**ĐẠI HỌC UEH**

**TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ UEH**

**KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH**



**DỰ ÁN NHÓM**

**Bộ Môn : BIỂU DIỄN TRỰC QUAN DỮ LIỆU**

**ĐỀ TÀI: PHÂN TÍCH VÀ DỰ ĐOÁN HIỆU SUẤT CỦA NHÂN VIÊN TRONG DOANH NGHIỆP**

**GV Hướng Dẫn: Nguyễn An Tế**

**Mã lớp học phần** : **23C1INF50908202**

**Sinh viên - MSSV** : **Đặng Yến Nhi 31211027659**

**Tất Diệu Ngân 31211027652**

**Hồng Nguyên 31211023184**

**Nguyễn Hoàng Hà My 31201024506**

**Nguyễn Tân Niên 31211027661**

TP Hồ Chí Minh, ngày 26 tháng 11 năm 2023

# 

**Giới thiệu đề tài**

Hiệu suất của nhân viên đóng vai trò quan trọng trong sự phát triển và thành công của một doanh nghiệp. Nắm vững và cải thiện hiệu suất nhân viên có thể giúp doanh nghiệp tối ưu hóa tài nguyên, gia tăng lợi nhuận, và duy trì sự cạnh tranh trên thị trường. Đề tài này có tính ứng dụng rộng rãi trong lĩnh vực quản lý doanh nghiệp và tài nguyên con người, giúp doanh nghiệp nắm bắt được nguồn lực quý báu trong nhân viên và tạo ra môi trường làm việc hiệu quả hơn. Một mô hình được đào tạo có thể dự đoán hiệu suất của nhân viên dựa trên các yếu tố đầu vào. Điều này sẽ được sử dụng để cải thiện môi trường của nhân viên và đề xuất để cải thiện hiệu suất của nhân viên dựa trên những hiểu biết sâu sắc từ phân tích. Dữ liệu được cung cấp từ IABAC cho dự án này với nguồn được thu thập là IABAC™. Dữ liệu được dựa trên INX Future Inc., (gọi tắt là INX). Đây là một trong những nhà cung cấp giải pháp tự động hóa và phân tích dữ liệu hàng đầu với hơn 15 năm kinh doanh trên toàn cầu. INX liên tục được đánh giá là top 20 nhà tuyển dụng tốt nhất trong 5 năm qua. Dữ liệu không phải từ tổ chức có thật.

**MỤC LỤC**

[I. Tổng quan dữ liệu 4](#_o3p6mouz2e3s)

[1. Tìm hiểu bộ dữ liệu 4](#_gzew4r632bc5)

[2. Mục tiêu nghiên cứu 4](#_v895z5ym0awt)

[3. Giải thích biến 4](#_1amin4fz15ow)

[3.1. Biến đầu vào 4](#_g41ww9s6m7am)

[3.2. Biến mục tiêu 6](#_26iqjn8mhqmg)

[4. Phương pháp thực hiện 6](#_aaosfoe89nco)

[5. Phân tích khám phá dữ liệu 6](#_575366w3ghpb)

[6. Thống kê dữ liệu 6](#_vzy7dykj6kwf)

[II. Tiền xử lí dữ liệu 8](#_uizi4b3n3mfy)

[1. Kiểm tra giá trị rỗng 9](#_8tasnwohw5m4)

[2. Kiểm tra các giá trị bị trùng: 9](#_xds6h7oulahb)

[3. Kiểm tra và xử lý outliers: 10](#_elgytwopqnzl)

[3.1. Xử lý các outliers của biến TotalWorkExperienceInYears: 11](#_lmk41kphlj40)

[3.2. Xử lý các outliers của biến TrainingTimesLastYear 12](#_8ef581e2w698)

[3.3. Xử lý các outliers của biến ExprienceYearsAtThisCompany 13](#_sg2qo0kehtf5)

[3.4. Xử lý các outliers của biến ExperienceYearsInCurrentRole 14](#_nvzaonpt2tub)

[3.5. Xử lý các outliers của biến YearsSinceLastPromotion 14](#_pn4i6kg1wd1y)

[3.6. Xử lý các outliers của biến YearsWithCurrentManager 16](#_2rvjsyd8ay0p)

[3.7. Xử lý các outliers của biến NumcompaniesWorked 17](#_4qe5m8rf8a64)

[4. Kiểm tra mối tương quan giữa các biến của bộ dữ liệu 18](#_t6h8gag2cts8)

[III. Trực quan hóa dữ liệu 20](#_8ly309ow808m)

[1. Trực quan đơn biến 20](#_voy6m8hgenw6)

[1.1. Qualitative data 20](#_ebpwtug5533q)

[1.2. Quantitative data 25](#_r7scu3ykav2u)

[2. Trực quan hai biến 29](#_oi2vwkv32wik)

[2.1. Sử dụng kiểm định Chi-square để xem những cặp biến có tương quan với nhau: 29](#_sq4o8a5c0jkk)

[2.2. Xét tương quan giữa biến nominal categorical và biến target 30](#_zi8csfjv3ra8)

[a. Biểu diễn trực quan cho các biến nominal categorical 30](#_2eorj3vj8ytk)

[b. Kiểm định mối tương quan giữa các biến nominal categorical và biến target (PerformanceRating) 32](#_srfpxrqr9661)

[c. Trực quan 2 biến overtime và biến target sau khi đã xác định được 2 biến này không độc lập 33](#_nghtz32rtj6l)

[d. Trực quan hai biến EmpDepartment (phòng ban nhân viên) và PerformanceRating (hiệu suất làm việc) 33](#_2cfcv03qxks3)

[e. Trực quan 2 biến EmpJobRole (Vị trí của nhân viên) và PerformanceRating (đánh giá hiệu suất) 35](#_1760r8yei9k2)

[2.3. Biểu diễn trực quan mối liên hệ giữa các biến ordinal và biến target (PerformanceRating) 36](#_zezbay2kfexf)

[2.4. Trực quan biến numerical với biến target (PerfomanceRating) 40](#_p7t6p3xyz92j)

[IV. Xử lý và chuẩn hóa dữ liệu để nâng cao mô hình được xây dựng 48](#_ukcn1ncmk0wb)

[1. Chuyển các thuộc tính phân loại sang dữ liệu số 48](#_o78kw8sn5lp5)

[2. Sử dụng các thư viện cần thiết để chuẩn hóa và giảm chiều dữ liệu 52](#_5knd40afem1y)

[IV. Xây dựng mô hình dự đoán 56](#_qzhob2xohiv)

[1. Mô hình Logistic Regression: 56](#_tz8ue9a5ps9b)

[2. Mô hình Decision Tree 61](#_kg0gl4o5d3dr)

[3. Mô hình Random Forest 65](#_4bmhtun15vsj)

[4. Sử dụng phương pháp kiểm định chéo (cross-validation) kiểm định ba mô hình trên để chọn ra mô hình tối ưu nhất 69](#_aa0ixlsdw0x)

[V. Tổng kết 72](#_8ssrlug1bcs3)

[TÀI LIỆU THAM KHẢO 74](#_no9ga1ub1sd6)

# **Tổng quan dữ liệu**

## Tìm hiểu bộ dữ liệu

* Nhân viên đã cho bao gồm 1200 hàng. Các tính năng có trong dữ liệu là 28 cột. Hình dạng của tập dữ liệu là 1200x28. 28 đặc điểm được phân loại thành định lượng và định tính trong đó 11 đặc điểm là định lượng và 16 đặc điểm là định tính. EmpNumber là dữ liệu định danh (giá trị riêng biệt) không đóng vai trò là tính năng liên quan để xếp hạng hiệu suất.

## Mục tiêu nghiên cứu

* Một trong những mục tiêu quan trọng của dự án này là tìm ra những biến quan trọng ảnh hưởng đến PerformanceRating. Các biến quan trọng được dự đoán trên sự tương quan khi phân tích và trực quan hóa dữ liệu. Ngoài ra còn có những mục tiêu song song như sau: khám phá dữ liệu, trực quan hóa dữ liệu, làm sạch dữ liệu, xây dựng mô hình máy học và rút ra những nhận xét, đánh giá dựa trên dữ liệu đã tổng hợp, phân tích.

## Giải thích biến

### 3.1. Biến đầu vào

* Biến phân loại:
* EmpNumber: ID định danh nhân viên
* Gender: Giới tính
* EducationBackground: Ngành học
* MaritalStatus: Tình trạng hôn nhân
* EmpDepartment: Phòng ban làm việc
* EmpJobRole: Vị trí làm việc
* BusinessTravelFrequency: Tần suất đi công tác ( non/ rarely/ frequently)
* OverTime: Tăng ca (yes/no)
* Attrition: Nhân viên rời công ty vì lý do cá nhân hoặc động lực nghề nghiệp (yes/no)
* EmpEducationLevel: Trình độ học vấn của nhân viên
* 1: Below College
* 2: College
* 3: Bachelor
* 4: Master
* 5: Doctor
* EmpEnvironmentSatisfaction: Độ hài lòng của nhân viên với công ty
* 1: Low
* 2: Medium
* 3: High
* 4: Very High
* EmpJobInvolvement: Mức độ tham gia vào công việc hoặc dự án của nhân viên
* 1: Low
* 2: Medium
* 3: High
* 4: Very High
* EmpJobLevel: Độ khó hoặc cấp độ của công việc mà nhân viên tham gia
* 1: Cơ bản
* 2: Dễ
* 3: Trung bình
* 4: Khó
* 5: Rất khó
* EmpJobSatisfaction: Mức độ hài lòng của nhân viên với công việc
* 1: Low
* 2: Medium
* 3: High
* 4: Very High
* EmpRelationshipSatisfaction: Mức độ hài lòng của nhân viên với đồng nghiệp tại công ty
* 1: Low
* 2: Medium
* 3: High
* 4: Very High
* EmpWorkLifeBalance: Sự cân bằng giữa công việc và cuộc sống của nhân viên
* 1: Bad
* 2: Good
* 3: Better
* 4: Best
* Biến số:
* Age: Tuổi của nhân viên
* DistanceFromHome: Khoảng cách (km) từ nhà đến công ty của nhân viên
* EmpHourlyRate: Mức lương theo giờ
* NumCompaniesWorked: Số công ty mà nhân viên đã làm việc
* EmpLastSalaryHikePercent: Hệ số tăng lương (%) vào năm ngoái
* TotalWorkExperienceInYears: Tổng số năm kinh nghiệm làm việc của nhân viên
* TrainingTimesLastYear: Tổng số thời gian được đào tạo
* ExperienceYearsAtThisCompany: Số năm kinh nghiệm tại công ty hiện tại
* ExperienceYearsInCurrentRole: Tổng số năm kinh nghiệm ở vị trí hiện tại
* YearsSinceLastPromotion: Số năm từ lần thăng chức cuối cùng
* YearsWithCurrManager: Số năm làm việc với quản lý hiện tại

## 

### 3.2. Biến mục tiêu

* PerformanceRating: Đây là biến target cho biết tổng đánh giá hiệu suất của nhân viên về công ty
* 1: Low
* 2: Good
* 3: Excellent
* 4: Outstanding

## Phương pháp thực hiện

* Để đánh giá chính xác những nhân tố có ảnh hưởng lớn đến hiệu suất của nhân viên trong một công ty, chúng ta cần tìm mối quan hệ y = f(x) và mục tiêu là tìm hàm f tốt nhất để khi có một giá trị x mới chúng ta có thể dự đoán được y
* Dữ liệu được phân tích bằng cách mô tả các tính năng có trong dữ liệu. Các tính năng đóng vai trò lớn trong việc phân tích. Các đặc điểm cho biết mối quan hệ giữa các biến phụ thuộc và biến độc lập.
* Phân tích đơn biến và hai biến:
* Phân tích đơn biến: Trong phân tích đơn biến, nhận được các nhãn duy nhất của các đặc điểm phân loại, cũng như phạm vi và mật độ của các số
* Phân tích hai biến: Trong phân tích hai biến, kiểm tra mối quan hệ đặc trưng với biến mục tiêu

# 

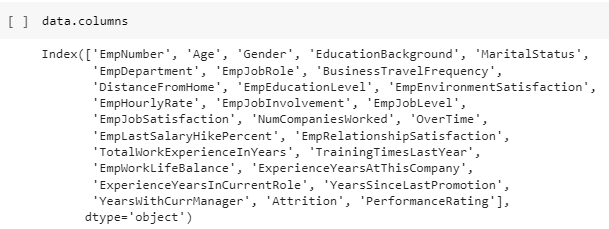
## 5. Phân tích khám phá dữ liệu

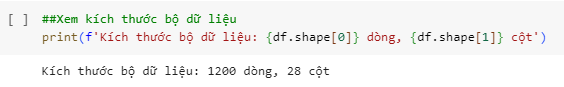
## Phân tích khám phá dữ liệu giúp chúng ta có cái nhìn tổng quát về dữ liệu. EDA giúp quan sát độ phức tạp của bài toán và xác định những bước cần làm.

* Có ba lệnh được sử dụng để khám phá dữ liệu:
* Data.head(): xem các cột và một vài hàng mẫu của dữ liệu
* Data.info(): cung cấp thông tin tóm tắt của dữ liệu
* Data.Describe(): cung cấp các chi tiết thống kê mô tả của dữ liệu

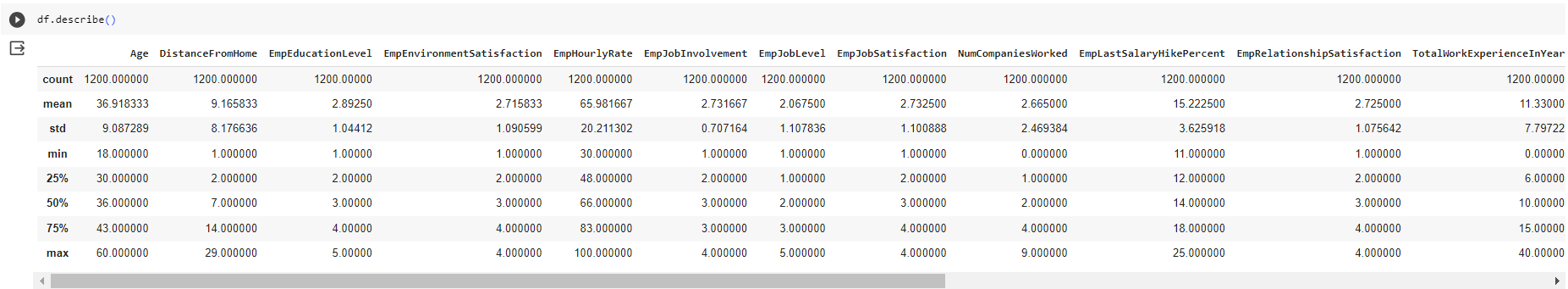
## 6. Thống kê dữ liệu

* Sử dụng data.columns() và data.shape() để quan sát các biến và kích thước bộ dữ liệu

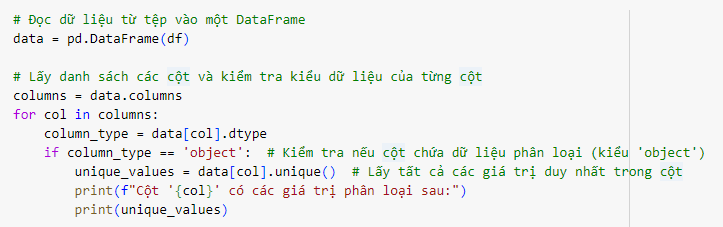


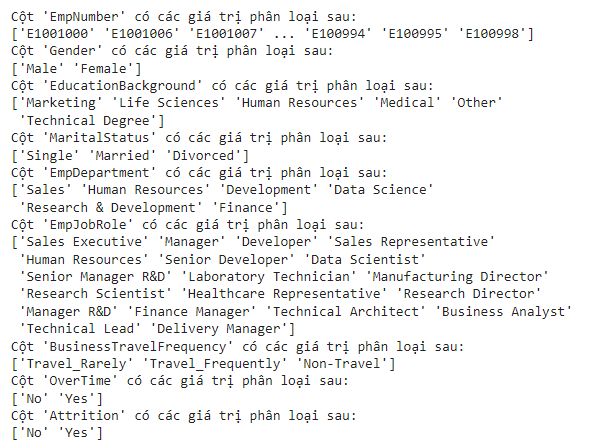


* Thống kê mô tả

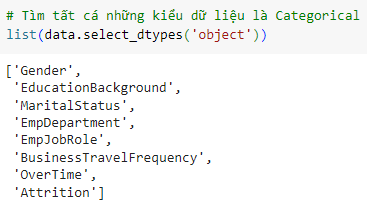


* Đọc dữ liệu vào DataFrame và lấy danh sách các cột để kiểm tra dữ liệu của từng cột





* Hiển thị tất cả những thuộc tính có kiểu dữ liệu là Categorical



* Quan sát số thuộc tính của bộ dữ liệu



* Loại bỏ thuộc tính không cần thiết cho bộ dữ liệu:



* **Nhận xét:**
* Bộ dữ liệu có kích thước là: 1200 dòng và 28 cột.
* Bộ dữ liệu bao gồm 28 thuộc tính trong đó có 17 biến phân loại và 11 biến số.
* Đối với bộ dữ liệu, cột EmpNumber để định danh phân biệt những nhân viên trong công ty với nhau nên ta có thể cắt bỏ cột này cho những phân tích về sau.
* Bộ dữ liệu không có giá trị rỗng nên không cần các bước xử lý rỗng tiếp theo.

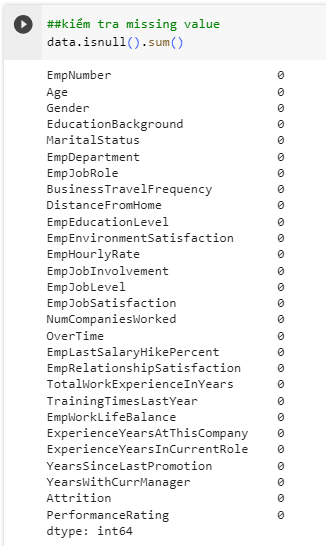
# **II. Tiền xử lí dữ liệu**

Tiền xử lí dữ liệu trước khi trực quan hóa để có thể hiển thị thông tin một cách rõ ràng và dễ hiểu. Mục tiêu của việc tiền xử lí dữ liệu trước khi biểu diễn trực quan bao gồm:

* Loại bỏ các dữ liệu không chính xác, thiếu xót, trùng lắp hoặc dữ liệu nhiễu.
* Đưa dữ liệu về một phạm vi chuẩn để dễ dàng so sánh và phân tích.
* Lọc bớt dữ liệu nhiễu để tăng tính rõ ràng, tin cậy của thông tin được hiển thị.

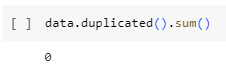
## Kiểm tra giá trị rỗng

* Kiểm tra các giá trị missing value bằng data.isnull().sum():
* data.isnull() sẽ trả về một DataFrame mới có cùng kích thước như data, trong đó mỗi ô có giá trị True nếu ô đó chứa giá trị thiếu và False nếu không.
* sum() được gọi trên DataFrame mới này để tính tổng số lượng giá trị True (hoặc số lượng giá trị thiếu) theo từng cột.



## Kiểm tra các giá trị bị trùng:

* Kiểm tra giá trị dữ liệu bị trùng bằng data.duplicated().sum():
* data.duplicated() kiểm tra các dòng trong DataFrame data và trả về một Series boolean, trong đó mỗi phần tử cho biết dòng tương ứng có bị trùng lặp (duplicate) không.
* sum(): Phương thức này tính tổng các giá trị trong Series. Vì Series boolean được trả về từ data.duplicated() có giá trị True tại các vị trí có dòng trùng lặp và False tại các vị trí không trùng lặp, nên việc gọi .sum() sẽ đếm số lượng các giá trị True, tức là số lượng các dòng có bản sao/trùng lặp.

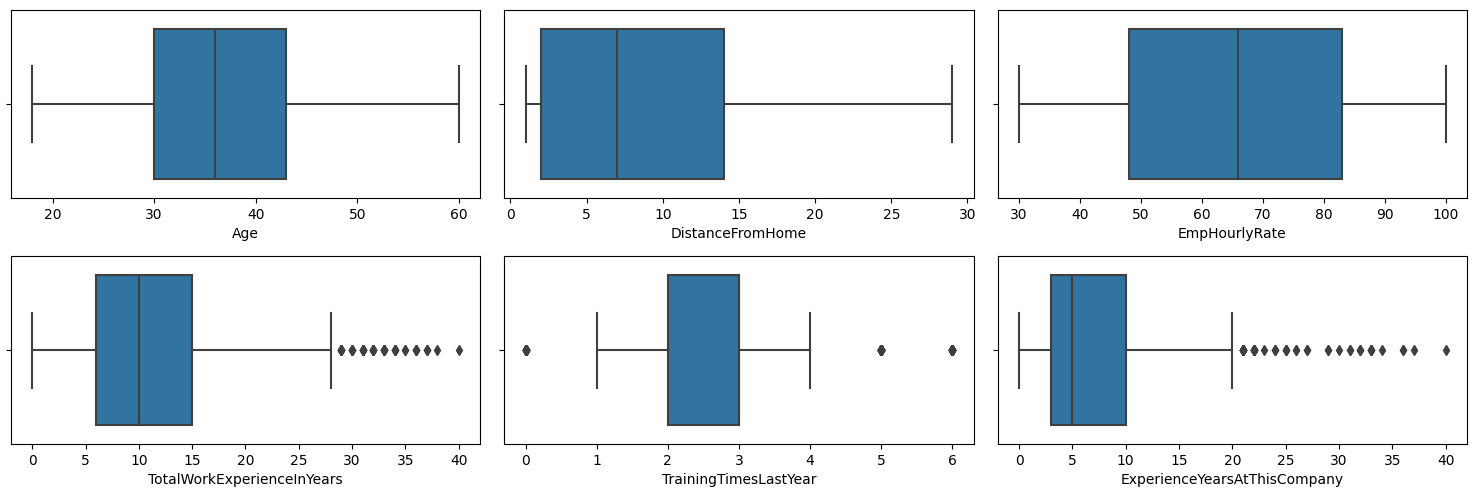


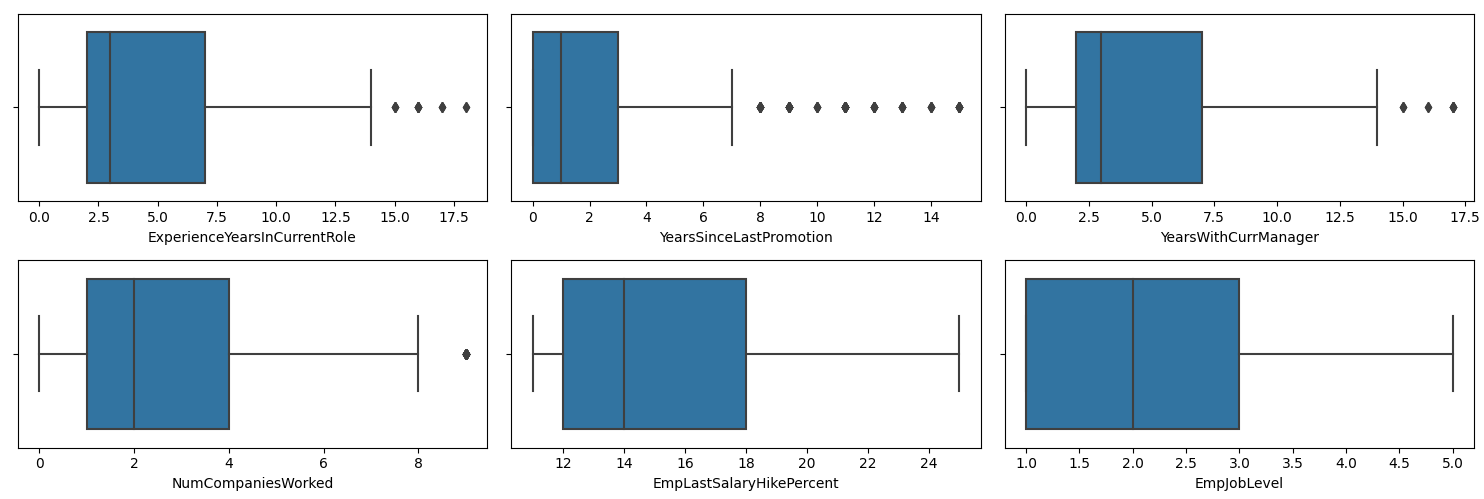
## Kiểm tra và xử lý outliers:

* Biểu diễn trực quan dữ liệu bằng biểu đồ boxplot để quan sát các giá trị ngoại lai của các biến số:



* Sau khi tạo dataframe cho các biến ta dùng khởi tạo biểu đồ và được kết quả như sau:





* Sau khi trực quan bằng biểu đồ, ta quan sát được có nhiều giá trị ngoại lai tại các biến ‘TotalWorkExperienceInYears’, ‘ExprienceYearsAtThisCompany’, ‘ExperienceYearsInCurrentRole’, ‘YearsSinceLastPromotion’, ‘YearsWithCurrentManager’, ‘NumcompaniesWorked’ và ‘TrainingTimesLastYear’.
* Sau khi quan sát dữ liệu nhiễu ta dùng phương pháp IQR để xử lí outiers:
* Tính toán tứ phân vị thứ nhất (Q1) và tứ phân vị thứ ba (Q3) của dữ liệu.
* Tính toán IQR bằng cách lấy hiệu của Q3 và Q1:
* Xác định ngưỡng cho outliers bằng cách sử dụng IQR bằng cách xác định các điểm dữ liệu nằm ngoài phạm vi và được coi là outliers.
* Xử lí các outliers dưới giới hạn dưới minimum limit và có giá trị lớn hơn giới hạn trên maximum limit sẽ được thay thế bằng giá trị trung vị trong cột đó.
* Sau khi xử lý outliers, quan sát lại dữ liệu để đảm bảo rằng các thay đổi không làm mất đi thông tin quan trọng và không làm biến đổi đáng kể tính chất của dữ liệu.

