

assignment-22-ipnyb

October 17, 2023

```
[ ]:
```

NAME -SAYANI ROY CHOUDHURY REGISTRATION NO.-21BCE10336

```
[ ]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
[ ]: #Import the Libraries.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
[ ]: # Load the Dataset.
df = pd.read_csv('/content/WA_Fn-UseC_-HR-Employee-Attrition.csv')
```

```
[ ]: df.head()
```

```
[ ]:
Age Attrition    BusinessTravel DailyRate    Department \
0  41      Yes      Travel_Rarely    1102      Sales
1  49      No  Travel_Frequently    279  Research & Development
2  37      Yes      Travel_Rarely    1373  Research & Development
3  33      No  Travel_Frequently    1392  Research & Development
4  27      No      Travel_Rarely    591  Research & Development

DistanceFromHome Education EducationField EmployeeCount EmployeeNumber \
0              1          2 Life Sciences              1              1
1              8          1 Life Sciences              1              2
2              2          2      Other              1              4
3              3          4 Life Sciences              1              5
4              2          1      Medical              1              7

... RelationshipSatisfaction StandardHours StockOptionLevel \
0  ...              1              80              0
```

1	...	4	80	1
2	...	2	80	0
3	...	3	80	0
4	...	4	80	1

	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	\
0	8	0	1	6	
1	10	3	3	10	
2	7	3	3	0	
3	8	3	3	8	
4	6	3	3	2	

	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager
0	4	0	5
1	7	1	7
2	0	0	0
3	7	3	0
4	2	2	2

[5 rows x 35 columns]

```
[ ]: df.shape
```

```
[ ]: (1470, 35)
```

```
[ ]: df.Attrition.value_counts()
```

```
[ ]: No      1233
      Yes      237
      Name: Attrition, dtype: int64
```

```
[ ]: df.info()
```

```
<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

```
[ ]: df.describe()
```

```

[ ]:
count      Age      DailyRate  DistanceFromHome  Education  EmployeeCount  \
count  1470.000000  1470.000000      1470.000000  1470.000000      1470.0
mean     36.923810   802.485714         9.192517    2.912925         1.0
std       9.135373   403.509100         8.106864    1.024165         0.0
min      18.000000   102.000000         1.000000    1.000000         1.0
25%      30.000000   465.000000         2.000000    2.000000         1.0
50%      36.000000   802.000000         7.000000    3.000000         1.0
75%      43.000000  1157.000000        14.000000    4.000000         1.0
max      60.000000  1499.000000        29.000000    5.000000         1.0

