In [109…
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import plotly.express as px
import seaborn as sns
sns.set()
from scipy import stats
```

In [110…
```
df=pd.read_excel('E:\Attrition data.xlsx')
```

In [111…
```
df
```

Out[111]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	Gender	...	TotalWork
0	1	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	Female	...	
1	2	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	Female	...	
2	3	32	No	Travel_Frequently	Research & Development	17	4	Other	1	Male	...	
3	4	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	Male	...	
4	5	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	Male	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4405	4406	42	No	Travel_Rarely	Research & Development	5	4	Medical	1	Female	...	
4406	4407	29	No	Travel_Rarely	Research & Development	2	4	Medical	1	Male	...	
4407	4408	25	No	Travel_Rarely	Research & Development	25	2	Life Sciences	1	Male	...	
4408	4409	42	No	Travel_Rarely	Sales	18	2	Medical	1	Male	...	
4409	4410	40	No	Travel_Rarely	Research & Development	28	3	Medical	1	Male	...	

4410 rows × 29 columns

In [112…
```
df.shape
```

Out[112]:
```
(4410, 29)
```

In [113…
```
# Dataset has 4410 rows and 29 columns
```

In [114…
```
# Assessing Data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 29 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   EmployeeID               4410 non-null   int64  
 1   Age                      4410 non-null   int64  
 2   Attrition                4410 non-null   object 
 3   BusinessTravel           4410 non-null   object 
 4   Department               4410 non-null   object 
 5   DistanceFromHome         4410 non-null   int64  
 6   Education                4410 non-null   int64  
 7   EducationField           4410 non-null   object 
 8   EmployeeCount            4410 non-null   int64  
 9   Gender                   4410 non-null   object 
 10  JobLevel                 4410 non-null   int64  
 11  JobRole                  4410 non-null   object 
 12  MaritalStatus            4410 non-null   object 
 13  MonthlyIncome            4410 non-null   int64  
 14  NumCompaniesWorked       4391 non-null   float64
 15  Over18                   4410 non-null   object 
 16  PercentSalaryHike        4410 non-null   int64  
 17  StandardHours            4410 non-null   int64  
 18  StockOptionLevel         4410 non-null   int64  
 19  TotalWorkingYears        4401 non-null   float64
 20  TrainingTimesLastYear    4410 non-null   int64  
 21  YearsAtCompany           4410 non-null   int64  
 22  YearsSinceLastPromotion  4410 non-null   int64  
 23  YearsWithCurrManager     4410 non-null   int64  
 24  EnvironmentSatisfaction  4385 non-null   float64
 25  JobSatisfaction          4390 non-null   float64
 26  WorkLifeBalance          4372 non-null   float64
 27  JobInvolvement           4410 non-null   int64  
 28  PerformanceRating        4410 non-null   int64  
dtypes: float64(5), int64(16), object(8)
memory usage: 999.3+ KB
```

In [115…

```
# Check for duplicate data
df.duplicated().sum()
```

Out[115]:

```
0
```

In [116…

```
#There is no duplicate value in the dataset.
```

In [117…
```
df.columns
```

Out[117]:
```
Index(['EmployeeID', 'Age', 'Attrition', 'BusinessTravel', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
       'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager',
       'EnvironmentSatisfaction', 'JobSatisfaction', 'WorkLifeBalance',
       'JobInvolvement', 'PerformanceRating'],
      dtype='object')
```

In [118…
```
df.dtypes
```

Out[118]:
```
EmployeeID                int64
Age                       int64
Attrition                object
BusinessTravel           object
Department               object
DistanceFromHome          int64
Education                 int64
EducationField           object
EmployeeCount             int64
Gender                   object
JobLevel                  int64
JobRole                  object
MaritalStatus            object
MonthlyIncome             int64
NumCompaniesWorked      float64
Over18                   object
PercentSalaryHike         int64
StandardHours             int64
StockOptionLevel          int64
TotalWorkingYears       float64
TrainingTimesLastYear     int64
YearsAtCompany            int64
YearsSinceLastPromotion   int64
YearsWithCurrManager      int64
EnvironmentSatisfaction  float64
JobSatisfaction          float64
WorkLifeBalance          float64
JobInvolvement            int64
PerformanceRating         int64
dtype: object
```

In [119…
```
# Handling missing data
df.isnull().sum()
```

Out[119]:
```
EmployeeID                0
Age                       0
Attrition                 0
BusinessTravel            0
Department                0
DistanceFromHome          0
Education                 0
EducationField            0
EmployeeCount             0
Gender                    0
JobLevel                  0
JobRole                   0
MaritalStatus             0
MonthlyIncome             0
NumCompaniesWorked       19
Over18                    0
PercentSalaryHike         0
StandardHours             0
StockOptionLevel          0
TotalWorkingYears         9
TrainingTimesLastYear     0
YearsAtCompany            0
YearsSinceLastPromotion   0
