```
# ==========================================
# STEP 1: Import Required Libraries
# ==========================================

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

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# ==========================================
# STEP 2: Load Dataset
# ==========================================

# Change file path if needed
df = pd.read_excel(r"C:\Users\grand\Downloads\hr_dataset_1000_rows.xlsx")

# Display first 5 rows
print("First 5 rows:")
print(df.head())

# Display dataset info
print("\nDataset Info:")
print(df.info())
```

```
# ==========================================
# STEP 3: Data Cleaning
# ==========================================

# Check missing values
print("\nMissing Values:")
print(df.isnull().sum())

# Drop employee_id, first_name, last_name (not useful for prediction)
df = df.drop(['employee_id','first_name','last_name'], axis=1)

# ==========================================
# STEP 4: Convert Categorical Data to Numeric
# ==========================================

le = LabelEncoder()

# Convert categorical columns
categorical_columns = ['gender','department','job_title',
                       'education','marital_status',
                       'attrition','location']

for col in categorical_columns:
    df[col] = le.fit_transform(df[col])

# Convert hire_date to years of service
df['hire_date'] = pd.to_datetime(df['hire_date'])
```

```
df['years_at_company'] = 2025 - df['hire_date'].dt.year

df = df.drop('hire_date', axis=1)

print("\nCleaned Data:")
print(df.head())

# ========================================
# STEP 5: Exploratory Data Analysis (EDA)
# ========================================

# Attrition count plot
plt.figure(figsize=(6,4))
sns.countplot(x='attrition', data=df)
plt.title("Attrition Count")
plt.show()

# Salary vs Attrition
plt.figure(figsize=(6,4))
sns.boxplot(x='attrition', y='salary', data=df)
plt.title("Salary vs Attrition")
plt.show()

# Experience vs Attrition
plt.figure(figsize=(6,4))
sns.boxplot(x='attrition', y='experience_years', data=df)
plt.title("Experience vs Attrition")
plt.show()
```

```
# ==========================================
# STEP 6: Feature Selection
# ==========================================

X = df.drop('attrition', axis=1)
y = df['attrition']

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

# ==========================================
# STEP 7: Train Machine Learning Model
# ==========================================

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# ==========================================
# STEP 8: Model Evaluation
# ==========================================

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
```

```
print("\nModel Accuracy:", accuracy)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print(cm)

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

# ==========================================
# STEP 9: Feature Importance
# ==========================================

importance = pd.DataFrame({
    'Feature': X.columns,
    'Importance': model.coef_[0]
})

importance = importance.sort_values(by='Importance', ascending=False)

print("\nFeature Importance:")
print(importance)

# Plot Feature Importance
plt.figure(figsize=(8,5))
sns.barplot(x='Importance', y='Feature', data=importance)
```

```
plt.title("Feature Importance")

plt.show()


# ==========================================

# STEP 10: Save Predictions

# ==========================================


output = X_test.copy()

output['Actual Attrition'] = y_test

output['Predicted Attrition'] = y_pred


output.to_excel("attrition_predictions.xlsx", index=False)


print("\nPrediction file saved as attrition_predictions.xlsx")


# ==========================================

# END OF PROJECT

# ==========================================
```

OUTPUT:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32

Enter "help" below or click "Help" above for more information.


= RESTART: C:\Users\grand\OneDrive\Documents\DATA ANALYSIS TASKS AND PROJECTS\data analyst project 1.py

First 5 rows:

  employee_id first_name last_name  ... performance_rating  attrition   location

0      E0001     Kavya     Patel  ...                 4       Yes  Bangalore

1      E0002     Kiran     Reddy  ...                 3       Yes  Bangalore
```

```
2     E0003    Arjun   Sharma ...            1       No  Hyderabad
3     E0004   Suresh   Sharma ...            4       No  Hyderabad
4     E0005     Ravi     Iyer ...            2       No  Bangalore

[5 rows x 15 columns]


Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 15 columns):
 #  Column           Non-Null Count  Dtype
---  ------          --------------  -----
 0  employee_id       1000 non-null   object
 1  first_name        1000 non-null   object
 2  last_name         1000 non-null   object
 3  gender           1000 non-null   object
 4  age              1000 non-null   int64
 5  department        1000 non-null   object
 6  job_title         1000 non-null   object
 7  education         1000 non-null   object
 8  marital_status     1000 non-null   object
 9  hire_date         1000 non-null   object
 10  salary           1000 non-null   int64
 11  experience_years    1000 non-null   int64
 12  performance_rating  1000 non-null   int64
 13  attrition         1000 non-null   object
 14  location          1000 non-null   object
dtypes: int64(4), object(11)
```

memory usage: 117.3+ KB

None

Missing Values:

employee_id 0

first_name 0

last_name 0

gender 0

age 0

department 0

job_title 0

education 0

marital_status 0

hire_date 0

salary 0

experience_years 0

performance_rating 0

attrition 0

location 0

dtype: int64

Cleaned Data:

	gender	age	department	...	attrition	location	years_at_company
0	1	45	4	...	1	0	2
1	0	27	1	...	1	0	9
2	0	30	3	...	0	4	6
3	1	47	4	...	0	4	6
4	1	40	3	...	0	0	3

[5 rows x 12 columns]

Model Accuracy: 0.435

Confusion Matrix:

[[56 33]

[80 31]]

Classification Report:

	precision	recall	f1-score	support
0	0.41	0.63	0.50	89
1	0.48	0.28	0.35	111
accuracy			0.43	200
macro avg	0.45	0.45	0.43	200
weighted avg	0.45	0.43	0.42	200

Feature Importance:

	Feature	Importance
10	years_at_company	7.944792e-03
5	marital_status	2.597258e-03
7	experience_years	2.492438e-03
6	salary	5.022347e-07
0	gender	-4.473819e-03
4	education	-5.363569e-03

1 age -6.045907e-03

2 department -9.560725e-03

8 performance_rating -2.291166e-02

9 location -2.908455e-02

3 job_title -3.027123e-02

Prediction file saved as attrition_predictions.xlsx