### 3.1. Xử lý các outliers của biến TotalWorkExperienceInYears:



* Giá trị Q1 (tứ phân vị thứ nhất) là 6.0 và Q3 (tứ phân vị thứ ba) là 15.0.
* Giá trị IQR (Interquartile Range) được tính là 9.0.
* Giới hạn dưới (Minimum limit) được tính là -7.5 và giới hạn trên (Maximum limit) là 28.5.
* Số lượng outliers được xác định là 51.
* Nhận xét:
* Dữ liệu có phân phối rộng, với phần lớn giá trị tập trung từ Q1 đến Q3, và khoảng cách lớn giữa Q3 và giá trị lớn nhất.
* Số lượng outliers là khá lớn, chiếm một phần đáng kể của dữ liệu nằm ngoài phạm vi giới hạn được xác định bởi phương pháp IQR.
* Giới hạn trên khá cao, cho thấy có một số lượng lớn các giá trị nằm xa hơn so với Q3.
* Sau khi thay thế các giá trị outliers bằng giá trị trung vị thì cột TotalWorkExperienceInYears đã không còn giá trị ngoại lại nào.

### 3.2. Xử lý các outliers của biến TrainingTimesLastYear



* Giá trị Q1 (tứ phân vị thứ nhất) là 2.0 và Q3 (tứ phân vị thứ ba) là 3.0.
* Giá trị IQR (Interquartile Range) được tính là 1.0.
* Giới hạn dưới (Minimum limit) được tính là 0.5 và giới hạn trên (Maximum limit) là 4.5.
* Số lượng outliers được xác định là 188.
* Nhận xét:
* Dữ liệu có phân phối tập trung chủ yếu trong một phạm vi hẹp từ Q1 đến Q3, với khoảng cách tương đối nhỏ giữa giá trị Q1 và Q3 (IQR = 1.0).
* Số lượng outliers được xác định là khá lớn (188), chiếm một tỷ lệ đáng kể so với tổng số lượng dữ liệu.
* Sau khi thay thế các giá trị outliers bằng giá trị trung vị thì cột TrainingTimesLastYear đã không còn giá trị ngoại lại nào.

### 3.3. Xử lý các outliers của biến ExprienceYearsAtThisCompany



* Giá trị Q1 (tứ phân vị thứ nhất) là 3.0 và Q3 (tứ phân vị thứ ba) là 10.0.
* Giá trị IQR (Interquartile Range) được tính là 7.0.
* Giới hạn dưới (Minimum limit) được tính là -7.5 và giới hạn trên (Maximum limit) là 20.5.
* Số lượng outliers được xác định là 56.
* Nhận xét:
* Dữ liệu có phân phối rộng, với giá trị Q1 là 3.0 và Q3 là 10.0, cho thấy một phạm vi lớn giữa 25% đến 75% của dữ liệu.
* Giá trị IQR là 7.0, chỉ ra một khoảng cách đáng kể giữa Q1 và Q3.
* Số lượng outliers được xác định là 56, tương đối nhỏ so với tổng số lượng dữ liệu.
* Sau khi thay thế các giá trị outliers bằng giá trị trung vị thì cột ExprienceYearsAtThisCompany đã không còn giá trị ngoại lại nào.

### 3.4. Xử lý các outliers của biến ExperienceYearsInCurrentRole



* Giá trị Q1 (tứ phân vị thứ nhất) là 2.0 và Q3 (tứ phân vị thứ ba) là 7.0.
* Giá trị IQR (Interquartile Range) được tính là 5.0.
* Giới hạn dưới (Minimum limit) được tính là -5.5 và giới hạn trên (Maximum limit) là 14.5.
* Số lượng outliers được xác định là 16.
* Nhận xét:
* Dữ liệu có phân phối từ Q1 (2.0) đến Q3 (7.0), với một khoảng tương đối hẹp giữa 25% và 75% của dữ liệu.
* Số lượng outliers được xác định là 16, tỷ lệ này so với tổng số lượng dữ liệu không lớn.
* Số lượng outliers ít có thể cho thấy dữ liệu có tính đáng tin cậy hơn và phân phối của nó có thể không bị ảnh hưởng nhiều bởi các giá trị ngoại lệ.
* Sau khi thay thế các giá trị outliers bằng giá trị trung vị thì cột ExperienceYearsInCurrentRole đã không còn giá trị ngoại lại nào.

### 3.5. Xử lý các outliers của biến YearsSinceLastPromotion



* Giá trị Q1 (tứ phân vị thứ nhất) là 0.0 và Q3 (tứ phân vị thứ ba) là 3.0.
* Giá trị IQR (Interquartile Range) được tính là 3.0.
* Giới hạn dưới (Minimum limit) được tính là -4.5 và giới hạn trên (Maximum limit) là 7.5.
* Số lượng outliers được xác định là 88.
* Nhận xét:
* Giá trị IQR là 3.0, chỉ ra một khoảng cách không quá lớn giữa Q1 và Q3.
* Số lượng outliers được xác định là 88, chiếm tỷ lệ lớn so với tổng số lượng dữ liệu.
* Sau khi thay thế các giá trị outliers bằng giá trị trung vị thì cột YearsSinceLastPromotion đã không còn giá trị ngoại lại nào.

### 3.6. Xử lý các outliers của biến YearsWithCurrentManager



* Giá trị Q1 (tứ phân vị thứ nhất) là 2.0 và Q3 (tứ phân vị thứ ba) là 7.0.
* Giá trị IQR (Interquartile Range) được tính là 5.0.
* Giới hạn dưới (Minimum limit) được tính là -5.5 và giới hạn trên (Maximum limit) là 14.5.
* Số lượng outliers được xác định là 11.
* Nhận xét:
* Giá trị IQR là 5.0, chỉ ra một khoảng cách đáng kể giữa Q1 và Q3.
* Số lượng outliers được xác định là 11, đây là một số lượng ít hơn so với tổng số lượng dữ liệu.
* Số lượng outliers ít có thể cho thấy dữ liệu có thể đáng tin cậy hơn và phân phối của nó không bị ảnh hưởng nhiều bởi các giá trị ngoại lệ.
* Sau khi thay thế các giá trị outliers bằng giá trị trung vị thì cột YearsWithCurrentManager đã không còn giá trị ngoại lại nào.

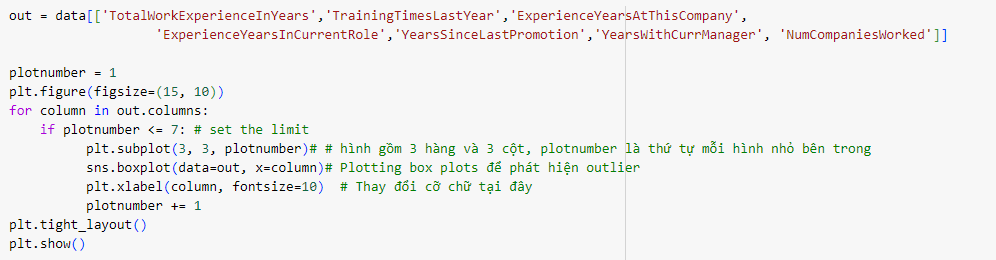
### 3.7. Xử lý các outliers của biến NumcompaniesWorked

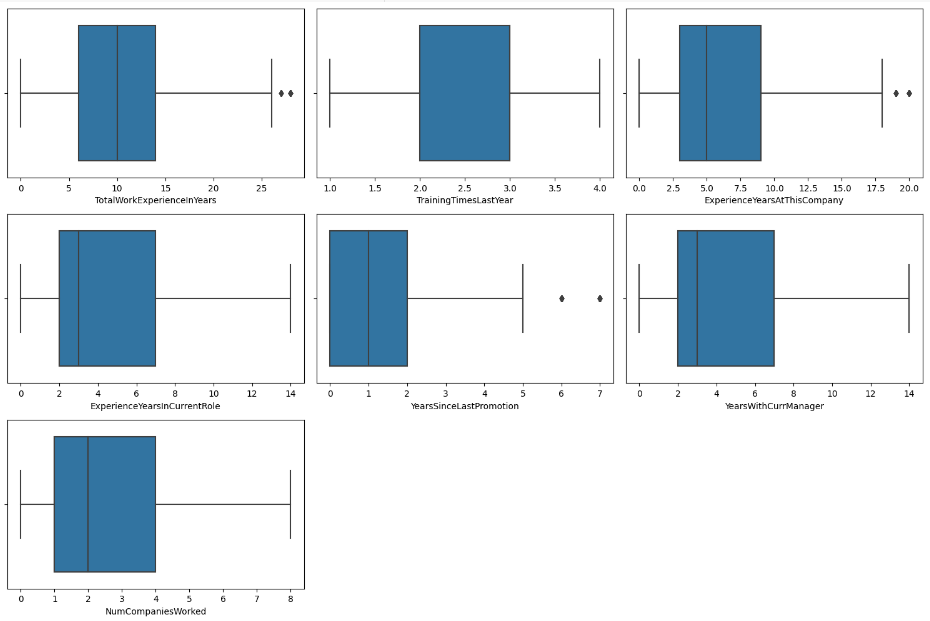


* Giá trị Q1 (tứ phân vị thứ nhất) là 1.0 và Q3 (tứ phân vị thứ ba) là 4.0.
* Giá trị IQR (Interquartile Range) được tính là 3.0.
* Giới hạn dưới (Minimum limit) được tính là -3.5 và giới hạn trên (Maximum limit) là 8.5.
* Số lượng outliers được xác định là 39.
* Nhận xét:
* Dữ liệu có phân phối từ Q1 (1.0) đến Q3 (4.0), với một phạm vi tương đối hẹp giữa 25% và 75% của dữ liệu.
* Số lượng outliers được xác định là 39, đây là một số lượng khá lớn so với tổng số lượng dữ liệu.

Kiểm tra lại Outliers sau khi sử dụng phương pháp IQR:

* Trực quan hóa dữ liệu để quan sát các thuộc tính đã được xử lý outliers:

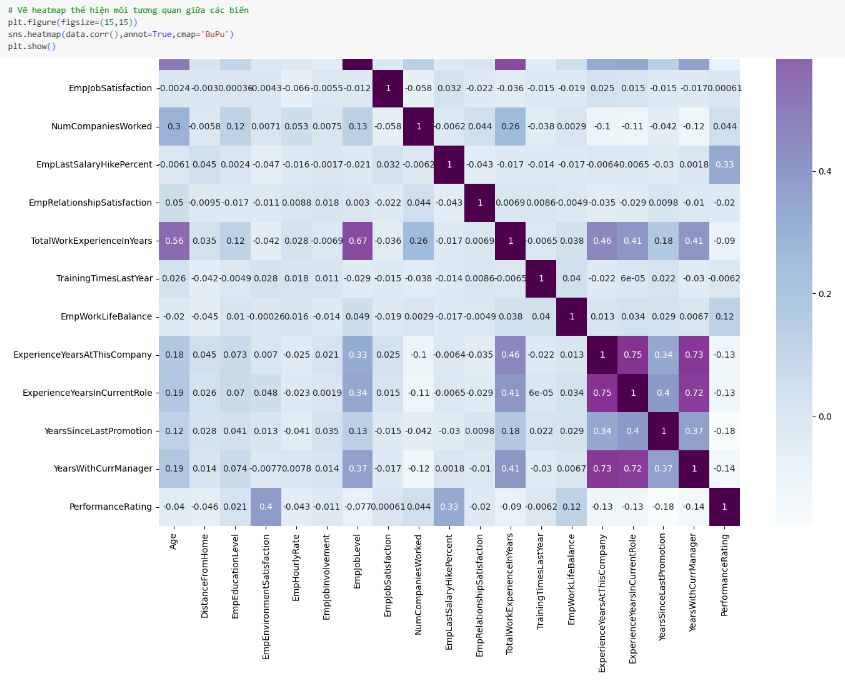




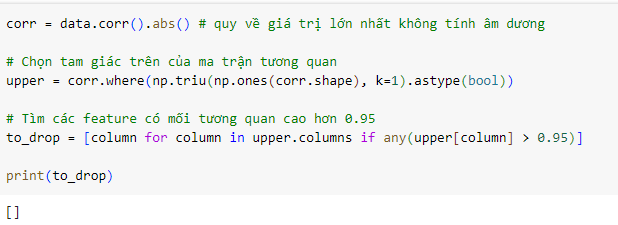
* Nhận xét: Sau khi xử lí những giá trị ngoại lai thì giá trị dữ liệu đã giảm bớt outliers trong từng biến. Từ đây chúng ta có thể thay đổi kết quả khi biểu diễn mô hình trực quan.

## 4. Kiểm tra mối tương quan giữa các biến của bộ dữ liệu

* Việc kiểm tra mối tương quan giữa các biến trong bộ dữ liệu rất quan trọng vì nó cung cấp thông tin về mức độ liên quan hoặc tương tác giữa các biến. Mục đích chính của việc kiểm tra mối tương quan là:
* Khi kiểm tra mối tương quan giữa các biến ta biết được thông tin về mức độ liên quan hoặc tương tác giữa các biến.
* Giúp ta biết được liệu một biến có thể được dự đoán hoặc ảnh hưởng bởi các biến khác hay không.
* Ngoài ra, kiểm tra mối tương quan giữa các biến cũng giúp đánh giá mức độ độc lập của chúng. Nếu hai hoặc nhiều biến có mối tương quan mạnh, điều này có thể dẫn đến hiện tượng đa cộng tuyến khi xét tương quan giữa các biến và biến target.
* Từ việc kiểm định mối tương quan ta có thể lựa chọn các biến quan trọng để xây dựng mô hình hình hóa dự đoán bộ dữ liệu.
* Sử dụng heatmap để quan sát sự tương quan giữa các biến:



* Nhận xét: Từ biểu đồ heatmap ta thấy răng các cặp biến không có mức độ tương quan cao hơn 0.95. Tuy nhiên ta cần kiểm tra lại bằng hàm tính ma trận tương quan (không xét dấu âm hay dương) giữa các cặp biến như sau:



* Kết quả cho thấy rằng những quan sát trong biểu đồ heatmap và kiểm tra bằng ma trận tương quan khớp với nhau

# **III. Trực quan hóa dữ liệu**

## Trực quan đơn biến

### 1.1. Qualitative data

* Nominal data

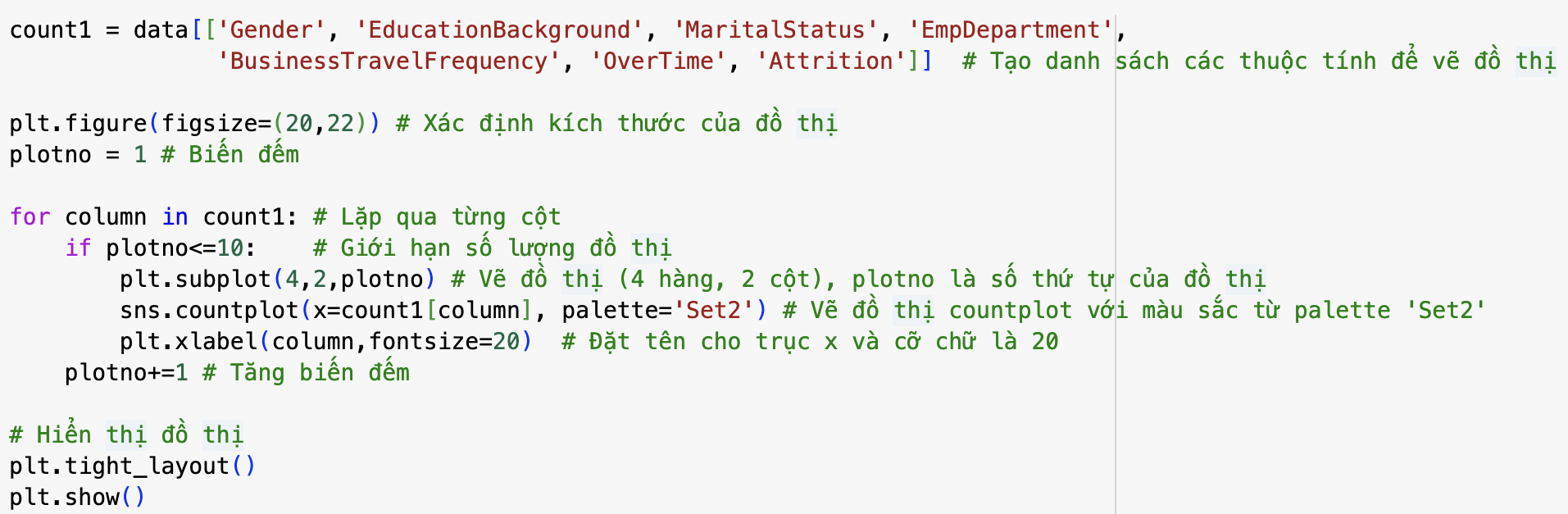
+ Được sử dụng để gắn nhãn các biến không có bất kỳ thứ tự để phân loại đối tượng, giá trị của chúng được phân phối cho các danh mục riêng biệt.

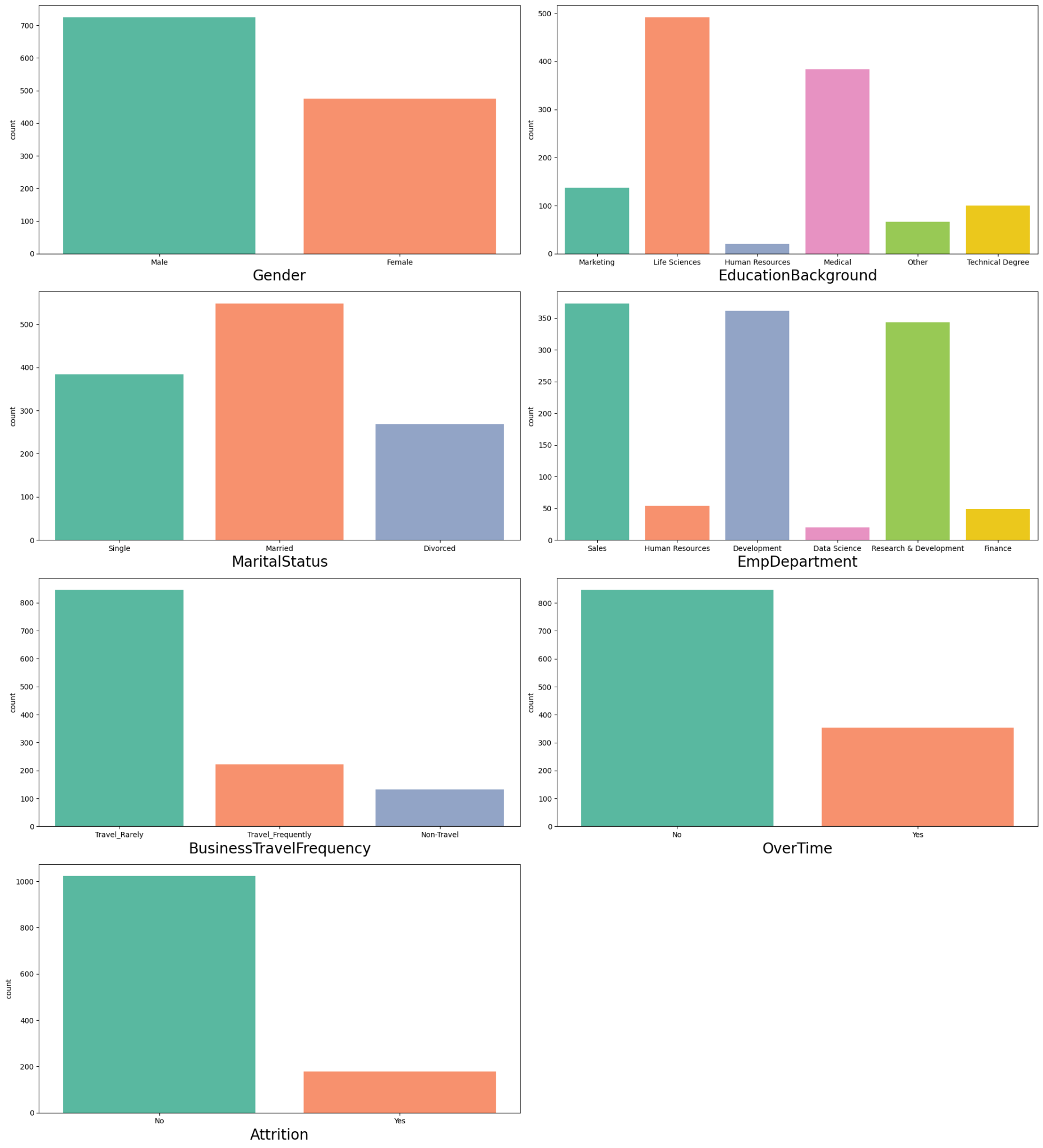
+ Có 8 biến thuộc Nominal data: ‘Gender’,‘EducationBackground’,

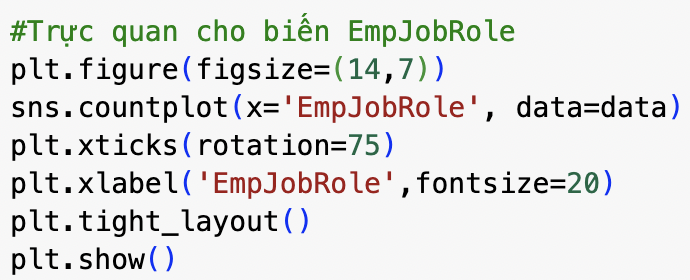
‘MaritalStatus’,‘EmpDepartment’,‘EmpJobRole’,‘BusinessTravelFrequen’, ‘OverTime’,‘Attrition’.

+ Sử dụng countplot từ thư viện ‘seaborn’ và ‘matplotlib’

để trực quan các biến trên.