      EmployeeNumber  EnvironmentSatisfaction  HourlyRate  JobInvolvement  \
count      1470.000000      1470.000000  1470.000000      1470.000000
mean      1024.865306         2.721769    65.891156         2.729932
std        602.024335         1.093082    20.329428         0.711561
min         1.000000         1.000000    30.000000         1.000000
25%        491.250000         2.000000    48.000000         2.000000

```

50%	1020.500000	3.000000	66.000000	3.000000
75%	1555.750000	4.000000	83.750000	3.000000
max	2068.000000	4.000000	100.000000	4.000000

	JobLevel	...	RelationshipSatisfaction	StandardHours	\
count	1470.000000	...	1470.000000	1470.0	
mean	2.063946	...	2.712245	80.0	
std	1.106940	...	1.081209	0.0	
min	1.000000	...	1.000000	80.0	
25%	1.000000	...	2.000000	80.0	
50%	2.000000	...	3.000000	80.0	
75%	3.000000	...	4.000000	80.0	
max	5.000000	...	4.000000	80.0	

	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	\
count	1470.000000	1470.000000	1470.000000	
mean	0.793878	11.279592	2.799320	
std	0.852077	7.780782	1.289271	
min	0.000000	0.000000	0.000000	
25%	0.000000	6.000000	2.000000	
50%	1.000000	10.000000	3.000000	
75%	1.000000	15.000000	3.000000	
max	3.000000	40.000000	6.000000	

	WorkLifeBalance	YearsAtCompany	YearsInCurrentRole	\
count	1470.000000	1470.000000	1470.000000	
mean	2.761224	7.008163	4.229252	
std	0.706476	6.126525	3.623137	
min	1.000000	0.000000	0.000000	
25%	2.000000	3.000000	2.000000	
50%	3.000000	5.000000	3.000000	
75%	3.000000	9.000000	7.000000	
max	4.000000	40.000000	18.000000	

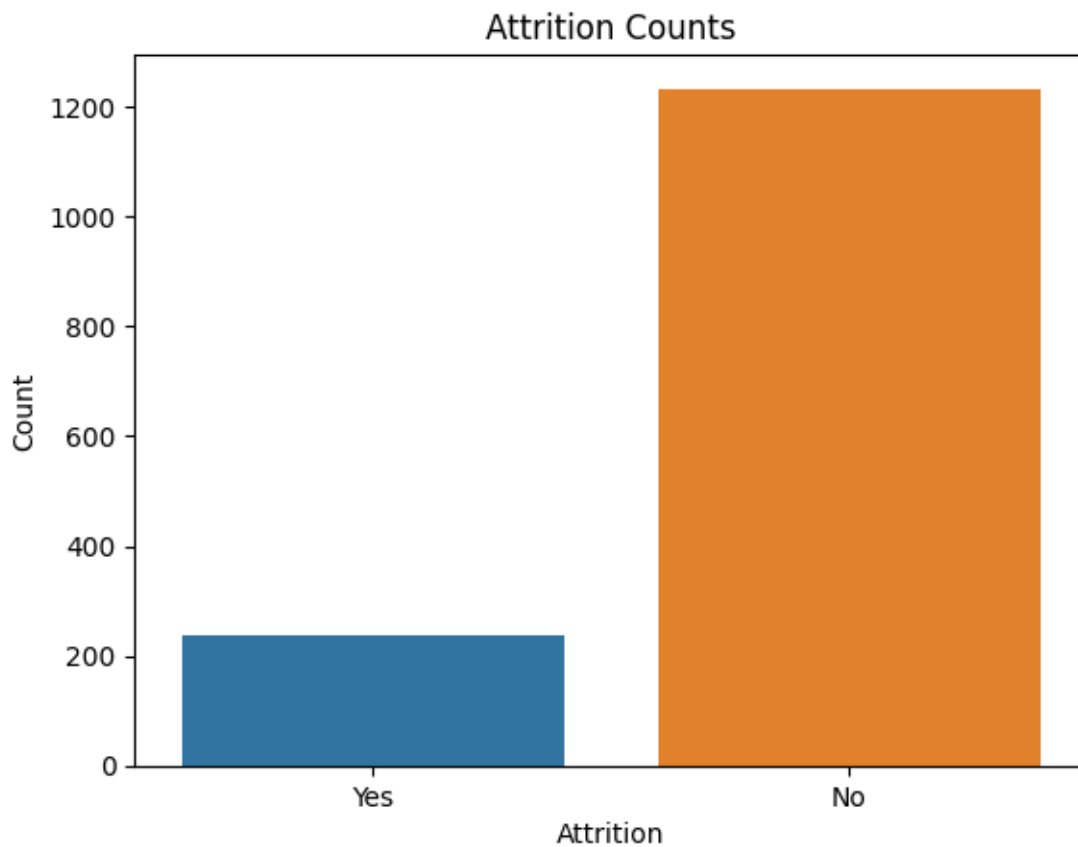
	YearsSinceLastPromotion	YearsWithCurrManager
count	1470.000000	1470.000000
mean	2.187755	4.123129
std	3.222430	3.568136
min	0.000000	0.000000
25%	0.000000	2.000000
50%	1.000000	3.000000
75%	3.000000	7.000000
max	15.000000	17.000000

[8 rows x 26 columns]

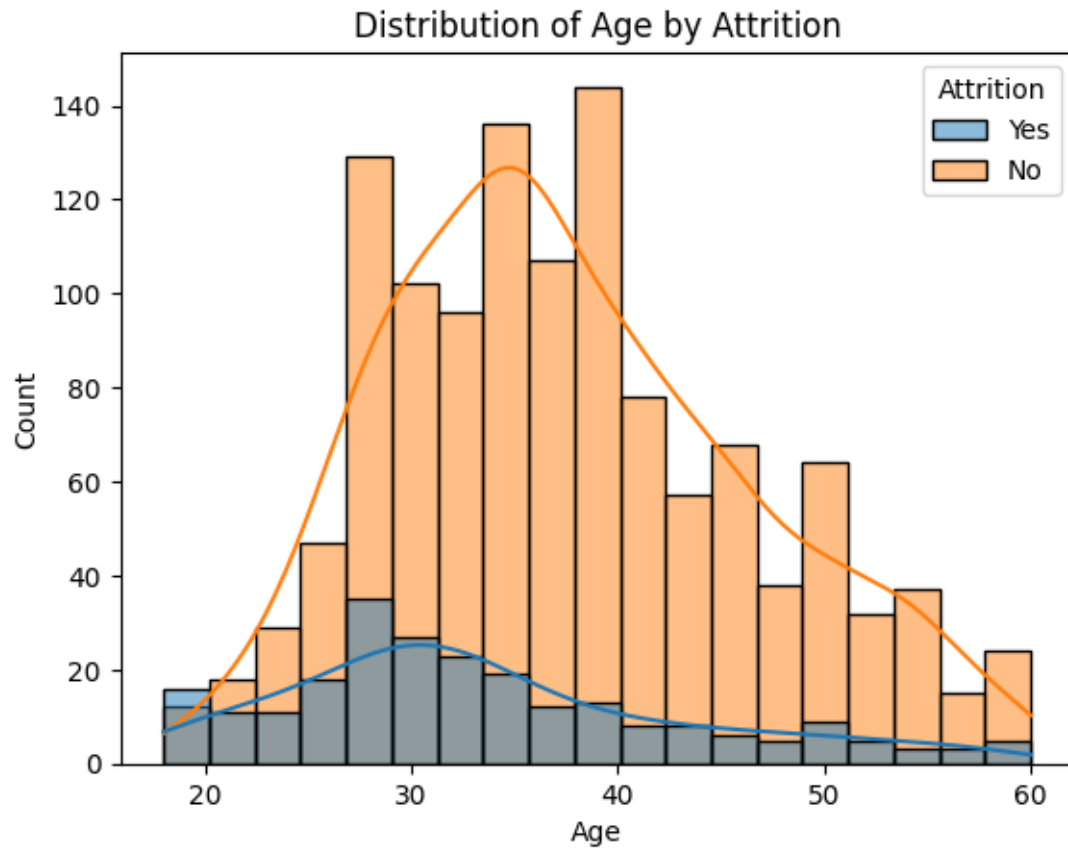
```
[ ]: #Checking for Null Values.  
df.isnull().any()
```

