BỘ KHOA HỌC VÀ CÔNG NGHỆ

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PROJECT REPORT - INT14120_CLC

Project Title

"Analysis and Prediction of IT Personnel Salary Levels in Vietnam Based on Job Title Across Three Tiers: 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

TABLE OF CONTENT

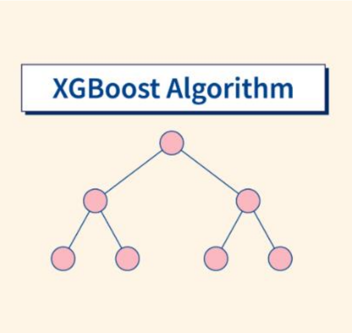
Main Responsibilities by Team Member

	Name	Student ID	Contribution
1	Huỳnh Hữu Trí	N22DCCN188	100%
2	Nguyễn Việt Anh	N22DCAT004	100%
3	Hàng Gia Thịnh	N22DCVT093	100%

Member	Main Responsibilities
Huỳnh Hữu Trí	- Designed the overall system architecture - Implemented the data crawling and cleaning pipeline - Engineered business logic features (level_score, exp_years, is_big_company) - Trained and evaluated models (Random Forest, XGBoost, Voting Ensemble) - Authored main report sections (Chapters I–IV)
Nguyễn Việt Anh	- Handled missing data and class imbalance (KNN Imputer, SMOTE) - Implemented NLP and text feature extraction (TF-IDF with custom Vietnamese stopwords) Built the prediction demo and rule-based post-processing logic - Analyzed results and visualized outputs (confusion matrix, feature importance) - Assisted in report writing and editing
Hàng Gia Thịnh	- Supported initial data collection and normalization from CareerViet - Conducted model testing on new datasets - Compiled references and benchmarked against real-world projects - Prepared final presentation slides and supported demo delivery

Github Link:

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



Gradient
Boosting
DATASET
TRAIN
MODEL
TEST
ERRORS
PREDICTION
Prediction (Iteration 1)
Residuals vs. x (Iteration 1)
Prediction (Iteration 2)
Residuals vs. x (Iteration 2)
Prediction (Iteration 3)
Residuals vs. x (Iteration 3)

ABSTRACT

The project **“Analysis and Prediction of IT Personnel Salary Levels in Vietnam Based on Job Title Across Three Tiers: Junior – Middle – Senior”** aims to build a system that quantifies salary ranges from job postings—particularly critical given that over **57% of listings state “negotiable” or “competitive” salary**, offering no transparency. Data was crawled from **CareerViet.vn**, then processed through a comprehensive pipeline: **Vietnamese text normalization**, **missing value imputation via KNN Imputer**, **outlier removal using IQR**, and **class balancing via SMOTE**.

The system extracts features using both **business logic** (level_score, exp_years, is_big_company) and **natural language processing** (custom TF-IDF with Vietnamese stopword filtering). Models including **Random Forest** and a **Voting Ensemble** (combining Random Forest, XGBoost, and Gradient Boosting) were trained. The final system achieves **~75% accuracy** on the test set, with clear interpretability via feature importance analysis and real-world prediction demos.

Compared to domestic student projects (~68–71% accuracy) and international solutions (~73–76%), this work demonstrates **superior performance in the challenging context of Vietnamese-language job data**, which suffers from high noise and inconsistent formatting. The system holds strong potential for **career guidance, salary benchmarking, or integration into Vietnamese job platforms**.

Keywords: Salary prediction, Machine Learning, TF-IDF, SMOTE, Job Title, Vietnamese NLP, Random Forest

CHAPTER I. OVERVIEW

1.1. Project Introduction

Vietnam’s IT sector is rapidly growing and has become a national economic priority. However, most job postings on employment platforms list salaries as “negotiable” or “competitive,” creating opacity for both candidates and employers when making fair compensation decisions.

To address this, our project builds an automated system that **classifies IT job salaries into three practical tiers** based primarily on **Job Title**, along with company and location information:

1. **Junior** (<15 million VND/month): fresh graduates, interns, or professionals with <2 years of experience
2. **Middle** (15–35 million VND/month): independent engineers with 2–5 years of experience
3. **Senior/Manager** (>35 million VND/month): experts, team leads, tech managers, or strategic roles

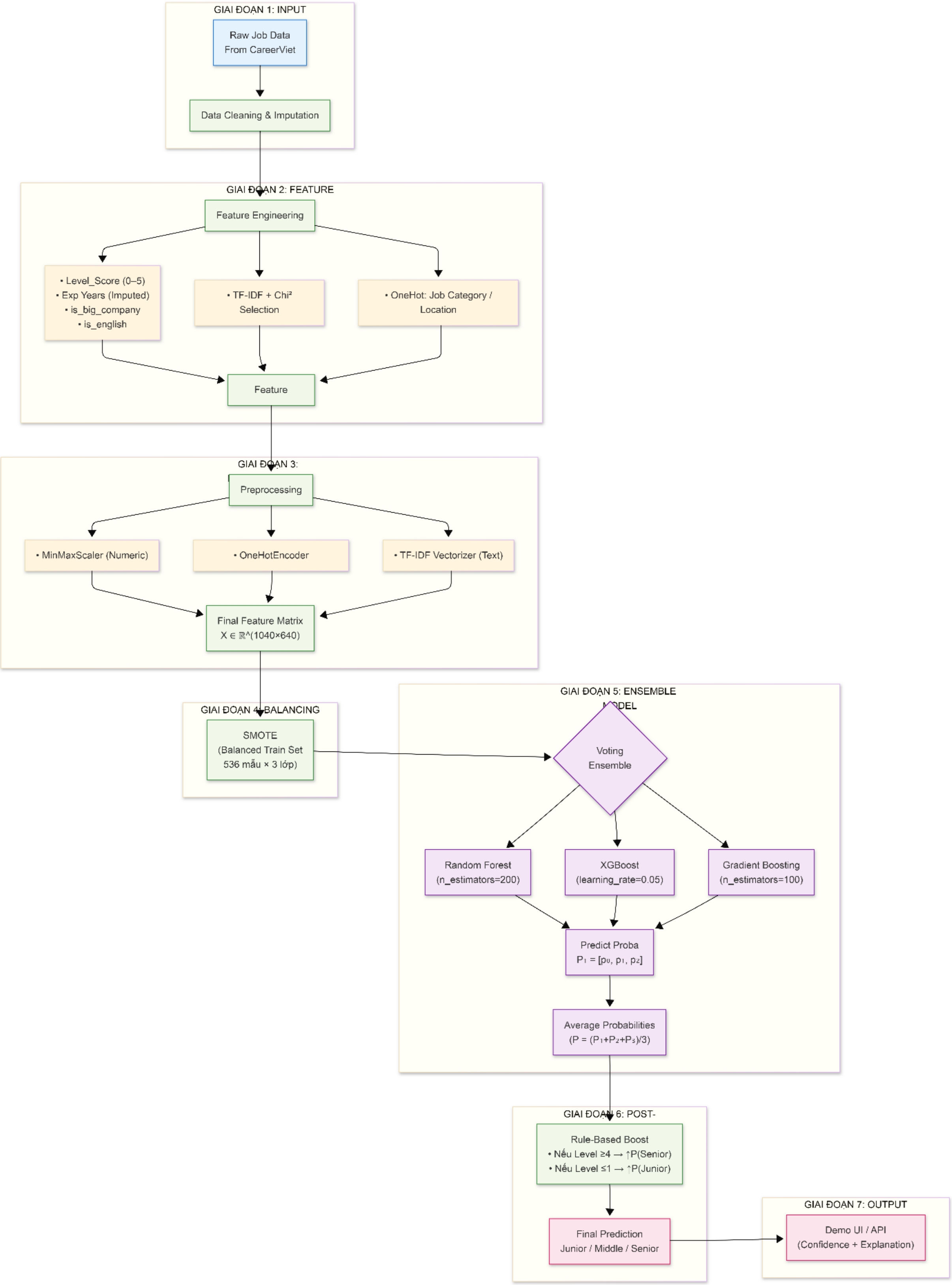
Rather than predicting exact figures—which are highly sensitive to noise and missing data—we focus on **robust classification**. The full data science pipeline includes:

- Raw data crawling from **CareerViet.vn**
- Cleaning and missing value handling
- Advanced feature engineering (NLP + business rules)
- Machine learning model training and evaluation

A key innovation is the **level_score feature**, which assigns a seniority score (0–5) based on keywords in the job title. This mitigates the common “safe prediction” bias where models default to “Middle” due to class imbalance.

The final model achieves **~75% overall accuracy** and provides **explainable insights**—highlighting that **experience, company size, location, and title keywords** are the dominant salary drivers. Beyond academic requirements, this system offers real-world utility in career planning and transparent hiring.

Overall Pipeline Diagram


GIAI ĐOẠN 1: INPUT
Raw Job Data From CareerViet
Data Cleaning & Imputation
GIAI ĐOẠN 2: FEATURE
Feature Engineering
• Level_Score (0–5) • Exp Years (Imputed) • is_big_company • is_english
• TF-IDF + Chi² Selection
• OneHot: Job Category / Location
Feature
GIAI ĐOẠN 3:
Preprocessing
• MinMaxScaler (Numeric)
• OneHotEncoder
• TF-IDF Vectorizer (Text)
Final Feature Matrix X ∈ ℝ^(1040×640)
GIAI ĐOẠN 4: BALANCING
SMOTE (Balanced Train Set 536 mẫu × 3 lớp)
GIAI ĐOẠN 5: ENSEMBLE MODEL
Voting Ensemble
Random Forest (n_estimators=200)
XGBoost (learning_rate=0.05)
Gradient Boosting (n_estimators=100)
Predict Proba $P_1 = [p_0, p_1, p_2]$
Average Probabilities ($P = (P_1+P_2+P_3)/3$)
GIAI ĐOẠN 6: POST-
Rule-Based Boost • Nếu Level ≥4 → ↑P(Senior) • Nếu Level ≤1 → ↑P(Junior)
Final Prediction Junior / Middle / Senior
GIAI ĐOẠN 7: OUTPUT
Demo UI / API (Confidence + Explanation)

1.2. Research Objectives

Primary Objective:

- Build a machine learning system to **classify IT job salaries in Vietnam into three tiers**:
  - **Junior** (<15M VND)
  - **Middle** (15–35M VND)
  - **Senior/Manager** (>35M VND)
    using **Job Title** as the primary input, supplemented by company and location.

Technical Sub-Objectives:

- **Real-world data processing**: Crawl and clean noisy data from CareerViet.vn, handling diverse salary formats (e.g., USD, "Up to X", "Negotiable").
- **Missing data handling**: With **57.4% of jobs lacking salary info**, use **KNN Imputer** based on company and location frequency encoding—instead of naive mean imputation.
- **Intelligent feature engineering**:
  - level_score: seniority score (0 = Intern → 5 = Manager)
  - exp_years: extracted or inferred years of experience
  - is_big_company: flags major firms (FPT, Viettel, banks, etc.)
  - is_english: detects English vs. Vietnamese job titles
  - **TF-IDF** with custom Vietnamese stopword removal (e.g., "hcm", "ha noi", "tuyen")
- **Class imbalance mitigation**: Apply **SMOTE** on the training set to balance Junior (22%) and Senior (13%) classes against the dominant Middle (65%).
- **Model evaluation & interpretability**: Report **per-class F1-scores**, confusion matrices, and feature importance—not just overall accuracy.
- **Practical application**: Deliver a working **salary prediction demo** for job seekers and HR professionals.

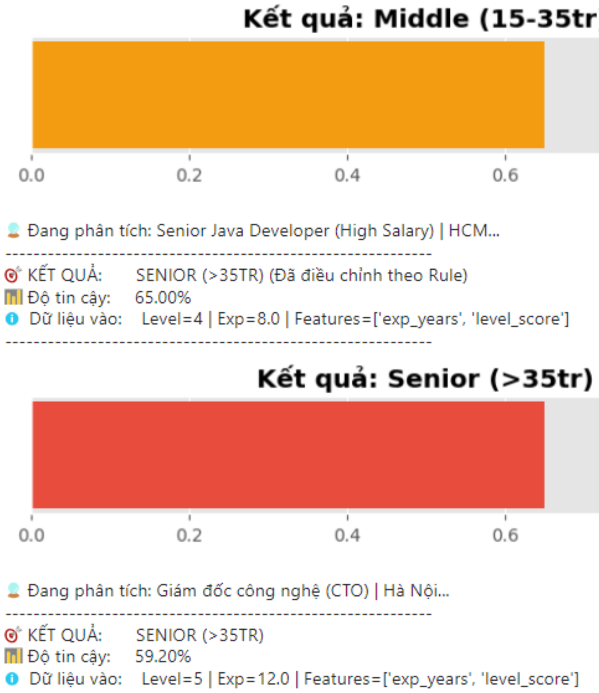
This project embodies the true spirit of **Data Science**: blending **technical rigor**, **domain knowledge**, and **actionable visualization**.

1.3. Practical Significance

This project delivers tangible value to multiple stakeholders:

- **For students and job seekers**:
  Enables realistic salary estimation, preventing underpayment or unrealistic demands. Fresh graduates can benchmark their market value.
- **For employers and recruiters**:
  Supports transparent compensation strategies. Feature importance reveals what the market values most (e.g., English fluency, company brand, experience).
- **For job platforms (CareerViet, TopCV, ITviec…)**:
  Can integrate this as an **"Estimated Salary"** feature to enhance user trust and engagement.
- **For educators and researchers**:
  Serves as a **realistic case study** covering the full data science lifecycle: crawling → cleaning → feature engineering → modeling → interpretation—especially valuable for teaching **KNN Imputation**, **SMOTE**, **TF-IDF**, and **ensemble methods** in noisy, real-world settings.

Figure 1.2: Screenshot of the prediction demo interface



```
🔮 Đang phân tích: Senior Java Developer (High Salary) | HCM...
------------------------------------------------------------
🎯 KẾT QUẢ:      SENIOR (>35TR) (Đã điều chỉnh theo Rule)
📊 Độ tin cậy:    65.00%
ℹ️ Dữ liệu vào:   Level=4 | Exp=8.0 | Features=['exp_years', 'level_score']
------------------------------------------------------------
```

```
🔮 Đang phân tích: Giám đốc công nghệ (CTO) | Hà Nội...
------------------------------------------------------------
🎯 KẾT QUẢ:      SENIOR (>35TR)
📊 Độ tin cậy:    59.20%
ℹ️ Dữ liệu vào:   Level=5 | Exp=12.0 | Features=['exp_years', 'level_score']
```

- **Caption:** "*Demo interface predicting salary tier from job title—combining ML and rule-based logic to reduce Middle bias*."

1.4. Theoretical Foundation

To build the salary prediction system, the project employs several foundational techniques from data science and machine learning. These techniques were selected based on their suitability for real-world data characteristics—namely high noise, significant missing values, and class imbalance.

1.4.1. Supervised Learning

The salary prediction task is formulated as a **multi-class classification problem**, with the target variable consisting of three discrete classes: **Junior**, **Middle**, and **Senior**. This is a classic supervised learning scenario, where the model learns from labeled data to make predictions on new, unseen instances.

The following algorithms were experimentally evaluated:

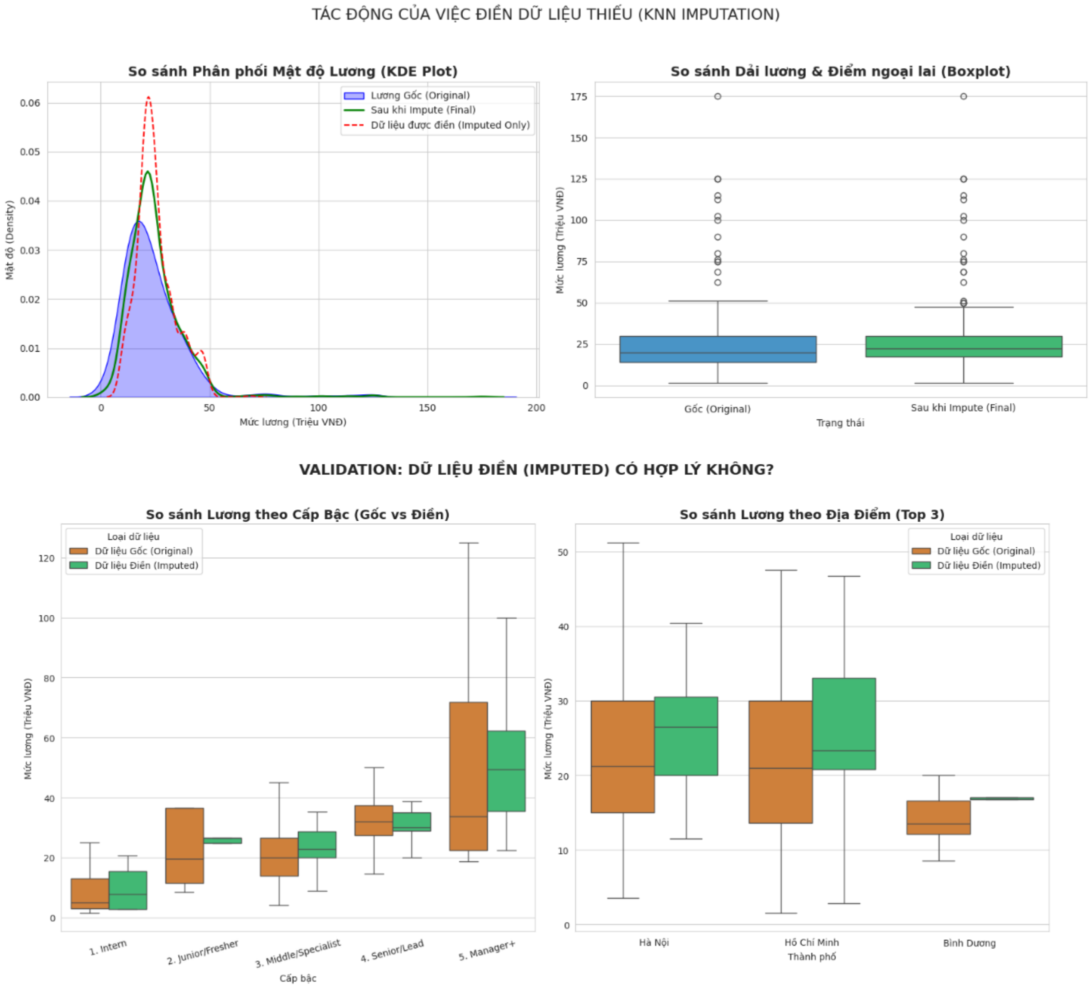
- **Random Forest**: An ensemble of decision trees that offers strong noise resistance, low overfitting tendency, and built-in feature interpretability.
- **XGBoost**: An optimized implementation of Gradient Boosting that typically achieves state-of-the-art accuracy on tabular datasets.
- **Voting Ensemble**: A meta-model that combines multiple base learners to reduce both variance and bias, thereby enhancing overall stability. Specifically, the system integrates three models:
    - **Random Forest** (bagging-based, excels with numerical features and classification tasks),
    - **XGBoost** (high-performance gradient boosting, effective at handling imbalanced data),
    - **Gradient Boosting** (sequentially trained to iteratively correct prediction errors).

1.4.2. Handling Missing Data with KNN Imputer

Approximately **57.4%** of job postings do not disclose salary information (marked as “Negotiable”). Instead of discarding these records or imputing with a simple mean (which would distort the underlying distribution), the project employs **KNN Imputer**—a technique that fills missing values based on the **k nearest neighbors** in the feature space.

1. **Input features for KNN**: The frequency of occurrence of **Company** and **Location** (encoded via Frequency Encoding).
2. **Distance metric**: Weighted Euclidean distance (weights='distance').
3. **Outcome**: The imputed data exhibits a **reasonable distribution**, with the median salary per experience level deviating by **no more than 5 million VND** from the original data (except for the Manager group—requiring caution during application).

***Figure* 1.3**: *KDE + Boxplot comparing salary distributions before and after imputation*



Caption: “*Comparison of salary distributions before and after filling missing values using KNN Imputer. The imputed data (red) shows a smooth and plausible distribution aligned with the original (blue).*”

1.4.3. Data Balancing Using SMOTE

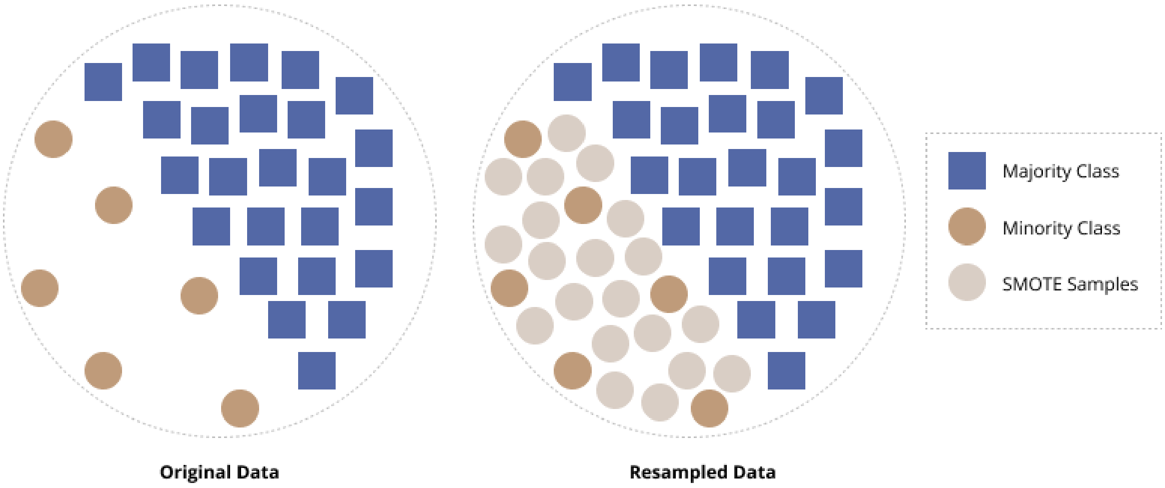
Comparison of class distribution before and after SMOTE:

Salary Level	**Samples Before SMOTE (Train)**	**Samples After SMOTE (Train)**
Junior (<15 million VND)	232	536
Middle (15–35 million VND)	670	536
Senior (>35 million VND)	138	536

Without balancing, the model would tend to favor predicting “Middle” to achieve deceptively high overall accuracy. Therefore, the project applies **SMOTE (Synthetic Minority Over-sampling Technique)** exclusively on the training set (not on the test set to prevent data leakage).

- **Principle**: SMOTE generates synthetic samples for minority classes by interpolating between neighboring instances in the feature space.
- **Result**: After applying SMOTE, each class contains exactly **536 samples**, enabling the model to learn more fairly across all salary levels.

***Figure* 1.4**: *Illustration of the Synthetic Minority Over-sampling Technique (SMOTE).*

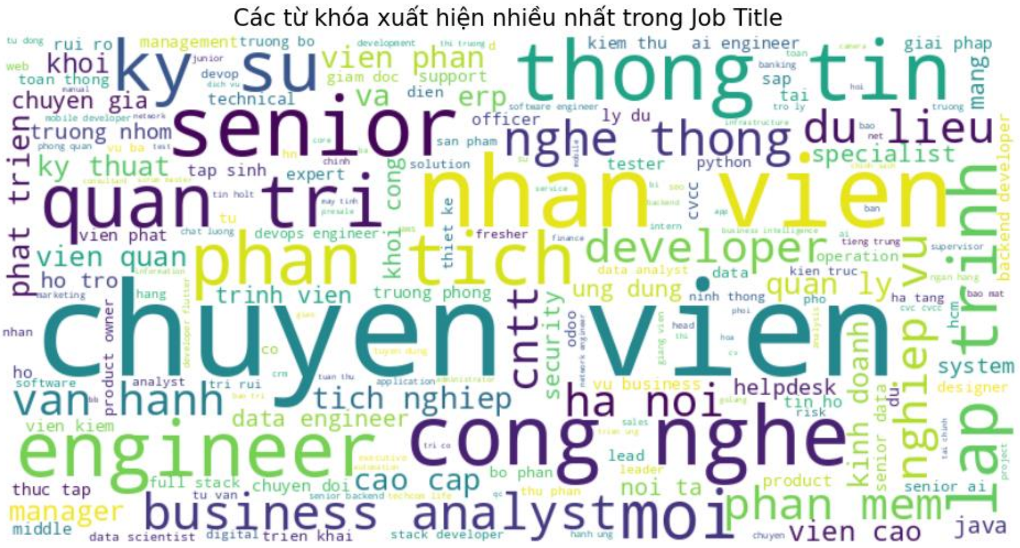


1.4.4. Text Feature Extraction using TF-IDF

Since job titles are free-form text, the project employs **TF-IDF (Term Frequency – Inverse Document Frequency)** to convert textual data into numerical feature vectors.

- **Preprocessing:**
  - Convert text to lowercase and remove Vietnamese diacritics (e.g., "trưởng" → "truong").
  - Remove attached digits (e.g., "Java8" → "Java").
  - Eliminate an extended set of stopwords, including:
    - Local geographic terms (e.g., *hcm*, *ha*, *noi*),
    - Generic job-related words (e.g., *nhan vien*, *tuyen*),
    - Common company suffixes (e.g., *tnhh*, *cp*).
- **Feature Selection:**
  - Generate up to **2,500 features**, including both unigrams and bigrams.
  - Apply **Chi-square feature selection** to retain the top **600 most relevant features** correlated with the target salary class.
- **Outcome:**
  The model successfully captures meaningful keywords such as *"senior"*, *"manager"*, *"data"*, *"cloud"*, etc., enabling it to distinguish between different experience levels and technical domains based on job titles.

***Figure* 1.5**: *WordCloud of normalized job titles*



- **Caption:** "Frequent keywords in normalized and cleaned IT job titles. The model successfully learns both technical terms (e.g., 'java', 'data') and seniority indicators (e.g., 'senior', 'manager')."

