Human Resources Dataset Analysis

Project Overview

This project aims to analyse a human resources dataset to provide meaningful insights for organizational decision-making. The analysis includes data cleaning, question formulation, forecasting trends, and visualization, culminating in a final report.

Week 1: Build Data Model, Data Cleaning

Objectives:

- Develop a structured data model.
- Clean raw data for analysis.

Tasks:

1. **Data Cleaning:**
    a. Handle missing values, duplicates, and inconsistencies.
    b. Standardize data formats to ensure consistency.
2. **Data Pre-processing:**
    a. Perform transformations such as normalization and encoding.
    b. Validate data integrity for analysis.

Tools Used:

1. SQL
2. Python (pandas, Matplotlib)

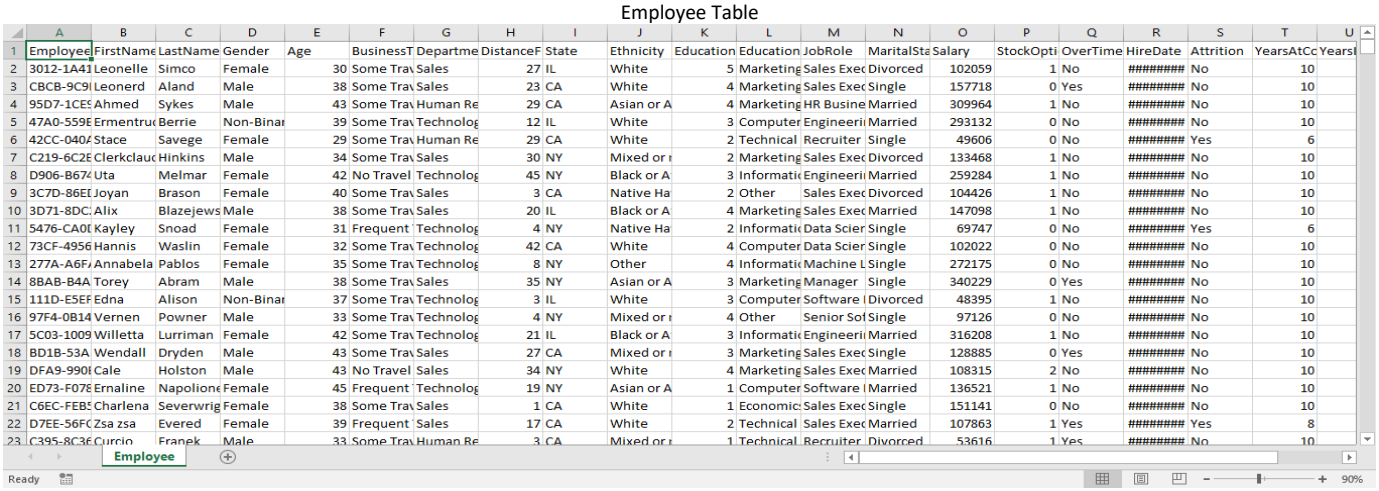
Steps Made To Reach Goals:

Step 1: Data Acquisition & Initial Assessment

We started with two Excel files:

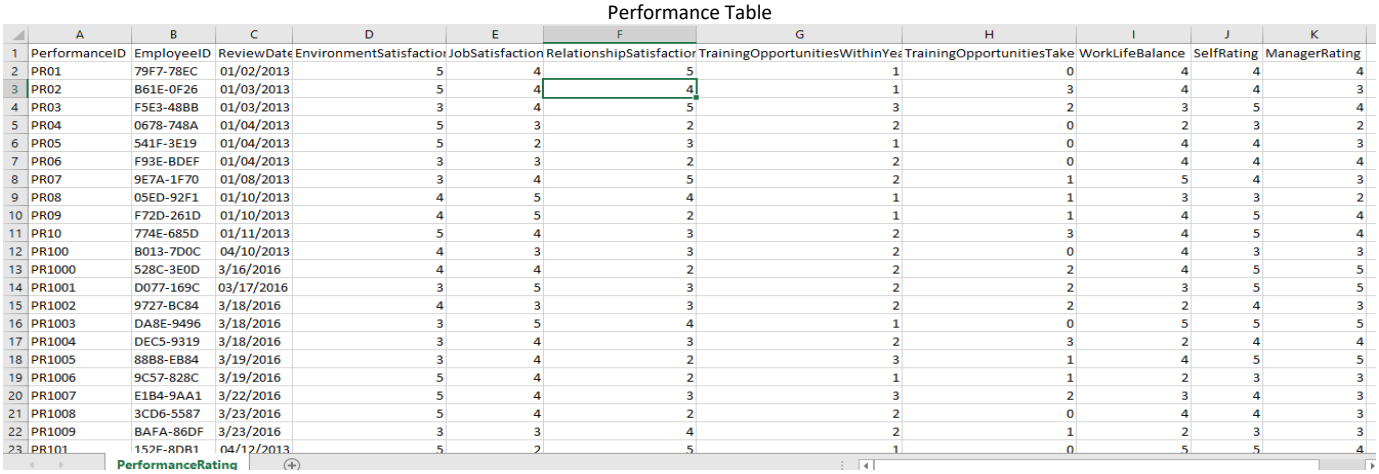
- Employees Data: Containing information about individual employees.
- Performance Data: Capturing various metrics related to employee efficiency.

Employee Table



Employee	FirstName	LastName	Gender	Age	BusinessT	Departme	DistanceF	State	Ethnicity	Education	Education	JobRole	MaritalSta	Salary	StockOpti	OverTime	HireDate	Attrition	YearsAtCc	Yearsl
3012-1A41	Leonelle	Simco	Female	30	Some Trav	Sales	27	IL	White	5	Marketing	Sales Exec	Divorced	102059	1	No	########	No	10	
CBCB-9C9	Leonerd	Aland	Male	38	Some Trav	Sales	23	CA	White	4	Marketing	Sales Exec	Single	157718	0	Yes	########	No	10	
95D7-1CE5	Ahmed	Sykes	Male	43	Some Trav	Human Re	29	CA	Asian or A	4	Marketing	HR Busine	Married	309964	1	No	########	No	10	
47A0-559E	Ermentrud	Berrie	Non-Binar	39	Some Trav	Technolog	12	IL	White	3	Computer	Engineeri	Married	293132	0	No	########	No	10	
42CC-040/	Stace	Savege	Female	29	Some Trav	Human Re	29	CA	White	2	Technical	Recruiter	Single	49606	0	No	########	Yes	6	
C219-6C2E	Clerkclauc	Hinkins	Male	34	Some Trav	Sales	30	NY	Mixed or r	2	Marketing	Sales Exec	Divorced	133468	1	No	########	No	10	
D906-B67/	Uta	Melmar	Female	42	No Travel	Technolog	45	NY	Black or A	3	Informatic	Engineeri	Married	259284	1	No	########	No	10	
3C7D-86EE	Joyan	Brason	Female	40	Some Trav	Sales	3	CA	Native Ha	2	Other	Sales Exec	Divorced	104426	1	No	########	No	10	
3D71-8DC	Alix	Blazejews	Male	38	Some Trav	Sales	20	IL	Black or A	4	Marketing	Sales Exec	Married	147098	1	No	########	No	10	
5476-CA0(	Kayley	Snoad	Female	31	Frequent	Technolog	4	NY	Native Ha	2	Informatic	Data Scier	Single	69747	0	No	########	Yes	6	
73CF-4956	Hannis	Waslin	Female	32	Some Trav	Technolog	42	CA	White	4	Computer	Data Scier	Single	102022	0	No	########	No	10	
277A-A6F/	Annabela	Pablos	Female	35	Some Trav	Technolog	8	NY	Other	4	Informatic	Machine L	Single	272175	0	No	########	No	10	
8BAB-B4A	Torey	Abram	Male	38	Some Trav	Sales	35	NY	Asian or A	3	Marketing	Manager	Single	340229	0	Yes	########	No	10	
111D-E5EF	Edna	Alison	Non-Binar	37	Some Trav	Technolog	3	IL	White	3	Computer	Software	Divorced	48395	1	No	########	No	10	
97F4-0B14	Vernen	Powner	Male	33	Some Trav	Technolog	4	NY	Mixed or r	4	Other	Senior Sof	Single	97126	0	No	########	No	10	
5C03-1009	Willetta	Lurriman	Female	42	Some Trav	Technolog	21	IL	Black or A	3	Informatic	Engineeri	Married	316208	1	No	########	No	10	
BD1B-53A	Wendall	Dryden	Male	43	Some Trav	Sales	27	CA	Mixed or r	3	Marketing	Sales Exec	Single	128885	0	Yes	########	No	10	
DFA9-990(	Cale	Holston	Male	43	No Travel	Sales	34	NY	White	4	Marketing	Sales Exec	Married	108315	2	No	########	No	10	
ED73-F078	Ernaline	Napolione	Female	45	Frequent	Technolog	19	NY	Asian or A	1	Computer	Software	Married	136521	1	No	########	No	10	
C6EC-FEB5	Charlena	Severwrig	Female	38	Some Trav	Sales	1	CA	White	1	Economic	Sales Exec	Single	151141	0	No	########	No	10	
D7EE-56F(	Zsa zsa	Evered	Female	39	Frequent	Sales	17	CA	White	2	Technical	Sales Exec	Married	107863	1	Yes	########	Yes	8	
C395-8C36	Curcio	Franek	Male	33	Some Trav	Human Re	3	CA	Mixed or r	1	Technical	Recruiter	Divorced	53616	1	Yes	########	No	10	

Performance Table



PerformanceID	EmployeeID	ReviewDate	EnvironmentSatisfactio	JobSatisfaction	RelationshipSatisfactior	TrainingOpportunitiesWithinYea	TrainingOpportunitiesTake	WorkLifeBalance	SelfRating	ManagerRating
PR01	79F7-78EC	01/02/2013	5	4	5	1	0	4	4	4
PR02	B61E-0F26	01/03/2013	5	4	4	1	3	4	4	3
PR03	F5E3-48BB	01/03/2013	3	4	5	3	2	3	5	4
PR04	0678-748A	01/04/2013	5	3	2	2	0	2	3	2
PR05	541F-3E19	01/04/2013	5	2	3	1	0	4	4	3
PR06	F93E-BDEF	01/04/2013	3	3	2	2	0	4	4	4
PR07	9E7A-1F70	01/08/2013	3	4	5	2	1	5	4	3
PR08	05ED-92F1	01/10/2013	4	5	4	1	1	3	3	2
PR09	F72D-261D	01/10/2013	4	5	2	1	1	4	5	4
PR10	774E-685D	01/11/2013	5	4	3	2	3	4	5	4
PR100	B013-7D0C	04/10/2013	4	3	3	2	0	4	3	3
PR1000	528C-3E0D	3/16/2016	4	4	2	2	2	4	5	5
PR1001	D077-169C	03/17/2016	3	5	3	2	2	3	5	5
PR1002	9727-BC84	3/18/2016	4	3	3	2	2	2	4	3
PR1003	DA8E-9496	3/18/2016	3	5	4	1	0	5	5	5
PR1004	DEC5-9319	3/18/2016	3	4	3	2	3	2	4	4
PR1005	88B8-EB84	3/19/2016	3	4	2	3	1	4	5	5
PR1006	9C57-828C	3/19/2016	5	4	2	1	1	2	3	3
PR1007	E1B4-9AA1	3/22/2016	5	4	3	3	2	3	4	3
PR1008	3CD6-5587	3/23/2016	5	4	2	2	0	4	4	3
PR1009	BAFA-86DF	3/23/2016	3	3	4	2	1	2	3	3
PR101	152F-8DB1	04/12/2013	5	2	5	1	0	5	5	4

However, we encountered several data integrity issues:

- Redundancy: Duplicate entries that needed to be eliminated.
- False Values: Erroneous or inconsistent values affecting accuracy.
- Null Values: Missing data points that required handling.

Step 2: Data Modelling

To enhance data structure, we used Python and SQL, utilizing libraries such as pandas for tabular manipulation and Matplotlib for visualization. We designed five separate Excel files, each representing:

1. Employees Table
2. Performance Table
3. Time Tracking Table
4. Rating Table
5. Satisfaction Metrics Table

Normalization:

- Data redundancy removal (e.g., eliminating duplicate employee records).

- Structuring data into multiple tables rather than keeping everything in one large, inefficient file.

