```
import numpy as np # for array
import pandas as pd #for dataframe
import matplotlib.pyplot as plt #for plots and graphs
import seaborn as sns #data visualization library ..again for plot
from sklearn.model_selection import train_test_split
from sklearn import metrics
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
manager_survey = pd.read_csv('/content/manager_survey_data.csv')
manager_survey
```

| | EmployeeID | JobInvolvement | PerformanceRating |
|---|---|---|---|
| **0** | 1 | 3 | 3 |
| **1** | 2 | 2 | 4 |
| **2** | 3 | 3 | 3 |
| **3** | 4 | 2 | 3 |
| **4** | 5 | 3 | 3 |
| **...** | ... | ... | ... |
| **4405** | 4406 | 3 | 3 |
| **4406** | 4407 | 2 | 3 |
| **4407** | 4408 | 3 | 4 |
| **4408** | 4409 | 2 | 3 |
| **4409** | 4410 | 4 | 3 |

4410 rows × 3 columns

```
employee_survey = pd.read_csv('/content/employee_survey_data.csv')
employee_survey
```

| | EmployeeID | EnvironmentSatisfaction | JobSatisfaction | WorkLifeBalance |
|---|---|---|---|---|
| **0** | 1 | 3.0 | 4.0 | 2.0 |
| **1** | 2 | 3.0 | 2.0 | 4.0 |
| **2** | 3 | 2.0 | 2.0 | 1.0 |
| **3** | 4 | 4.0 | 4.0 | 3.0 |
| **4** | 5 | 4.0 | 1.0 | 3.0 |

```
general_data = pd.read_csv('/content/general_data.csv')
general_data
```

| | Age | Attrition | BusinessTravel | Department | DistanceFromHome | Education | Edu |
|---|---|---|---|---|---|---|---|
| **0** | 51 | No | Travel_Rarely | Sales | 6 | 2 | |
| **1** | 31 | Yes | Travel_Frequently | Research & Development | 10 | 1 | |
| **2** | 32 | No | Travel_Frequently | Research & Development | 17 | 4 | |
| **3** | 38 | No | Non-Travel | Research & Development | 2 | 5 | |
| **4** | 32 | No | Travel_Rarely | Research & Development | 10 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | |
| **4405** | 42 | No | Travel_Rarely | Research & Development | 5 | 4 | |
| **4406** | 29 | No | Travel_Rarely | Research & Development | 2 | 4 | |
| **4407** | 25 | No | Travel_Rarely | Research & Development | 25 | 2 | |
| **4408** | 42 | No | Travel_Rarely | Sales | 18 | 2 | |
| **4409** | 40 | No | Travel_Rarely | Research & Development | 28 | 3 | |

4410 rows × 24 columns

```
general_data = general_data.join([manager_survey.drop('EmployeeID', axis=1), employee_surv
general_data.drop('EmployeeID', axis=1, inplace=True)
general_data
```

| | Age | Attrition | BusinessTravel | Department | DistanceFromHome | Education | Edu |
|---|---|---|---|---|---|---|---|
| **0** | 51 | No | Travel_Rarely | Sales | 6 | 2 | |
| **1** | 31 | Yes | Travel_Frequently | Research & Development | 10 | 1 | |
| **2** | 32 | No | Travel_Frequently | Research & Development | 17 | 4 | |
| **3** | 38 | No | Non-Travel | Research & Development | 2 | 5 | |
| **4** | 32 | No | Travel_Rarely | Research & Development | 10 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | |
| **4405** | 42 | No | Travel_Rarely | Research & Development | 5 | 4 | |
| **4406** | 29 | No | Travel_Rarely | Research & Development | 2 | 4 | |
| **4407** | 25 | No | Travel_Rarely | Research & Development | 25 | 2 | |
| **4408** | 42 | No | Travel_Rarely | Sales | 18 | 2 | |
| **4409** | 40 | No | Travel_Rarely | Research & Development | 28 | 3 | |