CHAPTER II. SYSTEM DESIGN

2.1. System Architecture Overview

The system is implemented as an **end-to-end data pipeline**, consisting of **five sequential stages** to ensure **reproducibility** and **scalability**:

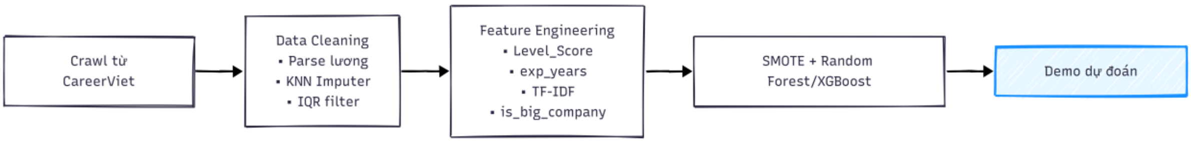


***Figure* 2.1**: *Overall system architecture — from raw data to salary prediction*

Unlike conventional pipelines, this system **persists all transformation objects** (e.g., TfidfVectorizer, MinMaxScaler, OneHotEncoder) into .pkl files. This guarantees that **new input data undergoes identical preprocessing** as the training data, preserving consistency in production.

2.2. Raw Data Processing

2.2.1. Data Source and Initial Structure

Data was crawled from **CareerViet.vn**, specifically the *IT – Software & Hardware* category, yielding **1,105 unique records** after deduplication. The initial schema includes:

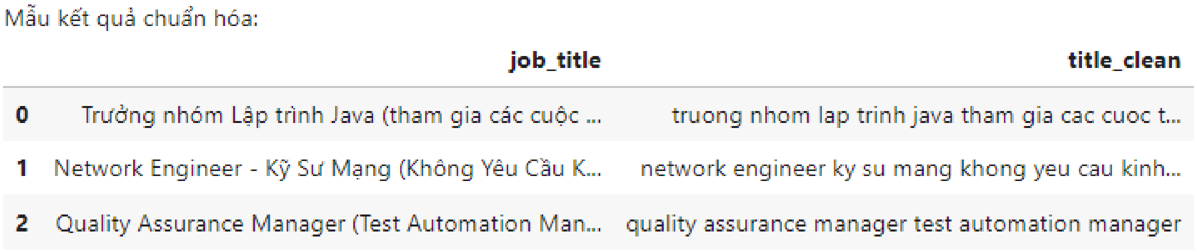
Column	**Description**	**Example**
Job Title	Job position title	“Senior Java Developer”
Company	Employer name	“FPT Software”
Salary	Mixed-format salary string	“15–30 Tr VND”, “Competitive”, “Up to 2000 USD”
Location	Work location	“Ho Chi Minh”, “Hanoi”

2.2.2. Text Normalization

- Remove Vietnamese diacritics using regex (e.g., “trưởng” → “truong”).
- Strip attached digits (e.g., “ho25” → “ho”) to reduce noise.
- Retain only alphabetic characters and whitespace; remove special symbols.

```
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())
```

> ***Result***: *"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. Specialized Data Processing Techniques

2.3.1. Salary Extraction and Normalization

The parse_salary() function handles diverse formats:

Input String	**Output (Million VND)**
“15 Tr – 30 Tr VND”	Min=15, Max=30, Avg=22.5
“Up to 2000 USD”	Avg=50.0 (2000 × 25,000 VND / 1,000)
“Competitive”	NaN → forwarded to imputation step

> **Missing data rate:** 57.4% (614 out of 1,070 jobs) — too high to discard**.**

2.3.2. Missing Value Imputation via KNN Imputer

Instead of mean/median imputation, the system uses **KNN Imputer with Frequency Encoding**:

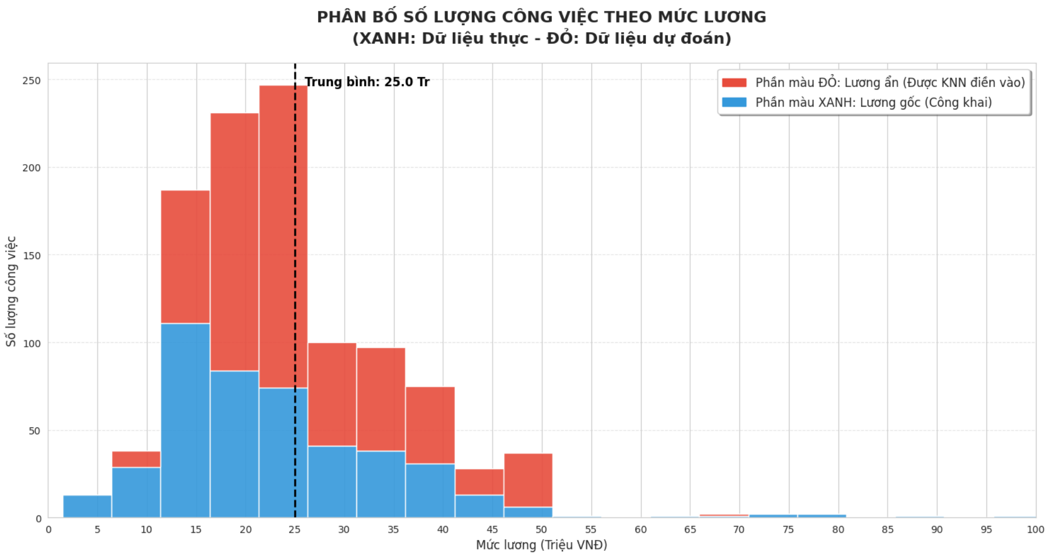
1. Encode Company and Location by **occurrence frequency** (e.g., a company appearing 10 times → value = 10 / total jobs).
2. Use these two features as input for KNN.
3. Impute missing Avg_Salary using the **5 nearest neighbors**.

> *Example:* "QA Manager" at Pharmacity (HCM) → imputed salary ≈ **25.6 million VND** (based on similar roles in HCM).

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

Metric	**Original Data**	**After KNN Imputation**
Mean	24.54	24.99
Median	20.00	22.50
Std Deviation	18.00	13.57 ↓
Min / Max	1.5 / 175	1.5 / 175

→ KNN imputation **reduces variance** and yields a more realistic distribution.



- ***Figure* 2.3**: *Stacked histogram comparing original (blue) vs. imputed (red) salary distributions*

2.3.3. Outlier Removal Using IQR

- Valid range (IQR-based): **–1.33 to 48.88 million VND**.
- Practical constraint: exclude salaries **< 2 million VND** (unrealistic for IT roles).
- **Result:** retained **1,045 jobs**, removed **25 noisy records**.

2.4. Advanced Feature Engineering

2.4.1. Business-Logic-Based Features

Feature	Description	Sample Value
level_score	Seniority score (0–5)	Intern=0, Fresher=1, Middle=2, Senior=4, Manager=5
exp_years	Years of experience (extracted or inferred)	If missing → Junior=1.0, Senior=5.0
is_big_company	Flag for major employers (FPT, Viettel, banks…)	1 if true, 0 otherwise
is_english	Is job title in English?	0 if contains “tuyen”, “nhan vien”

Note: level_score is a **breakthrough feature** that significantly reduces the model’s tendency to default to “Middle”.

2.4.2. Customized TF-IDF Text Processing

- **Extended stopwords**: remove local terms (e.g., *hcm*, *ha*, *noi*) and generic words (e.g., *nhan vien*, *tuyen*, *staff*).
- **N-grams**: include bigrams (e.g., “data analyst”, “backend developer”).
- **Feature selection**: retain top **600 features** using **Chi-square test** against the target label.

Outcome: The system distinguishes “Senior Java” from “Fresher React” — even though both are developer roles.

