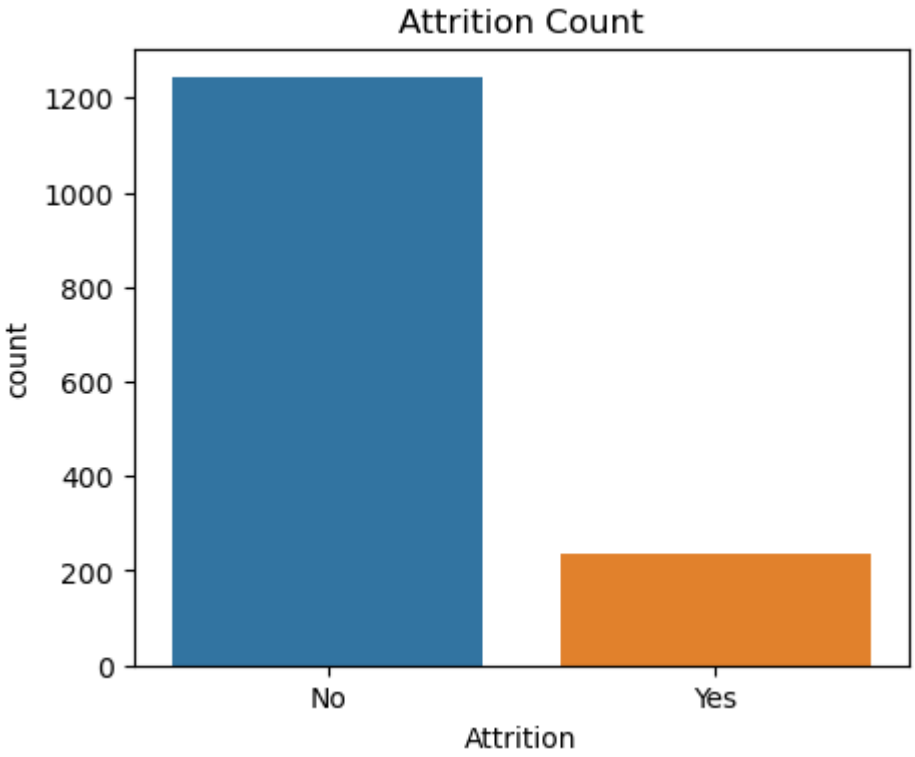
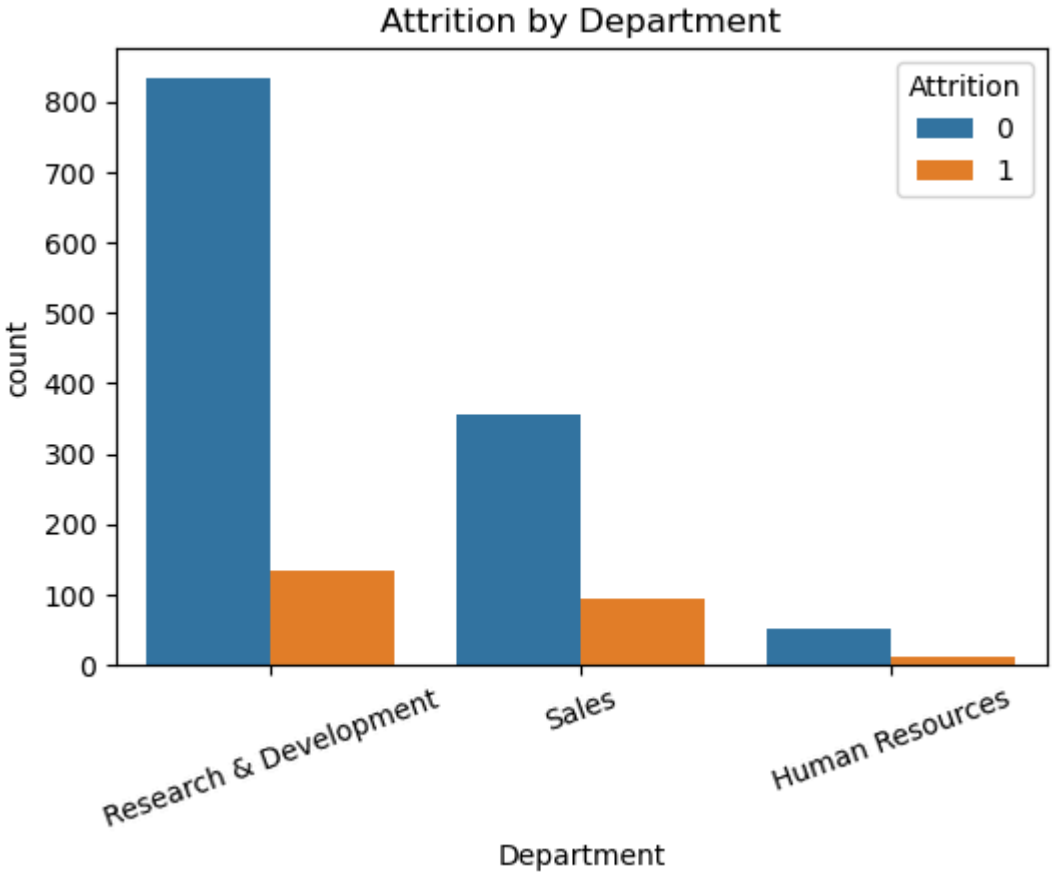
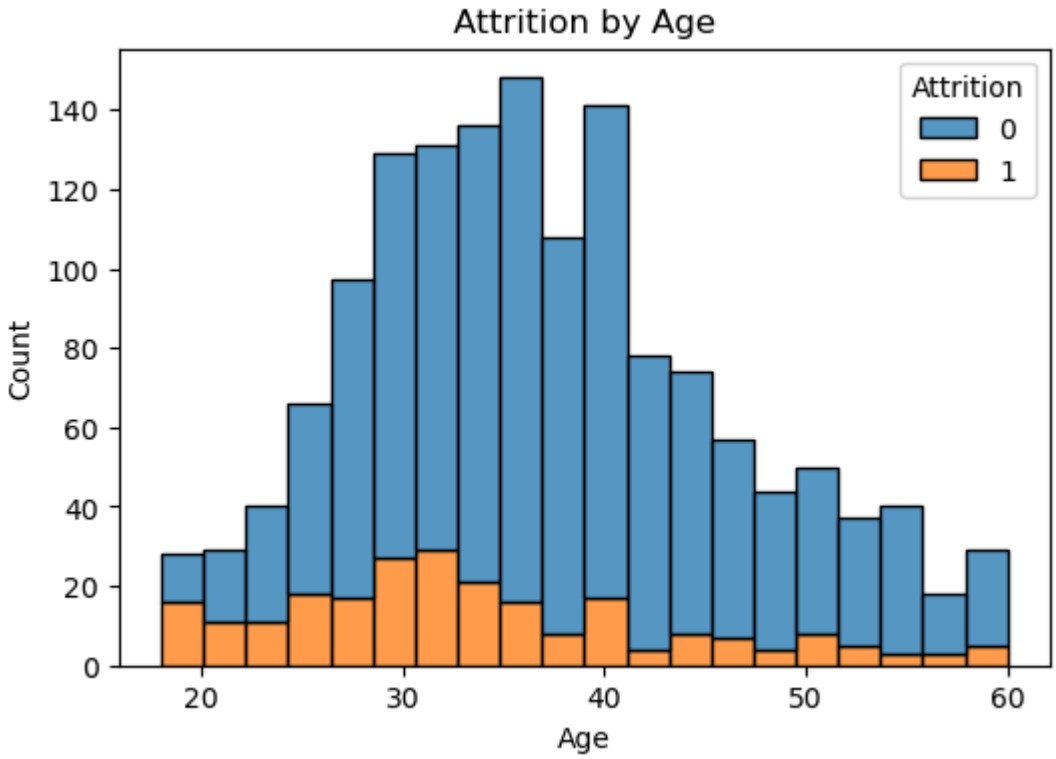
In [1]:
```
# Import necessary libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("HR_Analytics.csv")

# Basic overview
print(df.shape)
print(df.info())
print(df.describe())
print(df['Attrition'].value_counts())

# Convert 'Attrition' to binary
df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})

# Attrition count
plt.figure(figsize=(5, 4))
sns.countplot(x='Attrition', data=df)
plt.title('Attrition Count')
plt.xticks([0,1], ['No', 'Yes'])
plt.show()

# Attrition by Department
plt.figure(figsize=(6, 4))
sns.countplot(x='Department', hue='Attrition', data=df)
plt.title('Attrition by Department')
plt.xticks(rotation=20)
plt.show()

# Attrition by Age
plt.figure(figsize=(6, 4))
sns.histplot(data=df, x='Age', hue='Attrition', multiple='stack', bins=20)
plt.title('Attrition by Age')
plt.show()

# Attrition by Years at Company
plt.figure(figsize=(6, 4))
sns.boxplot(x='Attrition', y='YearsAtCompany', data=df)
plt.title('Attrition by Years at Company')
plt.show()

# Attrition by Job Role
plt.figure(figsize=(8, 4))
sns.countplot(y='JobRole', hue='Attrition', data=df)
plt.title('Attrition by Job Role')
plt.show()

# Attrition by Education Field
plt.figure(figsize=(8, 4))
sns.countplot(y='EducationField', hue='Attrition', data=df)
plt.title('Attrition by Education Field')
plt.show()
```