Importing Main Libraries

```python
import pandas as pd

employee_df = pd.read_csv(r"C:\Users\dell\Downloads\Employee.csv")
performance_rating_df = pd.read_csv(r"C:\Users\dell\Downloads\PerformanceRating.csv")

print("Employee Data:")
print(employee_df.head())
print("\nPerformance Rating Data:")
print(performance_rating_df.head())
```

Modelling of Employee Table

```python
print("أول 5 صفوف من جدول Employee:")
print(employee_df.head())
print("\nمعلومات عامة عن الجدول:")
print(employee_df.info())
print("\nإحصائيات أساسية:")
print(employee_df.describe())

print("\nعدد القيم المفقودة في كل عمود:")
print(employee_df.isnull().sum())
print("\nنسبة القيم المفقودة في كل عمود:")
print(employee_df.isnull().mean() * 100)

if employee_df['BusinessTravel'].isnull().sum() > 0:
    employee_df['BusinessTravel'].fillna('Non-Travel', inplace=True)
if employee_df['Salary'].isnull().sum() > 0:
    employee_df['Salary'].fillna(employee_df['Salary'].mean(), inplace=True)
if employee_df['EmployeeID'].isnull().sum() > 0:
    employee_df.dropna(subset=['EmployeeID'], inplace=True)
print("\nعدد القيم المفقودة بعد المعالجة:")
print(employee_df.isnull().sum())

duplicates = employee_df.duplicated(subset=['EmployeeID']).sum()
print(f"\nعدد الصفوف المكررة بناءً على EmployeeID: {duplicates}")
if duplicates > 0:
    employee_df.drop_duplicates(subset=['EmployeeID'], keep='first', inplace=True)
    print("تم حذف الصفوف المكررة.")
print(f"عدد الصفوف بعد حذف التكرارات: {len(employee_df)}")

print("\nأنواع البيانات لكل عمود:")
print(employee_df.dtypes)
employee_df['HireDate'] = pd.to_datetime(employee_df['HireDate'], errors='coerce')
employee_df['Age'] = pd.to_numeric(employee_df['Age'], errors='coerce')
employee_df['Salary'] = pd.to_numeric(employee_df['Salary'], errors='coerce')
employee_df['Gender'] = employee_df['Gender'].astype(str)
employee_df['MaritalStatus'] = employee_df['MaritalStatus'].astype(str)
print("\nأنواع البيانات بعد التحويل:")
print(employee_df.dtypes)

print("\nإحصائيات Age:")
print(employee_df['Age'].describe())
outliers_age = employee_df[(employee_df['Age'] < 18) | (employee_df['Age'] > 100)]
print(f"عدد القيم الشاذة في Age: {len(outliers_age)}")
employee_df = employee_df[(employee_df['Age'] >= 18) & (employee_df['Age'] <= 100)]

print("\nإحصائيات Salary:")
print(employee_df['Salary'].describe())
outliers_salary = employee_df[(employee_df['Salary'] <= 0) | (employee_df['Salary'] > 1_000_000)]
print(f"عدد القيم الشاذة في Salary: {len(outliers_salary)}")
employee_df = employee_df[(employee_df['Salary'] > 0) & (employee_df['Salary'] <= 1_000_000)]

print("\nالقيم الفريدة في Gender:")
print(employee_df['Gender'].unique())
employee_df['Gender'] = employee_df['Gender'].str.lower().str.strip()
employee_df['Gender'] = employee_df['Gender'].replace({'m': 'male', 'f': 'female'})
```

```python
print(employee_df['Gender'].unique())

print("\nالقيم الفريدة في MaritalStatus:")
print(employee_df['MaritalStatus'].unique())
employee_df['MaritalStatus'] = employee_df['MaritalStatus'].str.lower().str.strip()
employee_df['MaritalStatus'] = employee_df['MaritalStatus'].replace({'single': 'single', 'married': 'married', 'divorced': 'divorced'})
print("القيم الفريدة في MaritalStatus بعد التنظيف:")
print(employee_df['MaritalStatus'].unique())

print("\nالقيم الفريدة في Department:")
print(employee_df['Department'].unique())
employee_df['Department'] = employee_df['Department'].str.lower().str.strip()

print("\nالقيم الفريدة في JobRole:")
print(employee_df['JobRole'].unique())
employee_df['JobRole'] = employee_df['JobRole'].str.lower().str.strip()

print("\nالقيم الفريدة في Education:")
print(employee_df['Education'].unique())
employee_df['Education'] = employee_df['Education'].astype(int)

employee_df.to_csv('employee_cleaned.csv', index=False)
print("\nتم حفظ الجدول بعد التنظيف باسم employee_cleaned.csv")
```

Python

```python
employee_dim = employee_df.copy()

employee_dim = employee_dim[[
    'EmployeeID', 'FirstName', 'LastName', 'Gender', 'Age', 'BusinessTravel',
    'DistanceFromHome (KM)', 'State', 'Ethnicity', 'Education', 'MaritalStatus',
    'Salary', 'StockOptionLevel', 'OverTime', 'HireDate', 'Attrition',
    'YearsAtCompany', 'YearsInMostRecentRole', 'YearsSinceLastPromotion',
    'YearsWithCurrManager', 'Department', 'JobRole'
]]

employee_dim.columns = [
    'EmployeeID', 'FirstName', 'LastName', 'Gender', 'Age', 'BusinessTravel',
    'DistanceFromHome', 'State', 'Ethnicity', 'Education', 'MaritalStatus',
    'Salary', 'StockOptionLevel', 'OverTime', 'HireDate', 'Attrition',
    'YearsAtCompany', 'YearsInMostRecentRole', 'YearsSinceLastPromotion',
    'YearsWithCurrManager', 'Department', 'JobRole'
]

print("EmployeeDim:")
print(employee_dim.head())
```

Python

Modelling Satisfaction and Rating Tables

```python
satisfaction_levels = pd.DataFrame({
    'SatisfactionID': [1, 2, 3, 4, 5],
    'SatisfactionLevel': ['1 Very Dissatisfied', '2 Dissatisfied', '3 Neutral', '4 Satisfied', '5 Very Satisfied']
})
print("SatisfactionDim:")
print(satisfaction_levels.head())

rating_levels = pd.DataFrame({
    'RatingID': [1, 2, 3, 4, 5],
    'RatingLevel': ['1 Poor', '2 Fair', '3 Good', '4 Very Good', '5 Excellent']
})
print("\nRatingDim:")
print(rating_levels.head())

performance_rating_df['ReviewDate'] = pd.to_datetime(performance_rating_df['ReviewDate'])
dates = performance_rating_df['ReviewDate'].drop_duplicates().sort_values()
time_dim = pd.DataFrame({
    'TimeID': range(1, len(dates) + 1),
    'Date': dates,
    'Day': dates.dt.day,
    'Month': dates.dt.month,
    'Quarter': dates.dt.quarter,
    'Year': dates.dt.year
})
print("\nTimeDim:")
print(time_dim.head())
```

Python

Modelling Performance Table

```python
performance_rating_fact = performance_rating_df.copy()

performance_rating_fact['ReviewDate'] = pd.to_datetime(performance_rating_fact['ReviewDate'])
performance_rating_fact = performance_rating_fact.merge(
    time_dim[['TimeID', 'Date']],
    left_on='ReviewDate',
    right_on='Date',
    how='left'
)
performance_rating_fact = performance_rating_fact.drop(columns=['ReviewDate', 'Date'])

performance_rating_fact = performance_rating_fact.merge(
    satisfaction_levels,
    left_on='EnvironmentSatisfaction',
    right_on='SatisfactionID',
    how='left'
)
performance_rating_fact = performance_rating_fact.drop(columns=['EnvironmentSatisfaction', 'SatisfactionLevel'])
performance_rating_fact.rename(columns={'SatisfactionID': 'EnvironmentSatisfactionID'}, inplace=True)

performance_rating_fact = performance_rating_fact.merge(
    satisfaction_levels,
    left_on='JobSatisfaction',
    right_on='SatisfactionID',
    how='left'
)
performance_rating_fact = performance_rating_fact.drop(columns=['JobSatisfaction', 'SatisfactionLevel'])
performance_rating_fact.rename(columns={'SatisfactionID': 'JobSatisfactionID'}, inplace=True)

performance_rating_fact = performance_rating_fact.merge(
    satisfaction_levels,
    left_on='RelationshipSatisfaction',
    right_on='SatisfactionID',
    how='left'
)
performance_rating_fact = performance_rating_fact.drop(columns=['RelationshipSatisfaction', 'SatisfactionLevel'])
performance_rating_fact.rename(columns={'SatisfactionID': 'RelationshipSatisfactionID'}, inplace=True)

performance_rating_fact = performance_rating_fact.merge(
    satisfaction_levels,
    left_on='WorkLifeBalance',
    right_on='SatisfactionID',
    how='left'
)
performance_rating_fact = performance_rating_fact.drop(columns=['WorkLifeBalance', 'SatisfactionLevel'])
performance_rating_fact.rename(columns={'SatisfactionID': 'WorkLifeBalanceID'}, inplace=True)

performance_rating_fact = performance_rating_fact.merge(
    rating_levels,
    left_on='SelfRating',
    right_on='RatingID',
    how='left'
)
performance_rating_fact = performance_rating_fact.drop(columns=['SelfRating', 'RatingLevel'])
performance_rating_fact.rename(columns={'RatingID': 'SelfRatingID'}, inplace=True)

performance_rating_fact = performance_rating_fact[[
    'PerformanceID', 'EmployeeID', 'TimeID', 'EnvironmentSatisfactionID',
    'JobSatisfactionID', 'RelationshipSatisfactionID', 'WorkLifeBalanceID',
    'SelfRatingID', 'TrainingOpportunitiesWithinYear', 'TrainingOpportunitiesTaken'
]]

print("PerformanceRatingFact:")
print(performance_rating_fact.head())
```

Python

Generate + Code + Markdown

Saving Data In New Files

```python
employee_dim.to_csv('employee_dim.csv', index=False)
satisfaction_levels.to_csv('satisfaction_dim.csv', index=False)
rating_levels.to_csv('rating_dim.csv', index=False)
time_dim.to_csv('time_dim.csv', index=False)
performance_rating_fact.to_csv('performance_rating_fact.csv', index=False)

print("All tables have been saved as CSV files.")
```

Python

```
from graphviz import Digraph

dot = Digraph(comment='Star Schema', format='png')
dot.attr(rankdir='TB')
dot.node('PerformanceRatingFact', '''PerformanceRatingFact
PerformanceID (PK)
EmployeeID (FK)
TimeID (FK)
EnvironmentSatisfactionID (FK)
JobSatisfactionID (FK)
RelationshipSatisfactionID (FK)
WorkLifeBalanceID (FK)
SelfRatingID (FK)
TrainingOpportunitiesWithinYear
TrainingOpportunitiesTaken''', shape='box', style='filled', fillcolor='lightcoral')

dot.node('EmployeeDim', '''EmployeeDim
EmployeeID (PK)
FirstName
LastName
Gender
Age
BusinessTravel
DistanceFromHome
State
Ethnicity
Education
MaritalStatus
Salary
StockOptionLevel
OverTime
HireDate
Attrition
YearsAtCompany
YearsInMostRecentRole
YearsSinceLastPromotion
YearsWithCurrManager
Department
JobRole''', shape='box', style='filled', fillcolor='lightyellow')

dot.node('SatisfactionDim', '''SatisfactionDim
SatisfactionID (PK)
SatisfactionLevel''', shape='box', style='filled', fillcolor='lightyellow')

dot.node('RatingDim', '''RatingDim
RatingID (PK)
RatingLevel''', shape='box', style='filled', fillcolor='lightyellow')

dot.node('TimeDim', '''TimeDim
TimeID (PK)
Date
Day
Month
Quarter
Year''', shape='box', style='filled', fillcolor='lightyellow')

dot.edge('PerformanceRatingFact', 'EmployeeDim', label='EmployeeID', color='blue')
dot.edge('PerformanceRatingFact', 'TimeDim', label='TimeID', color='blue')
dot.edge('PerformanceRatingFact', 'SatisfactionDim', label='EnvironmentSatisfactionID', color='blue')
dot.edge('PerformanceRatingFact', 'SatisfactionDim', label='JobSatisfactionID', color='blue')
dot.edge('PerformanceRatingFact', 'SatisfactionDim', label='RelationshipSatisfactionID', color='blue')
dot.edge('PerformanceRatingFact', 'SatisfactionDim', label='WorkLifeBalanceID', color='blue')
dot.edge('PerformanceRatingFact', 'RatingDim', label='SelfRatingID', color='blue')

dot.render('star_schema_updated', view=True)
```

Python

Step 3: Data Cleaning

Using Python, we tackled several issues:

- False Data Identification: Filtering incorrect entries based on predefined logic.
- Duplicate Removal: Ensuring unique entries for employees and performance records.
- Standardization: Unifying inconsistent case-sensitive entries (e.g., "Manager" vs. "manager" now treated as the same entity).

```python
import pandas as pd
import numpy as np
```