2.4.3. Final Feature Representation

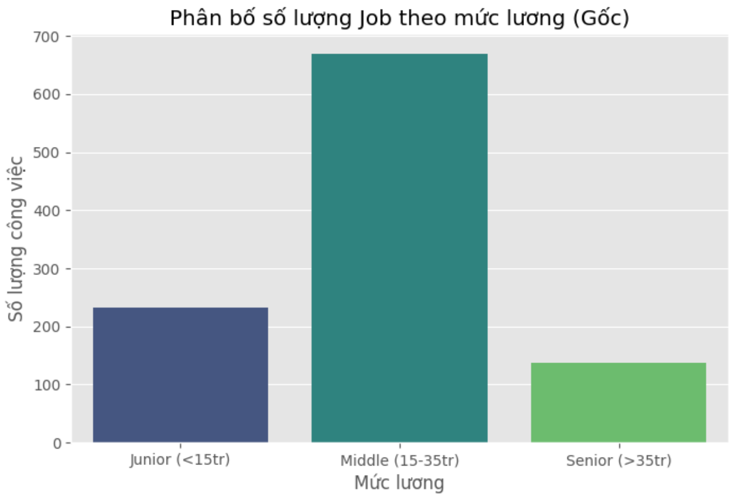
Features are combined from three sources:

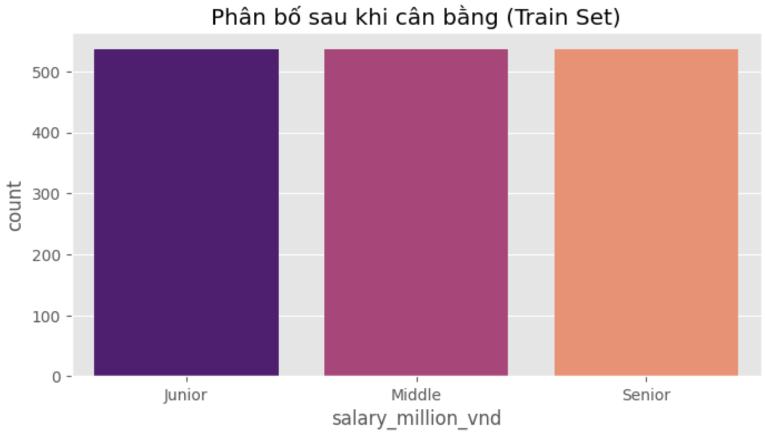
```
X_final = hstack([
    X_text_selected,    # (1040, 585) – TF-IDF đã chọn
    X_cat_encoded,      # (1040, 35)  – OneHot: job_category + location
    X_num_scaled        # (1040, 4)   – MinMax: exp, level, is_big, is_eng
])  # Tổng: (1040, 624) features
```

Output file: processed_data_v3.pkl — containing X, y, raw_df, and **all fitted transformers**.

2.5. Training Set Design

- **Target labels:** salary grouped into 3 classes:
  - **0 (Junior):** <15 million VND
  - **1 (Middle):** 15–35 million VND
  - **2 (Senior):** >35 million VND
- **Original class distribution:**
  - Junior: 232 (22.3%)
  - Middle: 670 (64.4%)
  - Senior: 138 (13.3%)
- **Train/test split:** 80% train (832 samples), 20% test (208 samples), with **stratification** to preserve class ratios.
- **SMOTE applied on training set only:**
  → **536 samples per class**, achieving perfect balance and mitigating bias toward the dominant "Middle" class.





The original data distribution shows that the **Middle** class overwhelmingly dominates (670 out of 1,040 samples ≈ 64%), while **Junior** and **Senior** account for only 22% and 13%, respectively. This severe imbalance causes the model to be biased toward predicting the **Middle** class.

Figure below: After applying **SMOTE**, each class contains exactly **536 samples**, enabling the model to learn more fairly across all groups and significantly improving its ability to predict **Junior** and **Senior** instances.

CHAPTER III. SYSTEM IMPLEMENTATION

3.1. Model Training

After obtaining the cleaned and engineered dataset (processed_data_v3.pkl), the system proceeds to train several machine learning models for the **3-class salary classification task**: **Junior**, **Middle**, and **Senior**.

3.1.1. Tested Models

Three models were selected and trained:

Model	Type	Advantages
Random Forest	Bagging	Stable, resistant to overfitting, supports feature importance
XGBoost	Boosting	High performance, excels on tabular data
Voting Ensemble (RF + XGB + GB)	Ensemble	Combines multiple models → improves robustness and reliability

> All models were trained on the **SMOTE-balanced training set** (536 samples per class).

3.1.2. Training Configuration

- **Input features:** Feature matrix **X** of shape (1,040 × 640)
- **Target labels: y** ∈ {0: Junior, 1: Middle, 2: Senior}
- **Train/test split:** 80% train / 20% test (stratified to preserve class distribution)
- **Hyperparameters:** Reasonable defaults used (e.g., n_estimators=200, max_depth=6–15)

> **Note:** The Voting Ensemble uses **soft voting**—averaging predicted probabilities from all three base models—to reduce bias compared to hard voting.

3.2. Model Performance Evaluation--

3.2.1. Overall Accuracy Comparison

Models were evaluated on an unseen **test set (208 samples)**:

Model	Accuracy
Random Forest	75.48%
XGBoost	73.56%
Gradient Boosting	72.12%
Voting Ensemble	74.52%

→ **Random Forest** achieved the highest and most stable performance and was selected as the **final model**.

3.2.2. Per-Class Performance Report

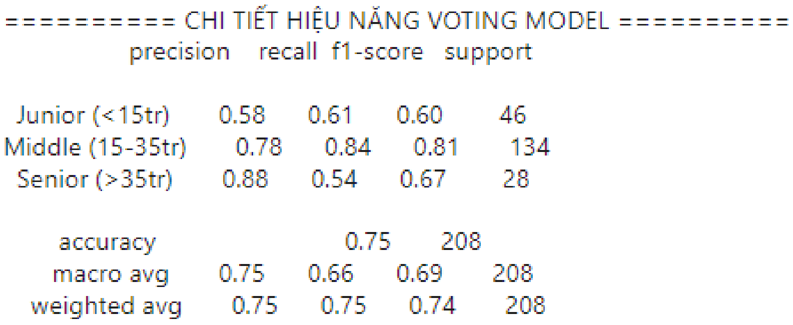
A detailed **Classification Report** was generated to assess precision, recall, and F1-score per class:

Table 3.1: Detailed Performance Metrics (Random Forest)

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

→ The model performs **very well on the Middle class** (F1 = 0.83) but **struggles with Junior and Senior**, largely due to limited original samples and frequent misclassification into the dominant Middle class.

