In [ ]:
```
### Imports
import mrmr
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.feature_selection import RFECV
from sklearn.linear_model import LogisticRegression
```

In [ ]:
```
### Import data
data = pd.read_csv("train.csv")

# Get general info
print(data.info(), "\n\n\n")
print(data.describe(), "\n\n\n")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1340 entries, 0 to 1339
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   EmployeeID                1340 non-null   int64 
 1   Age                       1340 non-null   int64 
 2   Attrition                 1340 non-null   object
 3   BusinessTravel            1340 non-null   object
 4   DailyRate                 1340 non-null   int64 
 5   Department                1340 non-null   object
 6   DistanceFromHome          1340 non-null   int64 
 7   Education                 1340 non-null   int64 
 8   EducationField            1340 non-null   object
 9   EmployeeCount             1340 non-null   int64 
 10  EnvironmentSatisfaction   1340 non-null   int64 
 11  Gender                    1340 non-null   object
 12  HourlyRate                1340 non-null   int64 
 13  JobInvolvement            1340 non-null   int64 
 14  JobLevel                  1340 non-null   int64 
 15  JobRole                   1340 non-null   object
 16  JobSatisfaction           1340 non-null   int64 
 17  MaritalStatus             1340 non-null   object
 18  MonthlyIncome             1340 non-null   int64 
 19  MonthlyRate               1340 non-null   int64 
 20  NumCompaniesWorked        1340 non-null   int64 
 21  Over18                    1340 non-null   object
 22  OverTime                  1340 non-null   object
 23  PercentSalaryHike         1340 non-null   int64 
 24  PerformanceRating         1340 non-null   int64 
 25  RelationshipSatisfaction  1340 non-null   int64 
 26  StandardHours             1340 non-null   int64 
 27  Shift                     1340 non-null   int64 
 28  TotalWorkingYears         1340 non-null   int64 
 29  TrainingTimesLastYear     1340 non-null   int64 
 30  WorkLifeBalance           1340 non-null   int64 
 31  YearsAtCompany            1340 non-null   int64 
 32  YearsInCurrentRole        1340 non-null   int64 
 33  YearsSinceLastPromotion   1340 non-null   int64 
 34  YearsWithCurrManager      1340 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 366.5+ KB
None
```

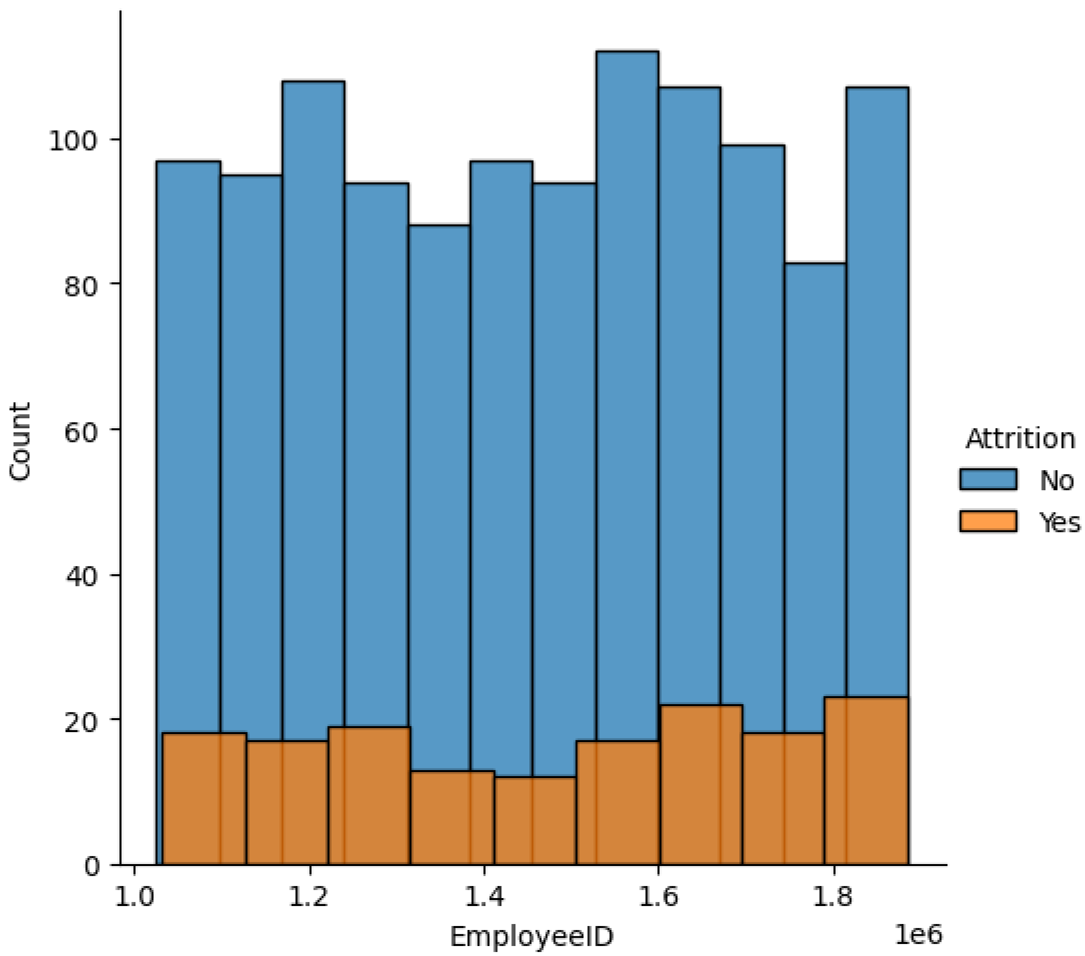
```
         EmployeeID          Age    DailyRate  DistanceFromHome    Educatio
n  \
count  1.340000e+03  1340.000000  1340.000000       1340.000000  1340.00000
0   
mean   1.460265e+06    36.580597   799.197761          9.193284     2.92462
7   
std    2.494821e+05     9.013072   399.333256          8.141621     1.03608
8   
min    1.025177e+06    18.000000   102.000000          1.000000     1.00000
0   
```

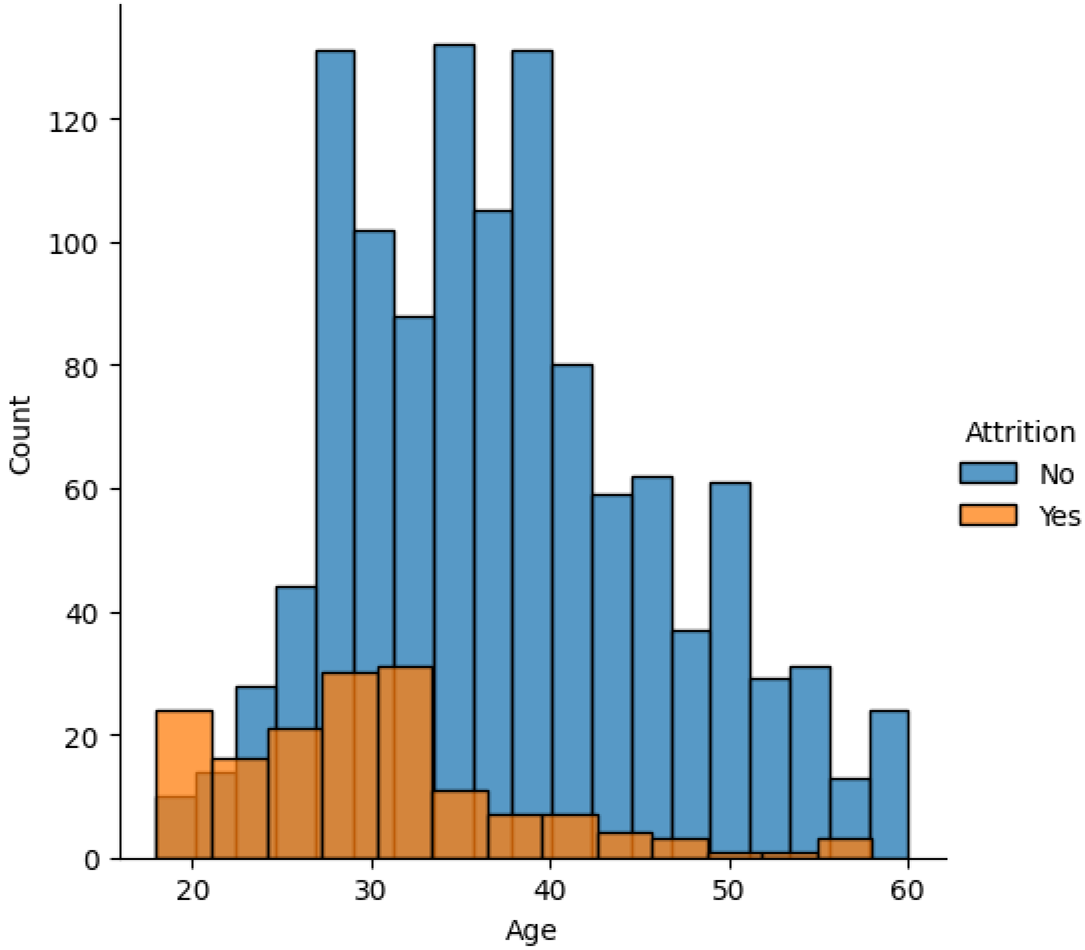
```
25%    1.237599e+06    30.000000   465.000000         2.000000     2.00000
0
50%    1.469862e+06    35.000000   796.000000         7.000000     3.00000
0
75%    1.670131e+06    42.000000  1153.000000        14.000000     4.00000
0
max    1.886378e+06    60.000000  1499.000000        29.000000     5.00000
0

       EmployeeCount  EnvironmentSatisfaction   HourlyRate  JobInvolvement
\
count         1340.0              1340.000000  1340.000000     1340.000000
mean             1.0                 2.709701    65.559701        2.717910
std              0.0                 1.099961    20.335025        0.717523
min              1.0                 1.000000    30.000000        1.000000
25%              1.0                 2.000000    48.000000        2.000000
50%              1.0                 3.000000    65.000000        3.000000
75%              1.0                 4.000000    83.000000        3.000000
max              1.0                 4.000000   100.000000        4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours        Shi
ft  \
count  1340.000000  ...               1340.000000         1340.0  1340.0000
00
mean      2.051493  ...                  2.700000           80.0     0.8082
09
std       1.104491  ...                  1.079858            0.0     0.8562
51
min       1.000000  ...                  1.000000           80.0     0.0000
00
25%       1.000000  ...                  2.000000           80.0     0.0000
00
50%       2.000000  ...                  3.000000           80.0     1.0000
00
75%       3.000000  ...                  4.000000           80.0     1.0000
00
max       5.000000  ...                  4.000000           80.0     3.0000
00

       TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
count        1340.000000            1340.000000      1340.000000
mean           11.222388               2.785821         2.771642
std             7.696043               1.263473         0.700007
min             0.000000               0.000000         1.000000
25%             6.000000               2.000000         2.000000
50%            10.000000               3.000000         3.000000
75%            15.000000               3.000000         3.000000
max            40.000000               6.000000         4.000000

       YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
count     1340.000000         1340.000000              1340.000000
mean         7.070149            4.272388                 2.175373
std          6.039663            3.677798                 3.222376
min          0.000000            0.000000                 0.000000
25%          3.000000            2.000000                 0.000000
50%          5.000000            3.000000                 1.000000
```

```
75%          10.000000           7.000000                         3.000000
max          40.000000          18.000000                        15.000000

       YearsWithCurrManager
count           1340.000000
mean               4.167164
std                3.581605
min                0.000000
25%                2.000000
50%                3.000000
75%                7.000000
max               17.000000

[8 rows x 26 columns]
```

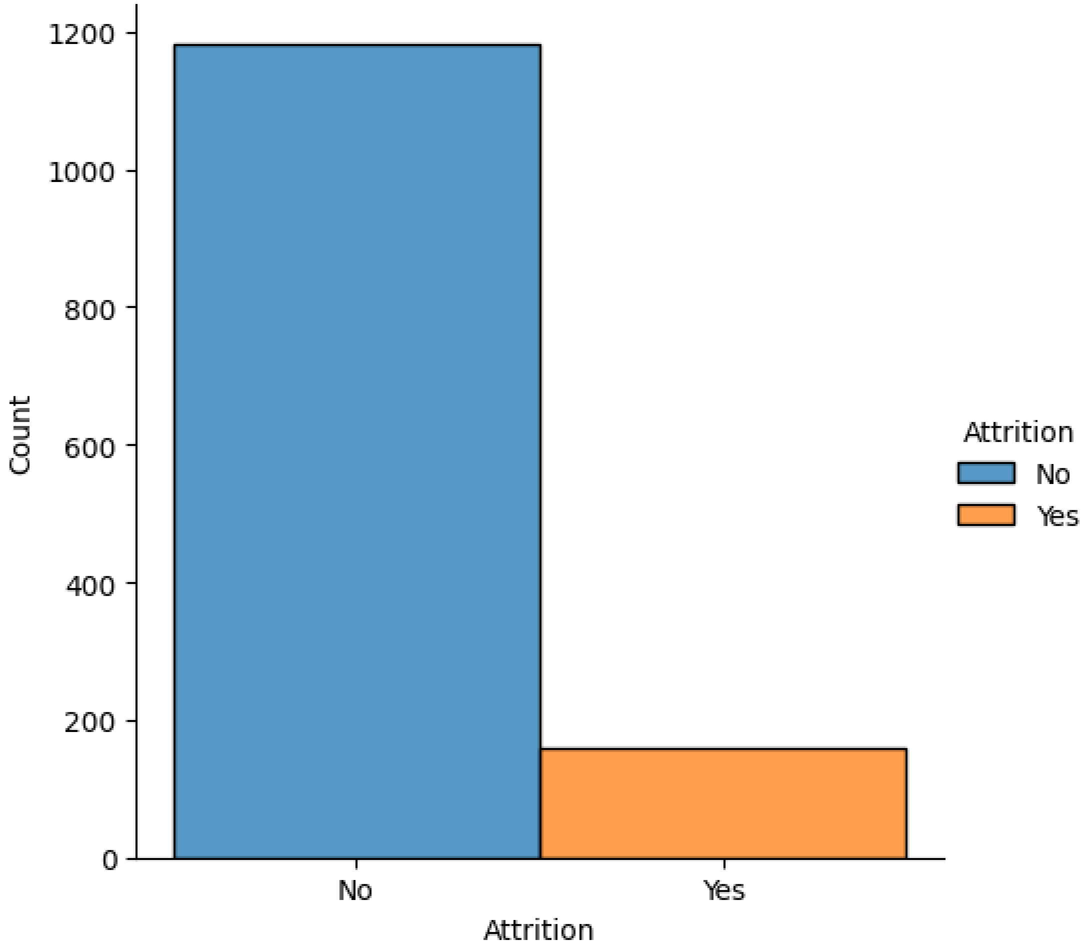
In [ ]:

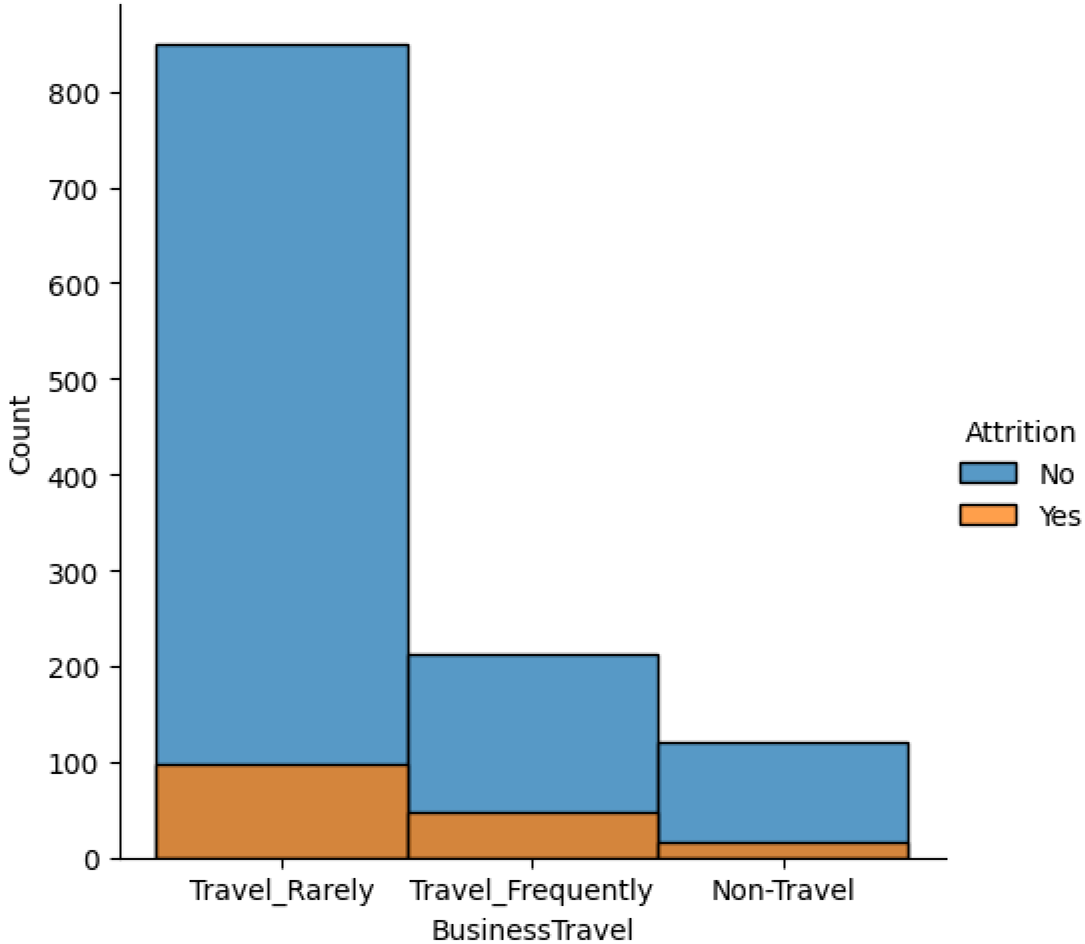
```
### Initial dataset visualization with histograms
for c in data.columns:
    sns.FacetGrid(data,
                  hue="Attrition",
                  height= 5).map(sns.histplot,c).add_legend()
```

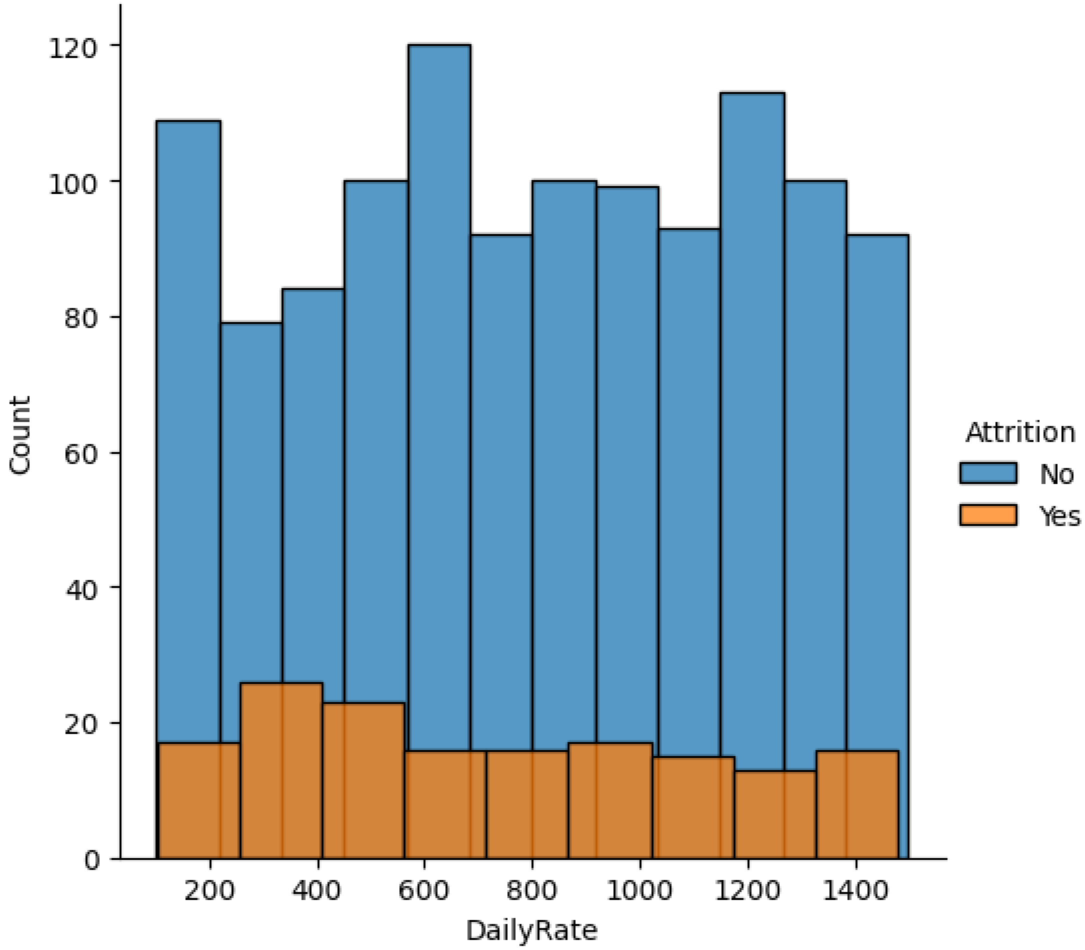




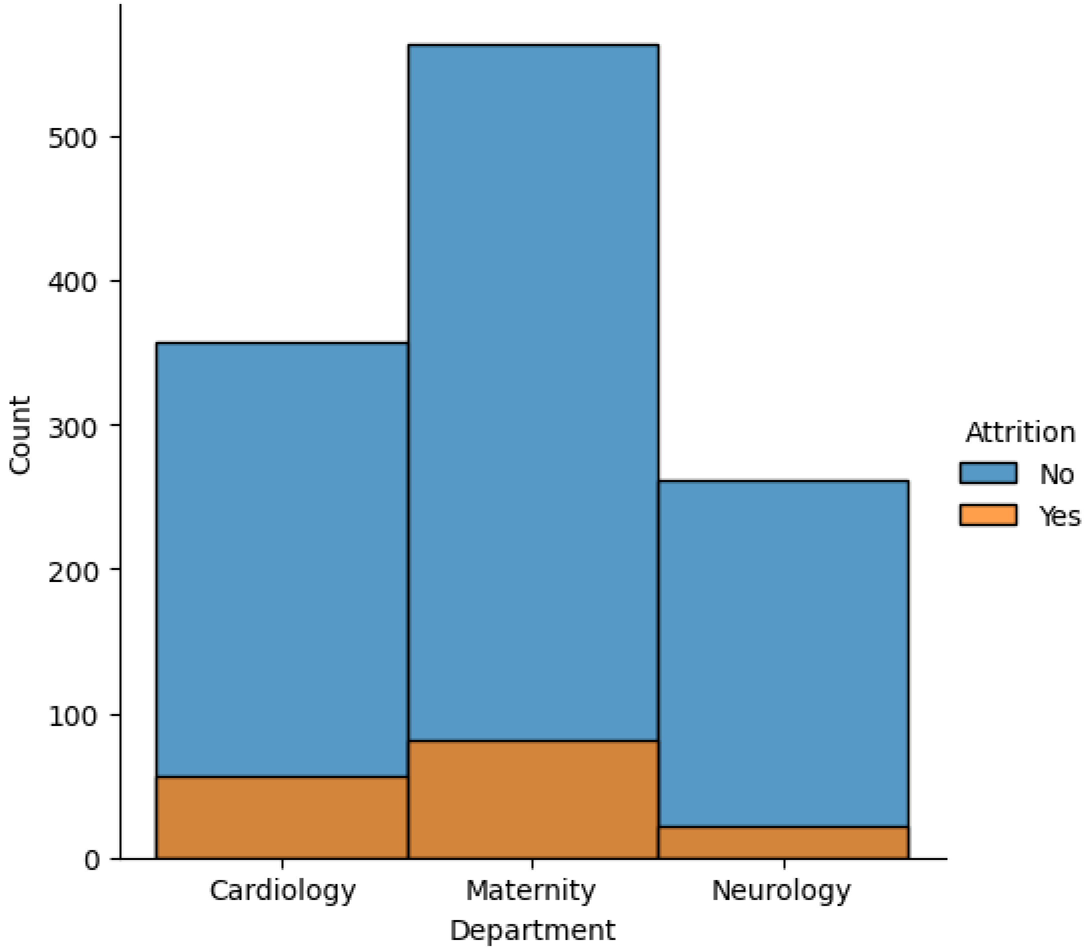
120
100
80
60
40
20
0
Count
20
30
40
50
60
Age
Attrition
No
Yes

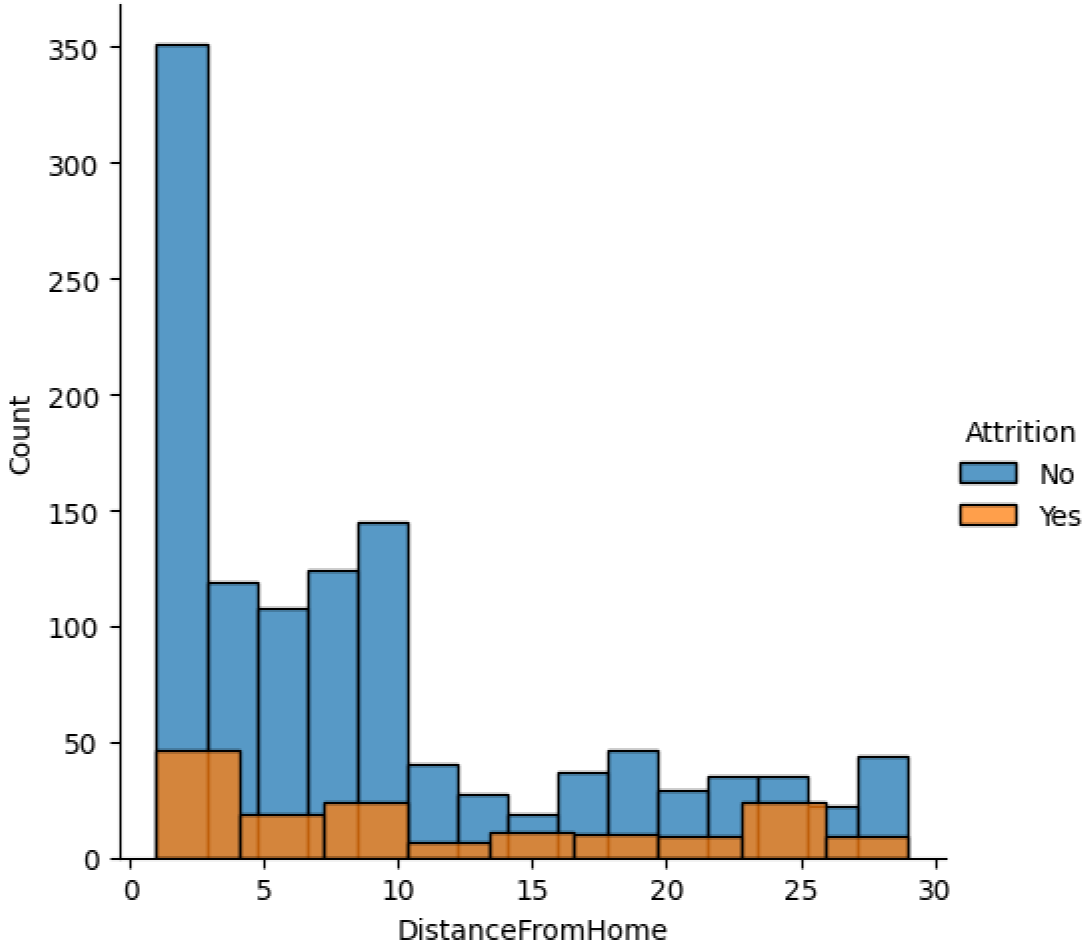

1200
1000
800
600
400
200
0
Count
No
Yes
Attrition
Attrition
No
Yes

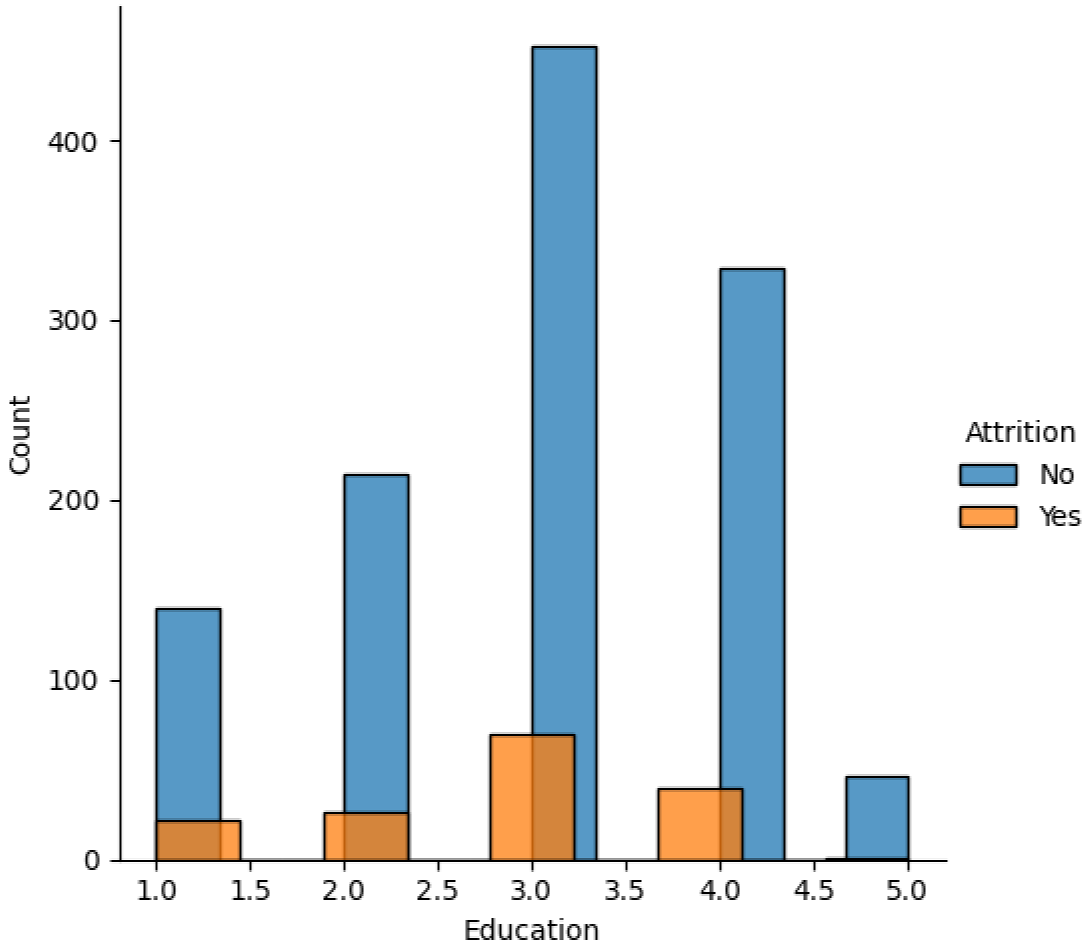

Count
800
700
600
500
400
300
200
100
0
Travel_Rarely
Travel_Frequently
Non-Travel
BusinessTravel
Attrition
No
Yes

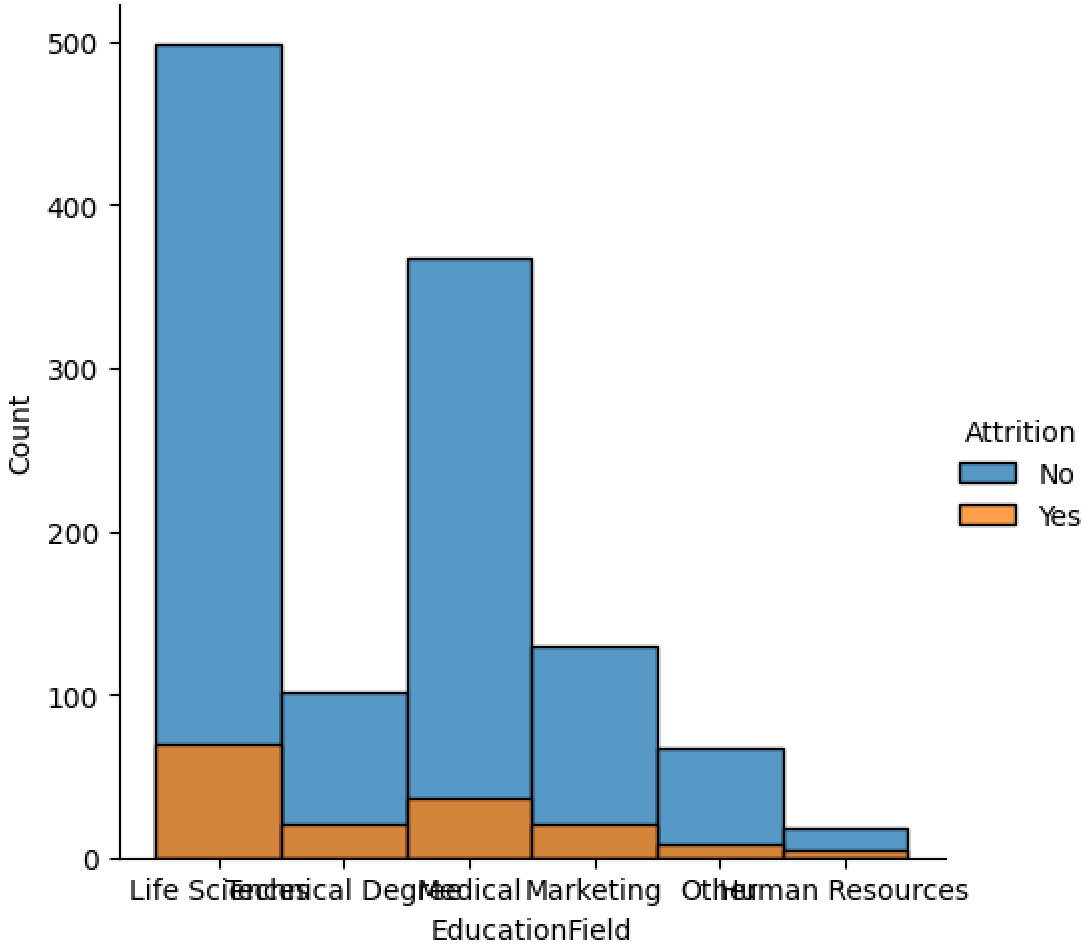


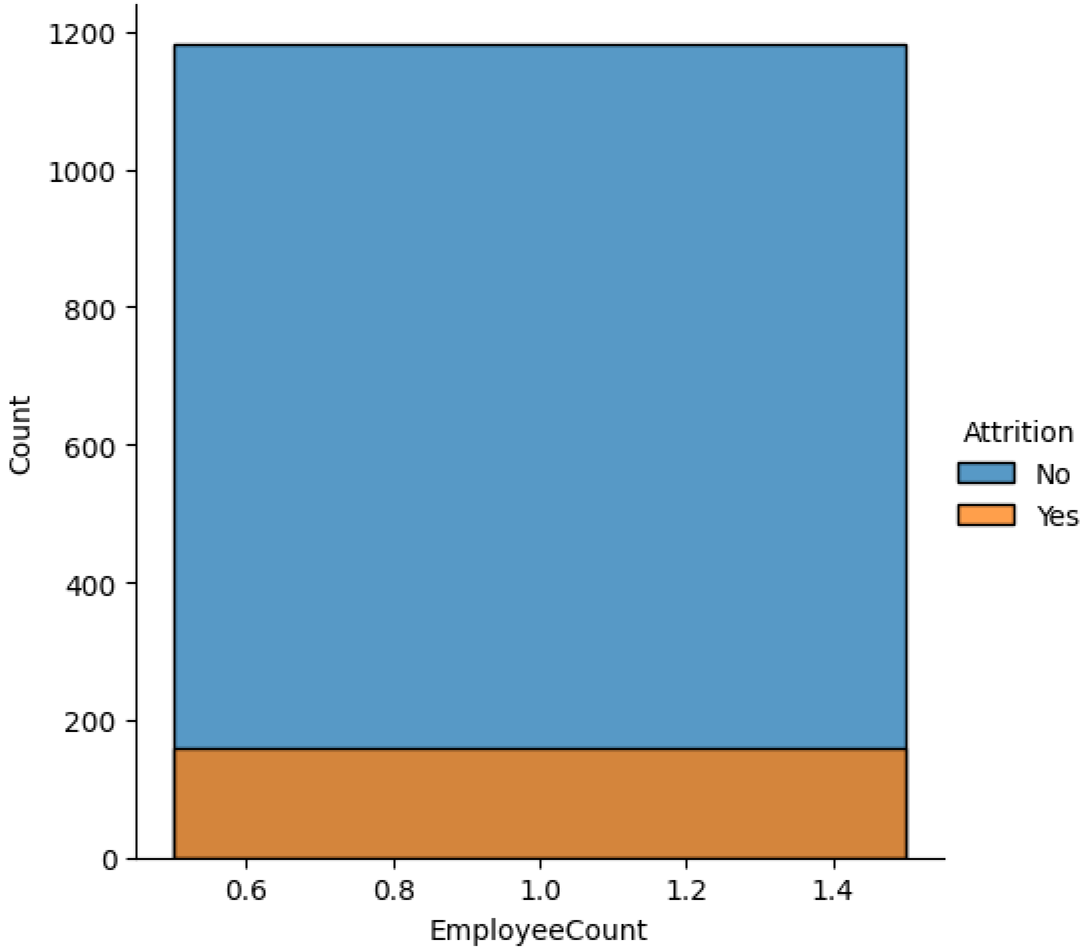
120
100
80
60
40
20
0
Count
200
400
600
800
1000
1200
1400
DailyRate
Attrition
No
Yes


0
100
200
300
400
500
Count
Cardiology
Maternity
Neurology
Department
Attrition
No
Yes

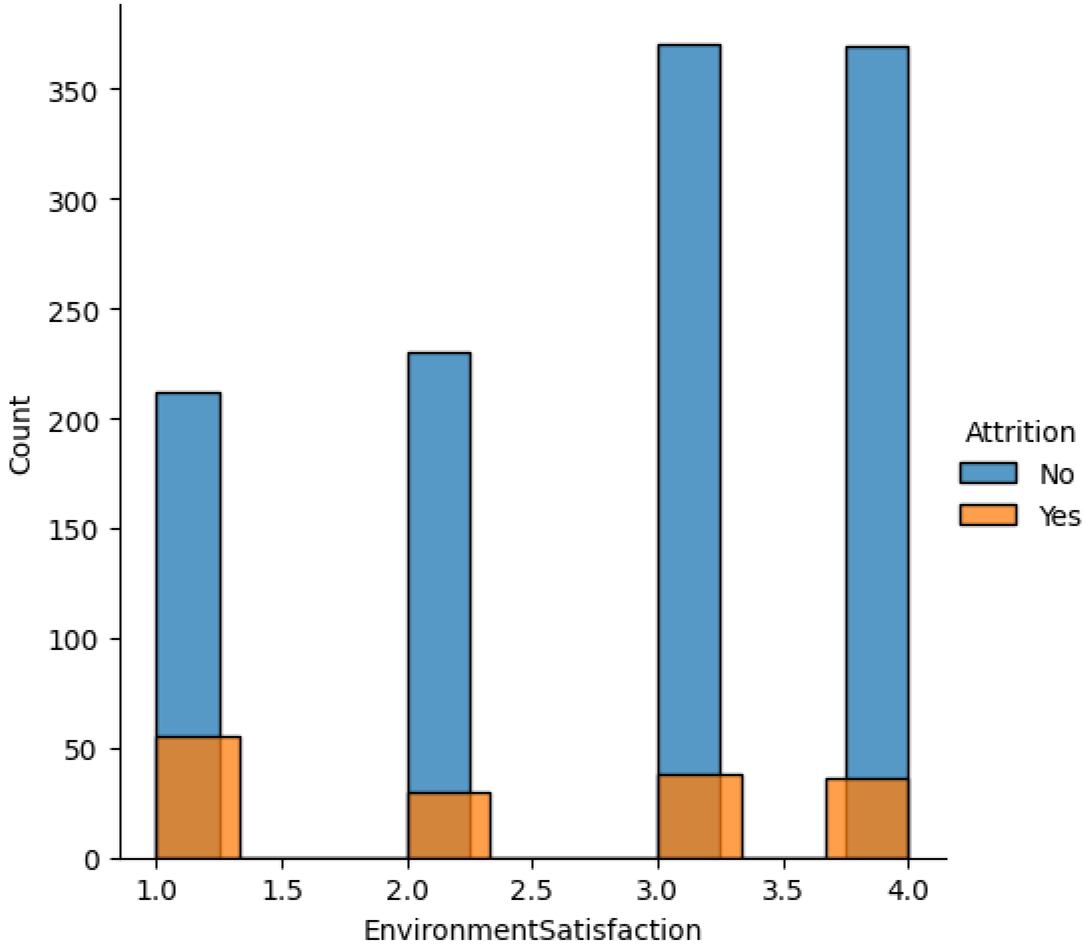

Count
350
300
250
200
150
100
50
0
0
5
10
15
20
25
30
DistanceFromHome
Attrition
No
Yes

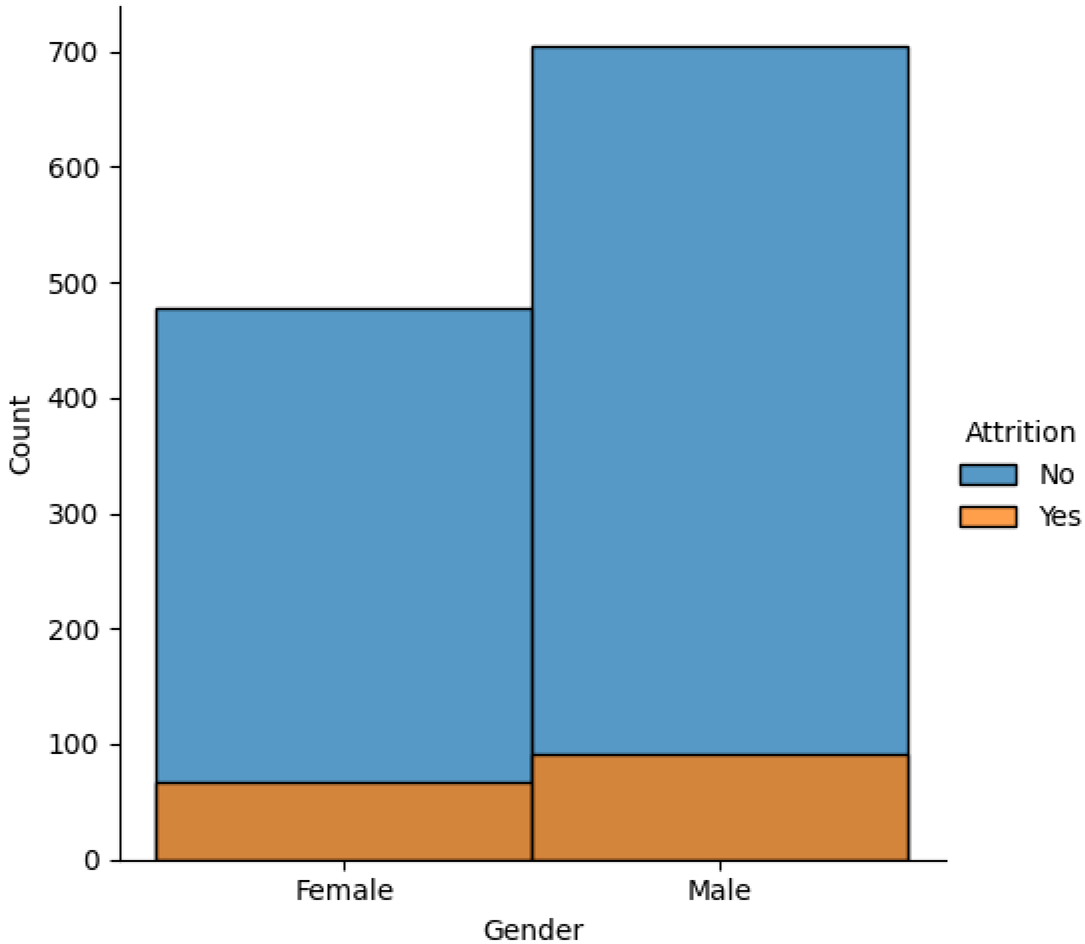

Attrition
No
Yes
Count
Education
0
100
200
300
400
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

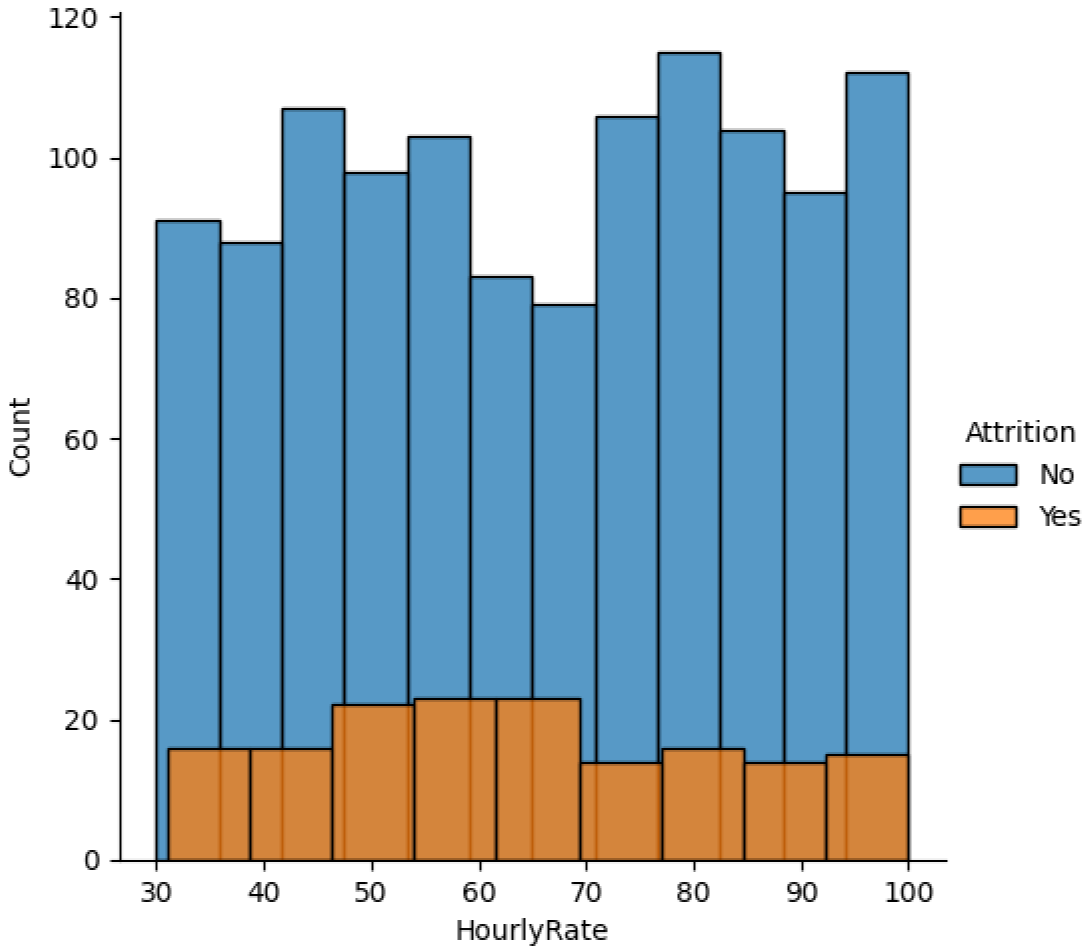

500
400
300
200
100
0
Count
Attrition
No
Yes
Life Sciences
Technical Degree
Medical
Marketing
Other
Human Resources
EducationField



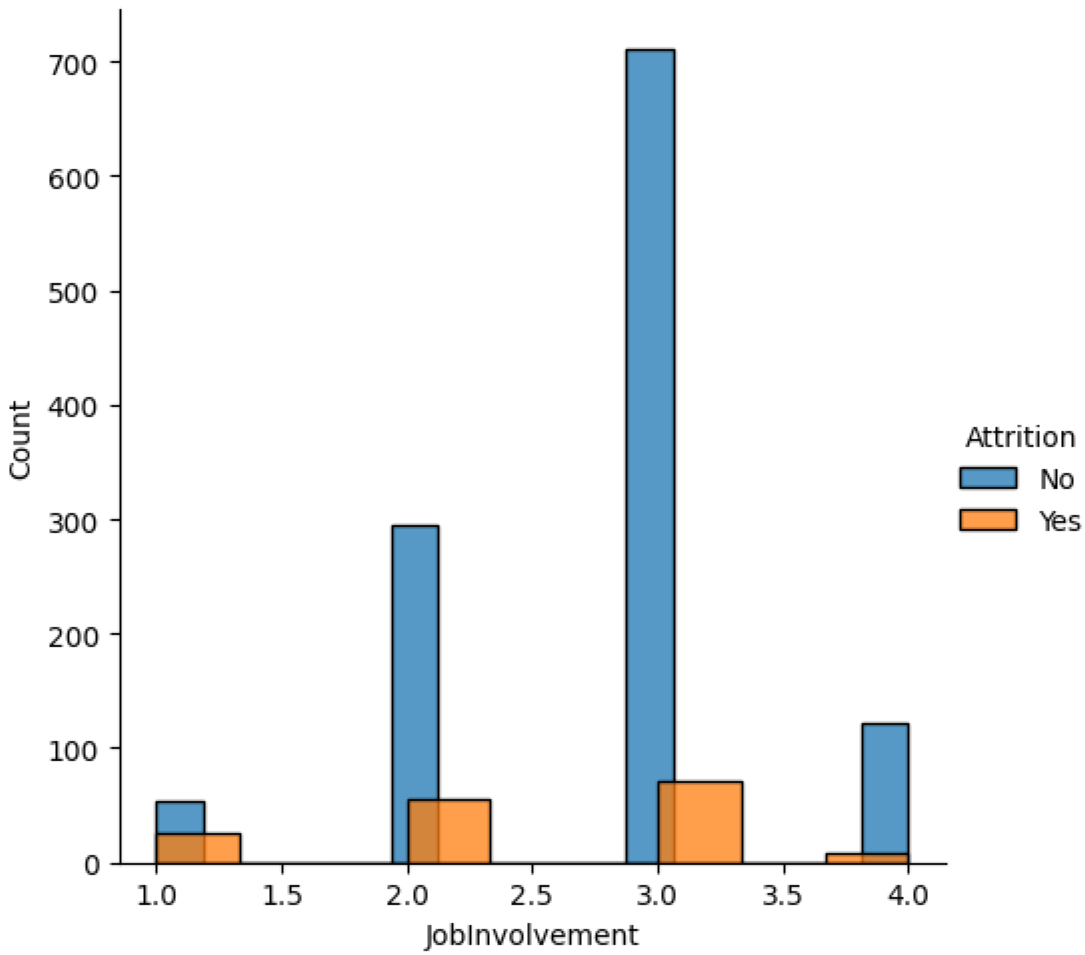
Count
1200
1000
800
600
400
200
0
0.6
0.8
1.0
1.2
1.4
EmployeeCount
Attrition
No
Yes

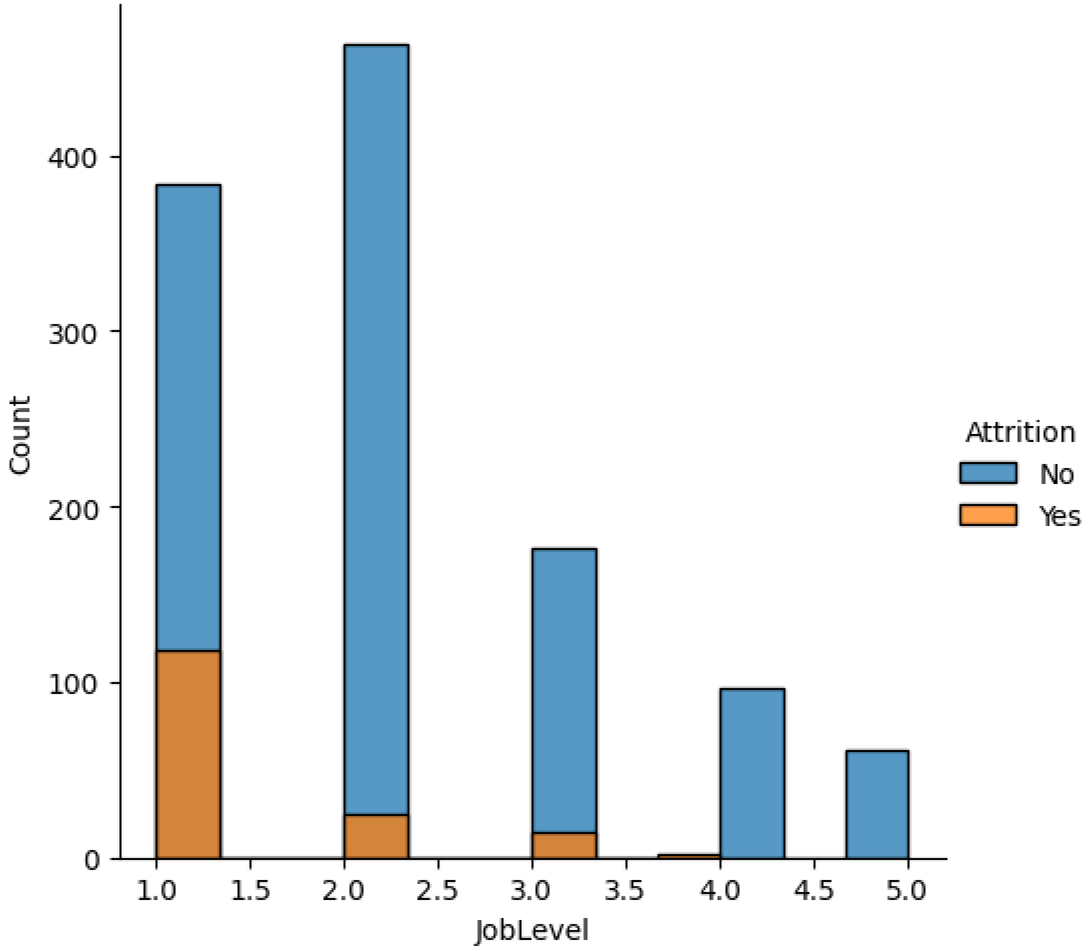

350
300
250
200
150
100
50
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
EnvironmentSatisfaction
Attrition
No
Yes

