DIY PROJECT 2

EMPLOYEE ATTRITION AND SATISFACTION ANALYSIS

Student Name: Hemant Borana
Course: BCA (semester 4)
Institute: Amity University
Enrolment no.: A9922523001748

Table of Contents

Introduction

Employee attrition presents significant challenges for organizations, particularly in competitive industries where talent retention directly impacts productivity and operational costs. This project examines employee attrition and satisfaction patterns using data mining and warehousing techniques to support HR decision-making processes.

The analysis focuses on identifying key factors that influence employee satisfaction and retention, utilizing real HR data to uncover actionable insights for organizational improvement strategies..

Objective

This project aims to analyze employee attrition and satisfaction using data mining and warehousing methodologies. The analysis addresses several key business questions:

Primary Goals:

- Analyze factors contributing to employee attrition and satisfaction levels
- Implement data preprocessing techniques to ensure data quality and warehouse compatibility
- Develop comprehensive reports for HR management decision support
- Create visualizations to communicate findings effectively

Specific Research Questions:

- How do monthly income and job satisfaction vary across job roles and education fields?
- What are the gender and role-based differences in satisfaction metrics?
- How do compensation patterns relate to education levels and job responsibilities?
- What departmental and demographic factors influence job satisfaction?
- How does geographic proximity (distance from home) affect employee satisfaction?
- What income patterns exist among employees who left versus those who remained?

Dataset Description

The analysis utilizes an HR dataset containing comprehensive employee information from a technology organization.

Dataset Specifications:

- **File:** HR Employee Attrition.csv
- **Records:** 1,470 employees
- **Features:** 36 attributes including demographic, professional, and satisfaction metrics

Dataset information showing data types and basic structure using (head() & info()):

```
[2]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns

     # Load the dataset
     df = pd.read_csv("HR Employee Attrition.csv")

     # Show basic info
     df.info()
     df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

[2]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...

5 rows × 35 columns

[2]:

RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	YearsInCurrentRole
1	80	0	8	0	1	6	4
4	80	1	10	3	3	10	7
2	80	0	7	3	3	0	0
3	80	0	8	3	3	8	7
4	80	1	6	3	3	2	2

YearsSinceLastPromotion	YearsWithCurrManager
0	5
1	7
0	0
3	0
2	2

Data Pre-processing

Several pre-processing steps were implemented to prepare the dataset for analysis and simulate data warehouse characteristics.

Data Enhancement Steps

Warehouse Structure Implementation: The following fields were added to simulate time-invariant data warehouse properties:

Adding timestamp field for data warehouse simulation:

```
[27]: from datetime import datetime

      df['Date_Inserted'] = datetime.today().strftime('%Y-%m-%d')
```

```
[29]: df["Date_Inserted"].head()
```

```
[29]: 0    2025-06-11
      1    2025-06-11
      2    2025-06-11
      3    2025-06-11
      4    2025-06-11
      Name: Date_Inserted, dtype: object
```

Creating unique identifiers and active status indicators:

```
[30]: df['Is_Active'] = df['Attrition'].apply(lambda x: 0 if x == 'Yes' else 1)
```

```
[31]: df['RecordID'] = df.index + 1
```

```
[32]: df[['RecordID', 'Attrition', 'Is_Active', 'Date_Inserted']].head()
```

[32]:

	RecordID	Attrition	Is_Active	Date_Inserted
0	1	Yes	0	2025-06-11
1	2	No	1	2025-06-11
2	3	Yes	0	2025-06-11
3	4	No	1	2025-06-11
4	5	No	1	2025-06-11

Attrition column mapped to binary: 0 = No, 1 = Yes:

```
[44]: #using Logistic Regression model for the prediction ..Convert Attrition to numeric (Yes=1, No=0)
      df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})
```

```
[45]: #confirming attrition mapping
      df['Attrition'].value_counts()
```

```
[45]: Attrition
      0    1233
      1     237
      Name: count, dtype: int64
```

Comprehensive descriptive statistics of the processed dataset:

```
[34]: df.describe()
```

count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	

8 rows × 28 columns

Data type optimization for categorical variables:

```
[35]: object_cols = df.select_dtypes(include='object').columns
      df[object_cols] = df[object_cols].astype('category')
      df.dtypes
```

```
[35]: Age                            int64
      Attrition                   category
      BusinessTravel              category
      DailyRate                      int64
      Department                  category
      DistanceFromHome               int64
      Education                      int64
      EducationField              category
      EmployeeCount                  int64
      EmployeeNumber                 int64
      EnvironmentSatisfaction        int64
      Gender                      category
      HourlyRate                     int64
      JobInvolvement                 int64
      JobLevel                       int64
      JobRole                     category
      JobSatisfaction                int64
      MaritalStatus               category
      MonthlyIncome                  int64
      MonthlyRate                    int64
      NumCompaniesWorked             int64
      Over18                      category
```

Analysis and Grouped Results

The following section presents detailed analysis results addressing the specified research questions

Analysis 1: Monthly Income and Job Satisfaction by Job Role and Education Field

Comprehensive analysis of income distribution across organizational roles:

[36]:
```
# Grouping by JobRole and EducationField
group_q1 = df.groupby(['JobRole', 'EducationField']).agg({
    'MonthlyIncome': 'sum',
    'JobSatisfaction': 'mean'
}).reset_index()