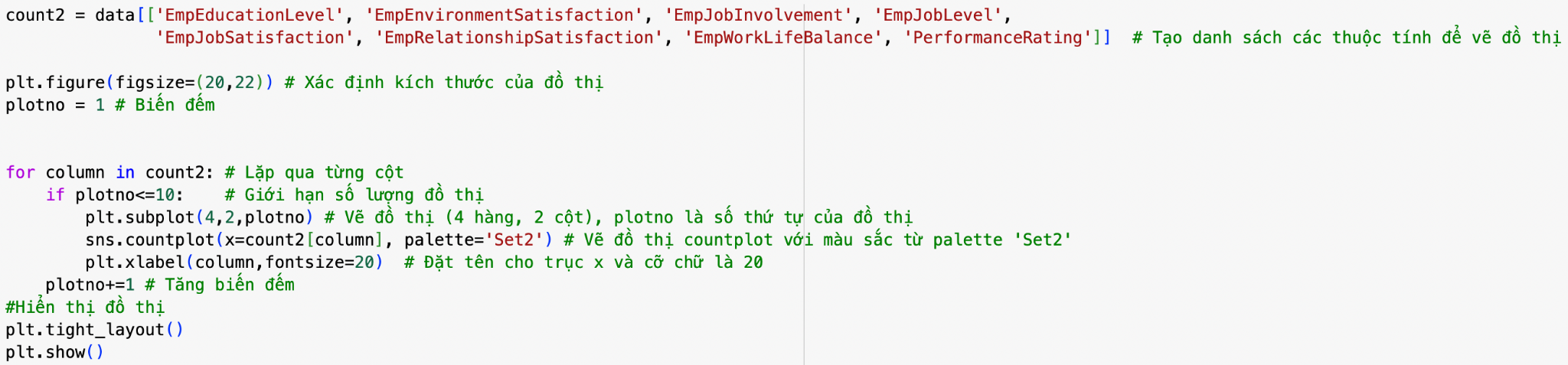
* Nhận xét:
* Gender: Hầu hết nhân viên trong công ty là Nam.
* EducationBackground: Hầu hết học vấn của nhân viên là ‘Life sciences’ và ‘Medical’, cũng như học vấn ‘Marketing’, với số lượng nhân viên có học vấn như này là hơn 100. Học vấn của nhân viên thuộc các lĩnh vực khác thì ít hơn 100.
* MaritalStatus: Hầu hết nhân viên đã kết hôn, cũng như có ít hơn 400 nhân viên độc thân. Gần 300 nhân viên có tình trạng hôn nhân là đã ly dị.
* EmpDepartment: Hầu hết nhân viên đến từ phòng ‘Sales’(Bán hàng), cũng như số nhân viên từ phòng Development (Phát triển) và Research & Development (Nghiên cứu & Phát triển) là cao nhất, trên dưới 350. Số lượng nhân viên từ phòng Human resources & Finance (Nhân sự & Tài chính) gần như bằng nhau. Phòng Data science (Khoa học dữ liệu) có số lượng nhân viên ít nhất trong công ty.
* BusinessTravelFrequency: Số lượng những nhân viên có tần suất đi công tác chiếm rất cao, trên 800.
* OverTime: Số lượng nhân viên không tăng cao chiếm hơn gấp đôi số lượng nhân viên còn lại.
* Attrition: Số nhân viên không rời khỏi công ty gấp 5 lần nhân viên rời khỏi.
* EmpJobRole: Vai trò công việc của nhân viên Sales executive (điều hành bán hàng) là cao nhất trên 250, cũng như số lượng nhân viên với vai trò Developer (Nhà phát triển) là 230. Nhân viên Manage R&D (quản lý R&D) và Research scientist (Nhà khoa học nghiên cứu) dưới 100 người và tất cả các vai trò công việc còn lại đều dưới 65 người.

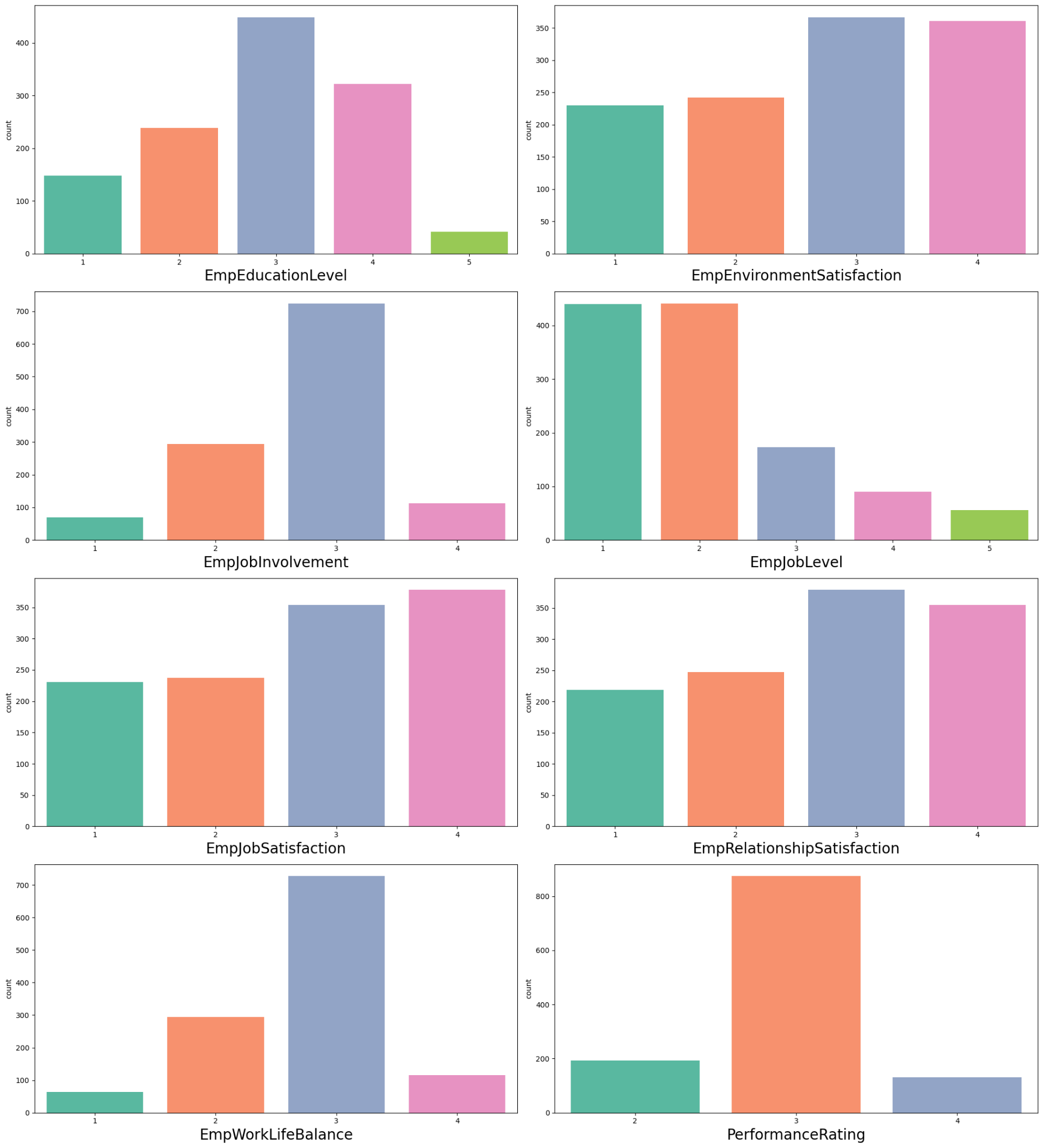
- Ordinal data

+ Là kiểu dữ liệu có thứ tự theo một thang đo nào đó. Những dữ liệu này được sử dụng để quan sát như mức độ hài lòng, mức độ hạnh phúc của khách hàng, v.v.

+ Có 8 biến thuộc Ordinal data: ‘EmpEducationLevel’, ‘EmpEnvironmentSatisfaction’,‘EmpJobInvolvement’,‘EmpJobLevel’,‘EmpJobSatisfaction’,‘EmpRelationshipSatisfaction’, ‘EmpWorkLifeBalance’, ‘PerformanceRating’.

+ Sử dụng countplot từ thư viện ‘seaborn’ và ‘matplotlib’ để trực quan các biến trên.





* Nhận xét:
* EmpEducationLevel: Đa số nhân viên có trình độ học vấn ở mức 3 và 4 (Bachelor và Master).
* EmpEnvironmentSatisfaction: Nnhân viên có độ hài lòng ở mức 3 và 4 (High, Very High) chiếm số lượng cao nhất. Nhưng mức 1 và 2 (Low, Medium) cũng khá cao gần 250.
* EmpJobInvolvement: Đa số nhân viên đều tham gia vào dự án công ty ở mức độ 3 (High).
* EmpJobLevel: Mức độ khó tăng dần từ 1 đến 5 và số lượng nhân viên có công việc có mức độ khó thì giảm dần từ 1 đến 5.
* EmpJobSatisfaction: Những nhân viên có mức độ hài lòng 3, 4 chiếm đa số. Nhưng số lượng chiếm 1, 2 cũng khá cao.
* EmpRelationshipSatisfaction: Đa số số nhân viên có mức đồ hài lòng với mối quan hệ với đồng nghiệp khá tốt. Nhưng những nhân viên không hài lòng lại chiếm hơn 200.
* EmpWorkLifeBalance: Đa số nhân viên có mức độ về việc cân bằng cuộc sống ở mức 3, chiếm ưu thế rõ rệt.
* PerformanceRating: Đây cũng là biến mục tiêu, nhân viên đều có hiệu suất từ 2 trở lên.

### 1.2. Quantitative data

* Discreate data

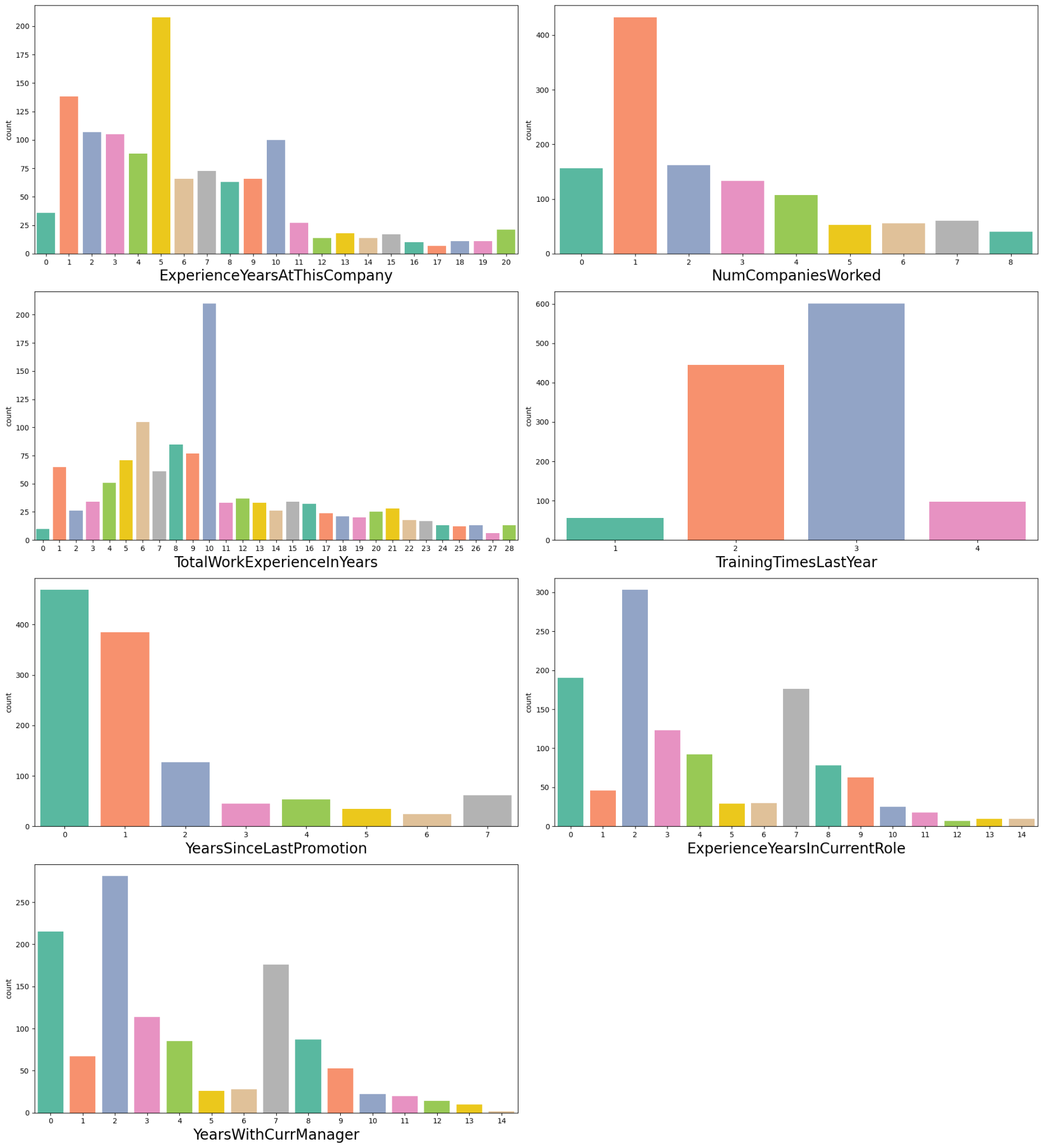
+ Dữ liệu rời rạc có thể đếm được và có giá trị hữu hạn; việc chia nhỏ của chúng là không thể. Là dữ liệu riêng biệt chứa các giá trị thuộc số nguyên.

+ Có 7 biến thuộc Discreate data: 'ExperienceYearsAtThisCompany', 'NumCompaniesWorked','TotalWorkExperienceInYears', 'TrainingTimesLastYear','YearsSinceLastPromotion',

'ExperienceYearsInCurrentRole', 'YearsWithCurrManager.'

+ Sử dụng countplot từ thư viện ‘seaborn’ và ‘matplotlib’ để trực quan các biến trên.



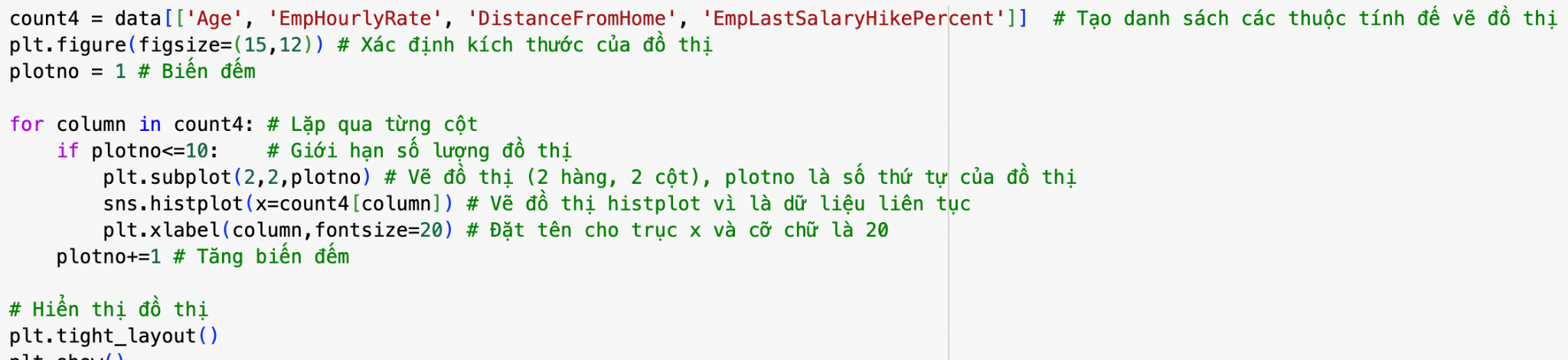


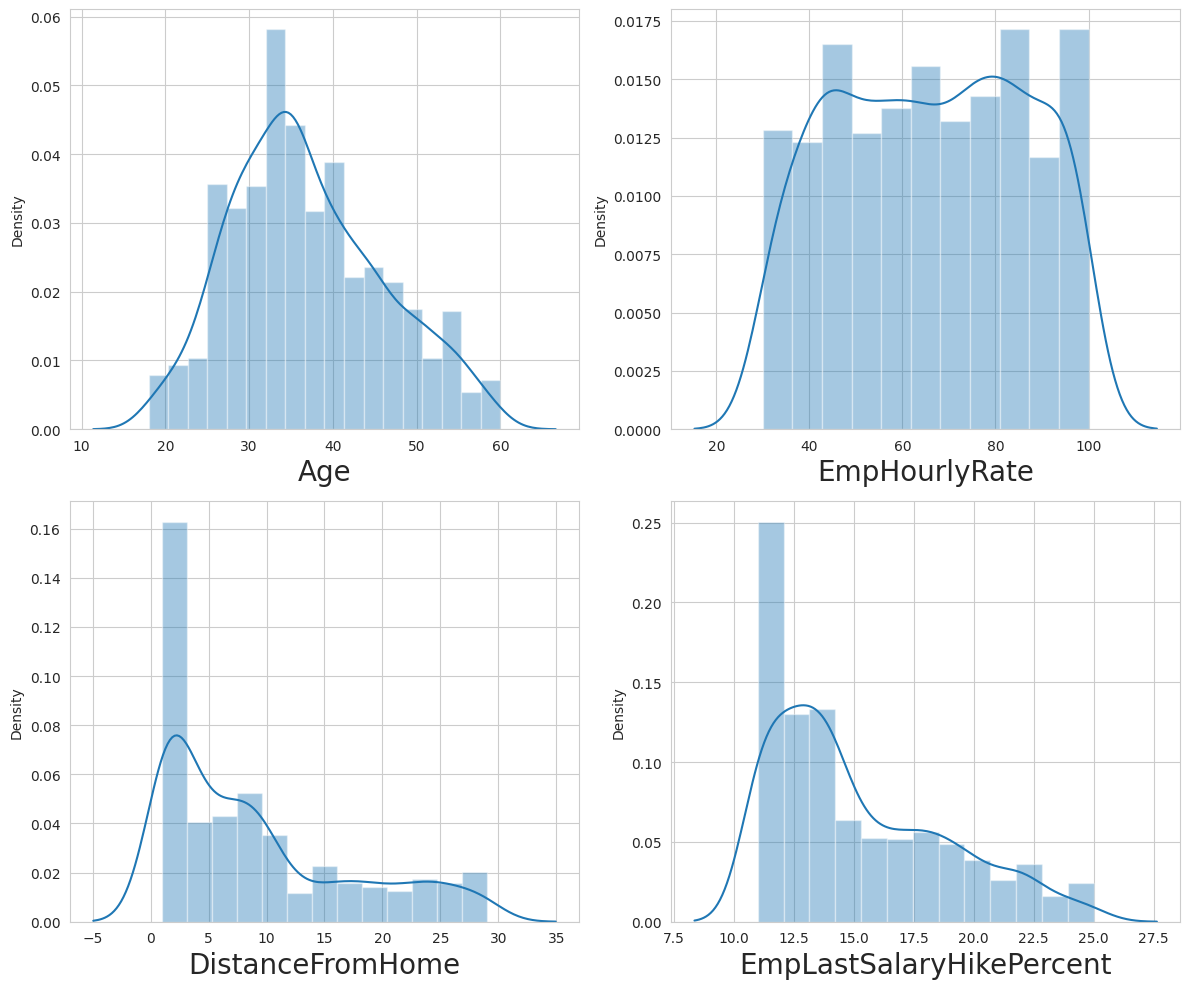
* Nhận xét:
* ExperienceYearsAtThisCompany: Phạm vi kinh nghiệm trong cùng một công ty là từ 0 đến 40, hầu hết nhân viên tham gia từ 0 đến 5.
* NumCompaniesWorked: Nhân viên chỉ từng làm 1 công ty chiếm số lượng đông nhất. Những nhân viên khác rải rác từ 2-9 công ty từng làm.
* TotalWorkExperienceInYears: Kinh nghiệm làm việc nằm trong khoảng từ 0 đến 40, Hầu hết trải nghiệm của nhân viên nằm trong khoảng từ 5 đến 10.
* TrainingTimesLastYear: Thời gian đào tạo trong năm ngoái từ 0 đến 6. Hầu hết nhân viên được đào tạo lần 2 & 3 và tất cả các lần còn lại đều dưới 100.
* YearsSinceLastPromotion: Khoảng thời gian kể từ lần khuyến mãi cuối cùng là từ 0 đến 15. Hầu hết nhân viên chưa được thăng chức trong năm qua, dưới 300 nhân viên đã được thăng chức ít nhất 1 năm.
* ExperienceYearsInCurrentRole: Kinh nghiệm đảm nhận vai trò hiện tại tối đa của nhân viên là 18. Hầu hết nhân viên có 2 năm kinh nghiệm ở vị trí hiện tại thì số năm kinh nghiệm của nhân viên là dưới 180 dưới 200 nhân viên không có kinh nghiệm đảm nhận vai trò hiện tại.
* YearsWithCurrManager: Hầu hết nhân viên đều có 2 năm làm việc với người quản lý hiện tại và tối thiểu 16 năm làm việc với người quản lý hiện tại. Phạm vi từ 0 đến 17 và dưới 180 nhân viên có 7 năm gắn bó với người quản lý hiện tại.
* Continuous data

+ Dữ liệu liên tục được mô tả như một tập hợp các quan sát không bị gián đoạn; có thể được đo trên thang điểm. Dữ liệu liên tục biểu thị thông tin có thể được chia thành các cấp độ nhỏ hơn. Biến liên tục có thể nhận bất kỳ giá trị nào trong một phạm vi.

+ Có 4 biến thuộc Continuous data: ‘Age’, ‘EmpHourlyRate’, ‘TotalWorkExperienceInYears’, ‘ExperienceYearsAtThisCompany’.

+ Sử dụng histplot từ thư viện ‘seaborn’ và ‘matplotlib’ để trực quan các biến trên.

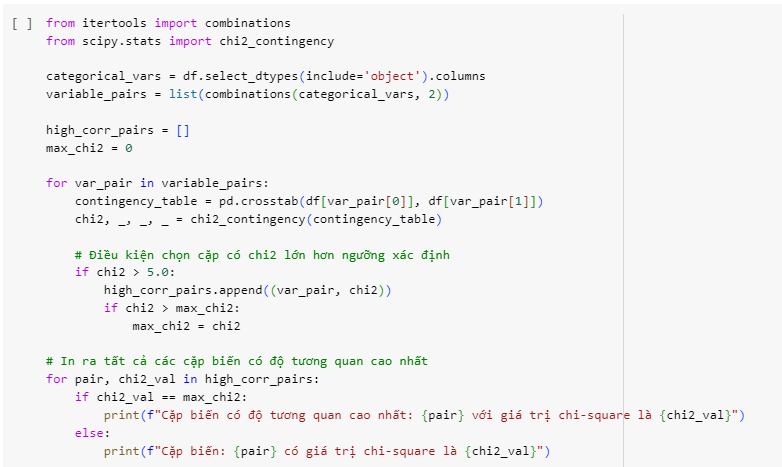


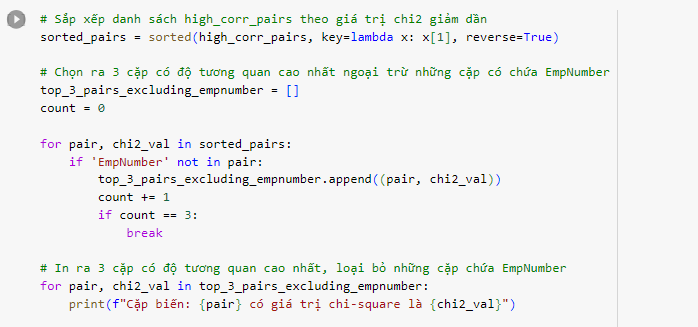


* Nhận xét:
* Age: Phạm vi độ tuổi từ 18 đến 60, phần lớn nhân viên ở độ tuổi từ 25 đến 40.
* EmpHourlyRate: Phạm vi mức lương theo giờ của nhân viên là từ 30 đến 100, hầu hết mức lương theo giờ của nhân viên là 45.
* DistanceFromHome: Khoảng cách từ nhà đến công ty trong phạm vi là từ 1 đến 29. Khoảng cách nhỏ hơn 10 chiếm lượng nhân viên khá đông.
* EmpLastSalaryHikePercent: Tỉ lệ tăng lương từ 11-14% chiếm đa số trong lượng nhân viên và giảm dần từ từ đến 25%.

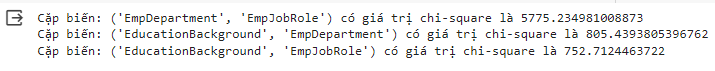
## Trực quan hai biến

### 2.1. Sử dụng kiểm định Chi-square để xem những cặp biến có tương quan với nhau:





* Sau khi lọc ra mối tương quan giữa biến định danh và các biến còn lại ta thu được ba cặp biến có mối tương quan cao như sau:

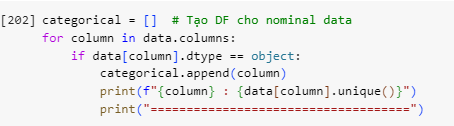


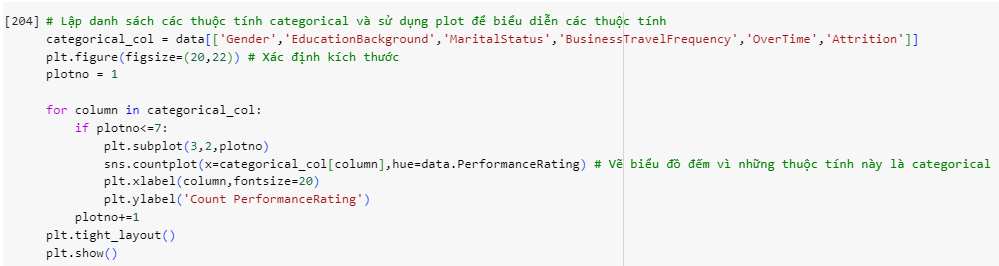
=> Nhận xét: Các biến xét độ tương quan cao nhất không phải là những cặp biến có chứa biến target nên ta đi xét từng biến với biến target để có thể đưa ra kết luận cho mục tiêu của bài là "Phân tích những yếu tố có ảnh hưởng đến hiệu suất làm việc của nhân viên". Từ đó xây dựng mô hình hồi quy dự đoán kết quả. Bên cạnh đó khi phân tích các biến có ảnh hưởng đến biến target để xây dựng mô hình cần lưu ý khi cặp biến có chứa những biến trên vì có thể xảy ra khả năng đa cộng tuyến dẫn đến việc đánh giá sai về tác động của mỗi biến lên mô hình.