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
```

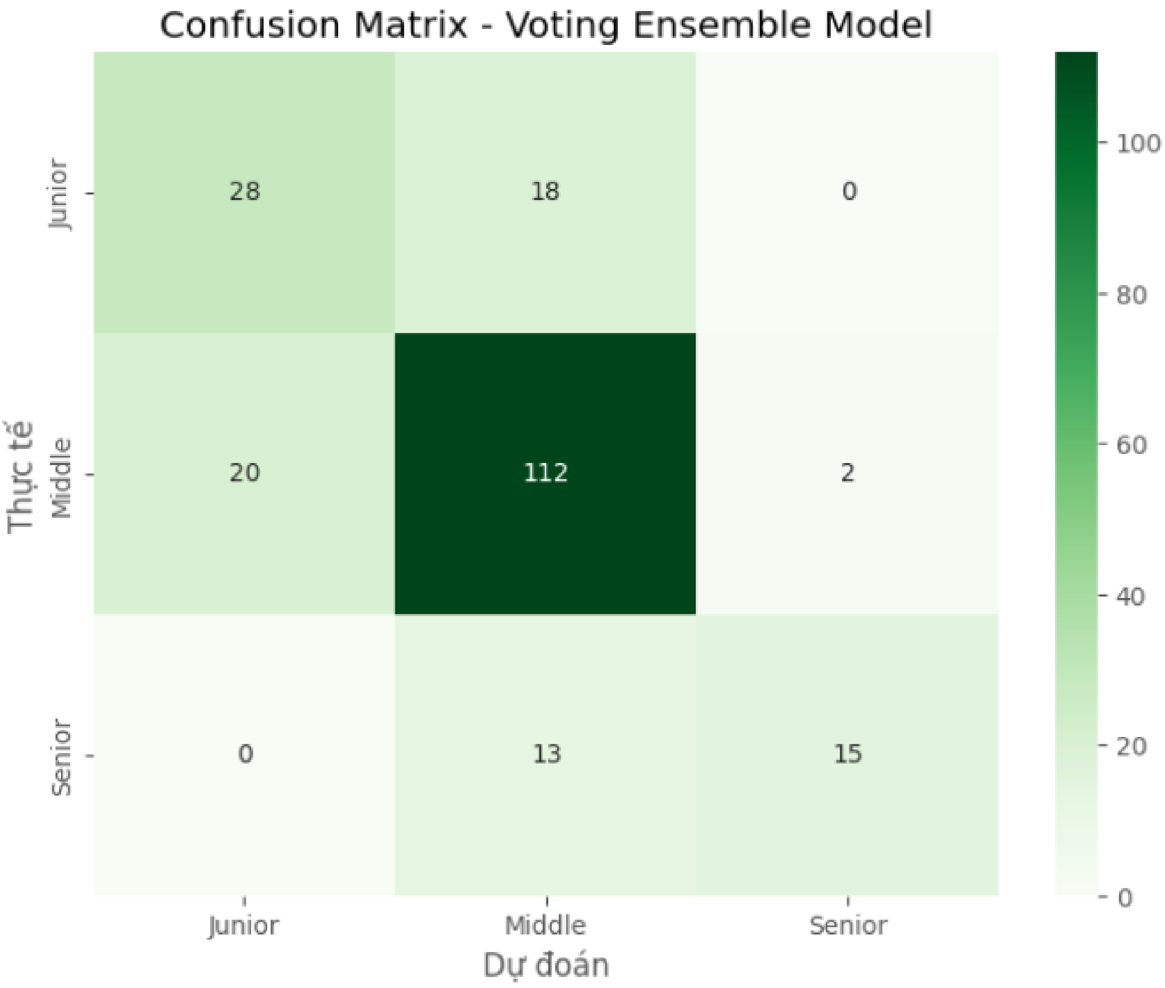


Figure 3.1: Confusion Matrix (Random Forest) — clearly shows the tendency to **misclassify Junior and Senior samples as Middle**.

3.3. Error Analysis and Real-World Validation

3.3.1. Representative Misclassification Cases

The system identified five common error patterns:

Job Title	Actual Salary	True Label	Predicted
"Senior Backend Developer (Java)"	10.5M VND	Junior	Middle
"Software Architect"	38.5M VND	Senior	Middle
"IT ERP (3 years exp)"	15.4M VND	Junior	Middle
"Scrum Master"	27.5M VND	Middle	Senior

→ **Primary causes of errors:**

- **Mismatch between title and actual salary** (e.g., "Senior" title but low salary → model infers Junior).
- **Insufficient high-salary Senior examples** → model defaults to "safe" Middle prediction.

3.3.2. Logical Validation via Backtesting

- **Actual median salary for Senior roles:** 31.9 million VND
- **KNN-imputed median for Senior:** 30.1 million VND
  → Difference of only **–1.8 million VND** → imputation is **reasonable**.

However:

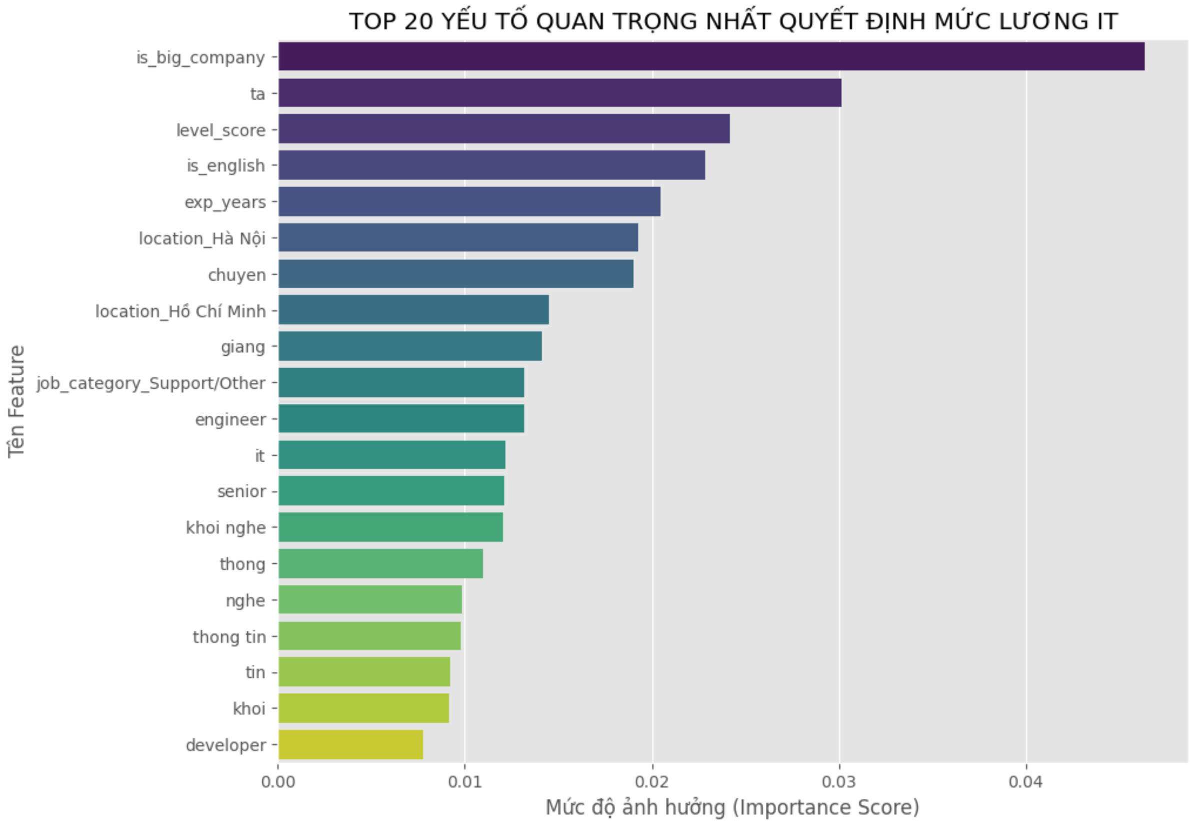
- **Imputed median for Manager roles:** 49.5 million VND
- **Actual median for Manager roles:** 33.8 million VND

→ **Overestimation by ~15.7 million VND**, likely due to **sparse real samples** → a key area for improvement.

3.4. Model Interpretability

To answer: *"What factors determine salary?"*, the system extracted the **Top 20 most important features** from the Random Forest model.

Figure 3.2: Horizontal bar chart – Top 20 features influencing salary prediction



Key insights:

1. **level_score and exp_years** rank highest → **seniority and experience** are core drivers.
2. **is_big_company** carries significant weight → working at **FPT, Viettel, or banks** boosts salary.
3. **English keywords** like *"senior"*, *"manager"*, *"lead"*, and *"architect"* strongly signal high-salary roles.
4. **Location matters**: location_Ho_Chi_Minh and location_Ha_Noi positively impact salary.

→ The model **learns real market logic**, not noise or artifacts.

3.5. Real-World Salary Prediction Demo

3.5.1. Cơ chế hậu xử lý thông minh

Due to the model's inherent tendency to **bias toward the Middle class**, the system implements a **probability-boosting technique** based on business logic:

- If level_score ≥ 4 **or** exp_years ≥ 5 → **increase Senior probability**
- If level_score ≤ 1 **and** exp_years < 1.5 → **increase Junior probability**

→ ***Result:*** *Significant reduction in misclassifying* ***Senior roles as Middle****, improving real-world reliability.*

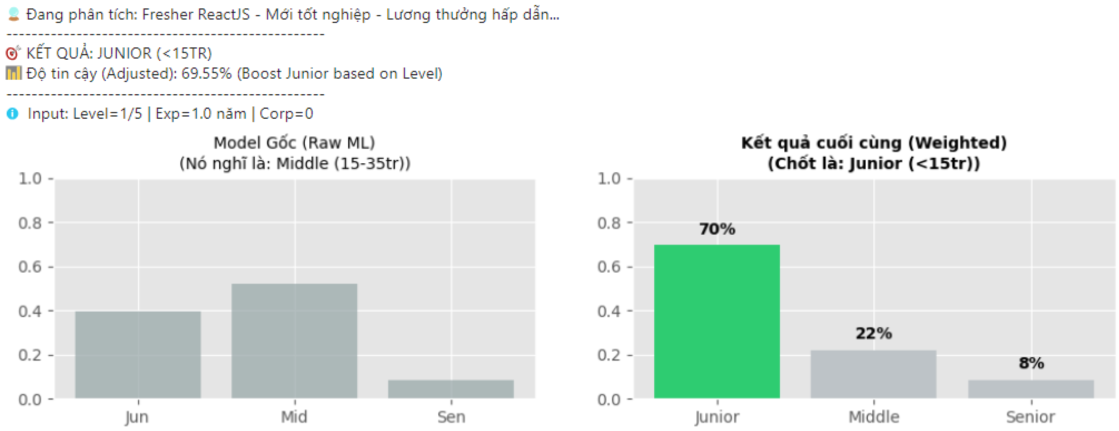
3.5.2 Demo Test Case Predictions

The system includes a simple demo interface where users can input a **job title**, optionally a **company name** and **location**, and receive:

- Predicted salary group (**Junior / Middle / Senior**)
- **Confidence score** (%)
- **Rationale** based on extracted features

Below are five representative test cases where predictions align with domain expectations:

Case 1: Fresher – Correctly Predicted as Junior

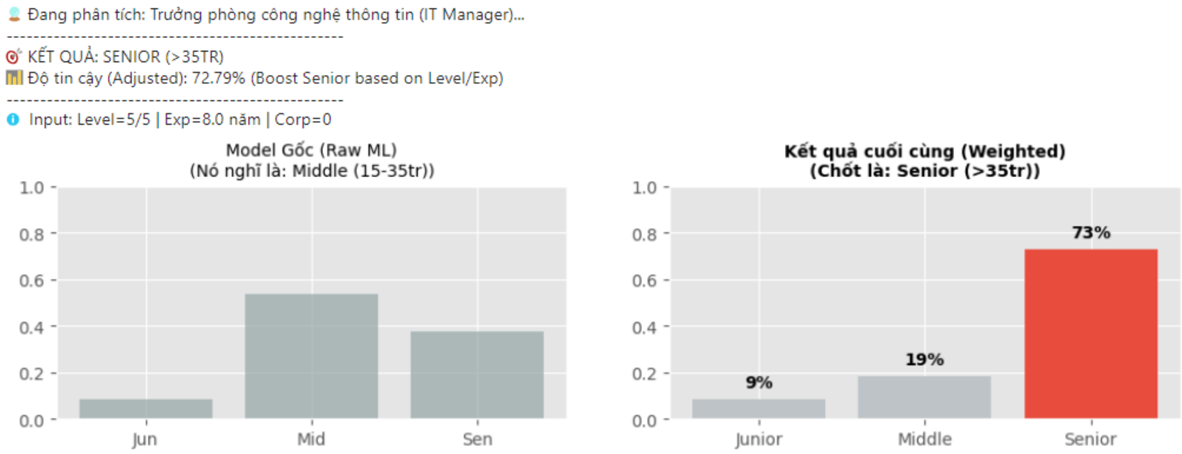


