```
#%%
##Load & Basic Checks##
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("D:/Elevate Lab/Task/project phase 2 /HR_Attrition.csv")

df.info()
df.isnull().sum()


#%%
df['Actual_Attrition'].value_counts()

#%%
df['Predicted_Attrition'].value_counts()

#%%
df['Attrition_Probability'].describe()

#%%
## Create images folder ##

import os
os.makedirs("images", exist_ok=True)

#%%
## Total Attrition Count Plot##

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(6,4))
sns.countplot(x='Actual_Attrition', data=df)
plt.title("Actual Attrition Count")
plt.savefig("images/Actual Attrition Count.png",)
plt.show()

#%%
```

```
41 ## Department-wise Attrition ##
42
43 plt.figure(figsize=(8,5))
44 sns.countplot(x='Department', hue='Actual_Attrition'
   , data=df)
45 plt.title("Attrition by Department")
46 plt.xlabel("Department")
47 plt.ylabel("Employee Count")
48
49 plt.savefig("images/attrition_by_department.png",
50             dpi=300,
51             bbox_inches='tight')
52 plt.show()
53
54 #%%
55 ## Attrition Probability Distribution ##
56
57 plt.figure(figsize=(8,5))
58 plt.hist(df['Attrition_Probability'], bins=10)
59 plt.title("Attrition Probability Distribution")
60 plt.xlabel("Probability")
61 plt.ylabel("Employee Count")
62 plt.savefig("images/attrition_probability.png",
63              dpi=300,
64             bbox_inches='tight')
65 plt.show()
66
67 #%%
68 ## Correlation Heatmap ##
69
70 plt.figure(figsize=(10,8))
71 sns.heatmap(df.select_dtypes(include='number').corr
   (), cmap='coolwarm')
72 plt.title("Correlation Heatmap")
73 plt.savefig("images/correlation_heatmap.png",
74             dpi=300,
75             bbox_inches='tight')
76 plt.show()
77
78 #%%
79 ## Department-wise Attrition Rate plot ##
```

```

dept_attrition = (
    df.groupby('Department')['Actual_Attrition']
      .value_counts(normalize=True)
      .unstack()
)

dept_attrition.plot(kind='bar', stacked=True, figsize=(8,5))
plt.title("Department-wise Attrition Rate")
plt.ylabel("Proportion")
plt.xlabel("Department")
plt.legend(title="Attrition")
plt.savefig("images/Department-wise Attrition Rate.png",
            dpi=300,
            bbox_inches='tight')
plt.show()

#%%
## Attrition Probability Distribution ##

sns.boxplot(x='Actual_Attrition', y='MonthlyIncome', data=df)
plt.title("Monthly Income vs Actual_Attrition")
plt.savefig("images/monthly_income.png",
            dpi=300,
            bbox_inches='tight')
plt.show()

#%%
## Promotions vs Attrition ##

sns.countplot(x='YearsSinceLastPromotion', hue='Actual_Attrition', data=df)
plt.title("Promotion Gap vs Actual_Attrition")
plt.savefig("images/Promotion Gap vs Actual_Attrition.png",
            dpi=300,
            bbox_inches='tight')
plt.show()
```

```
116 ____________________________________________________________________
117 #%%
118 ## Data Preprocessing ##
119
120 from sklearn.preprocessing import LabelEncoder
121
122 le = LabelEncoder()
123 for col in df.select_dtypes(include='object'):
124     df[col] = le.fit_transform(df[col])
125 ____________________________________________________________________
126 #%%
127 X = df.drop('Actual_Attrition', axis=1)
128 y = df['Actual_Attrition']
129 ____________________________________________________________________
130 #%%
131 ## Build Classification Model - Logistic Regression
     ##
132
133 from sklearn.preprocessing import StandardScaler
134
135 scaler = StandardScaler()
136
137 X_train_scaled = scaler.fit_transform(X_train)
138 X_test_scaled = scaler.transform(X_test)
139
140
141 ____________________________________________________________________
142 #%%
142 model = LogisticRegression(max_iter=2000, solver='
    lbfgs')
143 model.fit(X_train_scaled, y_train)
144 ____________________________________________________________________
145 #%%
146 y_pred = model.predict(X_test_scaled)
147 ____________________________________________________________________
148 #%%
149 ## Model Evaluation ##
150
151 print("Accuracy:", accuracy_score(y_test, y_pred))
152 print(confusion_matrix(y_test, y_pred))
153 print(classification_report(y_test, y_pred))
154 ____________________________________________________________________
```

```
#%%
## Confusion Matrix Map ##

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")

plt.savefig("images/confusion_matrix.png",
            dpi=300,
            bbox_inches='tight')
plt.show()

#%%
## SHAP Value Analysis (Explain Predictions) ##

import shap

explainer = shap.Explainer(model, X_train)
shap_values = explainer(X_test)

shap.summary_plot(shap_values, X_test)
plt.savefig("images/shap_summary_plot.png",
            dpi=300,
            bbox_inches='tight')
plt.close()

#%%
attrition_prob = model.predict_proba(X_test_scaled
)[:, 1]
#%%
# Add predictions & probability to test data ##

df_test = X_test.copy()

df_test['Actual_Attrition'] = y_test.values
```

```
df_test['Predicted_Attrition'] = y_pred
df_test['Attrition_Probability'] = attrition_prob

#%%
## Convert Attrition values to Yes / No ##

df_test['Actual_Attrition'] = df_test['Actual_Attrition'].map({1: 'Yes', 0: 'No'})
df_test['Predicted_Attrition'] = df_test['Predicted_Attrition'].map({1: 'Yes', 0: 'No'})

#%%
## Export CSV for Power BI ##

df_test.to_csv("HR_Attrition_Predictions.csv", index=False)

#%%

#%% md
## 🔍 Key Insights - HR Attrition Analysis

### 1️⃣ Overall Attrition Trend
- Employee attrition is **not random**.
- Salary level, overtime, and career growth are the **strongest drivers** of resignation.
- A small group of features explains the **majority of attrition cases**.

### 2️⃣ Department & Workload
- Employees working **overtime** show a significantly higher attrition rate.
- Certain departments show **consistently higher risk** than others.

### 3️⃣ Model Interpretation
- SHAP analysis confirms that **OverTime, MonthlyIncome, and JobLevel**
   have the highest impact on attrition prediction.


```