BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC KINH TẾ TP. HỒ CHÍ MINH

ĐỒ ÁN MÔN HỌC

LẬP TRÌNH PHÂN TÍCH DỮ LIỆU

PHÂN TÍCH BỘ DỮ LIỆU HR ANALYTICS

Họ và tên	**MSSV**	**Lớp**
Vũ Nguyễn Thảo Vi	31211027686	DS001
Nguyễn Quốc Việt	31211027687	DS001
Nguyễn Thanh Vy	31211027689	DS002
Nguyễn Nhật Thảo Vy	31211025542	DS001

GVHD: TS. Nguyễn An Tế

TP. Hồ Chí Minh, Ngày 19 tháng 12 năm 2023

Mục lục

Danh sách hình vẽ

Danh sách bảng

1 CHƯƠNG 1: TỔNG QUAN

1.1 Vai trò và ý nghĩa của lập trình phân tích dữ liệu

Dữ liệu là nguồn tài nguyên quan trọng và có giá trị trong thời đại số hóa ngày nay. Nó cho ta cái nhìn rõ hơn về vấn đề mà chúng ta quan tâm, tìm ra những thông tin hữu ích và hỗ trợ việc đưa ra quyết định chính xác hơn. Tuy nhiên, để khai thác được tiềm năng của dữ liệu, chúng ta cần có những phương pháp thu thập, xử lý và phân tích dữ liệu một cách chuyên nghiệp và hiệu quả. Vì vậy, kỹ năng lập trình phân tích dữ liệu giữ vai trò rất quan trọng trong nhiều lĩnh vực, đặc biệt là đối với ngành khoa học dữ liệu.

Lập trình phân tích dữ liệu là quá trình sử dụng các ngôn ngữ lập trình, các công cụ và thư viện để thu thập, xử lý, phân tích và mô hình hóa dữ liệu. Đồng thời những nhà phân tích có thể áp dụng các kỹ thuật như thống kê, trực quan hóa, hay học máy để đo lường kết quả, kiểm tra giả thuyết hay dự báo xu hướng trong lĩnh vực mà họ nghiên cứu. Có thể nói, lập trình phân tích dữ liệu là kỹ năng không thể thiếu đối với bất kỳ ai muốn bắt kịp sự phát triển nhanh chóng của thời đại công nghệ thông tin.

1.2 Giới thiệu đề tài

Một tổ chức có thể hoạt động tốt hay không phần lớn dựa vào khả năng của những người làm việc cho tổ chức đó. Từ lâu, các doanh nghiệp đã hiểu rõ rằng sức mạnh cạnh tranh của họ chủ yếu đến từ con người, và việc quản trị nhân sự nội bộ, giảm thiểu sự hao mòn nhân sự cũng là một yếu tố chiến lược quan trọng. Bối cảnh nền kinh tế đầy biến động đã đẩy không ít doanh nghiệp vào tình thế phải đưa ra nhiều quyết định quan trọng về quản trị nhân sự. Việc quản lý và duy trì được những nhân viên giỏi giúp công ty giảm được chi phí đào tạo cũng như giữ được sự ổn định cho hoạt động của họ. Bằng việc phân tích bộ dữ liệu chứa các thông tin cơ bản về nhân viên cùng những các thang đo có khả năng ảnh hưởng đến quyết định rời bỏ của họ, nhóm mong muốn đưa ra những phân tích mang lại giá trị đối với lĩnh vực quản trị nhân sự.

1.3 Mục đích đề tài

Mục tiêu của đề tài này là khai thác, xử lý và trình bày các dữ liệu liên quan đến nhân sự của một công ty. Bên cạnh một số phân tích về nhân khẩu học, đồ án đặc biệt quan tâm đến quyết định ở lại hay rời bỏ công ty của nhân viên. Bằng cách sử dụng các công cụ và phương pháp phân tích dữ liệu, kết hợp với các phương pháp thống kê và mô hình máy học, nhóm mong muốn có thể đưa ra những hiểu biết cơ bản về xu hướng của nguồn nhân sự này. Từ đó tìm hiểu về nguyên nhân, giải pháp hoặc đề xuất dành cho những vấn đề về quản trị nhân lực cho tổ chức, doanh nghiệp.

1.4 Tổng quan bộ dữ liệu

Bộ dữ liệu tổng hợp bao gồm 35 thuộc tính, 1477 quan sát.

STT	Tên thuộc tính	Mô tả	Ghi chú
1	Age	Tuổi của nhân viên	
2	Attrition	Nhân viên rời đi hoặc không	Target
3	BusinessTravel	Tần suất đi công tác của nhân viên	
4	DailyRate	Tiền lương theo ngày của nhân viên	
5	Department	Phòng ban của nhân viên	
6	DistanceFromHome	Khoảng cách từ nhà đến nơi làm việc	
7	Education	Trình độ học vấn của nhân viên	
8	EducationField	Lĩnh vực học thuật của nhân viên	
9	EmployeeCount	Số thứ tự của nhân viên	
10	EmployeeNumber	Mã số nhân viên	
11	EnvironmentSatisfaction	Độ hài lòng về môi trường làm việc của nhân viên	
12	Gender	Giới tính của nhân viên	
13	HourlyRate	Tiền lương theo giờ của nhân viên	
14	JobInvolvement	Mức độ tham gia công việc của nhân viên	
15	JobLevel	Cấp độ công việc của nhân viên	
16	JobRole	Vị trí công việc của nhân viên	
17	JobSatisfaction	Độ hài lòng về công việc của nhân viên	
18	MaritalStatus	Trình trạng hôn nhân	
19	MonthlyIncome	Thu nhập hàng tháng của nhân viên	
20	MonthlyRate	Tiền lương theo tháng của nhân viên	
21	NumCompaniesWorked	Số lượng công ty mà nhân viên đã làm việc trước đây	
22	Over18	Trên 18 tuổi hoặc không	

23	OverTime	Tăng ca hoặc không
24	PercentSalaryHike	Chỉ số tăng lương theo %
25	PerformanceRating	Đánh giá hiệu suất của nhân viên
26	RelationshipSatisfaction	Độ hài lòng về các mối quan hệ của nhân viên
27	StandardHours	Số giờ làm tiêu chuẩn mỗi tuần của nhân viên
28	StockOptionLevel	Mức độ lựa chọn cổ phiếu công ty của nhân viên
29	TotalWorkingYears	Tổng số năm làm việc của nhân viên
30	TrainingTimesLastYear	Thời lượng tập huấn trong năm
31	WorkLifeBalance	Mức độ cân bằng cuộc sống và công việc
32	YearsAtCompany	Số năm làm việc tại công ty
33	YearsInCurrentRole	Số năm làm việc ở vị trí hiện tại
34	YearsSinceLastPromotion	Số năm kể từ lần thăng chức cuối cùng
35	YearsWithCurrManager	Số năm làm việc với người quản lý hiện tại

Bảng 1: Bảng mô tả các thuộc tính bộ dữ liệu HR Analytics

2 CHƯƠNG 2: TIỀN XỬ LÍ DỮ LIỆU

2.1 Vai trò của tiền xử lí trong phân tích dữ liệu

Tiền xử lý dữ liệu là một bước quan trọng trong quá trình phân tích dữ liệu, đặc biệt là khi sử dụng các kỹ thuật học máy. Tiền xử lý dữ liệu bao gồm các hoạt động như làm sạch, xử lý giá trị bị thiếu, chuẩn hóa, biến đổi, xử lý ngoại lai và rút trích dữ liệu để làm cho chúng phù hợp với mục đích phân tích. Tiền xử lý dữ liệu có vai trò quyết định đến chất lượng và hiệu quả của các mô hình học máy, vì nếu dữ liệu không được tiền xử lý tốt, thì kết quả phân tích sẽ không chính xác và tin cậy. Tiền xử lý dữ liệu cũng giúp giảm thiểu thời gian và chi phí tính toán, cũng như tăng khả năng khai thác và hiểu biết về dữ liệu.

Mỗi bộ dữ liệu khác nhau, chúng ta sẽ đối mặt với những vấn đề khác nhau khi tiền xử lý. Vì vậy, tùy đặc điểm của bộ dữ liệu mà chúng ta có thể chọn những kỹ thuật và trình tự tiền xử lý khác nhau cho phù hợp.

2.2 Quan sát sơ bộ bộ dữ liệu.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1477 entries, 0 to 1476
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1477 non-null   int64
 1   Attrition                 1477 non-null   object
 2   BusinessTravel            1477 non-null   object
 3   DailyRate                 1477 non-null   int64
 4   Department                1476 non-null   object
 5   DistanceFromHome          1471 non-null   float64
 6   Education                 1477 non-null   int64
 7   EducationField            1474 non-null   object
 8   EmployeeCount             1477 non-null   int64
 9   EmployeeNumber            1467 non-null   float64
 10  EnvironmentSatisfaction   1477 non-null   int64
 11  Gender                    1477 non-null   object
 12  HourlyRate                1470 non-null   float64
 13  JobInvolvement            1476 non-null   float64
 14  JobLevel                  1477 non-null   int64
 15  JobRole                   1476 non-null   object
 16  JobSatisfaction           1474 non-null   float64
 17  MaritalStatus             1476 non-null   object
 18  MonthlyIncome             1473 non-null   float64
 19  MonthlyRate               1477 non-null   int64
 20  NumCompaniesWorked        1476 non-null   float64
 21  Over18                    1477 non-null   object
 22  OverTime                  1477 non-null   object
 23  PercentSalaryHike         1476 non-null   float64
 24  PerformanceRating         1477 non-null   int64
 25  RelationshipSatisfaction  1477 non-null   int64
 26  StandardHours             1477 non-null   int64
 27  StockOptionLevel          1477 non-null   int64
 28  TotalWorkingYears         1477 non-null   int64
 29  TrainingTimesLastYear     1477 non-null   int64
 30  WorkLifeBalance           1477 non-null   int64
 31  YearsAtCompany            1477 non-null   int64
 32  YearsInCurrentRole        1477 non-null   int64
 33  YearsSinceLastPromotion   1477 non-null   int64
 34  YearsWithCurrManager      1473 non-null   float64
dtypes: float64(9), int64(17), object(9)
memory usage: 404.0+ KB
```

2.2.1 Xoá cột

Xóa những cột thuộc các trường hợp sau:

- Chỉ có duy nhất một giá trị: 'EmployeeCount', 'Over18', 'StandardHours'
- Không có ý nghĩa trong việc giải thích biến target: 'EmployeeNumber'
- Không xác định được định nghĩa chính xác của biến: 'MonthlyRate'

Nguyên nhân xóa những cột thuộc 2 trường hợp đầu tiên là vì chúng không cung cấp thông tin có giá trị cho những phân tích về bộ dữ liệu. Đối với cột 'MonthlyRate', không tìm được định nghĩa chính xác và phù hợp với giá trị của biến nên xóa để tránh những nhận định sai lệch về dữ liệu khi không có đủ hiểu biết chính xác về ý nghĩa của biến này.

```
df.columns[df.nunique()==1]
```

```
Index(['EmployeeCount','Over18','StandardHours'], dtype='object')
```

```
df = df.drop(columns= ['EmployeeCount', 'Over18', 'StandardHours',
↪  'EmployeeNumber', 'MonthlyRate'])
```

2.2.2 Phân loại biến

Chia các biến thành biến định lượng, định tính (num) và biến định danh (cat).

```
num = df.select_dtypes(exclude='O')
cat = df.select_dtypes(include='O')

print(num.columns, num.shape[1])
print(cat.columns, cat.shape[1])
```

```
Index(['Age', 'DailyRate', 'DistanceFromHome', 'Education',
       'EnvironmentSatisfaction', 'HourlyRate', 'JobInvolvement', 'JobLevel',
       'JobSatisfaction', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
       'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object') 22
Index(['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender',
       'JobRole', 'MaritalStatus', 'OverTime'],
      dtype='object') 8
```

Sau đó tiếp tục chia những biến định lượng thành biến theo thang đo thứ bậc (interval_cols) và biến theo thang đo khoảng (ordinal_cols)

```
interval_cols = ['EnvironmentSatisfaction', 'JobInvolvement',
 ↪  'JobSatisfaction',
                  'PerformanceRating', 'RelationshipSatisfaction',
                   ↪  'WorkLifeBalance']
ordinal_cols = ['Education', 'JobLevel', 'StockOptionLevel']
```

Việc phân loại các biến giúp cho các phân tích trên từng phân loại biến dễ dàng hơn.

2.2.3 Chuyển kiểu dữ liệu

Chuyển kiểu dữ liệu của các biến từ dạng số thực về dạng số nguyên cho phù hợp với giá trị thực tế quan sát từ các biến.

```
#Chuyển kiểu dữ liệu
cols_to_convert = ['JobInvolvement', 'JobSatisfaction', 'NumCompaniesWorked',
 ↪  'YearsWithCurrManager']
df[cols_to_convert] = df[cols_to_convert].astype('Int64')
```

Xem miền giá trị của các biến định lượng, định tính và các lớp giá trị của biến định danh

```
#Biến định lượng, định tính
for col in num:
  print(f"Khoảng giá trị của cột {col}: {df[col].min()} đến {df[col].max()}")
```

```
Khoảng giá trị của cột Age: 18 đến 60
Khoảng giá trị của cột DailyRate: 102 đến 1499
Khoảng giá trị của cột DistanceFromHome: 1.0 đến 29.0
Khoảng giá trị của cột Education: 1 đến 5
Khoảng giá trị của cột EnvironmentSatisfaction: 1 đến 4
Khoảng giá trị của cột HourlyRate: 30.0 đến 100.0
Khoảng giá trị của cột JobInvolvement: 1 đến 4
Khoảng giá trị của cột JobLevel: 1 đến 5
Khoảng giá trị của cột JobSatisfaction: 1 đến 4
Khoảng giá trị của cột MonthlyIncome: 1009.0 đến 19999.0
Khoảng giá trị của cột NumCompaniesWorked: 0 đến 9
Khoảng giá trị của cột PercentSalaryHike: 11.0 đến 25.0
Khoảng giá trị của cột PerformanceRating: 3 đến 4
Khoảng giá trị của cột RelationshipSatisfaction: 1 đến 4
Khoảng giá trị của cột StockOptionLevel: 0 đến 3
Khoảng giá trị của cột TotalWorkingYears: 0 đến 40
Khoảng giá trị của cột TrainingTimesLastYear: 0 đến 6
Khoảng giá trị của cột WorkLifeBalance: 1 đến 4
Khoảng giá trị của cột YearsAtCompany: 0 đến 40
Khoảng giá trị của cột YearsInCurrentRole: 0 đến 18
Khoảng giá trị của cột YearsSinceLastPromotion: 0 đến 15
Khoảng giá trị của cột YearsWithCurrManager: 0 đến 17
```

```python
#Biến định danh
for col in cat:
  print(f"Các lớp giá trị của cột {col}: {df[col].unique()}")
```

```
Các lớp giá trị của cột Attrition: ['Yes' 'No']
Các lớp giá trị của cột BusinessTravel: ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
Các lớp giá trị của cột Department: ['Sales' 'Research & Development' 'Human Resources' nan]
Các lớp giá trị của cột EducationField: ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources' nan]
Các lớp giá trị của cột Gender: ['Female' 'Male']
Các lớp giá trị của cột JobRole: ['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources' nan
 'Laboratory ']
Các lớp giá trị của cột MaritalStatus: ['Single' 'Married' 'Divorced' nan]
Các lớp giá trị của cột OverTime: ['Yes' 'No']
```

2.3 Xử lí dữ liệu bị thiếu

Tạo hàm `check` để xem số quan sát, số lượng unique value và số lượng cùng với tỷ lệ dữ liệu bị thiếu của từng thuộc tính.

```python
def check(data):
    list=[]
    for col in data.columns:
        columns = data.columns
        dtype = data[col].dtypes
        instances = data[col].count()
        unique = data[col].nunique()
        sum_null = data[col].isnull().sum()
        misssing_rate = round(sum_null/ instances *100,2)
```

```
        list.append([dtype,instances,unique,sum_null, misssing_rate])
    data_check =
     ↪  pd.DataFrame(list,columns=["Type","Instances","Unique","Missing",
     ↪  'Missing rate'],index=data.columns)
    return data_check

check(df)
```

Attribute	Data Type	Instances	Unique	Missing Rate (%)
Age	int64	1477	43	0.00
Attrition	object	1477	2	0.00
BusinessTravel	object	1477	3	0.00
DailyRate	int64	1477	886	0.00
Department	object	1476	3	0.07
DistanceFromHome	float64	1471	29	0.41
Education	int64	1477	5	0.00
EducationField	object	1474	6	0.20
EmployeeCount	int64	1477	1	0.00
EmployeeNumber	float64	1467	1460	0.68
EnvironmentSatisfaction	int64	1477	4	0.00
Gender	object	1477	2	0.00
HourlyRate	float64	1470	71	0.48
JobInvolvement	float64	1476	4	0.07
JobLevel	int64	1477	5	0.00
JobRole	object	1476	10	0.07
JobSatisfaction	float64	1474	4	0.20
MaritalStatus	object	1476	3	0.07
MonthlyIncome	float64	1473	1346	0.27
MonthlyRate	int64	1477	1427	0.00
NumCompaniesWorked	float64	1476	10	0.07
Over18	object	1477	1	0.00
OverTime	object	1477	2	0.00
PercentSalaryHike	float64	1476	15	0.07
PerformanceRating	int64	1477	2	0.00
RelationshipSatisfaction	int64	1477	4	0.00
StandardHours	int64	1477	1	0.00

Bảng 2: Bảng thống kê giá trị bị thiếu của các thuộc tính

Xem những biến có dữ liệu bị thiếu thuộc nhóm biến định lượng, định tính và nhóm biến định danh

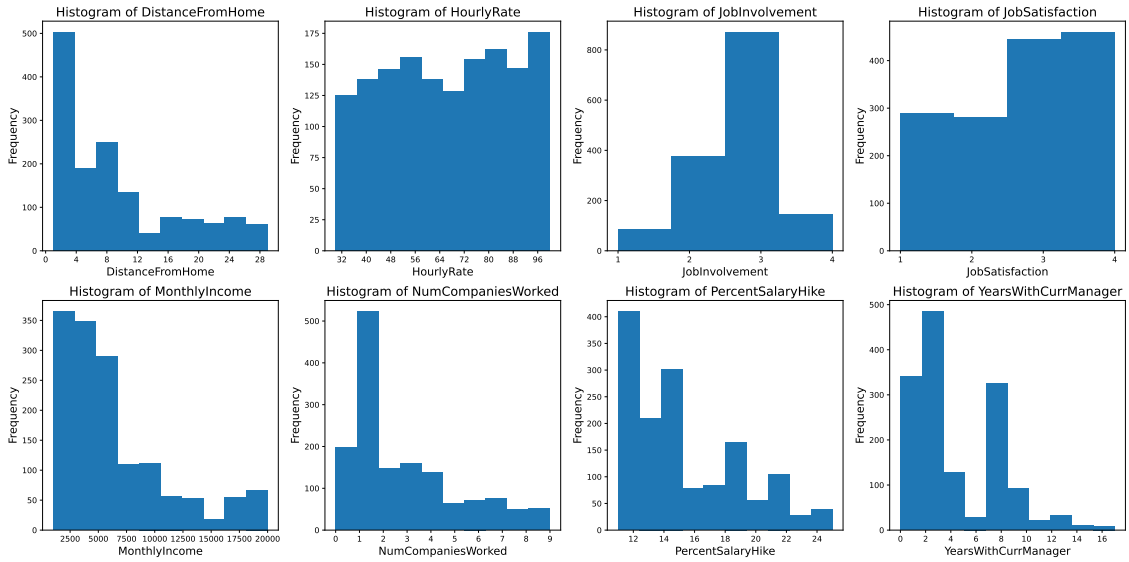
```
cols_with_missing = df[[col for col in df.columns if df[col].isnull().any()]]


num_with_missing = cols_with_missing.select_dtypes(exclude='O')
cat_with_missing = cols_with_missing.select_dtypes(include='O')


print(f'Biến định lượng, định tính: {",
↪  ".join(num_with_missing.columns.values)}\nSố lượng:
↪  {num_with_missing.shape[1]}')
print(f'Biến định danh: {", ".join(cat_with_missing.columns.values)}\nSố lượng:
↪  {cat_with_missing.shape[1]}')
```

```
Index(['DistanceFromHome', 'HourlyRate', 'JobInvolvement', 'JobSatisfaction',
       'MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike',
       'YearsWithCurrManager'],
      dtype='object') 8
Index(['Department', 'EducationField', 'JobRole', 'MaritalStatus'], dtype='object') 4
```

Xem phân phối của các biến định lượng, định tính.

```
fig, axs = plt.subplots(2, 4, figsize=(20, 10))


for i, col in enumerate(num_with_missing.columns):
    ax = axs[i//4, i%4]
    if col in ['JobInvolvement', 'JobSatisfaction']:
        bins = 4
    else:
        bins = None
    num_with_missing[col].hist(ax=ax, bins=bins)
    ax.set_title(f'Histogram of {col}', fontsize = 16)
    ax.set_xlabel(col, fontsize =14)
    ax.set_ylabel('Frequency', fontsize =14)
    ax.xaxis.set_major_locator(MaxNLocator(integer=True))
    ax.grid(False)
```

```
16
17
18 plt.tight_layout()
19 plt.savefig(f"figs/Histogram cols with missing value.pdf")
20 plt.show()
```



Hình 1: Histogram các biến có missing data

```
1  #Biến phân loại điền mode
2  for col in cat_with_missing.columns:
3      df[col].fillna(df[col].mode()[0], inplace=True)
4
5
6  #Biến định lượng điền median (không có phân phối chuẩn, dùng median thay mean)
7  exclude_cols = ['JobInvolvement', 'JobSatisfaction']
8
9
10 for col in num_with_missing.columns:
11     if col not in exclude_cols:
12         df[col].fillna(math.ceil(df[col].median()), inplace=True)
13
14
15 #JobInvolvement, JobSatisfaction điền mode
16 for col in exclude_cols:
17     df[col].fillna(df[col].mode()[0], inplace=True)
```

Đối với các biến phân loại, và biến rời rạc, điền giá trị yếu vị vì đây là phương pháp

phổ biến. Với các biến định lượng còn lại, vì phân phối của chúng không đối xứng nên điền median thay vì mean, do trung vị phù hợp hơn trong trường hợp này và nó ít bị ảnh hưởng bởi outliers hơn.

2.4 Xử lí dữ liệu bị trùng lắp

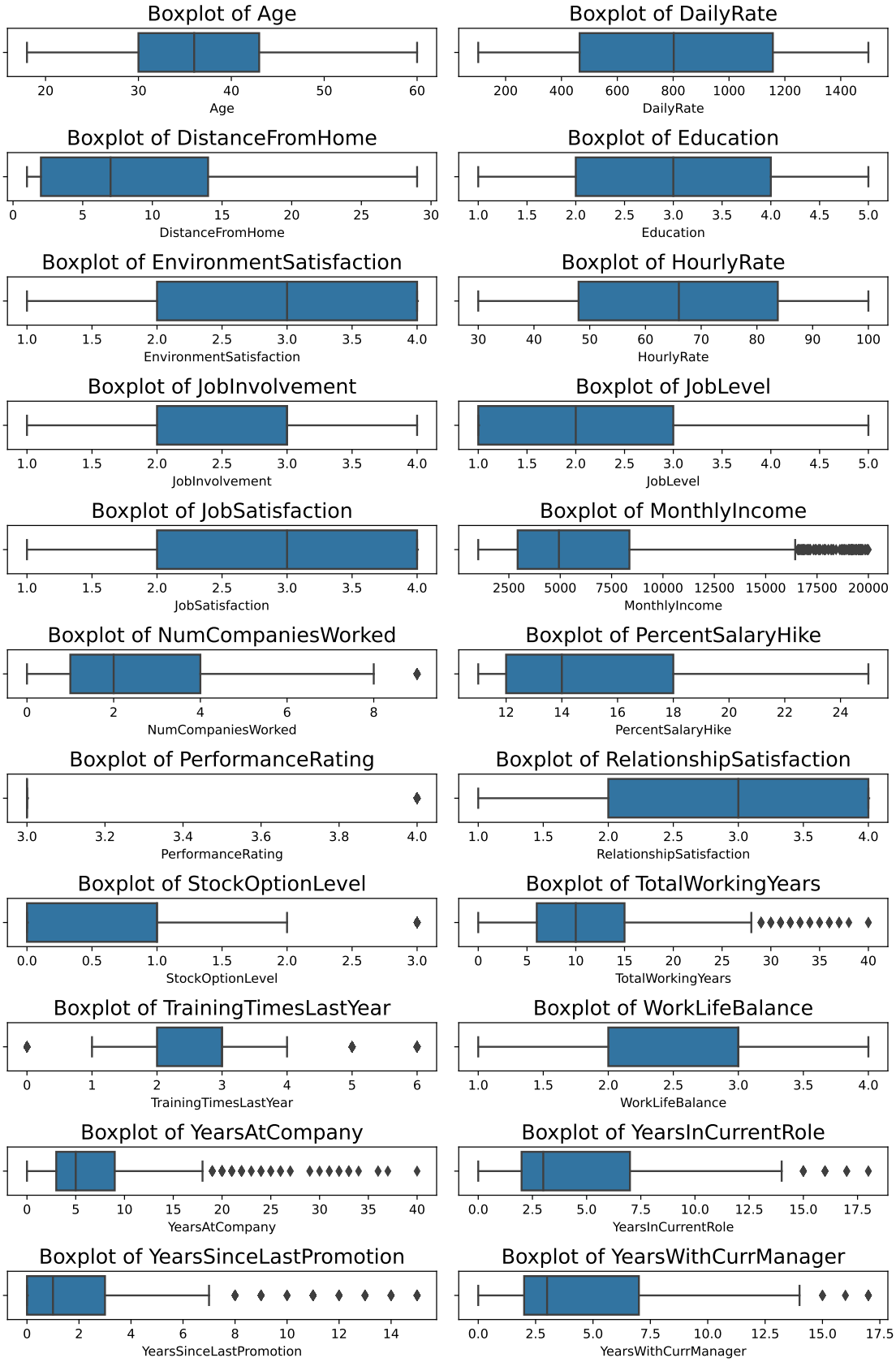
Xóa các dòng dữ liệu bị trùng lặp để đảm bảo tính chính xác cho dữ liệu.

```
#Số dòng bị trùng
df.duplicated().sum()
#Xóa các dòng bị trùng lắp
df = df.drop_duplicates()
```

2.5 Xử lí ngoại lai (Outliers)

Dùng phương pháp trực quan là xem boxplot của các biến định lượng, định tính để quan sát outliers có thể có của từng biến. Tuy nhiên, nếu dữ liệu càng bị lệch thì càng có nhiều quan sát bị nhận diện là outliers.

```
fig, axs = plt.subplots(11, 2, figsize=(10, 15))
axs = axs.flatten()  # Flatten the axes array to iterate over it easily

for i, column in enumerate(num.columns):
    sns.boxplot(x=df[column], ax=axs[i])
    axs[i].set_title(f'Boxplot of {column}', fontsize=16)

plt.tight_layout()
plt.savefig(f"figs/Boxplot num cols.pdf")
plt.show()
```



Hình 2: Boxplots các biến định lượng, định tính

Từ kết quả quan sát các boxplots, xác định được những biến xuất hiện outliers gồm: 'MonthlyIncome', 'NumCompaniesWorked', 'PerformanceRating', 'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager'.

Vì boxplot dựa trên phương pháp IQR nên tiếp tục dùng IQR để xem xét kỹ hơn các outliers. Phương pháp này phù hợp với dữ liệu không tuân theo phân phối chuẩn và ít nhạy cảm với các giá trị cực đoan.

```
#IQR
def detect_outliers_iqr(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    outliers = data[(data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 *IQR))]
    return outliers


for column in num.columns:
  outliers = detect_outliers_iqr(num[column])
  print(f"Outliers cột {column}: {np.sort(outliers.unique())}  Số lượng:
   ↪  {len(outliers)} ({(len(outliers)/ len(num[column]))* 100:.2f}%)")
```

```
Outliers cột Age: []  Số lượng: 0 (0.00%)
Outliers cột DailyRate: []  Số lượng: 0 (0.00%)
Outliers cột DistanceFromHome: []  Số lượng: 0 (0.00%)
Outliers cột Education: []  Số lượng: 0 (0.00%)
Outliers cột EnvironmentSatisfaction: []  Số lượng: 0 (0.00%)
Outliers cột HourlyRate: []  Số lượng: 0 (0.00%)
Outliers cột JobInvolvement: []  Số lượng: 0 (0.00%)
Outliers cột JobLevel: []  Số lượng: 0 (0.00%)
Outliers cột JobSatisfaction: []  Số lượng: 0 (0.00%)
Outliers cột MonthlyIncome: [16555. 16595. 16598. 16606. 16627. 16659. 16704. 16752. 16756. 16792.
 16799. 16823. 16835. 16856. 16872. 16880. 16885. 16959. 17007. 17046.
 17048. 17068. 17099. 17123. 17159. 17169. 17174. 17181. 17328. 17399.
 17426. 17444. 17465. 17567. 17584. 17603. 17639. 17650. 17665. 17779.
 17856. 17861. 17875. 17924. 18041. 18061. 18172. 18200. 18213. 18265.
 18300. 18303. 18430. 18606. 18665. 18711. 18722. 18740. 18789. 18824.
 18844. 18880. 18947. 19033. 19038. 19045. 19049. 19068. 19081. 19094.
 19141. 19144. 19161. 19187. 19189. 19190. 19197. 19202. 19232. 19237.
 19246. 19272. 19328. 19331. 19392. 19406. 19419. 19431. 19436. 19502.
 19513. 19517. 19537. 19545. 19566. 19586. 19613. 19626. 19627. 19636.
 19658. 19665. 19701. 19717. 19740. 19833. 19845. 19847. 19926. 19943.
 19973. 19999.]  Số lượng: 114 (7.76%)
Outliers cột NumCompaniesWorked: [9]  Số lượng: 52 (3.54%)
Outliers cột PercentSalaryHike: []  Số lượng: 0 (0.00%)
Outliers cột PerformanceRating: [4]  Số lượng: 226 (15.37%)
Outliers cột RelationshipSatisfaction: []  Số lượng: 0 (0.00%)
Outliers cột StockOptionLevel: [3]  Số lượng: 85 (5.78%)
Outliers cột TotalWorkingYears: [29 30 31 32 33 34 35 36 37 38 40]  Số lượng: 63 (4.29%)
Outliers cột TrainingTimesLastYear: [0 5 6]  Số lượng: 238 (16.19%)
Outliers cột WorkLifeBalance: []  Số lượng: 0 (0.00%)
Outliers cột YearsAtCompany: [19 20 21 22 23 24 25 26 27 29 30 31 32 33 34 36 37 40]  Số lượng: 104 (7.07%)
Outliers cột YearsInCurrentRole: [15 16 17 18]  Số lượng: 21 (1.43%)
Outliers cột YearsSinceLastPromotion: [ 8  9 10 11 12 13 14 15]  Số lượng: 107 (7.28%)
Outliers cột YearsWithCurrManager: [15 16 17]  Số lượng: 14 (0.95%)
```

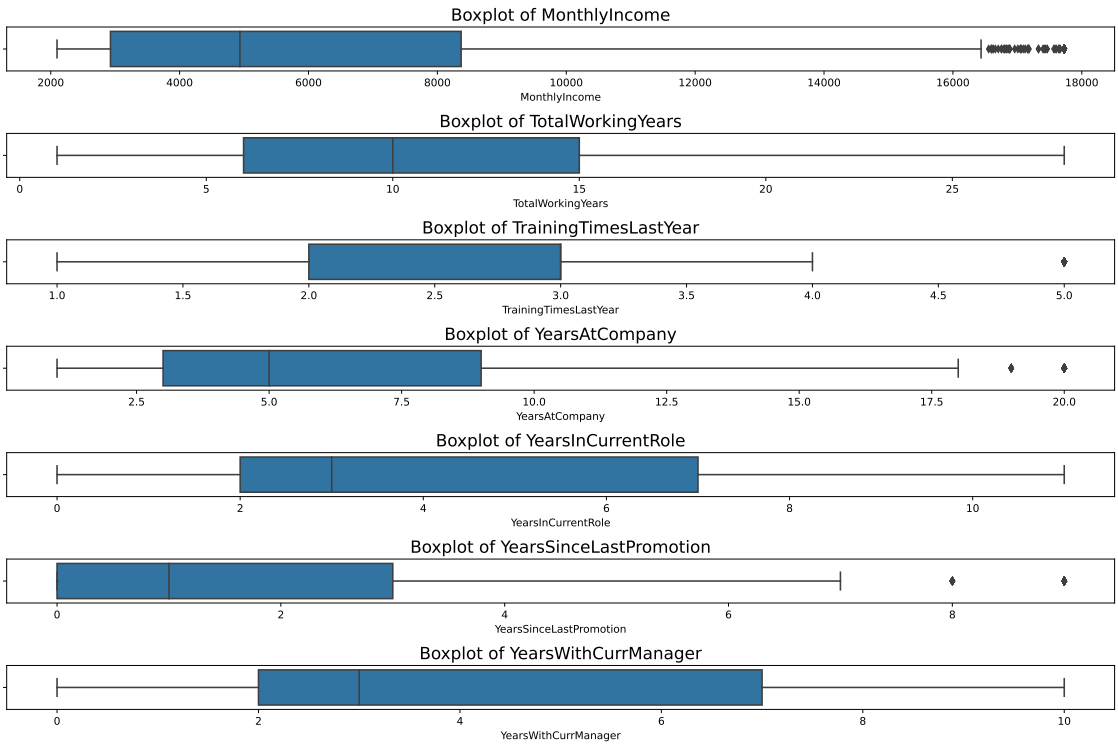
Các outliers này đều thuộc trường hợp natural outliers hay true outliers. Có nghĩa là tuy chúng là những ngoại lệ không phổ biến, chúng vẫn là một phần giá trị thực tế của dữ liệu. Việc giữ lại những outliers này hoàn toàn hợp lệ, tuy nhiên sự có mặt của chúng sẽ ảnh hưởng đến hiệu quả của các mô hình máy học và đồng thời ảnh hưởng đến độ chính xác của các giá trị thống kê. Do đó, nhóm chấp nhận giảm đi một phần tính thực tế của dữ liệu để xử lý các outliers này, đảm bảo hiệu quả cho các nội dung thống kê và xây dựng mô hình máy học.

Xử lý outliers bằng phương pháp thay thế giá trị dựa trên phân vị. Tức là thay thế các giá trị ngoại lai bằng các giá trị tại một phân vị cố định. Cụ thể, gán giới hạn dưới và trên lần lượt bằng giá trị tại phân vị thứ 5 và phân vị thứ 95. Tất cả các giá trị nhỏ hơn phân vị 5 sẽ được thay thế bằng giá trị của cận dưới và tất cả các giá trị lớn hơn phân vị 95 sẽ được thay thế bằng cận trên.

