TRƯỜNG ĐẠI HỌC KHOA HỌC HUẾ

KHOA CÔNG NGHỆ THÔNG TIN

- - - * - - -**

BÁO CÁO THỰC TẬP NIÊN LUẬN

ĐỀ TÀI: NHẬN DIỆN TIN TUYỂN DỤNG LỪA ĐẢO BẰNG CÁC MÔ HÌNH HỌC MÁY

Thành viên:

Lê Ngọc Ánh - 22T1020019

Nguyễn Bá Nhật - 22T1020289

Giáo viên hướng dẫn:

ThS.Trần Thị Phương Chi

HUẾ, 2025

MỤC LỤC

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 ThS.Cô Trần Thị Phương Chi đã tận tâm hướng dẫn, giúp đỡ trong quá trình định hướng, nghiên cứu và hoàn thành niên 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 khóa niên 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

DANH MỤC HÌNH VẼ

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Nghĩa tiếng Anh	Nghĩa tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
ML	Machine Learning	Học máy
DL	Deep Learning	Học sâu
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
TF	Term Frequency	Tần suất thuật ngữ
IDF	Inverse Document Frequency	thước đo độ hiếm của một từ
TF–IDF	Term Frequency–Inverse Document Frequency	Mã hóa TF–IDF
BoW	Bag-of-Words	Mô hình "túi từ"
N-gram	N-gram	Cụm n từ liên tiếp
LR	Logistic Regression	Hồi quy Logistic
SVM	Support Vector Machine	Máy véc-tơ hỗ trợ
NB	Naive Bayes	Naive Bayes
RF	Random Forest	Rừng ngẫu nhiên
OHE	One-Hot Encoding	Mã hóa one-hot
URL	Uniform Resource Locator	Địa chỉ liên kết (URL)
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
SMOTE	Synthetic Minority Over-sampling Technique	Tăng mẫu tổng hợp cho lớp thiểu số
CLASS WEIGHT	Class weight	Trọng số lớp

MỞ ĐẦU

Trong những năm gần đây, xu hướng tuyển dụng trực tuyến (tuyển dụng trực tuyến) ngày càng bùng nổ. Các nền tảng như Indeed, LinkedIn, Glassdoor, VietnamWorks… đã trở thành kênh chính để quản lý doanh nghiệp đăng tin và tìm kiếm cơ hội. Tuy nhiên, song với sự phát triển đó là sự gia tăng của các tin tuyển dụng lừa đảo (đăng tuyển dụng giả). Những kẻ giả mạo này thường giả mạo công ty uy tín, đưa ra mức lương "hấp dẫn" và yêu cầu ứng dụng cung cấp thông tin cá nhân, yêu cầu đóng phí đặt cọc hoặc bất kỳ biểu thức thanh toán nào thì mới được tiếp tục. Kết quả cuối cùng, ứng dụng có thể bị đánh cắp thông tin cá nhân, mất tiền hoặc mất nhiều thời gian vô ích.

Việc tự động phát hiện và loại bỏ những kẻ lừa đảo ngay tại nguồn không chỉ giúp bảo vệ quyền lợi của người lao động mà còn nâng cao uy tín của nền tảng tuyển dụng và giảm thiểu rủi ro cho doanh nghiệp. Trong khi nhiều nghiên cứu về email spam, tin tức giả (tin tức giả) đã đạt được những thành phần thiết bị nhất, việc áp dụng các kỹ thuật học máy (machine learning) để phân tích "tin thật" và "tin tuyển dụng lừa đảo" trong lĩnh vực tuyển dụng vẫn là một kỹ thức làm nội dung mô tả công việc thường đa dạng, có nhiều thông tin phi cấu trúc (văn bản phi cấu trúc) đi kèm với các thông số "meta" (ví dụ: ngành, Kinh nghiệm, vị trí, v.v).

CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN NHẬN DIỆN TIN TUYỂN DỤNG LỪA ĐẢO

1.1. Lý do chọn đề tài:

Trong bối cảnh chuyển đổi số, hình thức tuyển dụng trực tuyến ngày càng phổ biến. Hầu hết người tìm việc dựa vào các nền tảng đăng tin tuyển dụng (job boards), mạng xã hội nghề nghiệp, fanpage Facebook, nhóm Zalo, diễn đàn... Tuy nhiên, song hành với lợi ích lan tỏa nhanh, chi phí thấp là sự bùng nổ của các tin tuyển dụng giả (fake job postings) nhằm mục đích lừa đảo tiền bạc (phí xét duyệt hồ sơ, phí đồng phục, đặt cọc), đánh cắp dữ liệu cá nhân, phát tán spam hoặc dẫn dụ vào các hành vi bất hợp pháp. Nhiều nạn nhân là sinh viên, người mới ra trường, lao động phổ thông thiếu kinh nghiệm xác minh nguồn tin.

Theo thống kê từ nhiều chuyên trang tuyển dụng quốc tế (ví dụ các báo cáo cảnh báo lừa đảo việc làm hằng năm), tỷ lệ tin giả có thể dao động từ vài phần trăm đến trên 10% tùy nền tảng và giai đoạn cao điểm. Ở Việt Nam, các trường hợp bị lừa đặt cọc đi làm thời vụ, tuyển CTV online nhập liệu, đóng phí nhận việc ship hàng... xuất hiện thường xuyên trên mạng xã hội. Mặc dù đã có cảnh báo thủ công từ quản trị viên hoặc các bài viết kinh nghiệm phân biệt thật/giả, việc tự động hóa nhận diện tin tuyển dụng rủi ro vẫn còn hạn chế.

Trong khi đó, học máy (Machine Learning) và xử lý ngôn ngữ tự nhiên (NLP) đã chứng minh hiệu quả mạnh mẽ trong phân loại văn bản spam email, phát hiện bình luận độc hại, tin giả (fake news). Bài toán phân loại tin tuyển dụng thật/giả về bản chất cũng là một bài toán phân loại văn bản nhị phân – phù hợp để áp dụng ML.

Với mong muốn góp phần giải quyết vấn đề này, nhóm thực hiện chọn đề tài *"Nhận diện tin tuyển dụng giả bằng các mô hình học máy"* nhằm xây dựng một mô hình có khả năng phân loại các tin tuyển dụng thành thật hoặc giả một cách tự động và hiệu quả.

1.2. Mục tiêu nghiên cứu:

❖ **Mục tiêu tổng quát:** Nghiên cứu và xây dựng một hệ thống ứng dụng các mô hình học máy để *nhận diện tự động các tin tuyển dụng giả* dựa trên nội dung văn bản của tin tuyển dụng, từ đó góp phần giảm thiểu rủi ro cho người tìm việc và nâng cao hiệu quả quản lý nội dung của các nền tảng tuyển dụng trực tuyến.

❖ **Mục tiêu cụ thể:**

➢ Tìm hiểu tổng quan về tin tuyển dụng giả, đặc điểm nhận dạng và ảnh hưởng của chúng đến người lao động và xã hội.

➢ Thu thập và phân tích dữ liệu tin tuyển dụng thật và giả từ các nguồn dữ liệu có sẵn.

➢ Thực hiện tiền xử lý dữ liệu văn bản, bao gồm:

- Làm sạch văn bản (loại bỏ ký tự đặc biệt, HTML tag,…),
- Chuẩn hóa văn bản (chuyển về chữ thường, loại bỏ stopwords,…),
- Biến đổi văn bản sang dạng số (sử dụng TF–IDF, BoW,…).

➢ Xây dựng và huấn luyện các mô hình học máy có giám sát như: Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine (SVM).

1.3. Đối tượng và phạm vi nghiên cứu:

❖ **Đối tượng nghiên cứu:** Các tin tuyển dụng được đăng tải trên các nền tảng trực tuyến như VietnamWorks, TopCV, MyWork, hoặc các trang tổng hợp việc làm. Đối

tượng cụ thể là nội dung văn bản trong tin tuyển dụng, bao gồm: tiêu đề, mô tả công việc, yêu cầu ứng viên, quyền lợi, loại hình việc làm, công ty tuyển dụng,…

❖ **Phạm vi nghiên cứu:**

➢ **Về nội dung và loại dữ liệu:**

- Chỉ tập trung vào dữ liệu dạng văn bản (textual data) trong tin tuyển dụng. Các thành phần phi văn bản như hình ảnh, video, liên kết ngoài… không được xem xét trong phạm vi nghiên cứu này.
- Nghiên cứu dữ liệu đã được gán nhãn (labeled dataset), tức là các tin tuyển dụng đã được xác định rõ là "thật" hoặc "giả", nhằm phục vụ bài toán học máy có giám sát (supervised learning).
- Dữ liệu sử dụng chủ yếu là tiếng Anh, được trích xuất từ các tập dữ liệu công khai như "Fake Job Postings Dataset" hoặc các nguồn học thuật tương đương. Nếu thời gian và nguồn lực cho phép, có thể mở rộng sang dữ liệu tiếng Việt.

➢ **Về kỹ thuật và mô hình:**

- Đề tài áp dụng các kỹ thuật học máy truyền thống, bao gồm: Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine (SVM).
- Không triển khai các mô hình học sâu (Deep Learning) như RNN, LSTM, BERT do hạn chế về thời gian và tài nguyên tính toán.
- Dữ liệu được tiền xử lý bằng các kỹ thuật cơ bản trong xử lý ngôn ngữ tự nhiên (NLP) như: chuyển văn bản về chữ thường, loại bỏ stopwords, chuẩn hóa từ, biểu diễn văn bản bằng TF-IDF hoặc Bag-of-Words.

➢ **Về phạm vi đánh giá:**

- Hiệu quả của các mô hình được đánh giá dựa trên tập dữ liệu kiểm thử (test set) thông qua các chỉ số:
    - Accuracy (độ chính xác tổng thể).
    - Precision (độ chính xác cho từng lớp).
    - Recall (khả năng phát hiện tin giả).
    - F1-score (chỉ số cân bằng giữa Precision và Recall).
- Không tập trung phát triển giao diện người dùng hay ứng dụng thực tiễn cụ thể, mà chỉ tập trung vào xây dựng mô hình và đánh giá hiệu năng.