```
[ ]: Age                False  
Attrition              False  
BusinessTravel         False  
DailyRate              False  
Department             False  
DistanceFromHome       False  
Education              False  
EducationField          False  
EmployeeCount           False  
EmployeeNumber          False  
EnvironmentSatisfaction False  
Gender                 False  
HourlyRate             False  
JobInvolvement          False  
JobLevel               False  
JobRole                False  
JobSatisfaction         False  
MaritalStatus           False  
MonthlyIncome           False  
MonthlyRate            False  
NumCompaniesWorked      False  
Over18                 False  
OverTime                False  
PercentSalaryHike       False  
PerformanceRating       False  
RelationshipSatisfaction False  
StandardHours           False  
StockOptionLevel        False  
TotalWorkingYears       False  
TrainingTimesLastYear   False  
WorkLifeBalance         False  
YearsAtCompany          False  
YearsInCurrentRole       False  
YearsSinceLastPromotion  False  
YearsWithCurrManager     False  
dtype: bool
```

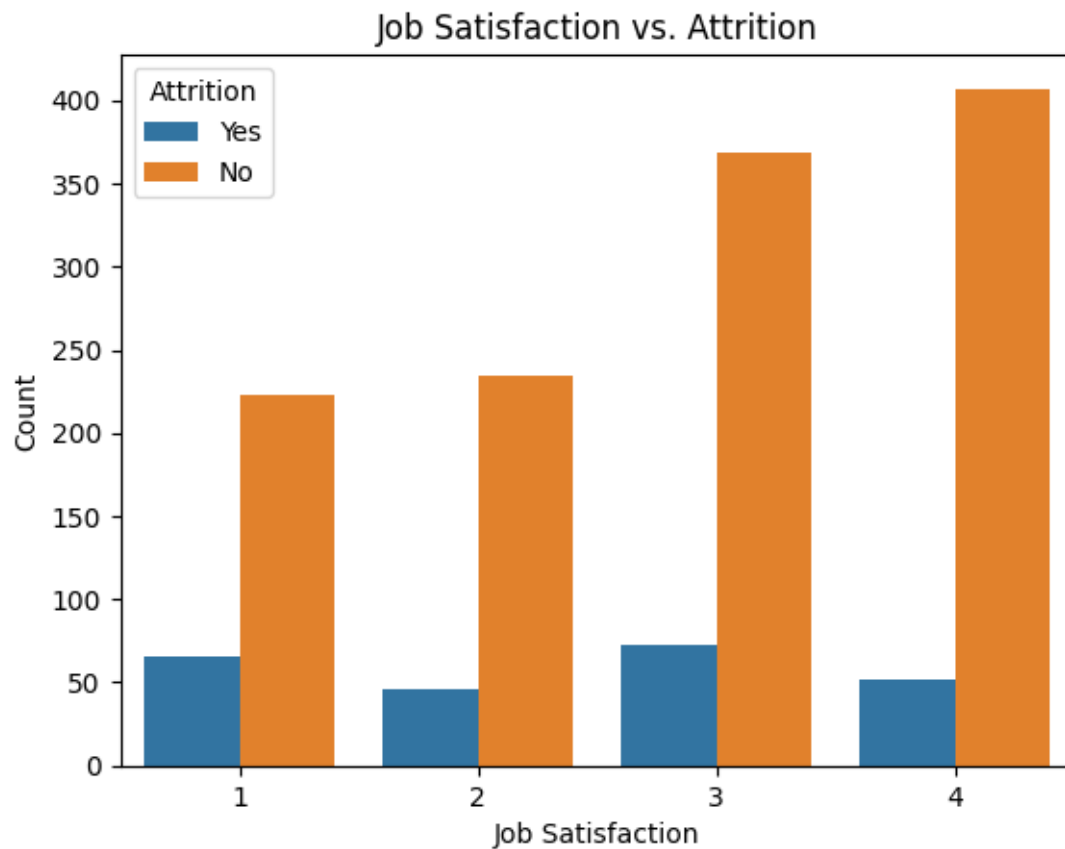
```
[ ]: sns.countplot(x='Attrition', data=df)  
plt.title('Attrition Counts')  
plt.xlabel('Attrition')  
plt.ylabel('Count')  
plt.show()
```



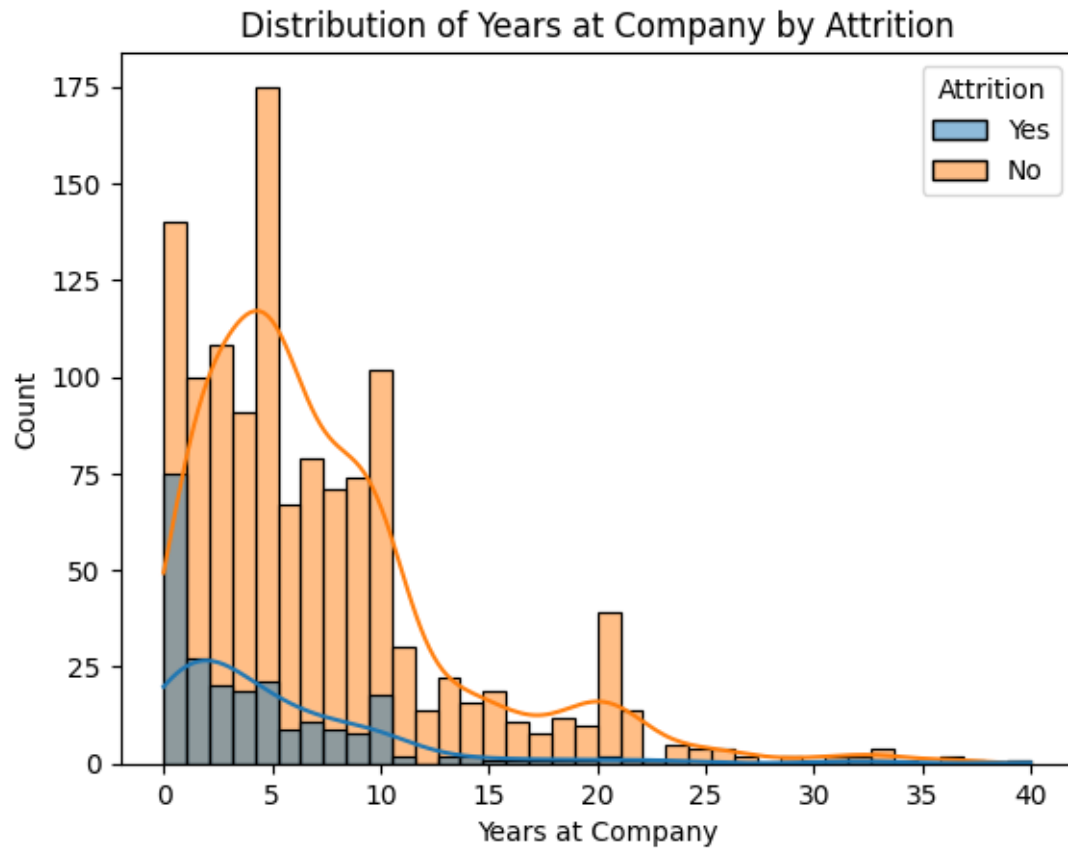
```
[ ]: sns.histplot(data=df, x='Age', hue='Attrition', kde=True)
plt.title('Distribution of Age by Attrition')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