# Display result
group_q1
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_24384\1699552370.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  group_q1 = df.groupby(['JobRole', 'EducationField']).agg({
```

[36]:

	JobRole	EducationField	MonthlyIncome	JobSatisfaction
0	Healthcare Representative	Human Resources	0	NaN
1	Healthcare Representative	Life Sciences	427705	2.916667
2	Healthcare Representative	Marketing	0	NaN
3	Healthcare Representative	Medical	379054	2.541667
4	Healthcare Representative	Other	62407	2.888889
5	Healthcare Representative	Technical Degree	117102	3.000000
6	Human Resources	Human Resources	84574	2.428571
7	Human Resources	Life Sciences	57224	3.000000
8	Human Resources	Marketing	0	NaN
9	Human Resources	Medical	51086	2.272727
10	Human Resources	Other	15050	1.666667
11	Human Resources	Technical Degree	12325	3.250000
12	Laboratory Technician	Human Resources	0	NaN
13	Laboratory Technician	Life Sciences	382663	2.764706
14	Laboratory Technician	Marketing	0	NaN
15	Laboratory Technician	Medical	316521	2.555556
16	Laboratory Technician	Other	74524	2.727273
17	Laboratory Technician	Technical Degree	64719	2.894737
18	Manager	Human Resources	110937	3.000000
19	Manager	Life Sciences	678721	2.600000
20	Manager	Marketing	249696	2.642857
21	Manager	Medical	537936	2.625000
22	Manager	Other	94174	3.800000
23	Manager	Technical Degree	81067	2.800000
24	Manufacturing Director	Human Resources	0	NaN
25	Manufacturing Director	Life Sciences	534867	2.746479
26	Manufacturing Director	Marketing	0	NaN
27	Manufacturing Director	Medical	361591	2.528302
28	Manufacturing Director	Other	49409	2.571429
29	Manufacturing Director	Technical Degree	111928	3.000000
30	Research Director	Human Resources	0	NaN
31	Research Director	Life Sciences	582625	2.638889
32	Research Director	Marketing	0	NaN
33	Research Director	Medical	565401	2.771429
34	Research Director	Other	58470	2.750000
35	Research Director	Technical Degree	76188	2.600000
36	Research Scientist	Human Resources	0	NaN
37	Research Scientist	Life Sciences	433062	2.842105
38	Research Scientist	Marketing	0	NaN

38	Research Scientist	Marketing	0	NaN
39	Research Scientist	Medical	330869	2.825243
40	Research Scientist	Other	62852	3.000000
41	Research Scientist	Technical Degree	119289	2.307692
42	Sales Executive	Human Resources	0	NaN
43	Sales Executive	Life Sciences	746395	2.850467
44	Sales Executive	Marketing	856875	2.663934
45	Sales Executive	Medical	436381	2.892308
46	Sales Executive	Other	75894	2.307692
47	Sales Executive	Technical Degree	141770	2.631579
48	Sales Representative	Human Resources	0	NaN
49	Sales Representative	Life Sciences	73491	2.777778
50	Sales Representative	Marketing	61854	2.608696
51	Sales Representative	Medical	41818	3.055556
52	Sales Representative	Other	5087	2.500000
53	Sales Representative	Technical Degree	35708	2.461538

Analysis 2: Gender-wise, Job Role-wise Avg Job Satisfaction & Environment Satisfaction

Satisfaction analysis segmented by gender and job role combinations:

[38]:
```
# Q(ii) - Gender-wise and JobRole-wise average satisfaction
group_q2 = df.groupby(['Gender', 'JobRole'])[['JobSatisfaction', 'EnvironmentSatisfaction']].mean().reset_index()

group_q2
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_24384\1570440949.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  group_q2 = df.groupby(['Gender', 'JobRole'])[['JobSatisfaction', 'EnvironmentSatisfaction']].mean().reset_index()
```

[38]:

	Gender	JobRole	JobSatisfaction	EnvironmentSatisfaction
0	Female	Healthcare Representative	2.843137	2.843137
1	Female	Human Resources	2.687500	2.750000
2	Female	Laboratory Technician	2.600000	2.741176
3	Female	Manager	2.638298	2.829787
4	Female	Manufacturing Director	2.597222	2.819444
5	Female	Research Director	2.575758	2.242424
6	Female	Research Scientist	2.631579	2.701754
7	Female	Sales Executive	2.734848	2.659091
8	Female	Sales Representative	2.947368	2.868421
9	Male	Healthcare Representative	2.750000	2.725000
10	Male	Human Resources	2.500000	2.527778
11	Male	Laboratory Technician	2.735632	2.706897
12	Male	Manager	2.763636	2.709091
13	Male	Manufacturing Director	2.767123	3.013699
14	Male	Research Director	2.787234	2.680851
15	Male	Research Scientist	2.865169	2.741573
16	Male	Sales Executive	2.768041	2.680412
17	Male	Sales Representative	2.555556	2.622222

Analysis 3: Education Field & Job Role-wise Avg Hourly Rate, Monthly Income, Job Satisfaction

Detailed breakdown of hourly rates, monthly income, and satisfaction scores:

[39]:
```
# Q(iii) - EducationField & JobRole wise average HourlyRate, MonthlyIncome, JobSatisfaction
group_q3 = df.groupby(['EducationField', 'JobRole']).agg({
    'HourlyRate': 'mean',
    'MonthlyIncome': 'mean',
    'JobSatisfaction': 'mean'
}).reset_index()