```
outlier_col = ['MonthlyIncome', 'TotalWorkingYears', 'TrainingTimesLastYear',
               'YearsAtCompany', 'YearsInCurrentRole',
               ↪  'YearsSinceLastPromotion',
               'YearsWithCurrManager']


#Outlier capping
for column in outlier_col:
    lower_bound = df[column].quantile(0.05)
    upper_bound = df[column].quantile(0.95)
    df[column] = df[column].clip(lower=lower_bound, upper=upper_bound)
```

Xem lại boxplot sau khi xử lý outliers.

```
fig, axs = plt.subplots(len(outlier_col), figsize=(15, 10))
for i, column in enumerate(outlier_col):
    sbn.boxplot(x=df[column], ax=axs[i])
    axs[i].set_title(f'Boxplot of {column}', fontsize = 16)


plt.tight_layout()
plt.savefig(f"figs/Boxplot after handling outliers.pdf")
plt.show()
```



Hình 3: Boxplots sau khi xử lý outliers

2.5.1 Lưu dữ liệu

Tiến hành lưu bộ dữ liệu đã tiền xử lý để thuận tiện cho việc sử dụng ở những công đoạn tiếp theo.

```
#Lưu dữ liệu
df.to_csv('data_iqr.csv')
```

3 CHƯƠNG 3: PHÂN TÍCH ĐƠN BIẾN

Phân tích đơn biến đóng vai trò quan trọng trong quá trình khai thác thông tin từ bộ dữ liệu HR Analytics, giúp nhóm tiếp cận một cách toàn diện đối với các biến quan trọng. Chương này sẽ chú trọng vào việc xác định các đại lượng mô tả xu thế trung tâm như trung bình, trung vị, và yếu vị. Đồng thời, ta cũng sẽ tìm hiểu về các đại lượng liên quan đến độ phân tán như tứ phân vị, phương sai, độ lệch chuẩn và những đại lượng đặc trưng về phân phối như độ lệch, độ nhọn.

Việc phân tích đơn biến giúp xây dựng nền tảng cho những phân tích đa biến và mô hình học máy ở các chương sau. Bằng cách hiểu rõ các đặc điểm cơ bản của các biến, ta có khả năng đưa ra giả định và lập kế hoạch cho việc phân tích chi tiết hơn về mối liên quan

giữa các biến, đồng thời giúp xây dựng các mô hình dự đoán một cách hiệu quả.

3.1 Biến định lượng

3.1.1 Thang đo khoảng (interval scale)

Thang đo khoảng là một loại thang đo dùng cho dữ liệu định lượng, các giá trị được xếp theo một quy ước về thứ bậc hay sự hơn kém, và ta có thể biết được khoảng cách giữa các giá trị.

Thông thường, biến định lượng sử dụng thang đo khoảng có dạng là một chuỗi số liên tục và đồng đều. Ví dụ, trong trường hợp mức độ hài lòng từ 1 đến 5, giá trị 1 thường đại diện cho trạng thái rất không hài lòng, trong khi giá trị 5 thường đại diện cho trạng thái rất hài lòng.

Một ưu điểm lớn của thang đo khoảng là khả năng thực hiện mọi phép toán thống kê, ngoại trừ phép chia. Điều này có nghĩa là có thể tính được trung bình, trung vị, yếu vị, phương sai và các đại lượng thống kê khác. Tuy nhiên, phép chia không thực hiện được trên thang đo khoảng.

Chọn các biến định lượng có thang đo khoảng trong bộ dữ liệu:

```
interval_col = ['EnvironmentSatisfaction','JobInvolvement','JobSatisfaction',
            'PerformanceRating','RelationshipSatisfaction','WorkLifeBalance']
```

Các đại lượng thống kê mô tả của từng biến định lượng có thang đo khoảng:

```
for col in interval_col:
  mean = np.mean(df[col]).round(3)
  mode = list(df[col].mode().round(3))
  median = np.median(df[col]).round(3)
  variance = np.var(df[col], ddof=1).round(3)
  std_deviation = np.std(df[col], ddof=1).round(3)
  skewness = skew(df[col]).round(2)
  kurt = kurtosis(df[col]).round(2)
  print(f'Các đại lượng về xu thế trung tâm của biến {col}')
  print(f'Mean: {mean}')
  print(f'Mode: {mode}')
  print(f'Median: {median}')
  print('')
  print(f'Các đại lượng về độ phân tán biến {col}')
  print(f'Phương sai (Variance): {variance}')
  print(f'Độ lệch chuẩn (Standard deviation): {std_deviation}')
```

```
  print('')
  print(f'Các đại lượng về hình dáng phân phối biến {col}')
  print(f'Độ lệch: {skewness}')
  print(f'Độ nhọn: {kurt}')
  print('-'*50)
```

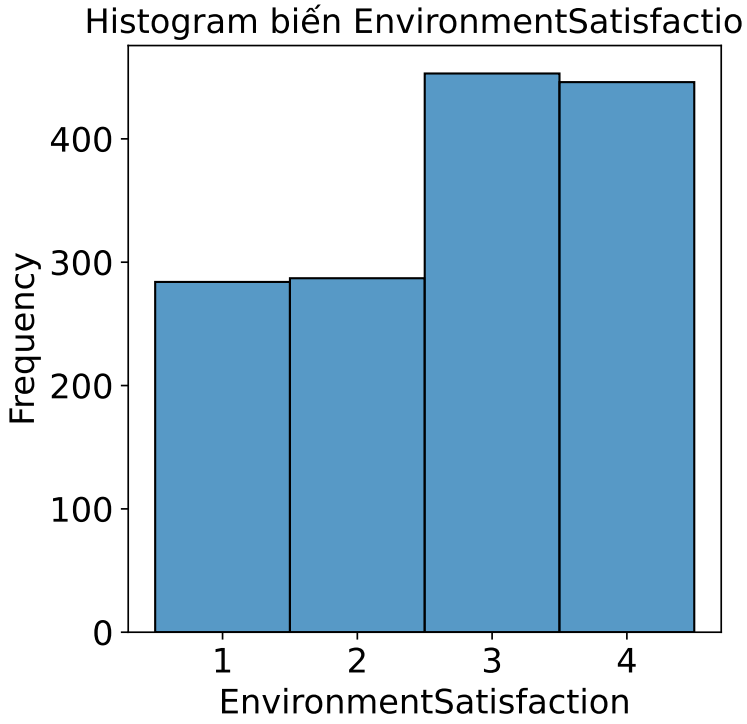
Vẽ biểu đồ tần số của từng biến định lượng có thang đo khoảng:

```
from matplotlib.ticker import MaxNLocator
for col in interval_col:
  if len(df[col].unique()) <= 25:
    plt.figure(figsize =(5,5))
    sbn.histplot(df, x=col, discrete=True)
    #Chuyển về số nguyên
    ax = plt.gca()
    ax.xaxis.set_major_locator(MaxNLocator(integer=True))
    plt.title(f'Histogram biến {col}',fontsize = 16)
    plt.xlabel(f'{col}',fontsize = 16)
    plt.ylabel('Frequency',fontsize = 16)
    plt.xticks(fontsize=16)
    plt.yticks(fontsize=16)
    plt.tight_layout()
    plt.savefig(f"figs/Histogram biến {col}.pdf")
    plt.show()
```

Biến EnvironmentSatisfaction

Biến EnvironmentSatisfaction cho biết mức độ hài lòng với môi trường làm việc được nhân viên đánh giá từ 1 đến 4 (1 = rất không hài lòng; 4 = rất hài lòng).

```
Các đại lượng về xu thế trung tâm của biến EnvironmentSatisfaction
Mean: 2.722
Mode: [3]
Median: 3.0

Các đại lượng về độ phân tán biến EnvironmentSatisfaction
Phương sai (Variance): 1.195
Độ lệch chuẩn (Standard deviation): 1.093

Các đại lượng về hình dáng phân phối biến EnvironmentSatisfaction
Độ lệch: -0.32
Độ nhọn: -1.2
```



Hình 4: Các đại lượng thống kê mô tả và biểu đồ tần số biến EnvironmentSatisfaction

Phân phối của biến ’EnvironmentSatisfaction’ có tập trung chủ yếu quanh giá trị 3, với một lượng lớn người lao động báo cáo mức độ hài lòng môi trường làm việc ở mức cao.

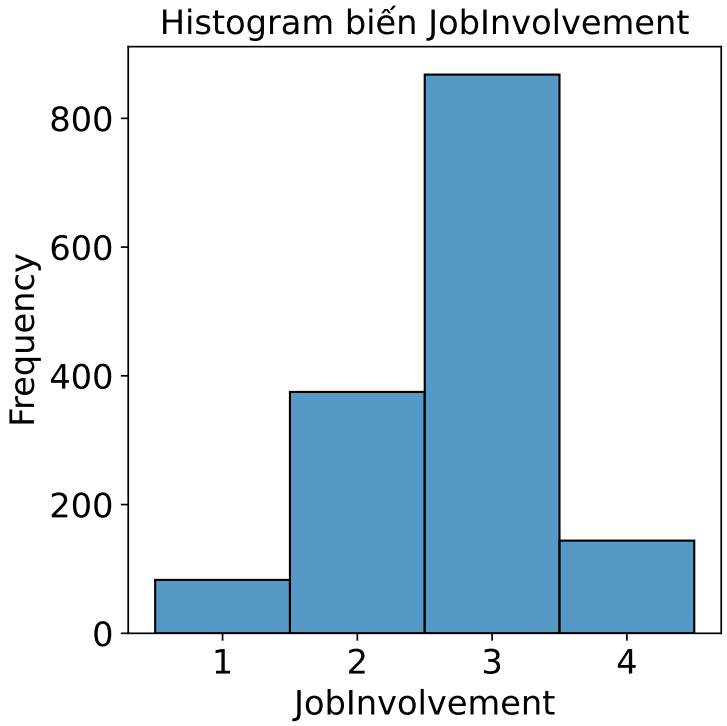
Giá trị trung bình và trung vị khá gần nhau và nằm ở mức trung bình của thang đo từ 1 đến 4, điều này cho thấy dữ liệu không bị lệch nhiều và có sự tập trung ở mức độ hài lòng trung bình đến cao. Độ phân tán không quá lớn, điều này cho thấy không có sự chênh lệch đáng kể về mức độ hài lòng môi trường làm việc giữa các nhân viên.

Nhìn chung, phân phối của biến ’EnvironmentSatisfaction’ trong tập dữ liệu cho thấy rằng mức độ hài lòng môi trường làm việc của nhân viên nghiêng về mức độ hài lòng trung bình đến cao, với mức độ biến động không lớn và một sự tập trung ở giá trị 3. Điều này có thể phản ánh rằng môi trường làm việc tại nơi nghiên cứu được đánh giá là khá tốt bởi đa số nhân viên.

Biến JobInvolvement

Biến JobInvolvement cho biết mức độ tham gia tích cực vào công việc của nhân viên được đánh giá từ 1 đến 4 (1 = không tích cực; 4 = rất tích cực).

```
Các đại lượng về xu thế trung tâm của biến JobInvolvement
Mean: 2.73
Mode: [3]
Median: 3.0

Các đại lượng về độ phân tán biến JobInvolvement
Phương sai (Variance): 0.506
Độ lệch chuẩn (Standard deviation): 0.712

Các đại lượng về hình dáng phân phối biến JobInvolvement
Độ lệch: -0.5
Độ nhọn: 0.27
```



Hình 5: Các đại lượng thống kê mô tả và biểu đồ tần số biến JobInvolvement

Phân phối của 'JobInvolvement' cho thấy một xu hướng chung của mức độ tham gia công việc ở mức từ trung bình đến cao trong tổ chức, với mức độ 3 là phổ biến nhất, phản ánh một môi trường làm việc nơi nhân viên cảm thấy công việc của họ có ý nghĩa và sẵn sàng tham gia vào công việc của mình.

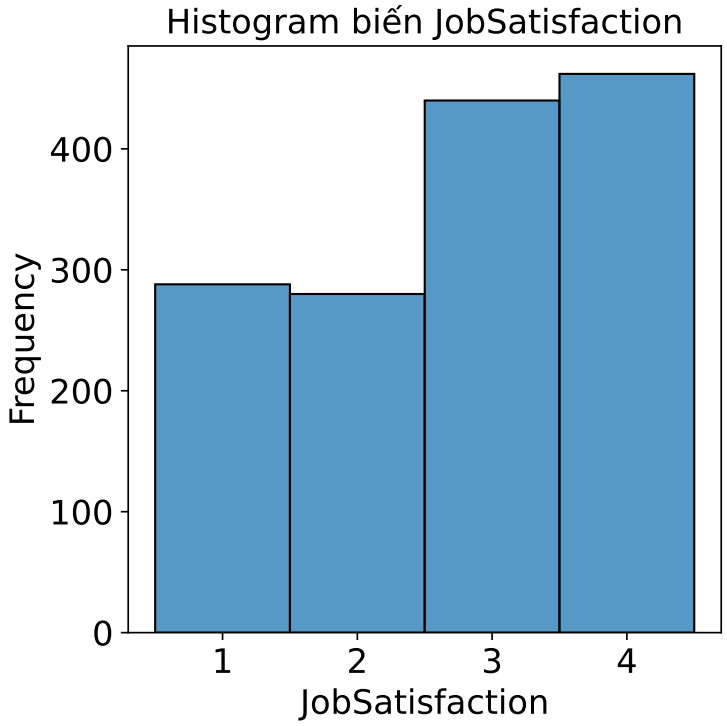
Có một lượng nhân viên ở mức độ tham gia 4, điều này có thể chỉ ra rằng một phần nhỏ của nhân viên cảm thấy rất gắn bó và hăng hái với công việc của họ. Tuy nhiên, có một lượng nhỏ nhân viên ở các mức độ tham gia thấp hơn là mức 1 và 2. Công ty nên xác định và giải quyết những nguyên nhân khiến cho mức độ tham gia vào công việc của nhân viên rất thấp, từ đó đưa ra giải pháp nhằm nâng cao hiệu suất, tăng mức độ hài lòng

đối với công việc cũng như mức độ tham vào công việc của nhân viên.

Biến JobSatisfaction

Biến JobSatisfaction cho biết mức độ hài lòng với công việc được nhân viên đánh giá từ 1 đến 4 (1 = rất không hài lòng; 4 = rất hài lòng).

```
Các đại lượng về xu thế trung tâm của biến JobSatisfaction
Mean: 2.732
Mode: [4]
Median: 3.0

Các đại lượng về độ phân tán biến JobSatisfaction
Phương sai (Variance): 1.217
Độ lệch chuẩn (Standard deviation): 1.103

Các đại lượng về hình dáng phân phối biến JobSatisfaction
Độ lệch: -0.33
Độ nhọn: -1.22
```



Hình 6: Các đại lượng thống kê mô tả và biểu đồ tần số biến JobSatisfaction

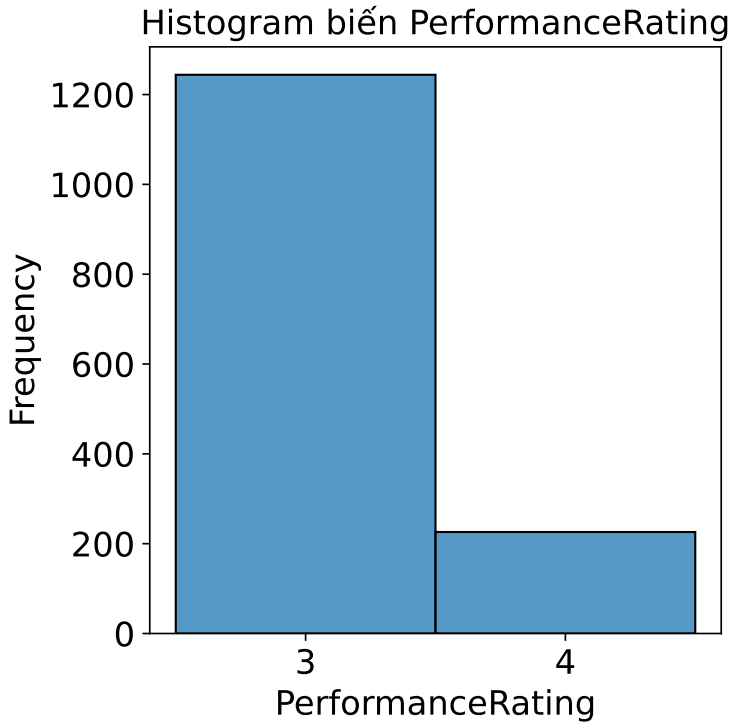
Trung bình mức độ hài lòng của mỗi nhân viên là 3, và đặc biệt, mức độ hài lòng được nhân viên đánh giá nhiều nhất là 4. Phân phối có chiều hướng lệch trái, được thể hiện qua độ lệch âm (-0.33), cho thấy mức độ hài lòng của nhân viên tập trung chủ yếu ở các mức độ cao là 3 và 4. Tuy nhiên, độ nhọn âm (-1.22) của phân phối là dấu hiệu của một phân phối có đỉnh phẳng và đuôi mỏng, đồng nghĩa rằng không có sự chênh lệch lớn giữa các mức đánh giá, số lượng nhân viên không hài lòng với công việc vẫn khá đáng kể.

Nhìn chung, dữ liệu cho thấy tỷ lệ lớn nhân viên đều hài lòng và mức độ hài lòng tập trung chủ yếu ở các mức độ cao. Đây là tín hiệu khả quan, cho thấy có một môi trường làm việc tích cực. Điều này có thể có ảnh hưởng tích cực đến sự hạnh phúc và hiệu suất của nhân viên, đồng thời tạo điều kiện thuận lợi cho sự phát triển và ổn định của công ty.

Biến PerformanceRating

Biến PerformanceRating cho biết đánh giá về khả năng và hiệu suất làm việc của nhân viên trong công ty đạt mức 3 hay 4.

```
Các đại lượng về xu thế trung tâm của biến PerformanceRating
Mean: 3.154
Mode: [3]
Median: 3.0

Các đại lượng về độ phân tán biến PerformanceRating
Phương sai (Variance): 0.13
Độ lệch chuẩn (Standard deviation): 0.361

Các đại lượng về hình dáng phân phối biến PerformanceRating
Độ lệch: 1.92
Độ nhọn: 1.69
```



Hình 7: Các đại lượng thống kê mô tả và biểu đồ tần số biến PerformanceRating

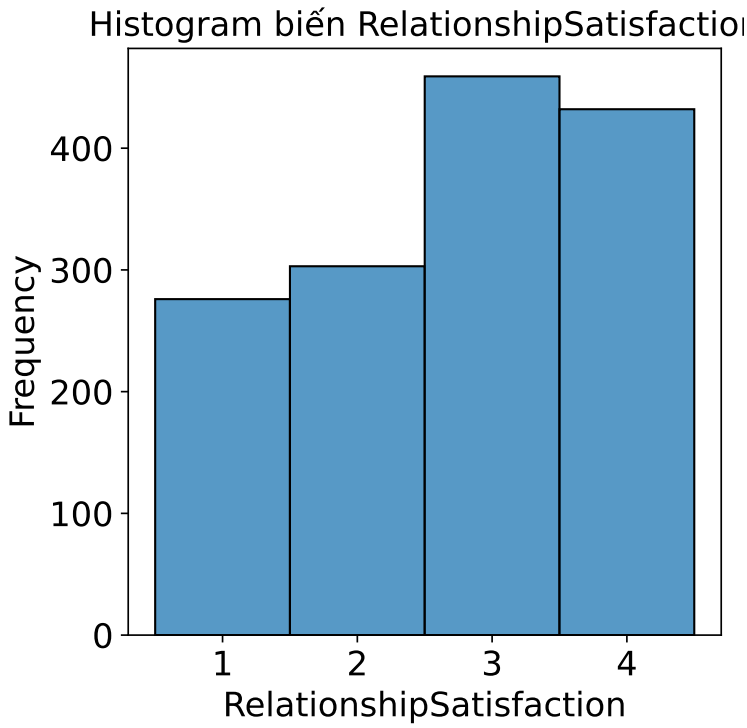
Tất cả các nhân viên đều được đánh giá Performance rating ở mức từ 3 đến 4, trong đó có khoảng hơn 1,200 nhân viên được đánh giá ở mức 3 và số lượng nhân viên được đánh giá ở mức 4 là ít hơn khá đáng kể, với khoảng 300 nhân viên. Điều này cho thấy rằng dù không có nhân viên nào bị đánh giá Performance Rating ở mức tệ (từ 1 đến 2), tuy nhiên

số lượng nhân viên xuất sắc hoàn toàn và đạt mức điểm tối đa cũng rất ít, khoảng 25% trên tổng số nhân viên trong bộ dữ liệu.

Biến RelationshipSatisfaction

Biến RelationshipSatisfaction cho biết mức độ hài lòng của nhân viên về các mối quan hệ từ 1 đến 4 (1 = rất không hài lòng; 4 = rất hài lòng).

```
Các đại lượng về xu thế trung tâm của biến RelationshipSatisfaction
Mean: 2.712
Mode: [3]
Median: 3.0

Các đại lượng về độ phân tán biến RelationshipSatisfaction
Phương sai (Variance): 1.169
Độ lệch chuẩn (Standard deviation): 1.081

Các đại lượng về hình dáng phân phối biến RelationshipSatisfaction
Độ lệch: -0.3
Độ nhọn: -1.18
```



Hình 8: Biểu đồ tần số biến RelationshipSatisfaction

Mức độ hài lòng với các mối quan hệ không chỉ là yếu tố quan trọng đối với tâm lý cá nhân mà còn ảnh hưởng đến cách mà nhân viên tương tác và hoạt động trong một tổ chức. Một môi trường làm việc tích cực về mối quan hệ có thể đóng góp quan trọng vào sự thành công và hiệu suất công ty.

Trung bình, trung vị và yếu vị đều là 3, cho thấy sự phân bố đều đặn giữa các mức đánh giá. Điều này thể hiện rằng có một sự cân bằng về số lượng giữa những người có đánh giá thấp hơn, bằng và cao hơn so với trung bình. Phân phối lệch nhẹ về phía trái và đỉnh

phẳng của phân phối cũng cho thấy sự đồng đều này nhưng cũng chỉ ra rằng mức độ hài lòng tập trung nhiều hơn ở mức 3 và 4.

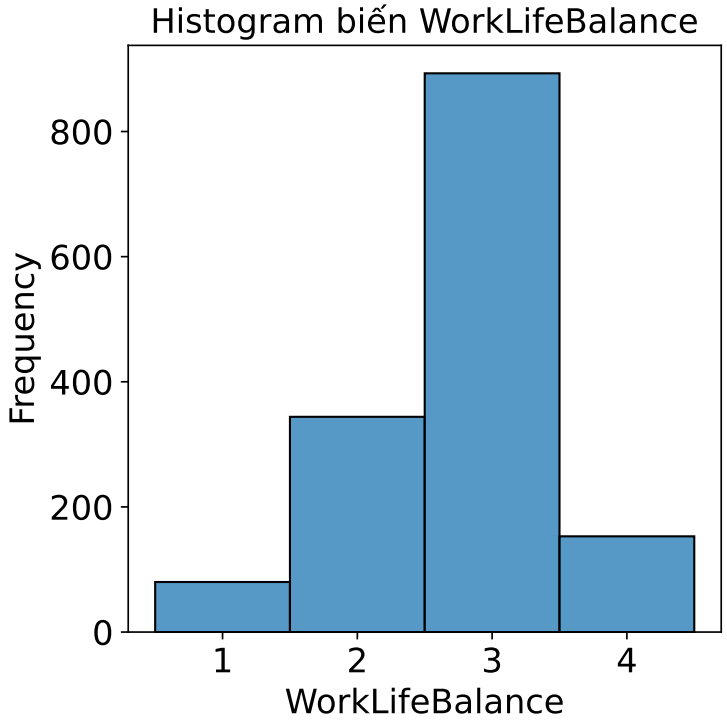
Bởi vì mức độ hài lòng với mối quan hệ có thể ảnh hưởng đến hiệu suất công việc, việc nâng cao mức độ hài lòng của nhân viên trong công ty trở nên quan trọng. Tuy nhiên, đây là một yếu tố nằm ngoài tầm kiểm soát của công ty, nơi chỉ có thể tập trung vào việc cải thiện mối quan hệ đồng nghiệp giữa các nhân viên. Điều này có thể được thực hiện bằng cách tạo ra cơ hội cho các nhân viên để gặp gỡ và làm việc chung, từ đó tạo dựng sự đoàn kết. Đồng thời, đối xử công bằng trong môi trường làm việc cũng là chìa khóa quan trọng để tránh xung đột trong công ty, giúp duy trì một không gian làm việc tích cực.

Biến WorkLifeBalance

Biến WorkLifeBalance cho biết mức độ cân bằng giữa công việc và của sống của nhân viên trên thang đo điểm 4.

Mode bằng 3 tương ứng với việc số nhân viên có WorkLifeBalance nhiều nhất là ở mức 3, với gần 900 nhân viên. Mức độ WorkLifeBalance trung bình của các nhân viên nằm ở mức khá tốt ở khoảng 3.15. Có ít hơn 100 nhân viên có mức độ WorkLifeBalance bằng 1.

```
Các đại lượng về xu thế trung tâm của biến WorkLifeBalance
Mean: 2.761
Mode: [3]
Median: 3.0

Các đại lượng về độ phân tán biến WorkLifeBalance
Phương sai (Variance): 0.499
Độ lệch chuẩn (Standard deviation): 0.706

Các đại lượng về hình dáng phân phối biến WorkLifeBalance
Độ lệch: -0.55
Độ nhọn: 0.41
```



Hình 9: Các đại lượng thống kê mô tả và biểu đồ tần số biến WorkLifeBalance

3.1.2 Thang đo tỷ lệ (ratio scale)

Thang đo tỷ lệ là loại thang đo có đầy đủ các đặc tính của thang đo khoảng nhưng cho phép người dùng có thể lấy tỷ lệ để so sánh giá trị giữa các biến số.

Chọn các biến định lượng có thang đo tỷ lệ trong bộ dữ liệu:

```
ratio_col = ['Age', 'DailyRate', 'DistanceFromHome','HourlyRate',
↪  'MonthlyIncome','NumCompaniesWorked','PercentSalaryHike',
↪  'TotalWorkingYears', 'TrainingTimesLastYear','YearsAtCompany',
↪  'YearsInCurrentRole','YearsSinceLastPromotion',
↪  'YearsWithCurrManager']
```

Các đại lượng thống kê mô tả của từng biến định lượng có thang đo tỷ lệ:

```
for col in ratio_col:
  mean = np.mean(df[col]).round(3)
  mode = list(df[col].mode().round(3))
  median = np.median(df[col]).round(3)
  data_range = np.ptp(df[col])
  variance = np.var(df[col], ddof=1).round(3)
  std_deviation = np.std(df[col], ddof=1).round(3)
  skewness = skew(df[col]).round(2)
  kurt = kurtosis(df[col]).round(2)
  print(f'Các đại lượng về xu thế trung tâm của biến {col}')
  print(f'Mean: {mean}')
  print(f'Mode: {mode}')
  print(f'Median: {median}')
  print('')
  print(f'Các đại lượng về độ phân tán biến {col}')
  print(f'Khoảng biến thiên (Range): {data_range}')
  print(f'Phương sai (Variance): {variance}')
  print(f'Độ lệch chuẩn (Standard deviation): {std_deviation}')
  print('')
  print(f'Các đại lượng về hình dáng phân phối biến {col}')
  print(f'Độ lệch: {skewness}')
  print(f'Độ nhọn: {kurt}')
  print('-'*50)
```

Vẽ biểu đồ phân phối của các biến định lượng liên tục có thang đo tỷ lệ:

```
#Distribution các biến liên tục
for col in float_cols.columns:
  plt.title(f'Distribution biến {col}',fontsize = 16)
  plt.xlabel(f'{col}',fontsize = 16)
  plt.ylabel('Density',fontsize = 16)
  sbn.distplot(float_cols[col], kde = True, norm_hist = True)
  plt.xticks(fontsize = 12)
  plt.yticks(fontsize = 12)
  plt.tight_layout()
  plt.savefig(f"figs/Distribution biến {col}.pdf")
  plt.show()
```

Vẽ biểu đồ phân phối của các biến định lượng rời rạc ít giá trị (<=25) có thang đo tỷ lệ:

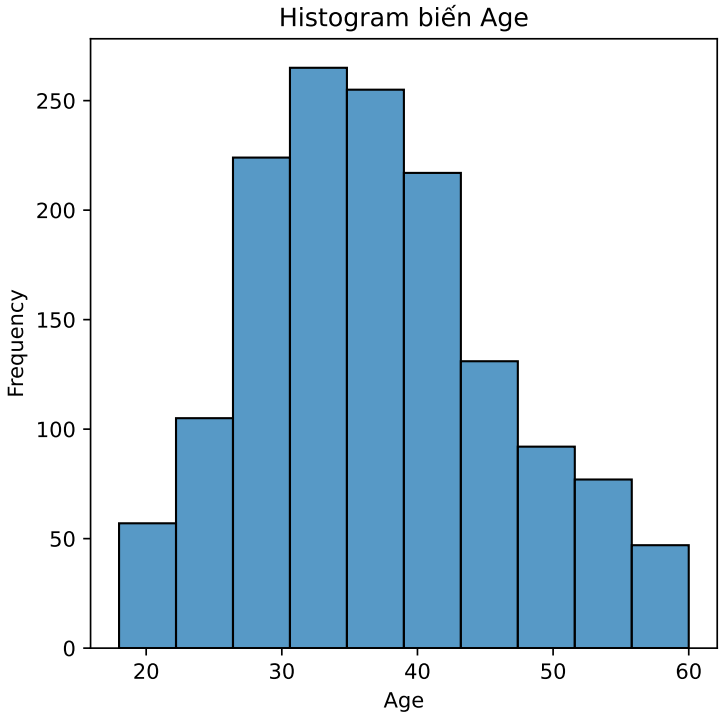
```
for col in int_cols.columns:
  if len(df[col].unique()) <= 25:
    plt.figure(figsize = (4,4))
    sbn.displot(df, x=col, discrete=True)
    #Chuyển về số nguyên
    ax = plt.gca()
    ax.xaxis.set_major_locator(MaxNLocator(integer=True))
    plt.title(f'Distribution biến {col}',fontsize = 16)
    plt.xlabel(f'{col}',fontsize = 16)
    plt.ylabel('Frequency',fontsize = 16)
    plt.xticks(fontsize = 16)
    plt.yticks(fontsize = 16)
    plt.tight_layout()
    plt.savefig(f"figs/Distribution biến {col}.pdf")
    plt.show()
```

Vẽ biểu đồ phân phối của các biến định lượng rời rạc nhiều giá trị (>25) có thang đo tỷ lệ:

```
for col in int_cols.columns:
  if len(df[col].unique()) > 25:
    plt.figure(figsize = (5,5))
    sns.histplot(df[col], bins=10)
    plt.title(f'Histogram biến {col}',fontsize=16)
    plt.xlabel(f'{col}',fontsize=16)
    plt.ylabel('Frequency',fontsize=16)
    plt.xticks(fontsize=12)
    plt.yticks(fontsize=12)
    plt.tight_layout()
    plt.savefig(f'figs/Phân phối {col}.pdf')
    plt.show()
```

Biến Age

```
Các đại lượng về xu thế trung tâm của biến Age
Mean: 36.924
Mode: [35]
Median: 36.0

Các đại lượng về độ phân tán biến Age
Khoảng biến thiên (Range): 42
Phương sai (Variance): 83.455
Độ lệch chuẩn (Standard deviation): 9.135

Các đại lượng về hình dáng phân phối biến Age
Độ lệch: 0.41
Độ nhọn: -0.41
```

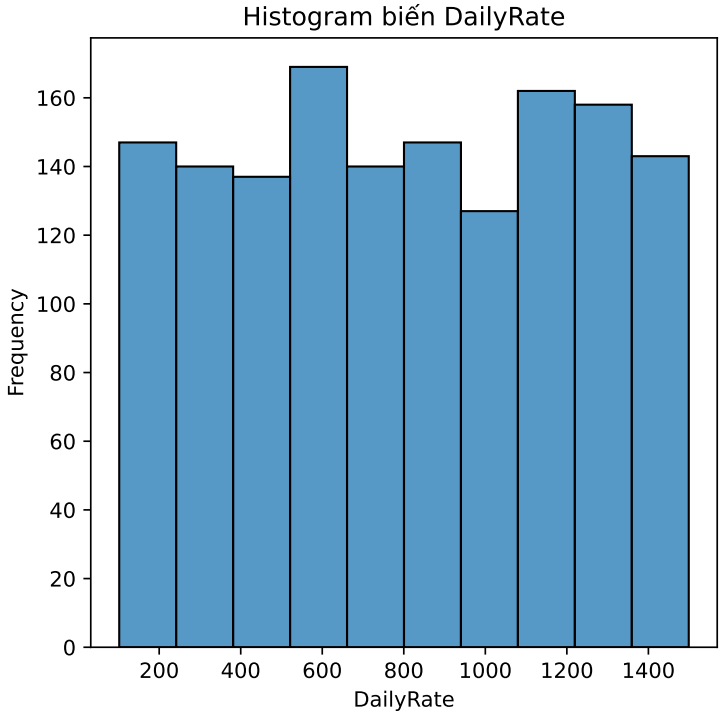


Hình 10: Các đại lượng thống kê mô tả và biểu đồ tần số biến Age

Độ tuổi trung bình và trung vị của nhóm nhân viên khá gần nhau. Độ lệch của phân phối biến "Age" dương (0.41) và độ nhọn âm (-0.41) cho thấy phân phối có xu hướng lệch về bên phải và phẳng nhưng không quá lớn, tức là số lượng nhân viên trẻ nhiều hơn so với những người ở độ tuổi già hơn. Độ tuổi phổ biến nhất của nhân viên rơi vào khoảng độ tuổi từ 30 đến 40, tập trung xung quanh độ tuổi trung bình và với một số lượng lớn nhân viên có độ tuổi là 35.

Biến DailyRate

```
Các đại lượng về xu thế trung tâm của biến DailyRate
Mean: 802.486
Mode: [691]
Median: 802.0

Các đại lượng về độ phân tán biến DailyRate
Khoảng biến thiên (Range): 1397
Phương sai (Variance): 162819.594
Độ lệch chuẩn (Standard deviation): 403.509