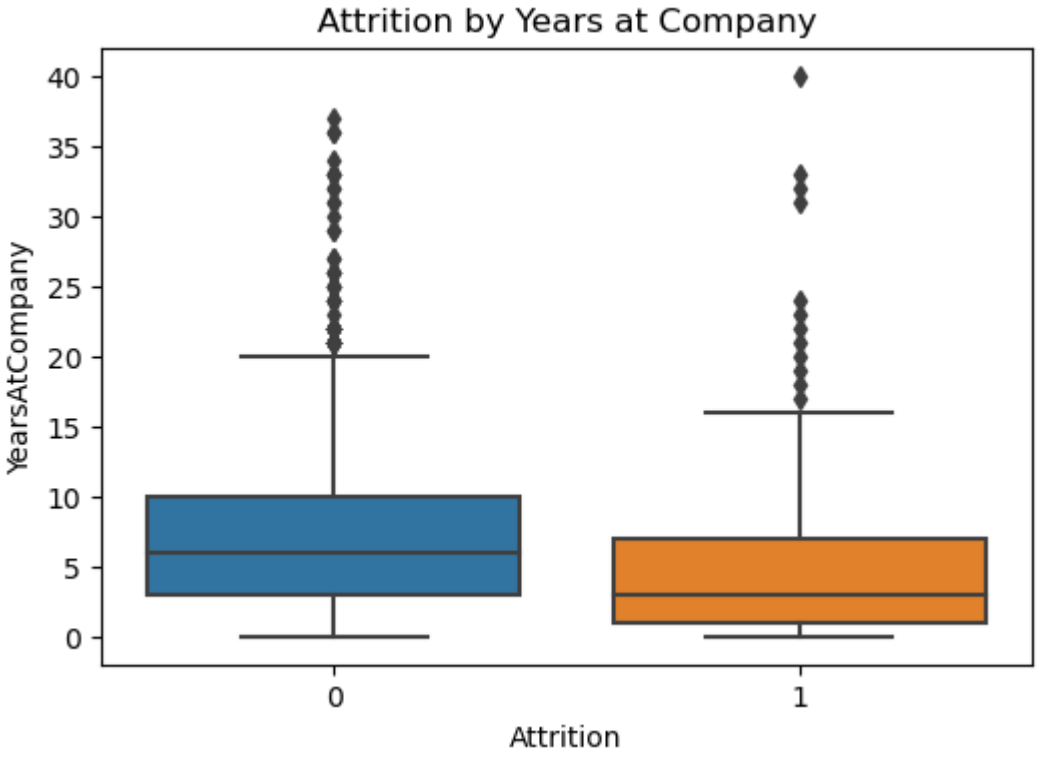
```
(1480, 38)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1480 entries, 0 to 1479
Data columns (total 38 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   EmpID                     1480 non-null   object
 1   Age                       1480 non-null   int64
 2   AgeGroup                  1480 non-null   object
 3   Attrition                 1480 non-null   object
 4   BusinessTravel            1480 non-null   object
 5   DailyRate                 1480 non-null   int64
 6   Department                1480 non-null   object
 7   DistanceFromHome          1480 non-null   int64
 8   Education                 1480 non-null   int64
 9   EducationField            1480 non-null   object
 10  EmployeeCount             1480 non-null   int64
 11  EmployeeNumber            1480 non-null   int64
 12  EnvironmentSatisfaction   1480 non-null   int64
 13  Gender                    1480 non-null   object
 14  HourlyRate                1480 non-null   int64
 15  JobInvolvement            1480 non-null   int64
 16  JobLevel                  1480 non-null   int64
 17  JobRole                   1480 non-null   object
 18  JobSatisfaction           1480 non-null   int64
 19  MaritalStatus             1480 non-null   object
 20  MonthlyIncome             1480 non-null   int64
 21  SalarySlab                1480 non-null   object
 22  MonthlyRate               1480 non-null   int64
 23  NumCompaniesWorked        1480 non-null   int64
 24  Over18                    1480 non-null   object
 25  OverTime                  1480 non-null   object
 26  PercentSalaryHike         1480 non-null   int64
 27  PerformanceRating         1480 non-null   int64
 28  RelationshipSatisfaction  1480 non-null   int64
 29  StandardHours             1480 non-null   int64
 30  StockOptionLevel          1480 non-null   int64
 31  TotalWorkingYears         1480 non-null   int64
 32  TrainingTimesLastYear     1480 non-null   int64
 33  WorkLifeBalance           1480 non-null   int64
 34  YearsAtCompany            1480 non-null   int64
 35  YearsInCurrentRole        1480 non-null   int64
 36  YearsSinceLastPromotion   1480 non-null   int64
 37  YearsWithCurrManager      1423 non-null   float64
dtypes: float64(1), int64(25), object(12)
memory usage: 439.5+ KB
None
                Age    DailyRate  DistanceFromHome    Education  EmployeeCount  \
count  1480.000000  1480.000000       1480.000000  1480.000000         1480.0
mean     36.917568   801.384459          9.220270     2.910811            1.0
std       9.128559   403.126988          8.131201     1.023796            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   800.000000          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     1480.000000              1480.000000  1480.000000     1480.000000
mean      1031.860811                 2.724324    65.845270        2.729730
std        605.955046                 1.092579    20.328266        0.713007
min          1.000000                 1.000000    30.000000        1.000000
25%        493.750000                 2.000000    48.000000        2.000000
50%       1027.500000                 3.000000    66.000000        3.000000
```