4410 rows × 28 columns

```
general_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 28 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      4410 non-null   int64  
 1   Attrition                4410 non-null   object 
 2   BusinessTravel           4410 non-null   object 
 3   Department               4410 non-null   object 
 4   DistanceFromHome         4410 non-null   int64  
 5   Education                4410 non-null   int64  
 6   EducationField           4410 non-null   object 
 7   EmployeeCount            4410 non-null   int64  
 8   Gender                   4410 non-null   object 
 9   JobLevel                 4410 non-null   int64  
 10  JobRole                  4410 non-null   object 
 11  MaritalStatus            4410 non-null   object 
 12  MonthlyIncome            4410 non-null   int64  
 13  NumCompaniesWorked       4391 non-null   float64
 14  Over18                   4410 non-null   object 
 15  PercentSalaryHike        4410 non-null   int64  
 16  StandardHours            4410 non-null   int64  
 17  StockOptionLevel         4410 non-null   int64  
 18  TotalWorkingYears        4401 non-null   float64
 19  TrainingTimesLastYear    4410 non-null   int64  
 20  YearsAtCompany           4410 non-null   int64  
```

```
     21  YearsSinceLastPromotion  4410 non-null   int64  
     22  YearsWithCurrManager     4410 non-null   int64  
     23  JobInvolvement           4410 non-null   int64  
     24  PerformanceRating        4410 non-null   int64  
     25  EnvironmentSatisfaction  4385 non-null   float64
     26  JobSatisfaction          4390 non-null   float64
     27  WorkLifeBalance          4372 non-null   float64
    dtypes: float64(5), int64(15), object(8)
    memory usage: 964.8+ KB
```

Few columns have missing data. The number of missing data in those columns are few, but since the number of observations in the dataset are few, those rows with missing data will not be removed. Instead I will be fillin those missing data with the mean values in the columns they're missing in.

```
general_data['NumCompaniesWorked'].fillna(general_data['NumCompaniesWorked'].mean(), inpla
general_data['TotalWorkingYears'].fillna(general_data['TotalWorkingYears'].mean(), inplace
general_data['EnvironmentSatisfaction'].fillna(general_data['EnvironmentSatisfaction'].mea
general_data['JobSatisfaction'].fillna(general_data['JobSatisfaction'].mean(), inplace=Tru
general_data['WorkLifeBalance'].fillna(general_data['WorkLifeBalance'].mean(), inplace=Tru
general_data.isnull().sum()
```

```
    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
    JobInvolvement             0
    PerformanceRating          0
    EnvironmentSatisfaction    0
    JobSatisfaction            0
    WorkLifeBalance            0
    dtype: int64
```

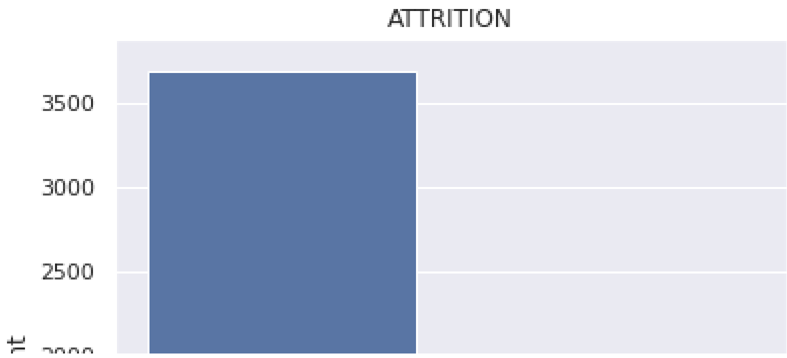
```
general_data.info()
```

```
    <class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 28 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      4410 non-null   int64  
 1   Attrition                4410 non-null   object 
 2   BusinessTravel           4410 non-null   object 
 3   Department               4410 non-null   object 
 4   DistanceFromHome         4410 non-null   int64  
 5   Education                4410 non-null   int64  
 6   EducationField           4410 non-null   object 
 7   EmployeeCount            4410 non-null   int64  
 8   Gender                   4410 non-null   object 
 9   JobLevel                 4410 non-null   int64  
 10  JobRole                  4410 non-null   object 
 11  MaritalStatus            4410 non-null   object 
 12  MonthlyIncome            4410 non-null   int64  
 13  NumCompaniesWorked       4410 non-null   float64
 14  Over18                   4410 non-null   object 
 15  PercentSalaryHike        4410 non-null   int64  
 16  StandardHours            4410 non-null   int64  
 17  StockOptionLevel         4410 non-null   int64  
 18  TotalWorkingYears        4410 non-null   float64
 19  TrainingTimesLastYear    4410 non-null   int64  
 20  YearsAtCompany           4410 non-null   int64  
 21  YearsSinceLastPromotion  4410 non-null   int64  
 22  YearsWithCurrManager     4410 non-null   int64  
 23  JobInvolvement           4410 non-null   int64  
 24  PerformanceRating        4410 non-null   int64  
 25  EnvironmentSatisfaction  4410 non-null   float64
 26  JobSatisfaction          4410 non-null   float64
 27  WorkLifeBalance          4410 non-null   float64
dtypes: float64(5), int64(15), object(8)
memory usage: 964.8+ KB
```

