Trường Đại học Sài Gòn
Khoa Toán - Ứng dụng

Thành phố Hồ Chí Minh, năm 2025

# DỰ ĐOÁN TIN TUYỂN DỤNG GIẢ MẠO BẰNG KỸ THUẬT KHAI PHÁ DỮ LIỆU

**Giáo viên hướng dẫn:** Thầy Đỗ Như Tài

**Học phần:**
Khai phá dữ liệu

**Năm học:**
HK1 - 2025

**Đề tài thi cuối kì**

# Giới thiệu thành viên

| STT | Họ & Tên | MSSV | Nhiệm vụ |
|---|---|---|---|
| 1 | Lê Phước Thành | 3123580045 | Trưởng nhóm, XGBoost, viết báo cáo, Quản lí Git |
| 2 | Nguyễn Hoàng Long | 3123580022 | Tiền xử lí dữ liệu, Random Forest, viết báo cáo |
| 3 | Đường Minh Đức | 3123580010 | EDA, Naive Bayes |
| 4 | Hà Tuấn Duy | 3123580006 | Logistic Regression, làm powerpoint |

# Nội dung trình bày

| | |
|---|---|
| **Chương 1** | **Giới thiệu đề tài** |
| **Chương 2** | **Dữ liệu và phương pháp đề xuất** |
| **Chương 3** | **Thực nghiệm, kết quả và thảo luận** |
| **Chương 4** | **Kết luận và trả lời câu hỏi nghiên cứu** |

# Chương 1: Giới thiệu đề tài

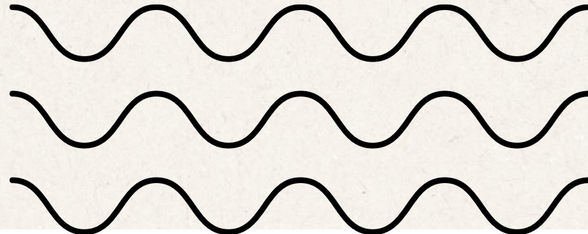
## Lí do chọn đề tài

- **Thực trạng:** Xu hướng tuyển dụng trực tuyến bùng nổ nhưng đi kèm rủi ro về tin giả mạo gia tăng.
- **Hệ lụy:** Gây thiệt hại tài chính, lộ thông tin cá nhân và giảm uy tín của doanh nghiệp chân chính.
- **Giải pháp:** Xây dựng hệ thống tự động phát hiện/phân loại tin tuyển dụng thật - giả là nhu cầu cấp thiết.

---

## Mục tiêu đề tài

- **Về mặt xã hội:** Bảo vệ người tìm việc, xây dựng môi trường tuyển dụng minh bạch.
- **Về mặt học thuật:** Ứng dụng kỹ thuật Khai phá dữ liệu (NLP, Machine Learning, Deep Learning) vào bài toán phân loại nhị phân thực tế.



# Chương 1: Giới thiệu đề tài

## Câu hỏi nghiên cứu

1. Thuật toán máy học nào cho hiệu suất cao nhất trong việc phân loại tin tuyển dụng?
2. Làm thế nào để cân bằng giữa chỉ số precision và recall
3. Việc áp dụng các kỹ thuật tái lấy mẫu như smote thì liệu có ảnh hưởng như thế nào đến khả năng phát hiện tin giả của mô hình?
4. Những đặc trưng chủ yếu về văn bản hoặc từ khóa (keywords) nào mang tính phân biệt cao nhất giúp nhận diện một tin tuyển dụng giả mạo?
5. Phương pháp trích xuất đặc trung nào (TF-IDF, Count Vectorized, Word Embeddings như Word2Vec) mang lại đầu vào tốt nhất cho mô hình phân loại trong ngữ cảnh này?

# Chương 1: Giới thiệu đề tài

## Các phương pháp nghiên cứu

- Dataset: EMSCAD (17,880 tin tuyển dụng)
- Ngôn ngữ: Python
- Thư viện: pandas, numpy, scikit-learn, matplotlib, seaborn
- Các thuật toán: Logistic Regression, Random Forest, XGBoost, Naïve Bayes, UnderSampling

# Chương 2: Dữ liệu và phương pháp đề xuất

## Mô tả tập dữ liệu

- **Tập dữ liệu:** Bao gồm thông tin bài đăng tuyển dụng từ trang web việc làm.
- **Mục đích:** Phân tích cấu trúc, đặc điểm để phát hiện bài đăng gian lận.
- **Tổng quan:** Dữ liệu có 17.880 bản ghi (rows) với 17.200 mẫu lớp đa số (không giả mạo) và 800 mẫu lớp thiểu số (giả mạo), với các cột (columns) chứa thông tin chi tiết về việc làm.

Mô tả dữ liệu:

- Job_id : id của công việc
- title : tiêu đề của bài báo tuyển dụng
- location : địa chỉ của công ty - nơi làm việc
- department : vị trí công việc
- salary_range : Khoảng lương
- company_profile : profile của công ty
- description : Mô tả
- requirements : yêu cầu của công việc
- benefits : quyền lợi
- telecommuting : có để số liên lạc hay không
- has_company_logo : công ty có logo hay không
- has_questions : có câu hỏi hay không
- employment_type : kiểu công việc (full-time, part-time, other)
- required_experience : yêu cầu kinh nghiệm
- required_education : yêu cầu học vị
- industry : lĩnh vực công việc
- function : vai trò công việc
- fraudulent : gian lận hay không

# Chương 2: Dữ liệu và phương pháp đề xuất

## Tiền xử lí dữ liệu (Data Preprocessing)

**Xử Lý Dữ Liệu Mất Cân Bằng (Imbalanced Data Handling)**

Vấn đề: Lớp thiểu số (gian lận) quá ít dẫn đến overfitting/underfitting.

Hai hướng xử lý:

**Hướng 1: Oversampling bằng SMOTE**

Tạo mẫu tổng hợp cho lớp thiểu số (Synthetic Minority Over-sampling Technique).

Tăng từ 800 lên ~16.400 mẫu gian lận.

Tỷ lệ mới: 1:1.1, tổng mẫu ~34.400.

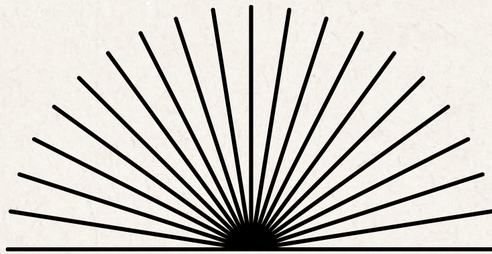
**Hướng 2: Kết hợp SMOTE và Undersampling**

SMOTE: Tăng lớp thiểu số lên ~4.000 (tăng ~5 lần).

Undersampling: Giảm lớp đa số xuống ~4.800-5.000 để tránh overfitting.

Tổng mẫu: ~8.800-9.000.