Python

```python
# مسار المجلد
folder_path = r"D:\Yousuf Gamal Eldeen\DEPI\Project\final"

# أسماء الملفات
file_names = [
    'department_dim.csv',
    'education_level_dim.csv',
    'employee_dim.csv',
    'job_role_dim.csv',
    'performance_rating_fact.csv',
    'rating_dim.csv',
    'satisfaction_dim.csv',
    'time_dim.csv'
]

# قراءة كل ملف في DataFrame
dataframes = [pd.read_csv(f"{folder_path}\\{file}") for file in file_names]

# تسمية الـ DataFrames بشكل فردي لو حابب تتعامل معاهم
department_dim = dataframes[0]
education_dim = dataframes[1]
employee_dim = dataframes[2]
jobrole_dim = dataframes[3]
performance_fact = dataframes[4]
rating_dim = dataframes[5]
satisfaction_dim = dataframes[6]
time_dim = dataframes[7]

# تنضيف لجدول Department
print("قبل التعديل", end="\n\n")
print("Number Of Rows Before Cleaning:", department_dim.shape[0])

# تعديل نوع الاعمده
department_dim["DepartmentID"] = department_dim["DepartmentID"].astype(int)
department_dim["DepartmentName"] = department_dim["DepartmentName"].astype(object)

# تنضيف القيم الناقصة
print("-----------------------------------")
print("Number Of null Values: ")
print(department_dim.isnull().sum())
department_dim.dropna(inplace= True)

# ازالة المسافات قبل و بعد الكلام و تكبير جميع الحروف
department_dim["DepartmentName"] = department_dim["DepartmentName"].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Each Value: ")
print(department_dim["DepartmentName"].value_counts())
department_dim.drop_duplicates(keep="first")

print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows After Cleaning:", department_dim.shape[0])
print("-----------------------------------")
print(department_dim)
```

```python
# تنضيف لجدول EducationLevel
print("قبل التعديل", end="\n\n")
print("Number Of Rows:", education_dim.shape[0])

# تعديل نوع الاعمده
education_dim["EducationLevelID"] = education_dim["EducationLevelID"].astype(int)
education_dim["EducationLevel"] = education_dim["EducationLevel"].astype(object)

# تنضيف القيم الناقصه
print("-----------------------------------")
print("Number Of null Values: ")
print(education_dim.isnull().sum())
education_dim.dropna(inplace= True)

# ازاله المسافات قبل و بعد الكلام و تكبير جميع الحروف
education_dim["EducationLevel"] = education_dim["EducationLevel"].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Each Value: ")
print(education_dim["EducationLevel"].value_counts())
education_dim.drop_duplicates(keep="first")

print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows:", education_dim.shape[0])
print("-----------------------------------")
print(education_dim)
```

```python
# تنضيف لجدول Job_Role
print("قبل التعديل", end="\n\n")
print("Number Of Rows:", jobrole_dim.shape[0])

# تعديل نوع الاعمده
jobrole_dim["JobRoleID"] = jobrole_dim["JobRoleID"].astype(int)
jobrole_dim["JobRoleName"] = jobrole_dim["JobRoleName"].astype(object)

# تنضيف القيم الناقصه
print("-----------------------------------")
print("Number Of null Values: ")
print(jobrole_dim.isnull().sum())
jobrole_dim.dropna(inplace= True)

# ازاله المسافات قبل و بعد الكلام و تكبير جميع الحروف
jobrole_dim["JobRoleName"] = jobrole_dim["JobRoleName"].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Each Value: ")
print(jobrole_dim["JobRoleName"].value_counts())
jobrole_dim.drop_duplicates(keep="first")

print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows:", jobrole_dim.shape[0])
print("-----------------------------------")
print(jobrole_dim)
```

```python
# تنضيف لجدول Rating
print("قبل التعديل", end="\n\n")
print("Number Of Rows:", rating_dim.shape[0])

# تعديل نوع الاعمده
rating_dim["RatingID"] = rating_dim["RatingID"].astype(int)
rating_dim["RatingLevel"] = rating_dim["RatingLevel"].astype(object)

# تنضيف القيم الناقصه
print("-----------------------------------")
print("Number Of null Values: ")
print(rating_dim.isnull().sum())
rating_dim.dropna(inplace= True)

# ازاله المسافات قبل و بعد الكلام و تكبير جميع الحروف
rating_dim["RatingLevel"] = rating_dim["RatingLevel"].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Each Value: ")
print(rating_dim["RatingLevel"].value_counts())
rating_dim.drop_duplicates(keep="first")


print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows:", rating_dim.shape[0])
print("-----------------------------------")
print(rating_dim)

# تنضيف لجدول Satisfaction
print("قبل التعديل", end="\n\n")
print("Number Of Rows:", satisfaction_dim.shape[0])

# تعديل نوع الاعمده
satisfaction_dim["SatisfactionID"] = satisfaction_dim["SatisfactionID"].astype(int)
satisfaction_dim["SatisfactionLevel"] = satisfaction_dim["SatisfactionLevel"].astype(object)

# تنضيف القيم الناقصه
print("-----------------------------------")
print("Number Of null Values: ")
print(satisfaction_dim.isnull().sum())
satisfaction_dim.dropna(inplace= True)

# ازاله المسافات قبل و بعد الكلام و تكبير جميع الحروف
satisfaction_dim["SatisfactionLevel"] = satisfaction_dim["SatisfactionLevel"].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Each Value: ")
print(satisfaction_dim["SatisfactionLevel"].value_counts())
satisfaction_dim.drop_duplicates(keep="first")


print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows:", satisfaction_dim.shape[0])
print("-----------------------------------")
print(satisfaction_dim)
```

```python
# تنضيف لجدول Employee
print("قبل التعديل", end="\n\n")
print("Number Of Rows:", employee_dim.shape[0])


# تعديل نوع الاعمده
cols = [
"EmployeeID", "FirstName", "LastName", "Gender", "Age", "DistanceFromHome", "State", "Ethnicity", "EducationField",
"MaritalStatus", "Salary", "StockOptionLevel", "OverTime", "Attrition", "YearsAtCompany", "YearsInMostRecentRole",
"YearsSinceLastPromotion", "YearsWithCurrManager", "DepartmentID", "JobRoleID", "EducationLevelID"
]

types =[
object, object, object, object, int, int, object, object, object, object, int, int, object, object, int,
int, int, int, int, int, int
]

for i in range(len(cols)):
    employee_dim[cols[i]] = employee_dim[cols[i]].astype(types[i])


emp_hiredate = pd.to_datetime(employee_dim["HireDate"], format="%Y-%m-%d")
employee_dim["HireDate"] = emp_hiredate.dt.strftime("%d-%m-%Y")

employee_W_time = pd.merge(employee_dim["HireDate"], time_dim[["TimeID", "Date"]], "left", left_on="HireDate", right_on="Date")

employee_dim["HireDate"] = employee_W_time["TimeID"]
employee_dim = employee_dim.rename(columns={"HireDate" : "HireDateID"})

# تنضيف القيم الناقصه
print("-----------------------------------")
Click to add a breakpoint
# تنضيف القيم الناقصة
print("-----------------------------------")
print("Number Of null Values: ")
print(employee_dim.isnull().sum())
employee_dim.dropna(inplace= True)

# ازاله المسافات قبل و بعد الكلام و تكبير جميع الحروف
for i in range(len(cols)):
    if (types[i] != object) :
        continue
    employee_dim[cols[i]] = employee_dim[cols[i]].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Unique ID:", len(employee_dim["EmployeeID"].unique()))
employee_dim.drop_duplicates(keep="first")

print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows:", employee_dim.shape[0])
```

```python
# تنضيف لجدول Performance
print("قبل التعديل", end="\n\n")
print("Number Of Rows:", performance_fact.shape[0])
# تعديل نوع الاعمده
cols = [
"PerformanceID", "EmployeeID", "TimeID", "EnvironmentSatisfactionID", "JobSatisfactionID",
"RelationshipSatisfactionID", "WorkLifeBalanceID", "SelfRatingID",
"TrainingOpportunitiesWithinYear", "TrainingOpportunitiesTaken"
]

types =[object, object, int, int, int, int, int, int, int, int]
for i in range(len(cols)):
    performance_fact[cols[i]] = performance_fact[cols[i]].astype(types[i])
# تنضيف القيم الناقصه
print("-----------------------------------")
print("Number Of null Values: ")
print(performance_fact.isnull().sum())
performance_fact.dropna(inplace= True)

# ازاله المسافات قبل و بعد الكلام و تكبير جميع الحروف
for i in range(len(cols)):
    if (types[i] != object) :
        continue
    performance_fact[cols[i]] = performance_fact[cols[i]].str.strip().str.capitalize()

# تنضيف القيم المكرره
print("-----------------------------------")
print("Number Of Unique PerformanceID:", len(performance_fact["PerformanceID"].unique()))
performance_fact.drop_duplicates(keep="first")

print("-----------------------------------")
print("بعد التعديل", end="\n\n")
print("Number Of Rows:", performance_fact.shape[0])
```

Step 4: Final Deliverables

At the end, we produced:

- Five cleaned and structured Excel files ready for further analysis.
- An SQL database organizing these tables efficiently.
- Graphical schema representation to clarify table relationships.

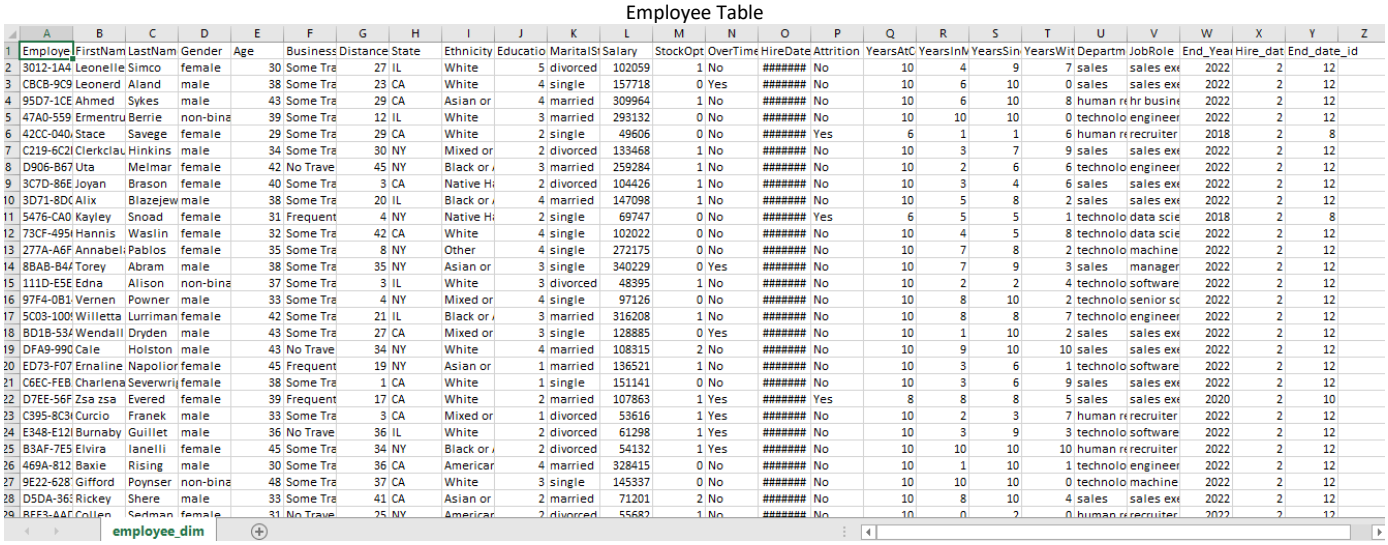
Rating Table

	A	B	C
1	RatingID	RatingLevel	
2	1	Poor	
3	2	Fair	
4	3	Good	
5	4	Very Good	
6	5	Excellent	
7			
8			
9			

Satisfaction Table

	A	B	C
1	SatisfactionID	SatisfactionLevel	
2	1	Very Dissatisfied	
3	2	Dissatisfied	
4	3	Neutral	
5	4	Satisfied	
6	5	Very Satisfied	
7			
8			
9			
10			