Các đại lượng về hình dáng phân phối biến DailyRate
Độ lệch: -0.0
Độ nhọn: -1.2
```



Hình 11: Các đại lượng thống kê mô tả và biểu đồ tần số biến DailyRate

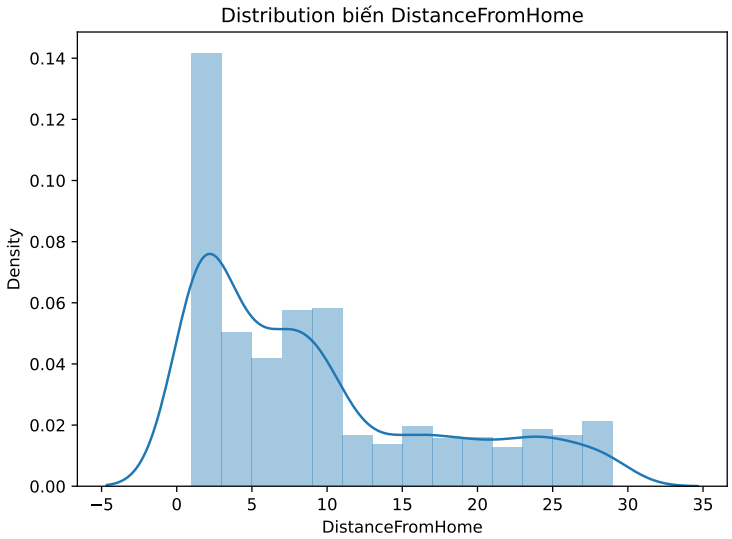
Tiền lương theo ngày của nhân viên có mức trung bình là 802.486, với giá trị xuất hiện nhiều nhất là 691.

Histogram cho thấy sự phân bố khá đều qua các khoảng của DailyRate, không có một khoảng giá trị nào chiếm ưu thế rõ rệt so với các khoảng giá trị khác. Điều này cho thấy sự đa dạng trong mức lương hàng ngày của nhân viên. Giá trị độ lệch gần như bằng 0 (-0.0), điều này chỉ ra rằng phân phối của biến DailyRate là tương đối đối xứng qua trung tâm của phân phối. Không có sự thiên vị về mức lương thấp hay cao. Sự phân tán của mức lương hàng ngày không tập trung chủ yếu quanh giá trị trung bình mà phân bố đều hơn.

Mức lương hàng ngày của nhân viên phân bố một cách đều khắp các khoảng giá trị, không có sự chênh lệch lớn giữa các nhóm lương, cho thấy một chính sách tiền lương có tính đồng đều và công bằng. Không có dấu hiệu của sự thiên vị về mức lương thấp hoặc cao, và không có sự phân biệt đáng kể nào trong việc phân phối lương hàng ngày giữa các nhân viên.

Biến DistanceFromHome

```
Các đại lượng về xu thế trung tâm của biến DistanceFromHome
Mean: 9.192
Mode: [2.0]
Median: 7.0

Các đại lượng về độ phân tán biến DistanceFromHome
Khoảng biến thiên (Range): 28.0
Phương sai (Variance): 65.442
Độ lệch chuẩn (Standard deviation): 8.09

Các đại lượng về hình dáng phân phối biến DistanceFromHome
Độ lệch: 0.96
Độ nhọn: -0.22
```



Hình 12: Các đại lượng thống kê mô tả và biểu đồ phân phối biến DistanceFromHome

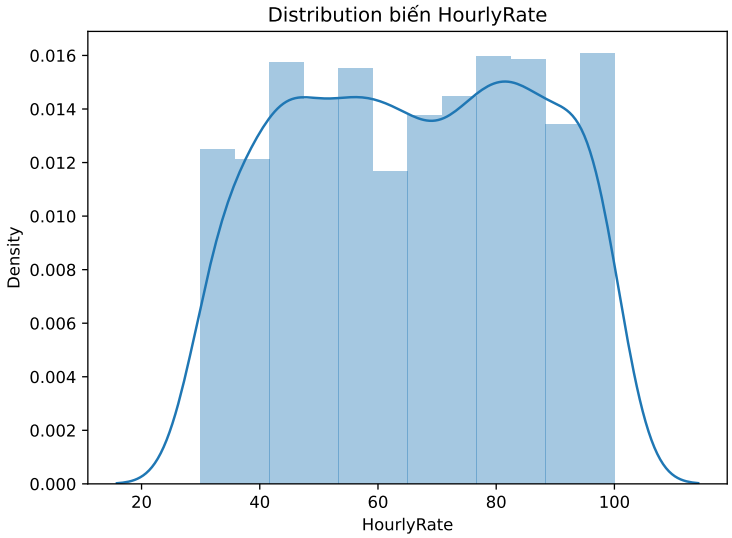
Khoảng cách từ nhà đến nơi làm việc có mức trung bình là 9.20. Độ lệch chuẩn khá lớn (8.088), chỉ ra rằng có sự biến động đáng kể trong dữ liệu về khoảng cách từ nhà đến nơi làm việc của nhân viên.

Giá trị độ lệch là 0.96, cho thấy phân phối bị lệch về phía bên phải. Điều này có nghĩa là có một số lượng không nhỏ nhân viên sống cách xa nơi làm việc, , có thể do lý do cá nhân hoặc thiếu hụt cơ hội việc làm gần khu vực họ sinh sống. Giá trị độ nhọn là -0.22, cho thấy phân phối này có độ nhọn thấp hơn một chút so với phân phối chuẩn (độ nhọn = 0). Điều này ngụ ý rằng phân phối có xu hướng phẳng hơn và đuôi phân bố rộng ra hơn, phản ánh việc có một số nhân viên sống cách xa nơi làm việc.

Những người quản lý nguồn nhân lực có thể xem xét thông tin này để cải thiện các chính sách hỗ trợ đi lại hoặc chính sách làm việc từ xa để giúp giảm bớt gánh nặng đi lại cho nhân viên.

Biến HourlyRate

```
Các đại lượng về xu thế trung tâm của biến HourlyRate
Mean: 65.901
Mode: [66.0]
Median: 66.0

Các đại lượng về độ phân tán biến HourlyRate
Khoảng biến thiên (Range): 70.0
Phương sai (Variance): 412.342
Độ lệch chuẩn (Standard deviation): 20.306

Các đại lượng về hình dáng phân phối biến HourlyRate
Độ lệch: -0.03
Độ nhọn: -1.19
```

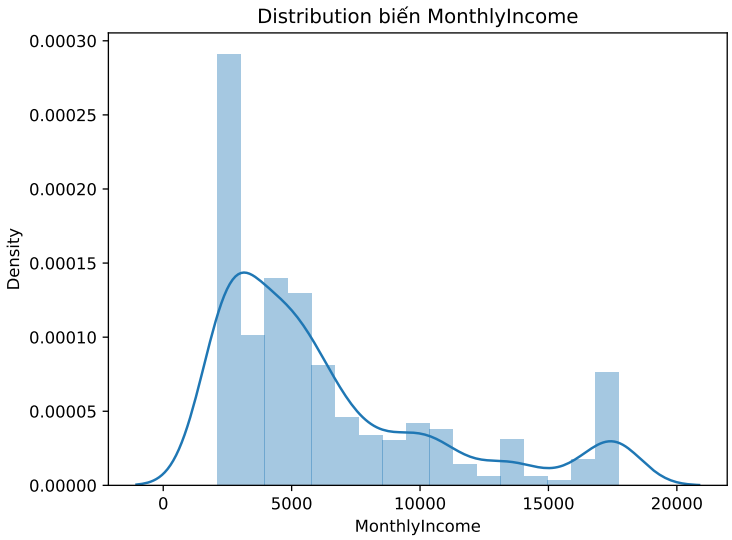


Hình 13: Các đại lượng thống kê mô tả và biểu đồ phân phối biến HourlyRate

Mức lương trung bình mà nhân viên có thể nhận được mỗi giờ làm việc là 65.868. Sự thiếu thiên lệch và độ nhọn thấp trong phân phối của biến HourlyRate cho thấy sự đồng đều giữa các mức lương theo giờ, không có sự tập trung quá mức ở bất kỳ mức lương cụ thể nào, điều này có thể phản ánh một chính sách lương bình đẳng và công bằng trong tổ chức.

Biến MonthlyIncome

```
Các đại lượng về xu thế trung tâm của biến MonthlyIncome
Mean: 6445.061
Mode: [2097.9, 17727.7]
Median: 4936.0

Các đại lượng về độ phân tán biến MonthlyIncome
Khoảng biến thiên (Range): 15629.799999999994
Phương sai (Variance): 20233058.323
Độ lệch chuẩn (Standard deviation): 4498.117

Các đại lượng về hình dáng phân phối biến MonthlyIncome
Độ lệch: 1.28
Độ nhọn: 0.63
```



Hình 14: Các đại lượng thống kê mô tả và biểu đồ phân phối biến MonthlyIncome

Biểu đồ phân phối thể hiện một đỉnh cao rõ rệt gần giá trị thấp của thu nhập hàng tháng, cho thấy một số lượng lớn nhân viên có thu nhập ở mức thấp. Giá trị độ lệch là 1.37, cho thấy phân phối có sự lệch phải. Điều này có nghĩa là có một số nhân viên có thu nhập hàng tháng cao hơn hẳn so với phần lớn những người khác.

Phân phối thu nhập hàng tháng cho thấy có sự chênh lệch đáng kể về thu nhập giữa các nhân viên. Sự lệch phải và độ nhọn cao của phân phối cho thấy sự không đồng đều trong thu nhập hàng tháng, chỉ có một số ít nhân viên có thu nhập rất cao so với phần còn lại.

Ta nhận thấy rõ có sự khác biệt giữa phân phối HourlyRate, DailyRate (tương đối đồng đều) và phân phối MonthlyIncome (có sự chênh lệch đáng kể). Điều này có thể phản ánh một cấu trúc lương có sự khác nhau dựa trên vị trí, kinh nghiệm, hoặc vai trò trong công ty. Có thể xem xét bởi các nguyên nhân sau:

HourlyRate và DailyRate: Các mức lương theo giờ hoặc ngày thường phản ánh lương cơ bản hoặc lương chuẩn mà một công ty trả cho nhiều nhân viên ở nhiều vị trí khác nhau,

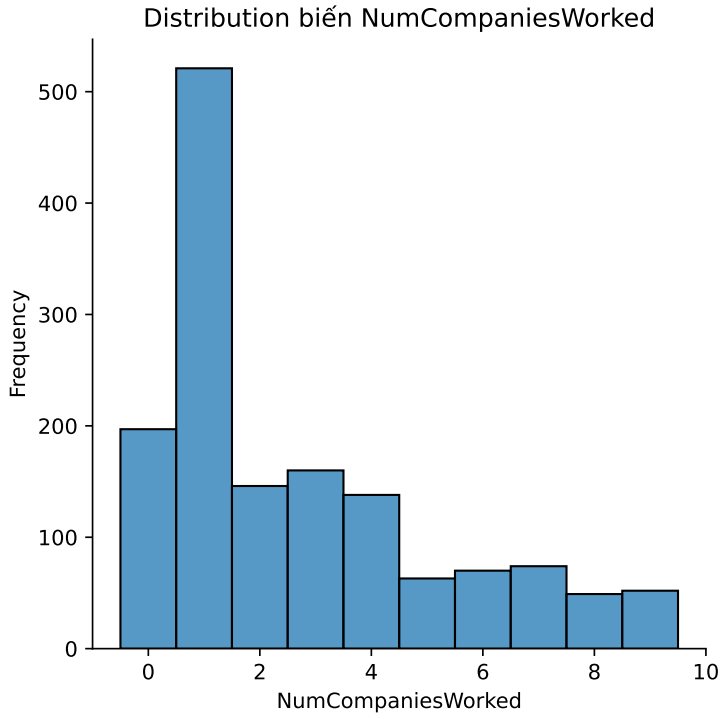
có thể do luật lao động hoặc hợp đồng lao động, thường cố định và ít biến đổi hơn.

MonthlyIncome: Thu nhập hàng tháng có thể bao gồm nhiều loại phụ cấp khác nhau, tiền thưởng, và các khoản thu nhập khác ngoài lương cơ bản phụ thuộc vào chức vụ của mỗi nhân viên là cao hay thấp. Điều này tạo ra sự chênh lệch lớn giữa thu nhập của các nhân viên, đặc biệt là giữa các cấp bậc khác nhau trong công ty.

Biến NumCompaniesWorked

Biến NumCompaniesWorked cho biết nhân viên đã làm việc tại bao nhiêu công ty trước đây hay nhân viên đã chuyển nơi làm việc bao nhiêu lần.

```
Các đại lượng về xu thế trung tâm của biến NumCompaniesWorked
Mean: 2.693
Mode: [1]
Median: 2.0

Các đại lượng về độ phân tán biến NumCompaniesWorked
Khoảng biến thiên (Range): 9
Phương sai (Variance): 6.239
Độ lệch chuẩn (Standard deviation): 2.498

Các đại lượng về hình dáng phân phối biến NumCompaniesWorked
Độ lệch: 1.03
Độ nhọn: 0.01
```



Hình 15: Các đại lượng thống kê mô tả và biểu đồ phân phối biến NumCompaniesWorked

Phân phối lệch phải với độ lệch 1.03 cho thấy xu hướng các nhân viên ít chuyển nơi làm việc. Cụ thể, trung bình mỗi nhân viên đã làm việc tại 3 công ty và đa số các nhân viên

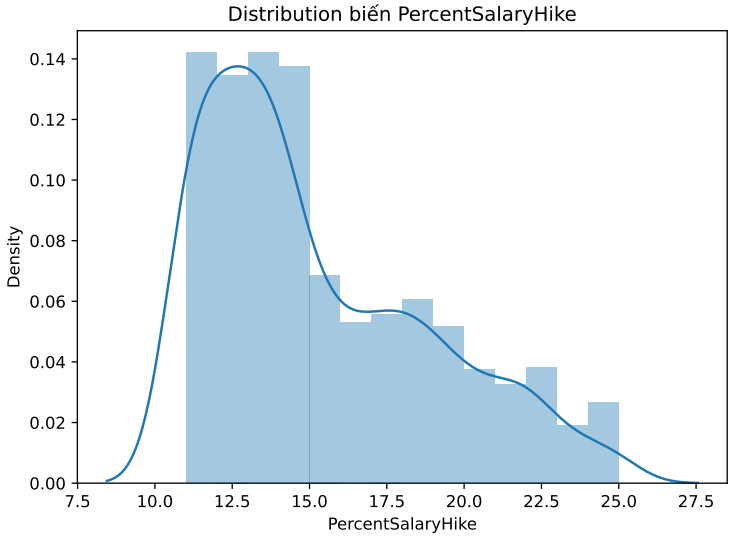
chỉ mới làm việc cho 1 công ty trước đây. Mặc dù khoảng biến thiên rộng (0-9) nhưng đến 50% nhân viên chỉ mới làm việc cho 2 công ty trở xuống.

Đa số nhân viên ít chuyển nơi làm việc có thể được xem là tích cực vì điều này cho thấy sự ổn định và cam kết cao trong công việc, dẫn đến khả năng những nhân viên này sẽ làm việc lâu dài cho công ty. Tuy nhiên, để tránh thiên vị cũng cần xem xét đến số năm làm việc của nhân viên vì có thể các nhân viên làm việc cho ít công ty trước đây cũng là nhân viên chưa có nhiều năm kinh nghiệm và những nhân viên làm việc cho nhiều công ty hơn đã có nhiều năm kinh nghiệm hơn.

Biến PercentSalaryHike

Biến PercentSalaryHike cho biết tỷ lệ phần trăm mức lương tăng so với mức lương trước đó.

```
Các đại lượng về xu thế trung tâm của biến PercentSalaryHike
Mean: 15.212
Mode: [11.0, 13.0]
Median: 14.0

Các đại lượng về độ phân tán biến PercentSalaryHike
Khoảng biến thiên (Range): 14.0
Phương sai (Variance): 13.383
Độ lệch chuẩn (Standard deviation): 3.658

Các đại lượng về hình dáng phân phối biến PercentSalaryHike
Độ lệch: 0.82
Độ nhọn: -0.3
```



Hình 16: Các đại lượng thống kê mô tả và biểu đồ phân phối biến PercentSalaryHike

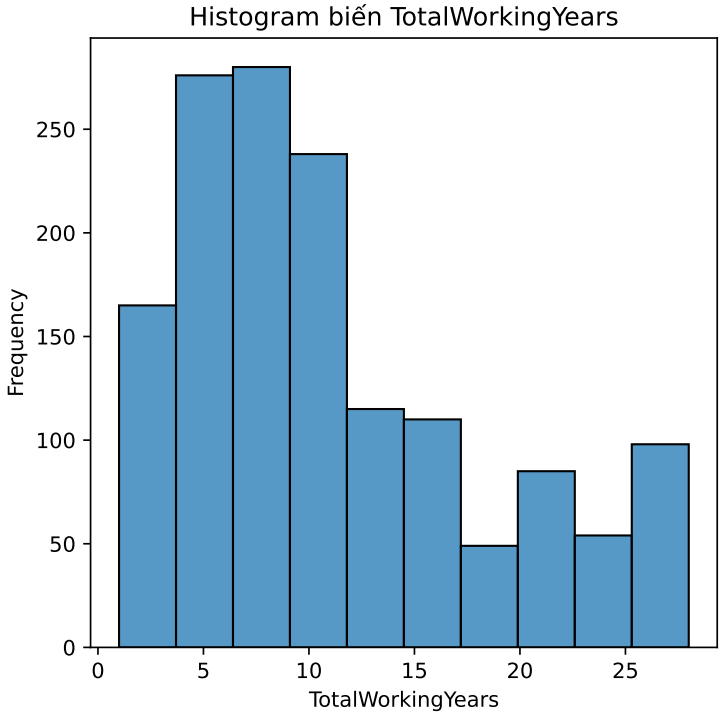
Mức tăng lương trung bình của nhân viên là 15.21% với mức lương tăng phổ biến là 11% và 13%. Tuy nhiên, phân phối dữ liệu có xu hướng lệch về phải với độ lệch là 0.82, cho thấy một số nhân viên đã nhận được mức tăng lương khá cao. Độ nhọn âm thể hiện rằng phân phối tương đối bằng phẳng, do đó, số lượng nhân viên nhận mức tăng lương

cao cũng không phải là hiếm. Nhìn chung, chính sách tăng lương của công ty này có vẻ ổn định và dàn trải đều giữa các mức tăng lương.

Chính sách lương ổn định và sự đồng đều trong mức tăng lương giữa các nhân viên không chỉ tạo ra một môi trường làm việc tích cực mà còn tăng cường lòng cam kết của nhân viên. Tuy nhiên, để đảm bảo sự công bằng, việc đánh giá cũng cần được thực hiện dựa trên đóng góp và hiệu suất cá nhân. Để đánh giá mức độ công bằng trong công ty, cần quan sát thêm về sự chênh lệch về mức lương tăng giữa các phòng/ban (Department).

Biến TotalWorkingYears

```
Các đại lượng về xu thế trung tâm của biến TotalWorkingYears
Mean: 11.089
Mode: [10]
Median: 10.0

Các đại lượng về độ phân tán biến TotalWorkingYears
Khoảng biến thiên (Range): 27
Phương sai (Variance): 52.452
Độ lệch chuẩn (Standard deviation): 7.242

Các đại lượng về hình dáng phân phối biến TotalWorkingYears
Độ lệch: 0.84
Độ nhọn: -0.11
```



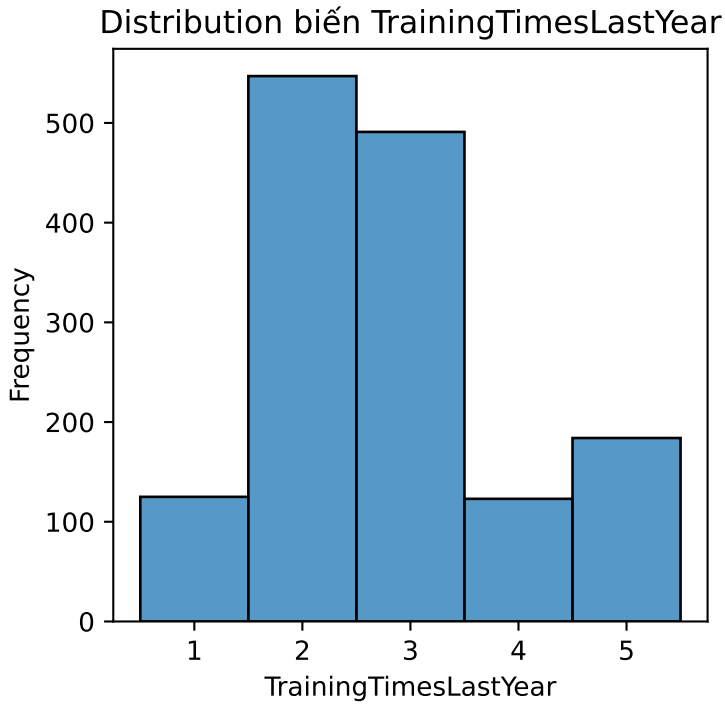
Hình 17: Các đại lượng thống kê mô tả và biểu đồ phân phối biến TotalWorkingYears

Trung bình số năm làm việc của nhân viên là 10. Độ lệch chuẩn khá nhỏ (6.335), cho thấy sự biến động trong dữ liệu không lớn. Tuy nhiên, có một số giá trị TotalWorkingYears rất cao làm tăng giá trị trung bình.

Phân phối có xu hướng lệch phải, có nghĩa là có nhiều cá nhân có tổng số năm làm việc ít hơn và giảm dần khi tổng số năm làm việc tăng lên. Phạm vi số năm làm việc phổ biến nhất là 5 đến 10 năm, vì phạm vi này có tần suất cao nhất. Có một sự giảm đáng kể về tần suất sau 10 năm rất ít cá nhân có hơn 20 năm làm việc tổng cộng, cho thấy rằng kinh nghiệm làm việc dài hạn như vậy là không phổ biến trong tập dữ liệu này. Phân phối không đều và cho thấy đa số lực lượng lao động có kinh nghiệm làm việc tương đối ít.

Biến TrainingTimesLastYear

```
Các đại lượng về xu thế trung tâm của biến TrainingTimesLastYear
Mean: 2.792
Mode: [2]
Median: 3.0

Các đại lượng về độ phân tán biến TrainingTimesLastYear
Khoảng biến thiên (Range): 4
Phương sai (Variance): 1.254
Độ lệch chuẩn (Standard deviation): 1.12

Các đại lượng về hình dáng phân phối biến TrainingTimesLastYear
Độ lệch: 0.59
Độ nhọn: -0.34
```



Hình 18: Biểu đồ phân phối biến TrainingTimesLastYear

Biểu đồ cho thấy số lần tập huấn phổ biến nhất mà nhân viên tham gia trong năm ngoái là 2 và 3 lần. Có rất ít nhân viên không tham gia bất kỳ khóa tập huấn nào. Sự biến động của số lần nhân viên tham huấn tập huấn không lớn.

Giá trị độ lệch là 0.55, cho thấy phân phối có sự lệch nhẹ về phía bên phải, có nghĩa là có một số nhân viên tham gia số lần tập huấn cao hơn trung bình. Giá trị độ nhọn là 0.49,

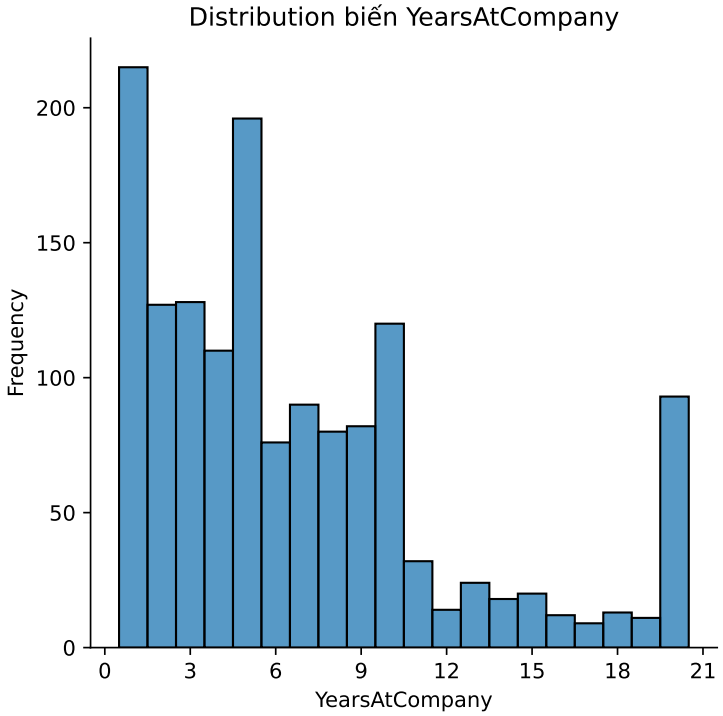
cho thấy phân phối có độ nhọn cao hơn một chút so với phân phối chuẩn, có sự tập trung một số lượng lớn các giá trị quanh trung tâm và ít dữ liệu ở phần đuôi.

Cần nghiên cứu để hiểu rõ nguyên nhân tại sao một lượng nhân viên không tham gia tập huấn và liệu điều này có phản ánh về mức độ không hài lòng trong công việc và hiệu suất làm việc hay không? Đồng thời cũng nên xem xét việc phát triển thêm các chương trình tập huấn đáp ứng đúng nhu cầu và mong muốn học hỏi của nhân viên để nhằm nâng cao tỷ lệ tham gia đào tạo của nhân viên.

Biến YearsAtCompany

Biến YearsAtCompany cho biết số năm làm việc tại công ty của nhân viên.

```
Các đại lượng về xu thế trung tâm của biến YearsAtCompany
Mean: 6.782
Mode: [1]
Median: 5.0

Các đại lượng về độ phân tán biến YearsAtCompany
Khoảng biến thiên (Range): 19
Phương sai (Variance): 27.818
Độ lệch chuẩn (Standard deviation): 5.274

Các đại lượng về hình dáng phân phối biến YearsAtCompany
Độ lệch: 1.11
Độ nhọn: 0.55
```



Hình 19: Các đại lượng thống kê mô tả và biểu đồ phân phối biến YearsAtCompany

Trung bình mỗi nhân viên đã gắn bó được 6-7 năm với công ty. Theo số liệu năm 2022 từ Cục Thống kê Lao động Hoa Kỳ, thời gian làm việc trung bình của một nhân viên tại

một công ty là 4.1 năm. Với thời gian làm việc trung bình cao hơn so với trung bình quốc gia, công ty có sự ổn định trong lực lượng lao động khi nhân viên thường xuyên gắn bó với công ty trong thời gian dài.

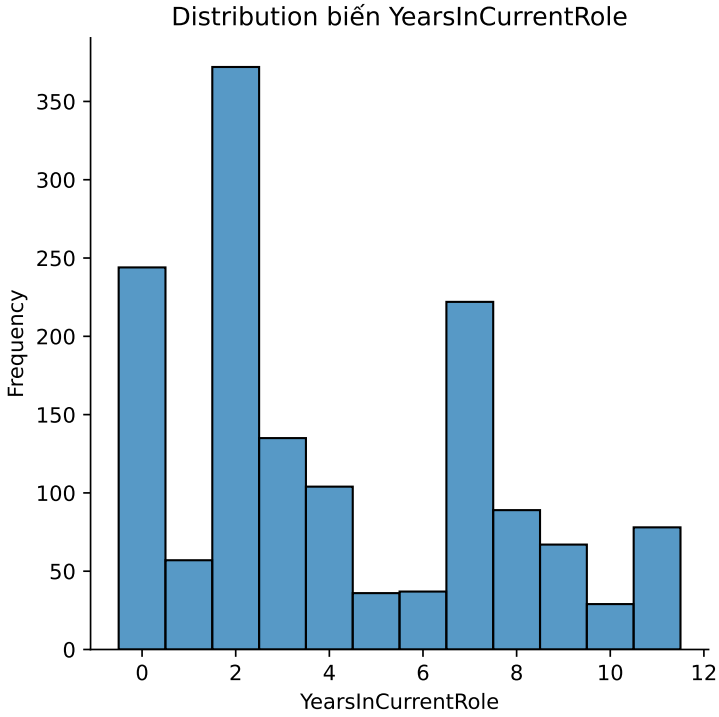
Số năm làm việc phổ biến nhất là 1 cho biết rằng gần đây công ty đã tuyển một lượng lớn nhân viên mới.

Độ nhọn gần bằng 0, cụ thể là 0.55, cho thấy sự tập trung nhẹ ở giá trị trung tâm với đuôi phân phối ít mở rộng hơn phân phối chuẩn. Điều này chứng tỏ có sự tập trung nhẹ ở giá trị trung tâm, có sự khác biệt trong thời gian làm việc của các nhóm nhân viên khác nhau.

Biến YearsInCurrentRole

Biến YearsInCurrentRole cho biết số năm giữ vị trí hiện tại của nhân viên.

```
Các đại lượng về xu thế trung tâm của biến YearsInCurrentRole
Mean: 4.11
Mode: [2]
Median: 3.0

Các đại lượng về độ phân tán biến YearsInCurrentRole
Khoảng biến thiên (Range): 11
Phương sai (Variance): 11.007
Độ lệch chuẩn (Standard deviation): 3.318

Các đại lượng về hình dáng phân phối biến YearsInCurrentRole
Độ lệch: 0.52
Độ nhọn: -0.93
```



Hình 20: Các đại lượng thống kê mô tả và biểu đồ phân phối biến YearsInCurrentRole

Trung bình, mỗi nhân viên đã giữ vị trí hiện tại trong khoảng 4 năm. Nhiều nhất là các nhân viên đã giữ vị trí hiện tại được 2 năm, điều này có thể là dấu hiệu cho thấy có sự chuyển động và sự đổi mới liên tục trong tổ chức. Tuy nhiên, không thể bỏ qua sự đa dạng trong đội ngũ nhân viên, với một số người đã giữ vị trí hiện tại suốt 7, 8 năm.

Sự đa dạng này phản ánh sự linh hoạt của công ty trong việc chấp nhận và giữ chân cả những người mới gia nhập lẫn những nhân viên có kinh nghiệm lâu dài. Tuy nhiên, để duy trì sự cân bằng và sự ổn định, cần chú ý đến những nhân viên thay đổi vị trí nhiều lần. Điều này có thể do có sự tham gia của nhân viên mới hoặc những người luôn đặt ra yêu cầu về sự đổi mới trong sự nghiệp.

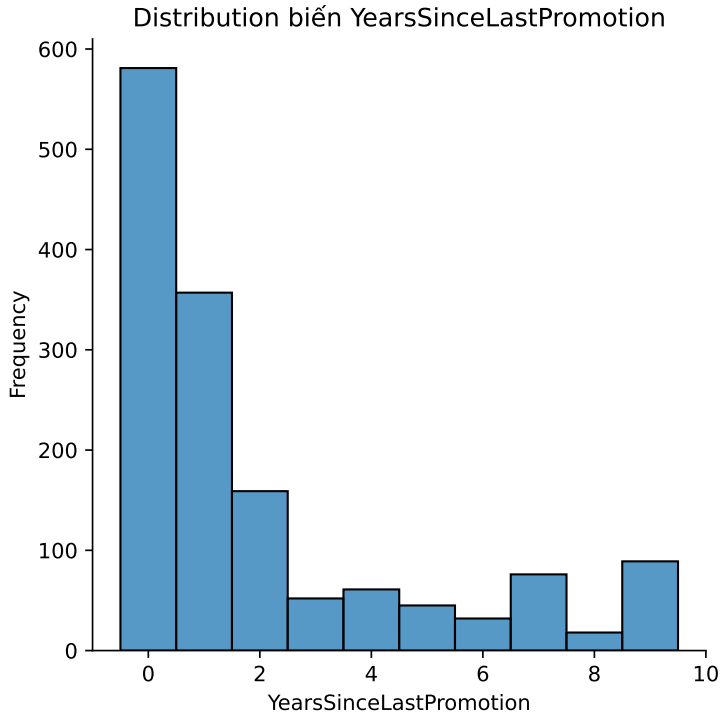
Biến YearsSinceLastPromotion

Biến YearsSinceLastPromotion thể hiện số năm kể từ lần thăng chức cuối cùng của mỗi nhân viên.

Trung bình, mỗi nhân viên đã được thăng chức cách đây 2 năm. Có đến 50% nhân viên được thăng chức từ 1 năm trở lại đây. Nhiều nhất là các nhân viên được thăng chức dưới 1 năm. Điều này chỉ ra rằng đa phần các nhân viên được thăng chức vào khoảng thời gian gần đây.

Tuy nhiên, cũng cần lưu ý đến một số nhân viên có thể đã không được thăng chức trong một khoảng thời gian dài, điều này có thể tạo ra tình trạng chán nản và có thể ảnh hưởng đến sự cam kết và giữ chân của nhân viên. Để duy trì sự hài lòng và động lực của nhân viên, cần xem xét chính sách thăng chức và đảm bảo công bằng và minh bạch trong quá trình đánh giá.

```
Các đại lượng về xu thế trung tâm của biến YearsSinceLastPromotion
Mean: 2.02
Mode: [0]
Median: 1.0

Các đại lượng về độ phân tán biến YearsSinceLastPromotion
Khoảng biến thiên (Range): 9
Phương sai (Variance): 7.354
Độ lệch chuẩn (Standard deviation): 2.712

Các đại lượng về hình dáng phân phối biến YearsSinceLastPromotion
Độ lệch: 1.44
Độ nhọn: 0.83
```

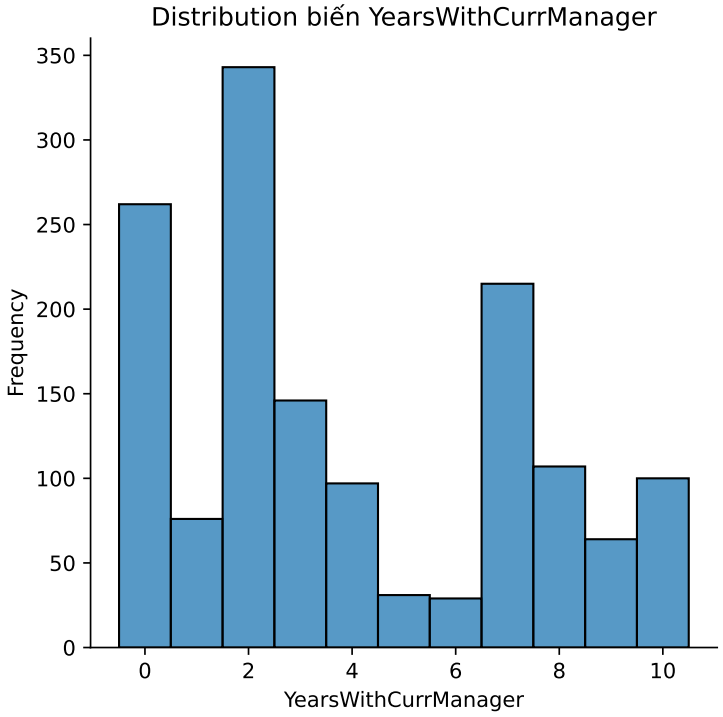


Hình 21: Các đại lượng thống kê mô tả và biểu đồ phân phối biến YearsSinceLastPromotion

Biến YearsWithCurrManager

Biến YearsWithCurrManager thể hiện số năm mà nhân viên đã làm việc với quản lý hiện tại của họ.