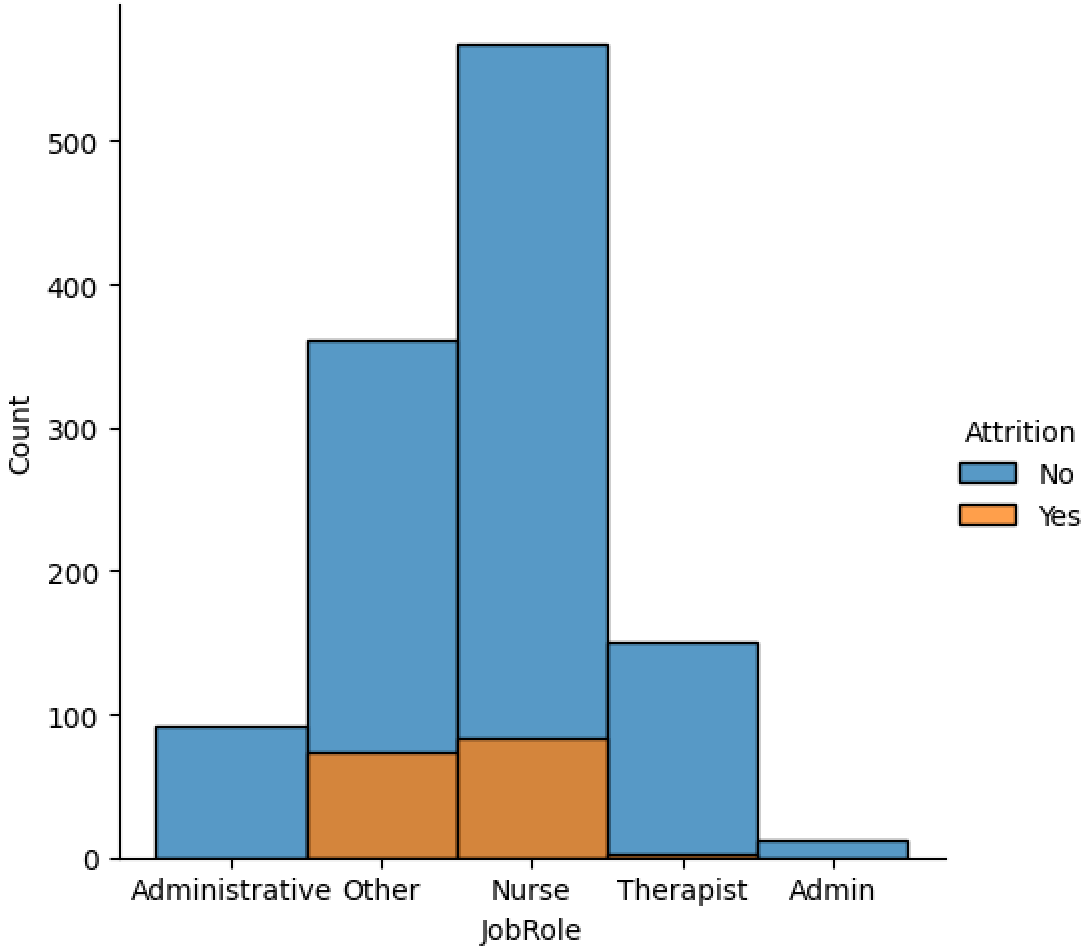

700
600
500
400
300
200
100
0
Count
Female
Male
Gender
Attrition
No
Yes

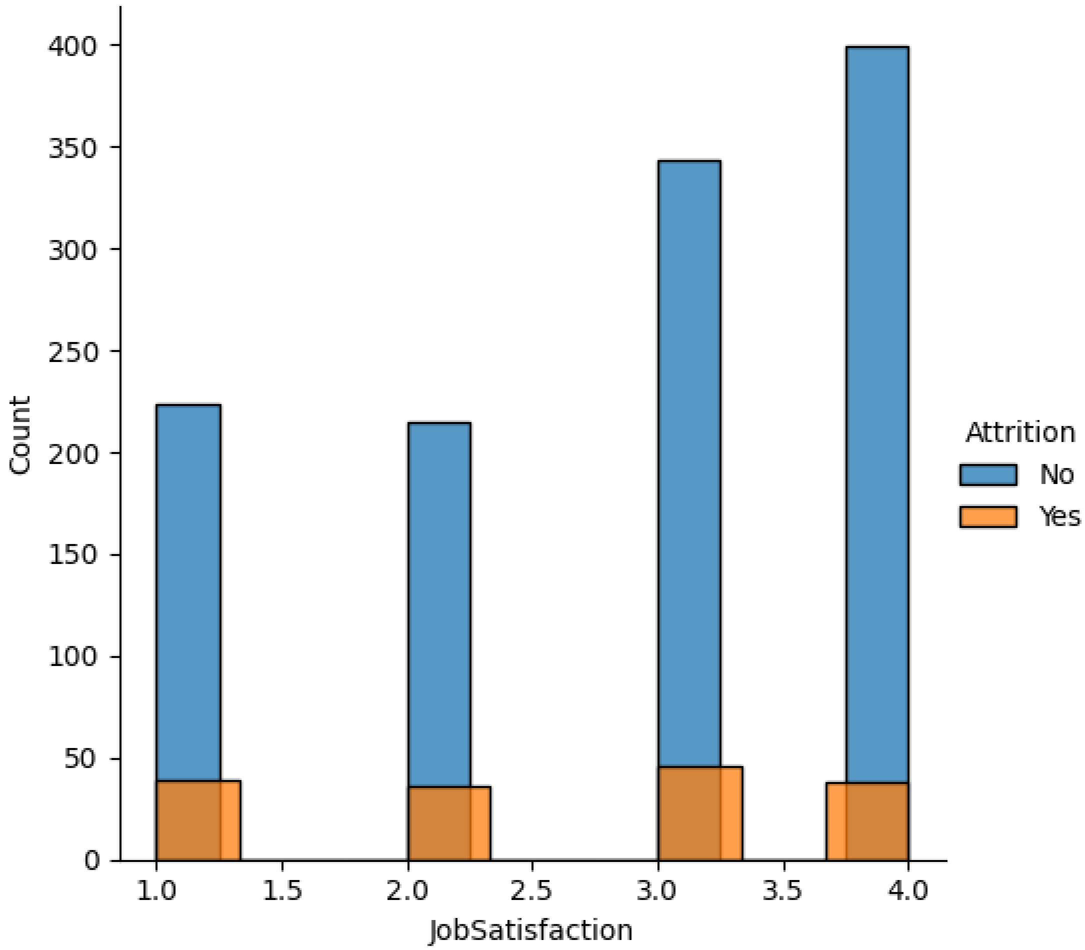


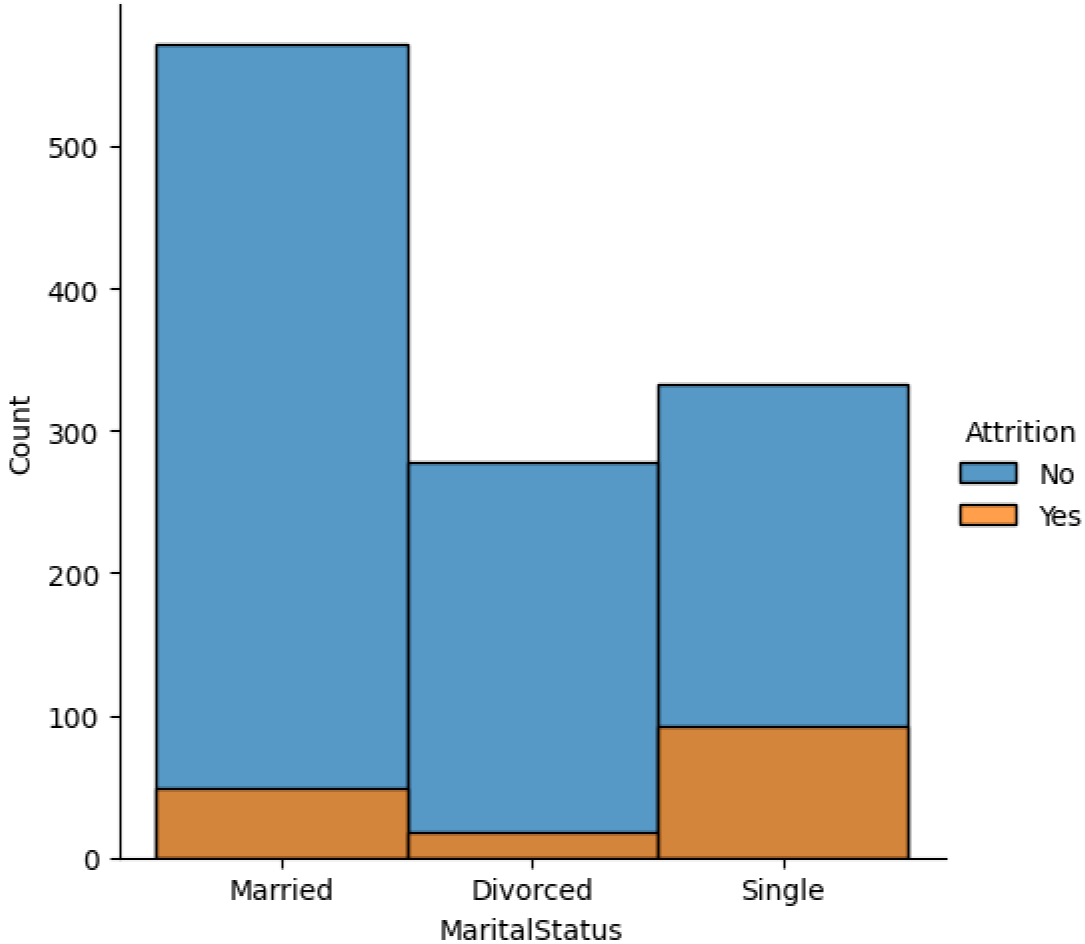
120
100
80
60
40
20
0
Count
30
40
50
60
70
80
90
100
HourlyRate
Attrition
No
Yes


700
600
500
400
300
200
100
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
JobInvolvement
Attrition
No
Yes

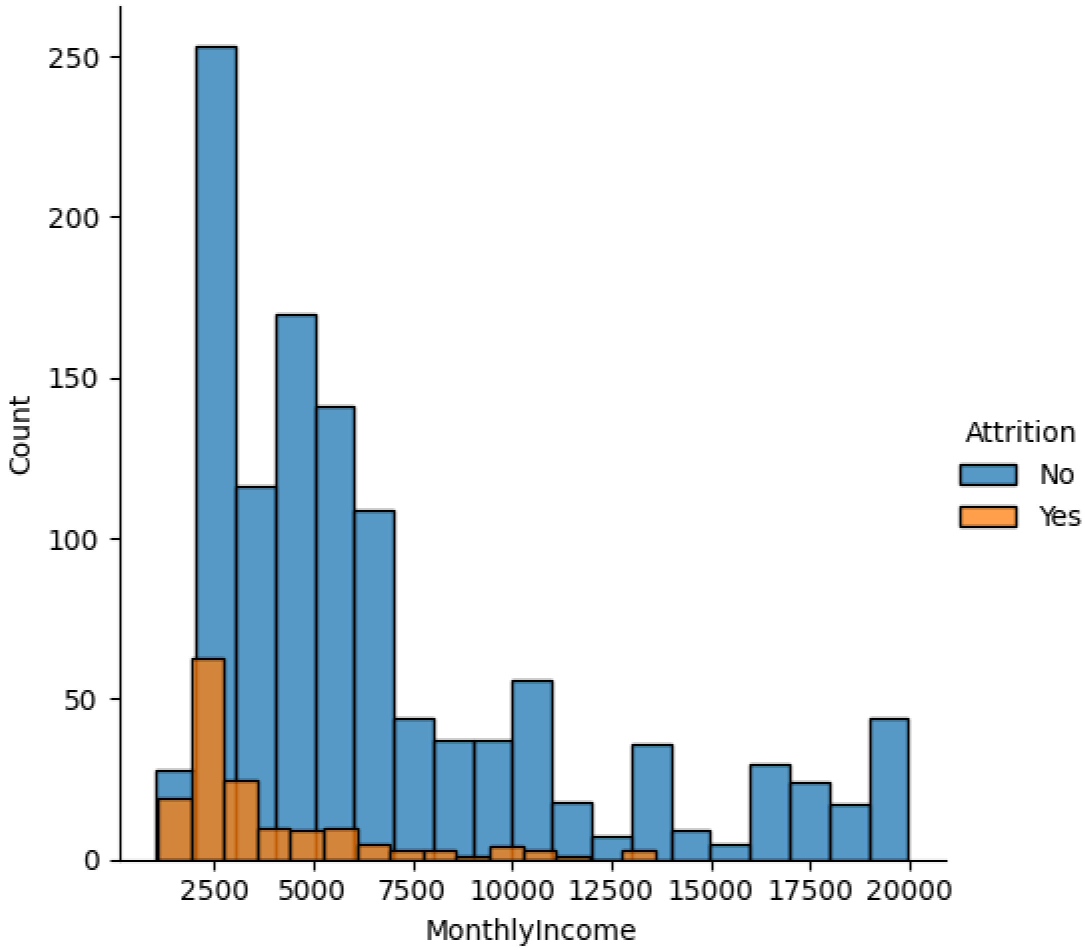

Count
JobLevel
Attrition
No
Yes
0
100
200
300
400
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

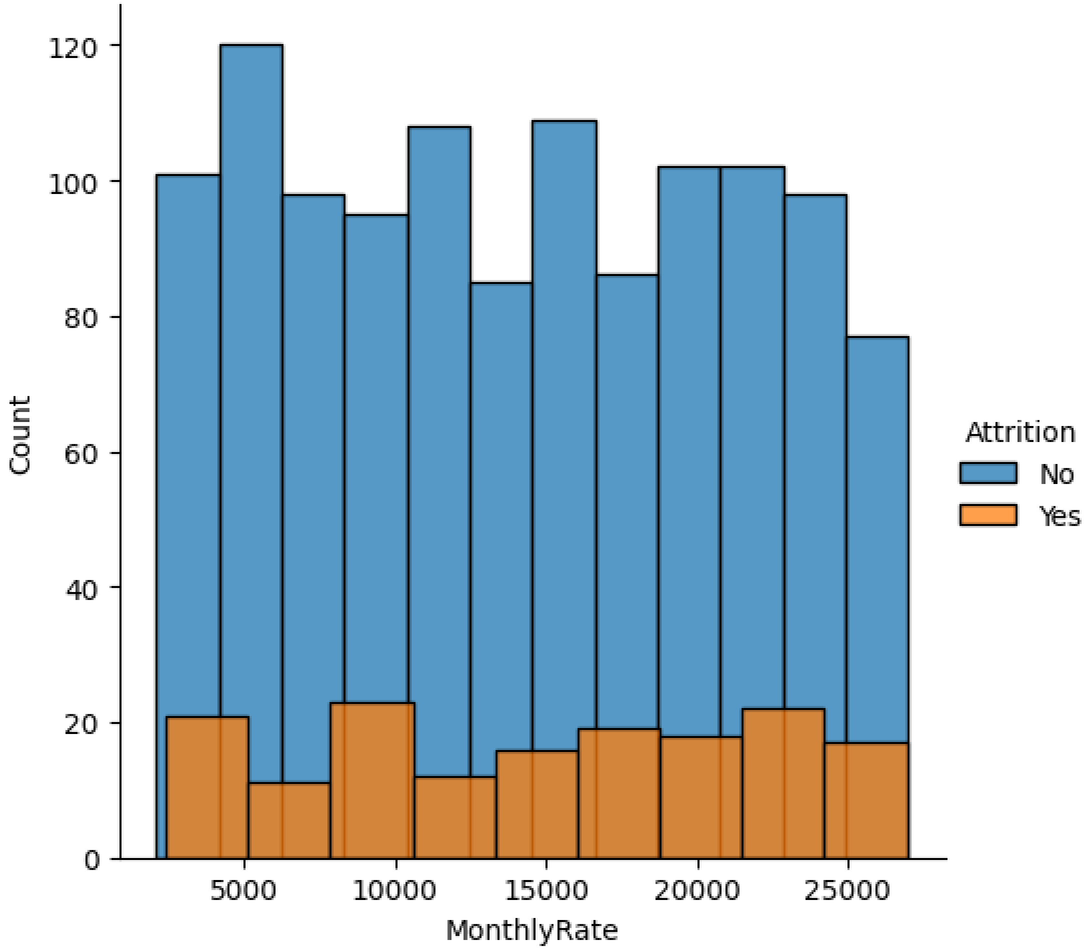

Count
500
400
300
200
100
0
Administrative
Other
Nurse
Therapist
Admin
JobRole
Attrition
No
Yes


400
350
300
250
200
150
100
50
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
JobSatisfaction
Attrition
No
Yes

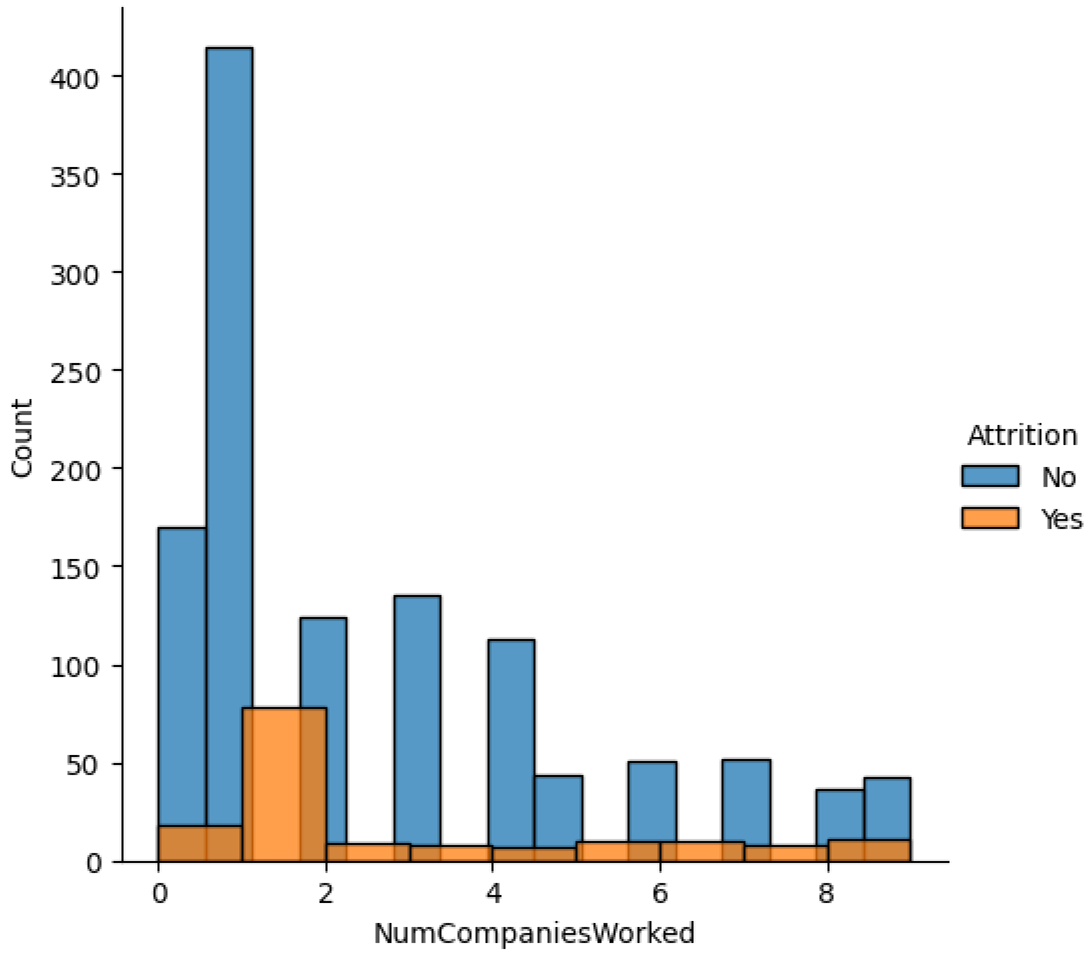


Count
500
400
300
200
100
0
Married
Divorced
Single
MaritalStatus
Attrition
No
Yes

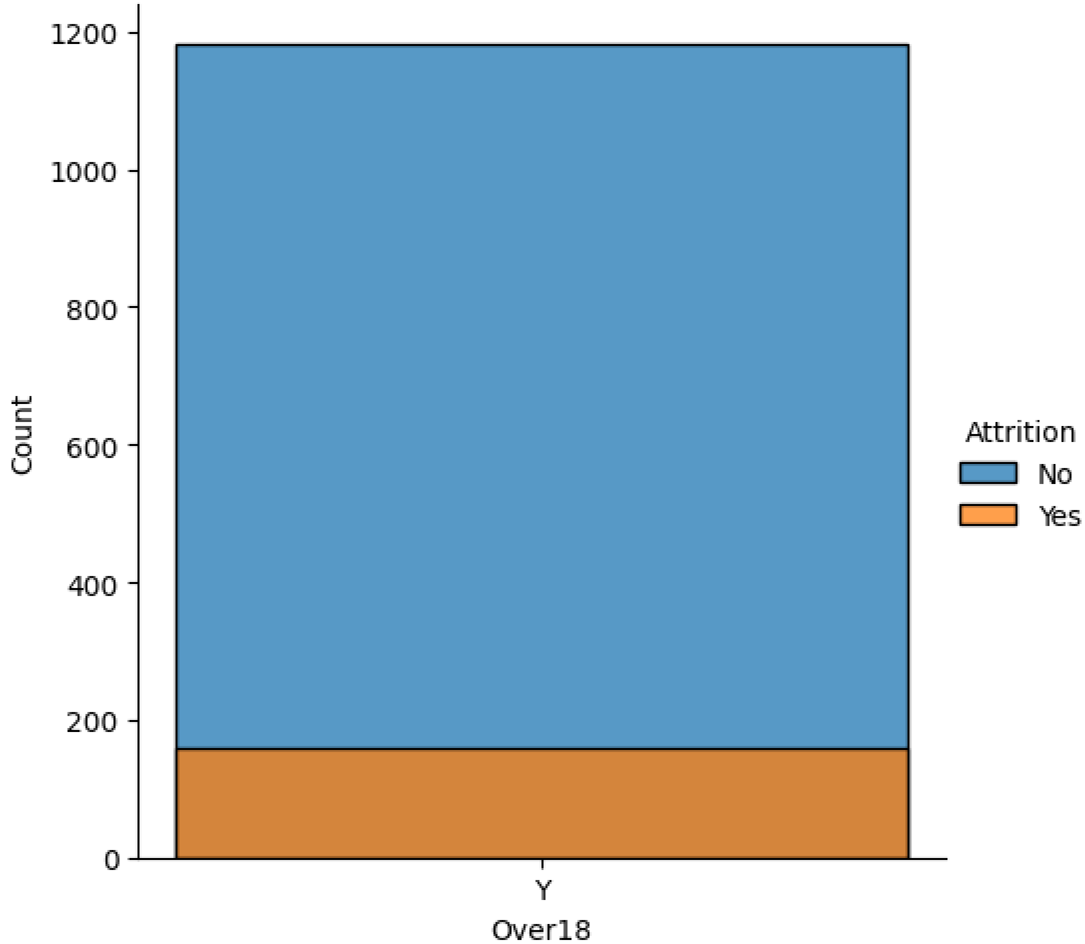

Count
250
200
150
100
50
0
2500
5000
7500
10000
12500
15000
17500
20000
MonthlyIncome
Attrition
No
Yes

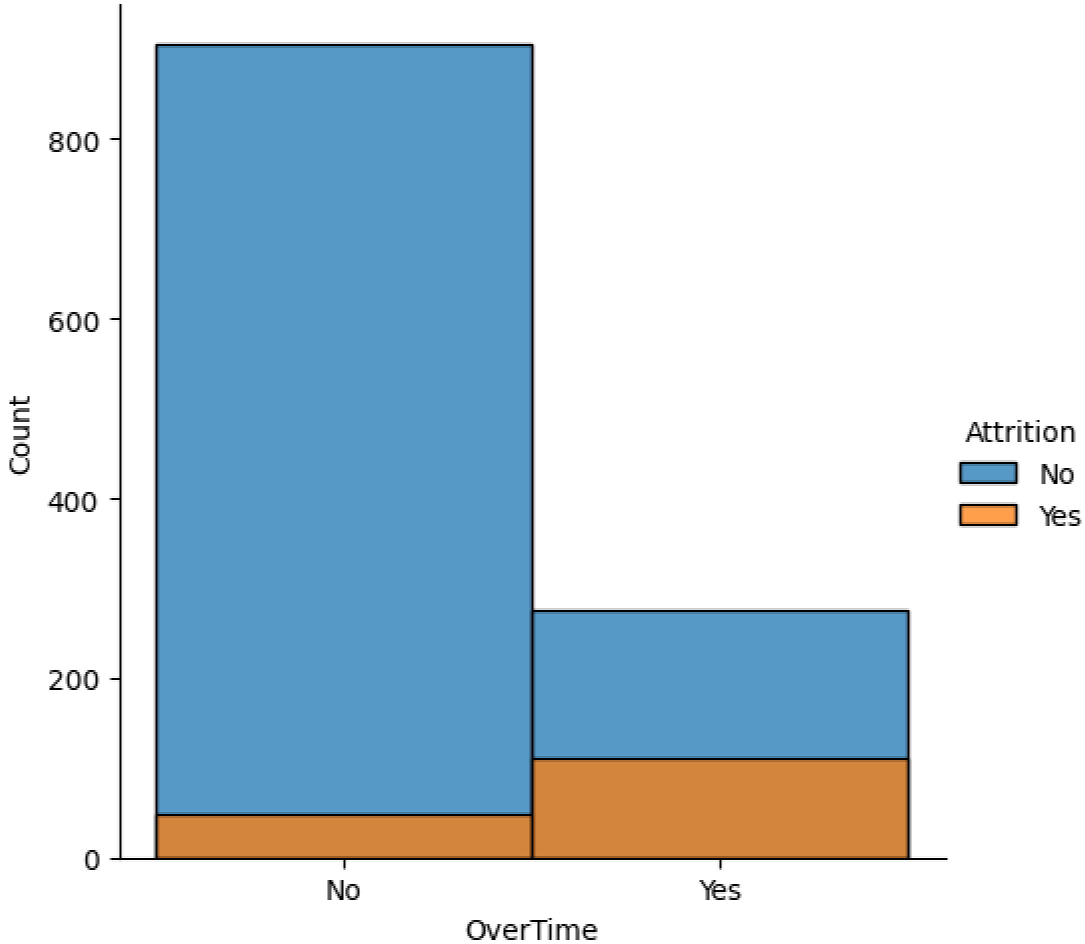


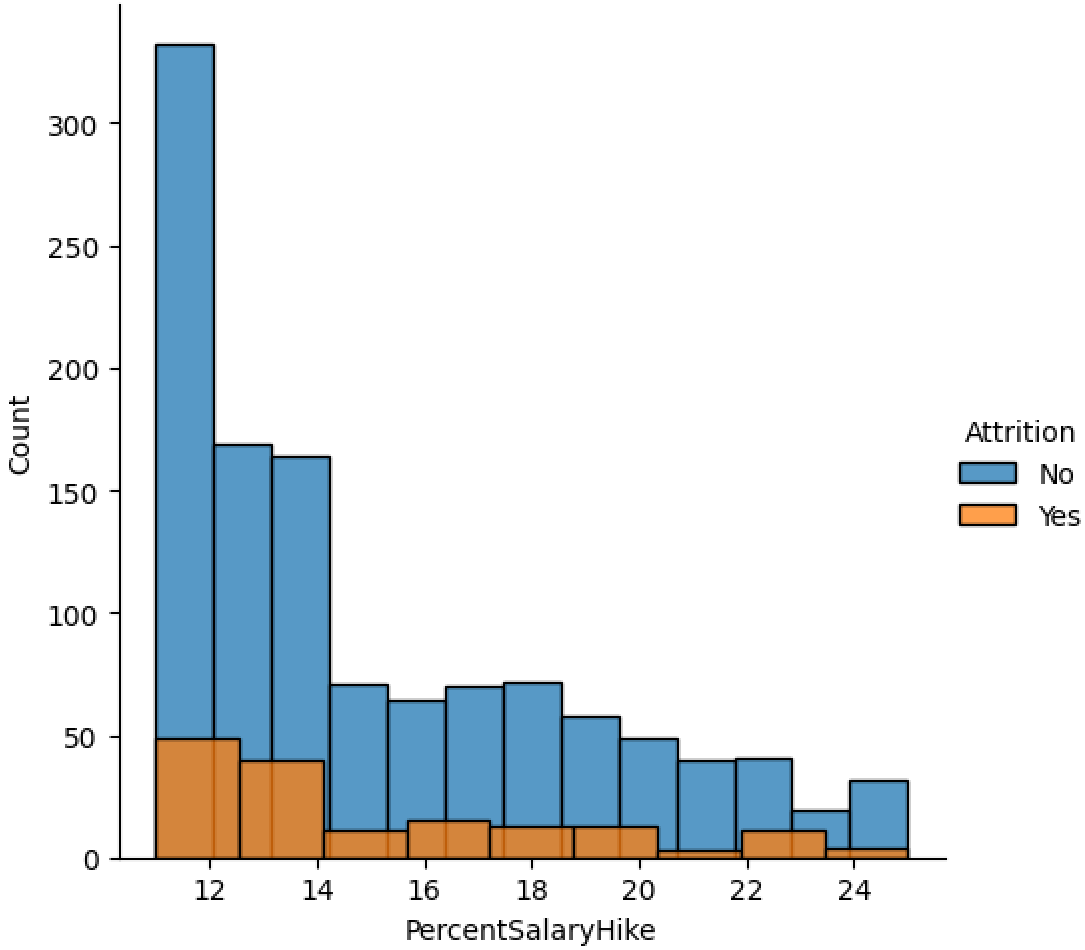
Count
120
100
80
60
40
20
0
5000
10000
15000
20000
25000
MonthlyRate
Attrition
No
Yes

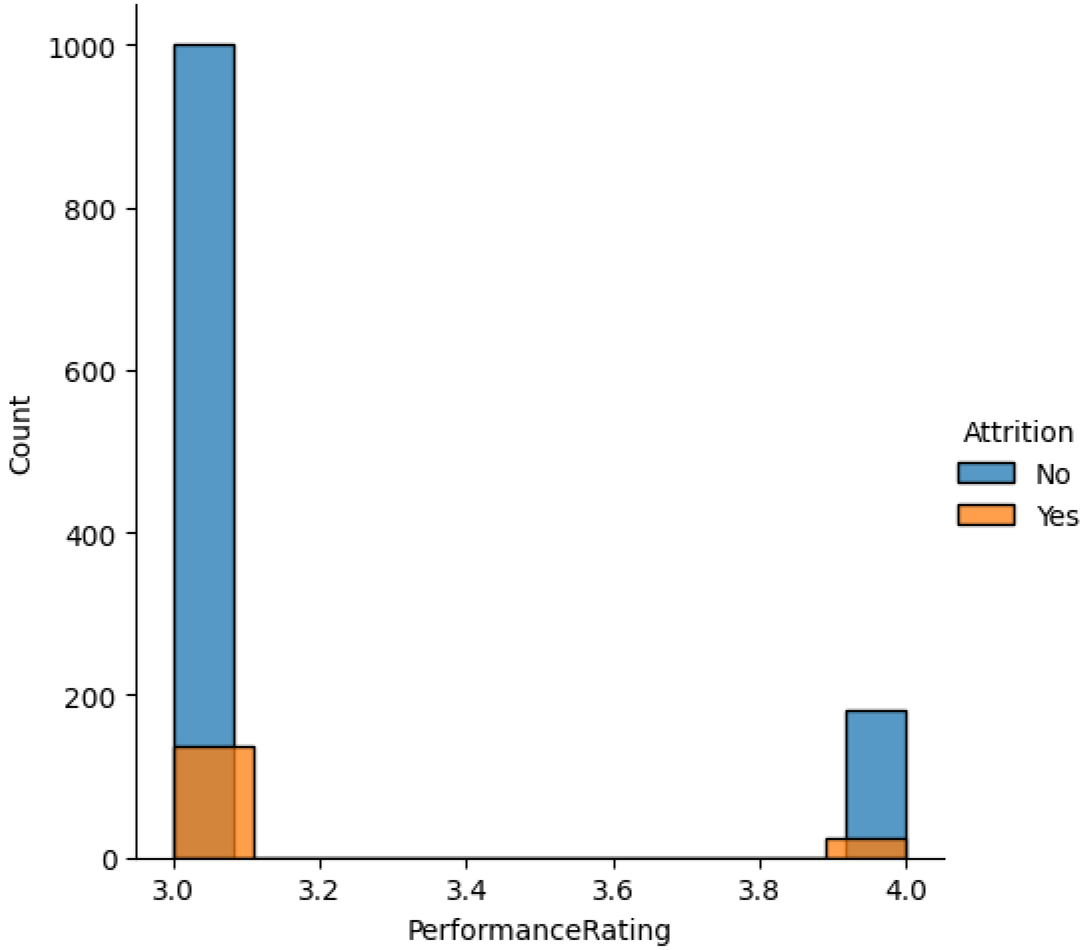


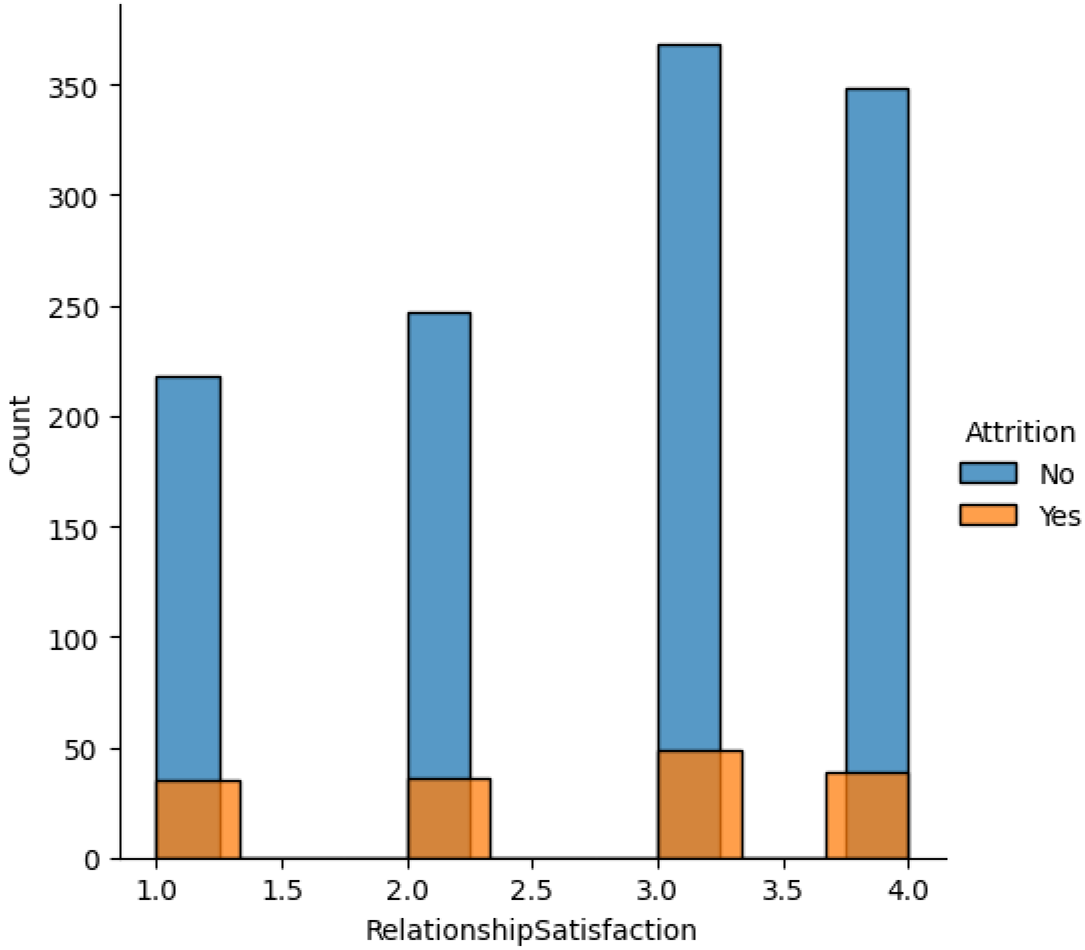
400
350
300
250
200
150
100
50
0
Count
0
2
4
6
8
NumCompaniesWorked
Attrition
No
Yes

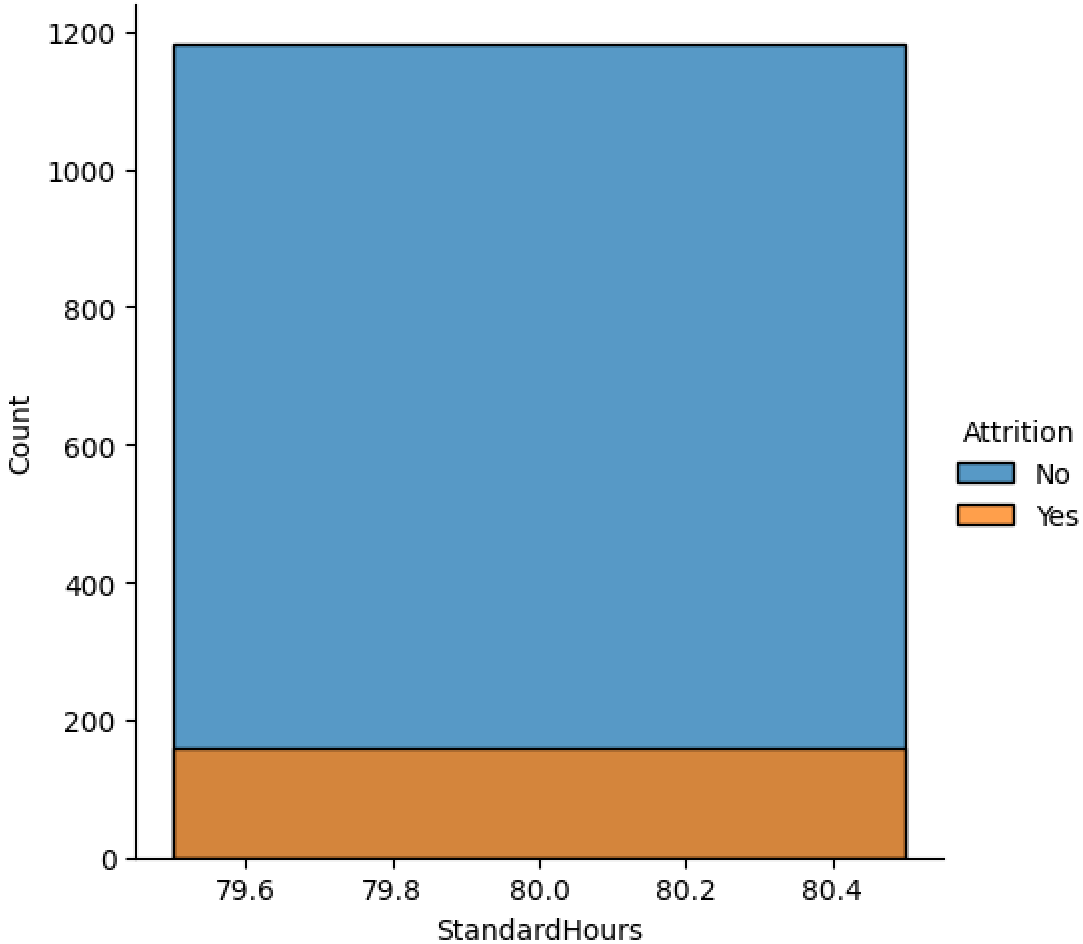

1200
1000
800
600
400
200
0
Count
Y
Over18
Attrition
No
Yes

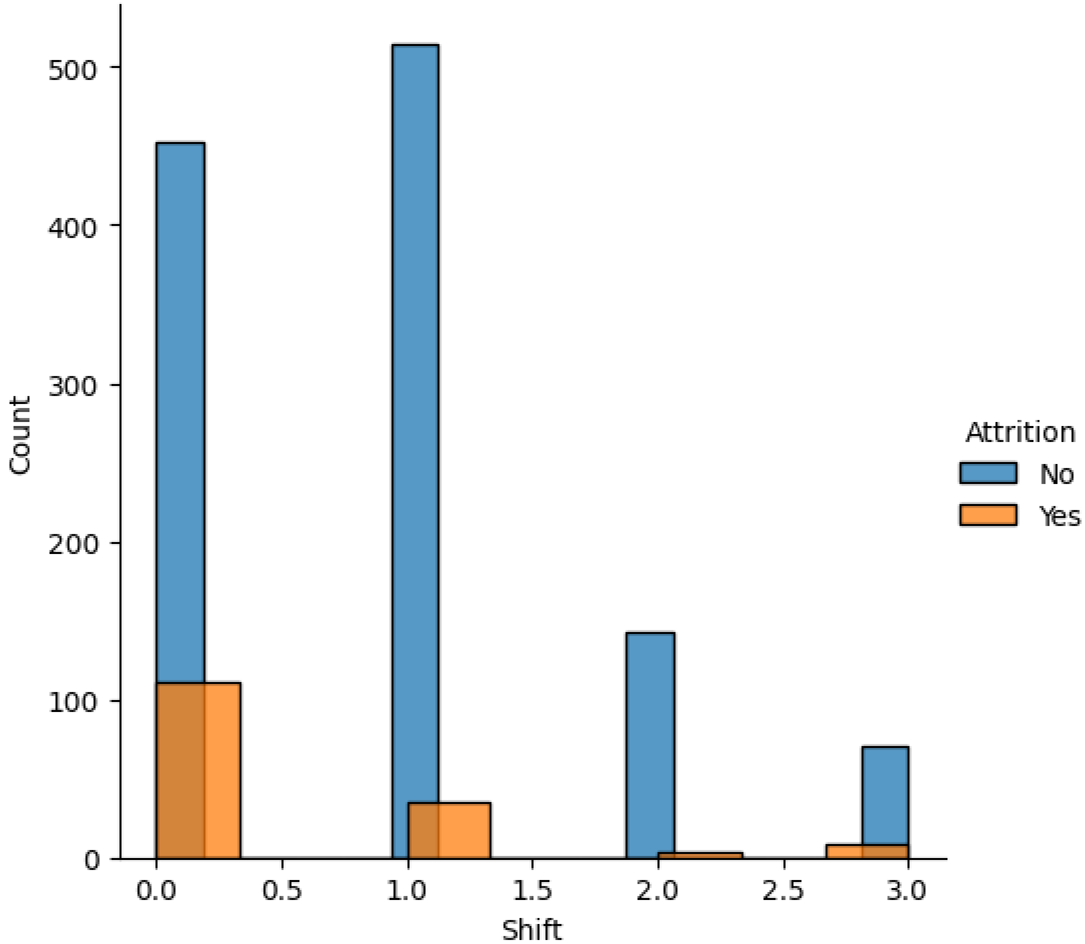

800
600
400
200
0
Count
No
Yes
OverTime
Attrition
No
Yes


300
250
200
150
100
50
0
Count
12
14
16
18
20
22
24
PercentSalaryHike
Attrition
No
Yes

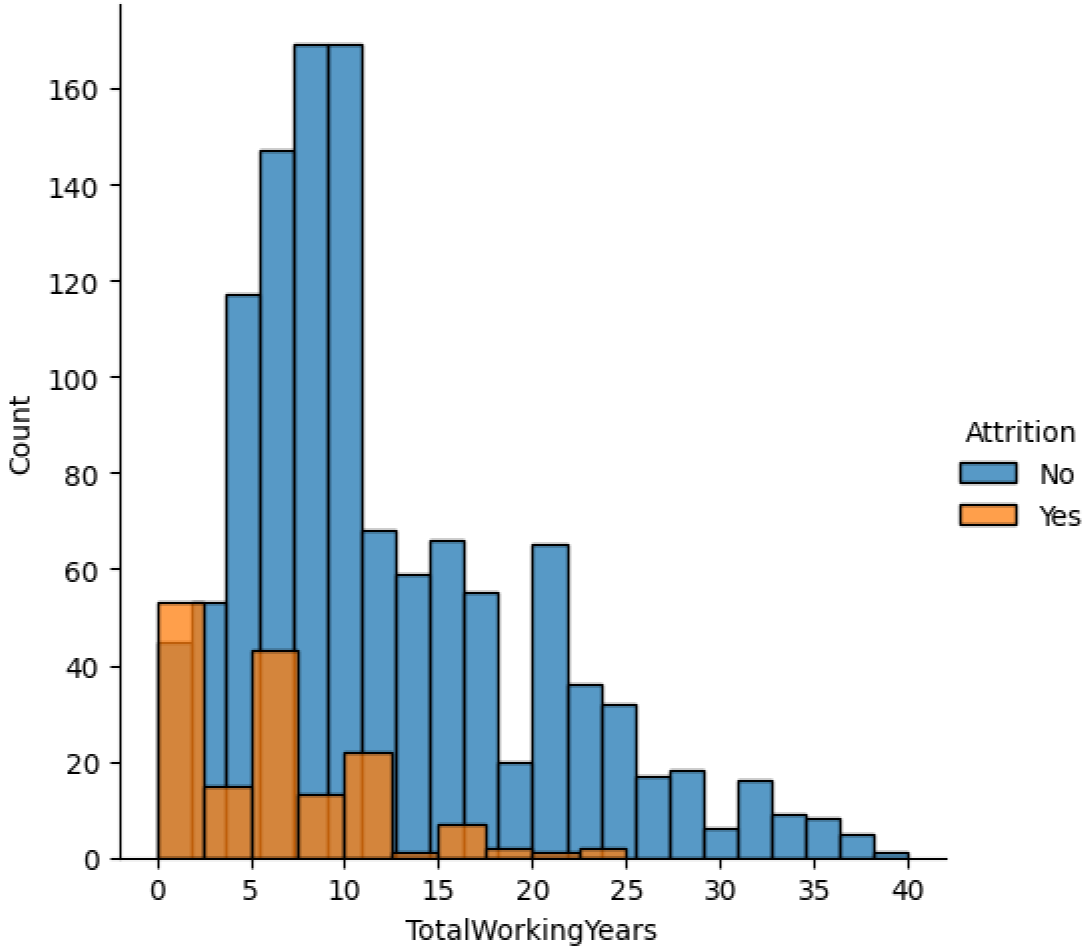

Count
1000
800
600
400
200
0
3.0
3.2
3.4
3.6
3.8
4.0
PerformanceRating
Attrition
No
Yes

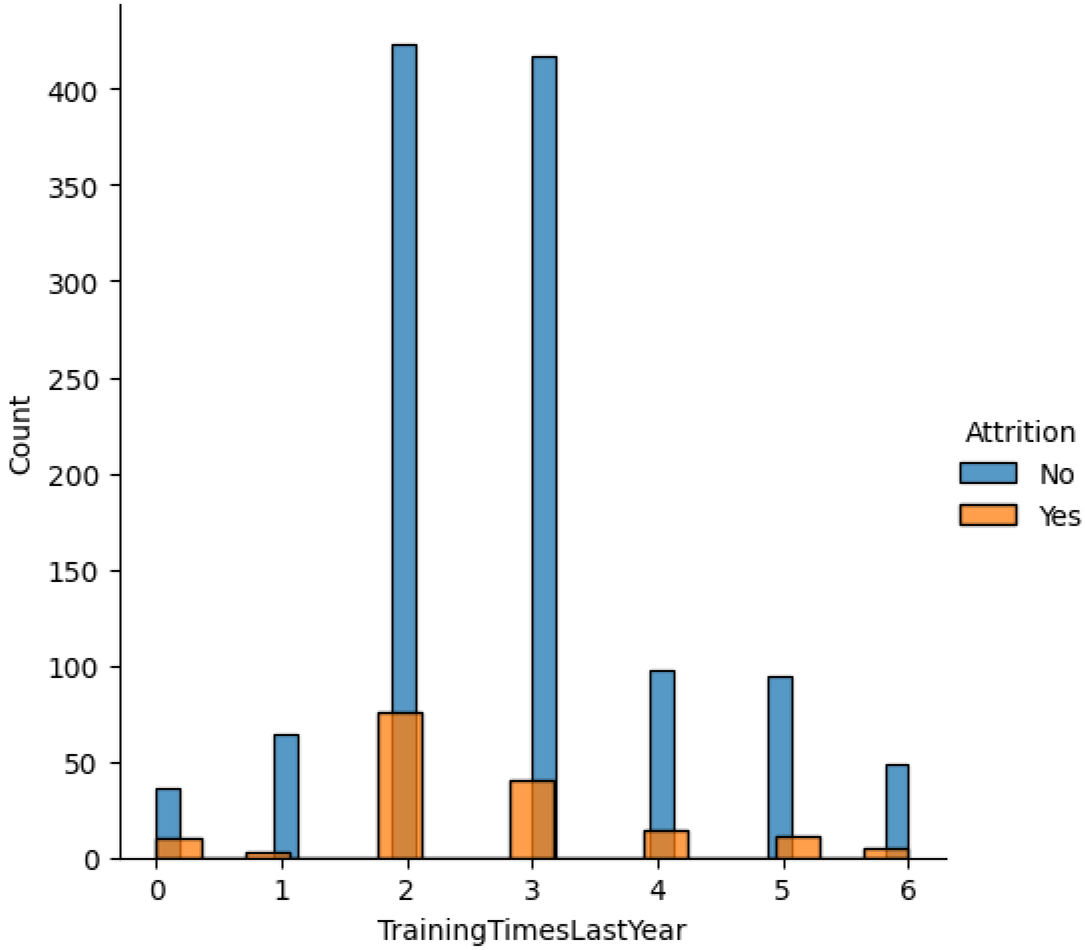

350
300
250
200
150
100
50
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
RelationshipSatisfaction
Attrition
No
Yes


1200
1000
800
600
400
200
0
Count
79.6
79.8
80.0
80.2
80.4
StandardHours
Attrition
No
Yes

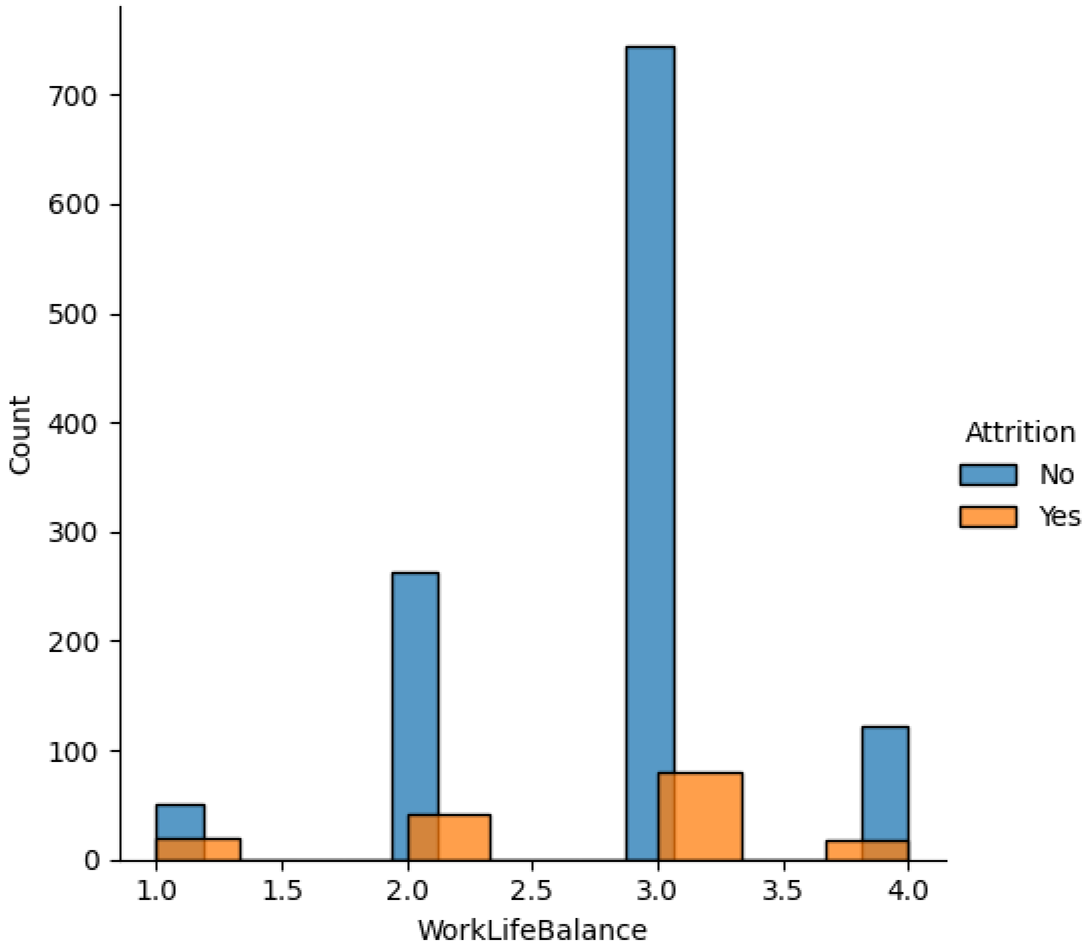


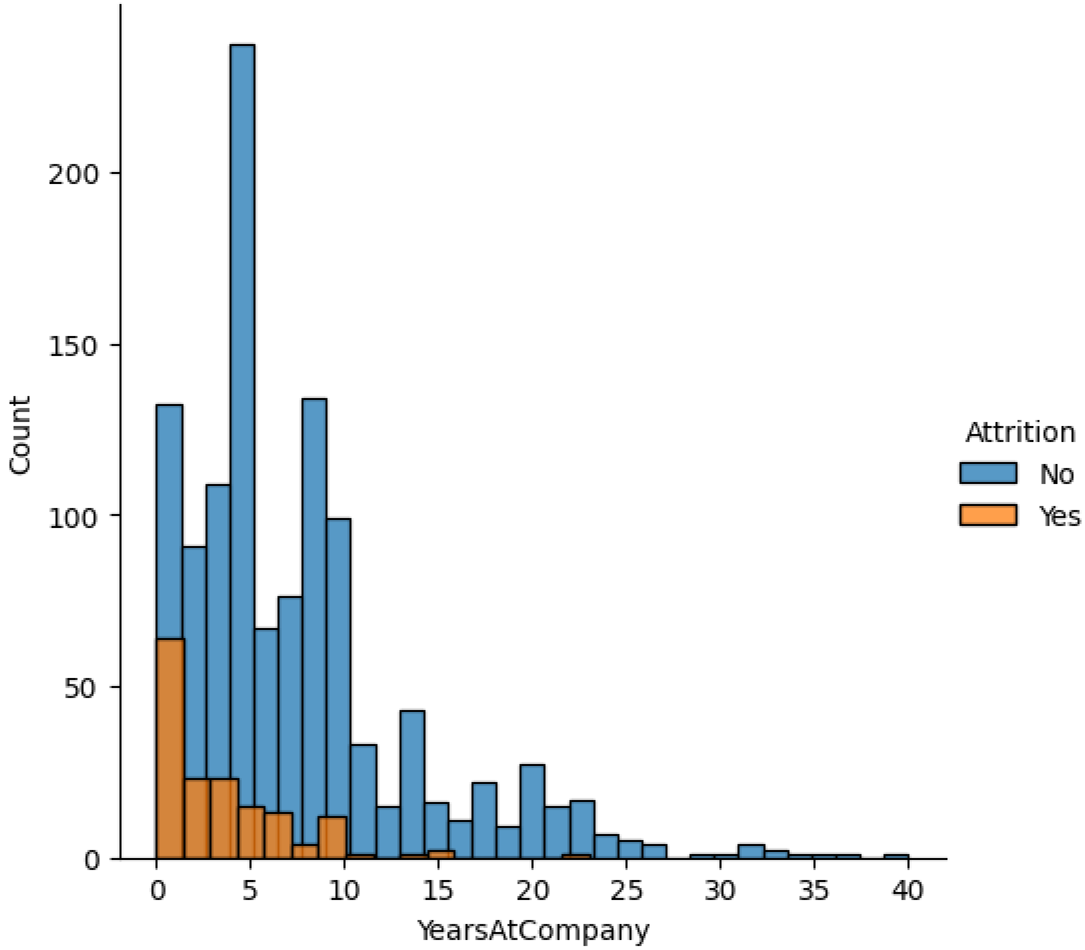
Count
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Shift
Attrition
No
Yes

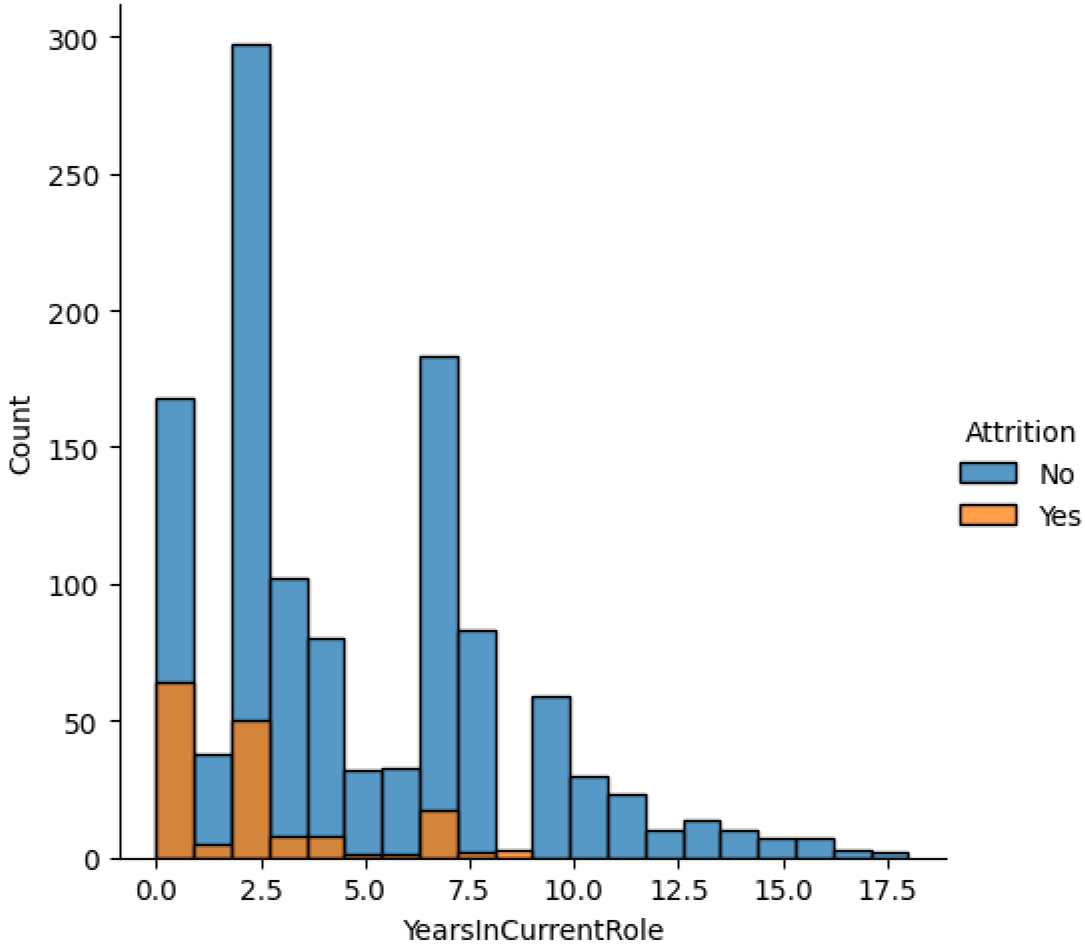

Count
0
20
40
60
80
100
120
140
160
0
5
10
15
20
25
30
35
40
TotalWorkingYears
Attrition
No
Yes



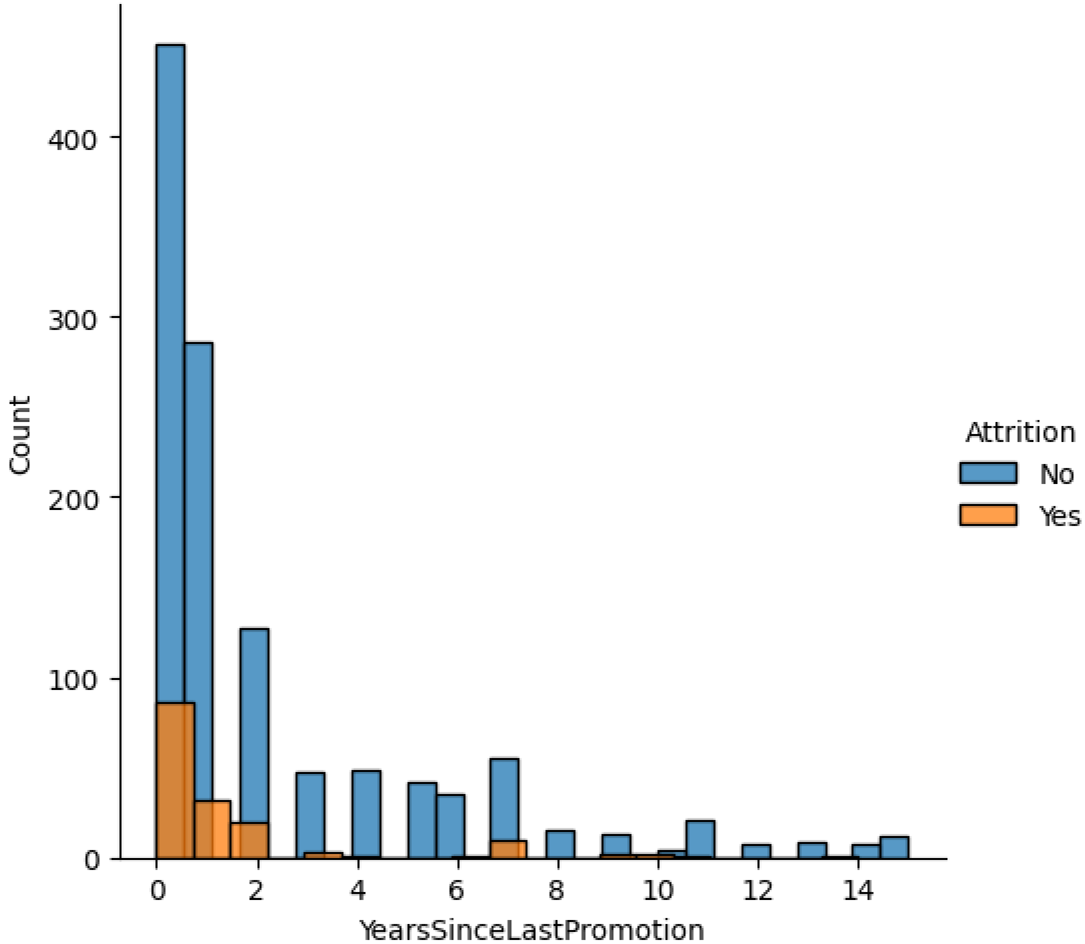
Attrition
No
Yes
Count
TrainingTimesLastYear
0
50
100
150
200
250
300
350
400
0
1
2
3
4
5
6