### 2.2. Xét tương quan giữa biến nominal categorical và biến target

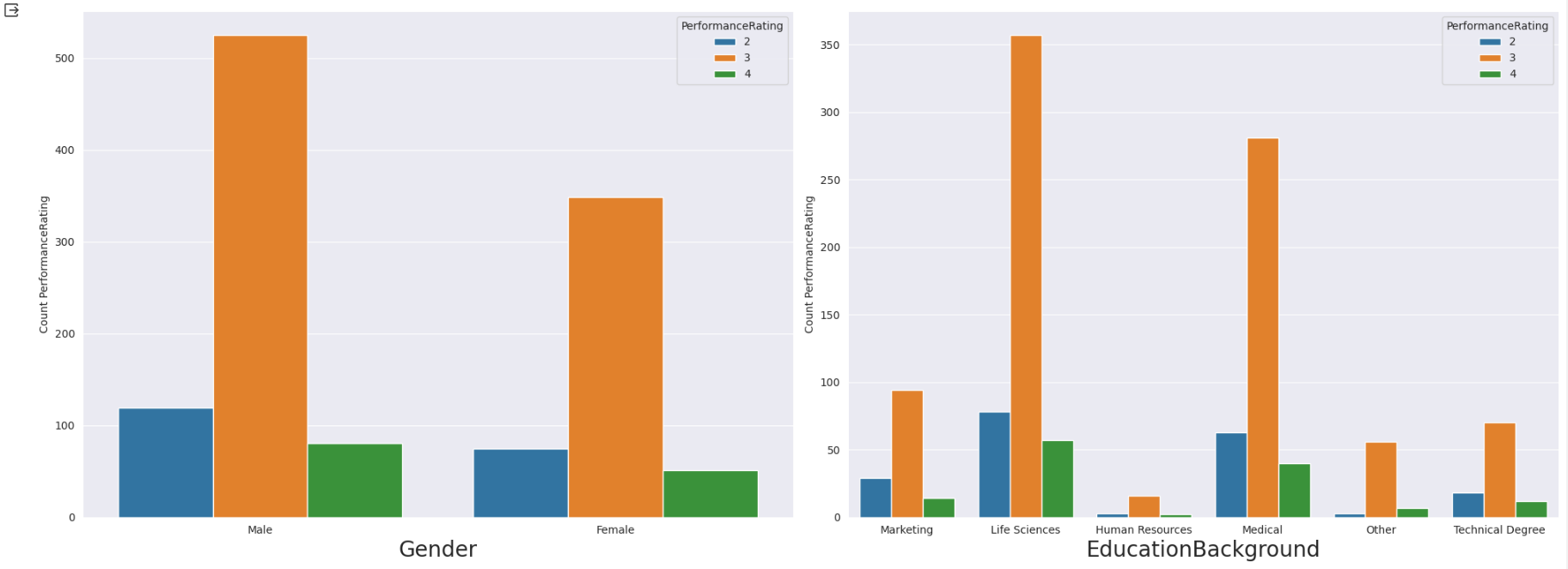
#### Biểu diễn trực quan cho các biến nominal categorical

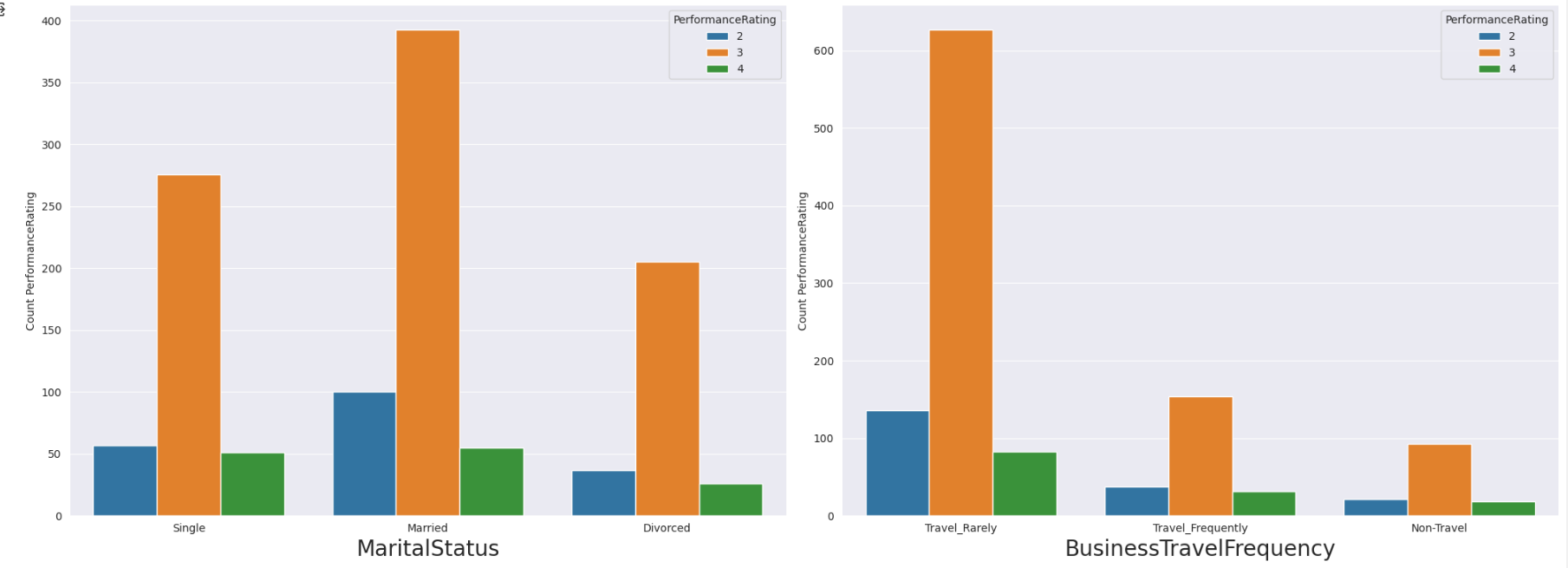
* Tại Dataframe cho biến nominal categorical, sau đó biểu diễn trực quan giữa các biến nominal và biến target để xác định tương quan:

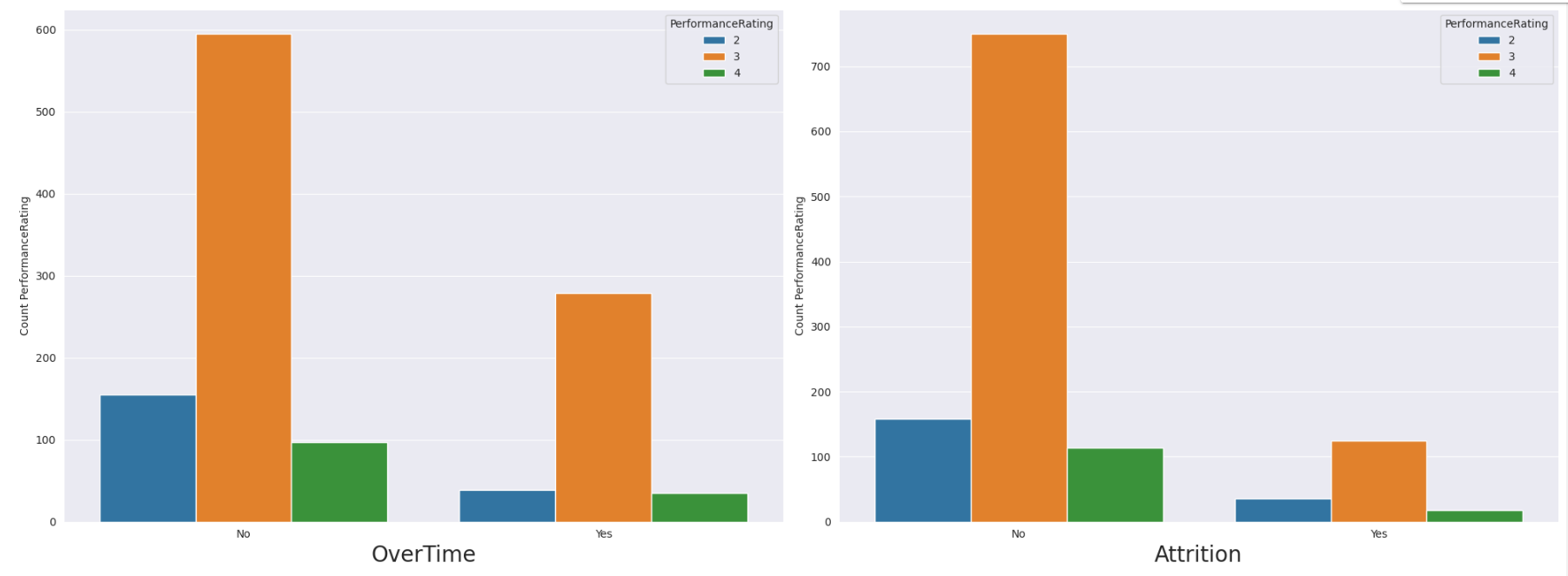




* Mã code Python được sử dụng để tạo biểu đồ đếm (countplot) cho các thuộc tính categorical trong bộ dữ liệu. Biểu đồ này sử dụng thư viện seaborn để biểu diễn số lượng các giá trị khác nhau của mỗi thuộc tính và phân loại theo hiệu suất (PerformanceRating).
* Biểu đồ countplot cho mỗi thuộc tính categorical sẽ được hiển thị trong các subplot tương ứng, mỗi biểu đồ sẽ biểu thị số lượng các giá trị của thuộc tính numerical và phân loại theo hiệu suất (PerformanceRating).







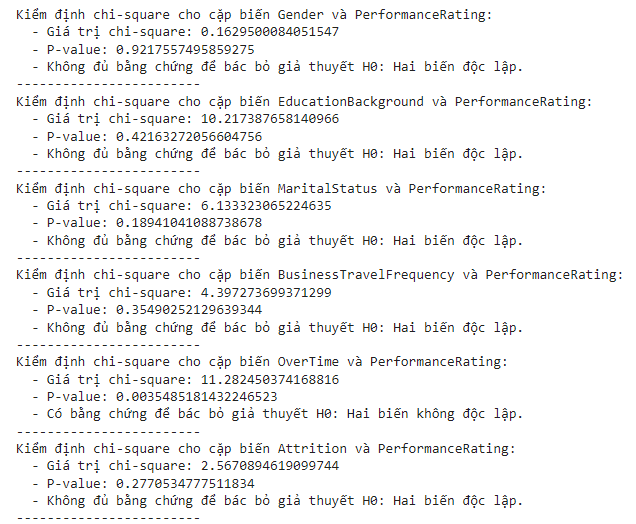
* Nhận xét:
* Nhìn chung, biến target PerformanceRating có sự biến động khá rõ khi giá trị của các thuộc tính thay đổi. Hiệu suất làm việc của nhân viên được thống kê cho thấy nhân viên làm việc đạt hiệu suất mức 3 (Excellent) chiếm phần lớn trong công ty khi khảo sát theo giới tính, ngành học, tình trạng hôn nhân, tần suất công tác, tăng ca và nghỉ việc vì lí do cá nhân.
* Quan sát biểu đồ ta có thể nhận thấy được rằng PerformanceRating ở mức 1 (Low) hầu như không xuất hiện tại các biến phân loại. Đây là tín hiệu tích cực của công ty. Bên cạnh đó, có sự chênh lệch khá lớn giữa hiệu suất làm việc mức 3 và các hiệu suất khác
* Năng suất làm việc có sự thay đổi trong các giá trị của biến nhưng xếp hạng các nhóm giá trị quy ước vẫn không thay đổi nhiều. Ví dụ trong biến Gender, dù có sự thay đổi giá trị từ ‘Male’ sang ‘Female’ số lượng năng suất làm việc của từng giá trị có sự thay đổi nhưng các giá trị quy ước vẫn giữ nguyên hiệu suất làm việc cao nhất là 3 (Excellent)> 2 (Good)> 4 (Outstanding) và tương tự với các biến phân loại khác.

#### Kiểm định mối tương quan giữa các biến nominal categorical và biến target (PerformanceRating)

* Ta gán các giá trị cho alpha và độ tin cậy, sau đó dùng vòng lặp để bỏ cột target đi, đồng thời tính giá trị chi-square và p-value cho từng cặp biến.



* Ta có kết quả về giá trị Chi-square và p-value của cặp biến đó, cùng kết luận bác bỏ hay chấp nhận H0.



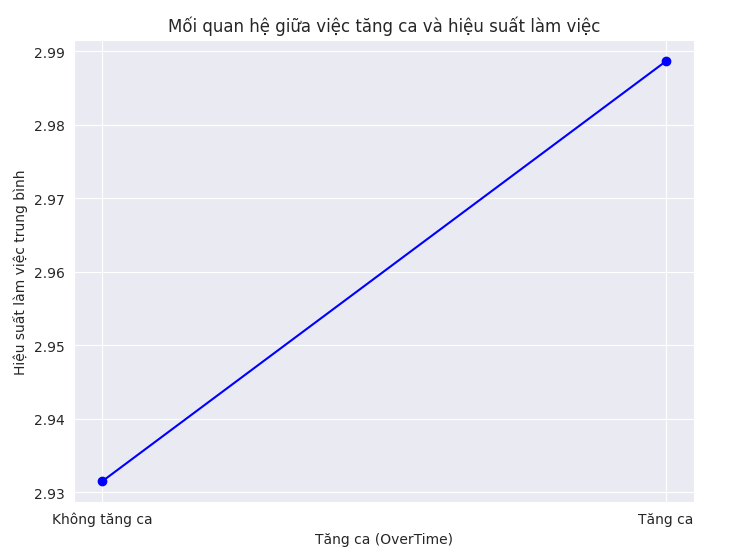
* **Nhận xét**: Từ biểu đồ trực quan và kiểm định ta có thể nhận ra rằng các biến phân loại không tương quan mạnh so với biến target, nghĩa là hai biến độc lập và ít có ảnh hưởng đến sự thay đổi của nhau

Khi các biến phân loại thay đổi thì hiệu suất của nhân viên vẫn không bị ảnh hưởng nhiều. Nên trong quá trình dự đoán hiệu suất của nhân viên ta không quan tâm nhiều về sự biến thiên của nó. Ngoài ra, trong kiểm định trên biến OverTime (làm thêm giờ) và PerformanceRating có p-value dưới mức ý nghĩa thông thường là 0.05 nên có thể bác bỏ giải thuyết H0.

- Biến OverTime có ảnh hưởng đến biến target PerformanceRating, nghĩa là việc tăng ca có thể ảnh hưởng đến hiệu suất đánh giá

#### Trực quan 2 biến overtime và biến target sau khi đã xác định được 2 biến này không độc lập

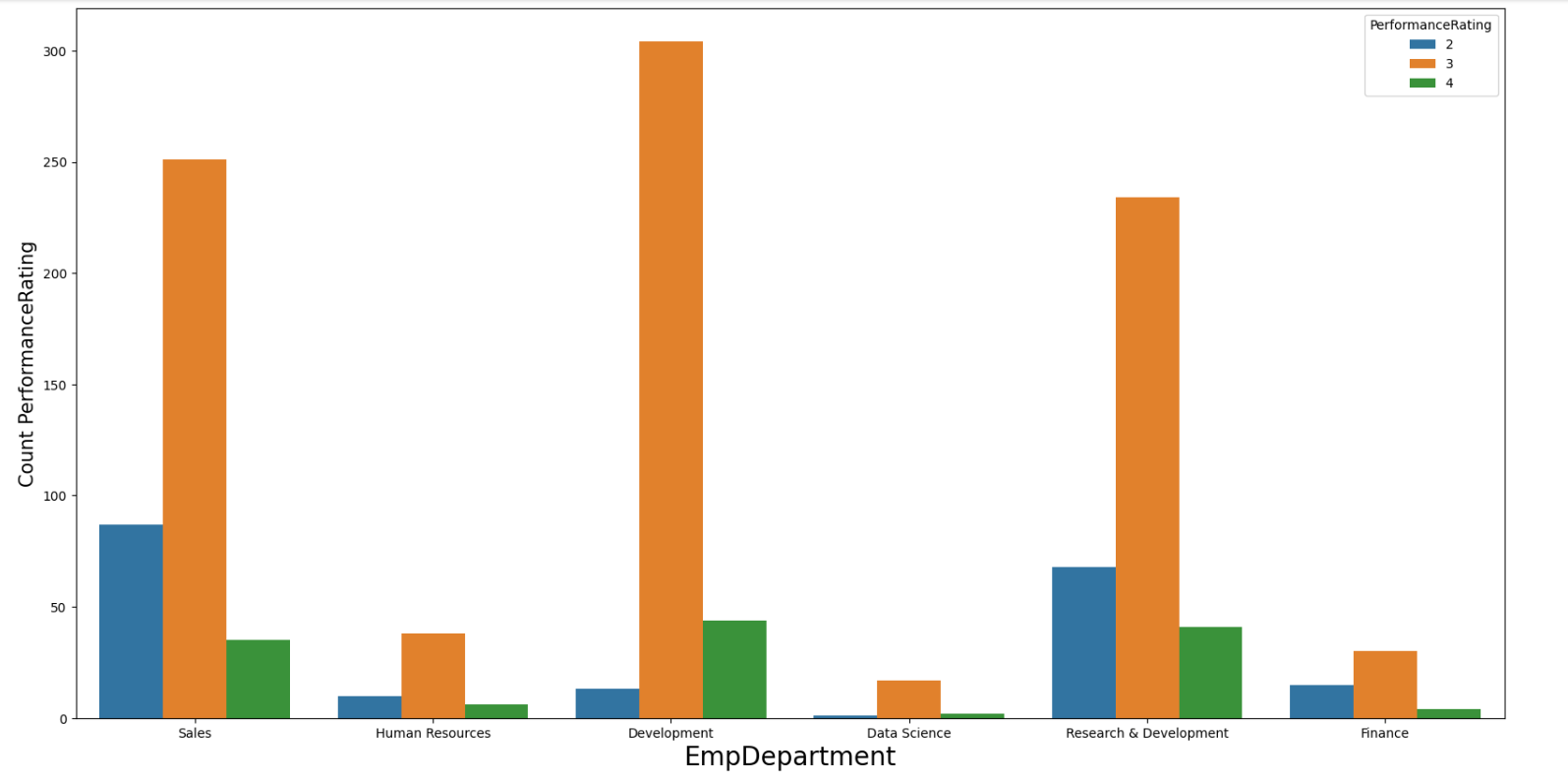
* Ta dùng lineplot để biểu diễn trực quan mỗi quan hệ giữa việc tăng ca và hiệu suất làm việc



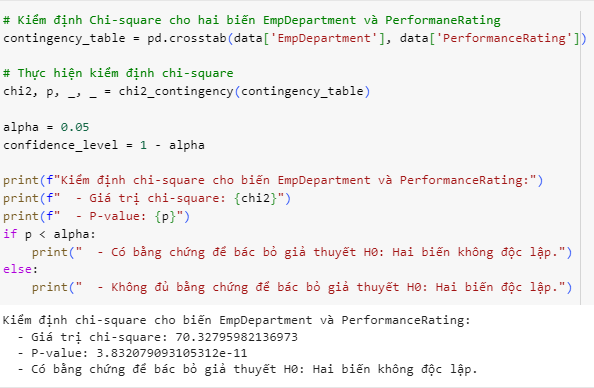
* Nhận xét: Từ biểu đồ trực quan cho thấy biến 'OverTime' và biến 'PerformanceRating' tương quan thuận với nhau (hai biến này có mối quan hệ đồng biến). Biến động của đường biểu diễn không lớn nên có thể kết luận rằng không có các yếu tố khác ảnh hưởng đến mối tương quan giữa tăng ca và hiệu suất làm việc

#### Trực quan hai biến EmpDepartment (phòng ban nhân viên) và PerformanceRating (hiệu suất làm việc)

* Dùng Countplot để trực quan mối quan hệ giữa 2 biến trên.



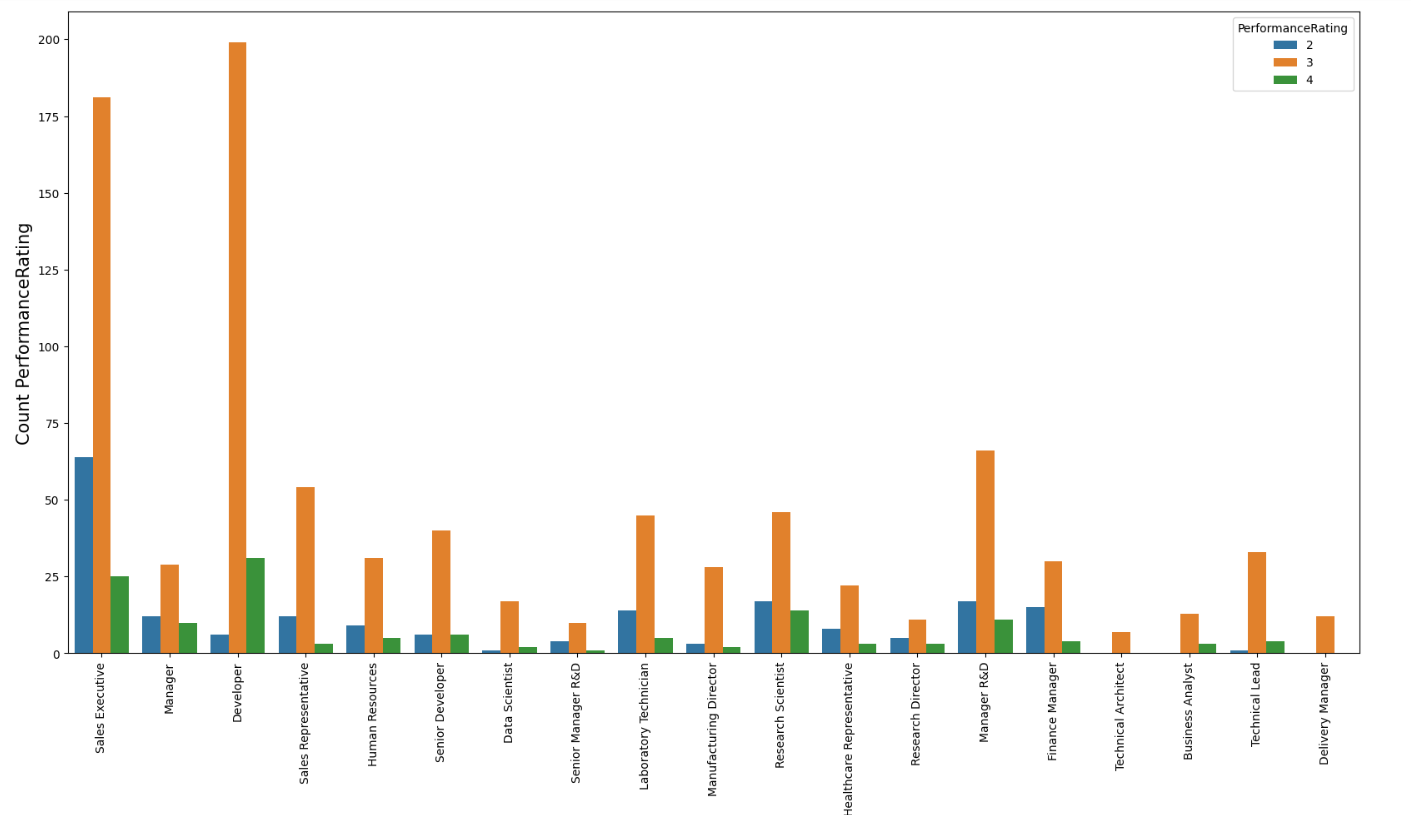
* Sau khi quan sát mối quan hệ, ta dùng kiểm định chi-square cho biến EmpDepartment và PerformanceRating.