**Kết quả:** Cân bằng dữ liệu, giảm bias mô hình.



# Chương 2: Dữ liệu và phương pháp đề xuất

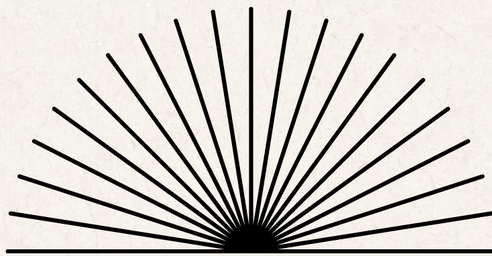
## Xử Lý Giá Trị Thiếu (Missing Values Imputation)

- Các giá trị trống trong dữ liệu được xử lý theo từng nhóm kiểu dữ liệu như sau:
- **Cột số (numerical features)**: Thay thế bằng giá trị trung vị (median) của cột để giảm ảnh hưởng của các giá trị ngoại lai.
- **Cột phân loại dạng chữ (categorical string features)**: Thay thế bằng chuỗi "unknown".
- **Cột văn bản (textual features)**: Thay thế bằng chuỗi "missing".

# Chương 2: Dữ liệu và phương pháp đề xuất

## Chỉnh Sửa Và Kỹ Thuật Đặc Trưng (Feature Engineering)

- **Loại bỏ cột**:
  - Job_id: Không mang thông tin dự đoán (tính duy nhất quá cao, không hữu ích cho mô hình).
- **Thêm cột mới**:
  - Keyword: Giá trị 0/1 dựa trên quy tắc domain knowledge (có/không chứa từ khóa liên quan đến gian lận).
  - Chain: Giá trị 0/1 dựa trên chuỗi liên quan trong tiêu đề/mô tả (dựa trên kiến thức chuyên môn).
- **Gộp đặc trưng văn bản**:
  - Kết hợp tất cả cột dạng chữ (title, description, requirements, v.v.) thành một cột duy nhất: **combined_text**.
  - Lợi ích: Tập trung thông tin văn bản, giảm chiều dữ liệu đầu vào, dễ xử lý hơn cho mô hình.
- **Lợi Ích Tổng Thể**
- Tăng độ chính xác dự đoán gian lận.
- Cải thiện hiệu suất mô hình bằng cách làm dữ liệu sạch hơn, thông tin hơn.



# Chương 2: Dữ liệu và phương pháp đề xuất

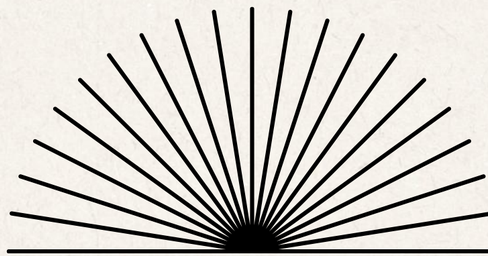
## Chuẩn Hóa Và Vector Hóa Dữ Liệu (Data Normalization & Text Vectorization)

### Xử lý đặc trưng

- **Đặc trưng số:** Giữ nguyên (do chủ yếu là hệ nhị phân 0/1).
- **Đặc trưng văn bản:** Làm sạch (loại bỏ ký tự đặc biệt, stop-words) & Chuẩn hóa (viết thường, lemmatization/stemming).

### Kỹ thuật Vector hóa (3 Pipeline riêng biệt)

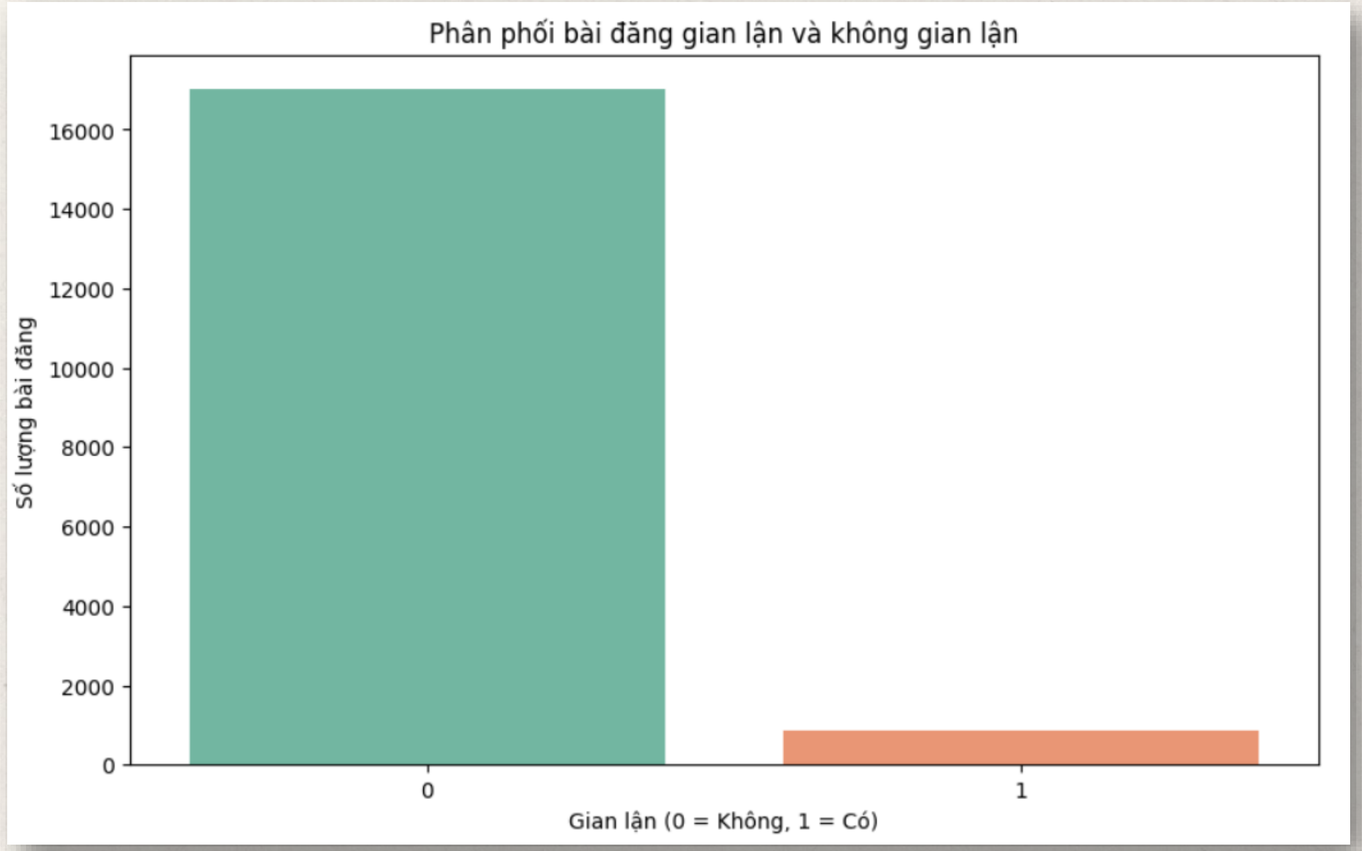
- **CountVectorizer:** Mô hình túi từ (Bag-of-Words).
- **TF-IDF Vectorizer:** Đánh trọng số tần suất từ và nghịch đảo văn bản.
- **Word2Vec:** Sử dụng Word Embeddings (trung bình cộng vector cho mỗi văn bản).



