HR Dataset Analysis Project

Context:

The transition from higher education to employment is a critical phase for graduates. Institutions in Singapore, such as universities and specialized colleges, produce a diverse pool of talent each year. However, the employment outcomes (including employment rates and salaries) vary significantly across fields of study, universities, and individual demographic factors. Recently, we have been reading news about premature retrenchments from many companies, especially those from the tech sector. Meanwhile, there is an increasing trend of graduates not finding jobs as per reported by The Straits Times. Although our chosen dataset is not a local dataset, understanding these trends is essential for enhancing educational programs, supporting graduates, and aligning their skills with market demands. We chose this dataset due it's extensive number of records and diverse predictors that can truly help us to find as many factors as possible that can help those seeking employment.

Problem Statement

What factors significantly influence graduate employment outcomes amid a more competitive job market?

Objective:

To address this gap, we aim to leverage predictive analytics and machine learning techniques to analyze factors influencing graduate employment outcomes. This project seeks to identify key trends and predictors that can be used to forecast the following:

1. **Attrition**: If an employee has left the company, regardless of cause, i.e. retrenched, resigned, etc.
2. **MonthlyIncome**: Prediction of monthly income for graduates.

In this notebook, we will:

- **Load and inspect** the HR dataset.
- **Clean and prepare** the data (including type conversion and handling duplicates).
- **Detect outliers** in numerical features.
- **Engineer new features** (for example, creating tenure buckets).
- **Perform Exploratory Data Analysis (EDA)** including univariate, categorical, and bivariate analyses.
- **Save the cleaned data** for further modeling if needed.

The dataset includes features like Age, Attrition, BusinessTravel, DailyRate, Department, DistanceFromHome, and many more.

Data Loading & Initial Inspection

We start by importing the necessary libraries and loading the dataset from a CSV file. Then we inspect the first few rows and check basic information.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Set plotting style and default figure size
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (12, 8)

# Load the dataset (ensure 'hr_data.csv' is in your working directory)
df = pd.read_csv("data.csv")

# Display the first few rows
print("Head of the DataFrame:")
df.head()
```

```
Head of the DataFrame:

   Age Attrition     BusinessTravel  DailyRate              Department
\
0   41       Yes      Travel_Rarely       1102                   Sales

1   49        No  Travel_Frequently        279  Research & Development

2   37       Yes      Travel_Rarely       1373  Research & Development

3   33        No  Travel_Frequently       1392  Research & Development

4   27        No      Travel_Rarely        591  Research & Development


   DistanceFromHome  Education EducationField  EmployeeCount
EmployeeNumber  \
0                 1          2  Life Sciences              1
1
1                 8          1  Life Sciences              1
2
2                 2          2          Other              1
4
3                 3          4  Life Sciences              1
5
4                 2          1        Medical              1
7

   ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
```

```
0  ...                           1           80                0
1  ...                           4           80                1
2  ...                           2           80                0
3  ...                           3           80                0
4  ...                           4           80                1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance
YearsAtCompany  \
0                  8                      0               1
6
1                 10                      3               3
10
2                  7                      3               3
0
3                  8                      3               3
8
4                  6                      3               3
2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