YearsWithCurrManager      0
EnvironmentSatisfaction  25
JobSatisfaction          20
WorkLifeBalance          38
JobInvolvement            0
PerformanceRating         0
dtype: int64
```

In [120…
```
df['NumCompaniesWorked'].fillna(df['NumCompaniesWorked'].mean(), inplace=True)
df['EnvironmentSatisfaction'].fillna(df['EnvironmentSatisfaction'].mean(), inplace=True)
df['WorkLifeBalance'].fillna(df['WorkLifeBalance'].mean(), inplace=True)
df['JobSatisfaction'].fillna(df['JobSatisfaction'].mean(), inplace=True)
df['TotalWorkingYears'].fillna(df['JobSatisfaction'].mean(), inplace=True)
```

In [121…
```
df.isnull().sum()
```

Out[121]:
```
EmployeeID                0
Age                       0
Attrition                 0
BusinessTravel            0
Department                0
DistanceFromHome          0
Education                 0
EducationField            0
EmployeeCount             0
Gender                    0
JobLevel                  0
JobRole                   0
MaritalStatus             0
MonthlyIncome             0
NumCompaniesWorked        0
Over18                    0
PercentSalaryHike         0
StandardHours             0
StockOptionLevel          0
TotalWorkingYears         0
TrainingTimesLastYear     0
YearsAtCompany            0
YearsSinceLastPromotion   0
YearsWithCurrManager      0
EnvironmentSatisfaction   0
JobSatisfaction           0
WorkLifeBalance           0
JobInvolvement            0
PerformanceRating         0
dtype: int64
```

In [122…
```
# checking unique values in categorical columns
df['BusinessTravel'].value_counts()
```

Out[122]:
```
BusinessTravel
Travel_Rarely        3129
Travel_Frequently     831
Non-Travel            450
Name: count, dtype: int64
```

In [123…
```
df['Department'].value_counts()
```

Out[123]:
```
Department
Research & Development    2883
Sales                     1338
Human Resources            189
Name: count, dtype: int64
```

In [124…
```
df['EducationField'].value_counts()
```

Out[124]:
```
EducationField
Life Sciences       1818
Medical             1392
Marketing            477
Technical Degree     396
Other                246
Human Resources       81
Name: count, dtype: int64
```

In [125…
```
df['Gender'].value_counts()
```

Out[125]:
```
Gender
Male      2646
Female    1764
Name: count, dtype: int64
```

In [126…
```
df['JobRole'].value_counts()
```

Out[126]:
```
JobRole
Sales Executive              978
Research Scientist           876
Laboratory Technician        777
Manufacturing Director       435
Healthcare Representative    393
Manager                      306
Sales Representative         249
Research Director            240
Human Resources              156
Name: count, dtype: int64
```

In [127…
```
df['MaritalStatus'].value_counts()
```

Out[127]:
```
MaritalStatus
Married     2019
Single      1410
Divorced     981
Name: count, dtype: int64
```

In [128…
```
df['Attrition'].value_counts()
```

Out[128]:
```
Attrition
No     3699
Yes     711
Name: count, dtype: int64
```

In [129…
```
df['Over18'].value_counts()
```

Out[129]:
```
Over18
Y    4410
Name: count, dtype: int64
```

In [130…
```
df.describe(include="all")
```

Out[130]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	Gender	...
count	4410.000000	4410.000000	4410	4410	4410	4410.000000	4410.000000	4410	4410.0	4410	...
unique	NaN	NaN	2	3	3	NaN	NaN	6	NaN	2	...
top	NaN	NaN	No	Travel_Rarely	Research & Development	NaN	NaN	Life Sciences	NaN	Male	...
freq	NaN	NaN	3699	3129	2883	NaN	NaN	1818	NaN	2646	...
mean	2205.500000	36.923810	NaN	NaN	NaN	9.192517	2.912925	NaN	1.0	NaN	...
std	1273.201673	9.133301	NaN	NaN	NaN	8.105026	1.023933	NaN	0.0	NaN	...
min	1.000000	18.000000	NaN	NaN	NaN	1.000000	1.000000	NaN	1.0	NaN	...
25%	1103.250000	30.000000	NaN	NaN	NaN	2.000000	2.000000	NaN	1.0	NaN	...
50%	2205.500000	36.000000	NaN	NaN	NaN	7.000000	3.000000	NaN	1.0	NaN	...
75%	3307.750000	43.000000	NaN	NaN	NaN	14.000000	4.000000	NaN	1.0	NaN	...
max	4410.000000	60.000000	NaN	NaN	NaN	29.000000	5.000000	NaN	1.0	NaN	...

11 rows × 29 columns

In [131…
```
# Reassign response variable
df['Attrition'] = df['Attrition'].apply(lambda x: 0 if x =='No' else 1)
```

In [132…
```
df.nunique().sort_values()
```

Out[132]:
```
Over18                      1
StandardHours               1
EmployeeCount               1
PerformanceRating           2
Gender                      2
Attrition                   2
BusinessTravel              3
MaritalStatus               3
Department                  3
JobInvolvement              4
StockOptionLevel            4
JobSatisfaction             5
Education                   5
JobLevel                    5
EnvironmentSatisfaction     5
WorkLifeBalance             5
EducationField              6
TrainingTimesLastYear       7
JobRole                     9
NumCompaniesWorked         11
PercentSalaryHike          15
YearsSinceLastPromotion    16
YearsWithCurrManager       18
DistanceFromHome           29
YearsAtCompany             37
TotalWorkingYears          41
Age                        43
MonthlyIncome            1349
EmployeeID               4410
dtype: int64
```

In [133…
```
df
```

Out[133]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	Gender	...	TotalWork
0	1	51	0	Travel_Rarely	Sales	6	2	Life Sciences	1	Female	...	