1.4. Phương pháp nghiên cứu:

❖ **Phân tích tài liệu:** Tìm hiểu các nghiên cứu liên quan đến nhận diện tin giả, kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và các mô hình học máy thường dùng như Naive Bayes, SVM, Logistic Regression và Random Forest, nhằm lựa chọn phương pháp phù hợp cho đề tài.

❖ **Thu thập dữ liệu:** Sử dụng tập dữ liệu có sẵn gồm các tin tuyển dụng thật và giả, trong đó mỗi tin chứa các trường như tiêu đề, mô tả, yêu cầu, quyền lợi và nhãn phân loại (real/fake).

❖ **Tiền xử lý dữ liệu:** Làm sạch văn bản: xóa ký tự đặc biệt, chuyển về chữ thường, loại bỏ stopwords, tách từ. Văn bản sau xử lý sẽ được dùng cho bước trích xuất đặc trưng.

❖ **Trích xuất đặc trưng:** Sử dụng phương pháp TF–IDF để chuyển nội dung văn bản thành vector số, giúp mô hình học máy hiểu và học được các đặc điểm từ dữ liệu văn bản.

❖ **Huấn luyện mô hình:** Áp dụng các mô hình học máy có giám sát như: Naive Bayes, Logistic Regression, Random Forest, SVM. Dữ liệu được chia thành tập huấn luyện và tập kiểm thử để đánh giá mô hình khách quan.

❖ **Đánh giá mô hình:** Sử dụng các chỉ số:

- Accuracy: độ chính xác tổng thể.
- Precision: độ chính xác với lớp "giả".
- Recall: khả năng phát hiện tin giả.
- F1-score: độ cân bằng giữa Precision và Recall.

❖ **Phân tích kết quả:** So sánh các mô hình để tìm mô hình tốt nhất. Phân tích lỗi và đưa ra đề xuất cải tiến như mở rộng dữ liệu, thêm đặc trưng hoặc thử nghiệm các phương pháp nâng cao trong tương lai.

1.5. Giới hạn và phạm vi của đề tài:

❖ **Giới hạn của đề tài:**

- Không xử lý dữ liệu đa phương tiện như hình ảnh, video, hoặc đường dẫn website.
- Không sử dụng các mô hình học sâu (deep learning) do giới hạn về thời gian và tài nguyên.
- Chưa triển khai hệ thống thực tế, chỉ đánh giá mô hình trên môi trường thử nghiệm.

❖ **Phạm vi của đề tài:**

- Đề tài tập trung vào phân loại tin tuyển dụng giả và thật dựa trên nội dung văn bản như tiêu đề, mô tả, yêu cầu, quyền lợi,...

- Áp dụng các mô hình học máy truyền thống như Naive Bayes, Logistic Regression, SVM và Random Forest.
- Dữ liệu sử dụng là tập dữ liệu có sẵn bằng tiếng Anh, đã được gán nhãn.

1.6. Nội dung của đề tài:

❖ Ngoài phần mở đầu và kết luận, nội dung đề tài gồm 3 chương:

➢ **Chương 1:** Tổng quan về đề tài "nhận diện tin tuyển dụng giả": Tổng quan về nội dung đề tài, đưa ra những lý do, phương pháp, ý nghĩa thực tiễn của đề tài và kết quả đạt được.

➢ **Chương 2:** Cơ sở lý thuyết về đề tài "nhận diện tin tuyển dụng giả":

- Giới thiệu các phương pháp học máy và kỹ thuật xử lý ngôn ngữ tự nhiên (NLP).
- Trình bày quy trình tiền xử lý dữ liệu văn bản.

➢ **Chương 3:** Triển khai và đánh giá các mô hình học máy sử dụng trong đề tài:

- Trình bày quá trình xây dựng và huấn luyện mô hình học máy.
- Đánh giá, so sánh hiệu quả giữa các mô hình.
- Đề xuất mô hình tối ưu và hướng phát triển tiếp theo.

➢ **Chương 4:** Kết luận và hướng phát triển:

- Đánh giá tổng quan.
- Hướng phát triển.

1.7. Ý nghĩa khoa học và thực tiễn của đề tài:

❖ **Ý nghĩa khoa học:** Đề tài là một ví dụ điển hình cho việc ứng dụng khoa học dữ liệu, học máy và xử lý ngôn ngữ tự nhiên (NLP) vào giải quyết một vấn đề thực tế mang

tính xã hội. Việc sử dụng các mô hình học máy như: Naive Bayes, Logistic Regression, SVM và Random Forest... trong việc phân loại tin tuyển dụng thật và giả giúp chứng minh tính khả thi và hiệu quả của các thuật toán trong bài toán xử lý văn bản có nhãn. Bên cạnh đó, nó còn:

➢ Góp phần làm rõ quy trình áp dụng học máy trong bài toán phân loại nhị phân trên dữ liệu văn bản, từ bước tiền xử lý, trích xuất đặc trưng đến đánh giá mô hình.

➢ Cung cấp một mô hình thực nghiệm có thể tham khảo, phát triển và mở rộng trong các nghiên cứu sau như: phát hiện tin lừa đảo nói chung, nhận diện thông tin sai lệch trên mạng xã hội, hoặc kiểm duyệt nội dung trực tuyến.

❖ **Ý nghĩa thực tiễn:** Trong bối cảnh số lượng tin tuyển dụng đăng tải trên Internet ngày càng nhiều, việc phát hiện và loại bỏ các tin tuyển dụng giả mạo là một nhu cầu bức thiết. Đề tài mang lại những giá trị thực tiễn rõ rệt:

➢ Đối với người tìm việc: Mô hình đề xuất có thể hỗ trợ cảnh báo người dùng về các tin tuyển dụng có dấu hiệu lừa đảo, giúp họ tránh được rủi ro mất tiền, bị lừa vào công việc bất hợp pháp hoặc lộ thông tin cá nhân.

➢Đối với các nền tảng tuyển dụng: Việc tích hợp mô hình học máy vào hệ thống kiểm duyệt giúp tự động hóa quy trình đánh giá tin đăng, giảm tải cho đội ngũ quản trị, đồng thời tăng độ tin cậy và minh bạch cho nền tảng.

➢ Đối với xã hội: Góp phần xây dựng một môi trường tìm việc lành mạnh và an toàn, từ đó nâng cao chất lượng tuyển dụng, hạn chế tình trạng trục lợi và vi phạm pháp luật trên không gian số.

1.8. Dự kiến kết quả đạt được của đề tài:

- Xây dựng thành công mô hình học máy có khả năng phân loại tin tuyển dụng thành hai nhóm: thật và giả, với độ chính xác tổng thể dự kiến đạt trên 95%. Mô

hình hoạt động dựa trên nội dung văn bản, không phụ thuộc vào yếu tố hình ảnh hay đường dẫn ngoài.

- Huấn luyện và đánh giá nhiều mô hình khác nhau, bao gồm: Naive Bayes, Logistic Regression, Random Forest và SVM. Việc so sánh được thực hiện dựa trên các chỉ số đánh giá như Accuracy, Precision, Recall và F1-score để lựa chọn mô hình tối ưu.
- Xây dựng quy trình tiền xử lý dữ liệu văn bản hiệu quả, bao gồm: làm sạch dữ liệu, tách từ, loại bỏ từ dừng, biểu diễn văn bản bằng TF-IDF,… Quy trình này có thể được tái sử dụng trong các bài toán phân loại văn bản khác như phân tích cảm xúc, phát hiện spam, phân loại đánh giá,…
- Đề xuất mô hình tối ưu cho bài toán nhận diện tin tuyển dụng giả, đồng thời đưa ra các hướng phát triển tiếp theo như: mở rộng dữ liệu tiếng Việt, thử nghiệm các mô hình deep learning (BERT, LSTM), và tích hợp vào hệ thống cảnh báo hoặc công cụ kiểm duyệt tin đăng tuyển dụng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VỀ BÀI TOÁN NHẬN DIỆN TIN TUYỂN DỤNG LỪA ĐẢO

2.1. Giới thiệu:

Xử lý ngôn ngữ tự nhiên (NLP) là lĩnh vực thuộc trí tuệ nhân tạo, nhằm giúp máy tính hiểu và xử lý ngôn ngữ con người. NLP kết hợp ngôn ngữ học, thống kê và khoa học máy tính để thực hiện các tác vụ như phân loại văn bản, trích xuất thông tin hay phân tích cảm xúc. Trong bối cảnh thông tin lan truyền nhanh chóng, NLP được ứng dụng rộng rãi để nhận diện và phân tích nội dung trên Internet, đặc biệt trong phát hiện tin tuyển dụng lừa đảo. Nhờ khả năng phân tích ngôn ngữ và mô hình hóa văn bản, NLP giúp xây dựng hệ thống phát hiện tin tuyển dụng lừa đảo tự động, với sự hỗ trợ của các thuật toán học máy như Naive Bayes hay SVM.

2.2. Xử lý ngôn ngữ tự nhiên:

2.2.1. Xây dựng bộ từ điển:

Trong bộ dữ liệu tin tuyển dụng, mỗi tin lại có số lượng từ khác nhau. Điều này dẫn đến khi mã hóa văn bản thô thành vector, ta thu được các vector có số chiều (số lượng đặc trưng) khác nhau, không phù hợp làm đầu vào cho các mô hình học máy thông thường vốn yêu cầu đầu vào có kích thước cố định. Để giải quyết vấn đề này, một bộ từ điển chung cần được xây dựng từ toàn bộ các từ xuất hiện trong tập dữ liệu. Cụ thể, bộ từ điển này là tập hợp duy nhất các từ xuất hiện trong văn bản.