# Chương 2: Dữ liệu và phương pháp đề xuất

## Phân tích dữ liệu thăm dò

- Sử dụng biểu đồ để so sánh tỷ lệ bài đăng quy mô và không gian lận theo từng thuộc tính trong quá trình phân tích dữ liệu thăm dò.
- Dựa và hình bên, ta có thể thấy dữ liệu không cân bằng.
- Tỷ lệ đăng không gian lận cao hơn nhiều lần so với bài đăng gian lận.
- Điều này phản ánh thực tế trong nền tảng mạng xã hội, nơi hầu hết bài đăng hợp pháp.

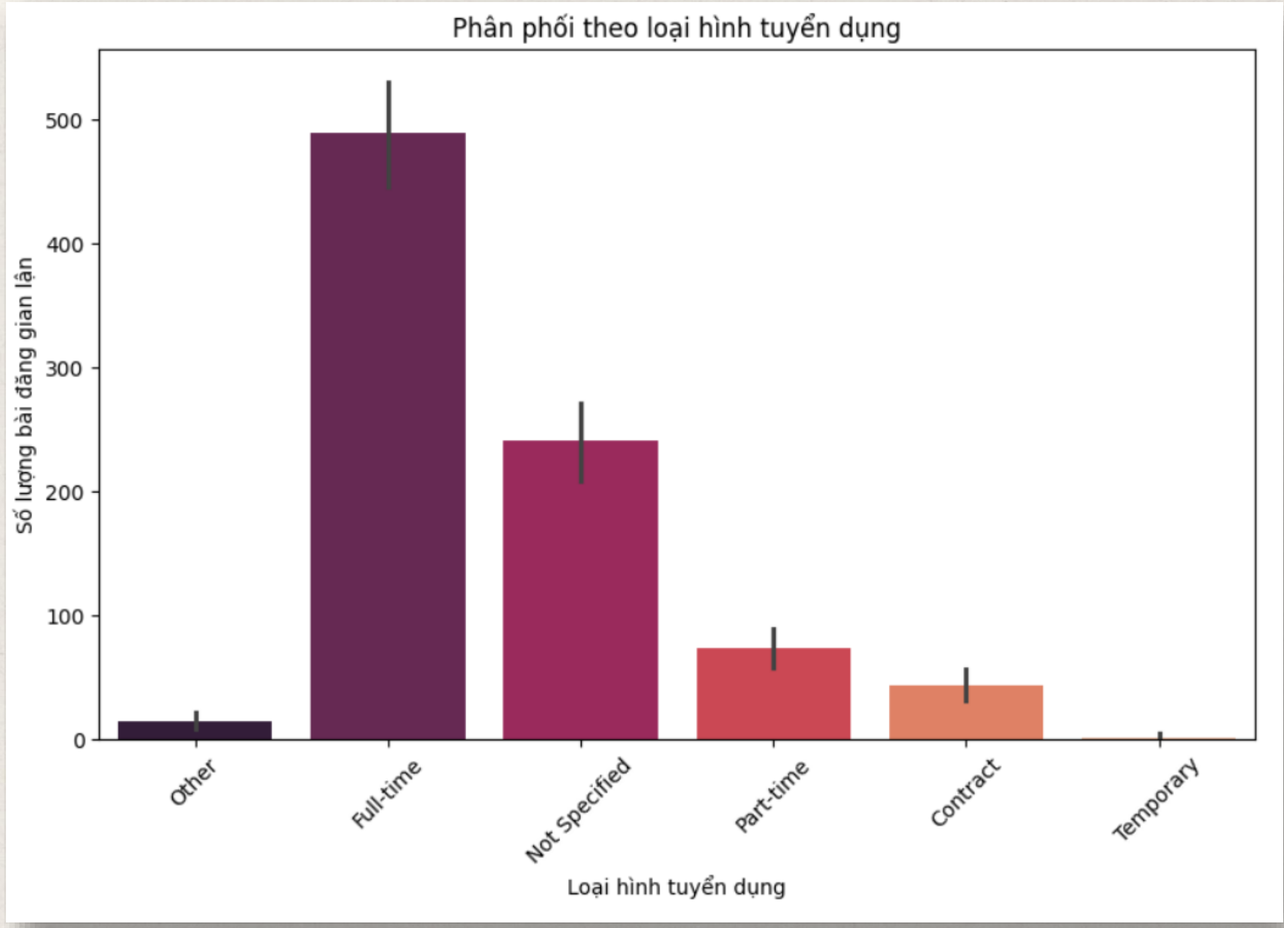


# Chương 2: Dữ liệu và phương pháp đề xuất

## Phân tích dữ liệu thăm dò

- Gian lận tập trung chủ yếu vào việc làm toàn thời gian (Full-time), có lẽ vì loại hình này hấp dẫn hơn với ứng viên, dễ lừa đảo (ví dụ: hứa hẹn lương cao, ổn định).

- "Not Specified" cũng phổ biến, cho thấy bài đăng mơ hồ dễ bị lợi dụng để gian lận.Rủi ro thấp: Các loại tạm thời (Temporary, Contract) hoặc khác (Other) ít bị ảnh hưởng, có thể do ít ứng viên quan tâm.