1	2	31	1	Travel_Frequently	Research & Development	10	1	Life Sciences	1	Female	...	
2	3	32	0	Travel_Frequently	Research & Development	17	4	Other	1	Male	...	
3	4	38	0	Non-Travel	Research & Development	2	5	Life Sciences	1	Male	...	1
4	5	32	0	Travel_Rarely	Research & Development	10	1	Medical	1	Male	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4405	4406	42	0	Travel_Rarely	Research & Development	5	4	Medical	1	Female	...	1
4406	4407	29	0	Travel_Rarely	Research & Development	2	4	Medical	1	Male	...	1
4407	4408	25	0	Travel_Rarely	Research & Development	25	2	Life Sciences	1	Male	...	
4408	4409	42	0	Travel_Rarely	Sales	18	2	Medical	1	Male	...	1
4409	4410	40	0	Travel_Rarely	Research & Development	28	3	Medical	1	Male	...	

4410 rows × 29 columns

In [134...
```
# Drop useless features and rename the dataframe
df1=df.drop(columns = ['Over18','StandardHours','EmployeeCount'], axis = 1)
```

In [135...
```
df1
```

Out[135]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	Gender	JobLevel	...	TotalWorkingYea
0	1	51	0	Travel_Rarely	Sales	6	2	Life Sciences	Female	1	...	1.00000
1	2	31	1	Travel_Frequently	Research & Development	10	1	Life Sciences	Female	1	...	6.00000
2	3	32	0	Travel_Frequently	Research & Development	17	4	Other	Male	4	...	5.00000
3	4	38	0	Non-Travel	Research & Development	2	5	Life Sciences	Male	3	...	13.00000
4	5	32	0	Travel_Rarely	Research & Development	10	1	Medical	Male	1	...	9.00000
...	...	...	...	...	...	...	...	...	...	...	...	
4405	4406	42	0	Travel_Rarely	Research & Development	5	4	Medical	Female	1	...	10.00000
4406	4407	29	0	Travel_Rarely	Research & Development	2	4	Medical	Male	1	...	10.00000
4407	4408	25	0	Travel_Rarely	Research & Development	25	2	Life Sciences	Male	2	...	5.00000
4408	4409	42	0	Travel_Rarely	Sales	18	2	Medical	Male	1	...	10.00000
4409	4410	40	0	Travel_Rarely	Research & Development	28	3	Medical	Male	2	...	2.72824

4410 rows × 26 columns

In [136…

```
# Changing object types to categories
cols = ['BusinessTravel','Department','EducationField','Gender','JobRole','MaritalStatus']
for col in cols:
    df1[col] = df1[col].astype('category')
```

Exploratory Data Analysis

I will try to analyze visually the trends in how and why employees are quitting their jobs. For that,

I will deep dive into the details about features and their relationships between each other.

In [137…
```
#Target Variable:
```

In [138…
```
df1['Target']=df1['Attrition'].apply(lambda x: 'Currently Working in Company' if x == 0  else 'Left the Company')
```

In [139…
```
df1.head(10)
```

Out[139]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	Gender	JobLevel	...	TrainingTimesLastYea
0	1	51	0	Travel_Rarely	Sales	6	2	Life Sciences	Female	1	...	
1	2	31	1	Travel_Frequently	Research & Development	10	1	Life Sciences	Female	1	...	
2	3	32	0	Travel_Frequently	Research & Development	17	4	Other	Male	4	...	
3	4	38	0	Non-Travel	Research & Development	2	5	Life Sciences	Male	3	...	
4	5	32	0	Travel_Rarely	Research & Development	10	1	Medical	Male	1	...	
5	6	46	0	Travel_Rarely	Research & Development	8	3	Life Sciences	Female	4	...	
6	7	28	1	Travel_Rarely	Research & Development	11	2	Medical	Male	2	...	
7	8	29	0	Travel_Rarely	Research & Development	18	3	Life Sciences	Male	2	...	
8	9	31	0	Travel_Rarely	Research & Development	1	3	Life Sciences	Male	3	...	
9	10	25	0	Non-Travel	Research & Development	7	4	Medical	Female	4	...	