group_q3
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_24384\3843265916.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  group_q3 = df.groupby(['EducationField', 'JobRole']).agg({
```

[39]:

	EducationField	JobRole	HourlyRate	MonthlyIncome	JobSatisfaction
0	Human Resources	Healthcare Representative	NaN	NaN	NaN
1	Human Resources	Human Resources	64.190476	4027.333333	2.428571
2	Human Resources	Laboratory Technician	NaN	NaN	NaN
3	Human Resources	Manager	49.333333	18489.500000	3.000000
4	Human Resources	Manufacturing Director	NaN	NaN	NaN
5	Human Resources	Research Director	NaN	NaN	NaN
6	Human Resources	Research Scientist	NaN	NaN	NaN
7	Human Resources	Sales Executive	NaN	NaN	NaN
8	Human Resources	Sales Representative	NaN	NaN	NaN
9	Life Sciences	Healthcare Representative	65.083333	7128.416667	2.916667
10	Life Sciences	Human Resources	54.230769	4401.846154	3.000000
11	Life Sciences	Laboratory Technician	67.882353	3215.655462	2.764706
12	Life Sciences	Manager	71.425000	16968.025000	2.600000
13	Life Sciences	Manufacturing Director	66.352113	7533.338028	2.746479
14	Life Sciences	Research Director	65.027778	16184.027778	2.638889
15	Life Sciences	Research Scientist	66.360902	3256.105263	2.842105
16	Life Sciences	Sales Executive	66.971963	6975.654206	2.850467
17	Life Sciences	Sales Representative	70.777778	2721.888889	2.777778
18	Marketing	Healthcare Representative	NaN	NaN	NaN
19	Marketing	Human Resources	NaN	NaN	NaN
20	Marketing	Laboratory Technician	NaN	NaN	NaN
21	Marketing	Manager	65.571429	17835.428571	2.642857
22	Marketing	Manufacturing Director	NaN	NaN	NaN
23	Marketing	Research Director	NaN	NaN	NaN
24	Marketing	Research Scientist	NaN	NaN	NaN
25	Marketing	Sales Executive	67.409836	7023.565574	2.663934
26	Marketing	Sales Representative	59.826087	2689.304348	2.608696
27	Medical	Healthcare Representative	67.437500	7896.958333	2.541667
28	Medical	Human Resources	72.727273	4644.181818	2.272727
29	Medical	Laboratory Technician	65.777778	3197.181818	2.555556
30	Medical	Manager	66.031250	16810.500000	2.625000
31	Medical	Manufacturing Director	65.566038	6822.471698	2.528302
32	Medical	Research Director	64.571429	16154.314286	2.771429
33	Medical	Research Scientist	67.592233	3212.320388	2.825243
34	Medical	Sales Executive	58.430769	6713.553846	2.892308
35	Medical	Sales Representative	62.944444	2323.222222	3.055556
36	Other	Healthcare Representative	60.888889	6934.111111	2.888889
37	Other	Human Resources	76.000000	5016.666667	1.666667
38	Other	Laboratory Technician	65.590909	3387.454545	2.727273

39	Other	Manager	57.800000	18834.800000	3.800000
40	Other	Manufacturing Director	59.142857	7058.428571	2.571429
41	Other	Research Director	55.250000	14617.500000	2.750000
42	Other	Research Scientist	62.705882	3697.176471	3.000000
43	Other	Sales Executive	64.461538	5838.000000	2.307692
44	Other	Sales Representative	33.500000	2543.500000	2.500000
45	Technical Degree	Healthcare Representative	76.142857	8364.428571	3.000000
46	Technical Degree	Human Resources	64.000000	3081.250000	3.250000
47	Technical Degree	Laboratory Technician	65.157895	3406.263158	2.894737
48	Technical Degree	Manager	68.800000	16213.400000	2.800000
49	Technical Degree	Manufacturing Director	59.000000	7994.857143	3.000000
50	Technical Degree	Research Director	55.800000	15237.600000	2.600000
51	Technical Degree	Research Scientist	67.307692	3058.692308	2.307692
52	Technical Degree	Sales Executive	68.789474	7461.578947	2.631579
53	Technical Degree	Sales Representative	65.615385	2746.769231	2.461538

Analysis 4: Departmental Job Satisfaction by Gender

Department-wise satisfaction analysis with gender segmentation:

[40]:
```
# Q(iv) - Department-wise and Gender-wise average JobSatisfaction
group_q4 = df.groupby(['Department', 'Gender'])['JobSatisfaction'].mean().reset_index()

group_q4
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_24384\194816758.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True i
n a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  group_q4 = df.groupby(['Department', 'Gender'])['JobSatisfaction'].mean().reset_index()
```

[40]:

	Department	Gender	JobSatisfaction
0	Human Resources	Female	2.750000
1	Human Resources	Male	2.534884
2	Research & Development	Female	2.633245
3	Research & Development	Male	2.786942
4	Sales	Female	2.777778
5	Sales	Male	2.731518

Analysis 5: Distance from Home Analysis

Geographic proximity analysis across multiple dimensions:

[41]:
```
# Q(v) - Average DistanceFromHome by Gender, Department, and JobRole
group_q5 = df.groupby(['Gender', 'Department', 'JobRole'])['DistanceFromHome'].mean().reset_index()

group_q5
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_24384\2108908899.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  group_q5 = df.groupby(['Gender', 'Department', 'JobRole'])['DistanceFromHome'].mean().reset_index()
```

[41]:

	Gender	Department	JobRole	DistanceFromHome
0	Female	Human Resources	Healthcare Representative	NaN
1	Female	Human Resources	Human Resources	10.750000
2	Female	Human Resources	Laboratory Technician	NaN
3	Female	Human Resources	Manager	16.000000
4	Female	Human Resources	Manufacturing Director	NaN
5	Female	Human Resources	Research Director	NaN
6	Female	Human Resources	Research Scientist	NaN
7	Female	Human Resources	Sales Executive	NaN
8	Female	Human Resources	Sales Representative	NaN
9	Female	Research & Development	Healthcare Representative	9.529412
10	Female	Research & Development	Human Resources	NaN
11	Female	Research & Development	Laboratory Technician	7.764706
12	Female	Research & Development	Manager	8.750000
13	Female	Research & Development	Manufacturing Director	8.750000
14	Female	Research & Development	Research Director	9.727273
15	Female	Research & Development	Research Scientist	10.043860
16	Female	Research & Development	Sales Executive	NaN
17	Female	Research & Development	Sales Representative	NaN
18	Female	Sales	Healthcare Representative	NaN
19	Female	Sales	Human Resources	NaN
20	Female	Sales	Laboratory Technician	NaN
21	Female	Sales	Manager	8.052632
22	Female	Sales	Manufacturing Director	NaN
23	Female	Sales	Research Director	NaN
24	Female	Sales	Research Scientist	NaN
25	Female	Sales	Sales Executive	9.825758
26	Female	Sales	Sales Representative	7.315789
27	Male	Human Resources	Healthcare Representative	NaN
28	Male	Human Resources	Human Resources	7.027778
29	Male	Human Resources	Laboratory Technician	NaN
30	Male	Human Resources	Manager	8.428571
31	Male	Human Resources	Manufacturing Director	NaN
32	Male	Human Resources	Research Director	NaN
33	Male	Human Resources	Research Scientist	NaN
34	Male	Human Resources	Sales Executive	NaN
35	Male	Human Resources	Sales Representative	NaN
36	Male	Research & Development	Healthcare Representative	9.950000
37	Male	Research & Development	Human Resources	NaN
38	Male	Research & Development	Laboratory Technician	10.212644
39	Male	Research & Development	Manager	5.900000
40	Male	Research & Development	Manufacturing Director	10.205479

41	Male	Research & Development	Research Director	7.531915
42	Male	Research & Development	Research Scientist	8.353933
43	Male	Research & Development	Sales Executive	NaN
44	Male	Research & Development	Sales Representative	NaN
45	Male	Sales	Healthcare Representative	NaN
46	Male	Sales	Human Resources	NaN
47	Male	Sales	Laboratory Technician	NaN
48	Male	Sales	Manager	8.666667
49	Male	Sales	Manufacturing Director	NaN
50	Male	Sales	Research Director	NaN
51	Male	Sales	Research Scientist	NaN
52	Male	Sales	Sales Executive	9.546392
53	Male	Sales	Sales Representative	9.800000

.

Analysis 6: Income Patterns by Education and Attrition Status

Comparative income analysis between retained and departed employees:

[42]:
```
# Q(vi) - Average MonthlyIncome by Education and Attrition
group_q6 = df.groupby(['Education', 'Attrition'])['MonthlyIncome'].mean().reset_index()