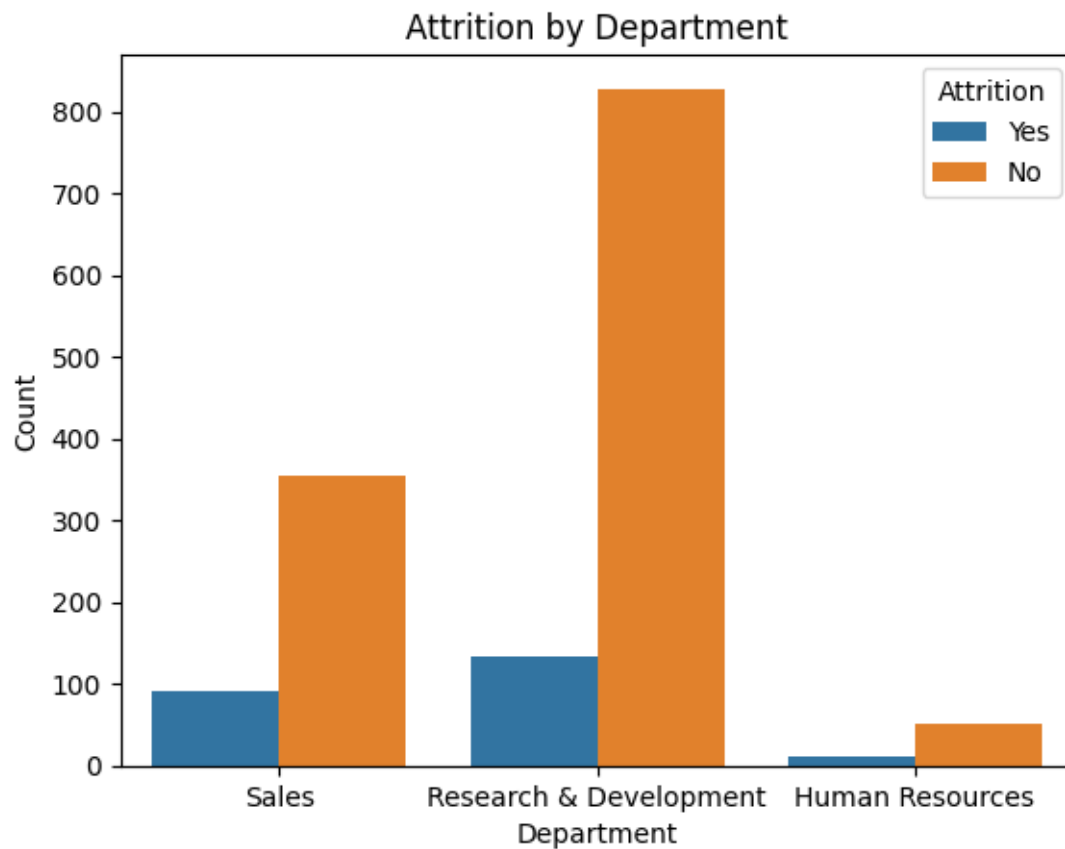
```
[ ]: sns.countplot(x='JobSatisfaction', hue='Attrition', data=df)
plt.title('Job Satisfaction vs. Attrition')
plt.xlabel('Job Satisfaction')
plt.ylabel('Count')
plt.show()
```



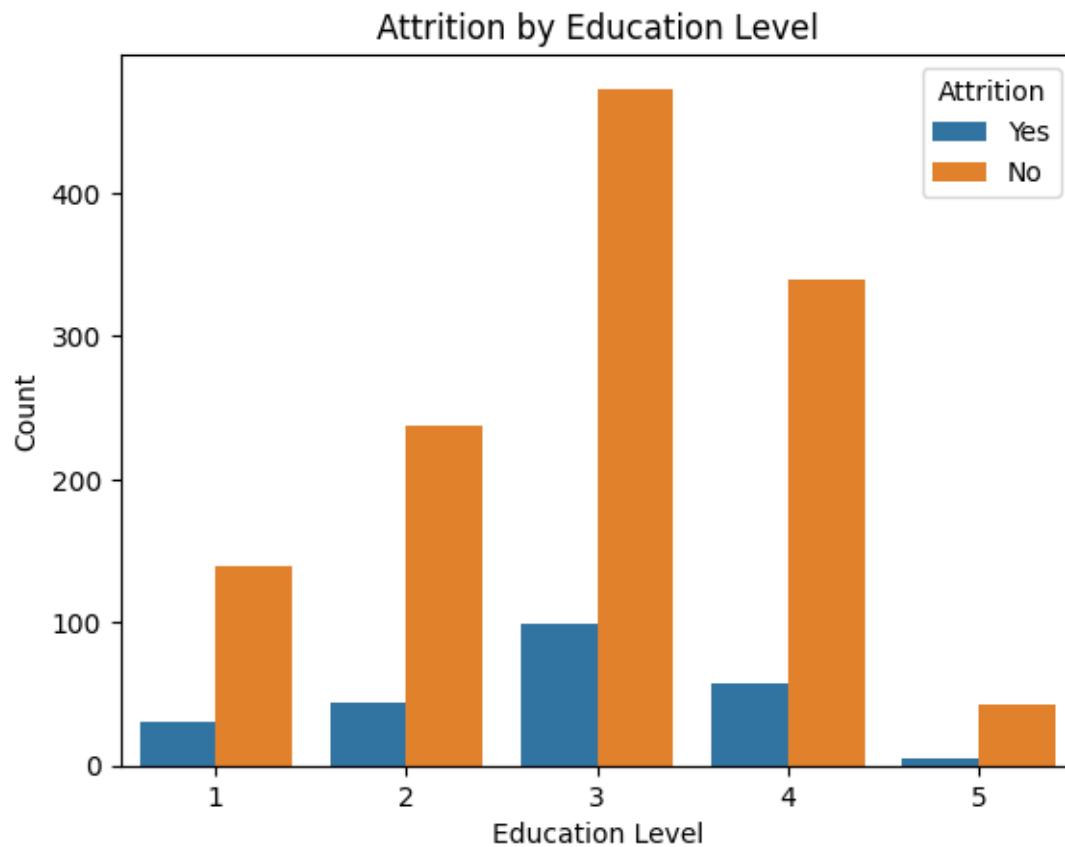
```
[ ]: sns.histplot(data=df, x='YearsAtCompany', hue='Attrition', kde=True)
plt.title('Distribution of Years at Company by Attrition')
plt.xlabel('Years at Company')
plt.ylabel('Count')
plt.show()
```

