Abhishek Gupta

Intern :- Bharat Intern

Object :- In this project, you will need to evaluate each factor and its relationship with attrition, for example, the distance from home to office, the job role impact on attrition, etc

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
```

In [4]:

```
df = pd.read_csv('HR-Employee-Attrition.csv')
df
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Environn
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	
5	32	No	Travel_Frequently	1005	Research & Development	2	2	Life Sciences	1	8	
...	...	...	...	...	Research &	...	...	...	...	...	

In [169]:

```
df.head()
```

Out[169]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [9]:

```
df.isna().sum()
```

Out[9]:

```
Age                         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
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        0
dtype: int64
```

In [11]:

```
df.describe()
```

Out[11]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000

In [12]:

```
df.describe(include=object)
```

Out[12]:

	Attrition	BusinessTravel	Department	EducationField	Gender	JobRole	MaritalStatus	Over18	OverTime
count	1470	1470	1470	1470	1470	1470	1470	1470	1470
unique	2	3	3	6	2	9	3	1	2
top	No	Travel_Rarely	Research & Development	Life Sciences	Male	Sales Executive	Married	Y	No
freq	1233	1043	961	606	882	326	673	1470	1054

In [70]:

```
df = df.drop(columns=['Over18','EmployeeCount','EmployeeNumber','StandardHours'])
```

In [72]:

```
df.shape
```

Out[72]:

```
(1470, 31)
```

In [40]:

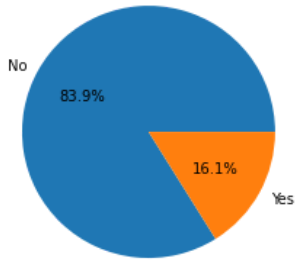
```
round(df['Attrition'].value_counts()/df.shape[0] *100,2)
```

Out[40]:

```
No     83.88
Yes    16.12
Name: Attrition, dtype: float64
```

In [50]:

```
x = round(df['Attrition'].value_counts()/df.shape[0] *100,2).values
y = round(df['Attrition'].value_counts()/df.shape[0] *100,2).index
plt.pie(x,labels = y, autopct='%1.1f%%')
plt.show()
```



In [27]:

```
df['Age'].value_counts().index.sort_values()
```

Out[27]:

```
Int64Index([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
            35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
            52, 53, 54, 55, 56, 57, 58, 59, 60],
           dtype='int64')
```

In [29]:

```
data = df[['Age','Attrition']]
data
```

Out[29]:

	Age	Attrition
0	41	Yes
1	49	No
2	37	Yes
3	33	No
4	27	No
5	32	No
6	59	No
7	30	No
8	38	No
9	36	No

In [166]:

```
data['Age_details'] = ['18-29' if 18<i<30  else '30-45' if 30<=i<45  else '45-60' if 45<=i<60  else 'Above 60' for i in df['Age
data
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_10904\1824373510.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['Age_details'] = ['18-29' if 18<i<30  else '30-45' if 30<=i<45  else '45-60' if 45<=i<60  else 'Above 60' for i in df['Age']]
```

Out[166]:

	Age	Attrition	Age_details
0	41	Yes	30-45
1	49	No	45-60
2	37	Yes	30-45
3	33	No	30-45

In [167]:

```
data.groupby(['Age_details','Attrition'])['Attrition'].count()
```

Out[167]:

```
Age_details  Attrition
18-29        No           231
             Yes           87
30-45        No           720
             Yes          110
45-60        No           273
             Yes           36
Above 60     No             9
             Yes            4
Name: Attrition, dtype: int64
```

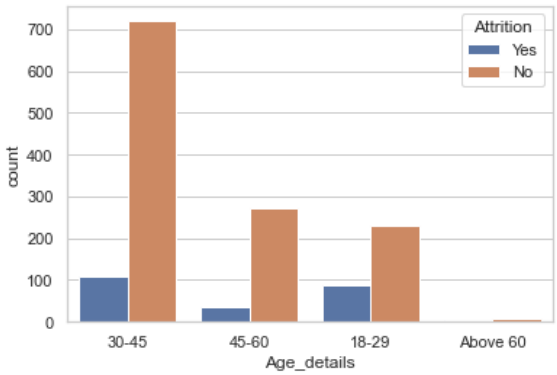
In [168]:

```
sns.countplot(data['Age_details'],hue=data['Attrition'])
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[168]:

```
<AxesSubplot:xlabel='Age_details', ylabel='count'>
```



In [65]:

```
df.groupby(['BusinessTravel','Attrition'])['Attrition'].count()
```

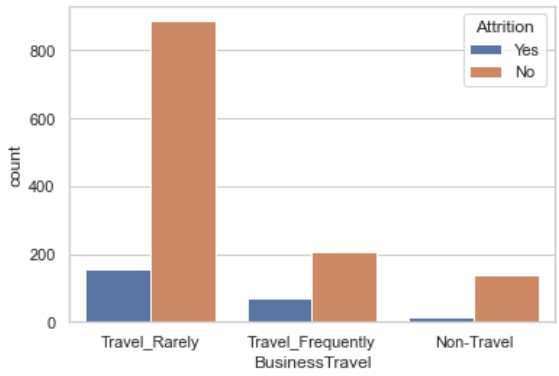
Out[65]:

```
BusinessTravel     Attrition
Non-Travel         No           138
                   Yes           12
Travel_Frequently  No           208
                   Yes           69
Travel_Rarely      No           887
                   Yes          156
Name: Attrition, dtype: int64
```

In [66]:

```
sns.set(style='whitegrid')
sns.countplot(df['BusinessTravel'],hue=df['Attrition'])
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [67]:

```
df.groupby(['Attrition']).mean()
```

Out[67]:

kOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	YearsInCurrentRole	YearsSinceLastPromotion	Yea
0.845093	11.862936	2.832928	2.781022	7.369019	4.484185	2.234388	
0.527426	8.244726	2.624473	2.658228	5.130802	2.902954	1.945148	

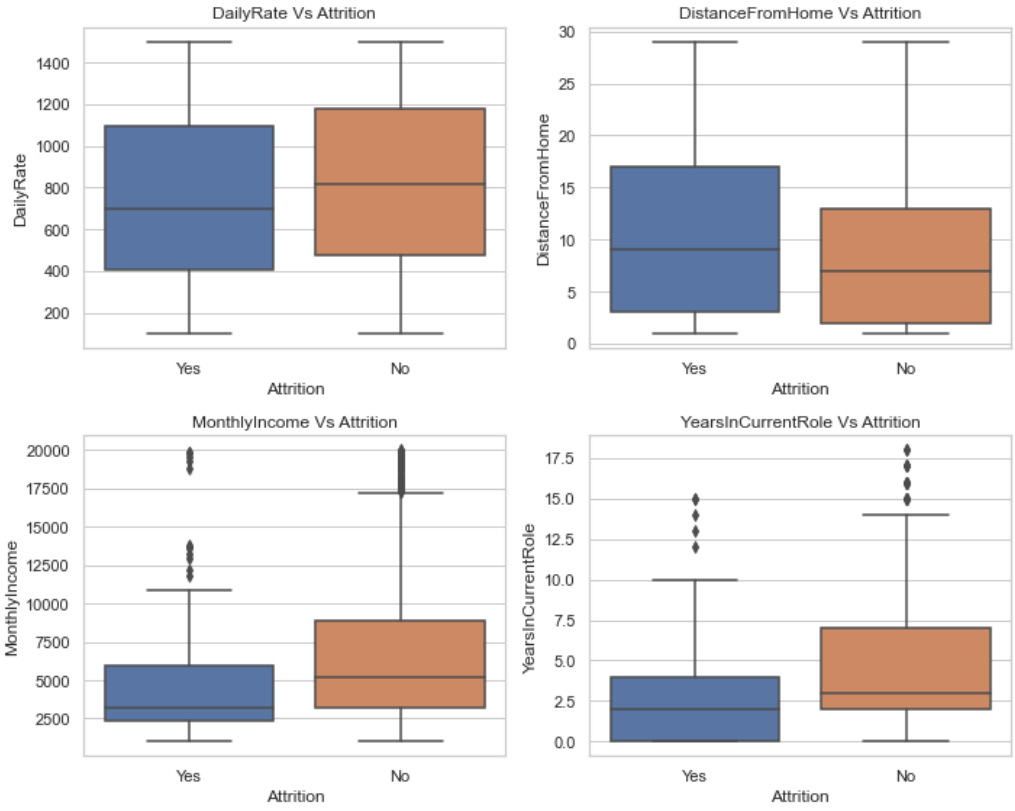
In [170]:

```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.title('DailyRate Vs Attrition')
sns.boxplot(data=df,x='Attrition',y='DailyRate')

plt.subplot(2,2,2)
plt.title('DistanceFromHome Vs Attrition')
sns.boxplot(data=df,x='Attrition',y='DistanceFromHome')


plt.subplot(2,2,3)
plt.title('MonthlyIncome Vs Attrition')
sns.boxplot(data=df,x='Attrition',y='MonthlyIncome')

plt.subplot(2,2,4)
plt.title('YearsInCurrentRole Vs Attrition')
sns.boxplot(data=df,x='Attrition',y='YearsInCurrentRole')
plt.tight_layout()
plt.show()
```



