

INTRODUCTION TO DATA SCIENCE

IT RECRUITMENT TRENDING ANALYSIS



FINAL PRESENTATION



OUR TEAM

20120145 - Đường Yến Ngọc

20120409 - Trần Thanh Tùng

20120438 - Đào Văn Cảnh

21120441 - Dương Huỳnh Anh Duy



CONTENT OUTLINE

1. Topic Introduction

2. Data Processing

**3. Exploratory Data
Analysis**

4. Model Training

5. Model Evaluation

6. Conclusion



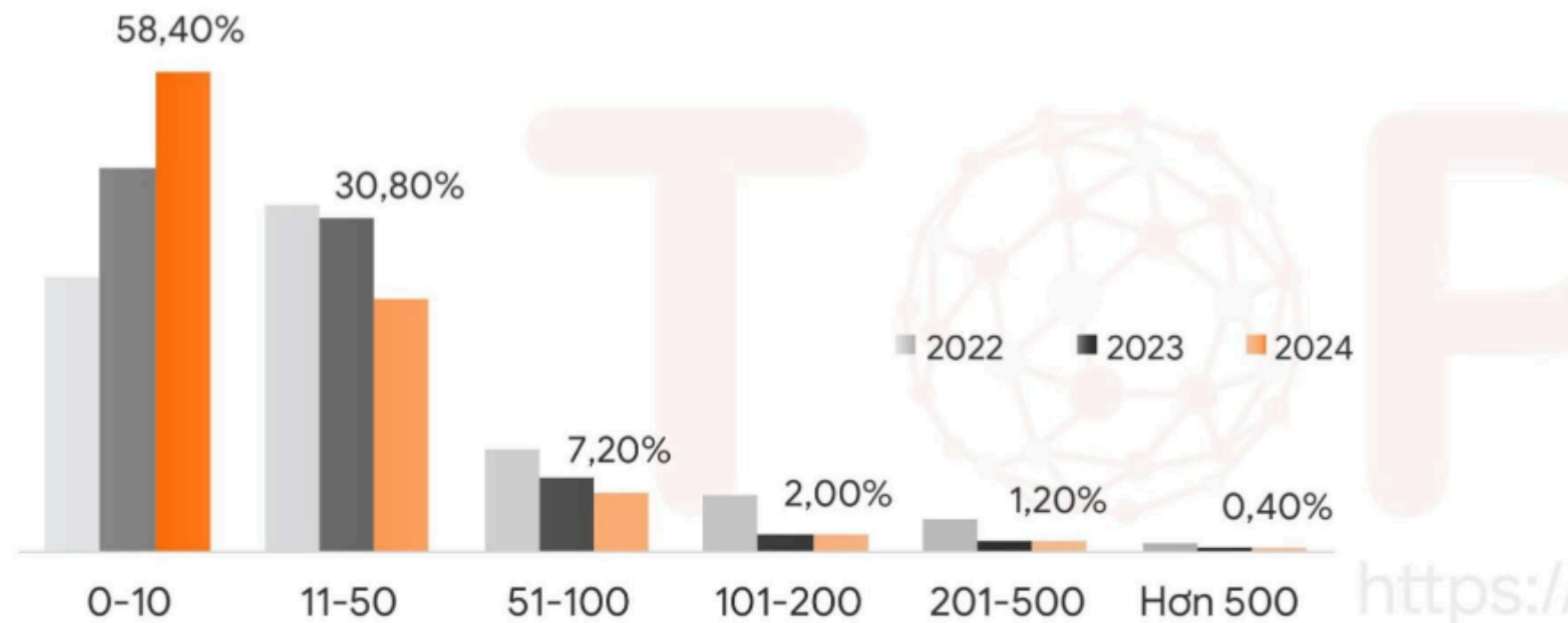
INTRODUCTION