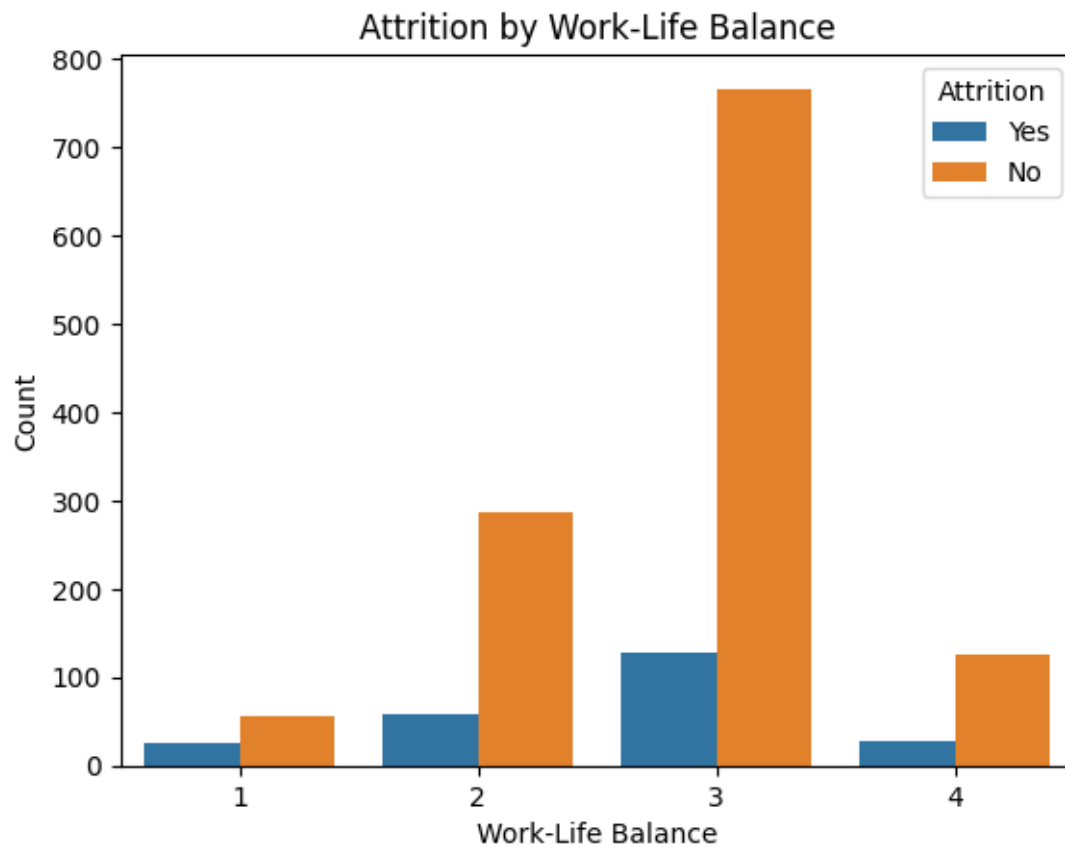
```
[ ]: sns.countplot(x='Department', hue='Attrition', data=df)
plt.title('Attrition by Department')
plt.xlabel('Department')
plt.ylabel('Count')
plt.show()
```



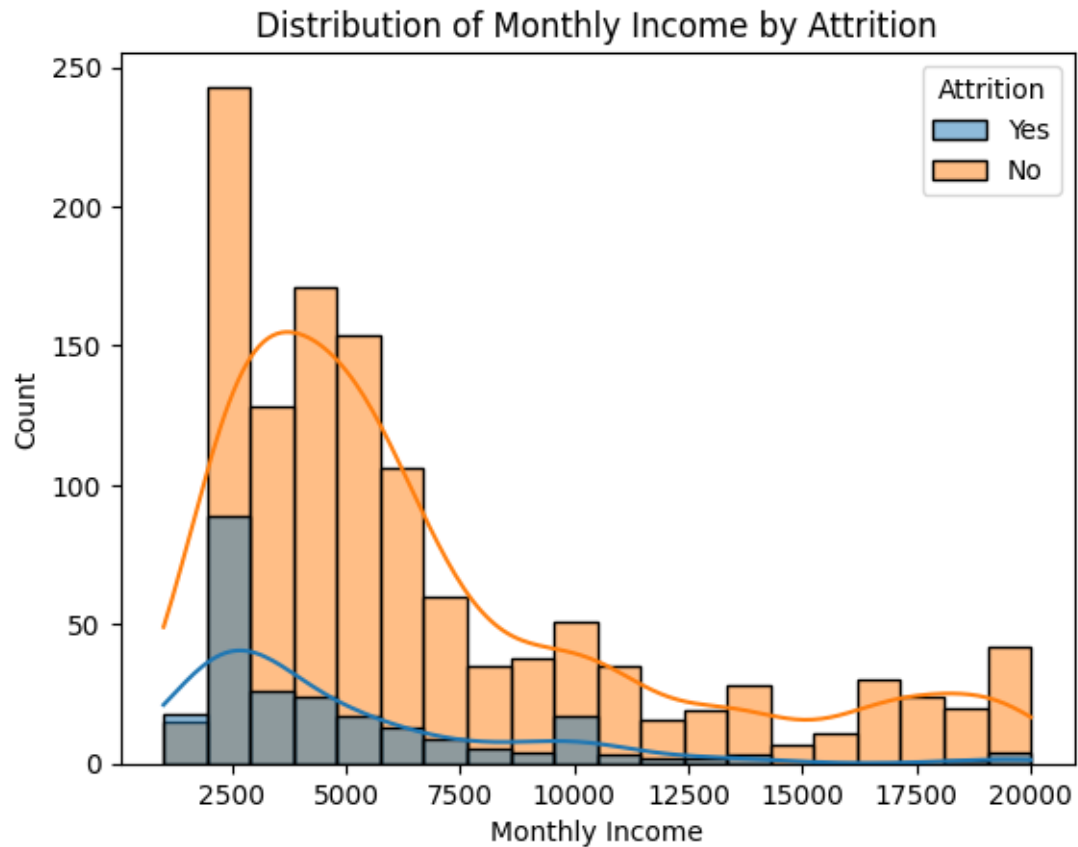
```
[ ]: sns.countplot(x='Education', hue='Attrition', data=df)
plt.title('Attrition by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.show()
```



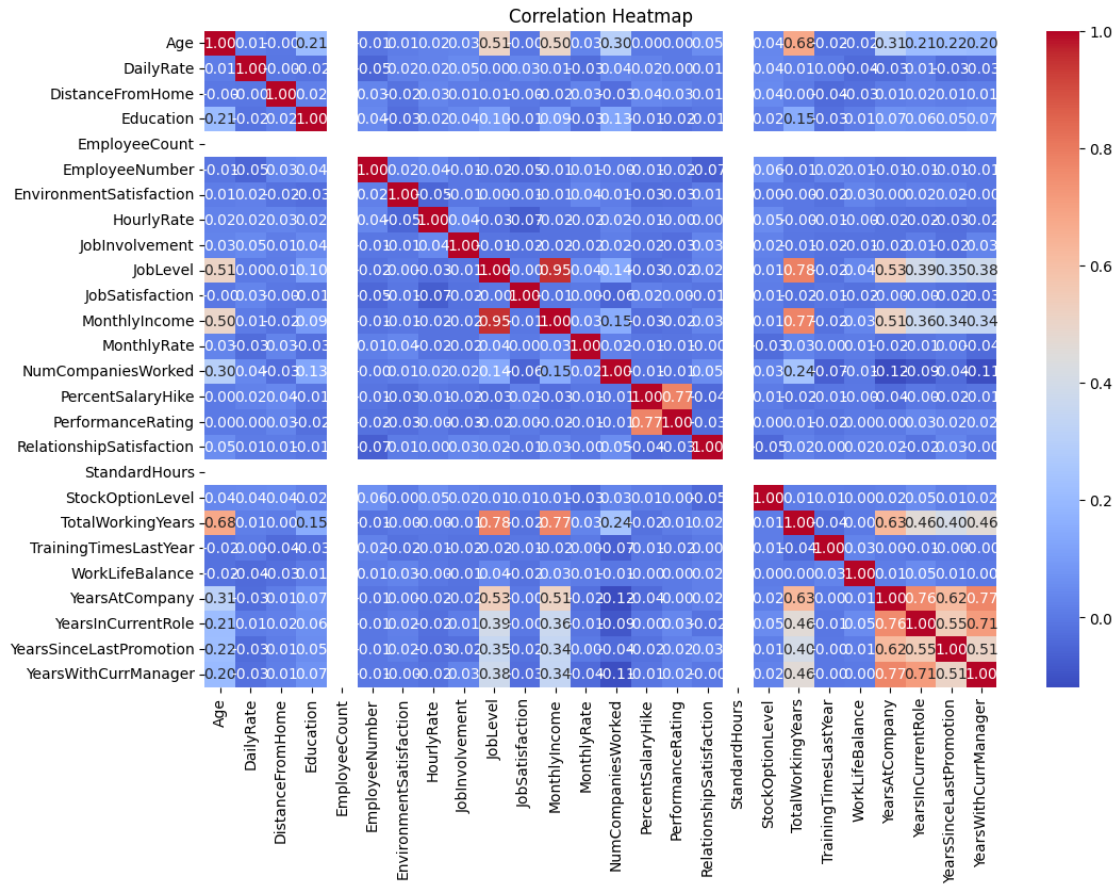
```
[ ]: sns.countplot(x='WorkLifeBalance', hue='Attrition', data=df)
plt.title('Attrition by Work-Life Balance')
plt.xlabel('Work-Life Balance')
plt.ylabel('Count')
plt.show()
```



