Employee Attrition Analysis code

1. Import necessary libraries.

In [4]:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from scipy.stats import chi2_contingency
import numpy as np
from scipy.stats import ttest_ind
```

2. Perform logistic regression and save to a .csv to upload into PowerBI.

In [8]:

```
df = pd.read_csv(r"C:\Users\etcok\OneDrive\Documents\Springboard\datasets\Capstone 2\hr_analysis.csv")

df = df[['satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours',
         'time_spend_company', 'Work_accident', 'promotion_last_5years', 'Department',
         'salary', 'Attrition']].dropna()

le_dep = LabelEncoder()
le_salary = LabelEncoder()
le_attr = LabelEncoder()

df['Department_encoded'] = le_dep.fit_transform(df['Department'])
df['salary_encoded'] = le_salary.fit_transform(df['salary'])
df['Attrition_encoded'] = le_attr.fit_transform(df['Attrition'])

X = df[['satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours',
        'time_spend_company', 'Work_accident', 'promotion_last_5years',
        'Department_encoded', 'salary_encoded']]
y = df['Attrition_encoded']

model = LogisticRegression(class_weight='balanced', max_iter=1000)
model.fit(X, y)

df['Prediction'] = model.predict(X)

df.to_csv(r"C:\Users\etcok\OneDrive\Documents\Springboard\datasets\Capstone 2\hr_predictions-full.csv", index=False)
```

3. Perform Chi-square test and Cramer's V for categorical variables.

```
In [9]: df = pd.read_csv(r"C:\Users\etcok\OneDrive\Documents\Springboard\datasets\Capstone 2\hr_analysis.csv")

        categorical_vars = ['Work_accident','Department','salary', 'promotion_last_5years']

        # Significance level
        alpha = 0.05

        for var in categorical_vars:
            print(f"\nTesting relationship between Attrition and {var}:")

            table = pd.crosstab(df['Attrition'], df[var])

            chi2, p, dof, expected = chi2_contingency(table)

            n = table.sum().sum()
            phi2 = chi2 / n
            r, k = table.shape
            cramers_v = np.sqrt(phi2 / min(k - 1, r - 1))

            print("Chi-square statistic:", round(chi2, 2))
            print("Degrees of freedom:", dof)
            print(f"p-value: {p:.10f}")
            print(f"Cramér's V: {cramers_v:.3f}")

            if p < alpha:
                print(f"→ Significant: Attrition is likely related to {var}.")
            else:
                print(f"→ Not significant: No evidence of a relationship with {var}.")
```

```
Testing relationship between Attrition and Work_accident:
Chi-square statistic: 357.56
Degrees of freedom: 1
p-value: 0.0000000000
Cramér's V: 0.154
→ Significant: Attrition is likely related to Work_accident.

Testing relationship between Attrition and Department:
Chi-square statistic: 86.83
Degrees of freedom: 9
p-value: 0.0000000000
Cramér's V: 0.076
→ Significant: Attrition is likely related to Department.

Testing relationship between Attrition and salary:
Chi-square statistic: 381.23
Degrees of freedom: 2
p-value: 0.0000000000
Cramér's V: 0.159
→ Significant: Attrition is likely related to salary.

Testing relationship between Attrition and promotion_last_5years:
Chi-square statistic: 56.26
Degrees of freedom: 1
p-value: 0.0000000000
Cramér's V: 0.061
→ Significant: Attrition is likely related to promotion_last_5years.
```

4. Perform t-test and Cohen's d for continuous variables.

In [10]:

```python
df = pd.read_csv(r"C:\Users\etcok\OneDrive\Documents\Springboard\datasets\Capstone 2\hr_analysis.csv")

def cohen_d(x, y):
    nx = len(x)
    ny = len(y)
    dof = nx + ny - 2
    pooled_std = np.sqrt(((nx - 1)*np.var(x, ddof=1) + (ny - 1)*np.var(y, ddof=1)) / dof)
    return (np.mean(x) - np.mean(y)) / pooled_std

continuous_vars = ['satisfaction_level', 'last_evaluation', 'number_project',
                   'average_montly_hours', 'time_spend_company']

for var in continuous_vars:
    yes_group = df[df['Attrition'] == 1][var].dropna()
    no_group = df[df['Attrition'] == 0][var].dropna()

    t_stat, p_value = ttest_ind(yes_group, no_group, equal_var=False)  # Welch's t-test
    print(f"{var}:\n  t = {t_stat:.3f}, p = {p_value:.4f}\n")
    effect_size = cohen_d(yes_group, no_group)
    print(f"Cohen's d for {var}: {effect_size:.3f}")
```

```
satisfaction_level:
  t = -46.636, p = 0.0000

Cohen's d for satisfaction_level: -0.989
last_evaluation:
  t = 0.725, p = 0.4683

Cohen's d for last_evaluation: 0.015
number_project:
  t = 2.166, p = 0.0303

Cohen's d for number_project: 0.056
average_montly_hours:
  t = 7.532, p = 0.0000

Cohen's d for average_montly_hours: 0.168
time_spend_company:
  t = 22.631, p = 0.0000

Cohen's d for time_spend_company: 0.344
```

In [ ]:

The Mann Whitney was run for the continuous variable to provide additional information.

In [15]:

```
import pandas as pd
from scipy.stats import mannwhitneyu

# Load the dataset
df = pd.read_csv(r"C:\Users\etcok\OneDrive\Documents\Springboard\datasets\Capstone 2\hr_analysis.csv")

# List of continuous variables
continuous_vars = ['satisfaction_level', 'last_evaluation', 'number_project',
                   'average_montly_hours', 'time_spend_company']

# Ensure Attrition is binary (0 and 1)
print(df['Attrition'].value_counts())  # Optional sanity check

# Loop through each continuous variable and run Mann-Whitney U test
for var in continuous_vars:
    group0 = pd.to_numeric(df[df['Attrition'] == 0][var], errors='coerce').dropna()
    group1 = pd.to_numeric(df[df['Attrition'] == 1][var], errors='coerce').dropna()
    stat, p = mannwhitneyu(group0, group1, alternative='two-sided')
    print(f"{var}: U-statistic={stat:.4f}, p-value={p:.4f}")
    if p < 0.05:
        print(" -> Significant difference between Attrition groups\n")
    else:
        print(" -> No significant difference between Attrition groups\n")
```

```
Attrition
0    11428
1     3571
Name: count, dtype: int64
satisfaction_level: U-statistic=30522915.0000, p-value=0.0000
 -> Significant difference between Attrition groups

last_evaluation: U-statistic=20472187.0000, p-value=0.7650
 -> No significant difference between Attrition groups

number_project: U-statistic=20930147.0000, p-value=0.0167
 -> Significant difference between Attrition groups

average_montly_hours: U-statistic=19119787.5000, p-value=0.0000
 -> Significant difference between Attrition groups

time_spend_company: U-statistic=13331224.0000, p-value=0.0000
 -> Significant difference between Attrition groups
```

In [ ]:

```
#code to export this as pdf
```