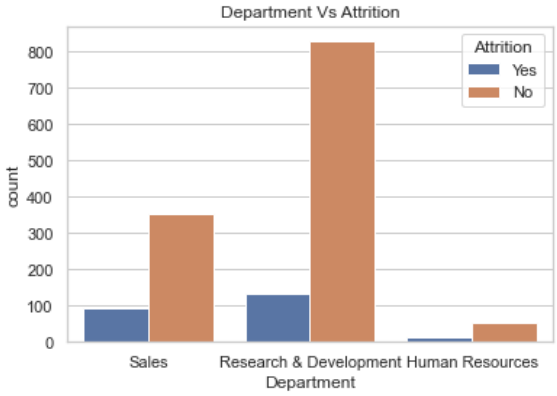
In [83]:

```
df['JobSatisfaction_details'] = ['Low' if i == 1 else 'Medium' if i == 2 else 'High' if i == 3 else 'Very High'
                                 for i in df['JobSatisfaction']]
```

In [77]:

```
plt.title('Department Vs Attrition')
sns.countplot(df['Department'],hue=df['Attrition'])
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [109]:

```
plt.title('Departmentwise jobSatisfaction Vs Attrition')
sns.boxplot(data=df,x='Department',y='JobSatisfaction',hue='Attrition')
plt.show()
```

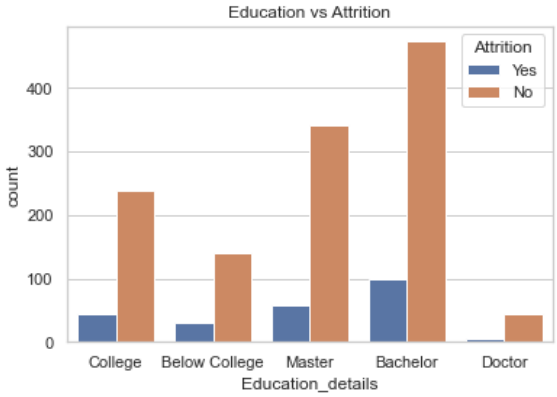
In [99]:

```
df['Education_details'] = ['Below College' if i == 1 else 'College' if i == 2 else 'Bachelor' if i == 3 else 'Master'
                           if i==4 else 'Doctor' for i in df['Education']]
```

In [104]:

```
plt.title('Education vs Attrition')
sns.countplot(df['Education_details'],hue=df['Attrition'])
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [117]:

```
df.groupby(['EducationField','Attrition'])['Attrition'].count()
```

Out[117]:

```
EducationField    Attrition
Human Resources   No            20
                  Yes            7
Life Sciences     No           517
                  Yes           89
Marketing         No           124
                  Yes           35
Medical           No           401
                  Yes           63
Other             No            71
                  Yes           11
Technical Degree  No           100
                  Yes           32
Name: Attrition, dtype: int64
```