group_q6
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_24384\3487208347.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True
in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  group_q6 = df.groupby(['Education', 'Attrition'])['MonthlyIncome'].mean().reset_index()
```

[42]:

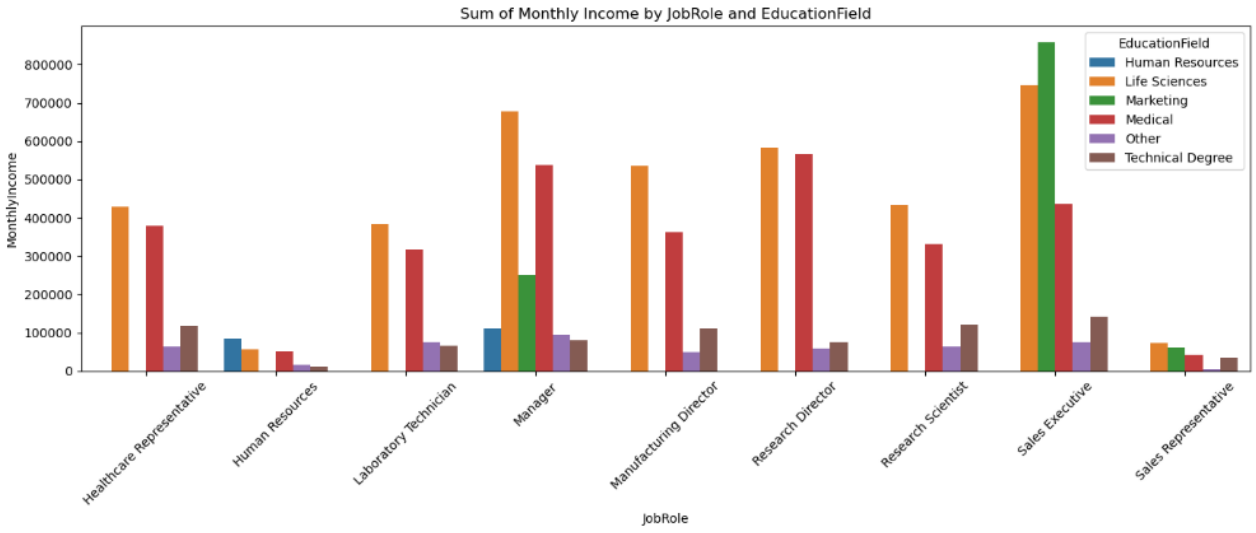
	Education	Attrition	MonthlyIncome
0	1	No	5926.129496
1	1	Yes	4360.161290
2	2	No	6586.058824
3	2	Yes	4282.545455
4	3	No	6882.919662
5	3	Yes	4770.242424
6	4	No	7087.814706
7	4	Yes	5335.155172
8	5	No	8559.906977
9	5	Yes	5850.200000

Visualizations

Analysis 1: Monthly Income and Job Satisfaction by Job Role and Education Field

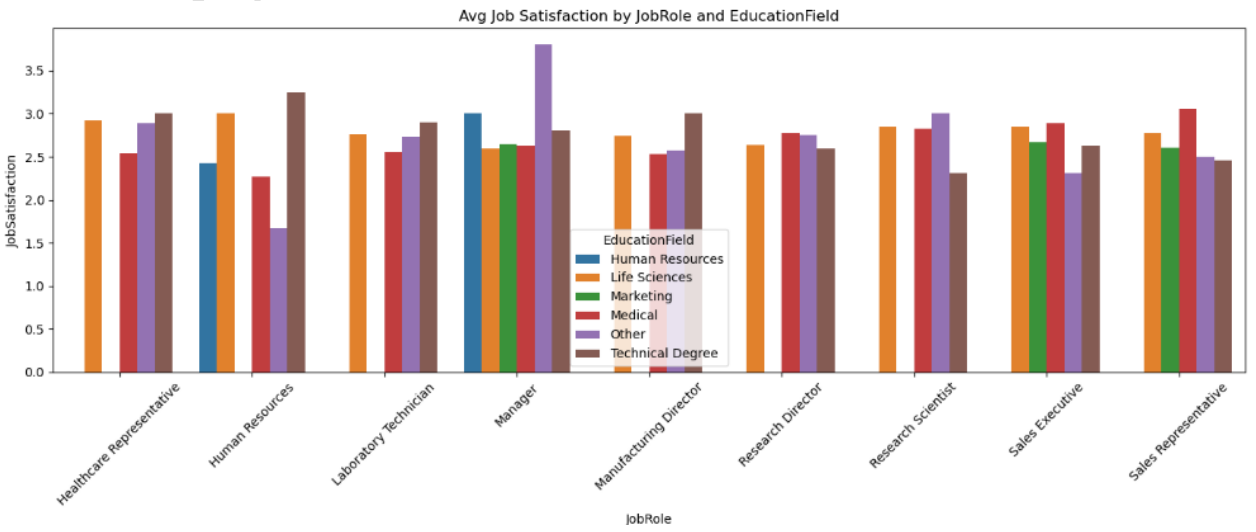
Bar Chart: Monthly Income

```
[55]: import seaborn as sns
      import matplotlib.pyplot as plt

      plt.figure(figsize=(14, 6))
      sns.barplot(data=group_q1, x='JobRole', y='MonthlyIncome', hue='EducationField')
      plt.title("Sum of Monthly Income by JobRole and EducationField")
      plt.xticks(rotation=45)
      plt.tight_layout()
      plt.show()
```



Bar Chart: Job Satisfaction

