PREDICT EMPLOYEE ATTRITION

Presented by: Le Hoang Minh Chau
Group: 6

NỘI DUNG CHÍNH

- **Data Overview_** *Giới thiệu tổng quan dữ liệu*
- **Data Cleasning_***Xử lý dữ liệu đầu vào*
- **Data Visualization_***Trực quan hóa dữ liệu*
- **Data Preprocessing_***Tiền xử lý dữ liệu*
- **Data Modeling & Evaluation_***Xây dựng mô hình và đánh giá*
- **Conclusion_** *Kết luận*

DATA OVERVIEW_GIỚI THIỆU

Tổng quan

- Đây là một tập dữ liệu hư cấu được tạo ra bởi các nhà khoa học dữ liệu của IBM với mục tiêu để tìm hiểu những yếu tố nào có thể tác động đến quyết định nghỉ việc của nhân viên
- Nguồn tải dữ liệu: Kaggle

Thông tin cơ bản

- Dữ liệu gồm 35 Cột và 1470 dòng

Mục tiêu

- Phân tích các yếu tố ảnh hướng đến sự nghỉ việc
- Từ các yếu tố đầu vào xây dựng mô hình dự đoán sự nghỉ việc của nhân viên

Thông tin các cột của dữ liệu

- **Age**: độ tuổi
- **Attrition**: Nhân viên có nghỉ việc hay không (Yes/No)
- **BusinessTravel**: Mức độ đi công tác (Non-Travel, Travel_frequently, Travel_Rarely)
- **DailyRate**: Mức lương tính theo ngày
- **Departmen**t: Phòng ban
- **DistanceFromHome**: Khoảng cách từ nhà
- **Education:** Mức độ giáo dục (1-5)
- **EducationField:** Ngành giáo dục
- **EmployeeCount:** Số nhân viên trong tổ chức
- **EmployeeNumber**: Mã định danh của mỗi hồ sơ nhân viên
- **EnvironmentSatisfaction:** Mức độ hài lòng
- **Gender:** Giới tính
- **HourlyRate:** Mức lương theo giờ
- **JobInvolvement:** Mức độ tham gia công việc
- **JobLevel:** Cấp bậc
- **JobRole:** Vị trí trong công việc
- **JobSatisfaction:** mức độ hài lòng trong công việc
- **MaritalStatus:** Tình trạng hôn nhân (Disvorced, married, single
- **MonthlyIncome:** Tổng thu nhập theo tháng
- **MonthlyRate:** Tỉ lệ lương theo tháng
- **NumCompaniesWorked:** Số lượng cty đã làm việc
- **Over18:** NV có trên 18 không
- **OverTime:** Có tăng ca không
- **PercentSalaryHike:** Tỉ lệ tăng lương cho nhân viên
- **PerformanceRating:** Đánh giá hiệu quả công việc
- **RelationshipSatisfaction:** Sự hài lòng của nhân viên về các mqh trong công việc
- **StandardHours:** Giờ làm việc tiêu chuẩn
- **StockOptionLevel:** Mức độ chọn cổ phiếu
- **TotalWorkingYears:** Tổng số năm làm việc
- **TrainingTimesLastYear:** số lần training trong năm trước
- **WorkLifeBalance:** mức độ cân bằng công việc và cuộc sống
- **YearsAtCompany:** số năm làm ở cty hiện tại
- **YearsInCurrentRole**: số năm ở vị trí hiện tại
- **YearsSinceLastPromotion:** số năm kể từ lần thắng chức gần nhất
- **YearsWithCurrManager:** số năm với quản lý hiện tại

DATA CLEASNING_XỬ LÝ DỮ LIỆU ĐẦU VÀO

```
# check dữ liệu trùng lặp
df['EmployeeNumber'].duplicated().sum()
```

```
0
```

1. **Kiểm tra dữ liệu trùng lặp (duplicate)**
   -> Không tìm thấy

```
[69] #check dữ liệu null
     df.isna().sum()
```

