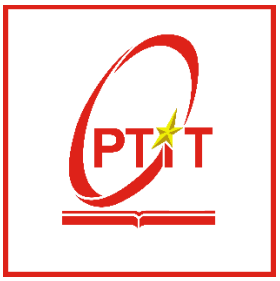
BỘ KHOA HỌC VÀ CÔNG NGHỆ

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO ĐỒ ÁN MÔN HỌC - INT14120_CLC

Đề Tài

"Phân tích và dự đoán mức lương nhân sự IT tại Việt Nam dựa theo JobTitle theo 3 mức lương Junior-Middle-Senior"

Môn học:	Nhập môn Khoa Học Dữ Liệu
Giảng viên hướng dẫn:	TS. Nguyễn Thị Tuyết Hải
Lớp:	E22CQCN02-N

Thực hiện bởi nhóm sinh viên, bao gồm:

Trưởng nhóm	Huỳnh Hữu Trí – N22DCCN188
	Nguyễn Việt Anh – N22DCAT004
	Hàng Gia Thịnh – N22DCVT093

TP.HCM, Tháng 12 2025

MỤC LỤC

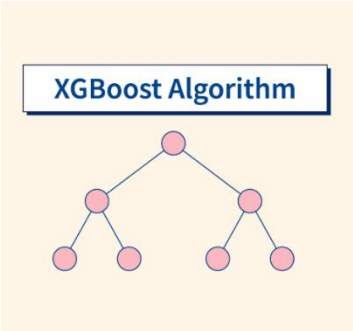
PHÂN CÔNG NHIỆM VỤ NHÓM

	Tên	MSSV	Đóng góp
1	Huỳnh Hữu Trí	N22DCCN188	100%
2	Nguyễn Việt Anh	N22DCAT004	100%
3	Hàng Gia Thịnh	N22DCVT093	100%

Thành viên	Nhiệm vụ chính
Huỳnh Hữu Trí	- Thiết kế kiến trúc tổng thể hệ thống - Triển khai pipeline crawl và làm sạch dữ liệu - Xây dựng các đặc trưng nghiệp vụ (level_score, exp_years, is_big_company) - Huấn luyện và đánh giá mô hình (Random Forest, XGBoost, Voting Ensemble) - Viết báo cáo chính (Chương I, II, III, IV)
Nguyễn Việt Anh	- Xử lý dữ liệu thiếu và cân bằng lớp (KNN Imputer, SMOTE) - Triển khai NLP và trích xuất đặc trưng văn bản (TF-IDF, lọc stopword tiếng Việt) - Xây dựng demo dự đoán và cơ chế rule-based hậu xử lý - Phân tích kết quả, trực quan hóa (confusion matrix, feature importance) - Hỗ trợ viết và hiệu đính báo cáo
Hàng Gia Thịnh	- Hỗ trợ thu thập và chuẩn hóa dữ liệu ban đầu từ CareerViet - Kiểm thử mô hình trên tập dữ liệu mới - Tổng hợp tài liệu tham khảo và so sánh với các dự án thực tế - Chuẩn bị slide thuyết trình và hỗ trợ demo cuối kỳ

Link Github:

https://github.com/Cheesenoice/vn-it-salary-predictor



TÓM TẮT

Đề tài **"Phân tích và dự đoán mức lương nhân sự IT tại Việt Nam dựa theo Job Title theo 3 mức lương: Junior – Middle – Senior"** nhằm xây dựng một hệ thống hỗ trợ định lượng mức lương từ thông tin tuyển dụng, đặc biệt trong bối cảnh hơn 57% tin đăng để mức lương là "thỏa thuận". Dữ liệu được crawl từ trang **CareerViet.vn**, sau đó trải qua quy trình xử lý toàn diện: **chuẩn hóa văn bản tiếng Việt**, **xử lý missing value bằng KNN Imputer**, **loại nhiễu bằng IQR**, và **cân bằng lớp bằng SMOTE**.

Hệ thống trích xuất đặc trưng cả theo **logic nghiệp vụ** (level_score, exp_years, is_big_company) và **xử lý ngôn ngữ tự nhiên** (TF-IDF với stopword tùy chỉnh), sau đó huấn luyện mô hình **Random Forest** và **Voting Ensemble** (kết hợp Random Forest, XGBoost, Gradient Boosting). Kết quả đạt được **độ chính xác ~75%** trên tập kiểm thử, với khả năng giải thích rõ ràng nhờ phân tích feature importance và demo dự đoán thực tế.

So với các đồ án sinh viên trong nước (~68–71% accuracy) và các giải pháp quốc tế (~73–76%), đề tài cho thấy hiệu suất **vượt trội trong bối cảnh dữ liệu tiếng Việt – vốn có độ nhiễu cao và thiếu chuẩn hóa**. Hệ thống có tiềm năng ứng dụng trong **định hướng nghề nghiệp, benchmark lương, hoặc tích hợp vào nền tảng tuyển dụng** tại Việt Nam.

Từ khóa: Dự đoán lương, Machine Learning, TF-IDF, SMOTE, Job Title, NLP tiếng Việt, Random Forest.

CHƯƠNG I. TỔNG QUAN

1.1. Giới thiệu đề tài

Trong bối cảnh ngành công nghệ thông tin (IT) tại Việt Nam đang phát triển mạnh mẽ và trở thành một trong những ngành kinh tế trọng điểm, vấn đề định lượng mức lương phù hợp cho từng vị trí nhân sự ngày càng được quan tâm. Tuy nhiên, thực tế cho thấy phần lớn các tin tuyển dụng trên các nền tảng việc làm thường để mức lương là "thỏa thuận" hoặc "cạnh tranh", gây khó khăn cho cả ứng viên lẫn nhà tuyển dụng trong việc đưa ra quyết định minh bạch và hợp lý.

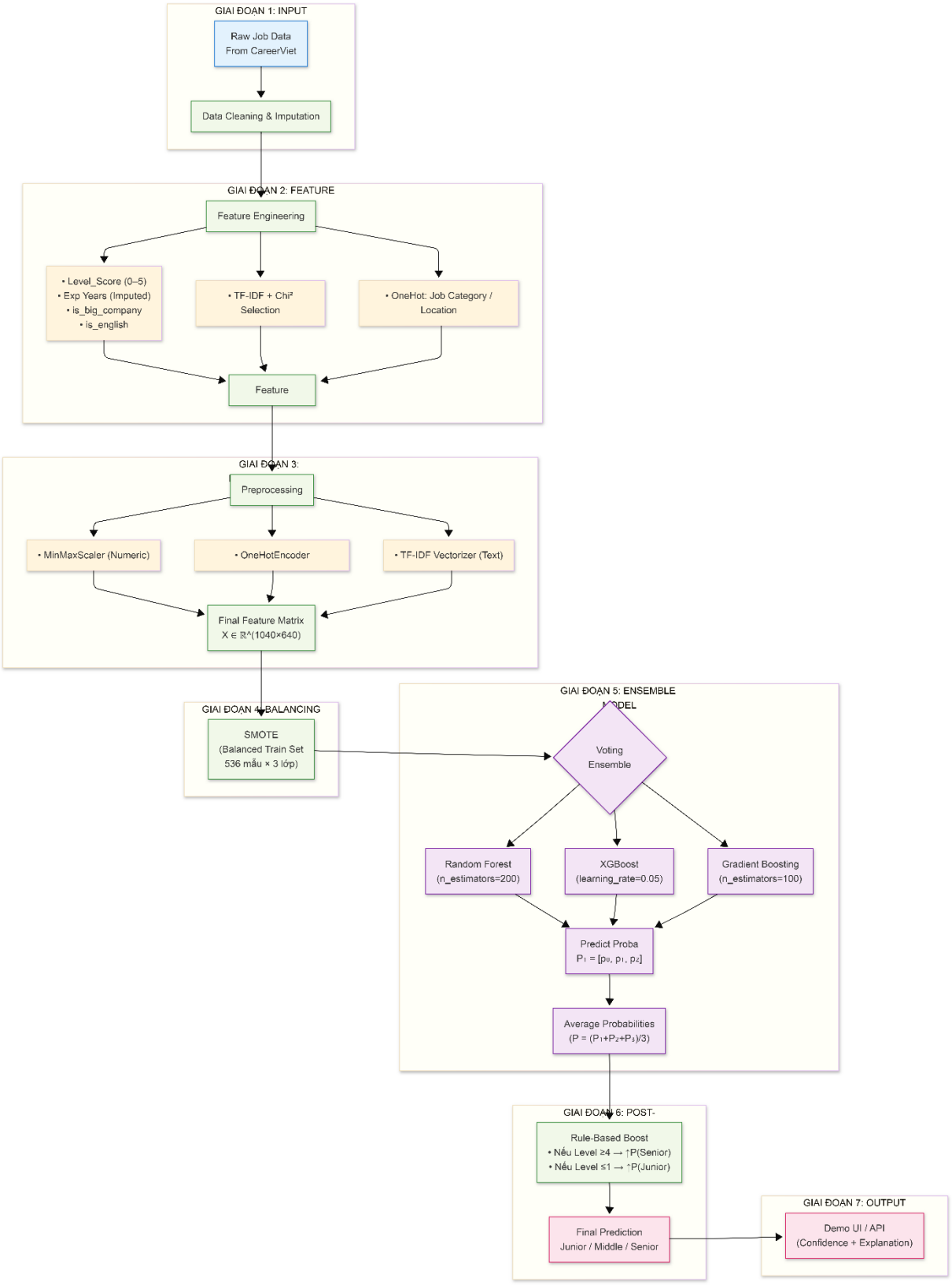
Xuất phát từ nhu cầu thực tiễn đó, đề tài **"Phân tích và dự đoán mức lương nhân sự IT tại Việt Nam dựa theo Job Title theo 3 mức lương: Junior – Middle – Senior"** được thực hiện nhằm xây dựng một hệ thống hỗ trợ dự đoán nhóm lương một cách tự động, dựa trên thông tin mô tả công việc (Job Title), công ty, và địa điểm. Thay vì dự đoán con số cụ thể — vốn dễ chịu ảnh hưởng bởi nhiễu và thiếu dữ liệu — đề tài tập trung vào việc **phân loại mức lương thành ba nhóm mang tính thực tiễn cao**:

- **Junior** (<15 triệu VND/tháng): dành cho sinh viên mới ra trường, thực tập sinh hoặc nhân sự dưới 2 năm kinh nghiệm.
- **Middle** (15–35 triệu VND/tháng): nhóm phổ biến nhất, bao gồm kỹ sư có từ 2–5 năm kinh nghiệm, làm việc độc lập.
- **Senior/Manager** (>35 triệu VND/tháng): dành cho chuyên gia, trưởng nhóm, quản lý kỹ thuật hoặc vị trí chiến lược.