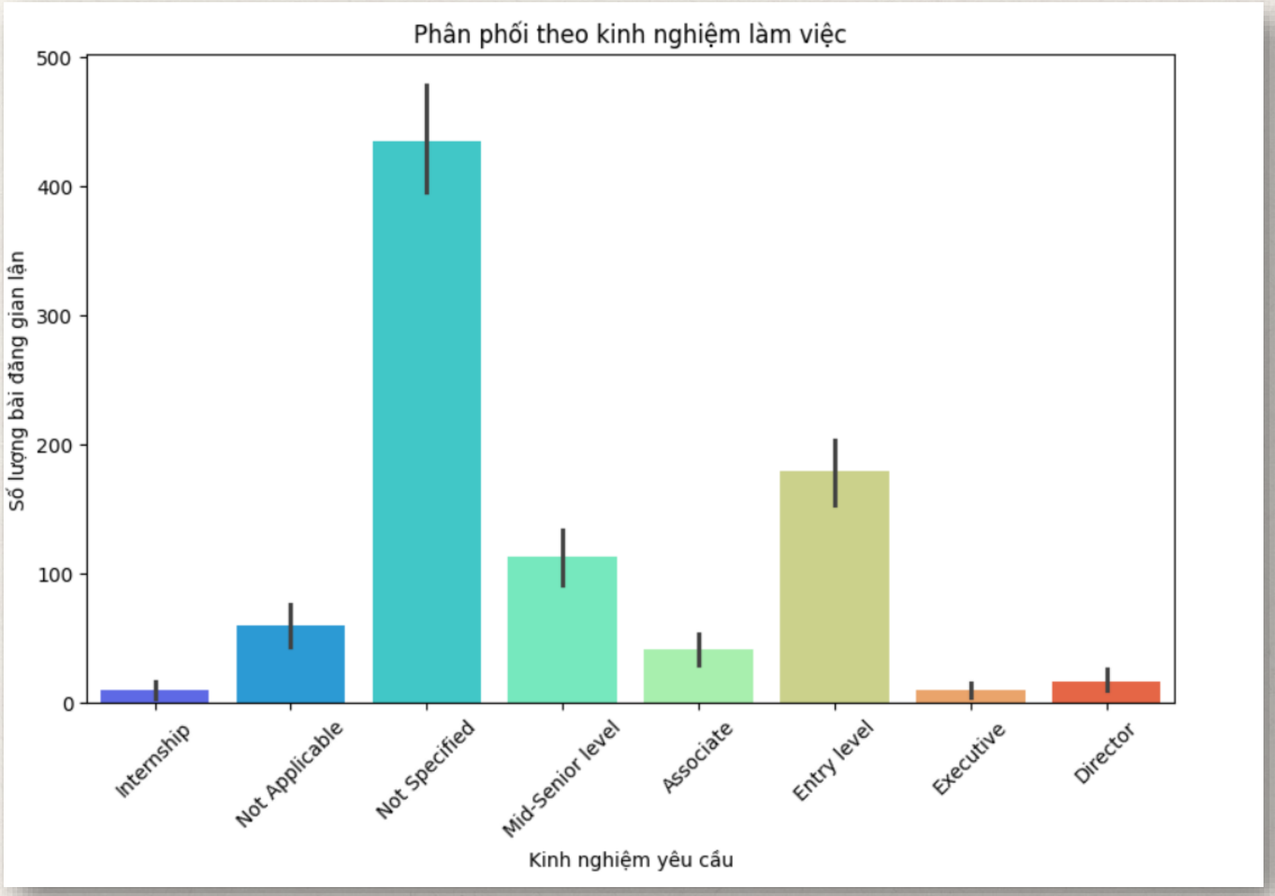


# Chương 2: Dữ liệu và phương pháp đề xuất

## Phân tích dữ liệu thăm dò

- Cột "Not Specified" cao áp đảo (hơn 400 bài), gấp đôi nhóm đứng thứ hai cho thấy kẻ lừa đảo thường cố tình để yêu cầu mơ hồ nhằm mở rộng "tệp nạn nhân".

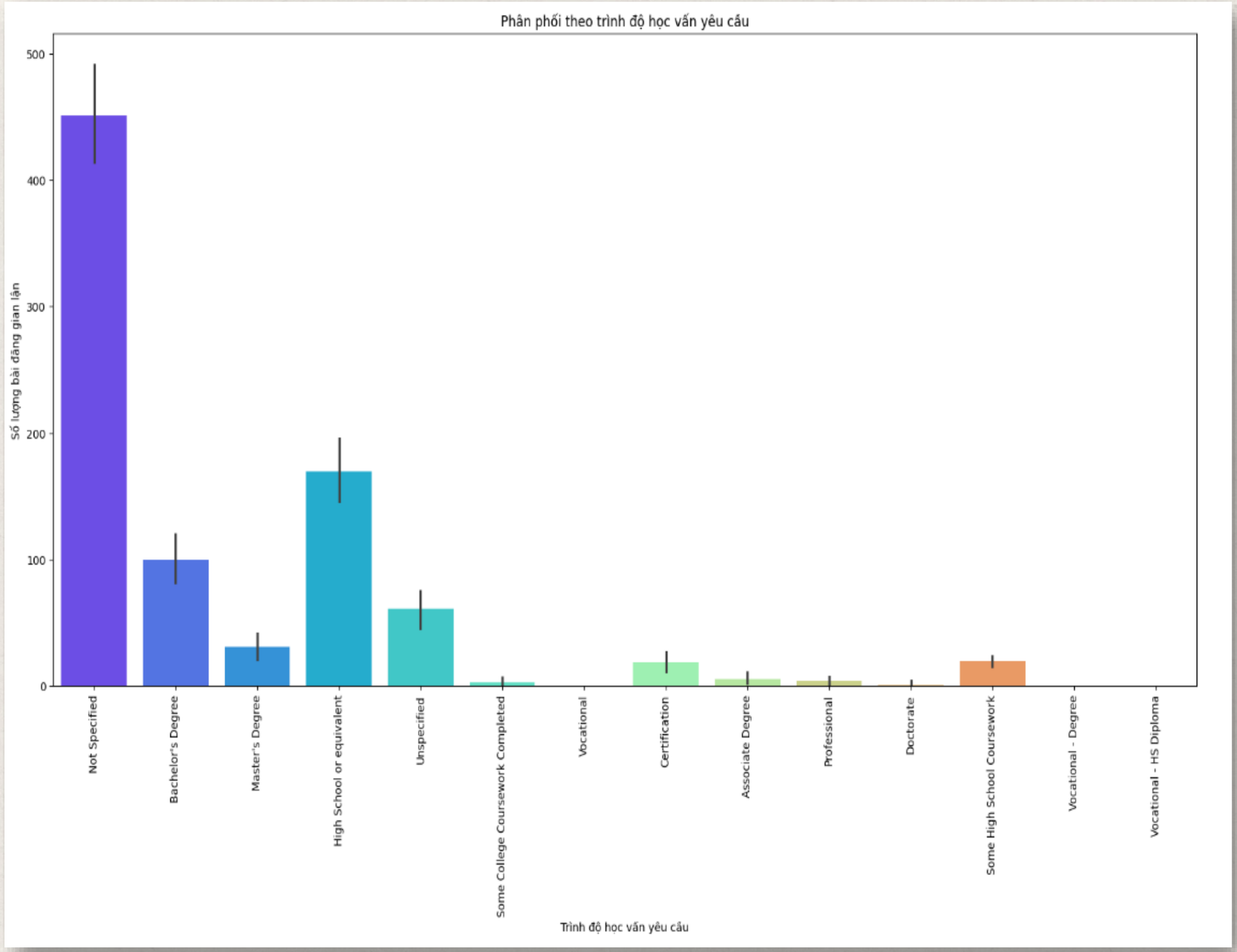
- Ta có thể thấy vị trí càng cao (Executive, Director), tỉ lệ lừa đảo càng thấp cho thấy kẻ gian ưu tiên tiếp cận đối tượng phổ thông, thiếu kinh nghiệm để dễ dàng "sập bẫy".