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	Gender	JobLevel	...	TrainingTimesLastYea

10 rows × 27 columns

In [140…
```
df1.groupby('EmployeeID')['Target'].sum().value_counts()
```

Out[140]:
```
Target
Currently Working in Company    3699
Left the Company                 711
Name: count, dtype: int64
```

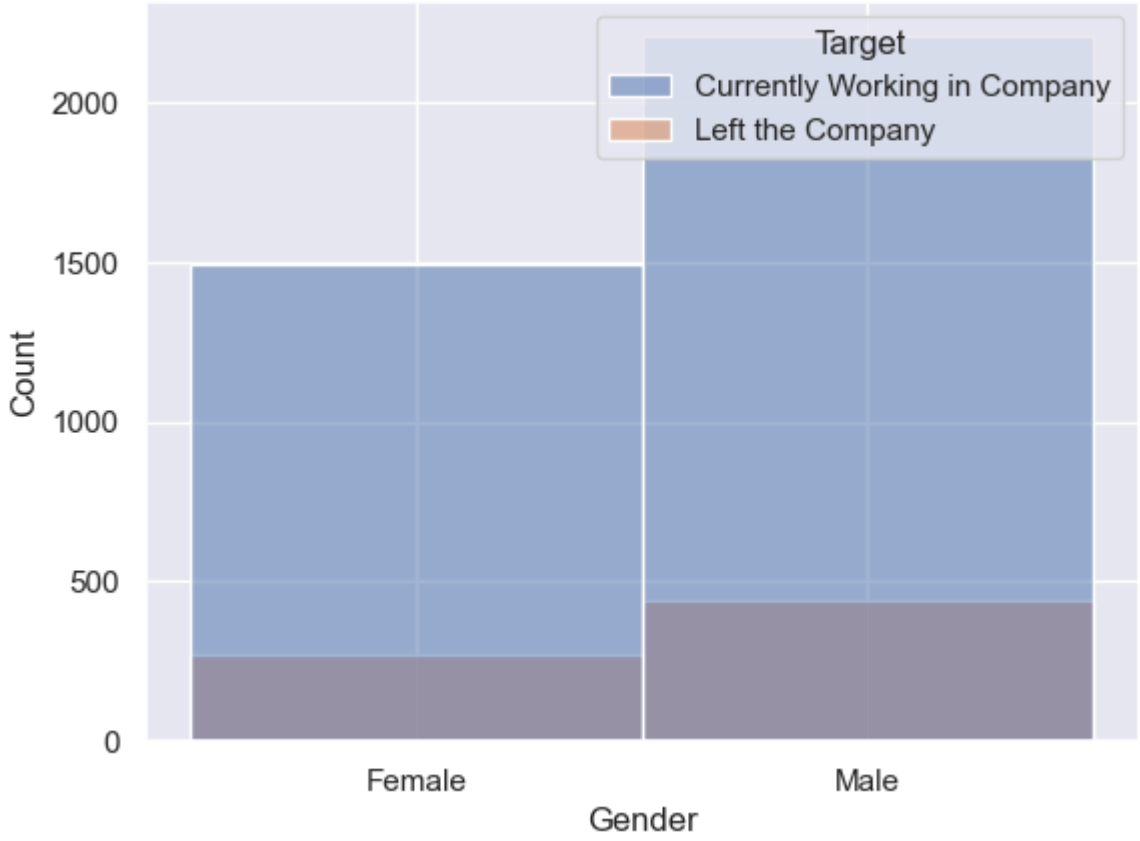
In [ ]:
```
```

In [141…
```
# plot showing employee count based on working in company and the others who left
sns.histplot(data=df1, x='Target')
```

Out[141]:
```
<Axes: xlabel='Target', ylabel='Count'>
```

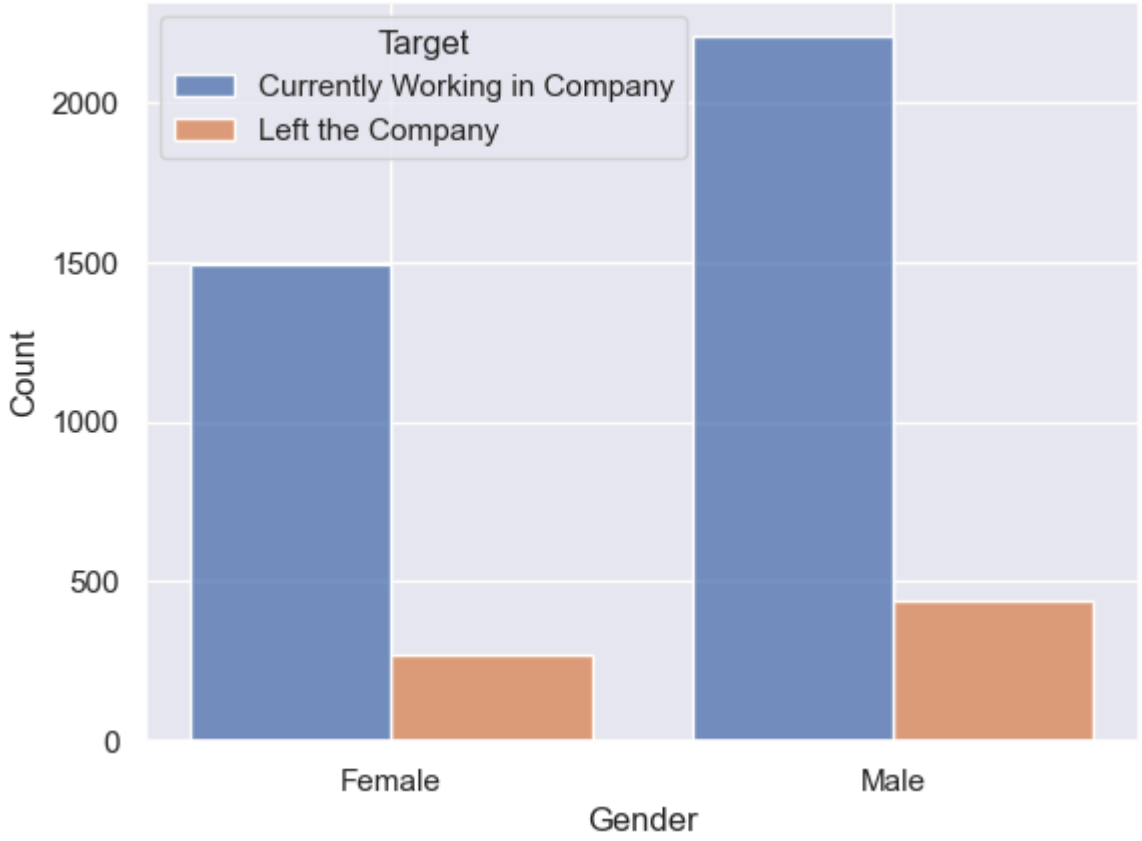
In [142…
```
# Gender Analysis
sns.histplot(data=df1, x='Gender',hue='Target')
```

Out[142]:
```
<Axes: xlabel='Gender', ylabel='Count'>
```



In [143…
```
sns.histplot(data=df1, x='Gender',hue='Target',multiple="dodge", shrink=.8)
```