Sau khi có từ điển, mỗi tin tuyển dụng sẽ được mã hóa thành một vector có độ dài cố định bằng với kích thước của từ điển bằng các phương pháp mã hóa như mã hóa đếm từ . Kết quả là mỗi tin tuyển dụng được biểu diễn bằng một vector có cùng số chiều, sẵn sàng để đưa vào huấn luyện các mô hình học máy. Ví dụ, bộ dữ liệu *D* của chúng ta có 3 câu như sau:

$D = [d_1$: "$The\ cat\ sat\ on\ the\ mat$",

d_2: "$i\ looked\ at\ the\ sky$",

d_3: "$i\ sat\ on\ my\ chair$"]

Từ điển của chúng ta thu thập được sẽ gồm những từ: the, cat, sat, on, mat, i, looked, at, sky, my, chair. Vậy mỗi văn bản của chúng ta sẽ được mã hóa thành vector 11 chiều như bảng sau:

t (từ)	the	cat	sat	on	mat	i	looked	at	sky	my	chair
d_1	2	1	1	1	1	0	0	0	0	0	0
d_2	1	0	0	0	0	1	1	1	1	0	0
d_3	0	0	1	1	0	1	0	0	0	1	1

Bảng 2.2. Mã hóa đếm từ các văn bản của bộ dữ liệu *D*

2.2.2. Mã hóa TF-IDF:

Tuy phương pháp đếm tần suất xuất hiện của từ (Word Count) là cách tiếp cận đơn giản và trực quan trong việc biểu diễn văn bản dưới dạng vector số, nhưng nó còn tồn tại nhiều hạn chế. Cụ thể, phương pháp này không phản ánh được mức độ quan trọng tương đối của một từ trong từng văn bản cũng như trong toàn bộ tập dữ liệu. Ví dụ, những từ phổ biến như “tuyển”, “việc”, hay “công ty” có thể xuất hiện thường xuyên trong hầu hết các tin tuyển dụng, khiến mô hình khó phân biệt được các đặc điểm riêng biệt của văn bản lừa đảo.

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 nhận diện tin tuyển dụng giả, lừa đảo. 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:

$$\text{TF(t, d)} = \frac{\text{Số từ t xuất hiện trong d}}{\text{Tổng số từ xuất hiện trong d}}$$

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:

$D = [d_1$: "$The\ cat\ sat\ on\ the\ mat$",
d_2: "$i\ looked\ at\ the\ sky$",
d_3: "$i\ sat\ on\ my\ chair$"]

t (từ)	the	cat	sat	on	mat	i	looked	at	sky	my	chair
$TF(t, d_1)$	0,33	0,17	0,17	0,17	0,17	0	0	0	0	0	0
$TF(t, d_2)$	0,2	0	0	0	0	0,2	0,2	0,2	0,2	0	0
$TF(t, d_3)$	0	0	0,2	0,2	0	0,2	0	0	0	0,2	0,2

Bảng 2.3. Giá trị TF của các văn bản trong bộ dữ liệu *D*

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

$$\text{IDF(t)} = \log\left(\frac{\text{Số lượng tài liệu có trong bộ dữ liệu}}{\text{Số lượng tài liệu có chứa t } + 1}\right)$$

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

t (từ)	the	cat	sat	on	mat	i	looked	at	sky	my	chair
IDF(*t*)	0	0,41	0	0	0,41	0	0,41	0,41	0,41	0,41	0,41

Bảng 2.4. Giá trị IDF của các từ trong bộ từ điển

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

t (từ)	the	cat	sat	on	mat	i	looked	at	sky	my	chair
$F\ -\ IDF(t,\ d_1)$	0	0,07	0	0	0,07	0	0	0	0	0	0
$F\ -\ IDF(t,\ d_2)$	0	0	0	0	0	0	0,08	0,08	0,08	0	0
$F\ -\ IDF(t,\ d_3)$	0	0	0	0	0	0	0	0	0	0,08	0,08

Bảng 2.4. Mã hóa TF-IDF của bộ dữ liệu *D*

Từ bảng trên ta có thể thấy được rằng nếu một từ phổ biến hay xuất hiện trong nhiều trong các văn bản thì giá trị IDF sẽ thấp kéo theo TF-IDF cũng thấp theo và ngược lại. Từ đó các từ phổ biến mà không mang ý nghĩa phân loại tin tuyển dụng lừa đảo như “the”, “and”, “an”, “a”, ”i”... sẽ có giá trị thấp hoặc bằng 0, giúp cho các thuật toán học máy tập trung vào những từ hiếm, mang ý nghĩa phân loại tin tuyển dụng lừa đảo hơn.

2.2.3. 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 độ quan trọng của từng từ đơn lẻ. Tuy nhiên, điều này đôi khi khiến mô hình khó phân biệt được sự khác nhau giữa các văn bản có cùng tập hợp từ nhưng khác về thứ tự và ngữ cảnh. Chẳng hạn, xét hai văn bản đơn giản: "people love dogs" và "dogs love people". Khi áp dụng TF–IDF với đơn vị từ là từng từ đơn lẻ (unigram), cả hai văn bản có thể được biểu diễn bằng cùng một vector, ví dụ như [–0.33, –0.33, –0.33], do chứa cùng ba từ nhưng theo thứ tự khác nhau. Điều này có thể gây nhầm lẫn cho mô hình trong việc nhận biết sự khác biệt về ngữ nghĩa.

Để khắc phục hạn chế này, N-gram được sử dụng như một kỹ thuật mở rộng. Thay vì chỉ xét các từ đơn lẻ, N-gram tạo ra các đặc trưng mới trong bộ từ điển bằng cách nhóm n từ liên tiếp trong văn bản lại với nhau. Ví dụ, với n = 2 (bigram), văn bản "people love dogs" sẽ được chuyển thành các cụm từ như: "people love" và "love dogs". Việc này giúp mô hình khai thác được thông tin về thứ tự và ngữ cảnh từ các cụm từ liền kề, từ đó cải thiện độ chính xác trong việc phân loại văn bản – đặc biệt quan trọng trong các bài toán như nhận diện tin tuyển dụng giả và lừa đảo.

2.2.4. Tối ưu từ điển:

Khi làm việc với tập dữ liệu văn bản lớn, việc sinh ra một bộ từ điển có kích thước rất lớn là điều tất yếu. Tuy nhiên, điều này lại không được các mô hình học máy "ưu ái". Việc biểu diễn văn bản dưới dạng vector với số lượng đặc trưng (features) quá lớn sẽ không chỉ làm tăng chi phí tính toán mà còn dễ dẫn đến hiện tượng nhiễu và quá khớp (overfitting). Do đó, cần có bước tối ưu từ điển nhằm giữ lại các từ có ý nghĩa quan trọng đối với nhiệm vụ phân loại, đồng thời loại bỏ những đặc trưng dư thừa hoặc gây nhiễu. Một số phương pháp tiền xử lý phổ biến giúp tối ưu từ điển bao gồm:

- Chuyển tất cả về chữ thường (lowercase): Giúp tránh việc coi hai từ giống nhau về nghĩa nhưng khác nhau về chữ hoa là hai từ khác biệt.
  Ví dụ: ['Hammer', 'hammer'] → ['hammer'].

- Loại bỏ liên kết web (URLs): Các đường dẫn không mang ý nghĩa ngữ cảnh cần thiết và thường gây nhiễu.
  Ví dụ: https://example.com → xóa khỏi văn bản.
- Loại bỏ ký tự không phải tiếng Anh: Giới hạn từ điển chỉ bao gồm từ ngữ thuộc ngôn ngữ được phân tích.
  Ví dụ: 'I love you', '我爱你' → chỉ giữ lại phần tiếng Anh.
- Xóa bỏ dấu câu và ký tự đặc biệt (punctuation): Để đảm bảo các từ được xử lý dưới dạng thuần túy.
  Ví dụ: ['traditional?', 'traditional.'] → ['traditional'].
- Loại bỏ các từ dừng (stop words): Là những từ có tần suất cao nhưng ít mang giá trị phân biệt ngữ nghĩa.
  Ví dụ: ["the", "a", "world"] → ["world"].
- Chuyển về nguyên mẫu (lemmatization hoặc stemming): Giúp gom các biến thể của một từ về một gốc chung.
  Ví dụ: ['go', 'went', 'gone'] → ['go'].

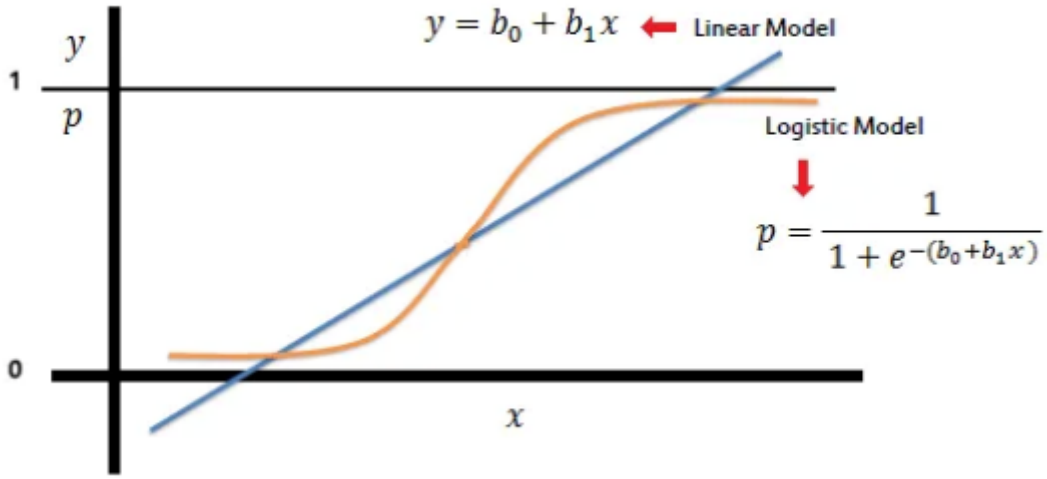
Những bước xử lý này không chỉ giúp giảm kích thước từ điển mà còn giúp tăng tính khái quát, giảm nhiễu và cải thiện hiệu quả học của mô hình – đặc biệt quan trọng trong những bài toán phân tích ngôn ngữ tự nhiên như nhận diện tin tuyển dụng giả, lừa đảo.

2.3. Các mô hình học máy:

2.3.1. Thuật toán Logistic Regression:

Logistic Regression là một thuật toán học có giám sát (supervised learning), thường được sử dụng trong các bài toán phân loại nhị phân, có đầu ra là hai lớp (2 class). Nên Logistic Regression chỉ áp dụng cho bài toán Classification. Thuật toán hoạt động dựa trên việc tính xác suất để suy ra được mỗi điểm dữ liệu thuộc một lớp (class) cụ thể hay tìm ước lượng hồi quy cho xác suất class 0 hay 1 (nhị nguyên) qua hàm

sigmoid trên dữ liệu train. Điểm testpoint được phân loại class dựa vào ước lượng xác suất tại điểm testpoint và ngưỡng threshold = 0.5.