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

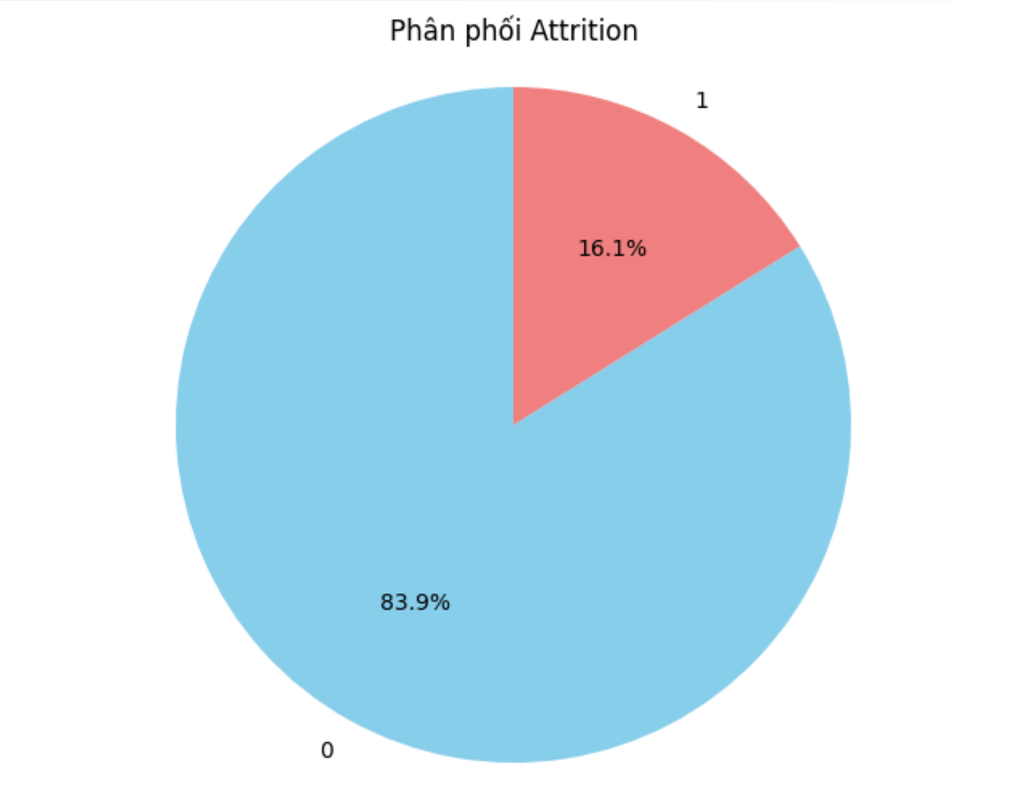
2. Kiểm tra dữ liệu null
-> Không tìm thấy

```
#Loại bỏ cột không cần thiết
df.drop(columns = ['EmployeeCount','StandardHours','EmployeeNumber','Over18'],inplace = True)
```

3. Loại bỏ các cột không cần thiết

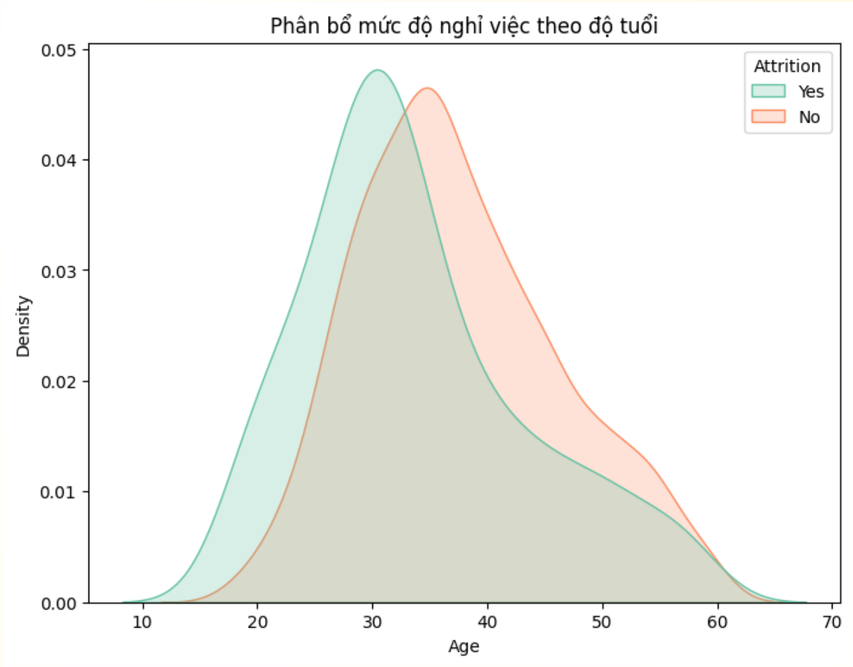
DATA VISUALIZATION_TRỰC QUAN HÓA DỮ LIỆU