# Chương 2: Dữ liệu và phương pháp đề xuất

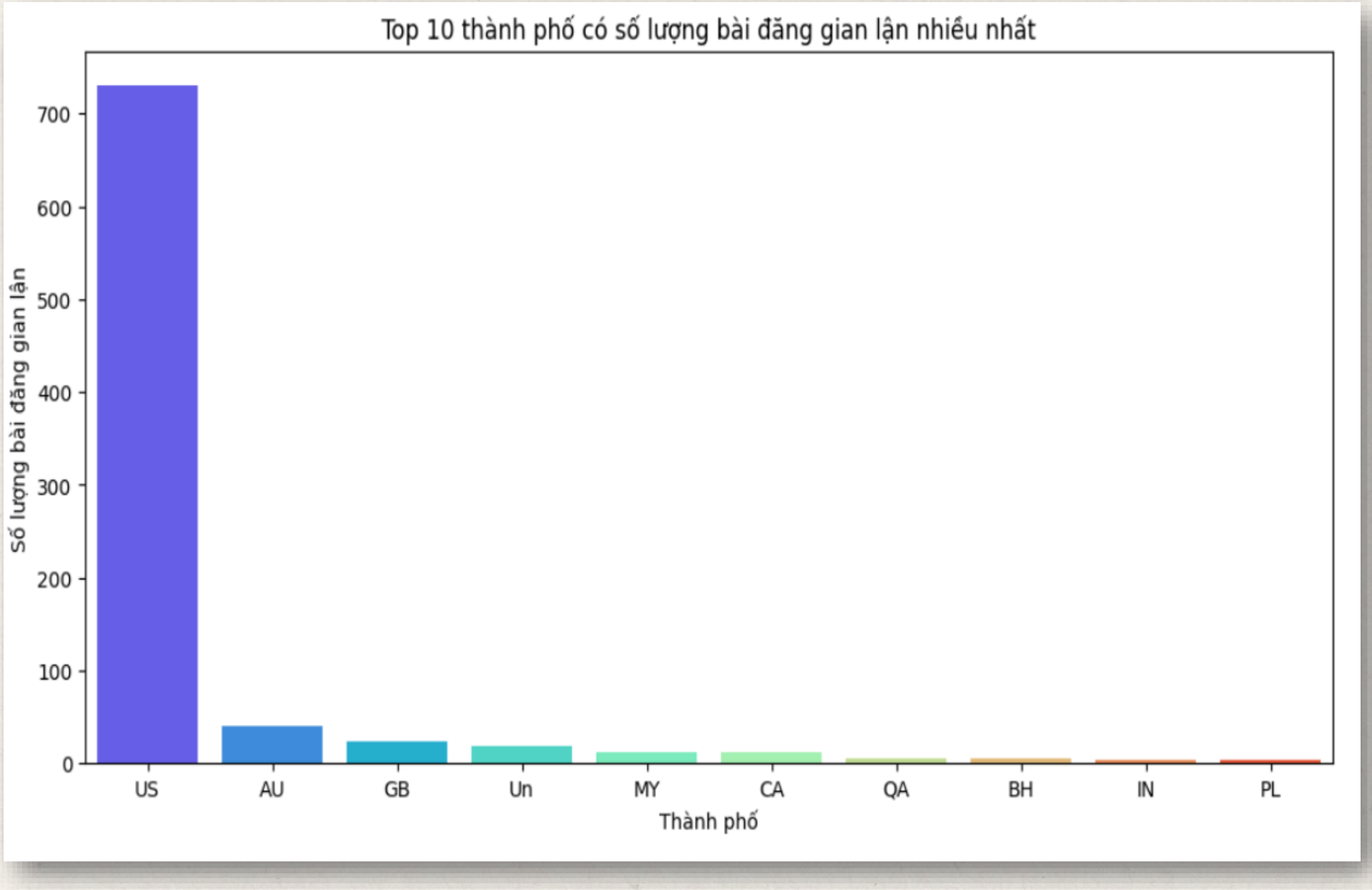
## Phân tích dữ liệu thăm dò

- Nhóm "Not Specified" (Không xác định) đứng đầu với hơn 450 bài đăng gian lận.
- Đối tượng mục tiêu tập trung mạnh vào người có bằng Cấp 3 (High School) và Cử nhân (Bachelor).
- Học vấn yêu cầu càng phổ thông hoặc mập mờ, nguy cơ lừa đảo càng lớn.
- Nhóm an toàn: Các trình độ chuyên sâu (Thạc sĩ, Tiến sĩ) gần như không xuất hiện dấu hiệu gian lận.



# Chương 2: Dữ liệu và phương pháp đề xuất

## Phân tích dữ liệu thăm dò




**Thị trường trọng điểm:**

- Mỹ (US) áp đảo hoàn toàn với hơn 700 bài đăng gian lận, bỏ xa các quốc gia còn lại.

**Khu vực tiếp theo:**

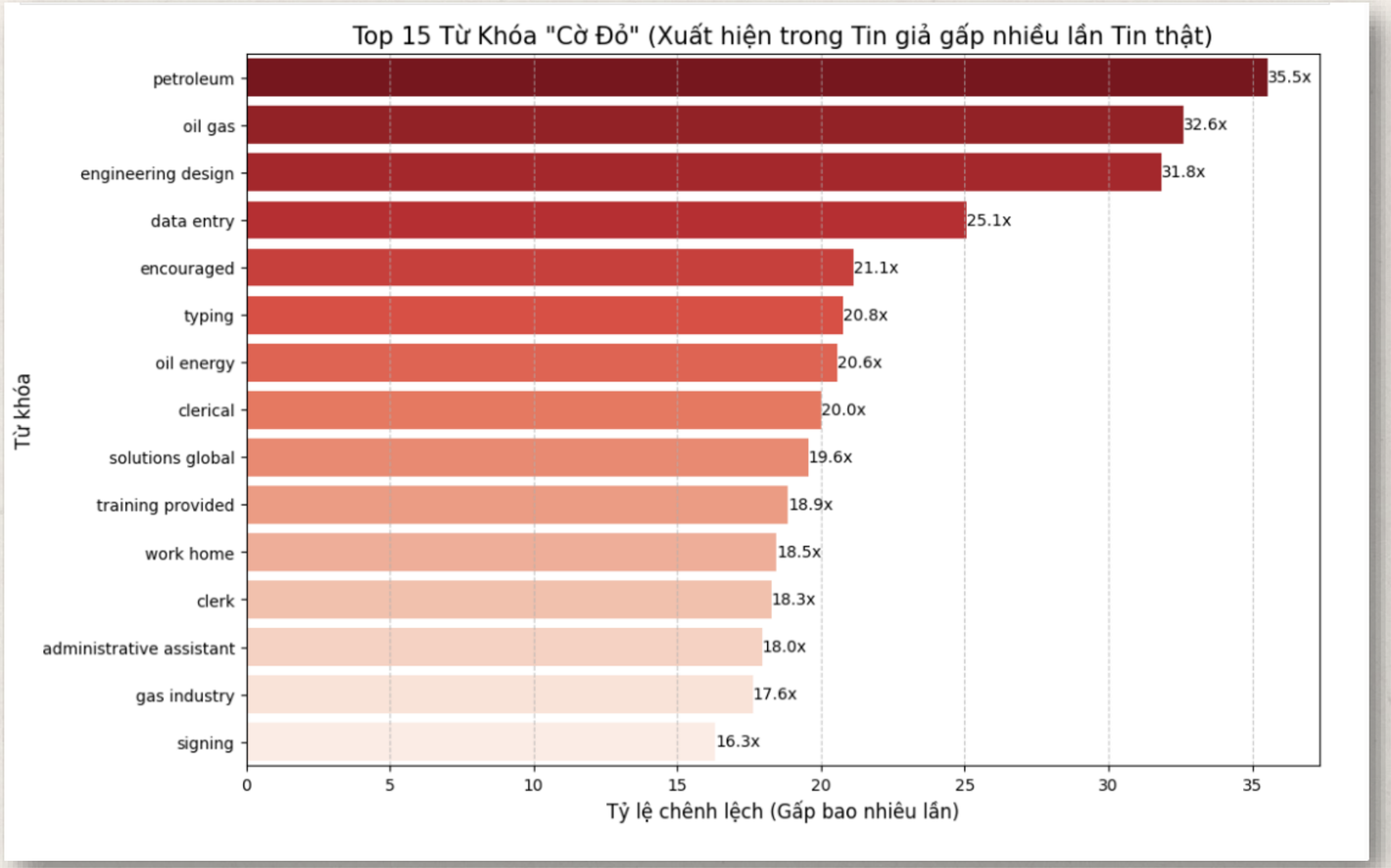
- Úc (AU) và Vương quốc Anh (GB) đứng vị trí thứ 2 và 3 nhưng với số lượng thấp hơn đáng kể (dưới 100 bài).
- Gian lận tuyển dụng trực tuyến tập trung cực kỳ mạnh mẽ tại thị trường Mỹ.