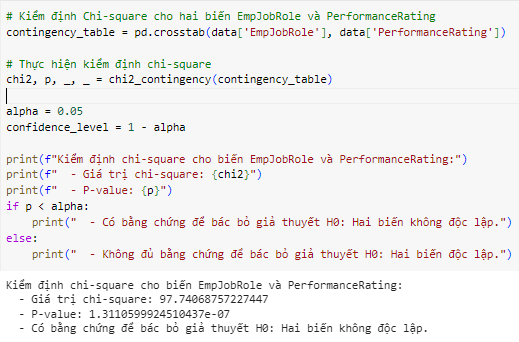
* Nhận xét: Kết quả của kiểm định trên cho thấy rằng cặp biến EmpDepartment (phòng ban nhân viên) và PerformanceRating (đánh giá hiệu suất) có bằng chứng đủ mạnh để bác bỏ giả thuyết H0 ("Hai biến không độc lập"). Có thể tồn tại sự ảnh hưởng của phòng ban làm việc đến hiệu suất làm việc của nhân viên. Có thể cho thấy sự khác biệt trong hiệu suất làm việc giữa các phòng ban khác nhau (được thể hiện qua trực quan biểu đồ cột ở trên).

#### Trực quan 2 biến EmpJobRole (Vị trí của nhân viên) và PerformanceRating (đánh giá hiệu suất)

* Tương tự như 2 biến EmpDepartment và target (PerformanceRating). Ta dùng Countplot để trực quan mối quan hệ, sau đó dùng kiểm định chi-square cho 2 biến này.

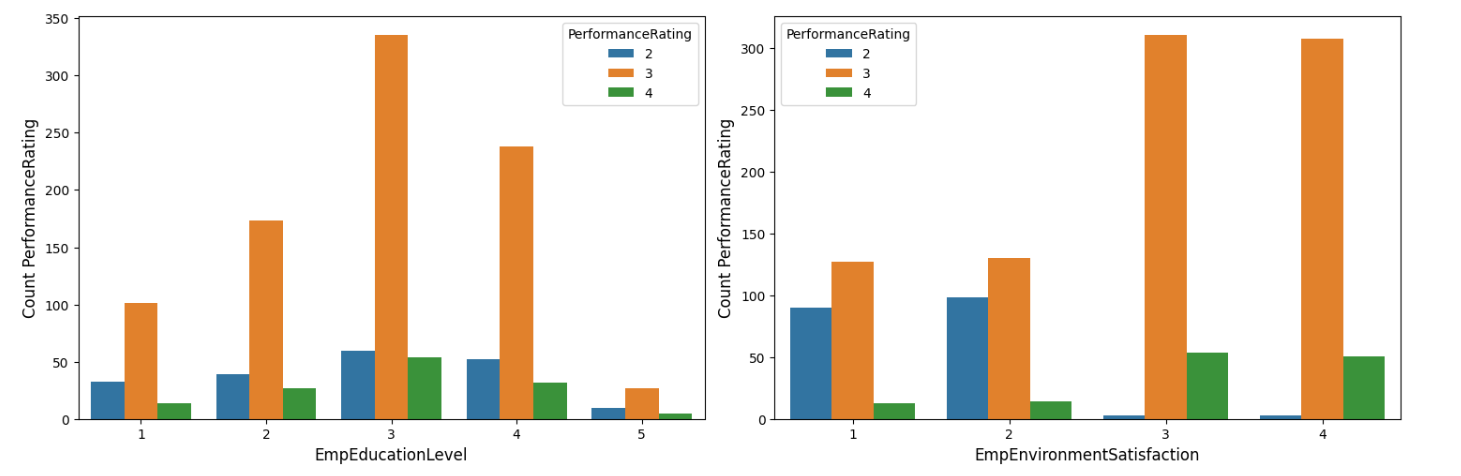


* Kiểm định Chi-square:

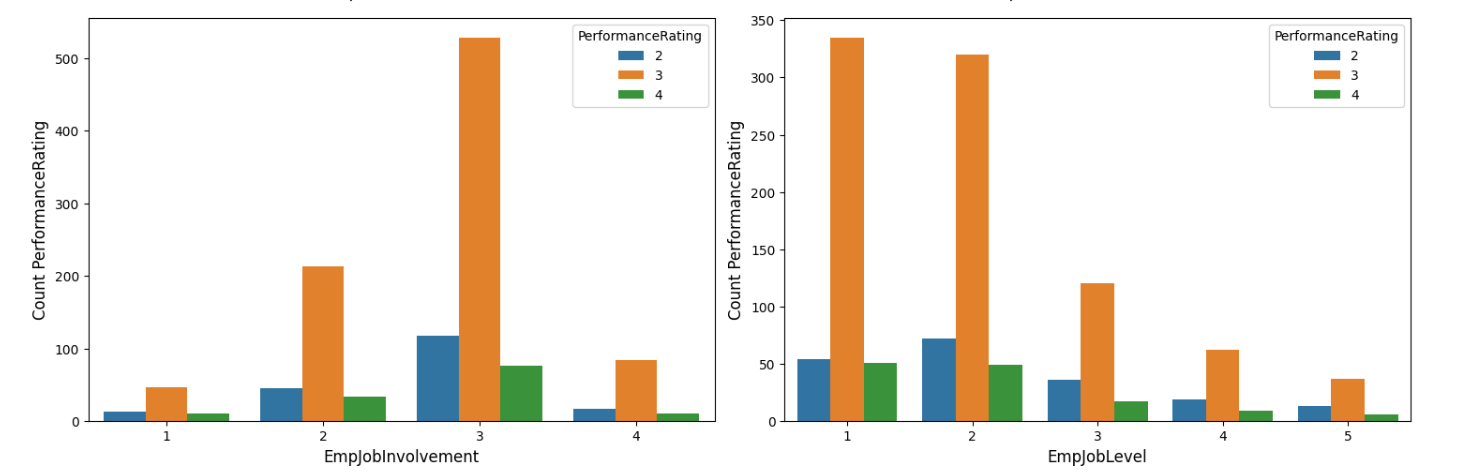


* Nhận xét: Kết quả kiểm định chi-square cho cặp biến EmpJobRole (vị trí công việc) và PerformanceRating (đánh giá hiệu suất) cho thấy giá trị p-value rất nhỏ có bằng chứng đủ mạnh để bác bỏ giả thuyết H0 ("Hai biến không độc lậ"). Có mối liên hệ đáng kể giữa vị trí công việc của nhân viên và Hiệu suất làm việc của họ. Có thể có sự khác biệt trong hiệu suất làm việc giữa các vị trí công việc khác nhau. Nghĩa là vị trí công việc của nhân viên có ảnh hưởng đáng kể đến hiệu suất làm việc.

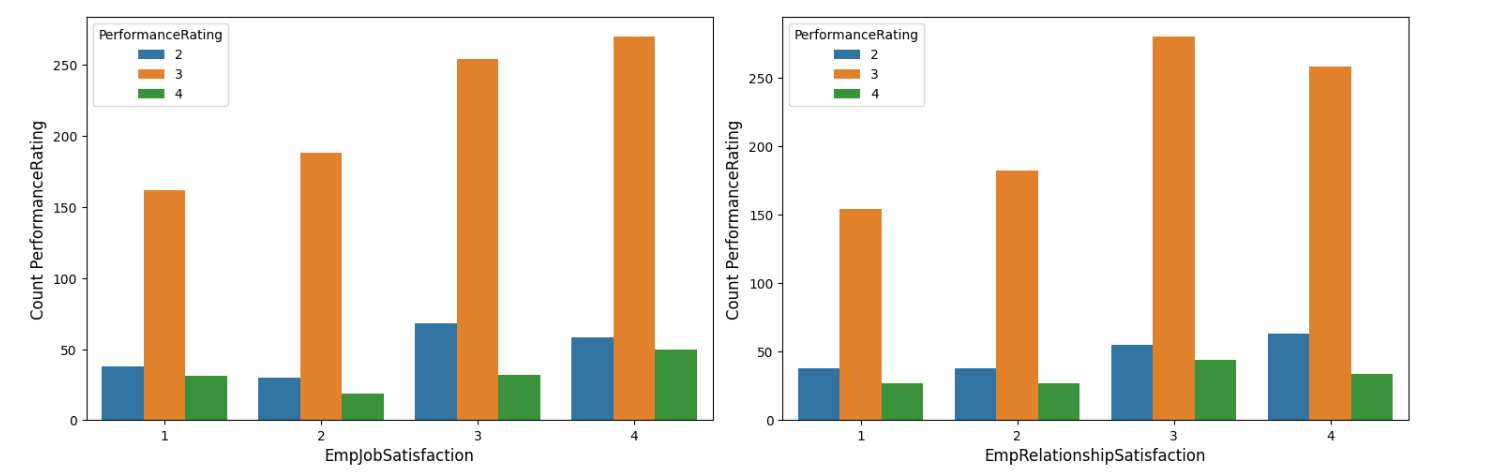
### 2.3. Biểu diễn trực quan mối liên hệ giữa các biến ordinal và biến target (PerformanceRating)



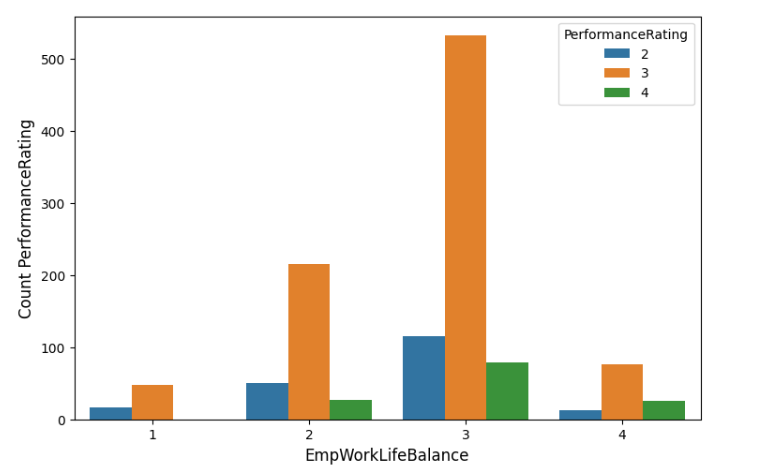
*Trực quan 2 biến EmpEducationLevel, EmpEnvironmentSatisfaction với target (PerformanceRating)*



*Trực quan 2 biến EmpJobInvolvement, EmpJobLevel với target (PerformanceRating)*

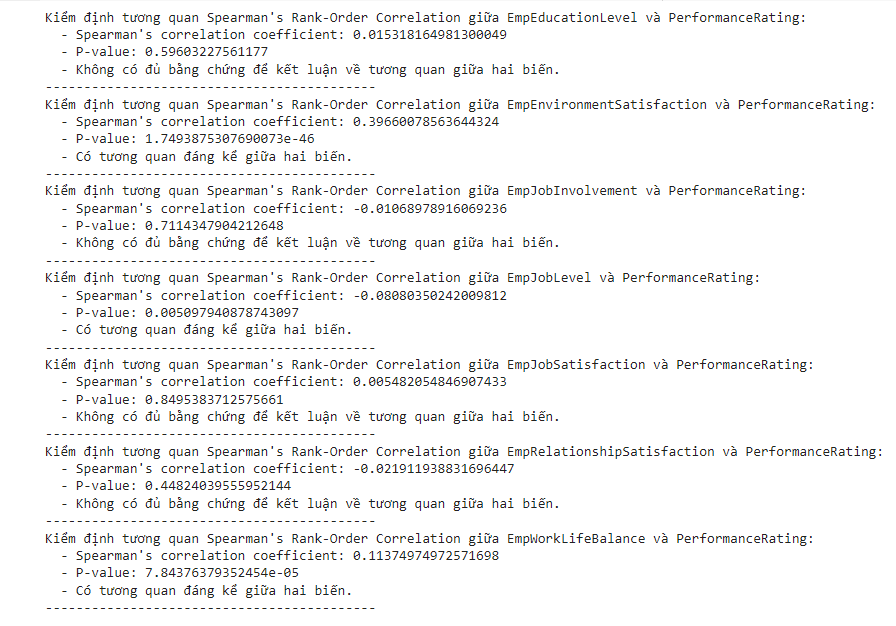


*Trực quan 2 biến EmpJobSatisfaction, EmpRelationshipSatisfaction với target (PerformanceRating)*

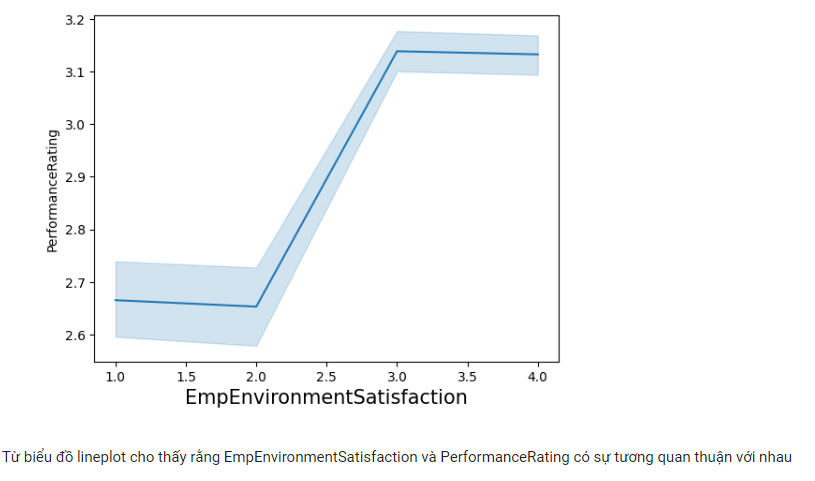


*Trực quan 2 biến EmpWorkLifeBalance và target (PerformanceRating)*

* Sau khi trực quan mối quan hệ giữa các biến ordinal với target, ta dùng kiểm định tương quan Spearman's Rank-Order Correlation để xác định độ tương quan giữa các biến ordinal và target

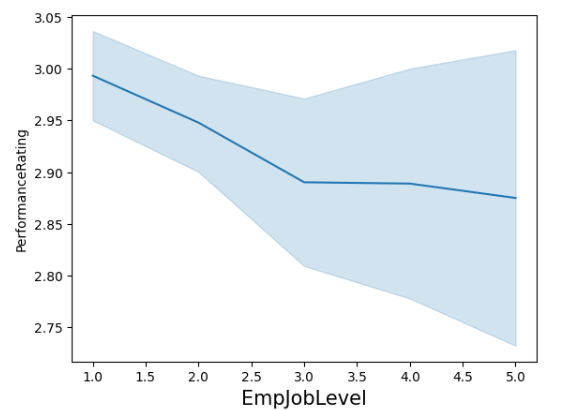


* Dựa trên kiểm định Spearman ở trên ta có thể rút ra một số biến có tương quan đến biến Target PerformanceRating như sau:
* EmpEnvironmentSatisfaction và PerformanceRating có giá trị p-value rất nhỏ chỉ ra sự tương quan mạnh mẽ giữa mức độ hài lòng với môi trường làm việc và hiệu suất làm việc.
* EmpJobLevel và PerformanceRating: Giá trị p-value là 0.005, dưới mức ý nghĩa thông thường. Có sự tương quan đáng kể giữa cấp bậc công việc và hiệu suất làm việc.
* EmpWorkLifeBalance và PerformanceRating: Giá trị p-value rất nhỏ chỉ ra mối tương quan đáng kể giữa cân bằng giữa công việc và cuộc sống cá nhân với hiệu suất làm việc.



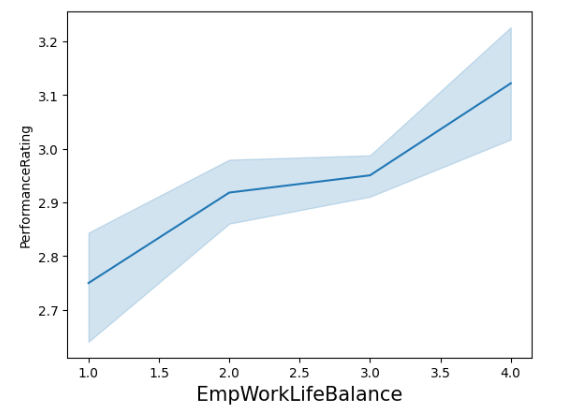
*Biểu diễn sự tương quan của biến EmpEnvironmentSatisfaction với biến target*

=> Từ biểu đồ lineplot cho thấy rằng EmpEnvironmentSatisfaction và PerformanceRating có sự tương quan thuận với nhau

**

*Biểu diễn sự tương quan của biến EmpJobLevel với biến target*

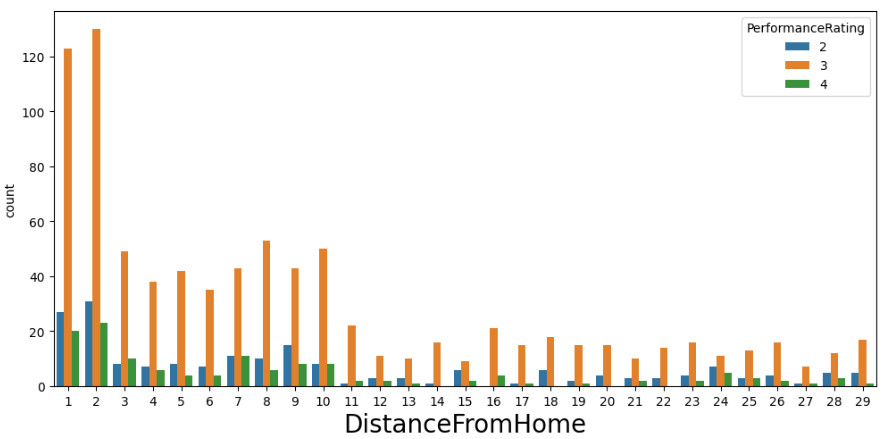
=> Từ biểu đồ lineplot cho thấy rằng EmpJobLevel và PerformanceRating có sự tương quan thuận với nhau

**

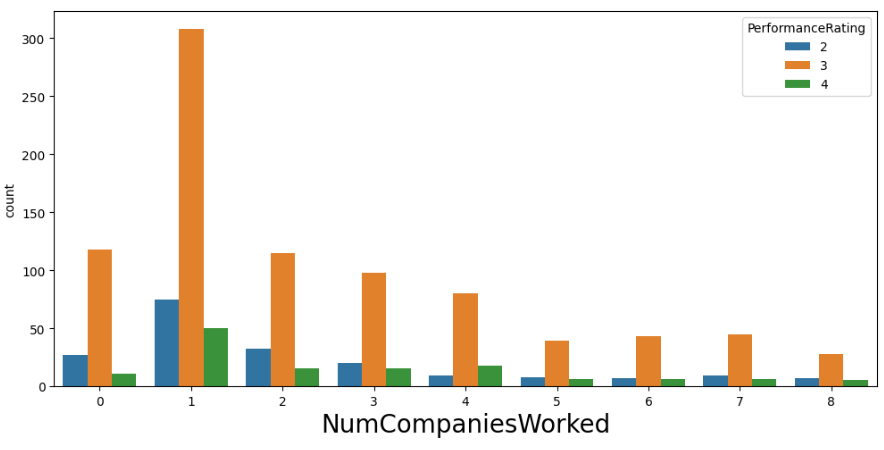
*Biểu diễn sự tương quan của biến EmpWorkLifeBalance với biến target*

=> Từ biểu đồ lineplot cho thấy rằng EmpWorkLifeBalance và PerformanceRating có sự tương quan thuận với nhau

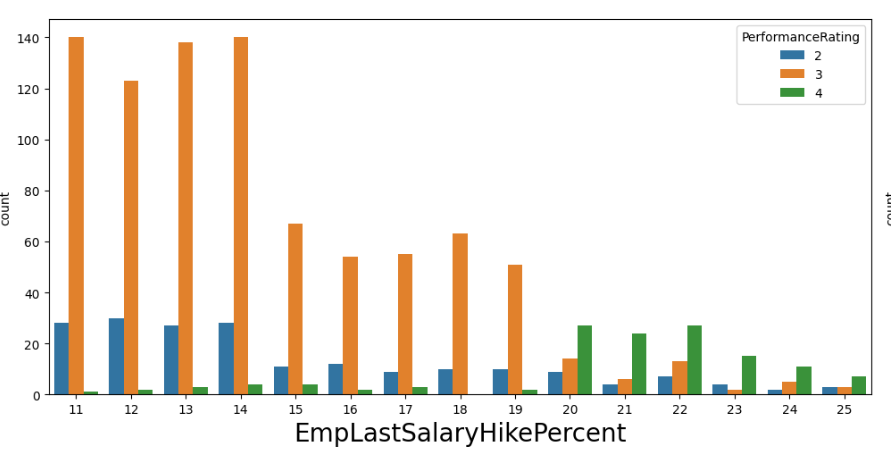
### 2.4. Trực quan biến numerical với biến target (PerfomanceRating)



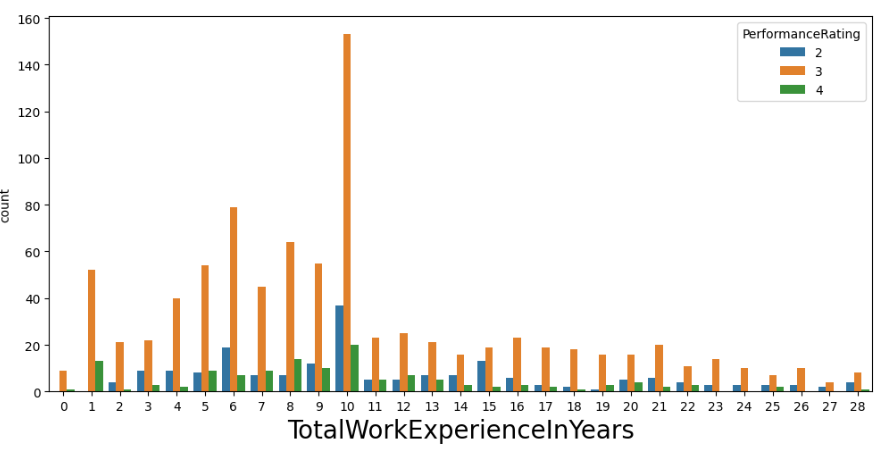
*Biểu diễn mối quan hệ của biến DistanceFromHome với biến target*



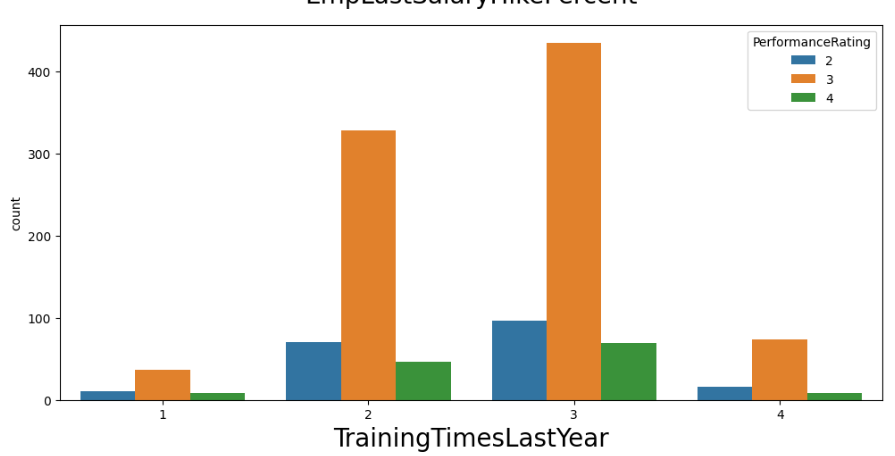
*Biểu diễn mối quan hệ của biến NumCompaniesWorked với biến target*