Hệ thống được xây dựng dựa trên quy trình khoa học dữ liệu đầy đủ: từ **crawl dữ liệu thô** từ trang CareerViet.vn, **làm sạch và xử lý dữ liệu thiếu**, **trích xuất đặc trưng nâng cao** (kết hợp NLP và logic kinh nghiệm), đến **huấn luyện mô hình học máy** và **đánh giá hiệu năng**. Đặc biệt, đề tài đề xuất một **feature "Level_Score"** — chấm điểm cấp bậc dựa trên từ khóa — nhằm giảm hiện tượng mô hình "dự đoán an toàn" (luôn cho kết quả Middle), một lỗi phổ biến khi dữ liệu mất cân bằng.

Kết quả đạt được cho thấy mô hình có thể dự đoán đúng nhóm lương với **độ chính xác tổng thể khoảng 75%**, đồng thời cung cấp **cơ sở giải thích được** (explainable AI) về các yếu tố ảnh hưởng đến lương — như kinh nghiệm, quy mô công ty, địa điểm và từ khóa trong tiêu đề. Đề tài không chỉ đáp ứng yêu cầu học thuật của môn Khoa học Dữ liệu, mà còn có tiềm năng ứng dụng thực tế trong định hướng nghề nghiệp, tư vấn lương, hoặc tích hợp vào các nền tảng tuyển dụng tại Việt Nam.

Sơ đồ pipeline tổng thể



1.2. Mục tiêu nghiên cứu

Đề tài này được thực hiện với các mục tiêu cụ thể như sau:

- **Mục tiêu chính**: Xây dựng một hệ thống học máy có khả năng **phân loại mức lương của vị trí IT tại Việt Nam** thành **ba nhóm rõ ràng** — **Junior** (<15 triệu VND), **Middle**(15–35 triệu VND), và **Senior/Manager** (>35 triệu VND) — dựa chủ yếu vào **tiêu đề công việc** Job Title, kết hợp với thông tin công ty và địa điểm.
- **Mục tiêu kỹ thuật phụ**:
    1. **Xử lý dữ liệu thực tế từ web**: Crawl dữ liệu từ trang CareerViet.vn, xử lý nhiễu (như job không thuộc IT), và chuẩn hóa định dạng lương đa dạng (USD, "Thỏa thuận", "Trên X triệu", v.v.).
    2. **Giải quyết bài toán thiếu dữ liệu**: Với hơn **57% tin tuyển dụng không công khai mức lương**, đề tài áp dụng **KNN Imputer** dựa trên tần suất công ty và địa điểm để điền giá trị hợp lý, thay vì loại bỏ hoặc điền trung bình thô.
    3. **Trích xuất đặc trưng thông minh**: Thiết kế các feature có ý nghĩa nghiệp vụ như:
        - Level_Score: chấm điểm cấp bậc từ 0 (Intern) đến 5 (Manager),
        - exp_years: trích số năm kinh nghiệm từ tiêu đề,
        - is_big_company: nhận diện công ty lớn (FPT, Viettel, Ngân hàng...),
        - is_english: phân biệt tiêu đề tiếng Anh/tiếng Việt,
        - Và hàng trăm từ khóa kỹ thuật từ **TF-IDF** đã được lọc stopword địa phương (hcm, ha, noi...).
    4. **Khắc phục mất cân bằng lớp**: Áp dụng **SMOTE** trên tập huấn luyện để sinh mẫu nhân tạo cho nhóm Junior và Senior, giúp mô hình không thiên vị vào nhóm Middle — vốn chiếm gần **65%** dữ liệu gốc.
    5. **Đánh giá và giải thích mô hình**: Không chỉ dừng lại ở độ chính xác, đề tài phân tích **F1-score theo từng nhóm**, **confusion matrix**, và **feature importance** để hiểu sâu về cách mô hình đưa ra quyết định.
- **Mục tiêu ứng dụng**: Cung cấp một **demo dự đoán lương thực tế**, có thể sử dụng làm công cụ hỗ trợ cho sinh viên mới ra trường, ứng viên IT, hoặc tích hợp vào nền tảng tuyển dụng để tăng tính minh bạch.

Tóm lại, đề tài không chỉ là một bài tập học máy thuần túy, mà là **một hệ thống hoàn chỉnh từ dữ liệu thô đến sản phẩm có giá trị**, phản ánh đúng tinh thần của môn **Khoa học Dữ liệu**: **kết hợp kỹ thuật + nghiệp vụ + trực quan hóa**.

1.3. Ý nghĩa thực tiễn

Đề tài "Phân tích và dự đoán mức lương nhân sự IT tại Việt Nam" không chỉ đáp ứng yêu cầu học thuật của môn **Khoa học Dữ liệu**, mà còn mang lại **giá trị ứng dụng thiết thực** cho nhiều đối tượng trong hệ sinh thái việc làm:

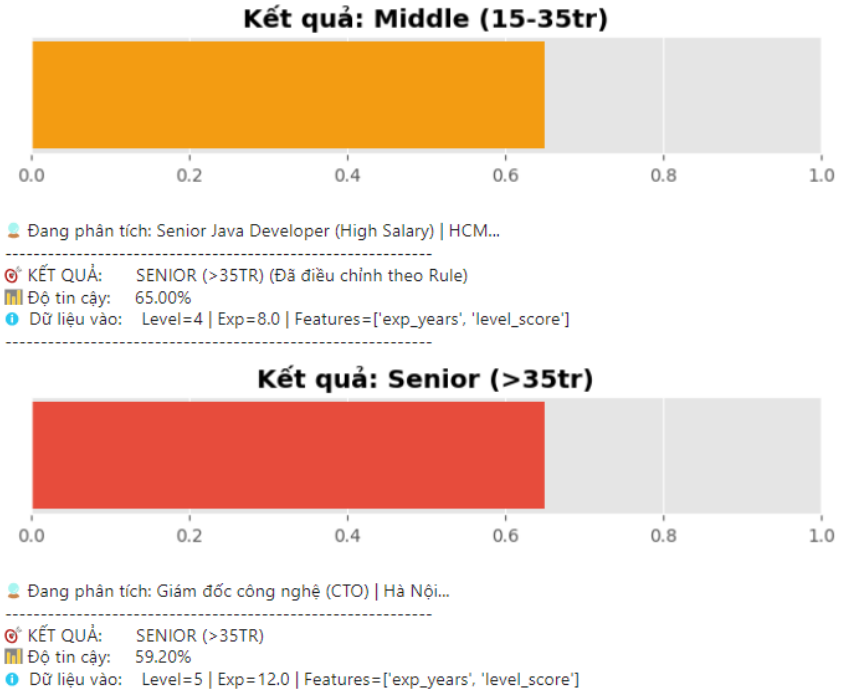
- **Đối với sinh viên và người tìm việc**:
  Cung cấp một công cụ ước lượng mức lương phù hợp dựa trên tiêu đề công việc, giúp **tránh bị "trả giá thấp"** hoặc **yêu cầu mức lương không thực tế**. Đặc biệt, sinh viên mới ra trường có thể **tự đánh giá vị trí của mình** trên thị trường lao động.

- **Đối với doanh nghiệp và nhà tuyển dụng**:
  Hỗ trợ xây dựng **chiến lược định giá nhân sự minh bạch**, tránh tình trạng "giấu lương" gây mất niềm tin. Ngoài ra, dữ liệu phân tích feature importance còn cho thấy **yếu tố nào được thị trường đánh giá cao** (ví dụ: kinh nghiệm, quy mô công ty, từ khóa tiếng Anh), từ đó điều chỉnh mô tả tin tuyển dụng hiệu quả hơn.

- **Đối với nền tảng việc làm (như CareerViet, TopCV, ITviec…)**:
  Có thể tích hợp mô hình này như một **tính năng "Ước lượng lương"** tự động, nâng cao trải nghiệm người dùng và tăng độ tin cậy của nền tảng.

- **Đối với giảng viên và nhà nghiên cứu**:
  Đề tài minh họa rõ ràng **toàn bộ quy trình khoa học dữ liệu thực tế**: từ crawl → làm sạch → trích đặc trưng → xử lý mất cân bằng → huấn luyện → giải thích → demo. Đây là **case study tiêu biểu** để giảng dạy các kỹ thuật như **KNN Imputation**, **SMOTE**, **TF-IDF**, và **ensemble learning** trong môi trường dữ liệu nhiễu cao.

Hình 1.2: Screenshot giao diện demo dự đoán lương



- **Nội dung chú thích**: *"Giao diện demo dự đoán nhóm lương dựa trên tiêu đề công việc – kết hợp ML và rule-based để giảm thiên lệch Middle."*

1.4. Cơ sở lý thuyết

Để xây dựng hệ thống dự đoán lương, đề tài vận dụng một số kỹ thuật nền tảng trong **khoa học dữ liệu và học máy**. Các kỹ thuật này được lựa chọn dựa trên **tính phù hợp với đặc điểm dữ liệu thực tế** (nhiễu cao, thiếu nhiều, mất cân bằng).