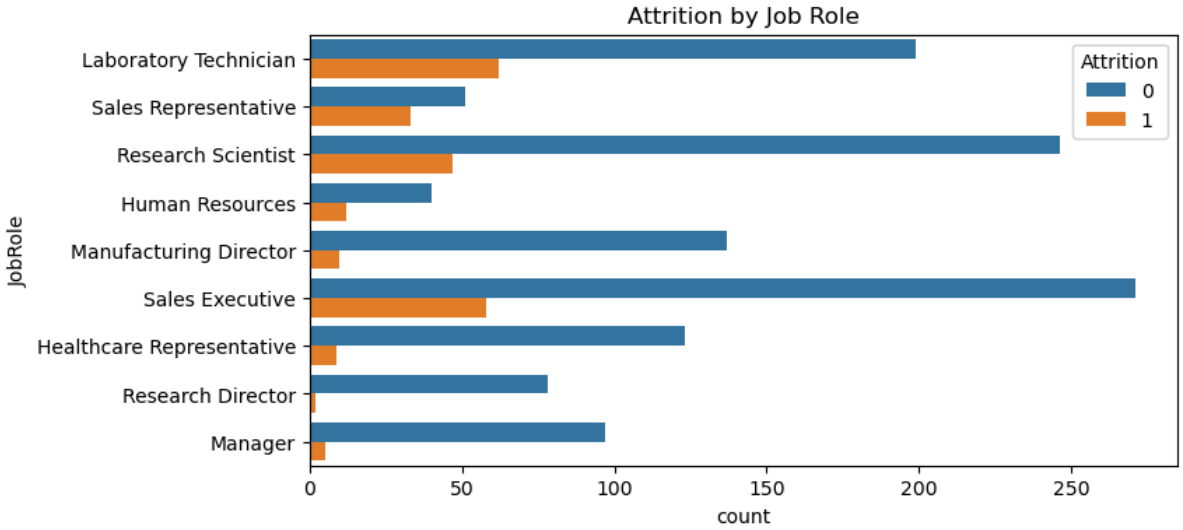
```
75%       1568.250000                4.000000    83.000000        3.000000
max       2068.000000                4.000000   100.000000        4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours  \
count  1480.000000  ...               1480.000000         1480.0
mean      2.064865  ...                  2.708784           80.0
std       1.105574  ...                  1.081995            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       1480.000000        1480.000000            1480.000000
mean           0.791892          11.281757               2.797973
std            0.850527           7.770870               1.288791
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      1480.000000     1480.000000         1480.000000
mean          2.760811        7.009459            4.228378
std           0.707024        6.117945            3.616020
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              1480.000000           1423.000000
mean                  2.182432              4.118060
std                   3.219357              3.555484
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]
Attrition
No     1242
Yes     238
Name: count, dtype: int64
```