**

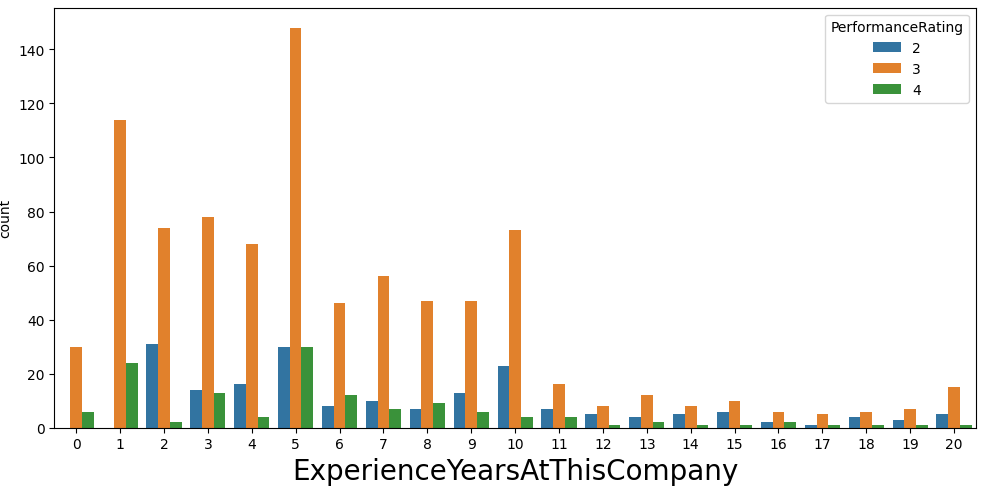
*Biểu diễn mối quan hệ của biến EmpLastSalaryHikePercent với biến target*

**

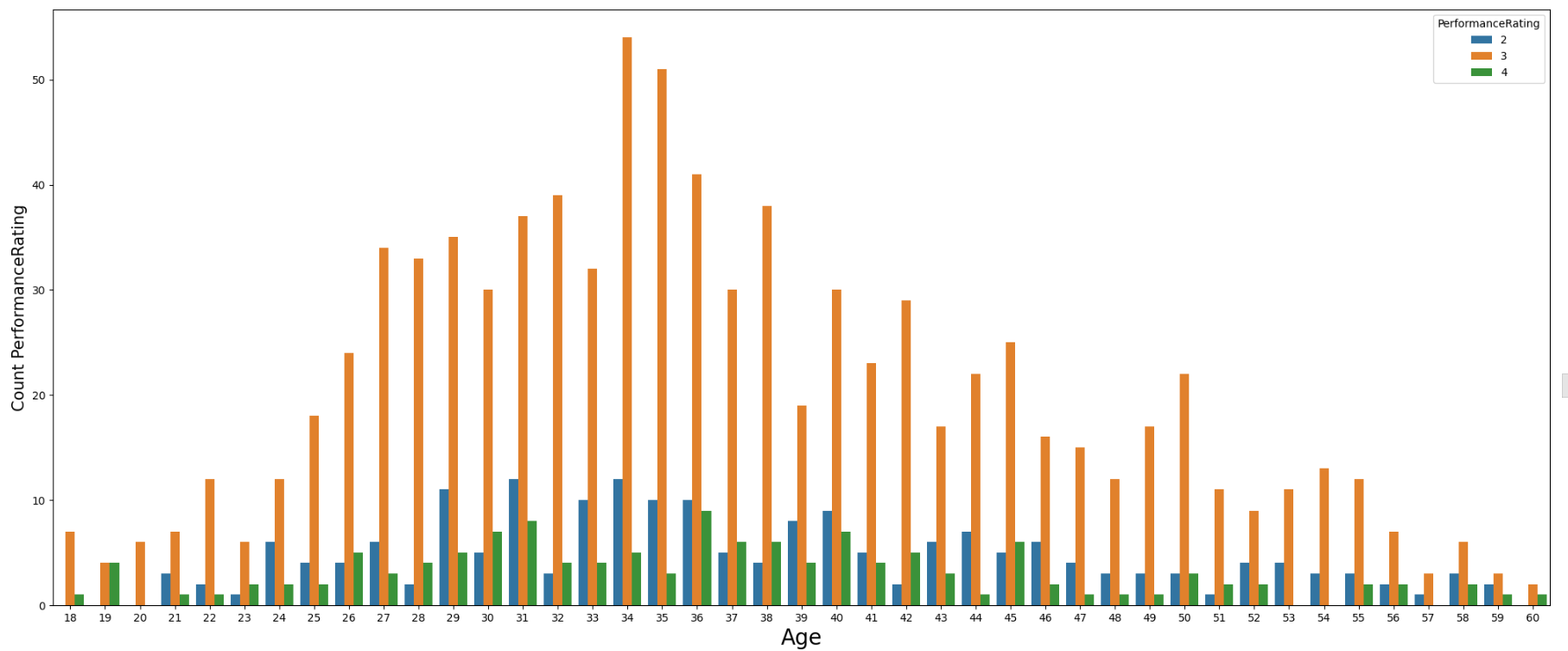
*Biểu diễn mối quan hệ của biến TotalWorkExperienceInYears với biến target*

**

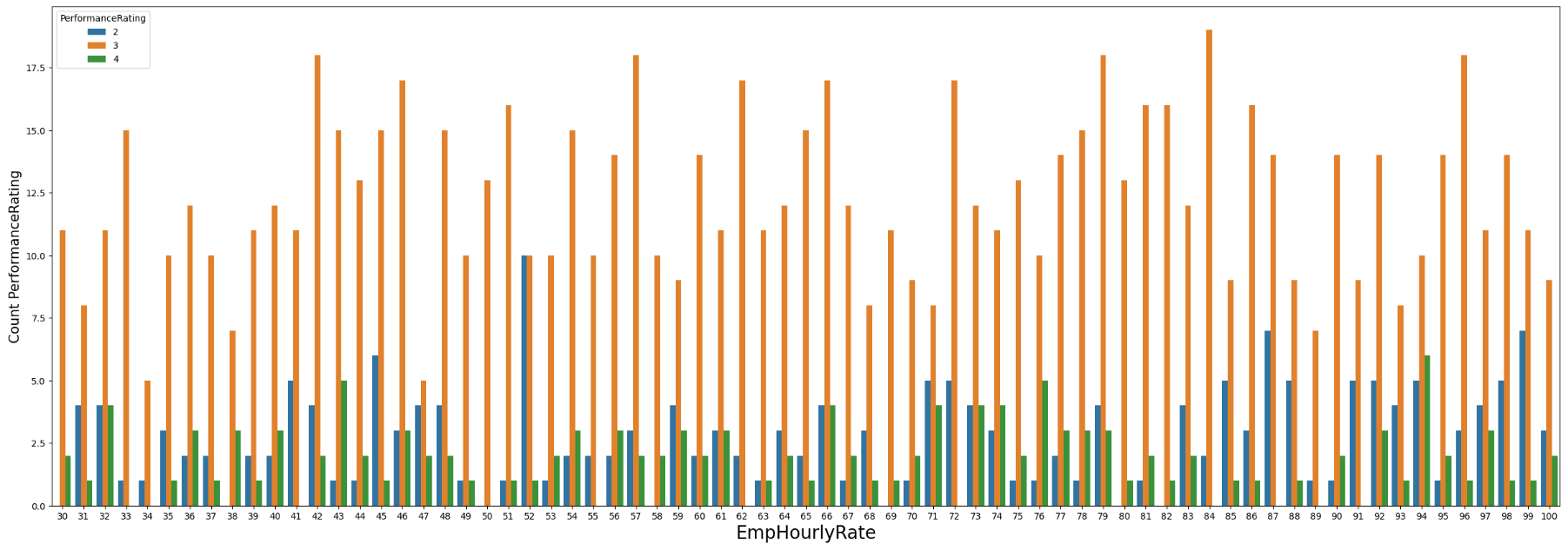
*Biểu diễn mối quan hệ của biến TrainingTimesLastYear với biến target*

**

*Biểu diễn mối quan hệ của biến ExperienceYearsAtThisCompany với biến target*

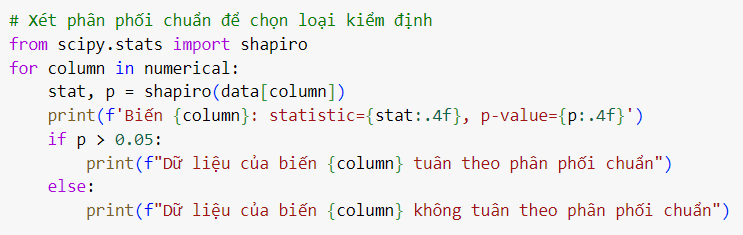
**

*Biểu diễn mối quan hệ của biến Age với biến target*

**

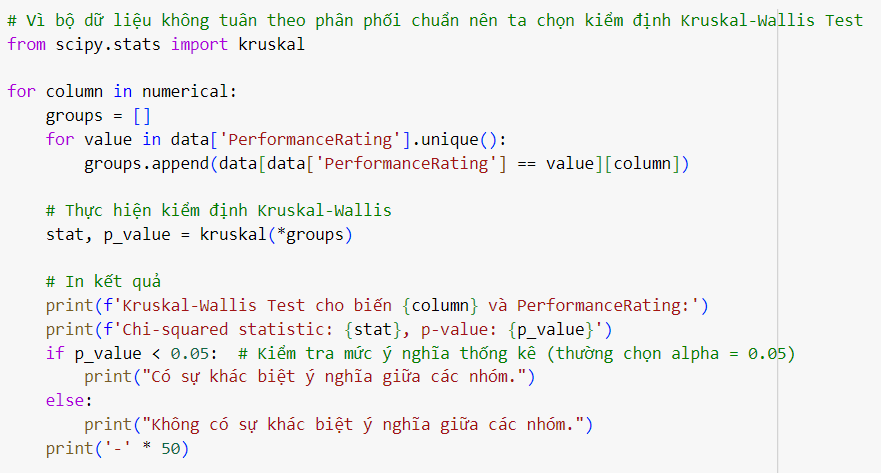
*Biểu diễn mối quan hệ của biến EmpHourlyRate với biến target*

* Đầu tiên, ta cần xét phân phối chuẩn để chọn loại kiểm định cho từng thuộc tính. Ta dùng vòng lặp để lược qua các cột, sau đó in ra chỉ số statistic và p-value của từng thuộc tính. Từ đó cho ra kết luận với các thuộc tính có tuân theo phân phối chuẩn hay không dựa trên điều kiện p-value phải lớn hơn alpha là 0.05.





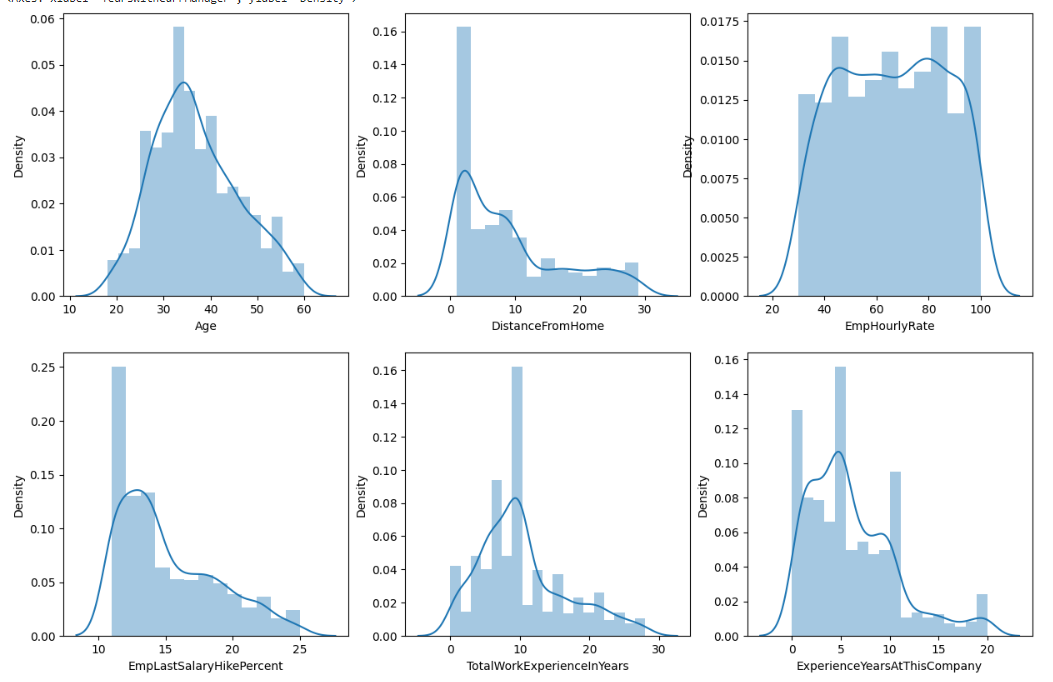
* Sau khi nhận thấy bộ dữ liệu không tuân theo phân phối chuẩn, ta chọn kiểm định Kruskal-Wallis Test để kiểm định các cặp biến.

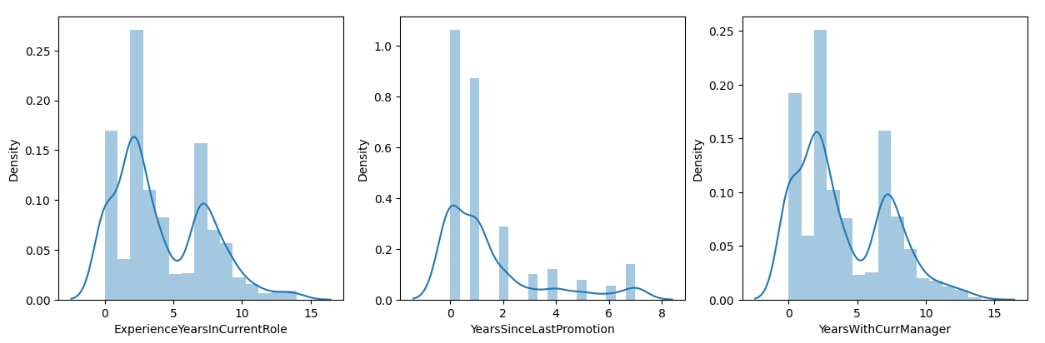




* Qua kiểm định Kruskal-Wallis ta rút ra được một số biến định lượng có ý nghĩa đến biến target PerformanceRating như sau:
* Biến EmpLastSalaryHikePercent và hiệu suất lao động (PerformanceRating) có mối tương quan đáng kể. Điều này thể hiện rằng những người nhận được mức tăng lương lớn hơn có khả năng có hiệu suất lao động tốt hơn.
* Tăng lương có thể tạo động lực và tác động tích cực đến thái độ làm việc của nhân viên. Tăng lương đáng kể có thể làm tăng cường cam kết và nỗ lực làm việc, dẫn đến hiệu suất lao động tốt hơn.
* Từ đó, có thể tham khảo để xây dựng chính sách lương cho nhân sự nhằm thúc đẩy hiệu suất làm việc của nhân viên.
* Khi xét biến số ToltalWorkExperienceInYears và biến target PerformanceRating ta nhận thấy được sự tương quan giữa kinh nghiệm và hiệu suất lao động:
* Mức độ kinh nghiệm làm việc theo năm có mối quan hệ ảnh hưởng đến hiệu suất làm việc của nhân viên. Những người có kinh nghiệm làm việc lâu dài thường có xu hướng có hiệu suất lao động tốt hơn so với những người có kinh nghiệm.
* Người có kinh nghiệm làm việc lâu dài thường tích lũy được nhiều kiến thức chuyên môn và kĩ năng trong lĩnh vực chuyên môn của họ. Điều này có thể ảnh hưởng tích cực đến khả năng thực hiện công việc.
* Sau khi có kết quả thống kê cần có những chính sách cung cấp chính sách và cơ hội để khuyến khích sự phát triển của nhân viên trong công ty.
* Mức độ kinh nghiệm làm việc tại công ty (ExperienceYearsAtThisCompany) có mối quan hệ ảnh hưởng đến hiệu suất làm việc của nhân viên (PerformanceRating):
* Tương tự, những người có nhiều kinh nghiệm làm việc tại công ty có hiệu suất lao động tốt hơn những người đang có ít kinh nghiệm
* Những người có kinh nghiệm tại công ty đã quen với môi trường làm việc, có kiến thức sâu về tổ chức và quy trình làm việcc. Chính vì vậy mà có tác động đến hiệu suất khi làm việc trong công ty.
* Từ kết luận này, có thể xem xét một số chính sách để thúc đẩy tinh thần làm việc của các nhân sự trong công ty.
* Tương tự với kinh nghiệm làm việc trong công ty và tổng số năm kinh nghiệm, thì mức độ kinh nghiệm trong vai trò hiện tại (ExperienceYearsInCurrentRole) có mối quan hệ đáng kể đến hiệu suất lao động (PerformanceRating). Có thể có xu hướng rằng những người có nhiều kinh nghiệm trong vai trò hiện tại có xu hướng có hiệu suất lao động tốt hơn.
* Người có nhiều kinh nghiệm trong vai trò hiện tại thường có độ thành thạo cao và hiểu biết sâu về công việc, quy trình và trách nhiệm của vai trò đó. Điều này có thể ảnh hưởng tích cực đến khả năng thực hiện và hiệu suất làm việc.
* Thời gian từ lần thăng tiến gần nhất (YearsSinceLastPromotion) có mối quan hệ đáng kể đến hiệu suất lao động (PerformanceRating). Có thể có xu hướng rằng những người đã trải qua thời gian lâu kể từ lần thăng tiến có hiệu suất lao động thấp hơn so với những người được thăng tiến gần đây hơn.
* Thời gian lâu kể từ lần thăng tiến có thể ảnh hưởng đến động lực và cam kết của nhân viên. Người không được thăng tiến trong một thời gian dài có thể cảm thấy ít được công nhận và động lực làm việc có thể giảm.
* Thời gian làm việc với người quản lý hiện tại (YearsWithCurrManager) có mối quan hệ đáng kể đến hiệu suất lao động (PerformanceRating). Có thể có xu hướng rằng những người đã làm việc lâu với người quản lý hiện tại có hiệu suất lao động tốt hơn so với những người làm việc với người quản lý trong thời gian ngắn.
* Thời gian làm việc lâu dài với người quản lý có thể xây dựng một mối quan hệ tốt giữa nhân viên và người quản lý, điều này có thể ảnh hưởng tích cực đến hiệu suất lao động và sự hỗ trợ trong công việc hàng ngày.
* Kết quả này có thể yêu cầu sự xem xét về vai trò của người quản lý và cách họ hỗ trợ và quản lý nhân viên. Có thể cần xem xét cách thức quản lý quan hệ và hỗ trợ từ người quản lý để tối ưu hóa hiệu suất lao động.
* **Kết luận:** sau khi trực quan và kiểm định thì có những biến sau có tương quan đến biến Target PerformanceRating là:
* Nominal: *OverTime, EmpDepartment, EmpJobRole*
* Ordinal: *EmpEnvironmentSatisfaction, EmpJobLevel, EmpWorkLifeBalance*
* Numerical: *EmpLastSalaryHikePercent, ToltalWorkExperienceInYears, ExperienceYearsAtThisCompany, ExperienceYearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager*
* Quan sát phân phối chuẩn của dữ liệu:





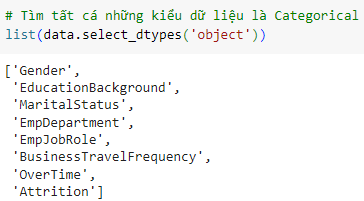


# **IV. Xử lý và chuẩn hóa dữ liệu để nâng cao mô hình được xây dựng**

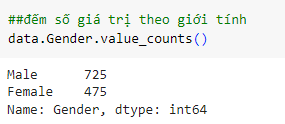
*Từ những phân tích về các biến có tương quan đến biến mục tiêu ‘PerformanceRating’ ta xây dựng mô hình dự đoán sự tác động giữa các biến phụ thuộc và biến độc lập cho bộ dữ liệu. Các mô hình có thể giúp hiểu các mẫu hoặc mối quan hệ phức tạp trong dữ liệu. Giúp chúng ta hiểu được sự tương quan chi tiết giữa các biến với nhau. Trong phần này, chúng ta sẽ xây dựng mô hình để dự đoán sự tương quan chi tiết giữa các biến được cho là có ảnh hưởng đến biến target ‘PerformanceRating’ được phân tích và biểu diễn trực quan.*

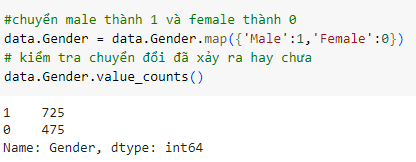
## Chuyển các thuộc tính phân loại sang dữ liệu số

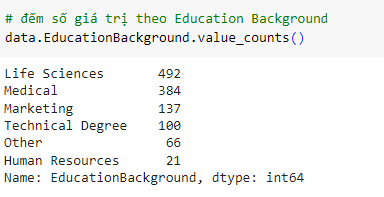
* Chuyển những biến phân loại (norminal categorical) thành biến số để có thể sử dụng PCA phục vụ cho việc tối ưu hóa mô hình
* Hiển thị tất cả biến nominal categirical để chuyển dạng dữ liệu:

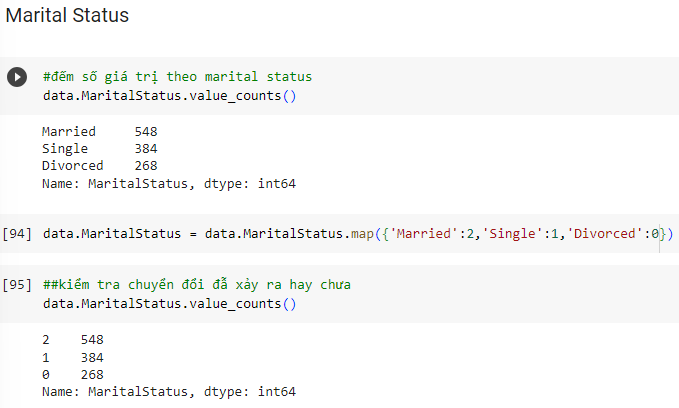


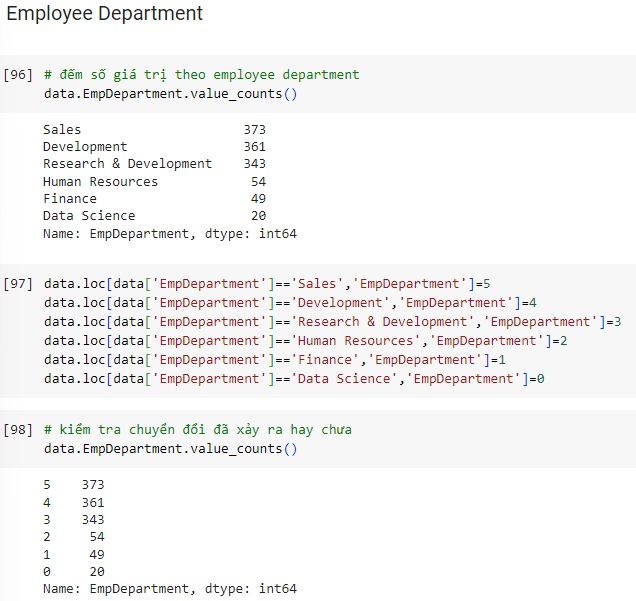
* Chuyển đổi những biến đã hiển thị ở trên thành biến số:

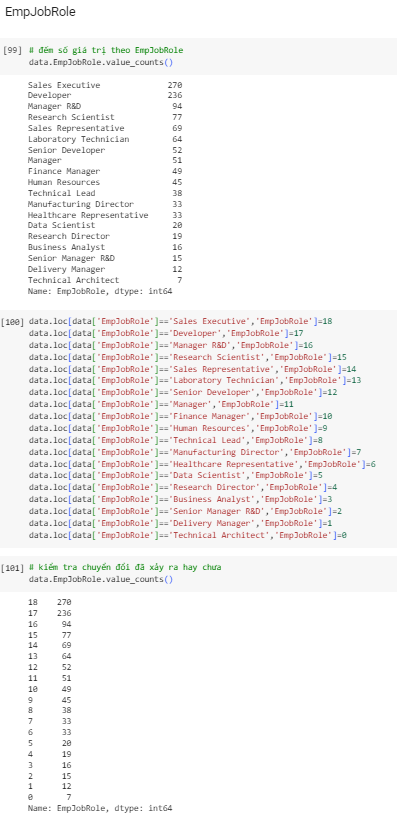




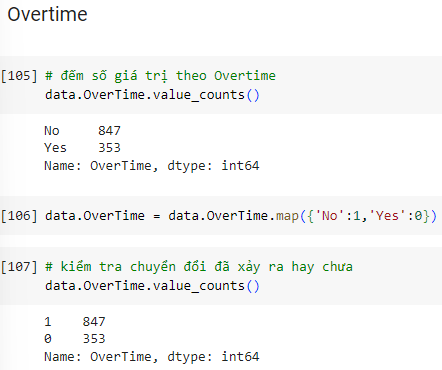


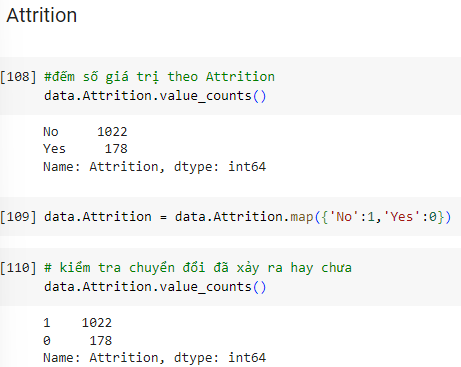








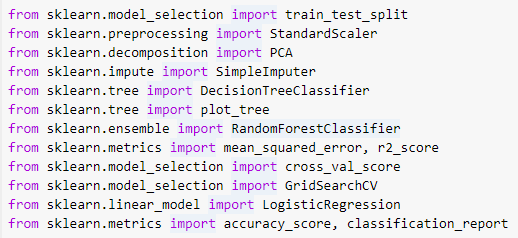




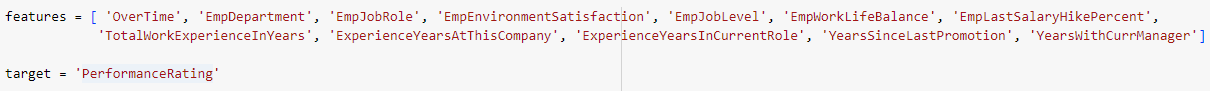
* Nhận xét: Cách làm chung khi chuyển đổi giá trị trị kiểu object sang number chính là đếm tổng từng giá trị có trong biến, sau đó ta gán cho từng giá trị theo số thứ tự. Cuối cùng là kiểm tra xem chuyển đổi kiểu dữ liệu đã xảy ra hay chưa.