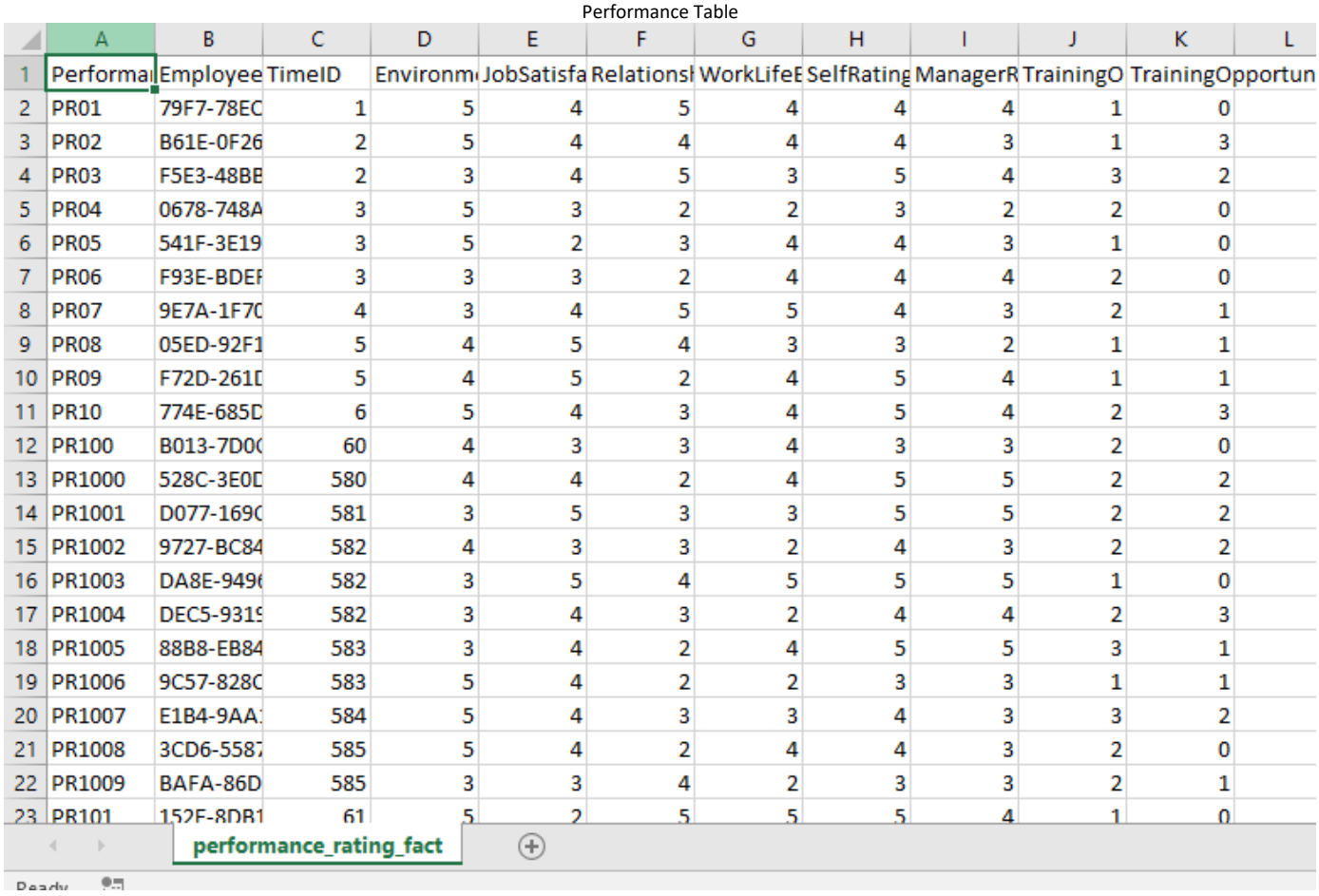
Employee Table



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Employe	FirstNam	LastNam	Gender	Age	Business	Distance	State	Ethnicity	Educatio	MaritalSt	Salary	StockOpt	OverTime	HireDate	Attrition	YearsAtC	YearsInV	YearsSin	YearsWit	Departm	JobRole	End_Yea	Hire_dat	End_date_id	
2	3012-1A4	Leonelle	Simco	female	30	Some Tra	27	IL	White	5	divorced	102059	1	No	#######	No	10	4	9	7	sales	sales exe	2022	2	12	
3	CBCB-9C9	Leonerd	Aland	male	38	Some Tra	23	CA	White	4	single	157718	0	Yes	#######	No	10	6	10	0	sales	sales exe	2022	2	12	
4	95D7-1CE	Ahmed	Sykes	male	43	Some Tra	29	CA	Asian or	4	married	309964	1	No	#######	No	10	6	10	8	human r	hr busine	2022	2	12	
5	47A0-559	Ermentru	Berrie	non-bina	39	Some Tra	12	IL	White	3	married	293132	0	No	#######	No	10	10	10	0	technolo	engineer	2022	2	12	
6	42CC-040	Stace	Savege	female	29	Some Tra	29	CA	White	2	single	49606	0	No	#######	Yes	6	1	1	6	human r	recruiter	2018	2	8	
7	C219-6C2	Clerkclau	Hinkins	male	34	Some Tra	30	NY	Mixed or	2	divorced	133468	1	No	#######	No	10	3	7	9	sales	sales exe	2022	2	12	
8	D906-B67	Uta	Melmar	female	42	No Trave	45	NY	Black or	3	married	259284	1	No	#######	No	10	2	6	6	technolo	engineer	2022	2	12	
9	3C7D-86E	Joyan	Brason	female	40	Some Tra	3	CA	Native Hi	2	divorced	104426	1	No	#######	No	10	3	4	6	sales	sales exe	2022	2	12	
10	3D71-8DC	Alix	Blazejew	male	38	Some Tra	20	IL	Black or	4	married	147098	1	No	#######	No	10	5	8	2	sales	sales exe	2022	2	12	
11	5476-CA0	Kayley	Snoad	female	31	Frequent	4	NY	Native Hi	2	single	69747	0	No	#######	Yes	6	5	5	1	technolo	data scie	2018	2	8	
12	73CF-495	Hannis	Waslin	female	32	Some Tra	42	CA	White	4	single	102022	0	No	#######	No	10	4	5	8	technolo	data scie	2022	2	12	
13	277A-A6F	Annabell	Pablos	female	35	Some Tra	8	NY	Other	4	single	272175	0	No	#######	No	10	7	8	2	technolo	machine	2022	2	12	
14	8BAB-B4A	Torey	Abram	male	38	Some Tra	35	NY	Asian or	3	single	340229	0	Yes	#######	No	10	7	9	3	sales	manager	2022	2	12	
15	111D-E5E	Edna	Alison	non-bina	37	Some Tra	3	IL	White	3	divorced	48395	1	No	#######	No	10	2	2	4	technolo	software	2022	2	12	
16	97F4-0B1	Vernen	Powner	male	33	Some Tra	4	NY	Mixed or	4	single	97126	0	No	#######	No	10	8	10	2	technolo	senior sc	2022	2	12	
17	5C03-100	Willetta	Lurriman	female	42	Some Tra	21	IL	Black or	3	married	316208	1	No	#######	No	10	8	8	7	technolo	engineer	2022	2	12	
18	BD1B-53A	Wendall	Dryden	male	43	Some Tra	27	CA	Mixed or	3	single	128885	0	Yes	#######	No	10	1	10	2	sales	sales exe	2022	2	12	
19	DFA9-990	Cale	Holston	male	43	No Trave	34	NY	White	4	married	108315	2	No	#######	No	10	9	10	10	sales	sales exe	2022	2	12	
20	ED73-F07	Ernaline	Napolior	female	45	Frequent	19	NY	Asian or	1	married	136521	1	No	#######	No	10	3	6	1	technolo	software	2022	2	12	
21	C6EC-FEB	Charlena	Severwri	female	38	Some Tra	1	CA	White	1	single	151141	0	No	#######	No	10	3	6	9	sales	sales exe	2022	2	12	
22	D7EE-56F	Zsa zsa	Evered	female	39	Frequent	17	CA	White	2	married	107863	1	Yes	#######	Yes	8	8	8	5	sales	sales exe	2020	2	10	
23	C395-8C3	Curcio	Franek	male	33	Some Tra	3	CA	Mixed or	1	divorced	53616	1	Yes	#######	No	10	2	3	7	human r	recruiter	2022	2	12	
24	E348-E12	Burnaby	Guillet	male	36	No Trave	36	IL	White	2	divorced	61298	1	Yes	#######	No	10	3	9	3	technolo	software	2022	2	12	
25	B3AF-7E5	Elvira	Ianelli	female	45	Some Tra	34	NY	Black or	2	divorced	54132	1	Yes	#######	No	10	10	10	10	human r	recruiter	2022	2	12	
26	469A-812	Baxie	Rising	male	30	Some Tra	36	CA	Americar	4	married	328415	0	No	#######	No	10	1	10	1	technolo	engineer	2022	2	12	
27	9E22-628	Gifford	Poynser	non-bina	48	Some Tra	37	CA	White	3	single	145337	0	No	#######	No	10	10	10	0	technolo	machine	2022	2	12	
28	D5DA-36	Rickey	Shere	male	33	Some Tra	41	CA	Asian or	2	married	71201	2	No	#######	No	10	8	10	4	sales	sales exe	2022	2	12	
29	BFE3-4AC	Collen	Sedman	female	31	No Trave	25	NY	Americar	2	divorced	55682	1	No	#######	No	10	0	2	0	human r	recruiter	2022	2	12	

employee_dim

Performance Table



	A	B	C	D	E	F	G	H	I	J	K	L
1	Performa	Employee	TimeID	Environm	JobSatisfa	Relationsl	WorkLifeE	SelfRating	ManagerR	TrainingO	TrainingOpportun	
2	PR01	79F7-78EC	1	5	4	5	4	4	4	1	0	
3	PR02	B61E-0F26	2	5	4	4	4	4	3	1	3	
4	PR03	F5E3-48BE	2	3	4	5	3	5	4	3	2	
5	PR04	0678-748A	3	5	3	2	2	3	2	2	0	
6	PR05	541F-3E19	3	5	2	3	4	4	3	1	0	
7	PR06	F93E-BDEF	3	3	3	2	4	4	4	2	0	
8	PR07	9E7A-1F7C	4	3	4	5	5	4	3	2	1	
9	PR08	05ED-92F1	5	4	5	4	3	3	2	1	1	
10	PR09	F72D-261D	5	4	5	2	4	5	4	1	1	
11	PR10	774E-685D	6	5	4	3	4	5	4	2	3	
12	PR100	B013-7D0C	60	4	3	3	4	3	3	2	0	
13	PR1000	528C-3E0D	580	4	4	2	4	5	5	2	2	
14	PR1001	D077-169C	581	3	5	3	3	5	5	2	2	
15	PR1002	9727-BC84	582	4	3	3	2	4	3	2	2	
16	PR1003	DA8E-949(	582	3	5	4	5	5	5	1	0	
17	PR1004	DEC5-931S	582	3	4	3	2	4	4	2	3	
18	PR1005	88B8-EB84	583	3	4	2	4	5	5	3	1	
19	PR1006	9C57-828C	583	5	4	2	2	3	3	1	1	
20	PR1007	E1B4-9AA1	584	5	4	3	3	4	3	3	2	
21	PR1008	3CD6-5587	585	5	4	2	4	4	3	2	0	
22	PR1009	BAFA-86D	585	3	3	4	2	3	3	2	1	
23	PR101	152F-8DB1	61	5	2	5	5	5	4	1	0	

performance_rating_fact

Ready

Week 2: Analysis Questions Phase

Objectives:

- Identify key analysis questions relevant to HR decision-makers.

Tasks:

1. **Determine Data Analysis Questions:**
    a. Explore possible relationships in employee data (e.g., age vs. satisfaction).
    b. Formulate data-driven insights for HR policies.
2. **Design Exploratory Analysis Techniques:**
    a. Conduct descriptive statistics and correlations.
    b. Generate preliminary visualizations.

Tools Used:

1. SQL
2. Python (pandas, Matplotlib)

Step 1: Defining Key Analysis Questions

In this stage, the goal was to frame relevant questions that HR could use to improve employee satisfaction and reduce attrition. Some of the fundamental inquiries included:

- Employee Satisfaction: What factors contribute most to employee happiness?
    - Correlations between work balance, self-rating, team relationships, and overall satisfaction.
- Attrition Analysis: Why do employees leave, and what patterns emerge?
    - Examining relationships between department, salary range, and age group to understand attrition causes.
- Performance & Retention: Do higher-rated employees tend to stay longer?

- Exploring how performance rating impacts employee tenure.

Step 2: Exploratory Data Analysis (EDA)

Once we identified questions, the next step was examining relationships between variables using statistical methods. we employed:

- Descriptive Statistics:
    - Mean, median, standard deviation for satisfaction scores.
    - Frequency distribution for attrition rates across age groups & departments.
- Correlations:
    - Identifying links between work-life balance and job satisfaction.
    - Checking how salary influences retention levels.
- Visualizations Using Python (Matplotlib, Pandas):

- Scatter plots to depict trends between age and attrition.
- Histograms for distribution of employee satisfaction scores.
- Heat maps showing correlations between multiple satisfaction metrics.

Step 3: Deliverables

By the end of this step, we had:

1. A refined set of analysis questions guiding HR strategies.
2. Charts and graphs visually representing emerging trends.

```sql
-------------------------------------------------- Which Department Has Highest Rating
SELECT e.Department, AVG(r.RatingID) AS AverageRating
FROM EmployeeDim e
JOIN PerformanceRatingFact p
ON e.EmployeeID = p.EmployeeID
JOIN RatingDim r
ON p.SelfRatingID = r.RatingID
GROUP BY e.Department
ORDER BY AverageRating DESC;
```

```sql
------------------------------------ Which Department Has The Highest Attrition Rate
SELECT Department, SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) as AttritionCount
FROM EmployeeDim
GROUP BY Department
ORDER BY AttritionCount DESC;
```

```sql
-------------------------------------------------- Relationship Between Years At Company and Job Satisfaction Per Department
SELECT
    e.Department,
    e.YearsAtCompany,
    AVG(CASE
        WHEN s.SatisfactionLevel = 'Very Dissatisfied' THEN 1
        WHEN s.SatisfactionLevel = 'Dissatisfied' THEN 2
        WHEN s.SatisfactionLevel = 'Neutral' THEN 3
        WHEN s.SatisfactionLevel = 'Satisfied' THEN 4
        WHEN s.SatisfactionLevel = 'Very Satisfied' THEN 5
        ELSE 0
    END) AS AverageJobSatisfaction
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
JOIN
    SatisfactionDim s ON p.JobSatisfactionID = s.SatisfactionID
GROUP BY
    e.Department,
    e.YearsAtCompany
ORDER BY
    e.Department,
    e.YearsAtCompany;
```

```sql
---------------------------------- Relationship Between JobRole and Avg Job Satisfaction Per Department
SELECT JobRole, Department,
       AVG(AvgJobSatisfaction) AS AvgJobSatisfaction
FROM (
    SELECT e.EmployeeID, Department, Jobrole, AVG(p.JobSatisfactionID) AS AvgJobSatisfaction
    FROM EmployeeDim e
    JOIN PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
    GROUP BY e.EmployeeID, Department, JobRole
) as tbl
GROUP BY JobRole, Department
ORDER BY Department, JobRole;

------------------------------------------------------------ Relationship Between Avgsalary and AvgSelfRating Per Department
SELECT
    e.Department,
    AVG(e.Salary) AS AvgSalary,
    AVG(p.SelfRatingID) AS AvgSelfRating
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
GROUP BY
    e.Department
ORDER BY
    AvgSalary DESC;
```

```sql
------------------------------------------------------------ How Many Training Opportonities Taken By Employees Per EducationLevel?
SELECT
    e.Education,
    AVG(p.TrainingOpportunitiesTaken) AS AvgTrainingTaken,
    COUNT(DISTINCT e.EmployeeID) AS EmployeeCount
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
GROUP BY
    e.Education
ORDER BY
    e.Education;

------------------------------------------------------------------ What Is The Average Salary Per Education Level and Number Of Employee
SELECT
    e.Education,
    AVG(e.Salary) AS AvgSalary,
    COUNT(*) AS EmployeeCount
FROM
    EmployeeDim e
GROUP BY
    e.Education
ORDER BY
    e.Education;
```

```sql
-------------------------------------------------------------------- What Is The Average Self rating Per Age Group?
SELECT
    CASE
        WHEN e.Age < 25 THEN 'Under 25'
        WHEN e.Age BETWEEN 25 AND 34 THEN '25-34'
        WHEN e.Age BETWEEN 35 AND 44 THEN '35-44'
        ELSE '45 and above'
    END AS AgeGroup,
    AVG(p.SelfRatingID) AS AvgSelfRating
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
GROUP BY
    CASE
        WHEN e.Age < 25 THEN 'Under 25'
        WHEN e.Age BETWEEN 25 AND 34 THEN '25-34'
        WHEN e.Age BETWEEN 35 AND 44 THEN '35-44'
        ELSE '45 and above'
    END
ORDER BY
    AgeGroup;
```

```sql
-------------------------------------------------------------------------- Does Satisfaction Level Affect Attrition?
SELECT s.SatisfactionLevel, COUNT(*) AS EmployeeCount
FROM  (
    SELECT e.EmployeeID, AVG(p.JobSatisfactionID) AS AvgJobSatisfaction
    FROM EmployeeDim e
    JOIN PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
    WHERE Attrition ='Yes'
    GROUP BY e.EmployeeID
) as tbl
LEFT JOIN SatisfactionDim s ON tbl.AvgJobSatisfaction = s.SatisfactionID
GROUP BY s.SatisfactionLevel
ORDER BY s.SatisfactionLevel;

-------------------------------------------------------------------------- Does Age or Martial Status Affect Attrition?
SELECT Age, SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) as AttritionCount
FROM EmployeeDim
GROUP BY Age
ORDER BY AttritionCount DESC;

SELECT MaritalStatus, SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) as AttritionCount
FROM EmployeeDim
GROUP BY MaritalStatus
ORDER BY AttritionCount DESC;
```

```sql
-------------------------------------------------------------------------- Are Employees With High Environment Satisfication Are More li
SELECT AVG(AvgEnvironmentSatisfaction) AS AvgEnvironmentSatisfaction,
       SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount
FROM (
    SELECT e.EmployeeID, AVG(p.EnvironmentSatisfactionID) AS AvgEnvironmentSatisfaction, e.Attrition
    FROM EmployeeDim e
    JOIN PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
    GROUP BY e.EmployeeID, e.Attrition
) as tbl
GROUP BY AvgEnvironmentSatisfaction
ORDER BY AvgEnvironmentSatisfaction;
```

```sql
----------------------------------- How Does BusinessTravel Affect Job Satisfaction and Attrition?
SELECT TravelGroup,
       AVG(AvgJobSatisfaction) AS AvgJobSatisfaction,
       SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount
FROM (
    SELECT e.EmployeeID, e.BusinessTravel as TravelGroup, AVG(p.JobSatisfactionID) AS AvgJobSatisfaction,
           e.Attrition
    FROM EmployeeDim e
    JOIN PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
    GROUP BY e.EmployeeID, e.BusinessTravel, e.Attrition
) as tbl
GROUP BY TravelGroup
ORDER BY TravelGroup;

------------------------------------------------------------- How Does State Affect Attrition? {Illinois - New York - California}
SELECT
    e.State,
    COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) AS AttritionCount,
    COUNT(*) AS TotalEmployees,
    (COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) * 100.0 / COUNT(*)) AS AttritionRate
FROM
    EmployeeDim e
GROUP BY
    e.State
ORDER BY
    AttritionRate DESC;
```

```sql
------------------------------------------------------------ How Does Promotion Affect Attrition?
SELECT
    CASE
        WHEN e.YearsSinceLastPromotion = 0 THEN 'promoted this year'
        WHEN e.YearsSinceLastPromotion <= 2 THEN '1-2 years ago'
        ELSE 'more than 2 years'
    END AS PromotionGroup,
    COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) AS AttritionCount,
    COUNT(*) AS TotalEmployees,
    (COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) * 100.0 / COUNT(*)) AS AttritionRate
FROM
    EmployeeDim e
GROUP BY
    CASE
        WHEN e.YearsSinceLastPromotion = 0 THEN 'promoted this year'
        WHEN e.YearsSinceLastPromotion <= 2 THEN '1-2 years ago'
        ELSE 'more than 2 years'
    END
ORDER BY
    PromotionGroup;
```

```sql
------------------------------------------------------------ How Does Satisfaction Affect Attrition?
WITH LatestRating AS (
    SELECT
        p.EmployeeID,
        p.JobSatisfactionID,
        ROW_NUMBER() OVER (PARTITION BY p.EmployeeID ORDER BY t.Year DESC) AS rn
    FROM
        PerformanceRatingFact p
    JOIN
        TimeDim t ON p.TimeID = t.TimeID
)
SELECT
    CASE
        WHEN lr.JobSatisfactionID = 1 THEN 'Very Dissatisfied'
        WHEN lr.JobSatisfactionID = 2 THEN 'Dissatisfied'
        WHEN lr.JobSatisfactionID = 3 THEN 'Neutral'
        WHEN lr.JobSatisfactionID = 4 THEN 'Satisfied'
        WHEN lr.JobSatisfactionID = 5 THEN 'Very Satisfied'
    END AS SatisfactionLevel,
    COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) AS AttritionCount,
    COUNT(DISTINCT e.EmployeeID) AS TotalEmployees,
    (COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) * 100.0 / COUNT(DISTINCT e.EmployeeID)) AS AttritionRate
FROM
    EmployeeDim e
JOIN
    LatestRating lr ON e.EmployeeID = lr.EmployeeID
WHERE
    lr.rn = 1
GROUP BY lr.JobSatisfactionID
ORDER BY lr.JobSatisfactionID;
```

```sql
------------------------------------------------------------------------ How Does OverTime Affect Attrition?
SELECT
    e.OverTime,
    COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) AS AttritionCount,
    COUNT(DISTINCT e.EmployeeID) AS TotalEmployees,
    ROUND((COUNT(CASE WHEN e.Attrition = 'Yes' THEN 1 END) * 100.0 / COUNT(DISTINCT e.EmployeeID)), 2) AS AttritionRate
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
JOIN
    TimeDim t ON p.TimeID = t.TimeID
WHERE EXISTS (
    SELECT 1
    FROM PerformanceRatingFact p2
    JOIN TimeDim t2 ON p2.TimeID = t2.TimeID
    WHERE p2.EmployeeID = p.EmployeeID
    AND t2.Year = (
        SELECT MAX(t3.Year)
        FROM TimeDim t3
        JOIN PerformanceRatingFact p3 ON t3.TimeID = p3.TimeID
        WHERE p3.EmployeeID = p.EmployeeID
    )
)
GROUP BY
    e.OverTime
ORDER BY
    e.OverTime;
```

```sql
---------------------------------------------------------------------- Does Training Opportunities Taken Affect Self Rating?
SELECT
    p.TrainingOpportunitiesTaken,
    AVG(r.RatingID) AS AverageRating
FROM
    PerformanceRatingFact p
JOIN
    RatingDim r ON p.SelfRatingID = r.RatingID
GROUP BY
    p.TrainingOpportunitiesTaken
ORDER BY
    p.TrainingOpportunitiesTaken;

---------------------------------------------------------------------- How Does JobRole Affect Work-Life Balance?
SELECT
    e.JobRole,
    AVG(s.SatisfactionID) AS AvgWorkLifeBalance
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
JOIN
    SatisfactionDim s ON p.WorkLifeBalanceID = s.SatisfactionID
GROUP BY
    e.JobRole
ORDER BY
    AvgWorkLifeBalance DESC;
```

```sql
---------------------------------------------------------------------------- Deos Employee Salary Affect Self Rating?
SELECT
    CASE
        WHEN e.Salary < 50000 THEN 'Under 50K'
        WHEN e.Salary BETWEEN 50000 AND 100000 THEN '50K-100K'
        WHEN e.Salary BETWEEN 100001 AND 150000 THEN '100K-150K'
        ELSE 'Over 150K'
    END AS SalaryRange,
    AVG(p.SelfRatingID) AS AvgSelfRating
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
GROUP BY
    CASE
        WHEN e.Salary < 50000 THEN 'Under 50K'
        WHEN e.Salary BETWEEN 50000 AND 100000 THEN '50K-100K'
        WHEN e.Salary BETWEEN 100001 AND 150000 THEN '100K-150K'
        ELSE 'Over 150K'
    END
ORDER BY
    SalaryRange;
```

```sql
---------------------------------------------------------------------- Does Married Employees Tend To Take More Training Opportunities Than Singles?
SELECT
    e.MaritalStatus,
    AVG(p.TrainingOpportunitiesTaken) AS AvgTrainingTaken,
    COUNT(DISTINCT e.EmployeeID) AS EmployeeCount
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
GROUP BY
    e.MaritalStatus
ORDER BY
    AvgTrainingTaken DESC;

---------------------------------------------------------------------- Does Ethnicity Affect Employee's Self Rating?
SELECT
    e.Ethnicity,
    AVG(p.SelfRatingID) AS AvgSelfRating,
    COUNT(DISTINCT e.EmployeeID) AS EmployeeCount
FROM
    EmployeeDim e
JOIN
    PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
GROUP BY
    e.Ethnicity
ORDER BY
    AvgSelfRating DESC;
```

```sql
------------------------------------ Does Long Years With the Same Manager Affects The Manager's Rating For The Employee?
SELECT YearsWithCurrentManager,
       AVG(AvgManagerRating) AS AvgManagerRating
FROM (
    SELECT e.EmployeeID, e.YearsWithCurrManager as YearsWithCurrentManager, AVG(p.ManagerRatingID) AS AvgManagerrating
    FROM EmployeeDim e
    JOIN PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
    GROUP BY e.EmployeeID, e.YearsWithCurrManager
) as tbl
GROUP BY YearsWithCurrentManager
ORDER BY YearsWithCurrentManager;

------------------------------------ How Many Training Opportunities Taken From Employees With High Self Rating?
SELECT SelfRatingID as SelfRating, SUM(TrainingOpportunitiesTaken)
FROM PerformanceRatingFact
GROUP BY SelfRatingID
ORDER BY SelfRatingID;

------------------------------------ Does Job Satisfaction Relates To Opportunities Provided?
SELECT JobSatisfactionID as JobSatisfaction, SUM(TrainingOpportunitiesWithinYear)
FROM PerformanceRatingFact
GROUP BY JobSatisfactionID
ORDER BY JobSatisfactionID;
```

```sql
------------------------------------ Relation Between Self Rating and Manager Rating
SELECT SelfRatingID as SelfRating, ManagerRatingID as ManagerRating, Count(*) as Occurances
FROM PerformanceRatingFact
GROUP BY SelfRatingID, ManagerRatingID
ORDER BY SelfRatingID, ManagerRatingID;

SELECT Count(*) FROM PerformanceRatingFact;

------------------------------------ Are Employees With High Stock Option Level Tend To Be More Satisfied In The Job?
SELECT StockOptionLevel, AVG(AvgJobSatisfaction) AS AvgJobSatisfaction
FROM (
    SELECT e.EmployeeID, StockOptionLevel, AVG(p.JobSatisfactionID) AS AvgJobSatisfaction
    FROM EmployeeDim e
    JOIN PerformanceRatingFact p ON e.EmployeeID = p.EmployeeID
    GROUP BY e.EmployeeID, StockOptionLevel
) as tbl
GROUP BY StockOptionLevel
ORDER BY StockOptionLevel;
```

Week 3: Forecasting Questions Phase

Objectives:

- Predict trends in HR metrics using forecasting techniques.

Tasks:

1. **Trend Analysis & Forecasting:**
    a. Identify patterns in employee behaviour.
    b. Apply machine learning models for predictions.
2. **Develop Visualization for Forecasted Trends:**
    a. Use plots to illustrate projections and insights.

Tools Used:

1. Python (scikit-learn, pandas, Matplotlib)

Deliverables:

2. Visualization plots addressing forecasting questions.
3. Model documentation explaining predictions.

```python
import pymysql

conn = pymysql.connect(
    host="localhost",
    user="root",
    password="OKASHA3210",
    database="DEPI"
)

cursor = conn.cursor()

cursor.execute("SHOW TABLES;")
tables = cursor.fetchall()

for table in tables:
    table_name = table[0]
    print(f"\n===== أول 5 صفوف من الجدول: {table_name} =====")

    try:
        cursor.execute(f"SELECT * FROM {table_name} LIMIT 5;")
        rows = cursor.fetchall()

        column_names = [desc[0] for desc in cursor.description]
        print(" | ".join(column_names))
        print("-" * 40)

        for row in rows:
            print(" | ".join(str(value) for value in row))

    except Exception as e:
        print(f"خطأ أثناء قراءة الجدول {table_name}: {e}")

cursor.close()
conn.close()
```

Python

```python
import pymysql
import pandas as pd

conn = pymysql.connect(
    host="localhost",
    user="root",
    password="OKASHA3210",
    database="DEPI"
)

employee_query = """
SELECT EmployeeID, Attrition,Salary, Department
FROM EmployeeDim;
"""
employee_df = pd.read_sql(employee_query, conn)

performance_query = """
SELECT EmployeeID, ReviewDate
FROM PerformanceRatingFact;
"""
performance_df = pd.read_sql(performance_query, conn)

df = pd.merge(employee_df, performance_df, on='EmployeeID', how='left')

conn.close()

print(df.head())
```

Python

```python
print(df.isnull().sum())
```

Python

```
EmployeeID      0
Attrition       0
Salary          0
Department      0
ReviewDate    190
dtype: int64
```

```python
df = df.dropna(subset=['ReviewDate'])

print(df.isnull().sum())
```

Python

```
EmployeeID    0
Attrition     0
Salary        0
Department    0
ReviewDate    0
dtype: int64
```

```python
df['ReviewDate'] = pd.to_datetime(df['ReviewDate'])
df = df.sort_values('ReviewDate').groupby('EmployeeID').last().reset_index()
print(f"عدد الموظفين بعد اختيار أحدث تقييم: {len(df)}")
```

Python

```
عدد الموظفين بعد اختيار أحدث تقييم: 1280
```

```python
df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})
print(df[['EmployeeID', 'Attrition']].head())
```

Python

```
  EmployeeID  Attrition
0  005C-E0FB          0
1  00A3-2445          0
2  00B0-F199          1
3  00D4-DD53          1
4  00E4-3D60          1
```

```python
df = df[df['Year'] < 2023]
print(df['Year'].unique())
```

Python

```python
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Department_Encoded'] = le.fit_transform(df['Department'])

print(df[['Department', 'Department_Encoded']].drop_duplicates())
```

Python

```
        Department  Department_Encoded
0            sales                   1
1       technology                   2
65  human resources                   0
```

```python
from sklearn.model_selection import train_test_split

X = df[['Salary', 'Department_Encoded']]
y = df['Attrition']

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

# التأكد من عدد الصفوف في التدريب والاختبار
print(f"عدد الصفوف في بيانات التدريب: {len(X_train)}")
print(f"عدد الصفوف في بيانات الاختبار: {len(X_test)}")
```

Python

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

model = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"دقة النموذج: {accuracy:.2f}")
```

Python