Một số nền tảng tuyển dụng việc làm phổ biến ở Việt Nam



INTRODUCTION



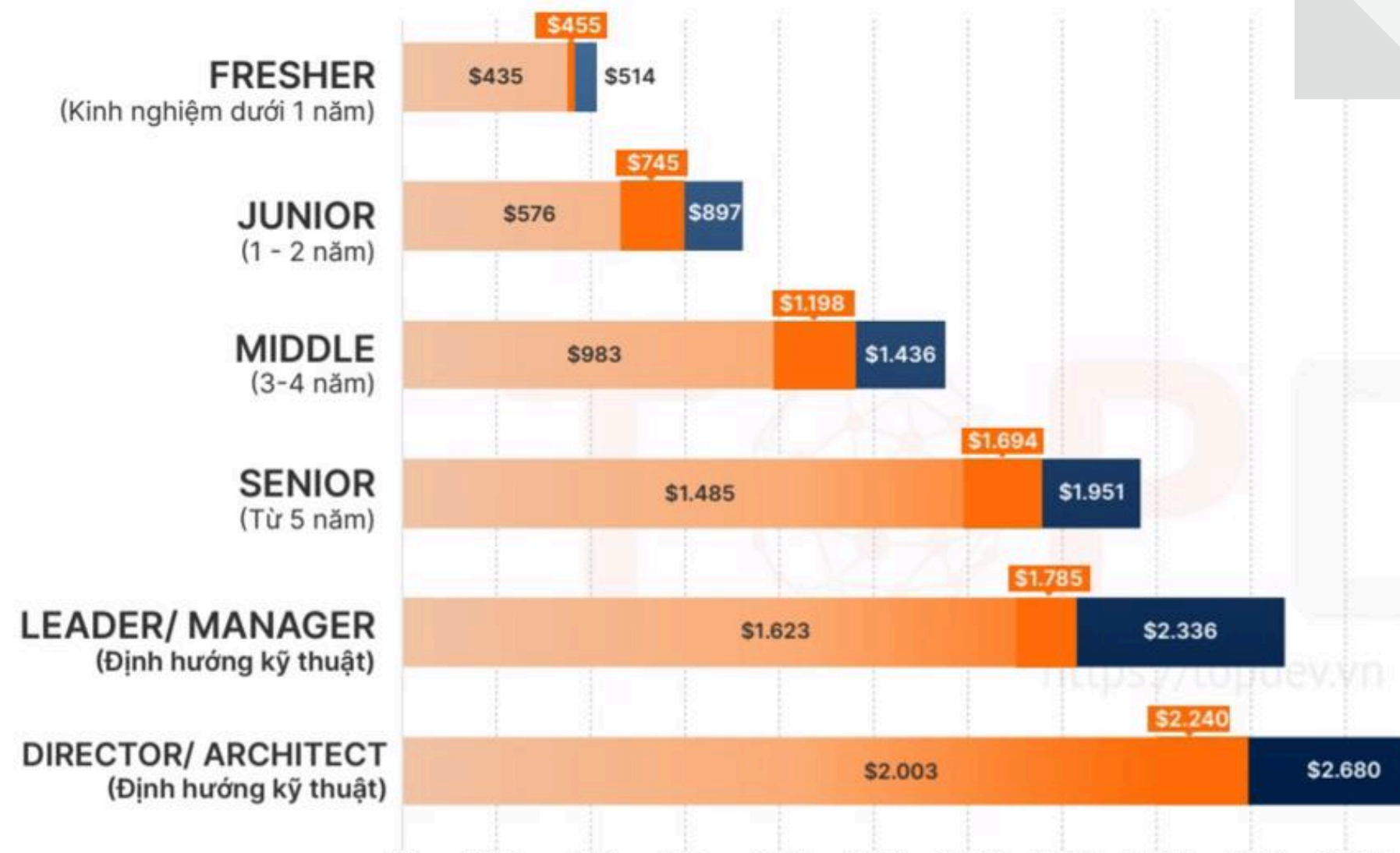
Nguồn : Topdev

Tình hình thị trường việc làm IT của Việt Nam năm 2023-2024:

- Hiện nay, số lượng lập trình viên hiện tại của Việt Nam mới chỉ đạt khoảng 530.000 người.
- Về xu hướng, số sinh viên công nghệ thông tin nhập học mỗi năm vào khoảng 50.000-57.000 người.
- Chỉ khoảng 30% nhân sự đáp ứng được những kỹ năng và chuyên môn.