# EDA

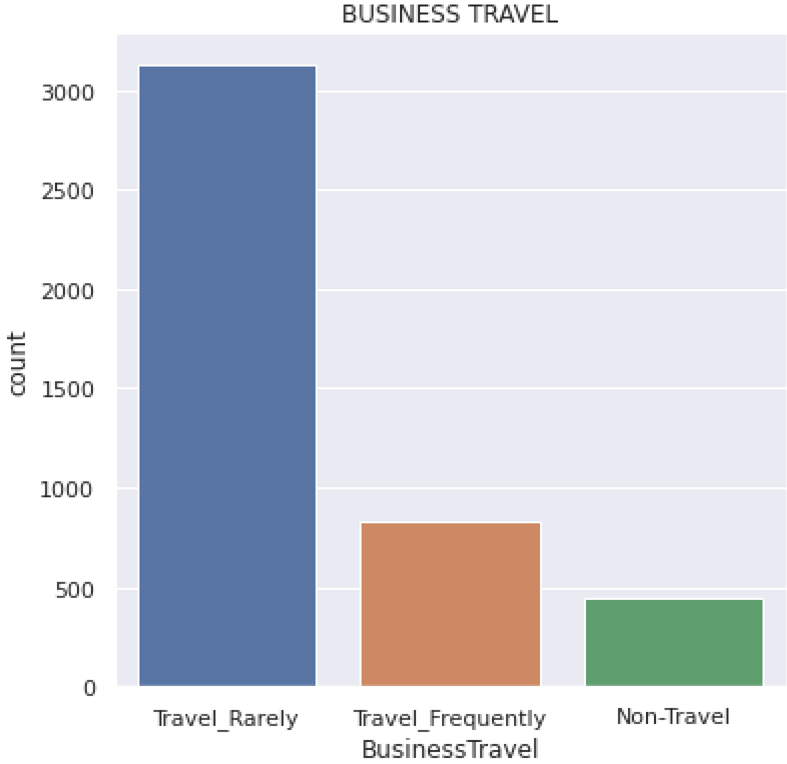
```
sns.set() #FOR BETTER THEMED PLOTS.
plt.figure(figsize=(6,6))
sns.countplot(general_data['Attrition'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGHOUT T
plt.title('ATTRITION')
plt.show()
```



```
# checking unique values in categorical columns
general_data['Attrition'].value_counts()
```

```
No     3699
Yes     711
Name: Attrition, dtype: int64
```

```
plt.figure(figsize=(6,6))
sns.countplot(general_data['BusinessTravel'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGH
plt.title('BUSINESS TRAVEL')
plt.show()
```