```
Các đại lượng về xu thế trung tâm của biến YearsWithCurrManager
Mean: 3.982
Mode: [2]
Median: 3.0

Các đại lượng về độ phân tán biến YearsWithCurrManager
Khoảng biến thiên (Range): 10
Phương sai (Variance): 10.475
Độ lệch chuẩn (Standard deviation): 3.236

Các đại lượng về hình dáng phân phối biến YearsWithCurrManager
Độ lệch: 0.43
Độ nhọn: -1.17
```



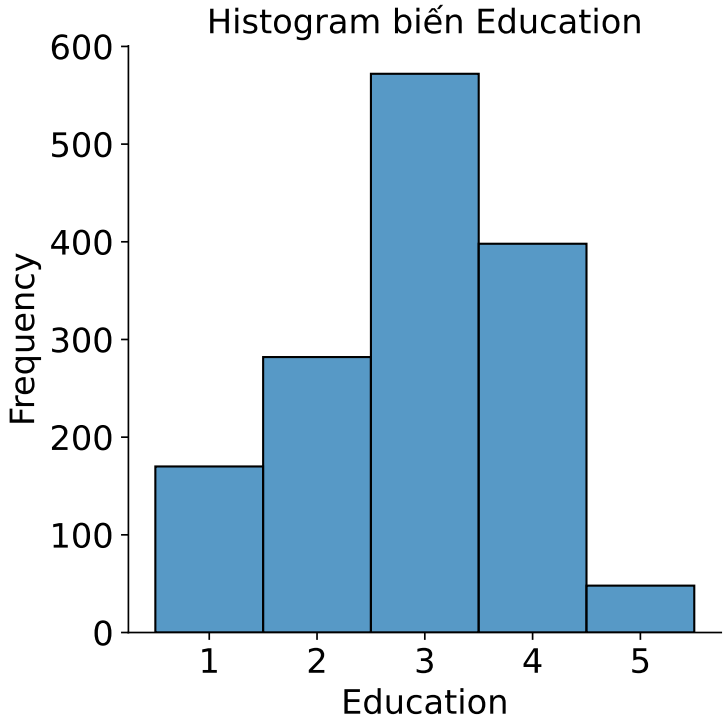
Hình 22: Các đại lượng thống kê mô tả và biểu đồ phân phối biến YearsWithCurrManager

Trung bình, mỗi nhân viên đã làm việc với quản lý hiện tại được gần 4 năm. Phân phối lệch phải nhẹ với độ lệch 0.43, độ nhọn âm chỉ ra rằng đỉnh phân phối thấp và đuôi phân phối mở rộng. Vì vậy, có sự đa dạng trong thời gian mà nhân viên đã làm việc với người quản lý hiện tại.

3.2 Biến định tính

Biến Education

```
Các đại lượng thống kê biến Education
Mode: 3
Median: 3.0
Tứ phân vị (Quartiles): [2. 3. 4.]
```



Hình 23: Các đại lượng thống kê mô tả và biểu đồ tần số biến Education

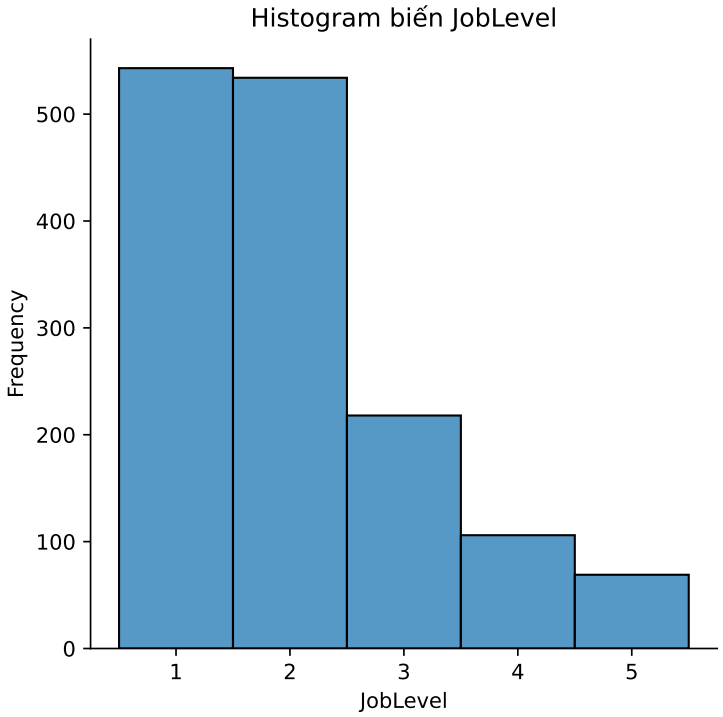
Trình độ học vấn có 5 mức độ, phần lớn nhân viên có trình độ học vấn là 3, tiếp theo là mức 2, mức 4, mức 1, và ít nhất là mức 5. Dữ liệu có sự phân tán khá lớn giữa các mức độ học vấn, do tỷ lệ phần trăm giữa mức 2 và mức 4 có sự chênh lệch đáng kể.

Các số liệu về trình độ học vấn có thể hữu ích trong quá trình quản lý nhân sự để đưa ra quyết định về đào tạo, phát triển sự nghiệp cho các nhân viên trong công ty. Nhận thấy nhóm nhân sự trình độ học vấn cấp 1 chiếm tỷ lệ đáng kể, do đó cần có chính sách cũng như tập trung đào tạo và phát triển cho nhóm này.

Biến JobLevel

Biến JobLevel cho biết cấp bậc của nhân viên trong công ty (theo quy ước của công ty).

```
Các đại lượng về xu thế trung tâm của biến JobLevel
Mode: [1]
Median: 2.0
Các đại lượng về độ phân tán biến JobLevel
Tứ phân vị (Quartiles): [1. 2. 3.]
```

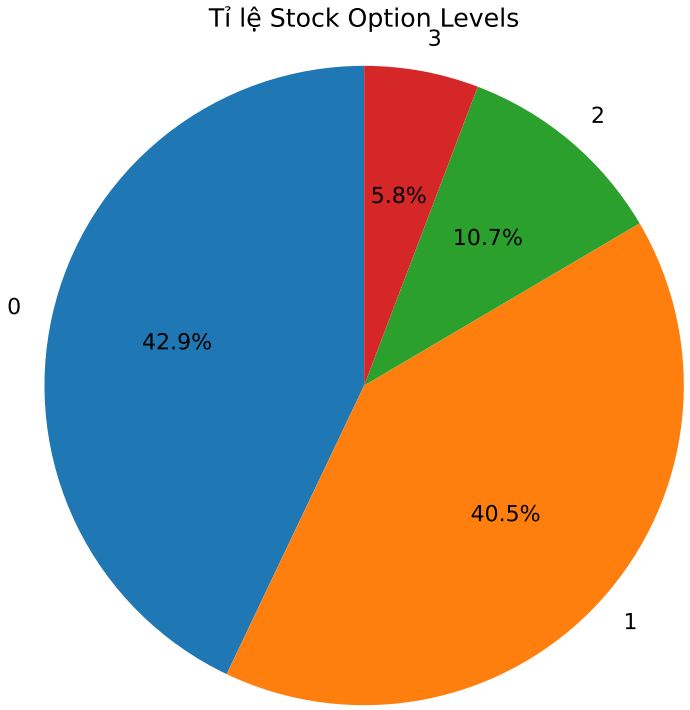


Hình 24: Các đại lượng thống kê mô tả và biểu đồ tần số biến JobLevel

Công ty có một tỷ lệ cao nhân viên ở các cấp bậc thấp, đặc biệt là cấp bậc 1 và 2. Hai cấp bậc này chiếm đến 50% số lượng nhân viên trong công ty. Điều này không hẳn là tiêu cực vì sự tập trung nhiều nhân viên ở các cấp bậc thấp có thể tạo ra cơ hội thăng tiến nội bộ cho nhân viên.

Biến StockOptionLevel

Phần lớn các nhân viên có mức Stock Option Level rơi vào khoảng 0 và 1, với gần 83% nhân viên. Mức Stock Option Level có ít nhân viên nhất là 3 với chỉ 5.8%, mức 2 có 10.7% trên tổng số nhân viên.



Hình 25: Biểu đồ tròn thể hiện tỉ lệ các giá trị Stock Option Levels

3.3 Biến định danh

Do các thuộc tính JobRole và EducationField đang có khá nhiều giá trị so với các thuộc tính định danh khác nên ta sẽ trực quan hoá riêng.

```
categorical_cols = df.select_dtypes(exclude='number')

categorical_cols =
↪  categorical_cols.drop(columns=['outlier','JobRole','EducationField'],axis=1)
```

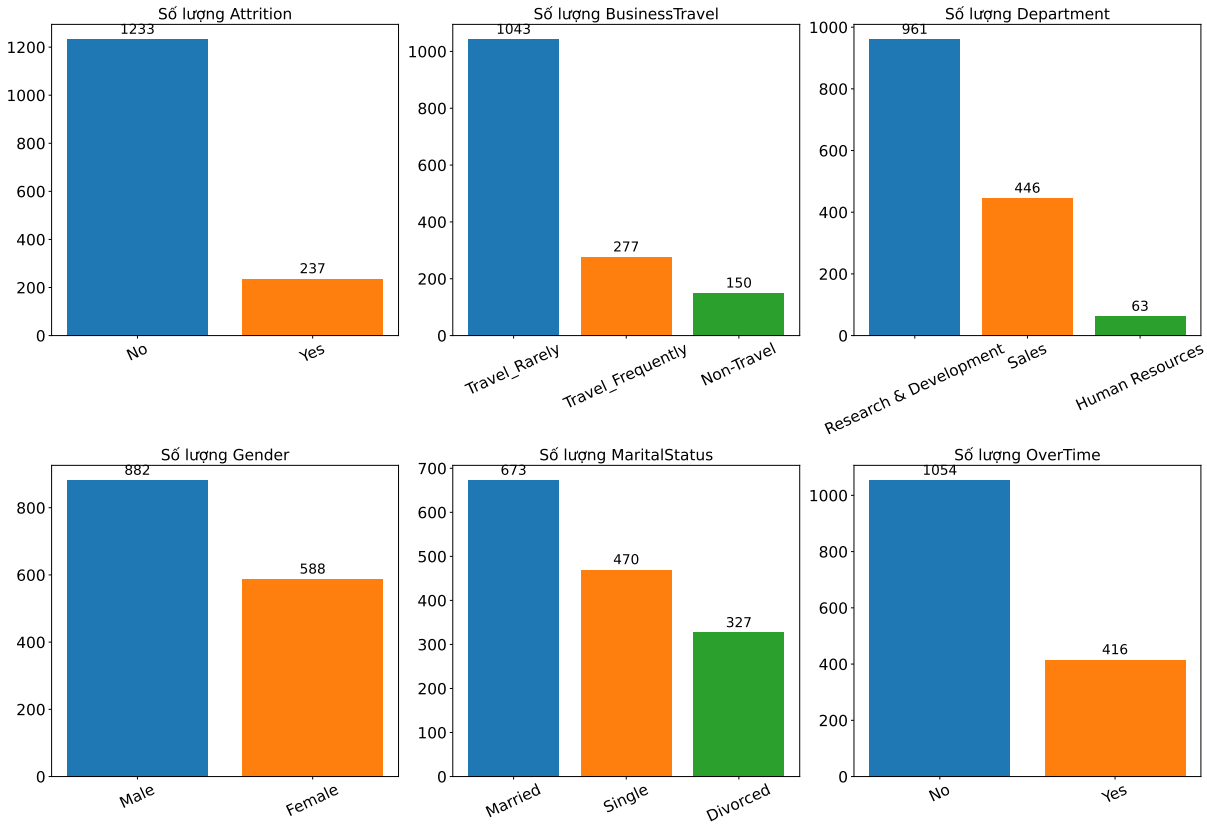
Đầu tiên ta sẽ đếm theo số lượng giá trị của từng thuộc tính định tính, bao gồm Attrition, BusinessTravel, Department, Gender, MaritalStatus, và Overtime.

```
colors = ['tab:blue', 'tab:orange', 'tab:green', 'tab:red', 'tab:purple',
          'tab:brown', 'tab:pink', 'tab:gray', 'tab:olive', 'tab:cyan']

fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(20, 20))

```

```
axes = axes.flatten()

for i, col in enumerate(categorical_cols.columns):
    value_counts = df[col].value_counts()
    bars = axes[i].bar(value_counts.index, value_counts.values,
    ↪  color=colors[:len(value_counts)])
    axes[i].set_title(f'Số lượng {col}',fontsize=16)
    axes[i].tick_params(axis='both',labelsize=16)
    axes[i].set_xticklabels(value_counts.index, fontsize=16)
    for bar in bars:
        height = bar.get_height()
        axes[i].annotate(f'{height}', xy=(bar.get_x() + bar.get_width() / 2,
        ↪  height),
                         xytext=(0, 3), textcoords='offset points',
                         ↪  ha='center', va='bottom',fontsize=16)

for j in range(len(categorical_cols.columns), 9):
    axes[j].axis('off')

plt.tight_layout()
plt.savefig('Thống kê theo các biến định tính 1.pdf')
plt.show()
```

Trực quan hoá số liệu thống kê các biến định tính trong dữ liệu cho ta thấy tỉ lệ nhân viên rời bỏ là khoảng 16% so với toàn bộ tổng số nhân viên. Đối với biến BusinessTravel, đa số các nhân viên hiếm khi đi công tác, với 1043 nhân viên. Trong đó, số nhân viên không có chuyến công tác nào là 150. Bộ phận (department) có nhiều nhân viên làm việc nhất là bộ phận Research % Development với 961 nhân viên, trong khi đó, số lượng nhân viên cho 2 bộ phận Sales và Human Resources lần lượt là 446 và 63.

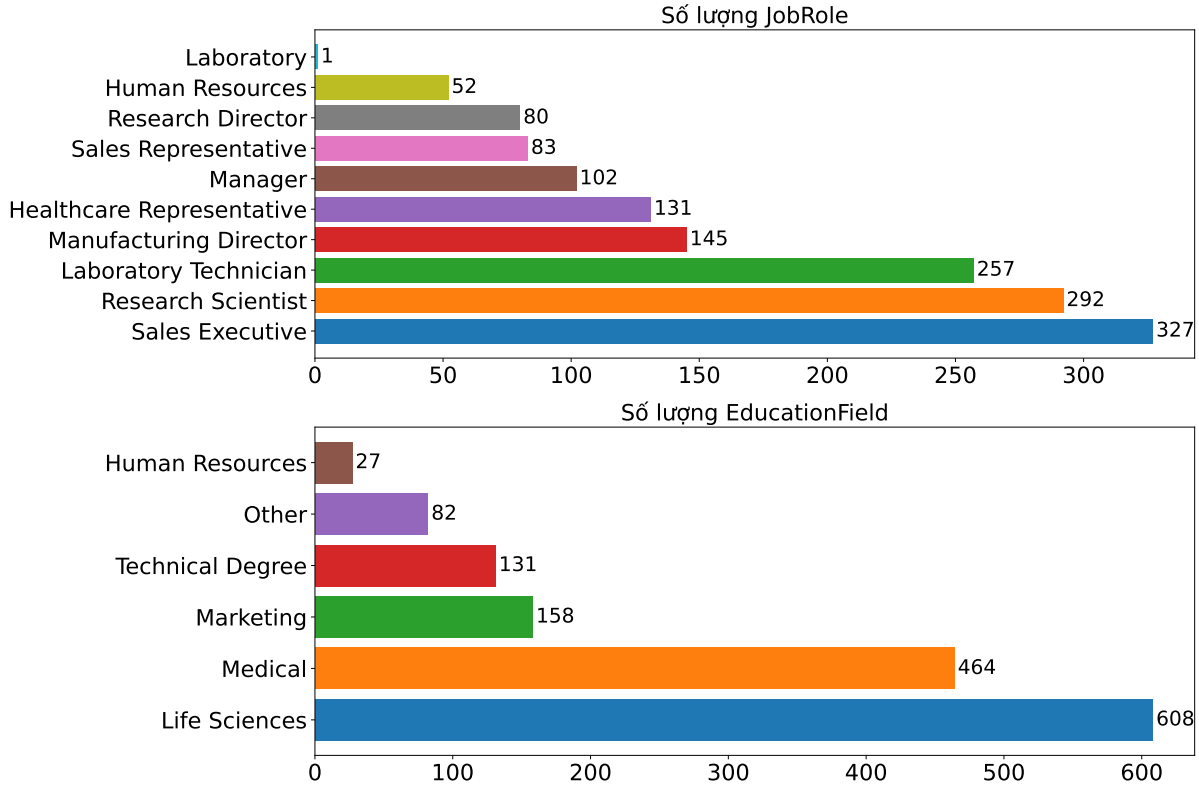
Về số lượng giới tính, nam giới chiếm nhiều hơn nữ giới với 882 nhân viên, so với số lượng nhân viên nữ giới là 588. Thêm vào đó, đa phần các nhân viên đã kết hôn với 673 nhân viên, chiếm 45% tổng số nhân viên. Số lượng nhân viên vẫn còn độc thân và đã li hôn lần lượt là 470 và 327. Đáng chú ý, số lượng nhân viên làm thêm giờ là 1054, nhiều gấp 2.5 lần số lượng nhân viên không làm thêm giờ với 416 nhân viên.



Hình 26: Thống kê các biến định danh

```
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(20, 10))
axes = axes.flatten()

for i, col in enumerate(categorical_cols.columns):
    value_counts = df[col].value_counts()
    bars = axes[i].barh(value_counts.index, value_counts.values,
    ↪  color=colors[:len(value_counts)])
    axes[i].set_title(f'Số lượng {col}',fontsize=16)
    axes[i].tick_params(axis='both',labelsize=16)
    axes[i].set_yticklabels(value_counts.index, fontsize=16)
    for bar in bars:
        width = bar.get_width()
        axes[i].annotate(f'{int(width)}', xy=(width, bar.get_y() +
        ↪  bar.get_height() / 2),
                         xytext=(3, 0), textcoords='offset points', ha='left',
                         ↪  va='center', fontsize=16)

plt.tight_layout()
plt.savefig('Thống kê theo các biến định tính 2.pdf')
plt.show()
```

Đa số các nhân viên làm việc ở vị trí Sales Executive với 327 nhân viên. Các vị trí khác có số lượng nhân viên nhiều bao gồm Research Scientist, Laboratory Technician, và Manufacturing Director với lần lươt 292, 257, và 145 nhân viên. Đáng chú ý, chỉ có 1 nhân viên ở vị trí Laboratory.



Hình 27: Thống kê theo các biến định danh

Đa phần các nhân viên có lĩnh vực giáo dục thuộc Life Sciences, với 608 nhân viên, sau đó là Medical và Marketing, với 464 và 158 nhân viên. Lĩnh vực giáo dục ít nhân viên theo đuổi nhất là Human Resources với 27 nhân viên.

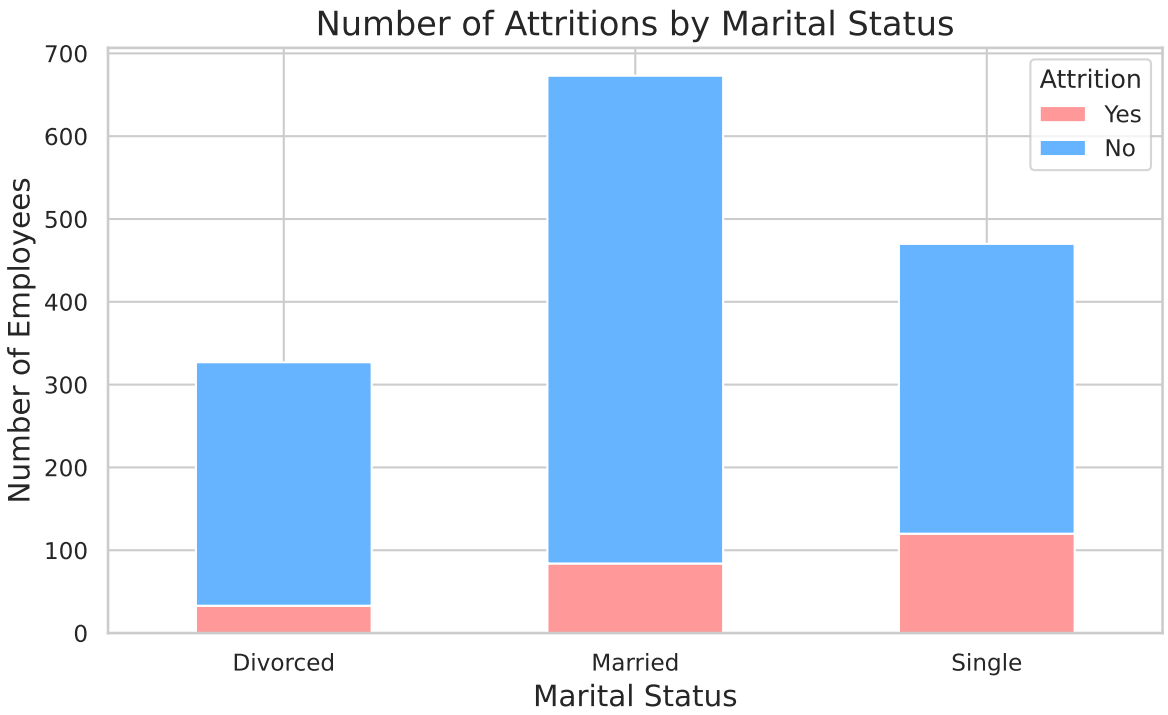
4 CHƯƠNG 4: PHÂN TÍCH ĐA BIẾN

4.1 Trực quan hoá dữ liệu

4.1.1 Biểu đồ số lượng rời bỏ của nhân viên theo tình trạng hôn nhân

```
marital_status_counts = df.groupby(['MaritalStatus', 'Attrition']).size().unstack()
marital_status_inverted = marital_status_counts.loc[:, ::-1]
marital_status_inverted.plot(kind='bar', stacked=True, figsize=(8, 5), color=['#ff9999','#66b3ff'])
plt.title('Number of Attritions by Marital Status', fontsize=16)
plt.xlabel('Marital Status',fontsize=14)
```

```
plt.ylabel('Number of Employees',fontsize=14)
plt.xticks(rotation=0)
plt.legend(title='Attrition', loc='upper right')
plt.tight_layout()
plt.savefig(f'figs/Number of Attritions by Marital Status.pdf')
plt.show()
```



Hình 28: Biểu đồ số lượng rời bỏ của nhân viên theo tình trạng hôn nhân

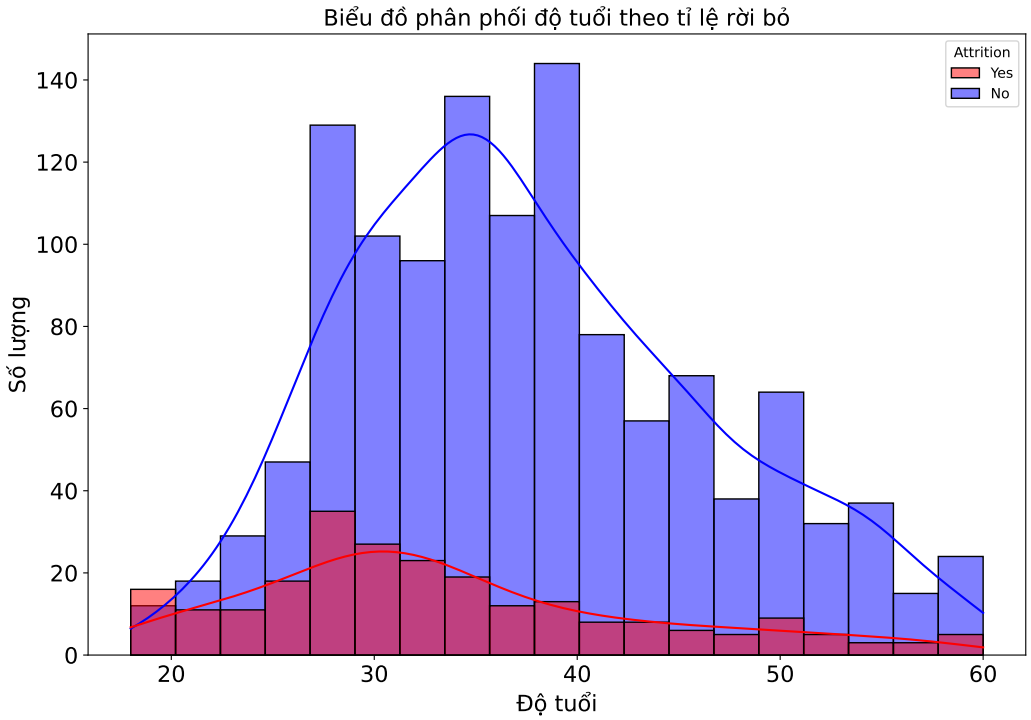
Biểu đồ chồng thể hiện số lượng nhân viên ở lại và rời bỏ công việc phân theo tình trạng hôn nhân. Ta nhận thấy số lượng nhân viên không rời đi ở các nhóm luôn chiếm đa số.

Mặc dù số lượng nhân viên còn độc thân ít hơn nhân viên đã kết hôn nhưng số lượng người rời bỏ công việc lại cao hơn, có thể cho rằng nhóm độc thân có thể có xu hướng rời đi nhiều hơn. Số lượng nhân viên đã kết hôn cao nhất và số nhân viên rời đi chiếm phần nhỏ, những người kết hôn ít có khả năng rời đi hơn so với nhóm độc thân. Nhóm nhân viên đã ly hôn có số lượng thấp nhất trong bảng và số lượng rời bỏ ở mức trung bình.

4.1.2 Biểu đồ phân phối độ tuổi theo tỉ lệ rời bỏ

```
plt.figure(figsize=(12, 8))

sbn.histplot(data=df, x='Age', hue='Attrition', kde=True, palette={'Yes':
↪  'red', 'No': 'blue'})

plt.title('Biểu đồ phân phối độ tuổi theo tỉ lệ rời bỏ',fontsize=16)
plt.xlabel('Độ tuổi',fontsize=16)
plt.ylabel('Tần suất',fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.savefig('Biểu đồ phân phối độ tuổi theo tỉ lệ rời bỏ.pdf')
plt.show()
```

Đa phần các nhân viên ở độ tuổi từ dưới 30 đến 40 tuổi. Tỉ lệ nhân viên rời bỏ nhiều nhất rơi vào khoảng độ tuổi 30.

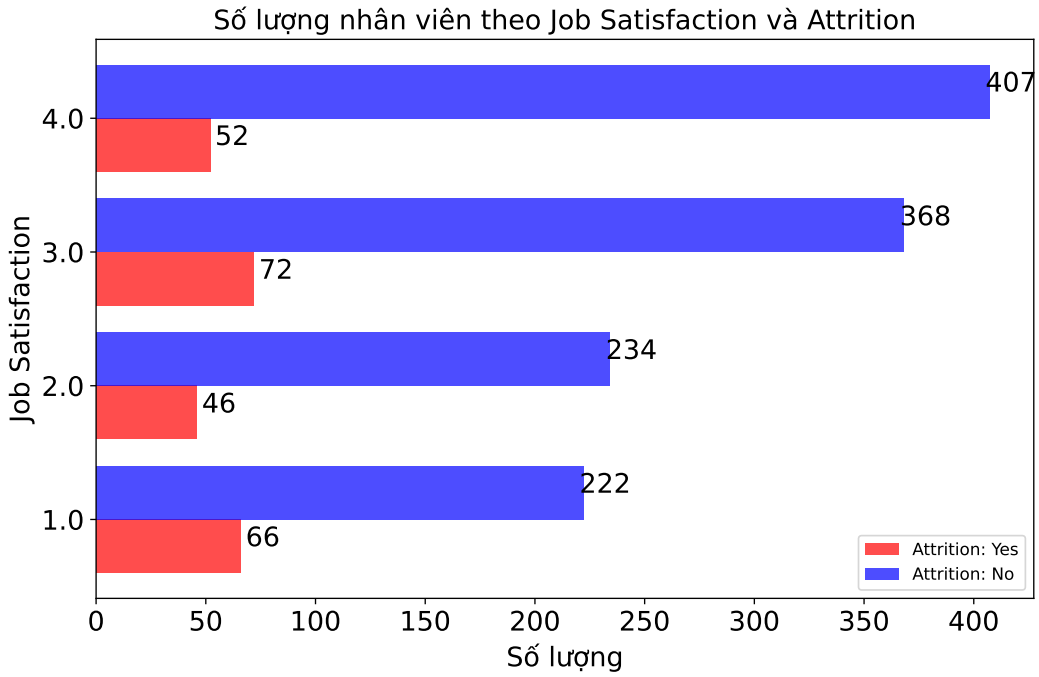


Hình 29: Biểu đồ phân phối độ tuổi theo tỉ lệ rời bỏ

4.1.3 Số lượng nhân viên theo Job Satisfaction và Attrition

```
counts = df.groupby(['JobSatisfaction', 'Attrition']).size().unstack().reset_index()

positions = np.arange(len(counts['JobSatisfaction']))

plt.figure(figsize=(10, 6))

plt.barh(positions - 0.2, counts['Yes'], height=0.4, label='Attrition: Yes', color='red', alpha=0.7)
plt.barh(positions + 0.2, counts['No'], height=0.4, label='Attrition: No', color='blue', alpha=0.7)

for i, count in enumerate(counts['Yes']):
    plt.text(count + 10, i - 0.2, str(count), color='black', ha='center',fontsize=16)

for i, count in enumerate(counts['No']):
    plt.text(count + 10, i + 0.2, str(count), color='black', ha='center',fontsize=16)

plt.yticks(positions, counts['JobSatisfaction'],fontsize=16)

plt.title('Số lượng nhân viên theo Job Satisfaction và Attrition',fontsize=16)
plt.xlabel('Số lượng',fontsize=16)
plt.ylabel('Job Satisfaction',fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.legend()
plt.savefig('Số lượng nhân viên theo Job Satisfaction và Attrition.pdf')
plt.show()
```

Số lượng nhân viên rời bỏ cao nhất nằm ở nhóm nhân viên đánh giá mức độ hài lòng về công việc ở mức 3 (trên thang điểm 4, 1 tương ứng với không hài lòng nhất và 4 tương ứng với hài lòng nhất) với 72 nhân viên. Số nhân viên rời bỏ đánh giá mức độ hài lòng với công việc ở mức thấp nhất là 66, số nhân viên đánh giá mức độ hài lòng công việc ở mức cao nhất và rời bỏ là 52.



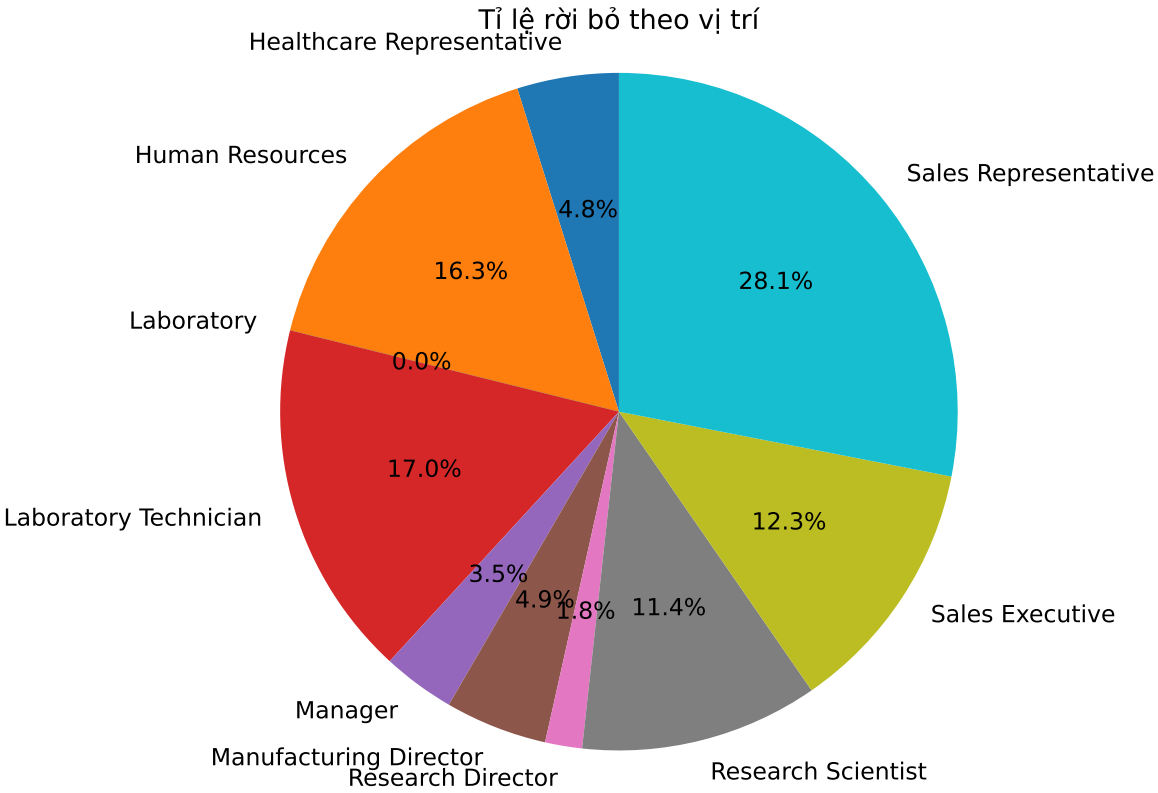
Hình 30: Số lượng nhân viên theo Job Satisfaction và Attrition

4.1.4 Tỉ lệ rời bỏ theo vị trí