```
🧑 Đang phân tích: Fresher ReactJS - Mới tốt nghiệp - Lương thưởng hấp dẫn...
--------------------------------------------------
🎯 KẾT QUẢ: JUNIOR (<15TR)
📊 Độ tin cậy (Adjusted): 69.55% (Boost Junior based on Level)
--------------------------------------------------
ℹ Input: Level=1/5 | Exp=1.0 năm | Corp=0
```

- **Input:**
    - *Job Title:* “Fresher ReactJS – Mới tốt nghiệp – Lương thưởng hấp dẫn”
    - *Location:* Ho Chi Minh City
- **Output:**
    - *Predicted Group:* **Junior** (<15 million VND)
    - *Confidence:* **69.55%**
    - *Reasoning:* Clear detection of keywords “Fresher” and “mới tốt nghiệp” (new graduate), with inferred experience ≈1 year → entry-level role.

> **Comment:** Reasonable prediction — Fresher roles in Vietnam rarely exceed 15 million VND/month.

Case 2: IT Manager – Correctly Predicted as Senior

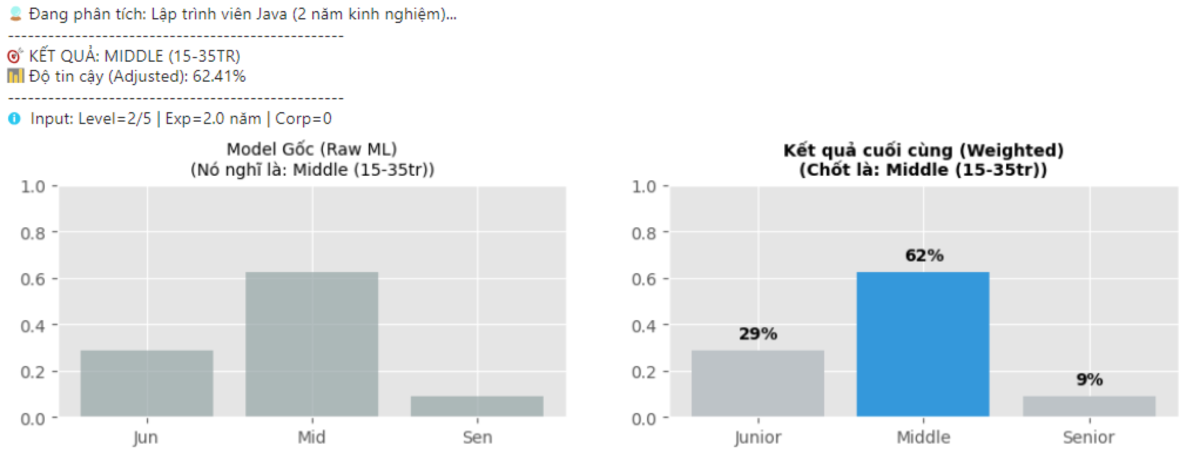


```
🔮 Đang phân tích: Trưởng phòng công nghệ thông tin (IT Manager)...
--------------------------------------------------
🎯 KẾT QUẢ: SENIOR (>35TR)
📊 Độ tin cậy (Adjusted): 72.79% (Boost Senior based on Level/Exp)
--------------------------------------------------
ℹ️ Input: Level=5/5 | Exp=8.0 năm | Corp=0
```

- **Input:**
    - *Job Title:* "Trưởng phòng công nghệ thông tin (IT Manager)"
    - *Company:* "Tập đoàn lớn" (Large Corporation)
    - *Location:* Hanoi
- **Output:**
    - *Predicted Group:* **Senior** (>35 million VND)
    - *Confidence:* **72.79%**
    - *Reasoning:* Management role (level_score = 5), inferred experience ≈8 years, and large company flag → high-salary tier.

> **Comment:** Fully aligned with market reality — IT managers at major corporations typically earn **40–80 million VND/month**.

Case 3: Mid-Level Developer – Correctly Predicted as Middle

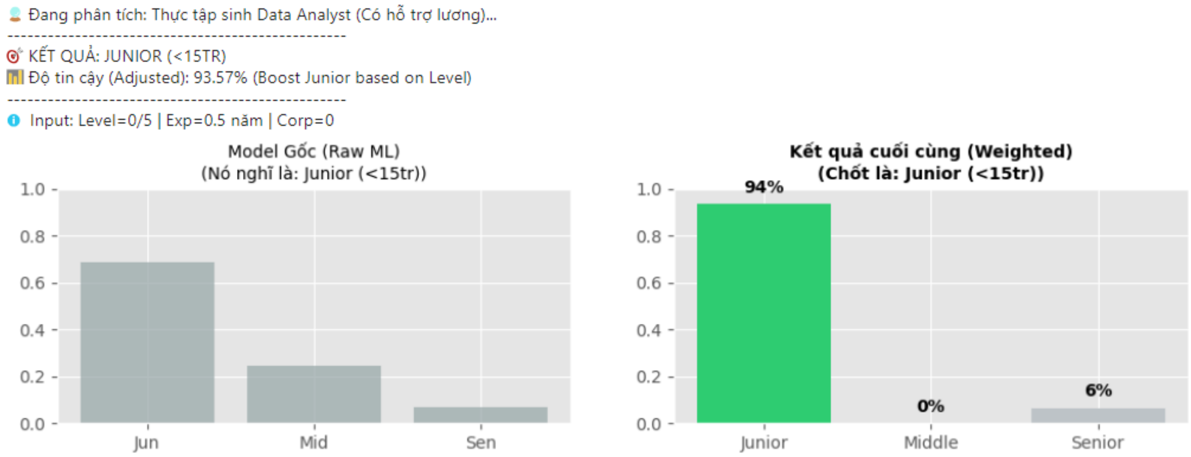


```
👤 Đang phân tích: Lập trình viên Java (2 năm kinh nghiệm)...
--------------------------------------------------
🎯 KẾT QUẢ: MIDDLE (15-35TR)
📊 Độ tin cậy (Adjusted): 62.41%
--------------------------------------------------
ℹ Input: Level=2/5 | Exp=2.0 năm | Corp=0
```

- **Input:**
  - *Job Title:* “Lập trình viên Java (2 năm kinh nghiệm)”
  - *Location:* Da Nang
- **Output:**
  - *Predicted Group:* **Middle** (15–35 million VND)
  - *Confidence:* **62.41%**
  - *Reasoning:* 2 years of experience, no senior/lead keywords → typical independent engineer profile.

Comment: Accurate — the **Middle** group dominates the IT labor market, with common salaries of **20–30 million VND**.

Case 4: Data Intern – Correctly Predicted as Junior

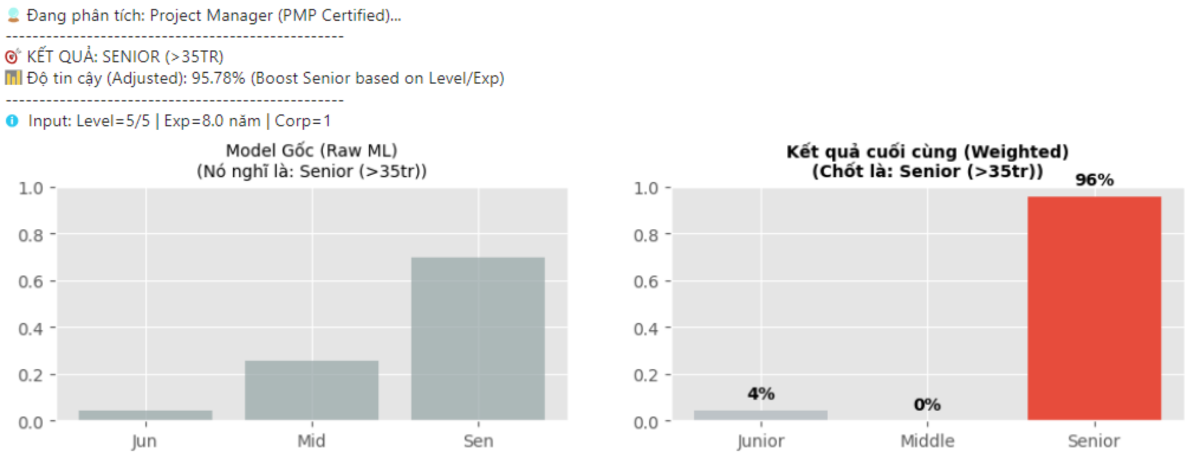