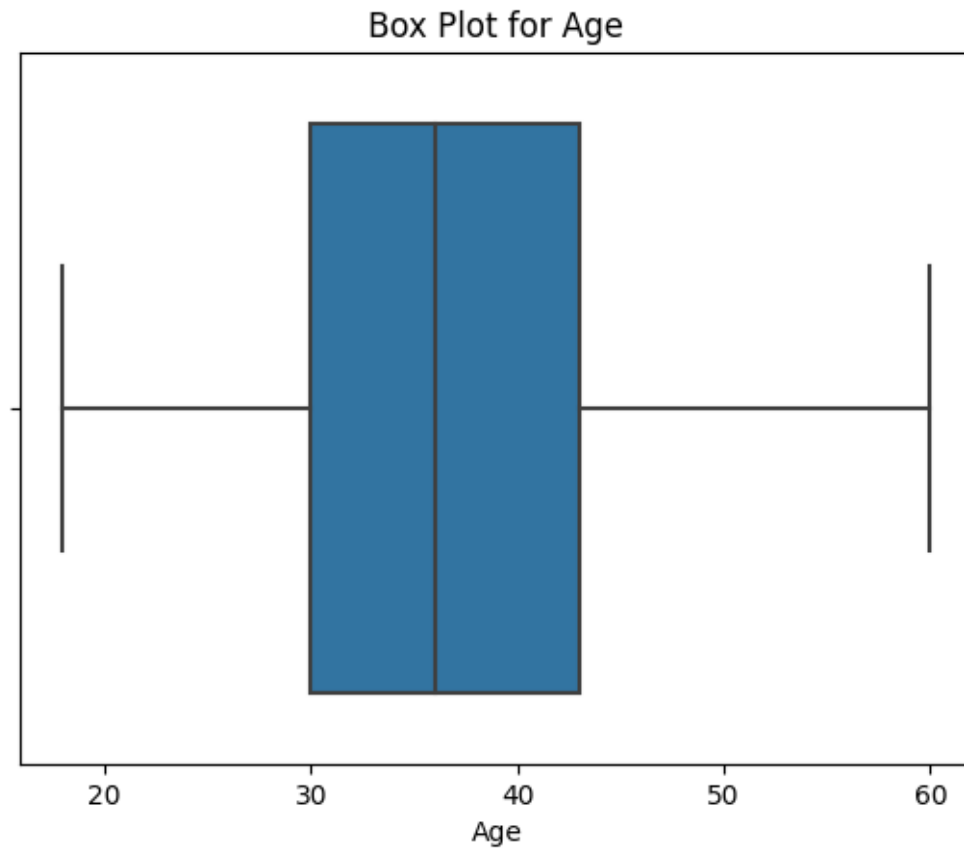
```
[ ]: sns.histplot(data=df, x='MonthlyIncome', hue='Attrition', kde=True)
plt.title('Distribution of Monthly Income by Attrition')
plt.xlabel('Monthly Income')
plt.ylabel('Count')
plt.show()
```