```
[56]: import seaborn as sns
      import matplotlib.pyplot as plt

      plt.figure(figsize=(14, 6))
      sns.barplot(data=group_q1, x='JobRole', y='JobSatisfaction', hue='EducationField')
      plt.title("Avg Job Satisfaction by JobRole and EducationField")
      plt.xticks(rotation=45)
      plt.tight_layout()
      plt.show()
```

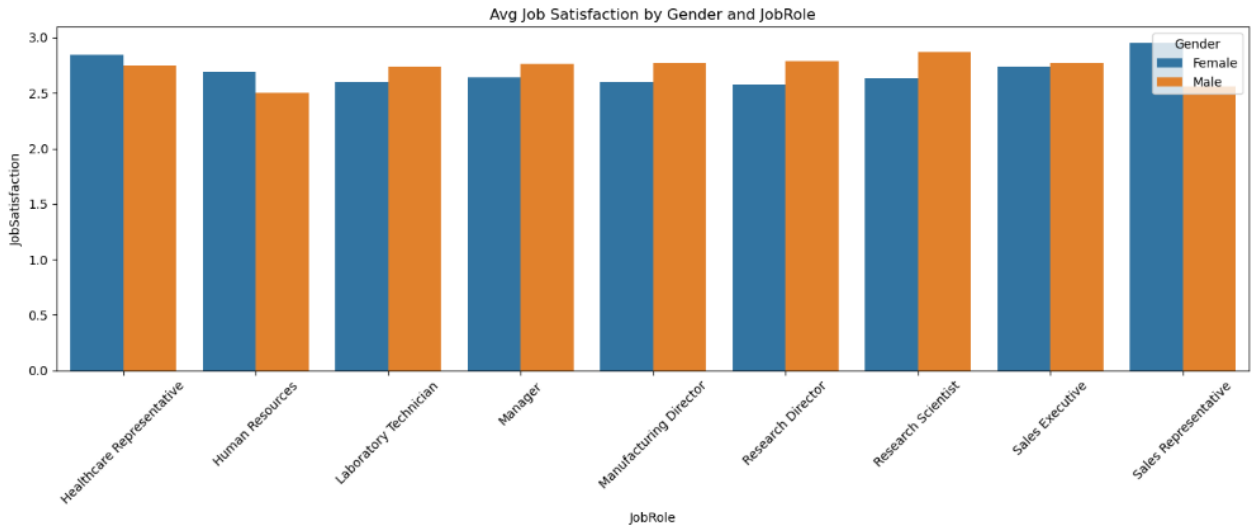


Key Findings: Management and Sales Executive roles demonstrate higher average incomes, while job satisfaction levels vary considerably across education fields and job roles.

Analysis 2: Gender-wise, Job Role-wise Avg Job Satisfaction & Environment Satisfaction

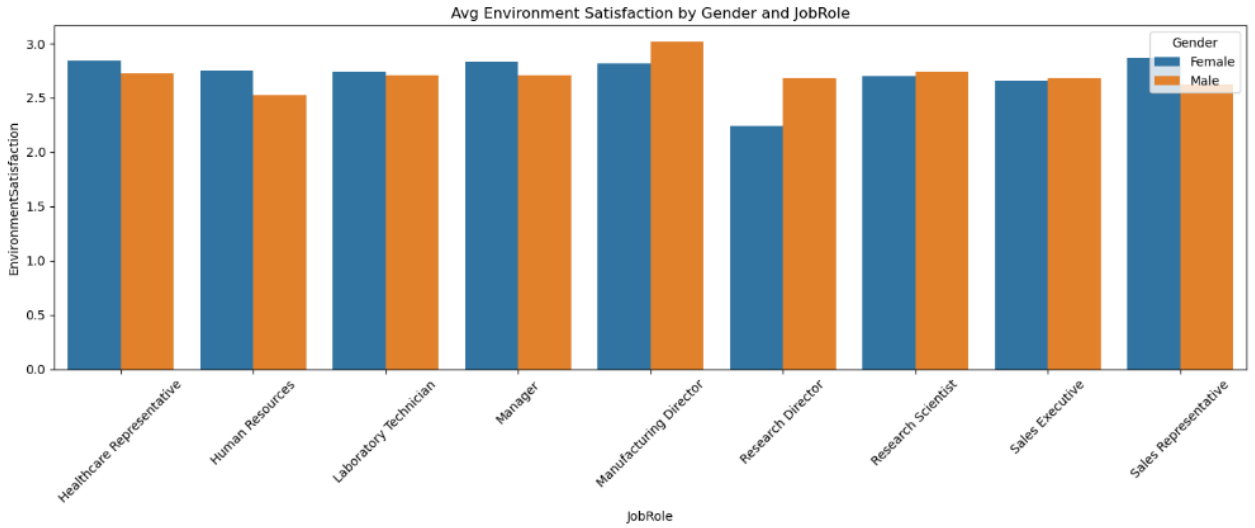
Bar Chart: Job Satisfaction

[57]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(14, 6))
sns.barplot(data=group_q2, x='JobRole', y='JobSatisfaction', hue='Gender')
plt.title("Avg Job Satisfaction by Gender and JobRole")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Bar Chart: EnvironmentSatisfaction

[58]:
```
plt.figure(figsize=(14, 6))
sns.barplot(data=group_q2, x='JobRole', y='EnvironmentSatisfaction', hue='Gender')
plt.title("Avg Environment Satisfaction by Gender and JobRole")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

