Employee Attrition Model Performance Report

1. Overview

This report summarizes the performance of machine learning models used to predict employee attrition in a corporate HR dataset. The goal was to identify employees at risk of leaving and the key drivers behind it.

2. Models Used

- Logistic Regression
- Decision Tree Classifier

3. Data Split

- Training Set: 1176 records
- Test Set: 294 records
- Attrition Distribution (Test):
  - No: 247
  - Yes: 47

4. Model Performance

Logistic Regression

- Accuracy: 87.4%
- Confusion Matrix:

```
 [[239   8]
  [ 29  18]]
```

- Classification Report:
  - Precision (Yes): 0.69
  - Recall (Yes): 0.38
  - F1-score (Yes): 0.49

Decision Tree

- Accuracy: 76.2%
- Confusion Matrix:

[[209 38]

 [32 15]]

- Classification Report:
  - Precision (Yes): 0.28
  - Recall (Yes): 0.32
  - F1-score (Yes): 0.30

5. Model Comparison

Model Comparison Table:

Model	Accuracy	F1-score (Yes)
Logistic Regression	87.4%	0.49
Decision Tree	76.2%	0.30

Best performing model: Logistic Regression due to better overall accuracy and precision.

6. SHAP Analysis

Top Features Influencing Attrition:

- OverTime
- MonthlyIncome
- YearsAtCompany
- JobRole
- DistanceFromHome

SHAP analysis was used to provide transparency in predictions and identify the most influential features for attrition.

7. Conclusion

The Logistic Regression model is the most reliable for predicting attrition. SHAP analysis helped identify actionable drivers, aiding HR teams in implementing data-driven retention strategies.

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv("HR-Employee-Attrition.csv")
df.head()
```

Out[2]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Education
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Scie
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Scie
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	C
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Scie
4	27	No	Travel_Rarely	591	Research & Development	2	1	Me

5 rows × 35 columns

In [3]:
```
df.info()
df.describe()
df['Attrition'].value_counts()
```

```
<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
```

Out[3]:

```
Attrition
No     1233
Yes     237
Name: count, dtype: int64
```

In [4]:

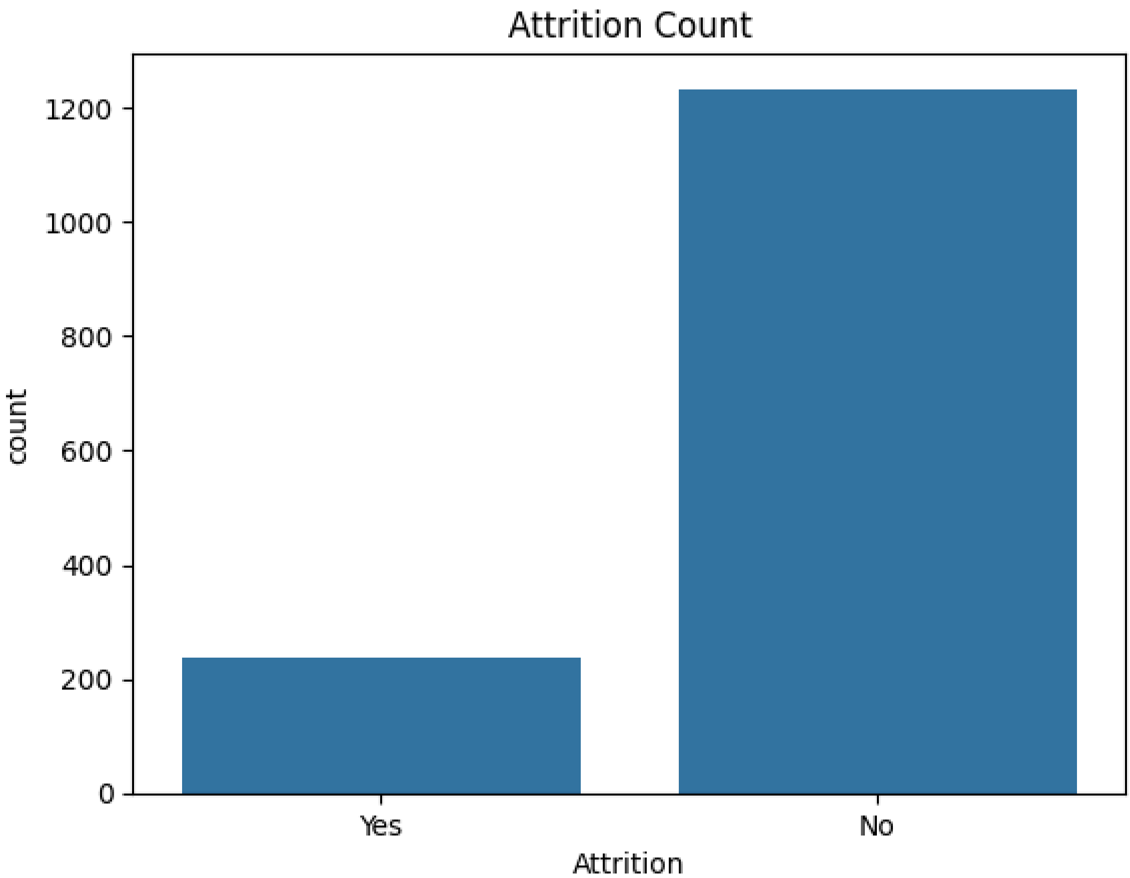
```
df.isnull().sum()
df.duplicated().sum()
```

Out[4]:

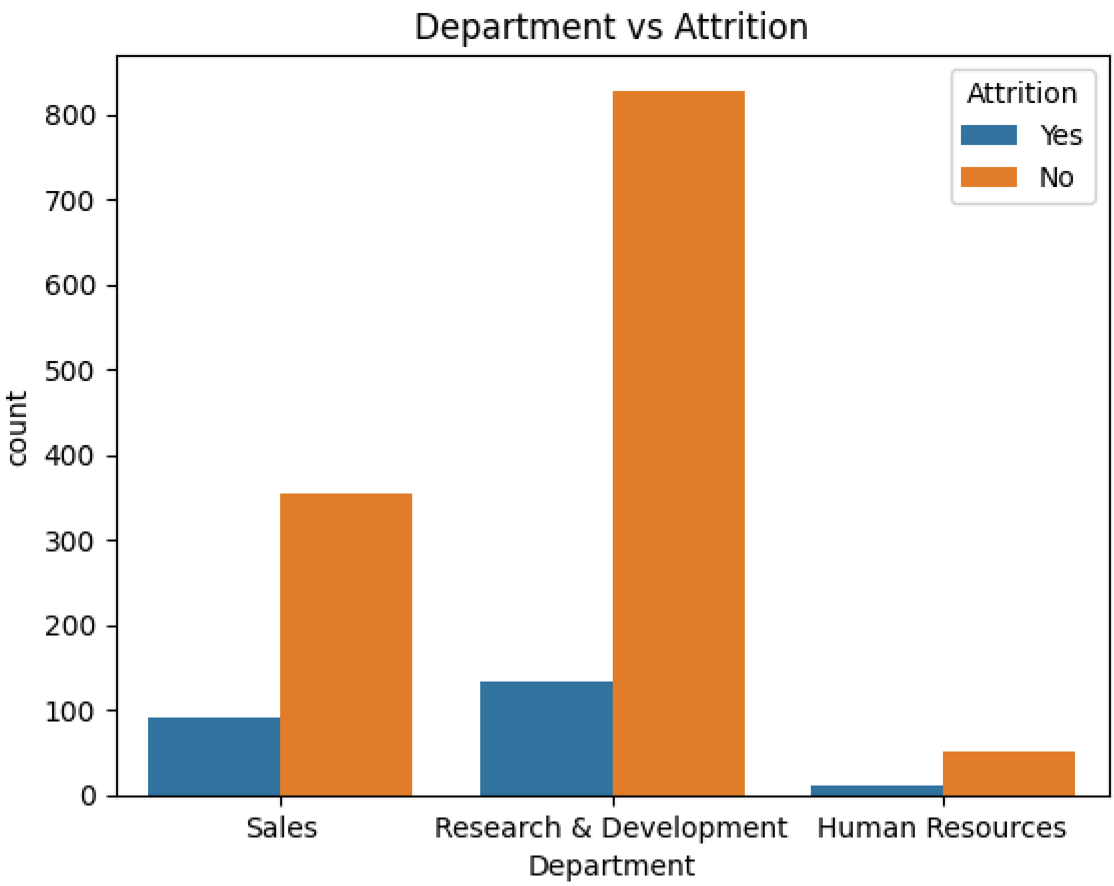
```
np.int64(0)
```

In [5]:

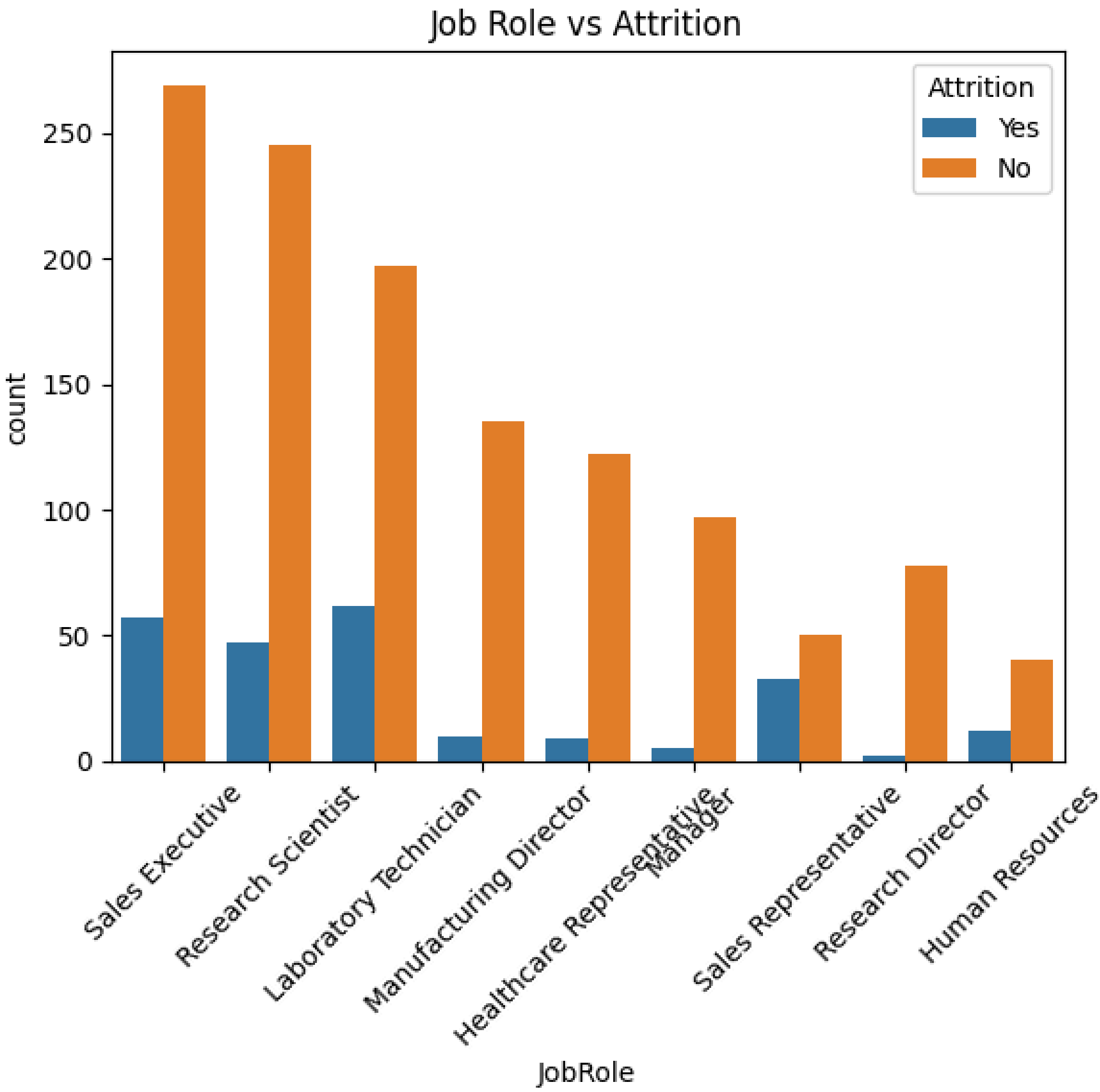
```
sns.countplot(x='Attrition', data=df)
plt.title("Attrition Count")
plt.show()
```



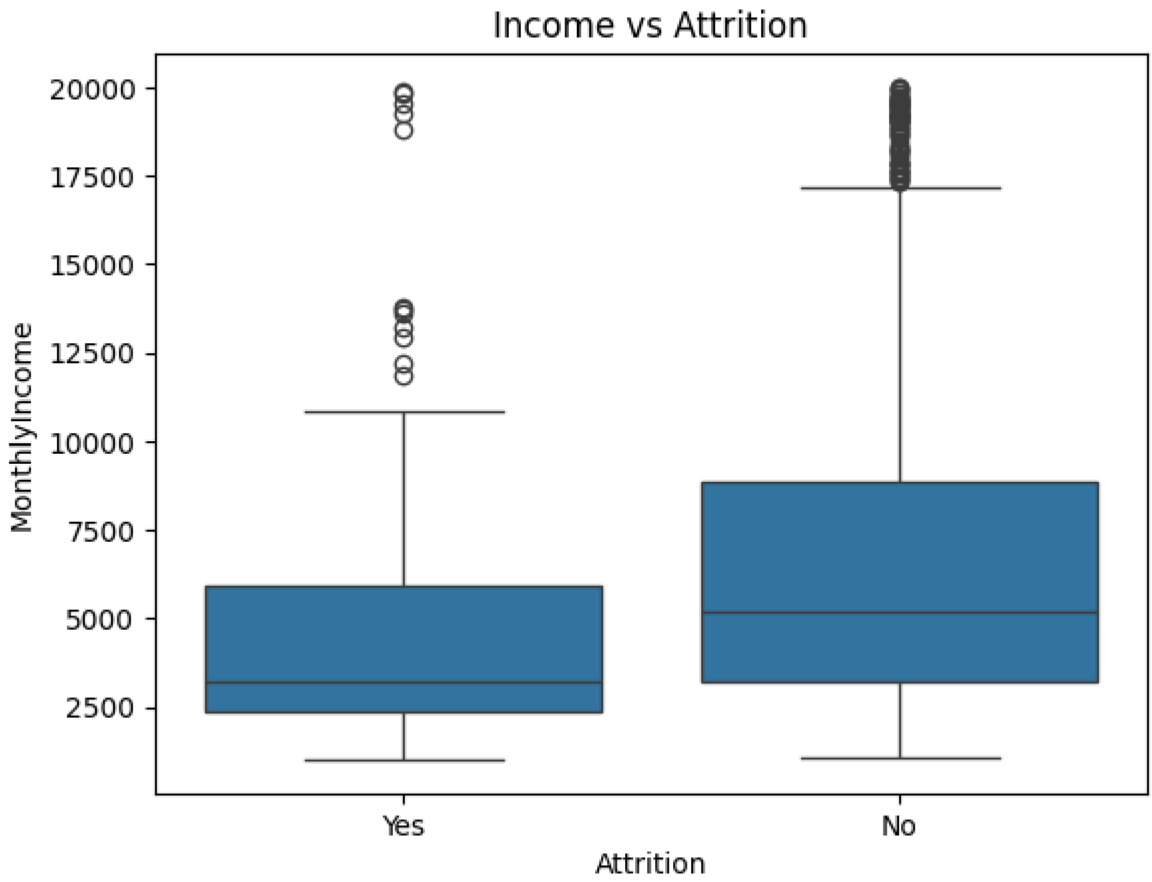
In [6]:
```
sns.countplot(x='Department', hue='Attrition', data=df)
plt.title("Department vs Attrition")
plt.show()
```



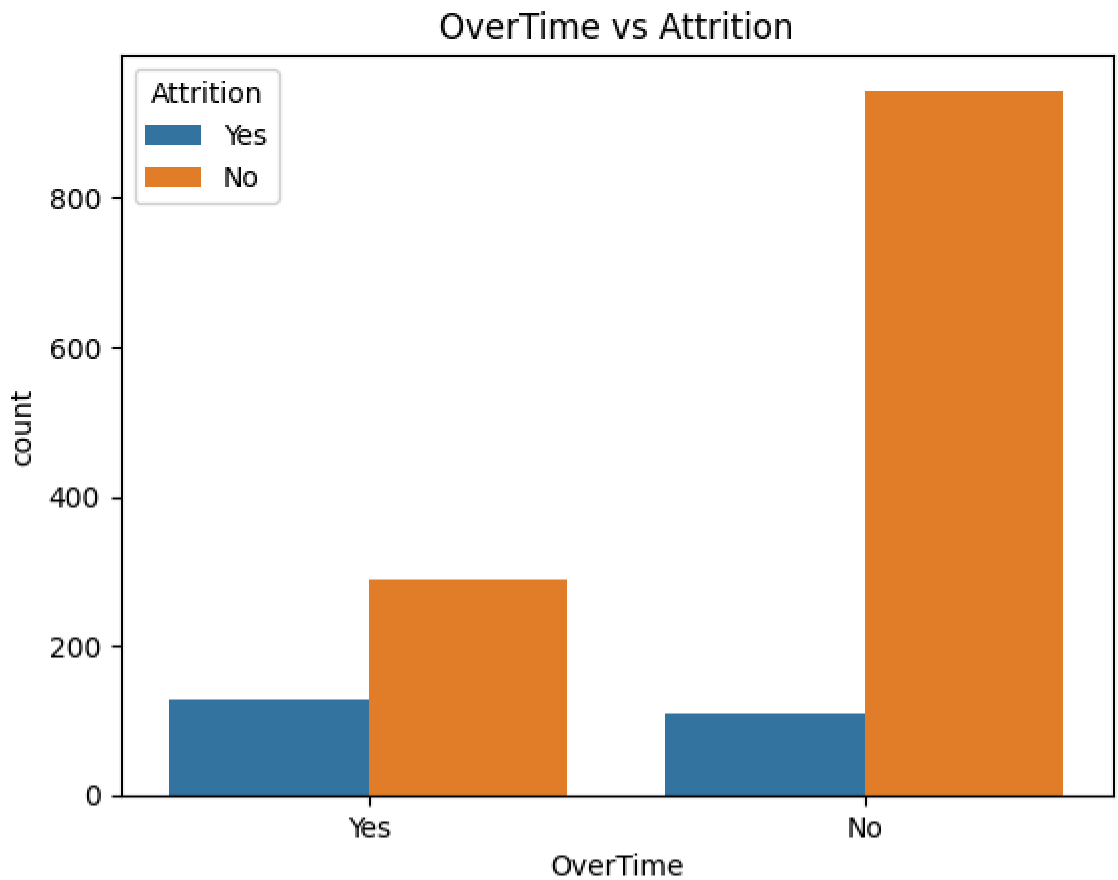
In [7]:
```
sns.countplot(x='JobRole', hue='Attrition', data=df)
plt.title("Job Role vs Attrition")
plt.xticks(rotation=45)
plt.show()
```



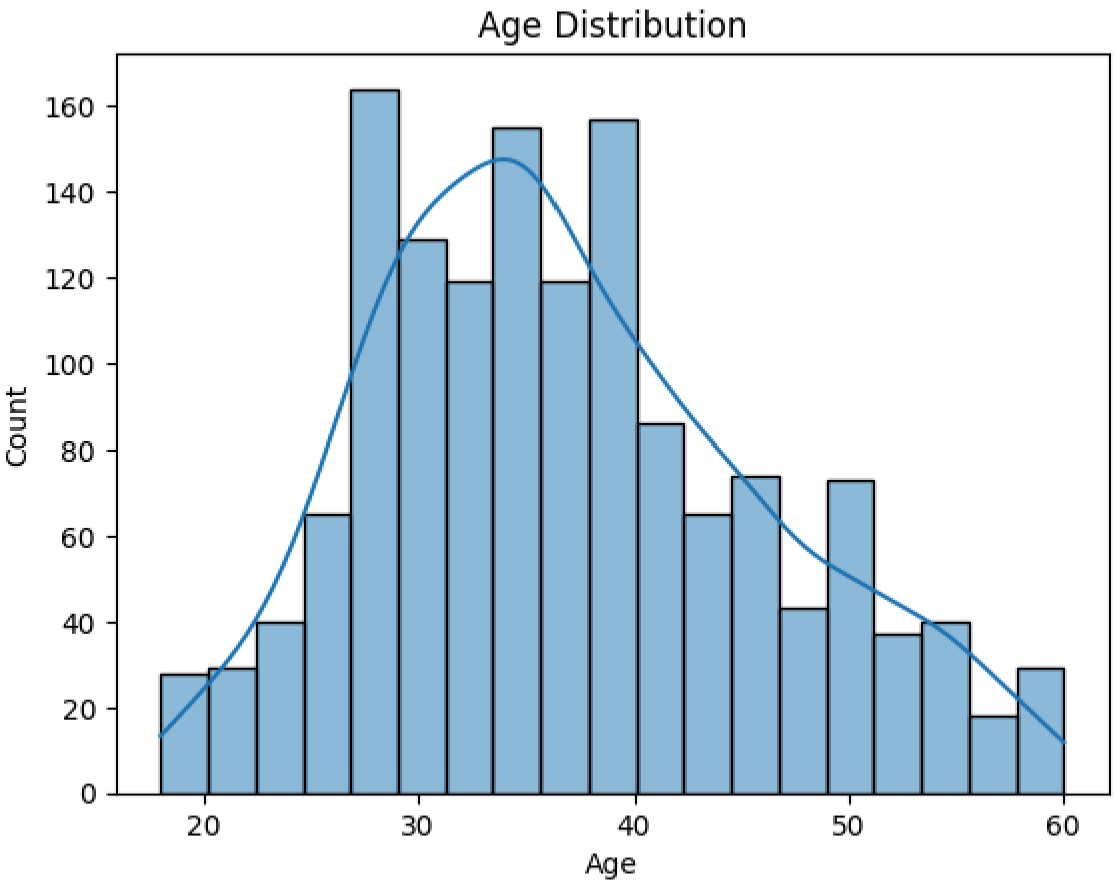
In [8]:
```
sns.boxplot(x='Attrition', y='MonthlyIncome', data=df)
plt.title("Income vs Attrition")
plt.show()
```



In [9]:
```
sns.countplot(x='OverTime', hue='Attrition', data=df)
plt.title("OverTime vs Attrition")
plt.show()
```



In [10]:
```
sns.histplot(df['Age'], kde=True)
plt.title("Age Distribution")
plt.show()
```



In [11]:
```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Attrition'] = le.fit_transform(df['Attrition'])  # No = 0, Yes = 1
df['OverTime'] = le.fit_transform(df['OverTime'])
df['Gender'] = le.fit_transform(df['Gender'])
```

In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [16]:
```
df_model = df.copy()
le = LabelEncoder()
for col in df_model.select_dtypes(include='object').columns:
    df_model[col] = le.fit_transform(df_model[col])

# Features & label
X = df_model.drop('Attrition', axis=1)
y = df_model['Attrition']

# Final split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
```

```
print("y_train distribution:\n", y_train.value_counts())
print("y_test distribution:\n", y_test.value_counts())
```

```
X_train shape: (1176, 34)
X_test shape: (294, 34)
y_train distribution:
 Attrition
0    986
1    190
Name: count, dtype: int64
y_test distribution:
 Attrition
0    247
1     47
Name: count, dtype: int64
```

In [23]:

```
from sklearn.preprocessing import StandardScaler

# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train Logistic Regression
lr_model = LogisticRegression(max_iter=2000)
lr_model.fit(X_train_scaled, y_train)

# Predict
y_pred_lr = lr_model.predict(X_test_scaled)
```

In [22]:

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

print("🔷 Logistic Regression")
print("Accuracy:", accuracy_score(y_test, y_pred_lr))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_lr))
print("Classification Report:\n", classification_report(y_test, y_pred_lr))
```

```
🔷 Logistic Regression
Accuracy: 0.8741496598639455
Confusion Matrix:
 [[239   8]
 [ 29  18]]
Classification Report:
               precision    recall  f1-score   support

           0       0.89      0.97      0.93       247
           1       0.69      0.38      0.49        47

    accuracy                           0.87       294
   macro avg       0.79      0.68      0.71       294
weighted avg       0.86      0.87      0.86       294
```

In [24]:

```
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

y_pred_dt = dt_model.predict(X_test)
print("🔶 Decision Tree")
print("Accuracy:", accuracy_score(y_test, y_pred_dt))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_dt))
print("Classification Report:\n", classification_report(y_test, y_pred_dt))
```