**Xu hướng toàn cầu:**

Các quốc gia khác (MY, CA, QA...) có số lượng bài đăng gian lận thấp hơn nhiều so với US.

# Chương 2: Dữ liệu và phương pháp đề xuất

## Phân tích dữ liệu thăm dò




**Nhóm ngành rủi ro cao nhất:**

- Các từ khóa liên quan đến Dầu khí & Năng lượng chiếm tỷ lệ áp đảo (Petroleum, Oil gas, Oil energy, Gas industry).
- Petroleum đứng đầu với tỷ lệ xuất hiện trong tin giả gấp 35.5 lần tin thật.

**Dấu hiệu lừa đảo tuyển dụng:**

- Các công việc đòi hỏi kỹ năng thấp nhưng mô tả hấp dẫn thường là "bẫy" tin giả:Data entry (gấp 25.1 lần), Typing (gấp 20.8 lần), Clerical/Clerk (~20 lần).

**Các cụm từ hứa hẹn:**

- Work home (gấp 18.5 lần), Training provided (gấp 18.9 lần).

**Đặc điểm ngôn ngữ:**

- Tin giả thường sử dụng các từ khóa mang tính cam kết hoặc mời gọi như Signing (16.3x) và Encouraged (21.1x).

# Chương 2: Dữ liệu và phương pháp đề xuất

**Từ những kết quả trên ta đã có thể trả lời được câu hỏi nghiên cứu là Các cột và keyword nào ảnh hưởng mạnh đến khả năng dự đoán của mô hình**

- **Dữ liệu dạng số:** Cột has_company có độ tương quan cao nhất với khả năng dự báo lừa đảo; ngược lại, job_id làm giảm độ chính xác của mô hình.
- **Dữ liệu dạng phân loại (Categorical):** Tất cả các cột thuộc nhóm này đều hỗ trợ tốt cho việc phân loại dữ liệu.
- **Dữ liệu dạng văn bản (Text):** * Xác định và giữ lại 50 stopWords có ý nghĩa quan trọng để tránh làm mất thông tin mô hình.
  - **Từ khóa (Keynote) tiêu biểu:** Petroleum, Oil gas, Data entry, Typing, Work home, Training provided... ảnh hưởng mạnh nhất đến kết quả dự đoán.
- **Giải pháp tối ưu:** Tạo thêm cột mới dựa trên sự xuất hiện của các từ khóa quan trọng (gán giá trị 1) để tăng mạnh khả năng dự đoán của mô hình.

# Chương 2: Dữ liệu và phương pháp đề xuất

## Phương pháp đề xuất

**Quy trình nghiên cứu (4 giai đoạn):**

- **Giai đoạn 1:** Thu thập dữ liệu và phân tích dữ liệu thăm dò (EDA)
- **Giai đoạn 2:** Tiền xử lý và làm sạch văn bản.
- **Giai đoạn 3:** Trích xuất đặc trưng (Feature Extraction).
- **Giai đoạn 4:** Huấn luyện và đánh giá mô hình.

---

**Mô hình huấn luyện:**

- **Mục tiêu:** Phân loại tin giả (Fake News Detection).
- **Tiếp cận:** Sử dụng 4 nhóm thuật toán đại diện gồm Xác suất, Tuyến tính, Bagging và Boosting để tìm mô hình tối ưu.

# Chương 2: Dữ liệu và phương pháp đề xuất

## Phương pháp đề xuất

### Naive Bayes

**Cơ sở:**

- Thuật toán nền tảng (baseline) cho NLP, dựa trên định lý Bayes với giả định các từ độc lập.

**Ưu điểm:**

- Tốc độ cực nhanh: Huấn luyện và dự đoán ít tốn tài nguyên.
- Phù hợp văn bản: Hiệu quả với dữ liệu số chiều lớn (TF-IDF, BoW) ngay cả khi tập dữ liệu nhỏ.

### Logistic Regression

**Cơ sở:**

- Thuật toán phân loại tuyến tính mạnh mẽ. Sử dụng hàm Sigmoid để ước tính xác suất (Tin thật/giả).

**Ưu điểm:**

- Dễ giải thích: Dựa vào trọng số (weights) để xác định các từ khóa quan trọng (như "breaking news").
- Ít Overfitting: Kiểm soát tốt hiện tượng quá khớp nhờ các kỹ thuật điều chuẩn (L1/L2).

### Random Forest

**Cơ sở:**

- Phương pháp học kết hợp (Ensemble Learning - Bagging) dựa trên hàng trăm cây quyết định và cơ chế bỏ phiếu.

**Ưu điểm:**

- **Tính ổn định cao:** Giảm phương sai (Variance), hạn chế lỗi của cây quyết định đơn lẻ.
- **Xử lý phi tuyến:** Nắm bắt tốt các mối quan hệ phức tạp giữa các cụm từ trong văn bản.

### XGBoost

**Cơ sở:**

- Thuật toán thuộc nhóm Boosting tiên tiến, tối ưu hóa hàm mất mát và hỗ trợ tính toán song song.

**Ưu điểm:**

- **Hiệu suất vượt trội**: Tốc độ xử lý cực nhanh nhờ tối ưu hóa phần cứng.
- **Xử lý dữ liệu nhiễu**: Hoạt động mạnh mẽ với dữ liệu thiếu (Missing values) hoặc mất cân bằng lớp (Imbalanced Data).

# Chương 3: Thực nghiệm, kết quả và thảo luận

## Thiết lập thực nghiệm

```
import pandas as pd
import numpy as np
import joblib
import json      You, 6 days ago • update func for prepro and train
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.naive_bayes import MultinomialNB
import optuna
from optuna.distributions import FloatDistribution, IntDistribution, CategoricalDistribution
from optuna_integration import OptunaSearchCV
import os
import sys
import traceback
```

*Các thư viện cần nạp để huấn luyện mô hình*

```
job_id                   0
title                    0
location                 0
department               0
salary_range             0
company_profile          0
description              0
requirements             0
benefits                 0
telecommuting            0
has_company_logo         0
has_questions            0
employment_type          0
required_experience      0
required_education       0
industry                 0
function                 0
fraudulent               0
dtype: int64
```

*Kết quả sau khi tiền xử lí*

# Chương 3: Thực nghiệm, kết quả và thảo luận

## Kết quả huấn luyện

### Kết Quả Naive Bayes

```
--- ĐÁNH GIÁ: NB (Threshold=0.5) ---
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      3403
           1       0.92      0.57      0.70       173

    accuracy                           0.98      3576
   macro avg       0.95      0.78      0.84      3576
weighted avg       0.98      0.98      0.97      3576

ROC-AUC: 0.9774
PR-AUC (AUPRC): 0.8152 (Quan trọng cho Fraud)
```

**Naive Bayes**: Có độ chính xác rất cao khi dự báo gian lận (Precision 0.92) nhưng lại bỏ sót khá nhiều trường hợp thực tế khi chỉ nhận diện được hơn một nửa số ca vi phạm (Recall 0.57).

### Kết Quả Random Forest

```
--- ĐÁNH GIÁ: RF (Threshold=0.5) ---
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      3403
           1       0.90      0.70      0.79       173

    accuracy                           0.98      3576
   macro avg       0.94      0.85      0.89      3576
weighted avg       0.98      0.98      0.98      3576

ROC-AUC: 0.9789
PR-AUC (AUPRC): 0.8619 (Quan trọng cho Fraud)
```

**Random Forest:** Đạt được sự cân bằng khá tốt giữa việc nhận diện đúng gian lận và hạn chế sai sót với F1-score là 0.79, đi kèm chỉ số AUPRC ở mức khá (0.8619) cho bài toán Fraud.

# Chương 3: Thực nghiệm, kết quả và thảo luận

## Kết quả huấn luyện

### Kết Quả XGB

```
--- ĐÁNH GIÁ: XGB (Threshold=0.5) ---
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      3403
           1       0.91      0.85      0.88       173

    accuracy                           0.99      3576
   macro avg       0.95      0.92      0.94      3576
weighted avg       0.99      0.99      0.99      3576

ROC-AUC: 0.9928
PR-AUC (AUPRC): 0.9446 (Quan trọng cho Fraud)
```

**XGBoost:** Hiệu suất cực kỳ ấn tượng với độ chính xác cao (Precision 0.91) cho lớp gian lận, giúp giảm thiểu tối đa các cảnh báo giả trong khi vẫn duy trì chỉ số AUPRC xuất sắc (0.9446).

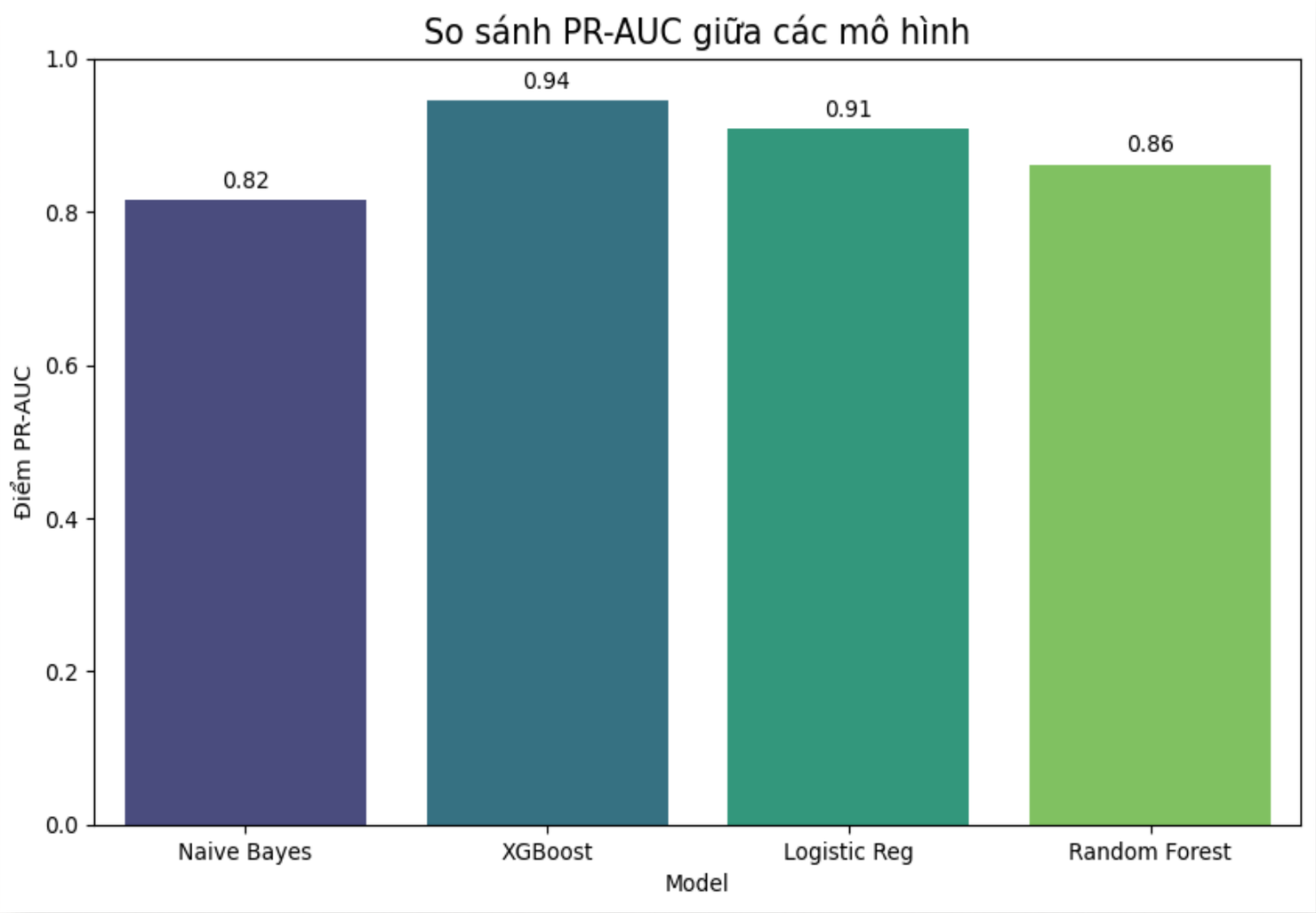
### Kết Quả Logistics Regression

```
--- ĐÁNH GIÁ: LR (Threshold=0.5) ---
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      3403
           1       0.67      0.90      0.77       173

    accuracy                           0.97      3576
   macro avg       0.83      0.94      0.88      3576
weighted avg       0.98      0.97      0.98      3576

ROC-AUC: 0.9900
PR-AUC (AUPRC): 0.9086 (Quan trọng cho Fraud)
```

**Logistic Regression:** Khả năng bắt lỗi rất tốt (Recall 0.90), tuy nhiên tỷ lệ dương tính giả còn cao (Precision chỉ 0.67), khiến mô hình này phù hợp hơn nếu ưu tiên không bỏ sót gian lận bất kể sai số.