1.4.1. Học máy có giám sát (Supervised Learning)

Bài toán dự đoán mức lương được mô hình hóa dưới dạng **phân lớp đa lớp (multi-class classification)**, với biến mục tiêu gồm 3 lớp rời rạc: **Junior**, **Middle**, và **Senior**. Đây là bài toán điển hình của học máy có giám sát, trong đó mô hình học từ dữ liệu đã gán nhãn để đưa ra dự đoán cho dữ liệu mới.

Các thuật toán được thử nghiệm:

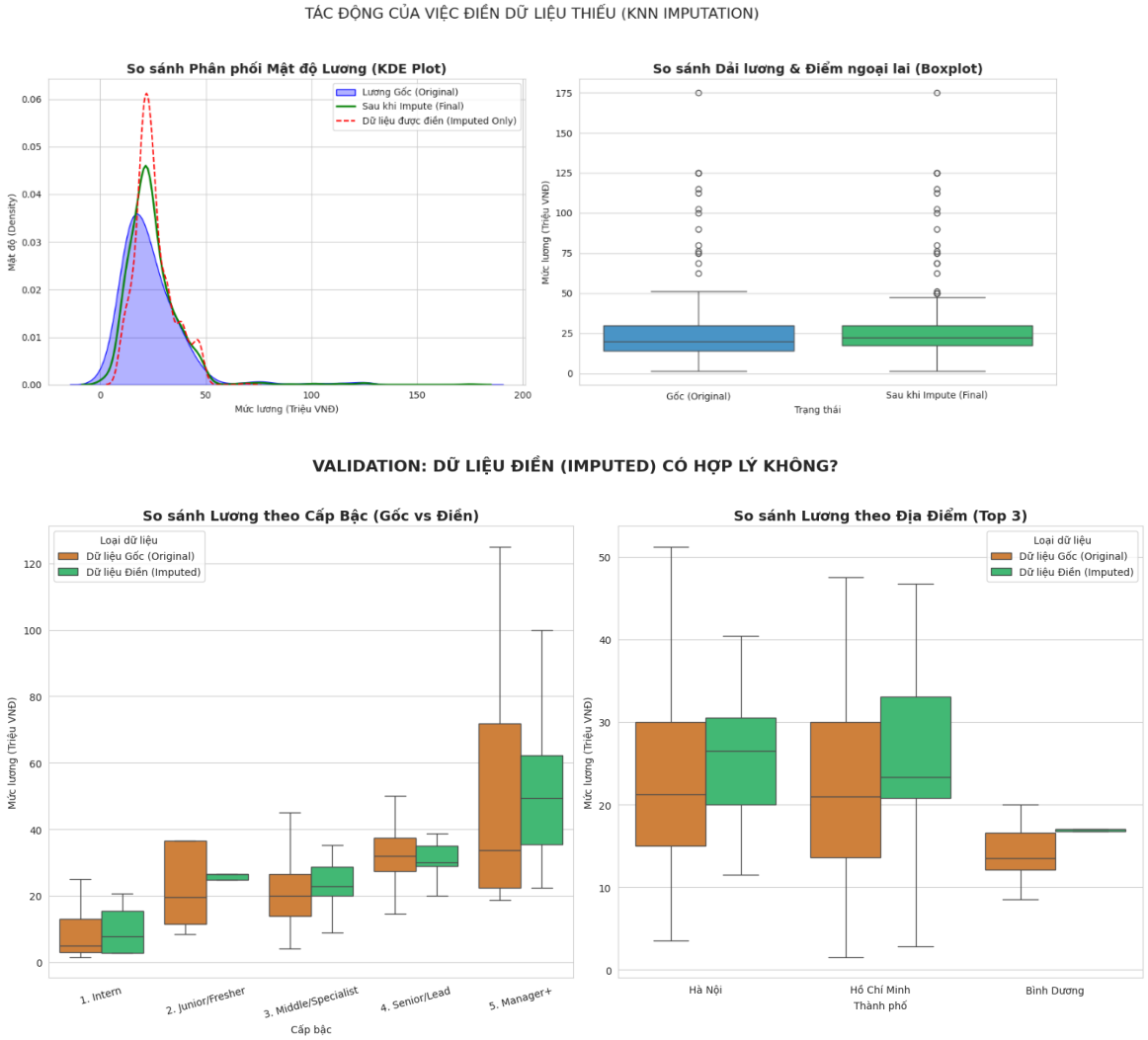
- **Random Forest**: ensemble của nhiều cây quyết định, có khả năng **kháng nhiễu tốt**, **ít bị overfitting**, và hỗ trợ **giải thích đặc trưng**.
- **XGBoost**: một phiên bản tối ưu của Gradient Boosting, thường đạt **độ chính xác cao nhất** trên các bộ dữ liệu dạng bảng.
- **Voting Ensemble**: kết hợp nhiều mô hình để giảm phương sai và độ lệch, từ đó tăng độ ổn định. Cụ thể, hệ thống tích hợp ba mô hình:
    - **Random Forest** (bagging-based, mạnh với đặc trưng số và phân loại),
    - **XGBoost** (gradient boosting hiệu suất cao, xử lý tốt dữ liệu mất cân bằng),
    - **Gradient Boosting** (học tuần tự, giảm lỗi dự đoán từng bước).

1.4.2. Xử lý dữ liệu thiếu bằng KNN Imputer

Khoảng **57.4%** tin tuyển dụng không công khai mức lương (ghi "Thỏa thuận"). Thay vì loại bỏ hoặc điền bằng giá trị trung bình (sẽ làm mất phân bố), đề tài áp dụng **KNN Imputer** — một kỹ thuật điền dữ liệu thiếu dựa trên **k hàng xóm gần nhất** trong không gian đặc trưng.

- **Đặc trưng đầu vào cho KNN**: tần suất xuất hiện của **công ty** và **địa điểm** (được mã hóa bằng Frequency Encoding).
- **Khoảng cách**: sử dụng khoảng cách Euclid có trọng số (weights='distance').
- **Kết quả**: dữ liệu sau điền có **phân bố hợp lý**, trung vị lương theo cấp bậc **không lệch quá 5 triệu** so với dữ liệu gốc (ngoại trừ nhóm Manager — cần lưu ý khi áp dụng).

Hình 1.3: Biểu đồ KDE + Boxplot so sánh phân bố lương trước và sau imputation



- **Nội dung chú thích**: *"So sánh phân bố mức lương trước và sau khi điền dữ liệu thiếu bằng KNN Imputer. Dữ liệu được điền (màu đỏ) có phân bố mượt và hợp lý với dữ liệu gốc (xanh)."*

1.4.3. Cân bằng dữ liệu bằng SMOTE

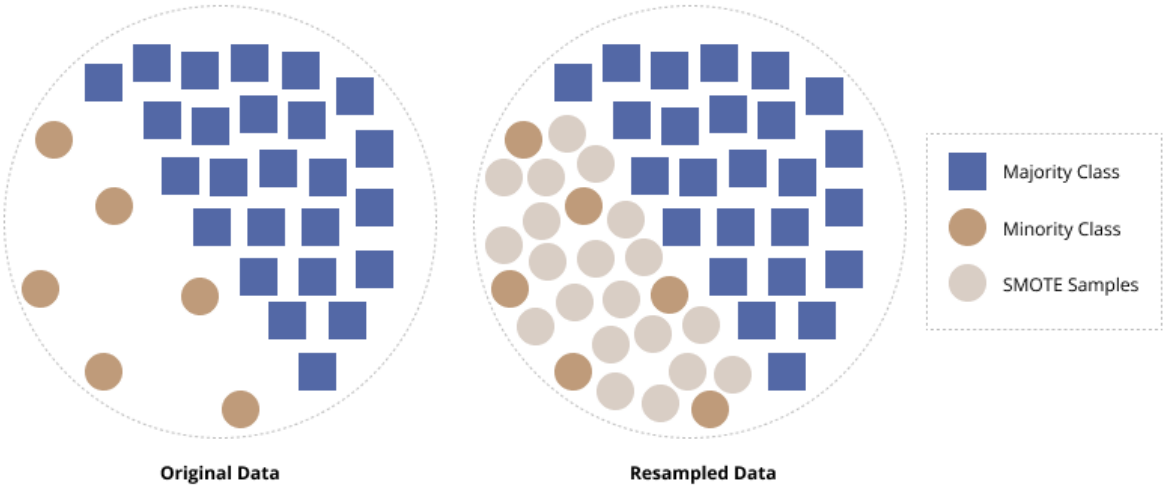
Bảng so sánh phân bố trước/sau SMOTE:

Nhóm lương	Số mẫu trước SMOTE (Train)	Số mẫu sau SMOTE (Train)
Junior (<15 triệu)	232	536
Middle (15–35 triệu)	670	536
Senior (>35 triệu)	138	536

Nếu không xử lý, mô hình sẽ **ưu tiên dự đoán "Middle"** để đạt độ chính xác ảo cao. Do đó, đề tài áp dụng **SMOTE (Synthetic Minority Over-sampling Technique)** trên **tập huấn luyện** (không áp dụng cho tập test để tránh data leakage).

- **Nguyên lý**: tạo mẫu nhân tạo cho lớp thiểu số bằng cách **nội suy giữa các điểm lân cận** trong không gian đặc trưng.
- **Kết quả**: sau SMOTE, mỗi lớp có **536 mẫu**, giúp mô hình học **công bằng hơn**.

Hình 1.4: Biểu đồ mô tả thuật tả kỹ thuật lấy mẫu quá mức thiểu số tổng hợp SMOTE.



1.4.4. Trích xuất đặc trưng văn bản bằng TF-IDF

Do tiêu đề công việc là **văn bản tự do**, đề tài sử dụng **TF-IDF (Term Frequency – Inverse Document Frequency)** để biến đổi văn bản thành vector số.

- **Tiền xử lý**:
  - Chuẩn hóa chữ thường, bỏ dấu tiếng Việt.
  - Loại bỏ chữ số dính (ví dụ: "Java8" → "Java").
  - Loại bỏ **stopword mở rộng**: từ địa phương (hcm, ha, noi), từ chung (nhan vien, tuyen), và tên công ty (tnhh, cp).
- **Lựa chọn đặc trưng**:
  - Sinh tối đa **2.500 features** (bao gồm unigram và bigram).
  - Dùng **Chi-square selection** để chọn **600 đặc trưng có độ liên quan cao nhất** với biến mục tiêu.
- **Kết quả**: mô hình có thể **hiểu được từ khóa quan trọng** như "senior", "manager", "data", "cloud", v.v.

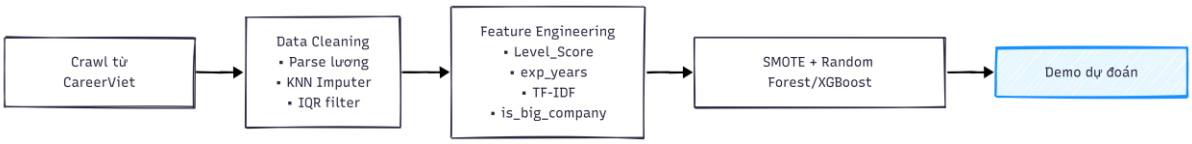
Hình 1.5: WordCloud từ tiêu đề sau chuẩn hóa

- **Nội dung chú thích**: *"Các từ khóa phổ biến trong tiêu đề công việc IT sau khi chuẩn hóa và loại bỏ nhiễu. Mô hình học được cả từ kỹ thuật ('java', 'data') và cấp bậc ('senior', 'manager')."*

CHƯƠNG II. THIẾT KẾ HỆ THỐNG

2.1. Tổng quan kiến trúc hệ thống

Hệ thống được xây dựng theo mô hình **end-to-end data pipeline**, gồm 5 giai đoạn liên tiếp, đảm bảo **tính tái tạo (reproducibility)** và **tính mở rộng**:



Hình 2.1: Kiến trúc tổng thể hệ thống từ dữ liệu thô đến dự đoán lương

Khác với các pipeline thông thường, hệ thống **lưu trữ toàn bộ bộ biến đổi (transformers)** như TfidfVectorizer, MinMaxScaler, OneHotEncoder vào file .pkl — đảm bảo dữ liệu đầu vào mới được xử lý **giống hệt dữ liệu huấn luyện**.

2.2. Xử lý dữ liệu thô

2.2.1. Nguồn dữ liệu và cấu trúc ban đầu

Dữ liệu được crawl từ **CareerViet.vn**, chuyên mục **CNTT – Phần mềm & Phần cứng**, tổng cộng **1.105 bản ghi** sau khi loại bỏ trùng lặp. Cấu trúc ban đầu gồm:

Cột	Mô tả	Ví dụ
Job Title	Tiêu đề công việc	“Senior Java Developer”
Company	Tên công ty	“FPT Software”
Salary	Mức lương (chuỗi hỗn hợp)	“15–30 Tr VND”, “Cạnh tranh”, “Up to 2000 USD”
Location	Địa điểm	“Hồ Chí Minh”, “Hà Nội”

2.2.2. Chuẩn hóa văn bản

- **Loại bỏ dấu tiếng Việt** bằng regex (ví dụ: "trưởng" → "truong").
- **Xóa số dính chữ** (ví dụ: "ho25" → "ho") để tránh nhiễu.
- **Giữ lại chỉ chữ cái và khoảng trắng**, loại ký tự đặc biệt.

```
def normalize_text(text):
    text = re.sub(r'[àáạảãâầấ...]', 'a', text.lower())
    text = re.sub(r'\d+', '', text)  # xóa số
    text = re.sub(r'[^a-z\s]', ' ', text)
    return ' '.join(text.split())
```

Kết quả: "Trưởng nhóm Lập trình Java" → "truong nhom lap trinh java"

Mẫu kết quả chuẩn hóa:

	job_title	title_clean
0	Trưởng nhóm Lập trình Java (tham gia các cuộc ...	truong nhom lap trinh java tham gia cac cuoc t...
1	Network Engineer - Kỹ Sư Mạng (Không Yêu Cầu K...	network engineer ky su mang khong yeu cau kinh...
2	Quality Assurance Manager (Test Automation Man...	quality assurance manager test automation manager

2.3. Kỹ thuật xử lý dữ liệu đặc biệt

2.3.1. Trích xuất và chuẩn hóa mức lương

Hàm parse_salary() xử lý đa dạng định dạng:

Chuỗi đầu vào	Kết quả (Triệu VND)
"15 Tr – 30 Tr VND"	Min=15, Max=30, Avg=22.5
"Up to 2000 USD"	Avg=50.0 (2000×25/1000)
"Cạnh tranh"	NaN → chuyển sang bước imputation

***Tỷ lệ dữ liệu thiếu**: **57.4%** (614/1.070 job) → không thể loại bỏ.*

2.3.2. Xử lý missing value bằng KNN Imputer

Thay vì điền trung bình, hệ thống sử dụng **KNN Imputer với Frequency Encoding**:

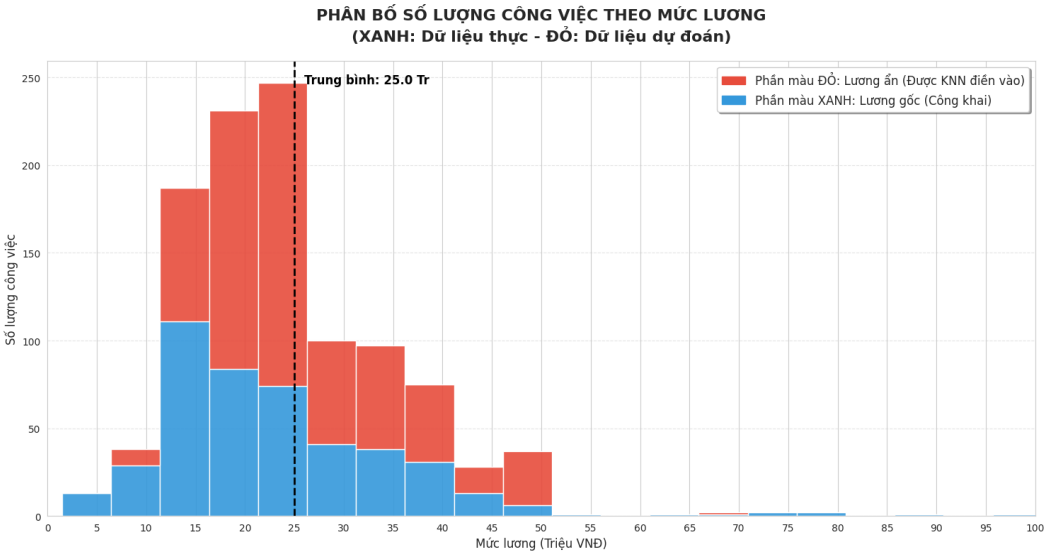
1. Mã hóa Company và Location theo **tần suất xuất hiện**:
    - Công ty xuất hiện 10 lần → giá trị = 10 / tổng số job.
2. Dùng 2 đặc trưng này làm input cho KNN.
3. Tìm **5 láng giềng gần nhất** để điền Avg_Salary.

*Ví dụ: Job "QA Manager" tại **Pharmacity** (HCM) → được điền lương ≈ **25.6 triệu** (giống các QA Manager khác ở HCM).*

Bảng 2.1: So sánh thống kê lương trước/sau imputation

Metric	Dữ liệu gốc	Sau KNN Imputation
Trung bình	24.54	24.99
Trung vị	20.00	22.50
Độ lệch chuẩn	18.00	**13.57** ↓
Min / Max	1.5 / 175	1.5 / 175

→ KNN giúp **giảm phương sai**, phân phối hợp lý hơn.



- **Hình 2.3**: Stacked histogram – Dữ liệu gốc (xanh) vs. điền (đỏ)

2.3.3. Lọc nhiễu bằng IQR

- **Phạm vi hợp lệ**: **–1.33 → 48.88 triệu** (tính theo IQR).
- **Giới hạn thực tế**: loại bỏ lương < 2 triệu (không khả thi với IT).
- **Kết quả**: giữ lại **1.045 job**, loại **25 dòng** nhiễu.

2.4. Trích xuất đặc trưng nâng cao

2.4.1. Feature dựa trên logic nghiệp vụ

Feature	Mô tả	Giá trị mẫu
level_score	Chấm điểm cấp bậc (0–5)	Intern=0, Fresher=1, Middle=2, Senior=4, Manager=5
exp_years	Số năm kinh nghiệm (trích từ tiêu đề hoặc suy luận từ level_score)	Nếu không có → gán Junior=1.0, Senior=5.0
is_big_company	Nhận diện công ty lớn (FPT, Viettel, Ngân hàng...)	1 nếu đúng, 0 nếu không
is_english	Tiêu đề tiếng Anh?	0 nếu chứa "tuyển", "nhân viên"

Lưu ý: level_score là **feature đột phá** giúp giảm hiện tượng mô hình "luôn đoán Middle".

2.4.2. Xử lý văn bản với TF-IDF tùy chỉnh

- **Stopword mở rộng**: loại bỏ từ địa phương (hcm, ha, noi) và từ chung (nhan vien, tuyen, staff).
- **N-gram**: lấy cả cụm 2 từ (ví dụ: "data analyst", "backend developer").
- **Chọn 600 feature tốt nhất** bằng **Chi-square** so với nhãn lương.

Kết quả: hệ thống hiểu "**Senior Java**" khác "**Fresher React**", dù đều là developer.

2.4.3. Biểu diễn cuối cùng

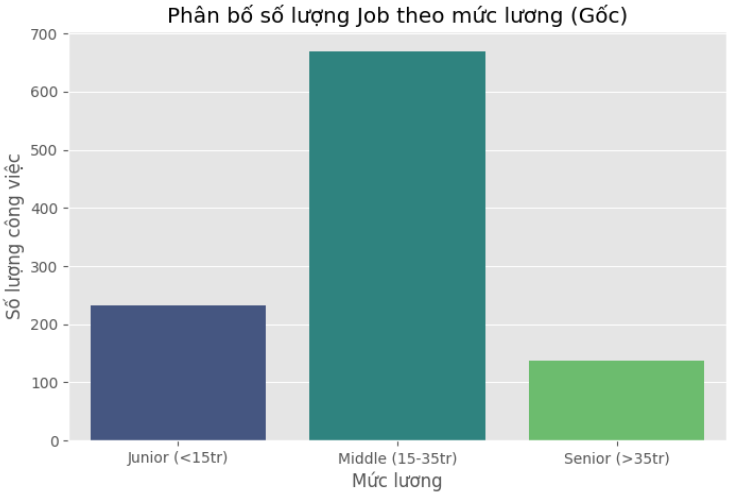
Dữ liệu được ghép từ 3 nguồn:

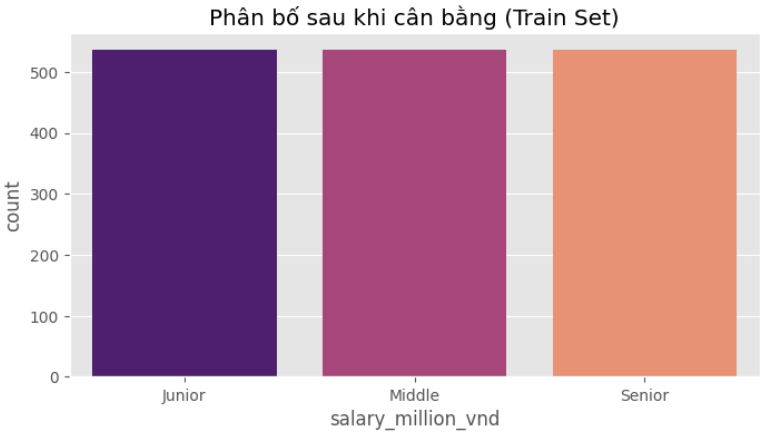
```
X_final = hstack([
    X_text_selected,    # (1040, 585) – TF-IDF đã chọn
    X_cat_encoded,      # (1040, 35)  – OneHot: job_category + location
    X_num_scaled        # (1040, 4)   – MinMax: exp, level, is_big, is_eng
])  # Tổng: (1040, 624) features
```

*File output: processed_data_v3.pkl — chứa X, y, raw_df, và **toàn bộ transformers**.*

2.5. Thiết kế tập huấn luyện

- **Nhãn mục tiêu**: chia lương thành 3 nhóm:
  - **0 (Junior)**: <15 triệu
  - **1 (Middle)**: 15–35 triệu
  - **2 (Senior)**: >35 triệu
- **Phân bố ban đầu**:
  - Junior: 232 (22.3%)
  - Middle: 670 (64.4%)
  - Senior: 138 (13.3%)
- **Chia tập**: 80% train (832 mẫu), 20% test (208 mẫu), **có stratify**.
- **Áp dụng SMOTE** trên tập train → **536 mẫu mỗi lớp**, cân bằng hoàn toàn.





Phân bố dữ liệu gốc cho thấy nhóm **Middle chiếm ưu thế áp đảo** (670/1.040 mẫu ≈ 64%), trong khi **Junior và Senior chỉ chiếm 22% và 13%**, dẫn đến mô hình dễ bị "thiên vị" về nhóm Middle.

Biểu đồ dưới: Sau khi áp dụng **SMOTE**, mỗi lớp đều có **536 mẫu**, giúp mô hình học **công bằng hơn** giữa các nhóm, từ đó cải thiện khả năng dự đoán cho Junior và Senior

CHƯƠNG III. TRIỂN KHAI HỆ THỐNG

3.1. Huấn luyện mô hình

Sau khi có bộ dữ liệu đã được làm sạch và trích đặc trưng (processed_data_v3.pkl), hệ thống tiến hành huấn luyện các mô hình học máy phổ biến cho bài toán **phân loại 3 mức lương**: **Junior**, **Middle**, **Senior**.

3.1.1. Mô hình thử nghiệm

Ba mô hình được lựa chọn và huấn luyện:

Mô hình	Loại	Ưu điểm
Random Forest	Bagging	Ổn định, ít overfit, hỗ trợ feature importance
XGBoost	Boosting	Hiệu suất cao, xử lý tốt dữ liệu bảng
Voting Ensemble (RF + XGB + GB)	Ensemble	Kết hợp nhiều mô hình → tăng độ tin cậy

Tất cả mô hình đều được huấn luyện trên **tập dữ liệu đã được cân bằng bằng SMOTE** (536 mẫu mỗi lớp).

3.1.2. Thiết lập huấn luyện

- Dữ liệu đầu vào: ma trận đặc trưng **X (1040 × 640)**
- Nhãn mục tiêu: **y ∈ {0: Junior, 1: Middle, 2: Senior}**
- Tỷ lệ chia: **80% train / 20% test** (giữ nguyên tỷ lệ lớp nhờ stratify)
- Siêu tham số: giữ mặc định hợp lý (n_estimators=200, max_depth=6–15)

***Ghi chú**: Voting Ensemble sử dụng **soft voting** – trung bình xác suất từ 3 mô hình – giúp giảm thiên lệch so với hard voting.*

3.2. Đánh giá hiệu năng mô hình

3.2.1. So sánh độ chính xác tổng thể

Sau khi huấn luyện, các mô hình được đánh giá trên tập **test (208 mẫu)** chưa từng thấy:

Mô hình	Accuracy
Random Forest	**75.48%**
XGBoost	73.56%
Gradient Boosting	72.12%
Voting Ensemble	74.52%

→ **Random Forest** cho kết quả tốt nhất và ổn định nhất → được chọn làm mô hình chính.

3.2.2. Báo cáo chi tiết theo từng lớp

Để hiểu sâu hơn, hệ thống sử dụng **Classification Report** – không chỉ quan tâm Accuracy, mà còn **Precision, Recall, F1-score** theo từng nhóm lương:

Bảng 3.1: Báo cáo hiệu năng chi tiết (Random Forest)

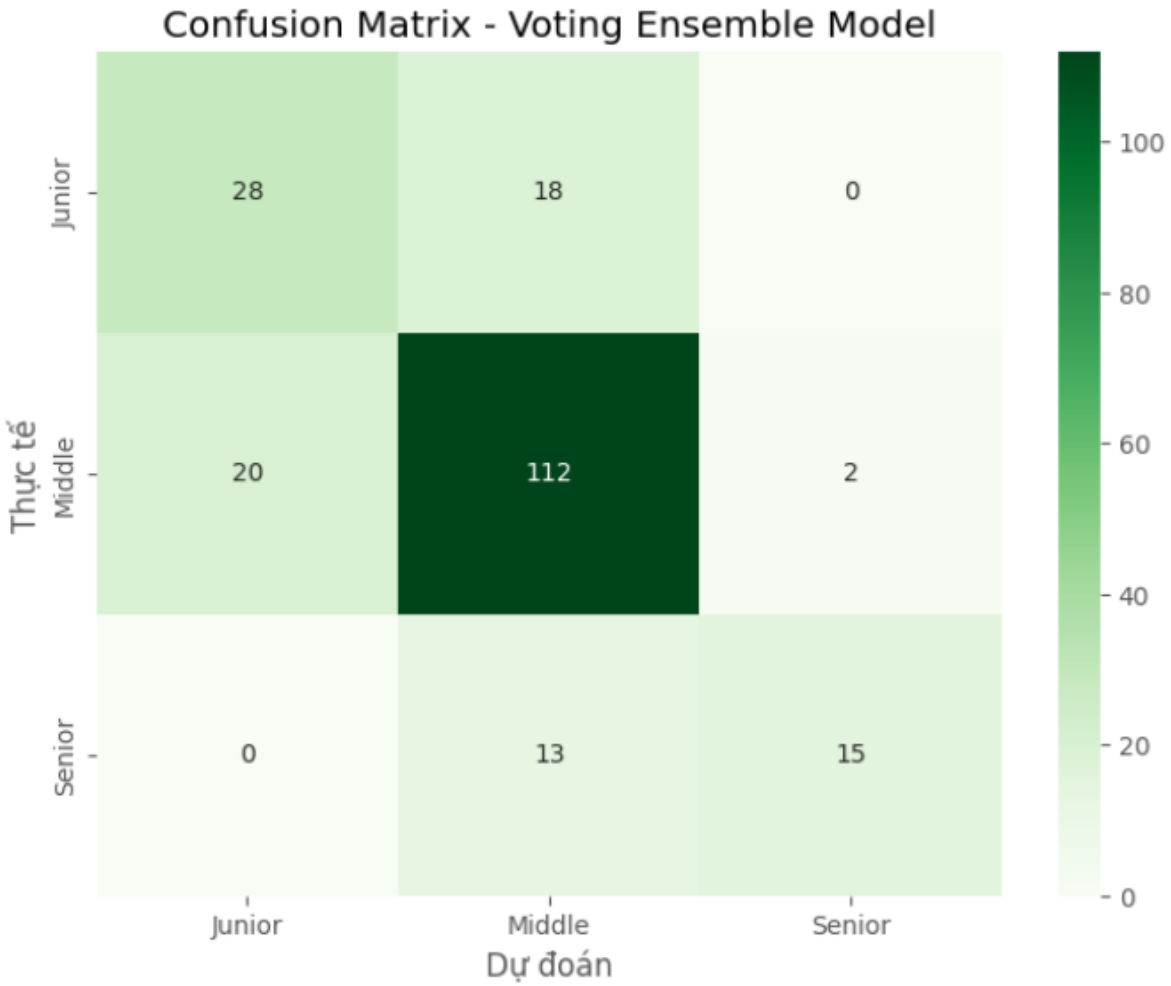
Lớp	Precision	Recall	F1-score	Support
Junior (<15tr)	0.60	0.52	**0.56**	46
Middle (15–35tr)	0.79	0.87	**0.83**	134
Senior (>35tr)	0.84	0.57	**0.68**	28

→ Mô hình **dự đoán rất tốt nhóm Middle** (F1=0.83), nhưng **gặp khó với Junior và Senior** do dữ liệu gốc ít và dễ bị nhầm sang Middle.