Data Inspection

We examine the dataset's structure, check data types, and look for missing values.

```
# DataFrame basic information
print("\nDataFrame Info:")
print(df.info())

# Summary statistics for numerical features
print("\nSummary Statistics (numerical features):")
print(df.describe())

# Check for missing values in each column
print("\nMissing values by column:")
print(df.isnull().sum())
```

```
DataFrame Info:
<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
None

Summary Statistics (numerical features):
               Age    DailyRate  DistanceFromHome    Education  
EmployeeCount  \
count  1470.000000  1470.000000       1470.000000  1470.000000  
1470.0
mean     36.923810   802.485714          9.192517     2.912925  
1.0
std       9.135373   403.509100          8.106864     1.024165  
```

```
0.0
min      18.000000   102.000000          1.000000     1.000000
1.0
25%      30.000000   465.000000          2.000000     2.000000
1.0
50%      36.000000   802.000000          7.000000     3.000000
1.0
75%      43.000000  1157.000000         14.000000     4.000000
1.0
max      60.000000  1499.000000         29.000000     5.000000
1.0

       EmployeeNumber  EnvironmentSatisfaction   HourlyRate
JobInvolvement  \
count     1470.000000              1470.000000  1470.000000
1470.000000
mean      1024.865306                 2.721769    65.891156
2.729932
std        602.024335                 1.093082    20.329428
0.711561
min          1.000000                 1.000000    30.000000
1.000000
25%        491.250000                 2.000000    48.000000
2.000000
50%       1020.500000                 3.000000    66.000000
3.000000
75%       1555.750000                 4.000000    83.750000
3.000000
max       2068.000000                 4.000000   100.000000
4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours  \
count  1470.000000  ...               1470.000000         1470.0
mean      2.063946  ...                  2.712245           80.0
std       1.106940  ...                  1.081209            0.0
min       1.000000  ...                  1.000000           80.0
25%       1.000000  ...                  2.000000           80.0
50%       2.000000  ...                  3.000000           80.0
75%       3.000000  ...                  4.000000           80.0
max       5.000000  ...                  4.000000           80.0

       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1470.000000        1470.000000            1470.000000
mean           0.793878          11.279592               2.799320
std            0.852077           7.780782               1.289271
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000
```

```
       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000   
mean          2.761224        7.008163            4.229252   
std           0.706476        6.126525            3.623137   
min           1.000000        0.000000            0.000000   
25%           2.000000        3.000000            2.000000   
50%           3.000000        5.000000            3.000000   
75%           3.000000        9.000000            7.000000   
max           4.000000       40.000000           18.000000   

       YearsSinceLastPromotion  YearsWithCurrManager  
count              1470.000000           1470.000000  
mean                  2.187755              4.123129  
std                   3.222430              3.568136  
min                   0.000000              0.000000  
25%                   0.000000              2.000000  
50%                   1.000000              3.000000  
75%                   3.000000              7.000000  
max                  15.000000             17.000000  

[8 rows x 26 columns]

Missing values by column:
Age                            0
Attrition                      0
BusinessTravel                 0
DailyRate                      0
Department                     0
DistanceFromHome               0
Education                      0
EducationField                 0
EmployeeCount                  0
EmployeeNumber                 0
EnvironmentSatisfaction        0
Gender                         0
HourlyRate                     0
JobInvolvement                 0
JobLevel                       0
JobRole                        0
JobSatisfaction                0
MaritalStatus                  0
MonthlyIncome                  0
MonthlyRate                    0
NumCompaniesWorked             0
Over18                         0
OverTime                       0
PercentSalaryHike              0
PerformanceRating              0
RelationshipSatisfaction       0
```

```
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

Removing Duplicates

If there are any duplicate rows, we remove them to ensure data quality.

```
df.drop_duplicates(inplace=True)
print("Shape after removing duplicates:", df.shape)
```

```
Shape after removing duplicates: (1470, 36)
```

```
# List of columns you want to treat as categories
cat_cols = ["Attrition", "BusinessTravel", "Department",
            "EducationField", "Gender", "MaritalStatus",
            "Over18", "OverTime", "JobRole"]

# Convert each to 'category' dtype
for col in cat_cols:
    df[col] = df[col].astype("category")

# Verify the new dtypes
print(df.dtypes)
```

```
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
OverTime                    category
PercentSalaryHike              int64
PerformanceRating              int64
RelationshipSatisfaction       int64
StandardHours                  int64
StockOptionLevel               int64
TotalWorkingYears              int64
TrainingTimesLastYear          int64
WorkLifeBalance                int64
YearsAtCompany                 int64
YearsInCurrentRole             int64
YearsSinceLastPromotion        int64
YearsWithCurrManager           int64
dtype: object
```

Feature Engineering

We create another category `TenureBucket`, by categorizing employees based on their years at the company.

```
# Define bins and labels for tenure buckets
bins = [0, 3, 6, 10, 20, np.inf]
labels = ["<3", "3-6", "6-10", "10-20", "20+"]
df["TenureBucket"] = pd.cut(df["YearsAtCompany"], bins=bins,
labels=labels)
df["TenureBucket"] = df["TenureBucket"].astype('category')

# Display the value counts for the new feature
print("\nValue counts for TenureBucket:")
print(df["TenureBucket"].value_counts())
```

```
Value counts for TenureBucket:
TenureBucket
<3       426
3-6      382
6-10     372
10-20    180
20+       66
Name: count, dtype: int64
```

Save Data

Finally we save the data to be used in part 2 of our EDA

```python
df.to_csv("hr_data_cleaned.csv", index=False)
print("Cleaned data saved to 'hr_data_cleaned.csv'.")
```

```
Cleaned data saved to 'hr_data_cleaned.csv'.
```