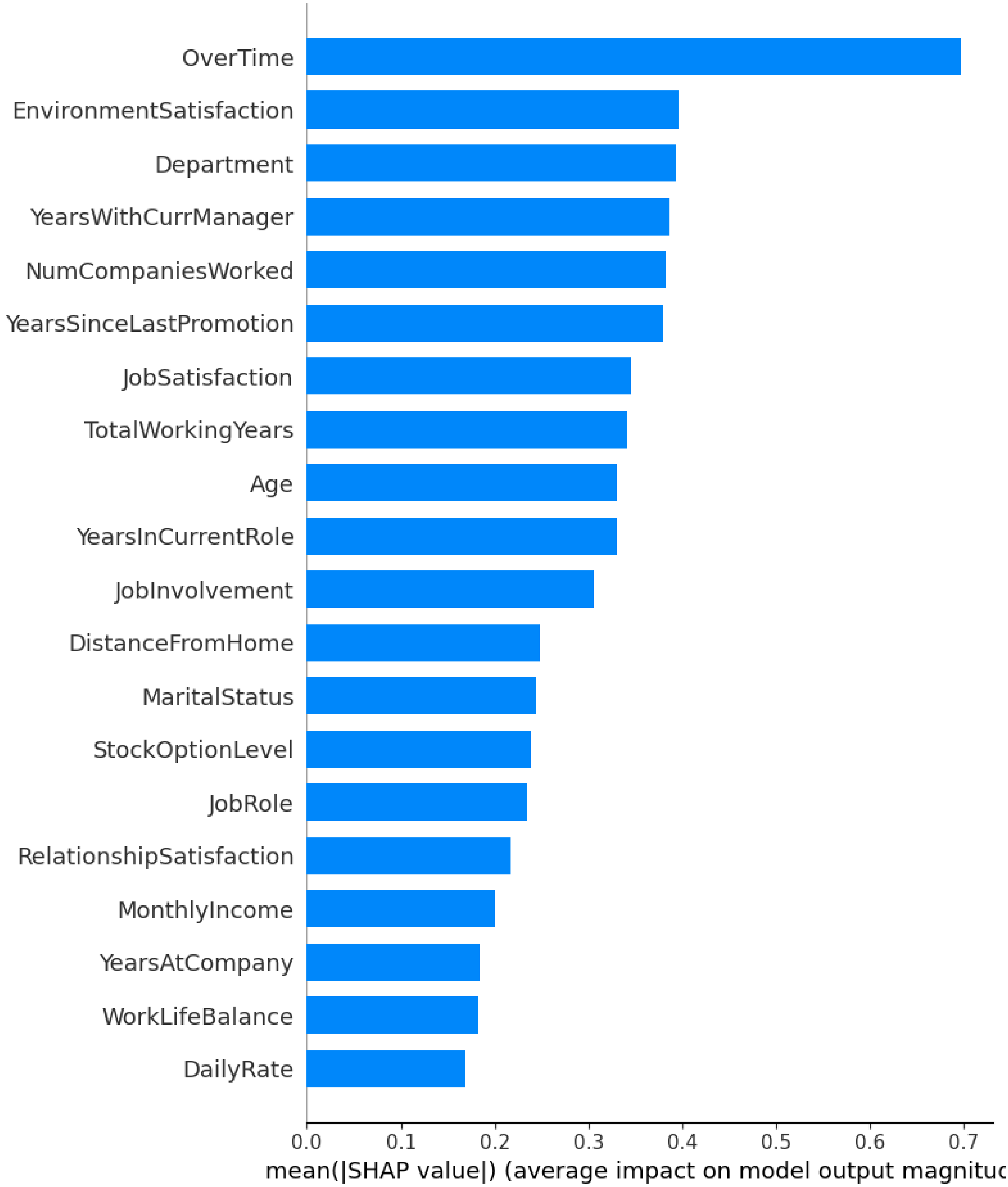
```
🔶 Decision Tree
Accuracy: 0.7619047619047619
Confusion Matrix:
 [[209  38]
 [ 32  15]]
Classification Report:
               precision    recall  f1-score   support

           0       0.87      0.85      0.86       247
           1       0.28      0.32      0.30        47

    accuracy                           0.76       294
   macro avg       0.58      0.58      0.58       294
weighted avg       0.77      0.76      0.77       294
```

In [25]:
```
!pip install shap
import shap
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: shap in c:\users\afra\appdata\local\programs\python\python313\l
ib\site-packages (0.48.0)
Requirement already satisfied: numpy in c:\users\afra\appdata\local\programs\python\python313
\lib\site-packages (from shap) (2.2.6)
Requirement already satisfied: scipy in c:\users\afra\appdata\local\programs\python\python313
\lib\site-packages (from shap) (1.15.3)
Requirement already satisfied: scikit-learn in c:\users\afra\appdata\local\programs\python\pyt
hon313\lib\site-packages (from shap) (1.6.1)
Requirement already satisfied: pandas in c:\users\afra\appdata\local\programs\python\python313
\lib\site-packages (from shap) (2.2.3)
Requirement already satisfied: tqdm>=4.27.0 in c:\users\afra\appdata\local\programs\python\pyt
hon313\lib\site-packages (from shap) (4.67.1)
Requirement already satisfied: packaging>20.9 in c:\users\afra\appdata\local\programs\python\p
ython313\lib\site-packages (from shap) (24.2)
Requirement already satisfied: slicer==0.0.8 in c:\users\afra\appdata\local\programs\python\py
thon313\lib\site-packages (from shap) (0.0.8)
Requirement already satisfied: numba>=0.54 in c:\users\afra\appdata\local\programs\python\pyth
on313\lib\site-packages (from shap) (0.61.2)
Requirement already satisfied: cloudpickle in c:\users\afra\appdata\local\programs\python\pyth
on313\lib\site-packages (from shap) (3.1.1)
Requirement already satisfied: typing-extensions in c:\users\afra\appdata\local\programs\pytho
n\python313\lib\site-packages (from shap) (4.13.2)
Requirement already satisfied: llvmlite<0.45,>=0.44.0dev0 in c:\users\afra\appdata\local\progr
ams\python\python313\lib\site-packages (from numba>=0.54->shap) (0.44.0)
Requirement already satisfied: colorama in c:\users\afra\appdata\local\programs\python\python3
13\lib\site-packages (from tqdm>=4.27.0->shap) (0.4.6)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\afra\appdata\local\programs
\python\python313\lib\site-packages (from pandas->shap) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\afra\appdata\local\programs\python\pyt
hon313\lib\site-packages (from pandas->shap) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\afra\appdata\local\programs\python\p
ython313\lib\site-packages (from pandas->shap) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\afra\appdata\local\programs\python\python3
13\lib\site-packages (from python-dateutil>=2.8.2->pandas->shap) (1.17.0)
Requirement already satisfied: joblib>=1.2.0 in c:\users\afra\appdata\local\programs\python\py
thon313\lib\site-packages (from scikit-learn->shap) (1.5.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\afra\appdata\local\programs\py
thon\python313\lib\site-packages (from scikit-learn->shap) (3.6.0)
```

In [26]:
```
explainer = shap.Explainer(lr_model, X_train_scaled)
shap_values = explainer(X_test_scaled)
```

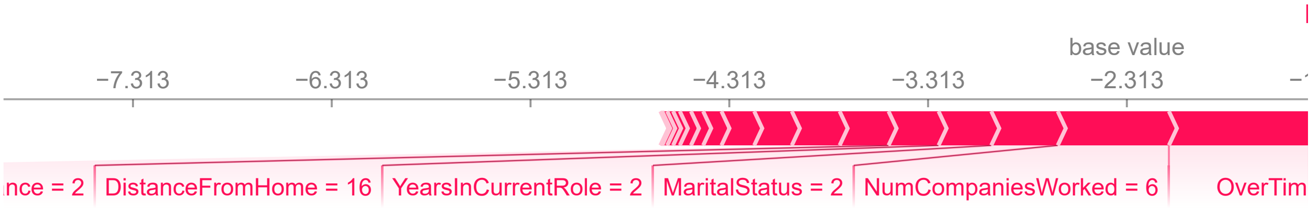
In [28]:
```
shap.summary_plot(shap_values, features=X_test_scaled, feature_names=X_test.columns, plot_typ
```



In [30]:

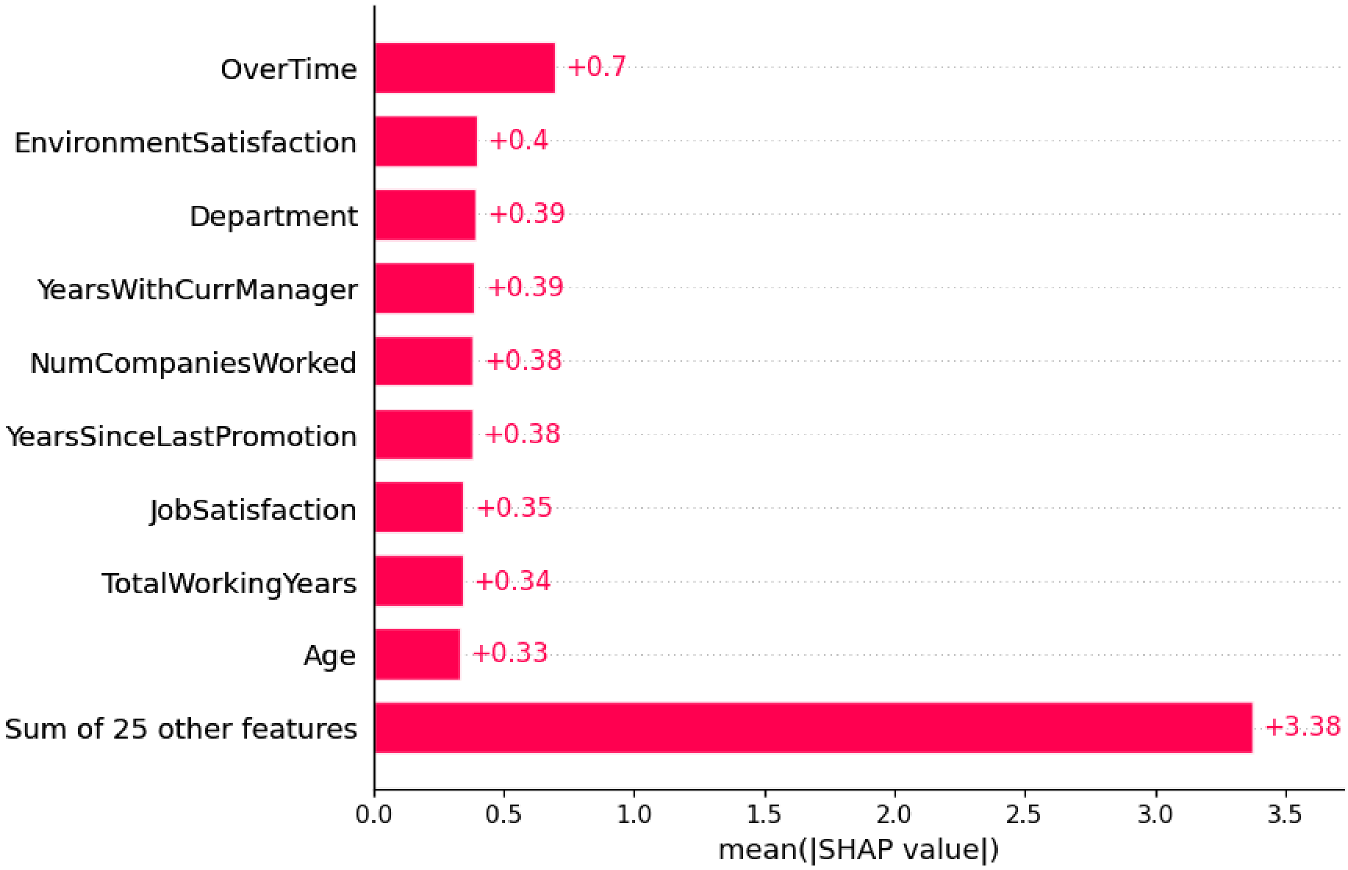
```
shap.initjs()
index = 5  # you can change this
shap.force_plot(
    explainer.expected_value,
    shap_values[index].values,
    X_test.iloc[index]
)
```

Out[30]:



In [33]:

```
# Rebuild SHAP Explanation with feature names attached
shap_values = shap.Explanation(
    values=shap_values.values,
    base_values=shap_values.base_values,
    data=X_test_scaled,
    feature_names=X_test.columns
)
shap.plots.bar(shap_values)
```



In [ ]:

ATTRITION DASHBOARD

1470
Total Employees

237
Total Attrition

16.1%
Attrition Rate

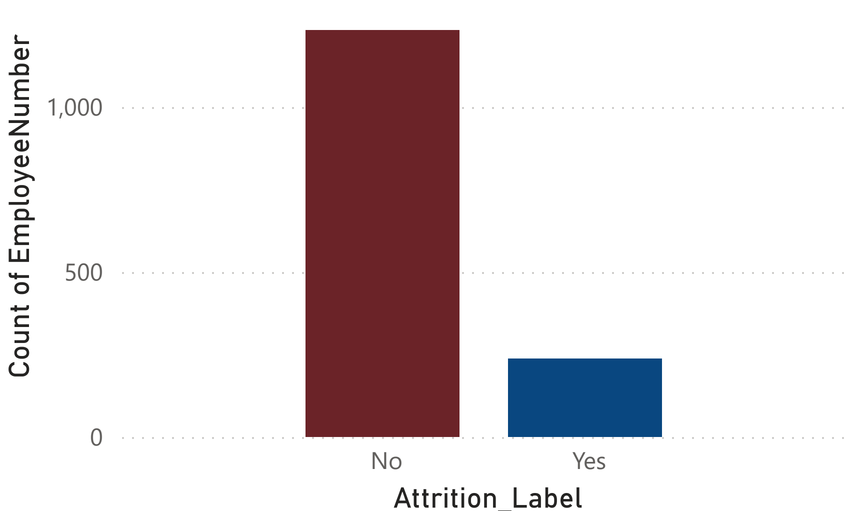
Gender
- [ ] Female
- [ ] Male

Department
- [ ] Human Resources
- [ ] Research & Development
- [ ] Sales

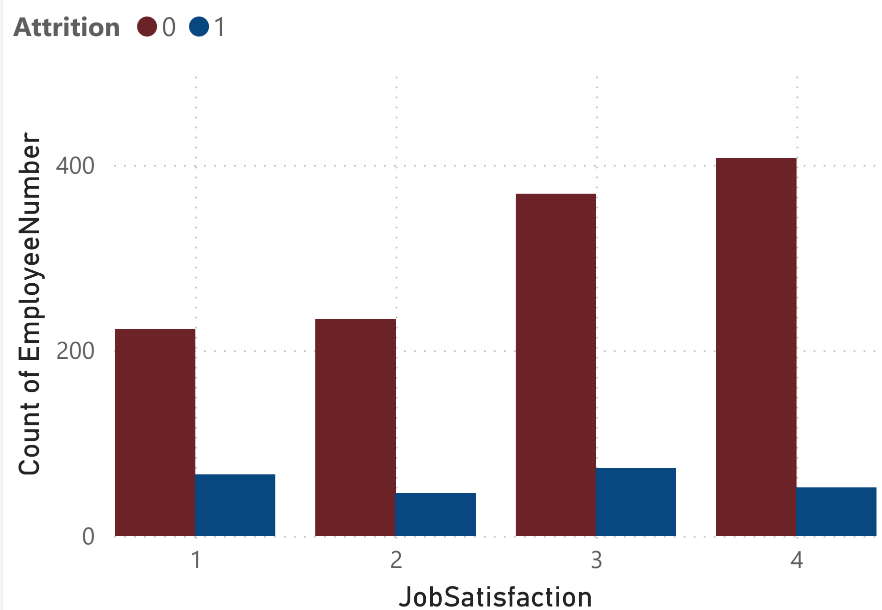
EducationField
- [ ] Human Resources
- [ ] Life Sciences
- [ ] Marketing
- [ ] Medical