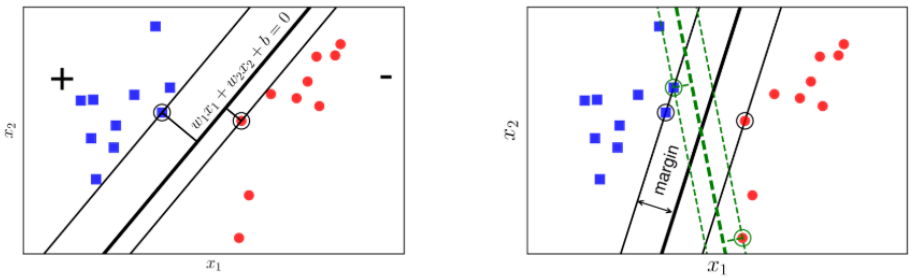
Hình 2.1. Hồi quy Logistic: biến đổi đầu ra tuyến tính thành xác suất bằng hàm sigmoid.

Logistic Regression sử dụng hàm sigmoid để chuẩn hóa đầu ra của mô hình về khoảng giá trị từ 0 đến 1. Hàm sigmoid đóng vai trò quan trọng trong việc chuyển đổi giá trị đầu ra tuyến tính (linear output) thành xác suất, qua đó cho phép mô hình đưa ra quyết định phân loại.

Những đặc điểm trên rất phù hợp cho các bài toán nhận biết công việc thật hay giả thông qua việc tính xác suất thuộc về một trong hai lớp. Một ưu điểm đáng kể của mô hình này là không yêu cầu các biến đầu vào phải tuân theo phân phối chuẩn, điều này đặc biệt phù hợp với các bộ dữ liệu văn bản đã được biến đổi thành dạng vector. Chẳng hạn như TF-IDF vốn không có phân phối chuẩn. Bên cạnh đó, Logistic Regression hoạt động hiệu quả trên dữ liệu có nhiễu, có tốc độ huấn luyện nhanh và dễ tối ưu hóa, nên rất thích hợp với các bộ dữ liệu vừa và lớn như tập dữ liệu Fake Job Postings, nơi yêu cầu xử lý nhanh và hiệu quả là điều cần thiết.

2.3.2. Thuật toán Support vector machine:

Support Vector Machine (SVM) là một trong những thuật toán học máy có giám sát mạnh mẽ và phổ biến, đặc biệt hiệu quả trong các bài toán phân loại và hồi quy. Được giới thiệu từ những năm 1990, SVM hoạt động dựa trên nguyên lý tìm kiếm một siêu phẳng (hyperplane) tối ưu để phân tách các lớp dữ liệu khác nhau trong không gian đặc trưng. Mục tiêu của SVM là tối đa hóa khoảng cách (margin) giữa siêu phẳng phân chia và các điểm dữ liệu gần nhất thuộc mỗi lớp – gọi là các vector hỗ trợ (support vectors)..



Hình 2.2. Siêu phẳng phân tách và khoảng cách biên (margin) trong SVM.

Đây là một thuật toán có hiệu quả cao trong các bài toán phân loại nhị phân, đặc biệt phù hợp với nhiệm vụ nhận biết công việc thật và giả. Một trong những điểm mạnh nổi bật của SVM là khả năng xử lý dữ liệu đã được vector hóa. Khi văn bản được biểu diễn bằng kỹ thuật TF-IDF, dữ liệu thường có số chiều rất lớn và cấu trúc phân tán phức tạp. Trong những trường hợp như vậy, SVM vẫn hoạt động ổn định và hiệu quả, nhờ khả năng hoạt động tốt trong không gian nhiều chiều và xử lý tốt cả những dữ liệu không tuyến tính. Không chỉ mạnh về mặt kỹ thuật, mô hình này còn mang lại hiệu quả thực tiễn rõ rệt khi được ứng dụng vào các hệ thống kiểm duyệt nội dung tuyển dụng, giúp tự động phát hiện các bài đăng công việc không đáng tin cậy.

2.3.3. Thuật toán Naive Bayes:

Naive Bayes (NB) là một thuật toán học có giám sát (supervised learning) đơn giản nhưng hiệu quả, nổi bật với khả năng huấn luyện nhanh và dễ triển khai. Thuật

toán này hoạt động dựa trên định lý Bayes với giả định độc lập giữa các đặc trưng, giúp giảm độ phức tạp trong quá trình tính toán.

Giả sử các đặc trưng X1, ... , Xk của dữ liệu là "độc lập" trên class C. Theo định lý Bayes, xác suất hậu nghiệm:

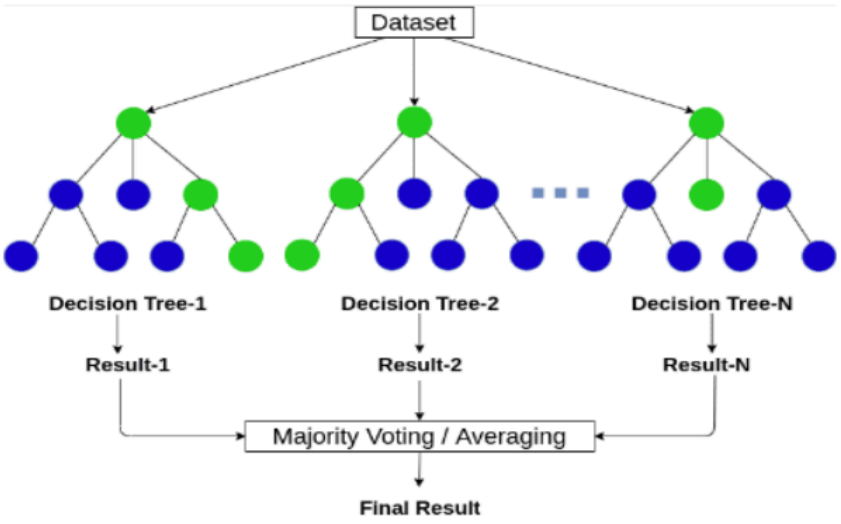
$$P\left(C\middle|X_1,\ldots,X_k\right)=\frac{P(C)P\left(X_1,\ldots,X_k\middle|C\right)}{P(X_1,\ldots,X_k)}$$

trong đó P(C) là phân bố các class (xác suất tiền nghiệm), P(Xi|C) là phân bố đặc trưng thứ i trên class C. Dựa vào tập train để ước lượng cho các phân bố trên và xây dựng mô hình ước lượng các xác suất hậu nghiệm. Khi đó, dự báo nhãn cho dữ liệu test dựa trên xác suất hậu nghiệm lớn nhất.

Đây là một thuật toán đơn giản, có tốc độ huấn luyện nhanh và mang lại hiệu quả cao trong các bài toán phân loại, đặc biệt là phân loại văn bản. Nhờ vào giả định độc lập giữa các đặc trưng, mô hình này giảm thiểu đáng kể độ phức tạp tính toán, giúp rút ngắn thời gian xử lý và phù hợp với các hệ thống cần phản hồi nhanh. Đối với đề tài nhận biết công việc thật – giả, Naive Bayes tỏ ra đặc biệt phù hợp vì dữ liệu văn bản đã được chuẩn hóa và vector hóa (chẳng hạn bằng TF-IDF), vốn là môi trường lý tưởng để thuật toán này phát huy hiệu quả. Tính đơn giản, dễ triển khai cùng tốc độ huấn luyện nhanh khiến Naive Bayes trở thành một lựa chọn hợp lý cho các hệ thống phát hiện và kiểm duyệt tin tuyển dụng giả mạo.

2.3.4. Thuật toán Random Forest:

Random Forest là một thuật toán học máy thuộc nhóm học có giám sát (supervised learning) và được sử dụng phổ biến trong các bài toán phân loại (classification) và hồi quy (regression). Thuật toán này là một dạng của tập hợp học (ensemble learning), nơi mà nhiều mô hình yếu (weak learners), cụ thể là các cây quyết định (decision trees), được kết hợp lại để tạo thành một mô hình mạnh mẽ hơn.



Hình 2.3. Mô hình Random Forest: nhiều cây quyết định bỏ phiếu đa số

Thuật ngữ "Random" trong Random Forest xuất phát từ hai yếu tố chính:

- Ngẫu nhiên trong chọn mẫu Thay vì sử dụng toàn bộ dữ liệu huấn luyện để xây dựng từng cây quyết định, thuật toán Random Forest chọn một mẫu ngẫu nhiên từ tập dữ liệu (với hoàn lại) để xây dựng mỗi cây. Kỹ thuật này được gọi là Bagging (Bootstrap Aggregating). Bagging giúp giảm thiểu phương sai của mô hình, cải thiện độ chính xác tổng thể.
- Ngẫu nhiên trong chọn đặc trưng Khi tạo các nút trong mỗi cây, chỉ một tập con ngẫu nhiên của tất cả các đặc trưng được xem xét để chọn đặc trưng tốt nhất tại mỗi bước. Điều này giúp các cây quyết định đa dạng hơn, giảm thiểu hiện tượng overfitting và đảm bảo rằng các cây không bị phụ thuộc quá mức vào một đặc trưng cụ thể nào đó.