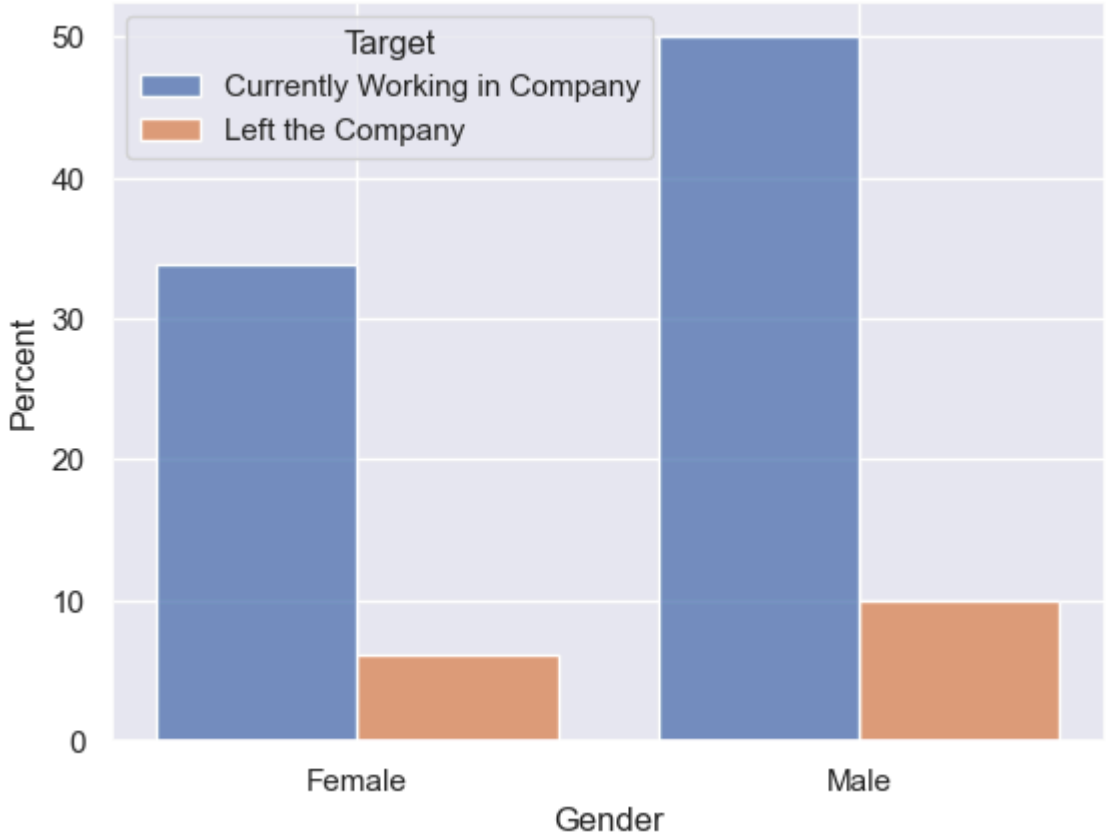
Out[143]:
```
<Axes: xlabel='Gender', ylabel='Count'>
```



In [144…
```
# It means no. of males working in company are more than females.
```

In [145…
```
sns.histplot(data=df1, x='Gender',stat="percent",hue='Target',multiple="dodge", shrink=.8, discrete=True)
```

Out[145]:
```
<Axes: xlabel='Gender', ylabel='Percent'>
```



In [146…
```
# more no. of males left company than female employees.
```

In [147…
```
# JobLevel vs Target(Attrition)
```

ax = sns.countplot(x="JobLevel", hue="Target", data=df1) for p in ax.patches: ax.annotate('{}'.format(p.get_height()), (p.get_x(), p.get_height()+1))

In [148…
```
# More No. Of Employees  left the company with lower job Level.
```

In [149…
```
sns.histplot(data=df1, x='Age',hue="Target")
```

Out[149]:
```
<Axes: xlabel='Age', ylabel='Count'>
```