```
دقة النموذج: 0.70
```

```python
#suppose
df_2023 = df[['Salary', 'Department_Encoded', 'Department']].copy()
df_2023['Salary'] = df_2023['Salary'] * 1.05
df_2023['Predicted_Attrition'] = model.predict(df_2023[['Salary', 'Department_Encoded']])
```

Python

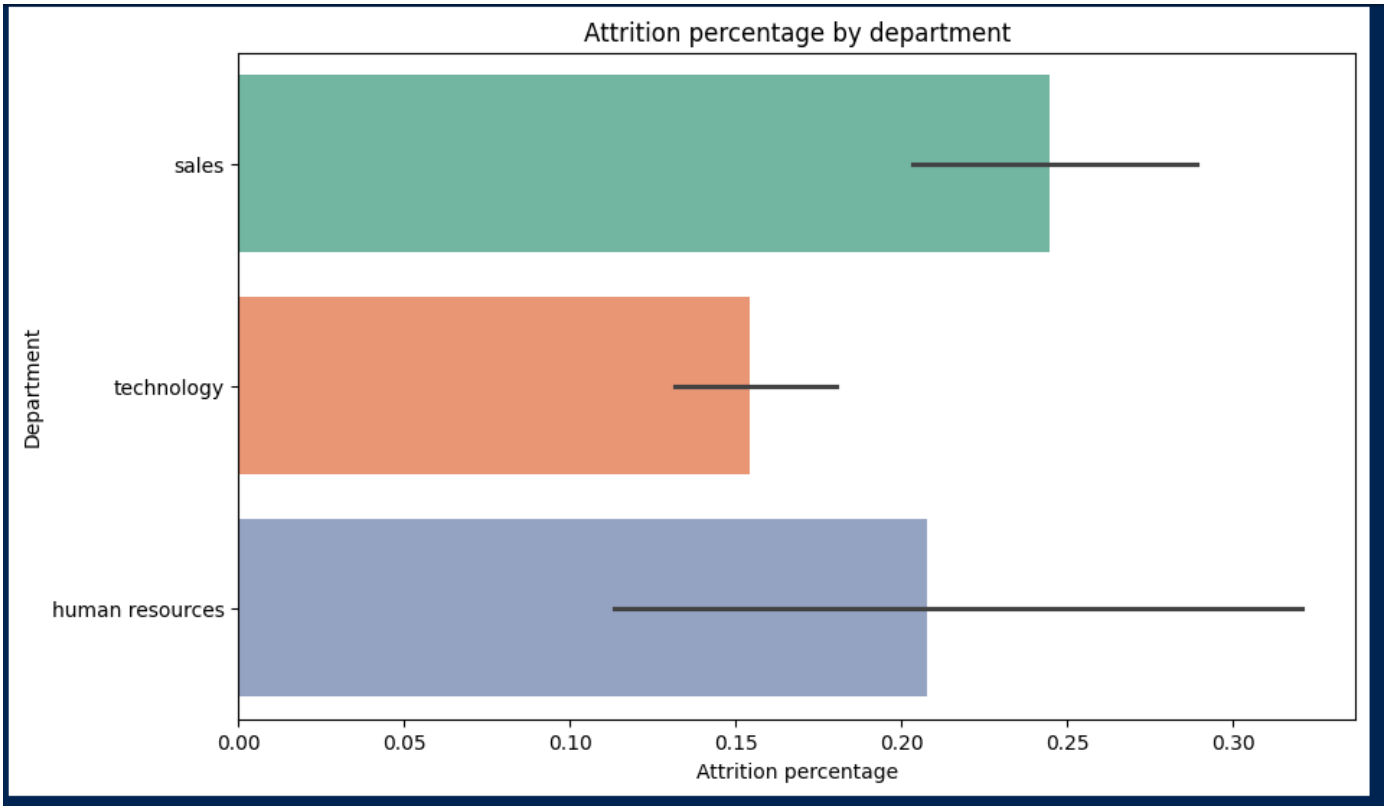
```python
attrition_by_dept = df_2023.groupby('Department')['Predicted_Attrition'].mean() * 100
print("المتوقعة لسنة 2023 لكل قسم Attrition نسبة الـ:")
print(attrition_by_dept)
```

Python

```
نسبة الـ Attrition لكل قسم 2023 المتوقعة لسنة:
Department
human resources    20.754717
sales              24.479167
technology         15.421115
Name: Predicted_Attrition, dtype: float64
```

```python
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
sns.barplot(x='Predicted_Attrition', y='Department', hue='Department', data=df_2023, palette='Set2')
plt.title('Attrition percentage by department')
plt.xlabel('Attrition percentage')
plt.ylabel('Department')
plt.show()
```

Python




```python
feature_importance = pd.DataFrame({
    'Feature': ['MonthlyIncome', 'Department_Encoded'],
    'Importance': model.feature_importances_
})
print("أهمية العوامل في النموذج:")
print(feature_importance.sort_values(by='Importance', ascending=False))
```

Python

```
أهمية العوامل في النموذج:
            Feature  Importance
0      MonthlyIncome    0.974575
1  Department_Encoded    0.025425
```

```python
from sqlalchemy import create_engine
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
engine = create_engine("mysql+mysqlconnector://root:OKASHA3210@localhost/DEPI")
employee_query = """
SELECT EmployeeID, Attrition, YearsAtCompany
FROM EmployeeDim;
"""
employee_df = pd.read_sql(employee_query, con=engine)
performance_query = """
SELECT EmployeeID, ReviewDate
FROM PerformanceRatingFact;
"""
performance_df = pd.read_sql(performance_query, con=engine)
df = pd.merge(employee_df, performance_df, on='EmployeeID', how='left')
df = df.dropna(subset=['ReviewDate'])
df['ReviewDate'] = pd.to_datetime(df['ReviewDate'])
df['Year'] = df['ReviewDate'].dt.year
df = df.sort_values('ReviewDate').groupby('EmployeeID').last().reset_index()
df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})
df = df[df['Year'] < 2023]

print(df[['EmployeeID', 'Attrition', 'YearsAtCompany', 'Year']].head())
```

```python
import pandas as pd
from sqlalchemy import create_engine
engine = create_engine("mysql+mysqlconnector://root:OKASHA3210@localhost/DEPI")

employee_query = """
SELECT EmployeeID, Attrition, YearsAtCompany, Salary, Department, Age
FROM EmployeeDim;
"""
employee_df = pd.read_sql(employee_query, con=engine)

performance_query = """
SELECT EmployeeID, ReviewDate
FROM PerformanceRatingFact;
"""
performance_df = pd.read_sql(performance_query, con=engine)
```

Python

```python
#merge
df = pd.merge(employee_df, performance_df, on='EmployeeID', how='left')
df = df.dropna(subset=['ReviewDate'])
df['ReviewDate'] = pd.to_datetime(df['ReviewDate'])
df['Year'] = df['ReviewDate'].dt.year

#last rate
df = df.sort_values('ReviewDate').groupby('EmployeeID').last().reset_index()
df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})
df = df[df['Year'] < 2023]
df['Department_Encoded'] = df['Department'].map({'sales': 0, 'technology': 1})

print("أول 5 صفوف من البيانات:")
print(df.head())
print("\nمعلومات عن البيانات:")
print(df.info())
```

Python

```
أول 5 صفوف من البيانات:
  EmployeeID  Attrition  YearsAtCompany    Salary  Department  Age ReviewDate  \
0  005C-E0FB          0               5   56155.0       sales   24 2022-06-17
1  00A3-2445          0              10  126238.0  technology   30 2022-06-18
2  00B0-F199          1               1   97824.0       sales   23 2022-05-20
3  00D4-DD53          1               5   68508.0  technology   30 2022-02-27
4  00E4-3D60          1               0  109778.0  technology   30 2022-03-28
```

```python
from sklearn.model_selection import train_test_split

X = df[['YearsAtCompany', 'Salary', 'Age', 'Department_Encoded']]
y = df['Attrition']

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

print("\nشكل بيانات التدريب والاختبار:")
print(f"X_train: {X_train.shape}, X_test: {X_test.shape}")
print(f"y_train: {y_train.shape}, y_test: {y_test.shape}")
```

Python

```
شكل بيانات التدريب والاختبار:
X_train: (1024, 4), X_test: (256, 4)
y_train: (1024,), y_test: (256,)
```

```python
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, classification_report

scale_pos_weight = len(y_train[y_train == 0]) / len(y_train[y_train == 1])

model = XGBClassifier(scale_pos_weight=scale_pos_weight, max_depth=3, n_estimators=50, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"\nModel accuracy: {accuracy:.2f}")

print("\nPerformance report:")
print(classification_report(y_test, y_pred))
```

Python

```
Model accuracy: 0.82

Performance report:
              precision    recall  f1-score   support

           0       0.89      0.88      0.89       200
           1       0.59      0.62      0.61        56

    accuracy                           0.82       256
   macro avg       0.74      0.75      0.75       256
weighted avg       0.83      0.82      0.83       256
```

```python
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, classification_report

scale_pos_weight = len(y_train[y_train == 0]) / len(y_train[y_train == 1])

model = XGBClassifier(scale_pos_weight=scale_pos_weight, max_depth=3, n_estim
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

```python
df_2023 = df[['YearsAtCompany', 'Salary', 'Age', 'Department_Encoded']].copy()
df_2023['YearsAtCompany'] = df_2023['YearsAtCompany'] + 1

probabilities = model.predict_proba(df_2023)[:, 1]
df_2023['Predicted_Attrition_Proba'] = probabilities

attrition_by_years = df_2023.groupby('YearsAtCompany')['Predicted_Attrition_Proba'].mean() * 100
print("\nExpected attrition ratio for 2023 based on number of years in the company:")
print(attrition_by_years)
```

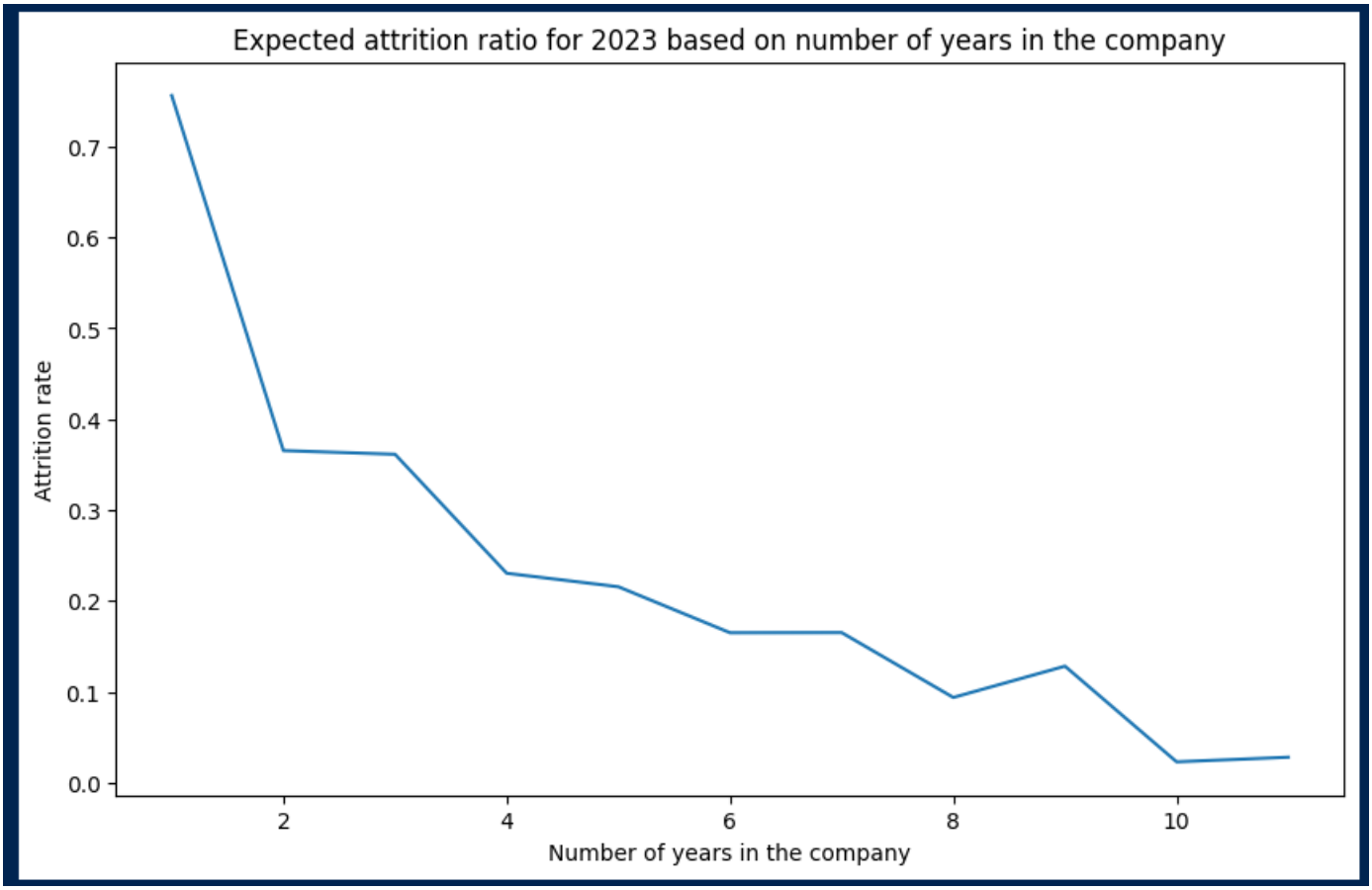
Python

```
Expected attrition ratio for 2023 based on number of years in the company:
YearsAtCompany
1     75.622826
2     36.545086
3     36.139942
4     23.044756
5     21.557045
6     16.497993
7     16.513783
8      9.372179
9     12.817442
10     2.286455
11     2.795860
Name: Predicted_Attrition_Proba, dtype: float32
```

```python
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.lineplot(x='YearsAtCompany', y='Predicted_Attrition_Proba', data=df_2023, estimator='mean', errorbar=None)
plt.title('Expected attrition ratio for 2023 based on number of years in the company')
plt.xlabel('Number of years in the company')
plt.ylabel('Attrition rate')
plt.show()
```

Python




```python
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine("mysql+mysqlconnector://root:OKASHA3210@localhost/DEPI")

employee_query = """
SELECT EmployeeID, Attrition, DistanceFromHome, Age
FROM EmployeeDim;
"""
employee_df = pd.read_sql(employee_query, con=engine)


performance_query = """
SELECT EmployeeID, ReviewDate, WorkLifeBalanceID, SelfRatingID, JobSatisfactionID, EnvironmentSatisfactionID
FROM PerformanceRatingFact;
"""
performance_df = pd.read_sql(performance_query, con=engine)
```

Python

```python
df = pd.merge(employee_df, performance_df, on='EmployeeID', how='left')

df = df.dropna(subset=['ReviewDate'])
df['ReviewDate'] = pd.to_datetime(df['ReviewDate'])
df['Year'] = df['ReviewDate'].dt.year

df = df.sort_values('ReviewDate').groupby('EmployeeID').last().reset_index()

df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})

df = df[df['Year'] < 2023]

cols_to_convert = ['WorkLifeBalanceID', 'SelfRatingID', 'JobSatisfactionID', 'EnvironmentSatisfactionID']
df[cols_to_convert] = df[cols_to_convert].apply(pd.to_numeric, errors='coerce')

df['OverallRating'] = df[cols_to_convert].mean(axis=1)


print("أول 5 صفوف من البيانات:")
print(df.head())
print("\nمعلومات عن البيانات:")
print(df.info())
```

Python

```
أول 5 صفوف من البيانات:
  EmployeeID  Attrition  DistanceFromHome  Age ReviewDate  WorkLifeBalanceID  \
0  005C-E0FB          0                17   24 2022-06-17                  4
1  00A3-2445          0                 6   30 2022-06-18                  4
2  00B0-F199          1                35   23 2022-05-20                  2
3  00D4-DD53          1                44   30 2022-02-27                  3
4  00E4-3D60          1                37   30 2022-03-28                  2

   SelfRatingID  JobSatisfactionID  EnvironmentSatisfactionID  Year  \
0             4                  4                          3  2022
1             4                  5                          4  2022
2             4                  3                          1  2022
3             4                  4                          3  2022
4             4                  2                          1  2022

   OverallRating
0           3.75
1           4.25
2           2.50
3           3.50
4           2.25

معلومات عن البيانات:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1280 entries, 0 to 1279
```

```python
from sklearn.model_selection import train_test_split

X = df[['DistanceFromHome', 'OverallRating']]
y = df['Attrition']

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

print("\nشكل بيانات التدريب والاختبار:")
print(f"X_train: {X_train.shape}, X_test: {X_test.shape}")
print(f"y_train: {y_train.shape}, y_test: {y_test.shape}")
```

Python

```
شكل بيانات التدريب والاختبار:
X_train: (1024, 2), X_test: (256, 2)
y_train: (1024,), y_test: (256,)
```

```python
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine("mysql+mysqlconnector://root:OKASHA3210@localhost/DEPI")

employee_query = """
SELECT EmployeeID, Attrition, DistanceFromHome, Age, YearsAtCompany, Salary
FROM EmployeeDim;
"""
employee_df = pd.read_sql(employee_query, con=engine)

performance_query = """
SELECT EmployeeID, ReviewDate, WorkLifeBalanceID, SelfRatingID, JobSatisfactionID, EnvironmentSatisfactionID
FROM PerformanceRatingFact;
"""
performance_df = pd.read_sql(performance_query, con=engine)
```

Python

```python
df = pd.merge(employee_df, performance_df, on='EmployeeID', how='left')

df = df.dropna(subset=['ReviewDate'])
df['ReviewDate'] = pd.to_datetime(df['ReviewDate'])
df['Year'] = df['ReviewDate'].dt.year


df = df.sort_values('ReviewDate').groupby('EmployeeID').last().reset_index()


df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})


df = df[df['Year'] < 2023]


cols_to_convert = ['WorkLifeBalanceID', 'SelfRatingID', 'JobSatisfactionID', 'EnvironmentSatisfactionID']
df[cols_to_convert] = df[cols_to_convert].apply(pd.to_numeric, errors='coerce')


df['OverallRating'] = df[cols_to_convert].mean(axis=1)
```

```python
# التحقق من البيانات
print("أول 5 صفوف من البيانات:")
print(df.head())
print("\nمعلومات عن البيانات:")
print(df.info())
```

```
أول 5 صفوف من البيانات:
   EmployeeID  Attrition  DistanceFromHome  Age  YearsAtCompany    Salary  \
0  005C-E0FB          0                17   24               5   56155.0
1  00A3-2445          0                 6   30              10  126238.0
2  00B0-F199          1                35   23               1   97824.0
3  00D4-DD53          1                44   30               5   68508.0
4  00E4-3D60          1                37   30               0  109778.0

   ReviewDate  WorkLifeBalanceID  SelfRatingID  JobSatisfactionID  \
0 2022-06-17                  4             4                  4
1 2022-06-18                  4             4                  5
2 2022-05-20                  2             4                  3
3 2022-02-27                  3             4                  4
4 2022-03-28                  2             4                  2

    EnvironmentSatisfactionID  Year  OverallRating
0                           3  2022           3.75
1                           4  2022           4.25
2                           1  2022           2.50
3                           3  2022           3.50
4                           1  2022           2.25
```

```python
from sklearn.model_selection import train_test_split

X = df[['DistanceFromHome', 'OverallRating', 'Age', 'YearsAtCompany', 'Salary']]
y = df['Attrition']

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

Python

```python
df['Attrition_Probability'] = model.predict_proba(df[['DistanceFromHome', 'OverallRating', 'Age', 'YearsAtCompany', 'Salary']])[:, 1] * 100

result = df[['EmployeeID', 'DistanceFromHome', 'OverallRating', 'Age', 'YearsAtCompany', 'Salary', 'Attrition_Probability']]
print("\nاحتمالية الـ Attrition لكل موظف:")
print(result)
```

Python

```
احتمالية الـ Attrition لكل موظف:
     EmployeeID  DistanceFromHome  OverallRating  Age  YearsAtCompany  \
0     005C-E0FB                17           3.75   24               5
1     00A3-2445                 6           4.25   30              10
2     00B0-F199                35           2.50   23               1
3     00D4-DD53                44           3.50   30               5
4     00E4-3D60                37           2.25   30               0
...         ...               ...            ...  ...             ...
1275  FE0F-498F                36           4.50   22               4
1276  FE2B-3DC7                23           3.50   28               5
1277  FEEC-A663                 1           3.75   27               7
1278  FF14-A43E                41           3.75   47              10
1279  FFCF-0BD5                42           3.75   23               5

        Salary  Attrition_Probability
0      56155.0               4.253017
1     126238.0               0.223444
2      97824.0              86.926544
3      68508.0              87.281456
4     109778.0              99.297066
...        ...                    ...
1275   57686.0               6.386410
1276   35606.0              84.525055
1277   48442.0              12.529579
1278   40786.0               8.029566
1279   30683.0               5.912015
```

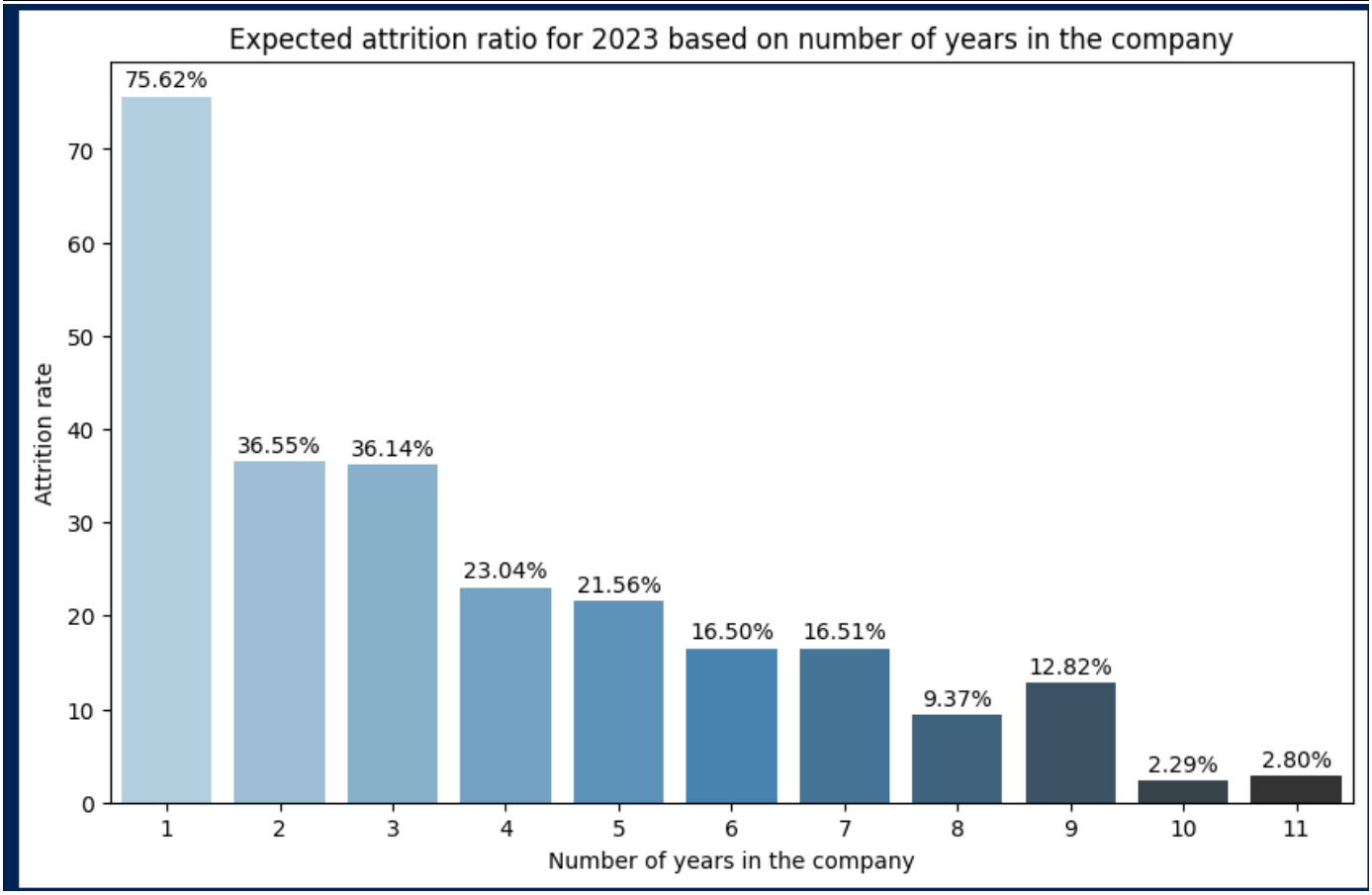
```python
import seaborn as sns
import matplotlib.pyplot as plt

bar_data = attrition_by_years.reset_index()
bar_data.columns = ['YearsAtCompany', 'Attrition_Probability']

plt.figure(figsize=(10, 6))
ax = sns.barplot(x='YearsAtCompany', y='Attrition_Probability', hue='YearsAtCompany', data=bar_data, palette='Blues_d', legend=False)

for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width() / 2., height + 0.5, f'{height:.2f}%', ha='center', va='bottom')

plt.title('Expected attrition ratio for 2023 based on number of years in the company')
plt.xlabel('Number of years in the company')
plt.ylabel('Attrition rate')
plt.show()
```



Week 4: Visualization Dashboard & Final Presentation

Objectives:

- Summarize findings with an interactive visualization dashboard.
- Present results in a structured report.

Tasks:

1. **Build a Visualization Dashboard:**
    a. Create Tableau dashboards to display HR insights.
2. **Prepare a Final Report & Presentation:**
    a. Compile findings, models, and methodology.
    b. Deliver an impactful presentation for stakeholders.

Tools Used:

1. SQL
2. Python (pandas, Matplotlib)
3. Tableau

Step 1: Building the Visualization Dashboard

The goal was to present insights dynamically, allowing HR stakeholders to explore patterns easily. We used Power BI and Tableau—both powerful tools for interactive dashboards. Here's how we structured it:

Data Integration

- Pulled processed data from SQL databases and cleaned Excel files.
- Merged employee satisfaction, performance metrics, and attrition trends into structured datasets.

Dashboard Components

- Attrition Analysis:
    - Created a heat map showing attrition rates across departments and salary ranges.
    - Used line graphs to display attrition trends over time.
- Employee Satisfaction Insights:
    - Designed bar charts comparing satisfaction levels across different departments.
    - Created drill-through reports allowing users to explore individual performance ratings.
- Work-Life Balance & Performance:

- Used scatter plots to analyse relationships between working hours and self-ratings.
- Built interactive slicers to filter data by age, department, and satisfaction levels.

Step 2: Preparing the Final Report & Presentation

Once the dashboard was ready, the next step was compiling the findings into a structured Word report and preparing a stakeholder presentation. The key components included:

Report Structure

1. Introduction:
    - Overview of objectives and methodology.
    - Summary of data modelling and cleaning phases.
2. Key Insights:
    - Trends discovered in employee satisfaction and attrition.
    - How various factors (age, department, salary) impact retention.
3. Dashboard Walkthrough:
    - Explanation of each visualization and its relevance.
    - How stakeholders can interact with the dashboard.
4. Recommendations:

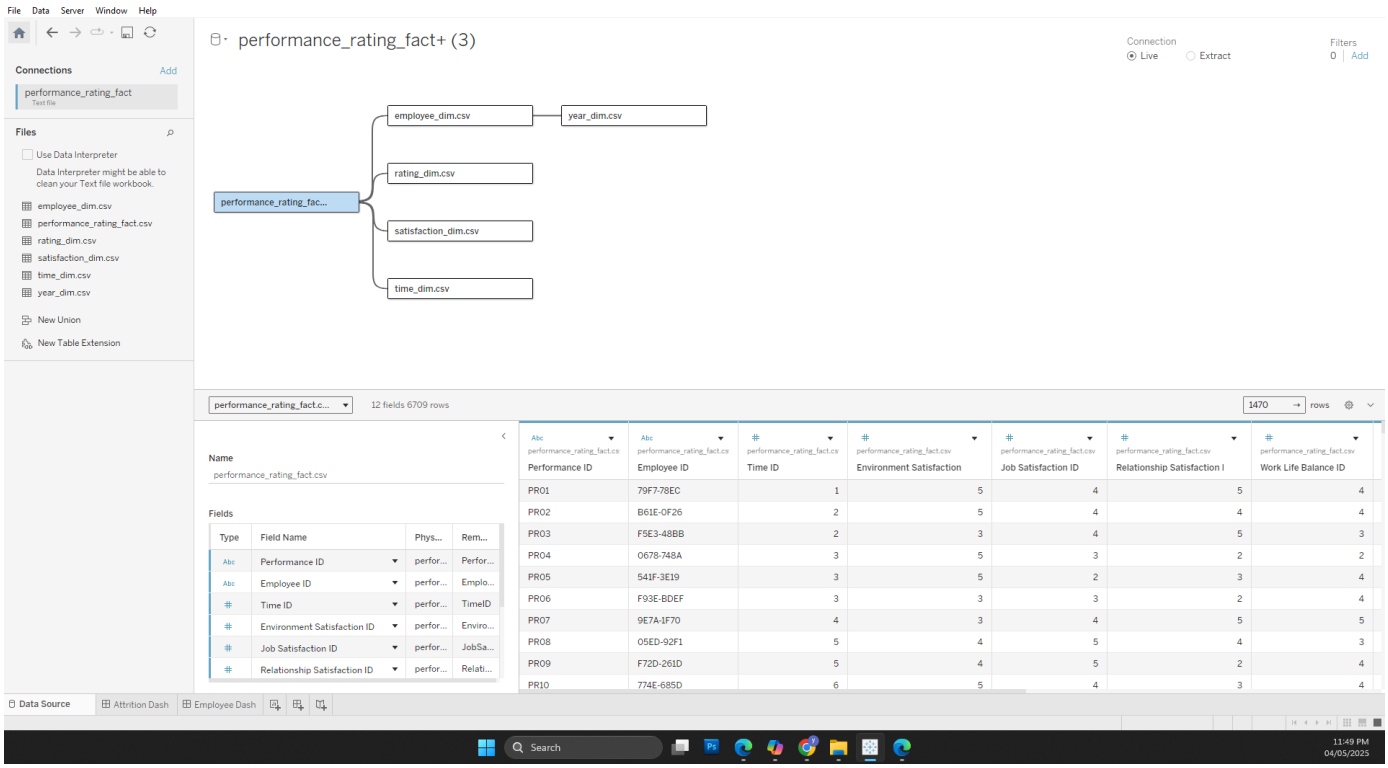
- Actionable strategies HR can implement based on findings.
- Policy suggestions for improving employee retention and satisfaction.

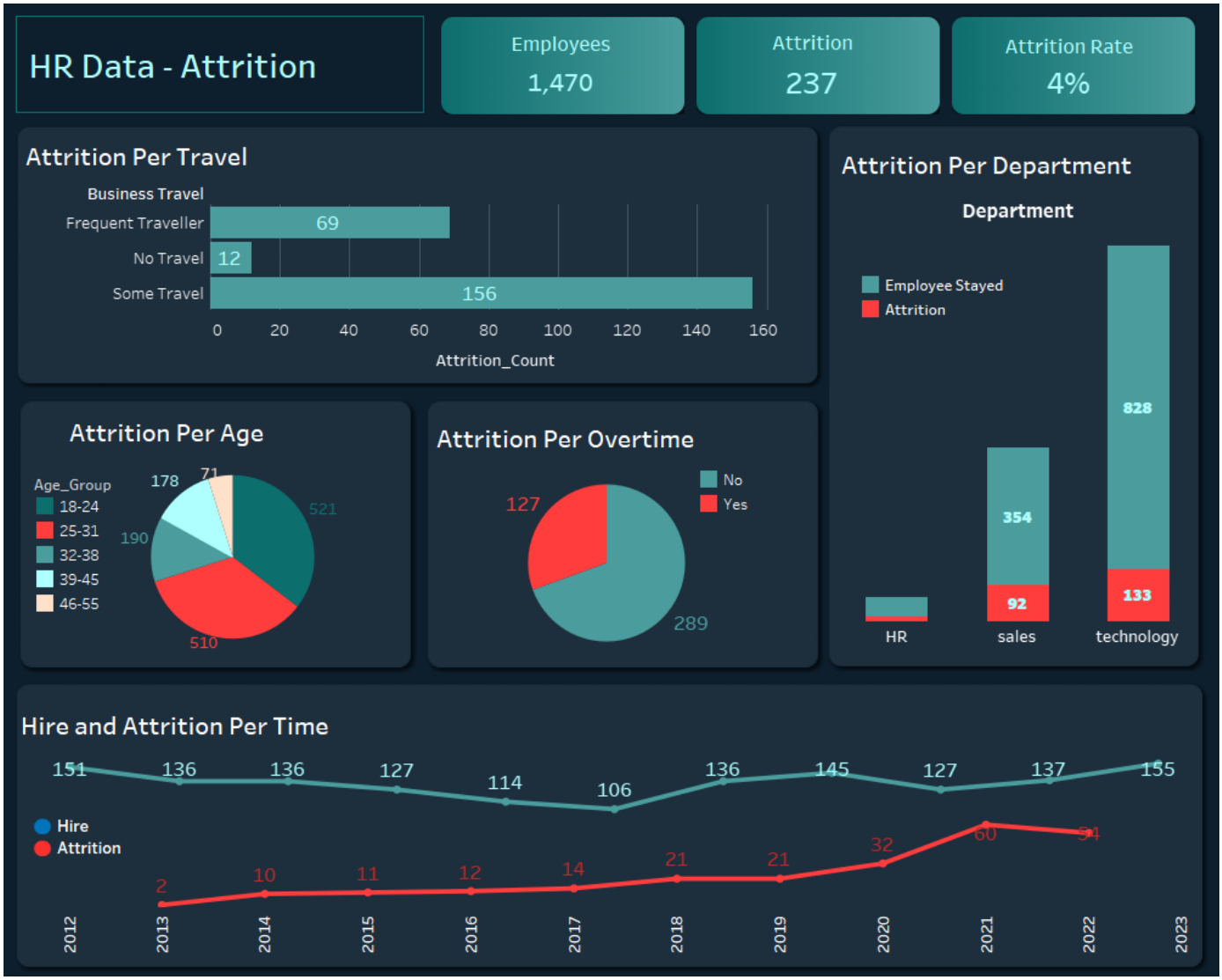
Presentation Design

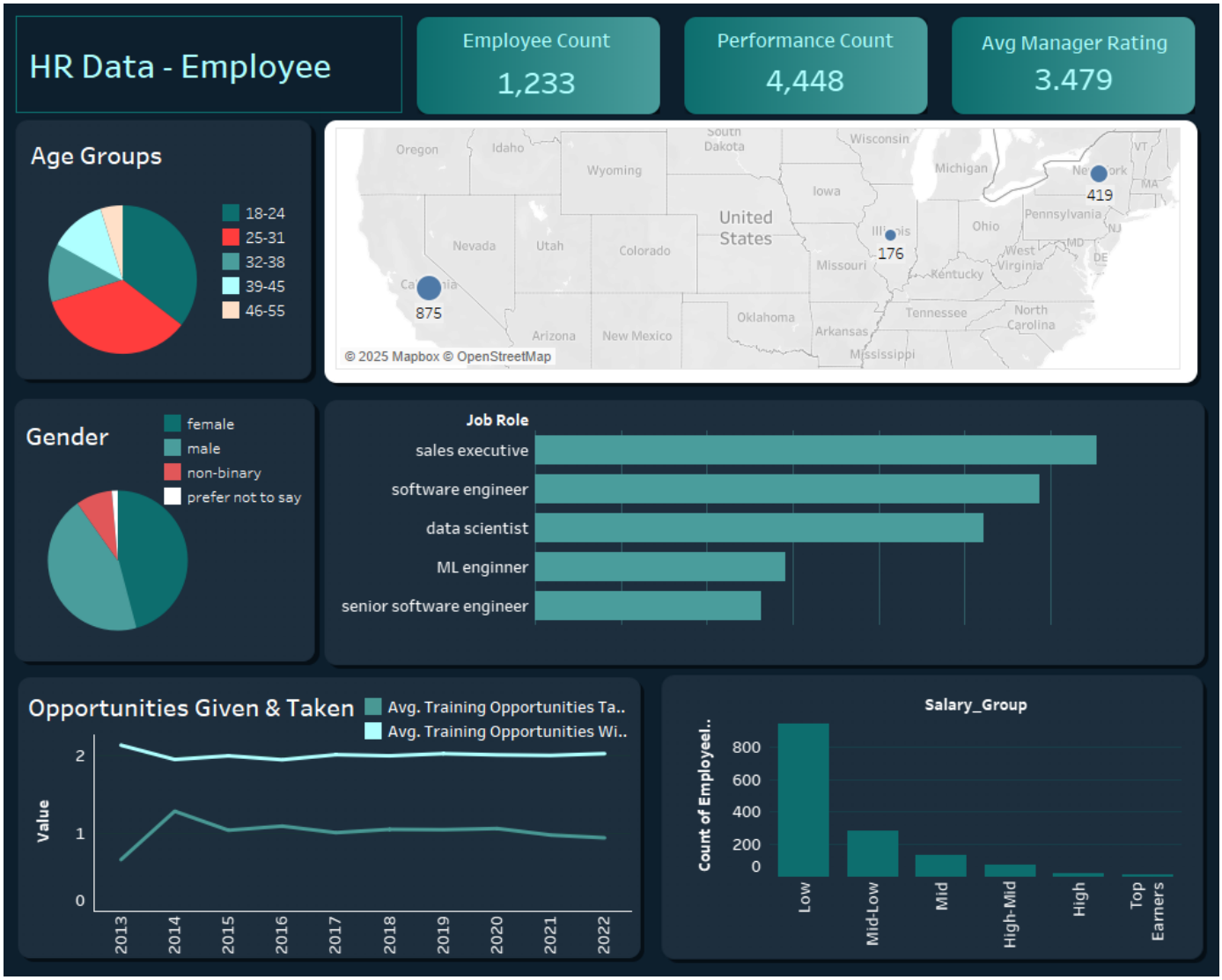
- Used Power BI and Tableau screenshots within slides.
- Simplified findings into visual storytelling for clarity.
- Included interactive elements (filters, drill-down options) for engagement.

Deliverables

1. Final Interactive Dashboard (Power BI & Tableau) for real-time exploration.
2. Structured Word Report summarizing findings, methodology, and insights.
3. Presentation slides highlighting key takeaways and recommendations.




HR Data - Attrition
Employees
1,470
Attrition
237
Attrition Rate
4%
Attrition Per Travel
Business Travel
Frequent Traveller
69
No Travel
12
Some Travel
156
0
20
40
60
80
100
120
140
160
Attrition_Count
Attrition Per Department
Department
Employee Stayed
Attrition
828
354
92
133
HR
sales
technology
Attrition Per Age
Age_Group
18-24
25-31
32-38
39-45
46-55
178
71
521
190
510
Attrition Per Overtime
No
Yes
127
289
Hire and Attrition Per Time
151
136
136
127
114
106
136
145
127
137
155
Hire
Attrition
2
10
11
12
14
21
21
32
60
54
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023


HR Data - Employee
Employee Count
1,233
Performance Count
4,448
Avg Manager Rating
3.479
Age Groups
18-24
25-31
32-38
39-45
46-55
Oregon
Idaho
Wyoming
South Dakota
Wisconsin
Michigan
Iowa
United States
Nevada
Utah
Colorado
Missouri
Illinois
Ohio
Pennsylvania
Kentucky
West Virginia
Oklahoma
Arkansas
Tennessee
North Carolina
Arizona
New Mexico
Mississippi
419
176
875
© 2025 Mapbox © OpenStreetMap
Gender
female
male
non-binary
prefer not to say
Job Role
sales executive
software engineer
data scientist
ML enginner
senior software engineer
Opportunities Given & Taken
Avg. Training Opportunities Ta..
Avg. Training Opportunities Wi..
Value
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Salary_Group
Count of Employeel..
Low
Mid-Low
Mid
High-Mid
High
Top Earners


HR ANALYTICS DASHBOARD
EMPLOYEE overview
Total Emp
1470
Avg Salary
112.96K
Employee By Department
Employee By Education
N.o Employees By JobRole
Employees By OverTime
Employee By State


timedim
Date
Day
Month
Quarter
Year
Collapse
performanceratingfact
Attrition (0,1)
employeedim.Attrition
EmployeeID
EnvironmentSatisfactionID
JobSatisfactionID
PerformanceID
RelationshipSatisfactionID
ReviewDate
SelfRatingID
Collapse
satisfactiondim
SatisfactionID
SatisfactionLevel
Collapse
ratingdim
RatingID
RatingLevel
Collapse
employeedim
Age
AgeGroup
Attrition
Attrition (0,1)
BusinessTravel
Department
DistanceFromHome
Education
EmployeeID
Ethnicity
FirstName
Gender
HireDate
HireYear
JobRole
LastName
Collapse


HR ANALYTICS
DASHBOARD
EMPLOYEE
overview
AgeGroup
Department
State
JobRole
Employee By State
Employee by StockOptionLevel
Employee By Business Travel
Employee by Age
Employee By Ethnicity


HR ANALYTICS
DASHBOARD
Performance Rating
overview
N.O Reviews
1001
N.O Employees Reviewed
573
Year
Ethnicity
Department
AgeGroup
State
JobRole
Employees By TrainingOpportunitiesTaken
Employee By Realationship Satisfaction
Employee By Work-Life Balance
EmployeeS By EnvironmentSatisfaction
Employee By Job Satisfaction
Employee By Self Rating


HR ANALYTICS
DASHBOARD
Attrition overview
Year
Ethnicity
Department
Employees By TrainingOpportunitiesTaken
Attrition Relationship Satisfaction
Employee By Work-Life Balance
Excellent
Very Good
Fair
Good
Poor
Neutral
Dissatisfied
Satisfied
Very Satisfied
EmployeeS By EnvironmentSatisfaction
Employee By Job Satisfaction
Employee By Self Rating
Employee 1
Employee 2
Performance Rating
Attrition 1
Attrition 2


HR ANALYTICS
DASHBOARD
Attrition overview
Ethnicity
Department
AgeGroup
State
Attrition By Year
Attrition By OverTime
Attrition By Ethnicity
EmployeeS By EnvironmentSatisfaction
Attrition By Department
Attrition By Age
Employee 1
Employee 2
Performance Rating
Attrition 1
Attrition 2