1. Nhóm các yếu tố cá nhân ảnh hưởng đến sự nghỉ việc



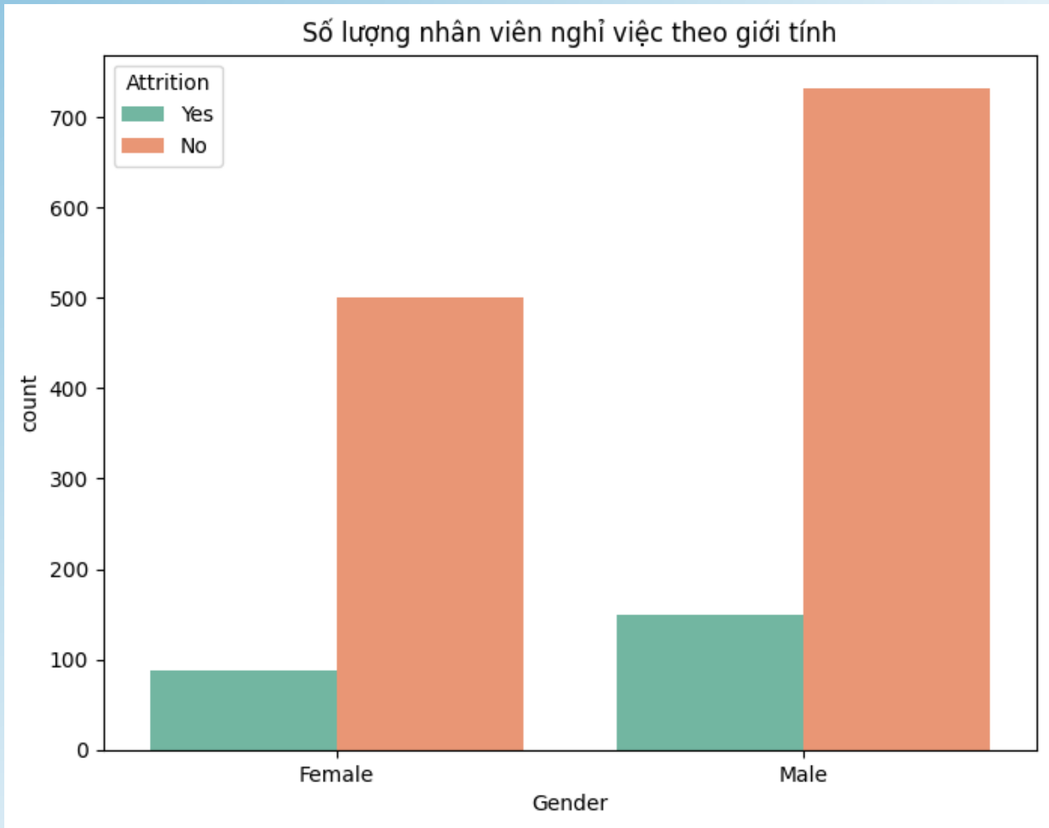
Biểu đồ thể hiện tỉ lệ nghỉ việc của nhân viên: Tỉ lệ nghỉ việc chiếm 16.12%.

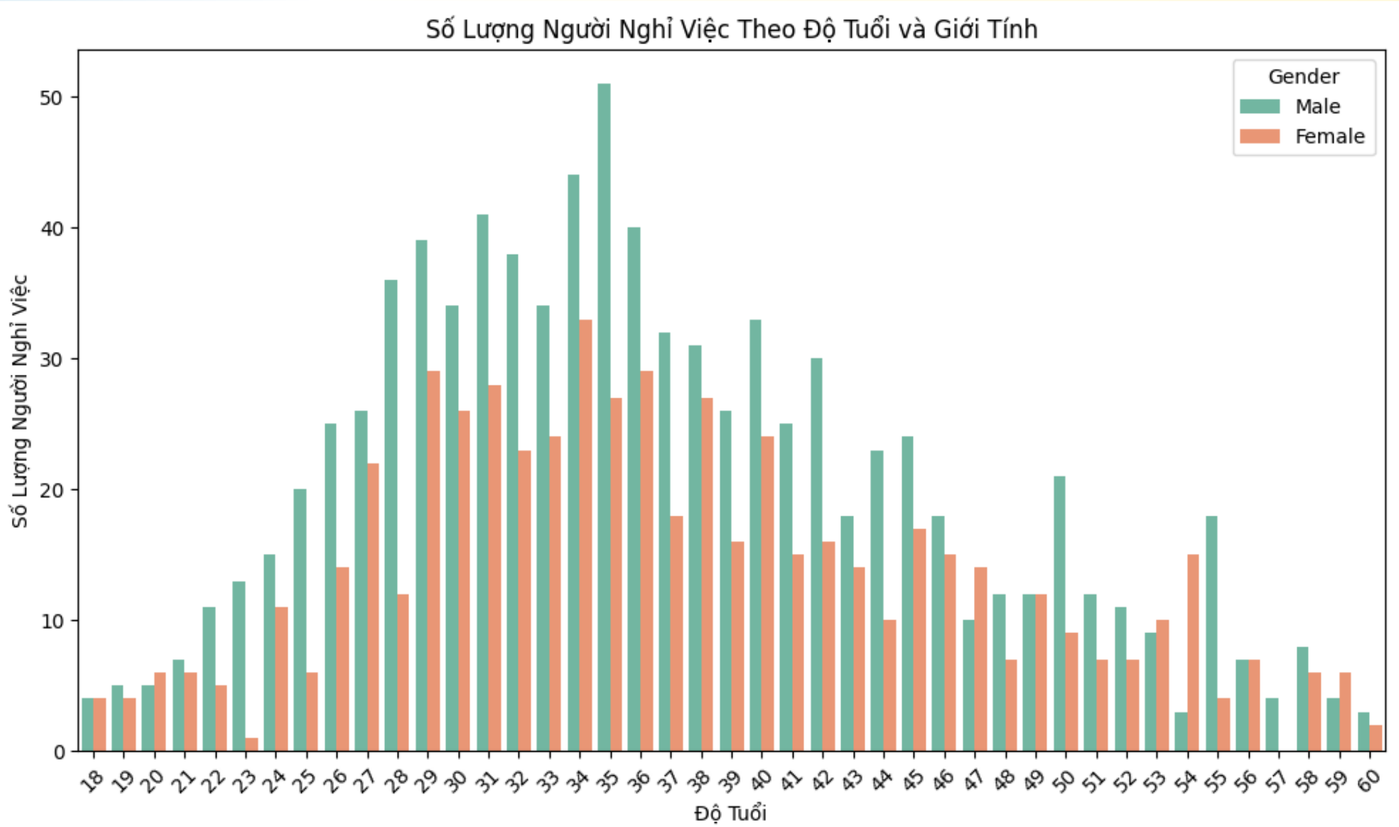
```
[106]  1 df['Age'].describe()

count    1470.000000
mean       36.923810
std         9.135373
min        18.000000
25%        30.000000
50%        36.000000
75%        43.000000
max        60.000000
Name: Age, dtype: float64
```



Nhân viên có độ tuổi từ 20-40 sẽ có xu hướng nghỉ việc nhiều hơn các độ tuổi còn lại

1. Nhóm các yếu tố cá nhân ảnh hưởng đến sự nghỉ việc

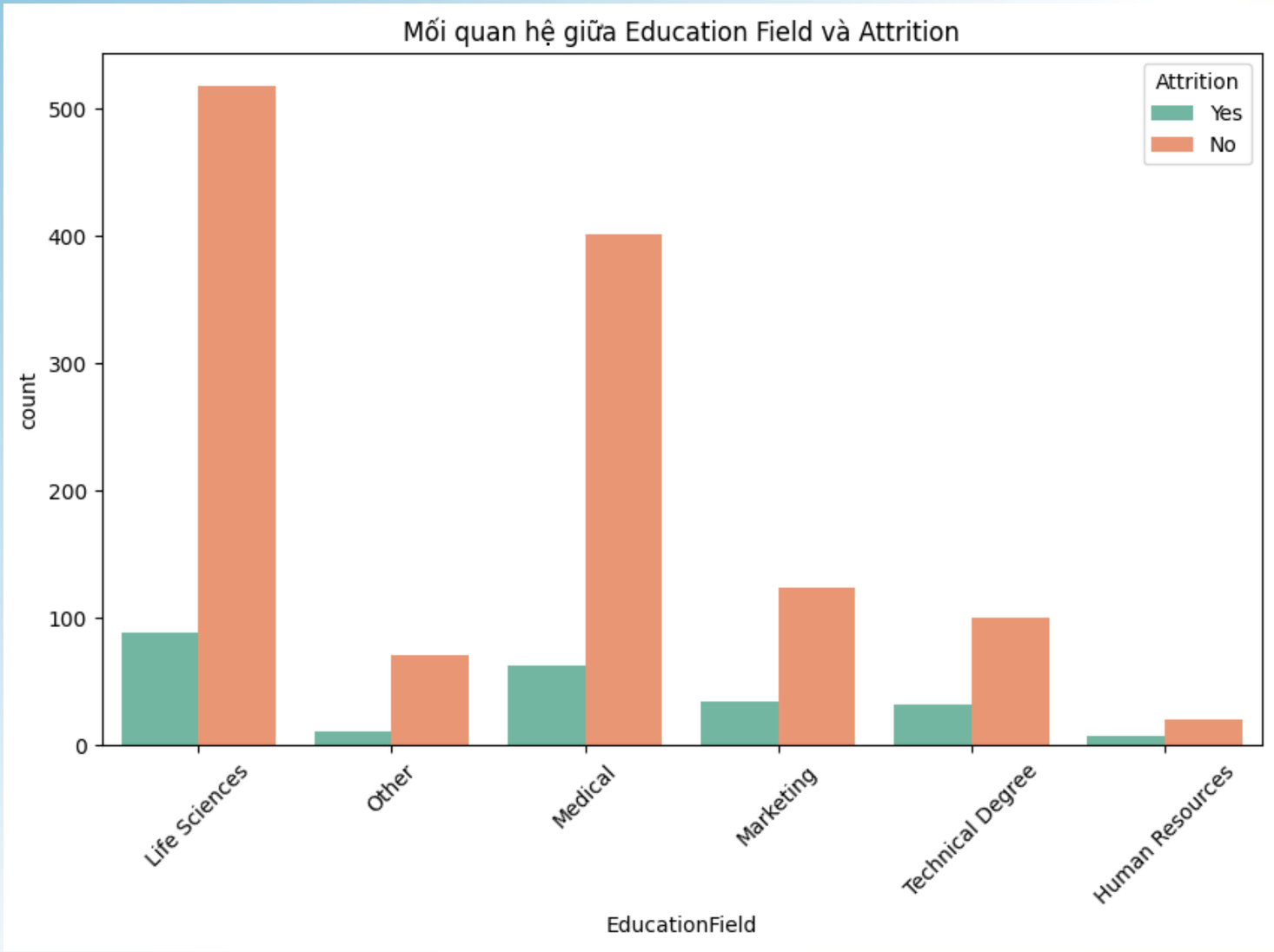




Biểu đồ thể hiện nam giới có xu hướng nghỉ việc nhiều hơn nữ giới

- Nam giới có độ tuổi từ 25 -40 có xu hướng nghỉ việc nhiều hơn so với các độ tuổi còn lại
- Nữ giới có độ tuổi từ 24 -38 có xu hướng nghỉ việc nhiều hơn so với các độ tuổi còn lại

1. Nhóm các yếu tố cá nhân ảnh hưởng đến sự nghỉ việc



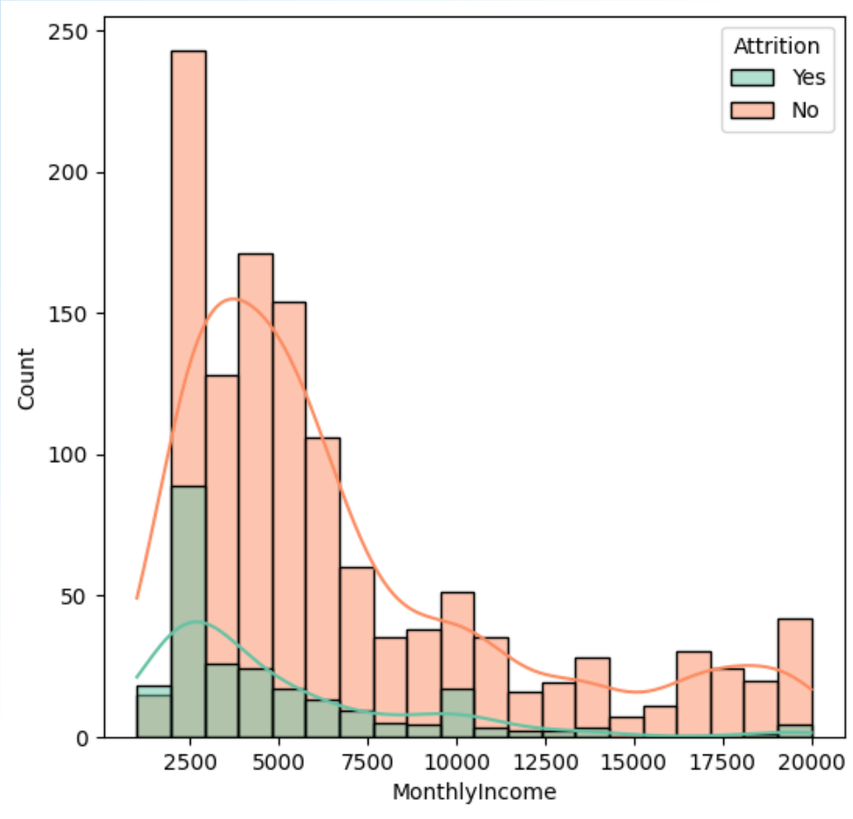
- Ngành học Life Sciences, Medical có số lượng nghỉ việc nhiều hơn những ngành còn lại

2. Nhóm các yếu tố công việc ảnh hưởng đến sự nghỉ việc

```
1 df['MonthlyIncome'].describe()
```

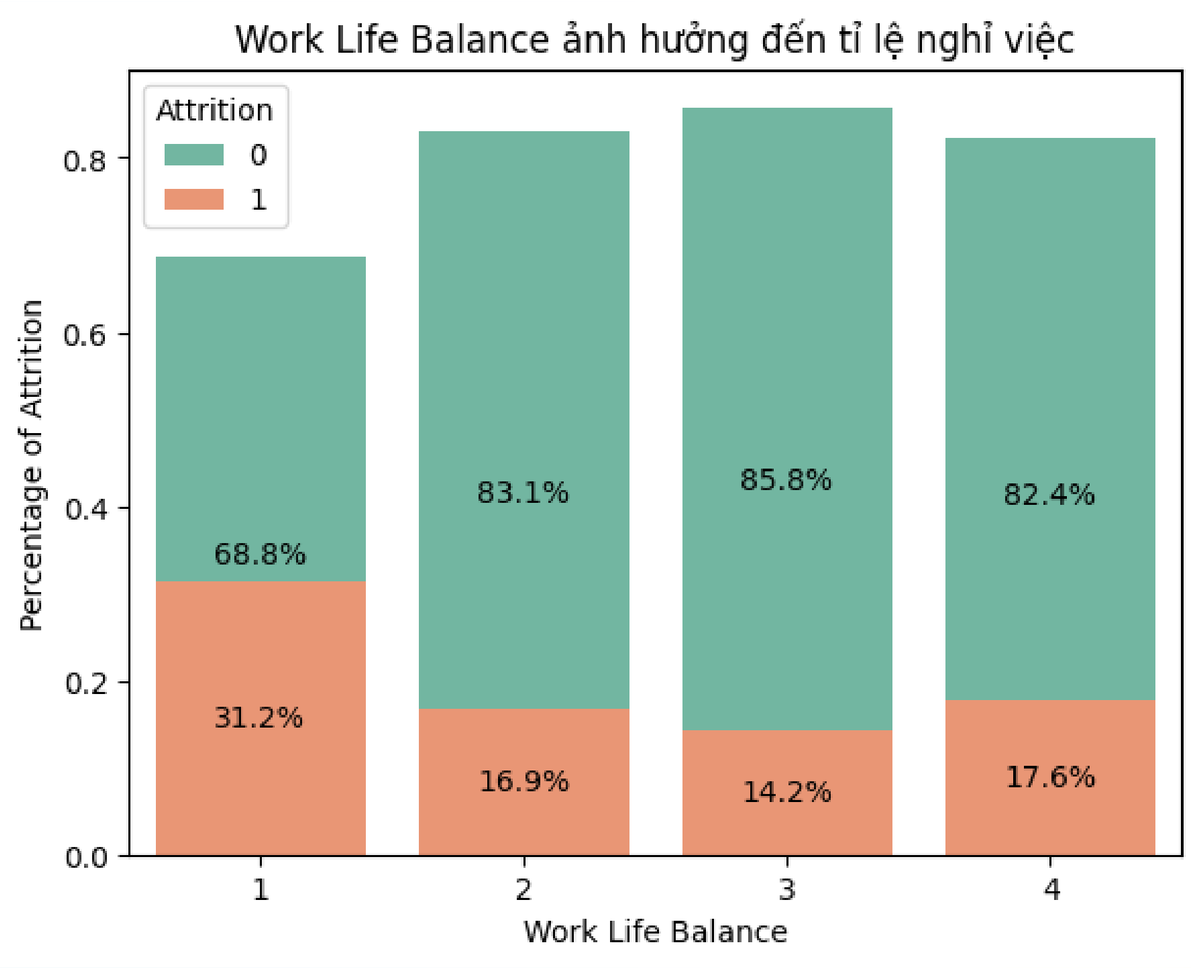
```
count     1470.000000
mean      6502.931293
std       4707.956783
min       1009.000000
25%       2911.000000
50%       4919.000000
75%       8379.000000
max      19999.000000
Name: MonthlyIncome, dtype: float64
```

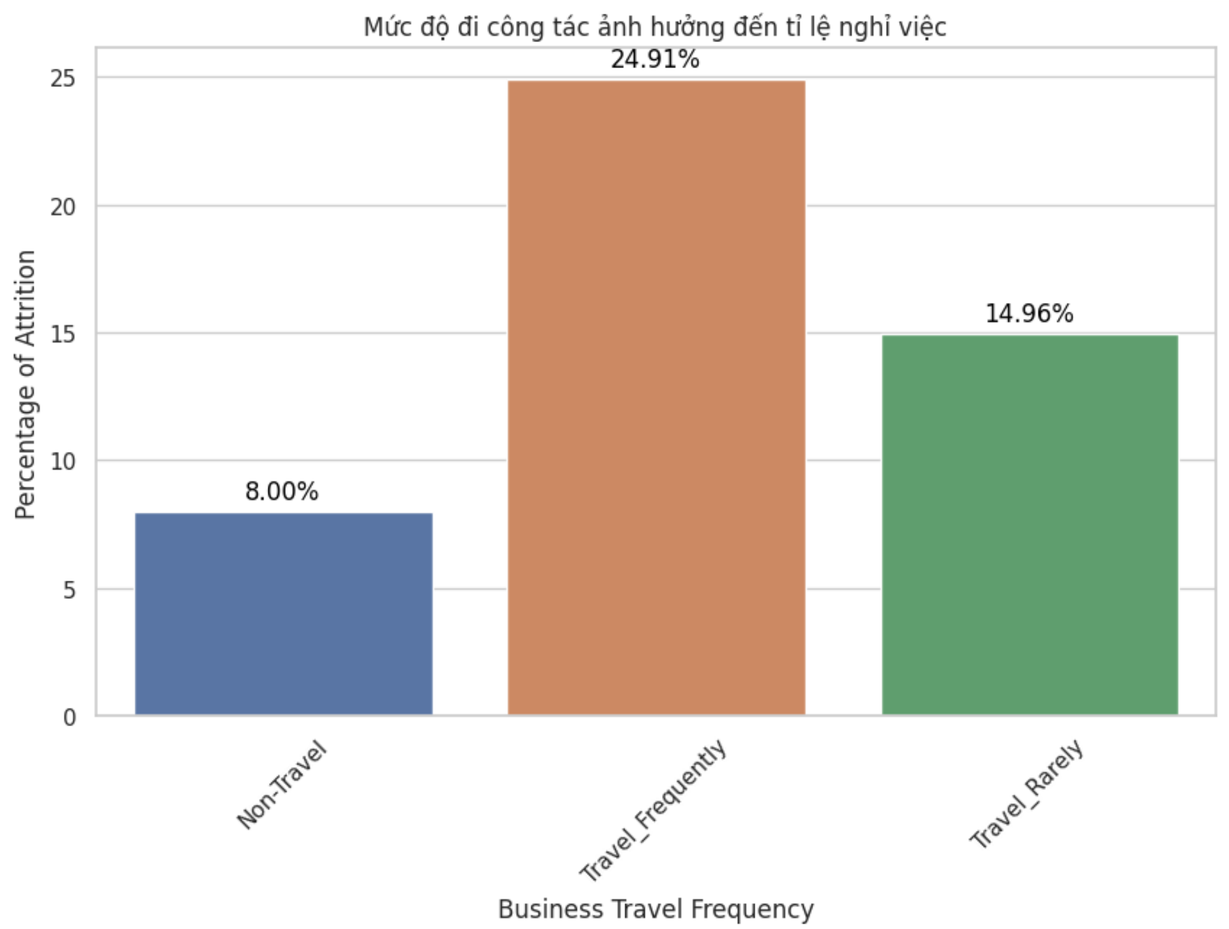
- Thu nhập trung bình 6502, không có nhân viên nào hưởng lương trên 20.000



- Thu nhập càng cao tỉ lệ nghỉ việc càng giảm

2. Nhóm các yếu tố công việc ảnh hưởng đến sự nghỉ việc





Mức độ Work Life Balance càng thấp thì tỉ lệ nghỉ việc càng cao

Mức độ đi công tác càng nhiều thì tỉ lệ nghỉ việc càng cao

PREPROCESSING DATA_TIỀN XỬ LÝ DỮ LIỆU

```
#Mã hóa dữ liệu
label_encoder = LabelEncoder()
df['Attrition'] = label_encoder.fit_transform(df['Attrition'])
df['BusinessTravel'] = label_encoder.fit_transform(df['BusinessTravel'])
df['Department'] = label_encoder.fit_transform(df['Department'])
df['EducationField'] = label_encoder.fit_transform(df['EducationField'])
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['JobRole'] = label_encoder.fit_transform(df['JobRole'])
df['MaritalStatus'] = label_encoder.fit_transform(df['MaritalStatus'])
df['OverTime'] = label_encoder.fit_transform(df['OverTime'])
df
```

Có một số trường đang ở dạng “object” -> tiến hành mã hóa để đảm bảo dữ liệu được đồng nhất về dạng số hóa

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceRating	RelationshipSatisfaction	StockOp
0	41	1	2	1102	2	1	2	1	2	0	...	3	1	
1	49	0	1	279	1	8	1	1	3	1	...	4	4	
2	37	1	2	1373	1	2	2	4	4	1	...	3	2	
3	33	0	1	1392	1	3	4	1	4	0	...	3	3	
4	27	0	2	591	1	2	1	3	1	1	...	3	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1465	36	0	1	884	1	23	2	3	3	1	...	3	3	
1466	39	0	2	613	1	6	1	3	4	1	...	3	1	
1467	27	0	2	155	1	4	3	1	2	1	...	4	2	
1468	49	0	1	1023	2	2	3	3	4	1	...	3	4	
1469	34	0	2	628	1	8	3	3	2	1	...	3	1	

1470 rows × 31 columns

PREPROCESSING DATA_TIỀN XỬ LÝ DỮ LIỆU

- Vẽ Heatmap để check nhanh tính tương quan của các biến. Do dữ liệu sau khi loại bỏ các cột cần thiết vẫn còn 31 trường -> Chỉ chọn ra các biến có tương quan nhiều với nhau để đưa vào mô hình đảm bảo quá trình xây dựng mô hình được tối ưu
- Tính lại tương quan và chọn các biến có tương quan với nhau (ngưỡng là 0.1) -> chọn được các biến

```
correlation_matrix = df.corr()
target_variable = 'Attrition'
correlations_with_target = abs(correlation_matrix[target_variable]).sort_values(ascending=False)
threshold = 0.1
selected_features = correlations_with_target[correlations_with_target > threshold].index.tolist()
selected_features
```

```
['Attrition',
 'OverTime',
 'TotalWorkingYears',
 'JobLevel',
 'MaritalStatus',
 'YearsInCurrentRole',
 'MonthlyIncome',
 'Age',
 'YearsWithCurrManager',
 'StockOptionLevel',
 'YearsAtCompany',
 'JobInvolvement',
 'JobSatisfaction',
 'EnvironmentSatisfaction']
```

PREPROCESSING DATA_TIỀN XỬ LÝ DỮ LIỆU

- Chọn X, y

```
#chọn X, y
y=df['Attrition']
X=df[selected_features].drop(columns=['Attrition'])
```

- Chuẩn hóa dữ liệu

```
# chuẩn hóa dữ liệu
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

```

- Chia tập train tập test

```
# Chia thành tập train và tập test
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42)
```

- Xử lý mất cân bằng dữ liệu bằng SMOTE

```
# Xử lý mất cân bằng dữ liệu bằng SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_scaled, y)
```

DATA MODELING & EVALUATION_XÂY DỰNG MÔ HÌNH VÀ ĐÁNH GIÁ

1. Random Forest

```python
# Xây dựng mô hình Random Forest Classifier
model1 = RandomForestClassifier(random_state=42)
# Huấn luyện mô hình trên tập train
model1.fit(X_train, y_train)
# Dự đoán trên tập test
y_pred = model1.predict(X_test)
# Đánh giá mô hình
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy: 0.8967611336032388
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.90      0.90       250
           1       0.90      0.89      0.89       244

    accuracy                           0.90       494
   macro avg       0.90      0.90      0.90       494
weighted avg       0.90      0.90      0.90       494
```

2. Logistic Regression

```python
# Xây dựng mô hình Logistic Regression
model2 = LogisticRegression(random_state=42)
# Huấn luyện mô hình trên tập huấn luyện
model2.fit(X_train, y_train)
# Dự đoán trên tập kiểm tra
y_pred = model2.predict(X_test)
# Đánh giá mô hình bằng accuracy
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy: 0.728744939271255
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.71      0.73       250
           1       0.71      0.75      0.73       244

    accuracy                           0.73       494
   macro avg       0.73      0.73      0.73       494
weighted avg       0.73      0.73      0.73       494
```

3. KNN

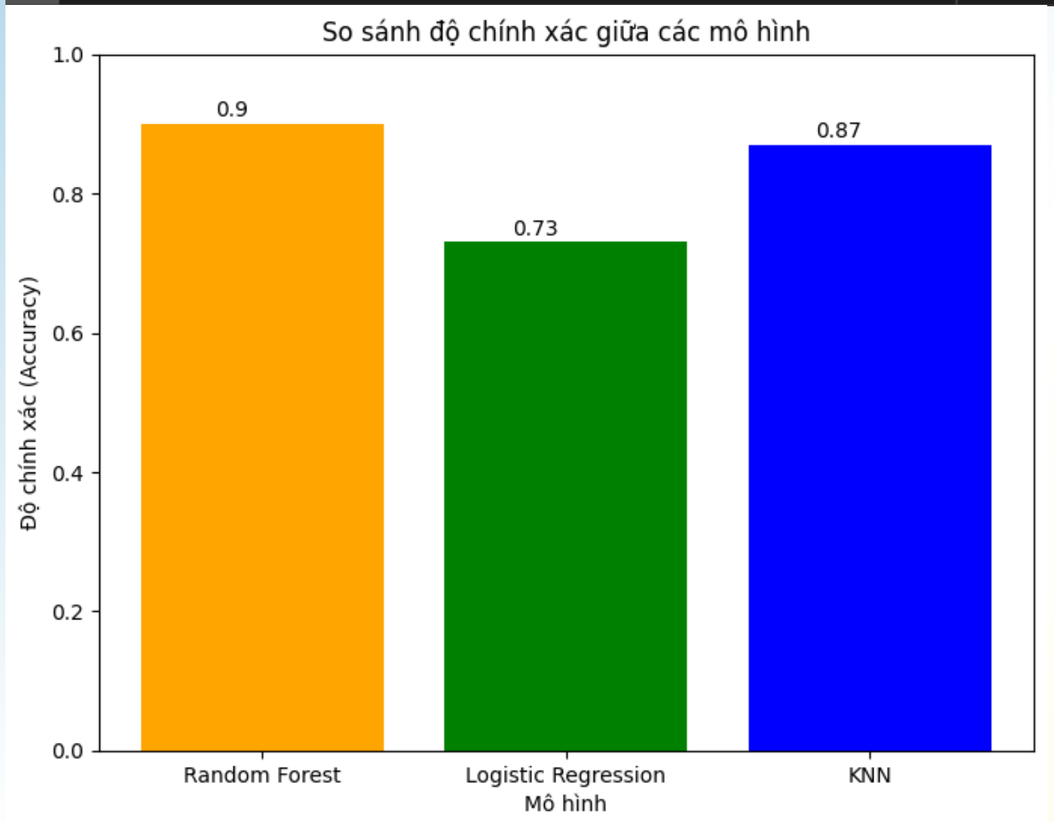
```python
# Xây dựng mô hình KNN
knn_model = KNeighborsClassifier(n_neighbors=4)
# Huấn luyện mô hình KNN trên tập train
knn_model.fit(X_train, y_train)
# Dự đoán trên tập test bằng mô hình KNN
y_pred_knn = knn_model.predict(X_test)
# Đánh giá mô hình KNN
accuracy_knn = accuracy_score(y_test, y_pred_knn)
report_knn = classification_report(y_test, y_pred_knn)
print("Accuracy (KNN):", accuracy_knn)
print("Classification Report (KNN):\n", report_knn)
```

```
Accuracy (KNN): 0.868421052631579
Classification Report (KNN):
              precision    recall  f1-score   support

           0       0.95      0.78      0.86       250
           1       0.81      0.96      0.88       244

    accuracy                           0.87       494
   macro avg       0.88      0.87      0.87       494
weighted avg       0.88      0.87      0.87       494
```

DATA MODELING & EVALUATION_XÂY DỰNG MÔ HÌNH VÀ ĐÁNH GIÁ

So sánh các mô hình

```python
# Kết quả từ ba mô hình
models = ['Random Forest', 'Logistic Regression', 'KNN']
accuracy_scores = [0.9, 0.73, 0.87]
# Vẽ biểu đồ bar chart
plt.figure(figsize=(8, 6))
bars=plt.bar(models, accuracy_scores, color=['orange', 'green', 'blue'])
for bar, score in zip(bars, accuracy_scores):
    plt.text(bar.get_x() + bar.get_width() / 2 - 0.1, score + 0.01, str(round(score, 2)), ha='center', color='black')
plt.xlabel('Mô hình')
plt.ylabel('Độ chính xác (Accuracy)')
plt.title('So sánh độ chính xác giữa các mô hình')
plt.ylim(0, 1)
plt.show()
```



- Mô hình Random Forest có độ chính xác (accuracy) cao nhất trong ba mô hình, với giá trị là 0.90.
- Mô hình KNN cũng có độ chính xác tốt với giá trị là 0.87.
- Mô hình Logistic Regression có độ chính xác thấp nhất trong ba mô hình với giá trị là 0.73.

CONCLUSION_ KẾT LUẬN

- Attrition có thể ảnh hưởng bởi 2 yếu tố là cá nhân (Tuổi, ngành học, tình trạng hôn nhân) và công việc (work-life balance, lương) -> Công ty nên chú ý tập trung vào các yếu tố này để có thể tuyển dụng và giữ các nhân viên chất lượng cho công ty
- Mô hình Random Forest có độ chính xác cao nhất và có thể được dùng để dự đoán attrition

THANKS FOR YOUR LISTENING