```
df['JobRole'] = df['JobRole'].str.strip()

attrition_counts = df.groupby(['JobRole', 'Attrition']).size().unstack()

attrition_counts = attrition_counts.fillna(0)
attrition_proportions = attrition_counts['Yes'] / (attrition_counts['Yes'] +
↪  attrition_counts['No'])

plt.figure(figsize=(10, 8))
plt.pie(attrition_proportions, labels=attrition_proportions.index,
↪  autopct='%1.1f%%', startangle=90,textprops={'fontsize': 14})
plt.title('Tỉ lệ rời bỏ theo vị trí',fontsize=16)
plt.axis('equal')
plt.savefig('Tỉ lệ rời bỏ theo vị trí.pdf')
plt.show()
```

Lượng nhân viên rời bỏ cao nhất là các nhân viên ở vị trí Sales Representative , với 28.1%. Tuy nhiên, đây cũng là vị trí có nhiều nhân viên làm việc nhất. Theo sau là các nhân viên ở vị trí Laboratory Technician và Human Resources. Phần trăm nhân viên ít rời bỏ nhất thường là các vị trí ở cấp cao. Cụ thể, số manager rời bỏ là 3.5%, số

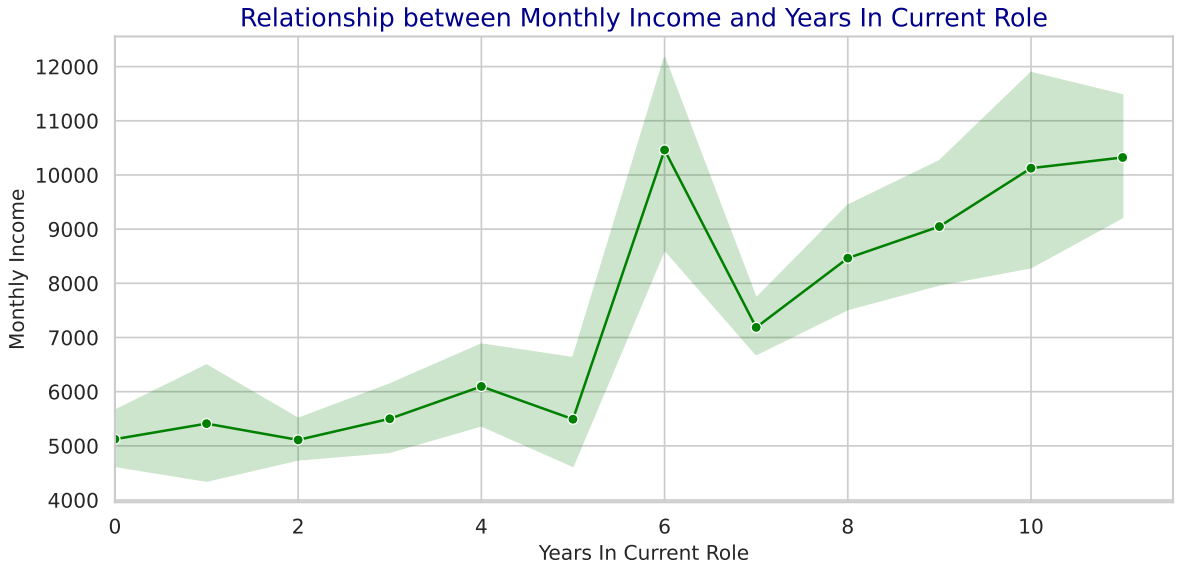
Manufacturing Director rời bỏ là 4.9% và số Research Director rời bỏ là 1.8%.



Hình 31: Tỉ lệ rời bỏ theo vị trí

4.1.5 Biểu đồ thể hiện mối quan hệ giữa YearsInCurrentRole và MonthlyIncome

```
plt.figure(figsize=(10, 5))
ax = sns.lineplot(data=df, x='YearsInCurrentRole', y='MonthlyIncome',
 ↪  marker='o', linestyle='-', color='green')
ax.set_xlim(left=0)
plt.xlabel('Years In Current Role',fontsize=12)
plt.ylabel('Monthly Income',fontsize=12)
plt.title('Relationship between Monthly Income and Years In Current
 ↪  Role',color='darkblue',fontsize=15)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.tight_layout()
plt.savefig('figs/Relationship between Monthly Income and Years In Current
 ↪  Role.pdf')
plt.show()
```



Hình 32: Biểu đồ thể hiện mối quan hệ giữa "YearsInCurrentRole"và "MonthlyIncome"

Nhìn chung, có vẻ có một xu hướng tăng về mặt thu nhập hàng tháng trung bình khi số năm ở vị trí hiện tại tăng lên. Điều này có thể liên quan đến việc nhân viên có cơ hội được thăng cấp và nhận mức lương cao hơn khi họ ở lâu hơn ở một vị trí. Tuy nhiên, có sự biến động đáng kể trong thu nhập hàng tháng ở một số năm cụ thể. Điều này có thể đang phản ánh sự chênh lệch thu nhập hàng tháng giữa các vị trí công việc.

4.1.6 Sự tương quan giữa thâm niên làm việc và thu nhập

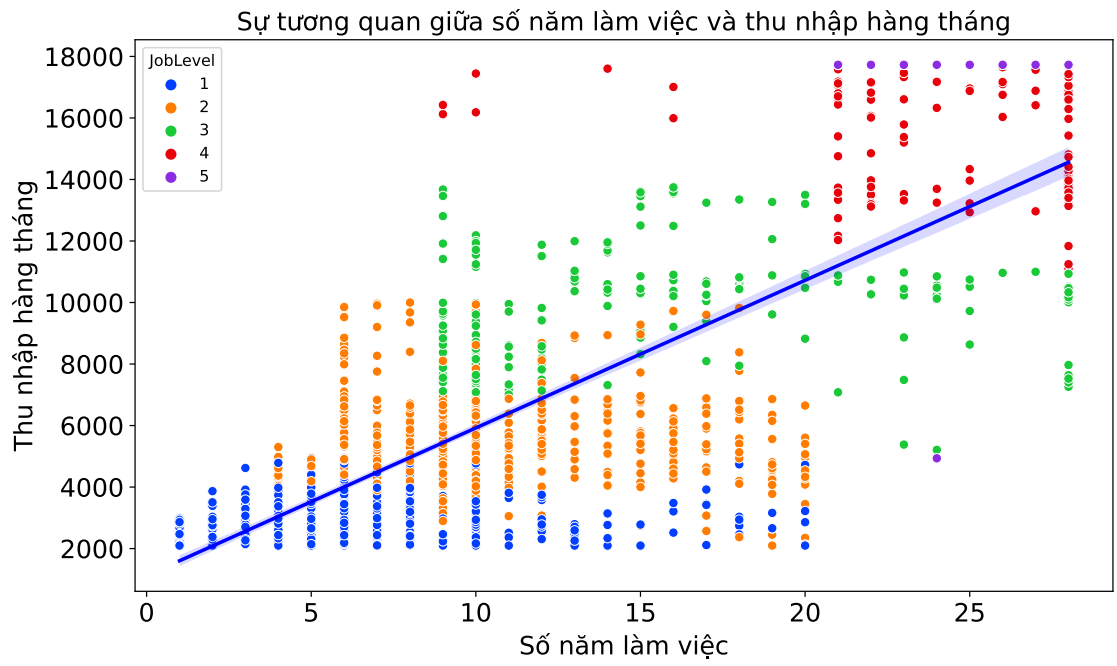
Tính hệ số tương quan Spearman giữa biến TotalWorkingYear và MonthlyIncome:

```
correlation = df['TotalWorkingYears'].corr(df['MonthlyIncome'],
    method='spearman').round(3)
print(f'Hệ số tương quan Spearman giữa biến TotalWorkingYears và biến
    MonthlyIncome: {correlation}')
```

```
Hệ số tương quan Spearman giữa biến TotalWorkingYears và biến MonthlyIncome: 0.707
```

Hệ số tương quan là 0.707 cho thấy hai biến này có mối tương quan thuận (cùng chiều), bởi vì hệ số tương quan dương và mức độ tương quan là khá mạnh, do hệ số gần 1. Khi TotalWorkingYears tăng lên thì MonthlyIncome có xu hướng tăng theo. Ngược lại, khi TotalWorkingYears giảm đi thì MonthlyIncome cũng có xu hướng giảm. Cụ thể hơn, để có thể xem mối tương quan của từng điểm dữ liệu, ta sử dụng Scatter plot thể hiện sự tương quan giữa biến TotalWorkingYear và MonthlyIncome (phân loại theo JobLevel).

```
plt.figure(figsize=(10, 6))
sbn.scatterplot(x=df['TotalWorkingYears'], y=df['MonthlyIncome'],
 ↪  hue=df['JobLevel'],palette='bright')
sbn.regplot(x=df['TotalWorkingYears'], y=df['MonthlyIncome'], scatter=False,
 ↪  color='blue')
plt.title('Sự tương quan giữa số năm làm việc và thu nhập hàng tháng', fontsize
 ↪  = 16)
plt.xlabel('Số năm làm việc', fontsize = 16)
plt.ylabel('Thu nhập hàng tháng', fontsize = 16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.tight_layout()
plt.savefig(f"figs/Sự tương quan giữa số năm làm việc và thu nhập hàng
 ↪  tháng.pdf")
plt.legend(title='Cấp độ công việc:', fontsize = 13)
plt.show()
```



Hình 33: Biểu đồ thể hiện tương quan giữa số năm làm việc và thu nhập hàng tháng

Các chỉ số thống kê mô tả cho mức thu nhập hàng tháng của các nhân viên làm việc dưới 5 năm:

```
df[df['TotalWorkingYears'] <= 5]['MonthlyIncome'].describe()
```

```
count     316.000000
mean     3017.141772
std       902.445731
min      2097.900000
25%      2304.250000
50%      2694.500000
75%      3731.750000
max      5301.000000
```

Đối với các nhân viên đã làm việc từ dưới 5 năm, đa phần họ đều ở cấp độ 1 hoặc 2, thu nhập hàng tháng của họ dao động ở mức thấp nhất, trung bình xấp xỉ 3,000 dollars. Các nhân viên đạt cấp độ 2 có mức thu nhập cao hơn một chút so với nhân viên cấp độ 1 và chỉ có một nhân viên cấp độ 1 có mức thu nhập cao vượt trội so với các nhân viên cấp độ 1 làm việc dưới 5 năm khác.

Các chỉ số thống kê mô tả cho mức thu nhập hàng tháng của các nhân viên làm việc từ 6 đến 20 năm:

```
df[(df['TotalWorkingYears'] >= 6)&(df['TotalWorkingYears'] <=
↪  20)]['MonthlyIncome'].describe()
```

```
count      947.000000
mean      5713.478141
std       2874.489428
min       2097.900000
25%       3452.000000
50%       5231.000000
75%       6880.000000
max      17603.000000
```

Các nhân viên đã làm việc từ 6 đến 20 năm trải dài ở cấp độ 1 đến 4, đa phần là nhân viên cấp độ 2 và 3, chỉ có 7 nhân viên làm việc ở cấp độ 4. Ở cấp độ càng cao thì lương cũng có xu hướng cao hơn, đối với các nhân viên cấp độ 4, mức thu nhập cao nhất lên

tới 17,603 dollars mặc dù thu nhập trung bình hàng tháng của các nhân viên đã có kinh nghiệm làm việc trong khoảng thời gian này chỉ là 5,713 dollars.

Các chỉ số thống kê mô tả cho mức thu nhập hàng tháng của các nhân viên làm việc trên 20 năm:

```
df[df['TotalWorkingYears'] > 20]['MonthlyIncome'].describe()
```

```
count      207.00000
mean     15024.92657
std       3261.39244
min       4936.00000
25%      13218.50000
50%      16752.00000
75%      17727.70000
max      17727.70000
```

Các nhân viên làm việc trên 20 năm đều đạt cấp độ từ 3 đến 5. Mức thu nhập trung bình hàng tháng của họ cũng cao hơn hẳn so với các nhân viên làm việc dưới 20 năm (15,024 dollars). Tuy nhiên, độ lệch chuẩn của nhóm nhân viên này khá cao (3,261 dollars), do có sự chênh lệch nhiều về mức thu nhập giữa các nhân viên cấp độ 3 so với nhân viên cấp độ 4 và 5. Ở cấp độ càng cao thì thu nhập càng cao, chỉ có một trường hợp nhân viên thuộc cấp độ 5 (màu tím) nhưng lại có mức thu nhập hàng tháng thấp khi so với các nhân viên làm việc trên 20 năm. Mức thu nhập thấp có thể đến từ các yếu tố như năng lực chưa đúng với cấp độ hoặc chính sách trả lương của công ty chưa hợp lý. Nhưng không thể ngoại trừ trường hợp nhân viên này mới vào làm tại công ty nên chưa có mức thu nhập cao và đang chờ xem xét tăng lương. Chính vì thế, ta cần xem số năm làm việc của nhân viên này tại công ty và quan trọng hơn hết là liệu nhân viên này có rời đi hay không:

```
# Lọc ra nhân viên làm việc trên 20 năm, đạt cấp độ 5 nhưng có mức thu nhập
 ↪ hàng tháng thấp
inc = df[(df['TotalWorkingYears'] >
 ↪ 20)&(df['JobLevel']==5)]['MonthlyIncome'].min()
df[(df['TotalWorkingYears'] >
 ↪ 20)&(df['JobLevel']==5)&(df['MonthlyIncome']==inc)][['Attrition',
'YearsAtCompany']]
```

	Attrition	YearsAtCompany
568	Yes	5

Nhân viên này không nằm trong trường hợp nhân viên mới chờ xem xét tăng lương vì nhân viên này đã làm việc tại công ty được 5 năm nhưng mức lương vẫn chưa cao. Vậy có thể suy đoán rằng nhân viên rời đi với lý do mức lương chưa phù hợp hoặc chưa được tăng lương trong một khoảng thời gian dài gắn bó với công ty.

4.2 Kiểm định giả thuyết

Khi phân tích dữ liệu, chúng ta cần tập trung vào mức độ ảnh hưởng của các biến độc lập lên hai biến mục tiêu quan trọng: MonthlyIncome (Thu nhập hàng tháng) và Attrition (Tỉ lệ nghỉ việc). Cả hai biến này đều đóng vai trò quan trọng trong đánh giá hiệu suất của tổ chức.

Mục tiêu chính của chương này là kiểm tra những tác động có thể phát sinh từ các biến khác nhau đối với MonthlyIncome và Attrition. Điều này giúp chúng ta không chỉ hiểu rõ hơn về sự tương tác giữa các yếu tố, mà còn là cơ sở cho việc ra quyết định và triển khai các biện pháp cải thiện hiệu suất.

Nhóm đã quyết định sử dụng mức ý nghĩa là 0.05 để thực hiện kiểm định giả thuyết. Quyết định này không chỉ là một tiêu chí thống kê, mà còn là sự cân nhắc kỹ lưỡng giữa việc phát hiện sự khác biệt có ý nghĩa và nguy cơ chấp nhận sai lầm.

4.2.1 Kiểm định sự khác biệt về tuổi giữa nhân viên ở lại và nhân viên rời đi

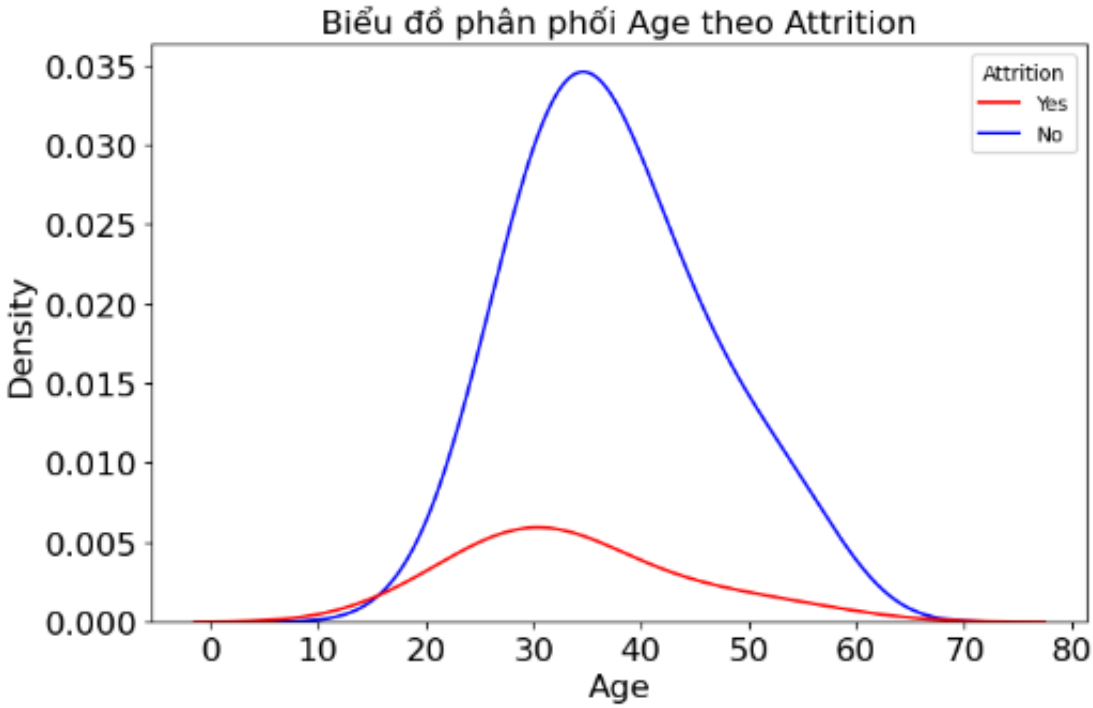
Kiểm định về mối quan hệ giữa độ tuổi và quyết định rời đi hay ở lại của nhân viên giúp chúng ta có những phân tích về sự ổn định của các độ tuổi khác nhau hay xu hướng tìm kiếm cơ hội việc làm mới của họ. Từ đó công ty có thể điều chỉnh phù hợp chẳng hạn như các chính sách phúc lợi để giữ chân người giỏi hoặc tăng lợi thế cạnh tranh trong tuyển dụng nhân sự.

Để kiểm định sự khác biệt giữa giá trị trung bình của hai nhóm, dùng kiểm định Student's T-test cho 2 mẫu độc lập. Các giả định cần thỏa mãn của kiểm định này là:

- Các quan sát của 2 nhóm độc lập với nhau
- Phân phối của 2 nhóm xấp xỉ phân phối chuẩn
- Phương sai đồng nhất: phương sai 2 nhóm bằng nhau

- Lấy mẫu ngẫu nhiên

Biến Age có phân phối gần giống với phân phối chuẩn, tuy nhiên phương sai 2 mẫu không đồng nhất nên có thể dùng kiểm định Welch's T-test. Welch's T-test là một lựa chọn thay thế cho Student's T-test khi giả định về phương sai đồng nhất không thỏa mãn.



Hình 34: Biểu đồ phân phối của biến Age theo Attrition

Giả thuyết được đặt ra là những người trẻ có xu hướng ít ổn định hơn trong việc gắn bó với nơi làm việc. Vậy tiến hành kiểm định 1 phía với giả thuyết:

- H_0: Độ tuổi của nhân viên rời đi lớn hơn hoặc bằng độ tuổi nhân viên ở lại.
- H_a: Độ tuổi của nhân viên rời đi bé hơn độ tuổi của nhân viên ở lại.

```
#Phân nhóm
df_yes = df[df['Attrition'] == 'Yes']
df_no = df[df['Attrition'] == 'No']


#Chọn mức ý nghĩa
alpha            = .05
confidence_level = 1 - alpha


## So sánh phương sai 2 mẫu dữ liệu
```

```
print(f'Var(Yes) = {df_yes.Age.var(ddof = 1):.2f}; Var(No) = {df_no.Age.var(ddof = 1):.2f}')
```

```
Var(Yes) = 93.88; Var(No) = 79.00
```

```
## Kiểm định T, 2 mẫu độc lập, khác phương sai
t, p = stats.ttest_ind(df_yes.Age, df_no.Age, equal_var = False)
## Kết luận theo phương pháp p-value (trị số p)
if (p < alpha):
    print(f'Trị số p = {p:.4f} < {alpha:.4f} nên bác bỏ H0 ==> Muy[1] <> Muy[2]')
else:
    print(f'Trị số p = {p:.4f} >= {alpha:.4f} KHÔNG bác bỏ H0 ==> Mean_1 = Mean_2')
```

```
Trị số p = 0.0000 < 0.0500 nên bác bỏ H0 ==> Muy[1] < Muy[2]
```

Kết luận: Với độ tin cậy 95%, tuổi trung bình của nhóm nhân viên rời đi thật sự trẻ hơn so với nhóm nhân viên ở lại. Điều này phản ánh rằng người trẻ có xu hướng hay "nhảy việc" hơn.

Nguyên nhân có thể là vì những người trẻ thích trải nghiệm và họ ít bị ràng buộc về tư tưởng hơn so với thế hệ đi trước nên có phần cởi mở với việc tìm kiếm một nơi làm việc mới. Bên cạnh đó, nhóm nhân viên trẻ tuổi này có thể bao gồm cả nhân viên chính thức lẫn thực tập sinh, và tỷ lệ thực tập sinh tiếp tục gắn bó với công ty sau kỳ thực tập cũng là một vấn đề cần tìm hiểu. Tuy nhiên phạm vi thông tin bộ dữ liệu này không đủ để tiến hành khảo sát. Vậy, để tìm hiểu sâu hơn về nguyên nhân để đưa ra những đề xuất phù hợp cần có kiến thức chuyên môn của lĩnh vực quản trị nhân lực.

4.2.2 Kiểm định giả thuyết về sự liên quan giữa giới tính và sự rời đi

Mục tiêu kiểm định nhằm xác định liệu có mối liên hệ thống kê giữa giới tính và quyết định rời bỏ công việc của nhân viên hay không.

- Giới tính (Gender): Nam, Nữ
- Rời bỏ (Attrition): Yes, No

Để kiểm định giả thuyết "Giới tính và Sự rời đi của nhân viên không có mối quan hệ với nhau", ta sẽ tiến hành kiểm định Chi-square với các giả thuyết như sau:

- H0: Giới tính và Sự rời đi của nhân viên không có mối quan hệ với nhau (độc lập nhau)
- H1: Giới tính và Sự rời đi của nhân viên có mối quan hệ với nhau (phụ thuộc nhau)

Trước hết, ta tạo bảng thống kê số lần xuất hiện của các quan sát cho hai biến "Gender" và "Attrition":

```
contingency_table_gender=pd.crosstab(df['Gender'],df['Attrition'])
print(contingency_table_gender)
```

Attrition	No	Yes
Gender		
Female	501	87
Male	732	150

Thực hiện kiểm định Chi-square với mức ý nghĩa 5%:

```
alpha = 0.05
chi2, p, dof, expected = chi2_contingency(contingency_table_gender)
print('\nH0: Giới tính và Sự rời đi của nhân viên không có mối quan hệ với nhau (độc lập nhau)')
print('H1: Giới tính và Sự rời đi của nhân viên có mối quan hệ với nhau (phụ thuộc nhau)')
print(f'\nAlpha: {alpha} (Mức độ tin cậy 95%)')
print(f'Chi-square value: {chi2:.4f}')
print(f'p-value: {p:.4f}')
if (p < alpha):
    print(f'Trị số p = {p:.4f} < {alpha:.4f} nên bác bỏ H0 ==> (Gender, Attrition) PHỤ THUỘC')
else:
    print(f'Trị số p = {p:.4f} >= {alpha:.4f} KHÔNG bác bỏ H0 ==> (Gender, Attrition) ĐỘC LẬP')
```

Kết quả thu được như sau:

```
Alpha: 0.05 (Mức độ tin cậy 95%)
Chi-square value: 1.1170
p-value: 0.2906
Trị số p = 0.2906 >= 0.0500 KHÔNG bác bỏ H0
```

Vì giá trị p (0.2906) lớn hơn mức alpha (0.05), không có đủ bằng chứng thống kê để bác bỏ giả thuyết không H0. Do đó, kết luận là giới tính và quyết định rời đi của nhân viên là độc lập; không có bằng chứng đáng tin cậy để chứng minh rằng giới tính ảnh hưởng đến quyết định rời bỏ công việc của nhân viên trong mẫu dữ liệu này.

4.2.3 Kiểm định giả thuyết về sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên

Thực hiện kiểm định giả thuyết về mối quan hệ giữa phòng ban làm việc và quyết định rời đi của nhân viên không chỉ là một bước quan trọng để đánh giá mối quan hệ giữa hai biến, mà còn giúp tìm ra những lợi ích quan trọng trong quản lý nhân sự.

Thông qua việc kiểm định giả thuyết này, chúng ta có thể xác định liệu phòng ban có sự ảnh hưởng đáng kể đến quyết định nghỉ việc của nhân viên hay không. Nếu có sự ảnh hưởng, điều này là dấu hiệu cho công ty biết rằng cần tiến hành các nghiên cứu chi tiết để hiểu rõ hơn về nguyên nhân gây ra sự chênh lệch. Dựa trên kết quả này, chúng ta có thể thực hiện những điều chỉnh và ứng dụng các biện pháp quản lý nhân sự phù hợp để giữ chân nhân viên và cải thiện môi trường làm việc.

Để kiểm định giả thuyết "Không có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên", ta sẽ tiến hành kiểm định Chi-square, bởi vì đây là mối quan hệ giữa hai biến Categorical. Với giả thuyết H0 là "Không có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên." hay "biến Department và biến Attrition độc lập", giả thuyết đối là "Có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên." hay "biến Department và biến Attrition phụ thuộc".

Trước khi tiến hành kiểm định Chi-square, cần có bước tạo bảng crosstab thể hiện số lượng nhân viên rời bỏ và ở lại theo mỗi phòng ban:

```
1 crosstab = pd.crosstab(df['Attrition'], df['Department'])
2 print(crosstab)
```

Department	Human Resources	Research & Development	Sales
Attrition			
No	51	828	354
Yes	12	133	92

Thực hiện kiểm định Chi-square:

```
## Kiểm định Chi-square
print('H0: Không có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên.')
print('H1:  Có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên.')
stat, p, dof, expected = stats.chi2_contingency(crosstab)
print(f'Trị số p = {p:4f}')
if p >= 0.05:
  print("Vì p >= alpha nên không bác bỏ H0")
else:
  print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Không có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên.
H1: Có sự liên quan giữa phòng ban làm việc và quyết định rời đi của nhân viên.
Trị số p = 0.004526
Vì p < alpha nên bác bỏ H0
```

Kết quả kiểm định là bác bỏ H0 cho thấy có mối quan hệ giữa phòng ban làm việc và quyết định rời bỏ của nhân viên với mức ý nghĩa 0.05. Vì vậy, công ty cần tiến hành điều tra nguyên nhân và kịp thời đưa ra các biện pháp điều chỉnh để giảm thiểu số lượng nhân viên rời bỏ ở từng phòng ban.

4.2.4 Kiểm định giả thuyết về sự liên quan giữa tăng ca và sự rời đi

Mục tiêu của kiểm định nhằm xác định liệu có mối liên hệ thống kê giữa việc tăng ca và quyết định rời bỏ công việc của nhân viên. Cụ thể, chúng ta muốn biết liệu rằng việc nhân viên tăng ca có ảnh hưởng đến quyết định rời bỏ công việc của họ hay không.

- Tăng ca (OverTime): Yes, No
- Rời bỏ (Attrition): Yes, No

Để kiểm định giả thuyết "Tăng ca và Sự rời đi của nhân viên không có mối quan hệ với nhau", ta sẽ tiến hành kiểm định Chi-square với các giả thuyết:

- H0: Tăng ca và Sự rời đi của nhân viên không có mối quan hệ với nhau (độc lập nhau)
- H1: Tăng ca và Sự rời đi của nhân viên có mối quan hệ với nhau (phụ thuộc nhau)

Trước hết, ta tạo bảng thống kê số lần xuất hiện của các quan sát cho hai biến "OverTime" và "Attrition":

```
contingency_table_OT = pd.crosstab(df['OverTime'], df['Attrition'])
print(contingency_table_OT)
```

Attrition	No	Yes
OverTime		
No	944	110
Yes	289	127

Thực hiện kiểm định Chi-square với mức ý nghĩa 5%:

```
alpha = 0.05
chi2, p, _, _ = chi2_contingency(contingency_table_OT)
print('\nH0: Tăng ca và Sự rời đi của nhân viên không có mối quan hệ với nhau (độc lập nhau)')
print('H1: Tăng ca và Sự rời đi của nhân viên có mối quan hệ với nhau (phụ thuộc nhau)')
print(f'\nAlpha: {alpha} (Mức độ tin cậy 95%)')
print(f'Chi-square value: {chi2:.4f}')
print(f'p-value: {p}')
if (p < alpha):
    print(f'Trị số p = {p} < {alpha:.4f} nên bác bỏ H0 ==> (OverTime, Attrition) PHỤ THUỘC')
else:
    print(f'Trị số p = {p} >= {alpha:.4f} KHÔNG bác bỏ H0 ==> (OverTime, Attrition) ĐỘC LẬP')
```

Kết quả thu được như sau:

```
Alpha: 0.05 (Mức độ tin cậy 95%)
Chi-square value: 87.5643
p-value: 8.15842372153832e-21
Trị số p = 8.15842372153832e-21 < 0.0500 nên bác bỏ H0
```

Kết quả kiểm định trên có thể bác bỏ H0 với mức ý nghĩa 5%, tức việc tăng ca hoặc làm thêm giờ có ảnh hưởng đến quyết định rời đi của nhân sự. Kết quả kiểm định cung cấp thông tin hữu ích cho các nhà quản lý nhân sự trong việc hiểu rõ hơn về tác động của việc làm thêm giờ đến sự hài lòng và quyết định ở lại hoặc rời bỏ của nhân viên. Điều này giúp họ đưa ra các quyết định quản lý tốt hơn, nhằm cải thiện môi trường làm việc và giảm tỷ lệ rời bỏ của nhân viên trong công việc.

4.2.5 Kiểm định sự khác biệt về mức độ hài lòng với công việc giữa nhân viên ở lại và nhân viên rời đi

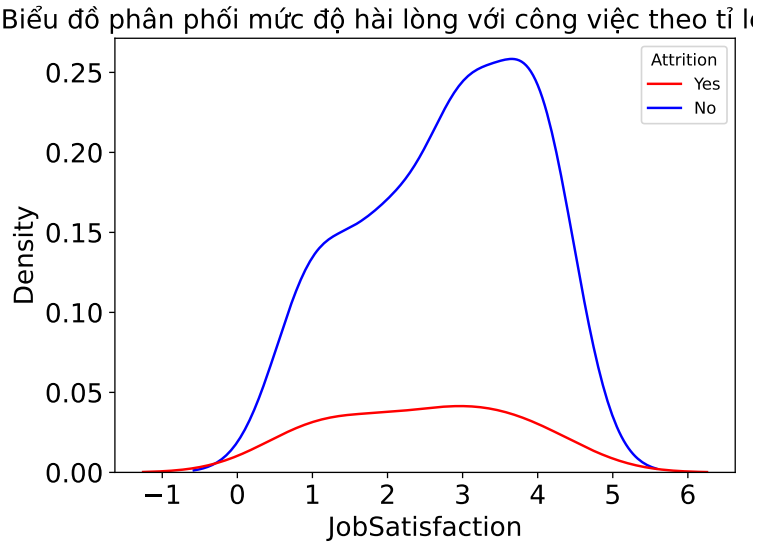
Mức độ hài lòng với công việc có thể là một yếu tố then chốt ảnh hưởng đến quyết định ở lại hay rời đi của nhân viên. Nhóm đưa ra giả định rằng những nhân viên càng hài lòng với công việc, họ sẽ càng có xu hướng gắn bó lâu dài với công ty. Trước hết, ta sẽ xem qua biểu đồ phân phối mức độ hài lòng với công việc giữa hai nhóm nhân viên ở lại và rời đi.

```
sbn.kdeplot(df, x='JobSatisfaction', hue='Attrition', palette={'Yes': 'red',
↪  'No': 'blue'},bw_adjust=2)
plt.title(f'Biểu đồ phân phối mức độ hài lòng với công việc theo tỉ lệ rời
↪  bỏ',fontsize=16)
plt.xlabel(f'JobSatisfaction',fontsize=16)
plt.ylabel('Density',fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.tight_layout()
plt.savefig(f"figs/Distribution biến JobSatisfaction theo Attrition.pdf")
plt.show()
```

Biểu đồ cho thấy có sự khác biệt về mức độ hài lòng giữa hai nhóm nhân viên ở lại và rời đi. Cụ thể, nhóm ở lại có phân phối mức hài lòng lệch về phía giá trị cao, trong khi phân phối của nhóm rời đi lại có đỉnh phẳng và đuôi mỏng hơn tức mức độ hài lòng phân tán hơn. Điều này cho thấy nhân viên ở lại có mức độ hài lòng với công việc cao hơn nhân viên rời đi. Tuy nhiên, để khẳng định kết luận đó, chúng ta cần phải tiến hành các kiểm định thống kê.

Bước kiểm định sẽ giúp xác định có thực sự tồn tại sự chênh lệch về mức độ hài lòng giữa hai nhóm hay không. Qua đó, chúng ta mới đưa ra các chiến lược phù hợp để nâng cao sự gắn kết của nhân viên với công ty.

Để kiểm định giả thuyết về chênh lệch giữa trung bình mức độ hài lòng với công việc của 2 nhóm nhân viên khi chưa biết độ lệch chuẩn tổng thể (σ_1 và σ_2), thì ta sử dụng s_1



Hình 35: Biểu đồ phân phối biến JobSatisfation theo Attrition

là ước lượng của σ_1 và s_2 là ước lượng của σ_2, thì thống kê t sẽ có dạng như sau:

$$t = \frac{(\bar{x} - \bar{y}) - D_0}{\sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}}$$

Với:

$$df = \frac{\left(\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}\right)^2}{\frac{1}{(n_X - 1)}\left(\frac{s_X^2}{n_X}\right)^2 + \frac{1}{(n_Y - 1)}\left(\frac{s_Y^2}{n_Y}\right)^2}$$

Tuy nhiên, kích thước mẫu giữa hai nhóm nhân viên có sự chênh lệch rất lớn. Cụ thể, nhóm nhân viên ở lại chiếm tới 84% tổng số quan sát, trong khi nhóm nhân viên rời đi chỉ có 16%, làm tăng xác suất xảy ra hiện tượng phương sai khác nhau giữa các nhóm. Nhóm có kích thước mẫu nhỏ hơn sẽ có xu hướng cho phương sai lớn hơn. Bên cạnh đó, các nhóm cũng không có phân phối chuẩn. Nhưng phép kiểm định T lại đòi hỏi phải thỏa các giả định về phân phối chuẩn và các phương sai của các mẫu phải gần bằng nhau.

Kiểm định Mann–Whitney U là một phương pháp phi tham số, ít giả định hơn về mặt lý thuyết so với kiểm định T. Do đó, nó phù hợp và đáng tin cậy hơn cho tập dữ liệu với cỡ mẫu khác nhau và không có phân phối chuẩn như trong trường hợp này.

Kiểm định Mann–Whitney U được dùng để kiểm định sự chênh lệch của 2 phân phối, cụ thể hơn là so sánh thứ hạng trung bình (mean ranks) của 2 nhóm. Các bước tính toán để tìm được p-value cho kiểm định Mann–Whitney U như sau:

```
import numpy as np
from scipy.stats import norm
U = min(U1, U2)
N = nx + ny
z = (U - nx*ny/2 + 0.5) / np.sqrt(nx*ny * (N + 1)/ 12)
p = 2 * norm.cdf(z)  # use CDF to get p-value from smaller statistic
```

Trong đó, $U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$ và $U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2$

Một số giả định cần kiểm tra trước khi tiến hành kiểm định Mann–Whitney U:

- Các quan sát độc lập: các quyết định rời đi hay ở lại của nhân viên là độc lập nhau.
- Biến phụ thuộc là biến định lượng liên tục hoặc biến định lượng có thứ bậc: biến JobSatisfaction là biến định lượng với thang đo interval (có thứ bậc) nên thỏa giả định này.
- Biến độc lập phải là biến định danh gồm 2 nhóm độc lập: biến Attrition là biến định danh gồm 2 nhóm là Yes và No.

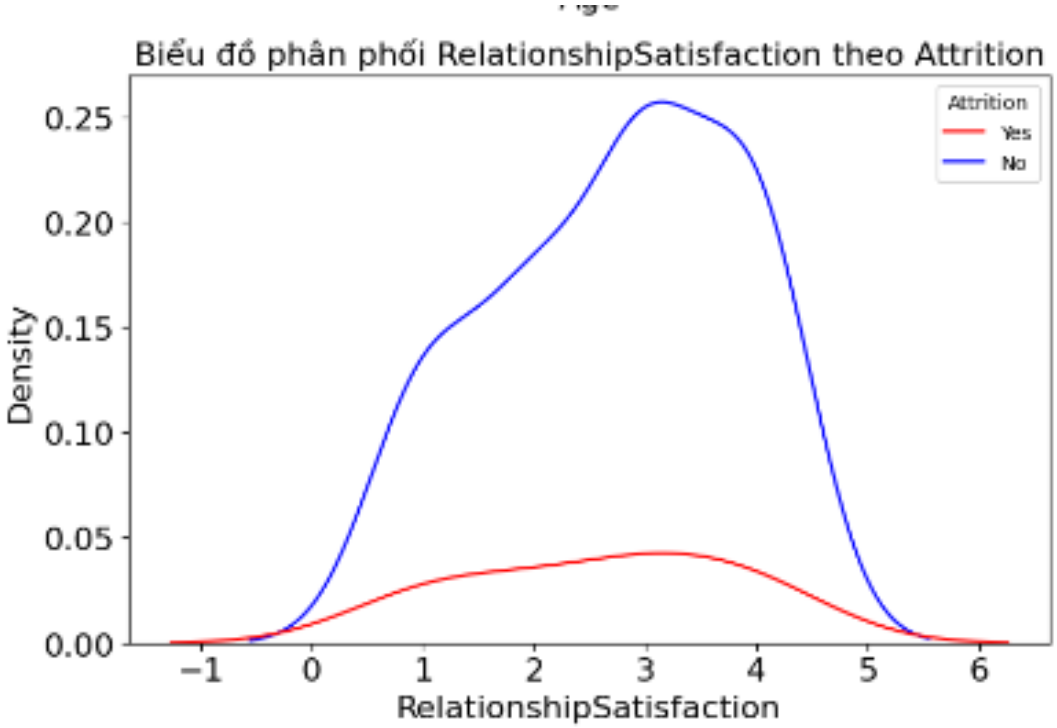
Khi đã kiểm tra các giả định, ta sử dụng hàm `scipy.stats.mannwhitneyu` của Python để kiểm định Mann–Whitney U với giả thuyết H0 là "Mức độ hài lòng với công việc của nhóm nhân viên rời đi lớn hơn hoặc bằng mức độ hài lòng với công việc của nhóm nhân viên ở lại", giả thuyết đối là "Mức độ hài lòng với công việc của nhóm nhân viên rời đi nhỏ hơn mức độ hài lòng với công việc của nhóm nhân viên ở lại". Đây là kiểm định 1 phía và kiểm định mức độ hài lòng nhóm nhân viên rời đi có nhỏ hơn hay không nên tham số alternative = "less":

```
## Kiểm định Mann-Whitney U
group1 = df[df['Attrition'] == 'Yes']['JobSatisfaction']
group2 = df[df['Attrition'] == 'No']['JobSatisfaction']
print('''H0: Mức độ hài lòng với công việc của nhóm nhân viên rời đi
lớn hơn hoặc bằng mức độ hài lòng với công việc của nhóm ở lại''')
print('''H1: Mức độ hài lòng với công việc của nhóm rời đi
nhỏ hơn mức độ hài lòng với công việc của nhóm ở lại''')
statistic, p = stats.mannwhitneyu(group1, group2,alternative='less')
print(f'Trị số p = {p:4f}')
if p >= 0.05:
  print("Vì p >= alpha nên không bác bỏ H0")
else:
  print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Mức độ hài lòng với công việc của nhóm nhân viên rời đi
lớn hơn hoặc bằng mức độ hài lòng với công việc của nhóm ở lại
H1: Mức độ hài lòng với công việc của nhóm rời đi
nhỏ hơn mức độ hài lòng với công việc của nhóm ở lại
Trị số p = 0.000044
Vì p < alpha nên bác bỏ H0
```

Kết quả kiểm định là bác bỏ H0 cho thấy mức độ hài lòng với công việc của nhóm nhân viên rời đi thấp hơn nhóm còn lại với mức ý nghĩa 0.05. Vì vậy, công ty cần tiến hành điều tra nguyên nhân vì sao mức độ hài lòng với công việc của nhóm nhân viên rời đi thấp hơn, có thể tiến hành các cuộc khảo sát hoặc phỏng vấn nhân viên để hiểu rõ hơn về vấn đề này. Ngay khi có dấu hiệu nhân viên không hài lòng với công việc, công ty cần đưa ra các biện pháp điều chỉnh để tránh tình trạng nhân viên rời bỏ.

4.2.6 Kiểm định sự khác biệt về độ hài lòng về mối quan hệ giữa nhân viên ở lại so với nhân viên rời đi



Hình 36: Biểu đồ phân phối mức độ hài lòng về mối quan hệ theo Attrition

```
#Phân nhóm
df_yes = df[df['Attrition'] == 'Yes']
df_no = df[df['Attrition'] == 'No']

#Chọn mức ý nghĩa
alpha           = .05
```

```
7
8  ## Kiểm định Mann-Whitney U
9  group1 = df[df['Attrition'] == 'Yes']['RelationshipSatisfaction']
10 group2 = df[df['Attrition'] == 'No']['RelationshipSatisfaction']
11 statistic, p = stats.mannwhitneyu(group1, group2,alternative='two-sided')
12 print(f'Trị số p = {p:.4f}')
13 if (p < alpha):
14     print(f'Trị số p = {p:.4f} < {alpha:.4f} nên bác bỏ H0 ==> Muy[1] <>
       ↪ Muy[2]')
15 else:
16     print(f'Trị số p = {p:.4f} >= {alpha:.4f} KHÔNG bác bỏ H0 ==> Muy[1] =
       ↪ Muy[2]')
```

```
Trị số p = 0.1020
Trị số p = 0.1020 >= 0.0500 KHÔNG bác bỏ H0 ==> Muy[1] = Muy[2]
```

Để kiểm tra xem thang đo mức độ hài lòng đối với mối quan hệ của nhân viên có ý nghĩa giải thích quyết định rời đi của họ không, tiến hành kiểm định khác biệt của biến RelationshipSatisfaction giữa 2 nhóm nhân viên.

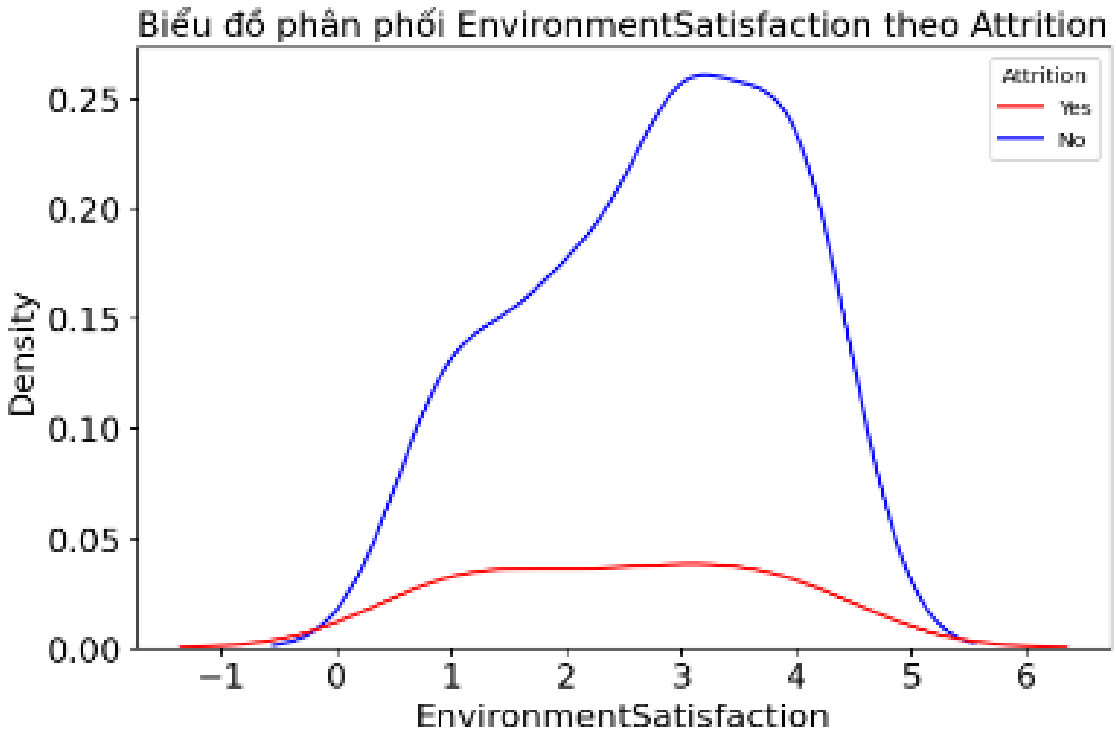
Tương tự với kiểm định trên, vì biến không có phân phối chuẩn, hình dạng phân phối 2 mẫu là khác nhau, hơn nữa RelationshipSatisfaction là biến dạng thang đo thứ bậc, kiểm định Mann_Whitney U sẽ phù hợp với những đặc điểm này.

Giả thuyết kiểm định là:

- H_0: Độ hài lòng về mối quan hệ của nhân viên rời đi bằng độ hài lòng về mối quan hệ của nhân viên ở lại.
- H_a: Độ hài lòng về mối quan hệ của nhân viên rời đi khác độ hài lòng về mối quan hệ của nhân viên ở lại.

Kết luận: Vì không thể bác bỏ giả thuyết H_0 nên không thể khẳng định rằng có sự khác nhau về mức độ hài lòng về mối quan hệ của nhân viên rời đi so với nhân viên ở lại. Vậy biến RelationshipSatisfaction không thật sự có tác động riêng lẻ đến khả năng rời đi hay ở lại của một nhân viên. Tuy nhiên, không nên lập tức bỏ qua yếu tố này, ta có thể xem xét đến trường hợp tác động đồng thời của các biến khác khi kết hợp với RelationshipSatisfaction để đảm bảo không bỏ sót tác động của sự kết hợp nhiều yếu tố.

4.2.7 Kiểm định sự khác biệt về độ hài lòng về môi trường làm việc giữa nhân viên ở lại và nhân viên rời đi



Hình 37: Biểu đồ phân phối biến EnvironmentSatisfaction theo Attrition

```
#Phân nhóm
df_yes = df[df['Attrition'] == 'Yes']
df_no = df[df['Attrition'] == 'No']

#Chọn mức ý nghĩa
alpha             = .05

## Kiểm định Mann-Whitney U
group1 = df[df['Attrition'] == 'Yes']['EnvironmentSatisfaction']
group2 = df[df['Attrition'] == 'No']['EnvironmentSatisfaction']

statistic, p = stats.mannwhitneyu(group1, group2,alternative='less')
print(f'Trị số p = {p:.4f}')
if (p < alpha):
    print(f'Trị số p = {p:.4f} < {alpha:.4f} nên bác bỏ H0 ==> Muy[1] <
    ↪ Muy[2]')
else:
    print(f'Trị số p = {p:.4f} >= {alpha:.4f} KHÔNG bác bỏ H0 ==> Muy[1] >=
    ↪ Muy[2]')
```

```
Trị số p = 0.0001
Trị số p = 0.0001 < 0.0500 nên bác bỏ H0 ==> Muy[1] < Muy[2]
```

Môi trường làm việc thường được xem là yếu tố quan trọng đối với người lao động và thường có tác động đến sự gắn bó lâu dài của họ với nơi làm việc. Vì vậy, có thể giả định rằng mức độ hài lòng đối với môi trường làm việc của nhân viên rời đi sẽ thấp hơn nhân viên ở lại.

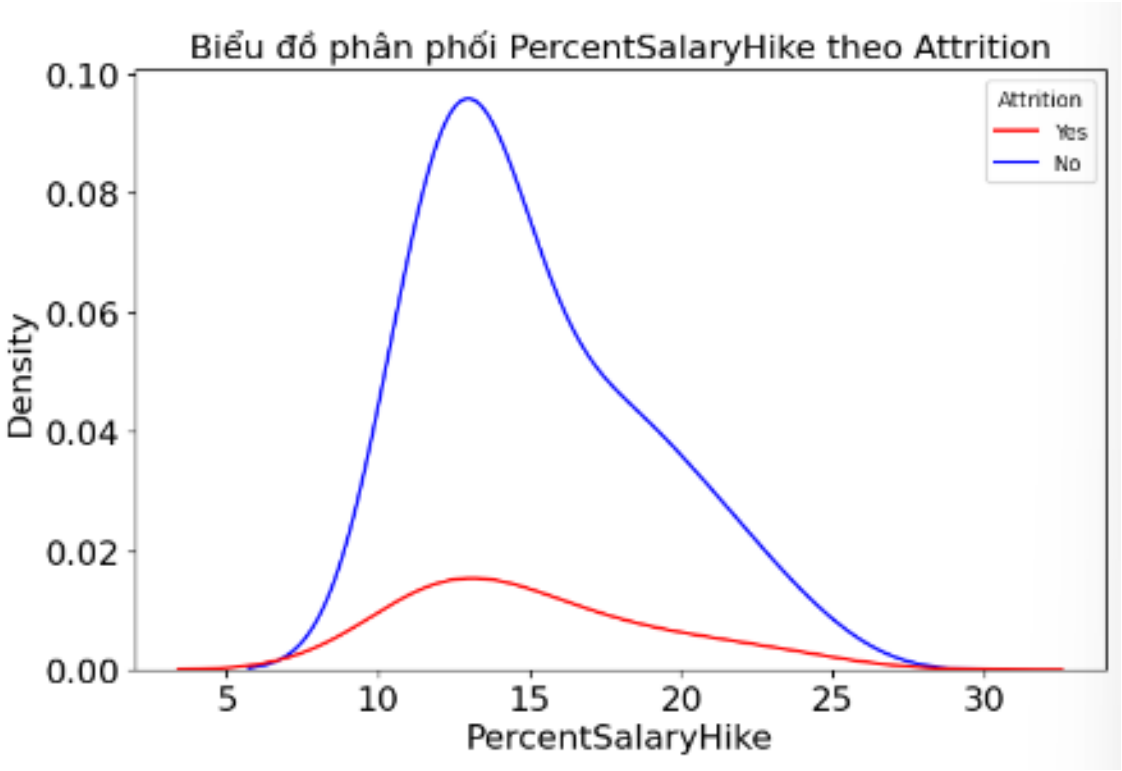
Với đặc điểm không có phân phối chuẩn, hình dạng phân phối 2 mẫu khác nhau và là thang đo thứ bậc, chọn kiểm định Mann-Whitney U.

Giả thuyết kiểm định là:

- H_0: Độ hài lòng về môi trường làm việc của nhân viên rời đi cao hơn hoặc bằng độ hài lòng về môi trường làm việc của nhân viên ở lại.
- H_a: Độ hài lòng về môi trường làm việc của nhân viên rời đi thấp hơn độ hài lòng về môi trường làm việc của nhân viên ở lại.

Kết luận: Với độ tin cậy 95%, có thể bác bỏ giả thuyết không và cho rằng nhân viên rời đi có độ hài lòng về môi trường làm việc thấp hơn so với nhân viên ở lại. Điều này cho thấy các yếu tố về môi trường làm việc, cơ sở vật chất nên được công ty quan tâm, đầu tư nếu không muốn đánh mất nhân sự tiềm năng.

4.2.8 Kiểm định sự khác biệt về mức tăng lương giữa nhân viên ở lại và nhân viên rời đi



Hình 38: Biểu đồ phân phối biến PecentSalaryHike theo Attrition

```
#Phân nhóm
df_yes = df[df['Attrition'] == 'Yes']
df_no = df[df['Attrition'] == 'No']

#Chọn mức ý nghĩa
alpha             = .05

## Kiểm định Mann-Whitney U
group1 = df[df['Attrition'] == 'Yes']['PercentSalaryHike']
group2 = df[df['Attrition'] == 'No']['PercentSalaryHike']
statistic, p = stats.mannwhitneyu(group1, group2,alternative='less')
print(f'Trị số p = {p:.4f}')
if (p < alpha):
    print(f'Trị số p = {p:.4f} < {alpha:.4f} nên bác bỏ H0 ==> Muy[1] <
    ↪ Muy[2]')
else:
    print(f'Trị số p = {p:.4f} >= {alpha:.4f} KHÔNG bác bỏ H0 ==> Muy[1] >=
    ↪ Muy[2]')
```

```
Trị số p = 0.1784
Trị số p = 0.1784 >= 0.0500 KHÔNG bác bỏ H0 ==> Muy[1] >= Muy[2]
```

Mức tăng lương cũng là một yếu tố được nghi ngờ rằng sẽ ảnh hưởng đến sự gắn bó của nhân viên với công ty. Kiểm định sự khác biệt về mức tăng lương giữa nhân viên ở lại và nhân viên rời đi nhằm đánh giá xem liệu phần trăm tăng lương có ý nghĩa giải thích cho quyết định rời đi hay ở lại của nhân viên hay không.

Vì phân phối của biến PercentSalaryHike theo Attrition của cả 2 nhóm đều không tuân theo phân phối chuẩn, nên sử dụng kiểm định Mann-Whitney U. Và vì hình dạng phân phối của cả 2 cũng không giống nhau, nên lưu ý rằng kết quả kiểm định là so sánh thứ hạng trung bình (mean ranks) của 2 nhóm.

Giả thuyết kiểm định là:

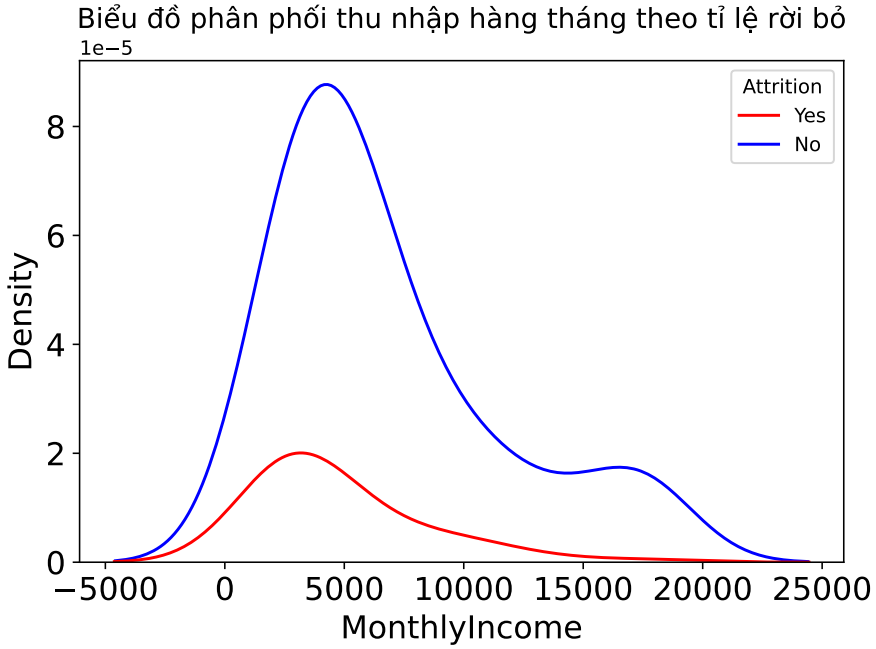
- H_0: Mức tăng lương của nhân viên rời đi lớn hơn hoặc bằng mức tăng lương của nhân viên ở lại.
- H_a: Mức tăng lương của nhân viên rời đi bé hơn mức tăng lương của nhân viên ở lại.

Kết luận: Không có đủ căn cứ để bác bỏ giả thuyết không. Vậy có khả năng mức tăng lương của nhân viên rời đi không khác biệt so với mức tăng lương của nhân viên ở lại. Do đó không thể kết luận rằng mức tăng lương có tác động đến khả năng rời đi của nhân viên.

4.2.9 Kiểm định sự khác biệt về thu nhập hàng tháng giữa nhân viên ở lại và nhân viên rời đi

Trước hết, ta cần quan sát phân phối thu nhập hàng tháng của nhân viên theo tỉ lệ rời bỏ:

```
sbn.kdeplot(df, x='MonthlyIncome', hue='Attrition', palette={'Yes': 'red',
↪  'No': 'blue'},bw_adjust=2)
plt.title(f'Biểu đồ phân phối thu nhập hàng tháng theo tỉ lệ rời
↪  bỏ',fontsize=14)
plt.xlabel(f'MonthlyIncome',fontsize=16)
plt.ylabel('Density',fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.tight_layout()
plt.savefig(f"figs/Distribution biến MonthlyIncome theo Attrition.pdf")
plt.show()
```



Hình 39: Phân phối thu nhập hàng tháng theo biến Attrition

Đường cong màu đỏ, tương ứng với nhóm nhân viên đã rời bỏ (Attrition = Yes), có đỉnh thấp hơn và phân bố rộng hơn so với đường cong màu xanh. Điều này cho thấy nhân viên rời bỏ có phạm vi thu nhập hàng tháng rộng lớn hơn nhưng tổng số lượng ít hơn so với những người không rời bỏ. Đỉnh của đường cong màu đỏ nằm ở một điểm thấp hơn trên trục thu nhập hàng tháng so với đường cong màu xanh. Từ đó, nhóm đã đặt ra giả thuyết rằng: nhân viên có thu nhập thấp hơn có thể có nhiều khả năng rời bỏ hơn.

Xem xét biểu đồ phân phối trên, nhóm nhận thấy phân phối thu nhập hàng tháng của 2 nhóm nhân viên không phân theo phân phối chuẩn, đồng thời các quan sát là độc lập với nhau. Bên cạnh đó biến "MonthlyIncome" là biến liên tục và "Attrition" là biến phân loại gồm 2 nhóm (Yes - No). Do đó, trong trường hợp này sẽ áp dụng phương pháp Mann-Whitney U nhằm kiểm định mối quan hệ của 2 biến trên.

Các giả thuyết được đặt ra như sau:

- H0: Thu nhập hàng tháng của nhóm nhân viên rời đi lớn hơn hoặc bằng tiền lương hàng tháng của nhóm nhân viên ở lại.
- H1: Thu nhập hàng tháng của nhóm nhân viên rời đi nhỏ hơn tiền lương hàng tháng của nhóm nhân viên ở lại.

```
income1 = df[df['Attrition'] == 'Yes']['MonthlyIncome']
income2 = df[df['Attrition'] == 'No']['MonthlyIncome']
print('H0: Thu nhập hàng tháng của nhóm nhân viên rời đi lớn hơn hoặc bằng thu
↪  nhập hàng tháng của nhóm nhân viên ở lại')
print('H1: Thu nhập hàng tháng của nhóm nhân viên rời đi nhỏ hơn thu nhập hàng
↪  tháng của nhóm nhân viên ở lại')
```

Ta sử dụng hàm scipy.stats.mannwhitneyu của Python để kiểm định Mann–Whitney U:

```
statistic, p = stats.mannwhitneyu(income1, income2,alternative='less')
alpha = 0.05
print(f'Alpha = {alpha} (Độ tin cậy 95%)')
print(f'Trị số p = {p}')
if p >= alpha:
  print("Vì p >= alpha nên không bác bỏ H0")
else:
  print("Vì p < alpha nên bác bỏ H0")
```

Sau khi kiểm định ta thu được kết quả dưới đây:

```
Alpha = 0.05 (Độ tin cậy 95%)
Trị số p = 1.2985087701945417e-14
Vì p < alpha nên bác bỏ H0
```

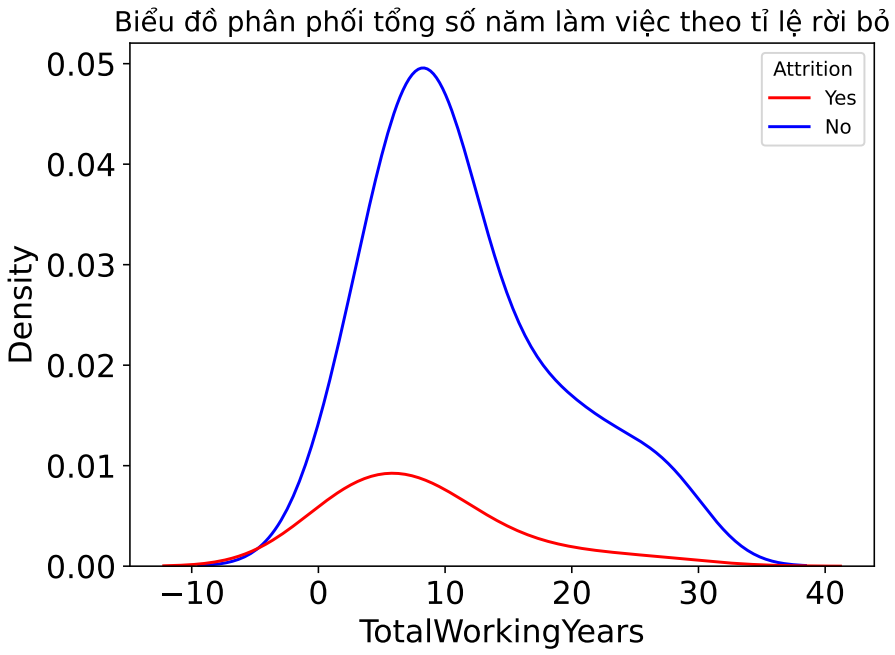
Giá trị p (p-value) là 1.2985087701945417e-14, một giá trị cực kỳ nhỏ, gần như bằng không. Vì giá trị p rất nhỏ và thấp hơn nhiều so với mức alpha đã chọn, ta có đủ cơ sở để bác bỏ giả thuyết không (H0), tức tiền lương hàng tháng của nhóm nhân viên rời đi nhỏ hơn so với tiền lương hàng tháng của nhóm nhân viên ở lại.

Từ đó rút ra kết luận rằng mức thu nhập hàng tháng có thể là một yếu tố ảnh hưởng đến quyết định của nhân viên khi cân nhắc việc rời bỏ công ty. Các nhà quản trị có thể xem xét thông tin này khi xây dựng chiến lược giữ chân nhân viên hoặc cải thiện cấu trúc lương bổng.

4.2.10 Kiểm định sự khác biệt về tổng số năm làm việc giữa nhân viên ở lại và nhân viên rời đi

Trước hết, ta cần quan sát phân phối tổng số năm làm việc của nhân viên theo tỉ lệ rời bỏ:

```
sbn.kdeplot(df, x='TotalWorkingYears', hue='Attrition', palette={'Yes': 'red', 'No': 'blue'},bw_adjust=2)
plt.title(f'Biểu đồ phân phối tổng số năm làm việc theo tỉ lệ rời bỏ',fontsize=14)
plt.xlabel(f'TotalWorkingYears',fontsize=16)
plt.ylabel('Density',fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.tight_layout()
plt.savefig(f"figs/Distribution biến TotalWorkingYears theo Attrition.pdf")
plt.show()
```



Hình 40: Phân phối tổng số năm làm việc theo biến Attrition

Đường cong màu đỏ, đại diện cho nhóm nhân viên đã rời bỏ (Attrition = Yes), có phạm vi rộng hơn nhưng đỉnh thấp hơn so với đường cong màu xanh, tương ứng với nhóm nhân viên không rời bỏ (Attrition = No). Điều này cho thấy nhóm nhân viên rời bỏ có tổng số năm làm việc phân bố rộng lớn hơn và mật độ thấp hơn. Đỉnh của đường cong màu xanh nằm ở phần giữa của trục tổng số năm làm việc, cho thấy phần lớn nhân viên không rời

bỏ có số năm làm việc tập trung ở một khoảng nhất định. Trong khi đó, đỉnh của đường cong màu đỏ nằm ở một vị trí thấp hơn, có thể cho thấy những nhân viên rời bỏ có tổng số năm làm việc ít hơn.

Có sự chồng chéo giữa hai đường cong nhưng không đáng kể, vì vậy nhóm đã đặt ra giả thuyết rằng: nhân viên có tổng số năm làm việc ít hơn có thể có nhiều khả năng rời bỏ hơn.

Xem xét biểu đồ phân phối trên, nhóm nhận thấy phân phối thu nhập hàng tháng của 2 nhóm nhân viên không phân theo phân phối chuẩn, đồng thời các quan sát là độc lập với nhau. Bên cạnh đó biến "TotalWorkingYears" là biến liên tục và "Attrition" là biến phân loại gồm 2 nhóm (Yes - No). Do đó, trong trường hợp này sẽ áp dụng phương pháp Mann-Whitney U nhằm kiểm định mối quan hệ của 2 biến trên.

Các giả thuyết được đặt ra như sau:

- H0: Tổng số năm làm việc của nhóm nhân viên rời đi lớn hơn hoặc bằng tổng số năm làm việc của nhóm nhân viên ở lại.
- H1: Tổng số năm làm việc của nhóm nhân viên rời đi nhỏ hơn tổng số năm làm việc của nhóm nhân viên ở lại.

```
total1 = df[df['Attrition'] == 'Yes']['TotalWorkingYears']
total2 = df[df['Attrition'] == 'No']['TotalWorkingYears']
print('H0: Tổng số năm làm việc của nhóm nhân viên rời đi lớn hơn hoặc tổng số
 ↪  năm làm việc của nhóm nhân viên ở lại')
print('H1: Tổng số năm làm việc của nhóm nhân viên rời đi nhỏ hơn tổng số năm
 ↪  làm việc của nhóm nhân viên ở lại')
```

Ta sử dụng hàm scipy.stats.mannwhitneyu của Python để kiểm định Mann–Whitney U:

```
statistic, p = stats.mannwhitneyu(total1, total2,alternative='less')
alpha = 0.05
print(f'\nAlpha = {alpha} (Độ tin cậy 95%)')
print(f'Trị số p = {p}')
if p >= alpha:
  print("Vì p >= alpha nên không bác bỏ H0")
else:
  print("Vì p < alpha nên bác bỏ H0")
```

Sau khi kiểm định ta thu được kết quả dưới đây:

```
Alpha = 0.05 (Độ tin cậy 95%)
Trị số p = 1.057614921355609e-14
Vì p < alpha nên bác bỏ H0
```

Giá trị p (p-value) là 1.057614921355609e-14, một giá trị cực kỳ nhỏ, gần như bằng không. Vì giá trị p nhỏ hơn nhiều so với mức alpha đã chọn (0.05), ta có đủ bằng chứng thống kê để bác bỏ giả thuyết không (H0), tức tổng số năm làm việc của nhóm nhân viên rời đi thực sự nhỏ hơn tổng số năm làm việc của nhóm nhân viên ở lại.

Kết quả này có thể chỉ ra rằng kinh nghiệm làm việc (tính theo số năm làm việc) có thể ảnh hưởng đến quyết định rời bỏ công ty của nhân viên. Các nhà quản lý và nhà hoạch định chính sách có thể cân nhắc việc này khi thiết kế chương trình phát triển sự nghiệp và cơ hội thăng tiến để giữ chân nhân viên có kinh nghiệm.

4.2.11 Kiểm định sự khác biệt về thu nhập giữa các phòng ban

Kiểm định sự khác biệt về mức thu nhập giữa các phòng ban là bước quan trọng để đảm bảo sự công bằng và xây dựng một môi trường làm việc tích cực. Từ đó, công ty có thể giữ vững sự hài lòng và cam kết của nhân viên.

Để kiểm định giả thuyết "Không có sự khác biệt về thu nhập hàng tháng giữa các phòng ban", ta sẽ tiến hành kiểm định ANOVA, bởi vì đây là mối quan hệ giữa một biến phụ thuộc (biến MonthlyIncome) và một biến độc lập (biến Department phân hoạch thành 3 nhóm: Sales, Research & Development, Human Resources). Với giả thuyết H0 là "Không có sự khác biệt về thu nhập giữa các phòng ban" hay "Thu nhập trung bình của 3 phòng ban bằng nhau", giả thuyết đối là "Có sự khác biệt về thu nhập giữa các phòng ban" hay "Thu nhập trung bình của 3 phòng ban khác nhau".

Trước khi tiến hành kiểm định ANOVA để so sánh sự khác biệt giữa các nhóm, cần kiểm tra các giả định:

- Các quan sát độc lập: Xét về mặt logic, các nhân viên khác phòng ban là đối tượng độc lập với nhau. Do đó, có thể giả định các quan sát độc lập.
- Các nhóm có phương sai giống nhau: Giả định phương sai của các nhóm là như nhau kiểm tra bằng kiểm định Levene.

```
import scipy.stats as stats


group1 = df[df['Department'] == 'Sales']['MonthlyIncome']
group2 = df[df['Department'] == 'Research & Development']['MonthlyIncome']
group3 = df[df['Department'] == 'Human Resources']['MonthlyIncome']


levene, p = stats.levene(group1, group2, group3)
print('H0: Các mẫu dữ liệu có phương sai bằng nhau')
print('H1: Các mẫu dữ liệu có phương sai khác nhau')
print(f'Trị thống kê Levene = {levene:4f}; p = {p:4f}')
if p > 0.05:
    print("Vì p > alpha nên không bác bỏ H0")
else:
    print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Các mẫu dữ liệu có phương sai bằng nhau
H1: Các mẫu dữ liệu có phương sai khác nhau
Trị thống kê Levene = 3.398108; p = 0.033700
Vì p < alpha nên bác bỏ H0
```

Như vậy, qua kiểm định Levene, ta có thể thấy phương sai các nhóm khác nhau với mức ý nghĩa 0.05.

- Các nhóm có phân phối chuẩn: Dữ liệu của các nhóm cần tuân theo phân phối chuẩn kiểm tra bằng kiểm định Shapiro.