In [150…
```
# Most of the Employees who left the company are between 25 to 35 age group.
```

In [151…
```
df1.groupby('Target')['Department'].value_counts()
```

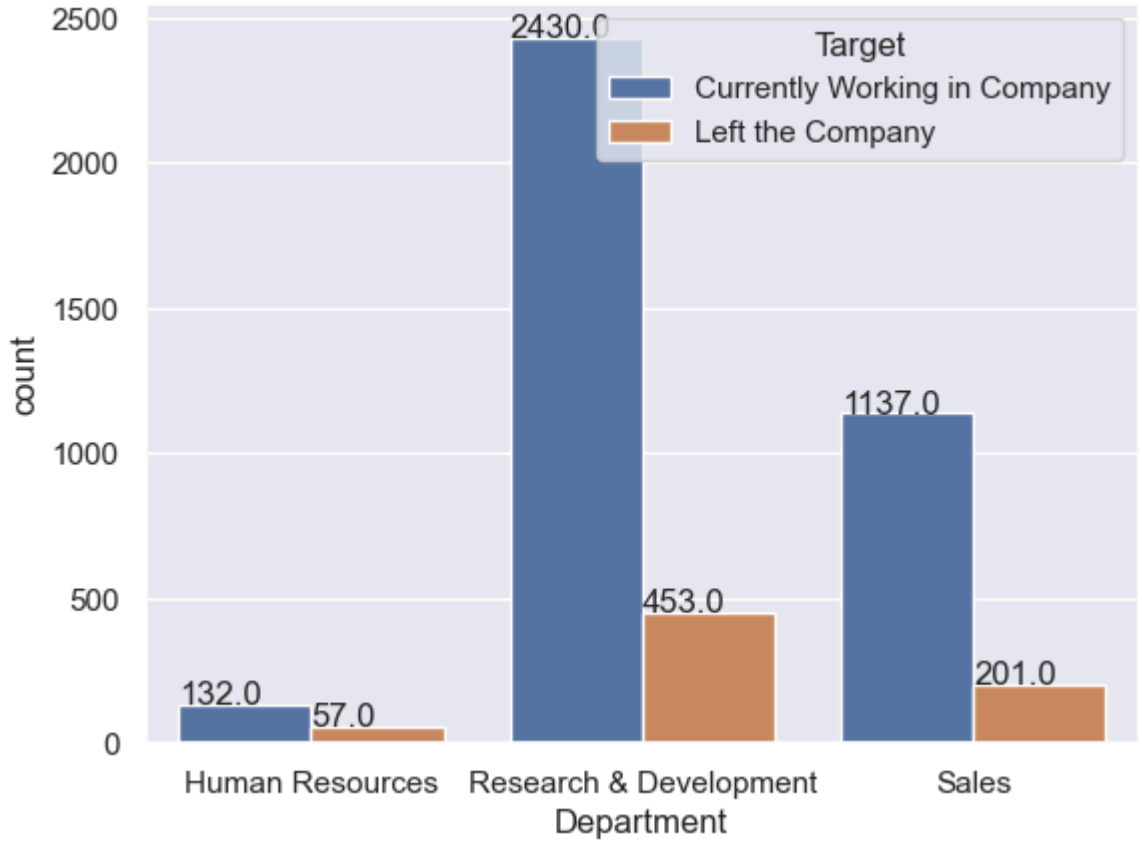
Out[151]:
```
Target                       Department
Currently Working in Company  Research & Development    2430
                              Sales                     1137
                              Human Resources            132
Left the Company              Research & Development     453
                              Sales                      201
                              Human Resources             57
Name: count, dtype: int64
```

In [ ]:

In [ ]:

In [152…
```
# Department vs Target(Attrition)
```

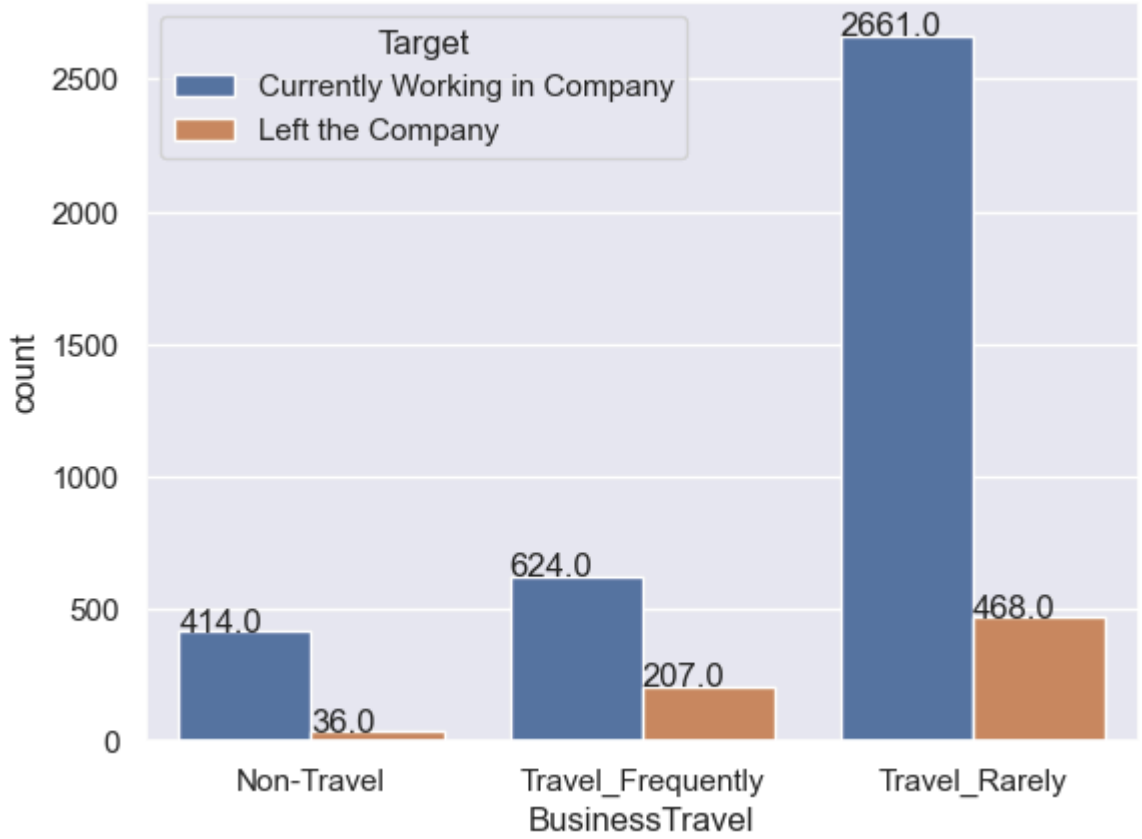
In [153…
```
ax = sns.countplot(x="Department", hue="Target", data=df1)
for p in ax.patches:
    ax.annotate('{}'.format(p.get_height()), (p.get_x(), p.get_height()+1))
```



In [154…
```
# Inference:
#1. 30.16% of employess from human resources department are likely to quit.
#2. 15.7% of employess from research and development department are likely to quit.
#3. 15.02% of employees from sales department are likely to quit.
```

In [155…
```
# BusinessTravel  vs Target(Attrition)
```

In [156…
```
ax = sns.countplot(x="BusinessTravel", hue="Target", data=df1)
for p in ax.patches:
    ax.annotate('{}'.format(p.get_height()), (p.get_x(), p.get_height()+1))
```



In [167…
```
# 1. 24.91% of total Travel Frequently employees are likely to quit.
# 2. 14.96%  of total Travel Rarely employees are likely to quit.
# 3. 8% of total non- travelling employees are likely to quit.
```

In [158…
```
# DistanceFromHome vs Target(Attrition)
```

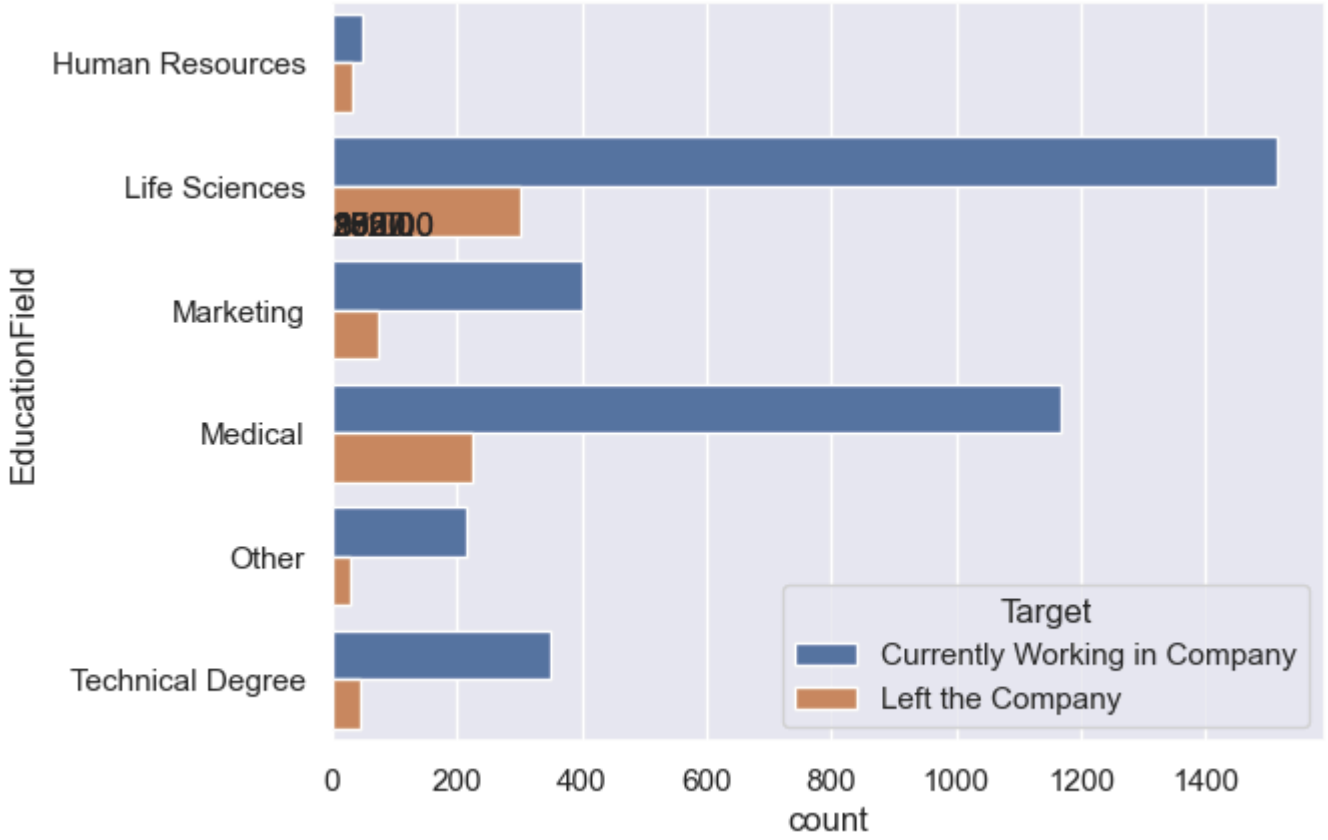
In [159…
```
sns.histplot(data=df1, x='DistanceFromHome',hue="Target")
```

Out[159]: <Axes: xlabel='DistanceFromHome', ylabel='Count'>

In [160…
```
