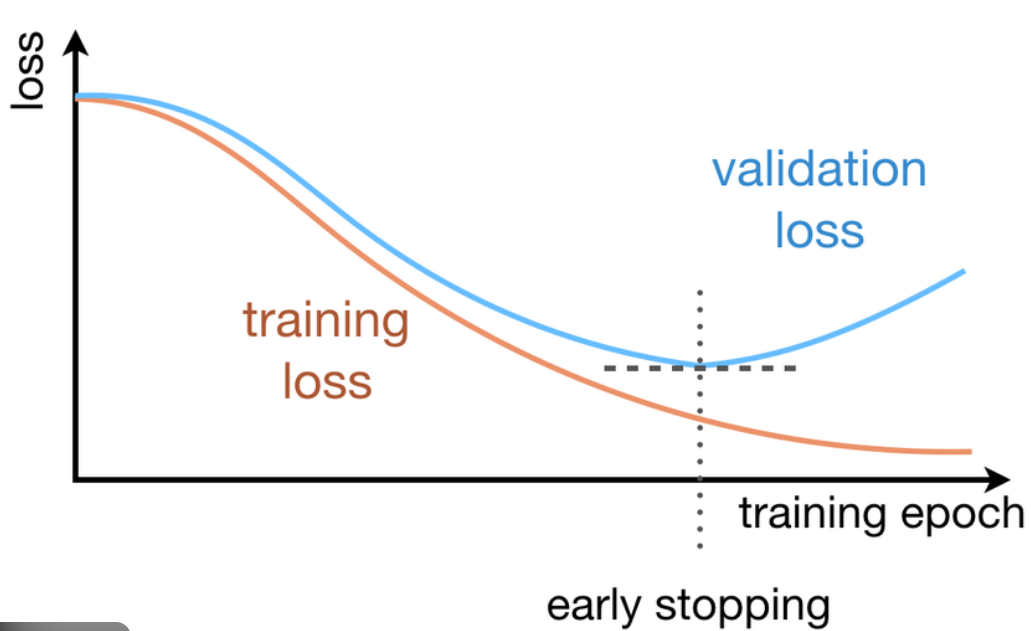
Do đặc thù của dữ liệu văn bản theo chủ đề, số lượng mẫu giữa các lớp không hoàn toàn đồng đều, trong đó một số chủ đề có số lượng văn bản lớn hơn đáng kể so với các chủ đề còn lại. Hiện tượng mất cân bằng lớp này có thể ảnh hưởng đến quá trình huấn luyện, khiến mô hình có xu hướng ưu tiên các lớp chiếm đa số nếu không có biện pháp xử lý phù hợp.

Đối với các mô hình học máy truyền thống, để đánh giá mô hình một cách ổn định và giảm ảnh hưởng của việc chia dữ liệu ngẫu nhiên, nghiên cứu áp dụng **Stratified 10-fold Cross-Validation** trên tập huấn luyện. Cụ thể, tập huấn luyện được chia thành 10 phần con (fold) sao cho mỗi fold đều duy trì tỷ lệ phân bố nhãn tương đương. Trong mỗi lần lặp, 9 fold được sử dụng để huấn luyện và 1 fold được sử dụng để xác thực; quá trình này được lặp lại cho đến khi mỗi fold đều lần lượt đóng vai trò là tập validation. Cách tiếp cận này giúp đánh giá mô hình trên nhiều tập dữ liệu khác nhau, giảm phụ thuộc vào một lần chia dữ liệu duy nhất và cung cấp ước lượng hiệu năng ổn định và đáng tin cậy hơn.



## Hình 3.2. Nguyên lý hoạt động của Stratified Fold Cross-Validation

Đối với các mô hình học sâu, tập kiểm tra được sử dụng làm tập validation trong quá trình huấn luyện, kết hợp với **kỹ thuật early {stopping** nhằm theo dõi sự hội tụ của mô hình và hạn chế hiện tượng overfitting, đồng thời vẫn đảm bảo rằng tập dữ liệu này không tham gia vào quá trình huấn luyện mô hình.



## Hình 3.3. Nguyên lý kỹ thuật Early Stopping dựa trên validation loss.

Nhờ quy mô dữ liệu lớn, đa dạng chủ đề và chiến lược chia dữ liệu hợp lý, bộ dữ liệu VNTC cung cấp nền tảng phù hợp để huấn luyện và đánh giá các mô hình học máy và học sâu trong bài toán phân loại văn bản tiếng Việt theo chủ đề.

## 3.2. Tiền xử lý:

### ***3.2.1. Tiền xử lý dữ liệu tổng thể***

Trong dự án này, bộ dữ liệu VNTC được sử dụng dưới dạng các tệp văn bản đã được gán nhãn chủ đề. Trước khi xử lý nội dung văn bản, dữ liệu được tiền xử lý ở mức tổng thể nhằm đảm bảo tính nhất quán và chất lượng đầu vào:

* Loại bỏ các văn bản trùng lặp (nếu có) để tránh việc một nội dung xuất hiện nhiều lần trong quá trình huấn luyện, gây thiên lệch kết quả.
* Kiểm tra tính hợp lệ của dữ liệu, chỉ giữ lại các văn bản có đầy đủ nội dung và nhãn chủ đề tương ứng.
* Chuẩn hóa cấu trúc dữ liệu, trong đó mỗi mẫu dữ liệu bao gồm hai thành phần chính: nội dung văn bản (text) và nhãn chủ đề (label).

### ***3.2.2. Tiền xử lý dữ liệu văn bản***

Quy trình tiền xử lý được áp dụng thống nhất cho toàn bộ tập dữ liệu văn bản tiếng Việt, bao gồm các bước sau:

**B1. Loại bỏ nhiễu trong văn bản**

* Xóa các URL (nếu xuất hiện) và các chuỗi ký hiệu không mang thông tin ngữ nghĩa.
* Loại bỏ các ký tự đặc biệt và dấu câu không cần thiết.

**B2. Chuẩn hóa văn bản**

* Chuyển toàn bộ văn bản về chữ thường.
* Loại bỏ các ký tự không thuộc bảng chữ cái tiếng Việt.

**B3. Chuẩn hóa teencode**

* Thay thế các từ viết tắt hoặc dạng viết không chuẩn bằng dạng từ chuẩn tương ứng thông qua từ điển teencode.

**B4. Tách từ tiếng Việt**

* Sử dụng công cụ VnTokenizer để tách từ, đảm bảo xác định đúng ranh giới từ trong văn bản tiếng Việt.

**B5. Loại bỏ stopwords**

* Loại bỏ các từ dừng trong tiếng Việt (liên từ, trợ từ, từ chức năng) nhằm giảm nhiễu và thu gọn không gian đặc trưng.

**B6. Chuẩn hóa định dạng văn bản**

* Gộp nhiều khoảng trắng liên tiếp thành một khoảng trắng duy nhất.
* Loại bỏ khoảng trắng dư thừa ở đầu và cuối văn bản.

**B7. Văn bản đầu ra sau tiền xử lý**

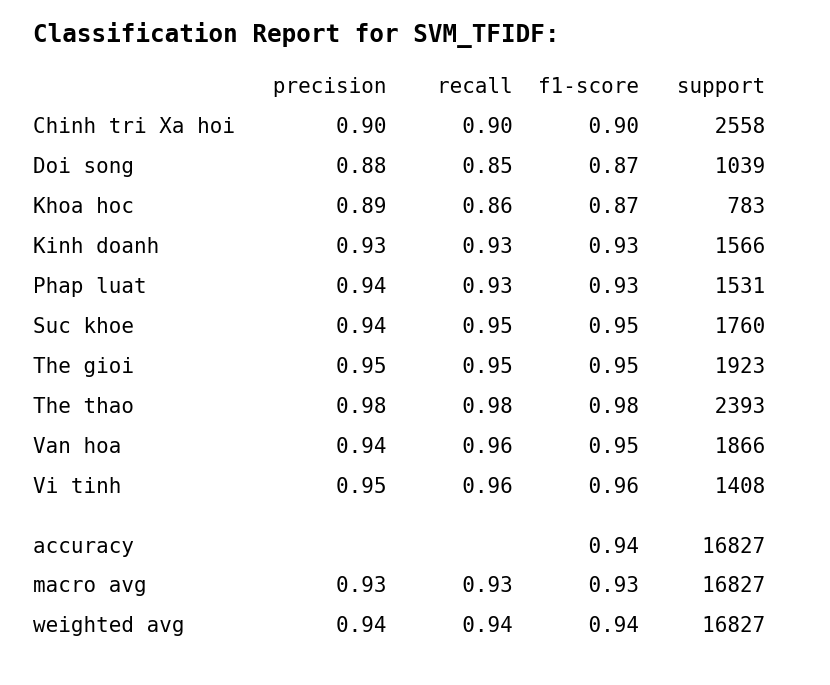
* Kết quả của bước tiền xử lý là nội dung văn bản tiếng Việt đã được làm sạch, chuẩn hóa và tách từ.
* Tập văn bản này được sử dụng trực tiếp cho các bước trích xuất đặc trưng (TF-IDF, n-gram, Word2Vec).

### ***3.2.3. Vai trò của tiền xử lý trong dự án***

* Đảm bảo tính nhất quán dữ liệu giữa tập huấn luyện và tập kiểm tra.
* Giảm nhiễu và độ thưa trong không gian đặc trưng.
  + Nâng cao hiệu quả của các phương pháp trích xuất đặc trưng và mô hình phân loại.
* Đảm bảo rằng sự khác biệt về hiệu năng giữa các mô hình phản ánh đúng lực thuật toán, thay vì do khác biệt trong xử lý dữ liệu đầu vào.