```
from statsmodels.formula.api     import ols
## Kiểm định Shapiro dựa trên Ordinary Least Squares (OLS) model
print('H0: Dữ liệu các nhóm có phân phối chuẩn')
print('H1: Dữ liệu các nhóm không có phân phối chuẩn')
model           = ols('MonthlyIncome ~ C(Department)', data =
↪  df[['Department','MonthlyIncome']]).fit()
shapiro, p      = stats.shapiro(model.resid)
print(f'Trị thống kê Shapiro = {shapiro:4f}; p = {p}')
if p > 0.05:
    print("Vì p > alpha nên không bác bỏ H0")
else:
```

```
    print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Dữ liệu các nhóm có phân phối chuẩn
H1: Dữ liệu các nhóm không có phân phối chuẩn
Trị thống kê Shapiro = 0.821791; p = 1.4318055166203084e-37
Vì p < alpha nên bác bỏ H0
```

Như vậy, qua kiểm định Shapiro, ta có thể thấy dữ liệu các nhóm không có phân phối chuẩn với mức ý nghĩa 0.05.

Kết quả kiểm định cho thấy phương sai của các nhóm khác nhau và phân phối của dữ liệu không chuẩn, thì không thỏa mãn 2 giả định quan trọng khi áp dụng ANOVA.

Trong trường hợp này, ta sẽ sử dụng kiểm định ANOVA không đòi hỏi phương sai đồng nhất và phân phối chuẩn như Welch's ANOVA test:

```
print('H0: Mức thu nhập trung bình của 3 phòng ban bằng nhau')
print('H1: Có sự khác biệt về mức thu nhập trung bình giữa các phòng ban')
# Kiểm định Welch's ANOVA
stat, p = stats.f_oneway(group1, group2, group3)
print(f'Trị số p = {p:4f}')
if p > 0.05:
  print("Vì p > alpha nên không bác bỏ H0")
else:
  print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Mức thu nhập trung bình của 3 phòng ban bằng nhau
H1: Có sự khác biệt về mức thu nhập trung bình giữa các phòng ban
Trị số p = 0.018127
Vì p < alpha nên bác bỏ H0
```

Kết quả kiểm định Welch's ANOVA là bác bỏ H0 cho thấy có sự khác biệt về mức thu nhập trung bình giữa các phòng ban với mức ý nghĩa 0.05. Vì vậy, để biết được sự khác biệt đáng kể nằm ở những cặp phòng ban nào, ta tiến hành hậu kiểm Tukey HSD.

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd


tukey = pairwise_tukeyhsd(endog=df['MonthlyIncome'],
                          groups=df['Department'],
                          alpha=0.05)


print(tukey)
```

```
                Multiple Comparison of Means - Tukey HSD, FWER=0.05
=============================================================================================
        group1                group2          meandiff p-adj    lower      upper   reject
---------------------------------------------------------------------------------------------
       Human Resources Research & Development -289.1974 0.8735 -1658.8215 1080.4267  False
       Human Resources                 Sales  439.7275  0.747  -977.7121 1857.1671  False
Research & Development                 Sales  728.9249 0.0129   125.5305 1332.3194   True
---------------------------------------------------------------------------------------------
```

Giá trị p-adj so sánh giữa Research & Development và Sales là 0.0129, nhỏ hơn ngưỡng ý nghĩa thống kê 0.05, và cột reject là True. Điều này cho thấy có đủ bằng chứng để bác bỏ giả thuyết H_0, có thể kết luận rằng có sự khác biệt về thu nhập giữa phòng Research & Development và phòng Sales. Cột meandiff cho biết hiệu giữa các trung bình của các nhóm so sánh. Giữa Research & Development và Sales, hiệu mức thu nhập trung bình là 728.9249 cho thấy phòng ban Research & Development có mức thu nhập cao hơn so với phòng ban Sales.

Ngược lại, giữa Human Resources và Research & Development, cũng như giữa Human Resources và Sales, không có đủ bằng chứng để bác bỏ giả thuyết H0, và do đó, không có sự khác biệt về thu nhập giữa các nhóm này.

Sự khác biệt về mức thu nhập giữa phòng Research & Development và phòng Sales có thể phản ánh sự chênh lệch về kỹ năng, trách nhiệm, hoặc sự đóng góp cho công ty giữa hai phòng ban này. Điều này có thể tạo ra mâu thuẩn tiềm ẩn và thậm chí có thể ảnh hưởng đến việc giữ chân nhân viên. Công ty nên xem xét lại chiến lược quản lý lương để đảm bảo sự công bằng và đồng đều giữa các bộ phận. Nếu có sự khác biệt về mức thu nhập do chênh lệch trình độ, kinh nghiệm, hoặc vị trí công việc, công ty có thể cần xem xét chiến lược quản lý nhân sự và phát triển nghề nghiệp để đảm bảo sự công bằng và cơ hội phát triển cho mọi nhân viên.

4.2.12 Kiểm định sự khác biệt về mức tăng lương giữa các phòng ban

Tương tự với thu nhập, mức tăng lương cũng là một yếu tố quan trọng trong quản lý nhân sự, ảnh hưởng đến động lực, cam kết, và hiệu suất của nhân viên. Việc kiểm định sự khác biệt về mức tăng lương giữa các phòng ban là để đảm bảo rằng mọi nhân viên ở mọi phòng ban đều có cơ hội công bằng để nhận được mức tăng lương xứng đáng với đóng góp và hiệu suất cá nhân. Nếu có sự chênh lệch đáng kể trong mức tăng lương giữa các phòng ban, điều này có thể tạo ra sự bất bình đẳng trong đội ngũ nhân viên.

Để kiểm định giả thuyết "Không có sự khác biệt về mức tăng lương giữa các phòng ban", ta sẽ tiến hành kiểm định ANOVA, bởi vì đây là mối quan hệ giữa một biến phụ thuộc (biến PercentHikeSalary) và một biến độc lập (biến Department phân hoạch thành 3 nhóm: Sales, Research & Development, Human Resources). Với giả thuyết H_0 là "Không có sự khác biệt về mức tăng lương giữa các phòng ban" hay "Mức tăng lương trung bình của 3 phòng ban bằng nhau", giả thuyết đối là "Có sự khác biệt về mức tăng lương giữa các phòng ban" hay "Mức tăng lương trung bình của 3 phòng ban khác nhau".

Trước khi tiến hành kiểm định ANOVA để so sánh sự khác biệt giữa các nhóm, cần kiểm tra các giả định:

- Các quan sát độc lập: Xét về mặt logic, các nhân viên khác phòng ban là đối tượng độc lập với nhau. Do đó, có thể giả định các quan sát độc lập.
- Các nhóm có phương sai giống nhau: Giả định phương sai của các nhóm là như nhau kiểm tra bằng kiểm định Levene.

```
group1 = df[df['Department'] == 'Sales']['PercentSalaryHike']
group2 = df[df['Department'] == 'Research &
↪  Development']['PercentSalaryHike']
group3 = df[df['Department'] == 'Human Resources']['PercentSalaryHike']


levene, p = stats.levene(group1, group2, group3)
print('H0: Các mẫu dữ liệu có phương sai bằng nhau')
print('H1: Các mẫu dữ liệu có phương sai khác nhau')
print(f'Trị thống kê Levene = {levene:4f}; p = {p:4f}')
if p > 0.05:
    print("Vì p > alpha nên không bác bỏ H0")
else:
    print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Các mẫu dữ liệu có phương sai bằng nhau
H1: Các mẫu dữ liệu có phương sai khác nhau
Trị thống kê Levene = 0.337159; p = 0.713850
Vì p > alpha nên không bác bỏ H0
```

Như vậy, qua kiểm định Levene, ta có thể thấy phương sai các nhóm bằng nhau với mức ý nghĩa 0.05.

- Các nhóm có phân phối chuẩn: Dữ liệu của các nhóm cần tuân theo phân phối chuẩn kiểm tra bằng kiểm định Shapiro.

```
## Kiểm định phân phối chuẩn Shapiro dựa trên Ordinary Least Squares (OLS) model
print('H0: Dữ liệu các nhóm có phân phối chuẩn')
print('H1: Dữ liệu các nhóm không có phân phối chuẩn')
model           = ols('PercentSalaryHike ~ C(Department)', data = df[['Department','PercentSalaryHike']]).fit()
shapiro, p      = stats.shapiro(model.resid)
print(f'Trị thống kê Shapiro = {shapiro:4f}; p = {p}')
if p > 0.05:
    print("Vì p > alpha nên không bác bỏ H0")
else:
    print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Dữ liệu các nhóm có phân phối chuẩn
H1: Dữ liệu các nhóm không có phân phối chuẩn
Trị thống kê Shapiro = 0.905488; p = 3.134589410704979e-29
Vì p < alpha nên bác bỏ H0
```

Như vậy, qua kiểm định Shapiro, ta có thể thấy dữ liệu các nhóm không có phân phối chuẩn với mức ý nghĩa 0.05.

Kết quả kiểm định cho thấy phương sai của các nhóm giống nhau, tuy nhiên, phân phối của dữ liệu không chuẩn, thì không thỏa mãn giả định quan trọng khi áp dụng ANOVA.

Trong trường hợp này, ta sẽ sử dụng kiểm định ANOVA không đòi hỏi phân phối chuẩn như Welch's ANOVA test:

```
# Kiểm định Welch's ANOVA
print('H0: Mức tăng lương trung bình của 3 phòng ban bằng nhau')
print('H1: Có sự khác biệt về mức tăng lương trung bình giữa các phòng ban')
stat, p = stats.f_oneway(group1, group2, group3)
print(f'Trị số p = {p:4f}')
if p > 0.05:
  print("Vì p > alpha nên không bác bỏ H0")
else:
  print("Vì p < alpha nên bác bỏ H0")
```

```
H0: Mức tăng lương trung bình của 3 phòng ban bằng nhau
H1: Có sự khác biệt về mức tăng lương trung bình giữa các phòng ban
Trị số p = 0.389468
Vì p > alpha nên không bác bỏ H0
```

Kết quả kiểm định Welch's ANOVA là chấp nhận H0 cho thấy không có sự khác biệt về mức tăng lương trung bình giữa các phòng ban với mức ý nghĩa 0.05. Vì vậy, có thể kết luận rằng công ty thực hiện chính sách mức tăng lương một cách công bằng đối với tất cả các phòng ban. Điều này có thể hỗ trợ trong việc duy trì lòng cam kết và động lực của nhân viên trong công ty.

5 CHƯƠNG 5: DỰ ĐOÁN VỚI CÁC MÔ HÌNH MÁY HỌC

Đối với bộ dữ liệu HR Analytics, biến target dùng để dự đoán là Attrition, tương ứng với việc dự đoán xem liệu một nhân viên có rời bỏ công ty hay không. Việc này có thể được thực hiện bằng cách áp dụng các mô hình máy học có giám sát như phổ biến thường được sử dụng để phân lớp như KNN, Naive Bayes Classifier, Support Vector Machine, và Decision Tree. Ngoài ra, đối với bài toán hồi quy, biến mục tiêu được chọn là MonthlyIncome để dự đoán lương thu nhập theo tháng của nhân viên. Trong phạm vi đồ án, thuật toán Linear Regression sẽ được áp dụng để dự đoán lương của nhân viên bằng hai phương pháp tối ưu khác nhau là Gradient Descent và tìm nghiệm tối ưu bằng ma trận nghịch đảo.

5.1 Cơ sở lý thuyết các thuật toán máy học phân lớp

5.1.1 KNN

K-Nearest Neighbors (KNN) là một thuật toán học máy có giám sát giúp phân lớp cho một quan sát mới bằng cách xem xét những điểm dữ liệu láng giềng (những điểm gần nhất) trong không gian đặc trưng. Điểm đặc biệt của KNN là thay vì học một mô hình

tường minh và chính xác, thuật toán KNN lưu trữ các quan sát trong tập dữ liệu để sử dụng cho việc phân loại hoặc dự đoán sau này.

Một điểm đặc trưng khác của KNN là nó không lưu lại các kết quả trung gian hay mô hình học được từ dữ liệu. Thay vào đó, KNN dựa vào việc lưu trữ trực tiếp các quan sát và thực hiện tính toán cho một quan sát mới dựa trên những láng giềng gần nhất.

KNN sử dụng độ tương đồng hoặc khoảng cách để xác định sự giống nhau giữa các điểm dữ liệu. Trong trường hợp dữ liệu có dạng numerical, các phương pháp tính độ tương đồng như cosine similarity hoặc tích vô hướng (scalar product) được sử dụng. Trong đó, chỉ số cosine similarity được định nghĩa như sau:

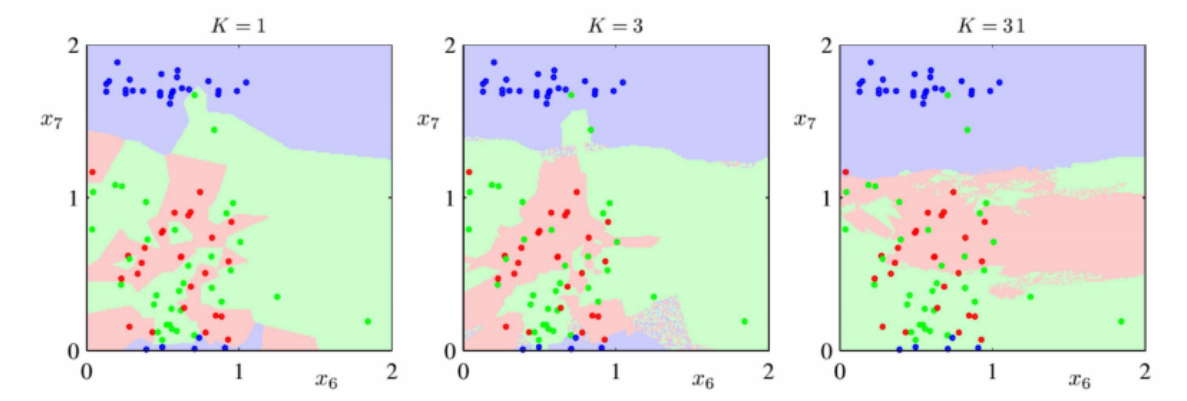
$$\text{sim}(x,y) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

và tích vô hướng được định nghĩa như sau:

$$\text{sim}(x,y) = \langle x,y \rangle = \sum_{i=1}^{n} x_i \cdot y_i$$

Một yếu điểm của cosine similarity là chỉ số này chỉ xét đến góc giữa hai vector chứ không xét đến độ dài của hai vector này. Các khoảng cách thường được sử dụng để tính toán độ tương đồng của các điểm dữ liệu trong KNN bao gồm khoảng cách Mahattan, khoảng cách Euclid, và khoảng cách Minkowski. Đối với dữ liệu phân loại, hàm Hamming thường được áp dụng để so sánh giá trị của các điểm dữ liệu này. Các láng giềng trong thuật toán KNN được xác định dựa trên mức độ tương đồng hoặc khoảng cách từ điểm dữ liệu cần dự đoán. Các mức độ tương đồng và khoảng cách này có thể có vai trò đồng đều với nhau trong việc quyết định hoặc có trọng số khác nhau dựa trên khoảng cách tuỳ vào bài toán cụ thể.

Việc xác định số lượng láng giềng k đôi khi là quyết định khó khăn. Khi $k = 1$, mô hình dễ bị ảnh hưởng bởi nhiễu và dữ liệu nhiễu có thể gây ra sai sót lớn. Giá trị k nhỏ có thể dẫn đến việc đường biên quyết định không trơn, như được minh hoạ ở Hình và có thể gây *overfitting*. Trong khi đó, k lớn có thể phá vỡ cấu trúc cục bộ trong dữ liệu. Một số khuyến nghị khi xác định k là chọn $k = \frac{\sqrt{n}}{2}$, nếu số lượng quan sát n đủ lớn.



Hình 41: Ảnh hưởng của các giá trị k đến đường biên quyết định

Ưu điểm của KNN bao gồm tính đơn giản, dễ triển khai, chi phí thấp trong giai đoạn học, và khả năng áp dụng cho cả bài toán phân loại và dự đoán. Thuật toán KNN cũng cho phép sự linh hoạt trong việc lựa chọn hàm khoảng cách.

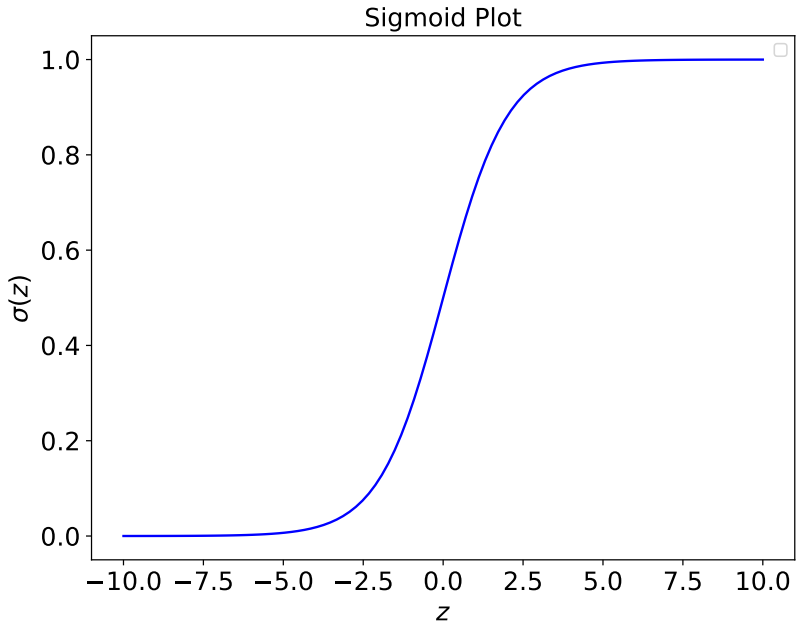
Tuy nhiên, KNN cũng có nhược điểm. Việc xác định giá trị k đôi khi là một quyết định khó khăn và đòi hỏi kiến thức chuyên môn của các chuyên gia. Ngoài ra, chi phí tính toán trong giai đoạn dự đoán cũng có thể tăng lên khi k lớn. Thuật toán KNN cũng tỏ ra không hiệu quả khi phân phối của mục tiêu (target) bị lệch. Trong trường hợp bài toán dự đoán việc nhân viên có rời bỏ hay không, do trong hầu hết các trường hợp, tỉ lệ nhân viên ở lại công ty lớn hơn nhiều so với số lượng nhân viên rời bỏ, nên mô hình KNN sẽ gặp khó khăn trong việc đưa ra dự đoán. Đây là vấn đề về dữ liệu bị mất cân bằng (imbalanced data). Trong thực tế, đây là vấn đề thường gặp phải với các bài toán phân loại phổ biến như phân loại các giao dịch bất thường (credit fraud detection), phân loại việc khách hàng có rời bỏ một dịch vụ mạng viễn thông hay không, ...

5.1.2 Logistic Regression

Hồi quy Logistic *(Logistic Regression)* là phương pháp học có giám sát đơn giản trong học máy. Khác với phương pháp hồi quy tuyến tính , mô hình Logistic Regression ước tính xác suất xảy ra sự kiện thay vì dự đoán các giá trị liên tục. Kết quả dự đoán của mô hình Logistic Regression có thể được xem như một giá trị xác suất thông qua hàm Logistic (hay còn được gọi là hàm Sigmoid) được định nghĩa như sau:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Đồ thị của hàm Sigmoid có thể được biểu diễn như hình bên dưới:



Hình 42: Đồ thị hàm Sigmoid

Hàm Sigmoid là hàm khả vi nên ta có thể sử dụng các thuật toán tối ưu dựa trên đạo hàm để cập nhật các tham số. Ngoài ra, ngưỡng giá trị đầu ra của hàm Sigmoid nằm trong khoảng [0,1], các giá trị này vì vậy mang một ý nghĩa biểu hiện xác suất.

Kết quả phân lớp của mô hình Logistic Regression thường sẽ được biểu diễn dưới dạng 0 hoặc 1 dựa trên một giá trị biên (threshold) nhất định, thường được chọn là 0.5 trong trường hợp phân lớp nhị phân.

$$\hat{y} = \begin{cases} 1 \text{ nếu } \sigma(x) > 0.5 \\ 0 \end{cases}$$

Hàm mất mát thường được sử dụng để huấn luyện mô hình Logistic Regression là hàm Cross entropy và được định nghĩa như sau:

$$J(\mathbf{w}) = -[y log(\sigma(\mathbf{z}) + (1-y) log(1-\sigma(\mathbf{z}))]$$

trong đó $\sigma(\mathbf{z}) = \dfrac{1}{1+e^{-\mathbf{z}}}$ và $\mathbf{z} = \mathbf{w}^T\mathbf{x}$

Việc cập nhật và tối ưu các tham số w có thể được thực hiện bằng thuật toán Gradient Descent để cực tiểu hoá hàm mất mát Cross Entropy theo công thức:

$$w_{t+1} = w_t - \rho \times \frac{\partial L(w)}{\partial w}$$

Trong đó α là giá trị learning rate hay tốc độ cập nhật cho mỗi lần tính toán lại trong số.

Mô hình Logistic Regression tuy đơn giản nhưng mang lại các kết quả tương đối tốt trong các bài toán phân loại nhị phân như chẩn đoán y khoa, phát hiện gian lận, phân loại email (spam hay không spam), và dự đoán liệu một nhân viên có rời bỏ công ty hay không.

5.1.3 Naive Bayes Classifier

Naive Bayes là một mô hình xác suất dựa trên nền tảng của định lý Bayes, được sử dụng cho các bài toán phân loại. Mô hình Naive Bayes giả định rằng các thuộc tính trong dữ liệu là độc lập với nhau, thuật ngữ Naive (ngây thơ) nhằm mục đích ám chỉ giả định này. Mặc dù giả định này có thể không đúng trong các tình huống thực tế, nhưng sự đơn giản và hiệu quả của Naive Bayes làm cho nó được sử dụng rộng rãi và hiệu quả trong nhiều bài toán.

Giả sử chúng ta có một tập hợp dữ liệu đầu vào (evidence), thường được biểu diễn dưới dạng tuple $X = (x_1, x_2, ..., x_n)$, trong đó mỗi x_j thuộc vào miền giá trị $DOM(A_j)$ tương ứng. Định lý Bayes liên quan đến việc xác định xác suất hậu nghiệm (posterior probability) $P(H|X)$, tức xác suất của một giả thuyết H dựa trên thông tin quan sát X. Ngoài ra, định lý Bayes cũng liên quan đến xác suất tiên nghiệm (prior probability) $P(H)$, tức xác suất của giả thuyết trước khi có thông tin quan sát.

Một phần quan trọng của định lý Bayes là khả năng tính toán xác suất hậu nghiệm dựa trên xác suất tiên nghiệm và xác suất của dữ liệu được quan sát, được mô tả bởi công thức:

$$P(H|X) = \frac{P(X)P(X|H)}{P(H)} \tag{1}$$

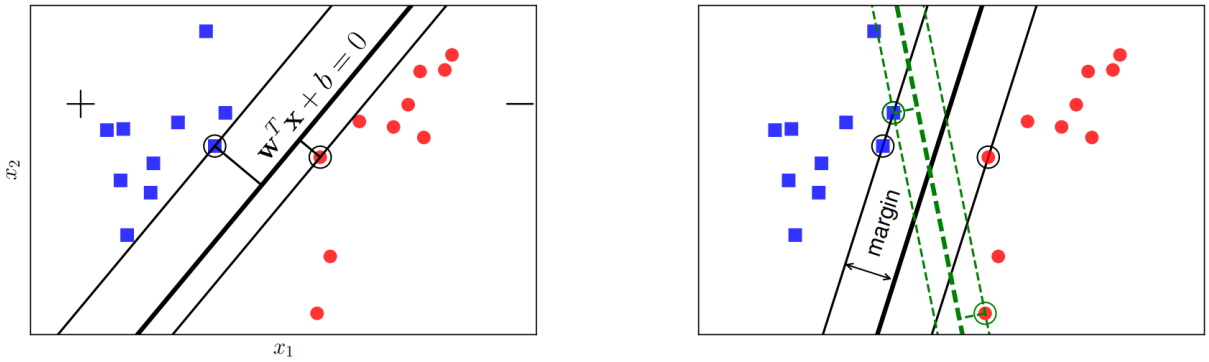
Trong đó:

- $P(H|X)$ là xác suất hậu nghiệm của giả thuyết H dựa trên dữ liệu quan sát X.
- $P(X|H)$ là xác suất của dữ liệu quan sát X nếu giả thuyết H đúng.
- $P(H)$ là xác suất tiên nghiệm của giả thuyết H.
- $P(X)$ là xác suất của dữ liệu quan sát X.

Định lý Bayes là một công cụ giúp cập nhật thông tin về một giả thuyết sau khi có thêm thông tin mới. Điều này làm nền tảng cho việc ra quyết định dựa trên xác suất trong nhiều lĩnh vực khác nhau.

5.1.4 Support Vector Machines

Support Vector Machine (SVM) là một thuật toán học máy dùng cho việc phân loại. Mục tiêu của SVM là tìm siêu phẳng *(hyperplane)* tối ưu nhất để phân chia các điểm dữ liệu của các lớp khác nhau. Siêu phẳng này cố gắng tối đa hoá biên cực đại *(maximum margin)*. Đường biên cực đại được định nghĩa là khoảng cách từ siêu phẳng đến điểm dữ liệu gần nhất của mỗi lớp. Các điểm này còn được gọi là vectơ hỗ trợ *(support vectors)*.



Hình 43: Minh họa thuật toán SVM. *Nguồn: Machine Learning Cơ Bản*

Giả sử ta có tập training set $T = \left\{\left(x^{(i)}, y_i\right)\right\}_{i=1}^{m}$ trong đó các điểm dữ liệu đầu vào $x^{(i)}$ là các vector có d chiều ($x^{(i)} \in \mathbb{R}^d$) và các target có giá trị là -1 hoặc 1 ($y_i \in \{-1, +1\}$).

Ta có khoảng cách từ điểm dữ liệu $\left(x^{(i)}, y_i\right)$ đến siêu phẳng H được tính bởi công thức:

$$d_i = \frac{y_i\left(w^T x^{(i)} + b\right)}{\|w\|}$$

Đường biên *(margin)* được định nghĩa là khoảng cách nhỏ nhất từ điểm dữ liệu $\left(x^{(i)}, y_i\right)$ đến siêu phẳng H, hay

$$\text{margin} = \min_i \frac{y_i\left(w^T x^{(i)} + b\right)}{\|w\|}$$

Bài toán tối ưu trong thuật toán SVM là đi tìm các tham số (w, b) sao cho đường biên là lớn nhất, hay:

$$(w,b) = \arg\max_{w,b} \left\{ \min_i \frac{y_i \left(w^T x^{(i)} + b \right)}{\|w\|} \right\} = \arg\max_{w,b} \left\{ \frac{1}{\|w\|} \min_i \left(y_i \left(w^T x^{(i)} + b \right) \right) \right\} \tag{2}$$

Phương trình 2 có thể được viết lại dưới dạng:

$$(w,b) = \arg\min_{w,b} \|w\|^2, \text{ với } 1 - y_i \left(w^T x^{(i)} + b \right) \leq 0 \tag{3}$$

Bài toán tối ưu ở phương trình 3 thường được giải qua phương pháp quy hoạch toàn phương *(quadratic programming)* với hàm mục tiêu là một chuẩn L_2 và các ràng buộc tuyến tính.

Khi có dữ liệu mới, SVM sử dụng siêu phẳng đã học để dự đoán lớp của dữ liệu đó. Quá trình này bao gồm việc tính toán giá trị của hàm quyết định dựa trên vector trọng số w và độ lệch (bias) b.

Thuật toán SVM được sử dụng rộng rãi trong nhiều ứng dụng thực tế nhờ vào khả năng phân loại tốt và tính linh hoạt trong xử lý dữ liệu. Một trong những ưu điểm nổi bật của SVM là khả năng xử lý cả dữ liệu được phân tách tuyến tính và phi tuyến, nhờ vào việc sử dụng các hàm kernel để ánh xạ dữ liệu vào không gian nhiều chiều.

Tuy nhiên, những hạn chế của SVM bao gồm việc thời gian huấn luyện có thể tăng đáng kể đối với các bộ dữ liệu lớn, và thuật toán này có thể trở nên kém hiệu quả khi số chiều của dữ liệu lớn hơn số mẫu dữ liệu huấn luyện.

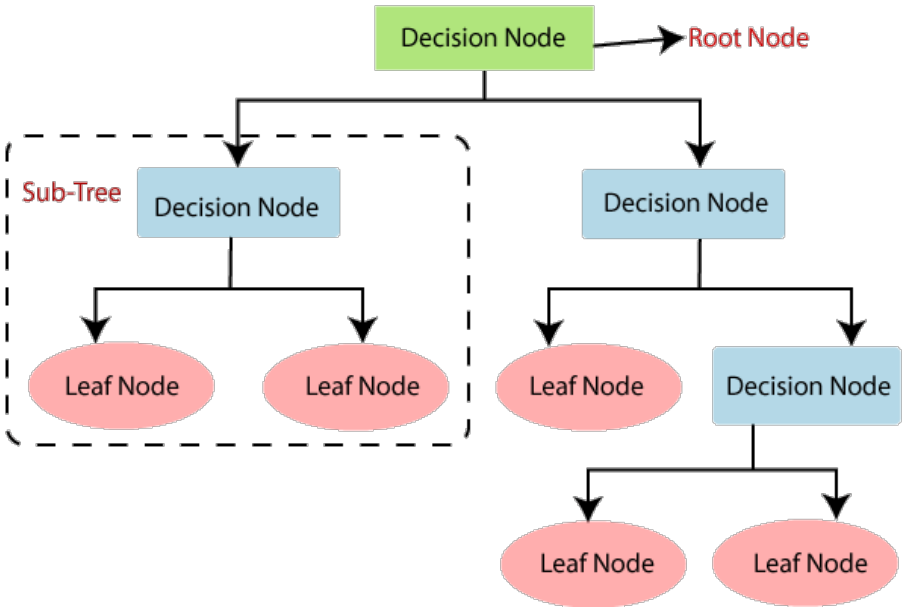
Mặc dù SVM có độ chính xác cao và ít bị overfitting, nhưng nó cũng có thể nhạy cảm với nhiễu và outliers trong dữ liệu. Vì vậy, khi áp dụng thuật toán, ta cần tinh chỉnh các tham số phù hợp để đảm bảo hiệu suất tốt nhất.

5.1.5 Decision Tree

Cây quyết định là một phương pháp hiệu quả trong học máy thường được sử dụng để giải quyết cả bài toán phân loại và dự đoán. Trong một cây quyết định đơn, mỗi bước trong quá trình xây dựng cây tập trung vào việc chọn một thuộc tính duy nhất để phân chia dữ liệu.

Cấu trúc của cây quyết định bao gồm các nút gốc *(root node)*, nút trong *(internal node)* và nút lá *(leaf node)*. Nút gốc đại diện cho toàn bộ tập dữ liệu và chứa thuộc tính được chọn để phân chia tốt nhất. Các nút trong, nằm giữa nút gốc và nút lá, thể hiện các quyết

định phân chia dữ liệu dựa trên thuộc tính và ngưỡng tương ứng. Còn nút lá, là các nút cuối cùng của cây, chứa thông tin về quyết định phân lớp cho một trường hợp dữ liệu cụ thể.



Hình 44: Minh hoạ một cấu trúc cây

Quá trình xây dựng cây quyết định bắt đầu từ nút gốc và tiếp tục lựa chọn thuộc tính và ngưỡng tốt nhất để phân chia dữ liệu mỗi bước. Mục tiêu là tạo ra các nhánh sao cho mỗi nhánh chứa các điểm dữ liệu cùng thuộc một lớp hoặc lớp gần nhau nhất có thể. Quá trình này lặp đi lặp lại cho đến khi đạt điều kiện dừng được đặt ra.

Một trong những ưu điểm lớn của cây quyết định là khả năng dễ diễn giải mô hình. Tuy nhiên, cũng cần lưu ý rằng cây quyết định có thể dễ bị overfitting nếu không được kiểm soát đúng cách và cũng có thể không phản ánh được tốt trong trường hợp các mô hình phức tạp hơn. Tuy nhiên, tính linh hoạt và khả năng áp dụng rộng rãi của nó làm cho cây quyết định trở thành một công cụ quan trọng trong học máy.

Cây quyết định là một phương pháp học phi tham số và không yêu cầu các giả thiết về phân phối của các lớp hoặc nhãn. Cấu trúc của cây không được xác định trước mà sẽ được tạo ra trực tiếp từ dữ liệu quan sát được. Quá trình xây dựng cây quyết định từ tập huấn luyện bắt đầu bằng việc sắp xếp các thuộc tính và tạo các nút,sau đó các nhánh của cây được tạo ra bằng cách phân chia tuples.

Ở mỗi bước trong quá trình xây dựng cây, thuật toán chọn thuộc tính tạo ra phân hoạch tốt nhất trên các quan sát liên quan, thông qua việc truyền từ nút cha đến các nút con. Điều kiện dừng của quá trình xây dựng có thể là khi phân hoạch hoàn toàn tất cả các quan sát hoặc khi tất cả các thuộc tính đã được sử dụng (mỗi thuộc tính chỉ được sử dụng

một lần duy nhất trong cây).

Cây quyết định là một phương pháp học máy linh hoạt và dễ hiểu, xây dựng cấu trúc từ dữ liệu huấn luyện thông qua việc sử dụng các độ đo như entropy, information gain và gain ratio. Entropy đo lường mức độ không chắc chắn trong dữ liệu, trong khi information gain ước lượng sự thay đổi về thông tin khi sử dụng một thuộc tính cụ thể để phân chia dữ liệu.

Mặc dù cây quyết định dễ hiểu và áp dụng rộng rãi cho nhiều loại dữ liệu, nhưng nó cũng có nhược điểm. Việc xây dựng mô hình có thể tốn nhiều thời gian với dữ liệu lớn và không hiệu quả với dữ liệu định lượng. Đồng thời, sự thay đổi nhỏ trong dữ liệu huấn luyện cũng có thể dẫn đến sự thay đổi lớn trong cấu trúc của cây quyết định.

5.2 Các chỉ số để đánh giá kết quả các thuật toán phân lớp

5.2.1 Accuracy

Độ đo accuracy trong bài toán phân loại là một phép đo quan trọng đo lường khả năng chính xác của mô hình. Đây là tỉ lệ phần trăm giữa số lượng các quan sát được phân loại đúng và tổng số lượng quan sát trong tập dữ liệu.