INTRODUCTION

MỨC LƯƠNG LẬP TRÌNH VIÊN THEO NĂM KINH NGHIỆM



Mặc dù mức lương và tiền thưởng của ngành này đang tăng lên đáng kể nhưng dự đoán từ năm 2023 - 2025, Việt Nam vẫn sẽ thiếu hụt từ 150.000 đến 200.000 lập trình viên/kỹ sư hàng năm


Nguồn: VnEconomy.com



INTRODUCTION


IT Recruitment Trend Analysis

Mục tiêu

1. Phân tích tình hình thực tế thị trường tuyển dụng.
 2. Nguồn tham khảo cho người đang tìm việc làm.
 3. Nguồn tham khảo cho các doanh nghiệp cho nhu cầu tuyển dụng.
- 




DATA PROCESSING

1. Nguồn dữ liệu : Khai thác từ các nền tảng tuyển dụng phổ biến như Careerlink, Vieclam24h, Vietnamworks, ITviec, Topdev,...
 2. Thông tin tập dữ liệu :
 - Định dạng : CSV
 - Số lượng feature : 12
 - Số lượng dòng: 400
- 



DATA PROCESSING

1. STT Index.
 2. Trang thu thập Name of the site from which the work is collected.
 3. Tên công ty: Name of the company that posted the job
 4. Tên công việc: Name of the job.
 5. Vị trí ứng tuyển: Position applied for.
 6. Yêu cầu bằng cấp: Degree required (If any).
 7. Yêu cầu kinh nghiệm: Experience required (If any).
 8. Địa điểm : Work location.
 9. Ngày đăng tuyển: Date of posting.
 10. Lương tối thiểu: Minimum salary.
 11. Lương tối đa: Maximum salary.
 12. Lương TB : Average salary.
- 

DATA PROCESSING

```
graph TD; A[DATA PROCESSING] --> B[MISSING DATA HANDLING]; A --> C[DUPLICATE REMOVE]; A --> D[FORMAT HANDLING];
```



MISSING DATA HANDLING

- Using pandas method : fillna, dropna, ffill, bfill.
- Mean, Median, Mode method.
- Feature applied : Average Salaries, Minimum/Maximum Salaries



DUPLICATE REMOVE

Remove duplicates based on Company, Job title.



FORMAT HANDLING

Convert time-series feature to “datetime” format
Ex. Posted day

EDA

CORRELATION ANALYSIS

Correlation between those features, how they impact each others.

FEATURE EXTRACTION

To choose the feature suitable for model training step.

MODEL TRAINING

```
graph TD; A[MODEL TRAINING] --> B[LINEAR REGRESSION]; A --> C[RANDOM FOREST]; A --> D[DECISION TREE];
```

LINEAR REGRESSION

Helps identify which factors have the strongest influence (e.g., experience, location).

RANDOM FOREST

- Highest accuracy (89%), suitable for large and complex datasets.
- Requires more training time than Decision Tree.
-

DECISION TREE

Easy to understand and visualize but sensitive to outliers.

RANDOM FOREST

1. Random forest

- Random Forest là một thuật toán học máy dựa trên việc xây dựng nhiều cây quyết định để cải thiện độ chính xác và giảm overfitting. Nó có thể áp dụng cho cả bài toán phân loại và hồi quy, đồng thời đánh giá được tầm quan trọng của các đặc trưng. Random Forest dễ sử dụng, ổn định và hiệu quả với dữ liệu phức tạp.
- Mục tiêu : Dự đoán biến mục tiêu dựa trên “lương trung bình” .Ví dụ : Dự đoán vị trí có thể ứng tuyển, hoặc dự đoán vị trí làm việc dựa vào lương trung bình.