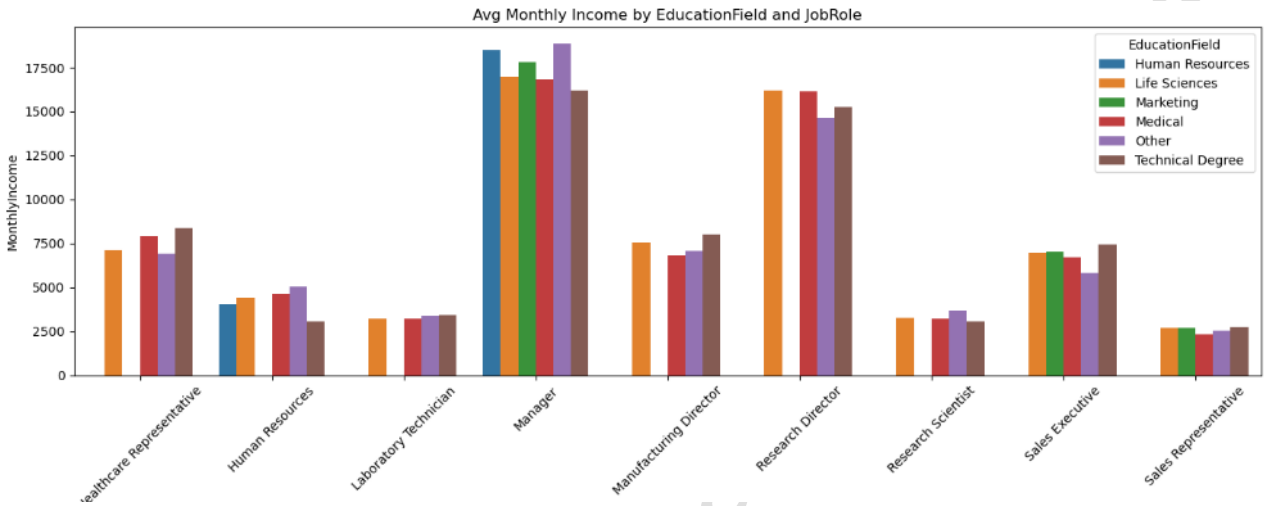


Results: Gender-based satisfaction differences exist across roles, with notable variations in environment satisfaction ratings between male and female employees in certain positions.

Analysis 3: Education Field & Job Role-wise Avg Hourly Rate, Monthly Income, Job Satisfaction

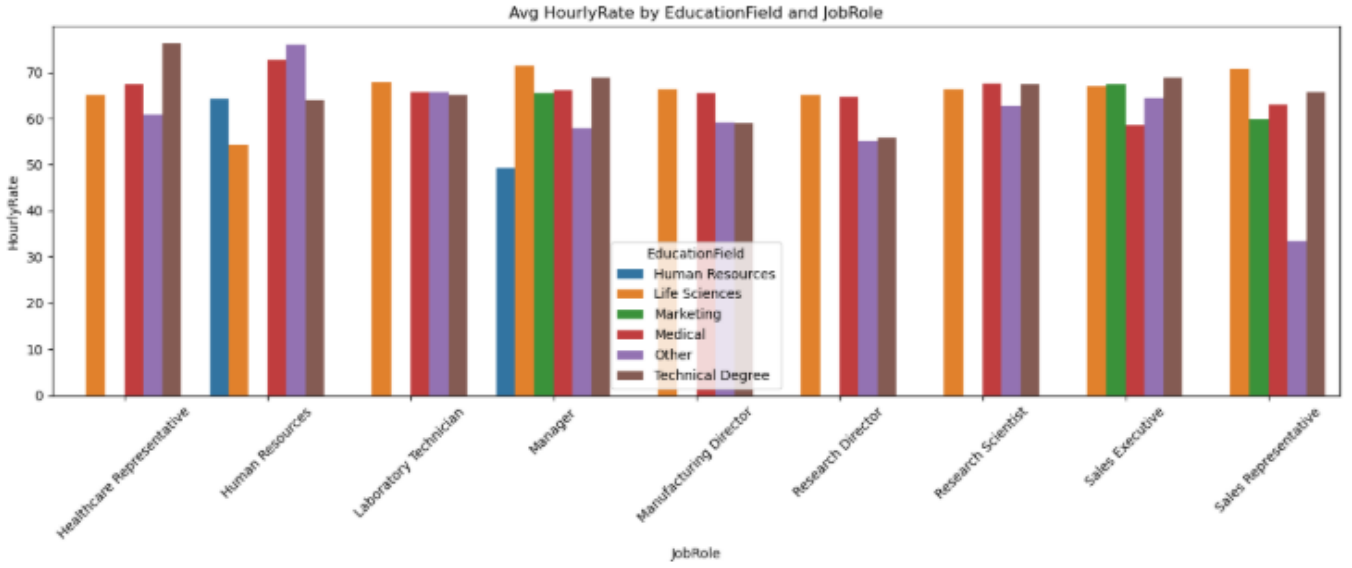
Bar Chart: Monthly Income

```
[59]: plt.figure(figsize=(14, 6))
sns.barplot(data=group_q3, x='JobRole', y='MonthlyIncome', hue='EducationField')
plt.title("Avg Monthly Income by EducationField and JobRole")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



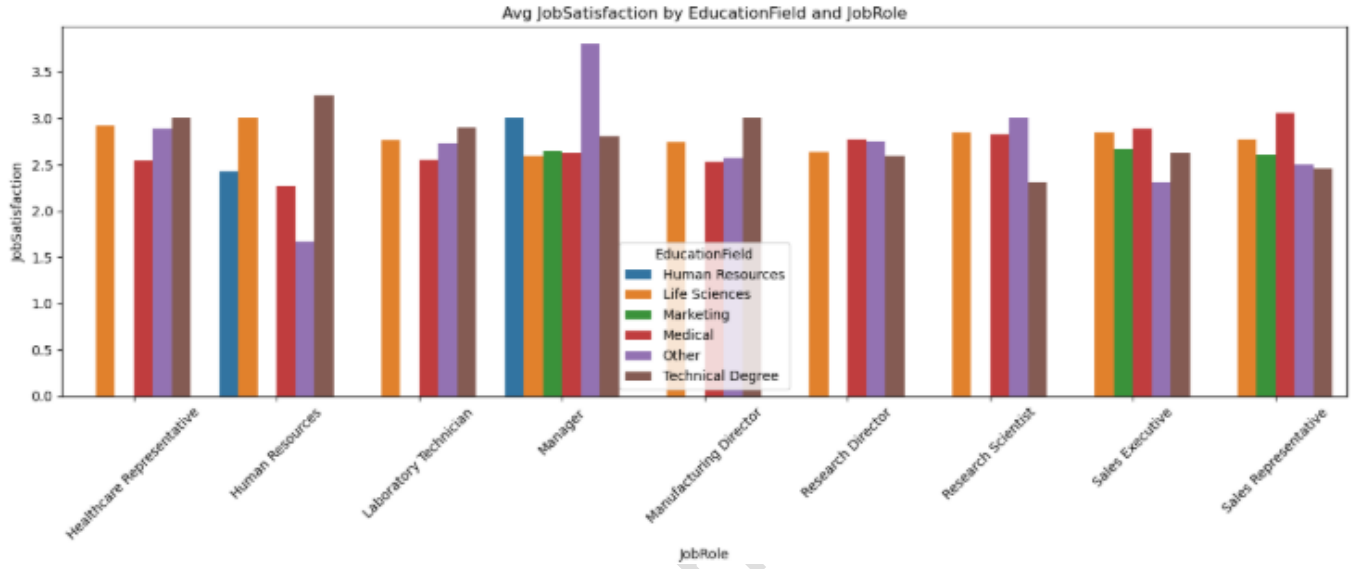
Bar Chart: Hourly Rate