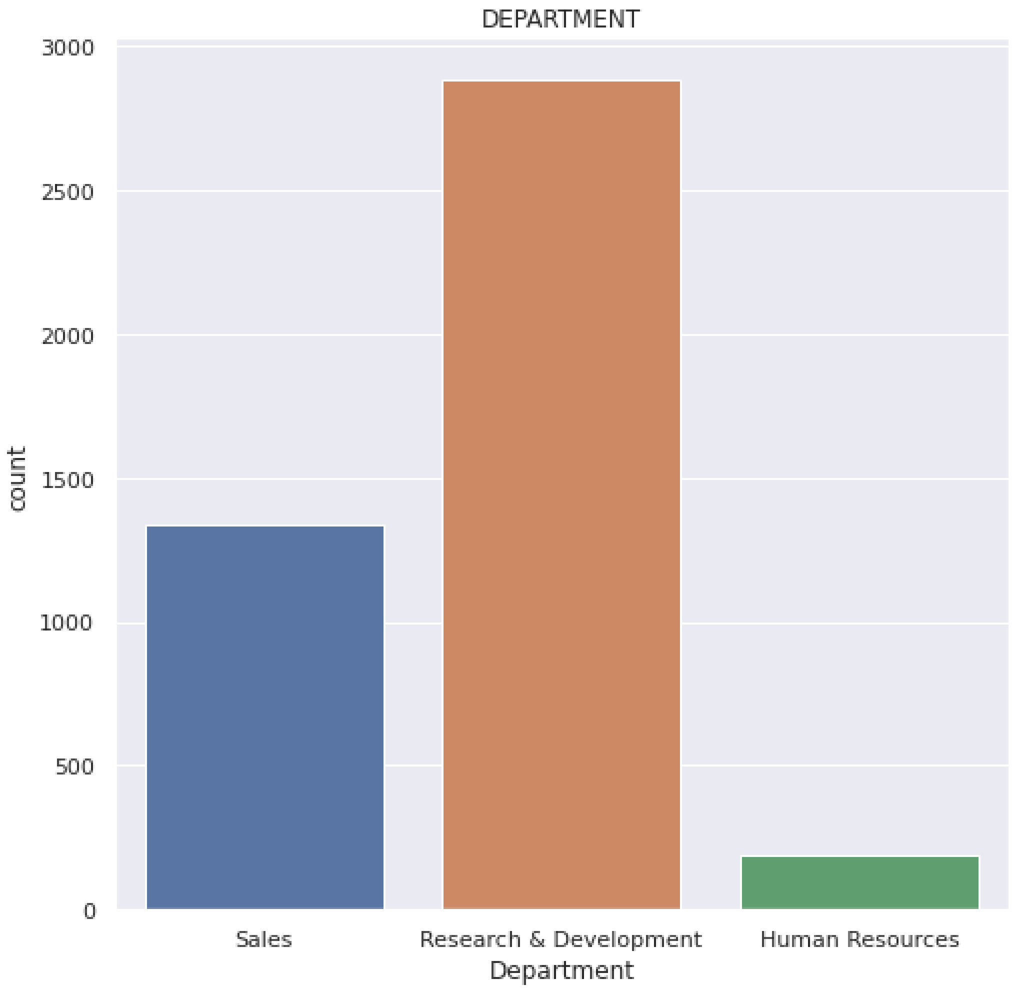


```
general_data['BusinessTravel'].value_counts()
```

```
Travel_Rarely        3129
Travel_Frequently     831
Non-Travel            450
Name: BusinessTravel, dtype: int64
```

```
plt.figure(figsize=(8,8))
sns.countplot(general_data['Department'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGHOUT
```