## 3.3. Đánh giá các mô hình học máy:

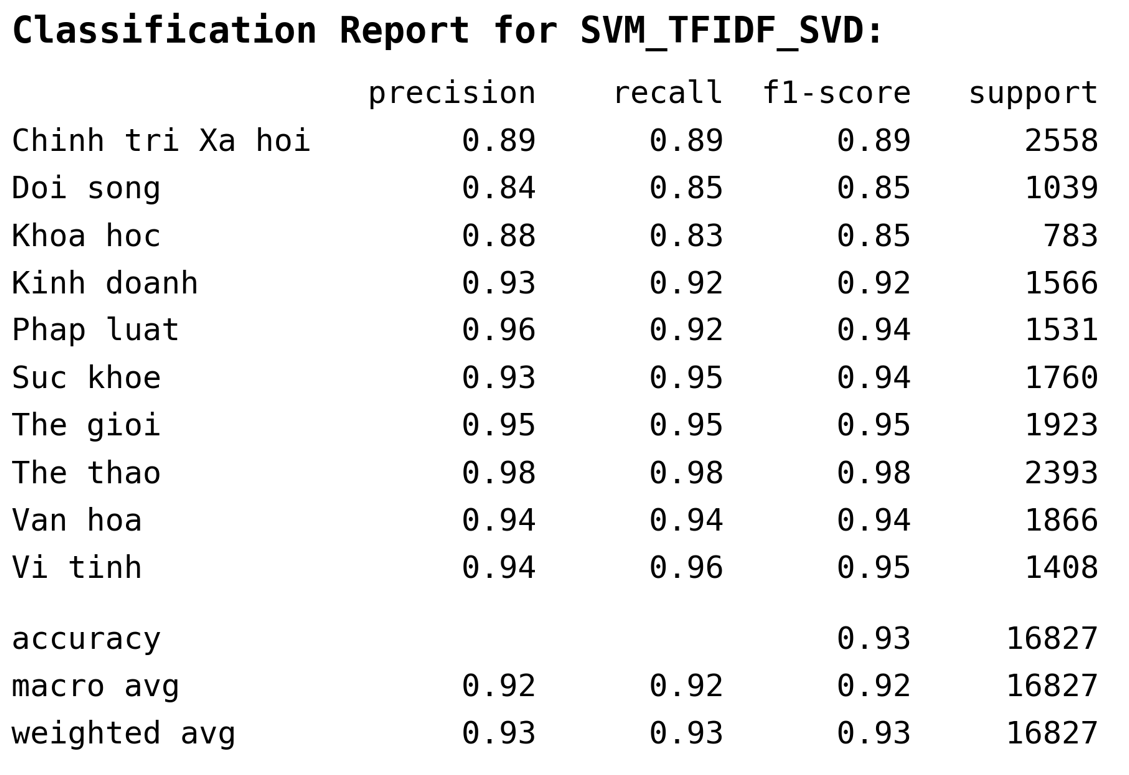
### ***3.3.1. Mô hình SVM kết hợp với đặc trưng TF-IDF:***



## Hình 3.4. Chỉ số đánh giá của mô hình SVM với đặc trưng TF-IDF.

Mô hình SVM kết hợp với đặc trưng TF-IDF đạt hiệu năng phân loại cao, với Accuracy = 0.94, cho thấy khả năng dự đoán chính xác và ổn định trên tập dữ liệu văn bản tiếng Việt theo chủ đề. Phần lớn các lớp đạt Precision, Recall và F1-score cao (0.87–0.98), phản ánh mô hình phân biệt tốt giữa các chủ đề và ít xảy ra sai lệch. Một số lớp như Đời sống và Khoa học có chỉ số thấp hơn so với các lớp còn lại, nguyên nhân có thể do sự giao thoa ngữ nghĩa giữa các chủ đề và đặc trưng nội dung chưa thực sự rõ ràng. Tuy nhiên, sự chênh lệch này không đáng kể và không ảnh hưởng lớn đến hiệu năng tổng thể của mô hình. Giá trị Macro avg = 0.93 và Weighted avg = 0.94 cho thấy mô hình hoạt động tương đối đồng đều giữa các lớp, đồng thời khai thác hiệu quả các lớp có nhiều dữ liệu. Nhìn chung, SVM kết hợp TF-IDF là mô hình phù hợp và đáng tin cậy để làm mô hình nền cho bài toán phân loại văn bản tiếng Việt theo chủ đề.

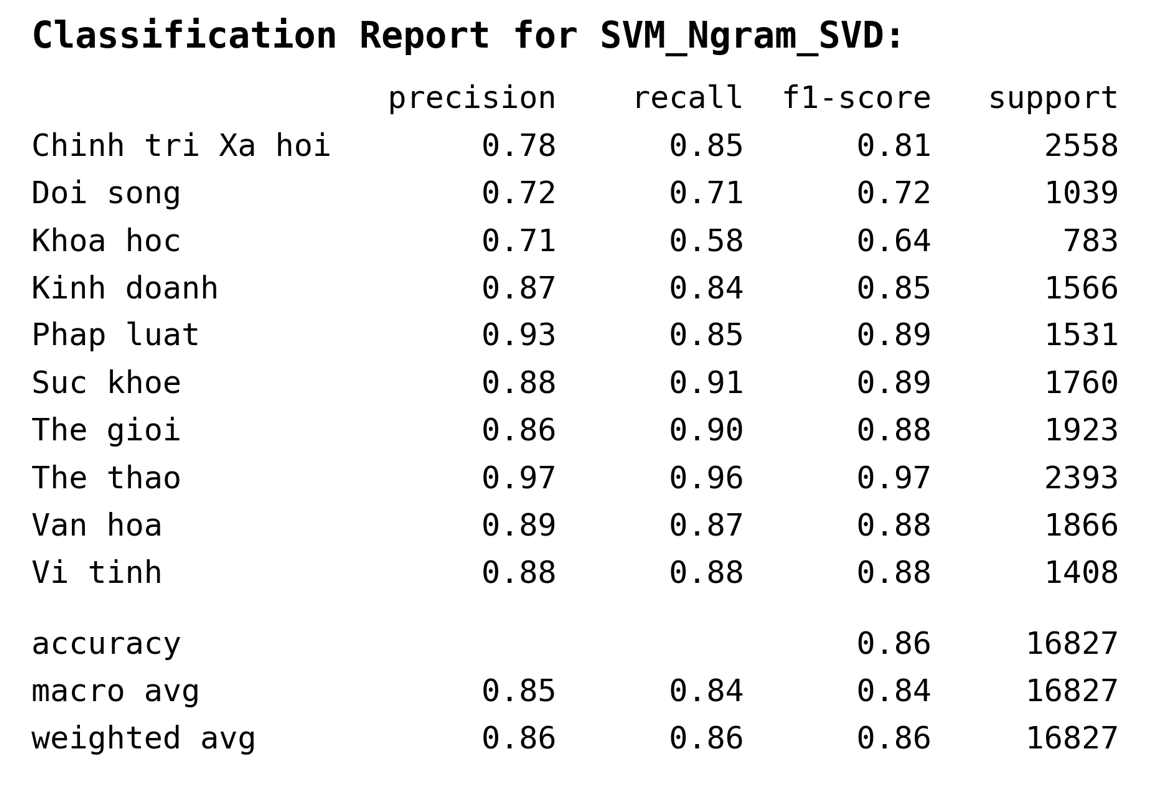
### ***3.3.2. Mô hình SVM kết hợp với TF-IDF và giảm chiều dữ liệu bằng SVD:***



## Hình 3.5. Chỉ số đánh giá của mô hình SVM kết hợp với TF-IDF và giảm chiều SVD

Mô hình SVM kết hợp với đặc trưng TF-IDF và giảm chiều dữ liệu bằng SVD đạt Accuracy = 0.93, cho thấy hiệu năng phân loại tốt và ổn định trên tập dữ liệu. Hầu hết các lớp đều có Precision, Recall và F1-score ở mức cao (khoảng 0.85–0.98), phản ánh khả năng phân biệt chủ đề hiệu quả sau khi giảm chiều đặc trưng. So với mô hình SVM chỉ sử dụng TF-IDF, việc áp dụng SVD giúp giảm độ phức tạp và nhiễu của không gian đặc trưng, tuy nhiên hiệu năng tổng thể giảm nhẹ. Một số lớp như Đời sống và Khoa học có F1-score thấp hơn, cho thấy việc nén chiều có thể làm mất đi một phần thông tin đặc trưng quan trọng của các chủ đề có nội dung giao thoa. Giá trị Macro average = 0.92 và Weighted average = 0.93 cho thấy mô hình hoạt động tương đối đồng đều giữa các lớp và không bị lệch đáng kể bởi sự mất cân bằng dữ liệu. Nhìn chung, SVM kết hợp TF-IDF và SVD là mô hình hiệu quả về mặt tính toán, phù hợp trong các trường hợp cần giảm chiều dữ liệu, nhưng chưa vượt trội so với mô hình SVM kết hợp TF-IDF nguyên bản về độ chính xác.

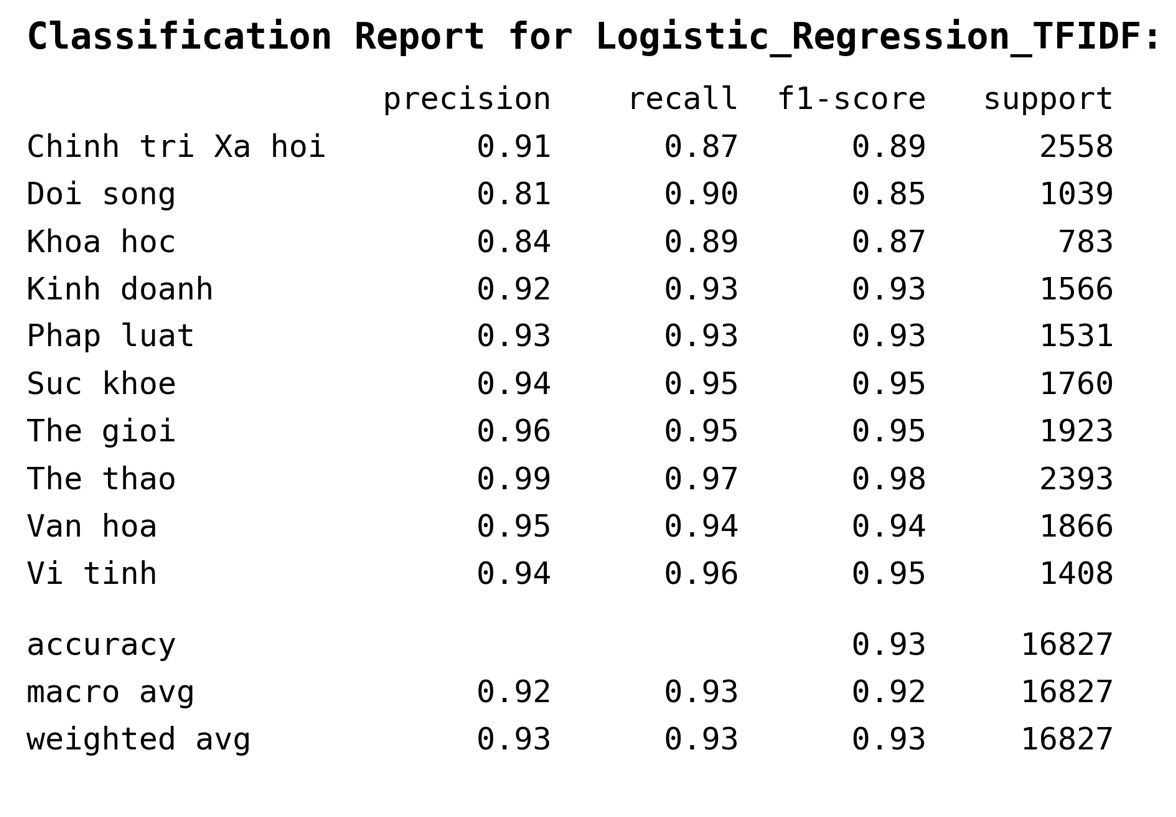
### ***3.3.3. Mô hình SVM kết hợp với đặc trưng N-gram và SVD:***



## Hình 3.6. Chỉ số đánh giá của mô hình SVM kết hợp với N-gram và giảm chiều SVD

Mô hình SVM kết hợp với đặc trưng N-gram và giảm chiều bằng SVD đạt Accuracy = 0.86, cho thấy hiệu năng phân loại ở mức khá nhưng thấp hơn so với các mô hình sử dụng TF-IDF. Các chỉ số Macro average = 0.84 và Weighted average = 0.86 phản ánh sự suy giảm hiệu quả trên nhiều lớp. Một số lớp như Chính trị – Xã hội, Đời sống và Khoa học có F1-score thấp, đặc biệt lớp Khoa học (0.64), cho thấy mô hình gặp khó khăn trong việc phân biệt các chủ đề có nội dung giao thoa khi sử dụng đặc trưng N-gram sau khi giảm chiều. Điều này cho thấy việc kết hợp N-gram với SVD có thể làm mất đi các mối quan hệ ngữ nghĩa quan trọng giữa các từ. Ngược lại, các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Pháp luật, Sức khỏe và Thế giới vẫn đạt kết quả tốt (F1-score từ 0.88 đến 0.97). Nhìn chung, SVM kết hợp N-gram và SVD không phải là lựa chọn tối ưu cho bài toán phân loại văn bản tiếng Việt theo chủ đề trong nghiên cứu này, và kém hiệu quả hơn so với các mô hình dựa trên TF-IDF.