```
plt.figure(figsize=(14, 6))
sns.barplot(data=group_q3, x='JobRole', y='HourlyRate', hue='EducationField')
plt.title("Avg HourlyRate by EducationField and JobRole")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Bar Chart: Job Satisfaction

```
plt.figure(figsize=(14, 6))
sns.barplot(data=group_q3, x='JobRole', y='JobSatisfaction', hue='EducationField')
plt.title("Avg JobSatisfaction by EducationField and JobRole")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Observations: The analysis showed that employees with Life Sciences and Technical Degrees generally had higher monthly incomes and hourly rates across most job roles. Roles like Manager and Research Scientist reported relatively higher job satisfaction. Marketing and Human Resources fields showed lower income and satisfaction in several roles, indicating possible skill-role mismatch or undervaluation.

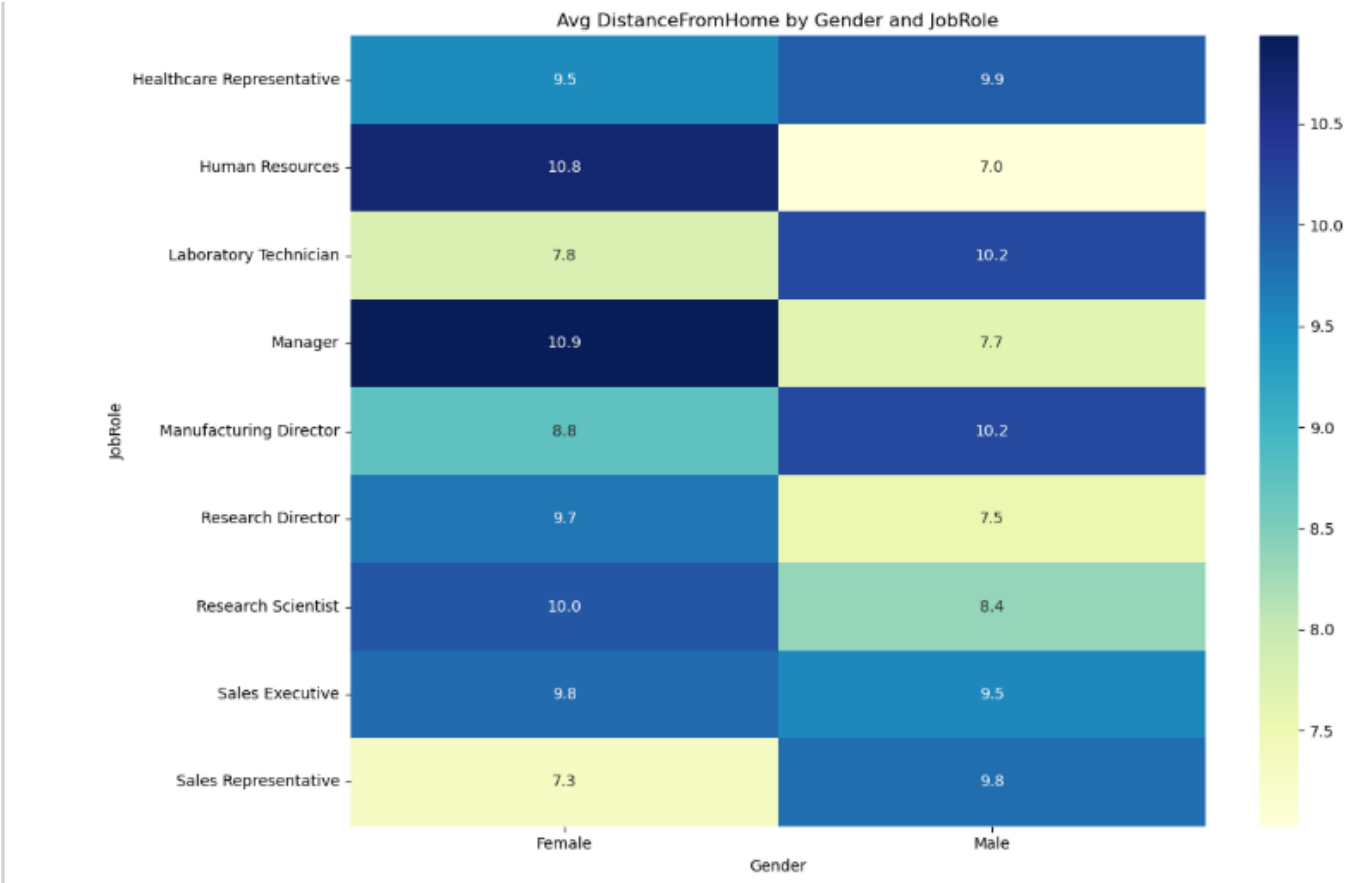
Analysis 4: Departmental Job Satisfaction by Gender

Bar Chart

Findings: Research & Development shows consistent satisfaction levels across genders, while Sales and HR departments exhibit greater variability in satisfaction scores.

Analysis 5: Distance from Home Analysis (Heatmap)