# Chương 3: Thực nghiệm, kết quả và thảo luận

## Đánh giá và so sánh




**Kết quả biểu đồ cho thấy:**

- **XGBoost** đạt hiệu suất cao nhất với PR-AUC = 0.94
- **Logistic Regression** đứng thứ hai với PR-AUC = 0.91
- **Random Forest** đạt PR-AUC = 0.86
- **Naive Bayes** có hiệu suất thấp nhất với PR-AUC = 0.82

# Chương 4: Kết luận và trả lời câu hỏi nghiên cứu

## Trả lời câu hỏi nghiên cứu

### Câu 1.Thuật toán máy học nào cho hiệu suất cao nhất trong việc phân loại tin tuyển dụng?

- Thuật toán XGBoost Classifier cho thấy hiệu suất cao nhất trong bài toán phân loại tin tuyển dụng vì khả năng xử lý hiệu quả dữ liệu không cân bằng, học được các mối quan hệ phi tuyến phức tạp giữa các đặc trưng và tích hợp cơ chế regularization mạnh mẽ giúp giảm hiện tượng overfitting.

---

### Câu 2.Làm thế nào để cân bằng giữa chỉ số precision và recall?

- Sử dụng F1-Score: Là trung bình điều hòa (harmonic mean) giữa Precision và Recall, dùng để tìm điểm cân bằng giữa hai chỉ số này.
- Công thức: $F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- Điều chỉnh Threshold: Thay đổi ngưỡng phân loại từ 0 đến 1 để tối ưu hóa diện tích dưới đường cong Precision-Recall (PR-AUC).

# Chương 4: Kết luận và trả lời câu hỏi nghiên cứu

## Trả lời câu hỏi nghiên cứu

**Câu 3.Việc áp dụng các kỹ thuật tái lấy mẫu như smote thì liệu có ảnh hưởng như thế nào đến khả năng phát hiện tin giả của mô hình?**

- Rủi ro từ SMOTE: Việc lạm dụng kỹ thuật này dễ gây ra hiện tượng Overfitting do mô hình phải học từ quá nhiều dữ liệu nhân tạo.
- Giải pháp thay thế: Đối với mô hình XGBoost, có thể sử dụng cơ chế scale_pos_weight để cân bằng các lớp trực tiếp trong hàm mất mát (loss function).
- Ưu điểm: Giúp giữ nguyên tính trung thực của dữ liệu gốc, tránh sinh dữ liệu giả và hạn chế tối đa nhiễu so với việc dùng SMOTE.

---

**Câu 4.Những đặc trưng chủ yếu về văn bản hoặc từ khóa (keywords) nào mang tính phân biệt cao nhất giúp nhận diện một tin tuyển dụng giả mạo.**

- petroleum, oil gas,oil energy, data entry, typing, clerical, work home,training provided, encouraged, administrative assitantm clerk.

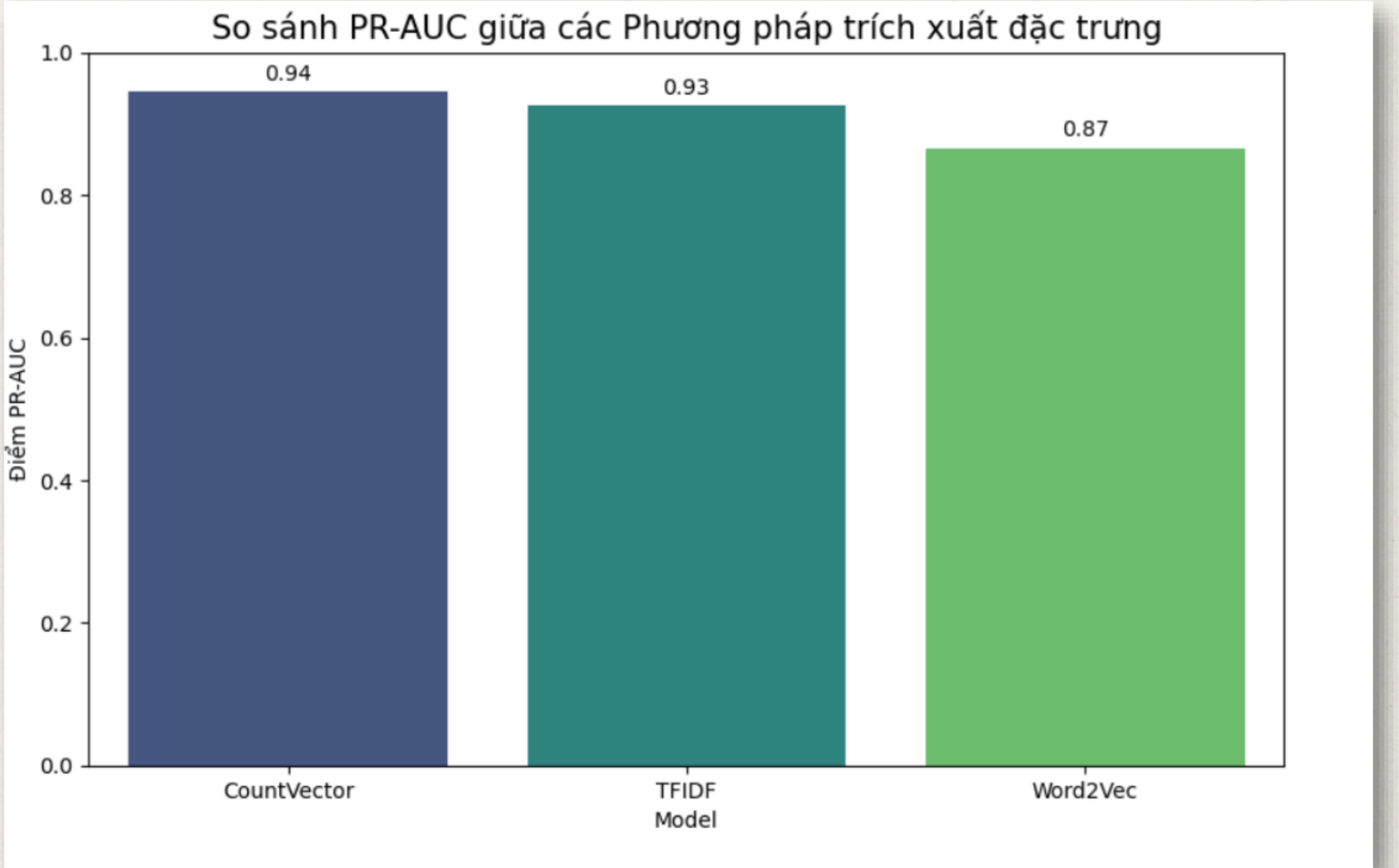
# Chương 4: Kết luận và trả lời câu hỏi nghiên cứu

## Trả lời câu hỏi nghiên cứu

**Câu 5.Phương pháp trích xuất đặc trưng nào (TF-IDF, Count Vectorized, Word Embeddings như Word2Vec) mang lại đầu vào tốt nhất cho mô hình phân loại trong ngữ cảnh này?**

- Count Vector

Chứng minh:



# Thank you