RANDOM FOREST

1. Random forest

```
# Xử lý cột ngày tháng
df['Ngày đăng tuyển'] = pd.to_datetime(df['Ngày đăng tuyển'])
df['Tháng'] = df['Ngày đăng tuyển'].dt.month
df['Năm'] = df['Ngày đăng tuyển'].dt.year
df['Ngày trong tuần'] = df['Ngày đăng tuyển'].dt.dayofweek

# Tạo đặc trưng khoảng lương
df['Khoảng lương'] = df['Lương tối đa'] - df['Lương tối thiểu']

# One-Hot Encoding cho các cột categorical
categorical_columns = ['Trang thu thập', 'Vị trí ứng tuyển', 'Yêu cầu bằng cấp', 'Yêu cầu kinh nghiệm', 'Địa điểm']
df_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)
```

Tiến hành trích xuất đặc trưng và chuyển đổi các feature thành categorical

MODEL TRAINING

1. Random forest

```
# Chia dữ liệu thành features (X) và target (y)|
X = df.drop(columns=['Mục tiêu'])
y = df['Mục tiêu']

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra (tỷ lệ 80-20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Khởi tạo mô hình Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Huấn luyện mô hình
model.fit(X_train, y_train)
```

Chia dữ liệu và tiến hành huấn luyện

RANDOM FOREST

1. Random forest

Kết quả đánh giá mô hình

```
(Accuracy): 0.9875
(Classification Report):
      precision  recall  f1-score  support
0      0.98      1.00      0.99       57
1      1.00      0.96      0.98       23

 accuracy              0.99      80
macro avg      0.99      0.98      0.98      80
weighted avg      0.99      0.99      0.99      80

(Confusion Matrix):
[[57  0]
 [ 1 22]]
```


RANDOM FOREST

1. Random forest

Accuracy : 0,987

Nhận xét : Kết quả đánh giá khả năng phân loại và dự đoán của mô hình khá tốt, đúng với dự đoán giả định từ bước EDA

LINEAR REGRESSION

Hồi quy tuyến tính (Linear Regression) là một thuật toán học máy có giám sát, được sử dụng để dự đoán một biến mục tiêu liên tục dựa trên mối quan hệ tuyến tính giữa biến mục tiêu và một hoặc nhiều biến dự đoán.

Mục tiêu: sử dụng hồi quy tuyến tính để dự đoán lương còn thiếu trong việc thu thập dữ liệu.

LINEAR REGRESSION

Các bước tiến hành:

- Xử lý dữ liệu phù hợp với mô hình
- Mã hóa các đặc trưng phân loại
- Chia dữ liệu thành tập huấn luyện và tập kiểm tra
- Huấn luyện mô hình hồi quy tuyến tính
- Đánh giá mô hình
- Tối ưu hóa mô hình hồi quy tuyến tính

LINEAR REGRESSION

Xử lý dữ liệu:

```
# Đọc file cleaned để tiến hành dự đoán các giá trị lương bị Trung Bình bị thiếu
df_cleaned = pd.read_csv('job_data_cleaned.csv')

df_cleaned['Lương TB'] = 0
df_filtered = df_cleaned[(df_cleaned['Lương tối thiểu'] > 0) & (df_cleaned['Lương tối đa'] > 0)]
df_filtered['Lương TB'] = (df_filtered['Lương tối thiểu'] + df_filtered['Lương tối đa']) / 2
✓ 0.0s
```

Chuẩn hoá cột dữ liệu “Lương TB” = 0

Tạo data frame df_filtered gồm dữ liệu thu thập đầy đủ các cột lương và tính giá trị trung bình