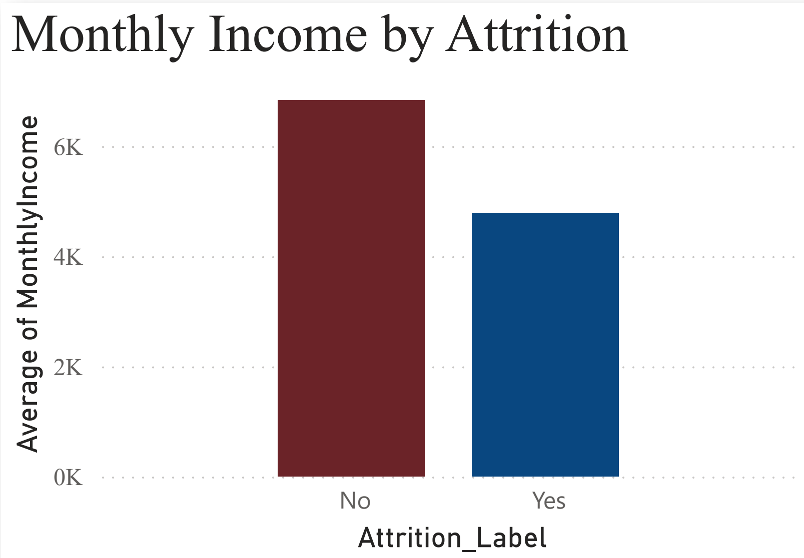
OverTime
- [ ] No
- [ ] Yes

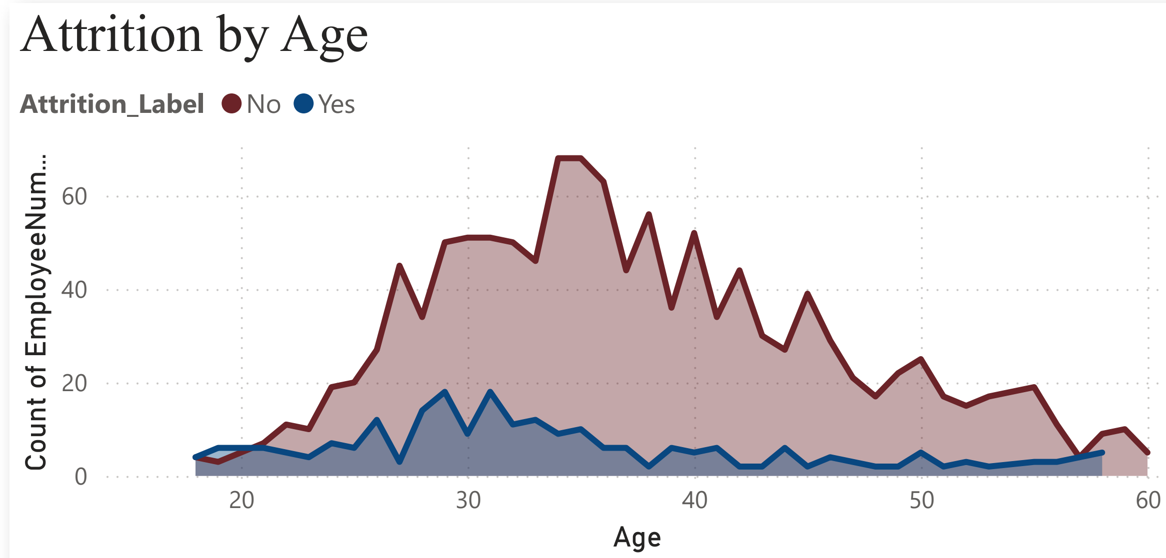
Overall Employee Attrition Distribution

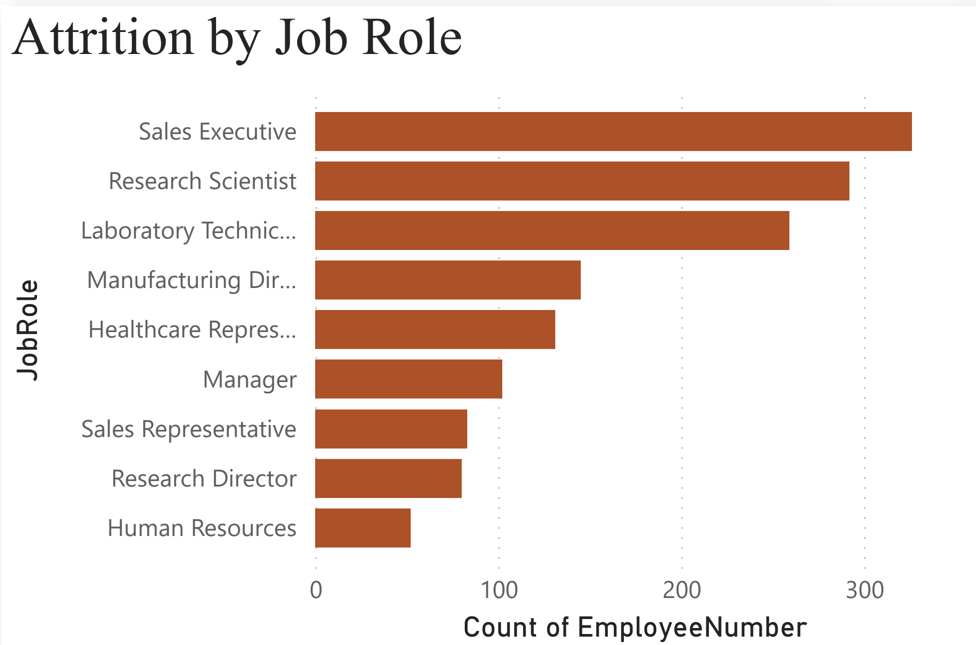


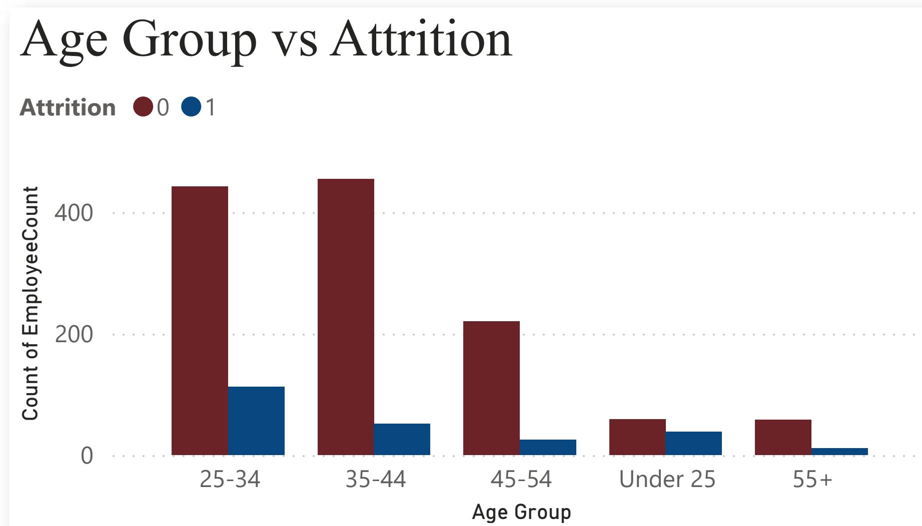
JobSatisfaction vs Attrition




Monthly Income by Attrition
Average of MonthlyIncome
6K
4K
2K
0K
No
Yes
Attrition_Label


Attrition by Age
Attrition_Label No Yes
Count of EmployeeNum...
60
40
20
0
20
30
40
50
60
Age


Attrition by Job Role
JobRole
Sales Executive
Research Scientist
Laboratory Technic...
Manufacturing Dir...
Healthcare Repres...
Manager
Sales Representative
Research Director
Human Resources
0
100
200
300
Count of EmployeeNumber


Age Group vs Attrition
Attrition 0 1
Count of EmployeeCount
400
200
0
25-34
35-44
45-54
Under 25
55+
Age Group

Employee Attrition Prevention Strategy

Overview

Based on the data analysis and SHAP explainability results from the employee attrition dataset, several high-risk factors have been identified that contribute to employee resignation. These include overtime work, low job satisfaction, low income, long commuting distance, and lack of promotion opportunities.

Workload & Overtime

- Monitor and limit overtime hours with upper caps.
- Introduce flexible working hours or hybrid/remote work options.

Job Satisfaction

- Conduct regular pulse surveys to measure satisfaction.
- Implement employee recognition programs.
- Improve autonomy and role clarity.

Compensation

- Review salary bands for at-risk roles.
- Provide performance-based bonuses or retention incentives.

Commute Issues

- Enable remote or hybrid work for employees who live far away.
- Offer relocation assistance or commuting benefits.

Career Development

- Launch a promotion tracking dashboard.
- Regularly provide training and development programs.
- Assign mentors to employees with longer tenures.

Employee Attrition Prevention Strategy

Retaining Younger Employees

- Offer career planning sessions for employees aged 25-34.
- Create fast-track programs for promotions and development.