700
600
500
400
300
200
100
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
WorkLifeBalance
Attrition
No
Yes

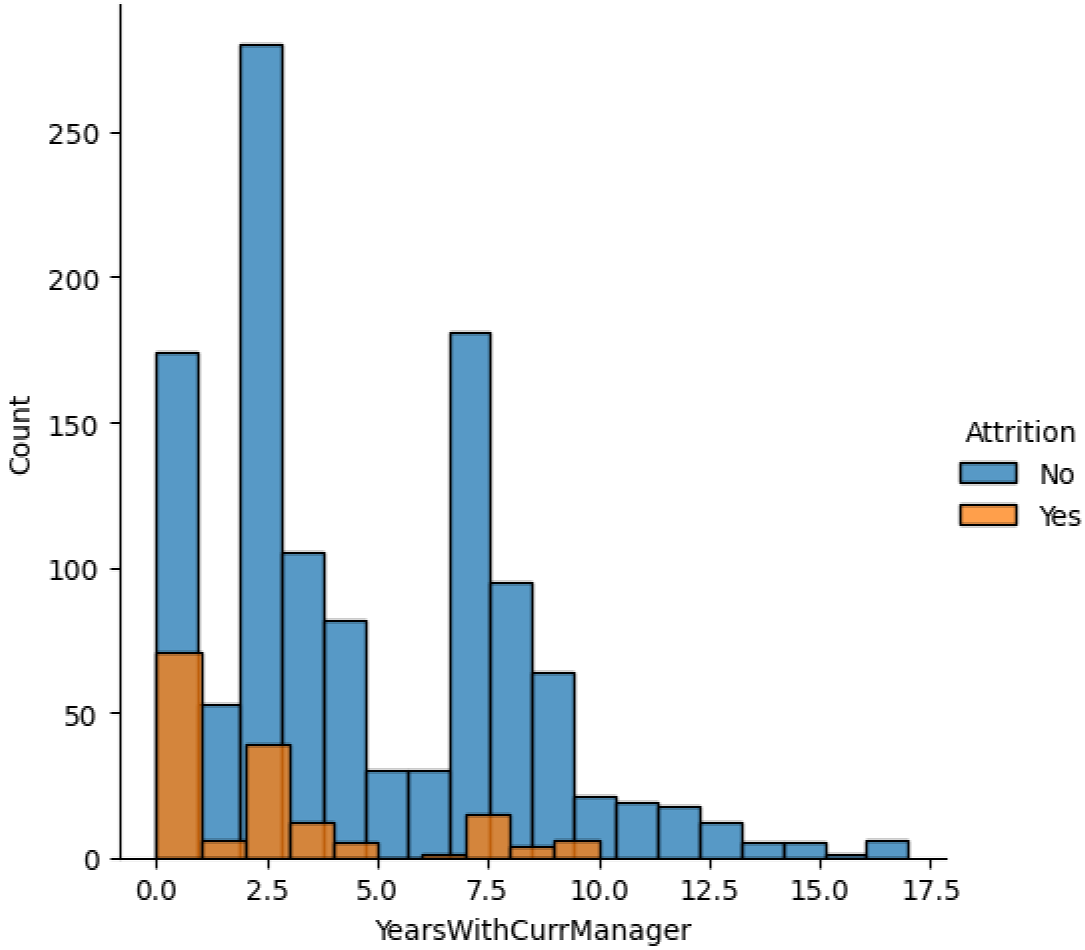

Count
200
150
100
50
0
0
5
10
15
20
25
30
35
40
YearsAtCompany
Attrition
No
Yes



300
250
200
150
100
50
0
Count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
YearsInCurrentRole
Attrition
No
Yes

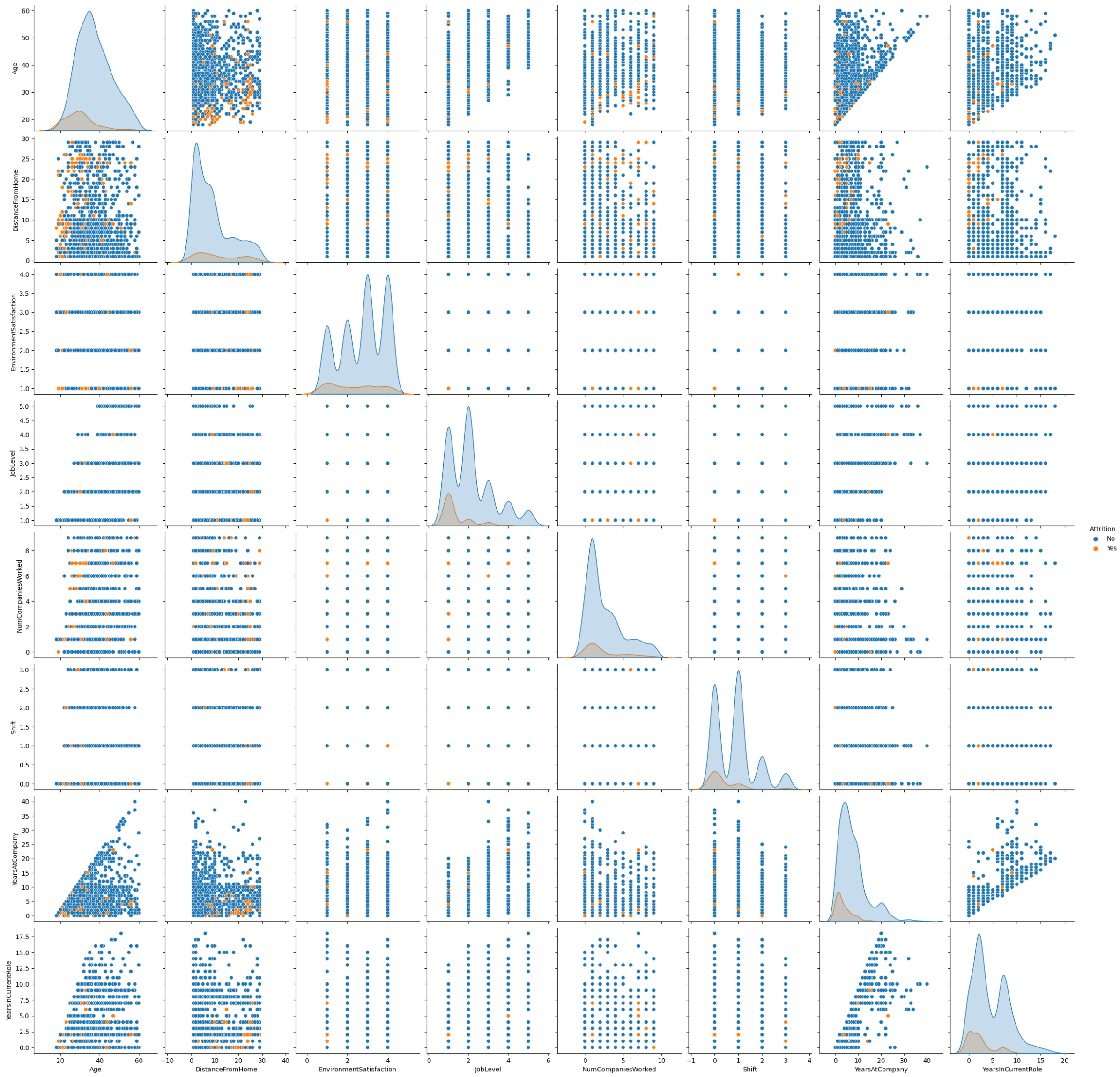


400
300
200
100
0
Count
0
2
4
6
8
10
12
14
YearsSinceLastPromotion
Attrition
No
Yes




In [ ]:
```
# Make pairwise scatter plots promising features seen in above analysis
promising_features = ["Age", "DistanceFromHome",
                      "EnvironmentSatisfaction",
                      "JobLevel", "JobRole",
                      "MaritalStatus",
                      "NumCompaniesWorked", "OverTime",
                      "Shift", "YearsAtCompany",
                      "YearsInCurrentRole",
                      "Attrition"]

promising_data = data[promising_features]

sns.pairplot(promising_data,
             hue="Attrition", height= 3)
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x2abee75b0>




In [ ]:
```python
# Import other data
labels = data.pop("Attrition")
submission_data = pd.read_csv("test.csv")
print(data.shape)
print(submission_data.shape)
```

```
(1340, 34)
(336, 34)
```

In [ ]:
```python
# drop columns with only 1 unique value
for c in data.columns:
    print(c, " num unique values: ", len(data[c].unique()), "\n")
    if len(data[c].unique()) == 1:
        data.drop([c], axis=1, inplace=True)
        submission_data.drop([c], axis=1, inplace=True)
print(data.shape)
print(submission_data.shape)
```

```
EmployeeID  num unique values:  1340
Age  num unique values:  43
BusinessTravel  num unique values:  3
DailyRate  num unique values:  793
Department  num unique values:  3
DistanceFromHome  num unique values:  29
Education  num unique values:  5
EducationField  num unique values:  6
EmployeeCount  num unique values:  1
EnvironmentSatisfaction  num unique values:  4
Gender  num unique values:  2
HourlyRate  num unique values:  71
JobInvolvement  num unique values:  4
JobLevel  num unique values:  5
JobRole  num unique values:  5
JobSatisfaction  num unique values:  4
MaritalStatus  num unique values:  3
MonthlyIncome  num unique values:  1134
MonthlyRate  num unique values:  1191
NumCompaniesWorked  num unique values:  10
Over18  num unique values:  1
OverTime  num unique values:  2
PercentSalaryHike  num unique values:  15
PerformanceRating  num unique values:  2
RelationshipSatisfaction  num unique values:  4
StandardHours  num unique values:  1
Shift  num unique values:  4
TotalWorkingYears  num unique values:  40
```

```
TrainingTimesLastYear  num unique values:  7

WorkLifeBalance  num unique values:  4

YearsAtCompany  num unique values:  37

YearsInCurrentRole  num unique values:  19

YearsSinceLastPromotion  num unique values:  16

YearsWithCurrManager  num unique values:  18

(1340, 31)
(336, 31)
```

See that there are 9 categorical columns which need to be converted to numerical.

See that there are many numerical columns which need to be binned by quantile.

Dropped columns with only 1 unique value -> can't get any information from those.

Can see that there are no null values, so we do not need to clean out rows or columns containing nulls.

In [ ]:
```
# Binning
int_cols = []
for c in data.columns:
    if data[c].dtype == "int64":
        print(c, len(data[c].unique()))
        print(data[c].describe(), "\n\n\n")
```

```
EmployeeID 1340
count    1.340000e+03
mean     1.460265e+06
std      2.494821e+05
min      1.025177e+06
25%      1.237599e+06
50%      1.469862e+06
75%      1.670131e+06
max      1.886378e+06
Name: EmployeeID, dtype: float64



Age 43
count    1340.000000
mean       36.580597
std         9.013072
min        18.000000
25%        30.000000
50%        35.000000
75%        42.000000
max        60.000000
Name: Age, dtype: float64



DailyRate 793
count    1340.000000
mean      799.197761
std       399.333256
min       102.000000
25%       465.000000
50%       796.000000
75%      1153.000000
max      1499.000000
Name: DailyRate, dtype: float64



DistanceFromHome 29
count    1340.000000
mean        9.193284
std         8.141621
min         1.000000
25%         2.000000
50%         7.000000
75%        14.000000
max        29.000000
Name: DistanceFromHome, dtype: float64



Education 5
count    1340.000000
mean        2.924627
std         1.036088
```

```
min          1.000000
25%          2.000000
50%          3.000000
75%          4.000000
max          5.000000
Name: Education, dtype: float64



EnvironmentSatisfaction 4
count    1340.000000
mean        2.709701
std         1.099961
min         1.000000
25%         2.000000
50%         3.000000
75%         4.000000
max         4.000000
Name: EnvironmentSatisfaction, dtype: float64



HourlyRate 71
count    1340.000000
mean       65.559701
std        20.335025
min        30.000000
25%        48.000000
50%        65.000000
75%        83.000000
max       100.000000
Name: HourlyRate, dtype: float64



JobInvolvement 4
count    1340.000000
mean        2.717910
std         0.717523
min         1.000000
25%         2.000000
50%         3.000000
75%         3.000000
max         4.000000
Name: JobInvolvement, dtype: float64



JobLevel 5
count    1340.000000
mean        2.051493
std         1.104491
min         1.000000
25%         1.000000
50%         2.000000
75%         3.000000
```

```
max          5.000000
Name: JobLevel, dtype: float64



JobSatisfaction 4
count    1340.000000
mean        2.746269
std         1.111328
min         1.000000
25%         2.000000
50%         3.000000
75%         4.000000
max         4.000000
Name: JobSatisfaction, dtype: float64



MonthlyIncome 1134
count     1340.000000
mean      6433.381343
std       4687.058380
min       1051.000000
25%       2870.000000
50%       4876.500000
75%       8038.750000
max      19973.000000
Name: MonthlyIncome, dtype: float64



MonthlyRate 1191
count     1340.000000
mean     14290.377612
std       7166.995911
min       2094.000000
25%       7967.250000
50%      14288.500000
75%      20472.500000
max      26997.000000
Name: MonthlyRate, dtype: float64



NumCompaniesWorked 10
count    1340.000000
mean        2.600000
std         2.472794
min         0.000000
25%         1.000000
50%         1.000000
75%         4.000000
max         9.000000
Name: NumCompaniesWorked, dtype: float64
```

```
PercentSalaryHike 15
count    1340.000000
mean       15.168657
std         3.661956
min        11.000000
25%        12.000000
50%        14.000000
75%        18.000000
max        25.000000
Name: PercentSalaryHike, dtype: float64
```

```
PerformanceRating 2
count    1340.000000
mean        3.152239
std         0.359386
min         3.000000
25%         3.000000
50%         3.000000
75%         3.000000
max         4.000000
Name: PerformanceRating, dtype: float64
```

```
RelationshipSatisfaction 4
count    1340.000000
mean        2.700000
std         1.079858
min         1.000000
25%         2.000000
50%         3.000000
75%         4.000000
max         4.000000
Name: RelationshipSatisfaction, dtype: float64
```

```
Shift 4
count    1340.000000
mean        0.808209
std         0.856251
min         0.000000
25%         0.000000
50%         1.000000
75%         1.000000
max         3.000000
Name: Shift, dtype: float64
```

```
TotalWorkingYears 40
count    1340.000000
mean       11.222388
```

```
std          7.696043
min          0.000000
25%          6.000000
50%         10.000000
75%         15.000000
max         40.000000
Name: TotalWorkingYears, dtype: float64



TrainingTimesLastYear 7
count    1340.000000
mean        2.785821
std         1.263473
min         0.000000
25%         2.000000
50%         3.000000
75%         3.000000
max         6.000000
Name: TrainingTimesLastYear, dtype: float64



WorkLifeBalance 4
count    1340.000000
mean        2.771642
std         0.700007
min         1.000000
25%         2.000000
50%         3.000000
75%         3.000000
max         4.000000
Name: WorkLifeBalance, dtype: float64



YearsAtCompany 37
count    1340.000000
mean        7.070149
std         6.039663
min         0.000000
25%         3.000000
50%         5.000000
75%        10.000000
max        40.000000
Name: YearsAtCompany, dtype: float64



YearsInCurrentRole 19
count    1340.000000
mean        4.272388
std         3.677798
min         0.000000
25%         2.000000
50%         3.000000
```

```
75%         7.000000
max        18.000000
Name: YearsInCurrentRole, dtype: float64



YearsSinceLastPromotion 16
count    1340.000000
mean        2.175373
std         3.222376
min         0.000000
25%         0.000000
50%         1.000000
75%         3.000000
max        15.000000
Name: YearsSinceLastPromotion, dtype: float64



YearsWithCurrManager 18
count    1340.000000
mean        4.167164
std         3.581605
min         0.000000
25%         2.000000
50%         3.000000
75%         7.000000
max        17.000000
Name: YearsWithCurrManager, dtype: float64
```

In [ ]:
```
# Binning
# found manually
cols_to_bin = ["Age", "DailyRate", "DistanceFromHome",
               "HourlyRate", "MonthlyIncome", "MonthlyRate",
               "PercentSalaryHike", "TotalWorkingYears",
               "YearsAtCompany","YearsInCurrentRole",
               "YearsWithCurrManager", "NumCompaniesWorked",
               "YearsSinceLastPromotion",]

print(data.shape)
print(submission_data.shape)

# uneven 4 groups
for c in cols_to_bin:
    try:
        data[c+"_Even"] = pd.cut(data[c], 4, labels=False)
        submission_data[c+"_Even"] = pd.cut(submission_data[c], 4, labels=Fa
    except:
        print("failed")
        pass
```

```
print(data.shape)
print(submission_data.shape)
```

```
(1340, 31)
(336, 31)
(1340, 44)
(336, 44)
```

In [ ]:
```
# get dummies from int categorical data

# found manually
already_categorical = ["Education", "EnvironmentSatisfaction",
                       "JobInvolvement", "JobLevel", "JobSatisfaction",
                       "PerformanceRating", "RelationshipSatisfaction",
                       "Shift", "TrainingTimesLastYear",
                       "WorkLifeBalance"]

for c in already_categorical:
    temp_dummy = pd.get_dummies(data[c], prefix=c)
    data = pd.concat([data, temp_dummy], axis=1)

    sub_temp_dummy = pd.get_dummies(submission_data[c], prefix=c)
    submission_data = pd.concat([submission_data, sub_temp_dummy], axis=1)

print(data.shape)
print(submission_data.shape)
```

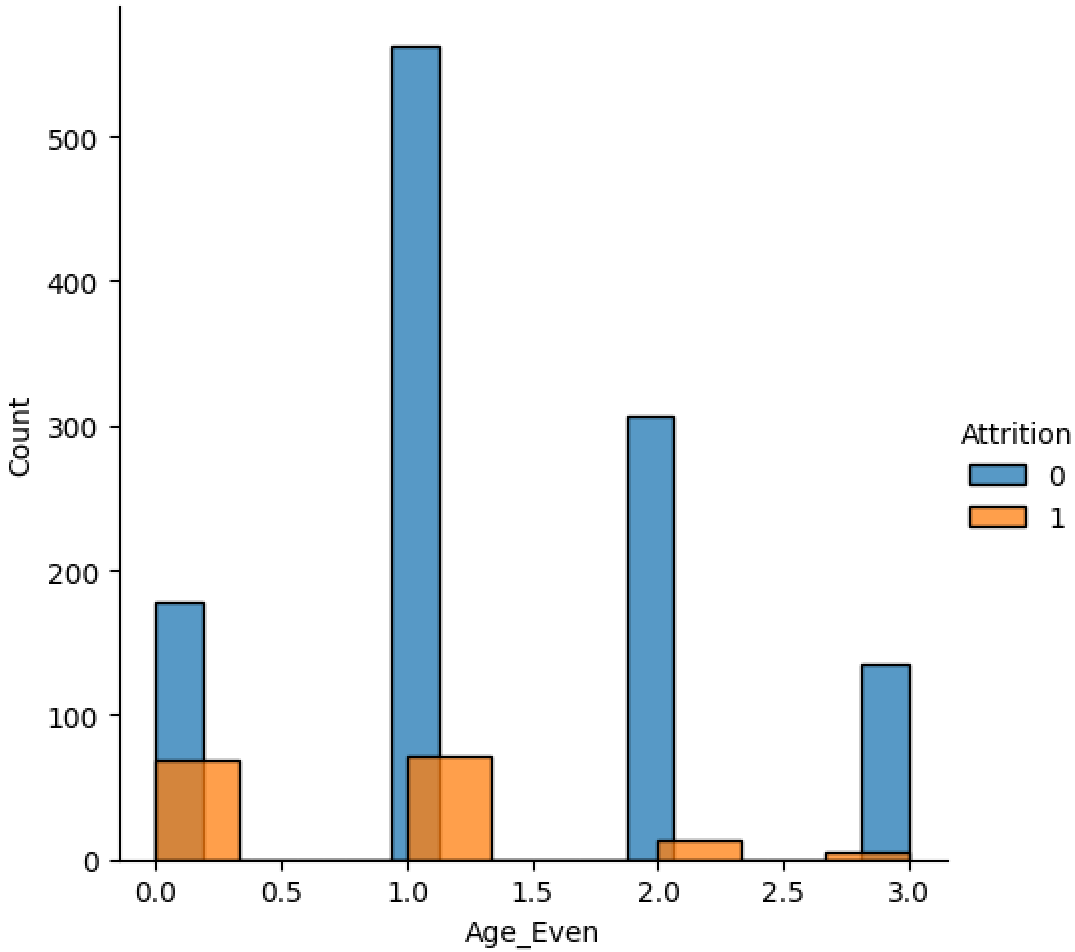
```
(1340, 87)
(336, 87)
```

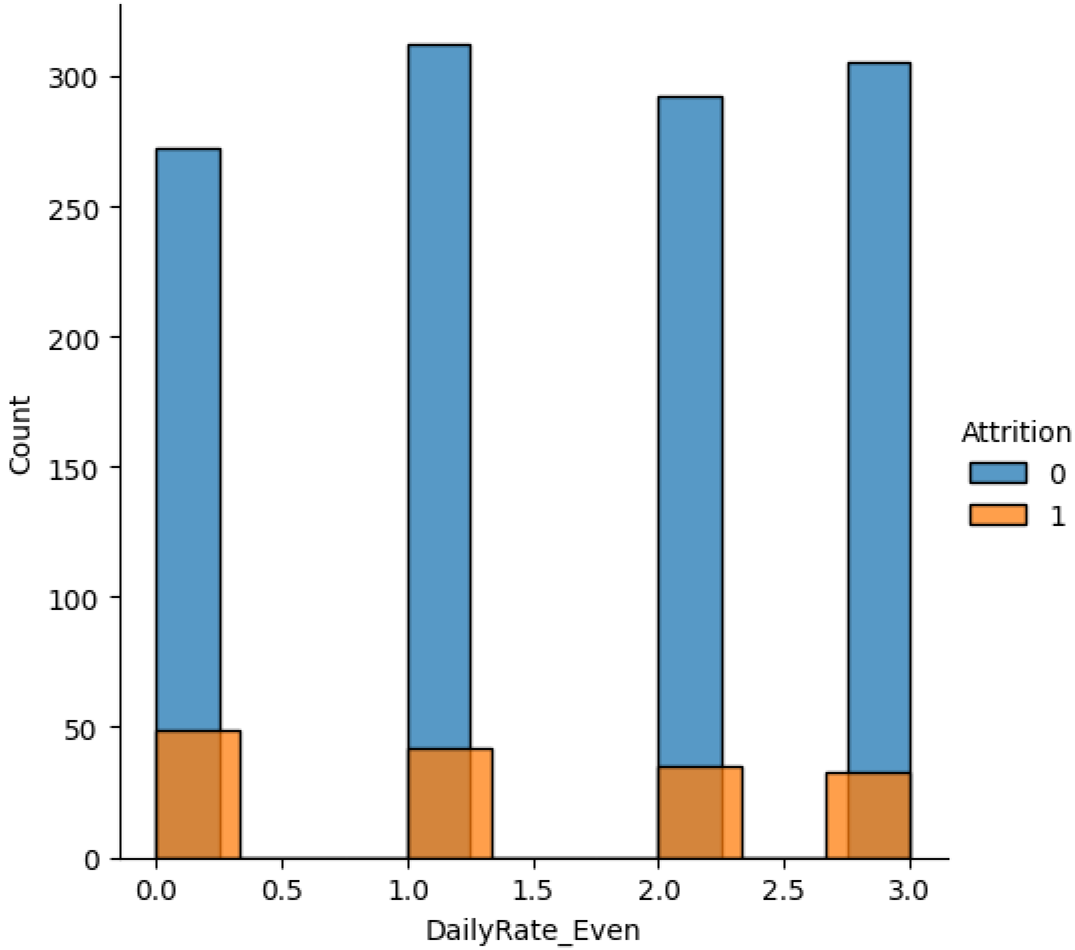
In [ ]:
```
# get dummies from obj categorical data
for c in data.columns:
    if data[c].dtype == "object":
        if len(data[c].unique()) == 2:
            data[c] = pd.factorize(data[c])[0]

            submission_data[c] = pd.factorize(submission_data[c])[0]
        else:
            temp_dummy = pd.get_dummies(data[c], prefix=c)
            data = pd.concat([data, temp_dummy], axis=1)
            data[c] = pd.factorize(data[c])[0]

            sub_temp_dummy = pd.get_dummies(submission_data[c], prefix=c)
            submission_data = pd.concat([submission_data, sub_temp_dummy], a
            submission_data[c] = pd.factorize(submission_data[c])[0]

print(data.shape)
print(submission_data.shape)

# turn label column to binary
labels = pd.DataFrame(pd.factorize(labels)[0], columns=["Attrition"])
```

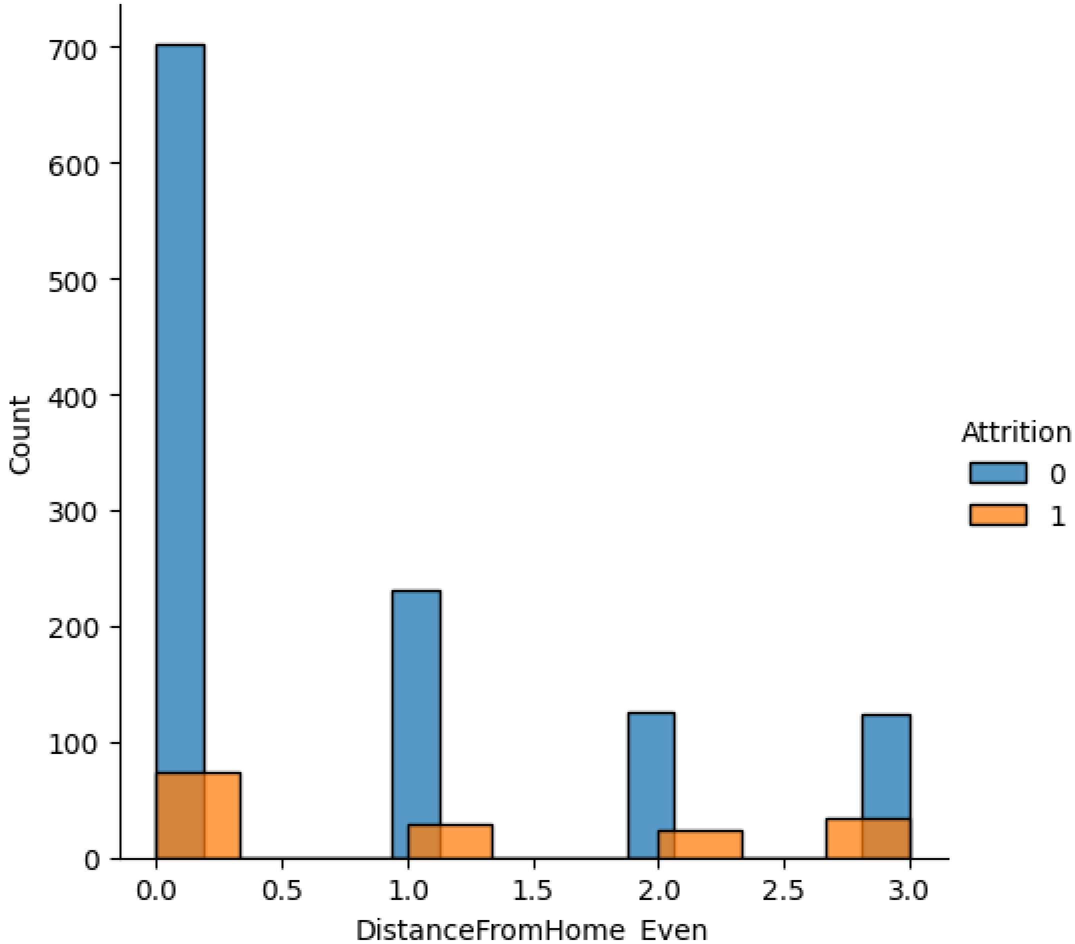
```
(1340, 107)
(336, 107)
```

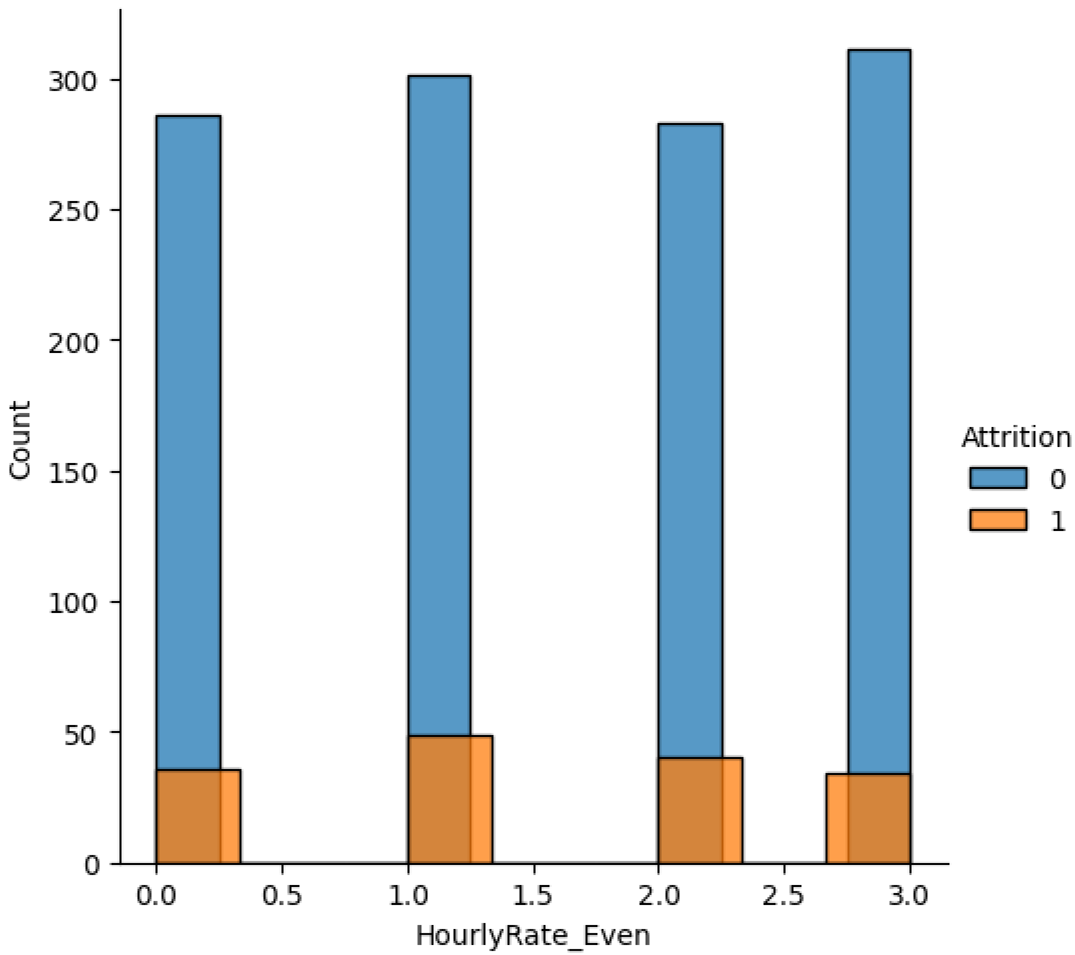
In [ ]:
```
# Histograms for binned data and dummmy data created
promising_data2 = data[data.columns[31:]]
```

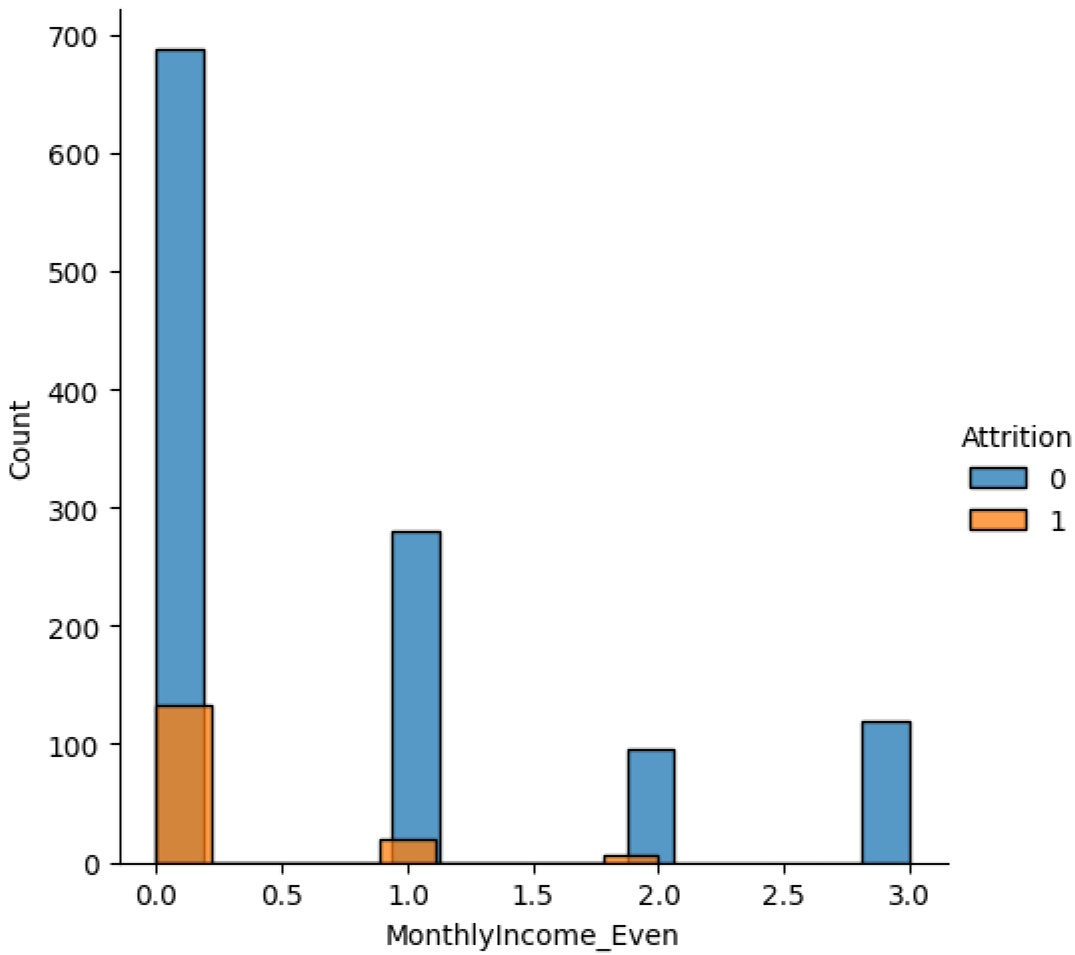
```
promising_data2 = pd.concat([promising_data2, labels], axis=1)