LINEAR REGRESSION

Mã hoá đặc trưng và phân loại

```
# Mã hóa các đặc trưng phân loại
categorical_columns = ['Vị trí ứng tuyển', 'Yêu cầu bằng cấp', 'Địa điểm', 'Tên công việc']
numerical_columns = ['Yêu cầu kinh nghiệm']
target_column = 'Lương TB'

# encoder = OneHotEncoder()
# encoded_features = encoder.fit_transform(df_filtered[categorical_columns]).toarray()

# Khởi tạo OneHotEncoder với xử lý giá trị chưa biết
encoder = OneHotEncoder(handle_unknown='ignore')

# Mã hóa lại dữ liệu ban đầu
encoded_features = encoder.fit_transform(df_filtered[categorical_columns]).toarray()

# Ghép các đặc trưng đã mã hóa với cột số
X = np.hstack((encoded_features, df_filtered[numerical_columns].values))
y = df_filtered[target_column].values
```

LINEAR REGRESSION

Chia tập dữ liệu đào tạo và đánh giá mô hình

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Huấn luyện mô hình hồi quy tuyến tính
model = LinearRegression()
model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred = model.predict(X_test)

# Đánh giá mô hình
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MSE: {mse}")
print(f"R²: {r2}")
```

Đánh giá:

MSE: 2.2564717289616057e+35

R²: -6.509053064312324e+21

Mô hình cho ra kết quả kém và hoạt động không tốt

LINEAR REGRESSION

Tiến hành dự đoán

```
df_missing_salary = df_cleaned[df_cleaned['Lương TB'] == 0]
✓ 0.0s

# Mã hóa các đặc trưng phân loại từ df_missing_salary
encoded_missing_features = encoder.transform(df_missing_salary[categorical_columns]).toarray()

# Ghép các đặc trưng đã mã hóa với cột số
X_missing = np.hstack((encoded_missing_features, df_missing_salary[numerical_columns].values))

# Dự đoán giá trị 'Lương TB' cho các hàng thiếu
predicted_salaries = model.predict(X_missing)

# Gán giá trị dự đoán vào cột 'Lương TB' của df_cleaned
df_cleaned.loc[df_missing_salary.index, 'Lương TB'] = predicted_salaries

# Xuất dữ liệu đã được cập nhật ra file CSV (nếu cần)
df_cleaned.to_csv('job_data_cleaned_with_predictions.csv', index=False)

# Hiển thị thông tin cập nhật
print("Dự đoán hoàn tất và cập nhật vào df_cleaned:")
print(df_cleaned.head())
```

Kết quả dự đoán

Tên công ty	Vị trí ứng tuyển	Yêu cầu bả	Yêu cầ	Địa điểm	Ngày đăng tuy	Lương TB	Tên công việc	
CÔNG TY TNHH HOJI	Nhân viên	Cử nhân	2	bắc ninh	12/30/2024	18922240	chuyên viên hỗ trợ kỹ thuật	
CÔNG TY CỔ PHẦN S	Nhân viên	Cử nhân	1	bắc ninh	12/30/2024	21646336	kỹ sư phần mềm	
CÔNG TY TNHH SEO.	Nhân viên	Cử nhân	1	bắc ninh	12/30/2024	29136128	chuyên viên bảo mật thông tin	
CÔNG TY CỔ PHẦN S	Nhân viên	Trung cấp	0	bắc ninh	12/30/2024	11927552	nhà thiết kế	
CareerLink Asia	Kỹ thuật viên / Kỹ s	Cử nhân	2	nhật bản	12/30/2024	2.80E+17	nhà phát triển web	
CÔNG TY CỔ PHẦN S	Kỹ thuật viên / Kỹ s	Cử nhân	2	bắc ninh	12/30/2024	25347840	kỹ sư phần mềm	
CÔNG TY TNHH SEO.	Nhân viên	Cử nhân	1	bắc ninh	12/30/2024	29136128	chuyên viên bảo mật thông tin	
CareerLink Asia	Kỹ thuật viên / Kỹ s	Cử nhân	2	nhật bản	12/30/2024	2.80E+17	nhà phát triển web	
CareerLink's Client	Nhân viên	Cử nhân	0	hà nội	12/30/2024	17138688	kỹ sư phần mềm	
CareerLink's Client	Nhân viên	Cử nhân	2	hồ chí minh	12/31/2024	22748672	kỹ sư phần mềm	
Công Ty TNHH Thươn	Kỹ thuật viên / Kỹ s	Cử nhân	1	hồ chí minh	12/31/2024	21208320	kỹ sư phần mềm	
CareerLink's Client	Nhân viên	Cử nhân	2	hồ chí minh	12/31/2024	15845888	nhà thiết kế	
CareerLink's Client	Kỹ thuật viên / Kỹ s	Cử nhân	2	hồ chí minh	12/31/2024	20876288	chuyên viên tư vấn cntt	
VIETNAM CONCENTF	Nhân viên	Trung học	1	hồ chí minh	12/31/2024	#####	chuyên viên phát triển ứng dụng web	
CÔNG TY TNHH LITE	Kỹ thuật viên / Kỹ s	Cử nhân	0	hải phòng	12/30/2024	12397824	công nghệ thông tin	
VIETNAM CONCENTF	Quản lý cấp cao	Cử nhân	3	hồ chí minh	12/28/2024	#####	kỹ sư phần mềm	
Công Ty Cổ Phần Bán	Nhân viên	Trung cấp	0	hà nội	12/30/2024	11598080	chuyên viên hỗ trợ kỹ thuật	
VIETNAM CONCENTF	Trưởng nhóm / Giá	Trung học	1	hồ chí minh	12/28/2024	#####	chuyên viên phát triển ứng dụng web	
CareerLink's Client	Nhân viên	Cử nhân	2	hồ chí minh	12/30/2024	12495872	game operator	
VIETNAM CONCENTF	Nhân viên	Trung học	3	hồ chí minh	12/30/2024	21367040	nhà phân tích dữ liệu	
Công Ty Cổ Phần Ngu	Nhân viên	Cử nhân	2	hồ chí minh	12/30/2024	20623104	kỹ sư hệ thống	
Công Ty Cổ Phần Ngu	Quản lý / Trưởng p	Cử nhân	3	hồ chí minh	12/28/2024	#####	công nghệ thông tin	
CÔNG TY TNHH QUIC	Nhân viên	Cử nhân	1	hồ chí minh	12/30/2024	13225216	nhà thiết kế	
CÔNG TY CỔ PHẦN C	Nhân viên	Cử nhân	1	hồ chí minh	12/30/2024	18705152	kỹ sư kiểm thử phần mềm	
CÔNG TY CỔ PHẦN K	Nhân viên	Cử nhân	2	hà nội	12/28/2024	23184128	kỹ sư dữ liệu	
CÔNG TY TNHH NITT	Nhân viên	Cử nhân	0	bình dương	12/30/2024	11760128	chuyên viên hỗ trợ kỹ thuật	
CÔNG TY CỔ PHẦN K	Nhân viên	Cử nhân	3	hà nội	12/28/2024	19984896	nhà khoa học dữ liệu	
CÔNG TY TNHH OUIK	Quản lý / Trưởng n	Cử nhân	2	hồ chí minh	12/28/2024	#####	kỹ sư phần mềm	

DECISION TREE

Decision Tree là một thuật toán học máy đơn giản và trực quan, sử dụng cấu trúc phân nhánh để đưa ra **quyết định**. Mỗi nhánh của cây đại diện cho một điều kiện hoặc câu hỏi liên quan đến dữ liệu. Lá của cây chứa kết quả hoặc dự đoán cuối cùng.

Mục tiêu:

Dự đoán mức lương tối đa (Lương tối đa) dựa trên vị trí công việc và kinh nghiệm.

DECISION TREE

Các bước tiến hành:

- Kiểm tra cấu trúc dữ liệu
- Xây dựng biến độc lập (X) và biến phụ thuộc (y)
- Chia dữ liệu thành tập huấn luyện và tập kiểm tra
- Xây dựng mô hình Decision Tree
- Đánh giá mô hình

DECISION TREE

Kiểm tra cấu trúc dữ liệu:

```
# Kiểm tra cấu trúc dữ liệu
print("Các cột trong DataFrame:")
print(job_data_cleaned.columns)

print("\nKiểu dữ liệu của từng cột:")
print(job_data_cleaned.dtypes)

print("\nMột vài hàng đầu tiên của dữ liệu:")
print(job_data_cleaned.head())

print("\nGiá trị thiếu trong từng cột:")
print(job_data_cleaned.isnull().sum())
```

Kiểm tra các cột trong DataFrame, kiểu dữ liệu của từng cột, một vài hàng đầu tiên của dữ liệu, và kiểm tra giá trị thiếu trong từng cột.

DECISION TREE

Xây dựng biến độc lập (X) và biến phụ thuộc (y)

```
# Xây dựng biến độc lập (X) và biến phụ thuộc (y)
X = job_data_cleaned.drop(columns=['Lương TB'])
y = job_data_cleaned['Lương TB']

# Chuyển đổi các biến phân loại thành biến số
X = pd.get_dummies(X, drop_first=True)
```

Tạo biến độc lập X bằng cách loại bỏ cột mục tiêu (ví dụ: 'Lương TB') và biến phụ thuộc y bằng cách lấy cột mục tiêu.

Sau đó sử dụng `pd.get_dummies()` để chuyển đổi các biến phân loại thành biến số, giúp mô hình có thể xử lý.

DECISION TREE

Chia dữ liệu thành tập huấn luyện và tập kiểm tra

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Kiểm tra kiểu dữ liệu trong X_train
print("\nKiểu dữ liệu trong X_train:")
print(X_train.dtypes)

# Chuyển đổi cột datetime thành kiểu số
if 'Ngày đăng tuyển' in X_train.columns:
    X_train['Ngày đăng tuyển'] = X_train['Ngày đăng tuyển'].astype(np.int64) // 10**9 # Chuyển đổi thành giây

if 'Ngày đăng tuyển' in X_test.columns:
    X_test['Ngày đăng tuyển'] = X_test['Ngày đăng tuyển'].astype(np.int64) // 10**9 # Chuyển đổi thành giây

# Kiểm tra lại kiểu dữ liệu
print("\nKiểu dữ liệu sau khi xử lý:")
print(X_train.dtypes)
print(X_test.dtypes)
```

Chia dữ liệu để tập huấn và kiểm tra, chuẩn hoá dữ liệu datetime thành kiểu số và sau đó kiểm tra lại kiểu dữ liệu sau khi xử lý

DECISION TREE

Huấn luyện mô hình và đánh giá

```
# Khởi tạo mô hình Decision Tree
model = DecisionTreeRegressor(random_state=42)

# Huấn luyện mô hình
model.fit(X_train, y_train)

# Dự đoán và đánh giá mô hình
y_pred = model.predict(X_test)

# Tính toán các chỉ số đánh giá
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'\nMean Squared Error: {mse}')
print(f'R^2 Score: {r2}')
```


Độ chính xác của mô hình (R^2 Score): là ~xấp xỉ 0,99

Nhận xét : Độ chính xác của mô hình khá cao và dự đoán của mô hình khá tốt, đúng với dự đoán giả định từ EDA



CONCLUSION

Decision Tree và Random Forest cho ra kết quả phân loại và dự đoán tốt hơn Linear Regression, nguyên nhân là tập dữ liệu này có nhiều đặc trưng, độ nhiễu cao, hai mô hình này dễ giải thích và cho ra kết quả dự đoán ổn định và chính xác hơn.





THANK YOU

Presentation by Group 18