## Sử dụng các thư viện cần thiết để chuẩn hóa và giảm chiều dữ liệu

* Import các thư viện cần thiết để xây dựng ba mô hình dự đoán cho biến target ‘PerformanceRating’ (ordinal categorical) là: Logistic Regression, Decision Tree Classifier và Random Forest Classifier.



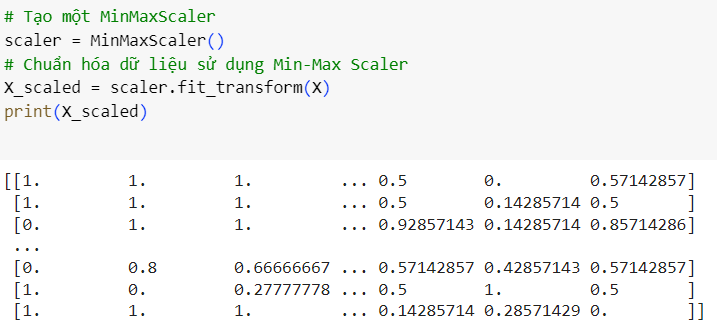
* Sau khi phân tích và trực quan hóa các biến để tìm ra những biến có tương quan đến biến target ta kết luận được có tất cả 12 biến có ảnh hưởng đến ‘PerformanceRating’. Nên ta sẽ xây dựng mô hình với features là 12 biến đã được phân tích và target là biến ‘PerformanceRating’. Để dễ dàng nhận xét trong các mô hình ta có quy ước như sau:
* Biến 1: OverTime
* Biến 2: EmpDepartment
* Biến 3: EmpJobRole
* Biến 4: EmpEnvironmentSatisfaction
* Biến 5: EmpJobLevel
* Biến 6: EmpWorkLifeBalance
* Biến 7: EmpLastSalaryHikePercent
* Biến 8: ToltalWorkExperienceInYears
* Biến 9: ExperienceYearsAtThisCompany
* Biến 10: ExperienceYearsInCurrentRole
* Biến 11: YearsSinceLastPromotion
* Biến 12: YearsWithCurrManager
* Tiếp theo ta tạo danh sách các biến độc lập (features) gồm 12 thuộc tính đã được phân tích và đánh giá là có ảnh hưởng đến biến mục tiêu và tạo biến target muốn dự đoán là ‘PerformanceRating’



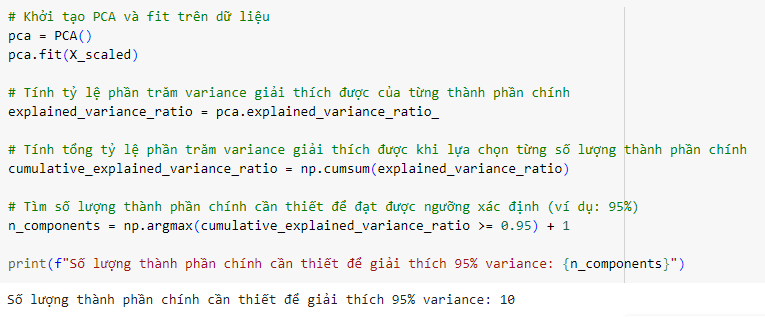
* Chọn các cột từ tập dữ liệu gốc (data) để tạo ra dữ liệu đầu vào (X) và dữ liệu đầu ra (y) cho mô hình hoặc phân tích.



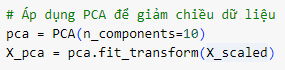
* Vì dữ liệu không tuân theo phân phối chuẩn nên ta sử dụng MinMaxScaler làm cho các biến có cùng phạm vi và đồng nhất, tăng tính ổn định và thuận tiện cho việc huấn luyện mô hình hoặc thực hiện các phân tích dữ liệu.



* *fit\_transform* được sử dụng để tính toán các giá trị cần thiết cho việc chuẩn hóa dữ liệu và áp dụng biến đổi lên dữ liệu X.
* Dùng PCA xác định số lượng thành phần chính cần thiết để giữ lại một tỷ lệ variance nhất định từ dữ liệu. Điều này giúp giảm số chiều dữ liệu một cách hiệu quả và lựa chọn số lượng thành phần chính phù hợp trong quá trình phân tích hoặc huấn luyện mô hình.



* Khởi tạo và fit mô hình PCA: Tạo một mô hình Phân tích thành phần chính (PCA) từ dữ liệu đã được chuẩn hóa (X\_scaled).
* Tính tỷ lệ phần trăm variance giải thích được: Dùng mô hình PCA đã fit để tính toán tỷ lệ phần trăm variance mà mỗi thành phần chính giải thích được từ dữ liệu.
* Tính toán tổng tích lũy của tỷ lệ phần trăm variance giải thích được từng thành phần chính. Điều này giúp hiểu rõ hơn về tỷ lệ phần trăm variance mà ta có thể giữ lại khi chọn một số lượng nhất định các thành phần chính.
* Tìm số lượng thành phần chính cần thiết để giải thích 95% variance: Từ tổng tỷ lệ phần trăm variance tính được từ mỗi thành phần chính, ta tìm số lượng thành phần chính cần thiết sao cho tổng tỷ lệ phần trăm variance giải thích được từ các thành phần này đạt ít nhất 95%.
* Sử dụng mô hình PCA để thực hiện phân tích thành phần chính với số lượng thành phần chính được xác định là 10. Mô hình tạo ra một ma trận mới (X\_pca) chứa dữ liệu đã được giảm số chiều xuống còn 10 thành phần chính được chọn từ dữ liệu gốc. Ma trận mới này sẽ giữ lại thông tin quan trọng nhất từ dữ liệu ban đầu và có số chiều ít hơn, tiện lợi cho việc huấn luyện mô hình hoặc phân tích dữ liệu.

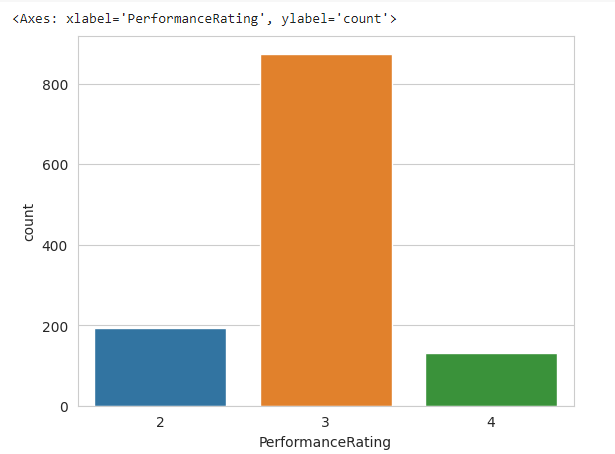


* *PCA(n\_components=10):* Khởi tạo một mô hình PCA với tham số n\_components được thiết lập là 10. Điều này chỉ định rằng sau quá trình phân tích, PCA sẽ giữ lại 10 thành phần chính từ dữ liệu.
* *pca.fit\_transform(X\_scaled):* Áp dụng phương thức fit\_transform() của mô hình PCA lên dữ liệu đã được chuẩn hóa (X\_scaled). Quá trình này tính toán các thành phần chính từ dữ liệu và biến đổi dữ liệu ban đầu thành một phiên bản mới với số lượng chiều được giảm xuống còn 10 chiều, tương ứng với 10 thành phần chính đã chọn.
* Chia dữ liệu thành các tập huấn luyện và kiểm tra (training set và testing set) để chuẩn bị dữ liệu cho quá trình huấn luyện và kiểm tra mô hình.



* *X\_pca:* Đây là ma trận chứa dữ liệu đã được giảm số chiều bằng phương pháp PCA, được chuẩn bị từ quá trình giảm chiều dữ liệu trước đó.
* *y:* Biến mục tiêu (target variable) hoặc nhãn tương ứng với dữ liệu.
* *test\_size=0.2:* Tham số này chỉ định tỷ lệ dữ liệu được chia cho tập kiểm tra. Ở đây, dữ liệu sẽ được chia thành 80% dữ liệu huấn luyện (X\_train, y\_train) và 20% dữ liệu kiểm tra (X\_test, y\_test).
* *random\_state=1:* Đây là một hằng số hoặc một số ngẫu nhiên được sử dụng để tạo ra các phân chia ngẫu nhiên nhưng có thể tái tạo được. Bằng cách sử dụng một số nguyên cụ thể, việc chia dữ liệu sẽ giống nhau mỗi khi bạn chạy mã này. Điều này giúp đảm bảo tính nhất quán khi chia dữ liệu thành các tập huấn luyện và kiểm tra.
* Vẽ countplot thể hiện số lượng các nhóm trong biến target để có thể đánh gía sau khi mô hình dự đoán





*Biểu đồ biểu diễn số lượng giá trị trong biến Performancerating*

* Từ trực quan hóa ta có thể thấy rằng số lượng giá trị ở nhóm 3: ‘Excellent’ là giá trị chiếm số lượng nhiều nhất, nên khi xây dựng mô hình dự đoán có thể mang độ chính xác cao hơn so với hai nhóm giá trị còn lại.

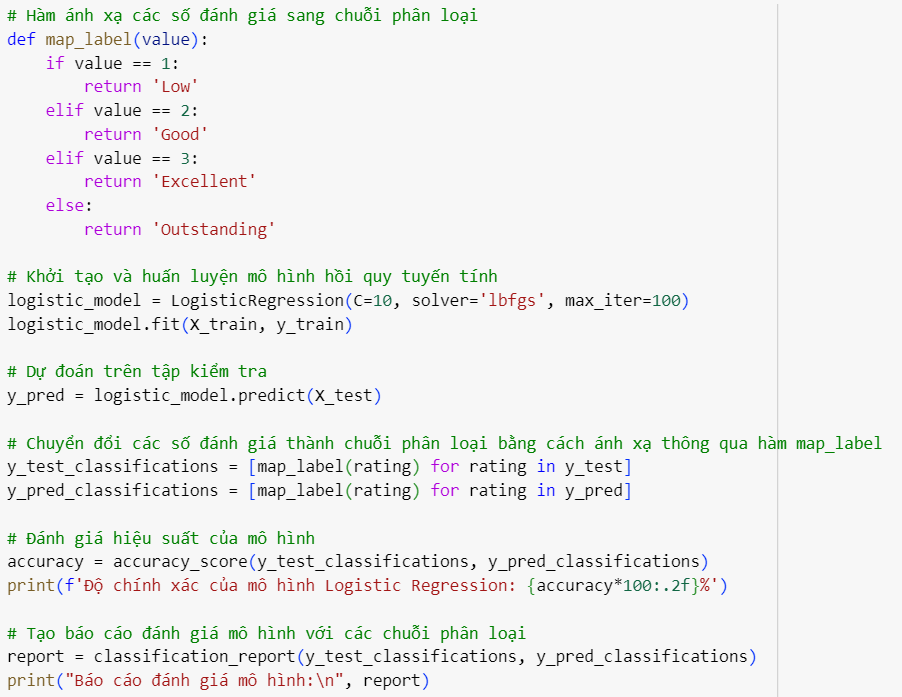
# **IV. Xây dựng mô hình dự đoán**

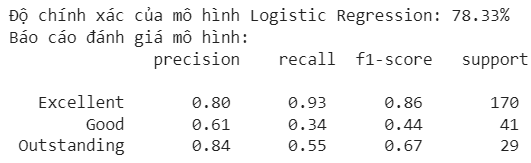
## Mô hình Logistic Regression:

* Thực hiện việc tìm kiếm siêu tham số tốt nhất cho mô hình Logistic Regression thông qua quá trình Grid Search

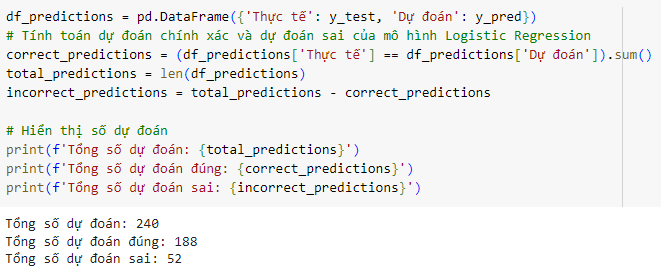


* *LogisticRegression(random\_state=42):* Đây là một mô hình hồi quy logistic được khởi tạo với một số tham số mặc định và một số giá trị c-ố định nhất định của các tham số không ảnh hưởng đến quá trình tinh chỉnh siêu tham số.
* *param\_grid:* Là một từ điển chứa các danh sách các giá trị tham số cần tinh chỉnh. Đối với mỗi tham số, GridSearchCV sẽ thử tất cả các giá trị có thể kết hợp được định nghĩa trong param\_grid.
* *GridSearchCV:* Là một công cụ tìm kiếm siêu tham số tự động, trong đó thực hiện kiểm định chéo (cross-validation) trên các tập con khác nhau của dữ liệu huấn luyện để đánh giá hiệu suất của từng bộ tham số.
* estimator: Mô hình sẽ được sử dụng để huấn luyện.
* param\_grid: Tập hợp các tham số cần tinh chỉnh.
* cv=5: Số lượng fold trong cross-validation, ở đây là 5-fold cross-validation.
* scoring='accuracy': Đánh giá hiệu suất mô hình dựa trên độ chính xác.
* *grid\_search.fit(X\_train, y\_train):* Tiến hành tìm kiếm siêu tham số tốt nhất bằng cách sử dụng GridSearchCV trên dữ liệu huấn luyện (X\_train, y\_train).
* *grid\_search.best\_params\_:* In ra bộ tham số tốt nhất tìm được thông qua Grid Search.
* *best\_model = grid\_search.best\_estimator\_:* Gán mô hình có siêu tham số tốt nhất vào biến best\_model.
* Huấn luyện một mô hình Logistic Regression với các siêu tham số đã chọn trước đó và đánh giá hiệu suất của mô hình trên tập kiểm tra dựa trên độ chính xác và báo cáo đánh giá chi tiết:

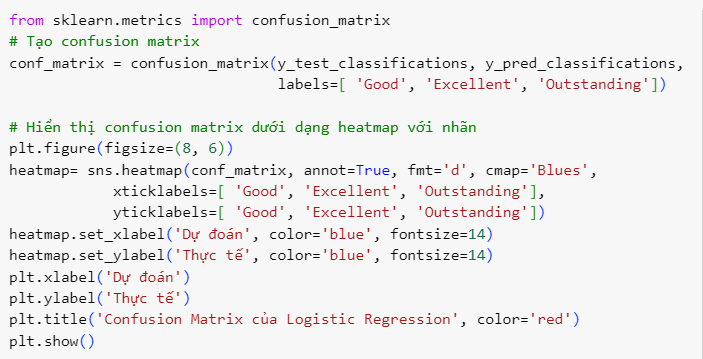


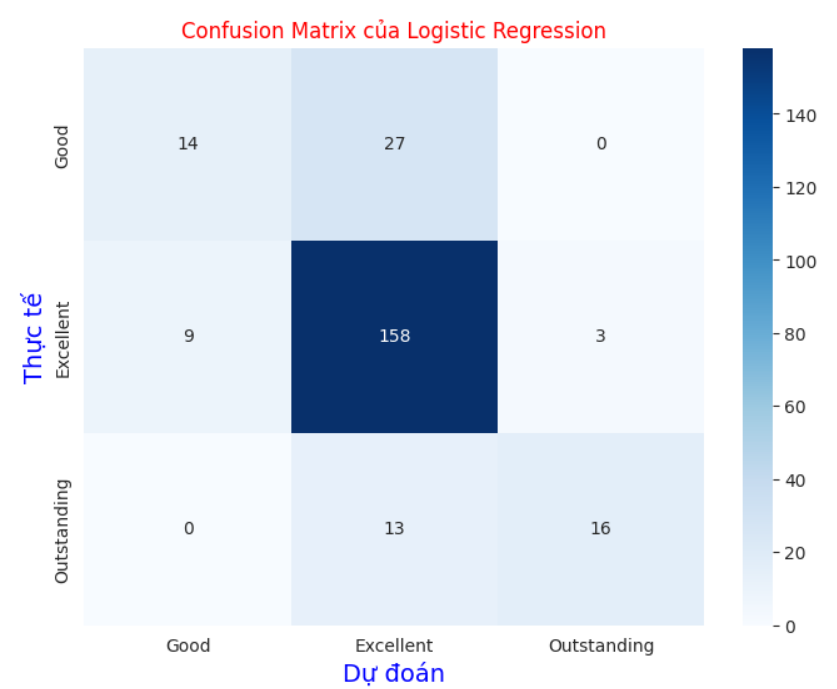


* Khởi tạo mô hình Logistic Regression: Một mô hình Logistic Regression được tạo với các siêu tham số nhất định: C=10 (tham số điều chuẩn), solver='lbfgs' (phương pháp tối ưu hóa), max\_iter=100 (số lần lặp tối đa).
* Huấn luyện mô hình: Dữ liệu huấn luyện (X\_train, y\_train) được sử dụng để huấn luyện mô hình Logistic Regression thông qua phương pháp fit().
* Dự đoán và đánh giá mô hình:
* *y\_pred\_logis\_reg = logistic\_model.predict(X\_test):* Mô hình được sử dụng để dự đoán nhãn trên tập dữ liệu kiểm tra (X\_test), và kết quả được lưu vào y\_pred\_logis\_reg.
* *accuracy = accuracy\_score(y\_test, y\_pred):* Tính toán độ chính xác của mô hình dựa trên dự đoán và nhãn thực tế trên tập kiểm tra.
* *print(f'Độ chính xác của mô hình Logistic Regression: {accuracy\*100:.2f}%'):* In ra độ chính xác của mô hình Logistic Regression trên tập kiểm tra.
* *report = classification\_report(y\_test, y\_pred):* Tạo báo cáo đánh giá mô hình dựa trên y\_test (nhãn thực tế trên tập kiểm tra) và y\_pred (nhãn dự đoán từ mô hình), trong đó bao gồm các thông tin như precision, recall, f1-score và support cho từng lớp nhãn.
* *print("Báo cáo đánh giá mô hình:\n", report):* In ra báo cáo đánh giá chi tiết của mô hình trên tập kiểm tra.
* **Nhận xét về chỉ số của mô hình:**
* Độ chính xác tổng thể (Accuracy) đạt khoảng 78.33%, nghĩa là mô hình dự đoán đúng khoảng 78.33% trên tổng số lượng mẫu trong tập kiểm tra.
* Precision, Recall và F1-score:
* Lớp Good: Precision đạt khoảng 61%, Recall đạt 34%, và F1-score đạt 44%. Mô hình có độ chính xác cao trong việc dự đoán các trường hợp thực sự là lớp Good (Precision), sự kết hợp của precision và recall có mức độ tốt khá, nhưng không cao.
* Lớp Excellent: mô hình có khả năng dự đoán lớp "Excellent" với mức độ chính xác (precision) là 80%, và nó cũng có khả năng bao quát rộng (recall) với 93% trên tất cả các trường hợp thực tế thuộc lớp "Excellent". F1-score đánh giá tổng thể khá cao (0.86), cho thấy mô hình có hiệu suất tốt trên lớp "Excellent".
* Lớp 4: mô hình có khả năng dự đoán lớp "Outstanding" với mức độ chính xác (precision) là 84%, nhưng recall chỉ đạt khoảng 55%. F1-score, mặc dù tốt (0.67), nhưng không cao như "Excellent" và vẫn có thể cần cải thiện. Điều này có thể ám chỉ rằng mô hình có thể dự đoán tốt một số trường hợp "Outstanding" nhưng không phải tất cả.
* Nhận xét: Mô hình có khả năng dự đoán tốt nhất cho lớp Excellent (Performance Rating 3), trong khi đó có kết quả dự đoán kém hơn cho lớp Good và lớp Outstanding.
* Tính toán độ chính xác của mô hình bằng cách in ra tổng số dự đoán đúng và số dự đoán sai của mô hình:



* Nhìn chung, mô hình có xu hướng dự đoán đúng nhiều hơn dự đoán sai trên tổng số dự đoán. Nhưng có thể thấy rằng tổng số dự đoán đúng và tổng số dự đoán sai khi cộng lại không đủ để tạo thành tổng số dự đoán. Điều này có thể bị ảnh hưởng bởi mất cân bằng trong tập dữ liệu (số lượng mẫu thuộc vào lớp 3 nhiều hơn hẳn so với lớp 2 và lớp 4) nên khi thực hiện dự đoán, mô hình có thể tập trung dự đoán vào các lớp có số lượng mẫu lớn hơn, dẫn đến số lượng dự đoán sai ở các lớp ít mẫu nhỏ hơn .
* Tạo ma trận nhầm lẫn (confusion matrix) từ các dự đoán được thực hiện và so sánh chúng với giá trị thực tế:





*Heatmap thể hiện sự số lượng các dự đoán được phân loại đúng hoặc sai theo từng nhãn dự đoán và thực tế từ mô hình Logistic Regression*

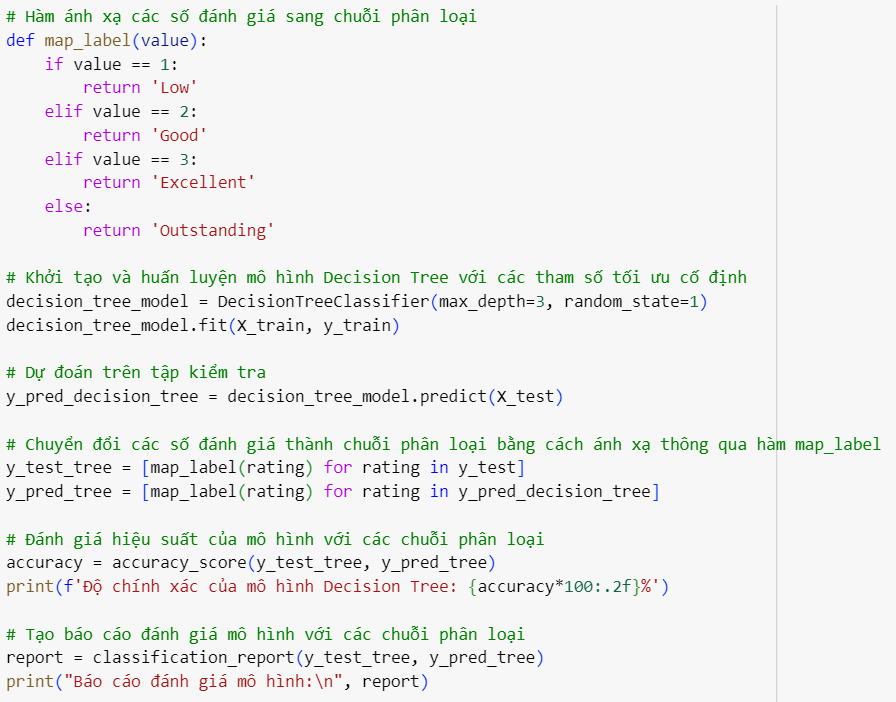
* Biểu đồ confusion matrix cho thấy mô hình dự đoán PerformanceRating của nhân viên trong công ty có độ chính xác khá tốt. Nhưng mô hình vẫn mắc lỗi như:
* Sai lầm loại 1 (false positive) có 9 trường hợp mà mô hình dự đoán là ‘Good’ nhưng thực tế là ‘Excellent’.
* Sai lầm loại 2 (false negative) có 27 trường hợp mô hình dự đoán là ‘Excellent’ nhưng thực tế là ‘Good’.

