In [1]:

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

In [2]:

```python
# Load the hr_dataset cleaned
df= pd.read_csv('hr_cleaned_dataset.csv')
df.head()
```

Out[2]:

	EmployeeID	Age	Department	SatisfactionScore	LastEvaluationScore	NumProjects	AvgM
0	896999	41	finance	0.42	0.08	5	
1	331148	41	hr	0.91	0.73	5	
2	559437	36	operations	0.93	0.82	7	
3	883201	41	finance	0.03	0.53	7	
4	562242	41	finance	0.66	0.72	3	

FEATURE ENGINEERING

1. Hours per Project to measure work intensity.

- Monthly Snapshot – Both AvgMonthlyHours and NumProjects represent monthly values, providing a consistent time frame for workload analysis.
- Workload per Project – Hours_Per_Project = AvgMonthlyHours / NumProjects measures how much time an employee spends on each project per month. High values indicate potential overwork or burnout risk.

In [3]:

```python
df['HoursPerProject']= df['AvgMonthlyHours']/df['NumProjects']
```

In [4]:

```python
# Replace any infinite values (e.g., NumProjects = 0) with median
median_hours_per_project = df['HoursPerProject'].median()
df['HoursPerProject'] = df['HoursPerProject'].replace([np.inf, -np.inf, np.nan], medi
```

2. Performance Ratio ~ Evaluation / Satisfaction.

- Calculates Performance_Ratio to flag high-performing but low-satisfaction employees, replacing invalid values with the median for clean analysis.

In [5]:

```python
df['PerformanceRatio'] = df['LastEvaluationScore']/df['SatisfactionScore']
```

In [6]:

```
 #Replace infinite or undefined values with median
median_performance_ratio = df['PerformanceRatio'] .median()
df['PerformanceRatio']  = df['PerformanceRatio'] .replace([np.inf, -np.inf, np.nan],
```

3. Tenure / Experience Feature- Categorize employees based on YearsAtCompany.

- Creates TenureCategory to group employees by experience level, enabling HR to tailor retention, promotions, and training strategies.

In [7]:

```
df['TenureCategory'] = pd.cut(
    df['YearsAtCompany'],
    bins=[0, 2, 5, 10],
    labels=['New', 'Mid', 'Experienced'],
    right=True,  # includes the upper bound of each bin
    include_lowest=True
)
```

4. High-Risk Employee Flag-employees with high workload AND low satisfaction as "at risk".

- Flagging employees who have high workload but low satisfaction as High_Risk_Employee helps identify employees most likely at risk of leaving, providing actionable HR insights.

In [8]:

```
df['High_Risk_Employee'] = np.where(
    (df['HoursPerProject'] > df['HoursPerProject'].median()) &
    (df['SatisfactionScore'] < df['SatisfactionScore'].median()),
    1, 0
)
```

5.Attrition Insights by Department & Tenure

- Calculate average attrition by Department and TenureCategory to reveal patterns in employee turnover.

- Identify departments and tenure groups with higher attrition to guide HR retention strategies.

In [9]:

```
# Calculate average attrition by Department
dept_attrition = df.groupby('Department')['Attrition'].mean().sort_values(ascending=F

# Calculate average attrition by TenureCategory
tenure_attrition = df.groupby('TenureCategory',observed=True)['Attrition'].mean().sor
```

In [16]:

```
dept_attrition*100
```

```
Out[16]: Department
         hr            33.347090
         operations    33.091398
         it            32.917014
         unknown       32.889237
         finance       32.772765
         sales         32.480211
         Name: Attrition, dtype: float64
```

```
In [11]: tenure_attrition
```

```
Out[11]: TenureCategory
         Mid             0.336042
         Experienced     0.329341
         New             0.320611
         Name: Attrition, dtype: float64
```

```
In [12]: # New Features
         print(df[['EmployeeID','AvgMonthlyHours','NumProjects','HoursPerProject',
                   'SatisfactionScore','LastEvaluationScore','PerformanceRatio',
                   'YearsAtCompany','TenureCategory','High_Risk_Employee','Attrition']].head(1
```

```
   EmployeeID  AvgMonthlyHours  NumProjects  HoursPerProject  \
0      896999              200            5        40.000000   
1      331148              180            5        36.000000   
2      559437              200            7        28.571429   
3      883201              297            7        42.428571   
4      562242              186            3        62.000000   
5      538510              200            4        50.000000   
6      585585              200            5        40.000000   
7      689574              227            5        45.400000   
8      394433              205            3        68.333333   
9      314638              227            5        45.400000   

   SatisfactionScore  LastEvaluationScore  PerformanceRatio  YearsAtCompany  \
0               0.42                 0.08          0.190476               6   
1               0.91                 0.73          0.802198               7   
2               0.93                 0.82          0.881720               4   
3               0.03                 0.53         17.666667               6   
4               0.66                 0.72          1.090909               5   
5               0.85                 0.76          0.894118               6   
6               0.65                 0.39          0.600000               5   
7               0.89                 0.80          0.898876               6   
8               0.33                 0.92          2.787879               7   
9               0.57                 0.42          0.736842               6   

  TenureCategory  High_Risk_Employee  Attrition  
0    Experienced                   0          1  
1    Experienced                   0          1  
2            Mid                   0          1  
3    Experienced                   1          1  
4            Mid                   0          0  
5    Experienced                   0          1  
6            Mid                   0          0  
7    Experienced                   0          1  
8    Experienced                   1          0  
9    Experienced                   0          1  
```

In [13]:

```
#overview
df.head()
```

Out[13]:

	EmployeeID	Age	Department	SatisfactionScore	LastEvaluationScore	NumProjects	AvgM
0	896999	41	finance	0.42	0.08	5	
1	331148	41	hr	0.91	0.73	5	
2	559437	36	operations	0.93	0.82	7	
3	883201	41	finance	0.03	0.53	7	
4	562242	41	finance	0.66	0.72	3	

In [15]:

```
# Save Feature-Enhanced Dataset
# ------------------------------
df.to_csv("hr_features_dataset.csv", index=False)
print("\nFeature-engineered HR dataset saved.")
```

```
Feature-engineered HR dataset saved.
```