Nói cách khác, accuracy là tỉ lệ giữa số lượng dự đoán chính xác và tổng số lượng dự đoán được thực hiện bởi mô hình. Accuracy được tính bằng công thức:

$$Accuracy = \frac{TP + TN}{n}$$

Một accuracy cao (gần 1 hoặc 100%) chỉ ra rằng mô hình đang dự đoán chính xác một phần lớn các điểm dữ liệu trong tập kiểm tra. Tuy nhiên, việc sử dụng accuracy cần cân nhắc đối với các tập dữ liệu mất cân bằng (có sự chênh lệch lớn giữa số lượng các lớp). Nếu một lớp có số lượng quan sát nhiều hơn mức trung bình, accuracy có thể không phản ánh chính xác khả năng dự đoán của mô hình. Trong trường hợp này, các độ đo khác như precision, recall hoặc F1-score có thể cung cấp cái nhìn tổng quát và chi tiết hơn về hiệu suất của mô hình trên các lớp khác nhau.

Accuracy cũng có thể không phù hợp trong những trường hợp mà việc phân loại sai lớp này thành lớp khác có ý nghĩa lớn hơn (ví dụ: trong bài toán dự đoán nhân viên rời bỏ, việc phân loại sai một nhân viên rời bỏ thành không rời bỏ có thể có hậu quả nghiêm trọng hơn việc phân loại sai ngược lại).

Vậy nên, trong khi accuracy là một độ đo quan trọng, việc hiểu rõ về bản chất của dữ liệu và bài toán cũng như sử dụng các độ đo khác cùng với accuracy là quan trọng để đánh giá chính xác hiệu suất của mô hình.

5.2.2 Precision

Precision là tỉ lệ của số lượng các dự đoán positive đúng chia cho tổng số lượng dự đoán positive (bao gồm cả dự đoán đúng và dự đoán sai) và được định nghĩa như sau:

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

Đối với dữ liệu không cân bằng, mô hình có thể dễ dàng đạt được precision cao bằng cách dự đoán tất cả các quan sát thuộc lớp ít phổ biến (minority class) là positive, kể cả khi dự đoán này không mang lại giá trị thực sự. Điều này có thể dẫn đến tình trạng precision cao nhưng mô hình không thực sự hữu ích, vì nó không phản ánh khả năng dự đoán chính xác trên toàn bộ dữ liệu.

Ví dụ, trong việc dự đoán các nhân viên rời bỏ (đây thường là các trường hợp ít phổ biến), mô hình có thể dự đoán toàn bộ trường hợp là rời bỏ để tăng precision. Tuy nhiên, điều này không mang lại giá trị thực tế nếu mô hình bỏ qua hoặc dự đoán sai nhiều trường hợp không rời bỏ.

Do đó, trong dữ liệu không cân bằng, việc sử dụng precision cần phải kết hợp với các độ đo khác như recall, F1-score hoặc ROC-AUC để đánh giá một cách toàn diện hơn về hiệu suất của mô hình, đặc biệt là trên các lớp thiểu số. Ngoài ra, việc áp dụng các kỹ thuật như undersampling, oversampling hoặc sử dụng các thuật toán đối với dữ liệu không cân bằng cũng có thể cải thiện đáng kể hiệu suất của mô hình và đáng tin cậy của các độ đo như precision.

5.2.3 Recall

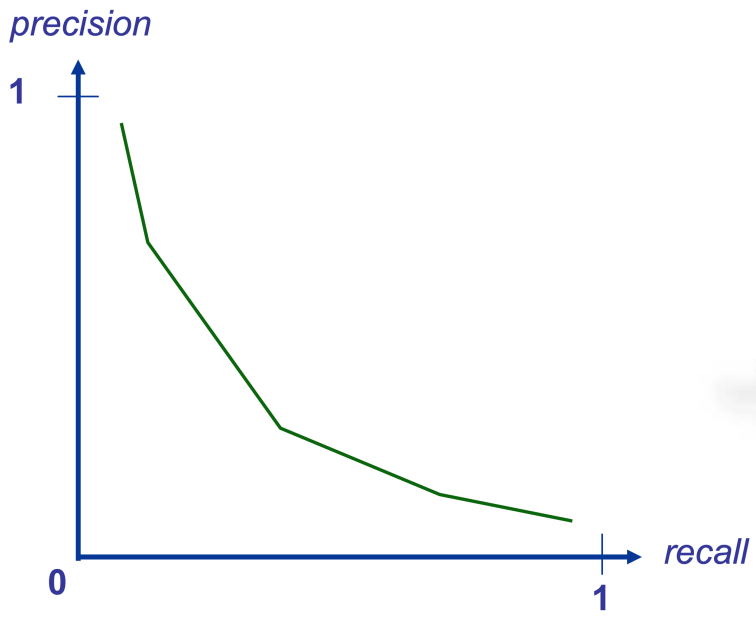
Độ đo recall là một trong những phép đo quan trọng để đánh giá hiệu suất của mô hình phân loại, đặc biệt là trên lớp thiểu số (positive class). Recall (còn được gọi là sensitivity) đo lường khả năng của một mô hình phân loại trong việc nhận diện đúng các quan sát thuộc lớp positive so với tổng số lượng các quan sát thực sự thuộc lớp positive, được định nghĩa như sau:

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

Recall cao chỉ ra rằng mô hình có khả năng nhận diện đúng một phần lớn các trường hợp positive trong tập kiểm tra. Tuy nhiên, trong tình huống dữ liệu không cân bằng, mô hình có thể đạt được recall cao bằng cách dự đoán tất cả các trường hợp là positive, kể cả

khi dự đoán này không chính xác. Điều này có thể dẫn đến tình trạng recall cao nhưng mô hình không hữu ích thực sự.

Ngoài ra, ta có thể dễ dàng nhận thấy Precision và Recall có tương quan đối nghịch nhau. Nói cách khác, việc tối ưu mô hình để tăng Precision sẽ làm giảm Recall và ngược lại. Mối quan hệ này có thể được biểu diễn như hình sau:



5.2.4 F_1 score

F1-score là một chỉ được sử dụng rộng rãi trong học máy kết hợp chỉ số Precision và Recall thành một thước đo duy nhất. Chỉ số F1 score cung cấp đánh giá cân bằng về hiệu suất của mô hình, đặc biệt là trong các vấn đề phân loại nhị phân.

F1-score được tính là trung bình điều hoà *(harmonic mean)* của độ đo Precision và Recall và được định nghĩa như sau:

$$F_1 \text{ score } = 2 \times \frac{precision \times recall}{precision + recall}$$

Khi Precision và Recall có giá trị gần nhau, F1-score có xu hướng cao, cho thấy sự cân bằng giữa Precision và Recall.

Chỉ số F1 score rất quan trọng trong việc đánh giá các mô hình phân loại, đặc biệt là trong các tình huống mà False Positive và False Negative mang ý nghĩa quan trọng. F1 score cao tương ứng với việc hiệu suất mô hình có hiệu quả cao về cả Precision và Recall.

5.3 Kết quả dự đoán việc rời bỏ của nhân viên

Do dữ liệu đang bị mất cân bằng vì số lượng nhân viên không rời bỏ đang lớn hơn rất nhiều so với số lương nhân viên rời bỏ, nằm ở tỉ lệ 80%-20%, nên đầu tiên ta sẽ tiến hành sampling lại dữ liệu:

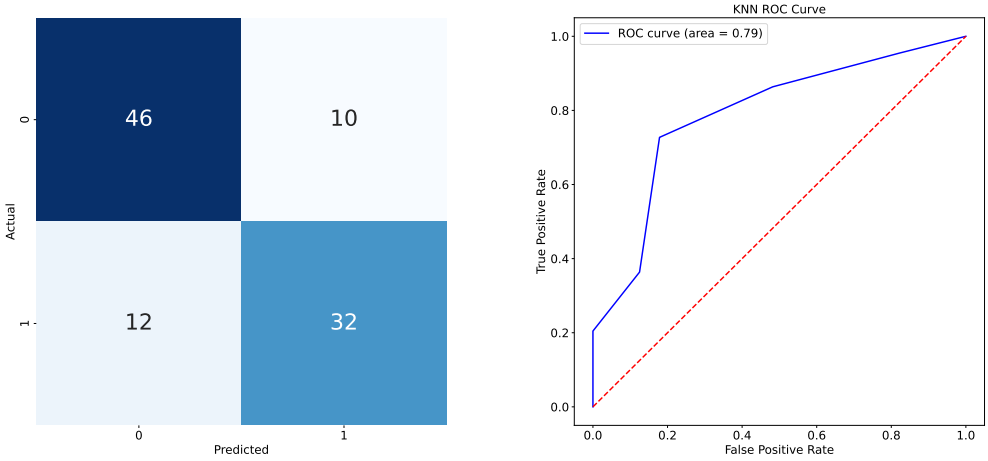
```
attrition_0 = df[df['Attrition']==0]
attrition_1 = df[df['Attrition']==1]
count_attrition_0 = len(attrition_0)
count_attrition_1 = len(attrition_1)

attrition_0_downsampled = resample(attrition_0, replace=False,
↪  n_samples=int(count_attrition_1*1.1), random_state=42)

df = pd.concat([attrition_0_downsampled, attrition_1])
```

Tỉ lệ nhân viên không rời bỏ và rời bỏ trong bộ dữ liệu lúc này là 53%-47%. Kết quả dự đoán của các mô hình có thể được tóm tắt trong bảng sau:

Model	**Metrics**			
	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.720	0.654	0.772	0.708
Naive Bayes	0.700	0.629	0.772	0.693
Support Vector Classifier	0.720	0.674	0.809	0.688
Decision Tree	0.680	0.630	0.659	0.644
KNN	0.780	0.762	0.727	0.744

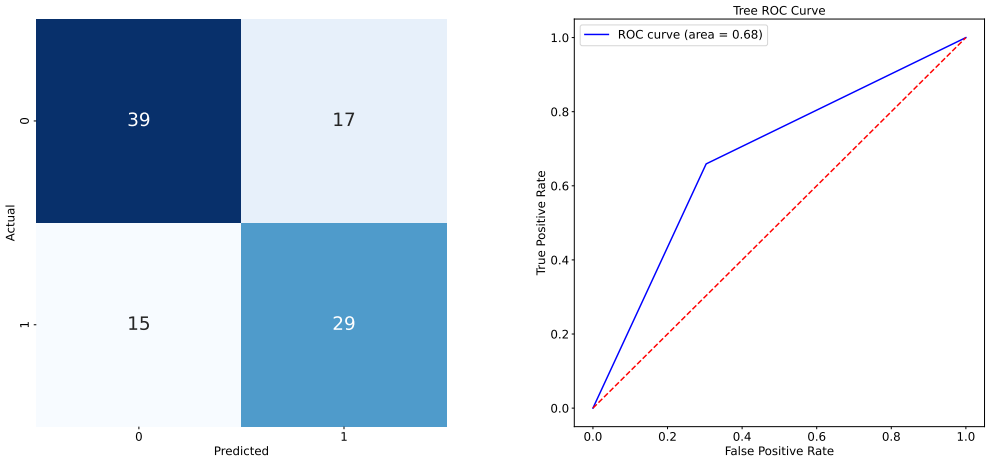
Bảng 3: Đánh giá các mô hình phân lớp

Trong đó KNN là mô hình có kết quả phân lớp tốt nhất với điểm Accuracy, Precision, Recall, và F1 Score lần lượt là 0.78; 0.762; 0.727; và 0.744.

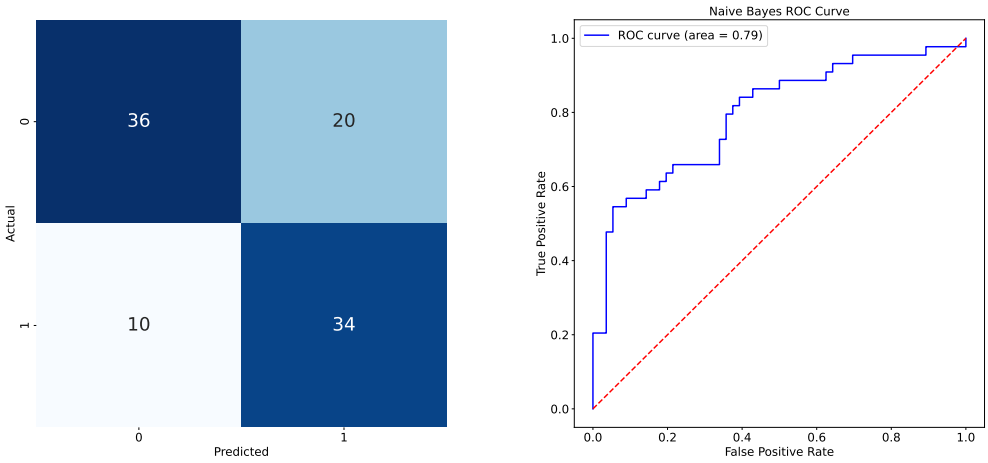


Hình 45: Ma trận nhầm lẫn và đường cong ROC của mô hình KNN

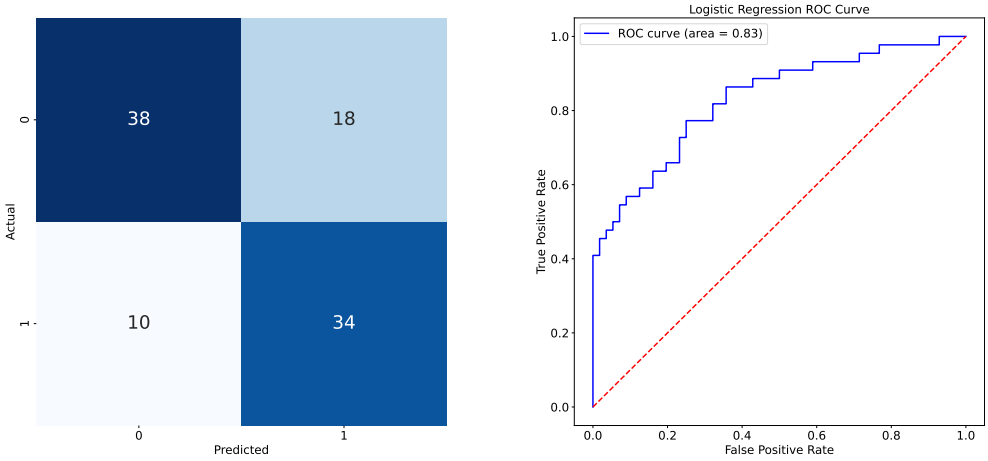
Mô hình Decision Tree có kết quả phân lớp kém nhất trong lớp 5 mô hình trong phạm vi đồ án, với điểm Accuracy là 0.68 và F1 Score là 0.644.



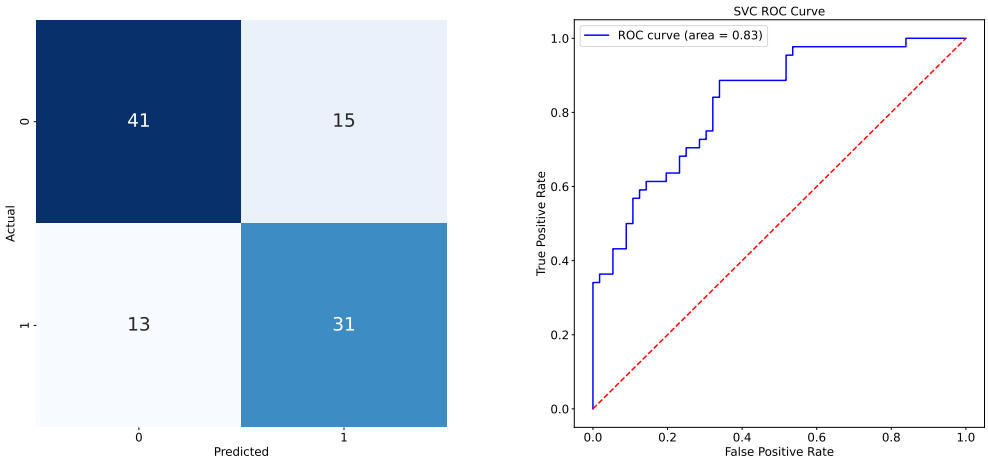
Hình 46: Ma trận nhầm lẫn và đường cong ROC của mô hình Decision Tree



Hình 47: Ma trận nhầm lẫn và đường cong ROC của mô hình Naive Bayes



Hình 48: Ma trận nhầm lẫn và đường cong ROC của mô hình Logistic Regression



Hình 49: Ma trận nhầm lẫn và đường cong ROC của mô hình Support Vector Classifier

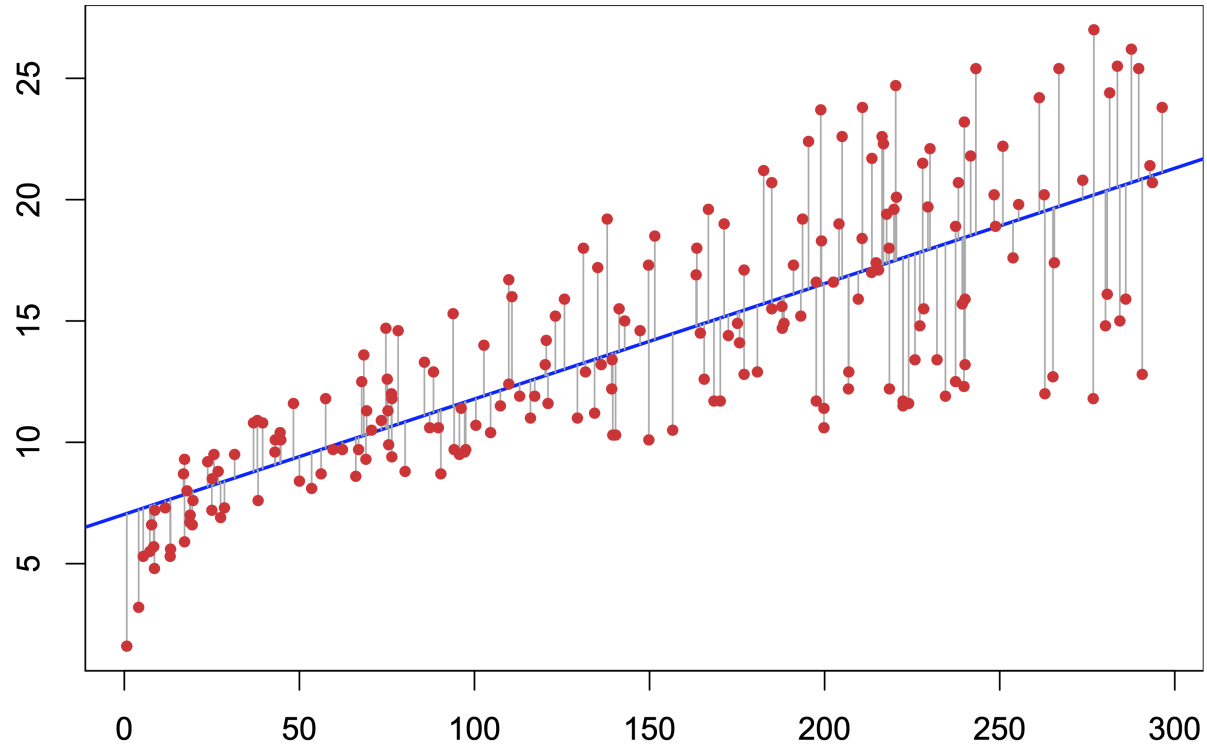
5.4 Hồi quy dự đoán lương nhân viên bằng mô hình Linear Regression

Hồi quy tuyến tính *(Linear regression)* là một phương pháp thống kê nền tảng và là một trong những phương pháp máy học cơ bản được sử dụng để hiểu và định lượng mối quan hệ giữa hai hoặc nhiều biến. Mục tiêu của hồi quy tuyến tính là đi tìm một phương trình tuyến tính thể hiện tốt nhất mối quan hệ giữa một biến phụ thuộc y và một hoặc nhiều biến độc lập X, như được minh hoạ ở Hình 50. Nói cách khác, với mỗi điểm dữ liệu đầu vào x có thể được biểu diễn dưới dạng ma trận

$$\hat{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & \dots & x_{md} \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_m \end{bmatrix}$$

và các output y được biểu diễn dưới dạng vector

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$



Hình 50: Ví dụ về thuật toán Linear regression. *Nguồn: James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). An introduction to statistical learning*

Linear Regression là việc đi tìm vector: $w = (w_0, w_1, \cdots, w_d)$ sao cho $\hat{Y} = \hat{X} \cdot w \approx Y$ tốt nhất.

Về bản chất, hồi quy tuyến tính tìm đi tìm một đường thẳng sao cho sự khác biệt giữa các giá trị dự đoán và các giá trị quan sát thực tế là nhỏ nhất. Đường thẳng này này được thể hiện bằng phương trình có dạng

$$\hat{y} = w_1\hat{x}_1 + w_2\hat{x}_2 + \cdots + w_n\hat{x}_n + w_0 = \mathbf{w}^T\mathbf{x} \tag{4}$$

Trong đó $w_1, w_2, \cdots w_n$ là các tham số cần được tối ưu và $\hat{x}_1, \hat{x}_2, \cdots \hat{x}_n$ là các giá trị đầu vào.

Các tham số trong phương trình 4 được tối ưu bằng cách cực tiểu hoá hàm mất mát thông qua thuật toán Gradient Descent, trong đó hàm mất mát (loss function) được định nghĩa như sau:

$$L(w) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{x}_i w)^2 \tag{5}$$

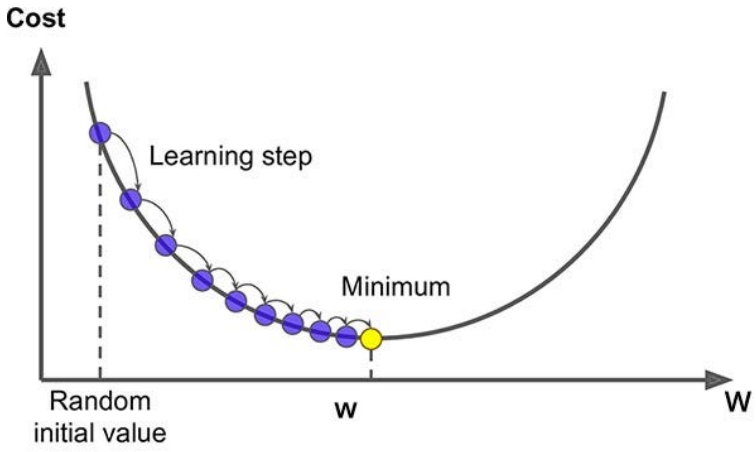
Trong đó

- $\hat{x}_i, w$ lần lượt là các điểm dữ liệu trong tập dữ liệu huấn luyện và tham số của mô hình.
- y_i là giá trị thật của biến target
- m là số điểm dữ liệu

Hệ số $\frac{1}{2}$ trước phương trình được dùng để khử hệ số 2 khi lấy đạo hàm cho hàm $L(w)$, việc bình phương sai số giữa giá trị thật và giá trị dự đoán làm cho hàm $L(w)$ khả vi tại mọi điểm, từ đó việc thực hiện thuật toán Gradient Descent sẽ dễ dàng hơn. Việc chia giá trị cho m giúp giảm tình trạng khi có càng nhiều điểm dữ liệu thì giá trị của hàm mất mát sẽ càng lớn.

5.4.1 Gradient Descent

Gradient Descent là thuật toán tối ưu dựa vào đạo hàm bậc 1 để tìm ra các điểm tối ưu (cực đại hoặc cực tiểu) của một hàm số bằng cách cập nhật các tham số w ngược dấu với đạo hàm. Trong Linear Regression, Gradient Descent là thuật toán nhằm đi tìm các tham số w tối ưu sao cho giá trị của phương trình . là nhỏ nhất.



Hình 51: Minh họa thuật toán Gradient Descent

Đầu tiên ta sẽ khởi tạo (initialize) các tham số w. Các tham số w sau đó sẽ được cập nhật dựa trên giá trị đạo hàm của hàm mất mát theo công thức như sau:

$$w_{t+1} = w_t - \rho \times \frac{\partial L}{\partial w}$$

Các trọng số sẽ được cập nhật theo công thức trên cho đến khi không còn sự thay đổi nào trong sự thay đổi tiếp theo, lúc này thuật toán đã hội tụ.

Nói cách khác, Gradient Descent là thuật toán giúp tìm ra vector w^* thoả mãn:

$$w^* = \underset{w}{\operatorname{argmin}}\, L(w)$$

5.4.2 Normal Equation

Một phương pháp khác để tối ưu các tham số trong mô hình Linear Regression là đi tìm trực tiếp ma trận **W** sao cho giá trị của hàm mất mát là nhỏ nhất. Thực hiện việc tính đạo hàm riêng của hàm mất mát $L(w)$ theo w_i, ta được:

$$\frac{\partial L(w)}{\partial w} = \begin{pmatrix} \frac{\partial L(w)}{\partial w_1} \\ \frac{\partial L(w)}{\partial w_2} \\ \vdots \\ \frac{\partial L(w)}{\partial w_m} \end{pmatrix} = \frac{1}{m}\hat{X}^T(\hat{X}\cdot w - Y) \tag{6}$$

Để tìm cực trị của hàm $L(w)$, ta đặt phương trình trên bằng 0:

$$\frac{\partial L(w)}{\partial w} = \frac{1}{m}\hat{X}^T(\hat{X}\cdot w - Y) = 0 \tag{7}$$

Và giải bằng cách nhân với ma trận nghịch đảo:

$$\hat{X}^T\hat{X}\cdot w = \hat{X}^T Y \rightarrow w = (\hat{X}^T\hat{X})^{-1}\hat{X}^T Y$$

Trong trường hợp ma trận $\hat{X}^T\hat{X}$ không có nghịch đảo, ta sẽ tìm w tối ưu bằng cách nhân với ma trận giả nghịch đảo *(Pseudo-inverse)*

$$\hat{X}^T\hat{X}\cdot w = \hat{X}^T Y \rightarrow w = (\hat{X}^T\hat{X})^{\dagger}\hat{X}^T Y$$

5.4.3 Đánh giá kết quả

MSE

Độ chính xác của thuật toán Linear Regression có thể được đánh giá qua các chỉ số khác nhau, trong đó trung bình sai số bình phương (Mean Squared Error) là một trong những chỉ số được dùng phổ biến nhất và được định nghĩa như sau:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

RMSE

Một điểm yếu của MSE là đại lượng bình phương $(\hat{y}_i - y_i)^2$ dẫn đến sự không đồng nhất về mặt đơn vị. Vì vậy một chỉ số đo khác thường được sử dụng là *Root Mean Squared Error* (RMSE), được tính toán bằng cách lấy căn bậc hai của MSE và được định nghĩa như sau:

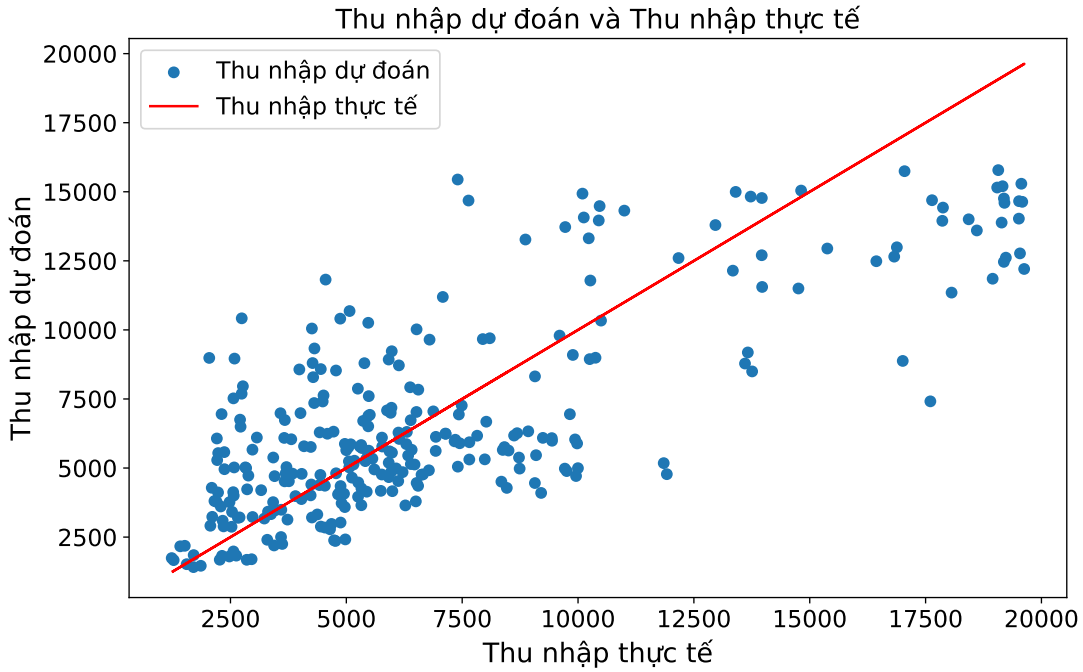
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

5.4.4 Dự đoán lương hằng tháng của nhân viên trên bộ dữ liệu HR Analytics

Đối với phương pháp tối ưu Gradient Descent

```
selected_features = ['Age', 'DailyRate', 'DistanceFromHome', 'HourlyRate',
↪  'NumCompaniesWorked',
                     'PercentSalaryHike', 'PerformanceRating',
                     ↪  'TotalWorkingYears', 'TrainingTimesLastYear',
                     'YearsAtCompany', 'YearsInCurrentRole',
                     ↪  'YearsSinceLastPromotion', 'YearsWithCurrManager',
                     'MonthlyIncome']

subset_df = df[selected_features].dropna()

X = subset_df.drop('MonthlyIncome', axis=1)
y = subset_df['MonthlyIncome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↪  random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

predictions = model.predict(X_test)

plt.figure(figsize=(10, 6))

plt.scatter(y_test, predictions, label='Thu nhập dự đoán')
plt.plot(y_test, y_test, color='red', label='Thu nhập thực tế')
```

```
22
23 plt.xlabel('Thu nhập thực tế', fontsize=16)
24 plt.ylabel('Thu nhập dự đoán', fontsize=16)
25 plt.title('Thu nhập dự đoán và Thu nhập thực tế', fontsize=16)
26 plt.xticks(fontsize=14)
27 plt.yticks(fontsize=14)
28 plt.legend(fontsize=14)
29 plt.savefig('figs/linear_regression.pdf')
30 plt.show()
```



Hình 52: Biểu đồ thể hiện lương tháng dự đoán bởi mô hình Linear Regression và lương thực tế của nhân viên

Các chỉ số đánh giá cho mô hình Linear Regression trong việc dự đoán lương nhân viên như sau:

- MSE: 8015332.445
- RMSE: 2831.136
- R2 Score: 0.591

Tìm nghiệm trực tiếp bằng cách nhân với ma trận nghịch đảo

```
X_train_np = X_train.values
y_train_np = y_train.values
X_train_np = np.c_[np.ones(X_train_np.shape[0]), X_train_np]
theta =
 ↪  np.linalg.inv(X_train_np.T.dot(X_train_np)).dot(X_train_np.T).dot(y_train_np)
```

Cả 2 phương pháp đều cho cùng một nghiệm w tối ưu:

```
Coefficients: [ 3.46709368e+03 -2.49369388e+01  7.93184100e-02 -1.10664194e+01
 -4.49263025e+00 -3.84181651e+01  2.79621178e+01 -5.41542170e+02
  4.92096977e+02  2.88099342e+01  1.05108119e+02 -4.05905332e+01
  5.79682946e+01 -1.56105760e+02]
```

và vì vậy các sai số của phương pháp nhân ma trận nghịch đảo cũng giống với phương pháp tối ưu Gradient Descent.

6 ĐÁNH GIÁ VÀ KẾT LUẬN

Đồ án Phân tích dữ liệu HR Analytics đã cung cấp một góc nhìn toàn vẹn và khách quan về bộ dữ liệu HR Analytics. Đồ án đã thực hiện nhiều công đoạn từ tiền xử lí dữ liệu đến dự đoán bằng các mô hình máy học và đưa ra những đánh giá về khả năng của mô hình. Thông qua đồ án, các phân tích chi tiết liên quan đến bộ dữ liệu, bao gồm cả các thuộc tính định lượng và định tính đã được đưa ra để khai thác và hiểu hơn về bộ dữ liệu. Ngoài ra, đồ án cũng đã thực hiện các phương pháp biểu diễn trực quan dữ liệu và kiểm định giả thuyết để phân tích mối quan hệ của các yếu tố trong lĩnh vực nhân sự, đem lại những thông tin cần thiết cho bộ dữ liệu nói riêng và lĩnh vực nhân sự nói chung.

Tuy nhiên, đồ án vẫn tồn tại một số hạn chế. Phương án xử lý ngoại lai của nhóm tuy là để đảm bảo hiệu quả cho mô hình máy học và những phương pháp thống kê, nó cũng dẫn đến việc đánh đối tính thực tế của dữ liệu. Vì vậy những phân tích được đưa ra chỉ áp dụng trong phạm vi của môn học này mà chưa thực sự có tính thực tiễn trong lĩnh vực của đề tài đồ án.

Bảng phân công

Thành viên	Phân công	Đánh giá
Vũ Nguyễn Thảo Vi	Kiểm tra các đại lượng về xu thế trung tâm Kiểm tra sự tương quan giữa các đại lượng Thống kê mô tả và BDTQDL Kiểm định tương quan giữa 2 biến liên tục	100%
Nguyễn Quốc Việt	Thống kê mô tả và BDTQDL Mô hình máy học phân lớp dự đoán nhân viên rời bỏ Mô hình máy học hồi quy dự đoán thu nhập	100%
Nguyễn Thanh Vy	Kiểm định tương quan giữa biến liên tục và biến phân loại Kiểm định tương quan giữa hai biến phân loại Thống kê mô tả và BDTQDL	100%
Nguyễn Nhật Thảo Vy	Tổng quan Tiền xử lý dữ liệu Kiểm định trung bình tuổi của nhân viên rời đi Kết luận	100%

TÀI LIỆU THAM KHẢO

1. James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). An introduction to statistical learning, volume 112. Springer.

2. Cristianini, N., Ricci, E. (2008). Support Vector Machines. In: Kao, MY. (eds) Encyclopedia of Algorithms. Springer, Boston, MA.

3. Z. (2020, December 20). Welch’s t-test: When to Use it + Examples. Statology. https://www.statology.org/welchs-t-test/

4. Z. (2021, March 16). The Four Assumptions Made in a T-Test. Statology. https://www.statology.org/t-test-assumptions/

5. Assumptions of the Mann-Whitney U test | Laerd Statistics. (n.d.). https://statistics.laerd.com/statistical-guides/mann-whitney-u-test-assumptions.php?fbclid=IwAR1jG3n6kZF5uRLqA9xLDzdXcXEBHxrOCDyiWng_l8NfyqE17V0RjYJedX0

6. Sang, D. (n.d.). Phân biệt các loại thang đo trong nghiên cứu - Hệ thống thông tin Thống kê &CN. http://thongke.cesti.gov.vn/dich-vu-thong-ke/tai-lieu-phan-tich-thong-ke/720-phan-biet-thang-do-trong-nghien-cuu