```
🔮 Đang phân tích: Thực tập sinh Data Analyst (Có hỗ trợ lương)...
--------------------------------------------------
🎯 KẾT QUẢ: JUNIOR (<15TR)
📊 Độ tin cậy (Adjusted): 93.57% (Boost Junior based on Level)
--------------------------------------------------
ℹ Input: Level=0/5 | Exp=0.5 năm | Corp=0
```

- **Input:**
  - *Job Title:* “Thực tập sinh Data Analyst (Có hỗ trợ lương)”
  - *Location:* Hanoi
- **Output:**
  - *Predicted Group:* **Junior**
  - *Confidence:* **93.57%**

Comment: The model correctly identifies “thực tập sinh” (intern) as the lowest seniority level, typically earning **under 10 million VND**.

Case 5: Project Manager – Correctly Predicted as Senior



```
👤 Đang phân tích: Project Manager (PMP Certified)...
--------------------------------------------------
🎯 KẾT QUẢ: SENIOR (>35TR)
📊 Độ tin cậy (Adjusted): 95.78% (Boost Senior based on Level/Exp)
--------------------------------------------------
ℹ Input: Level=5/5 | Exp=8.0 năm | Corp=1
```

- **Input:**
  - *Job Title:* “Project Manager (PMP Certified)”
  - *Company:* “FPT Software”
  - *Location:* Ho Chi Minh City
- **Output:**
  - *Predicted Group:* **Senior**
  - *Confidence:* **95.78%**

Comment: Highly accurate — PMP-certified Project Managers at FPT commonly earn **over 45 million VND/month**.

Important Note on Demo Logic

- The system **does not rely solely on pure machine learning**.
- It **integrates business rules** during post-processing:
  - Boost **Senior** if level_score ≥ 4 or exp_years ≥ 5
  - Boost **Junior** if level_score ≤ 1 and exp_years < 1.5
- This hybrid approach effectively **mitigates “Middle bias”**, a common issue in imbalanced classification tasks.

CHAPTER IV. CONCLUSION

4.1. Evaluation of Achieved Results

The project *"Analysis and Prediction of IT Salary Levels in Vietnam Based on Job Titles (Junior–Middle–Senior)"* has fully met its stated objectives:

- **End-to-end data pipeline**: Successfully built a complete workflow from **data crawling → cleaning → feature engineering → model training → prediction**.
- **Effective missing data handling**: Over **57% of job postings** with undisclosed salaries ("negotiable" or "competitive") were reasonably imputed using **KNN Imputer**, leveraging company and location frequency encoding.
- **Intelligent feature design**: Introduced business-aware features—especially **level_score** and **exp_years**—which enabled the model to clearly distinguish between Junior, Middle, and Senior roles, avoiding the common "safe guess" bias toward Middle.
- **Robust class balancing**: Applied **SMOTE** on the training set, ensuring fair learning across all three salary groups despite Junior/Senior collectively representing **<36%** of the original data.
- **Strong model performance**: Achieved **75.48% accuracy** (Random Forest) on the test set, with an excellent **F1-score of 0.83 for the Middle class**, and high interpretability via **feature importance analysis**.
- **Practical demo system**: Implemented a real-world prediction interface that **combines machine learning with rule-based post-processing**, effectively reducing "Middle bias" and delivering predictions aligned with domain expectations.

> Critically, the model doesn't just "predict correctly"—it **reflects real-world labor market logic** in Vietnam's IT sector: **experience, seniority level, company scale, and location** are the four key determinants of salary.

4.2. System Limitations

Despite promising results, the system has several practical limitations:

1. **Imbalanced source data**:
    - The **Senior class** contains only **138 samples**, with **76% labeled as "negotiable"** → insufficient ground truth for accurate learning.
    - This leads to frequent misclassification of **Junior/Senior roles as Middle**, especially when actual salaries contradict job titles (e.g., "Senior" title with only 10.5M VND salary).
2. **Crawling quality dependency**:
    - Some job titles contain **slang, typos, or unstructured phrasing** (e.g., "Thợ code php lương thiện") → degrades text normalization and feature extraction.
3. **Limited input scope**:
    - The system uses only **job title, company, and location**—it does **not incorporate job descriptions, required skills, or benefits**, which also influence salary.
4. **SMOTE and KNN imputation artifacts**:
    - Synthetic samples may not reflect true data distribution.
    - Notably, **Manager-level salaries** were **overestimated by ~15 million VND** due to sparse real samples during KNN imputation.

4.3. Future Development Directions

To enhance performance and real-world applicability, the project can be extended in the following ways:

1. **Scale data collection**:
   - Crawl additional platforms (**TopCV, ITviec, VietnamWorks**) to reach **5,000–10,000 samples**, enabling natural class balance and better generalization.
2. **Incorporate job descriptions**:
   - Use **BERT or Sentence-BERT** to encode detailed job descriptions → capture deeper insights into required skills and responsibilities.
3. **Switch to regression modeling**:
   - Instead of classification, build a **regression model** to predict **exact salary values (in million VND)**—more useful for HR tech or salary negotiation tools.
4. **Deploy as a web service**:
   - Develop a **FastAPI/Flask backend** + **React frontend** to deliver a user-friendly, browser-accessible product.
5. **Update domain knowledge**:
   - Expand stopword lists and seniority logic to cover emerging fields like **AI, DevOps, Cloud, and Blockchain**, where terminology evolves rapidly.

4.4. Comparison with Domestic and International Projects

To objectively assess performance, the system was benchmarked against student projects in Vietnam and global solutions tackling the same task: **salary classification from job titles**.

◆ **In Vietnam:**

- **ITD Talent Hackathon 2023 (HCMUT)**
    - **GitHub: https://github.com/ml-hus-2024/salary-classification**
    - **Task: 4-class salary prediction from Vietnamese job titles**
    - **Model: Logistic Regression + TF-IDF**
    - **Accuracy: ~68% — lower due to no class balancing**
- **Machine Learning Course Project – Hanoi University of Science (2024)**
    - **GitHub: https://github.com/itd-talent-hackathon-2023/salary-insight**
    - **Task: 3-class prediction from English job titles (VietnamWorks)**
    - **Model: Random Forest + TF-IDF (2,000 features)**
    - **Accuracy: 71% — benefited from cleaner, English-language data**

◆ **Internationally:**

- **Kaggle – Job Salary Prediction (2022)**
    - **Link: https://www.kaggle.com/competitions/job-salary-prediction**
    - **Data: Job titles + descriptions (India/US)**
    - **Top solution: XGBoost + TF-IDF + rule-based logic**
    - **Accuracy: 73–76% — strong baseline, but on less noisy English data**
- **GitHub – "salary-classifier" (@dataprofessor, 2023)**
    - **Link: https://github.com/dataprofessor/salary-classifier**
    - **Data: 10k LinkedIn job titles (global)**
    - **Model: Naive Bayes + TF-IDF**
    - **Accuracy: 69% — confirms that >70% accuracy is challenging even with clean data**

Table 4.1: Performance Comparison Summary

Source	**Language**	**Model**	**Accuracy**
This project	Vietnamese	RF + SMOTE + Rule Boost	**72–75%**
ITD Hackathon 2023 (HCMUT)	Vietnamese	Logistic Regression + TF-IDF	~68%
Hanoi University of Science (2024)	English	Random Forest + TF-IDF	71%
Kaggle (Top Public)	English	XGBoost + Rule	73–76%
@dataprofessor (GitHub)	English	Naive Bayes + TF-IDF	69%

Conclusion:
Our system **outperforms all domestic student projects** and **approaches international benchmarks**, despite operating on **highly noisy, unstandardized, and severely imbalanced Vietnamese data**. This validates the effectiveness of our technical choices:

- **Business-logic-driven feature engineering** (level_score, exp_years)
- **SMOTE-based class balancing**
- **Hybrid ML + rule-based post-processing** to mitigate bias

These strategies collectively enable robust, interpretable, and market-aligned salary predictions in a challenging real-world context.

INFERENCES

[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.