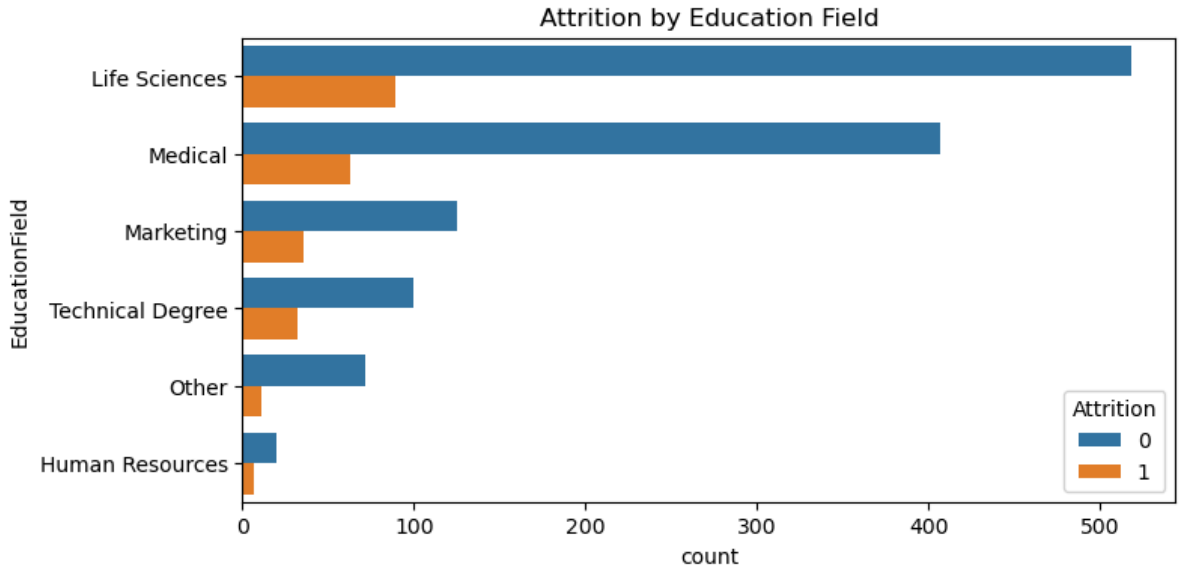

Attrition Count
1200
1000
800
600
400
200
0
count
No
Yes
Attrition


Attrition by Department
Attrition
0
1
800
700
600
500
400
300
200
100
0
count
Research & Development
Sales
Human Resources
Department


Attrition by Age
Attrition
0
1
Count
Age
0
20
40
60
80
100
120
140
20
30
40
50
60


Attrition by Years at Company
YearsAtCompany
Attrition
0
5
10
15
20
25
30
35
40
0
1





In [7]:
```
from sklearn.preprocessing import LabelEncoder

# Identify categorical columns
cat_cols = df.select_dtypes(include='object').columns
print(cat_cols)

# Apply Label Encoding for simplicity (One-Hot can also be considered)
le = LabelEncoder()
for col in cat_cols:
    df[col] = le.fit_transform(df[col])
```