for c in promising_data2.columns:
    sns.FacetGrid(promising_data2,
                  hue="Attrition",
                  height= 5).map(sns.histplot,c).add_legend()
```

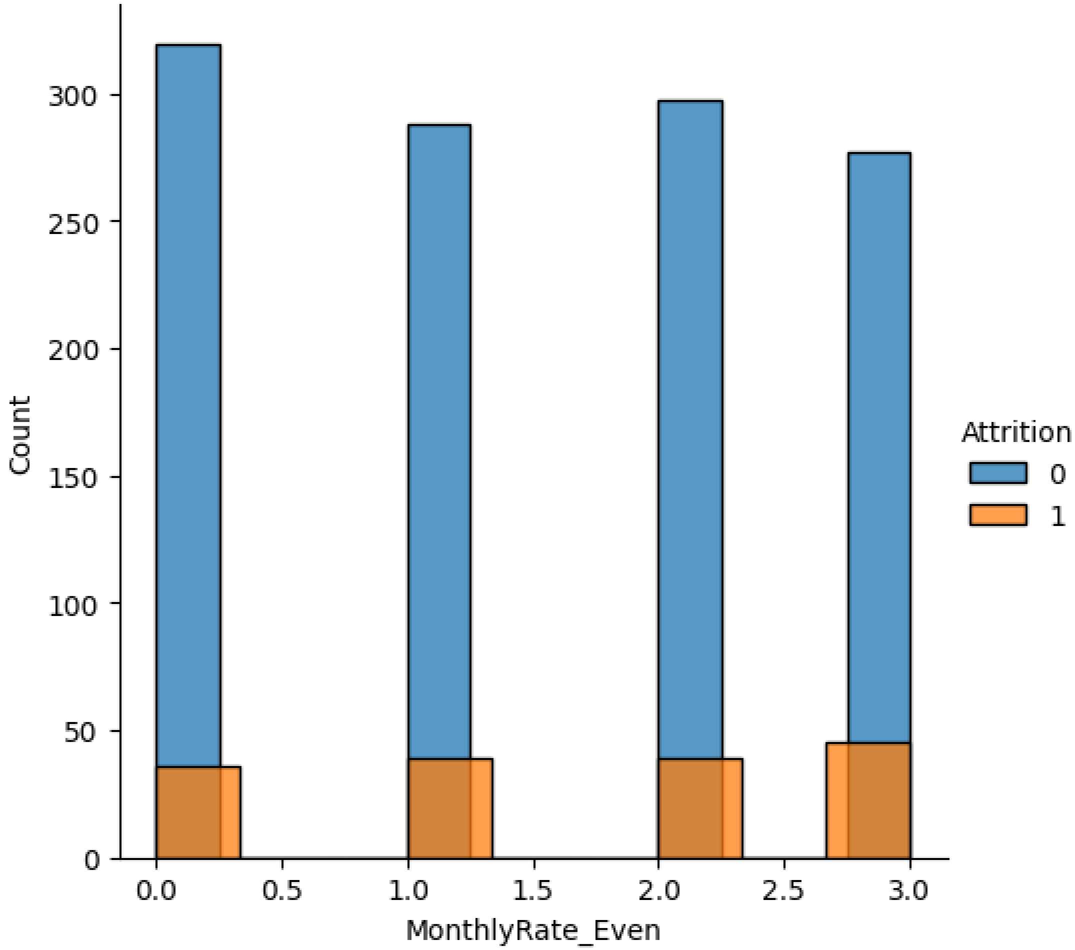



300
250
200
150
100
50
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
DailyRate_Even
Attrition
0
1

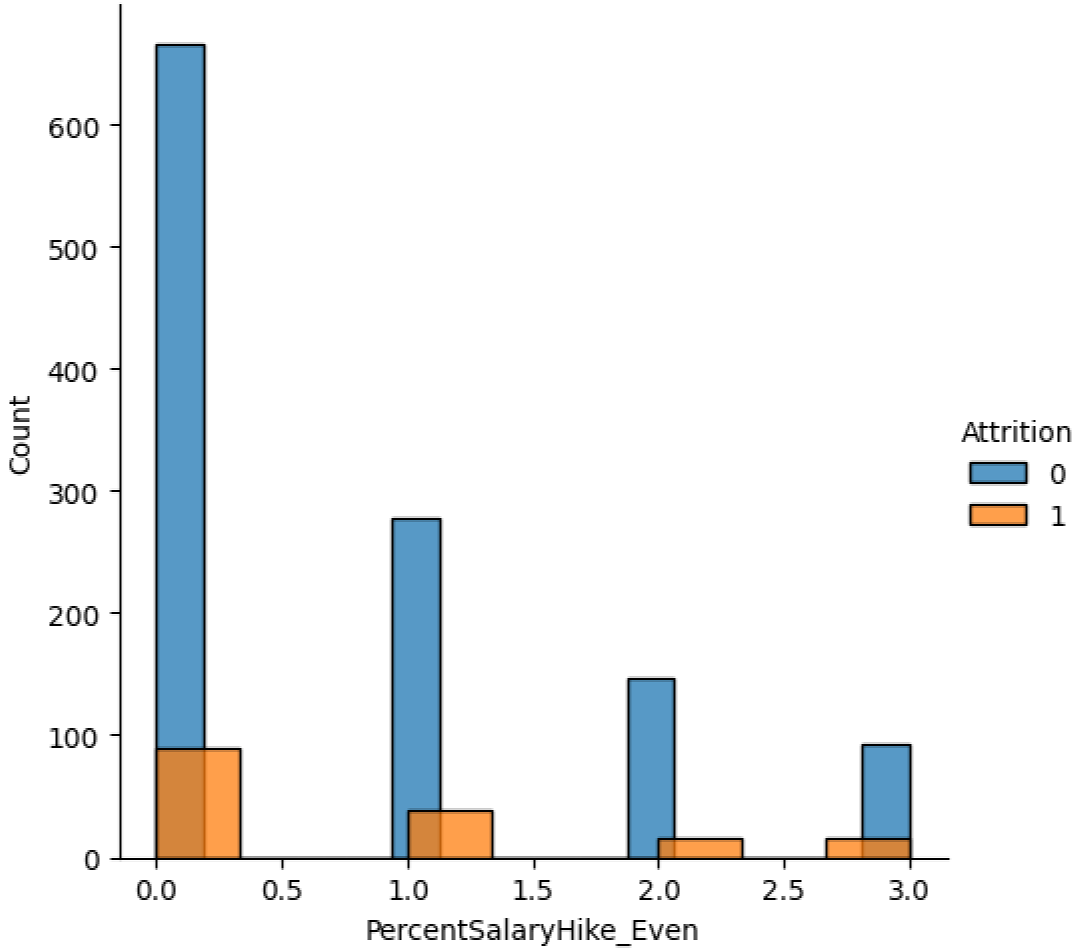

700
600
500
400
300
200
100
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
DistanceFromHome_Even
Attrition
0
1

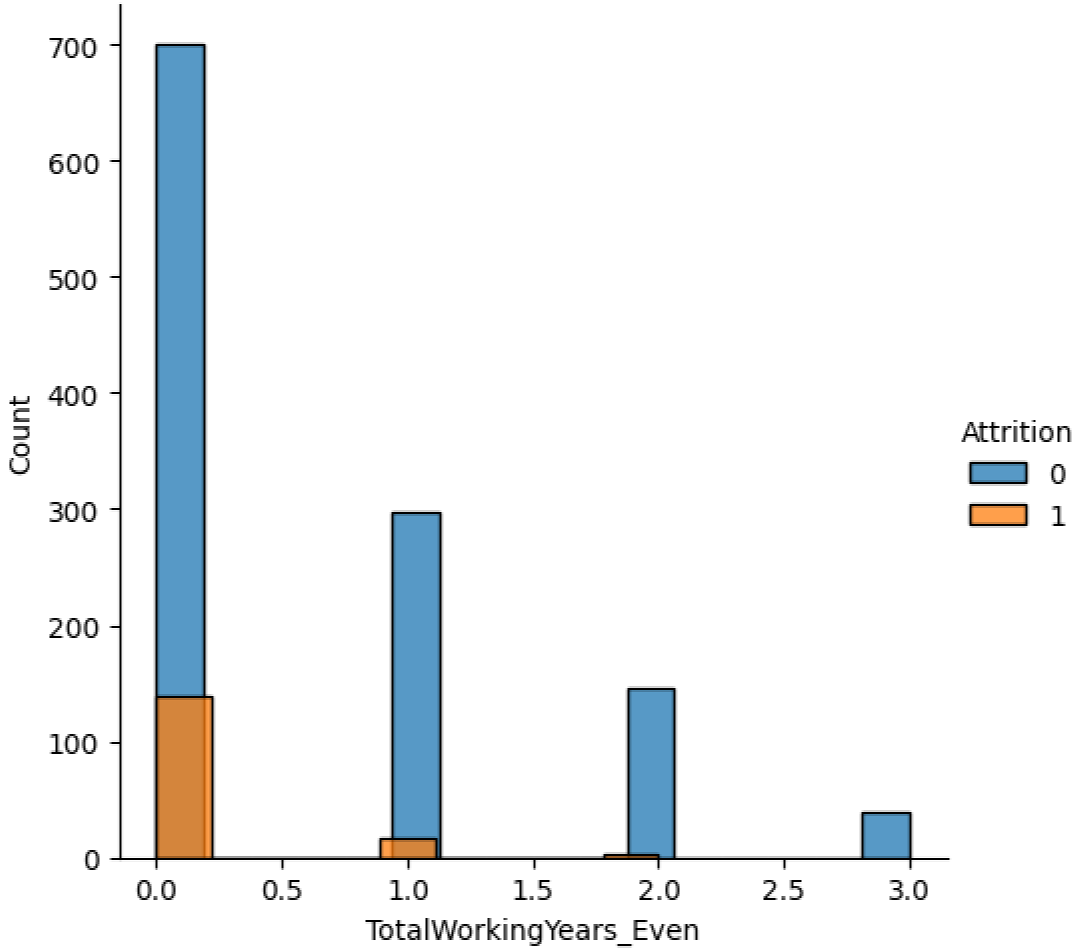

300
250
200
150
100
50
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
HourlyRate_Even
Attrition
0
1


700
600
500
400
300
200
100
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
MonthlyIncome_Even
Attrition
0
1

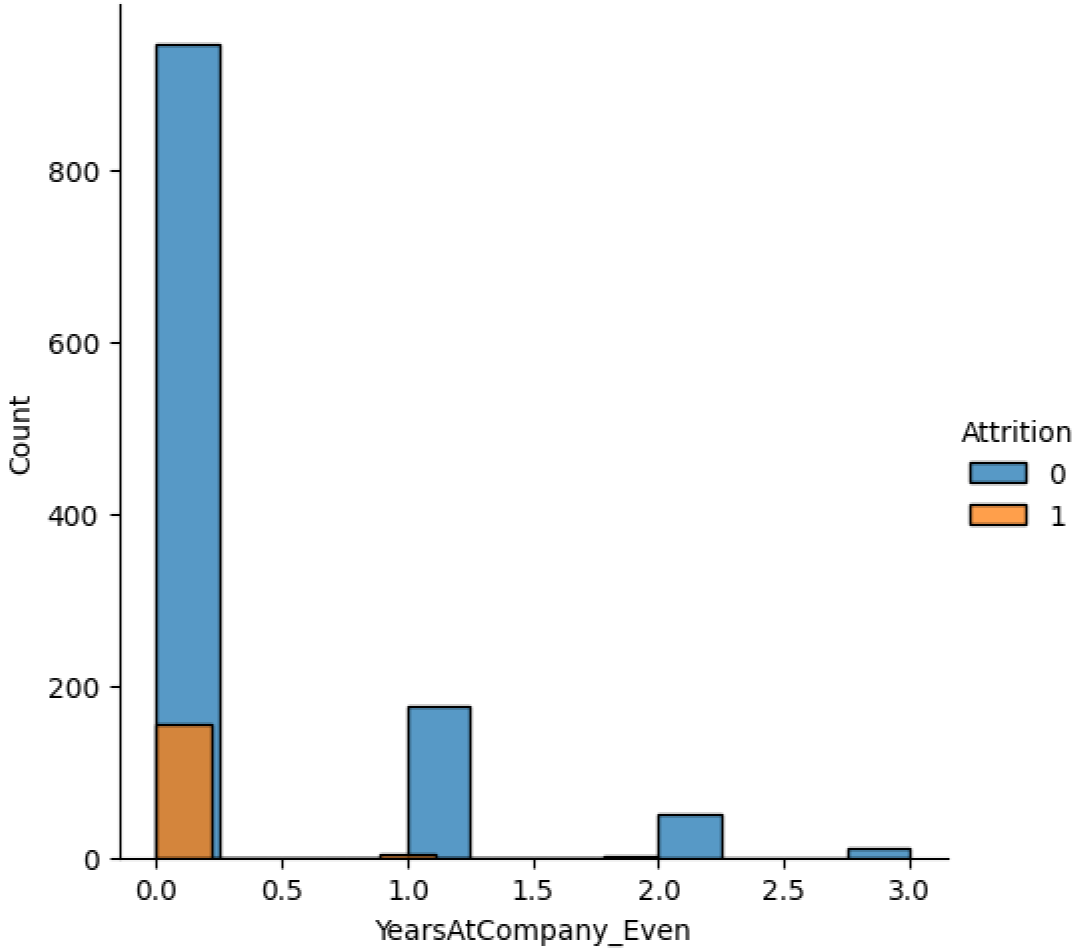


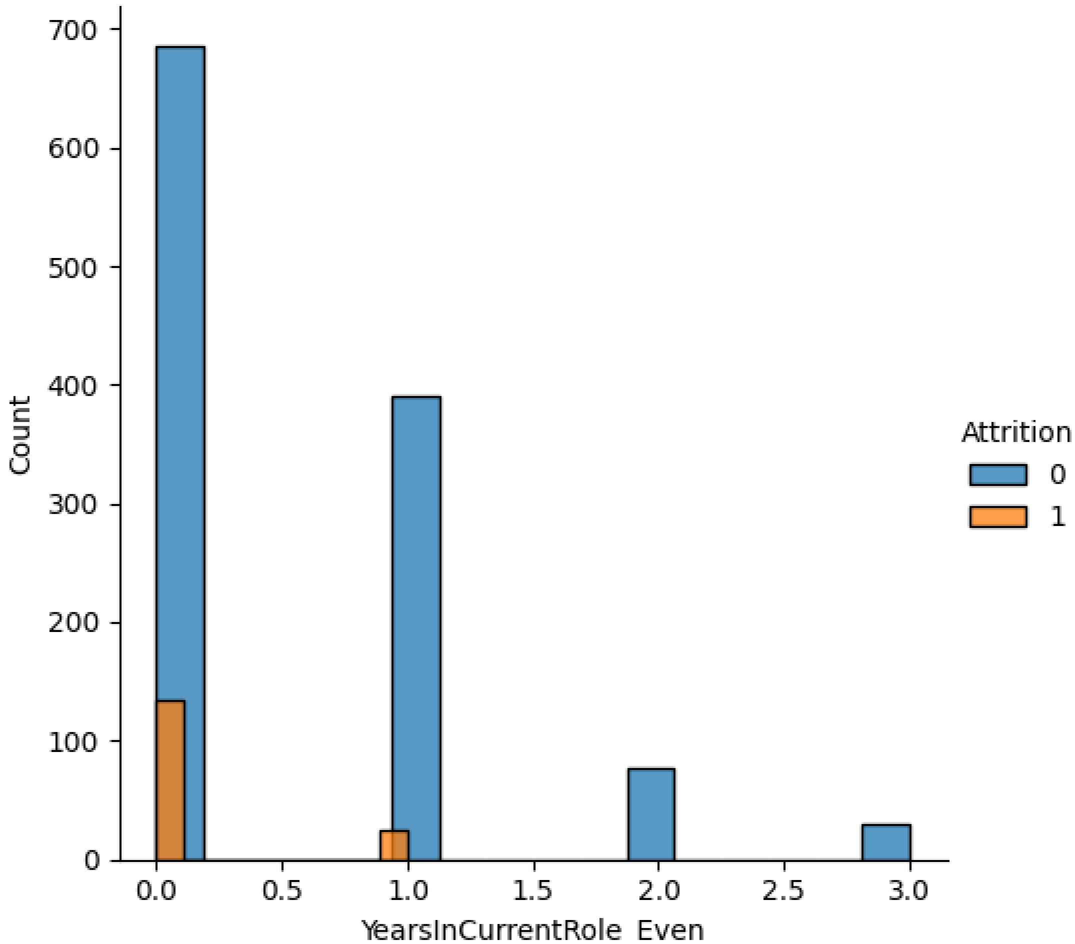
300
250
200
150
100
50
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
MonthlyRate_Even
Attrition
0
1

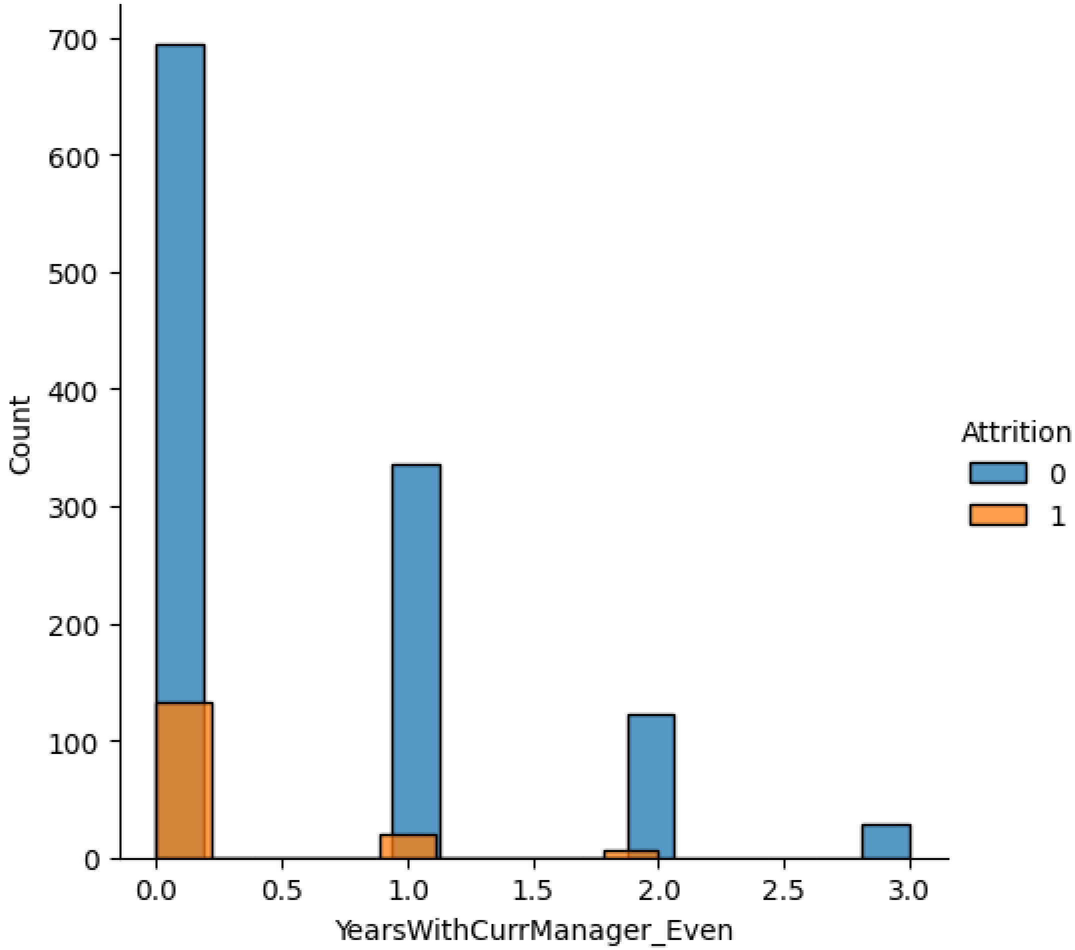

Count
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
PercentSalaryHike_Even
Attrition
0
1



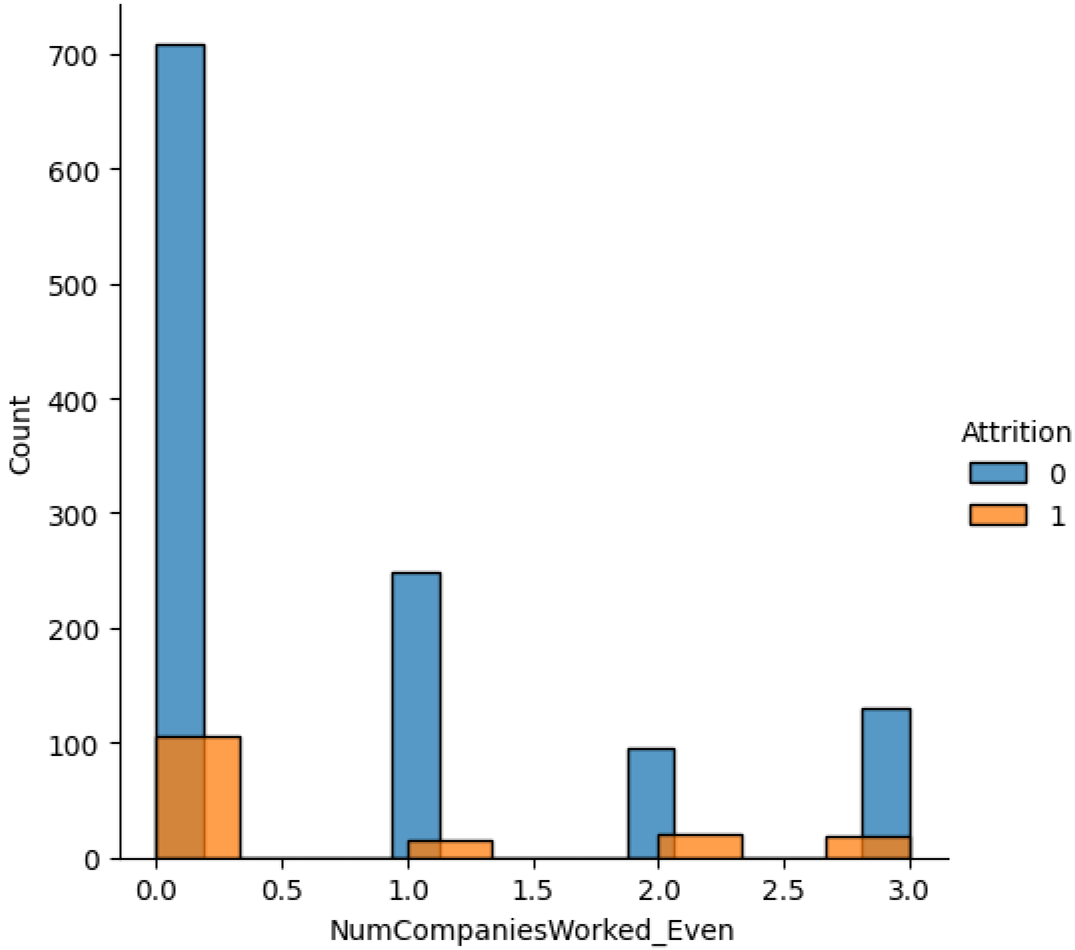
Count
700
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
TotalWorkingYears_Even
Attrition
0
1

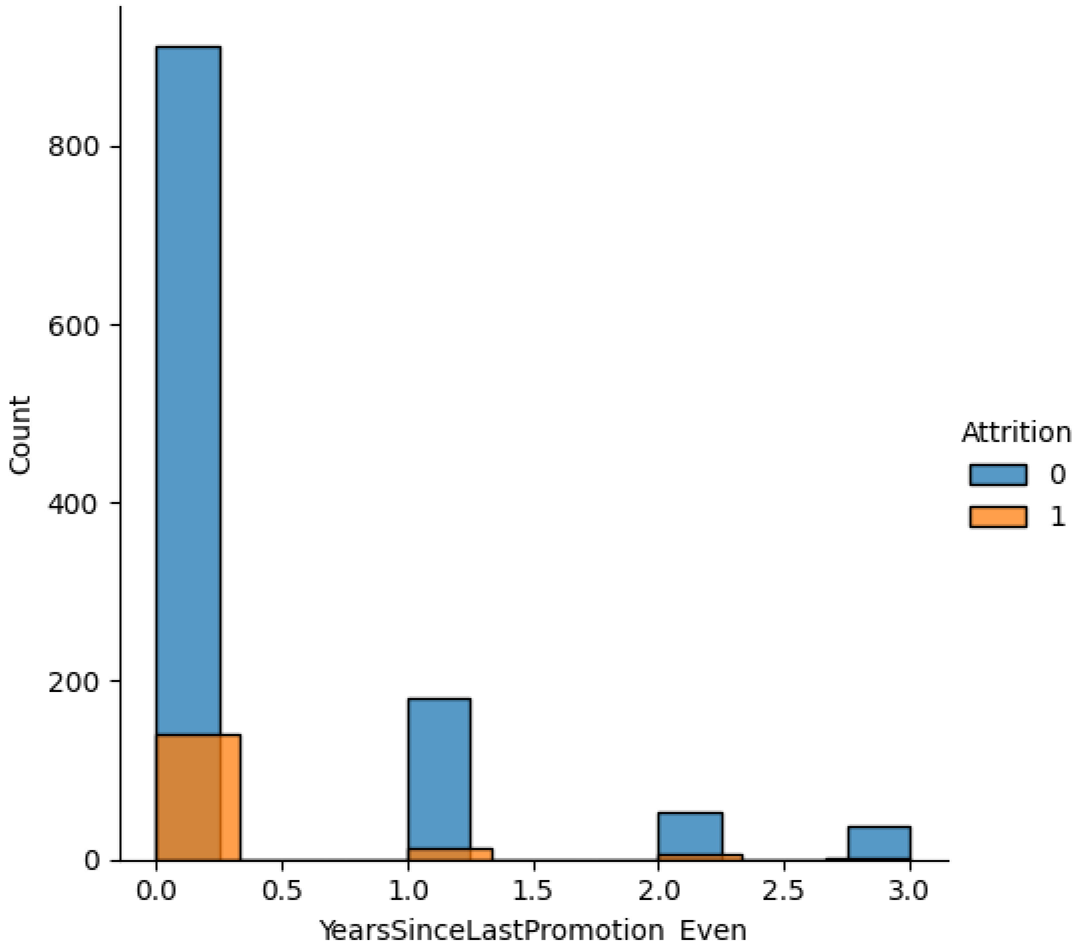

Count
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
YearsAtCompany_Even
Attrition
0
1


700
600
500
400
300
200
100
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
YearsInCurrentRole_Even
Attrition
0
1

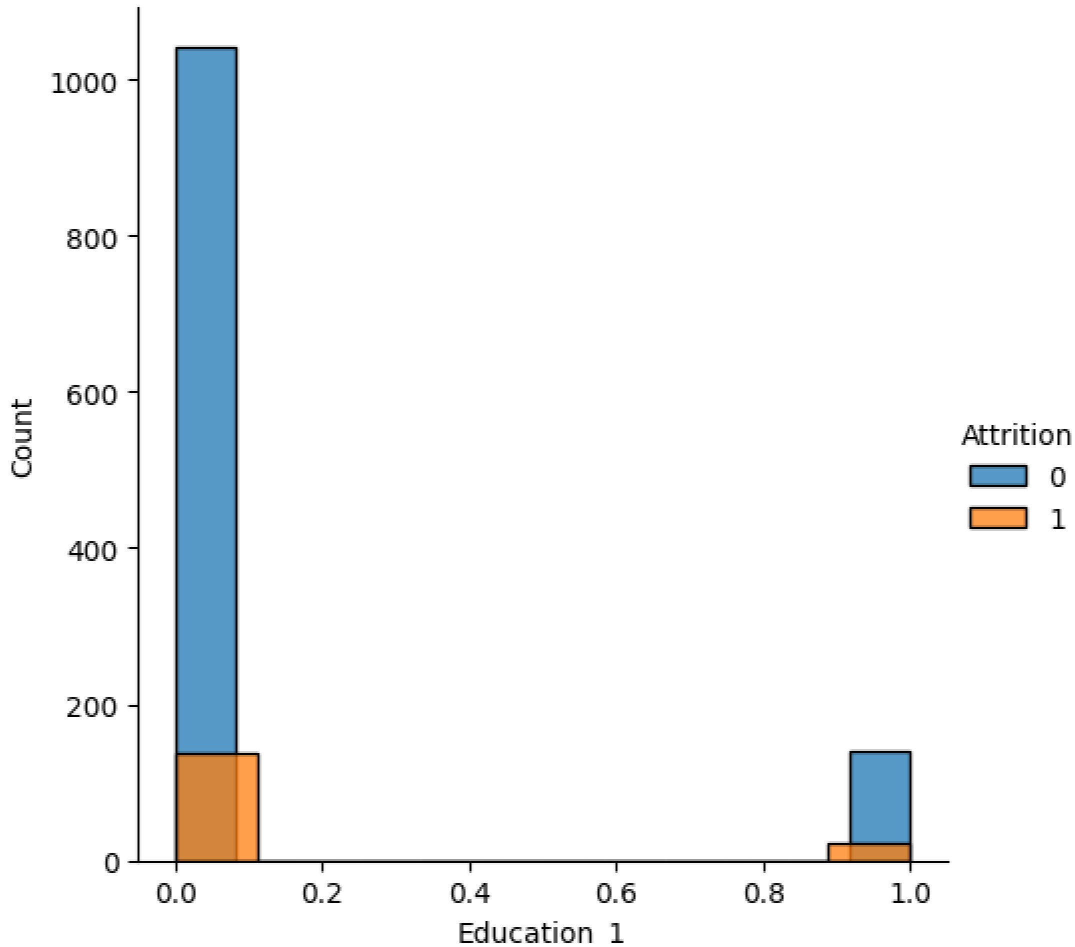


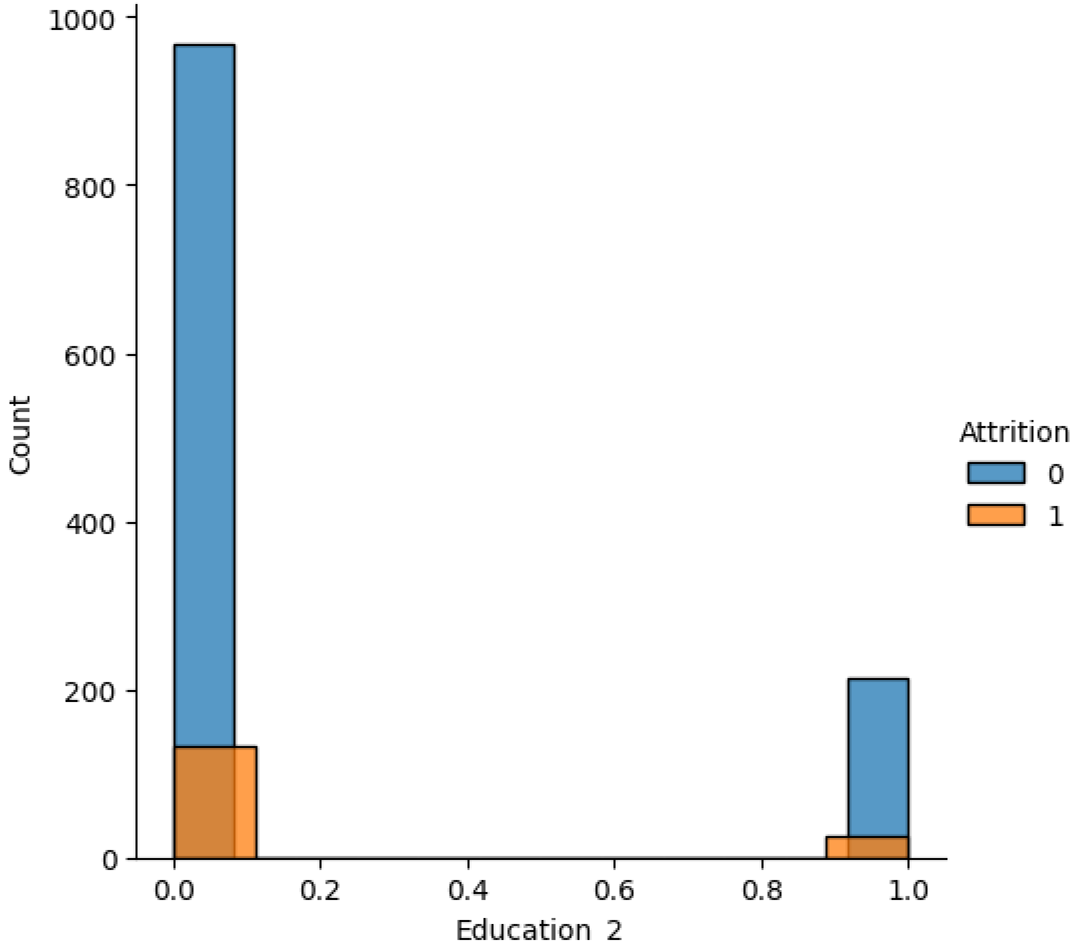
700
600
500
400
300
200
100
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
YearsWithCurrManager_Even
Attrition
0
1

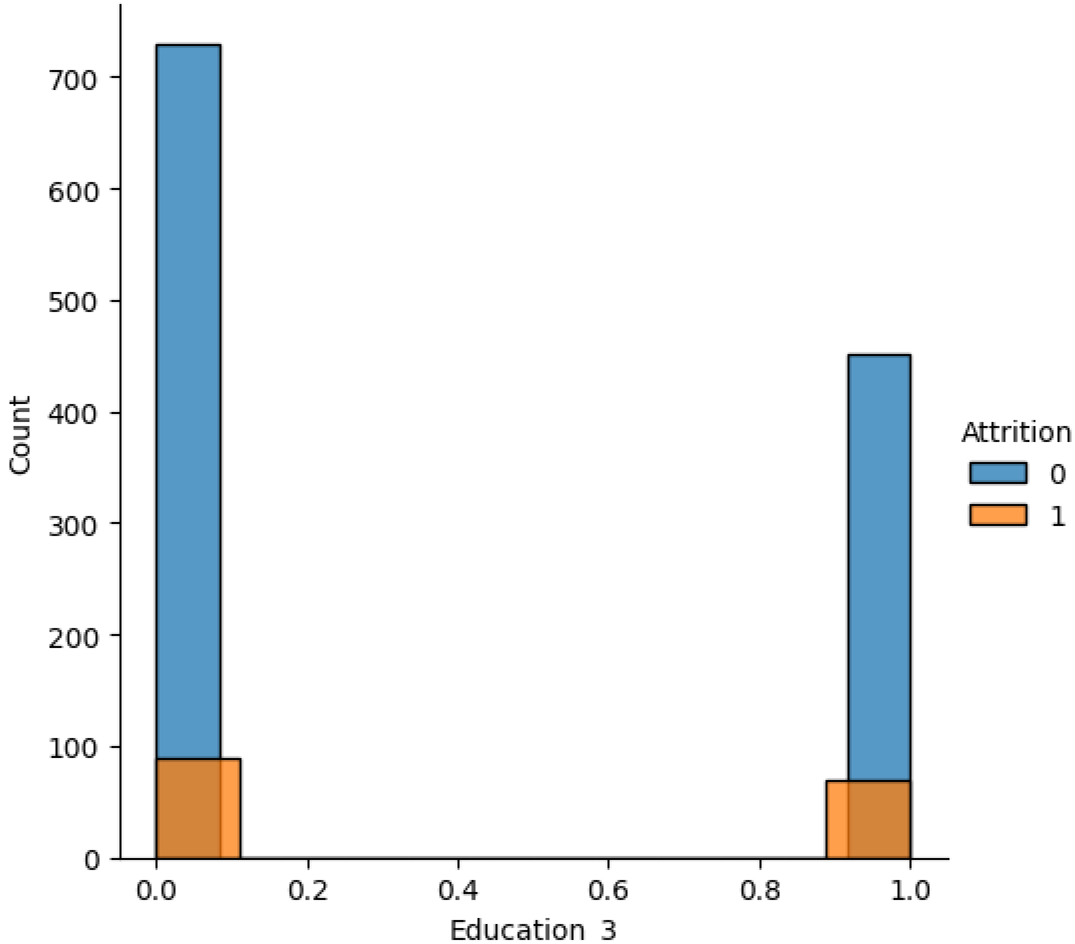

Count
700
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
NumCompaniesWorked_Even
Attrition
0
1

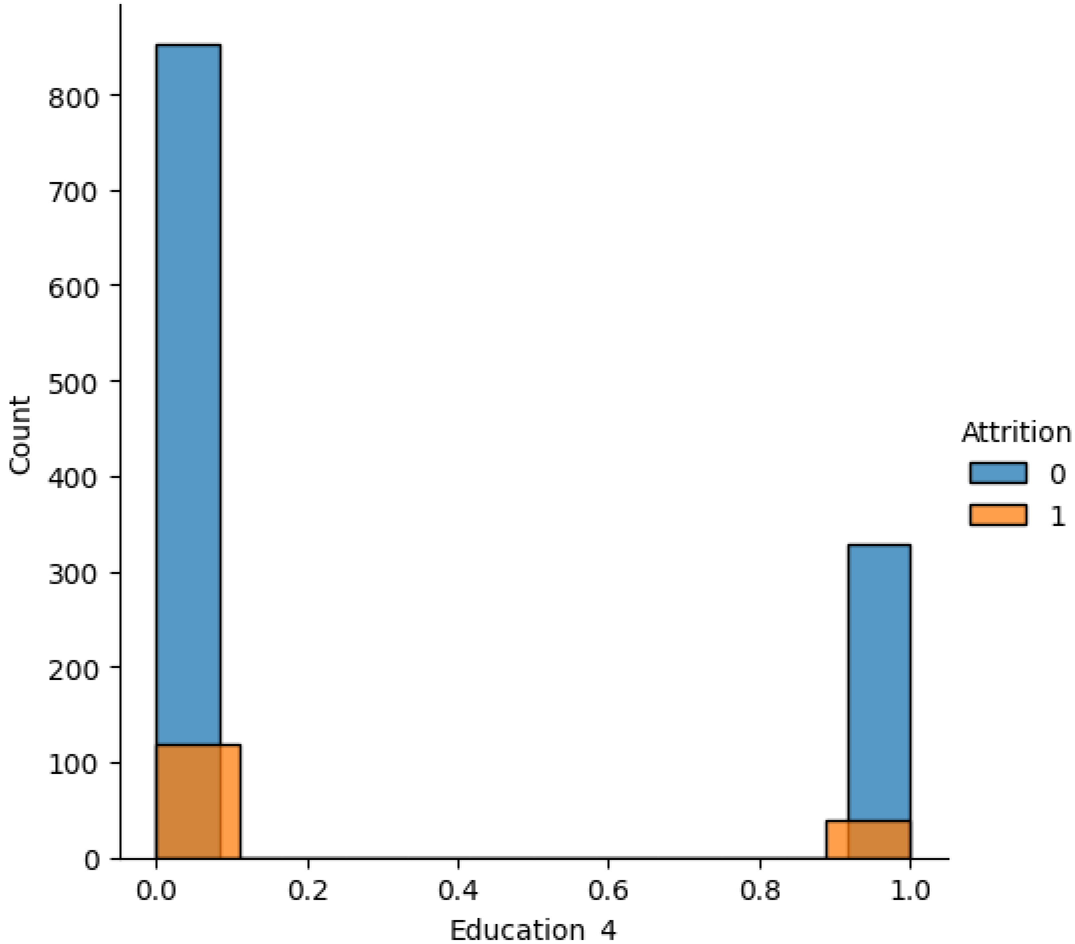


Count
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
YearsSinceLastPromotion_Even
Attrition
0
1

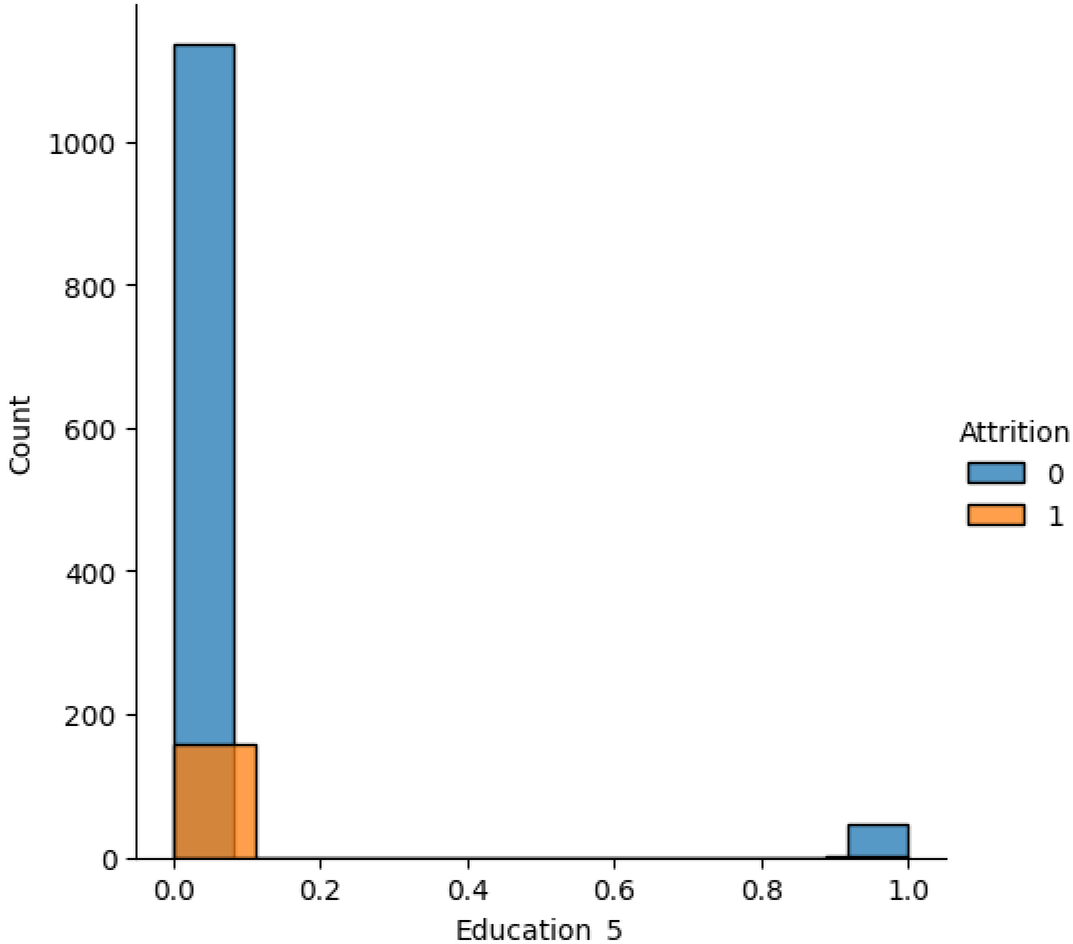

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Education_1
Attrition
0
1


1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Education_2
Attrition
0
1

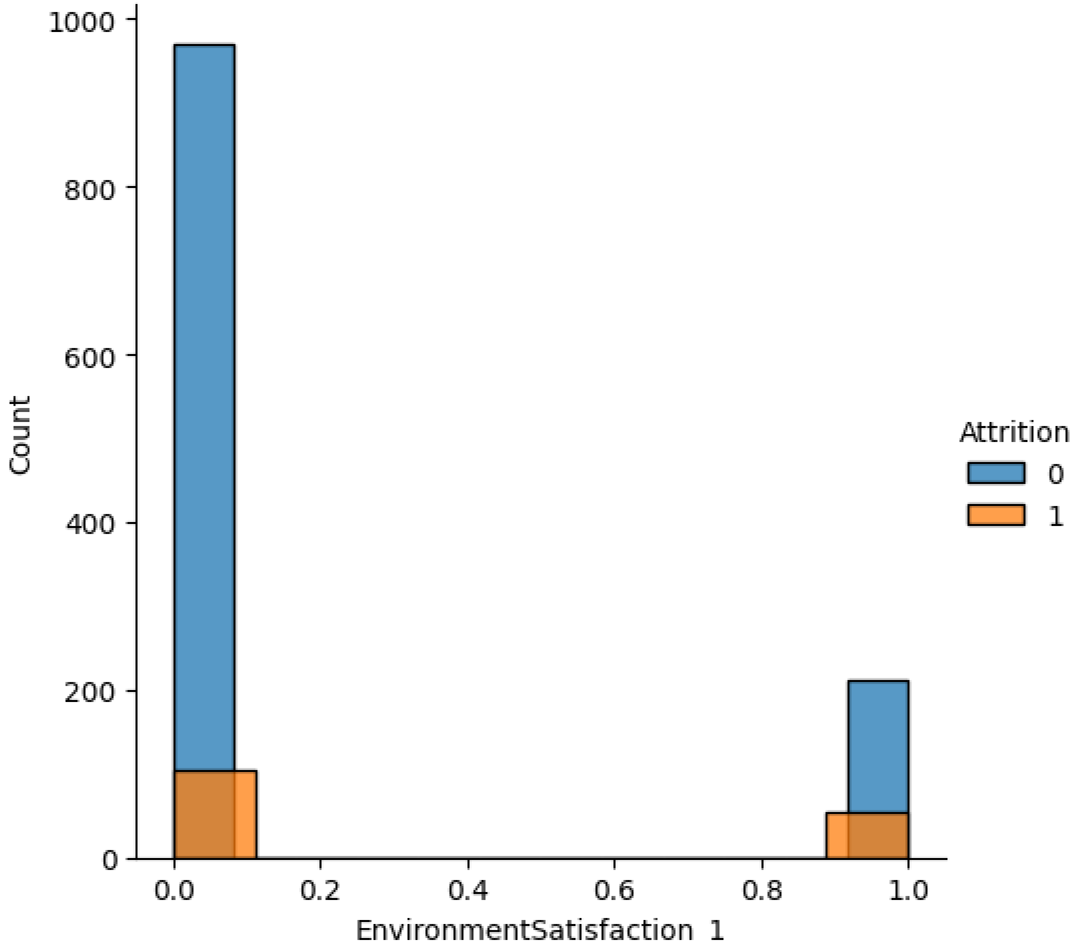

700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Education_3
Attrition
0
1



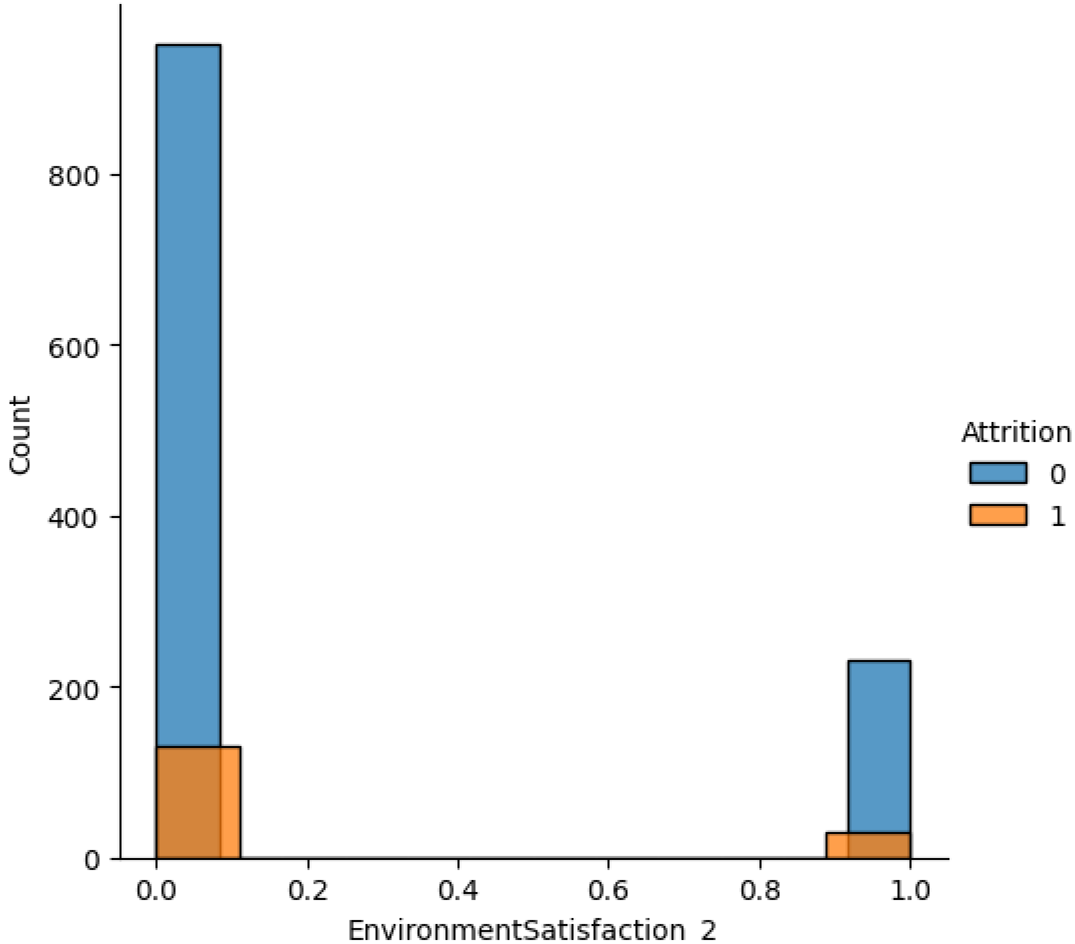
800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Education_4
Attrition
0
1

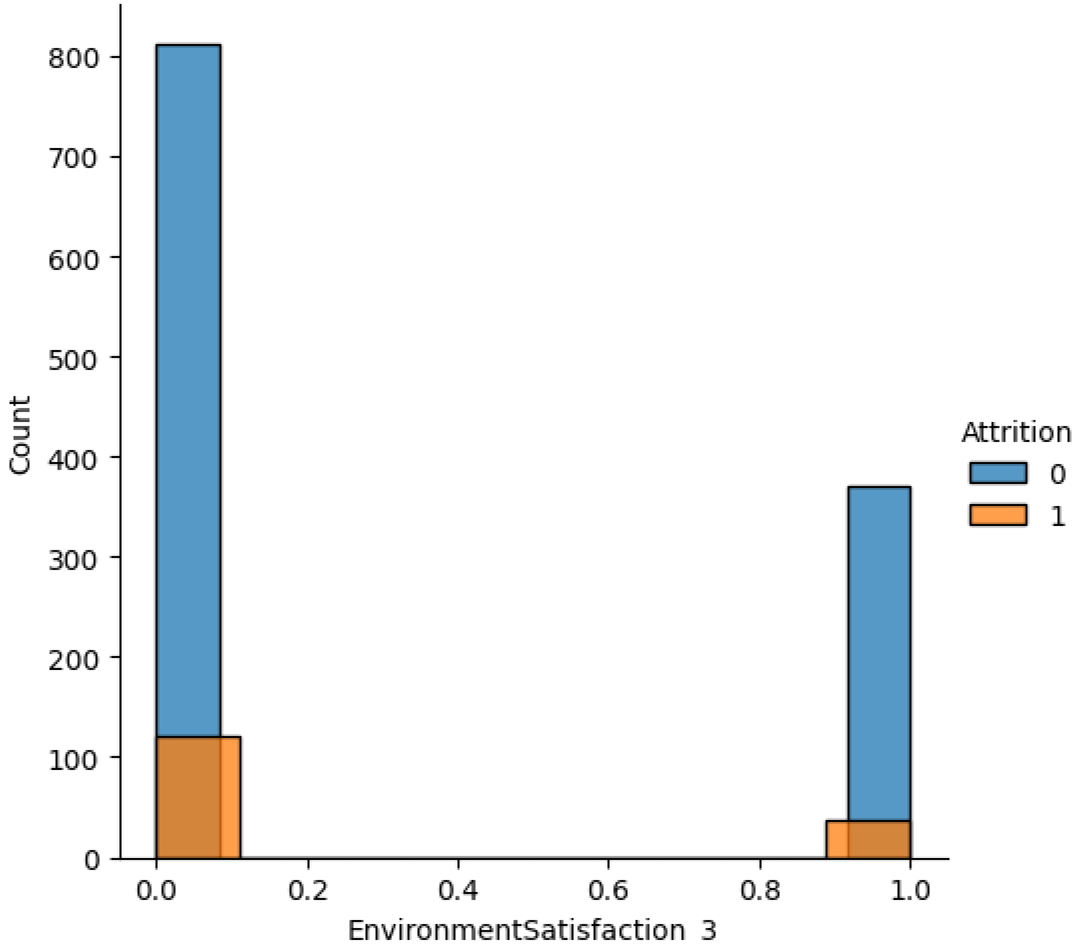


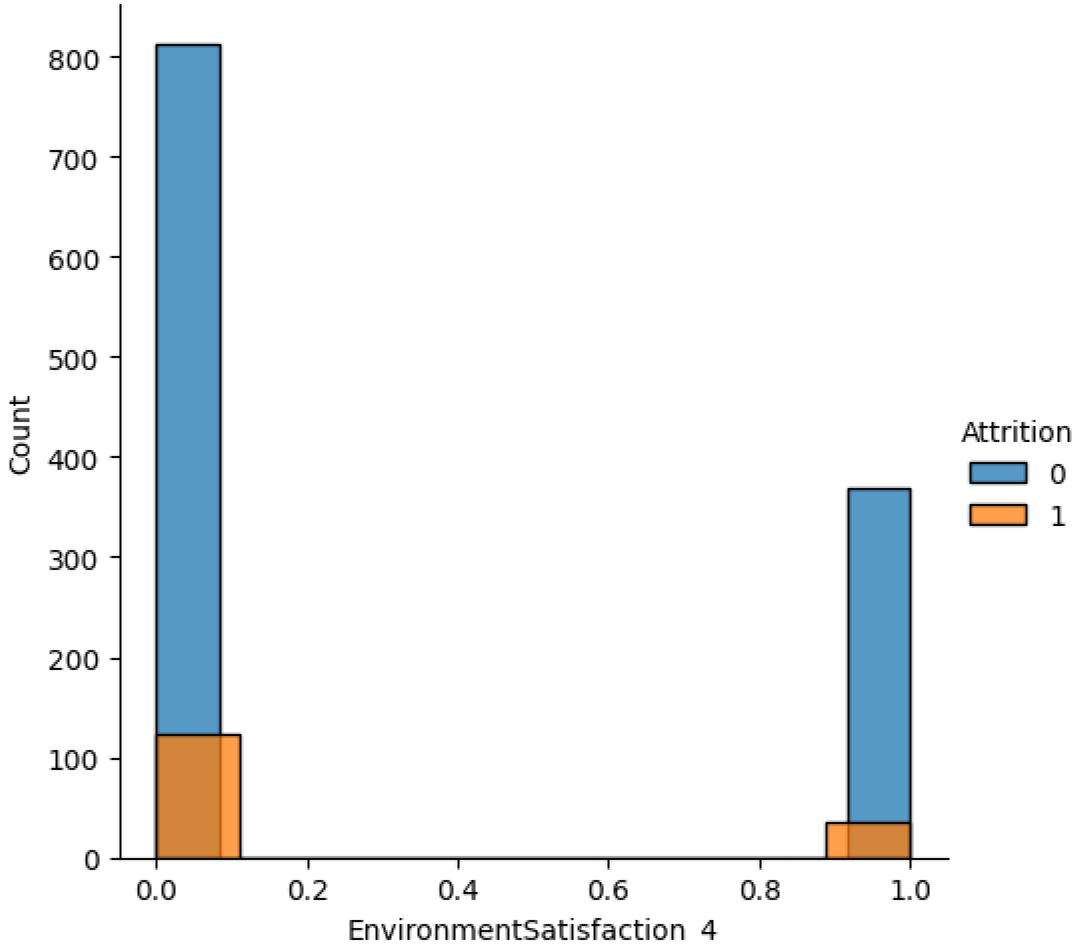
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Education_5
Attrition
0
1

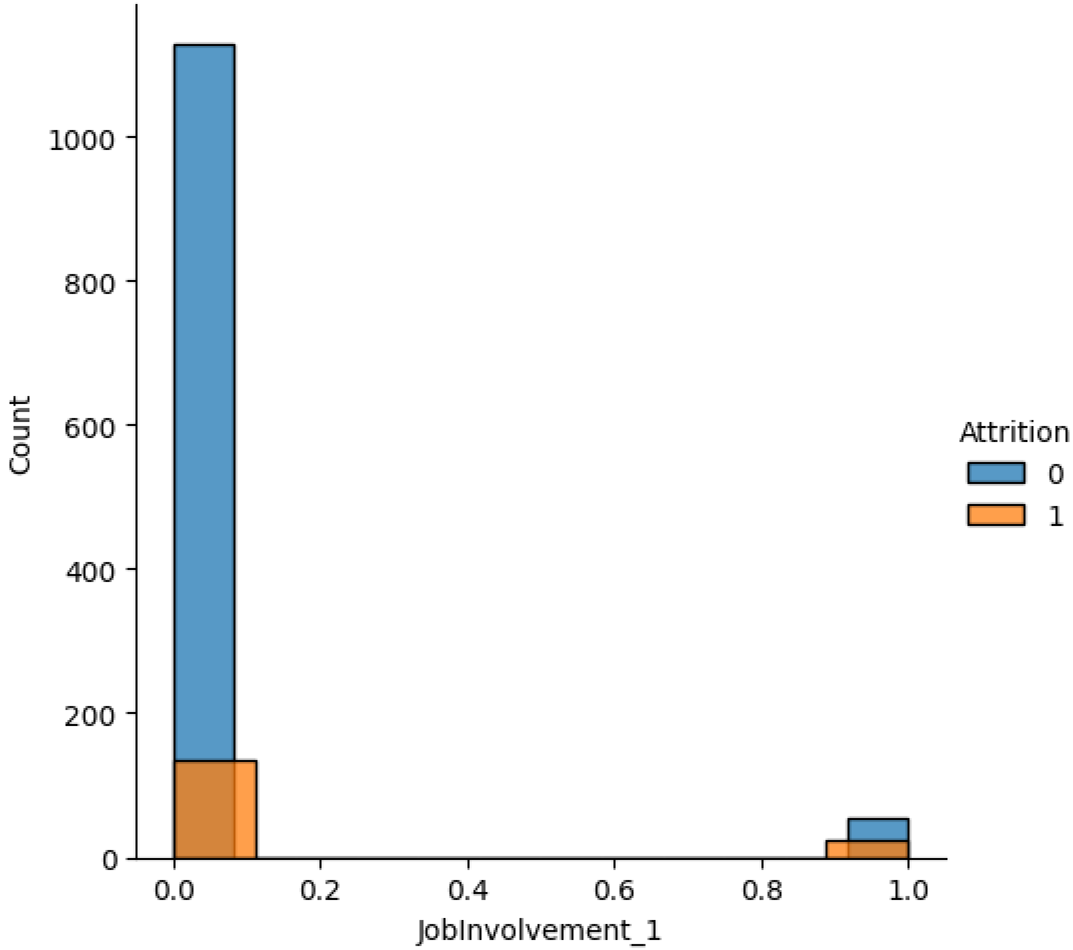


1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
EnvironmentSatisfaction_1
Attrition
0
1

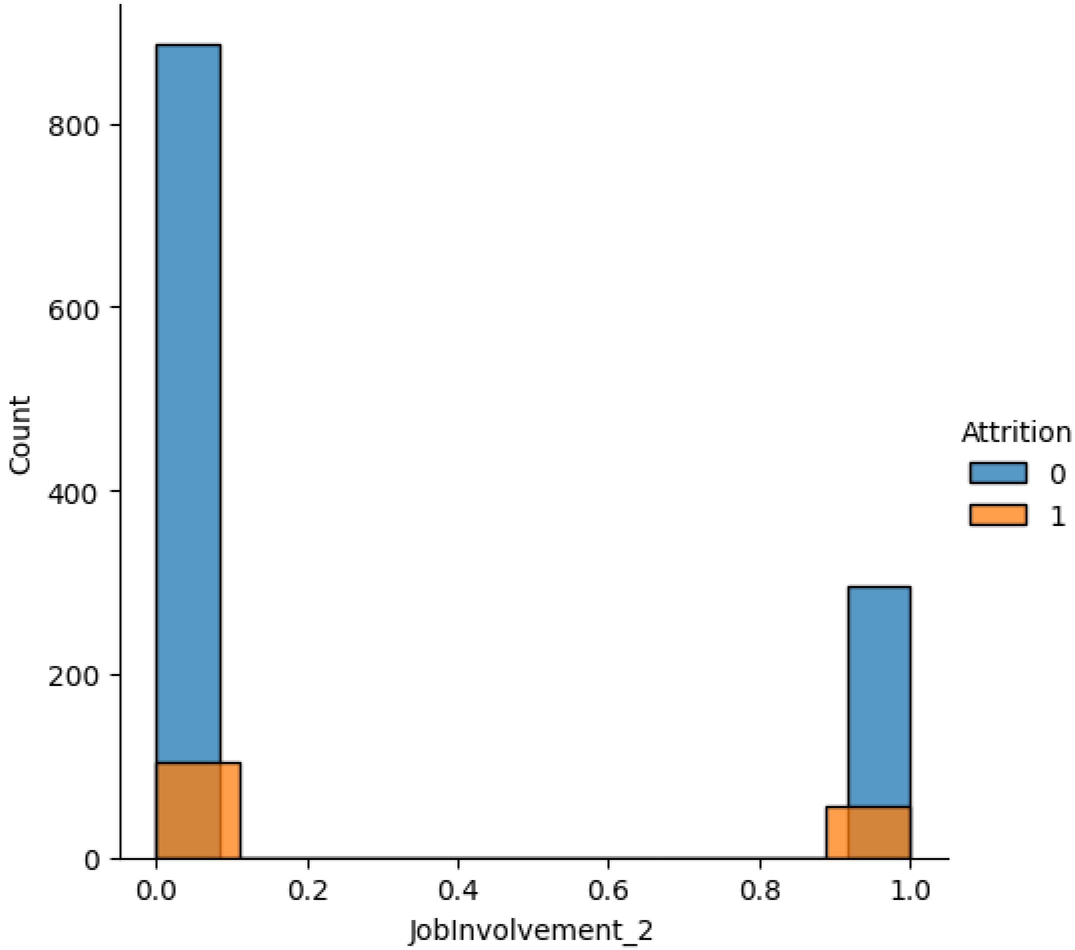

Count
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
EnvironmentSatisfaction_2
Attrition
0
1


800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
EnvironmentSatisfaction_3
Attrition
0
1

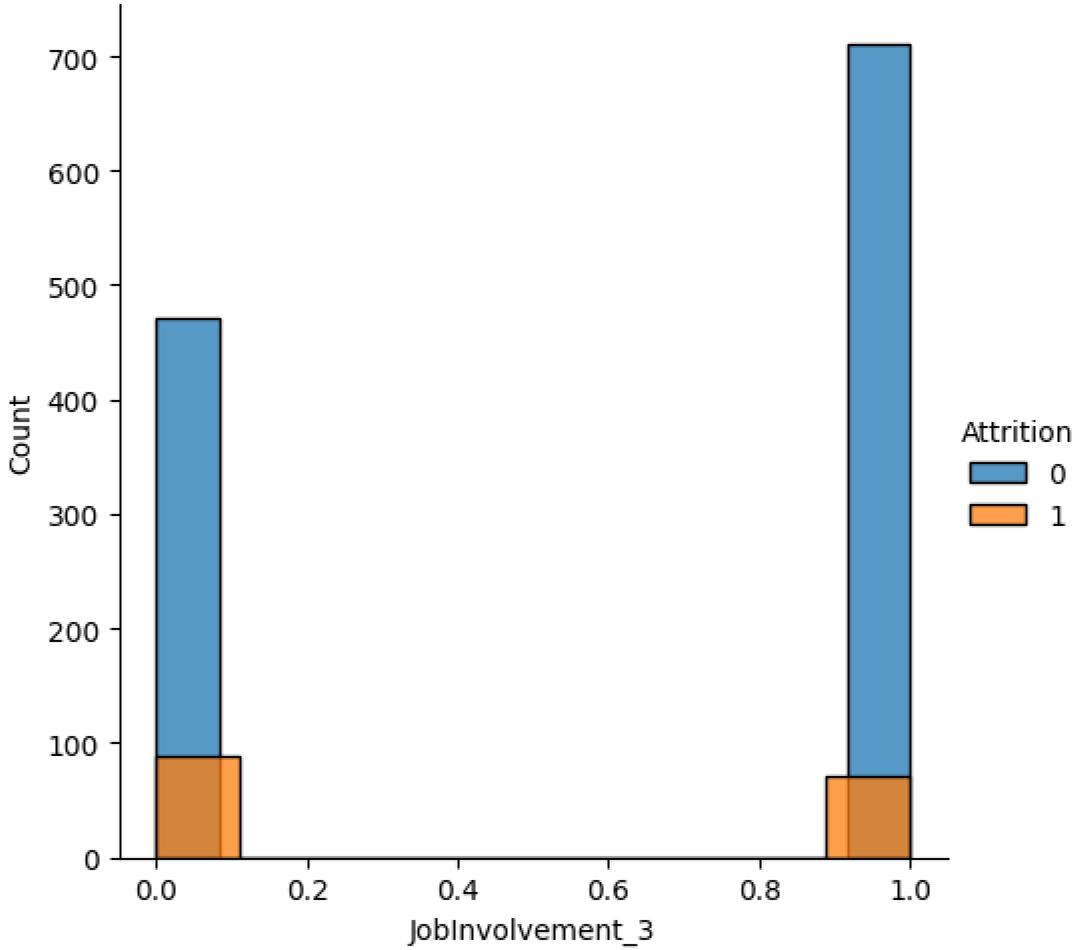

Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
EnvironmentSatisfaction_4
Attrition
0
1



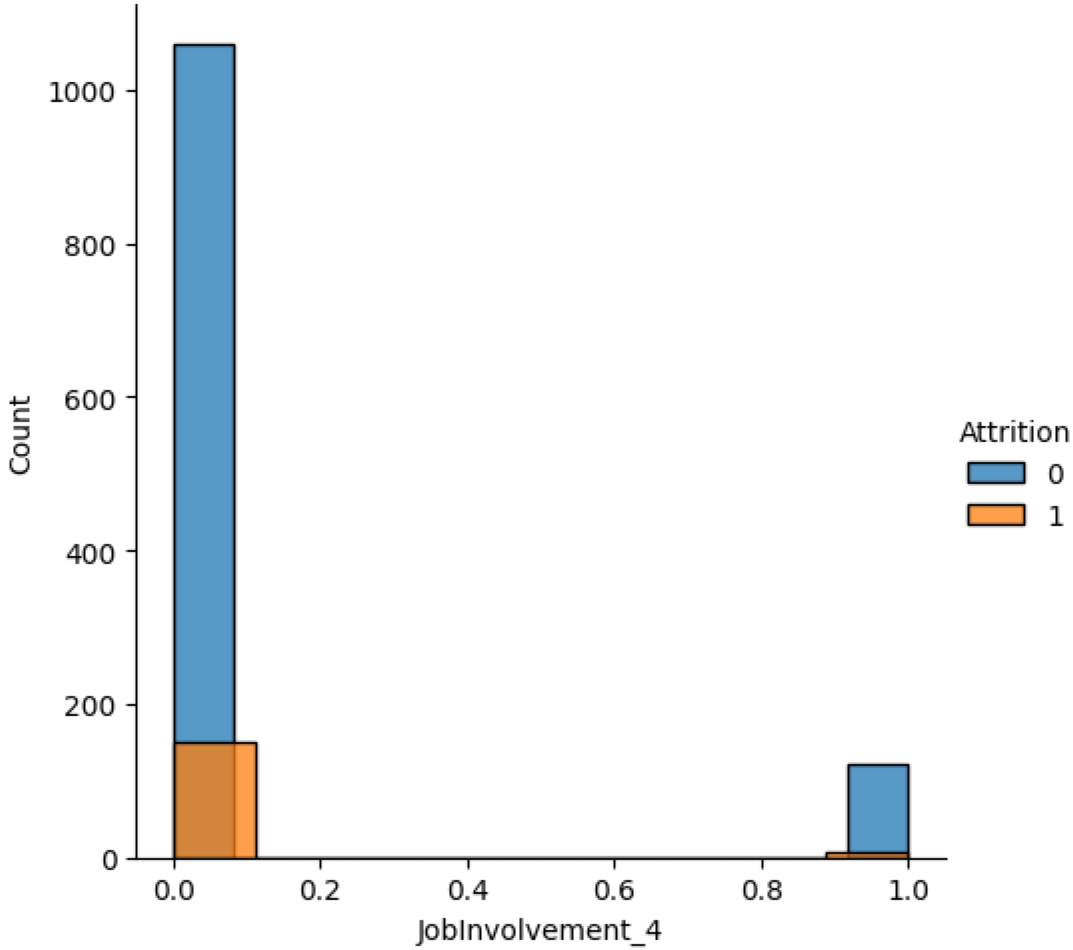
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobInvolvement_1
Attrition
0
1

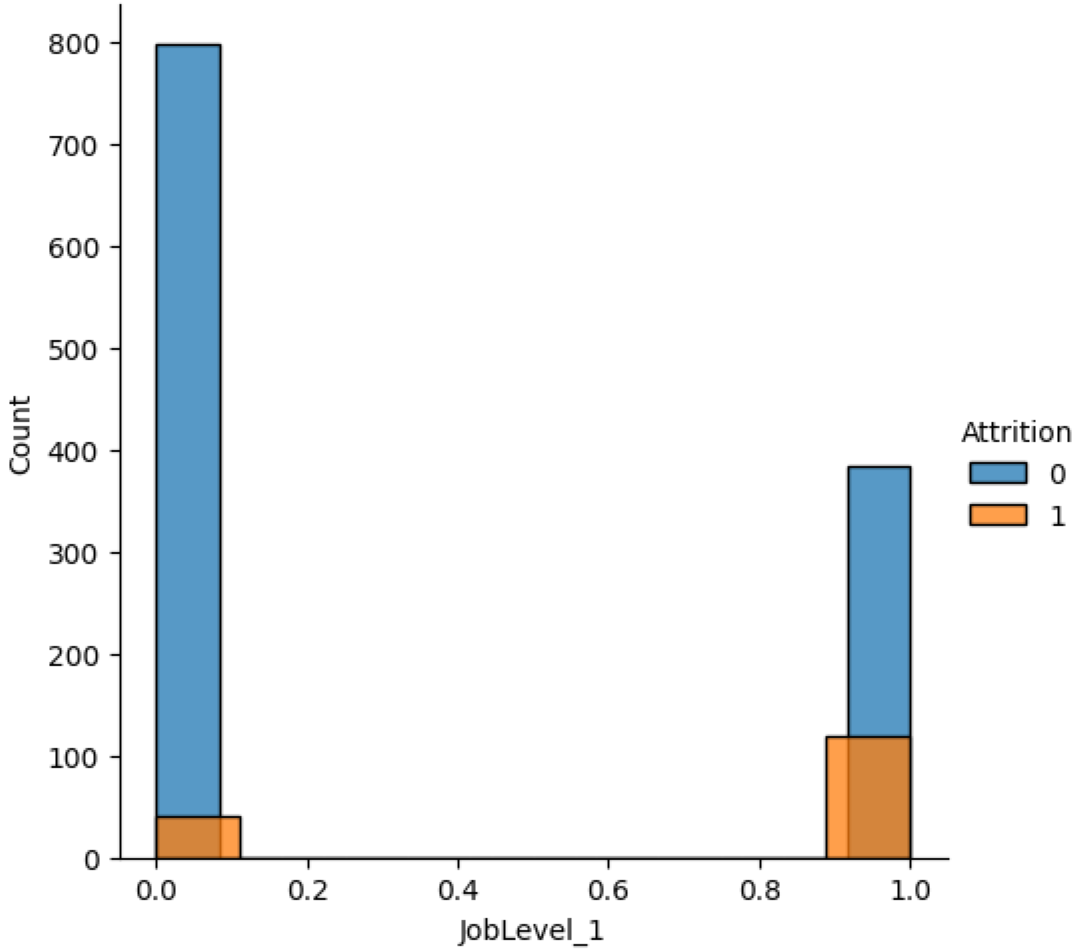


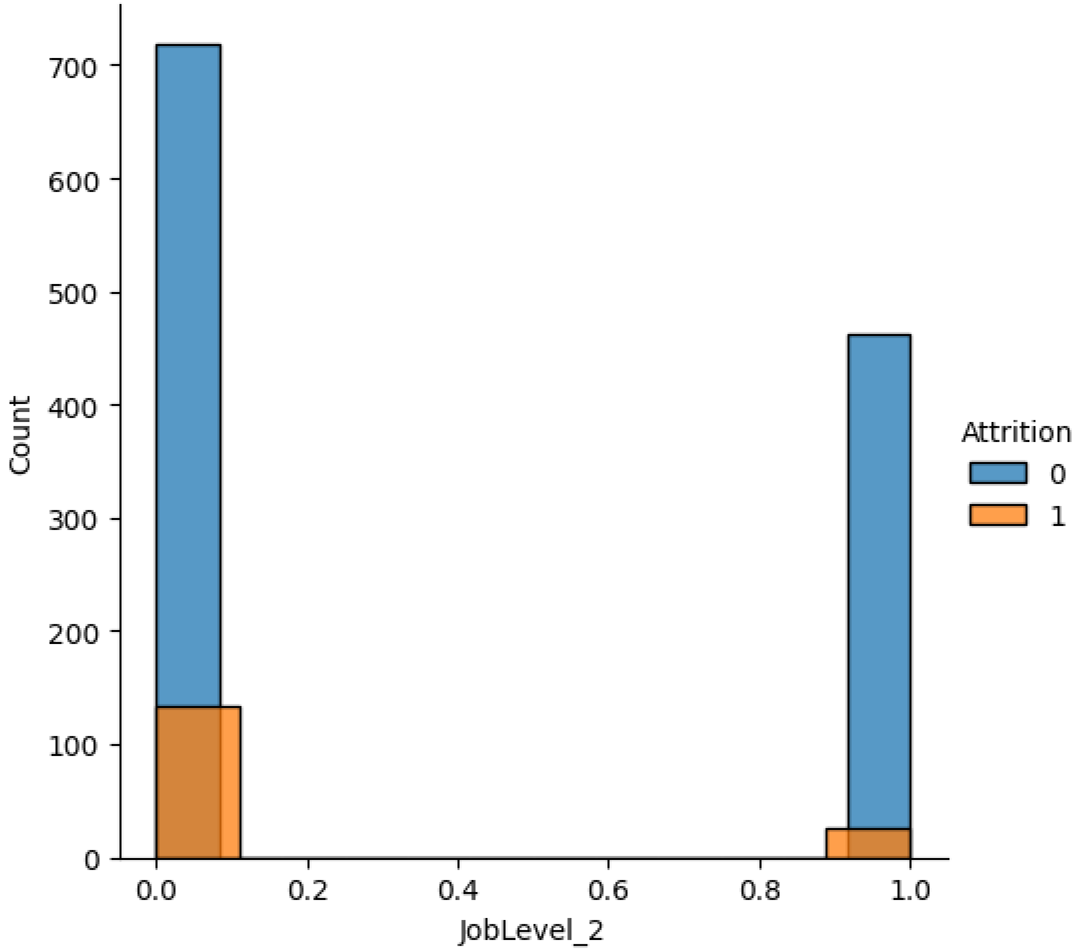
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobInvolvement_2