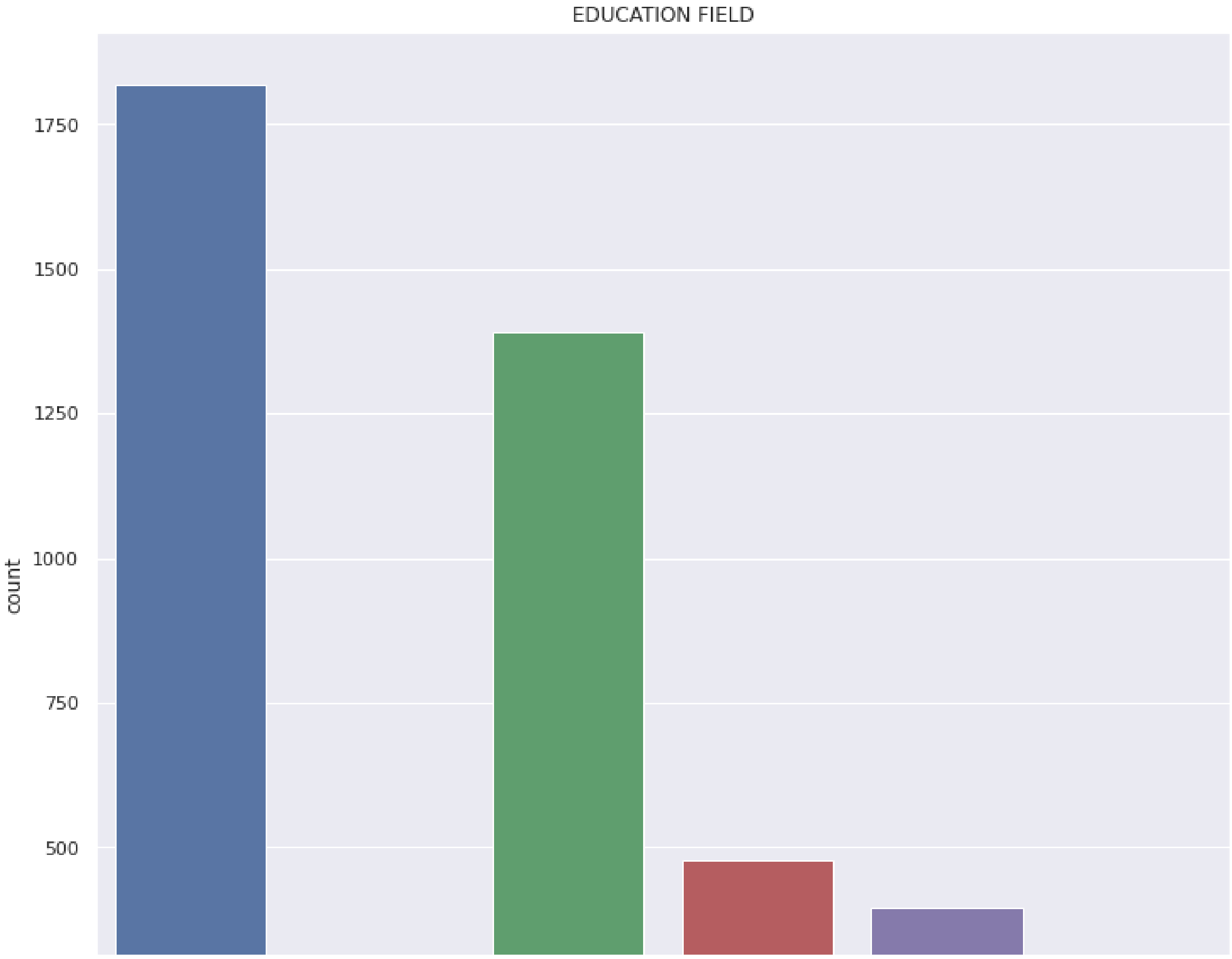
```
plt.title('DEPARTMENT')
plt.show()
```



```
general_data['Department'].value_counts()
```

```
Research & Development    2883
Sales                     1338
Human Resources            189
Name: Department, dtype: int64
```