In [125]:

```
pd.crosstab(index=df['EducationField'],columns=df['Attrition'],margins=True)
```

Out[125]:

Attrition	No	Yes	All
EducationField			
Human Resources	20	7	27
Life Sciences	517	89	606
Marketing	124	35	159
Medical	401	63	464
Other	71	11	82
Technical Degree	100	32	132
All	1233	237	1470

In [137]:

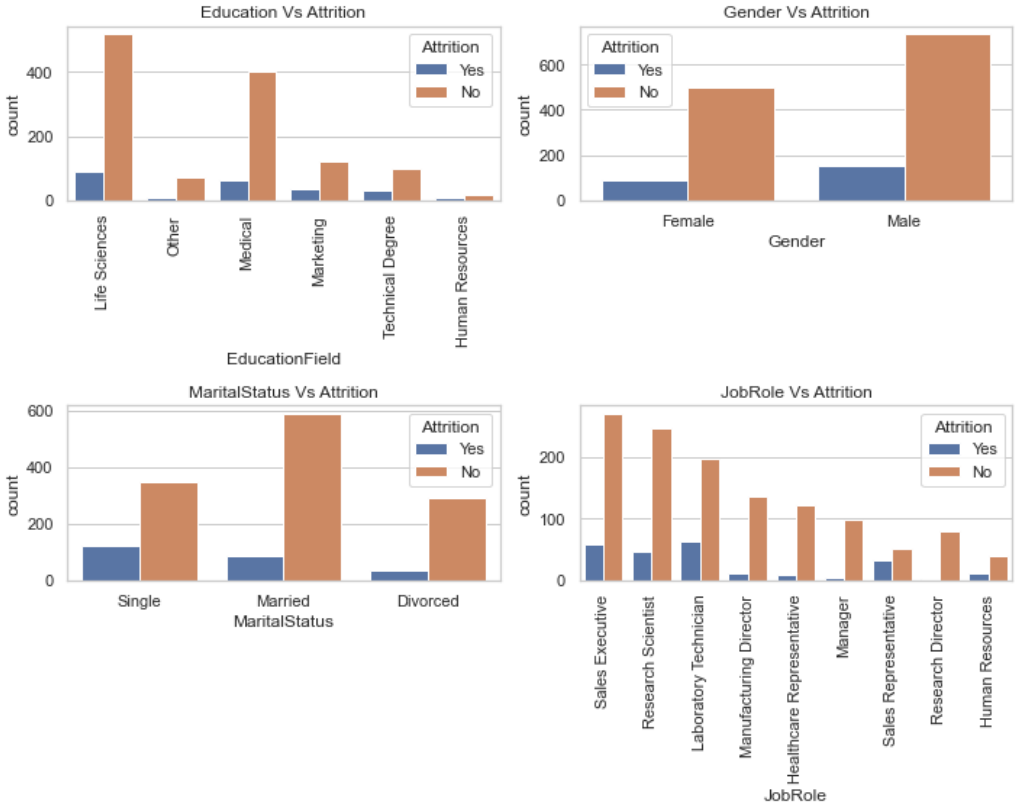
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.title('Education Vs Attrition')
sns.countplot(data = df, x='EducationField',hue='Attrition')
plt.xticks(rotation=90)

plt.subplot(2,2,2)
plt.title('Gender Vs Attrition')
sns.countplot(data=df,x='Gender',hue='Attrition')

plt.subplot(2,2,3)
plt.title('MaritalStatus Vs Attrition')
sns.countplot(data=df,x='MaritalStatus',hue='Attrition')

plt.subplot(2,2,4)
plt.title('JobRole Vs Attrition')
sns.countplot(data=df,x='JobRole',hue='Attrition')
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()
```



In [153]:

```
df['EnvironmentSatisfaction_details'] = ['Low' if i == 1 else 'Medium' if i == 2 else 'High' if i == 3 else 'Very High' for i i
```

In [155]:

```
df['JobInvolvement_details'] = ['Low' if i == 1 else 'Medium' if i == 2 else 'High' if i == 3 else 'Very High'
                                for i in df['JobInvolvement']]
df['PerformanceRating_details'] = ['Low' if i == 1 else 'Good' if i == 2 else 'Excellent' if i == 3 else 'Outstanding'
                                   for i in df['PerformanceRating']]
df['RelationshipSatisfaction_details'] = ['Low' if i == 1 else 'Medium' if i == 2 else 'High' if i == 3 else 'Very High'
                                          for i in df['RelationshipSatisfaction']]
df['WorkLifeBalance_details'] = ['Bad' if i == 1 else 'Good' if i == 2 else 'Better' if i == 3 else 'Best'
                                 for i in df['WorkLifeBalance']]
```

In [158]:

```
df['PerformanceRating_details'].value_counts()
```

Out[158]:

```
Excellent      1244
Outstanding     226
Name: PerformanceRating_details, dtype: int64
```

In [159]:

```
df['PerformanceRating'].value_counts()
```

Out[159]:

```
3    1244
4     226
Name: PerformanceRating, dtype: int64
```

In [163]:

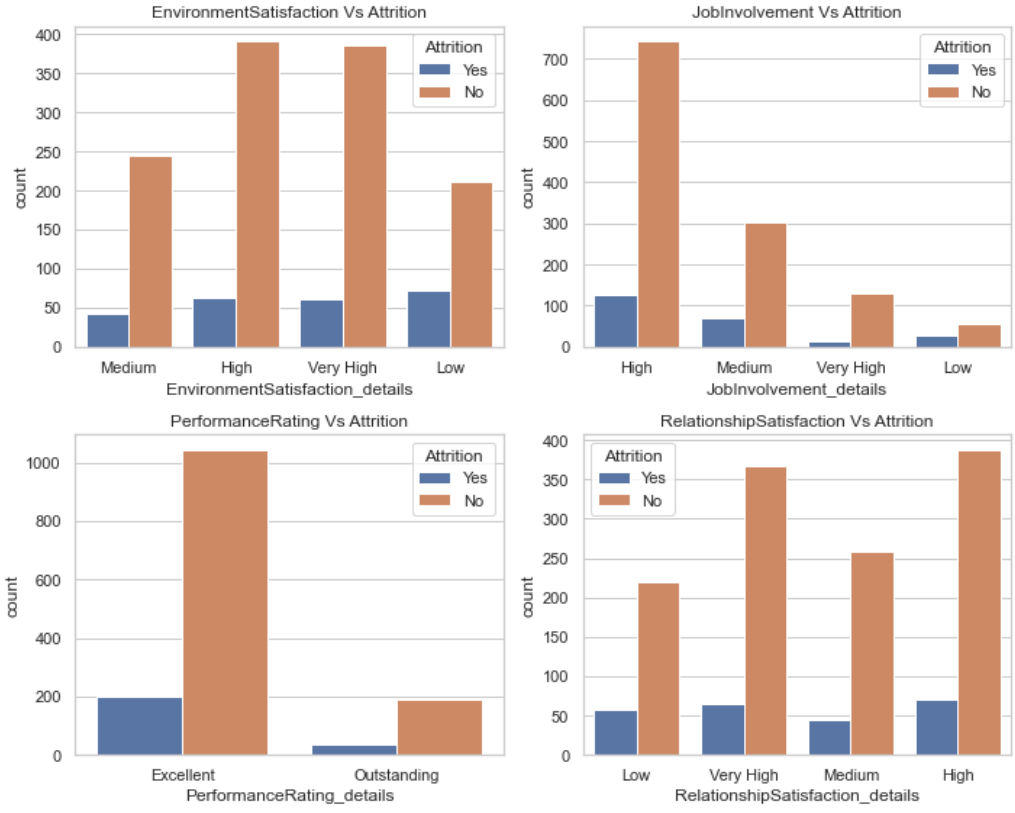
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.title('EnvironmentSatisfaction Vs Attrition')
sns.countplot(data=df,x='EnvironmentSatisfaction_details',hue='Attrition')

plt.subplot(2,2,2)
plt.title('JobInvolvement Vs Attrition')
sns.countplot(data=df,x='JobInvolvement_details',hue='Attrition')

plt.subplot(2,2,3)
plt.title('PerformanceRating Vs Attrition')
sns.countplot(data=df,x='PerformanceRating_details',hue='Attrition')

plt.subplot(2,2,4)
plt.title('RelationshipSatisfaction Vs Attrition')
sns.countplot(data=df,x='RelationshipSatisfaction_details',hue='Attrition')

plt.tight_layout()
plt.show()
```



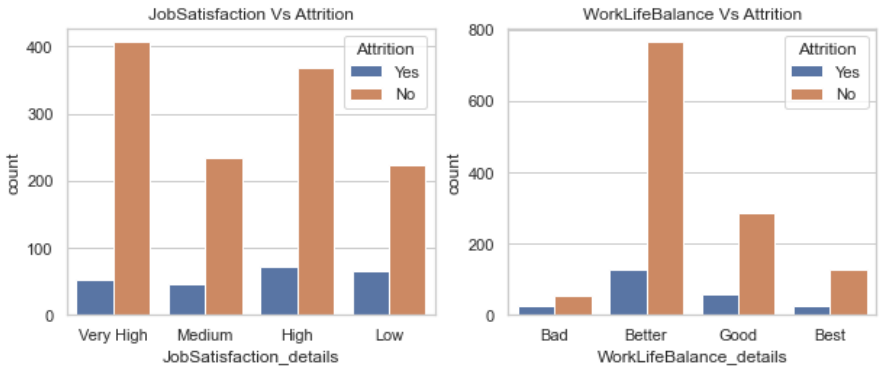
In [165]:

```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.title('JobSatisfaction Vs Attrition')
sns.countplot(df['JobSatisfaction_details'],hue=df['Attrition'])

plt.subplot(2,2,2)
plt.title('WorkLifeBalance Vs Attrition')
sns.countplot(df['WorkLifeBalance_details'],hue=df['Attrition'])


plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [ ]:

```
Insights:-
1. Dataset Overview:
Our dataset comprises 1470 records and 35 columns, predominantly featuring integer and object data types. Notably, we are pleas

2. Attrition Rate:
Our analysis reveals an attrition rate of 16.12% among our employees. This information serves as a crucial baseline for underst

3. Identified High-Risk Groups:
Our data-driven visualizations demonstrate that certain employee demographics exhibit a higher tendency towards attrition:

a). Employees with lower daily rates, longer commutes, and relatively lower monthly incomes.
b). Those holding lower-ranking positions within the company.
c). Employees aged between 30-45, 18-29, and 45-60.
d). Frequent travelers and those who travel rarely.
e). Employees who belong to departments such as R&D, Sales, and HR.
f). Individuals with education levels of Bachelor's, Master's, and College degrees.
g). Employees in the Life Science and Medical fields, with Marketing and Technical fields showing equal vulnerability to
    attrition.
h). Male employees are more likely to be affected by attrition compared to females.
i). Marital status, including singles and married employees, as well as those in sales representation roles, show
    increased attrition risk.
j). Job roles such as Lab Technician, Sales Executive, and Researcher are notably prone to attrition.
h). Performance ratings ranging from "Excellent" to "Outstanding" exhibit higher attrition rates.
i). Employees with higher satisfaction ratings for Relationship, Employment, and Job satisfaction are more prone to attrition.
j). Work-life balance ratings of "Better" and "Good" are indicative of higher attrition rates, while "Bad" and "Best" show
    equal susceptibility.

Recommendation:-
These insights underscore the importance of addressing specific areas to mitigate employee attrition.
By focusing on improving work-life balance, optimizing compensation and benefits, enhancing job satisfaction,
and creating a more inclusive and supportive work environment, we can effectively reduce attrition rates
and foster a more engaged and committed workforce.
```