Attrition
0
1



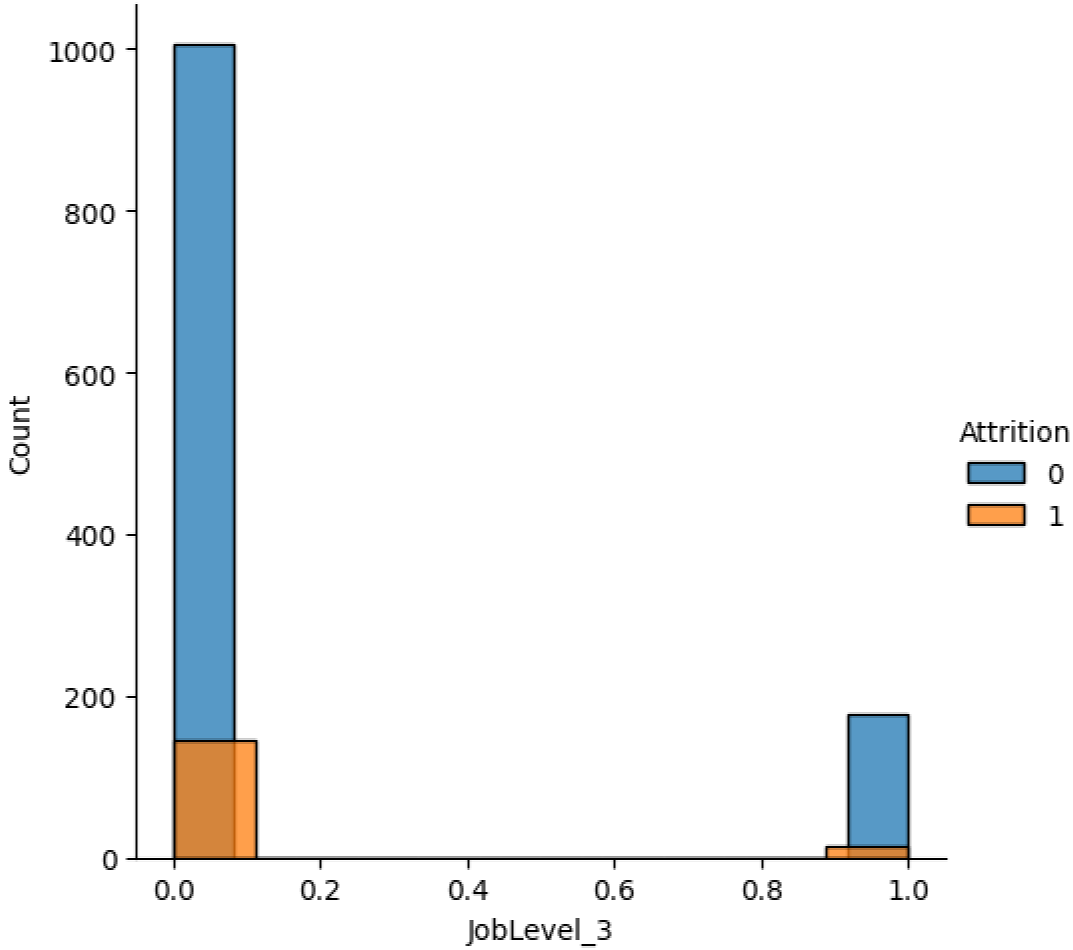
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobInvolvement_3
Attrition
0
1

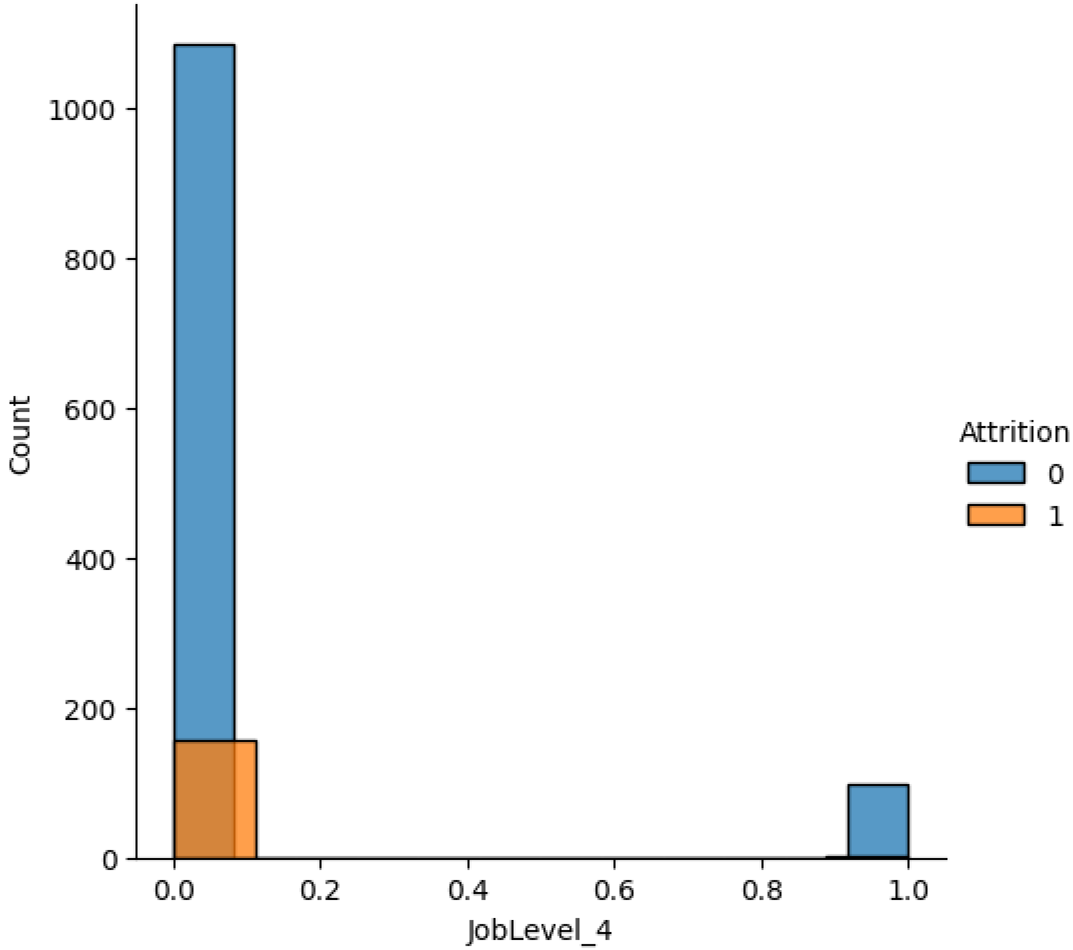

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
JobInvolvement_4
Attrition
0
1

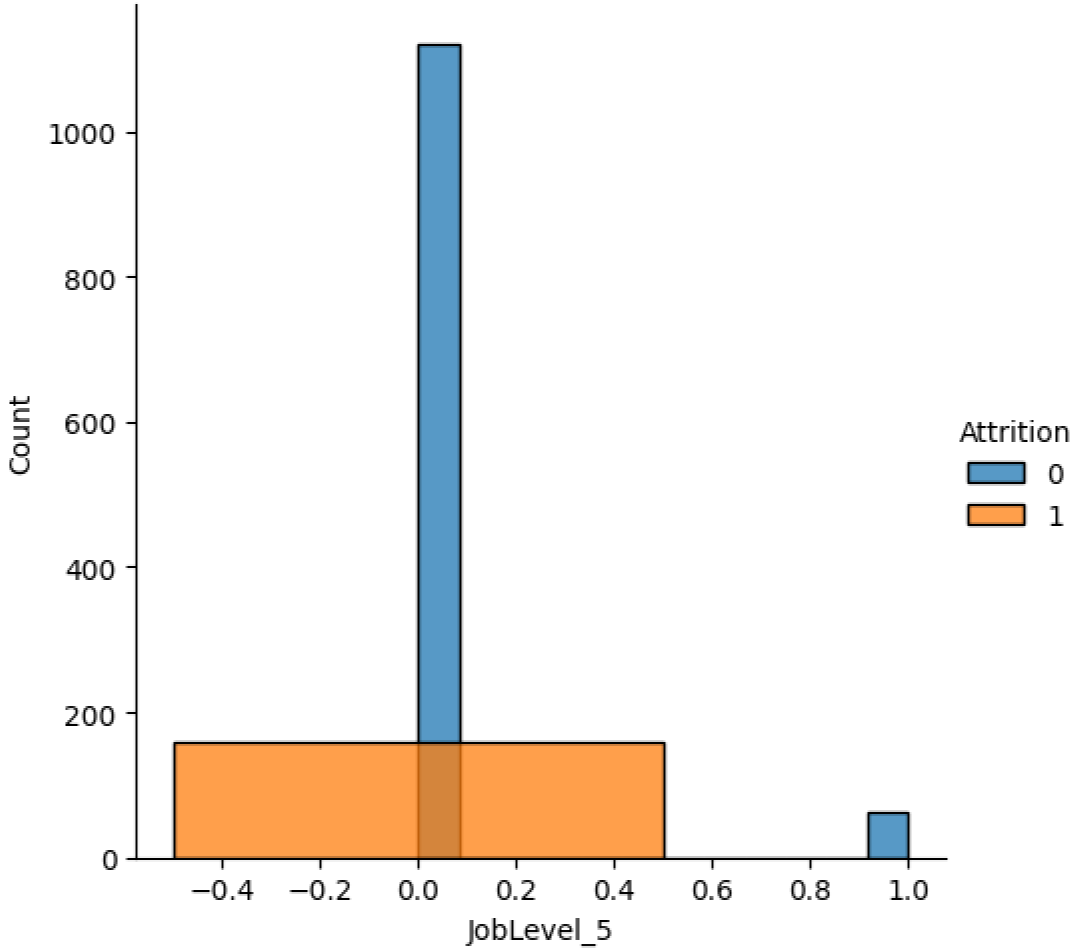

800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_1
Attrition
0
1



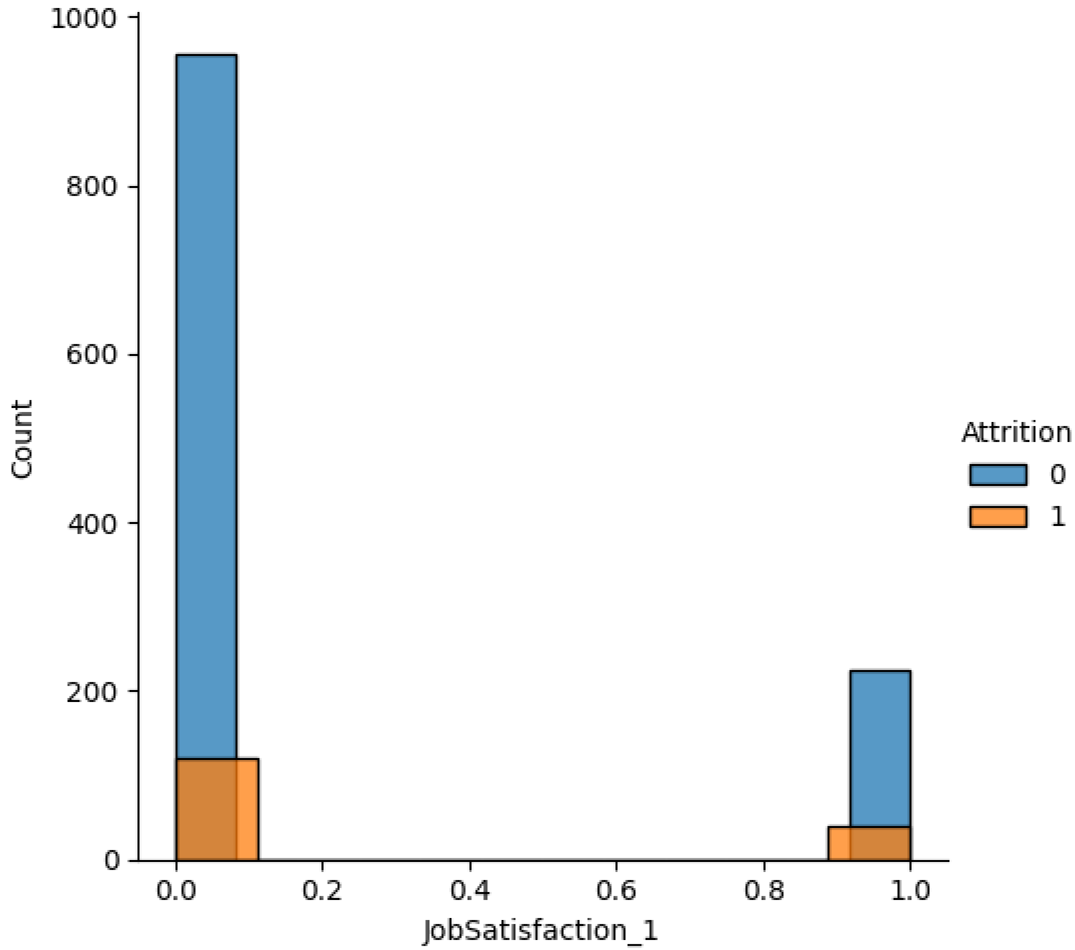
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_2
Attrition
0
1

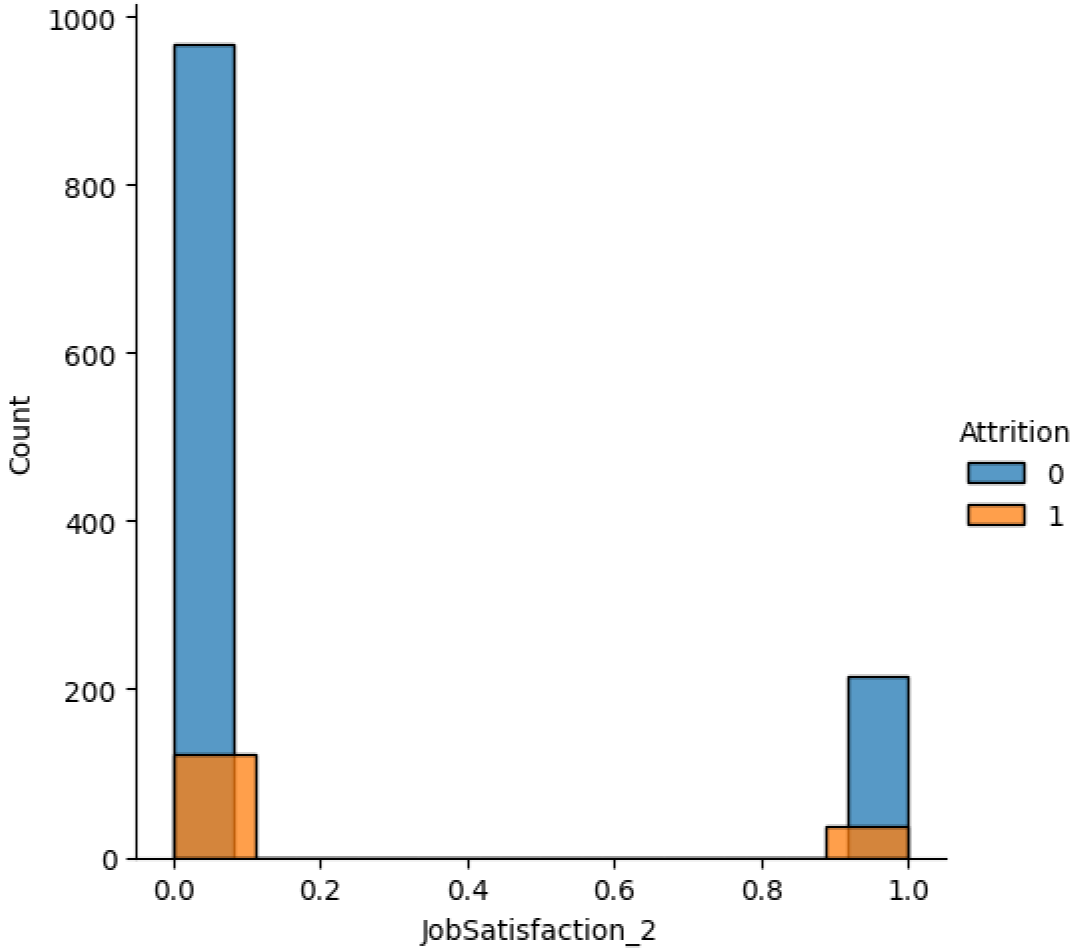

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_3
Attrition
0
1

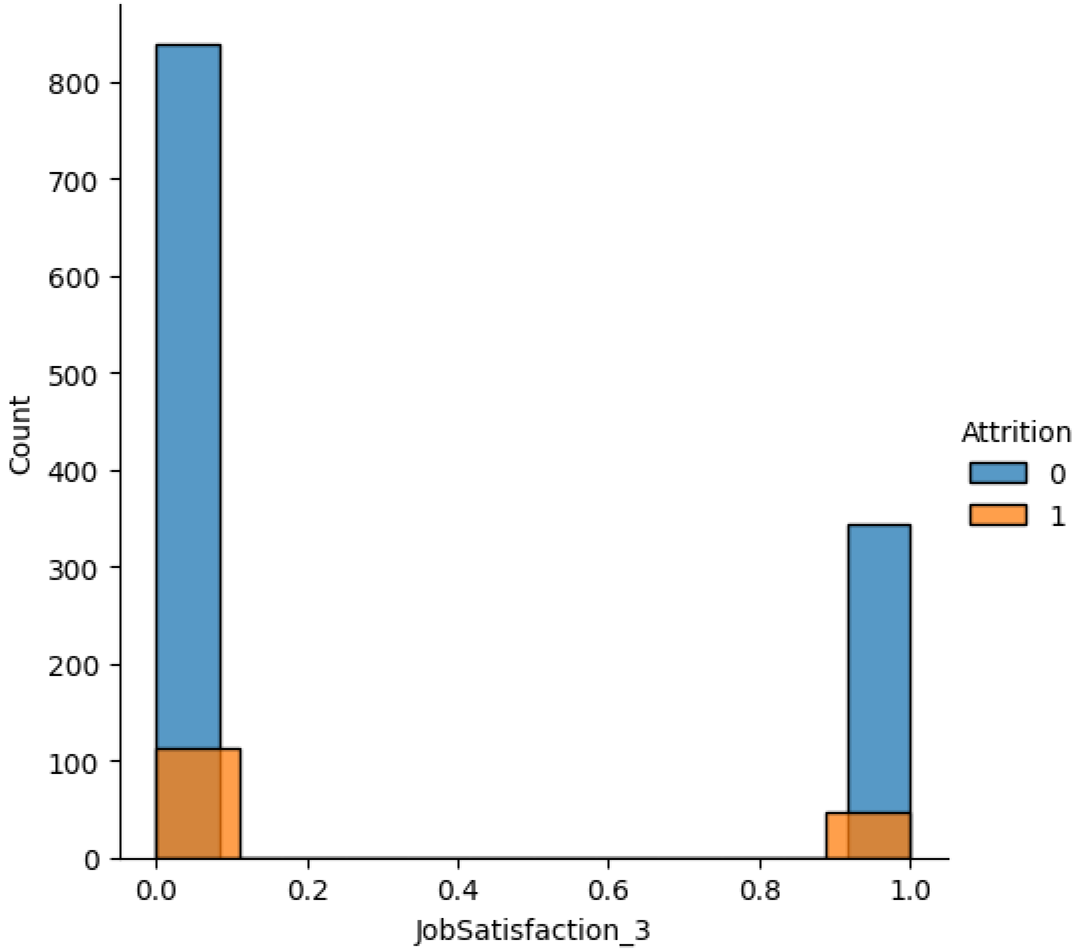

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_4
Attrition
0
1

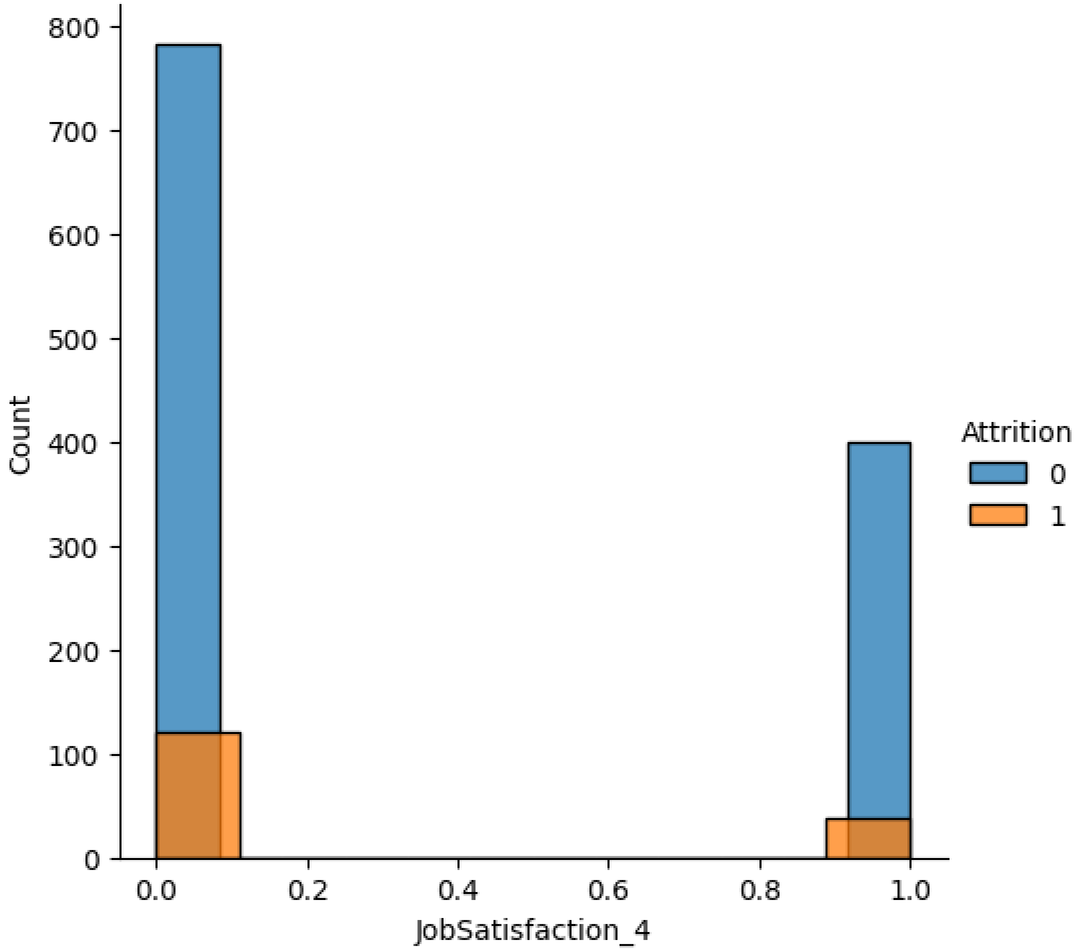


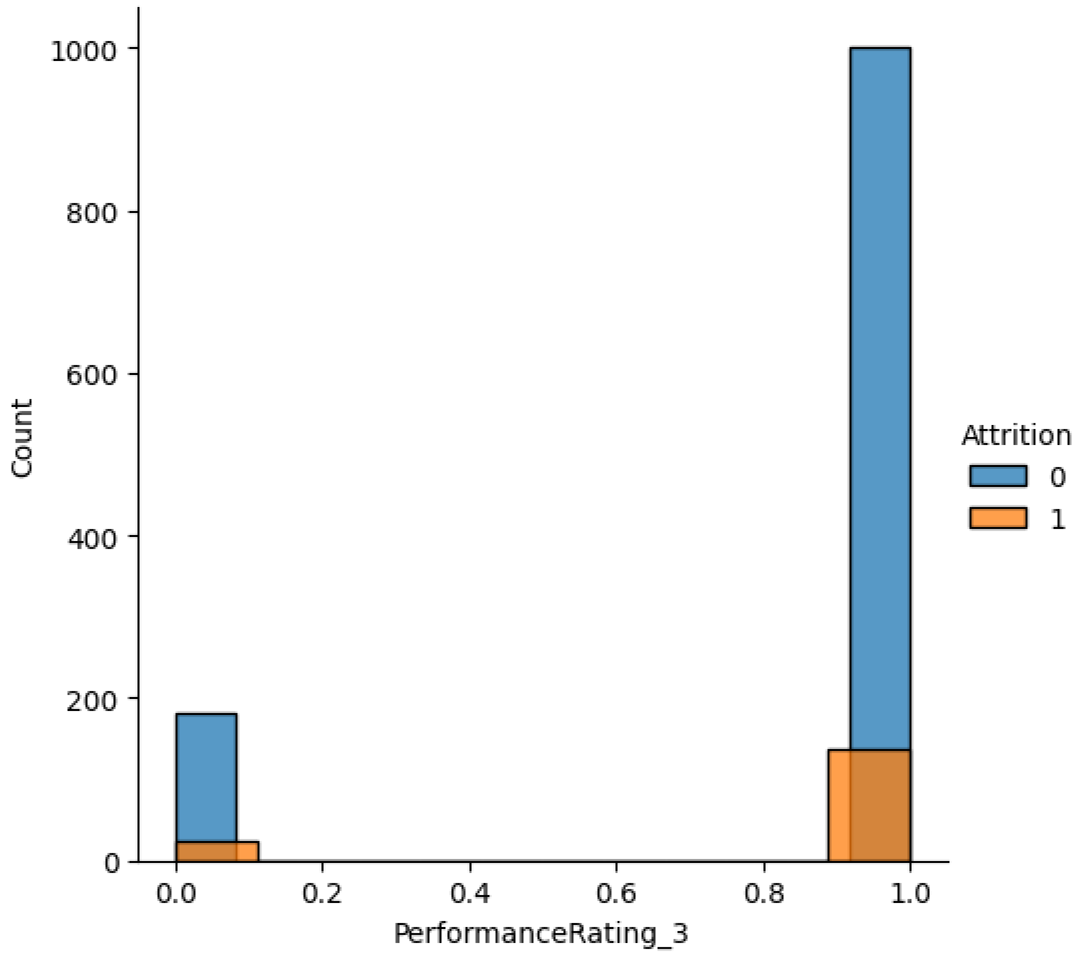
Count
1000
800
600
400
200
0
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_5
Attrition
0
1

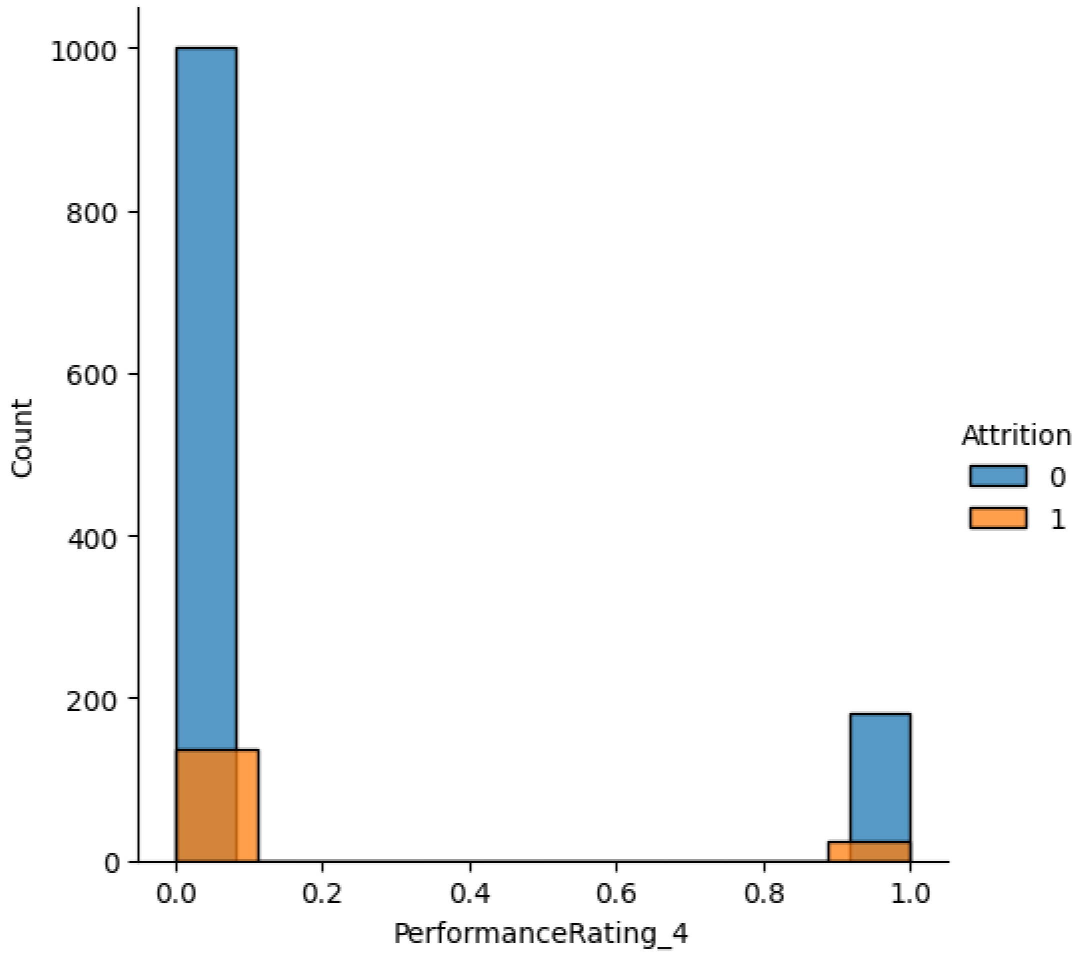

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobSatisfaction_1
Attrition
0
1

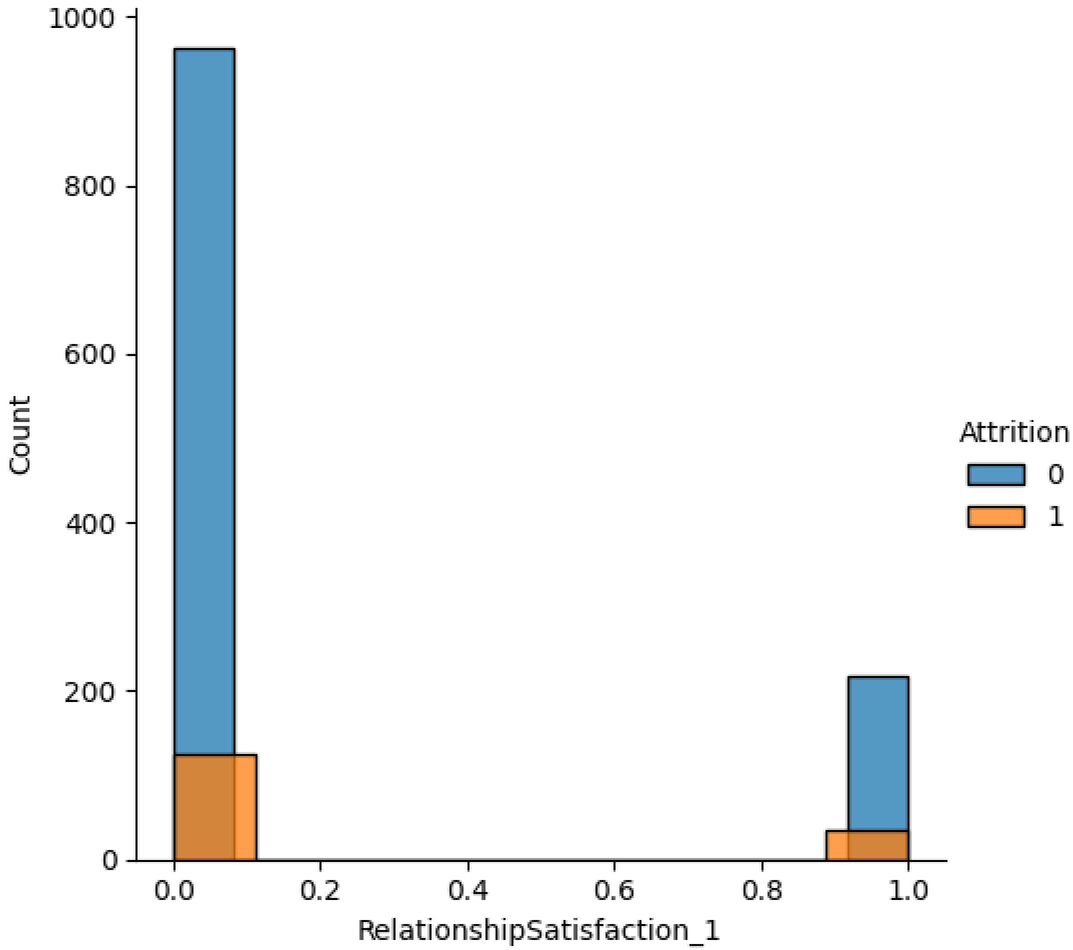

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobSatisfaction_2
Attrition
0
1


Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
JobSatisfaction_3
Attrition
0
1

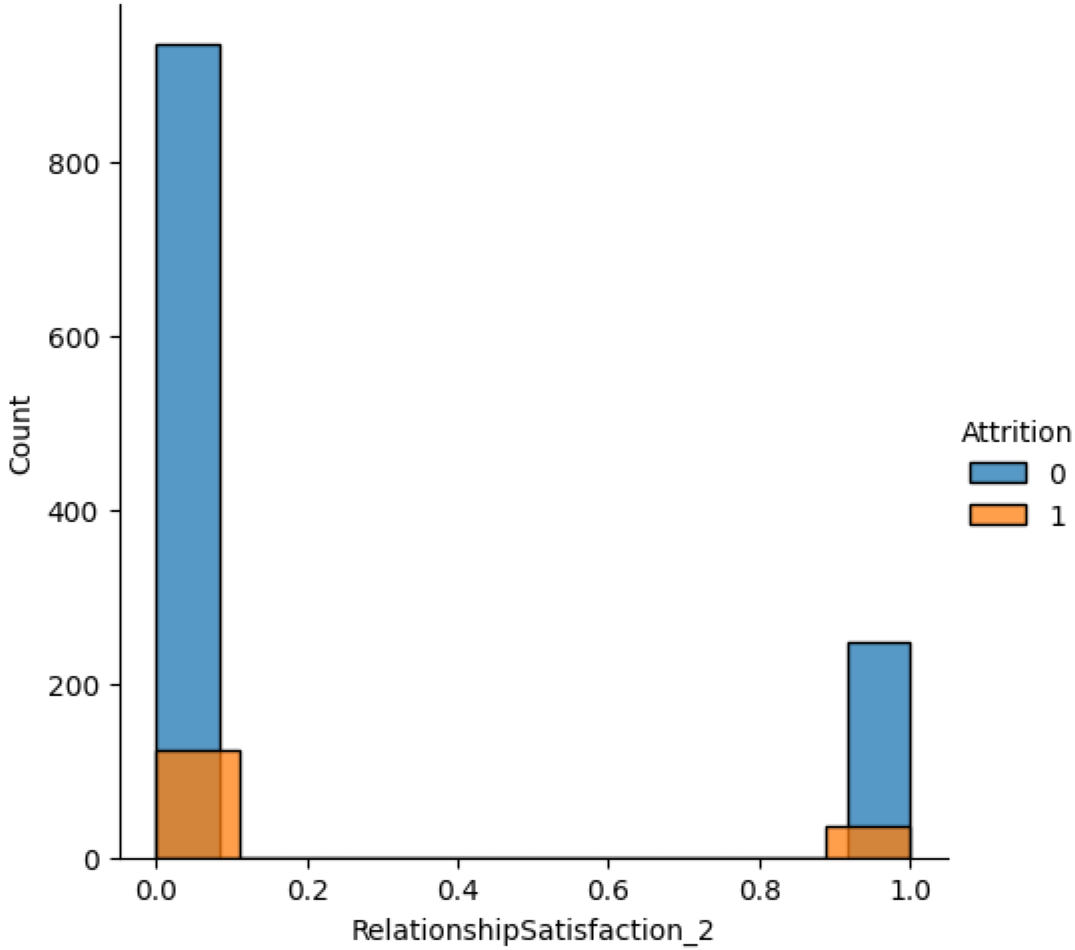

800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobSatisfaction_4
Attrition
0
1

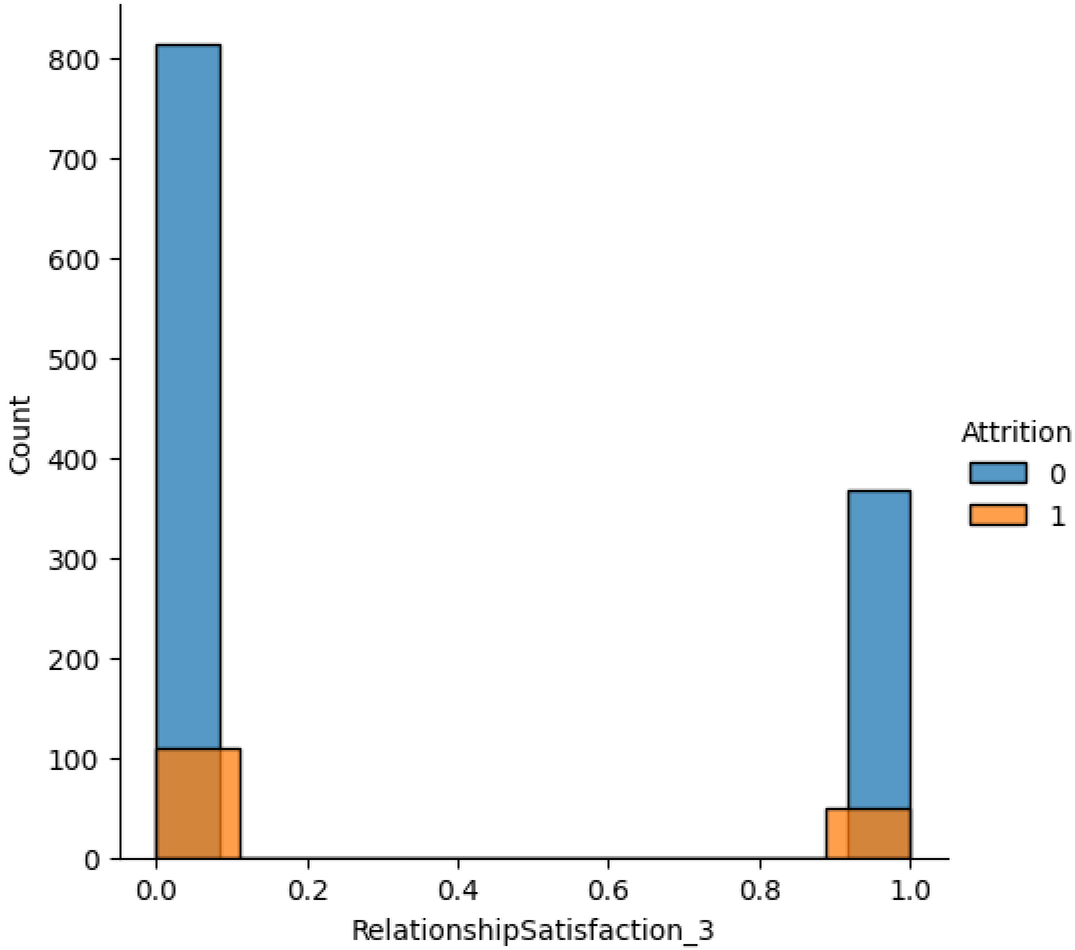

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
PerformanceRating_3
Attrition
0
1

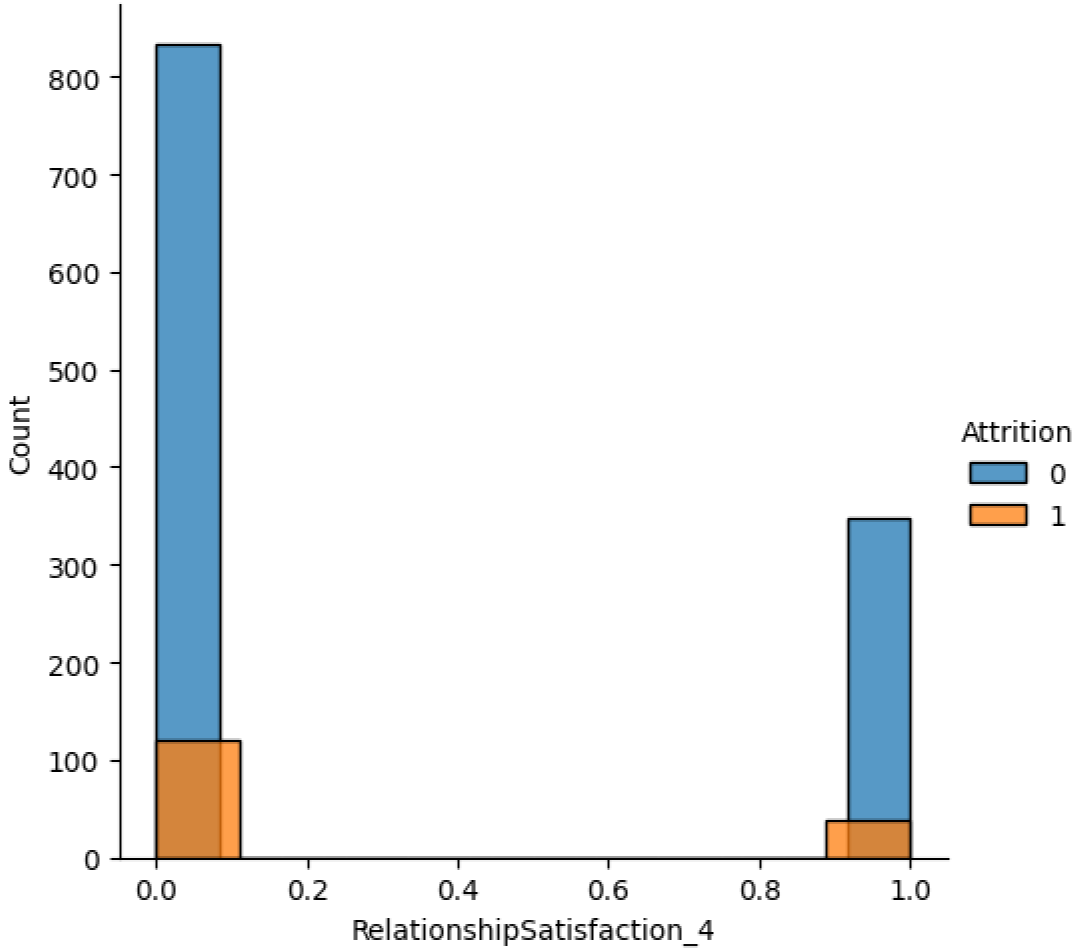

1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
PerformanceRating_4
Attrition
0
1



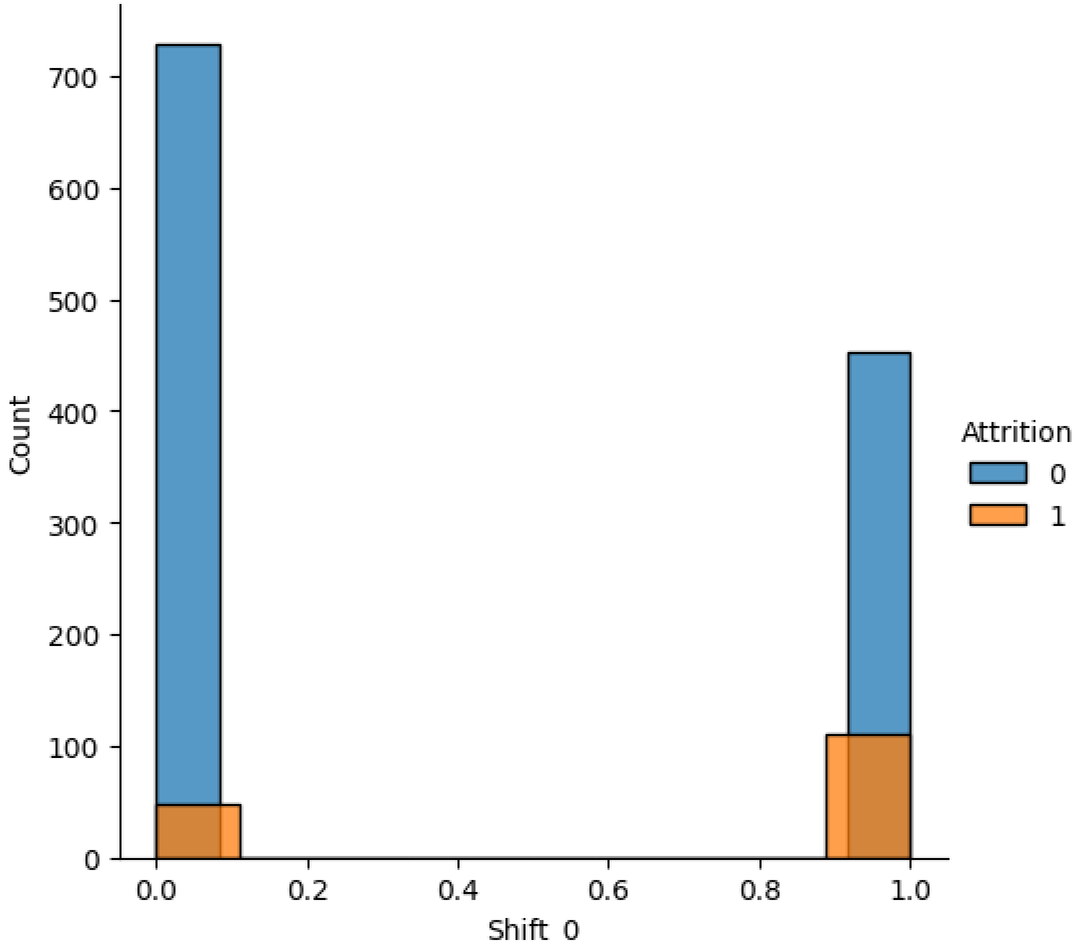
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
RelationshipSatisfaction_1
Attrition
0
1


Count
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
RelationshipSatisfaction_2
Attrition
0
1

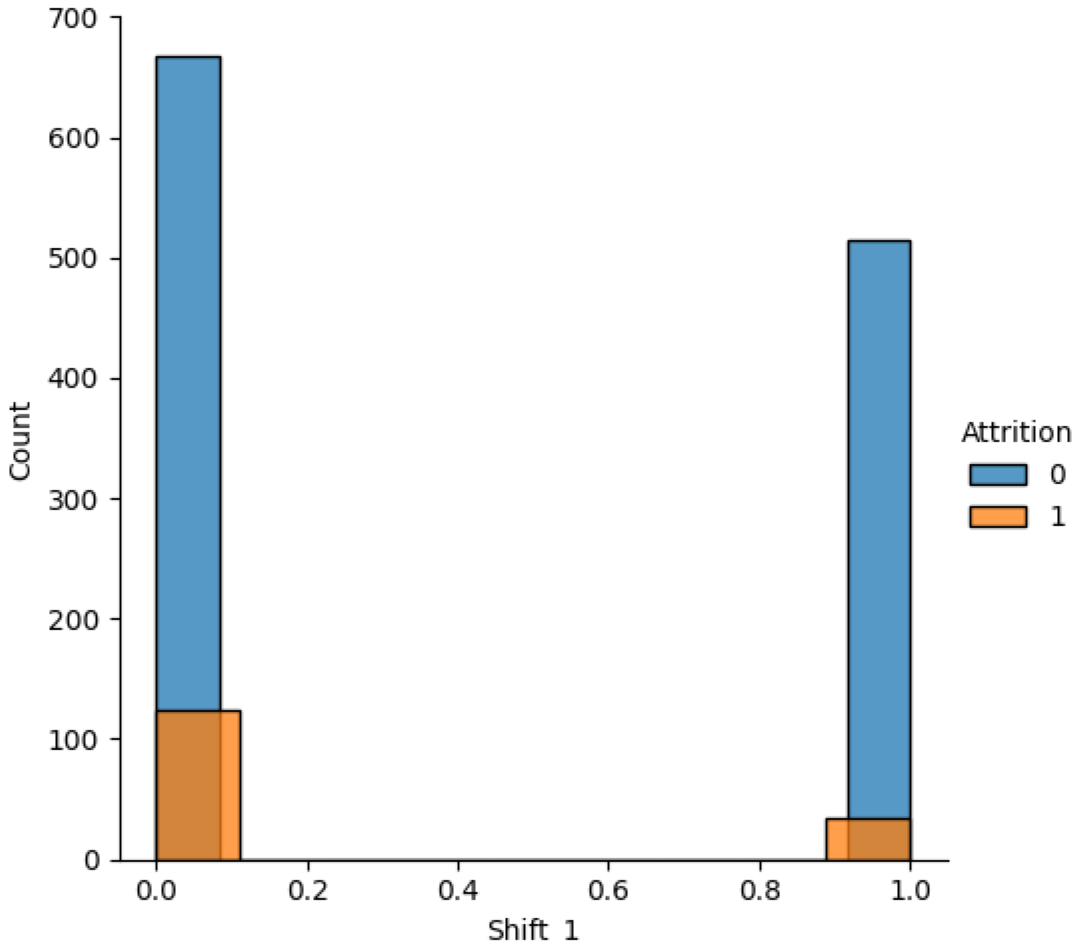

Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
RelationshipSatisfaction_3
Attrition
0
1



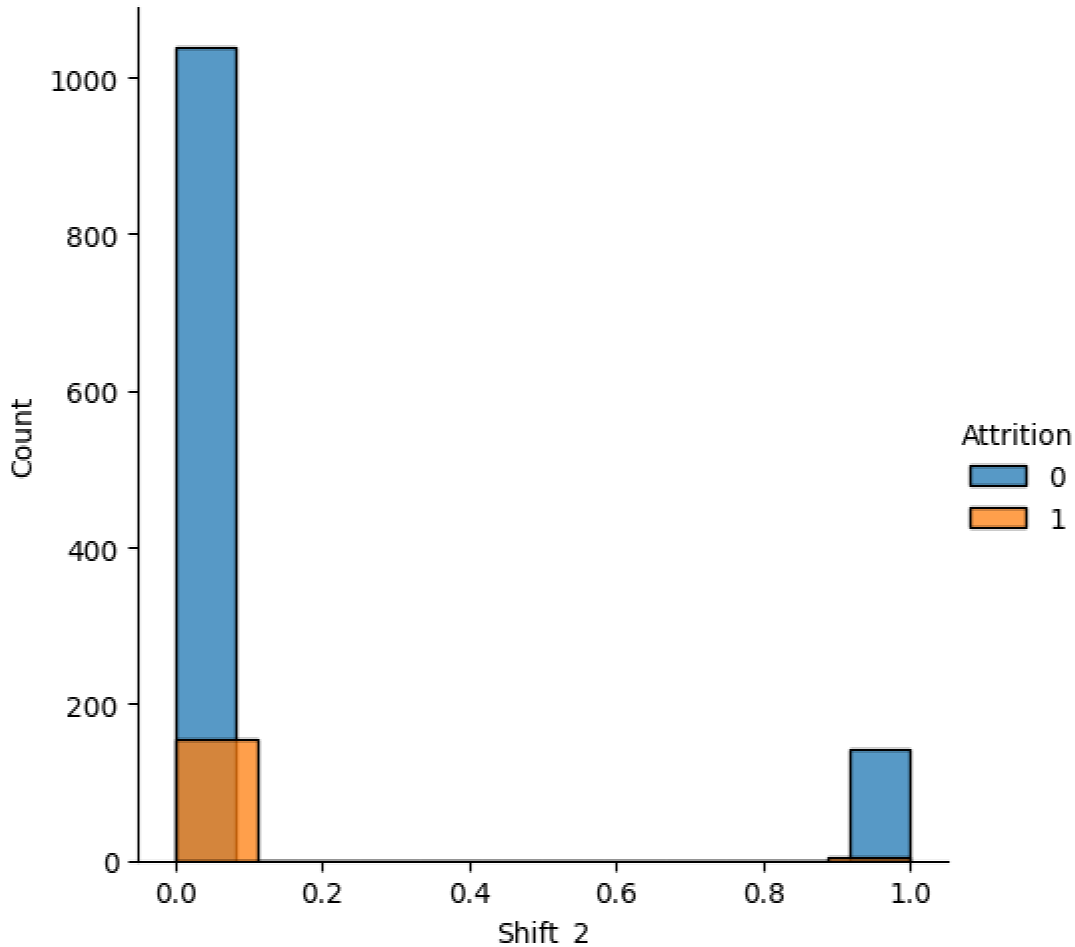
Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
RelationshipSatisfaction_4
Attrition
0
1



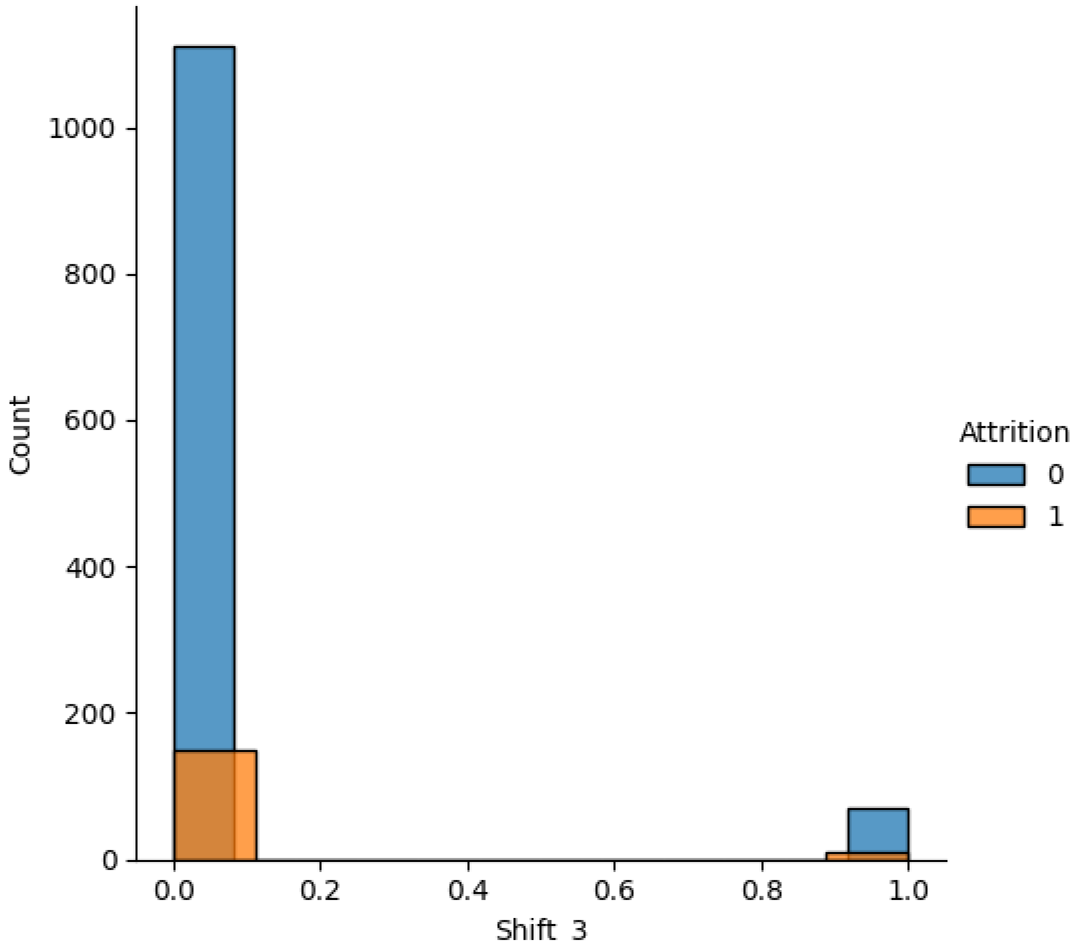
Count
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
Shift_0
Attrition
0
1

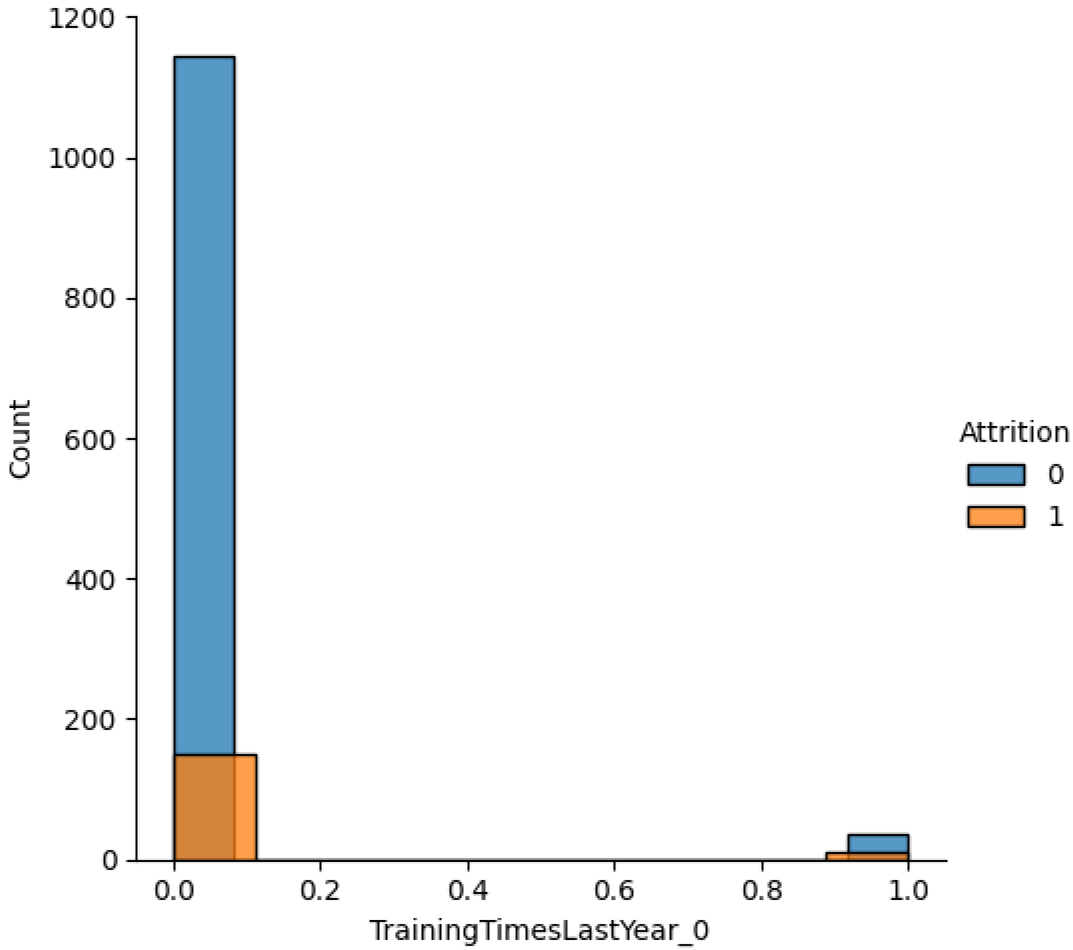


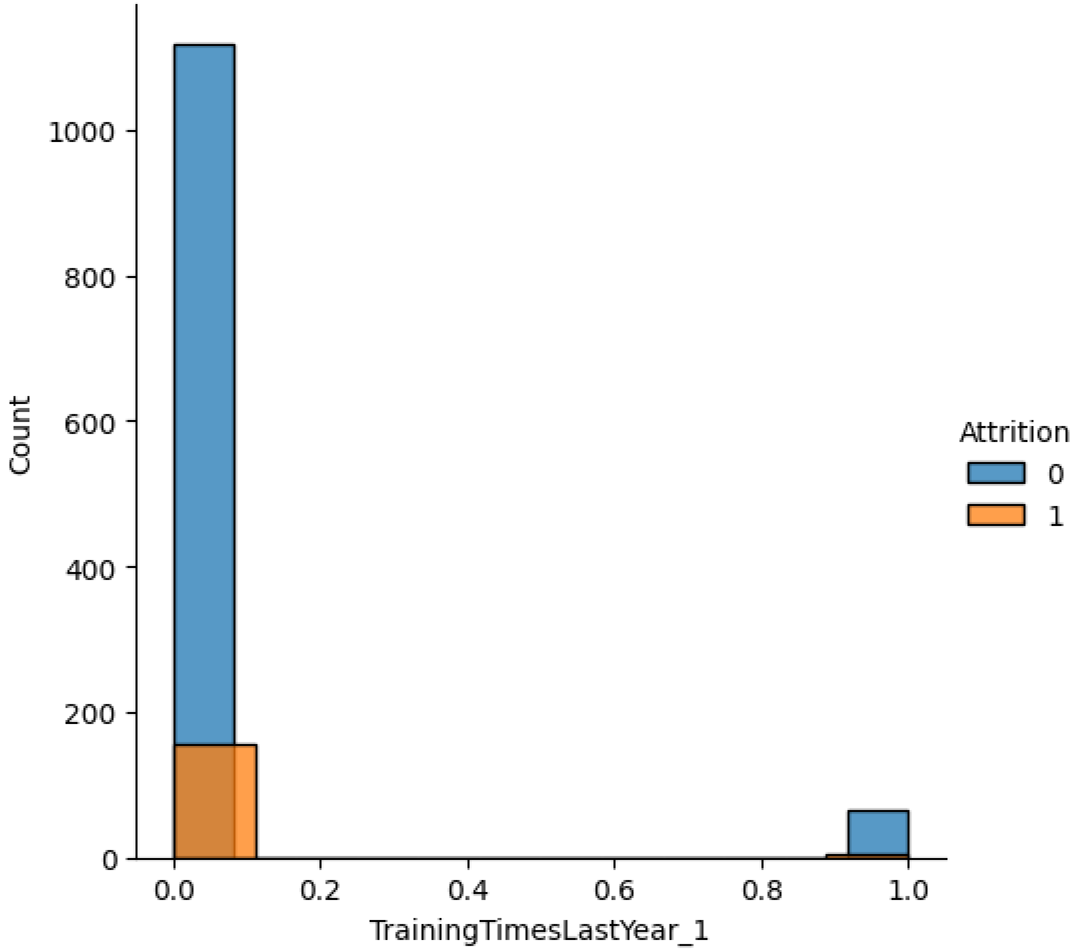
Count
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
Shift_1
Attrition
0
1

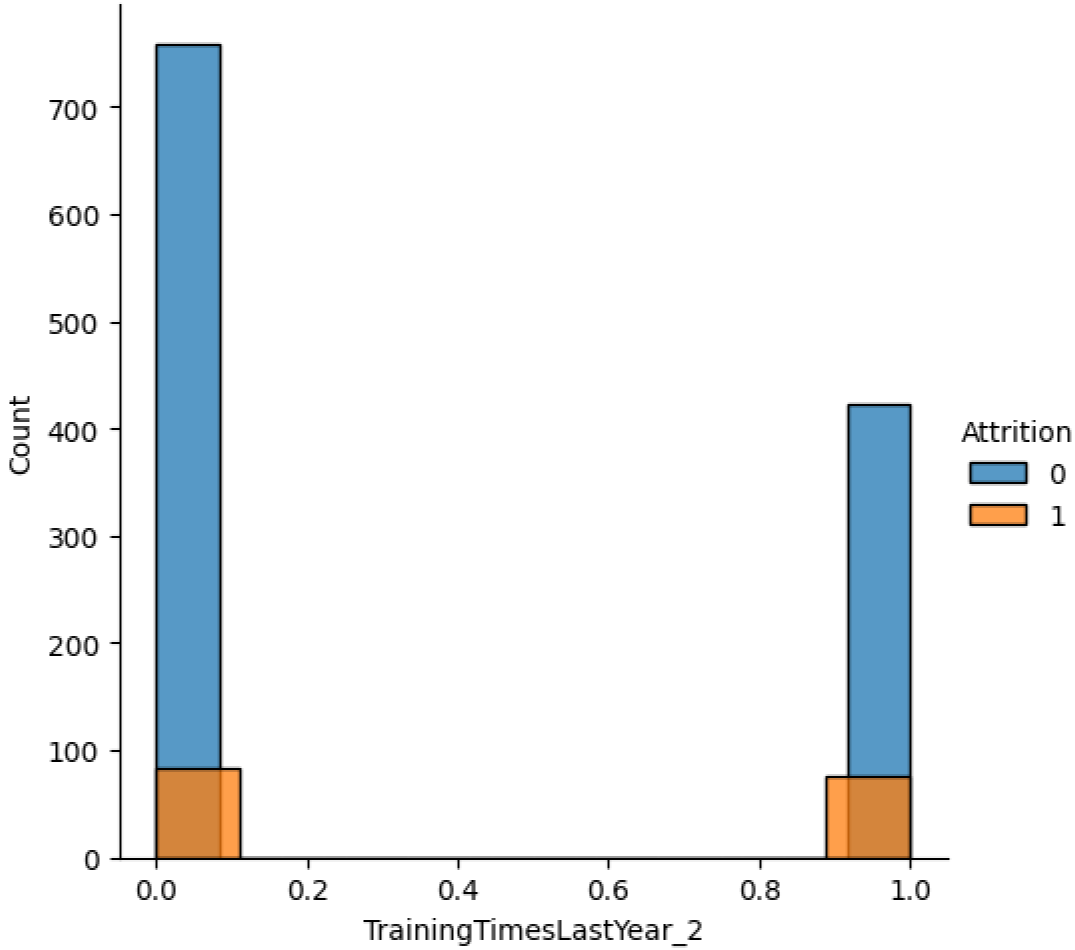


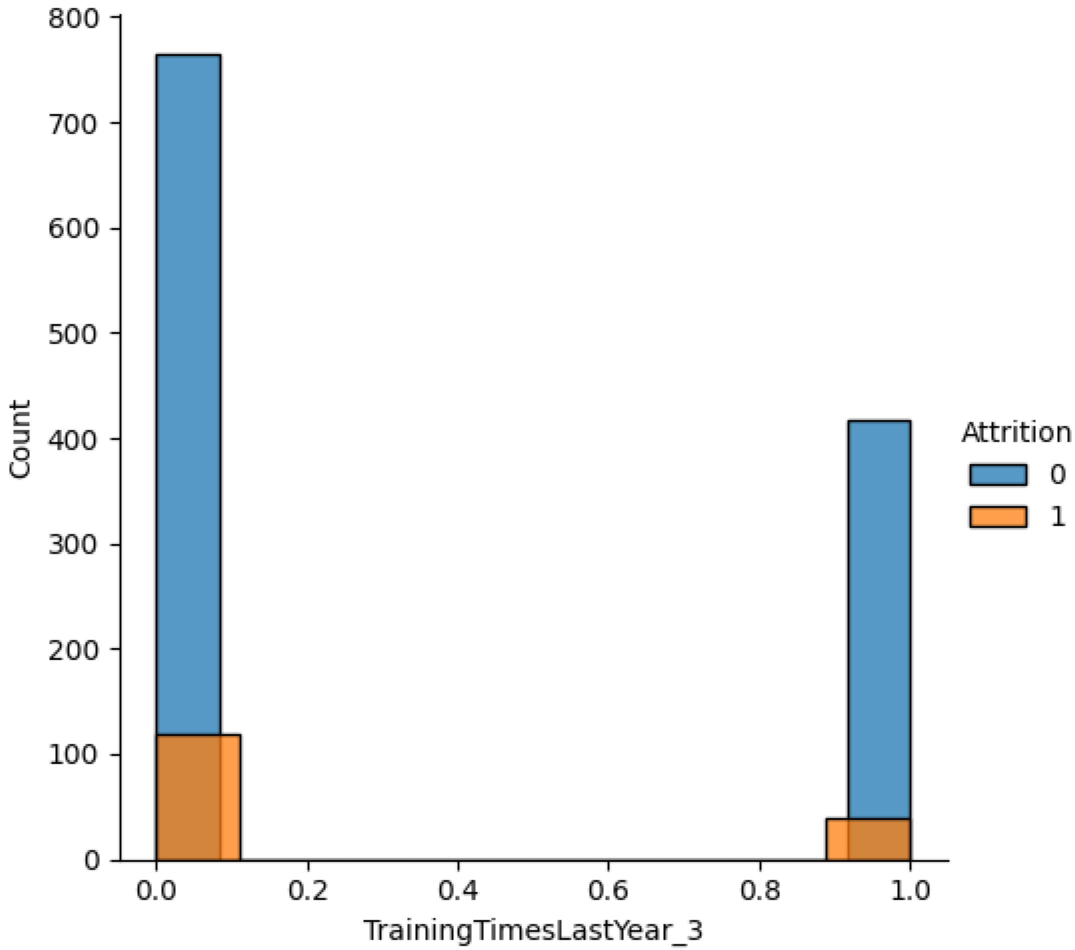
Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
Shift_2
Attrition
0
1


Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
Shift_3
Attrition
0
1

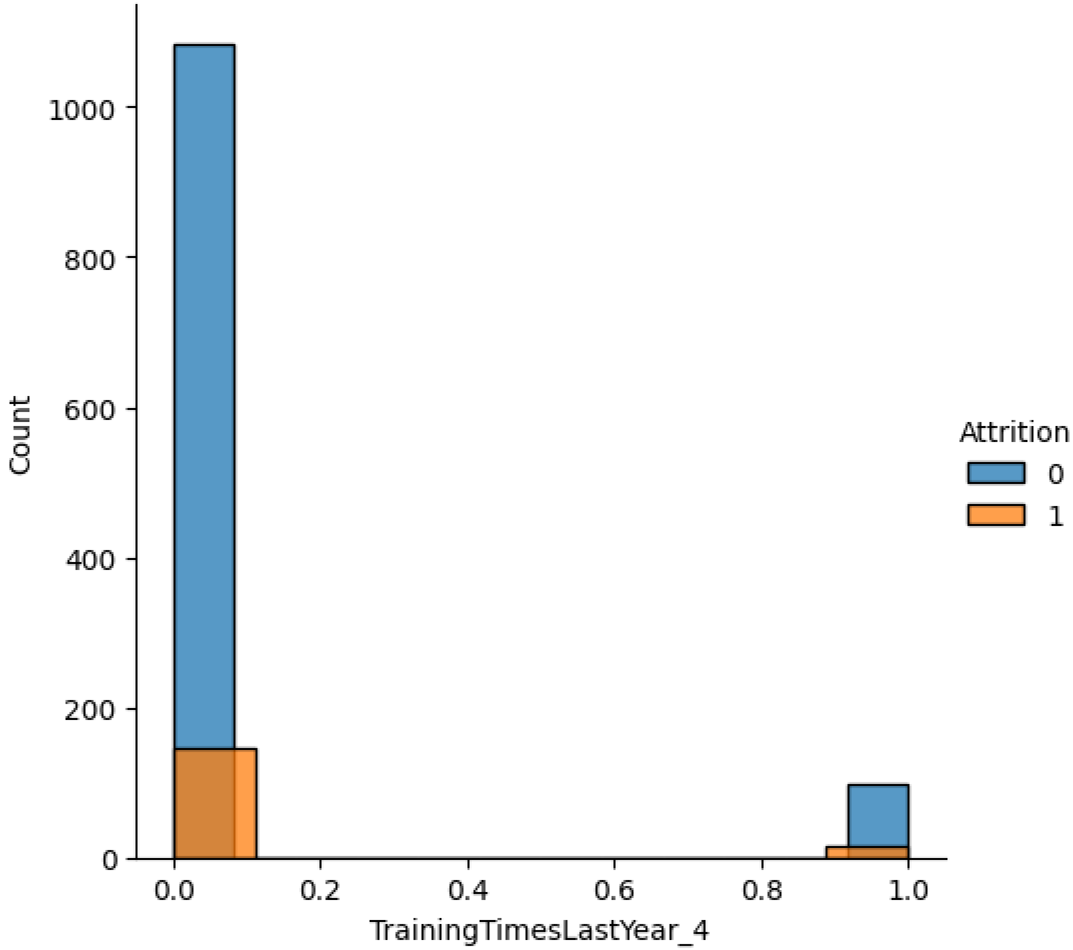

1200
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_0
Attrition
0
1

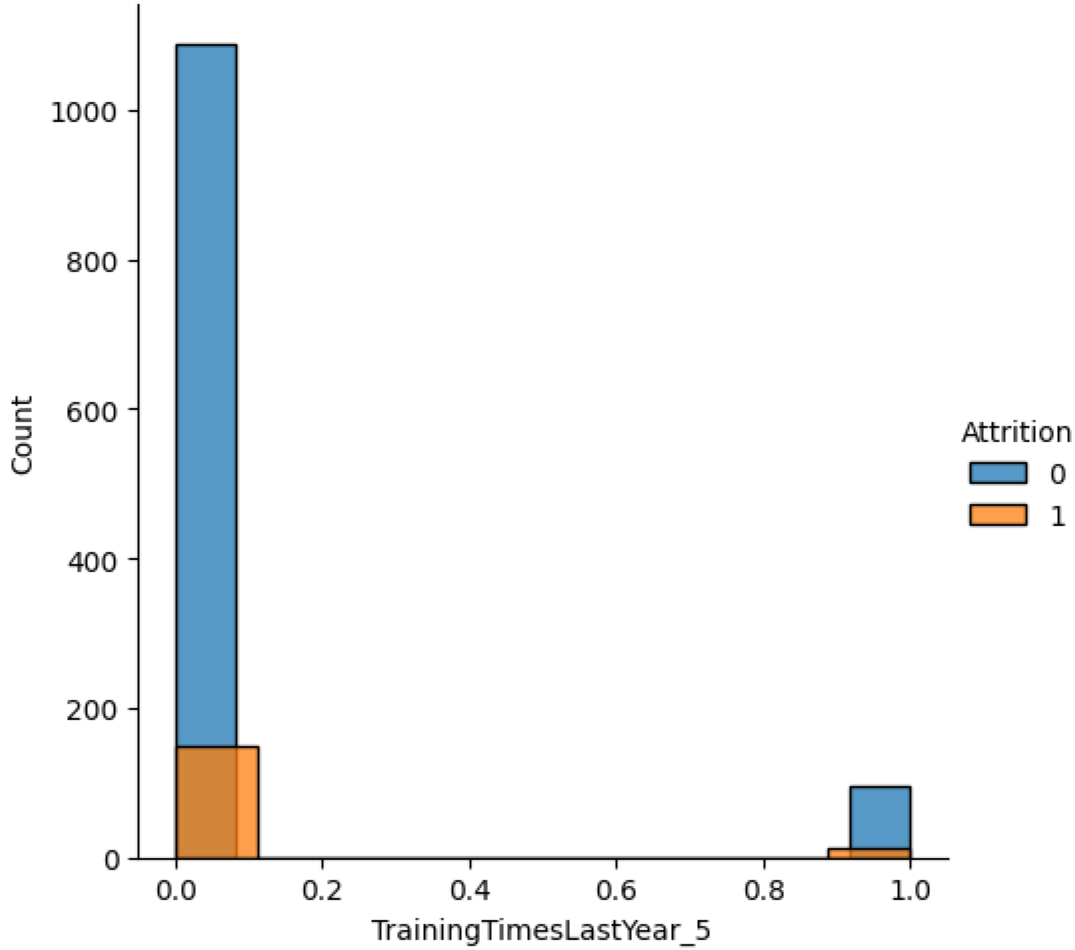

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_1
Attrition
0
1


700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_2
Attrition
0
1

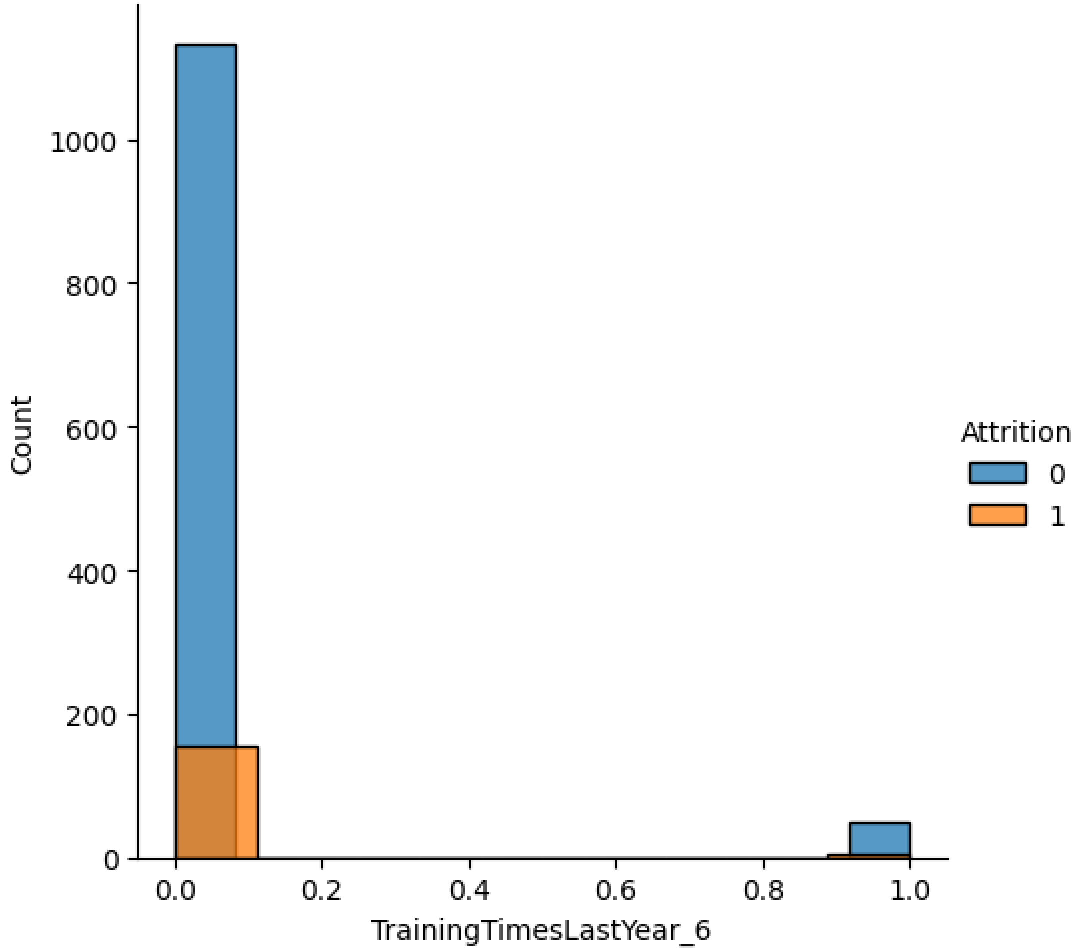


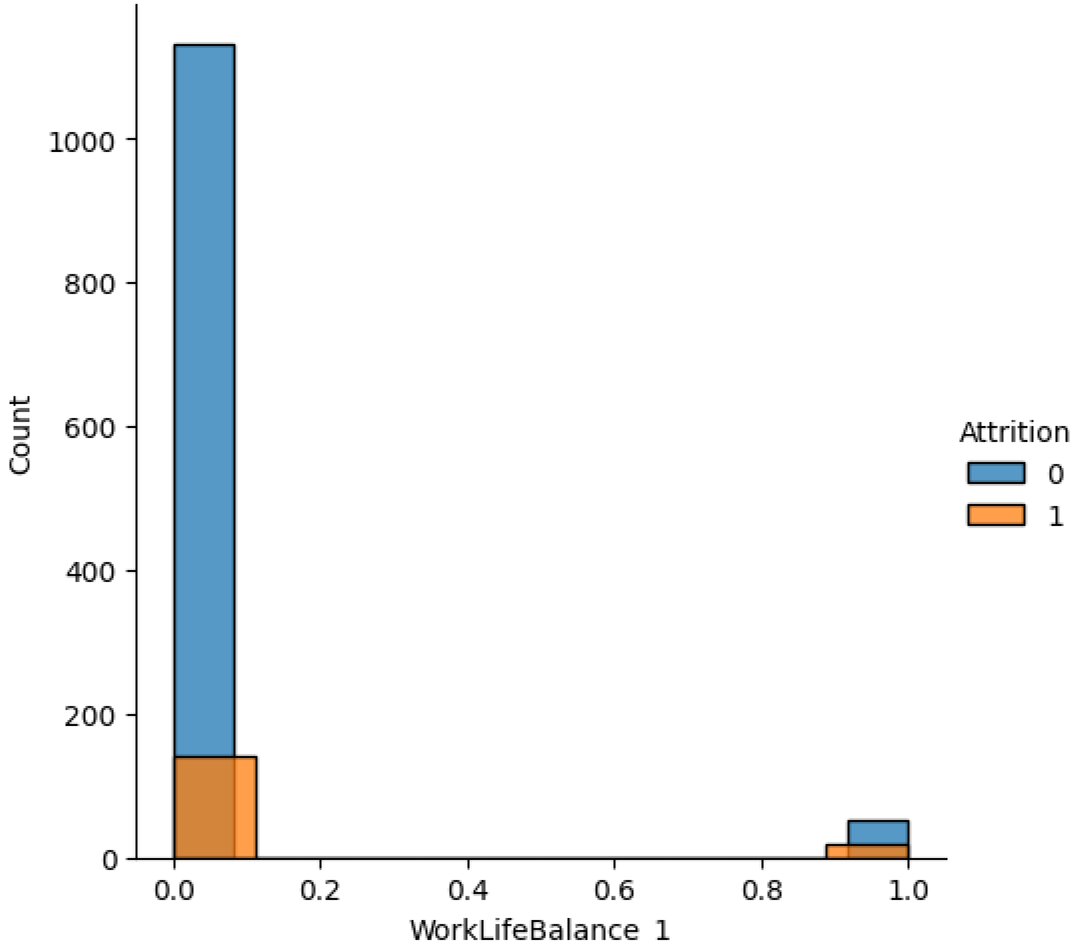
800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_3
Attrition
0
1


1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_4
Attrition
0
1

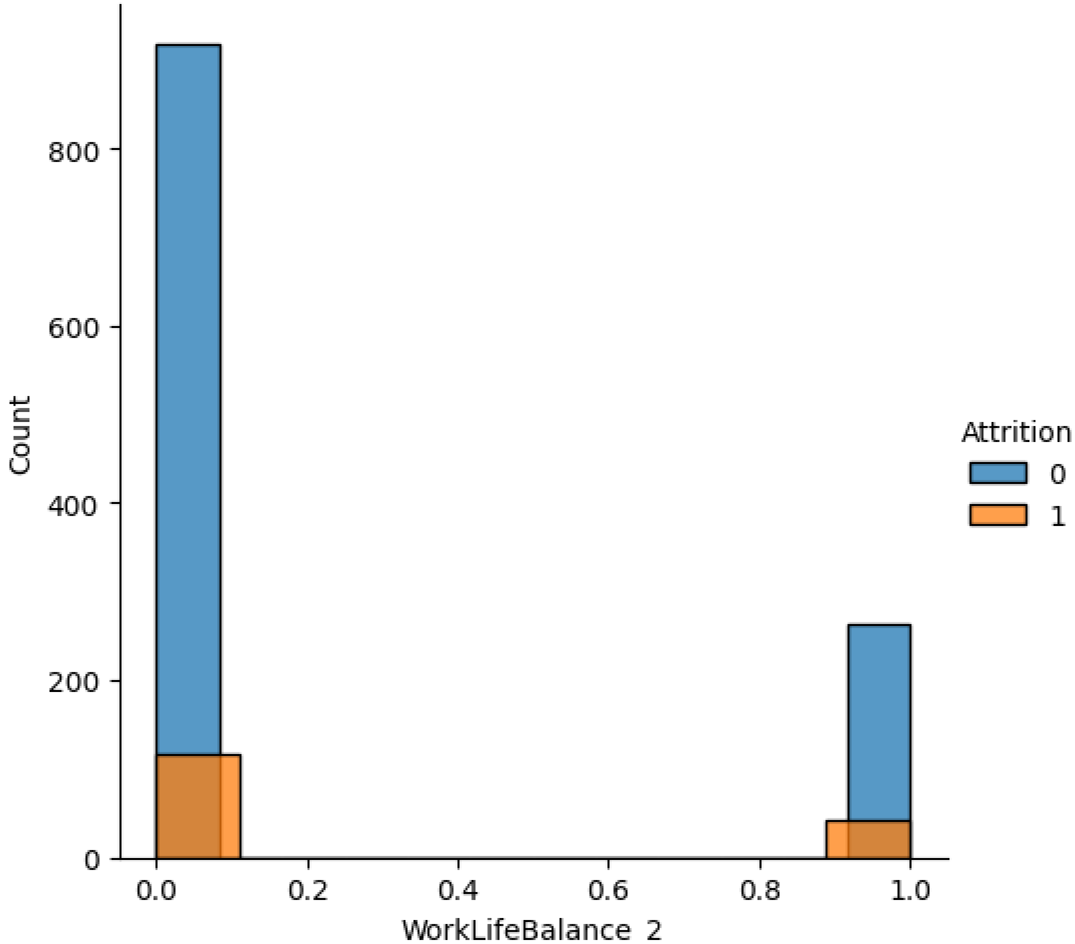


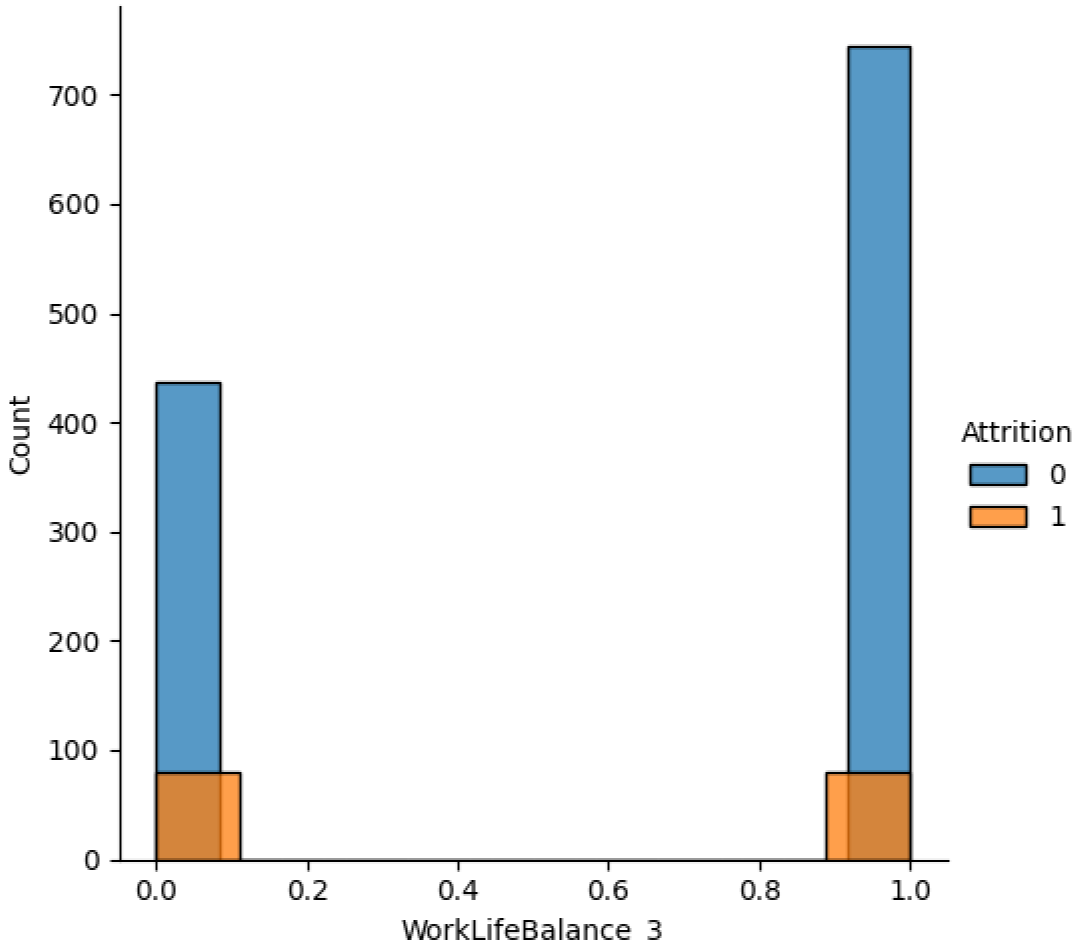
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_5