Công thức tổng quát

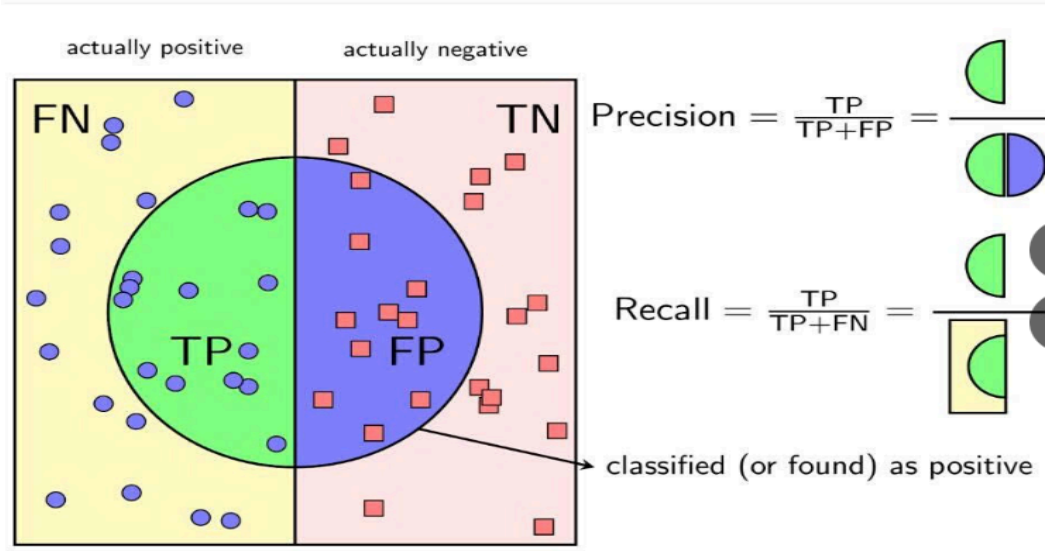
Một cây quyết định trong Random Forest thực hiện phân loại hoặc hồi quy bằng cách chia nhỏ không gian đặc trưng thành các vùng con. Các phân vùng này được xác định dựa trên các điều kiện phân tách tại mỗi nút trong cây. Giả sử có một đặc trưng $\boldsymbol{X}$ và một ngưỡng phân tách $\boldsymbol{t}$, việc phân tách tại một nút có thể được biểu diễn bằng cách chọn một hàm chỉ thị $\boldsymbol{I}$, trong đó:

$$I(X \leq t) \text{ và } I(X > t)$$

Nếu đặc trưng ***X*** tại mẫu đó nhỏ hơn hoặc bằng ***t***, mẫu sẽ được chuyển đến nhánh trái; ngược lại, nó sẽ được chuyển đến nhánh phải. Quá trình này tiếp tục cho đến khi đạt đến một nút lá, nơi giá trị đầu ra của nút đó được sử dụng làm dự đoán.

Random Forest là thuật toán học máy mạnh mẽ, hoạt động bằng cách xây dựng nhiều cây quyết định và tổng hợp kết quả để tăng độ chính xác và giảm overfitting. Nhờ khả năng xử lý tốt dữ liệu phức tạp và nhiễu, Random Forest đặc biệt phù hợp cho bài toán nhận diện tin tuyển dụng thật – giả, nơi dữ liệu văn bản sau khi vector hóa (TF-IDF) chứa nhiều đặc trưng đa chiều. Thuật toán này giúp xác định các yếu tố quan trọng, đảm bảo mô hình hoạt động ổn định và hiệu quả. Tính chính xác cao, linh hoạt và khả năng chịu nhiễu tốt khiến Random Forest trở thành lựa chọn hợp lý cho các hệ thống phát hiện và kiểm duyệt tin tuyển dụng giả mạo.

2.3.5. Các chỉ số đánh giá hiệu suất mô hình:



Hình 2.4. Trực quan Precision–Recall với ma trận nhầm lẫn

Để đánh giá hiệu quả của các mô hình học máy trong bài toán nhận diện tin tuyển dụng giả, tôi sử dụng một tập hợp các chỉ số đánh giá phân loại nhị phân phổ biến. Các chỉ số

này sẽ cung cấp những góc nhìn đa chiều về khả năng phân biệt giữa tin giả (positive) và tin thật (negative), từ đó giúp chọn được mô hình tối ưu nhất. Việc kết hợp nhiều chỉ số sẽ đảm bảo đánh giá toàn diện cả về độ chính xác tổng thể, khả năng phát hiện tin giả, cũng như mức độ tin cậy khi báo một tin là giả.

Dưới đây là ma trận nhầm lẫn (confusion matrix) và các chỉ số tính toán:

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 chính xác bao nhiêu phần trăm trong toàn bộ các trường hợp, bao gồm cả dự đoán đúng cho lớp tin tuyển dụng giả (positive) và lớp tin tuyển dụng thật (negative):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Trong đó:

- TP (True Positive): Số tin giả được dự đoán đúng.
- TN (True Negative): Số tin thật được dự đoán đúng.
- FP (False Positive): Số tin thật nhưng bị dự đoán nhầm là tin giả.
- FN (False Negative): Số tin giả nhưng bị bỏ sót (dự đoán thành tin thật).

Mặc dù trực quan và dễ hiểu, Accuracy có thể đánh giá sai lệch khi dữ liệu mất cân bằng, ví dụ trong bài toán nhận diện tin tuyển dụng giả,nơi số lượng tin thật thường áp đảo so với tin giả.

Độ chính xác dương tính (Precision): đo lường tỷ lệ trong số các tin mà mô hình gắn nhãn là giả, có bao nhiêu phần trăm thực sự là tin tuyển dụng lừa đảo:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Trong đó:

TP (True Positive): Số tin giả được dự đoán đúng.

FP (False Positive): Số tin thật nhưng bị gắn nhãn nhầm là tin giả.

Precision quan trọng khi hậu quả của cảnh báo giả (FP) là tốn kém hoặc gây ảnh hưởng không cần thiết. Trong bài toán nhận diện tin tuyển dụng giả, cảnh báo giả nghĩa là gắn nhãn nhầm tin thật thành tin giả, gây gián đoạn quy trình tuyển dụng, tốn công kiểm duyệt và ảnh hưởng đến uy tín của nhà tuyển dụng. Tuy nhiên, nếu tối ưu Precision mà Recall lại thấp, mô hình sẽ bỏ sót nhiều tin giả thực sự (FN), làm suy giảm vai trò phòng ngừa rủi ro đối với người tìm việc.

Độ nhạy (Recall) – Tỷ lệ phát hiện đúng: đo lường trong số các tin giả thực sự, mô hình dự đoán được bao nhiêu phần trăm:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Trong đó:

TP (True Positive): Tin giả được dự đoán đúng là giả.

FN (False Negative): Tin giả nhưng bị bỏ sót (dự đoán thành tin thật).

Trong bài toán nhận diện tin tuyển dụng giả, Recall thường được ưu tiên vì bỏ sót một tin lừa đảo (FN) có thể gây hại cho người tìm việc và làm giảm uy tín nền tảng. Trong khi đó, cảnh báo nhầm (FP) thường "chỉ" khiến tin thật bị rà soát thêm.

Điểm F1 (F1-Score): là trung bình giữa Precision và Recall (với lớp dương tính là tin tuyển dụng giả). Chỉ số này đặc biệt hữu ích khi cần một thước đo cân bằng giữa việc phát hiện tin giả (Recall) và ít nhầm với tin thật (Precision), nhất là trong bối cảnh dữ liệu mất cân bằng nơi tin thật áp đảo.

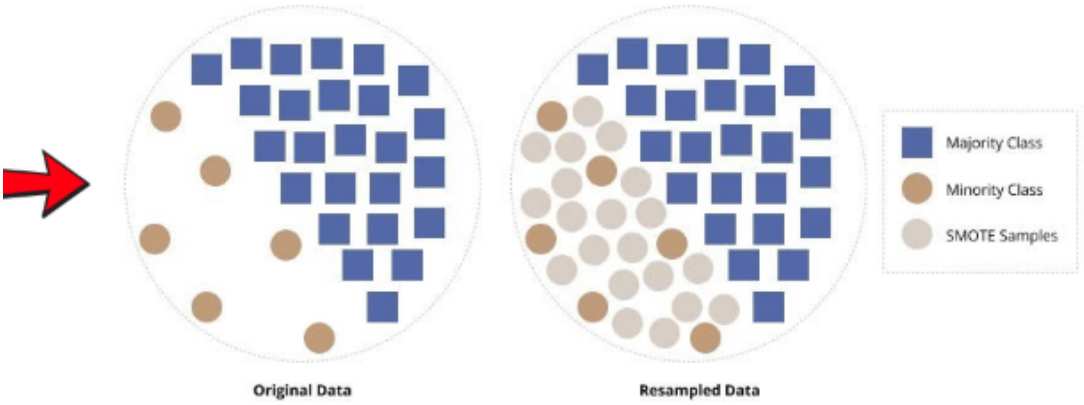
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Điểm F1 cao cho thấy mô hình phát hiện được nhiều tin tuyển dụng giả (Recall cao) đồng thời ít nhầm lẫn tin thật (Precision cao) — tức cân bằng tốt giữa khả năng phát hiện và độ tin cậy của cảnh báo.

2.3.6. Xử lý mất cân bằng lớp:

Trong nhiều bài toán học máy, dữ liệu thường bị mất cân bằng khi lớp thiểu số có rất ít mẫu, như trong bài toán phát hiện tin tuyển dụng lừa đảo – nơi đa số tin đăng là hợp pháp. Nếu không xử lý, mô hình dễ bỏ qua lớp lừa đảo quan trọng. Vì vậy, cần áp dụng các kỹ thuật như SMOTE hoặc gán trọng số (class weight) để giúp mô hình học tốt cả hai lớp và phát hiện chính xác các trường hợp bất thường.

- **SMOTE** (Synthetic Minority Over-sampling Technique) là một kỹ thuật xử lý mất cân bằng lớp trong học máy bằng cách **tăng cường dữ liệu của lớp thiểu số thông qua việc tạo ra các mẫu tổng hợp mới**. Thay vì chỉ sao chép các điểm dữ liệu hiện có, SMOTE tạo ra các điểm mới bằng cách nội suy giữa một điểm dữ liệu thiểu số và các láng giềng gần nhất của nó trong không gian đặc trưng. Bằng cách này, SMOTE giúp mở rộng vùng không gian của lớp thiểu số, từ đó giảm hiện tượng overfitting và giúp mô hình học được ranh giới phân lớp tốt hơn. Kỹ thuật này đặc biệt hữu ích trong các bài toán mà tỷ lệ giữa lớp đa số và lớp thiểu số bị chênh lệch nghiêm trọng, như trong phát hiện gian lận hay chẩn đoán bệnh hiếm gặp.



Hình 2.5. Mất cân bằng lớp và dữ liệu sau tái lấy mẫu bằng SMOTE

❖ **CLASS WEIGHT** là một phương pháp đơn giản nhưng hiệu quả để xử lý vấn đề mất cân bằng lớp bằng cách **gán trọng số lớn hơn cho lớp thiểu số trong quá trình huấn luyện mô hình.** Khi sử dụng class weight, các lỗi dự đoán sai ở lớp thiểu số sẽ bị phạt nặng hơn so với lớp đa số, từ đó buộc mô hình chú ý nhiều hơn đến lớp ít dữ liệu. Phương pháp này thường được tích hợp sẵn trong nhiều thuật toán học máy như Logistic Regression, SVM. Ưu điểm lớn của kỹ thuật class weight là không làm thay đổi dữ liệu gốc, dễ áp dụng và tiết kiệm tài nguyên tính toán, đặc biệt phù hợp với các mô hình nhạy cảm với phân phối lớp và dữ liệu hiếm.

$$Trọng\ số\ của\ lớp = \frac{1}{số\ lượng\ mẫu\ của\ lớp}$$

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ MÔ HÌNH

3.1. Bộ dữ liệu:

Dữ liệu sử dụng trong đề tài được lấy từ bộ "Fake Job Posting" trên nền tảng Kaggle. Bộ dữ liệu này cung cấp thông tin chi tiết về các tin tuyển dụng thật và giả, bao gồm các trường như tiêu đề công việc, mô tả, yêu cầu, địa điểm… Đây là nguồn dữ liệu phù hợp để huấn luyện và đánh giá mô hình phát hiện tin tuyển dụng lừa đảo dựa trên các kỹ thuật học máy và xử lý ngôn ngữ tự nhiên. Bộ dữ liệu bao gồm 18 cột (cột), mỗi dòng (bản ghi) là một tuyển dụng. Dưới đây là danh sách đầy đủ các cột được giải thích ngắn gọn:

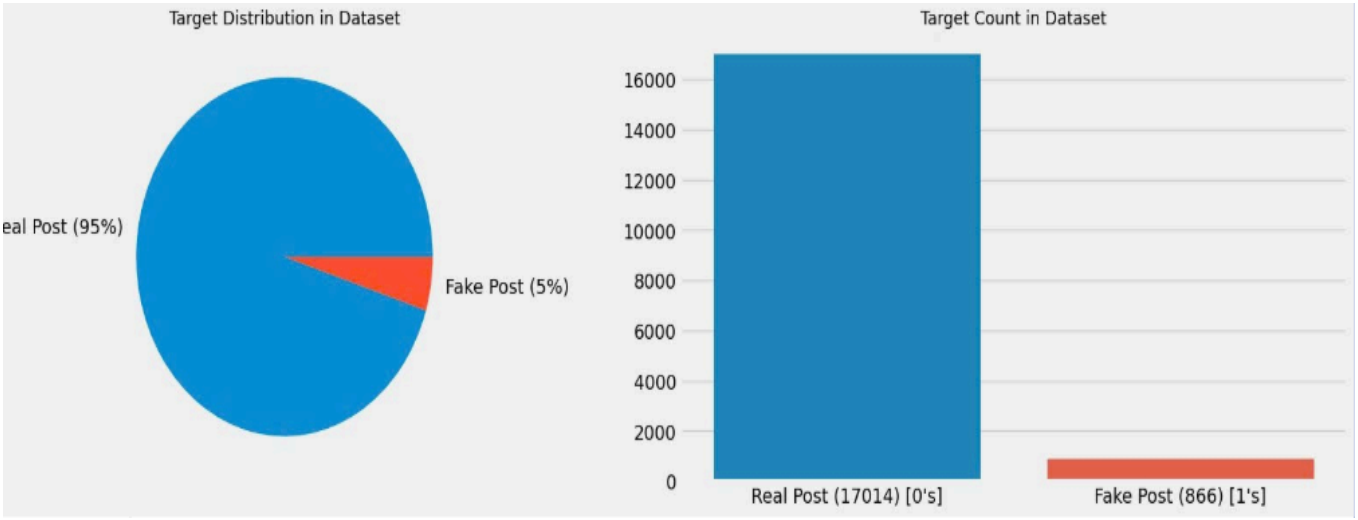
Tên cột	Kiểu dữ liệu	Miêu tả
job_id	integer	Mã định danh duy nhất cho mỗi tin tuyển dụng(có thể bỏ khi xử lý).
title	string	Tiêu đề công việc (ví dụ: "Data Scientist", "Marketing Intern").
location	string	Địa điểm làm việc (ví dụ: "New York, NY, US" hoặc "Remote").
department	string	Bộ phận/nhóm (ví dụ: "Engineering", "Marketing").
salary_range	string	Khoảng lương (nếu có; có thể chuyển đổi thành "80-100", hoặc NA).
company_profile	string	Mô tả ngắn gọn về công ty.
description	string	Mô tả chi tiết công việc.
requirements	string	Yêu cầu công việc (bằng cấp).
benefits	string	Quyền lợi/phúc lợi đi kèm (nếu có).
telecommuting	integer(0/1)	1 nếu cho phép làm việc từ xa (remote);

		0 nếu không.
has_company_logo	nteger(0/1)	1 nếu tin đăng có kèm logo công ty; 0 nếu không.
has_questions	nteger(0/1)	1 nếu có trường câu hỏi/bài kiểm tra (contact questions) ; 0 nếu không.
employment_type	string	Loại hình tuyển dụng (“Full-time”, “Part-time”, “Contract”, “Internship”,…).
required_experience	string	Mức độ kinh nghiệm (“Cấp đầu vào”, “Cấp trung cấp”, “Giám đốc”, “Điều hành”, “Không xác định”).
required_education	string	Trình độ học vấn (“Trung học phổ thông”, “Cử nhân”, “Thạc sĩ”, “Tiến sĩ”, “Không xác định”).
industry	string	Ngành nghề (ví dụ: “Công nghệ thông tin”, “Tiếp thị và Quảng cáo”, “Ngân hàng”).
function	string	Chức năng/chuyên môn (ví dụ: “Kỹ thuật”, “Bán hàng”, “Tài chính”).
fraudulent	integer(0/1)	Nhãn mục tiêu: 0 = tin thật, 1 = tin tuyển dụng lừa đảo (việc làm giả).

❖ **Mô tả dữ liệu:**

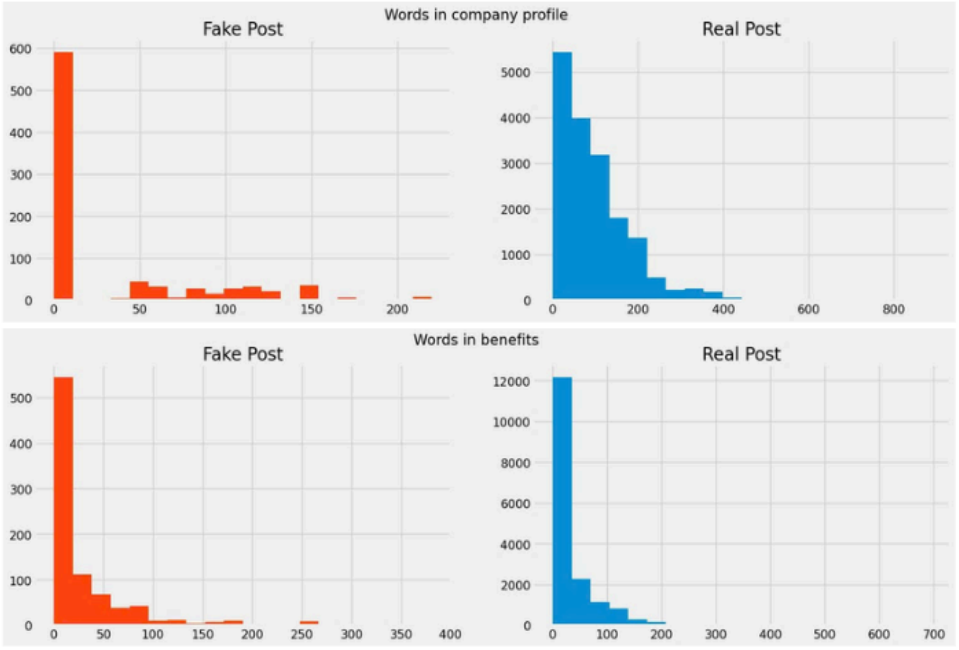
Các thuộc tính Fake-job posting:

- Thuộc tính văn bản: title, salary_range, description, requirements, benefits, company_profile, location.
- Thuộc tính hạng mục: department, industry, employment_type, required_experience, required_education, function.
- Thuộc tính nhị phân: telecommuting, has_company_logo, has_questions.



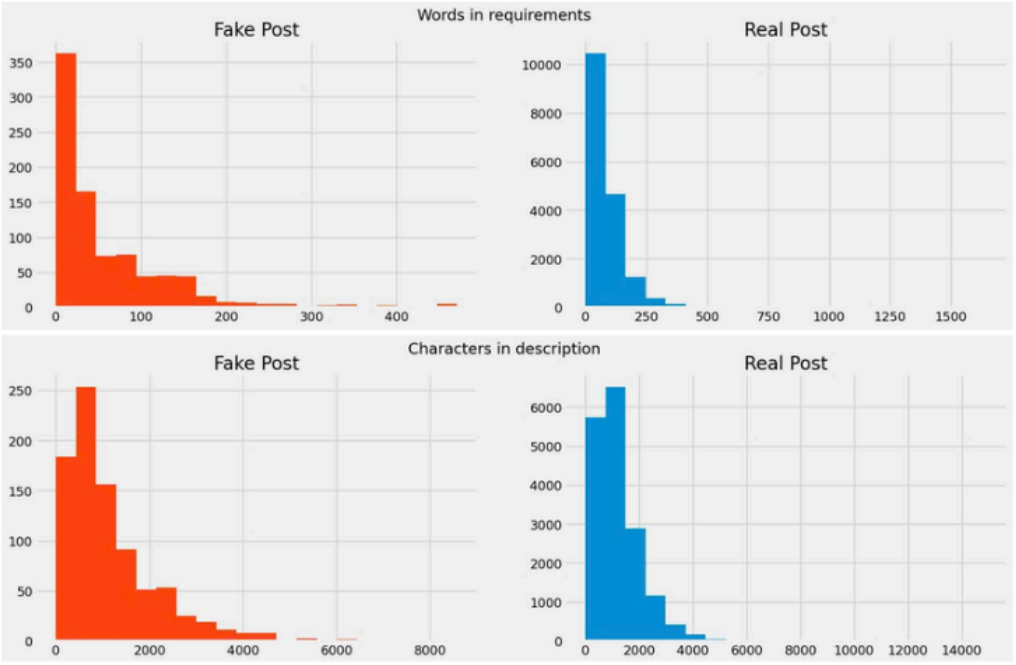
Hình 3.1. Biểu đồ tròn và cột cho thấy dữ liệu mất cân bằng, với Real (0) chiếm ~95% (17.014 mẫu) và Fake (1) chiếm ~5% (866 mẫu).

❖ **Số lượng từ trong các cột văn bản:**

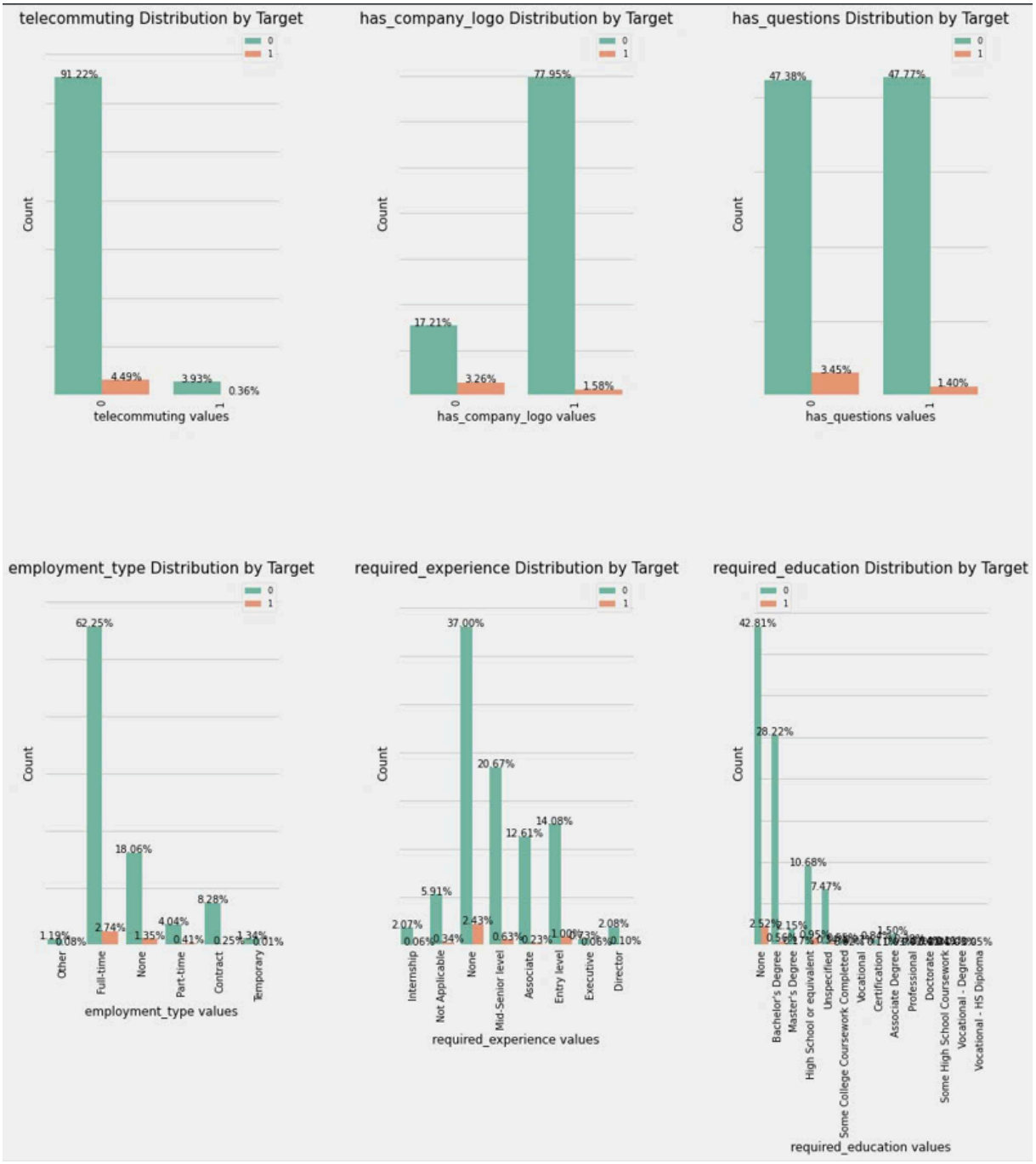


- Thông tin công ty
- Lợi ích công việc

Hình 3.2. Phân bố số từ trong hai trường mô tả: Company profile, Benefits theo nhãn bài đăng. Tin giả (đỏ) hầu hết rất ngắn (≈≤50 từ), trong khi tin thật (xanh) dài và phân tán hơn.



Hình 3.3. Phân bố độ dài nội dung theo nhãn bài đăng: Requirements, Description. Tin giả (đỏ) chủ yếu rất ngắn và tập trung ở mức thấp, trong khi tin thật (xanh) dài hơn và phân tán rộng.



Hình 3.4 .Phân bố thuộc tính theo nhãn: Tin thật(Real) áp đảo,có logo thường thuộc lớp 0; Tin giả(Fake) thường rơi vào None/Other và yêu cầu thấp.

3.2. Tiền xử lý:

3.2.1. Tiền xử lý cho các cột không phải là cột văn bản:

- Xử lý chung: xóa các mẫu dữ liệu trùng nhau, xóa cột job_id(không có ý nghĩa đến việc phân loại).
- Hạng mục nhị phân: không có missing.(dữ liệu thiếu)
- Hạng mục nominal:

B1. Điền missing bằng cách suy luận từ cột khác(requirement -> require education; requirement,title -> require experience).

B2. Điền mode(employment type).

B3. One hot encoding.

3.2.2. Tiền xử lý các cột văn bản:

B1. Loại bỏ mọi URL và các ký hiệu #URL…#.

B2. Xóa entity HTML (&, <…) và thẻ HTML, chỉ giữ lại text thuần.

B3. Xóa các chữ cái không phải tiếng anh.

B4. Dùng WordNinja tách từ dính (Ví dụ “machinelearning” → “machine learning”).

B5. Gộp nhiều khoảng trắng thành một,xóa khoảng trắng hai đầu chuẩn hóa chúng.

B6. Quy trình sẽ áp dụng cho cột text được ghép từ “title”,” location”, “department”, “description”, “company_profile”, “benefits”, “requirements”, “salary_range”.

B7. Sử dụng mã hóa TF-IDF cho cột text vừa được ghép.

3.3. Đánh giá các mô hình học máy:

3.3.1. Mô hình Logistic Regression:

	precision	recall	f1-score	support
0	0.97	1.00	0.99	3331
1	0.94	0.54	0.69	189
accuracy			0.97	3520
macro avg	0.96	0.77	0.84	3520
weighted avg	0.97	0.97	0.97	3520

Accuracy: 0.9739

Hình 3.5. Mô hình logistic mặc định

Accuracy: 0.9869

	precision	recall	f1-score	support
0	0.99	0.99	0.99	3331
1	0.89	0.87	0.88	189
accuracy			0.99	3520
macro avg	0.94	0.93	0.94	3520
weighted avg	0.99	0.99	0.99	3520

Hình 3.6. Mô hình logistic sau khi đã tối ưu

Kết quả cho thấy việc tối ưu mô hình logistic đã cải thiện đáng kể hiệu suất phân loại, đặc biệt với lớp lừa đảo (lớp 1)– đối tượng chính của bài toán. Ở mô hình mặc định, mặc dù độ chính xác tổng thể khá cao (accuracy = 97.39%), nhưng recall của lớp 1 chỉ đạt 0.54, cho thấy mô hình bỏ sót nhiều tin tuyển dụng lừa đảo. Điều này không phù hợp với mục tiêu phát hiện các trường hợp lừa đảo.

Sau khi tối ưu, accuracy tăng lên 98.69%, và đặc biệt recall của lớp 1 tăng mạnh lên 0.87, đồng thời F1-score đạt 0.88, thể hiện sự cân bằng tốt hơn giữa precision và recall. Dù precision lớp 1 giảm nhẹ từ 0.94 xuống 0.89, nhưng sự đánh đổi này là hợp lý để cải thiện khả năng phát hiện tin lừa đảo. Ngoài ra, các chỉ số trung bình như macro avg và weighted avg cũng được nâng cao rõ rệt, cho thấy mô hình tối ưu hoạt động tốt hơn trên cả hai lớp.

3.3.2. Mô hình SVM:

Cũng như mô hình logistic regression trước ở mô hình mặc định, độ chính xác đạt 97.30%, tuy nhiên recall của lớp tin tuyển dụng lừa đảo chỉ ở mức 0.52, cho thấy mô hình bỏ sót nhiều trường hợp quan trọng. Sau khi áp dụng các kỹ thuật tối ưu, độ chính xác tăng lên 98.69%, đồng thời recall của lớp tin tuyển dụng lừa đảo tăng mạnh lên 0.87, precision đạt 0.89 và F1-score đạt 0.88. Điều này cho thấy mô hình sau tối ưu không chỉ phân loại chính xác hơn mà còn cân bằng tốt hơn giữa hai lớp, đặc biệt là trong việc nhận diện tin tuyển dụng lừa đảo – lớp có số lượng mẫu ít hơn.

Accuracy: 0.9730
None

	precision	recall	f1-score	support
0	0.97	1.00	0.99	3331
1	0.95	0.52	0.68	189
accuracy			0.97	3520
macro avg	0.96	0.76	0.83	3520
weighted avg	0.97	0.97	0.97	3520

Hình 3.7. Mô hình SVM mặc định

Accuracy: 0.9869
None

	precision	recall	f1-score	support
0	0.99	0.99	0.99	3331
1	0.89	0.87	0.88	189
accuracy			0.99	3520
macro avg	0.94	0.93	0.94	3520
weighted avg	0.99	0.99	0.99	3520

Hình 3.8. Mô hình SVM sau khi đã tối ưu

3.3.3. Mô hình Naive Bayes:

Sau khi tối ưu, mô hình Naive Bayes đạt độ chính xác 96%, cao hơn mô hình mặc định ban đầu (93.8%). Đặc biệt, F1-score của lớp tin tuyển dụng lừa đảo (lớp 1) tăng từ 0.60 lên 0.69, với recall giữ ở mức cao (0.85), cho thấy khả năng phát hiện tin tuyển dụng lừa đảo đã được cải thiện rõ rệt. Tuy vậy, precision của lớp này (0.58) vẫn thấp hơn so với hai mô hình còn lại. So với Logistic Regression và SVM sau tối ưu, Naive Bayes có hiệu suất tổng thể thấp hơn một chút (F1-score lớp 1 lần lượt là 0.88 ở cả hai mô hình kia), nhưng có ưu điểm là nhẹ, dễ triển khai và vẫn đảm bảo được mức cân bằng tốt giữa precision và recall. Do đó, Naive Bayes có thể là lựa chọn phù hợp trong các hệ thống yêu cầu phản hồi nhanh hoặc tài nguyên hạn chế.

Accuracy: 0.9383522727272727

	precision	recall	f1-score	support
Class 0	0.99	0.94	0.97	3331
Class 1	0.46	0.88	0.60	189
accuracy			0.94	3520
macro avg	0.73	0.91	0.79	3520
weighted avg	0.96	0.94	0.95	3520

Hình 3.9. Mô hình Naïve Bayes mặc định

	precision	recall	f1-score	support
0	0.99	0.97	0.98	3331
1	0.58	0.85	0.69	189
accuracy			0.96	3520
macro avg	0.79	0.91	0.83	3520
weighted avg	0.97	0.96	0.96	3520

Hình 3.10. Mô hình Naïve Bayes sau khi đã tối ưu

3.3.4. Mô hình Random Forest:

Sau khi tối ưu các tham số mô hình Random Forest đạt Accuracy 97.93%, với lớp tin tuyển dụng giả (lớp 1) cho Precision 1.00, Recall 0.57 và F1-score 0.73. So với mô hình mặc định. Kết quả đã cải thiện nhẹ về khả năng phát hiện tin giả đồng thời duy trì độ tin cậy cảnh báo rất cao (Precision = 1.00). Tuy nhiên, nếu so với Logistic Regression và SVM sau tối ưu (Accuracy 98.69%, Recall lớp 1 = 0.87, F1 = 0.88), thì Recall,F1 của Random Forest còn thấp, nghĩa là vẫn bỏ sót một tỉ lệ đáng kể tin giả và cần tiếp tục điều chỉnh hoặc trọng số lớp để nâng khả năng phát hiện. Mô hình phù hợp cho các hệ thống cần dự đoán ổn định, chịu nhiễu tốt và vẫn đáp ứng thời gian phản hồi nhanh, có thể triển khai hiệu quả trên tài nguyên hạn chế khi tối ưu số cây và độ sâu mô hình.

Accuracy: 0.9759

	precision	recall	f1-score	support
0	0.98	1.00	0.99	3331
1	1.00	0.55	0.71	189
accuracy			0.98	3520
macro avg	0.99	0.78	0.85	3520
weighted avg	0.98	0.98	0.97	3520

Hình 3.11. Mô hình Random Forest mặc định

Accuracy: 0.97926136363636

	precision	recall	f1-score	support
0	0.98	1.00	0.99	3349
1	1.00	0.57	0.73	171
accuracy			0.98	3520
macro avg	0.99	0.79	0.86	3520
weighted avg	0.98	0.98	0.98	3520

Hình 3.12. Mô hình Random Forest khi đã tối ưu

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Đánh giá tổng quan:

Đề tài đã xây dựng được một quy trình phát hiện tin tuyển dụng giả gồm: làm sạch dữ liệu văn bản (loại URL, HTML, chuẩn hóa…), ghép các trường văn bản và mã hóa TF–IDF để đưa vào các mô hình có giám sát (Logistic Regression, SVM, Naive Bayes). Việc đánh giá được thực hiện trên tập kiểm thử bằng các chỉ số Accuracy, Precision, Recall, F1-score, với lớp dương tính là tin giả (fraudulent = 1).

❖ **Những kết quả đạt được:**

- Pipeline tiền xử lý và đặc trưng: Hoàn thiện chuỗi xử lý văn bản và đặc trưng TF–IDF, là nền tảng cho toàn bộ thí nghiệm mô hình.

➢ **Hiệu quả mô hình (tóm tắt các con số nổi bật):**

- **Logistic Regression**: sau tối ưu, Accuracy 98.69%, Recall (lớp giả) 0.87, F1 = 0.88; chấp nhận đánh đổi Precision lớp giả từ 0.94 xuống 0.89 để tăng khả năng phát hiện tin giả. So với mặc định (Accuracy 97.39%, Recall lớp 1 = 0.54), mức cải thiện là rõ rệt.
- **SVM**: mặc định Recall lớp giả chỉ 0.52, sau tối ưu đạt Accuracy 98.69%, Recall 0.87, Precision 0.89, F1 = 0.88 – cân bằng tốt hơn giữa hai lớp và đáp ứng mục tiêu phát hiện tin giả.
- **Naive Bayes**: sau tối ưu Accuracy ~96%; tuy nhanh và đơn giản, nhưng hiệu quả tổng thể kém hơn LR/SVM trên dữ liệu văn bản của đề tài.
- **Random Forest:**

=> **Thiết lập đánh giá rõ ràng:** nhấn mạnh Precision/Recall/F1 cho lớp dương tính (tin giả) – phù hợp mục tiêu bảo vệ người tìm việc.

❖ **Một số hạn chế:**

- **Mất cân bằng lớp:** đặc thù dữ liệu thật–giả lệch nhiều (đa số tin hợp pháp) khiến mô hình có nguy cơ bỏ qua lớp hiếm nếu không xử lý/điều chỉnh ngưỡng cẩn thận.
- **Phạm vi triển khai:** báo cáo tập trung vào xây dựng và đánh giá mô hình, chưa ưu tiên phần giao diện/ứng dụng thực tiễn.

4.2. Hướng phát triển:

- **Mở rộng dữ liệu & làm giàu đặc trưng**: Thu thập thêm tin ở nhiều nguồn/thời điểm; bổ sung đặc trưng "phi văn bản" (mức lương, liên hệ, số liên kết, độ dài mô tả…) để hỗ trợ mô hình.
- **Cải thiện biểu diễn văn bản**: Thử nghiệm N-gram mở rộng, stopword chuyên biệt lĩnh vực, và đặc biệt mô hình ngôn ngữ tiền huấn luyện (BERT, RoBERTa) để nắm bắt ngữ nghĩa tốt hơn.
- **Tối ưu theo mục tiêu nghiệp vụ**: Hiệu chỉnh ngưỡng xác suất để đạt Recall/Precision mong muốn; calibration xác suất (Platt/Isotonic) nhằm tăng độ tin cậy.
- **Xử lý mất cân bằng**: So sánh class_weight, SMOTE (với đặc trưng phi văn bản) và undersampling; kiểm tra Balanced Random Forest như một baseline bổ sung.
- **Diễn giải & kiểm định**: Dùng permutation importance/SHAP(tham khảo) để phân tích từ khóa/thuộc tính tác động; k-fold CV và thí nghiệm nhiều seed để đo độ ổn định.
- **Triển khai thử nghiệm**: Tích hợp mô hình vào quy trình kiểm duyệt; thiết kế cảnh báo hai mức (tự động/đề nghị duyệt tay) theo xác suất, ghi log FP/FN để học tăng cường (active learning).

❖ KẾT LUẬN:

Bài toán phát hiện tin tuyển dụng lừa đảo là một thách thức quan trọng trong xử lý ngôn ngữ tự nhiên, đòi hỏi mô hình phân loại vừa chính xác vừa nhạy bén để hạn chế bỏ sót tin tuyển dụng lừa đảo. Qua việc triển khai và tối ưu ba mô hình Logistic Regression, SVM Naive Bayes và Random Forest, kết quả cho thấy SVM và Logistic Regression vượt trội hơn về độ chính xác và khả năng phân loại lớp tin tuyển dụng lừa đảo với F1-score cao và recall tốt. Mô hình Naive Bayes tuy đơn giản và nhanh nhưng gặp hạn chế về độ chính xác và độ tin cậy trong phát hiện tin tuyển dụng lừa đảo.

Dựa trên các đánh giá toàn diện, Logistic Regression được lựa chọn làm mô hình tối ưu nhờ hiệu năng ổn định, khả năng tổng quát tốt và phù hợp với đặc điểm dữ liệu hiện tại. Kết quả này mở ra cơ hội ứng dụng rộng rãi trong các hệ thống kiểm duyệt và phân tích tin tức tự động, góp phần nâng cao chất lượng thông tin trên mạng xã hội và các nền tảng truyền thông số.

Để phát triển tiếp theo, có thể thử nghiệm các mô hình phức tạp hơn hoặc kết hợp nhiều kỹ thuật nhằm nâng cao hơn nữa hiệu quả phát hiện và giảm thiểu sai sót trong thực tế.

TÀI LIỆU THAM KHẢO

[1] Bakirarar, B., & Elhan, A. H. (2023). Class Weighting Technique to Deal with Imbalanced Class Problem in Machine Learning: Methodological Research. Turkiye Klinikleri Journal of Biostatistics, 15(1), 19–29. https://doi.org/10.5336/biostatic.2022-93961

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[3] Cramer, J. (2003). The origins of logistic Regression. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.360300

[4] Hapke, H., Howard, C., & Lane, H. (2019). Natural language processing in action: Understanding, analyzing, and generating text with Python. Simon and Schuster.

[5] Pattern recognition and machine learning. (2006). In Springer eBooks. https://doi.org/10.1007/978-0-387-45528-0

[6] Seel, N. M. (2012). Pattern classification. In Springer eBooks (p. 2560). https://doi.org/10.1007/978-1-4419-1428-6_5195