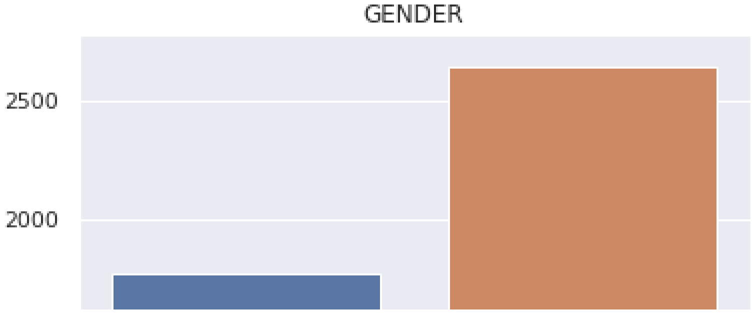
```
plt.figure(figsize=(12,12))
sns.countplot(general_data['EducationField'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGH
plt.title('EDUCATION FIELD')
plt.show()
```



```
general_data['EducationField'].value_counts()

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

```
plt.figure(figsize=(6,6))
sns.countplot(general_data['Gender'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGHOUT THE
plt.title('GENDER')
plt.show()
```



```
general_data['Gender'].value_counts()
```

```
     Male      2646
     Female    1764
     Name: Gender, dtype: int64
```

```
plt.figure(figsize=(20,20))
sns.countplot(general_data['JobRole'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGHOUT THE
plt.title('JOB ROLE')
plt.show()
```

```
general_data['JobRole'].value_counts()
```

```
     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: JobRole, dtype: int64
```

```
general_data['MaritalStatus'].value_counts()
```

```
     Married     2019
     Single      1410
     Divorced     981
     Name: MaritalStatus, dtype: int64
```

```
plt.figure(figsize=(6,6))
sns.countplot(general_data['Over18'])#TELLS HOW THE VALUES ARE DISTRIBUTED THROUGHOUT THE
plt.title('OVER 18 AGE')
plt.show()
```

OVER 18 AGE