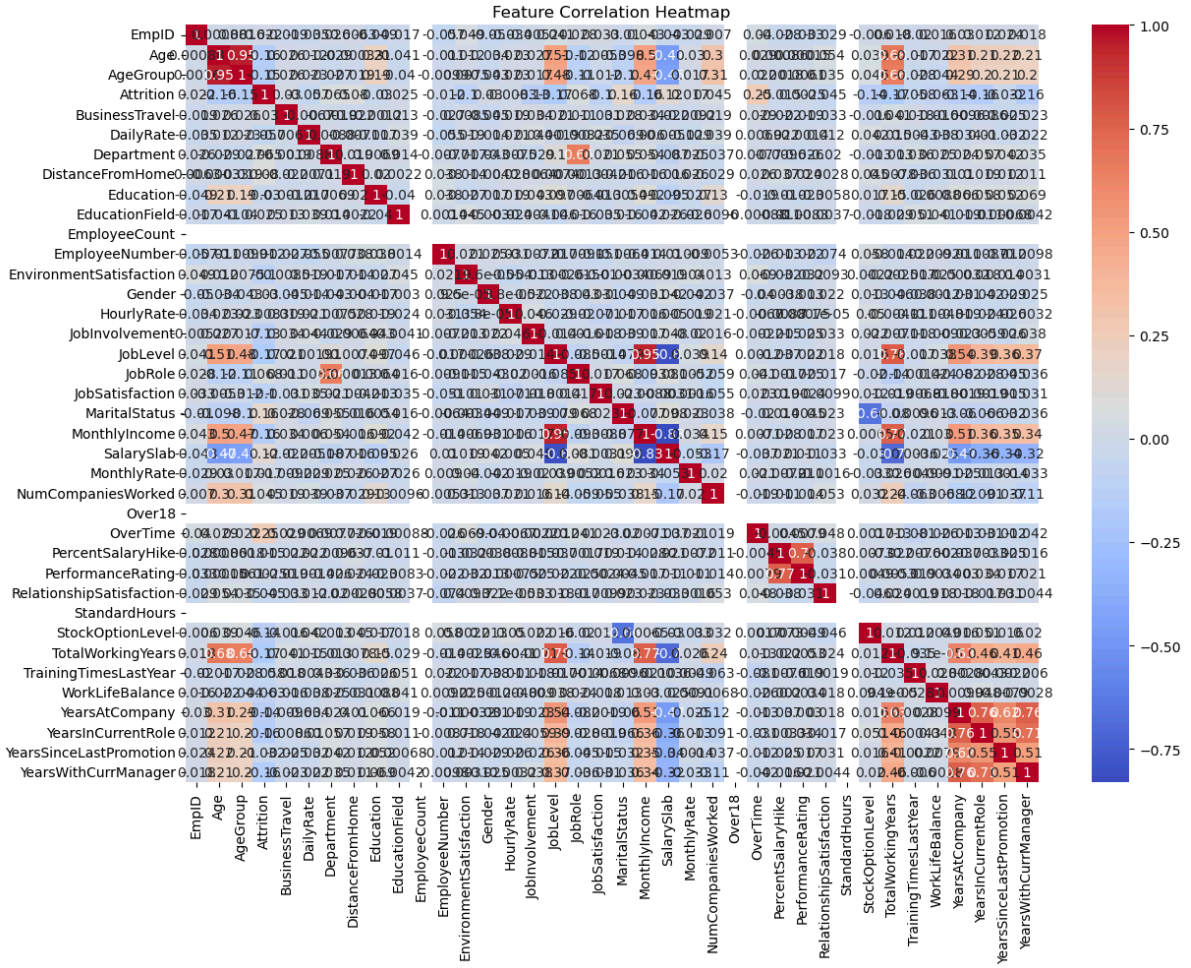
```
Index(['EmpID', 'AgeGroup', 'Attrition', 'BusinessTravel', 'Department',
       'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'SalarySlab',
       'Over18', 'OverTime'],
      dtype='object')
```

In [8]:
```
print(df.isnull().sum())
```

```
EmpID                       0
Age                         0
AgeGroup                    0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
SalarySlab                  0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager       57
dtype: int64
```

In [9]:

```
plt.figure(figsize=(14,10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Feature Correlation Heatmap')
plt.show()
```



In [12]:
```
from sklearn.model_selection import train_test_split

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

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [13]:
```
print(X_train.isnull().sum().sort_values(ascending=False).head(10))
```

```
YearsWithCurrManager      46
MonthlyIncome              0
MonthlyRate                0
NumCompaniesWorked         0
Over18                     0
OverTime                   0
PercentSalaryHike          0
PerformanceRating          0
RelationshipSatisfaction   0
StandardHours              0
dtype: int64
```

In [14]:
```
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='median')