Attrition
0
1

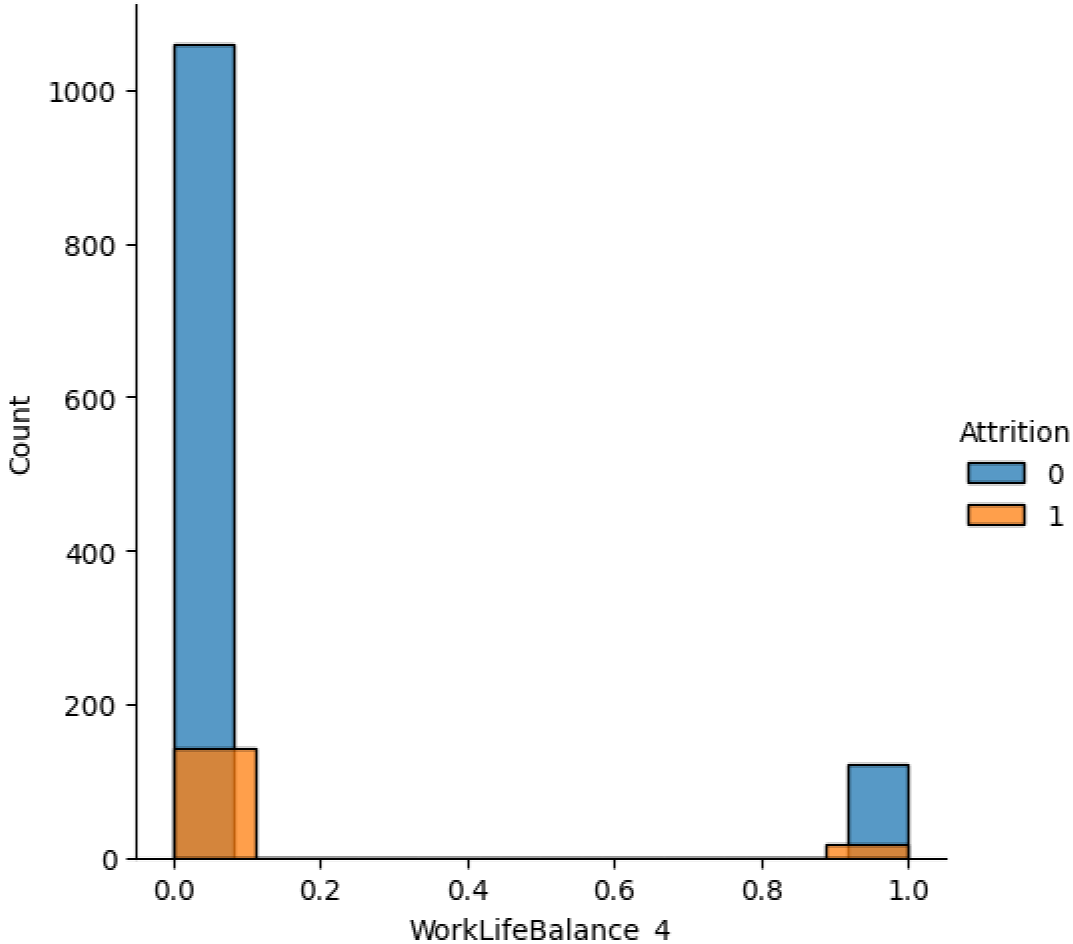

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
TrainingTimesLastYear_6
Attrition
0
1

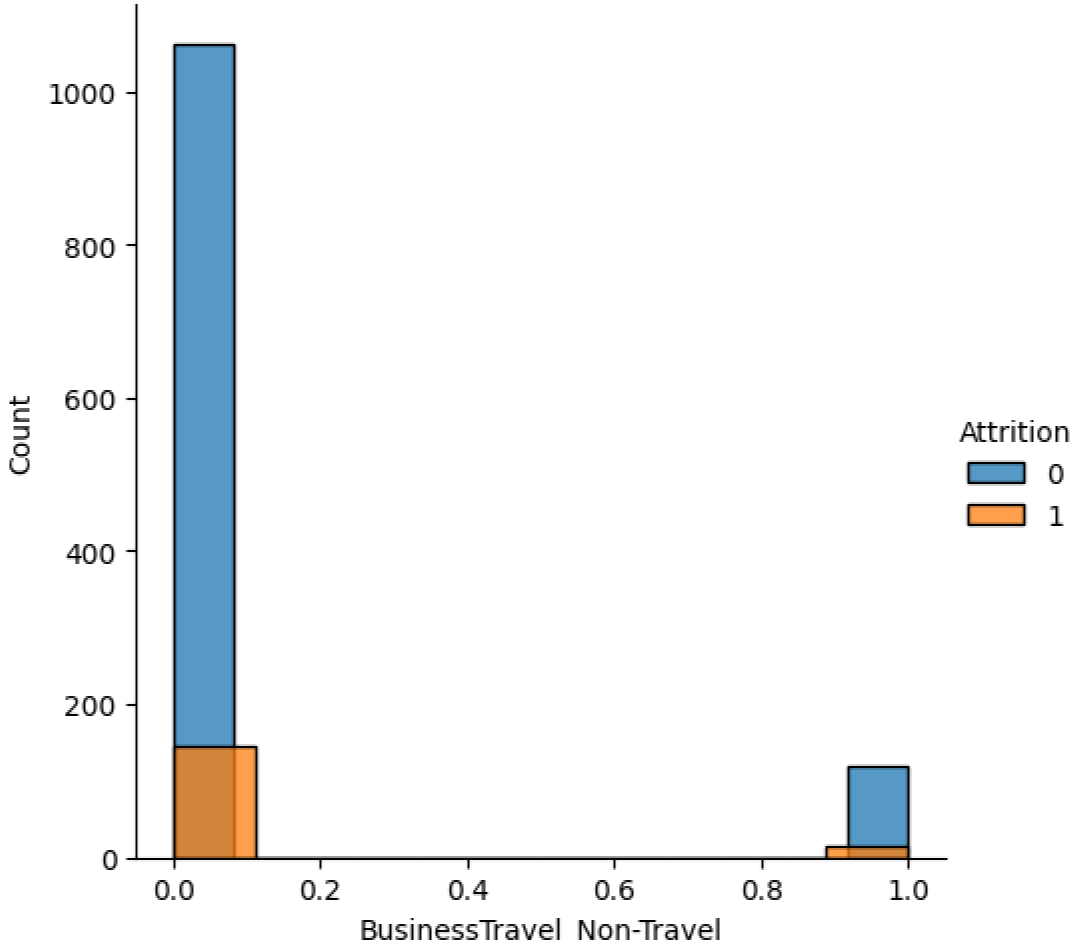


Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
WorkLifeBalance_1
Attrition
0
1

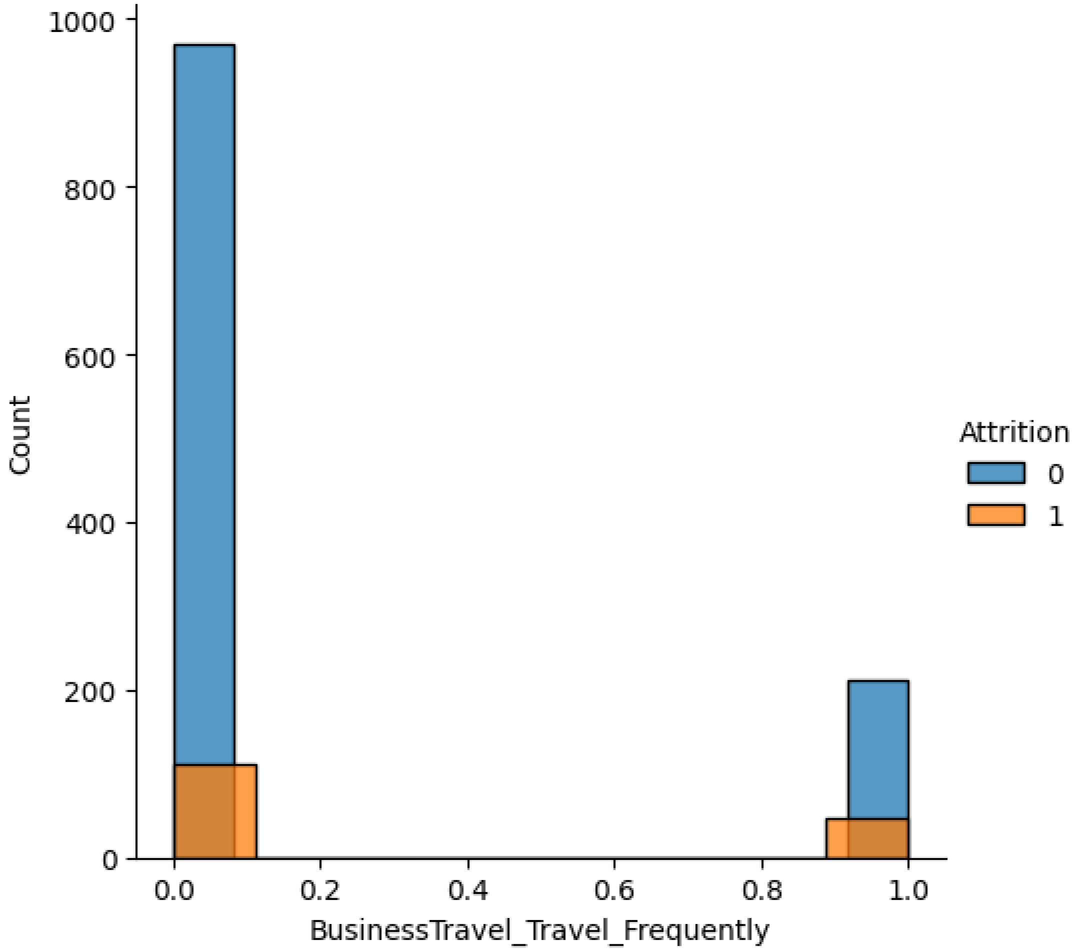

Count
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
WorkLifeBalance_2
Attrition
0
1


Count
WorkLifeBalance_3
Attrition
0
1
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0

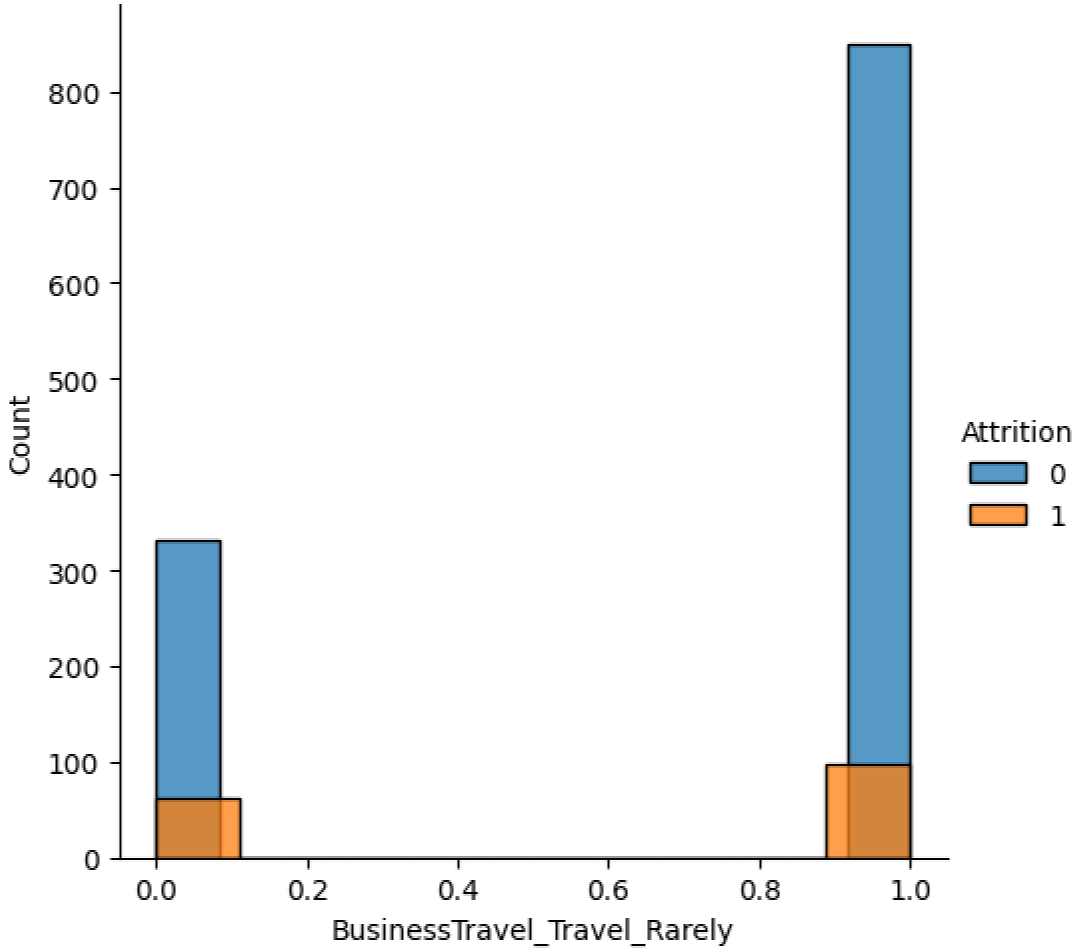

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
WorkLifeBalance_4
Attrition
0
1



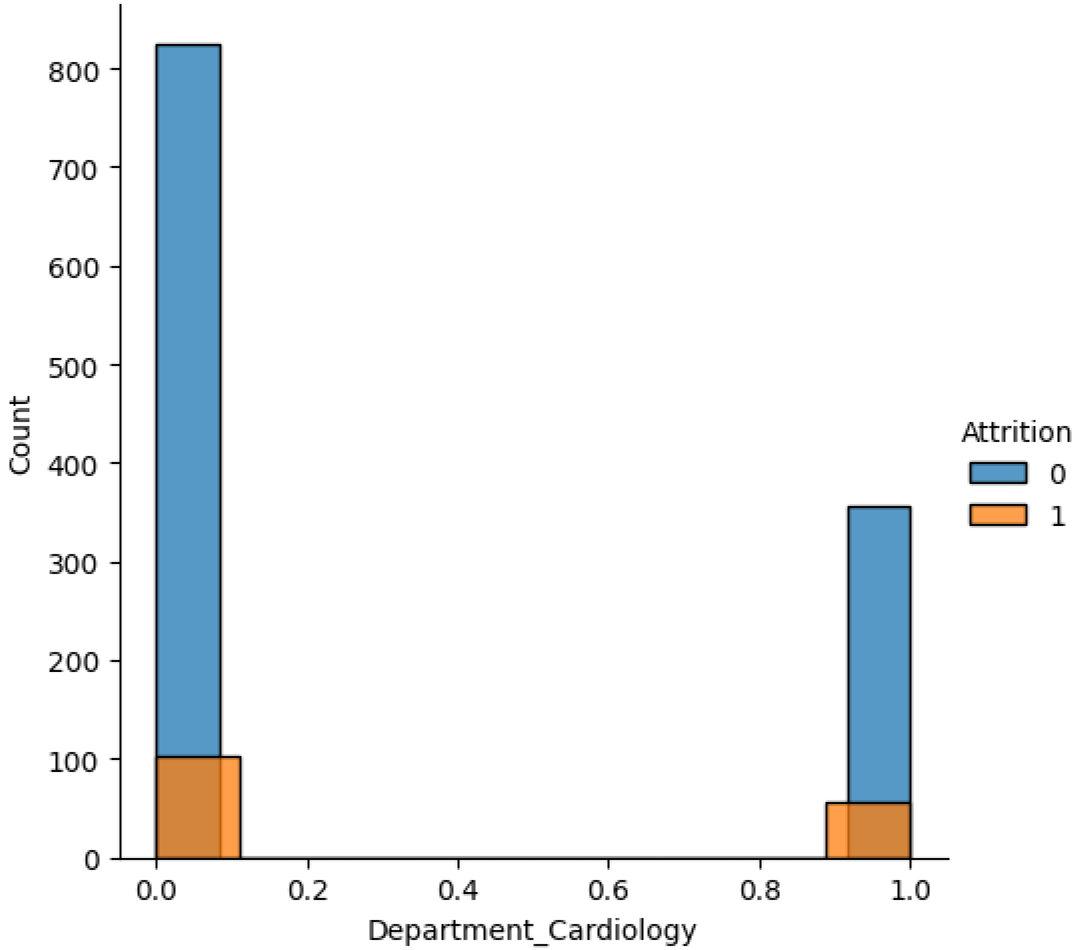
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
BusinessTravel_Non-Travel
Attrition
0
1



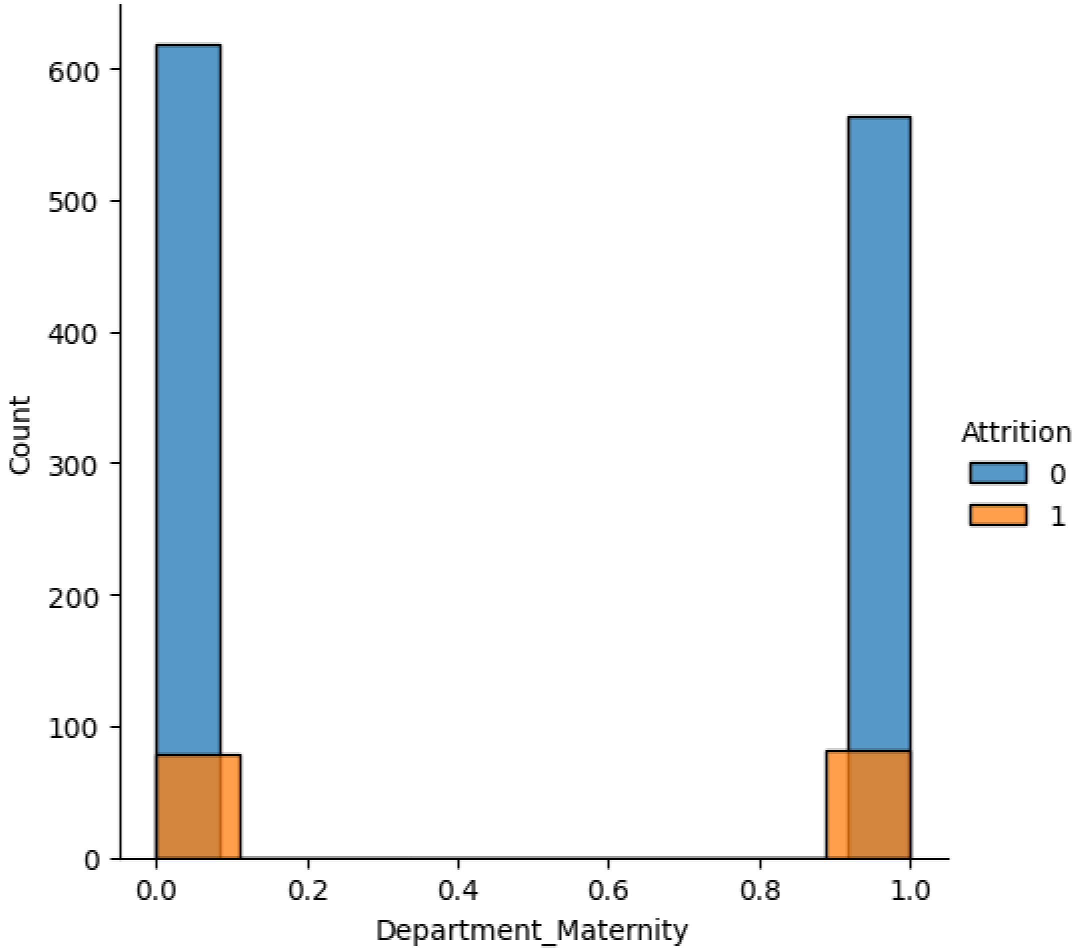
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
BusinessTravel_Travel_Frequently
Attrition
0
1

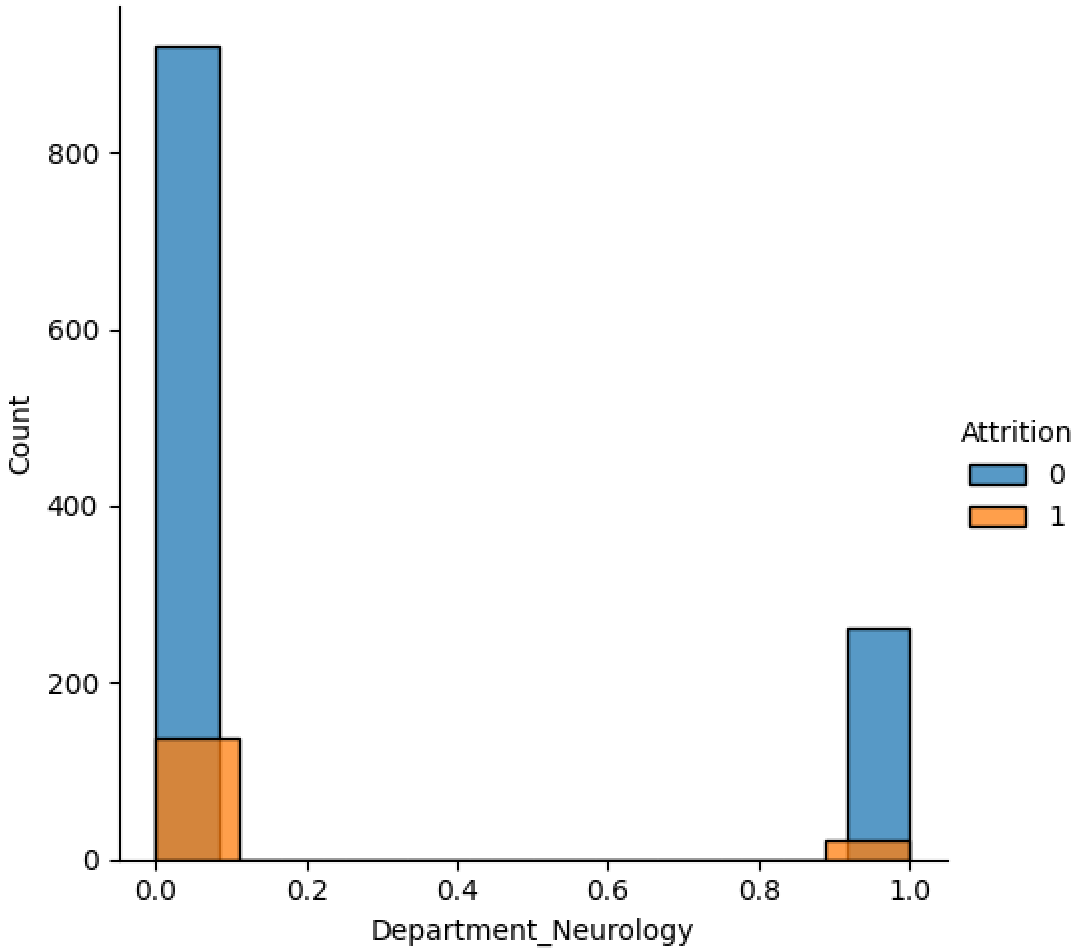


800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
BusinessTravel_Travel_Rarely
Attrition
0
1

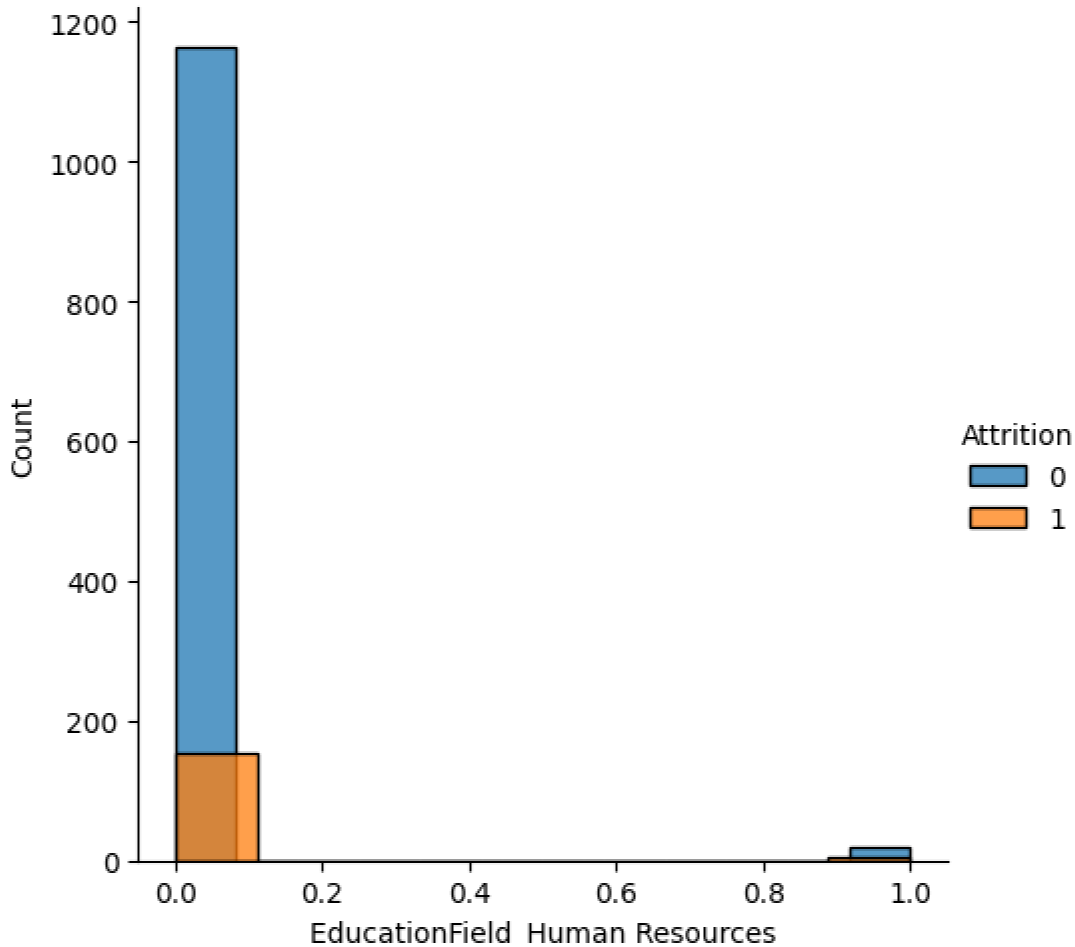


Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
Department_Cardiology
Attrition
0
1

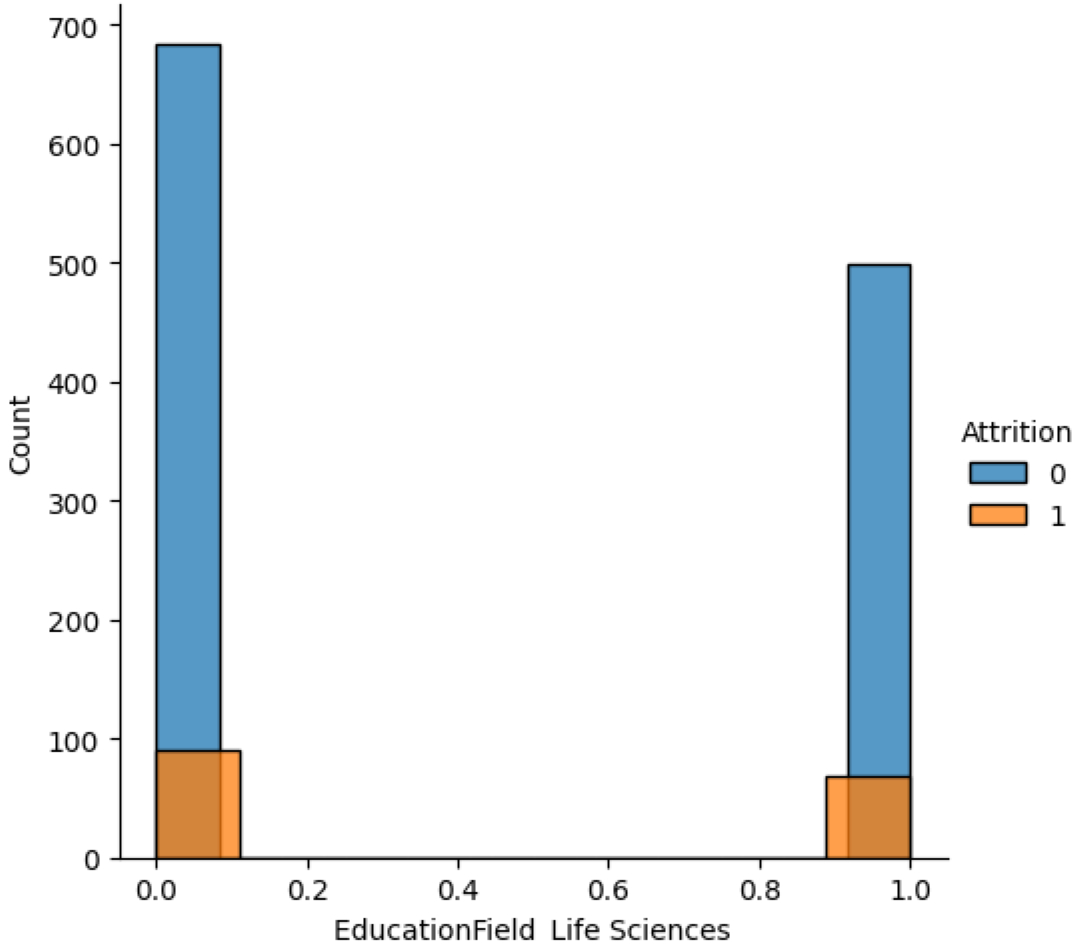

600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Department_Maternity
Attrition
0
1

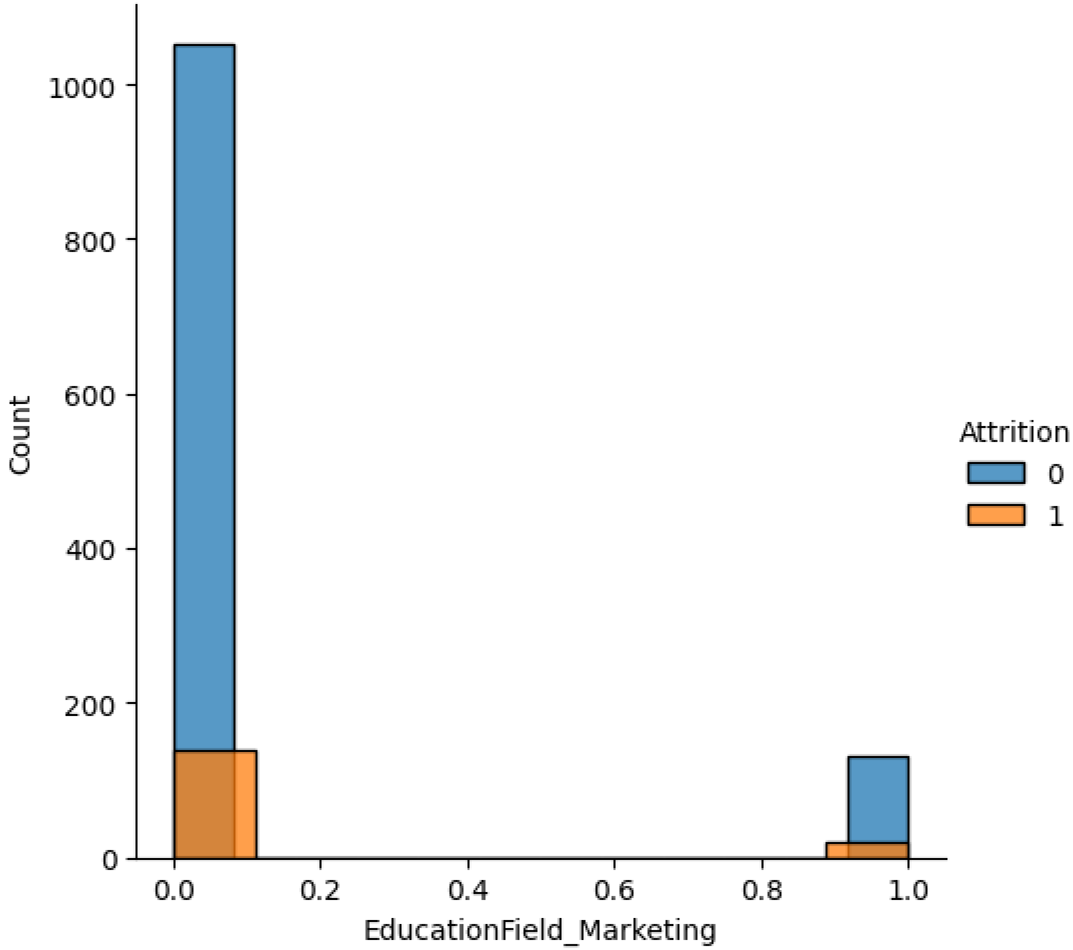


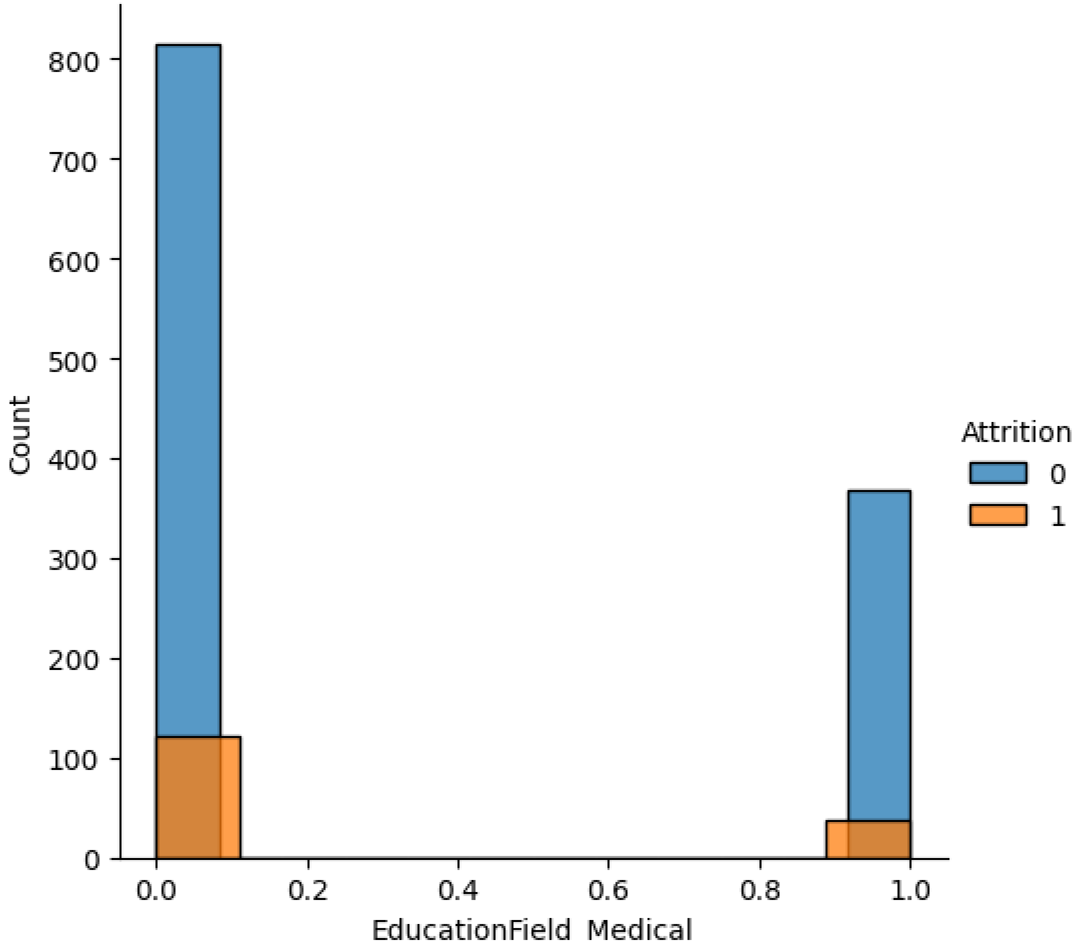
Count
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
Department_Neurology
Attrition
0
1

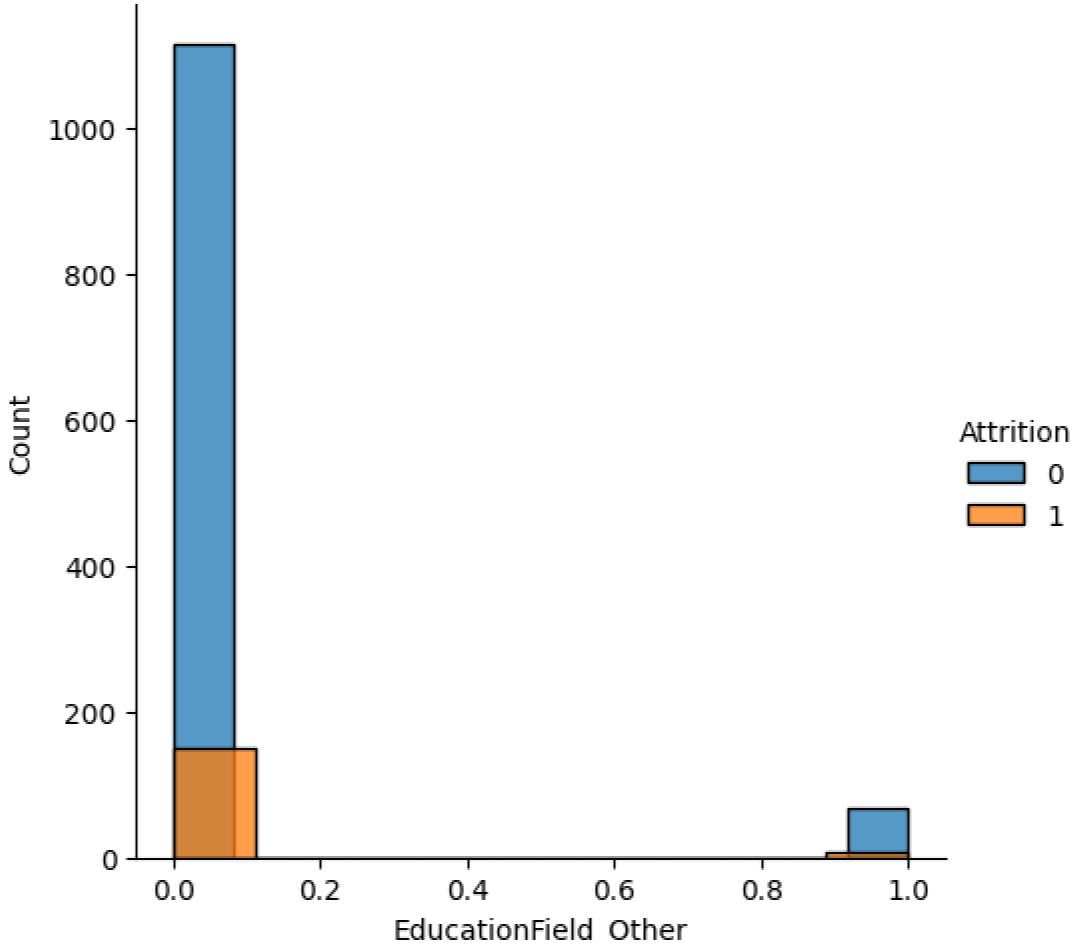


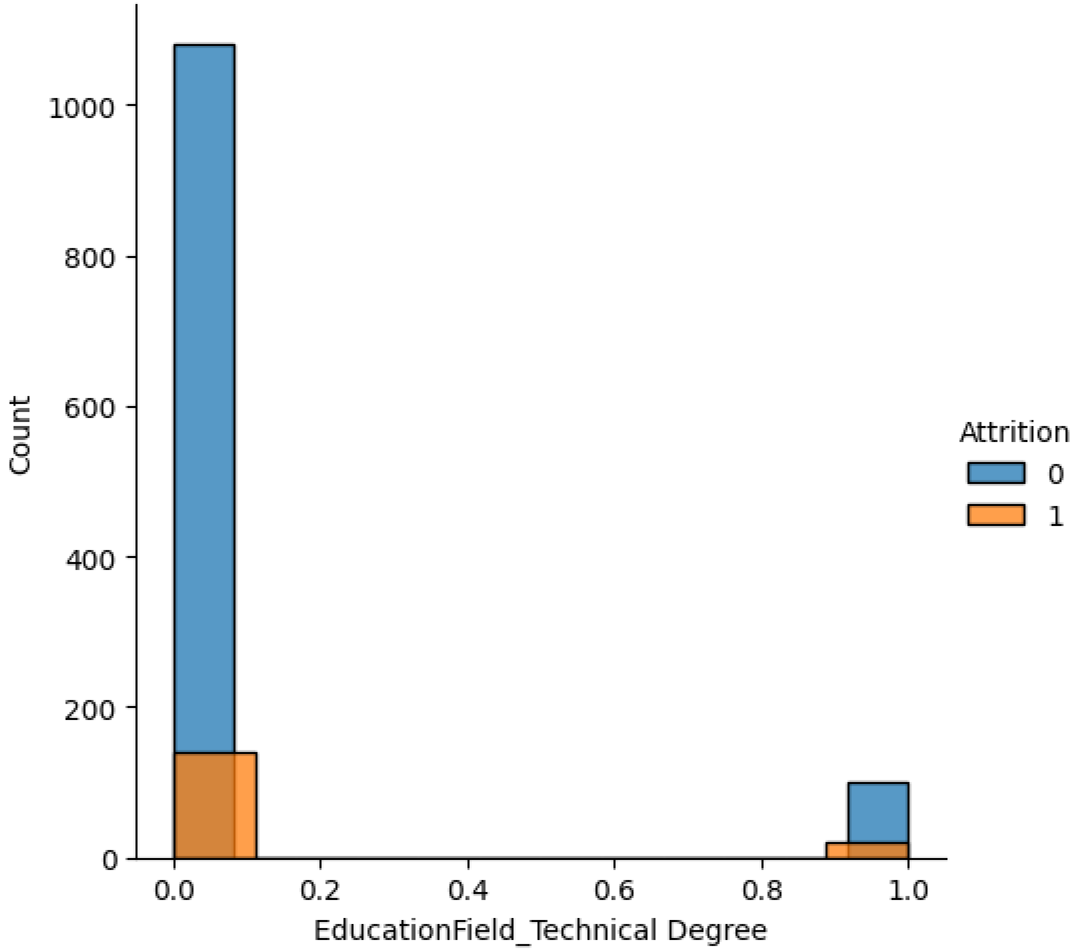
Count
1200
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
EducationField_Human Resources
Attrition
0
1


700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
EducationField_Life Sciences
Attrition
0
1

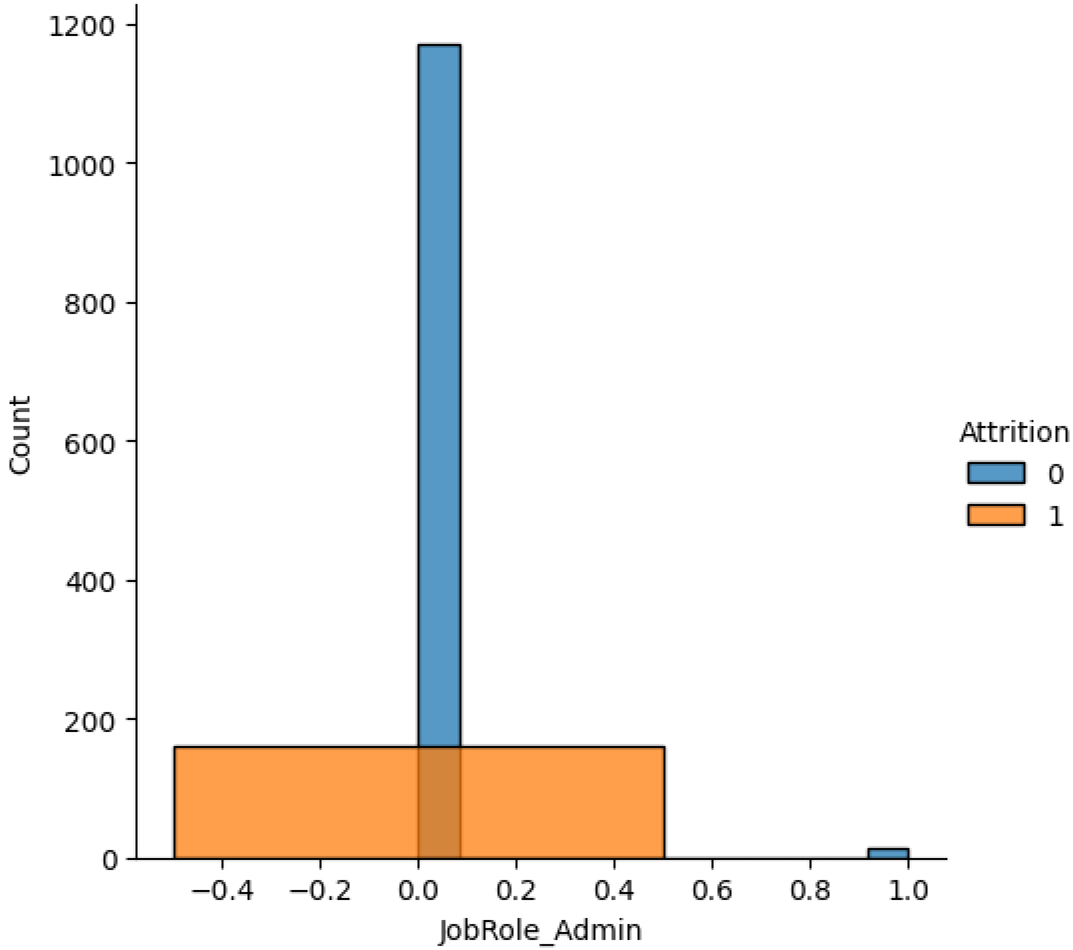

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
EducationField_Marketing
Attrition
0
1

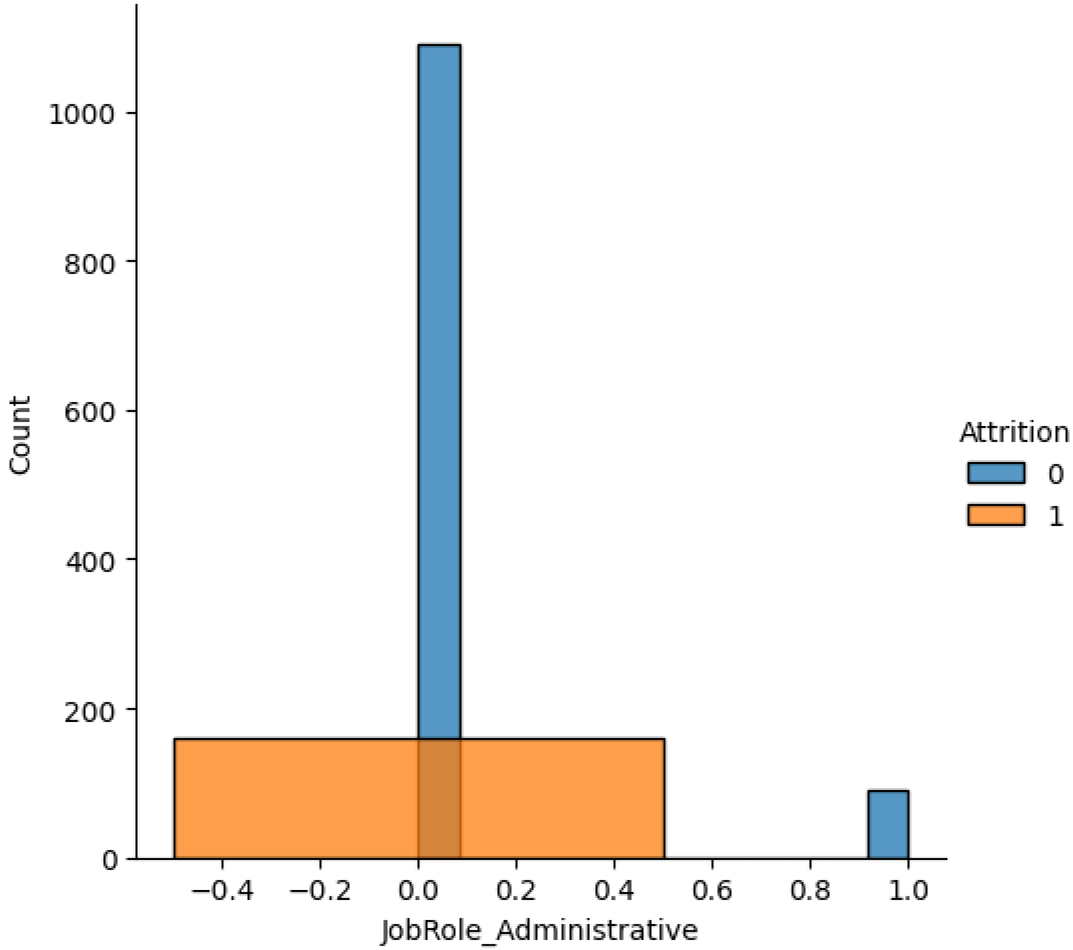

Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
EducationField_Medical
Attrition
0
1

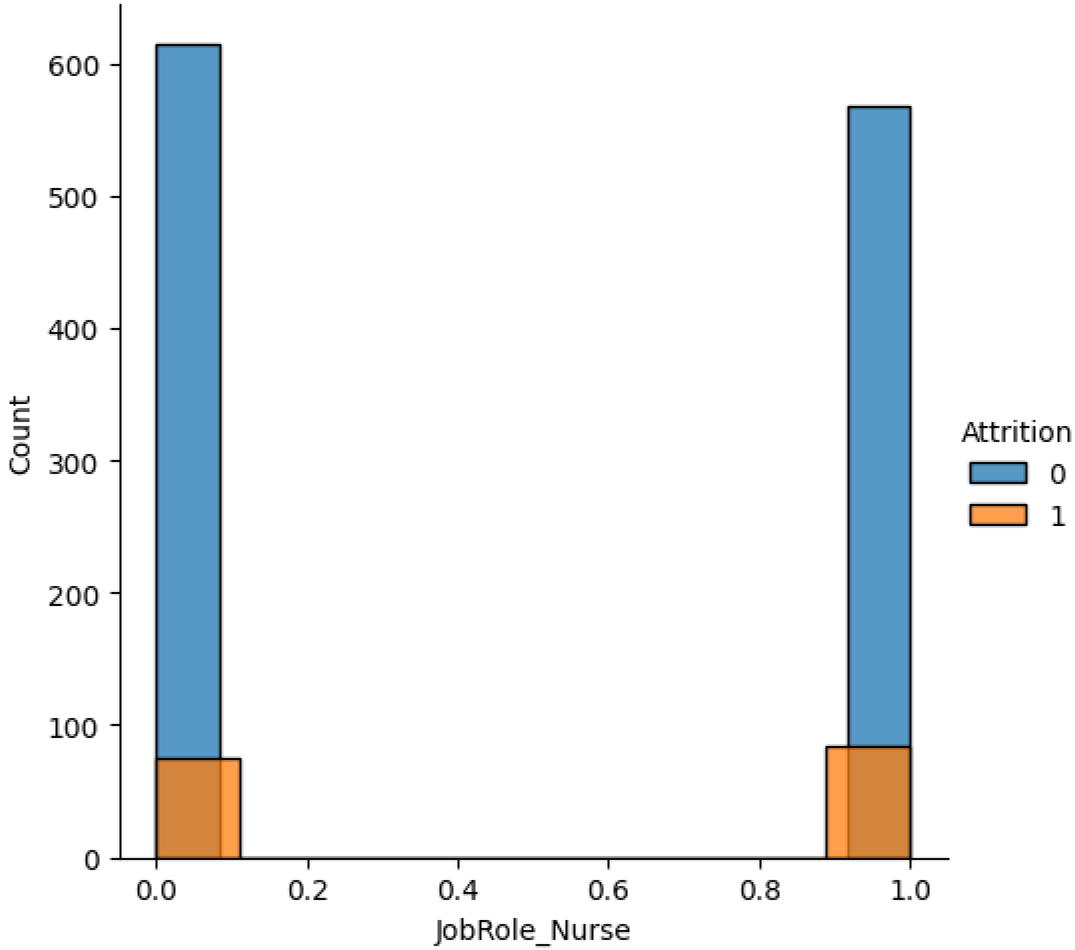

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
EducationField_Other
Attrition
0
1

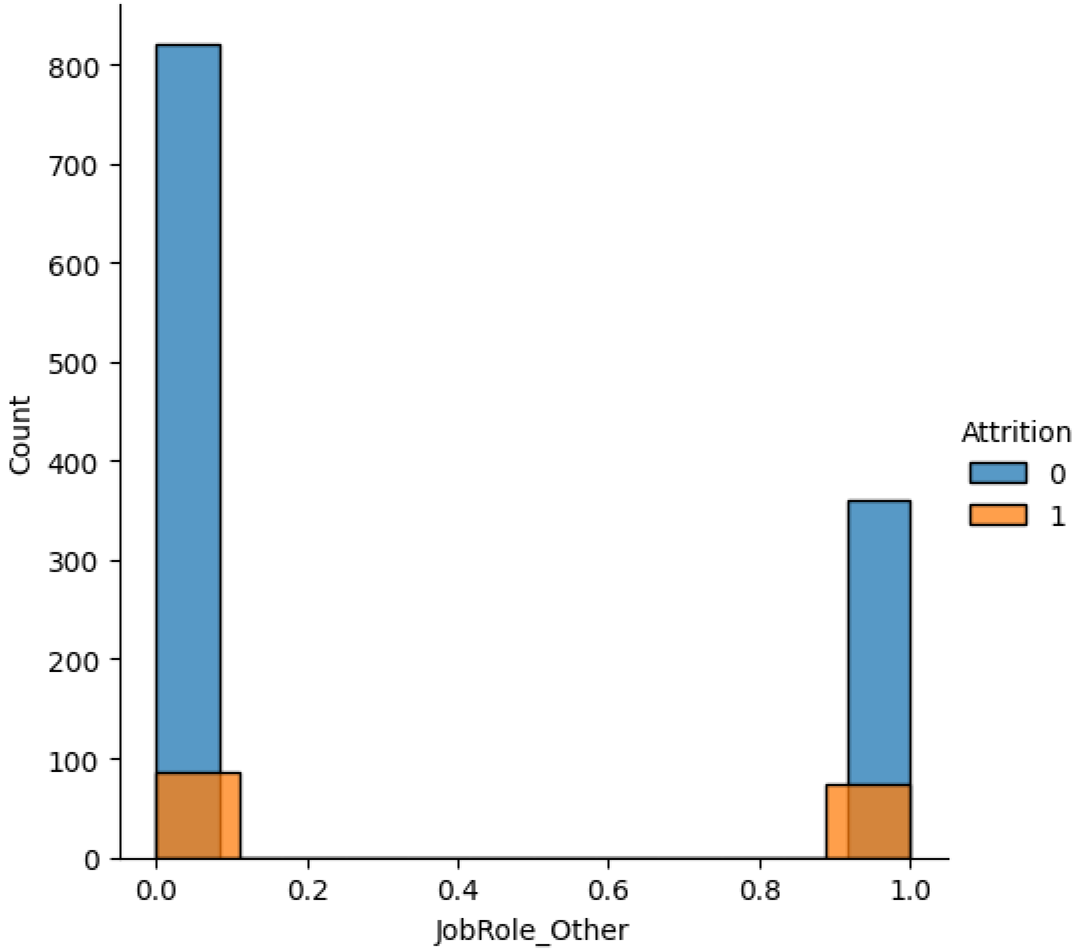


Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
EducationField_Technical Degree
Attrition
0
1

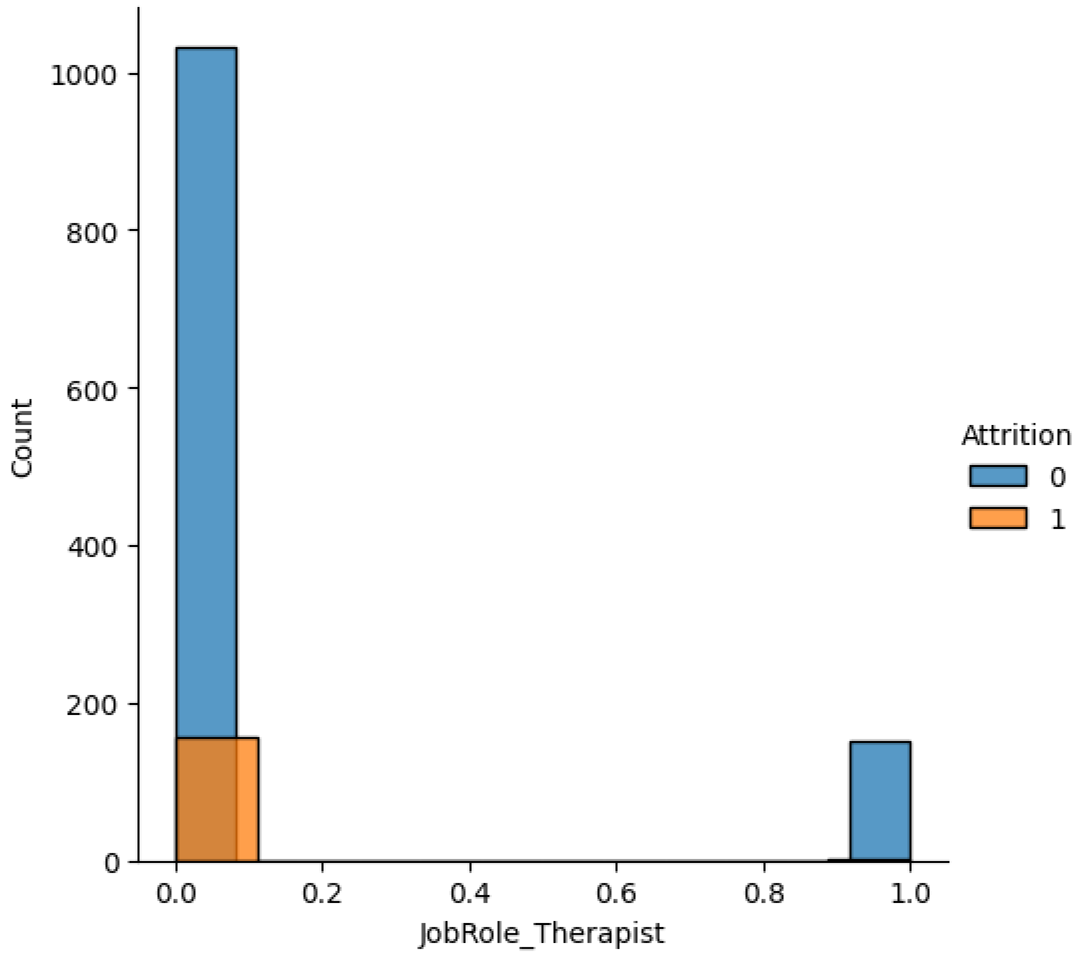

1200
1000
800
600
400
200
0
Count
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
JobRole_Admin
Attrition
0
1


1000
800
600
400
200
0
Count
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
JobRole_Administrative
Attrition
0
1

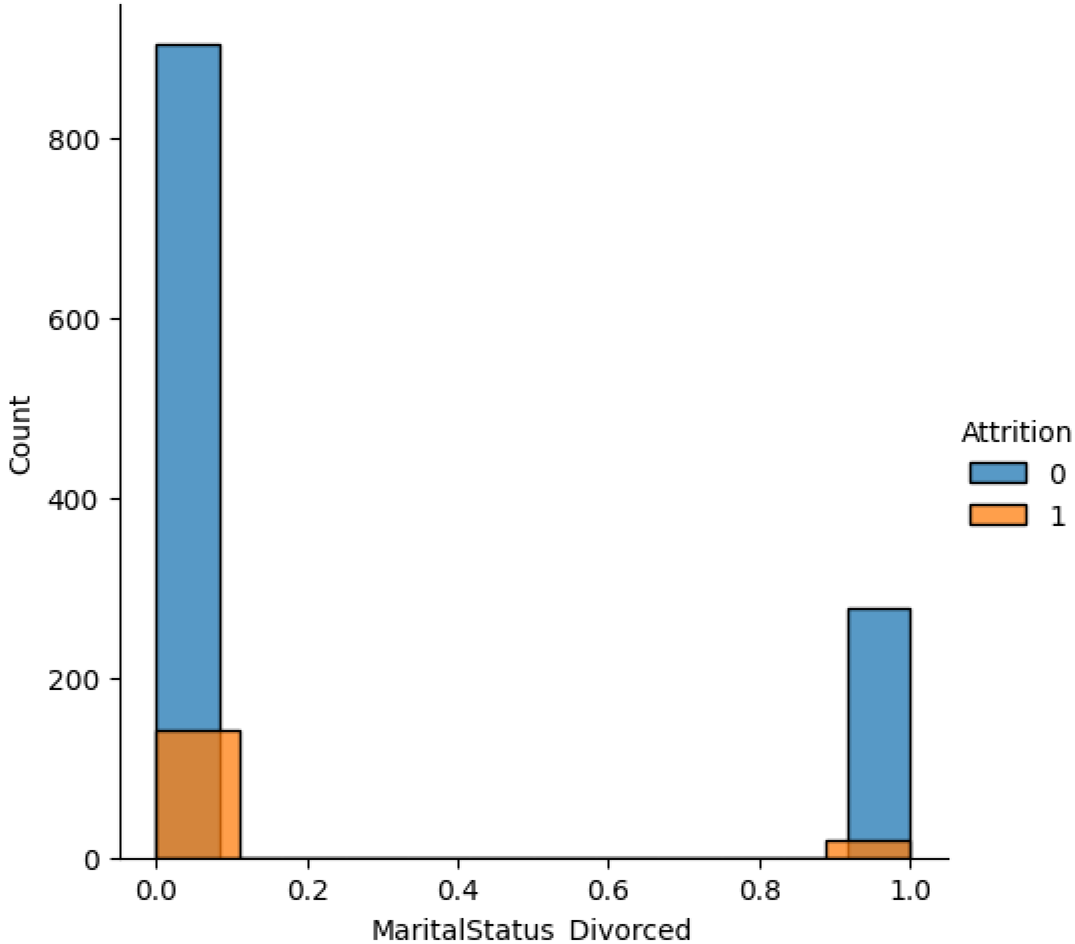

600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobRole_Nurse
Attrition
0
1



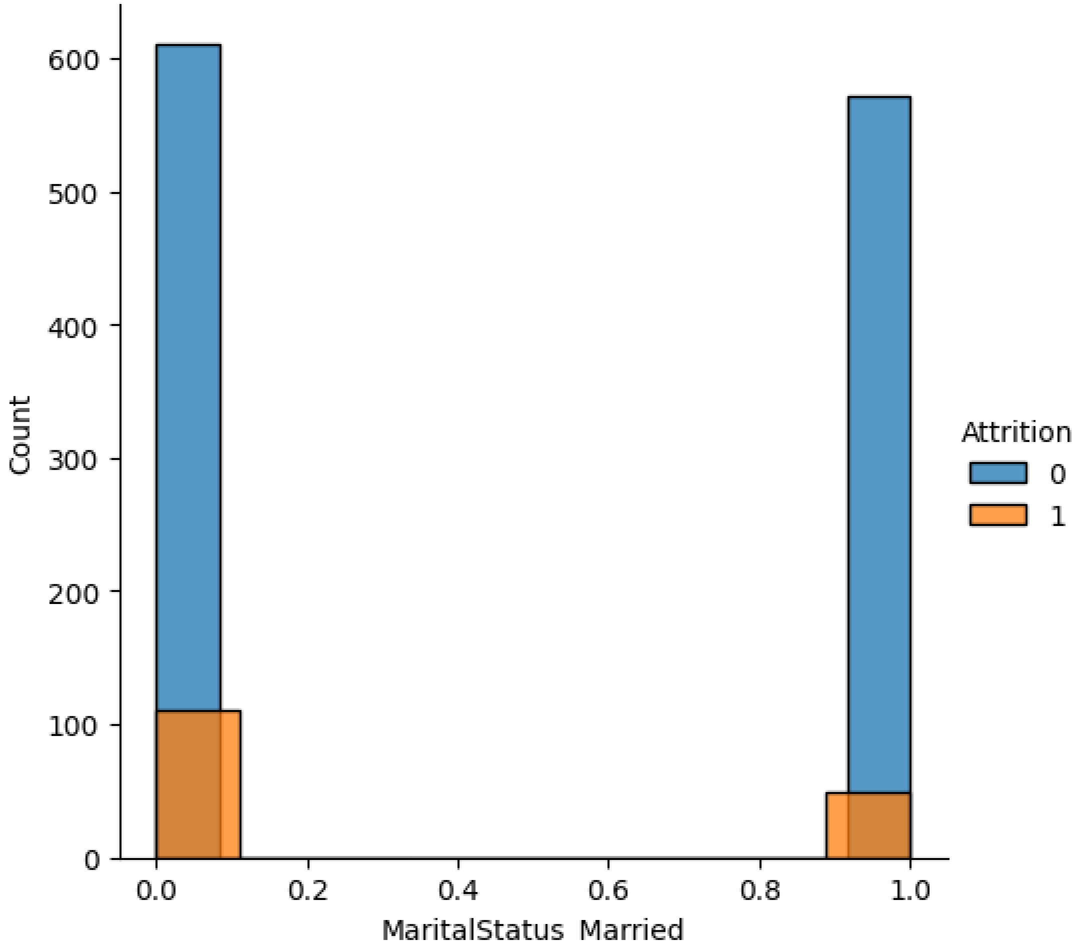
800
700
600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobRole_Other
Attrition
0
1



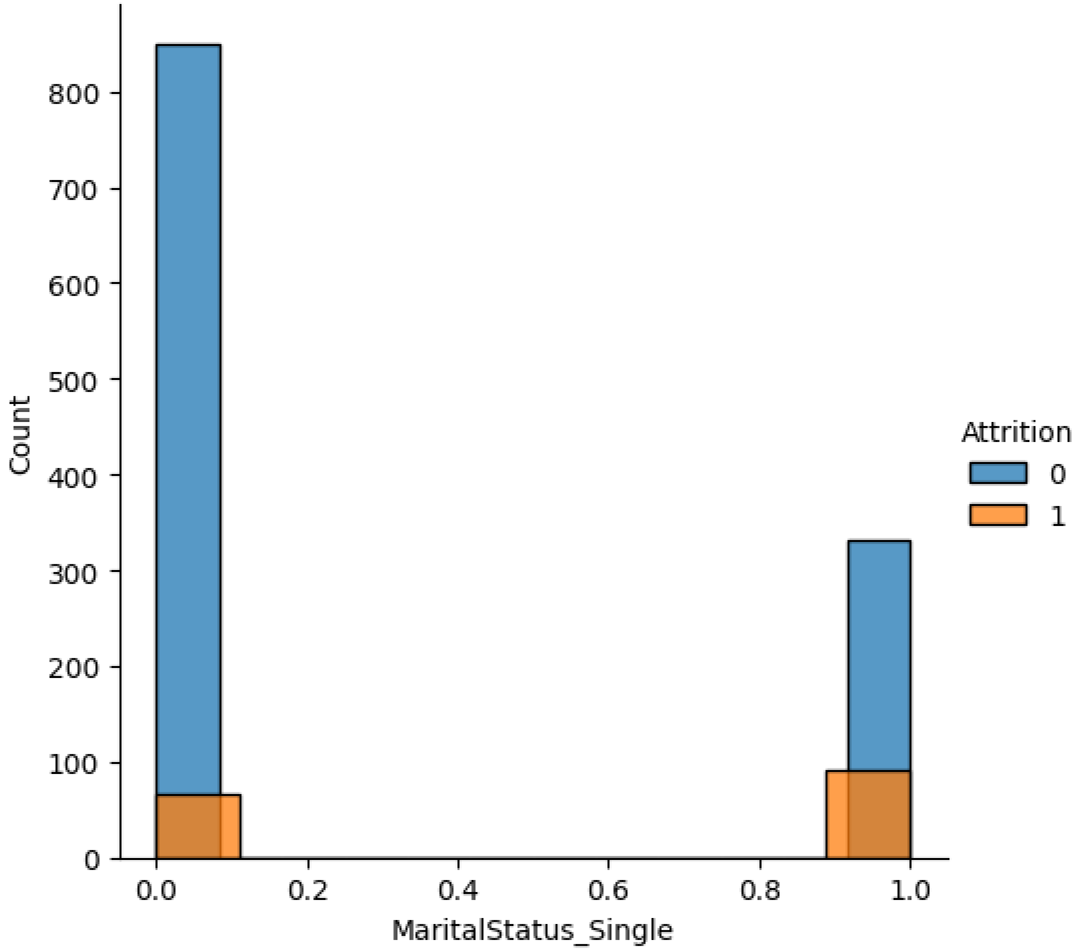
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobRole_Therapist
Attrition
0
1

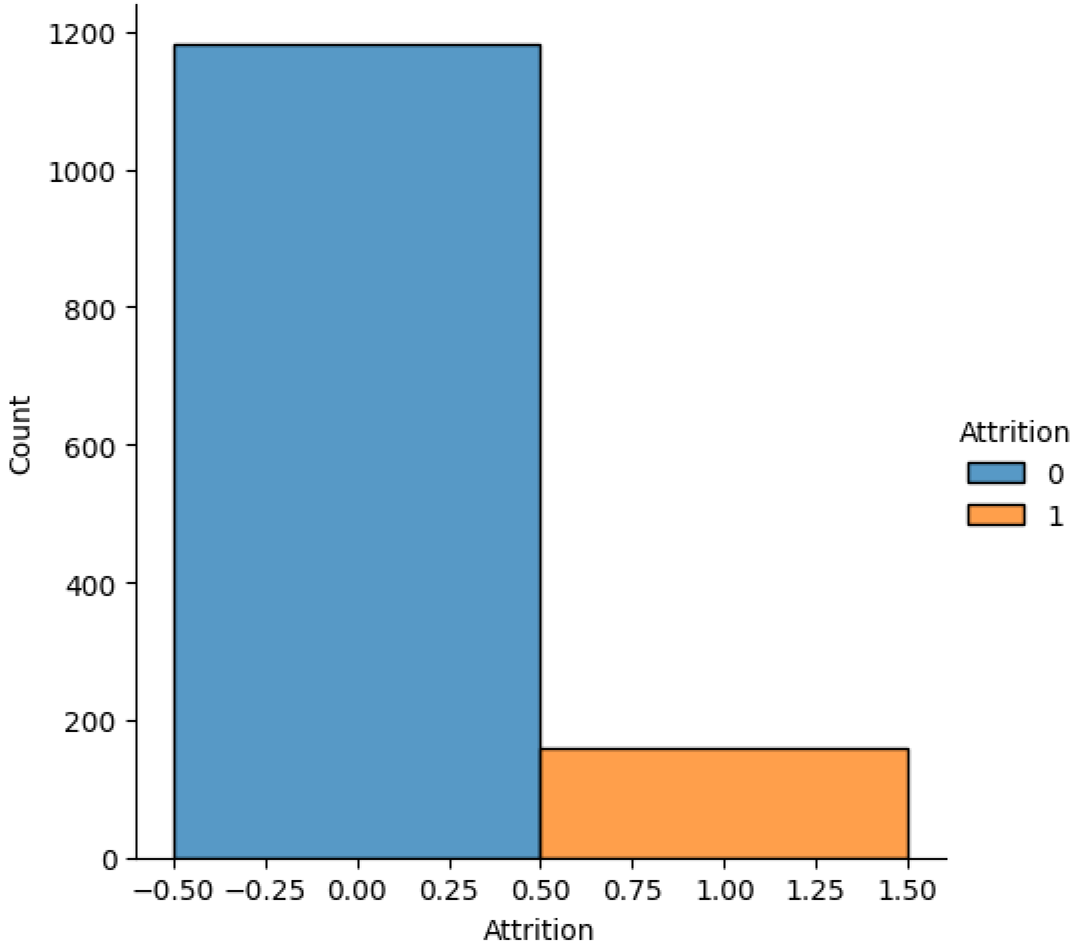


800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
MaritalStatus_Divorced
Attrition
0
1



600
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
MaritalStatus_Married
Attrition
0
1


Count
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
MaritalStatus_Single
Attrition
0
1




In [ ]:

```
# Feature selection using coefficient weights
estimator = LogisticRegression(max_iter=120)
selector = RFECV(estimator, min_features_to_select=20,
                 step=1, cv=5)
selector = selector.fit(data, np.ravel(labels))
names = selector.get_feature_names_out()

# See which features are correlated with Attrition
print(data[names].columns)
print(data[names].shape)
```

```
Index(['BusinessTravel', 'Department', 'DistanceFromHome', 'Education',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'NumCompaniesWorked', 'OverTime', 'PercentSalaryHike',
       'PerformanceRating', 'RelationshipSatisfaction', 'Shift',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager', 'Age_Even', 'DailyRate_Even',
       'DistanceFromHome_Even', 'HourlyRate_Even', 'MonthlyIncome_Even',
       'MonthlyRate_Even', 'TotalWorkingYears_Even', 'YearsInCurrentRole_Ev
en',
       'YearsWithCurrManager_Even', 'NumCompaniesWorked_Even',
       'EnvironmentSatisfaction_1', 'EnvironmentSatisfaction_3',
       'EnvironmentSatisfaction_4', 'JobInvolvement_1', 'JobInvolvement_2',
       'JobInvolvement_3', 'JobInvolvement_4', 'JobLevel_1', 'JobLevel_2',
       'JobLevel_3', 'JobSatisfaction_1', 'JobSatisfaction_4',
       'RelationshipSatisfaction_4', 'Shift_0', 'Shift_1', 'Shift_2',
       'TrainingTimesLastYear_2', 'TrainingTimesLastYear_3',
       'WorkLifeBalance_1', 'WorkLifeBalance_3',
       'BusinessTravel_Travel_Frequently', 'BusinessTravel_Travel_Rarely',
       'Department_Cardiology', 'Department_Maternity',
       'EducationField_Marketing', 'EducationField_Medical',
       'JobRole_Therapist', 'MaritalStatus_Divorced', 'MaritalStatus_Marrie
d',
       'MaritalStatus_Single'],
      dtype='object')
(1340, 65)
```

In [ ]:
```
# Eliminate redundant features
good_data = data[names]
selected_features = mrmr.mrmr_classif(good_data,
                                      np.ravel(labels),
                                      K=20)
print(selected_features)

uncorr_data = good_data[selected_features]
uncorr_sub_data = submission_data[selected_features]
```

```
100%|██████████| 20/20 [00:00<00:00, 86.50it/s]
['OverTime', 'JobLevel_1', 'BusinessTravel_Travel_Rarely', 'JobInvolvemen
t', 'Shift_0', 'WorkLifeBalance_1', 'Age_Even', 'DistanceFromHome_Even', 'E
nvironmentSatisfaction', 'YearsInCurrentRole', 'MaritalStatus_Single', 'Tot
alWorkingYears', 'JobSatisfaction', 'JobLevel_2', 'JobInvolvement_1', 'Year
sWithCurrManager', 'JobLevel', 'EnvironmentSatisfaction_1', 'TrainingTimesL
astYear_2', 'MaritalStatus']
```

In [ ]:
```
# Save the selected data
fin_data = pd.concat([uncorr_data, labels], axis=1)
fin_data.to_csv("uncorr20_data.csv")
uncorr_sub_data.to_csv("uncorr20_sub_data.csv")
```

In [ ]:
```
# Feature engineering by making polynomial features
poly = PolynomialFeatures(degree=2)

poly_data = poly.fit_transform(uncorr_data)
```

```
poly_sub = poly.fit_transform(uncorr_sub_data)

poly_names = poly.get_feature_names_out()
# print(poly_names)

# print(poly_data.shape)

poly_data = pd.DataFrame(poly_data, columns=poly_names)
poly_sub = pd.DataFrame(poly_sub, columns=poly_names)
# print(poly_data.head())
```

In [ ]:
```
# Feature selection using coefficient weights
estimator = LogisticRegression(max_iter=120)
selector = RFECV(estimator, min_features_to_select=20,
                 step=1, cv=5)
selector = selector.fit(poly_data, np.ravel(labels))
names = selector.get_feature_names_out()

print(poly_data[names].columns)
print(poly_data[names].shape)
```

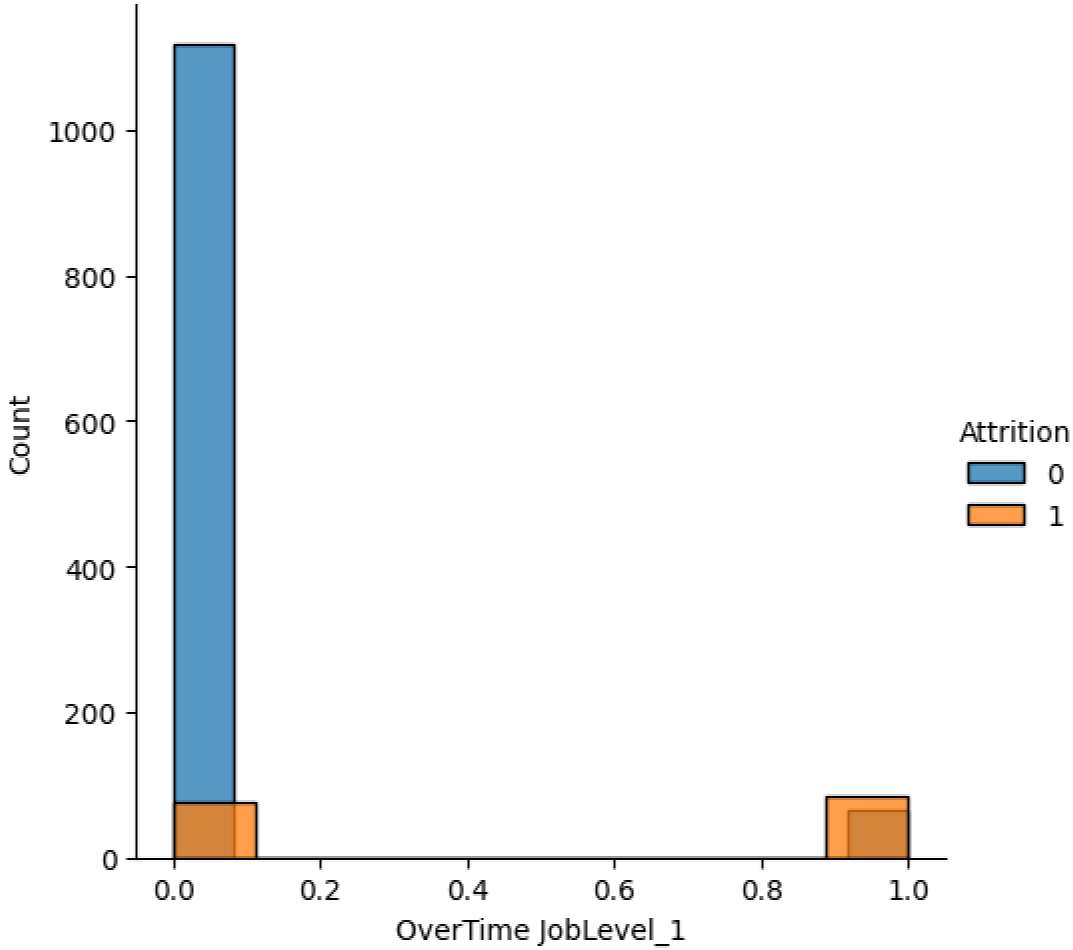
```
Index(['OverTime', 'YearsInCurrentRole', 'MaritalStatus_Single',
       'YearsWithCurrManager', 'OverTime^2', 'OverTime JobLevel_1',
       'OverTime YearsInCurrentRole', 'OverTime JobSatisfaction',
       'OverTime EnvironmentSatisfaction_1',
       'OverTime TrainingTimesLastYear_2', 'OverTime MaritalStatus',
       'JobLevel_1 WorkLifeBalance_1', 'JobLevel_1 Age_Even',
       'JobLevel_1 DistanceFromHome_Even', 'JobLevel_1 MaritalStatus_Singl
e',
       'JobLevel_1 EnvironmentSatisfaction_1',
       'BusinessTravel_Travel_Rarely WorkLifeBalance_1',
       'BusinessTravel_Travel_Rarely Age_Even',
       'BusinessTravel_Travel_Rarely DistanceFromHome_Even',
       'BusinessTravel_Travel_Rarely JobInvolvement_1',
       'BusinessTravel_Travel_Rarely JobLevel',
       'BusinessTravel_Travel_Rarely TrainingTimesLastYear_2',
       'BusinessTravel_Travel_Rarely MaritalStatus',
       'JobInvolvement MaritalStatus_Single', 'JobInvolvement JobSatisfacti
on',
       'JobInvolvement JobLevel_2', 'JobInvolvement EnvironmentSatisfaction
_1',
       'JobInvolvement MaritalStatus', 'Shift_0 WorkLifeBalance_1',
       'Shift_0 DistanceFromHome_Even', 'Shift_0 EnvironmentSatisfaction',
       'Shift_0 YearsInCurrentRole', 'Shift_0 MaritalStatus_Single',
       'Shift_0 JobInvolvement_1', 'Shift_0 YearsWithCurrManager',
       'Shift_0 EnvironmentSatisfaction_1', 'Shift_0 TrainingTimesLastYear_
2',
       'Shift_0 MaritalStatus', 'WorkLifeBalance_1 DistanceFromHome_Even',
       'WorkLifeBalance_1 YearsInCurrentRole',
       'WorkLifeBalance_1 TotalWorkingYears',
       'WorkLifeBalance_1 JobSatisfaction', 'WorkLifeBalance_1 JobLevel_2',
       'WorkLifeBalance_1 YearsWithCurrManager', 'Age_Even^2',
       'Age_Even EnvironmentSatisfaction', 'Age_Even YearsInCurrentRole',
       'Age_Even MaritalStatus_Single', 'Age_Even JobSatisfaction',
       'Age_Even JobLevel_2', 'Age_Even MaritalStatus',
       'DistanceFromHome_Even EnvironmentSatisfaction',
       'DistanceFromHome_Even JobLevel_2',
       'DistanceFromHome_Even JobInvolvement_1',
       'DistanceFromHome_Even TrainingTimesLastYear_2',
       'DistanceFromHome_Even MaritalStatus',
       'EnvironmentSatisfaction YearsInCurrentRole',
       'EnvironmentSatisfaction MaritalStatus_Single',
       'EnvironmentSatisfaction JobInvolvement_1',
       'EnvironmentSatisfaction YearsWithCurrManager',
       'EnvironmentSatisfaction TrainingTimesLastYear_2',
       'EnvironmentSatisfaction MaritalStatus',
       'YearsInCurrentRole MaritalStatus_Single',
       'YearsInCurrentRole MaritalStatus', 'MaritalStatus_Single^2',
       'MaritalStatus_Single TotalWorkingYears',
       'MaritalStatus_Single JobLevel_2',
       'MaritalStatus_Single JobInvolvement_1',
       'MaritalStatus_Single JobLevel',
       'MaritalStatus_Single EnvironmentSatisfaction_1',
       'MaritalStatus_Single MaritalStatus', 'JobSatisfaction JobLevel_2',
       'JobSatisfaction TrainingTimesLastYear_2',
       'JobSatisfaction MaritalStatus', 'JobLevel_2 JobInvolvement_1',
       'JobLevel_2 MaritalStatus', 'JobInvolvement_1 YearsWithCurrManager',
```

```
       'JobInvolvement_1 JobLevel', 'JobInvolvement_1 TrainingTimesLastYear
_2',
       'JobInvolvement_1 MaritalStatus', 'YearsWithCurrManager JobLevel',
       'YearsWithCurrManager EnvironmentSatisfaction_1',
       'YearsWithCurrManager MaritalStatus',
       'EnvironmentSatisfaction_1 MaritalStatus', 'TrainingTimesLastYear_2^
2',
       'MaritalStatus^2'],
      dtype='object')
(1340, 86)
```

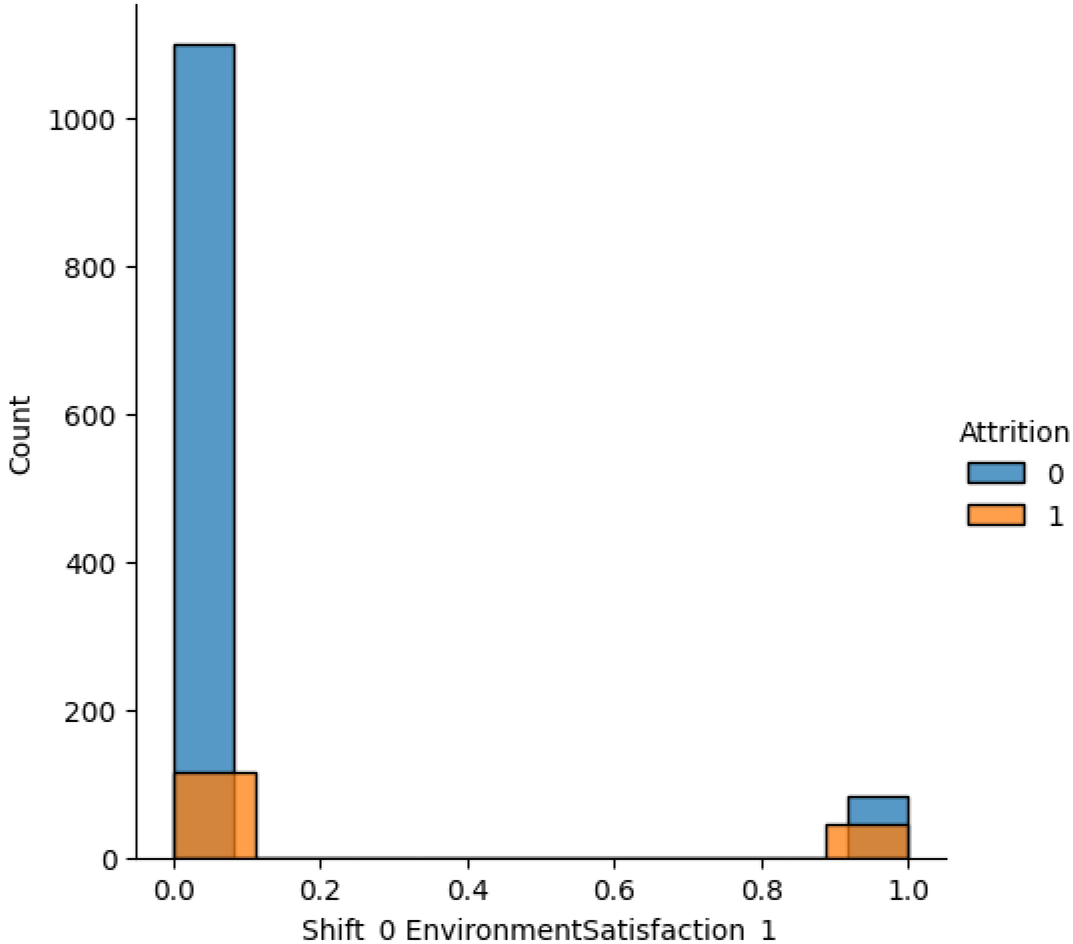
In [ ]:
```python
# Eliminate redundant features
good_poly_data = poly_data[names]
selected_features = mrmr.mrmr_classif(good_poly_data,
                                      np.ravel(labels),
                                      K=20)
print(selected_features)