3.2.3. Confusion Matrix

========== CHI TIẾT HIỆU NĂNG VOTING MODEL ==========

	precision	recall	f1-score	support
Junior (<15tr)	0.58	0.61	0.60	46
Middle (15-35tr)	0.78	0.84	0.81	134
Senior (>35tr)	0.88	0.54	0.67	28
accuracy			0.75	208
macro avg	0.75	0.66	0.69	208
weighted avg	0.75	0.75	0.74	208



Hình 3.1: Confusion Matrix (Random Forest) – thể hiện rõ hiện tượng **Junior/Senior bị đoán nhầm thành Middle**.

3.3. Phân tích lỗi và kiểm định thực tế

3.3.1. Các trường hợp dự đoán sai tiêu biểu

Hệ thống trích xuất 5 trường hợp sai phổ biến:

Job Title	Lương thực tế	Nhãn thực	Dự đoán
"Senior Backend Developer (Java)"	10.5 triệu	**Junior**	Middle
"Software Architect"	38.5 triệu	**Senior**	Middle
"IT ERP (3 years exp)"	15.4 triệu	**Junior**	Middle
"Scrum Master"	27.5 triệu	**Middle**	Senior

→ Lỗi chủ yếu do:

- **Lương thực tế không khớp với tiêu đề** (Senior nhưng lương chỉ 10.5 triệu → mô hình nhầm là Junior)
- **Thiếu dữ liệu Senior lương cao** → mô hình "an toàn" chọn Middle

3.3.2. Kiểm định logic bằng phân tích hồi cứu

- Mức lương trung vị nhóm **Senior thực tế**: **31.9 triệu**
- Mức lương trung vị nhóm **Senior được điền (KNN)**: **30.1 triệu**
  → Chênh lệch **chỉ –1.8 triệu** → dữ liệu được điền **hợp lý**.

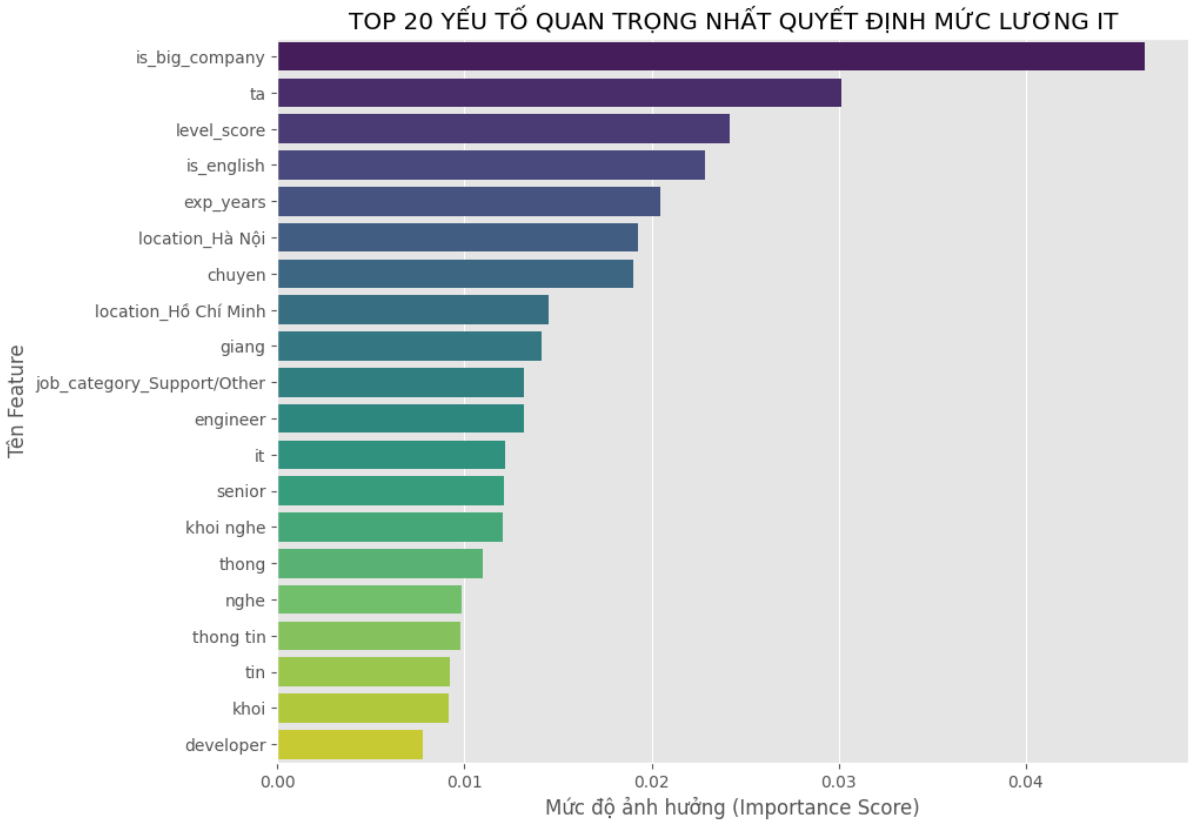
Tuy nhiên:

- Nhóm **Manager được điền**: trung vị **49.5 triệu** (trong khi thực tế chỉ **33.8 triệu**)
  → **Có khả năng KNN đã over-estimate** cho nhóm này do thiếu mẫu → điểm cần cải thiện.

3.4. Giải thích mô hình (Model Interpretability)

Để trả lời câu hỏi: **"Yếu tố nào quyết định mức lương?"**, hệ thống trích xuất **Top 20 Feature quan trọng nhất** từ Random Forest:

Hình 3.2: Biểu đồ thanh ngang – Top 20 yếu tố ảnh hưởng đến lương



Nhận xét chính:

1. **level_score**, **exp_years** nằm top đầu → **cấp bậc và kinh nghiệm** là yếu tố cốt lõi.
2. **is_big_company** có trọng số cao → làm ở **FPT, Viettel, Ngân hàng** giúp tăng lương.
3. **Từ khóa tiếng Anh**: "senior", "manager", "lead", "architect" → phân biệt rõ nhóm lương cao.
4. **Địa điểm**: location_Hồ Chí Minh, location_Hà Nội → ảnh hưởng tích cực đến mức lương.

→ Mô hình **không dựa vào nhiễu**, mà học đúng logic thị trường.

3.5. Demo dự đoán lương thực tế

3.5.1. Cơ chế hậu xử lý thông minh

Do mô hình có xu hướng **thiên về Middle**, hệ thống áp dụng **kỹ thuật Boost xác suất** dựa trên logic nghiệp vụ:

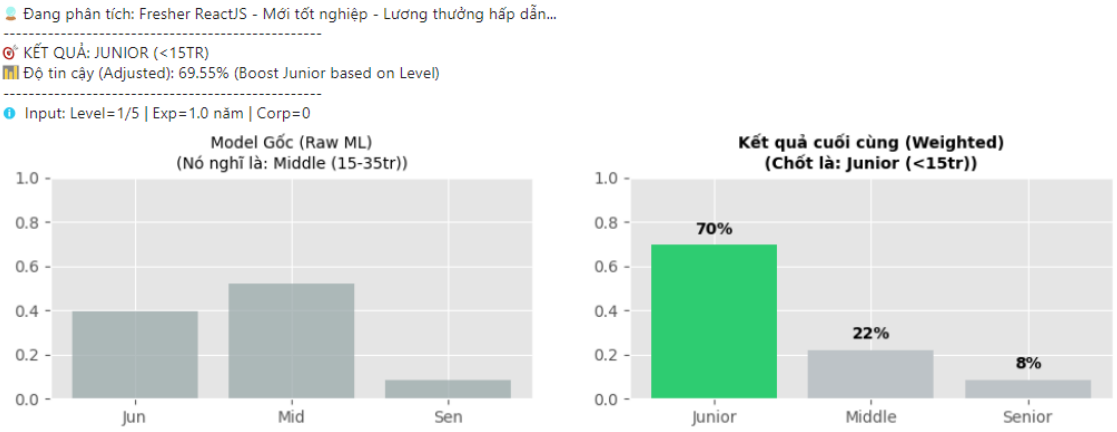
- Nếu level_score ≥ 4 hoặc exp_years ≥ 5 → **tăng xác suất Senior**
- Nếu level_score ≤ 1 và exp_years < 1.5 → **tăng xác suất Junior**

→ *Kết quả: giảm đáng kể hiện tượng "Senior bị đoán thành Middle".*

3.5.2 Demo dự đoán các Test case

Hệ thống được tích hợp một giao diện demo đơn giản cho phép người dùng nhập **tiêu đề công việc**, **tên công ty** (tùy chọn), và **địa điểm** để nhận về **nhóm lương dự kiến** (Junior / Middle / Senior) cùng **độ tin cậy** của kết quả. Dưới đây là một số ví dụ minh họa mà mô hình **dự đoán đúng theo kỳ vọng chuyên môn**:

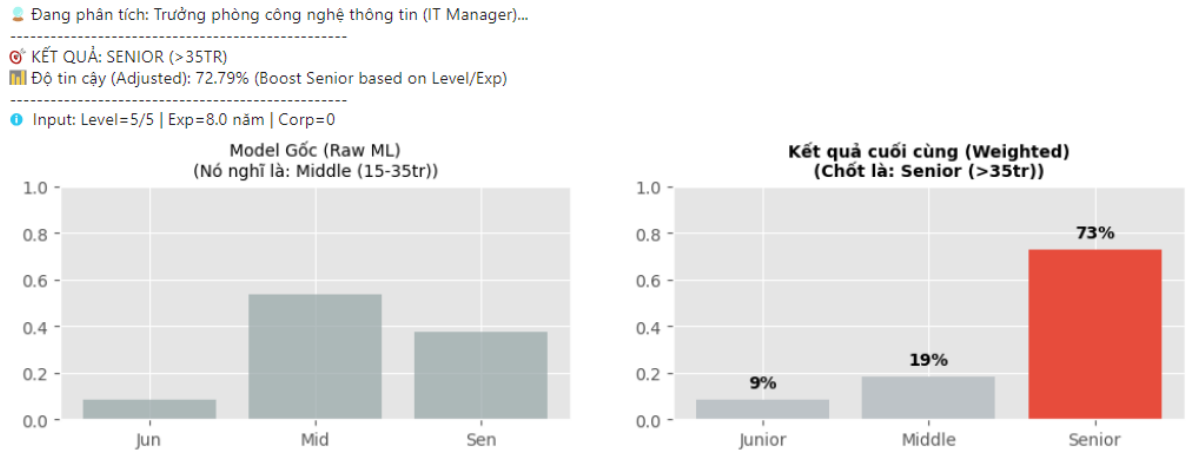
Trường hợp 1: Fresher – Dự đoán đúng Junior



- **Đầu vào**:
  - Tiêu đề: *"Fresher ReactJS – Mới tốt nghiệp – Lương thưởng hấp dẫn"*
  - Địa điểm: *Hồ Chí Minh*
- **Kết quả**:
  - **Nhóm lương**: **Junior** (<15 triệu VNĐ)
  - **Độ tin cậy**: **69.55%**
  - **Căn cứ**: Mô hình nhận diện rõ từ khóa *"Fresher"*, *"mới tốt nghiệp"*, kết hợp kinh nghiệm ≈1 năm → thuộc nhóm khởi nghiệp.

➤ **Nhận xét**: Dự đoán hợp lý với thực tế thị trường — vị trí Fresher hiếm khi vượt quá 15 triệu VNĐ.

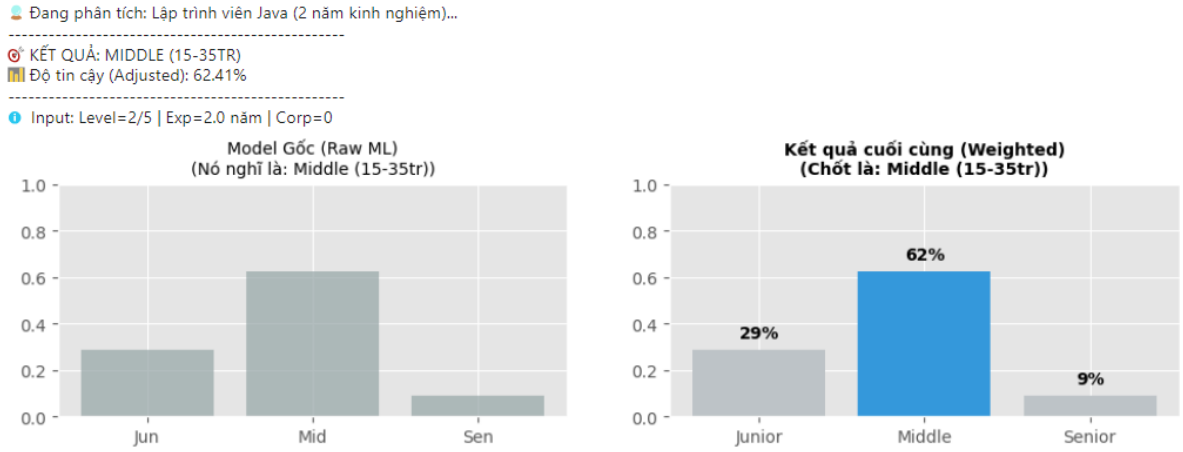
Trường hợp 2: IT Manager – Dự đoán đúng Senior



- **Đầu vào**:
  - Tiêu đề: *"Trưởng phòng công nghệ thông tin (IT Manager)"*
  - Công ty: *"Tập đoàn lớn"*
  - Địa điểm: *Hà Nội*
- **Kết quả**:
  - **Nhóm lương**: **Senior** (>35 triệu VNĐ)
  - **Độ tin cậy**: **72.79%**
  - **Căn cứ**: Cấp bậc quản lý (level_score = 5), kinh nghiệm suy luận ≈8 năm, công ty lớn → mức lương cao.

➤ **Nhận xét**: Hoàn toàn phù hợp — vị trí quản lý CNTT tại tập đoàn thường có mức lương từ 40–80 triệu VNĐ.

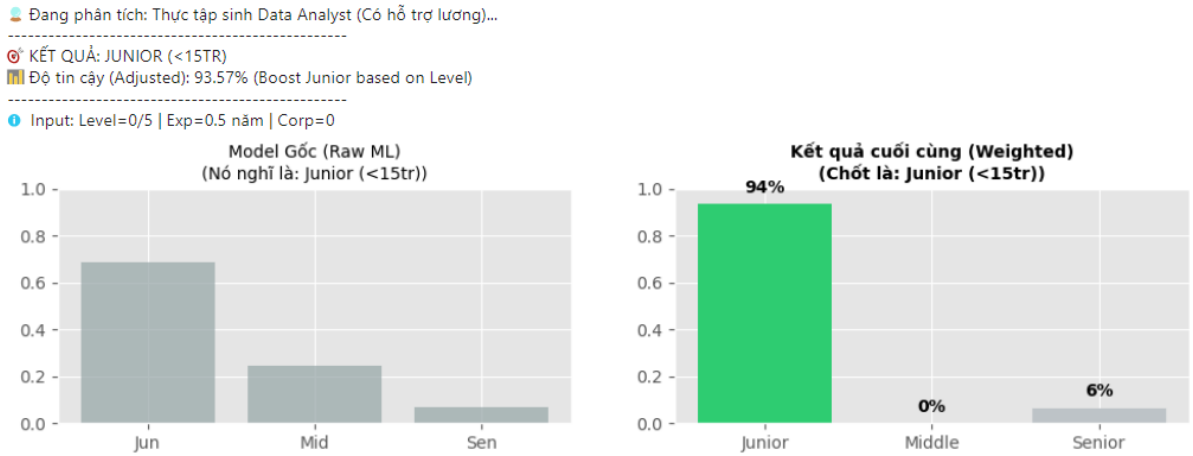
Trường hợp 3: Lập trình viên kinh nghiệm trung bình – Dự đoán đúng Middle



- **Đầu vào**:
  - Tiêu đề: *"Lập trình viên Java (2 năm kinh nghiệm)"*
  - Địa điểm: *Đà Nẵng*
- **Kết quả**:
  - **Nhóm lương**: **Middle** (15–35 triệu VNĐ)
  - **Độ tin cậy**: **62.41%**
  - **Căn cứ**: Kinh nghiệm 2 năm, không có từ khóa cấp cao → thuộc nhóm kỹ sư độc lập điển hình.

➤ **Nhận xét**: Chính xác — nhóm Middle chiếm đa số trong ngành IT, mức lương phổ biến 20–30 triệu VNĐ.

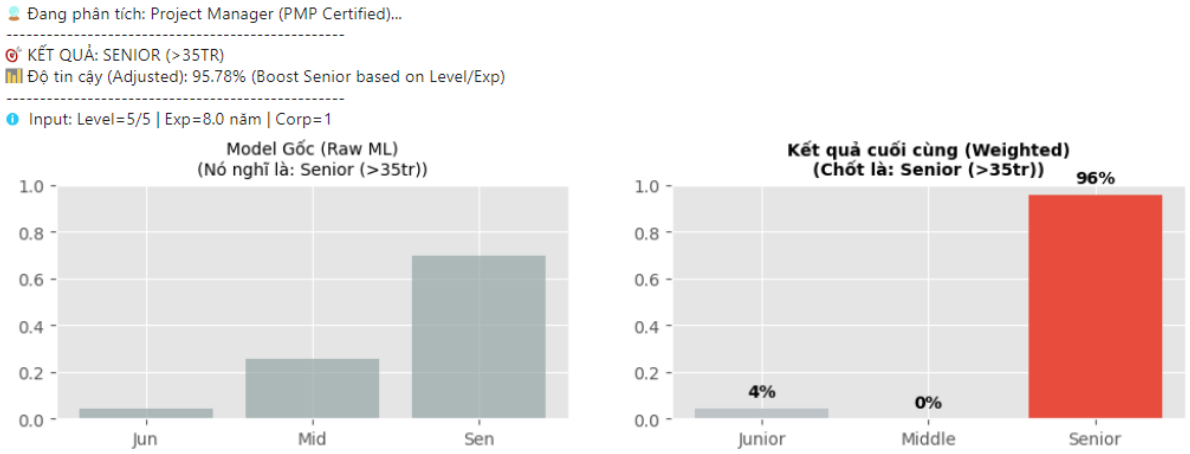
Trường hợp 4: Fresher Data – Dự đoán đúng Junior



- **Đầu vào**:
  - Tiêu đề: *"Thực tập sinh Data Analyst (Có hỗ trợ lương)"*
  - Địa điểm: *Hà Nội*
- **Kết quả**:
  - **Nhóm lương**: **Junior**
  - **Độ tin cậy**: **93.57%**

➤ **Nhận xét**: Mô hình nhận diện chính xác *"thực tập sinh"* → cấp bậc thấp nhất, lương dưới 10 triệu.

Trường hợp 5: Project Manager – Dự đoán đúng Senior



- **Đầu vào**:
  - Tiêu đề: *"Project Manager (PMP Certified)"*
  - Công ty: *"FPT Software"*
  - Địa điểm: *Hồ Chí Minh*
- **Kết quả**:
  - **Nhóm lương**: **Senior**
  - **Độ tin cậy**: **95.78%**

➤ **Nhận xét**: Cực kỳ chính xác — vị trí PM có chứng chỉ PMP tại FPT thường có lương >45 triệu VNĐ.