```
general_data['Over18'].value_counts()
```

```
Y    4410
Name: Over18, dtype: int64
```

```
# using labelencoding for columns with only two categories
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
general_data['Attrition'] = le.fit_transform(general_data['Attrition'])
general_data['Gender'] = le.fit_transform(general_data['Gender'])
general_data['Over18'] = le.fit_transform(general_data['Over18'])
general_data.info()
```

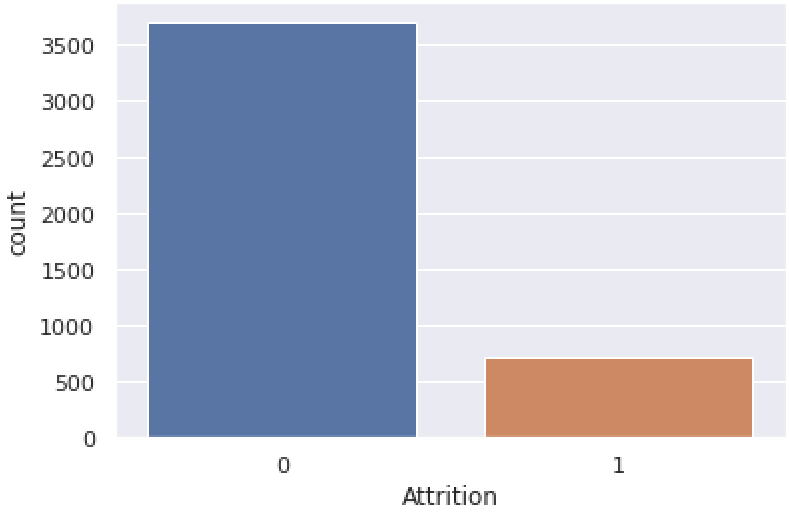
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 28 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      4410 non-null   int64  
 1   Attrition                4410 non-null   int64  
 2   BusinessTravel           4410 non-null   object 
 3   Department               4410 non-null   object 
 4   DistanceFromHome         4410 non-null   int64  
 5   Education                4410 non-null   int64  
 6   EducationField           4410 non-null   object 
 7   EmployeeCount            4410 non-null   int64  
 8   Gender                   4410 non-null   int64  
 9   JobLevel                 4410 non-null   int64  
 10  JobRole                  4410 non-null   object 
 11  MaritalStatus            4410 non-null   object 
 12  MonthlyIncome            4410 non-null   int64  
 13  NumCompaniesWorked       4410 non-null   float64
 14  Over18                   4410 non-null   int64  
 15  PercentSalaryHike        4410 non-null   int64  
 16  StandardHours            4410 non-null   int64  
 17  StockOptionLevel         4410 non-null   int64  
 18  TotalWorkingYears        4410 non-null   float64
 19  TrainingTimesLastYear    4410 non-null   int64  
 20  YearsAtCompany           4410 non-null   int64  
 21  YearsSinceLastPromotion  4410 non-null   int64  
 22  YearsWithCurrManager     4410 non-null   int64  
 23  JobInvolvement           4410 non-null   int64  
 24  PerformanceRating        4410 non-null   int64  
 25  EnvironmentSatisfaction  4410 non-null   float64
 26  JobSatisfaction          4410 non-null   float64
 27  WorkLifeBalance          4410 non-null   float64
dtypes: float64(5), int64(18), object(5)
memory usage: 964.8+ KB
```

```
# using dummies for columns with more than two categories
general_data = pd.get_dummies(general_data, columns=['BusinessTravel', 'Department', 'Educ
                                                    'JobRole', 'MaritalStatus'])
general_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 47 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Age                                4410 non-null   int64  
 1   Attrition                          4410 non-null   int64  
 2   DistanceFromHome                   4410 non-null   int64  
 3   Education                          4410 non-null   int64  
 4   EmployeeCount                      4410 non-null   int64  
 5   Gender                             4410 non-null   int64  
 6   JobLevel                           4410 non-null   int64  
 7   MonthlyIncome                      4410 non-null   int64  
 8   NumCompaniesWorked                 4410 non-null   float64
 9   Over18                             4410 non-null   int64  
 10  PercentSalaryHike                  4410 non-null   int64  
 11  StandardHours                      4410 non-null   int64  
 12  StockOptionLevel                   4410 non-null   int64  
 13  TotalWorkingYears                  4410 non-null   float64
 14  TrainingTimesLastYear              4410 non-null   int64  
 15  YearsAtCompany                     4410 non-null   int64  
 16  YearsSinceLastPromotion            4410 non-null   int64  
 17  YearsWithCurrManager               4410 non-null   int64  
 18  JobInvolvement                     4410 non-null   int64  
 19  PerformanceRating                  4410 non-null   int64  
 20  EnvironmentSatisfaction            4410 non-null   float64
 21  JobSatisfaction                    4410 non-null   float64
 22  WorkLifeBalance                    4410 non-null   float64
 23  BusinessTravel_Non-Travel          4410 non-null   uint8  
 24  BusinessTravel_Travel_Frequently   4410 non-null   uint8  
 25  BusinessTravel_Travel_Rarely       4410 non-null   uint8  
 26  Department_Human Resources         4410 non-null   uint8  
 27  Department_Research & Development  4410 non-null   uint8  
 28  Department_Sales                   4410 non-null   uint8  
 29  EducationField_Human Resources     4410 non-null   uint8  
 30  EducationField_Life Sciences       4410 non-null   uint8  
 31  EducationField_Marketing           4410 non-null   uint8  
 32  EducationField_Medical             4410 non-null   uint8  
 33  EducationField_Other               4410 non-null   uint8  
 34  EducationField_Technical Degree    4410 non-null   uint8  
 35  JobRole_Healthcare Representative  4410 non-null   uint8  
 36  JobRole_Human Resources            4410 non-null   uint8  
 37  JobRole_Laboratory Technician      4410 non-null   uint8  
 38  JobRole_Manager                    4410 non-null   uint8  
 39  JobRole_Manufacturing Director     4410 non-null   uint8  
 40  JobRole_Research Director          4410 non-null   uint8  
 41  JobRole_Research Scientist         4410 non-null   uint8  
 42  JobRole_Sales Executive            4410 non-null   uint8  
 43  JobRole_Sales Representative       4410 non-null   uint8  
 44  MaritalStatus_Divorced             4410 non-null   uint8  
 45  MaritalStatus_Married              4410 non-null   uint8  
 46  MaritalStatus_Single               4410 non-null   uint8  
dtypes: float64(5), int64(18), uint8(24)
memory usage: 895.9 KB
```

```
import seaborn as sns
sns.countplot(x='Attrition', data=general_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90d69407d0>
```



```
len(general_data[general_data['Attrition']==1])/len(general_data)
```

```
0.16122448979591836
```

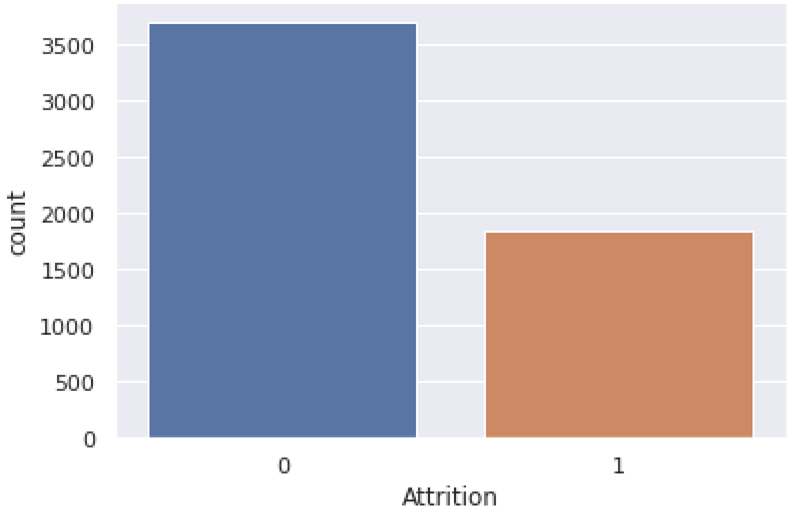
Only 16% of employees in this dataset left the company so there is a large class imbalance

```
from imblearn.over_sampling import SMOTE

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

# Resample data
X, y = SMOTE(sampling_strategy=0.5, random_state=0).fit_resample(X, y)
sns.countplot(x=y)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90d5b2b990>
```



```
from sklearn.model_selection import train_test_split
X_train, X_test,y_train, y_test = train_test_split(X, y, test_size=20, random_state=0)
X_train
```

| | Age | DistanceFromHome | Education | EmployeeCount | Gender | JobLevel | MonthlyInc |
|---|---|---|---|---|---|---|---|
| **3475** | 28 | 1 | 1 | 1 | 1 | 2 | 634 |
| **3314** | 42 | 13 | 4 | 1 | 0 | 2 | 228 |
| **4363** | 30 | 17 | 4 | 1 | 1 | 2 | 643 |
| **5305** | 29 | 4 | 2 | 1 | 1 | 1 | 285 |
| **2986** | 39 | 5 | 4 | 1 | 0 | 2 | 210 |
| **...** | ... | ... | ... | ... | ... | ... | |
| **4931** | 30 | 2 | 3 | 1 | 1 | 2 | 400 |
| **3264** | 40 | 10 | 4 | 1 | 1 | 2 | 656 |
| **1653** | 42 | 2 | 4 | 1 | 0 | 1 | 297 |
| **2607** | 31 | 2 | 1 | 1 | 1 | 3 | 709 |
| **2732** | 48 | 2 | 4 | 1 | 0 | 1 | 459 |

5528 rows × 46 columns

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
acc = accuracy_score(pred, y_test)
acc
```

```
0.65
```

Colab paid products - Cancel contracts here

✓ 0s completed at 10:27 AM ● ×