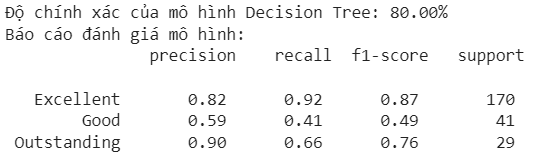
## Mô hình Decision Tree

* Tạo một GridSearchCV để tìm kiếm siêu tham số tốt nhất cho mô hình Decision Tree:



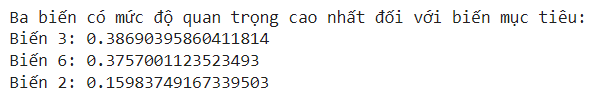
* Khởi tạo mô hình Decision Tree và tạo các tham số để thử nghiệm:
* Mô hình Decision Tree được khởi tạo với các tham số mà chúng ta muốn tìm kiếm giá trị tối ưu, bao gồm max\_depth và random\_state.
* *max\_depth* là độ sâu tối đa của cây quyết định. Các giá trị được thử nghiệm là [3, 5, 7, 9, None].
* *random\_state* là seed hoặc số ngẫu nhiên được sử dụng cho việc tạo cây quyết định. Các giá trị được thử nghiệm là [1, 42, 100].
* *GridSearchCV* được sử dụng để thử nghiệm tất cả các tổ hợp của các tham số được chỉ định để tìm ra bộ tham số tối ưu nhất cho mô hình.
* *cv=5* chỉ định cross-validation với số lượng fold là 5.
* *scoring='accuracy'* đánh giá hiệu suất của mô hình bằng độ chính xác.
* Fit dữ liệu:Mô hình được huấn luyện trên tập huấn luyện (X\_train và y\_train) với tất cả các tổ hợp của các tham số được chỉ định để tìm ra bộ tham số tốt nhất.
* In ra bộ tham số tốt nhất: {'max\_depth': 3, 'random\_state': 42}, nghĩa là mô hình Decision Tree tốt nhất được tìm thấy có max\_depth là 7 và random\_state là 42.
* Huấn luyện một mô hình Decision Tree với các tham số tối ưu cố định đã được chọn trước và in ra độ chính xác của mô hình Decision Tree trên tập kiểm tra và báo cáo đánh giá chi tiết của mô hình.





* Khởi tạo mô hình Decision Tree với các tham số tối ưu:
* max\_depth=3: Đây là độ sâu tối đa của cây quyết định, giới hạn độ phức tạp của cây.
* random\_state=1: Tham số này giúp đảm bảo rằng kết quả của mô hình sẽ ổn định và dự đoán được tái tạo.
* decision\_tree\_model.fit(X\_train, y\_train): Mô hình được huấn luyện trên tập dữ liệu huấn luyện (X\_train, y\_train). Mục tiêu là học cách phân loại dữ liệu để dự đoán nhãn mục tiêu.
* *y\_pred\_decision\_tree = decision\_tree\_model.predict(X\_test):* Mô hình được sử dụng để dự đoán nhãn trên tập dữ liệu kiểm tra (X\_test). Các dự đoán được lưu vào biến y\_pred\_decision\_tree.
* *accuracy\_score:* Tính độ chính xác của mô hình bằng cách so sánh giữa nhãn thực tế trên tập kiểm tra (y\_test) và nhãn mô hình dự đoán (y\_pred\_decision\_tree).
* *classification\_report:* In ra báo cáo đánh giá hiệu suất của mô hình dựa trên các chỉ số như precision, recall và f1-score.
* **Nhận xét:**
* Mô hình có độ chính xác tổng thể là 80.00%, tức là mô hình đưa ra dự đoán đúng cho khoảng 80.00% các trường hợp trong tập kiểm tra.
* Đối với lớp Excellent’:
* Precision ở mức 0.82, cho thấy tỷ lệ dự đoán đúng là "Excellent" trong số các dự đoán được gán nhãn là ‘Excellent’ là cao.
* Recall đạt 0.92, tức là mô hình có khả năng dự đoán đúng hầu hết các trường hợp thực tế thuộc lớp ‘Excellent’.
* F1-score là 0.87, kết hợp giữa precision và recall cho thấy mức độ hiệu suất tốt trên lớp này.
* Đối với lớp ‘Good’:
* Precision và recall đều thấp, lần lượt là 0.59 và 0.41, cho thấy mô hình không dự đoán đúng và bao quát đủ tốt trên lớp này.
* F1-score chỉ đạt 0.49, thấp hơn so với lớp ‘Excellent"’, cho thấy hiệu suất của mô hình trên lớp ‘Good’ chưa cao.
* Đối với lớp ‘Outstanding’:
* Precision cao (0.90) cho thấy mô hình dự đoán chính xác hầu hết các trường hợp thuộc lớp này.
* Recall đạt 0.66, không cao bằng precision, có thể mô hình đã bỏ sót một số trường hợp thực tế thuộc lớp ‘Outstanding’.
* F1-score ở mức 0.76, tương đối tốt nhưng vẫn có thể cần cải thiện.
* Sử dụng thuộc tính feature\_importances\_ của mô hình Decision Tree để hiển thị 3 biến có mức độ quan trọng nhất đối với biến target và vẽ biểu đồ để so sánh tổng thể giữa mức độ quan trọng của các biến. Các giá trị càng cao cho thấy đặc trưng đó càng quan trọng trong việc phân loại các mẫu dữ liệu.





* Mô hình Decision Tree dự đoán không tương thích với kết quả sau khi biểu diễn trực quan và kiểm định các thuộc tính có sự tương quan cao đến biến target là PerformanceRating

## Mô hình Random Forest

* Tìm kiếm siêu tham số tốt nhất cho mô hình Random Forest thông qua GridSearchCV:

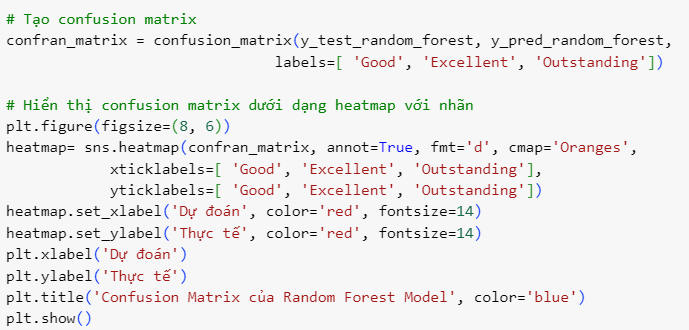


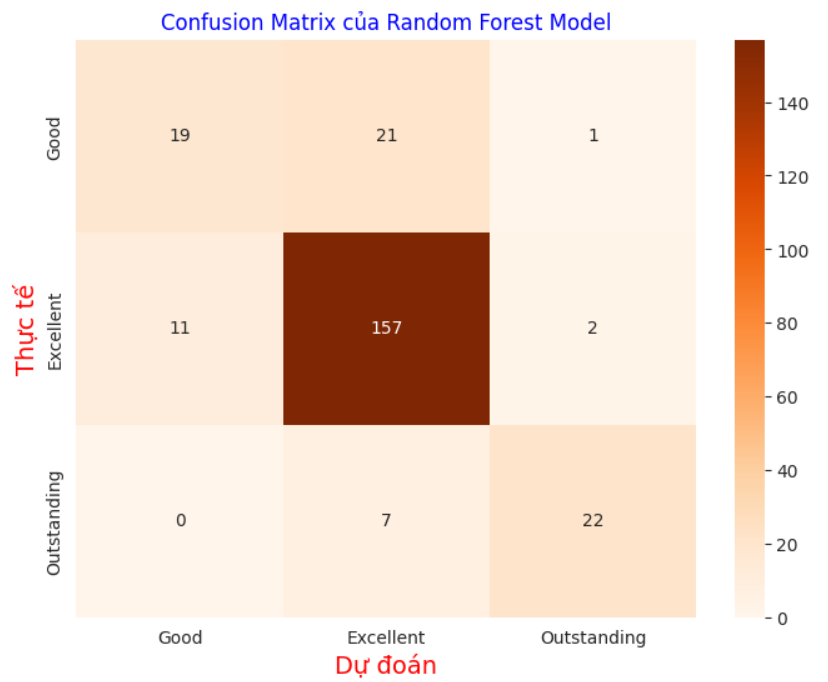
* *RandomForestClassifier()* khởi tạo một mô hình RandomForest với các giá trị mặc định.
* *param\_grid* là một từ điển chứa các giá trị mà chúng ta muốn thử nghiệm để tìm ra siêu tham số tốt nhất. Trong trường hợp này:
* 'n\_estimators': Số lượng cây quyết định trong mô hình Random Forest, thử nghiệm với các giá trị là 50, 100, và 150.
* 'max\_depth': Độ sâu tối đa của các cây quyết định, thử nghiệm với các giá trị là 5, 10, và 15.
* 'random\_state': Giá trị để cố định kết quả ngẫu nhiên, thử nghiệm với các giá trị là 42, 100, và 2023.
* *GridSearchCV* thực hiện tìm kiếm siêu tham số thông qua việc thử nghiệm các giá trị của param\_grid và sử dụng
* *cross-validation* với số fold là 5 (cv=5). Scoring được thiết lập thành 'accuracy' để đánh giá hiệu suất của mô hình dựa trên độ chính xác.
* grid\_search.fit(X\_train, y\_train) chạy quá trình tìm kiếm siêu tham số tốt nhất dựa trên tập dữ liệu huấn luyện X\_train và y\_train.
* *print("Bộ tham số tối ưu:", grid\_search.best\_params\_)* in ra bộ siêu tham số tốt nhất mà GridSearchCV đã tìm thấy sau khi huấn luyện mô hình. Đây là bộ tham số tối ưu mà cần sử dụng cho mô hình RandomForestClassifier.
* Huấn luyện một mô hình Random Forest trên dữ liệu huấn luyện và sau đó đánh giá độ chính xác và hiệu suất của mô hình trên tập kiểm tra:





* *RandomForestClassifier(n\_estimators=150, max\_depth=10, random\_state=42):* Tạo một mô hình Random Forest với các siêu tham số như sau:
* n\_estimators=150: Số lượng cây quyết định trong mô hình Random Forest là 150.
* max\_depth=10: Độ sâu tối đa của các cây quyết định là 10.
* random\_state=42: Sử dụng một giá trị cố định để đảm bảo kết quả tái lập được.
* random\_forest\_model.fit(X\_train, y\_train): Huấn luyện mô hình Random Forest trên dữ liệu huấn luyện X\_train và nhãn tương ứng y\_train.
* *y\_pred\_random\_forest = random\_forest\_model.predict(X\_test):* Dự đoán nhãn trên tập kiểm tra X\_test sử dụng mô hình đã được huấn luyện.
* *accuracy = accuracy\_score(y\_test, y\_pred\_random\_forest):* Tính toán độ chính xác của mô hình bằng cách so sánh nhãn dự đoán (y\_pred\_random\_forest) với nhãn thực tế trên tập kiểm tra (y\_test).
* *print(f'Độ chính xác của mô hình Random Forest: {accuracy\*100:.2f}%'):* In ra màn hình độ chính xác của mô hình Random Forest trên tập kiểm tra.
* *report = classification\_report(y\_test, y\_pred\_random\_forest):* Tạo báo cáo đánh giá hiệu suất của mô hình trên tập kiểm tra sử dụng các metric như precision, recall và f1-score cho từng lớp.
* *print("Báo cáo đánh giá mô hình:\n", report):* In báo cáo đánh giá hiệu suất của mô hình, bao gồm các metric precision, recall, f1-score và support cho từng lớp và tổng quan của toàn bộ mô hình.
* **Nhận xét:**
* Độ chính xác (Accuracy): Đạt được khoảng 82.50%, cho thấy tỷ lệ các dự đoán đúng trên tổng số mẫu trong tập kiểm tra.
* Lớp ‘Excellent’:
* Precision cao (0.85) chỉ ra rằng mô hình dự đoán đúng hầu hết các trường hợp khi được gán nhãn là "Excellent".
* Recall cao (0.92) cho thấy mô hình có khả năng bao quát rộng (recall cao) trên lớp này, tức là có ít trường hợp thực tế thuộc lớp "Excellent" mà mô hình bỏ sót.
* F1-score là 0.88, kết hợp giữa precision và recall, đạt mức cao, cho thấy hiệu suất tốt trên lớp này.
* Lớp ‘Good’:
* Precision và recall đều thấp hơn so với lớp "Excellent", lần lượt là 0.63 và 0.46. Điều này cho thấy mô hình không dự đoán đúng và bao quát đủ tốt trên lớp này.
* F1-score là 0.54, thấp hơn so với lớp "Excellent", cho thấy hiệu suất của mô hình trên lớp "Good" chưa cao.
* Lớp ‘Outstanding’:
* Precision cao (0.88) chỉ ra rằng mô hình dự đoán chính xác hầu hết các trường hợp thuộc lớp này.
* Recall đạt 0.76, không cao bằng precision, có thể mô hình đã bỏ sót một số trường hợp thực tế thuộc lớp "Outstanding".
* F1-score là 0.81, tương đối tốt nhưng cần cải thiện để tăng cường hiệu suất tổng thể của mô hình.
* Tổng thể, mô hình Random Forest đạt được hiệu suất tốt trên tập kiểm tra, với độ chính xác cao và có thể dự đoán các lớp dữ liệu khá chính xác.
* ***Kết luận:*** Mô hình Random Forest dự đoán khá chính xác những biến có tương quan cao đến biến target. Dự đoán của mô hình giống với kiểm định và biểu diễn trực quan các biến ở phần trên.
* Tạo confusion matrix để đánh giá cách mà mô hình Random Forest đã thực hiện dự đoán cho các nhãn:



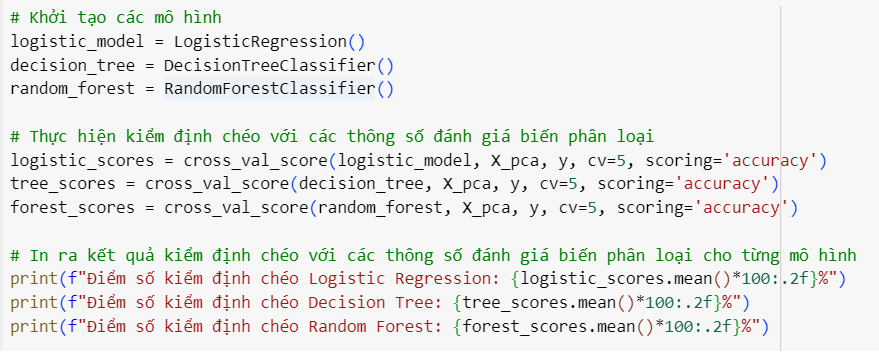


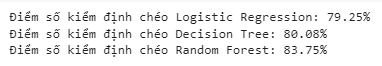
*Confusion Matrix đánh giá dự đoán từ mô hình và giá trị thực tế của Random Forest*

* Biểu đồ confusion matrix cho thấy mô hình dự đoán PerformanceRating của nhân viên trong công ty có độ chính xác khá tốt. Tuy nhiên, mô hình vẫn mắc lỗi:
* Sai lầm loại 1 (Type I Error) - False Positive: có 21 trường hợp dự đoán sai là ‘Good’ nhưng thực tế là ‘Excellent’ và có 7 trường hợp bị dự đoán sai ‘Good’ thực tế ‘Outstanding’
* Sai lầm loại 2 (False Negative) có 11 trường hợp lớp ‘Excellent’ bị dự đoán sai thành ‘Good’.

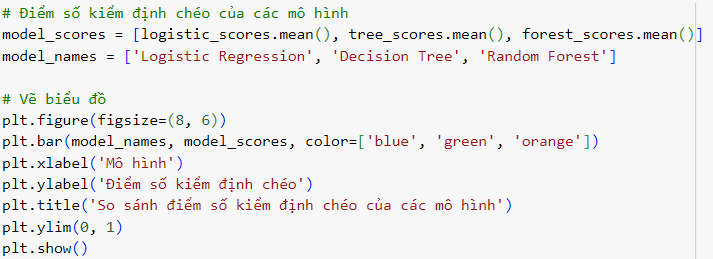
## Sử dụng phương pháp kiểm định chéo (cross-validation) kiểm định ba mô hình trên để chọn ra mô hình tối ưu nhất

* So sánh hiệu suất trung bình của ba mô hình trên dữ liệu cross-validation:





* *cross\_val\_score* từ thư viện sklearn.model\_selection được sử dụng để tính điểm số của mỗi mô hình trong quá trình cross-validation.
* X\_pca là dữ liệu đã được giảm chiều bằng phương pháp PCA và y là biến mục tiêu.
* cv=5 chỉ định số lượng fold (phần chia tập dữ liệu) trong quá trình cross-validation.
* scoring='accuracy' chỉ định việc đánh giá hiệu suất dựa trên độ chính xác của mô hình.
* cross\_val\_score trả về danh sách các điểm số của từng fold và mean() được sử dụng để tính trung bình của các điểm số này.
* **Nhận xét:**
* Logistic Regression: Điểm số kiểm định chéo đạt được là 79.25%. Điều này cho thấy mô hình Logistic Regression có khả năng dự đoán khá tốt trên dữ liệu kiểm tra khi được kiểm định chéo.
* Decision Tree: Điểm số kiểm định chéo đạt được là 80.08%. Điểm số này cao hơn so với Logistic Regression. Điều này có thể chỉ ra rằng mô hình Decision Tree hoạt động tốt nhất trên tập kiểm tra khi được kiểm định chéo.
* Random Forest: Điểm số kiểm định chéo đạt được là 83.75%. Đây là điểm số cao nhất trong ba mô hình, cho thấy Random Forest có khả năng dự đoán tốt nhất trên tập kiểm tra khi được kiểm định chéo.
* Nhìn chung, Random Forest có xu hướng cho kết quả tốt nhất trong ba mô hình khi được đánh giá bằng phương pháp kiểm định chéo trên tập dữ liệu đã chia.
* Tạo biểu đồ cột so sánh độ chính xác giữa ba mô hình: Logistic Regression, Decision Tree và Random Forest





*Biểu đồ cột so sánh độ chính xác giữa ba mô hình*

* **Kết luận:** Từ các kết quả trên ta có thể thấy rằng mô hình Random Forest có điểm số cao nhất trong số ba mô hình đã được đánh giá, với độ chính xác trung bình ở mức 83.75%. Điều này cho thấy rằng mô hình Random Forest có xu hướng dự đoán chính xác hơn trên tập dữ liệu chưa từng được sử dụng trong quá trình huấn luyện so với hai mô hình còn lại. Dựa trên độ chính xác trung bình từ quá trình kiểm định chéo, ta có thể kết luận rằng mô hình Random Forest có hiệu suất tốt nhất trong số ba mô hình này đối với tập dữ liệu.

# **V. Tổng kết**

Đồ án này nhằm mục đích phân tích và xây dựng mô hình dự đoán cho những yếu tố chính ảnh hưởng đến hiệu suất của nhân viên trong doanh nghiệp dựa trên các thông tin được thu thập từ công ty. Quá trình được thực hiện bao gồm các bước chính như sau:

* Tiền xử lý dữ liệu: Lựa chọn các biến quan trọng liên quan đến hiệu suất của nhân viên từ tập dữ liệu và thực hiện tiền xử lý như xử lý dữ liệu thiếu, chuẩn hóa các biến số, mã hóa biến phân loại nếu cần.
* Trực quan hóa dữ liệu
* Trực quan một biến: Sử dụng biểu đồ histogram, boxplot hoặc count plot để hiểu phân phối của các biến đơn lẻ, giúp nhận biết tính phân phối, outliers và các thông tin quan trọng của từng biến.
* Trực quan hai biến: Sử dụng heatmap hoặc scatter plot matrix để xem xét mức độ tương quan giữa các biến, đặc biệt với biến mục tiêu (PerformanceRating), từ đó sử dụng các phương pháp kiểm định để có thể đánh giá được mối quan hệ tương quan giữa các cặp biến và chọn lọc các biến quan trọng cho mô hình dự đoán.
* Xây dựng và đánh giá mô hình
* Xây dựng mô hình dự đoán: Sử dụng ba loại mô hình phổ biến là Logistic Regression, Decision Tree và Random Forest để dự đoán mức độ quan trọng của các biến với biến mục tiêu PerformanceRating.
* Đánh giá mô hình: Sử dụng kiểm định chéo (cross-validation) để đánh giá hiệu suất của mô hình, đánh giá các mô hình dự đoán và chọn ra mô hình có hiệu suất tốt nhất dựa trên các độ đo như accuracy, precision, recall, f1-score.

Sau quá trình phân tích, xây dựng và đánh giá mô hình, mô hình Random Forest được chọn là mô hình tốt nhất, đem lại độ chính xác cao nhất trong việc dự đoán mức độ hiệu suất của nhân viên (PerformanceRating). Mô hình này sử dụng các biến như EmpEnvironmentSatisfaction, EmpLastSalaryHikePercent, EmpWorkLifeBalance, và các biến khác và tính ra trọng số để cho ra kết quả những biến có độ ảnh hưởng cao đến biến target là ‘PerformanceRating’.

Nhìn chung, dự án đã biểu diễn, phân tích và xây dựng mô hình dự đoán hiệu suất nhân viên dựa trên dữ liệu được thu thập từ IABAC™, và kết quả từ mô hình Random Forest có thể được sử dụng để đưa ra những yếu tố ảnh hưởng nhiều đến hiệu suất làm việc. Từ biểu diễn và dự đoán này, bộ phận nhân sự có thể tham khảo để cải thiện chính sách nhân sự cũng như đưa ra những chính sách phù hợp cho nhân viên của công ty từ đó nâng cao hiệu suất giúp công ty hoạt động tốt thúc đẩy doanh thu.

Bảng phân chia công việc

| STT | MSSV | Họ và tên | Công việc | Mức độ hoàn thành |
| --- | --- | --- | --- | --- |
| 1 | 31211023184 | Nguyễn Hoàng Hà My | * Phân tích biến phân loại và biến target * Xây dựng mô hình * Làm mục xây dựng mô hình cho powerpoint * Làm word phần tổng quan và thống kê dữ liệu. * Trực quan hóa dữ liệu | 100% |
| 2 | 31211027661 | Nguyễn Tân Niên | * Trực quan 2 biến, powerpoint | 100% |
| 3 | 31201024506 | Nguyễn Hoàng Nhật Hồng Nguyên | * Thống kê dữ liệu * PCA * Feature section * Làm word, powerpoint phần thống kê dữ liệu | 100% |
| 4 | 31211027652 | Tất Diệu Ngân | * Tiền xử lí dữ liệu * PCA * Chuyển thuộc tính phân loại theo dữ liệu số * Chỉnh sửa file * tiền xử lí dữ liệu, PCA, chuyển thuộc tính phân loại theo dữ liệu số power point | 100% |
| 5 | 31211027659 | Đặng Yến Nhi | * Trực quan đơn biến (quantitative and qualitative) * Làm powper point cho mục trực quan đơn biến, giới thiệu đề tài, bộ dữ liệu. * Làm doc phần trực quan đơn biến. | 100% |

# **TÀI LIỆU THAM KHẢO**

1. Giáo trình và slide môn học “Biểu diễn trực quan dữ liệu” Đại học Kinh tế
2. Data Visualization - ĐẠI HỌC UEH TRƯỜNG KINH DOANH KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH Đề tài môn. (n.d.). Retrieved from <https://www.studocu.com/vn/document/truong-dai-hoc-kinh-te-thanh-pho-ho-chi-minh/giao-tiep-kinh-doanh/data-visualization/50106853?origin=home-recent-1>
3. Góc học tập. (n.d.). Retrieved from <https://kqtkd.duytan.edu.vn/Home/ArticleDetail/vn/88/4184/cac-phuong-phap-thuong-su-dung-de-kiem-dinh-mo-hinh-trong-nghien-cuu-dinh-luong>

\_HẾT\_