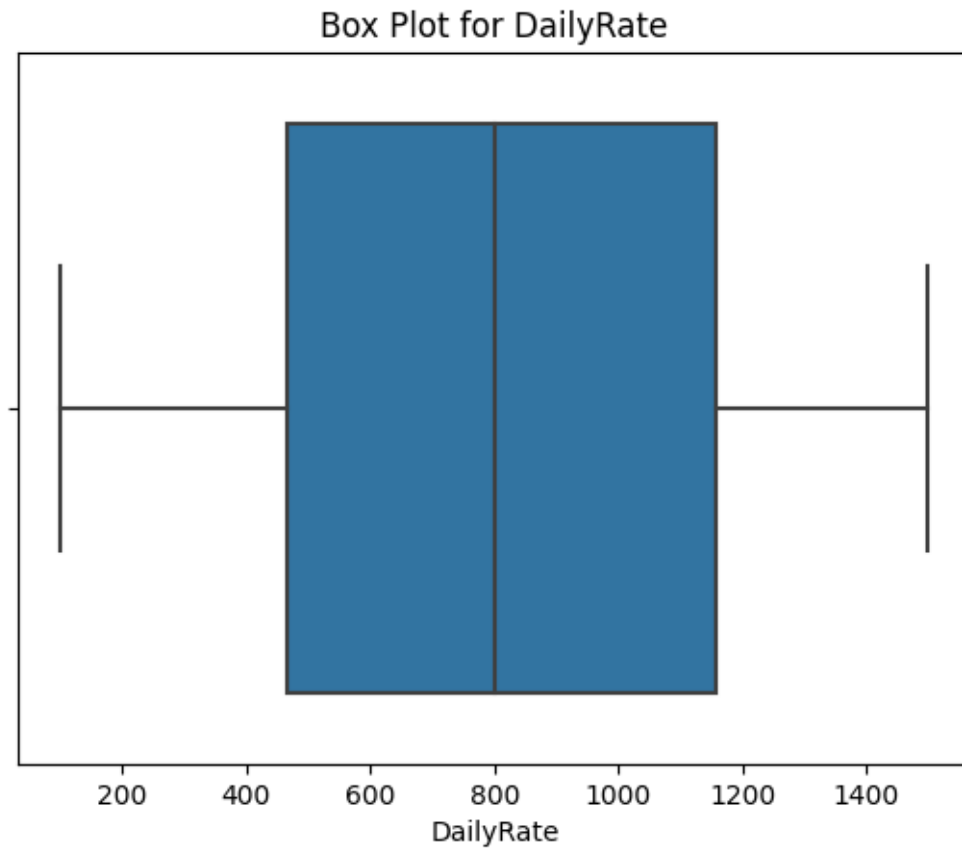
```
[ ]: # Correlation Heatmap
correlation_matrix = df.corr(numeric_only=True)
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



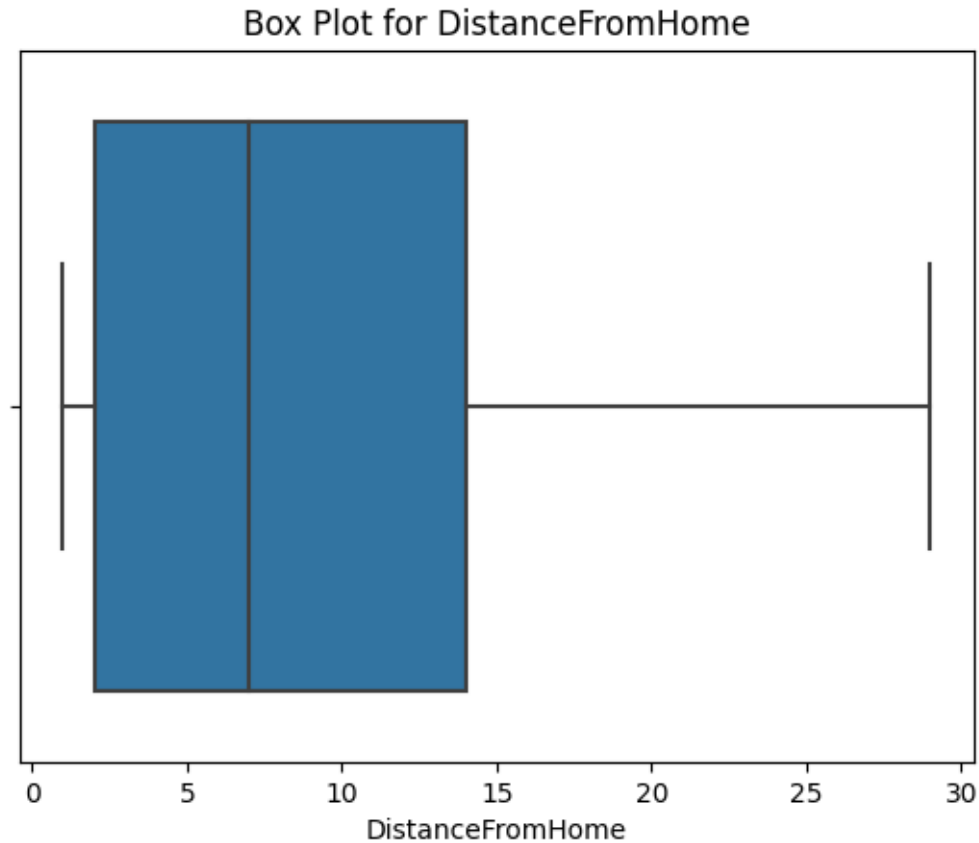
```
[ ]: # Box plot for Age
sns.boxplot(x='Age', data=df)
plt.title('Box Plot for Age')
plt.show()
```



```
[ ]: # Box plot for DailyRate
sns.boxplot(x='DailyRate', data=df)
plt.title('Box Plot for DailyRate')
plt.show()
```



```
[ ]: # Box plot for DistanceFromHome
sns.boxplot(x='DistanceFromHome', data=df)
plt.title('Box Plot for DistanceFromHome')
plt.show()
```

```
[ ]: # Define the target variable (dependent variable)
y = df['Attrition']

# Define the independent variables (features)
X = df.drop('Attrition', axis=1)
```

```
[ ]: # Perform one-hot encoding on categorical columns
X_encoded = pd.get_dummies(X, drop_first=True)
```

```
[ ]: X_encoded
```

```
[ ]:
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	\
0	41	1102	1	2	1	
1	49	279	8	1	1	
2	37	1373	2	2	1	
3	33	1392	3	4	1	
4	27	591	2	1	1	
...	
1465	36	884	23	2	1	
1466	39	613	6	1	1	

1467	27	155	4	3	1
1468	49	1023	2	3	1
1469	34	628	8	3	1

	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	\
0	1	2	94	3	
1	2	3	61	2	
2	4	4	92	2	
3	5	4	56	3	
4	7	1	40	3	
...	
1465	2061	3	41	4	
1466	2062	4	42	2	
1467	2064	2	87	4	
1468	2065	4	63	2	
1469	2068	2	82	4	

	JobLevel	...	JobRole_Laboratory Technician	JobRole_Manager	\
0	2	...	0	0	
1	2	...	0	0	
2	1	...	1	0	
3	1	...	0	0	
4	1	...	1	0	
...	
1465	2	...	1	0	
1466	3	...	0	0	
1467	2	...	0	0	
1468	2	...	0	0	
1469	2	...	1	0	

	JobRole_Manufacturing Director	JobRole_Research Director	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
...	
1465	0	0	
1466	0	0	
1467	1	0	
1468	0	0	
1469	0	0	

	JobRole_Research Scientist	JobRole_Sales Executive	\
0	0	1	
1	1	0	
2	0	0	

3		1		0
4		0		0
...	
1465		0		0
1466		0		0
1467		0		0
1468		0		1
1469		0		0

	JobRole_Sales Representative	MaritalStatus_Married \
0	0	0
1	0	1
2	0	0
3	0	1
4	0	1
...
1465	0	1
1466	0	1
1467	0	1
1468	0	1
1469	0	1

	MaritalStatus_Single	OverTime_Yes
0	1	1
1	0	0
2	1	1
3	0	1
4	0	0
...
1465	0	0
1466	0	0
1467	0	1
1468	0	0
1469	0	0

[1470 rows x 47 columns]

```
[ ]: from sklearn.preprocessing import MinMaxScaler

# Initialize the scaler
scaler = MinMaxScaler()

# Fit and transform the scaled features
X_scaled = scaler.fit_transform(X_encoded)

# Convert the scaled features back to a DataFrame (optional)
X_scaled_df = pd.DataFrame(X_scaled, columns=X_encoded.columns)
```

```
[ ]: X_scaled_df
```

```
[ ]:
      Age  DailyRate  DistanceFromHome  Education  EmployeeCount  \
0    0.547619    0.715820          0.000000        0.25           0.0
1    0.738095    0.126700          0.250000        0.00           0.0
2    0.452381    0.909807          0.035714        0.25           0.0
3    0.357143    0.923407          0.071429        0.75           0.0
4    0.214286    0.350036          0.035714        0.00           0.0
...    ...    ...    ...    ...    ...
1465  0.428571    0.559771          0.785714        0.25           0.0
1466  0.500000    0.365784          0.178571        0.00           0.0
1467  0.214286    0.037938          0.107143        0.50           0.0
1468  0.738095    0.659270          0.035714        0.50           0.0
1469  0.380952    0.376521          0.250000        0.50           0.0

      EmployeeNumber  EnvironmentSatisfaction  HourlyRate  JobInvolvement  \
0          0.000000          0.333333    0.914286    0.666667
1          0.000484          0.666667    0.442857    0.333333
2          0.001451          1.000000    0.885714    0.333333
3          0.001935          1.000000    0.371429    0.666667
4          0.002903          0.000000    0.142857    0.666667
...    ...    ...    ...    ...
1465      0.996613          0.666667    0.157143    1.000000
1466      0.997097          1.000000    0.171429    0.333333
1467      0.998065          0.333333    0.814286    1.000000
1468      0.998549          1.000000    0.471429    0.333333
1469      1.000000          0.333333    0.742857    1.000000

      JobLevel  ...  JobRole_Laboratory Technician  JobRole_Manager  \
0          0.25  ...                                0.0           0.0
1          0.25  ...                                0.0           0.0
2          0.00  ...                                1.0           0.0
3          0.00  ...                                0.0           0.0
4          0.00  ...                                1.0           0.0
...    ...    ...    ...    ...
1465      0.25  ...                                1.0           0.0
1466      0.50  ...                                0.0           0.0
1467      0.25  ...                                0.0           0.0
1468      0.25  ...                                0.0           0.0
1469      0.25  ...                                1.0           0.0

      JobRole_Manufacturing Director  JobRole_Research Director  \
0                                0.0                                0.0
1                                0.0                                0.0
2                                0.0                                0.0
3                                0.0                                0.0
4                                0.0                                0.0
```

...
1465	0.0	0.0
1466	0.0	0.0
1467	1.0	0.0
1468	0.0	0.0
1469	0.0	0.0

	JobRole_Research Scientist	JobRole_Sales Executive	\
0	0.0	1.0	
1	1.0	0.0	
2	0.0	0.0	
3	1.0	0.0	
4	0.0	0.0	
...	
1465	0.0	0.0	
1466	0.0	0.0	
1467	0.0	0.0	
1468	0.0	1.0	
1469	0.0	0.0	

	JobRole_Sales Representative	MaritalStatus_Married	\
0	0.0	0.0	
1	0.0	1.0	
2	0.0	0.0	
3	0.0	1.0	
4	0.0	1.0	
...	
1465	0.0	1.0	
1466	0.0	1.0	
1467	0.0	1.0	
1468	0.0	1.0	
1469	0.0	1.0	

	MaritalStatus_Single	OverTime_Yes
0	1.0	1.0
1	0.0	0.0
2	1.0	1.0
3	0.0	1.0
4	0.0	0.0
...
1465	0.0	0.0
1466	0.0	0.0
1467	0.0	1.0
1468	0.0	0.0
1469	0.0	0.0

[1470 rows x 47 columns]

```
[ ]: from sklearn.model_selection import train_test_split

# Split the data into training and testing sets (e.g., 80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
↳ random_state=42)
```

```
[ ]: from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import joblib # For model saving
```

```
[ ]: # Initialize the models
logistic_model = LogisticRegression(random_state=42)
decision_tree_model = DecisionTreeClassifier(random_state=42)
random_forest_model = RandomForestClassifier(random_state=42)
```

```
[ ]: # Training and testing the Logistic Regression model
logistic_model.fit(X_train, y_train)
logistic_predictions = logistic_model.predict(X_test)

# Training and testing the Decision Tree model
decision_tree_model.fit(X_train, y_train)
decision_tree_predictions = decision_tree_model.predict(X_test)

# Training and testing the Random Forest model
random_forest_model.fit(X_train, y_train)
random_forest_predictions = random_forest_model.predict(X_test)
```

```
[ ]: # Evaluation of Logistic Regression model
logistic_accuracy = accuracy_score(y_test, logistic_predictions)
logistic_report = classification_report(y_test, logistic_predictions)

print("Logistic Regression Model Accuracy:", logistic_accuracy)
print("Logistic Regression Model Classification Report:")
print(logistic_report)
```

Logistic Regression Model Accuracy: 0.891156462585034

Logistic Regression Model Classification Report:

	precision	recall	f1-score	support
No	0.91	0.97	0.94	255
Yes	0.67	0.36	0.47	39
accuracy			0.89	294
macro avg	0.79	0.67	0.70	294
weighted avg	0.88	0.89	0.88	294

```
[ ]: # Evaluation of Decision Tree model
decision_tree_accuracy = accuracy_score(y_test, decision_tree_predictions)
decision_tree_report = classification_report(y_test, decision_tree_predictions)

print("Decision Tree Model Accuracy:", decision_tree_accuracy)
print("Decision Tree Model Classification Report:")
print(decision_tree_report)
```

Decision Tree Model Accuracy: 0.7721088435374149

Decision Tree Model Classification Report:

	precision	recall	f1-score	support
No	0.87	0.86	0.87	255
Yes	0.17	0.18	0.17	39
accuracy			0.77	294
macro avg	0.52	0.52	0.52	294
weighted avg	0.78	0.77	0.78	294

```
[ ]: # Evaluation of Random Forest model
random_forest_accuracy = accuracy_score(y_test, random_forest_predictions)
random_forest_report = classification_report(y_test, random_forest_predictions)

print("Random Forest Model Accuracy:", random_forest_accuracy)
print("Random Forest Model Classification Report:")
print(random_forest_report)
```

Random Forest Model Accuracy: 0.8775510204081632

Random Forest Model Classification Report:

	precision	recall	f1-score	support
No	0.88	1.00	0.93	255
Yes	0.80	0.10	0.18	39
accuracy			0.88	294
macro avg	0.84	0.55	0.56	294
weighted avg	0.87	0.88	0.83	294