# People who live closeby (0-10 miles) are likely to quit more based on the data
```

In [161…
```
# EducationField vs Target(Attrition)
```

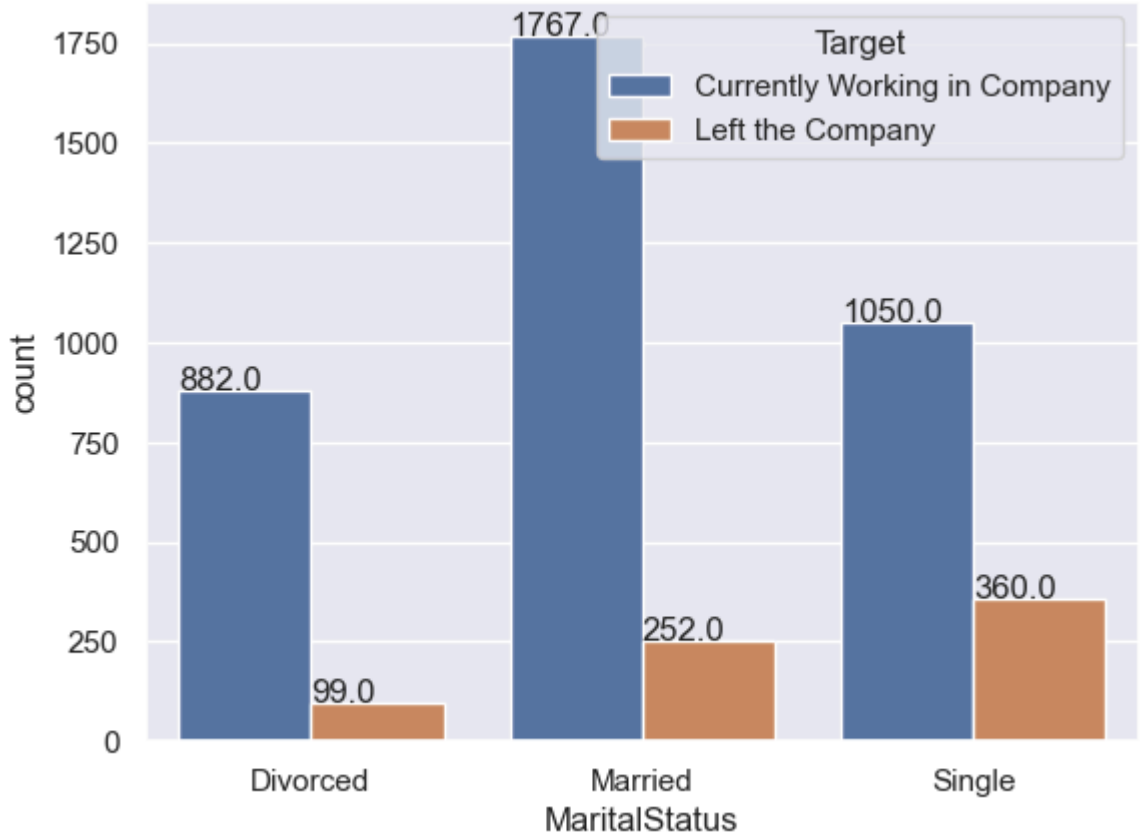
In [168…
```
ay = sns.countplot(y="EducationField", hue="Target", data=df1)
for p in ax.patches:
    ay.annotate('{}'.format(p.get_height()), (p.get_y(), p.get_width()+1))
```



In [169…
```
# Most of the Employees who are likely to quit are having Human Resources as Education Field .
```

In [170…
```
# MaritalStatus vs Target(Attrition)
```

In [171…
```
ax = sns.countplot(x="MaritalStatus", hue="Target", data=df1)
for p in ax.patches:
    ax.annotate('{}'.format(p.get_height()), (p.get_x(), p.get_height()+1))
```



In [172…
```
# 1. 25.53% of total single employees are likely to quit.
# 2. 12.48%  of total married employees are likely to quit.
# 3. 10.09% of total divorced employees  are likely to quit.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: