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

HỌC MÁY - CO3117

Bài tập lớn 2

Phân loại lĩnh vực câu hỏi

Subject classification from Questions

Giảng viên hướng dẫn: TS. Lê Thành Sách
Lớp: TN01
Nhóm: DNA05

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 10 - 2025

BÁO CÁO KẾT QUẢ LÀM VIỆC NHÓM

STT	MSSV	Họ và Tên	Nhiệm vụ phụ trách	Mức độ hoàn thành
1	2310167	Tăng Hồng Ái	Tiền xử lý dữ liệu: loại bỏ dữ liệu trùng lặp. Chuẩn hoá ký tự và xử lý dữ liệu thiếu. Chuẩn bị dữ liệu đầu vào cho pipeline ML/DL. Tham gia viết phần báo cáo về tiền xử lý dữ liệu.	100%
2	2310510	Phạm Khánh Duy	Thu thập và lựa chọn dataset phù hợp. Phân tích khám phá dữ liệu (EDA), trực quan hóa và thống kê tổng quan. Hỗ trợ đánh giá chất lượng dữ liệu và viết báo cáo.	100%
3	2312506	Nguyễn Trần Yến Nhi	Tìm tham số tối ưu và huấn luyện các mô hình ML/DL. Đánh giá kết quả: Accuracy, Precision, Recall, F1-score. Viết phần báo cáo về kết quả mô hình và phân tích.	100%

Bảng 1: Phân công công việc và mức độ hoàn thành nhóm

Mục lục

1 EDA

1.1 Tổng quan dữ liệu

Bộ dữ liệu `student_questions.csv` có 122.519 mẫu, và 2 cột, tương ứng với hơn một trăm hai mươi nghìn câu hỏi của học sinh chuẩn bị cho các kỳ thi IIT-JEE, NEET và AIIMS - là các kỳ thi đầu vào quan trọng tại Ấn Độ. Mỗi dòng đại diện cho một câu hỏi bằng tiếng Anh cùng với môn học tương ứng mà câu hỏi thuộc về. Bộ dữ liệu gồm hai cột chính:

- `eng` (object): nội dung câu hỏi bằng tiếng Anh.
- `Subject` (object): thể hiện môn học tương ứng với câu hỏi, chẳng hạn như Maths, Physics, Chemistry, hoặc Biology.

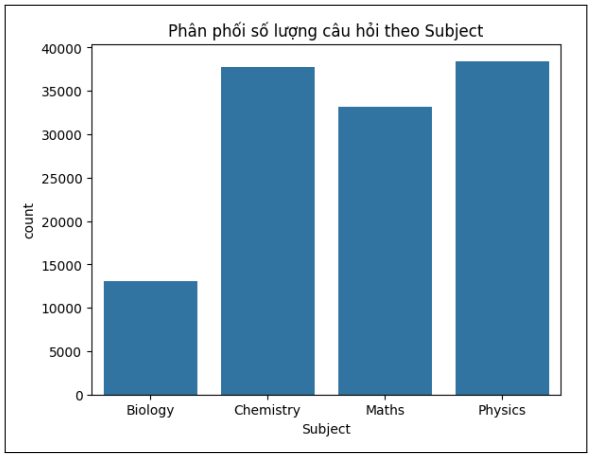
Cả hai cột đều có kiểu dữ liệu là object, tức là dạng chuỗi ký tự (textual data).

Feature	Number of Missing Values
`eng`	0
`Subject`	0

Bảng 2: Thống kê số lượng Missing Values cho từng cột.

Điều này có nghĩa là bộ dữ liệu hoàn toàn đầy đủ, không có giá trị bị thiếu ở bất kỳ cột nào. Vì vậy, ta không cần áp dụng kỹ thuật xử lý giá trị khuyết như điền trung bình, loại bỏ dòng hoặc suy luận giá trị.

1.2 Phân bố câu hỏi theo môn học



Hình 1.1: Thống kê tần suất nhãn

Subject	Proportion
Physics	31.37%
Chemistry	30.83%
Maths	27.09%
Biology	10.71%

Bảng 3: Thống kê tỉ lệ nhãn

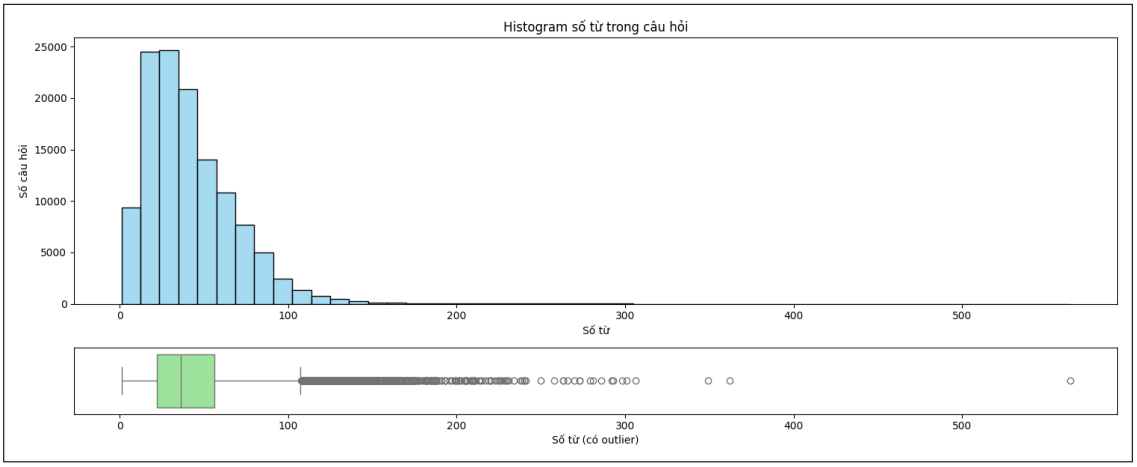
Bộ dữ liệu bao gồm bốn môn học chính: Physics, Chemistry, Maths, và Biology. Từ các bảng thống kê trên có thể nhận thấy rằng:

- Physics và Chemistry chiếm tỉ trọng gần như tương đương, mỗi môn khoảng 30% tổng số câu hỏi, cho thấy hai lĩnh vực này được nhấn mạnh nhiều trong bộ dữ liệu.
- Maths chiếm khoảng 27%, tức là cũng có lượng câu hỏi khá lớn, chỉ thấp hơn một chút so với hai môn đầu.
- Biology là môn có số lượng câu hỏi ít nhất (khoảng 10.7%), thấp hơn đáng kể so với các môn còn lại.

Điều này phản ánh sự mất cân bằng (imbalance) nhất định trong dữ liệu — một yếu tố cần lưu ý nếu sau này tiến hành các bài toán phân loại theo môn học (text classification), vì mô hình có thể dễ dàng nghiêng về các lớp có nhiều dữ liệu hơn (như Physics hoặc Chemistry).

Về mặt nội dung, sự phân bố này cũng hợp lý nếu xét đến thực tế rằng bộ dữ liệu kết hợp từ các kỳ thi IIT-JEE (thiên về Physics, Chemistry, và Maths) và NEET/AIIMS (có thêm Biology). Do đó, sự chênh lệch tỉ lệ phản ánh rõ đặc trưng của từng kỳ thi.

1.3 Phân tích mô tả dữ liệu văn bản



Hình 1.2: Phân phối số từ trong câu hỏi

Biểu đồ trên thể hiện phân phối số từ trong các câu hỏi thuộc cột `eng`. Histogram biểu diễn tần suất xuất hiện theo độ dài câu hỏi. Boxplot giúp quan sát giá trị trung vị, độ phân tán và các outlier. Kết quả cho thấy:

- Phần lớn các câu hỏi có độ dài ngắn đến trung bình, tập trung trong khoảng 10 – 60 từ.
- Phân phối nghiêng về bên phải (right-skewed), nghĩa là có một số ít câu hỏi dài bất thường kéo dài đến hơn 200 từ, thậm chí có trường hợp vượt quá 500 từ.
- Outlier (giá trị ngoại lai) chiếm 2.533 câu hỏi, tương đương khoảng 2% tổng dữ liệu, chủ yếu là các câu mô tả tình huống dài hoặc có nhiều công thức.

ID	Question
1172	Define refraction
1714	Define ecosystem.
1968	Define Pteridophyta
2824	tood\nE
4159	tood\nyo
4998	Match following
5177	Define Eccentricity?
5633	Explain rancidity.
5857	1
8410	2

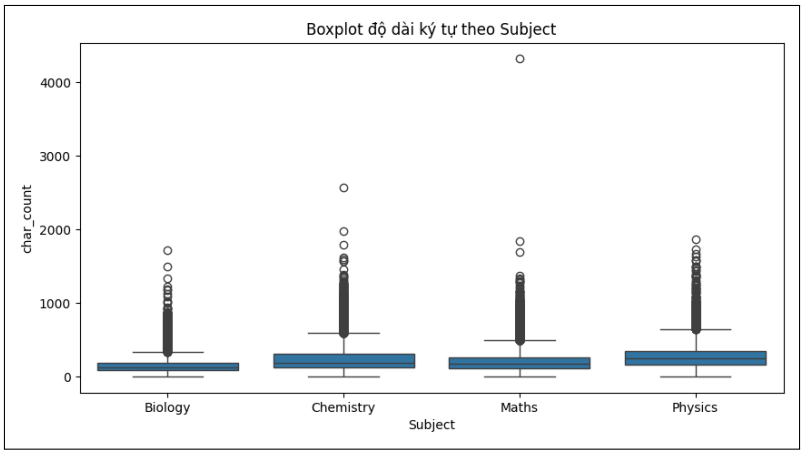
Bảng 4: Ví dụ các câu hỏi quá ngắn

ID	Question
427	Which one of the following statements\nare cor...
3296	The position vector of a particle of mass \(\...
4135	Q Type your question\n\[\n\underset{\mathbf{M ...
8943	toppr\nQ Type your question-\n\(A \)\n\begin{...
9383	toppr\nQ Type your question\nreactions that ar...

Bảng 5: Ví dụ các câu hỏi quá dài

Trong toàn bộ dữ liệu, có 98 câu quá ngắn (dưới 3 từ) và 64 câu quá dài (trên 200 từ). Từ các ví dụ cho thấy:

- Câu quá ngắn thường là những câu yêu cầu định nghĩa hoặc cụm lệnh ngắn, những trường hợp này có thể là lỗi nhập liệu hoặc câu hỏi chưa hoàn chỉnh.
- Câu quá dài thường chứa nhiều dòng mô tả, công thức hoặc dữ liệu định dạng toán học, điển hình cho các đề bài mô tả thí nghiệm hoặc bài toán phức tạp.



Hình 1.3: Phân phối số kí tự trong câu hỏi theo môn học

Boxplot cho thấy sự khác biệt giữa các môn học:

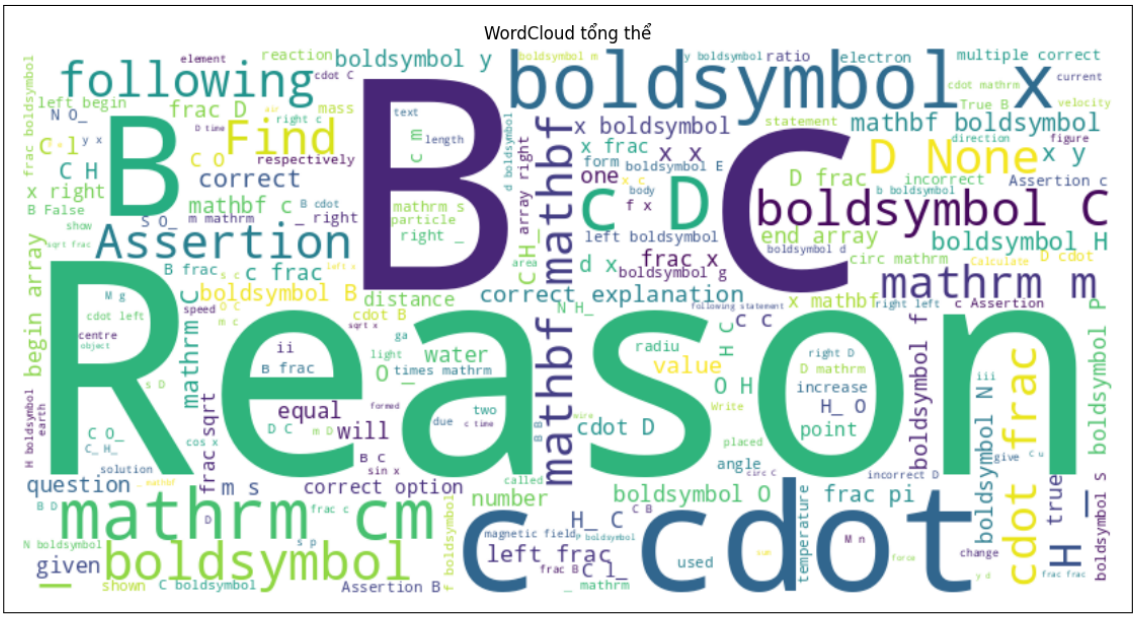
- Physics và Chemistry có phân phối độ dài tương đối tương đồng, nhiều câu hỏi trung bình 50 – 80 ký tự, nhưng có đuôi dài do chứa công thức.
- Câu hỏi Mathematics có xu hướng ngắn hơn, tập trung quanh 40 – 60 ký tự.
- Biology có nhiều câu dài hơn trung bình, thường dùng mô tả ngôn ngữ tự nhiên nhiều hơn công thức.

Về đa dạng từ vựng, số lượng từ duy nhất (vocabulary size) trong từng môn được thống kê như sau:

Subject	**Vocabulary**
Biology	20.064
Chemistry	44.104
Maths	49,628
Physics	39,098

Bảng 6: Thống kê từ vựng trong từng môn học

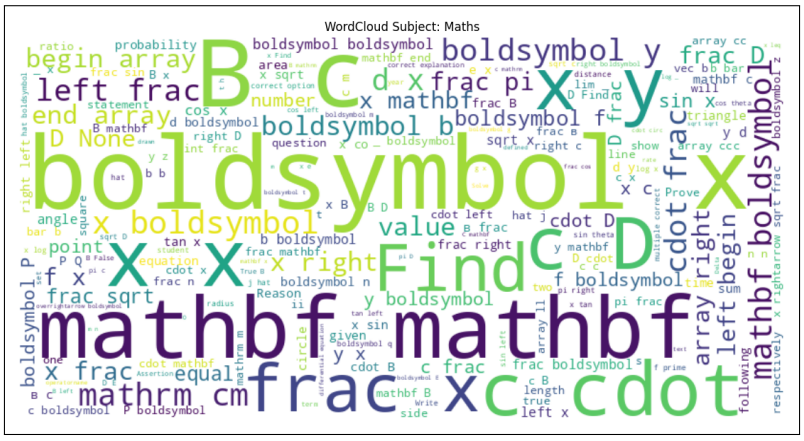
Điều này cho thấy Maths có quy mô từ vựng lớn nhất, đạt gần 50.000 từ. Chemistry và Physics có số lượng từ vựng trung bình. Biology có quy mô nhỏ hơn, khoảng 20.000 từ. Sự chênh lệch này phản ánh độ đa dạng và đặc trưng ngôn ngữ khác nhau giữa các môn, do đó cần chiến lược xử lý riêng cho các câu hỏi trong từng môn học.



Hình 1.4: Word Cloud cho toàn bộ dữ liệu

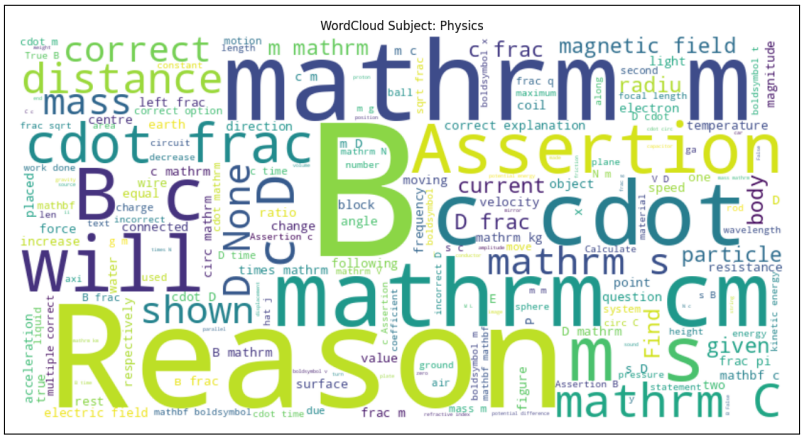
Hình trên thể hiện WordCloud tổng thể của toàn bộ tập câu hỏi trong cột `eng`. Mỗi từ được hiển thị với kích thước tỷ lệ thuận với tần suất xuất hiện trong tập dữ liệu, giúp quan sát nhanh những từ phổ biến nhất. Có thể nhận thấy rằng:

- Các từ xuất hiện nhiều nhất gồm `Reason`, `Assertion`, `correct`, `following`, `option`, `find`, `true`, `false`, phản ánh đặc trưng của dạng câu hỏi trắc nghiệm suy luận.
- Các ký hiệu LaTeX như `mathbf`, `cdot`, `frac` cho thấy dữ liệu chứa nhiều công thức toán – lý, mang tính học thuật cao.



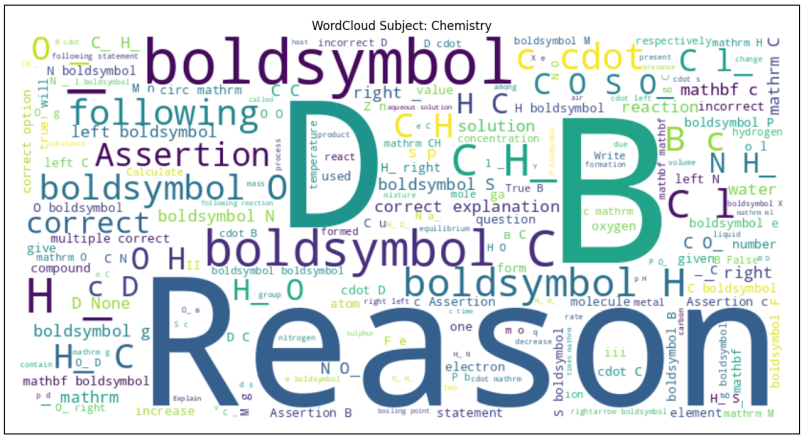
Hình 1.5: Word Cloud cho môn Maths

WordCloud của Maths có nhiều ký hiệu và từ khóa toán học như `value`, `x`, `y`, `find`, `ratio`, `area`, `probability` thể hiện đặc trưng ngôn ngữ của Maths là ngắn gọn, cấu trúc chặt, độ lặp cao của biến và ký hiệu.



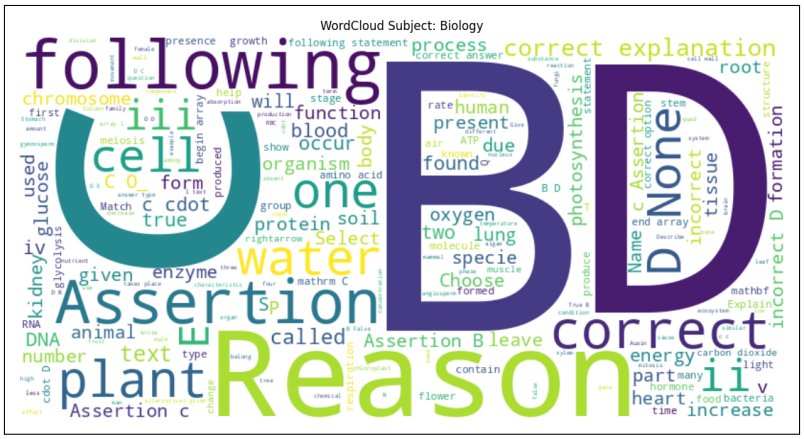
Hình 1.6: Word Cloud cho môn Physics

WordCloud của Physics nhiều ký hiệu và đại lượng vật lý như `velocity`, `force`, `field`, `mass`, `charge`, `motion`, `current`. Các từ này thường đi kèm với ký hiệu toán học hoặc đơn vị đo, thể hiện đặc trưng ngôn ngữ mang tính công thức và định lượng.



Hình 1.7: Word Cloud cho môn Chemistry

WordCloud của Chemistry xuất hiện nhiều tên chất và các ký hiệu hóa học như `molecule`, `water`, `electron`, `equilibrium`, `compound`. Đặc trưng nổi bật là sự pha trộn giữa từ ngữ tự nhiên và ký hiệu hóa học, cho thấy dạng câu hỏi thường bao gồm cả mô tả phản ứng lẫn công thức biểu diễn.



Hình 1.8: Word Cloud cho môn Biology

WordCloud của Biology thể hiện nhiều từ mang tính mô tả và danh từ học thuật như `organism`, `cell`, `enzyme`, `water`, `energy`. Khác với các môn còn lại, Biology ít chứa ký hiệu, thay vào đó là câu hỏi dạng định nghĩa hoặc mô tả hiện tượng sinh học. Từ vựng phân tán hơn, ít lặp lại so với các môn còn lại, cho thấy ngôn ngữ mô tả trong nhóm này đa dạng và có tính tự nhiên hơn.

Nhìn chung, các WordCloud cho thấy sự đồng nhất trong cấu trúc câu hỏi trắc nghiệm, nhưng vẫn phản ánh đặc trưng ngữ nghĩa riêng của từng môn, rất hữu ích cho việc huấn luyện mô hình phân loại hoặc sinh câu hỏi theo chủ đề. Tuy nhiên, việc xuất hiện dày đặc các token LaTeX như `boldsymbol`, `cdot`, `frac`... cho thấy cần bước xử lý làm sạch văn bản (text cleaning) để loại bỏ ký hiệu không mang ý nghĩa ngữ nghĩa trước khi huấn luyện mô hình ngôn ngữ.

1.4 Kiểm tra các giá trị trùng lặp

Trong quá trình kiểm tra dữ liệu, có tổng cộng 1.577 câu hỏi trùng lặp được phát hiện trong cột `eng`.

Question	Count	Subjects
`Match the column.`	12	Chemistry
`Match the following`	9	Biology, Chemistry, Maths, Physics
`Match the column`	7	Chemistry, Maths, Physics
`\( \frac{k}{k} \)`	7	Physics
`8\n8\n8\n8`	7	Chemistry

Bảng 7: Ví dụ các câu hỏi trùng lặp nhiều lần nhất

Những câu hỏi trên lặp lại nhiều lần trong bộ dữ liệu, với nội dung ngắn, không mang thông tin ngữ nghĩa đặc thù mà chỉ đóng vai trò hướng dẫn. Đặc biệt, các câu như `Match the column` hay `Match the following` thường xuất hiện trong phần đề gốc có hình minh họa hoặc bảng dữ

liệu đi kèm. Tuy nhiên trong tập dữ liệu này, chúng bị tách riêng, dẫn đến mất ngữ cảnh.

Việc xuất hiện nhiều câu hỏi trùng lặp, đặc biệt là các mẫu ngắn và vô nghĩa, cho thấy bộ dữ liệu có chứa nhiễu văn bản (text noise). Trong các bước xử lý tiếp theo, cần thực hiện bước lọc trùng lặp để tránh ảnh hưởng đến độ đa dạng và chất lượng của tập dữ liệu.

2 Tiền xử lý dữ liệu

2.1 Loại bỏ các giá trị trùng lặp

Bước đầu tiên trong quá trình tiền xử lý là loại bỏ các câu hỏi trùng lặp trong cột `eng`. Kết quả cho thấy có tổng cộng 1.577 câu hỏi bị trùng, chiếm một tỉ lệ nhỏ nhưng đáng kể trong toàn bộ dữ liệu. Ví dụ một số câu hỏi trùng lặp trong bộ dữ liệu

Question	Subject
`What type of a relation is "Less than" in the ...`	Maths
`Consider the following statements:\n1. The gen...`	Maths
`To determine the Young modulus of a wire, seve...`	Physics
`For the same mass, which one of the\nfollowing...`	Physics
`Which of the following steps of respiration is...`	Biology

Bảng 8: Ví dụ các câu hỏi trùng lặp

Sau khi loại bỏ các bản ghi trùng lặp, số lượng câu hỏi còn lại là 120.942, đảm bảo dữ liệu đầu vào không bị lặp thông tin, giúp mô hình huấn luyện phản ánh chính xác hơn phân bố ngôn ngữ của từng môn học.

2.2 Làm sạch dữ liệu

Do dữ liệu chứa nhiều dạng ký tự khác nhau (ký hiệu toán học, hóa học, đơn vị đo lường, công thức, v.v.), cần áp dụng một pipeline làm sạch có tính tổng quát và linh hoạt. Quy trình làm sạch được triển khai thông qua một **pipeline tiền xử lý văn bản**, với các bước tuần tự như sau:

- **Xóa ký tự xuống dòng** (`\n`) nhằm tránh việc tách câu hoặc dòng không mong muốn.
- **Loại bỏ dấu câu** để mô hình không bị nhiễu bởi các ký tự đặc biệt và tăng độ nhất quán của dữ liệu.
- **Chuẩn hóa từ nối (hyphenated words)**, ví dụ: `state-of-the-art` được chuyển thành `state of the art` để đảm bảo tokenizer hiểu đúng ngữ cảnh.
- **Chuẩn hóa bullet points**, loại bỏ các ký tự liệt kê như •, `-`, `*` thường xuất hiện trong đề bài hoặc danh sách.
- **Chuẩn hóa dấu ngoặc kép và khoảng trắng** giúp định dạng văn bản đồng nhất, hỗ trợ tokenizer xử lý chính xác hơn.

Sau khi áp dụng pipeline trên lên dữ liệu gốc, hai phiên bản dữ liệu được tạo ra để phục vụ cho các hướng mô hình khác nhau:

1. `ML-ready` – phục vụ cho **các mô hình Machine Learning truyền thống**, bao gồm các bước bổ sung:
   - **Chuyển toàn bộ về chữ thường (lowercase)** để giảm phân mảnh từ vựng (ví dụ "Apple" và "apple" được coi là một).
   - **Loại bỏ stopwords**: các mô hình dựa trên tần suất từ như Naive Bayes, SVM, Logistic Regression thường không cần các từ dừng, việc loại bỏ giúp giảm nhiễu và cải thiện độ khái quát.
2. `DL-ready` – phục vụ cho **các mô hình Deep Learning hiện đại**, trong đó:
   - Giữ nguyên chữ hoa/thường và stopwords nhằm bảo toàn ngữ cảnh ngữ pháp, giúp các mô hình như RNN, LSTM, BERT, Transformer học được mối quan hệ và trọng số ngữ nghĩa của từ trong câu.

Như vậy, tập `ML-ready` phù hợp cho các mô hình dựa trên vector đặc trưng như Bag-of-Words, TF-IDF hoặc CountVectorizer, trong khi `DL-ready` phù hợp cho các mô hình học ngữ cảnh sâu, vốn khai thác được cấu trúc và ngữ nghĩa tự nhiên của văn bản.

2.3 Mã hóa nhãn

Vì nhiều thuật toán học máy như SVM (Support Vector Machine), Logistic Regression, hay Neural Network chỉ có thể xử lý dữ liệu dạng số mà không thể làm việc trực tiếp với chuỗi ký tự, nên ta cần chuyển đổi các giá trị phân loại (categorical) sang dạng số tương ứng. Trong trường hợp này, cột `Subject` chứa các giá trị tên môn học nên cần được mã hoá lại.

Ta sử dụng công cụ `LabelEncoder` của thư viện `sklearn.preprocessing` để thực hiện việc mã hoá này. Bộ mã hoá này sẽ tự động gán cho mỗi nhãn (label) trong cột một giá trị số nguyên duy nhất, đảm bảo rằng mô hình có thể hiểu và xử lý được dữ liệu.

Ví dụ, danh sách `["Maths", "Physics", "Biology"]` sau khi được mã hoá sẽ trở thành `[0, 2, 1]`. Mỗi số nguyên tương ứng với một môn học cụ thể, nhưng thứ tự gán không mang ý nghĩa về độ lớn hay mức độ, nó chỉ là một cách biểu diễn định danh để phục vụ cho quá trình huấn luyện mô hình.

2.4 Chia tập train, test

Dữ liệu được chia thành **tập huấn luyện** (train set) dùng để huấn luyện mô hình và **tập kiểm thử** (test set) dùng để đánh giá độ chính xác mô hình trên dữ liệu chưa từng thấy. Trong đó, 20% dữ liệu dành cho kiểm thử, 80% cho huấn luyện.

Đồng thời, hương pháp **stratified split** được áp dụng, để chia dữ liệu theo tỷ lệ nhãn (class proportion), giúp tránh mất cân bằng giữa train và test (đặc biệt quan trọng với dữ liệu phân loại nhiều lớp).

Do có các phiên bản dữ liệu khác nhau về đặc trưng đầu vào, nên cần dùng cùng `random_state` để đảm bảo cùng mẫu chia lớp, điều này giúp việc so sánh kết quả giữa hai pipeline công bằng hơn.

Question	Subject
mathbf 2 boldsymbol x boldsymbol + mathbf 3 boldsymbol z mathbf 1 = mathbf 0 mathbf 2 boldsymbol x boldsymbol + mathbf 3 boldsymbol z + mathbf 3 = mathbf 0	Maths
among following one gives output 1 gate a=0 b=0 B a=1 b=1 c a=1 b=0 a=0 b=1	Physics
complex number frac 2^ n 1+i ^ 2 n + frac 1+i ^ 2 n 2^ n quad n boldsymbol z quad boldsymbol quad boldsymbol e q u l quad boldsymbol	Maths
area rhombus 90 mathrm cm ^ 2 one diagonal 14 mathrm cm length diagonal	Maths
marks obtained four students 25 35 45 55 average deviations mean 10 b 9 c 7 none	Maths

Bảng 9: Ví dụ các mẫu trong tập train dành cho pipeline Machine Learning

Question	Subject
mathbf 2 boldsymbol x boldsymbol y + mathbf 3 boldsymbol z mathbf 1 = mathbf 0 and mathbf 2 boldsymbol x boldsymbol y + mathbf 3 boldsymbol z + mathbf 3 = mathbf 0	Maths
Among the following which one gives output 1 in the AND gate A A=0 B=0 B A=1 B=1 c A=1 B=0 D A=0 B=1	Physics
The complex number frac 2^ n 1+i ^ 2 n + frac 1+i ^ 2 n 2^ n quad n in boldsymbol Z quad boldsymbol i s quad boldsymbol e q u a l quad boldsymbol t o	Maths
The area of a rhombus is 90 mathrm cm ^ 2 If one of the diagonal is 14 mathrm cm what is the length of the other diagonal	Maths
Marks obtained by four students are 25 35 45 55 The average deviations from the mean is A 10 B 9 c 7 D none of these	Maths

Bảng 10: Ví dụ các mẫu trong tập train dành cho pipeline Deep Learning

3 Trích xuất đặc trưng

Sau khi hoàn tất bước tiền xử lý dữ liệu, bước tiếp theo là **trích xuất đặc trưng** từ văn bản để đưa vào mô hình học máy. Mục tiêu của bước này là biến đổi dữ liệu văn bản dạng chuỗi thành các **vector** số, đồng thời giữ được các thông tin ngữ nghĩa và ngữ cảnh quan trọng, nhằm nâng cao hiệu quả phân loại.

3.1 Tổng quan phương pháp

Trong nghiên cứu này, chúng tôi triển khai cả hai hướng:

- **Phương pháp truyền thống**: Dựa trên thống kê tần suất từ và các kết hợp n-gram, bao gồm Bag-of-Words (BoW), TF-IDF và n-gram.
- **Phương pháp hiện đại / học sâu**: Sử dụng các *embedding* liên tục (dense vector) như GloVe và contextual embeddings từ DistilBERT.

Các phương pháp này được lựa chọn để so sánh khả năng biểu diễn văn bản đơn giản nhưng hiệu quả (truyền thống) với biểu diễn giàu ngữ cảnh và ngữ nghĩa (hiện đại).

3.2 Phương pháp truyền thống

Các phương pháp truyền thống dựa trên việc đếm từ và tần suất xuất hiện của n-gram trong văn bản. Đặc trưng sinh ra thường là **sparse matrix**, tiết kiệm bộ nhớ, và có thể sử dụng cho các mô hình học máy cơ bản như Naive Bayes, Logistic Regression hay SVM.

Bag-of-Words (BoW) Phương pháp BoW biểu diễn mỗi câu bằng vector đếm số lần xuất hiện của từng từ trong từ điển. Các tham số quan trọng bao gồm:

- `max_features`: giới hạn số lượng từ tối đa được sử dụng (ví dụ 10.000 từ phổ biến nhất).
- `ngram_range`: xác định loại n-gram được xét, ví dụ unigram, bigram hoặc kết hợp.

BoW là biểu diễn cơ bản nhưng mạnh mẽ, đặc biệt hữu ích khi dữ liệu văn bản ngắn hoặc mô hình cần nắm bắt sự xuất hiện riêng lẻ của từ.

TF-IDF TF-IDF (Term Frequency – Inverse Document Frequency) là phương pháp điều chỉnh trọng số của từ dựa trên tần suất xuất hiện trong tập dữ liệu. Những từ phổ biến được giảm trọng số, trong khi các từ mang thông tin phân loại quan trọng được nâng cao. TF-IDF giúp mô hình tập trung vào các từ có khả năng phân biệt các lớp, thay vì các từ dừng hoặc từ phổ biến.

N-gram Ngoài unigram, n-gram (2-gram, 3-gram) được sử dụng để nắm bắt một phần ngữ cảnh ngắn hạn. Ví dụ, cụm từ "average deviation" sẽ được coi như một đặc trưng riêng biệt, giúp tăng khả năng phân biệt các lớp có ngữ nghĩa tương tự nhưng cấu trúc câu khác nhau.

3.3 Phương pháp hiện đại / học sâu

Các phương pháp hiện đại sử dụng **embedding** liên tục (dense vector) để biểu diễn văn bản, giữ thông tin ngữ cảnh và ngữ nghĩa phong phú hơn so với các phương pháp truyền thống.

GloVe Embeddings GloVe (Global Vectors for Word Representation) học vector từ dựa trên ma trận *co-occurrence* giữa các từ. Mỗi từ được ánh xạ vào một vector 100 chiều, và để biểu diễn một câu, các vector từ trong câu được trung bình lại (*mean pooling*). Cách biểu diễn này giúp mô hình học sâu khai thác các mối quan hệ ngữ nghĩa giữa các từ.

DistilBERT Embeddings DistilBERT là một mô hình Transformer nhẹ, cung cấp **contextual embeddings** cho từng câu. Khác với GloVe, vector của mỗi từ phụ thuộc vào ngữ cảnh xung quanh trong câu. Vector câu được tính bằng trung bình các vector token (*mean pooling*) trên toàn bộ câu. Sử dụng DistilBERT giúp mô hình học sâu nhận diện các mối quan hệ ngữ pháp, logic và ngữ nghĩa phức tạp, từ đó nâng cao hiệu quả phân loại.

3.4 Tóm tắt

- **Traditional Methods**: BoW, TF-IDF, n-gram → sparse matrix, nhanh, nhẹ, hiệu quả với dữ liệu ngắn.
- **Modern Embeddings**: GloVe, DistilBERT → dense vector, giữ ngữ cảnh, ngữ nghĩa, phù hợp với mô hình học sâu.
- Việc kết hợp hoặc so sánh các phương pháp này giúp đánh giá được hiệu quả biểu diễn văn bản và lựa chọn đặc trưng tối ưu cho nhiệm vụ phân loại.

4 Huấn luyện mô hình

Trong bước này, dữ liệu sau tiền xử lý và trích xuất đặc trưng được đưa vào các pipeline khác nhau để huấn luyện và đánh giá mô hình. Pipeline được chia thành hai nhánh chính: **Machine Learning dựa trên feature/embedding** và **Deep Learning end-to-end**. Mục tiêu là trình bày cách triển khai, các mô hình, tham số thử nghiệm và quy trình train/evaluate.

4.1 Pipeline Machine Learning dựa trên feature/embedding

4.1.1 Các loại feature và embedding

- **Sparse features:** Bao gồm Bag-of-Words (BoW), TF-IDF và n-gram. Những feature này thể hiện tần suất xuất hiện của từ hoặc chuỗi từ trong văn bản.
- **Dense embeddings:** Bao gồm GloVe và DistilBERT. Các embeddings này mã hóa ngữ nghĩa và ngữ cảnh của từ, phù hợp cho các mô hình ML phi tuyến như Random Forest, Gradient Boosting hoặc GaussianNB.

4.1.2 Các mô hình thử nghiệm

- **Linear models:** Logistic Regression và Linear SVM, thường áp dụng với feature sparse, phù hợp cho các bài toán phân loại nhiều lớp.
- **Tree-based models:** Decision Tree, Random Forest, Gradient Boosting, thích hợp với feature dense và dữ liệu phi tuyến.
- **Naive Bayes:** MultinomialNB (cho sparse features) và GaussianNB (cho dense embeddings).

4.1.3 Quy trình triển khai

Pipeline ML được triển khai theo các bước sau:

1. Load feature tương ứng cho tập train và test.
2. Train từng mô hình với feature đã chọn.
3. Predict trên tập test.
4. Đánh giá hiệu năng sử dụng các metrics: Accuracy, Precision, Recall, F1-score.
5. Quan sát phân bố dự đoán qua Confusion Matrix.
6. Tổng hợp kết quả để so sánh hiệu quả của từng feature và từng mô hình.

4.1.4 Các tham số thử nghiệm

- **Logistic Regression:** regularization coefficient C, solver, max_iter, class_weight.
- **Decision Tree:** max_depth, class_weight.
- **Random Forest:** n_estimators, max_depth, class_weight.
- **Gradient Boosting:** n_estimators, learning_rate, max_depth.
- Lưu ý: GaussianNB không có tham số tối ưu hóa, MultinomialNB phù hợp với feature sparse.

4.2 Pipeline Deep Learning end-to-end

4.2.1 Chuẩn bị dữ liệu

- Dữ liệu DL-ready giữ nguyên chữ hoa/thường và stopwords để bảo toàn ngữ cảnh.
- Tokenizer theo từ, xây dựng vocab từ tập train, bao gồm các token đặc biệt `<pad>` và `<unk>`.
- Dataset và DataLoader được chuẩn hóa với **padding sequence** để xử lý batch có độ dài khác nhau.

4.2.2 Mô hình LSTM

- Sử dụng **Bidirectional LSTM** để học ngữ cảnh theo cả hai hướng.
- Embedding layer được train từ đầu trên dữ liệu của bài toán.
- Linear layer cuối cùng ánh xạ ra số lớp *num_classes*.
- Các tham số chính đã thử nghiệm:
  - embedding_dim = 128
  - hidden_dim = 128
  - bidirectional = True
  - batch_size = 64
  - learning_rate = 1e-3
  - epochs = 8
- Hỗ trợ **load pretrained model** nếu đã train trước đó, giúp tiết kiệm thời gian huấn luyện.

4.2.3 Quy trình train/evaluate LSTM

1. Chuẩn bị batch train và test với padding.
2. Huấn luyện theo epoch, tối ưu bằng CrossEntropyLoss và Adam optimizer.
3. Theo dõi loss trong quá trình train để giám sát hội tụ.
4. Sau khi train xong, evaluate trên tập test với các metrics: Accuracy, Precision, Recall, F1-score.
5. Lưu model, optimizer, config và logs để có thể tái sử dụng hoặc fine-tune tiếp.

4.3 Tổng quan pipeline triển khai

- **ML + feature/embedding:** từ sparse (BoW/TF-IDF/n-gram) và dense (GloVe/DistilBERT) → train/evaluate nhiều mô hình ML khác nhau → tổng hợp kết quả.
- **End-to-End Deep Learning:** từ văn bản thô → tokenize → DataLoader → LSTM → train → evaluate → lưu model/config.
- Pipeline được thiết kế **dễ mở rộng**, cho phép thử nghiệm nhiều feature, embedding và mô hình khác nhau.
- Mỗi nhánh đều có quy trình train/evaluate chuẩn hóa, phục vụ so sánh hiệu quả giữa các hướng tiếp cận truyền thống và hiện đại.

5 So sánh đánh giá các mô hình

Trong phần này, ta tổng hợp và phân tích kết quả đánh giá các mô hình phân loại văn bản dựa trên nhiều loại feature/embedding khác nhau, bao gồm các mô hình Machine Learning (ML) truyền thống và mô hình Deep Learning (LSTM).

5.1 Bảng kết quả chi tiết

Feature	Model	Accuracy	Precision	Recall	F1-score
BoW	Logistic Regression	0.9197	0.9205	0.9197	0.9197
BoW	Decision Tree	0.4175	0.5772	0.4175	0.4332
BoW	Random Forest	0.7093	0.8138	0.7093	0.7339
BoW	MultinomialNB	0.9016	0.9027	0.9016	0.9017
TF-IDF	Logistic Regression	0.8920	0.8946	0.8920	0.8924
TF-IDF	Decision Tree	0.4178	0.5791	0.4178	0.4325
TF-IDF	Random Forest	0.7151	0.8153	0.7151	0.7383
TF-IDF	MultinomialNB	0.9099	0.9099	0.9099	0.9098
N-gram	Logistic Regression	0.9190	0.9204	0.9190	0.9192
N-gram	Decision Tree	0.4173	0.5766	0.4173	0.4332
N-gram	Random Forest	0.6648	0.8027	0.6648	0.6954
N-gram	MultinomialNB	0.8829	0.8901	0.8829	0.8844
GloVe	Logistic Regression	0.8455	0.8455	0.8455	0.8451
GloVe	Decision Tree	0.6262	0.6338	0.6262	0.6235
GloVe	Random Forest	0.8090	0.8111	0.8090	0.8086
DistilBERT	Logistic Regression	0.9080	0.9079	0.9080	0.9078
DistilBERT	Decision Tree	0.7391	0.7465	0.7391	0.7389
DistilBERT	Random Forest	0.8766	0.8774	0.8766	0.8762

Bảng 11: So sánh các mô hình Machine Learning với nhiều loại feature/embedding

Model	Accuracy	Precision	Recall	F1-score
LSTM	0.9235	0.9245	0.9235	0.9238

Bảng 12: Kết quả mô hình Deep Learning LSTM

5.2 Nhận xét và phân tích

Từ bảng kết quả, rút ra một số nhận xét chính:

- **Hiệu quả của các mô hình ML với sparse features:**
  - Logistic Regression và MultinomialNB hoạt động rất tốt với BoW, TF-IDF và n-gram, đạt Accuracy trên 0.88–0.92.

  - Logistic Regression đạt Accuracy cao nhất (0.92) với BoW/n-gram vì mô hình tuyến tính phù hợp với feature sparse: các từ xuất hiện độc lập và dữ liệu nhiều chiều, giúp mô hình học ranh giới phân lớp tối ưu.
  - Decision Tree đơn lẻ cho kết quả thấp (0.42), chứng tỏ cây quyết định đơn lẻ dễ overfit dữ liệu sparse, nhiều chiều và không khai thác tốt sự phân bố của từ.

- **Tree-based models với dense embeddings:**
  - Random Forest và Decision Tree cải thiện khi dùng GloVe hoặc DistilBERT.
  - Random Forest với DistilBERT đạt Accuracy 0.8766, F1-score 0.8762, cho thấy mô hình phi tuyến và ensemble khai thác tốt thông tin ngữ cảnh từ embeddings.
  - Embeddings contextual phát huy hiệu quả rõ rệt khi kết hợp với mô hình phi tuyến, vì chúng nắm bắt được ngữ cảnh và mối quan hệ phức tạp giữa các từ.

- **So sánh embeddings hiện đại và phương pháp truyền thống:**
  - Logistic Regression với BoW hoặc n-gram đạt Accuracy cao (0.92), gần tương đương với DistilBERT + Logistic Regression (0.908). Feature sparse vẫn rất mạnh với mô hình tuyến tính.
  - DistilBERT kết hợp Random Forest đạt F1-score 0.8762, nhấn mạnh lợi thế của embeddings contextual khi dùng mô hình phi tuyến.

- **Deep Learning LSTM:**
  - Mô hình LSTM end-to-end đạt Accuracy 0.9235, F1-score 0.9238, cao hơn hoặc tương đương với các mô hình ML tốt nhất.
  - LSTM học trực tiếp từ chuỗi văn bản, vừa học embedding vừa học phân loại, nắm bắt ngữ cảnh và mối quan hệ từ trong câu.

- **Nhận xét tổng quát:**
  - Mô hình truyền thống (BoW, TF-IDF, n-gram) + Linear model vẫn là baseline mạnh cho phân loại văn bản ngắn, dữ liệu sparse.
  - Embeddings hiện đại (GloVe, DistilBERT) cải thiện hiệu quả khi dùng với mô hình phi tuyến (Random Forest).
  - LSTM end-to-end là giải pháp tổng thể tốt nhất, tối ưu cho dữ liệu có ngữ cảnh phức tạp.

5.3 Kết luận rút ra

- Logistic Regression với BoW/n-gram đạt Accuracy cao nhất trong các mô hình ML truyền thống, nhờ tính tuyến tính phù hợp với feature sparse và dữ liệu ngắn, độc lập.

- Embeddings hiện đại phát huy hiệu quả rõ khi kết hợp với mô hình phi tuyến (Random Forest), nhờ khả năng nắm bắt ngữ cảnh và mối quan hệ phức tạp giữa các từ.
- LSTM end-to-end đạt hiệu quả cao nhất và ổn định, học trực tiếp từ chuỗi văn bản, vừa học embedding vừa học phân lớp, tối ưu cho dữ liệu ngữ cảnh phức tạp.
- Decision Tree đơn lẻ không phù hợp với dữ liệu sparse, cần ensemble hoặc embeddings để cải thiện.
- Lựa chọn mô hình tùy theo mục tiêu:
  - Nhanh, dễ triển khai: ML với BoW/TF-IDF/n-gram.
  - Tối đa hóa độ chính xác và khai thác ngữ cảnh: LSTM hoặc embeddings + Random Forest.

6 Kết luận

Trong báo cáo này, ta đã triển khai và đánh giá nhiều hướng tiếp cận khác nhau cho bài toán phân loại câu hỏi theo môn học, bao gồm:

- Các mô hình Machine Learning truyền thống với feature sparse: BoW, TF-IDF, n-gram.
- Các mô hình Machine Learning với dense embeddings: GloVe, DistilBERT.
- Mô hình Deep Learning end-to-end: LSTM.

Qua quá trình huấn luyện và đánh giá, rút ra một số kết luận chính:

1. **Hiệu quả của mô hình ML truyền thống:** Logistic Regression và MultinomialNB với BoW/n-gram đạt Accuracy và F1-score rất cao (trên 0.88), chứng tỏ các feature truyền thống vẫn là baseline mạnh, đặc biệt với dữ liệu ngắn, sparse.
2. **Tác dụng của embeddings hiện đại:** Khi sử dụng GloVe hoặc DistilBERT kết hợp với các mô hình phi tuyến (Random Forest), mô hình có khả năng khai thác ngữ nghĩa và ngữ cảnh, nâng cao độ chính xác, đặc biệt khi dữ liệu có cấu trúc phức tạp hơn.
3. **Ưu thế của LSTM end-to-end:** Mô hình LSTM đạt Accuracy 0.9235 và F1-score 0.9238, cao hơn hoặc tương đương với các mô hình ML tốt nhất. LSTM học trực tiếp từ chuỗi văn bản, nắm bắt các mối quan hệ ngữ nghĩa và ngữ cảnh phức tạp, do đó phù hợp cho các bài toán phân loại văn bản tổng thể.
4. **Những hạn chế quan sát được:** Decision Tree đơn lẻ kém hiệu quả với dữ liệu sparse. Các mô hình ML truyền thống cần careful tuning và feature engineering để đạt hiệu quả tối ưu.
5. **Hướng ứng dụng:**
    - Nếu ưu tiên tính nhanh, dễ triển khai: ML với BoW/TF-IDF/n-gram là lựa chọn hợp lý.
    - Nếu ưu tiên độ chính xác và khả năng học ngữ cảnh: embeddings hiện đại kết hợp Random Forest hoặc mô hình LSTM end-to-end là giải pháp tối ưu.

Tóm lại, kết quả đánh giá cho thấy rằng, kết hợp embeddings hiện đại với mô hình phi tuyến hoặc sử dụng học sâu end-to-end là chiến lược hiệu quả nhất để phân loại văn bản, trong khi các mô hình truyền thống vẫn là baseline đáng tin cậy và dễ triển khai trong nhiều tình huống thực tế.

7 Phụ lục

Dataset: IITJEE NEET AIIMS Students Questions Data
Github: DNA05's github page
Colab Notebook: DNA05-BTL2

Tài liệu tham khảo

[1] Hossain Hedayati. Heart disease prediction with 83.8% accuracy. https://www.kaggle.com/code/hossainhedayati/heart-disease-prediction-with-83-8-accuracy/notebook, 2023. Accessed: 2025-09-18.

[2] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R and Python*. Springer, 2nd edition, 2021.

[3] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[4] Scikit-learn developers. Classification metrics — scikit-learn documentation. https://scikit-learn.org/stable/modules/model_evaluation.html, 2025. Accessed: 2025-09-18.

[5] Scikit-learn developers. Preprocessing data — scikit-learn documentation. https://scikit-learn.org/stable/modules/preprocessing.html, 2025. Accessed: 2025-09-18.