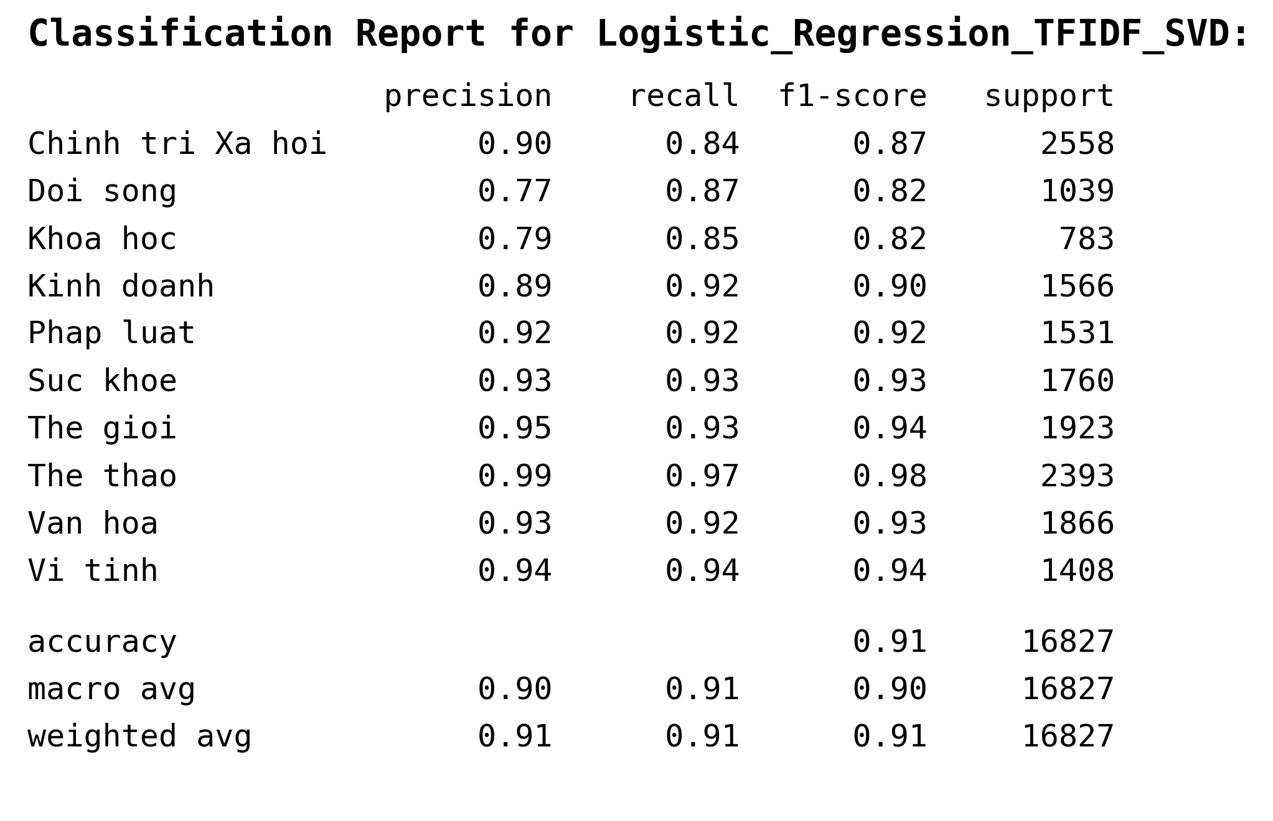
### ***3.3.4. Mô hình Logistic Regression kết hợp với đặc trưng TF-IDF:***



## Hình 3.7. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với TF-IDF

Mô hình Logistic Regression kết hợp với đặc trưng TF-IDF đạt Accuracy = 0.93, cho thấy hiệu năng phân loại cao và ổn định trên tập dữ liệu văn bản tiếng Việt theo chủ đề. Hầu hết các lớp đều đạt Precision, Recall và F1-score cao (khoảng 0.85–0.98), phản ánh khả năng phân biệt chủ đề hiệu quả trong không gian đặc trưng TF-IDF. Một số lớp như Đời sống có Recall cao nhưng Precision thấp hơn, cho thấy mô hình có xu hướng dự đoán bao phủ rộng hơn, dẫn đến một số trường hợp nhầm lẫn với các chủ đề gần nghĩa. Tuy nhiên, sự chênh lệch này không ảnh hưởng đáng kể đến hiệu năng tổng thể của mô hình. Giá trị Macro average = 0.92 và Weighted average = 0.93 cho thấy mô hình hoạt động tương đối đồng đều giữa các lớp và tận dụng tốt các lớp có nhiều dữ liệu. Nhìn chung, Logistic Regression kết hợp TF-IDF là mô hình đơn giản, hiệu quả và có tính ổn định cao, phù hợp làm mô hình so sánh và đối chứng với các phương pháp phân loại khác trong nghiên cứu.

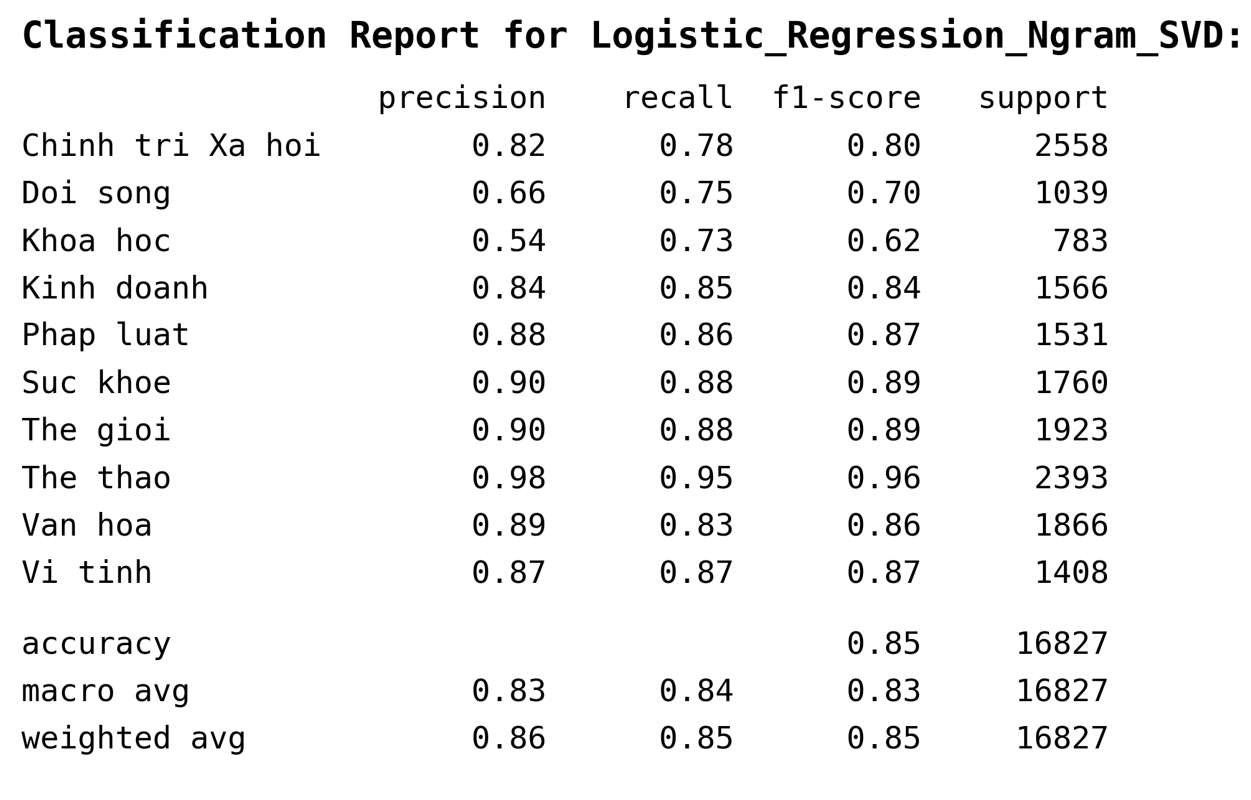
### ***3.3.5. Mô hình Logistic Regression kết hợp với TF-IDF và SVD:***



## Hình 3.8. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với TF-IDF và SVD

Mô hình Logistic Regression kết hợp với đặc trưng TF-IDF và giảm chiều bằng SVD đạt Accuracy = 0.91, cho thấy hiệu năng phân loại tốt nhưng giảm so với mô hình Logistic Regression chỉ sử dụng TF-IDF. Các chỉ số Macro average = 0.90 và Weighted average = 0.91 phản ánh sự suy giảm nhẹ về hiệu quả khi áp dụng kỹ thuật giảm chiều. Một số lớp như Đời sống và Khoa học có F1-score thấp hơn, cho thấy việc nén chiều bằng SVD có thể làm mất đi một phần thông tin đặc trưng quan trọng đối với các chủ đề có nội dung giao thoa. Trong khi đó, các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Thế giới, Sức khỏe và Pháp luật vẫn duy trì kết quả tốt với F1-score cao. Nhìn chung, Logistic Regression kết hợp TF-IDF và SVD giúp giảm độ phức tạp mô hình và chi phí tính toán, tuy nhiên không mang lại cải thiện về độ chính xác trong bài toán này, và kém hiệu quả hơn so với mô hình Logistic Regression kết hợp TF-IDF nguyên bản.

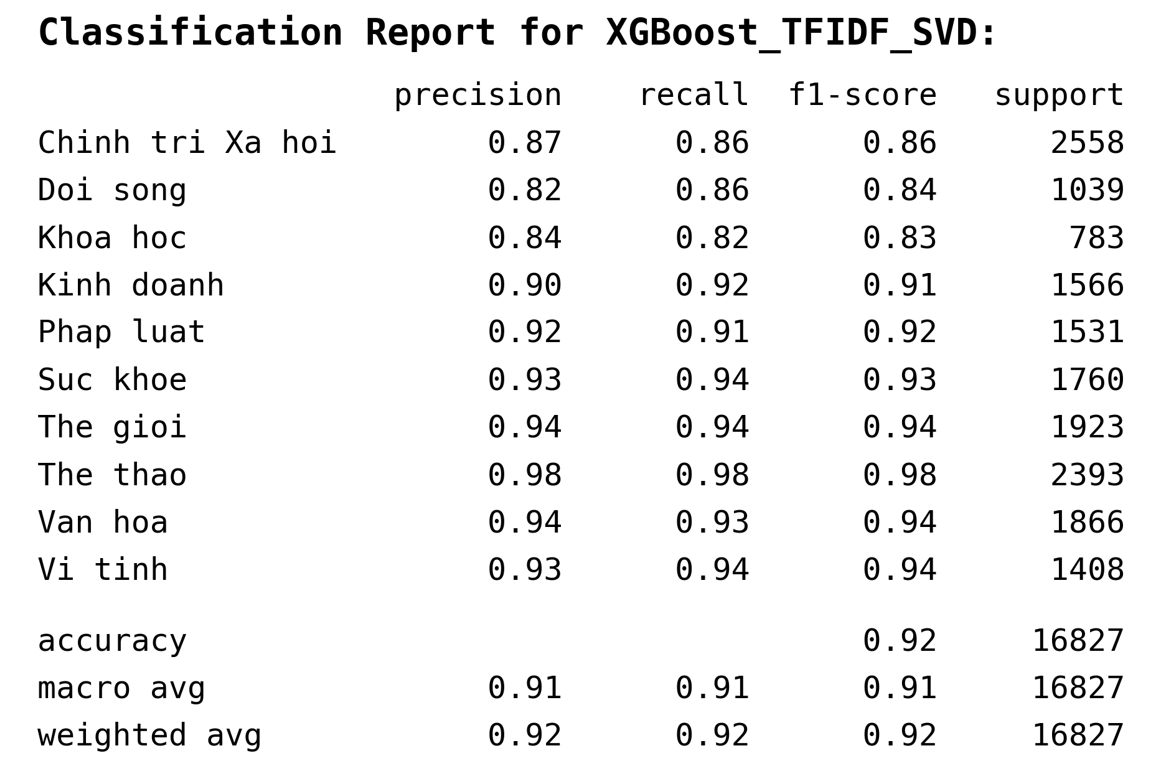
### ***3.3.6. Mô hình Logistic Regression kết hợp với đặc trưng N-gram và SVD:***

******

## Hình 3.9. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với N-gram và SVD

Mô hình Logistic Regression kết hợp với đặc trưng N-gram và giảm chiều bằng SVD đạt Accuracy = 0.85, cho thấy hiệu năng phân loại ở mức khá và thấp hơn so với các mô hình sử dụng TF-IDF. Các chỉ số Macro average = 0.83 và Weighted average = 0.85 phản ánh sự suy giảm hiệu quả trên nhiều lớp. Một số lớp như Đời sống và đặc biệt là Khoa học có F1-score thấp (lần lượt là 0.70 và 0.62), cho thấy mô hình gặp khó khăn trong việc phân biệt các chủ đề có nội dung giao thoa khi sử dụng đặc trưng N-gram sau khi giảm chiều. Điều này cho thấy việc kết hợp N-gram với SVD làm mất đi nhiều thông tin ngữ nghĩa quan trọng. Ngược lại, các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Sức khỏe, Thế giới và Pháp luật vẫn duy trì kết quả tương đối tốt với F1-score từ 0.87 đến 0.96. Nhìn chung, Logistic Regression kết hợp N-gram và SVD không phải là lựa chọn phù hợp cho bài toán phân loại văn bản tiếng Việt theo chủ đề trong nghiên cứu này và kém hiệu quả hơn so với các mô hình dựa trên TF-IDF.

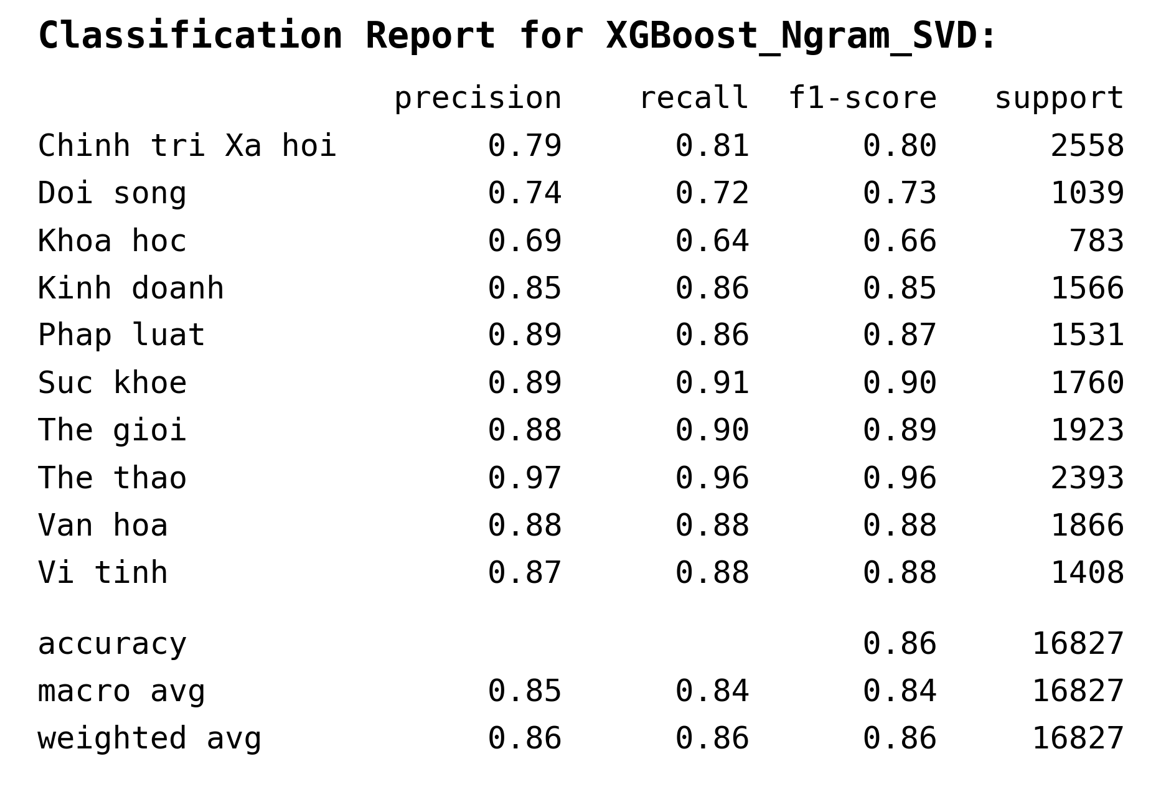
### ***3.3.7. Mô hình XGBoost kết hợp với TF-IDF và SVD:***



## Hình 3.10. Chỉ số đánh giá của mô hình XGBoost kết hợp với TF-IDF và SVD

Mô hình XGBoost kết hợp với đặc trưng TF-IDF và giảm chiều bằng SVD đạt Accuracy = 0.92, cho thấy hiệu năng phân loại tốt và ổn định trên tập dữ liệu văn bản tiếng Việt theo chủ đề. Các chỉ số Macro average = 0.91 và Weighted average = 0.92 phản ánh khả năng tổng quát hóa tương đối đồng đều giữa các lớp. Hầu hết các lớp đều đạt Precision, Recall và F1-score cao (khoảng 0.83–0.98), đặc biệt các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Thế giới, Văn hóa và Vi tính duy trì kết quả rất tốt. Một số lớp như Chính trị – Xã hội và Đời sống có chỉ số thấp hơn, cho thấy mô hình vẫn gặp khó khăn nhất định với các chủ đề có nội dung giao thoa. Nhìn chung, XGBoost kết hợp TF-IDF và SVD mang lại sự cân bằng tốt giữa hiệu năng và khả năng tổng quát hóa, tuy nhiên không vượt trội so với mô hình SVM kết hợp TF-IDF trong nghiên cứu này. Mô hình phù hợp để so sánh và làm phong phú thêm tập phương pháp trong bài toán phân loại văn bản tiếng Việt.

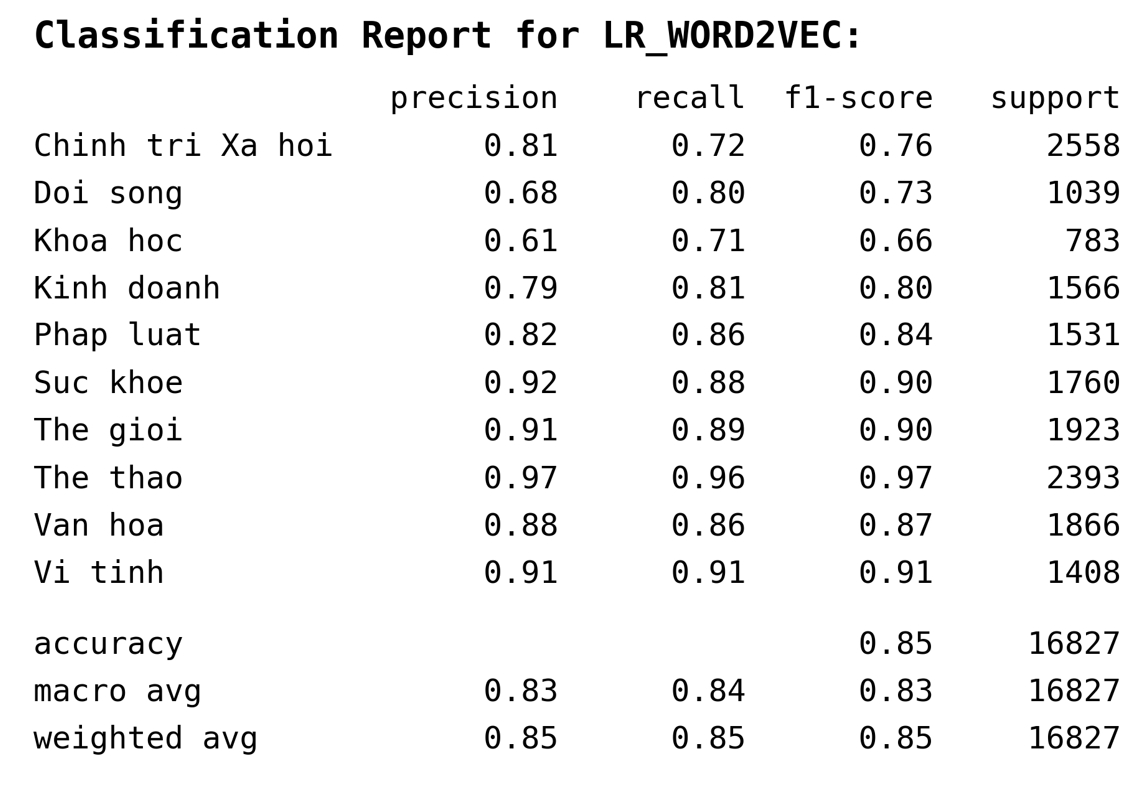
### ***3.3.8. Mô hình XGBoost kết hợp với đặc trưng N-gram và SVD:***



## Hình 3.11. Chỉ số đánh giá của mô hình XGBoost kết hợp với đặc trưng N-gram và SVD

Mô hình XGBoost kết hợp với đặc trưng N-gram và giảm chiều bằng SVD đạt Accuracy = 0.86, cho thấy hiệu năng phân loại ở mức khá nhưng thấp hơn so với các mô hình sử dụng TF-IDF. Các chỉ số Macro average = 0.84 và Weighted average = 0.86 phản ánh sự suy giảm hiệu quả trên nhiều lớp. Một số lớp như Đời sống và đặc biệt là Khoa học có F1-score thấp (0.73 và 0.66), cho thấy mô hình gặp khó khăn trong việc phân biệt các chủ đề có nội dung giao thoa khi sử dụng đặc trưng N-gram sau khi giảm chiều. Việc nén chiều bằng SVD trong trường hợp này có thể làm mất đi các thông tin ngữ nghĩa quan trọng. Ngược lại, các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Sức khỏe, Thế giới và Pháp luật vẫn đạt kết quả tốt với F1-score từ 0.87 đến 0.96. Nhìn chung, XGBoost kết hợp N-gram và SVD không mang lại hiệu quả vượt trội, và kém hơn đáng kể so với mô hình XGBoost kết hợp TF-IDF và SVD trong nghiên cứu này.

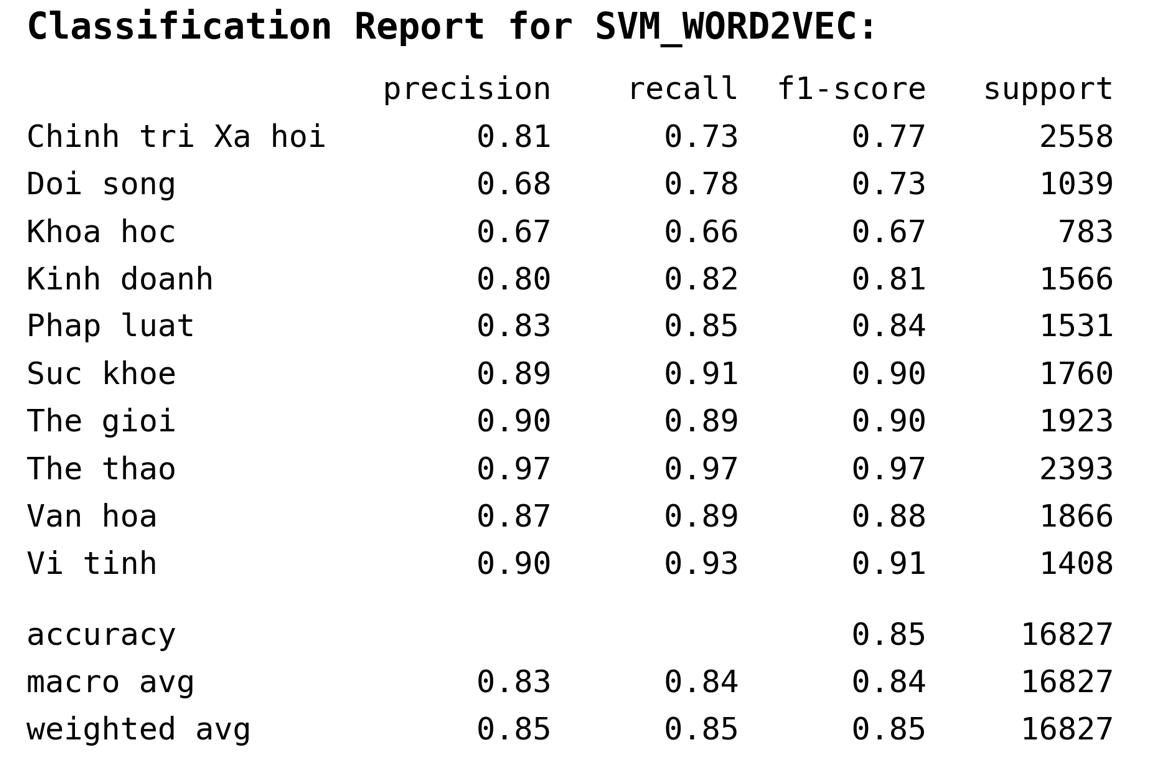
### ***3.3.9. Mô hình Logistic Regression kết hợp với Word2Vec tiền huấn luyện:***



## Hình 3.12. Chỉ số đánh giá của mô hình Logistic Regression kết hợp với Word2Vec tiền huấn luyện.

Mô hình Logistic Regression kết hợp với Word2Vec tiền huấn luyện đạt Accuracy = 0.85, cho thấy hiệu năng phân loại ở mức trung bình và thấp hơn so với các mô hình sử dụng TF-IDF. Các chỉ số Macro average = 0.83 và Weighted average = 0.85 phản ánh khả năng tổng quát hóa còn hạn chế trên một số lớp. Một số lớp như Chính trị – Xã hội, Đời sống và Khoa học có F1-score thấp, cho thấy mô hình tuyến tính gặp khó khăn khi khai thác các đặc trưng ngữ nghĩa phân bố trong không gian embedding Word2Vec. Điều này cho thấy Word2Vec, khi không kết hợp với mô hình học sâu, chưa phát huy tối đa hiệu quả trong bài toán phân loại chủ đề. Ngược lại, các lớp có đặc trưng ngữ nghĩa rõ ràng như Thể thao, Thế giới, Sức khỏe và Vi tính vẫn đạt kết quả tốt với F1-score từ 0.90 đến 0.97. Nhìn chung, Logistic Regression kết hợp Word2Vec phù hợp làm mô hình tham khảo, nhưng không phải là lựa chọn tối ưu cho bài toán phân loại văn bản tiếng Việt trong nghiên cứu này.

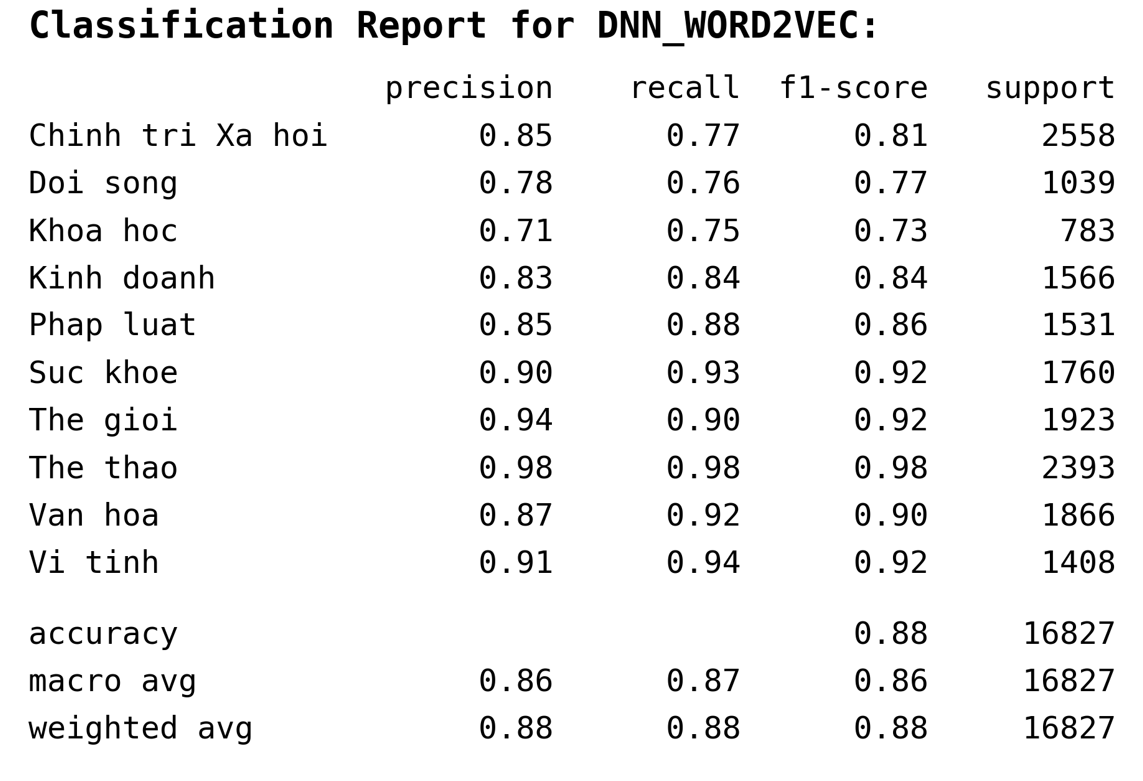
### ***3.3.10. Mô hình SVM kết hợp với Word2Vec tiền huấn luyện:***



## Hình 3.13. Chỉ số đánh giá của mô hình SVM kết hợp với Word2Vec tiền huấn luyện.

Mô hình SVM kết hợp với Word2Vec tiền huấn luyện đạt Accuracy = 0.85, cho thấy hiệu năng phân loại ở mức trung bình và thấp hơn đáng kể so với các mô hình sử dụng TF-IDF. Các chỉ số Macro average = 0.84 và Weighted average = 0.85 phản ánh khả năng tổng quát hóa chưa cao trên toàn bộ tập dữ liệu. Một số lớp như Chính trị – Xã hội, Đời sống và Khoa học có F1-score thấp, cho thấy SVM gặp khó khăn khi khai thác đặc trưng ngữ nghĩa liên tục từ Word2Vec trong bài toán phân loại tuyến tính theo chủ đề. Điều này cho thấy Word2Vec không phù hợp khi kết hợp trực tiếp với SVM cho bài toán này nếu không có thêm các cơ chế học đặc trưng sâu. Ngược lại, các lớp có đặc trưng từ vựng và ngữ nghĩa rõ ràng như Thể thao, Sức khỏe, Thế giới và Vi tính vẫn đạt kết quả tốt với F1-score từ 0.90 đến 0.97. Nhìn chung, SVM kết hợp Word2Vec chỉ phù hợp làm mô hình tham khảo, và kém hiệu quả hơn so với các mô hình SVM dựa trên TF-IDF trong nghiên cứu này.

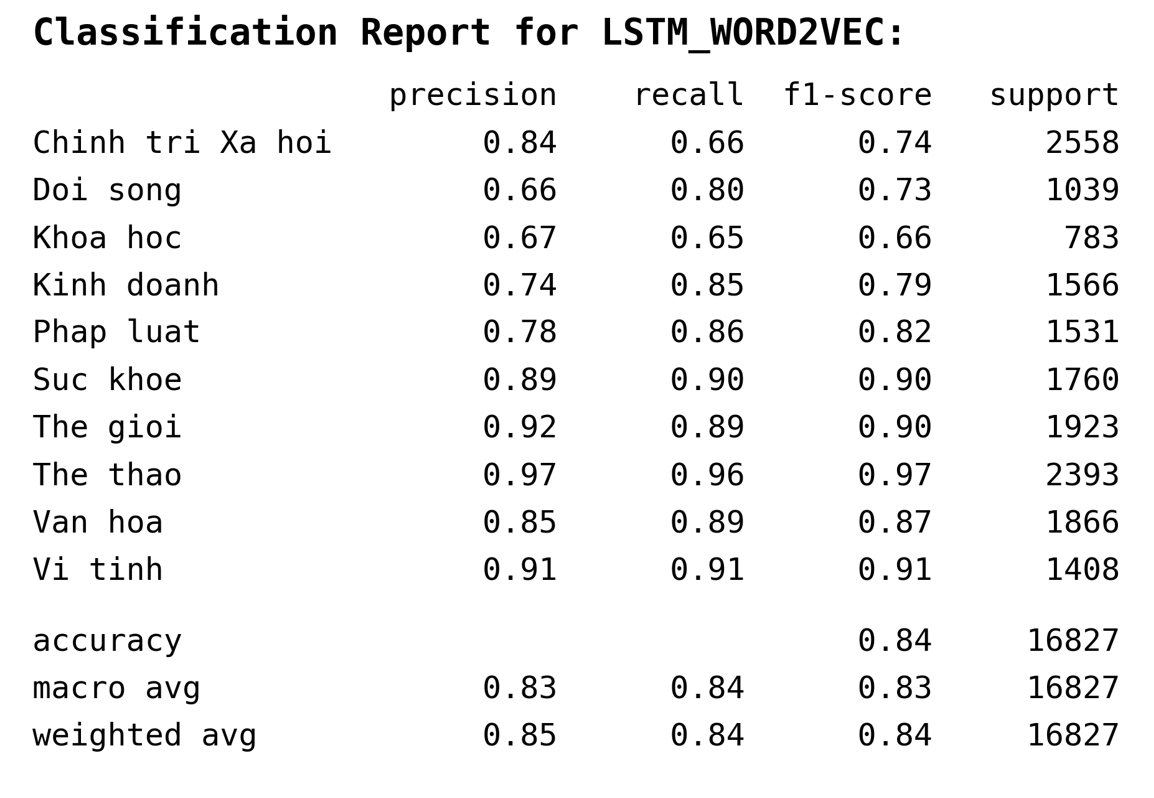
### ***3.3.11. Mô hình DNN kết hợp với Word2Vec tiền huấn luyện:***



## Hình 3.14. Chỉ số đánh giá của mô hình Mô hình Mạng nơ-ron (DNN) kết hợp với Word2Vec tiền huấn luyện.

Mô hình mạng nơ-ron (DNN) kết hợp với Word2Vec tiền huấn luyện đạt Accuracy = 0.88, cho thấy hiệu năng phân loại tốt hơn so với các mô hình tuyến tính sử dụng Word2Vec, nhưng vẫn chưa vượt trội so với các mô hình dựa trên TF-IDF. Các chỉ số Macro average = 0.86 và Weighted average = 0.88 phản ánh khả năng tổng quát hóa ở mức khá. So với Logistic Regression và SVM khi kết hợp với Word2Vec, DNN cho thấy khả năng khai thác thông tin ngữ nghĩa tốt hơn, đặc biệt ở các lớp như Văn hóa, Vi tính, Thế giới và Sức khỏe, với F1-score đạt từ 0.90 đến 0.92. Điều này cho thấy mô hình học sâu có lợi thế trong việc học các mối quan hệ phi tuyến trong không gian embedding. Tuy nhiên, một số lớp như Đời sống và Khoa học vẫn có F1-score thấp hơn, cho thấy Word2Vec dù mang thông tin ngữ nghĩa vẫn chưa đủ phân biệt rõ ràng các chủ đề có nội dung giao thoa. Nhìn chung, DNN kết hợp Word2Vec là lựa chọn phù hợp trong nhóm mô hình Word2Vec, nhưng chưa phải là phương pháp tối ưu cho bài toán phân loại văn bản tiếng Việt trong nghiên cứu này.

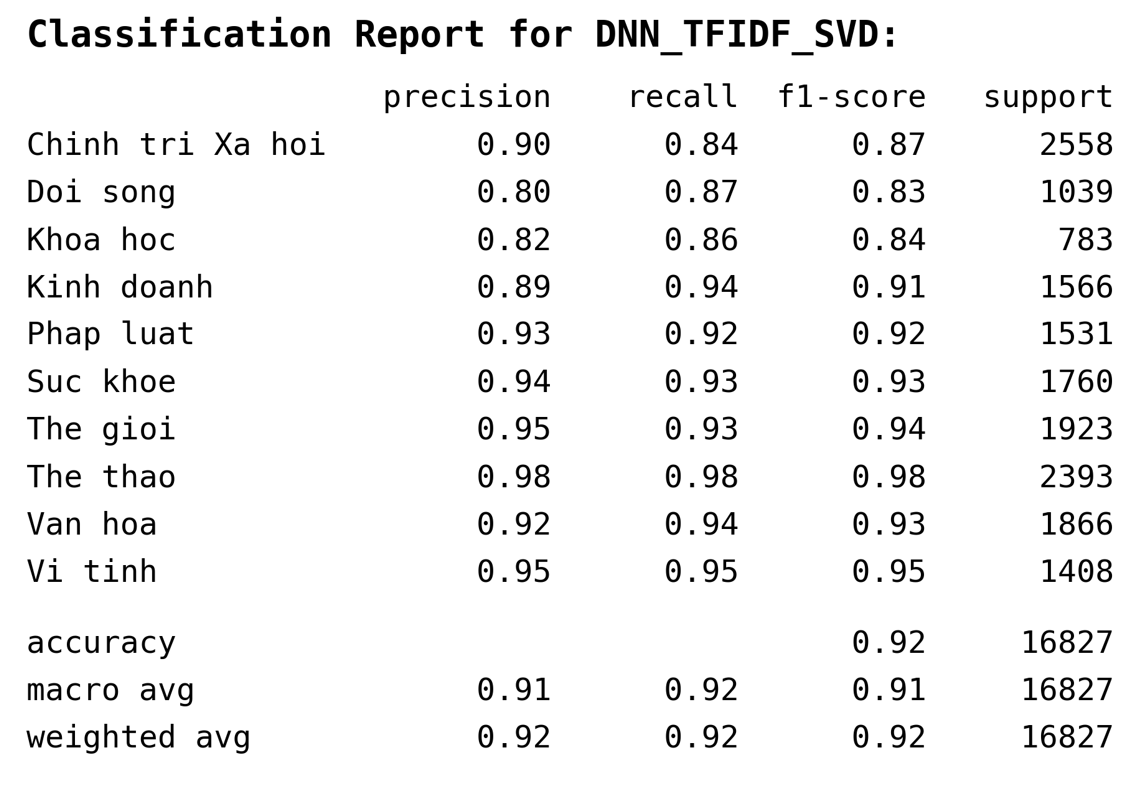
### ***3.3.12. Mô hình LSTM kết hợp với Word2Vec tiền huấn luyện:***



## Hình 3.15. Chỉ số đánh giá của mô hình LSTM kết hợp với Word2Vec tiền huấn luyện.

Mô hình LSTM kết hợp với Word2Vec tiền huấn luyện đạt Accuracy = 0.84, cho thấy hiệu năng phân loại ở mức trung bình và chưa vượt trội so với các mô hình học sâu khác trong nghiên cứu. Các chỉ số Macro average = 0.83 và Weighted average = 0.84 phản ánh khả năng tổng quát hóa còn hạn chế giữa các lớp. Một số lớp như Chính trị – Xã hội, Đời sống và Khoa học có F1-score thấp, cho thấy LSTM chưa khai thác hiệu quả ngữ cảnh chuỗi trong trường hợp văn bản có nội dung giao thoa và độ dài không đồng nhất. Điều này có thể xuất phát từ việc Word2Vec là embedding tĩnh, chưa phản ánh đầy đủ ngữ cảnh phụ thuộc vào vị trí từ trong câu. Ngược lại, các lớp có cấu trúc và từ vựng đặc trưng rõ ràng như Thể thao, Sức khỏe, Thế giới và Vi tính vẫn đạt kết quả tốt với F1-score từ 0.90 đến 0.97. Nhìn chung, LSTM kết hợp Word2Vec không mang lại cải thiện đáng kể so với DNN kết hợp Word2Vec, và chưa phải là lựa chọn tối ưu cho bài toán phân loại văn bản tiếng Việt trong nghiên cứu này.

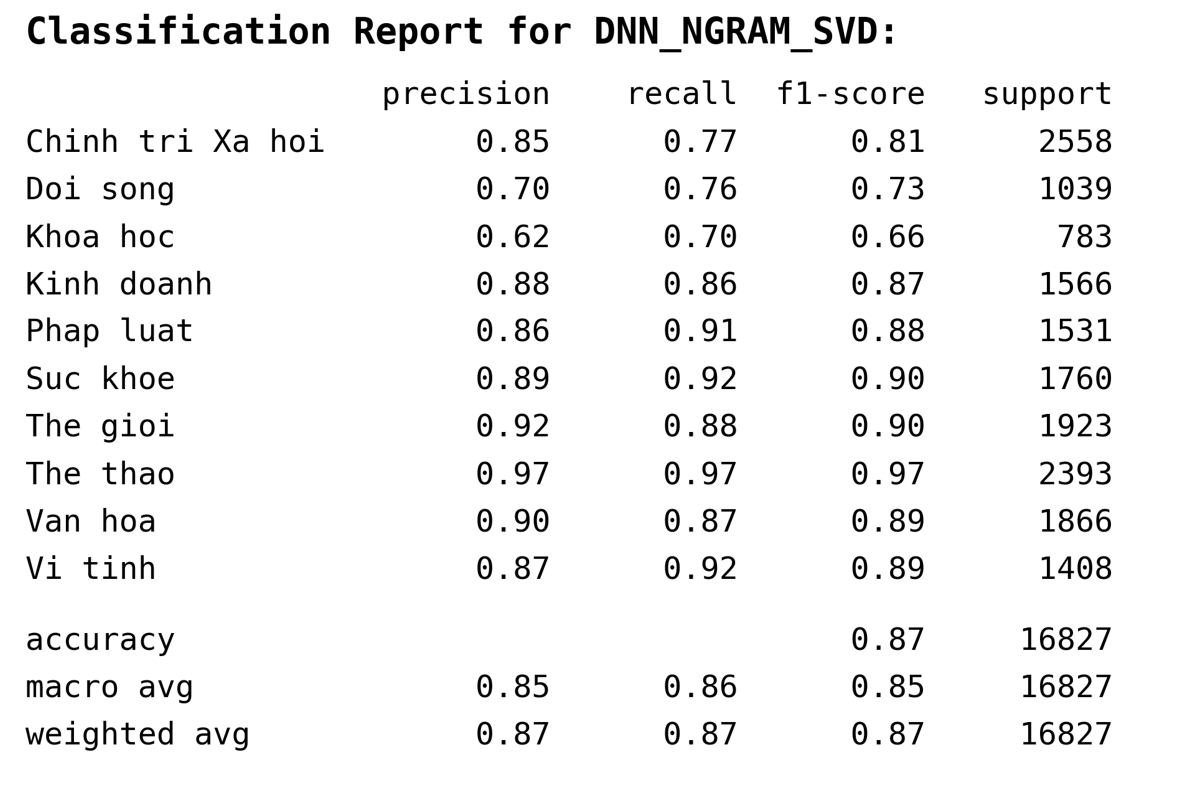
### ***3.3.13. Mô hình DNN kết hợp với TF-IDF và SVD:***



## Hình 3.16. Chỉ số đánh giá của mô hình Mạng nơ-ron sâu DNN kết hợp với TF-IDF và SVD.

Mô hình mạng nơ-ron sâu (DNN) kết hợp với đặc trưng TF-IDF và giảm chiều bằng SVD đạt Accuracy = 0.92, cho thấy hiệu năng phân loại cao và ổn định trên tập dữ liệu văn bản tiếng Việt theo chủ đề. Các chỉ số Macro average = 0.91 và Weighted average = 0.92 phản ánh khả năng tổng quát hóa tốt giữa các lớp. Hầu hết các lớp đều đạt Precision, Recall và F1-score cao, đặc biệt các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Vi tính, Thế giới và Sức khỏe với F1-score đạt từ 0.93 đến 0.98. Điều này cho thấy DNN có khả năng học các mối quan hệ phi tuyến hiệu quả trong không gian đặc trưng đã được giảm chiều. Tuy nhiên, so với mô hình SVM kết hợp TF-IDF, hiệu năng của DNN kết hợp TF-IDF và SVD không vượt trội, cho thấy việc giảm chiều bằng SVD có thể làm mất đi một phần thông tin quan trọng đối với mô hình học sâu. Nhìn chung, DNN kết hợp TF-IDF và SVD là phương pháp hiệu quả, nhưng chưa phải là lựa chọn tối ưu nhất trong nghiên cứu này.

### ***3.3.14. Mô hình DNN kết hợp với đặc trưng N-gram và SVD:***



## Hình 3.17. Chỉ số đánh giá của mô hình DNN kết hợp với đặc trưng N-gram và SVD.

Mô hình mạng nơ-ron sâu (DNN) kết hợp với đặc trưng N-gram và giảm chiều bằng SVD đạt Accuracy = 0.87, cho thấy hiệu năng phân loại ở mức khá nhưng thấp hơn so với các mô hình sử dụng TF-IDF. Các chỉ số Macro average = 0.85 và Weighted average = 0.87 phản ánh sự suy giảm hiệu quả trên một số lớp. Một số lớp như Đời sống và đặc biệt là Khoa học có F1-score thấp (0.73 và 0.66), cho thấy việc sử dụng đặc trưng N-gram sau khi giảm chiều chưa khai thác tốt các mối quan hệ ngữ nghĩa phức tạp. Điều này cho thấy DNN không phát huy tối đa lợi thế khi kết hợp với N-gram và SVD trong bài toán phân loại chủ đề. Ngược lại, các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Sức khỏe, Thế giới và Văn hóa vẫn đạt kết quả tốt với F1-score từ 0.89 đến 0.97. Nhìn chung, DNN kết hợp N-gram và SVD mang lại hiệu quả hạn chế, và kém hơn đáng kể so với mô hình DNN kết hợp TF-IDF và SVD trong nghiên cứu này.

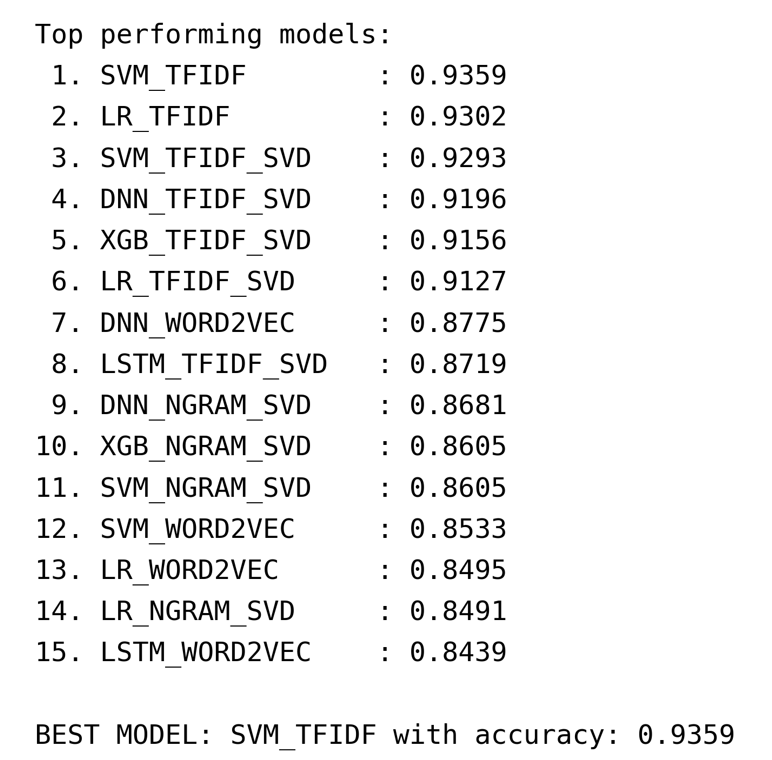
### ***3.3.15. Mô hình LSTM kết hợp với TF-IDF và SVD:***



## Hình 3.18. Chỉ số đánh giá của mô hình Mạng nơ-ron hồi tiếp LSTM kết hợp với TF-IDF và SVD.

Mô hình mạng nơ-ron hồi tiếp LSTM kết hợp với đặc trưng TF-IDF và giảm chiều bằng SVD đạt Accuracy = 0.87, cho thấy hiệu năng phân loại ở mức khá nhưng không vượt trội so với các mô hình học máy truyền thống và DNN trong nghiên cứu. Các chỉ số Macro average = 0.86 và Weighted average = 0.87 phản ánh khả năng tổng quát hóa ở mức trung bình. Một số lớp như Chính trị – Xã hội, Đời sống và Khoa học có F1-score tương đối thấp, cho thấy việc sử dụng TF-IDF (vốn là đặc trưng không tuần tự) chưa khai thác được lợi thế mô hình hóa ngữ cảnh theo chuỗi của LSTM. Điều này làm hạn chế hiệu quả của LSTM trong bài toán này. Ngược lại, các lớp có đặc trưng từ vựng rõ ràng như Thể thao, Sức khỏe, Thế giới và Vi tính vẫn đạt kết quả tốt với F1-score từ 0.90 đến 0.97. Nhìn chung, LSTM kết hợp TF-IDF và SVD không phải là lựa chọn phù hợp, và kém hiệu quả hơn so với việc kết hợp LSTM với các biểu diễn ngữ nghĩa tuần tự như Word2Vec trong nghiên cứu này.

## 3.4. Tổng hợp kết quả:



**Nhận xét:**

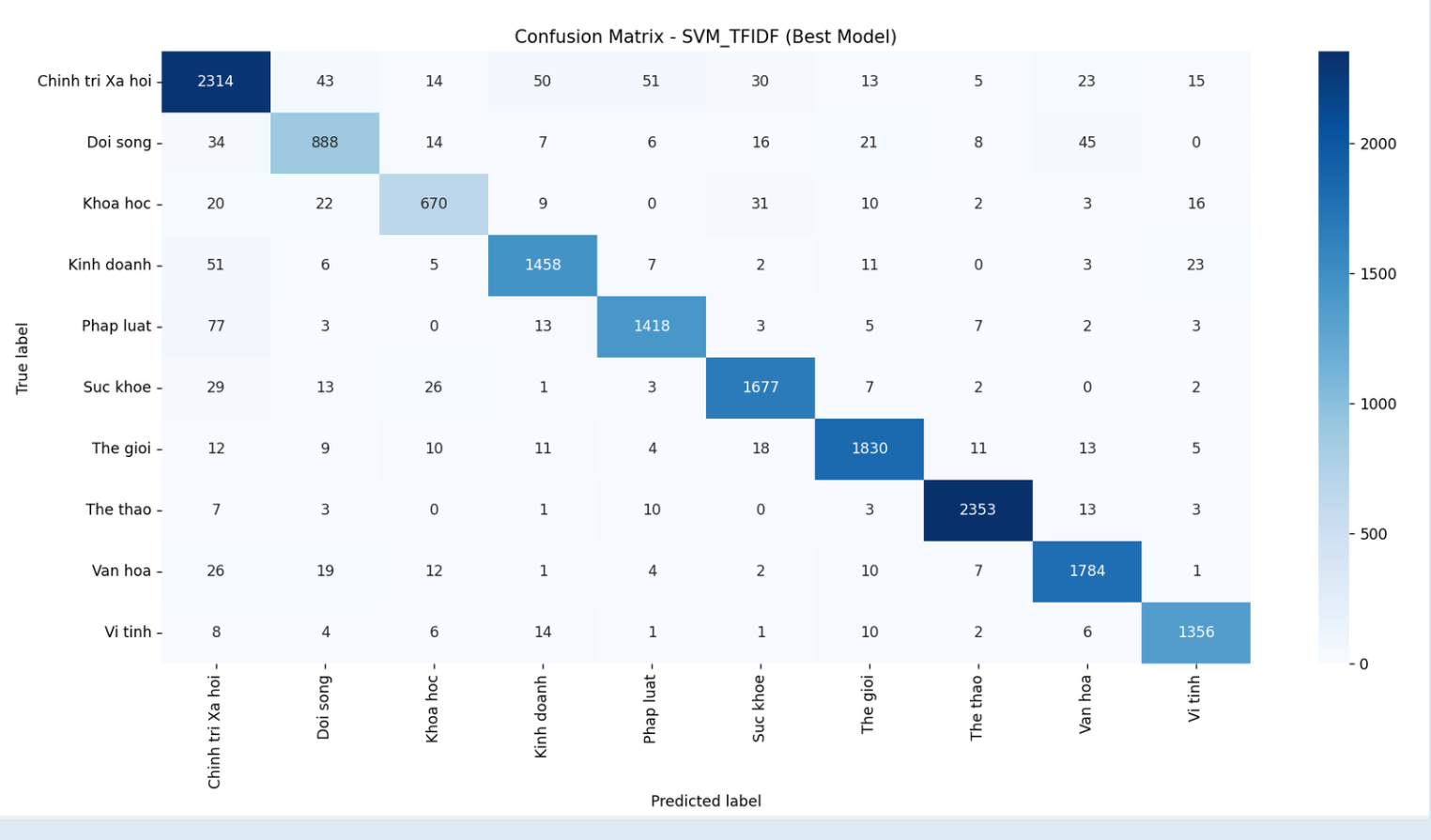
Thông qua quá trình thực nghiệm và so sánh nhiều mô hình phân loại văn bản khác nhau, kết quả cho thấy cách biểu diễn đặc trưng đóng vai trò quyết định đến hiệu năng của mô hình trong bài toán phân loại văn bản tiếng Việt theo chủ đề. Nhìn chung, các mô hình sử dụng đặc trưng TF-IDF cho kết quả vượt trội hơn rõ rệt so với các mô hình dựa trên N-gram hoặc Word2Vec tiền huấn luyện.

Trong số các mô hình được khảo sát, mô hình Máy vector hỗ trợ (SVM) kết hợp với TF-IDF đạt độ chính xác cao nhất (Accuracy = 0.9359) và cho kết quả ổn định trên hầu hết các lớp chủ đề. Điều này cho thấy TF-IDF vẫn là biểu diễn đặc trưng rất hiệu quả trong việc nắm bắt thông tin từ vựng đặc trưng của từng chủ đề, đồng thời SVM có khả năng phân tách tốt trong không gian đặc trưng có chiều cao. Việc áp dụng giảm chiều dữ liệu bằng SVD giúp giảm độ phức tạp của không gian đặc trưng và cải thiện hiệu quả tính toán, tuy nhiên không làm tăng đáng kể độ chính xác, thậm chí trong một số trường hợp còn làm suy giảm nhẹ hiệu năng do mất mát thông tin đặc trưng. Các mô hình học sâu như DNN kết hợp với TF-IDF và SVD cho kết quả khá tốt, nhưng vẫn chưa vượt qua được các mô hình học máy truyền thống trong thiết lập thí nghiệm của đề tài.

Đối với các mô hình sử dụng Word2Vec tiền huấn luyện, mặc dù có khả năng biểu diễn ngữ nghĩa, nhưng khi kết hợp với các mô hình tuyến tính hoặc học sâu đơn giản, hiệu năng chưa cao. Nguyên nhân chủ yếu là do Word2Vec là embedding tĩnh, chưa phản ánh đầy đủ ngữ cảnh, và chưa phù hợp tối ưu cho bài toán phân loại chủ đề trong phạm vi dữ liệu nghiên cứu.

Tổng hợp các kết quả trên, mô hình SVM kết hợp với TF-IDF được lựa chọn là mô hình tốt nhất cho hệ thống, nhờ đạt độ chính xác cao, ổn định giữa các lớp và có chi phí tính toán hợp lý. Mô hình này được xem là phương án phù hợp để triển khai cho bài toán phân loại văn bản tiếng Việt theo chủ đề trong thực tế.

## 3.5. Đánh giá ma trận nhầm lẫn:



## Hình 3.19. Ma trận nhầm lẫn của mô hình SVM kết hợp TF-IDF.

Ma trận nhầm lẫn của mô hình SVM kết hợp với TF-IDF (mô hình tốt nhất) cho thấy phần lớn các mẫu được phân loại đúng, thể hiện qua việc các giá trị trên đường chéo chính chiếm ưu thế rõ rệt. Điều này phản ánh khả năng phân biệt chủ đề hiệu quả của mô hình trong không gian đặc trưng TF-IDF. Các lớp có số lượng mẫu lớn và đặc trưng từ vựng rõ ràng như Thể thao, Thế giới, Sức khỏe, Văn hóa và Vi tính có tỷ lệ dự đoán đúng rất cao. Ví dụ, lớp Thể thao có số lượng mẫu được phân loại đúng áp đảo so với số mẫu bị nhầm sang các lớp khác, cho thấy mô hình học tốt các từ khóa đặc trưng và cấu trúc nội dung riêng biệt của lớp này. Điều này phù hợp với các chỉ số Precision, Recall và F1-score rất cao đã được trình bày trong báo cáo đánh giá.

Ngược lại, một số lớp như Đời sống, Khoa học và Chính trị – Xã hội xuất hiện hiện tượng nhầm lẫn nhiều hơn. Cụ thể, các mẫu thuộc lớp Đời sống có xu hướng bị dự đoán nhầm sang Chính trị – Xã hội hoặc Văn hóa, trong khi lớp Khoa học đôi khi bị nhầm với Sức khỏe hoặc Vi tính do các bài viết khoa học thường đề cập đến y sinh, công nghệ hoặc ứng dụng kỹ thuật. Đây là hiện tượng hợp lý do sự giao thoa về nội dung và từ vựng giữa các chủ đề này trong thực tế. Điều này cho thấy mô hình có tính nhất quán về mặt ngữ nghĩa, và các sai sót chủ yếu đến từ ranh giới chủ đề không rõ ràng, thay vì do mô hình học kém.

Tập dữ liệu trong nghiên cứu có sự mất cân bằng giữa các lớp, thể hiện qua sự chênh lệch lớn về số lượng mẫu (support) giữa các chủ đề. Tuy nhiên, mô hình vẫn duy trì hiệu năng tốt trên các lớp ít mẫu hơn nhờ biện pháp xử lý mất cân bằng đã được áp dụng trong quá trình huấn luyện. Cụ thể, mô hình SVM được huấn luyện với trọng số lớp (class weight = balanced), giúp tăng mức độ quan tâm của mô hình đối với các lớp có số lượng mẫu ít. Nhờ đó, mặc dù các lớp như Khoa học hoặc Đời sống có số lượng mẫu thấp hơn so với Chính trị – Xã hội hay Thể thao, mô hình vẫn đạt Recall và F1-score ở mức chấp nhận được, tránh hiện tượng bỏ sót nghiêm trọng các lớp thiểu số. Điều này được thể hiện rõ qua việc không có lớp nào bị suy giảm mạnh về Recall, và các lỗi phân loại chủ yếu mang tính cục bộ giữa các chủ đề gần nghĩa, thay vì do thiên lệch dữ liệu.

Tóm lại, ma trận nhầm lẫn cho thấy mô hình SVM kết hợp TF-IDF không chỉ đạt độ chính xác cao về mặt tổng thể mà còn duy trì được sự cân bằng tương đối giữa các lớp, ngay cả trong bối cảnh dữ liệu không đồng đều. Các lỗi phân loại xảy ra có tính hợp lý về ngữ nghĩa và phản ánh đúng đặc thù của bài toán phân loại văn bản theo chủ đề. Kết quả này khẳng định rằng mô hình đã học được các đặc trưng phân biệt quan trọng, xử lý tốt vấn đề mất cân bằng dữ liệu và hoàn toàn phù hợp để lựa chọn làm mô hình cuối cùng cho hệ thống phân loại văn bản tiếng Việt theo chủ đề.

# 

# Chương IV. KẾT LUẬN

Trong bài nghiên cứu này nhóm đã thử nghiệm và đánh giá hiệu quả của các phương pháp phân loại văn bản Tiếng Việt, sử dụng các trích xuất đặc trưng tĩnh kết hợp với các phương pháp học máy và học sâu. Trong đó, SVM kết hợp với Tf-Idf cho kết quả tốt nhất. Phương pháp học máy cho ra kết quả vượt trội khi sử dụng cùng Tf-Idf. Kết quả cũng cho thấy Word2Vec cho ra kết quả khá tệ so với tiềm năng nó có thể mang lại cho thấy với bài toán đơn giản như phân loại văn bản không cần phụ thuộc vào ngữ cảnh của nó trong câu mà là dựa vào đặc điểm của các từ khóa hơn. Nhóm vẫn chưa thực nghiệm với phương pháp contextual embedding bởi hạn chế về phần cứng.

Trong tương lai, Nhóm sẽ áp dụng phương pháp contextual embedding và nghiên cứu sâu hơn về cơ chế Attention của Transformer, đồng thời mở rộng sang các tác vụ xử lý ngôn ngữ khác như phân tích cảm xúc, tóm tắt văn bản tiếng Việt.

# ĐÓNG GÓP CỦA CÁC THÀNH VIÊN

| Trần Gia Vũ | Nghiên cứu bài toán và các bài báo |
| --- | --- |
| Nguyễn Bá Nhật | xây dựng và phát triển mô hình phân loại |
| Lê Ngọc Ánh | Soạn báo cáo powerpoint và word |

# 

# 

# TÀI LIỆU THAM KHẢO

**[1]** [Comparative Study of Static and Contextual Text Vectorization for Sentiment Analysis (2025) - A D Bhargavi.](https://www.ijraset.com/research-paper/static-and-contextual-text-vectorization-for-sentiment-analysis)

**[2]** [A comparative study of word embedding techniques for classification of star ratings - Expert Systems with Applications (Volume 297, Part A,) - H.A, C.M, M.W.](https://www.researchgate.net/publication/394193261_A_Comparative_Study_of_Word_Embedding_Techniques_for_Classification_of_Star_Ratings)

**[3]** [Word Embedding: TF-IDF (2024) - Hex.](https://youtu.be/x1u5TotQ0G0?si=HiQRijpLDcwSEKvv)

**[4]** [Phân tích và thực nghiệm các phương pháp phân loại bài báo tiếng Việt (2025) - Kiệt Bùi Quang Tấn, Lang Nguyen Thi.](https://www.researchgate.net/publication/393163546_Phan_tich_va_thuc_nghiem_cac_phuong_phap_phan_loai_bai_bao_tieng_Viet)

**[5]** [Word Embeddings: Word2Vec (2024) - Hex.](https://youtu.be/iErmK_sJtag?si=h_erAnGUiEjzqV8n)

**[6]** [Transformer Embeddings (2022) - CodeEmporium.](https://youtu.be/l4is4uHvKlU?si=KTzUWZMLu9mbyOiA)

**[7]** [What are the main differences between the word embeddings of ELMo, BERT, Word2vec, and GloVe? - Quora.](https://www.quora.com/What-are-the-main-differences-between-the-word-embeddings-of-ELMo-BERT-Word2vec-and-GloVe)

**[8]** [Khảo sát các mô hình phân loại văn bản tiếng việt (2022) - Nguyễn Chí Hiếu.](https://jst.iuh.edu.vn/index.php/jst-iuh/article/download/4395/663/10039)