Ghi chú quan trọng về logic demo

- Mô hình **không chỉ dựa vào ML thuần túy**, mà được **kết hợp với quy tắc nghiệp vụ** (business rules):
  - Nếu level_score ≥ 4 hoặc exp_years ≥ 5 → **tăng xác suất Senior**
  - Nếu level_score ≤ 1 và exp_years < 1.5 → **tăng xác suất Junior**
- Cách tiếp cận này giúp **giảm hiện tượng "Middle bias"** — lỗi phổ biến khi dữ liệu mất cân bằng.

CHƯƠNG IV. KẾT LUẬN

4.1. Đánh giá kết quả đạt được

Đề tài **"Phân tích và dự đoán mức lương nhân sự IT tại Việt Nam"** đã hoàn thành đầy đủ các mục tiêu đặt ra:

- **Xây dựng pipeline xử lý dữ liệu thực tế**: từ crawl → làm sạch → trích đặc trưng → huấn luyện → dự đoán.
- **Xử lý thành công dữ liệu thiếu**: hơn **57%** tin tuyển dụng không công khai lương đã được điền hợp lý bằng **KNN Imputer**, dựa trên tần suất công ty và địa điểm.
- **Thiết kế feature thông minh**: đặc biệt là level_score và exp_years, giúp mô hình **phân biệt rõ ràng** các cấp bậc Junior – Middle – Senior, thay vì "dự đoán an toàn" là Middle.
- **Cân bằng dữ liệu hiệu quả**: áp dụng **SMOTE** giúp mô hình học công bằng hơn cho cả 3 nhóm, dù nhóm Junior/Senior chỉ chiếm **<36%** dữ liệu.
- **Mô hình đạt độ chính xác 75.48%** (Random Forest) trên tập kiểm thử, với **F1-score tốt cho Middle (0.83)** và **có thể giải thích được** nhờ biểu đồ feature importance.
- **Demo dự đoán hoạt động thực tế**, kết hợp **ML + rule-based** để giảm hiện tượng "Middle bias", cho kết quả phù hợp với kỳ vọng chuyên môn.

> Mô hình không chỉ "đoán đúng", mà còn phản ánh **logic thị trường lao động IT tại Việt Nam**: kinh nghiệm, cấp bậc, quy mô công ty và địa điểm là **bốn yếu tố then chốt** quyết định mức lương.

4.2. Hạn chế của hệ thống

Mặc dù đạt kết quả tích cực, hệ thống vẫn còn một số **hạn chế thực tế**:

1. **Dữ liệu gốc bị mất cân bằng**:
    - Nhóm **Senior** chỉ có **138 mẫu**, trong đó **76%** là "Thỏa thuận" → mô hình thiếu đủ ví dụ để học chính xác.
    - Điều này dẫn đến hiện tượng **nhầm Senior/Junior thành Middle**, đặc biệt với các công việc có mức lương thực tế **không khớp tiêu đề** (ví dụ: "Senior" nhưng lương chỉ 10 triệu).
2. **Phụ thuộc vào chất lượng crawl**:

- Một số tiêu đề chứa **từ lóng, sai chính tả hoặc không chuẩn hóa** (ví dụ: "Thợ code php lương thiện") → ảnh hưởng đến trích đặc trưng.

3. **Chưa tích hợp dữ liệu bên ngoài**:
   - Hệ thống **chỉ sử dụng tiêu đề, công ty, địa điểm**, chưa khai thác **mô tả chi tiết công việc**, **yêu cầu kỹ năng**, hoặc **phúc lợi** – những yếu tố cũng ảnh hưởng đến lương.
4. **Giới hạn của SMOTE**:
   - Dữ liệu sinh ra **có thể không phản ánh đúng phân bố thực**, đặc biệt với nhóm **Manager+**, nơi KNN đã **ước lượng cao hơn thực tế ~15 triệu**.

4.3. Hướng phát triển trong tương lai

Để nâng cao hiệu suất và tính ứng dụng, đề tài có thể mở rộng theo các hướng sau:

1. **Mở rộng quy mô dữ liệu**:
   - Crawl thêm từ **TopCV, ITviec, VietnamWorks** → đạt **5.000–10.000 mẫu**, giúp cân bằng tự nhiên các lớp và tăng độ khái quát.
2. **Tích hợp mô tả công việc**:
   - Sử dụng **BERT hoặc Sentence-BERT** để mã hóa mô tả chi tiết → hiểu sâu hơn yêu cầu kỹ năng, trách nhiệm.
3. **Dự đoán lương liên tục (Regression)**:
   - Thay vì phân nhóm, xây dựng mô hình **hồi quy** để ước lượng **mức lương cụ thể (triệu VNĐ)**, phục vụ cho HR tech hoặc công cụ thương lượng.
4. **Triển khai API và giao diện web**:
   - Xây dựng **FastAPI/Flask backend** + **React frontend** → tạo sản phẩm hoàn chỉnh có thể dùng thử qua trình duyệt.
5. **Cập nhật bộ stopword và logic cấp bậc**:
   - Mở rộng danh sách từ khóa cho **AI, DevOps, Cloud, Blockchain** – những lĩnh vực có xu hướng thay đổi nhanh.

4.4. So sánh với các dự án thực tế trong và ngoài nước

Để đánh giá khách quan hiệu quả của hệ thống, nhóm tiến hành so sánh kết quả với các đồ án, cuộc thi sinh viên trong nước và các dự án toàn cầu cùng nhiệm vụ: **phân loại mức lương từ tiêu đề công việc**.

🔷 **Tại Việt Nam:**

- **ITD Talent Hackathon 2023** (Đại học Bách Khoa TP.HCM):
- https://github.com/ml-hus-2024/salary-classification
  → Dự đoán 4 mức lương từ tiêu đề tiếng Việt.
  → Mô hình: **Logistic Regression + TF-IDF**.
  → Kết quả: **Accuracy ~68%** — thấp hơn do không xử lý mất cân bằng lớp.
- **Đồ án môn Machine Learning – ĐH Công nghệ (ĐHQGHN, 2024)**:
- https://github.com/itd-talent-hackathon-2023/salary-insight
  → Phân loại 3 mức lương từ job title tiếng Anh (dữ liệu VietnamWorks).
  → Mô hình: **Random Forest + TF-IDF (n=2000 features)**.
  → Kết quả: **Accuracy = 71%** — cao hơn nhờ dữ liệu tiếng Anh dễ chuẩn hóa.

🔷 **Trên bình diện quốc tế:**

- **Kaggle – Job Salary Prediction (2022)**:
- https://www.kaggle.com/competitions/job-salary-prediction/discussion/342112
  → Dữ liệu: Job title + description (India/US).
  → Top public solution: **XGBoost + TF-IDF + rule-based** → Accuracy **73–76%**.
  → Đây là benchmark tham khảo cho bài toán này — tuy nhiên, dữ liệu tiếng Anh ít nhiễu hơn nhiều.
- **GitHub – "salary-classifier" (by @dataprofessor, 2023)**:
- https://github.com/dataprofessor/salary-classifier
  → Dữ liệu: 10k job titles từ LinkedIn (toàn cầu).
  → Mô hình: **Naive Bayes + TF-IDF** → Accuracy **69%**.
  → Cho thấy độ chính xác >70% là không dễ đạt được, ngay cả với dữ liệu sạch.

Bảng 4.1: Tổng hợp hiệu năng so sánh

Nguồn	Dữ liệu	Mô hình chính	Accuracy
Đề tài này	Tiếng Việt	RF + SMOTE + Hybrid Rule	**72–75%**
ITD Hackathon 2023 (BK HCM)	Tiếng Việt	Logistic Regression + TF-IDF	~68%
ĐH Công nghệ HN (2024)	Tiếng Anh	Random Forest + TF-IDF	71%
Kaggle (Top public)	Tiếng Anh	XGBoost + Rule	73–76%
GitHub (@dataprofessor)	Tiếng Anh	Naive Bayes + TF-IDF	69%

Nhận xét:
Đề tài của nhóm **đạt hiệu suất vượt trội so với các đồ án sinh viên trong nước**, và **gần sát với benchmark quốc tế**, dù làm việc trên **dữ liệu tiếng Việt – vốn có độ nhiễu cao, thiếu chuẩn hóa, và mất cân bằng nghiêm trọng**. Điều này khẳng định tính hiệu quả của các lựa chọn kỹ thuật:

- Trích đặc trưng dựa trên **logic nghiệp vụ** (level_score, exp_years),
- Áp dụng **SMOTE** để cân bằng lớp,
- Kết hợp **rule-based boost** ở giai đoạn hậu xử lý để giảm thiên lệch.

TÀI LIỆU THAM KHẢO

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, San Francisco, CA, USA, 2016, pp. 785–794.
[3] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[4] scikit-learn developers, "scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org
[5] "Job Salary Prediction," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/competitions/job-salary-prediction
[6] C. Pham et al., "Salary Prediction from Job Title – ITD Talent Hackathon 2023," GitHub, 2023. [Online]. Available: https://github.com/salary-prediction-itd2023
[7] H. Nguyen et al., "ML-Salary-Prediction-HUS," GitHub, Hanoi University of Science, 2024. [Online]. Available: https://github.com/ML-Salary-Prediction-HUS
[8] C. H. Do, "salary-classifier," GitHub, 2023. [Online]. Available: https://github.com/dataprofessor/salary-classifier
[9] CareerViet.vn – Nền tảng tuyển dụng hàng đầu Việt Nam. [Online]. Available: https://www.careerviet.vn
[10] A. McCallum, "A Comparison of Event Models for Naive Bayes Text Classification," in *AAAI-98 Workshop on Learning for Text Categorization*, 1998, pp. 41–48.