uncorr_poly_data = good_poly_data[selected_features]
uncorr_poly_sub_data = poly_sub[selected_features]
```

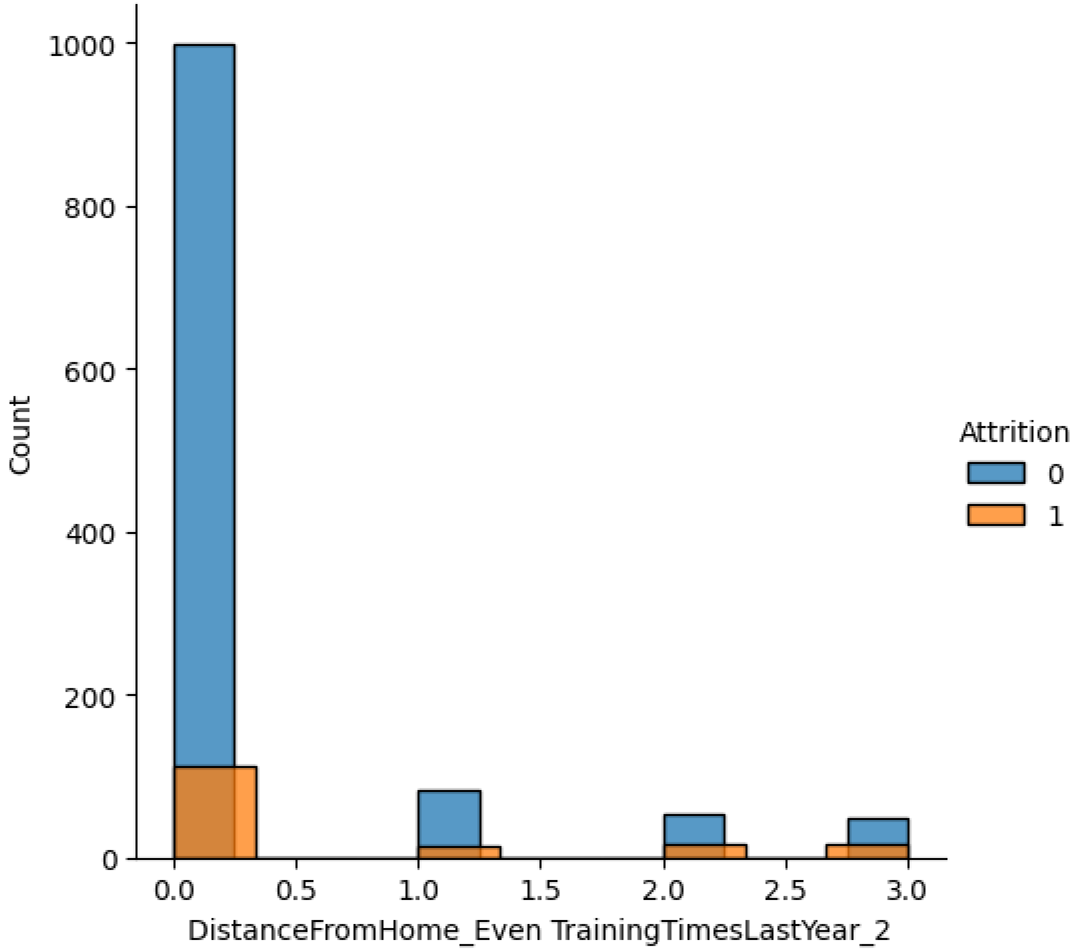
```
100%|██████████| 20/20 [00:00<00:00, 50.45it/s]
['OverTime JobLevel_1', 'Shift_0 EnvironmentSatisfaction_1', 'DistanceFromH
ome_Even TrainingTimesLastYear_2', 'JobInvolvement JobSatisfaction', 'OverT
ime MaritalStatus', 'BusinessTravel_Travel_Rarely Age_Even', 'JobLevel_1 Ma
ritalStatus_Single', 'OverTime', 'Shift_0 JobInvolvement_1', 'YearsWithCurr
Manager', 'OverTime EnvironmentSatisfaction_1', 'JobLevel_1 WorkLifeBalance
_1', 'OverTime^2', 'Age_Even EnvironmentSatisfaction', 'Shift_0 DistanceFro
mHome_Even', 'JobInvolvement JobLevel_2', 'OverTime TrainingTimesLastYear_
2', 'YearsInCurrentRole', 'MaritalStatus_Single EnvironmentSatisfaction_1',
'OverTime JobSatisfaction']
```

In [ ]:
```python
# Generate histograms to check usefulness of features selected
fin_poly_data = pd.concat([uncorr_poly_data, labels], axis=1)
for c in fin_poly_data.columns:
    sns.FacetGrid(fin_poly_data,
                  hue="Attrition",
                  height= 5).map(sns.histplot,c).add_legend()
```

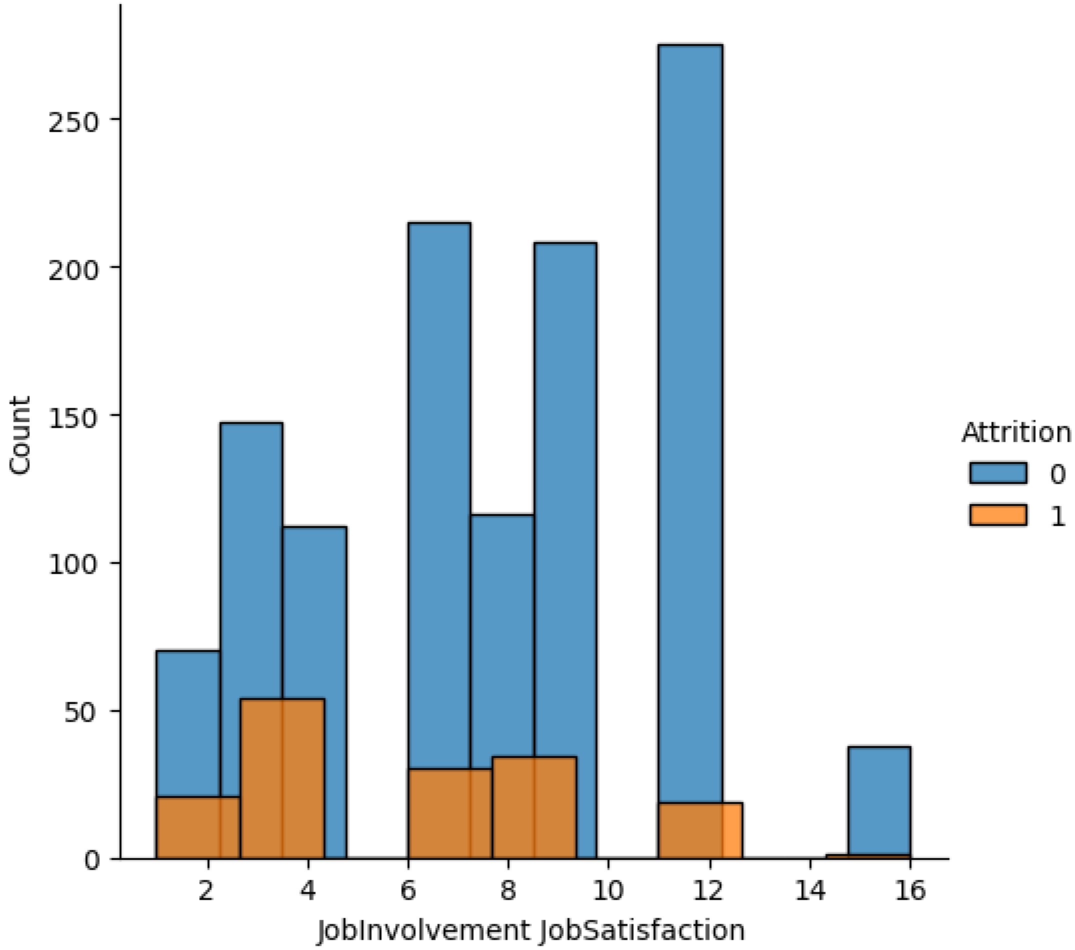


Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
OverTime JobLevel_1
Attrition
0
1

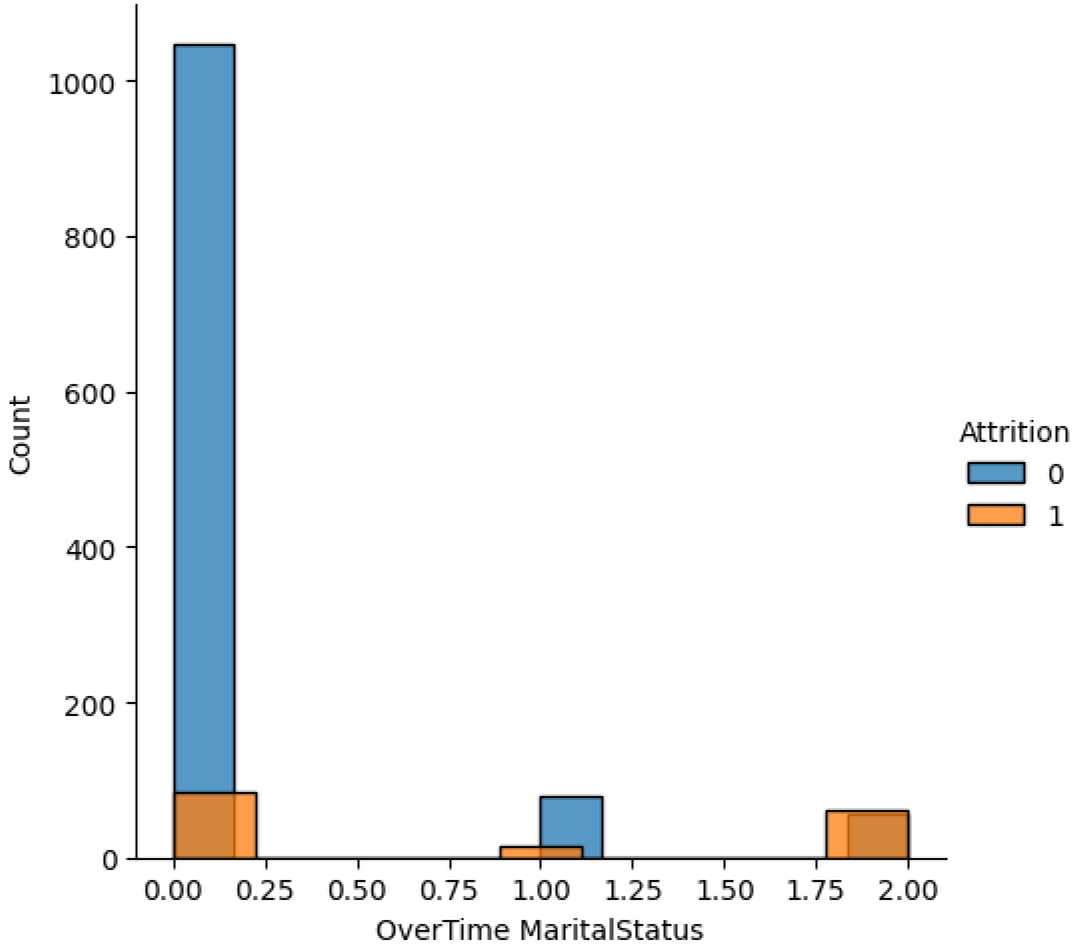

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
Shift_0 EnvironmentSatisfaction_1
Attrition
0
1

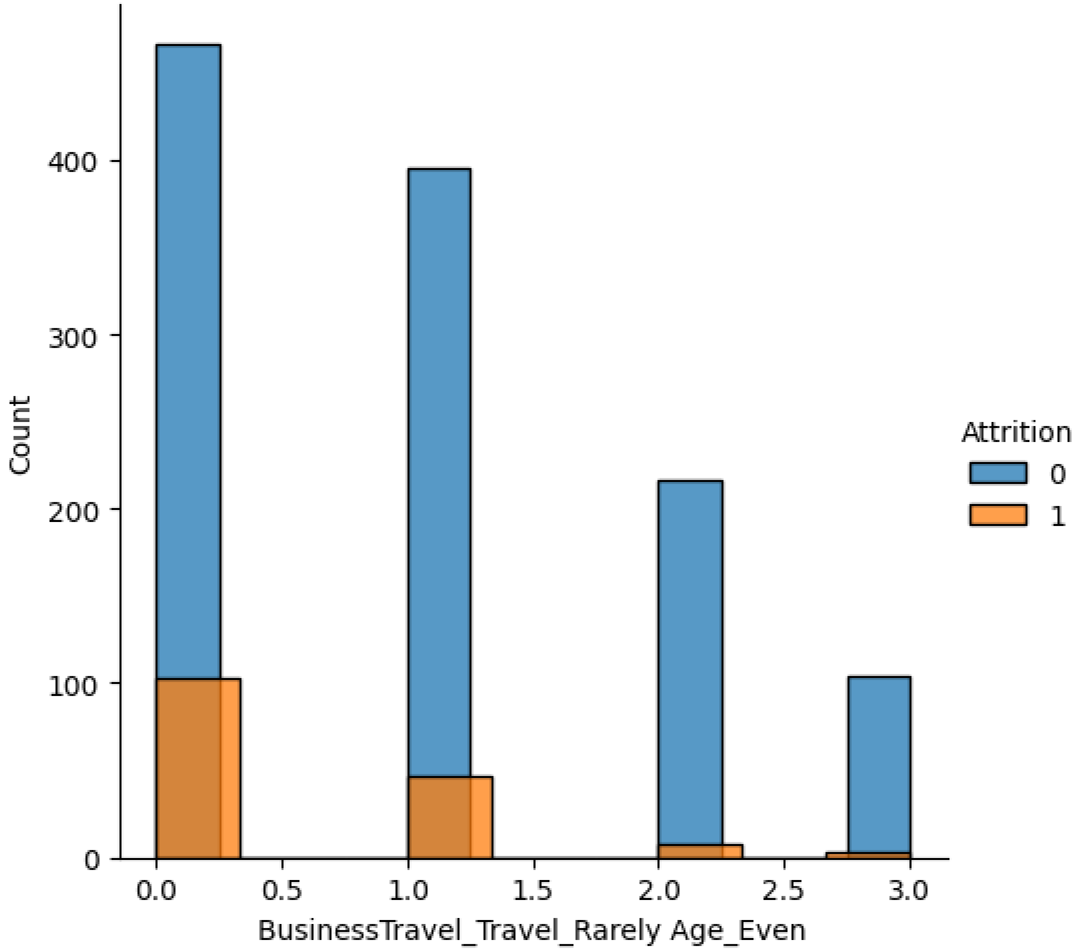


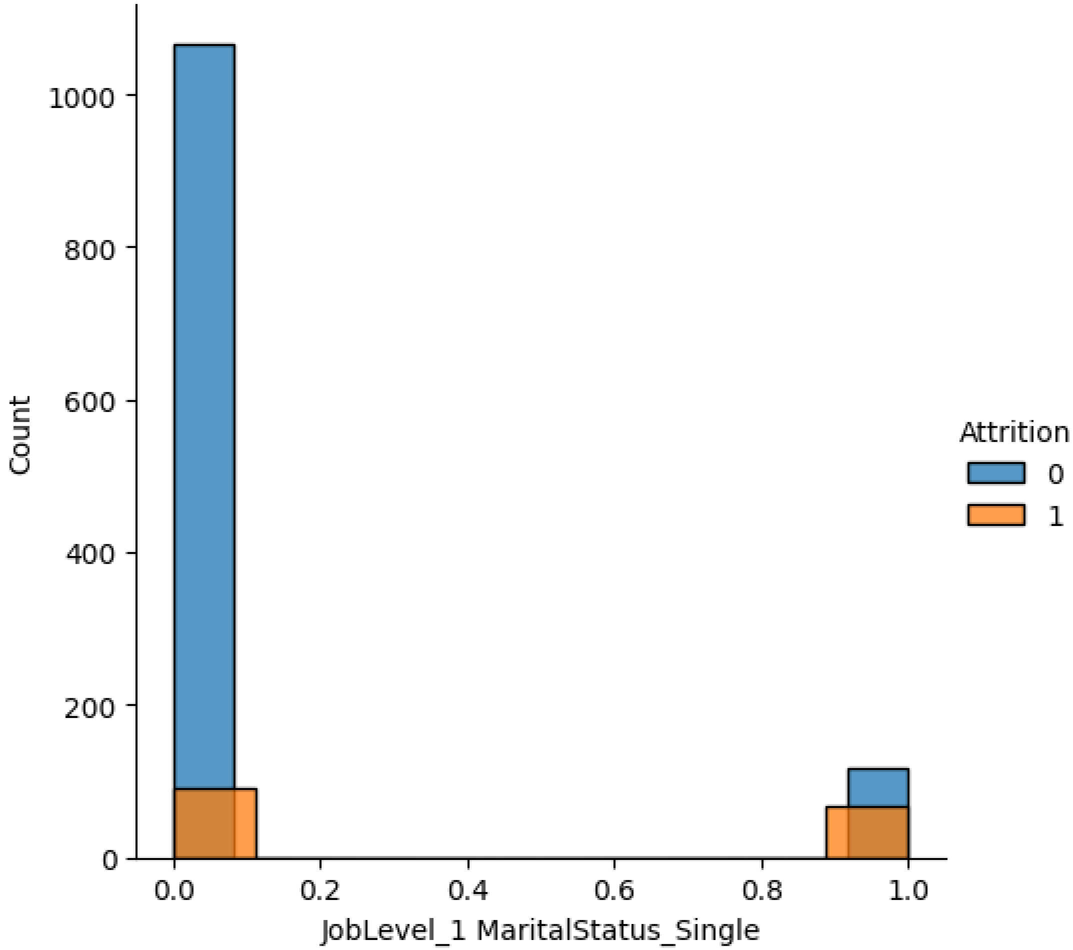
1000
800
600
400
200
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
DistanceFromHome_Even TrainingTimesLastYear_2
Attrition
0
1

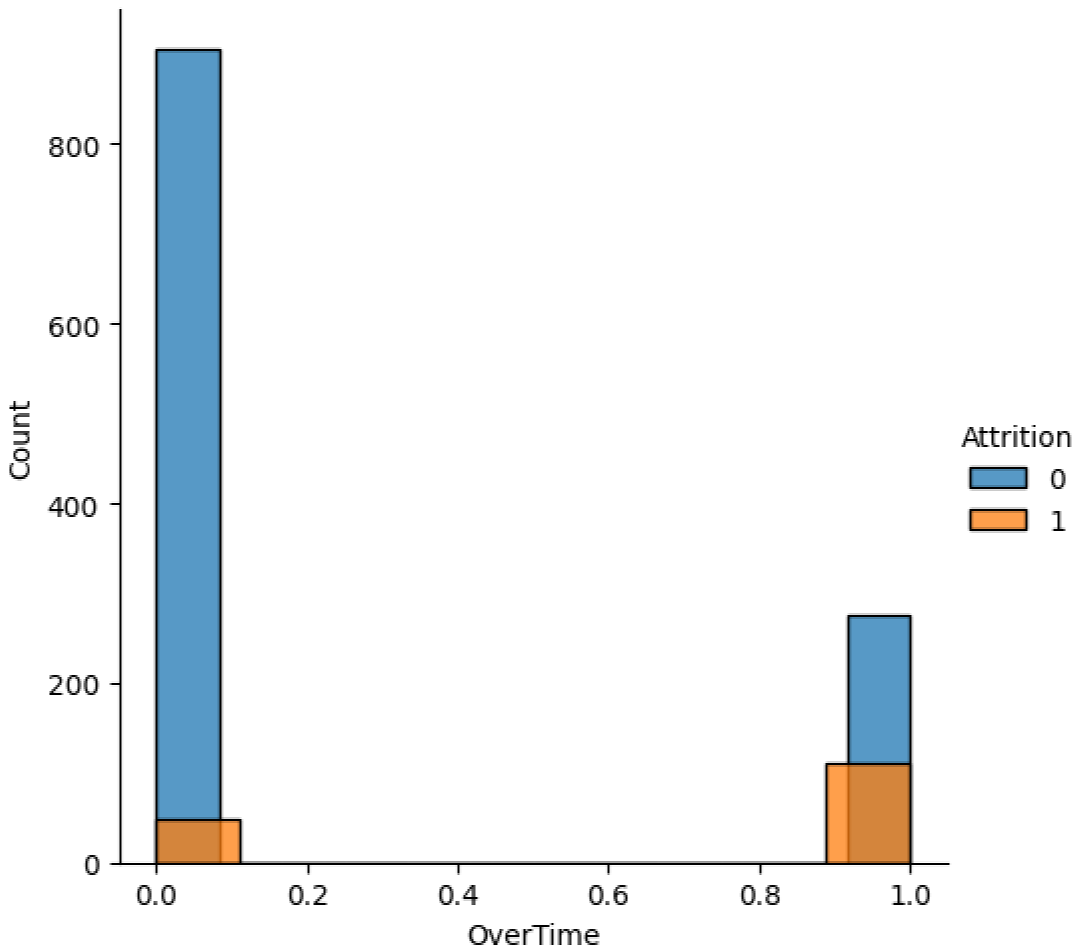


Count
250
200
150
100
50
0
2
4
6
8
10
12
14
16
JobInvolvement JobSatisfaction
Attrition
0
1

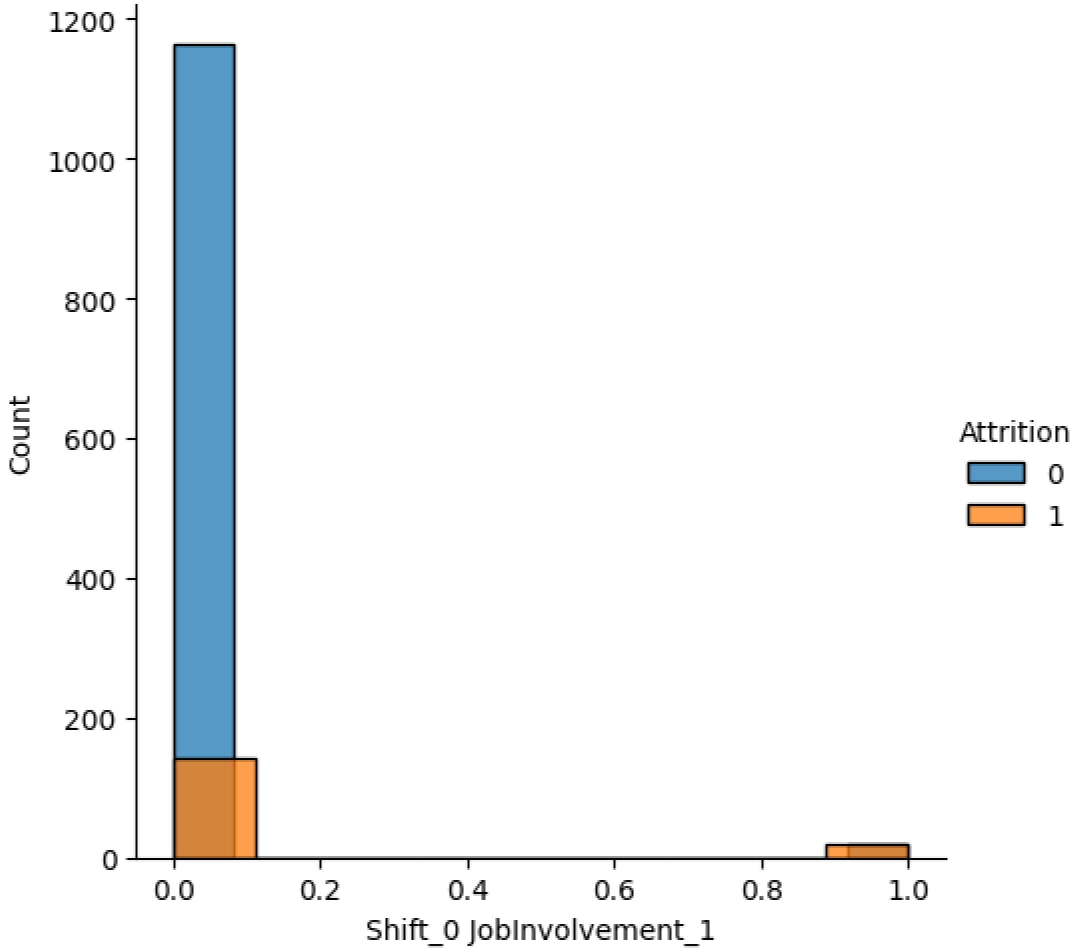

1000
800
600
400
200
0
Count
0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00
OverTime MaritalStatus
Attrition
0
1

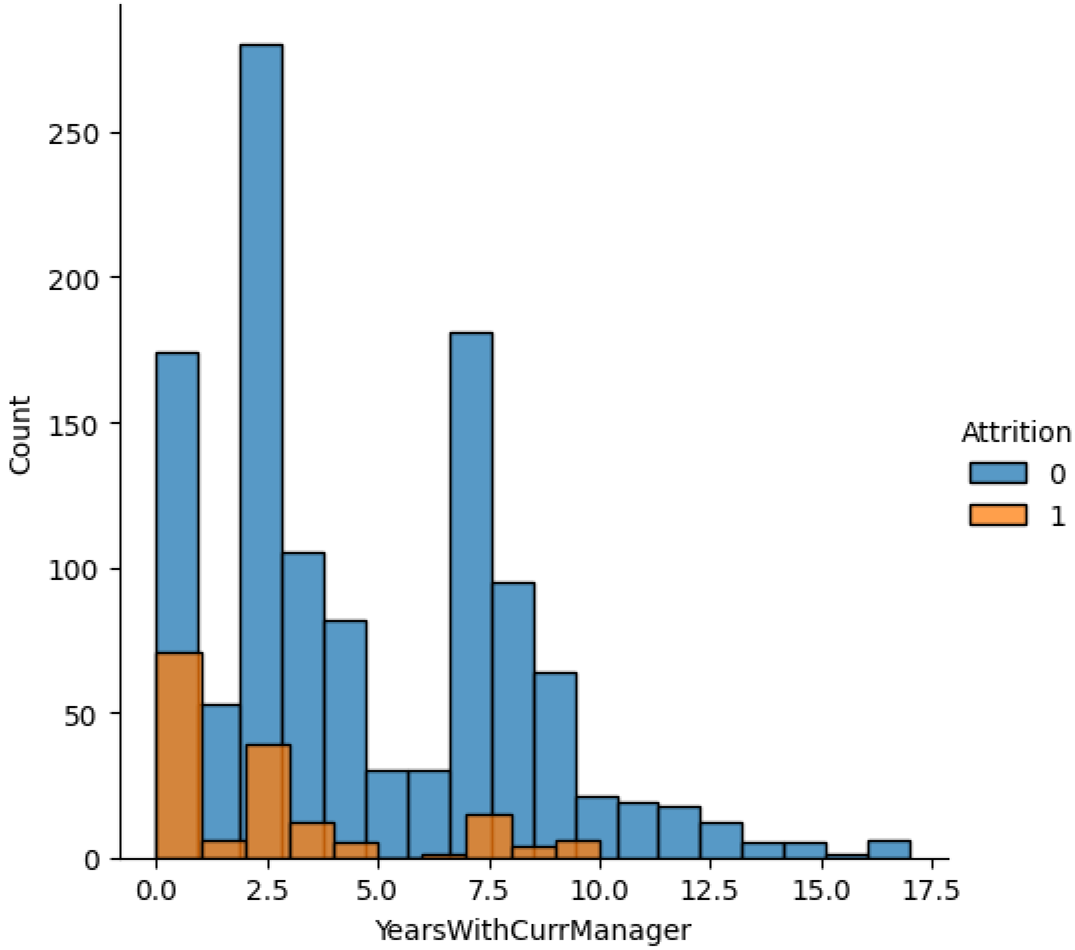

Count
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
BusinessTravel_Travel_Rarely Age_Even
Attrition
0
1

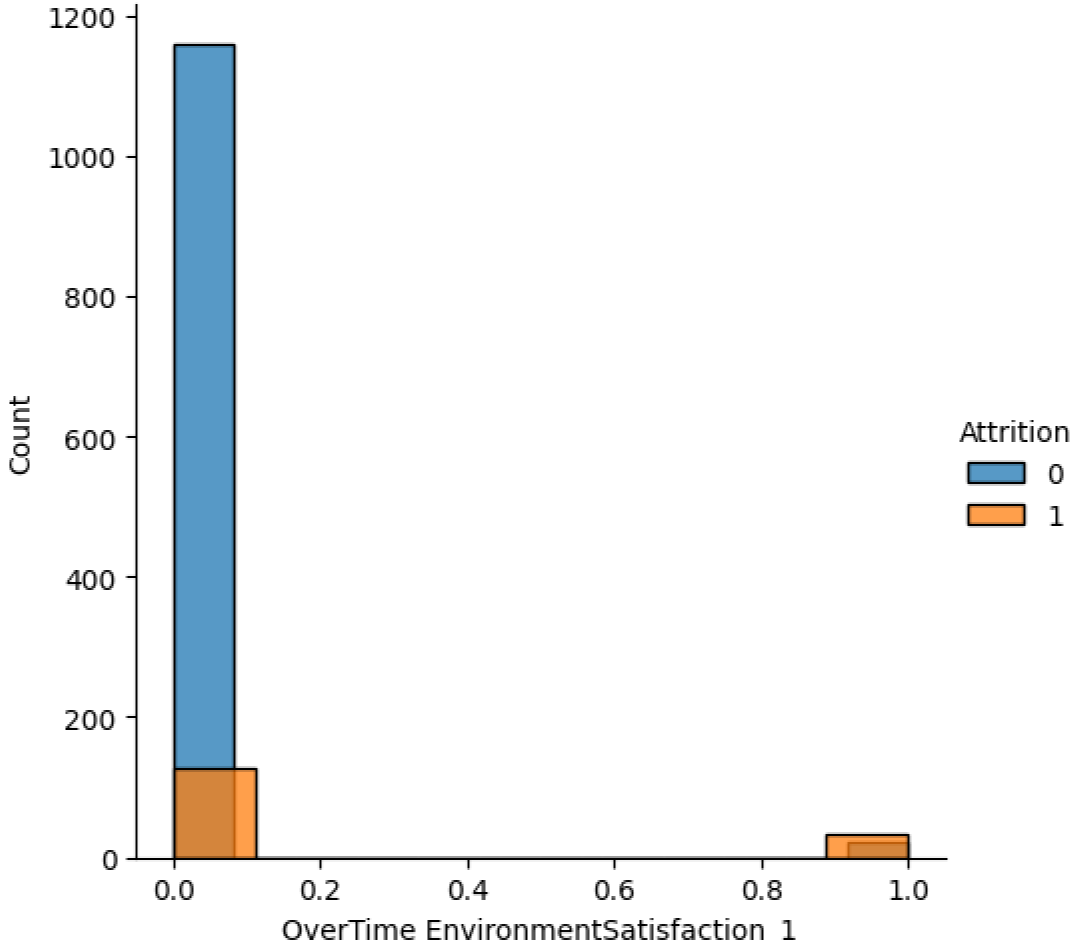

Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_1 MaritalStatus_Single
Attrition
0
1

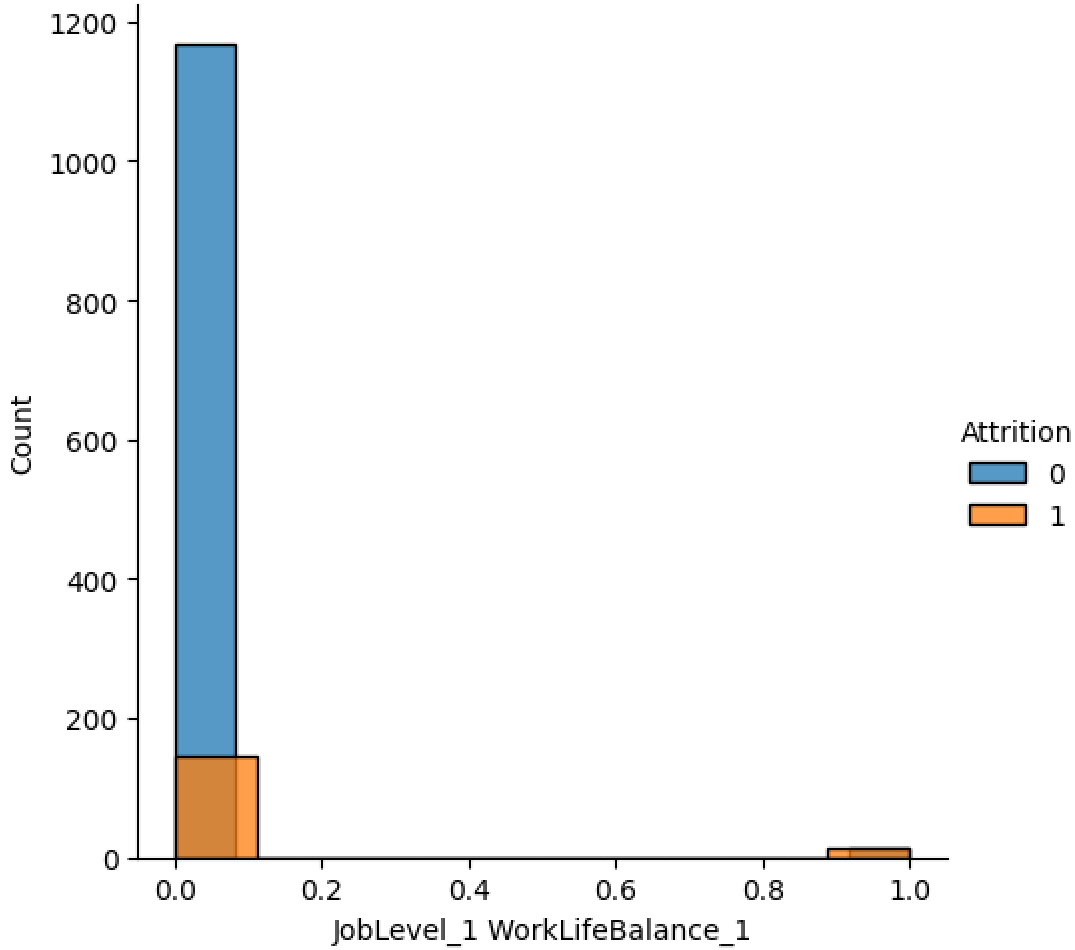


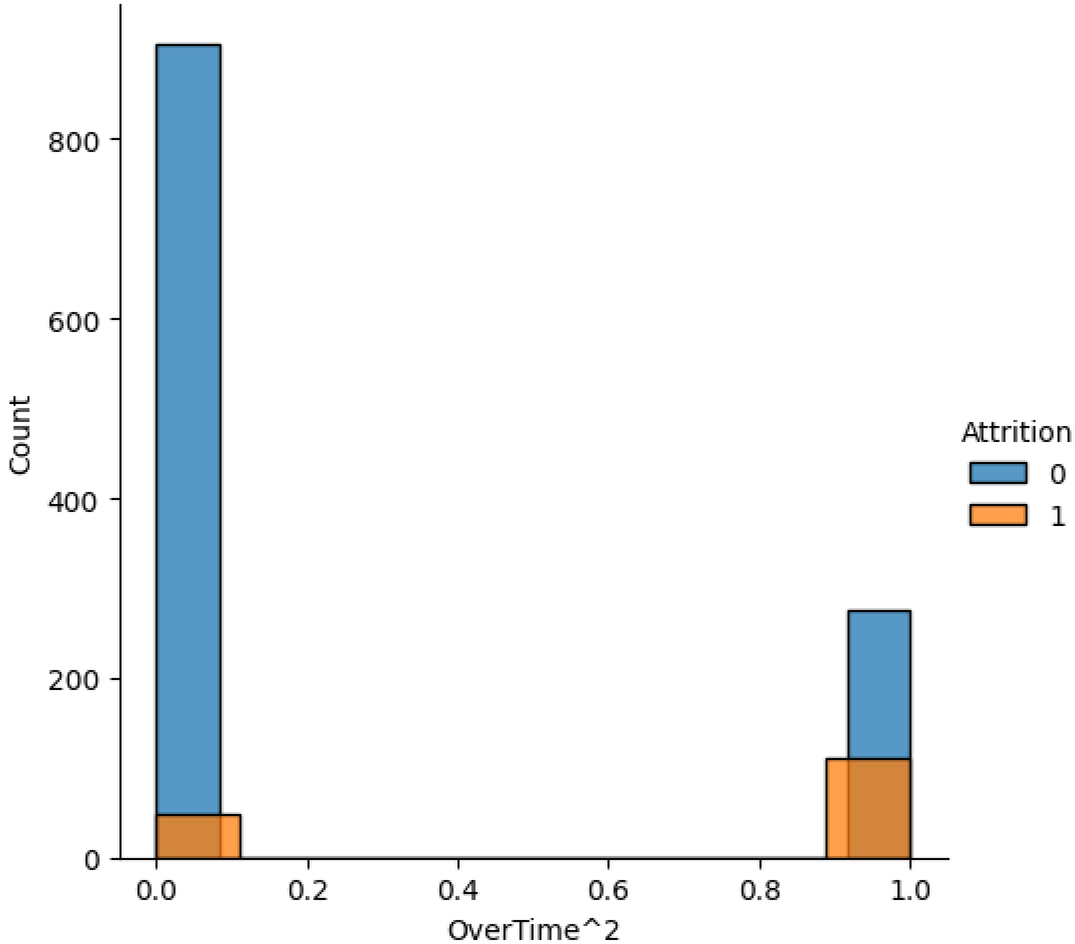
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
OverTime
Attrition
0
1


1200
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
Shift_0 JobInvolvement_1
Attrition
0
1

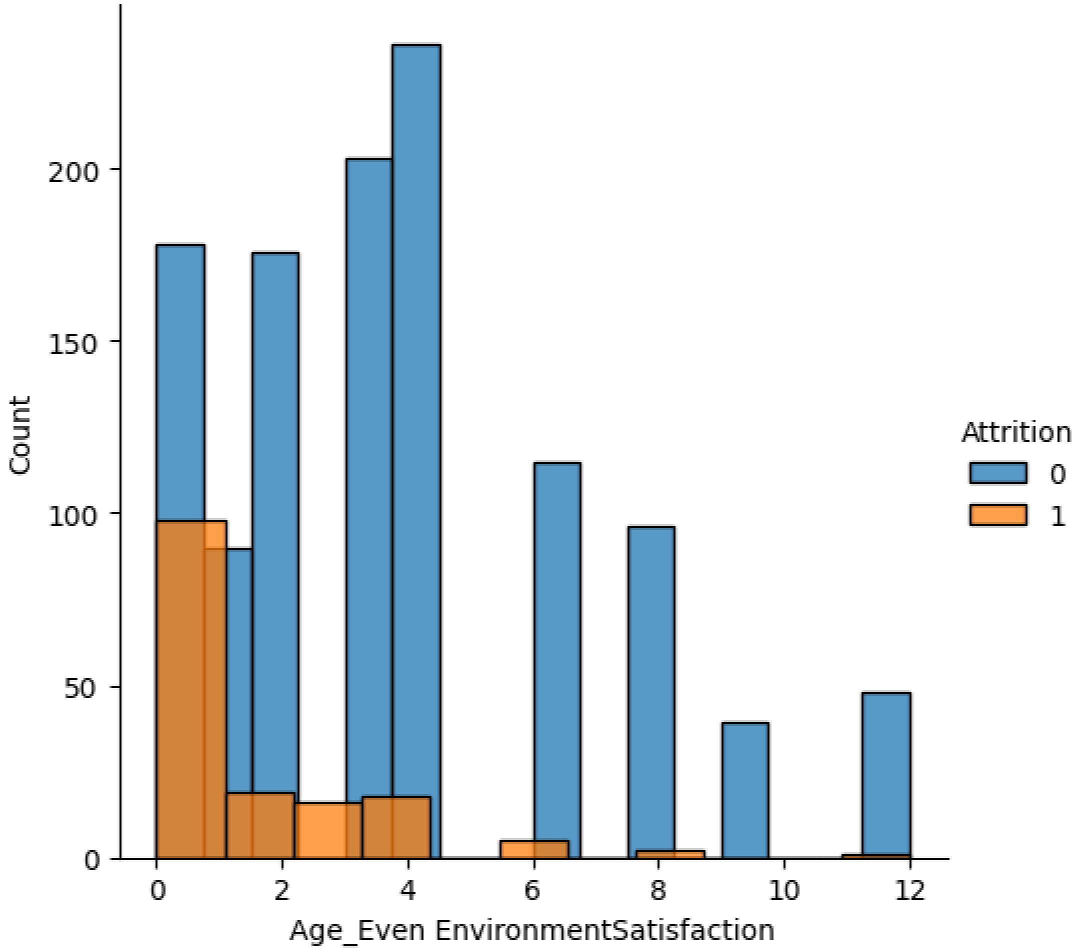

Count
250
200
150
100
50
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
YearsWithCurrManager
Attrition
0
1

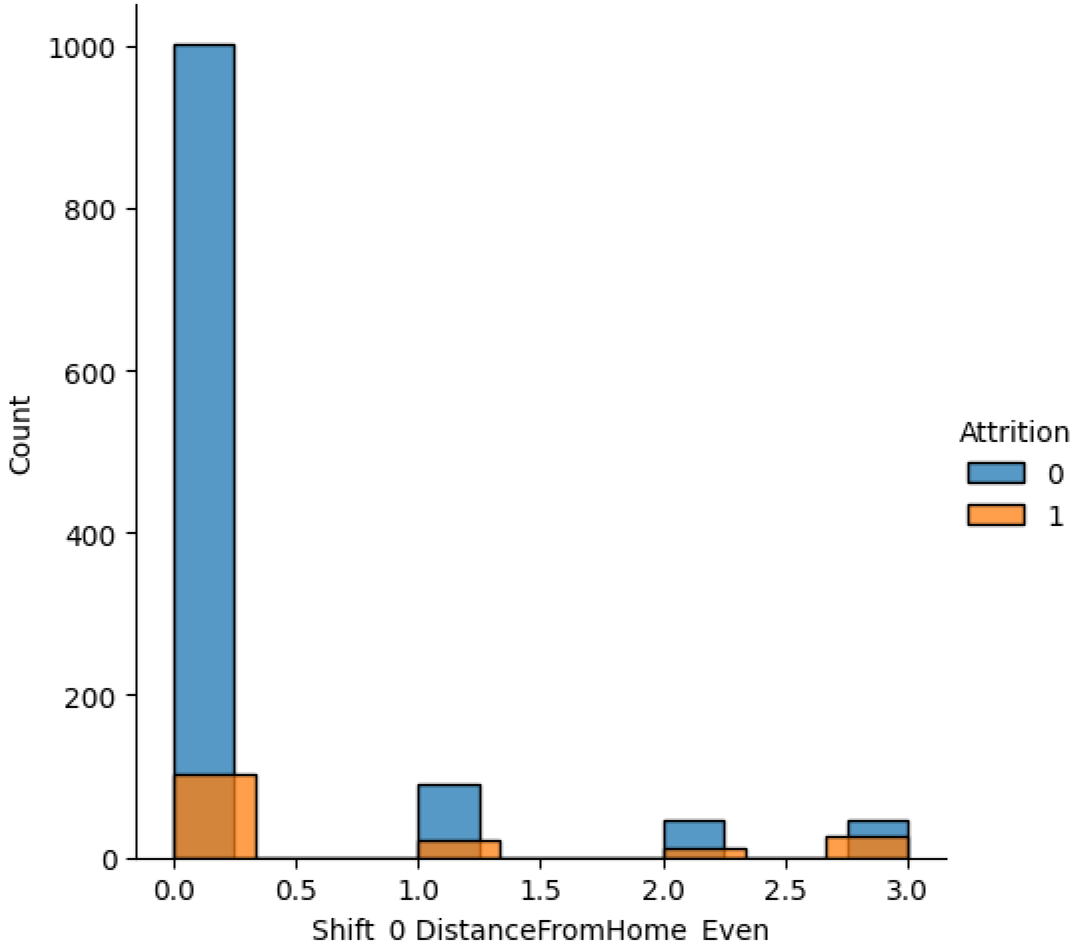

1200
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
OverTime EnvironmentSatisfaction_1
Attrition
0
1

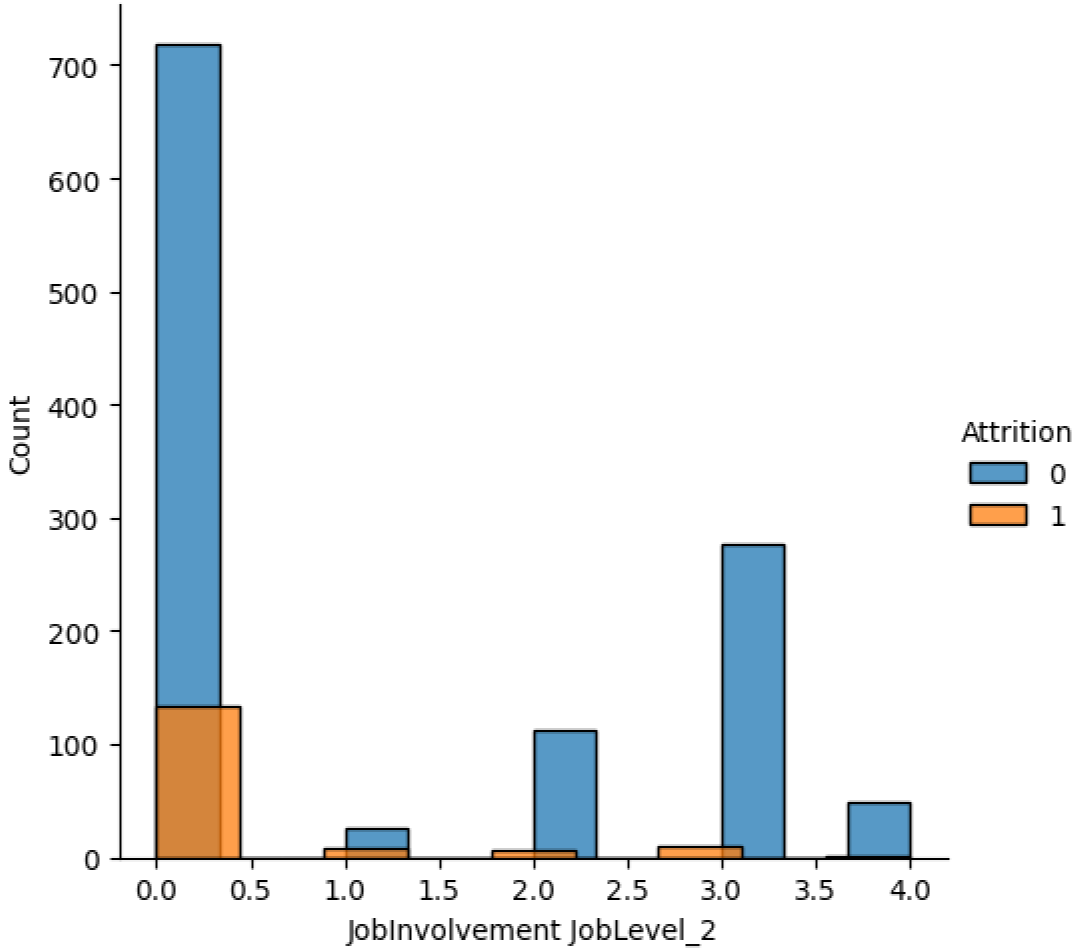

1200
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
JobLevel_1 WorkLifeBalance_1
Attrition
0
1

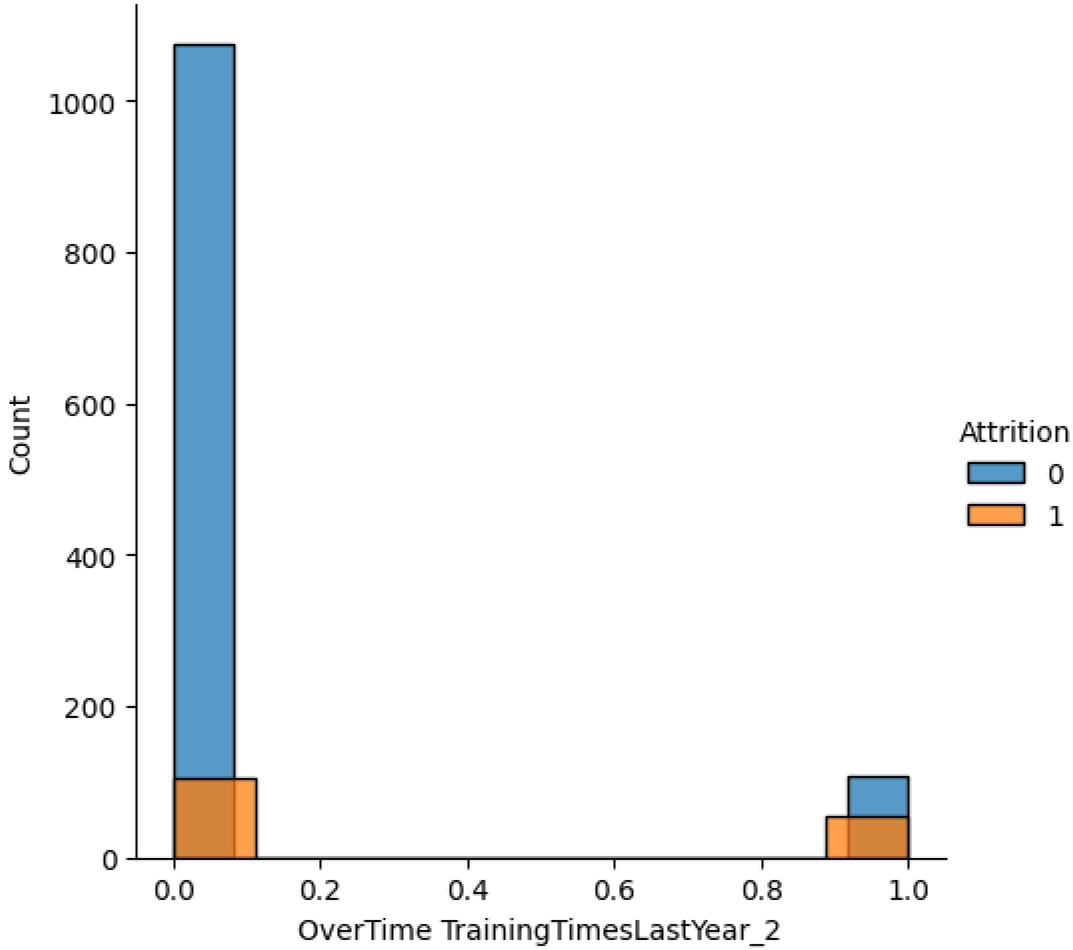


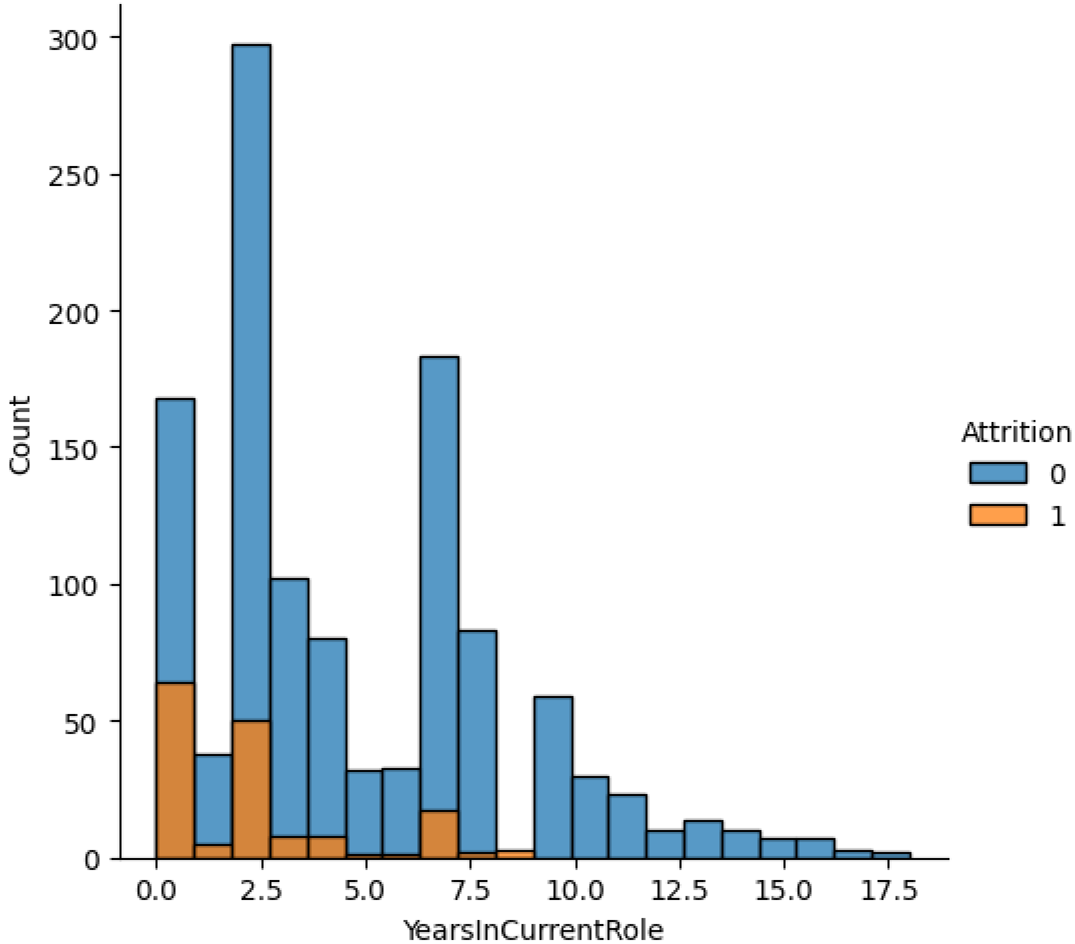
Count
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
OverTime^2
Attrition
0
1

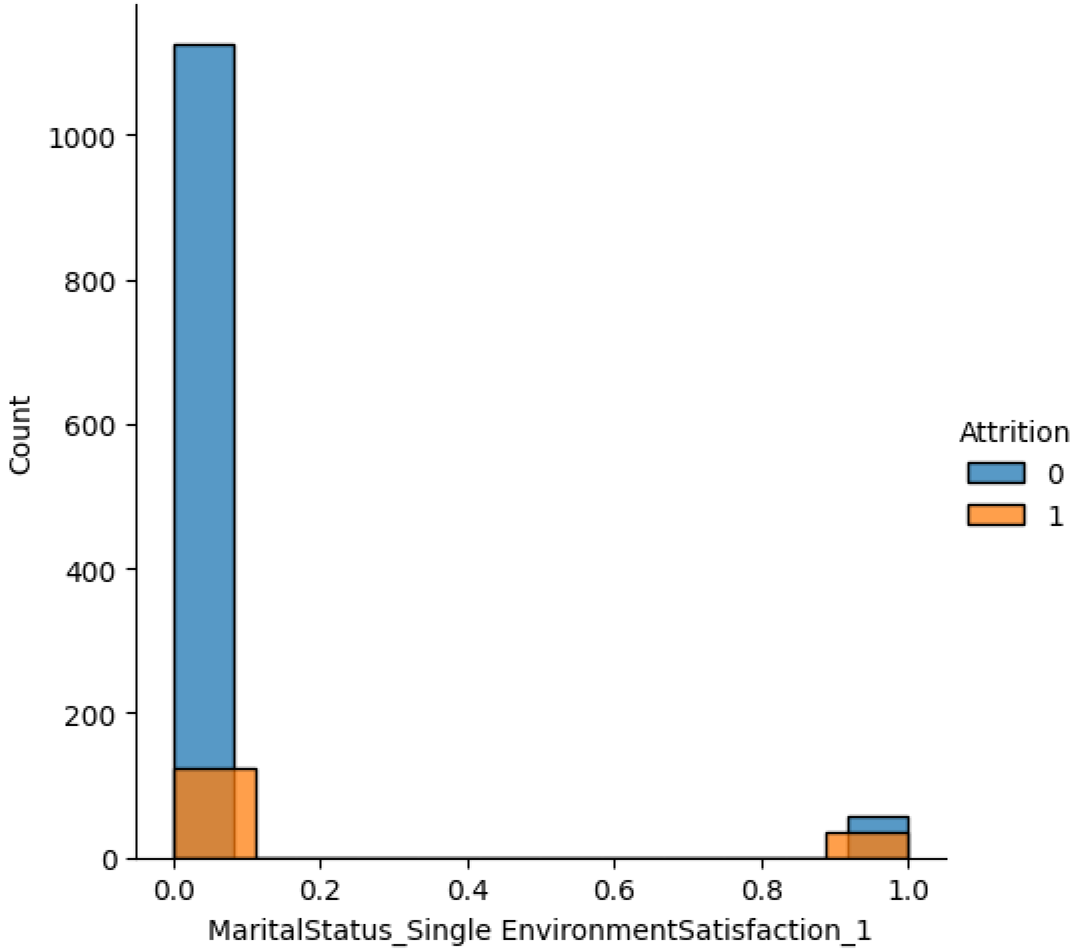

Count
200
150
100
50
0
0
2
4
6
8
10
12
Age_Even EnvironmentSatisfaction
Attrition
0
1


Count
1000
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Shift_0 DistanceFromHome_Even
Attrition
0
1

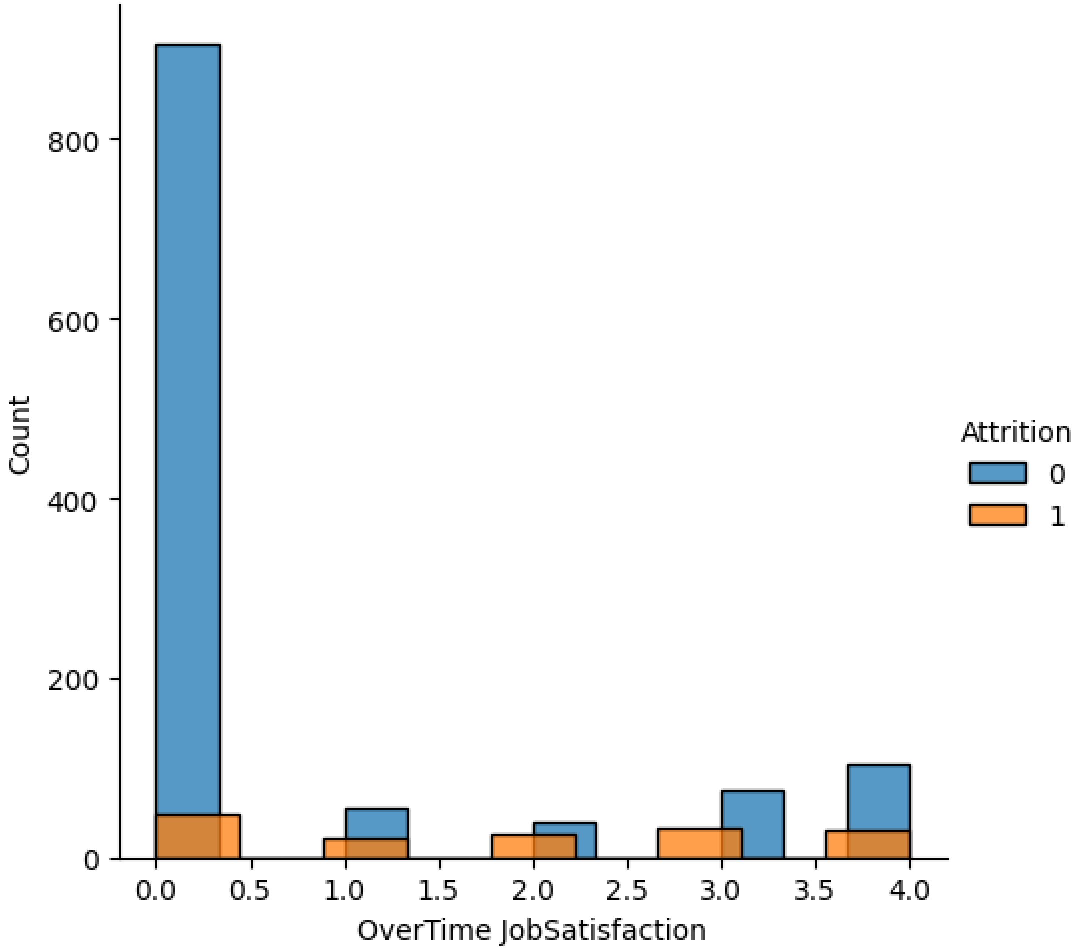

Count
700
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
JobInvolvement JobLevel_2
Attrition
0
1


Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
OverTime TrainingTimesLastYear_2
Attrition
0
1

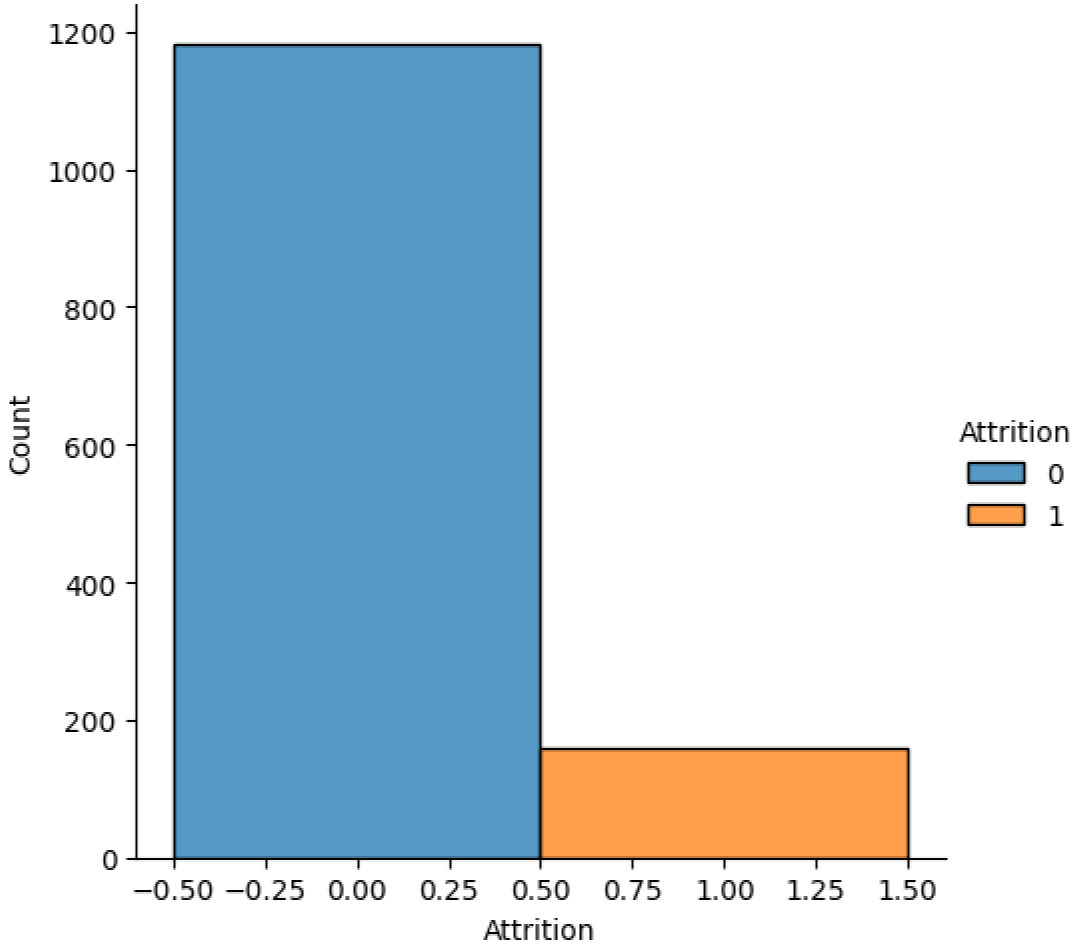

300
250
200
150
100
50
0
Count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
YearsInCurrentRole
Attrition
0
1



Count
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
MaritalStatus_Single EnvironmentSatisfaction_1
Attrition
0
1



Count
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
OverTime JobSatisfaction
Attrition
0
1




In [ ]:
```
# save the final poly data
fin_poly_data = pd.concat([uncorr_poly_data, labels], axis=1)
fin_poly_data.to_csv("uncorr20_poly_data.csv")
uncorr_poly_sub_data.to_csv("uncorr20_poly_sub_data.csv")
```