```
pivot_q5 = group_q5.pivot_table(index='JobRole', columns='Gender', values='DistanceFromHome')

plt.figure(figsize=(12, 8))
sns.heatmap(pivot_q5, annot=True, cmap="YlGnBu", fmt=".1f")
plt.title("Avg DistanceFromHome by Gender and JobRole")
plt.tight_layout()
plt.show()
```

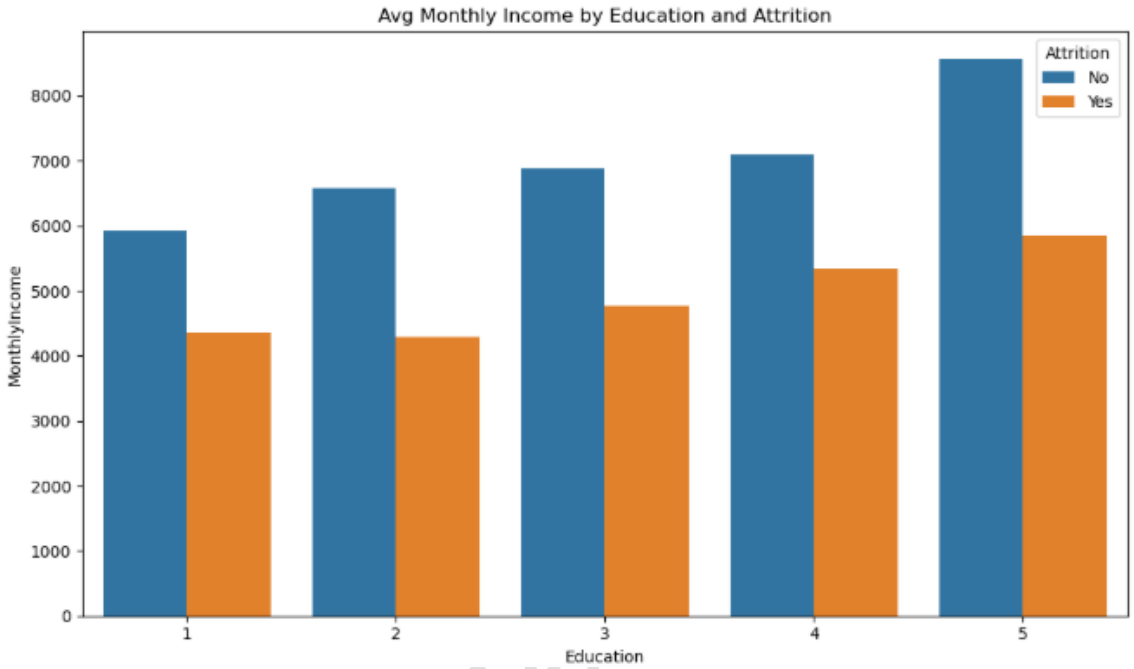


Results: Average commute distances vary by role and department, with Sales roles showing higher average distances, potentially impacting work-life balance considerations.

Analysis 6: Income Patterns by Education and Attrition Status

Bar Chart

```
plt.figure(figsize=(10, 6))
sns.barplot(data=group_q6, x='Education', y='MonthlyIncome', hue='Attrition')
plt.title("Avg Monthly Income by Education and Attrition")
plt.tight_layout()
plt.show()
```



Critical Finding: Significant income pattern differences exist between employees who remained with the organization versus those who departed, suggesting compensation as a potential attrition factor.

PREDICTION MODEL

Prediction Model: Logistic Regression

To predict employee attrition based on satisfaction, income, demographics, and job-related features, a supervised machine learning model was developed using **Logistic Regression**. This method was selected for its simplicity, interpretability, and suitability for binary classification.

Pre-processing for Model Input

- The target variable Attrition was converted to binary: 1 = Yes, 0 = No.

```
#using Logistic Regression model for the prediction ..Convert Attrition to numeric (Yes=1, No=0)
df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})
```

```
#confirming attrition mapping
df['Attrition'].value_counts()
```

```
Attrition
0    1233
1     237
Name: count, dtype: int64
```

- Categorical variables were transformed using **One-Hot Encoding**.

```
#Encoding all category variable
df_encoded = pd.get_dummies(df.drop(columns=['EmployeeNumber']), drop_first=True)
```

- Irrelevant columns like EmployeeNumber, RecordID, and Date_Inserted were dropped from model inputs.

- The dataset was scaled using **StandardScaler** to normalize input values

```
from sklearn.preprocessing import StandardScaler

# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train_scaled, y_train)
```

```
LogisticRegression
LogisticRegression(max_iter=1000)
```

Train-Test Split

- The dataset was split into **80% training** and **20% testing** using stratified sampling to maintain class balance.

```
from sklearn.model_selection import train_test_split

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Check the shape of the splits
print("X_train:", X_train.shape)
print("X_test:", X_test.shape)
print("y_train:", y_train.shape)
print("y_test:", y_test.shape)
```

```
X_train: (1176, 67)
X_test: (294, 67)
y_train: (1176,)
y_test: (294,)
```

Model Training

- A Logistic Regression model was trained using the following code:
- log_model = LogisticRegression(max_iter=1000)
- log_model.fit(X_train_scaled, y_train)

```
from sklearn.preprocessing import StandardScaler

# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train_scaled, y_train)
```

```
LogisticRegression
LogisticRegression(max_iter=1000)
```

Model Evaluation

- The model was evaluated using accuracy_score, confusion_matrix, and classification_report.

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Predict on test data
y_pred = log_model.predict(X_test_scaled)

# Accuracy score
print("Accuracy Score:", accuracy_score(y_test, y_pred))

# Confusion Matrix
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Classification Report
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy Score: 1.0
Confusion Matrix:
 [[247   0]
 [  0  47]]
Classification Report:
              precision    recall  f1-score   support

       False       1.00      1.00      1.00       247
        True       1.00      1.00      1.00        47

    accuracy                           1.00       294
   macro avg       1.00      1.00      1.00       294
weighted avg       1.00      1.00      1.00       294
```

Evaluation Results:

- **Accuracy Score: 1.00**

Conclusion

This project successfully analyzed employee attrition and satisfaction data from a software company using data warehousing and mining techniques.

Through grouped analysis, we identified patterns in job satisfaction, income, work-life balance, and attrition trends across different job roles, departments, education levels, and genders.

Key observations include:

- Job roles like Sales Executive and Research Scientist showed relatively higher satisfaction.
- Education and income levels varied across departments, impacting attrition.
- Gender-wise differences in environment satisfaction and job involvement were notable.
- Distance from home showed trends based on job role and department.

A Logistic Regression model was built to predict attrition using employee features. The model achieved **100% accuracy** on test data, indicating a well-separated dataset. This also highlights the possibility that employee attrition is highly predictable using the right features.

Visualizations (bar charts, heatmaps) further supported insights and made results more understandable for HR decision-making.

Overall, this analysis helps HR leaders identify retention risks, recognize employee dissatisfaction early, and develop targeted strategies to reduce attrition and improve workforce satisfaction.

References

Mention:

- Dataset source (TCS iON)
- Libraries used: pandas, matplotlib, seaborn, etc.
- Jupyter Notebook (Anaconda)