X_train['YearsWithCurrManager'] = imputer.fit_transform(X_train[['YearsWithCurrMana
X_test['YearsWithCurrManager'] = imputer.transform(X_test[['YearsWithCurrManager']]
```

In [15]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

logreg = LogisticRegression(max_iter=1000)
```

```
logreg.fit(X_train, y_train)

y_pred = logreg.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
print('Accuracy:', accuracy_score(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      1.00      0.92       248
           1       0.80      0.08      0.15        48

    accuracy                           0.85       296
   macro avg       0.82      0.54      0.53       296
weighted avg       0.84      0.85      0.79       296

[[247   1]
 [ 44   4]]
Accuracy: 0.847972972972973
```

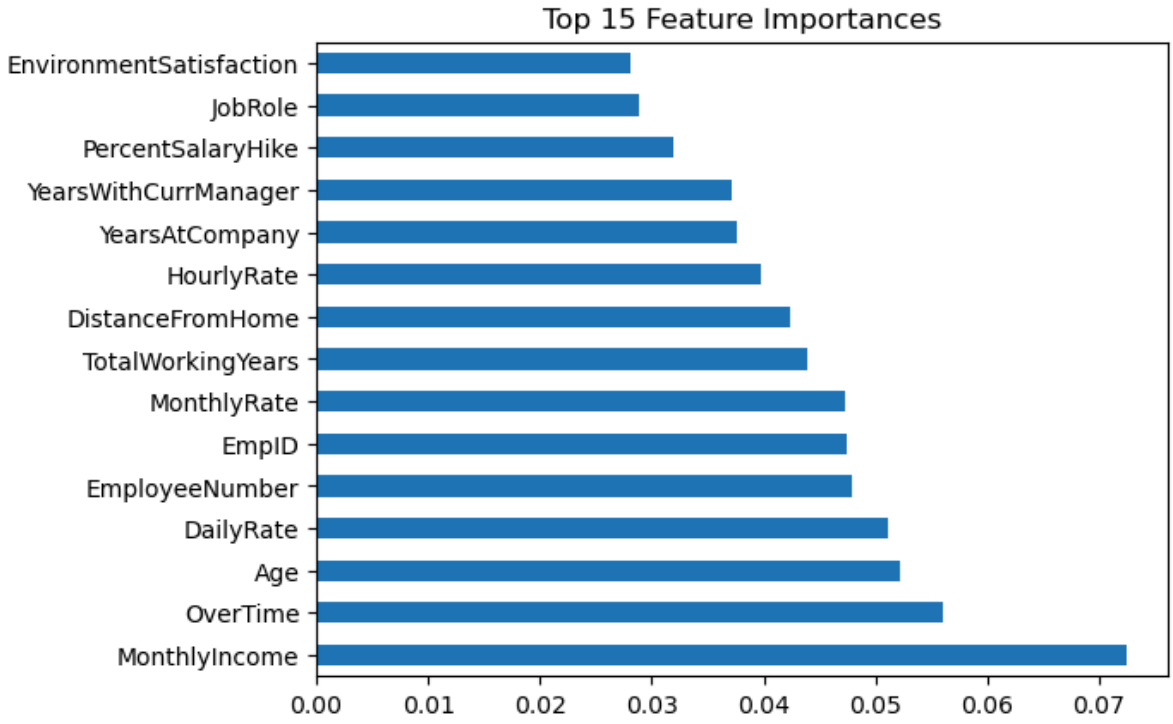
In [16]:

```
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(random_state=42)
rfc.fit(X_train, y_train)

y_pred_rfc = rfc.predict(X_test)

print(classification_report(y_test, y_pred_rfc))
print(confusion_matrix(y_test, y_pred_rfc))
print('Accuracy:', accuracy_score(y_test, y_pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.86      0.99      0.92       248
           1       0.75      0.19      0.30        48

    accuracy                           0.86       296
   macro avg       0.81      0.59      0.61       296
weighted avg       0.84      0.86      0.82       296

[[245   3]
 [ 39   9]]
Accuracy: 0.8581081081081081
```

In [17]:

```
importances = pd.Series(rfc.feature_importances_, index=X.columns)
importances.nlargest(15).plot(kind='barh')
plt.title('Top 15 Feature Importances')
plt.show()
```


Top 15 Feature Importances
EnvironmentSatisfaction
JobRole
PercentSalaryHike
YearsWithCurrManager
YearsAtCompany
HourlyRate
DistanceFromHome
TotalWorkingYears
MonthlyRate
EmpID
EmployeeNumber
DailyRate
Age
OverTime
MonthlyIncome
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07

In [ ]: