TechnoHacks: Data science

Task3: Employee turnover prediction

Use a dataset of employee information and build a model that can predict which employees are most likely to leave the company

Author: Sujata Gaikwad

In [1]:
```
# import librires
import pandas as pd
import numpy as np
```

In [2]:
```
data=pd.read_csv(r"C:\Users\HP\Downloads\WA_Fn-UseC_-HR-Employee-Attrition.csv
data
```

Out[2]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	E
0	41	Yes	Travel_Rarely	1102	Sales	1	2	
1	49	No	Travel_Frequently	279	Research & Development	8	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	
4	27	No	Travel_Rarely	591	Research & Development	2	1	
...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

1470 rows × 35 columns

In [3]:
```
data.size
```

Out[3]:
```
51450
```

In [4]:
```
data.shape
```

Out[4]:
```
(1470, 35)
```

In [5]:
```
data.describe()
```

Out[5]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNu
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.00
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.86
std	9.135373	403.509100	8.106864	1.024165	0.0	602.02
min	18.000000	102.000000	1.000000	1.000000	1.0	1.00
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.25
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.50
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.75
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.00

8 rows × 26 columns

In [6]:
```
data.corr()
```

Out[6]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Er
Age	1.000000	0.010661	-0.001686	0.208034	NaN	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	
Education	0.208034	-0.016806	0.021042	1.000000	NaN	
EmployeeCount	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	
StandardHours	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	

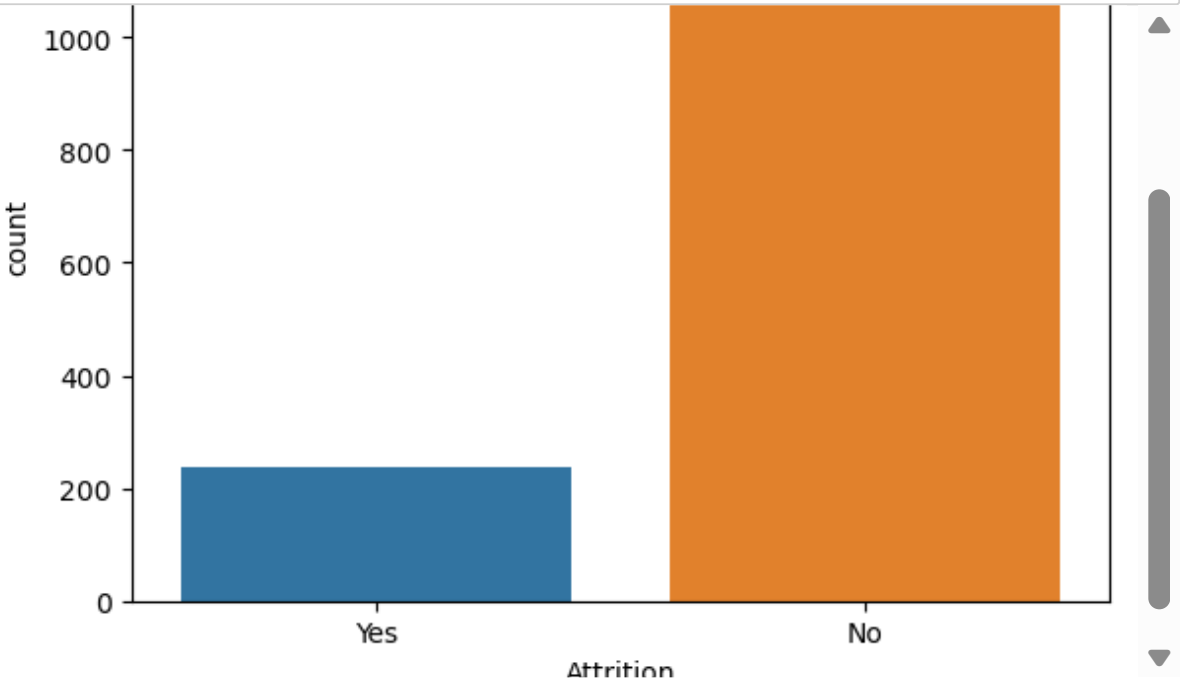
26 rows × 26 columns

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
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
```

Visuallization

In [8]:
```
import seaborn as sns
sns.countplot(data=data,x="Attrition")
```



In [9]:
```
data.drop(["EmployeeCount","EmployeeNumber","Over18","StandardHours"],axis="co
categorical_col=[]
for column in data.columns:
    if data[column].dtype==object and len(data[column].unique())<=50:
        categorical_col.append(column)
data["Attrition"]=data.Attrition.astype("category").cat.codes
```

In [10]:
```
categorical_col.remove("Attrition")
```

In [11]:
```
## Transform categorical data into dummies
data1=pd.get_dummies(data,columns=categorical_col)
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 52 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Age                                   1470 non-null   int64
 1   Attrition                             1470 non-null   int8
 2   DailyRate                             1470 non-null   int64
 3   DistanceFromHome                      1470 non-null   int64
 4   Education                             1470 non-null   int64
 5   EnvironmentSatisfaction               1470 non-null   int64
 6   HourlyRate                            1470 non-null   int64
 7   JobInvolvement                        1470 non-null   int64
 8   JobLevel                              1470 non-null   int64
 9   JobSatisfaction                       1470 non-null   int64
 10  MonthlyIncome                         1470 non-null   int64
 11  MonthlyRate                           1470 non-null   int64
 12  NumCompaniesWorked                    1470 non-null   int64
 13  PercentSalaryHike                     1470 non-null   int64
 14  PerformanceRating                     1470 non-null   int64
 15  RelationshipSatisfaction              1470 non-null   int64
 16  StockOptionLevel                      1470 non-null   int64
 17  TotalWorkingYears                     1470 non-null   int64
 18  TrainingTimesLastYear                 1470 non-null   int64
 19  WorkLifeBalance                       1470 non-null   int64
 20  YearsAtCompany                        1470 non-null   int64
 21  YearsInCurrentRole                    1470 non-null   int64
 22  YearsSinceLastPromotion               1470 non-null   int64
 23  YearsWithCurrManager                  1470 non-null   int64
 24  BusinessTravel_Non-Travel             1470 non-null   uint8
 25  BusinessTravel_Travel_Frequently      1470 non-null   uint8
 26  BusinessTravel_Travel_Rarely          1470 non-null   uint8
 27  Department_Human Resources            1470 non-null   uint8
 28  Department_Research & Development     1470 non-null   uint8
 29  Department_Sales                      1470 non-null   uint8
 30  EducationField_Human Resources        1470 non-null   uint8
 31  EducationField_Life Sciences          1470 non-null   uint8
 32  EducationField_Marketing              1470 non-null   uint8
 33  EducationField_Medical                1470 non-null   uint8
 34  EducationField_Other                  1470 non-null   uint8
 35  EducationField_Technical Degree       1470 non-null   uint8
 36  Gender_Female                         1470 non-null   uint8
 37  Gender_Male                           1470 non-null   uint8
 38  JobRole_Healthcare Representative     1470 non-null   uint8
 39  JobRole_Human Resources               1470 non-null   uint8
 40  JobRole_Laboratory Technician         1470 non-null   uint8
 41  JobRole_Manager                       1470 non-null   uint8
 42  JobRole_Manufacturing Director        1470 non-null   uint8
 43  JobRole_Research Director             1470 non-null   uint8
 44  JobRole_Research Scientist            1470 non-null   uint8
 45  JobRole_Sales Executive               1470 non-null   uint8
 46  JobRole_Sales Representative          1470 non-null   uint8
 47  MaritalStatus_Divorced                1470 non-null   uint8
 48  MaritalStatus_Married                 1470 non-null   uint8
 49  MaritalStatus_Single                  1470 non-null   uint8
 50  OverTime_No                           1470 non-null   uint8
 51  OverTime_Yes                          1470 non-null   uint8
```

```
dtypes: int64(23), int8(1), uint8(28)
memory usage: 305.9 KB
```

In [12]:
```
from sklearn.preprocessing import LabelEncoder
l1=LabelEncoder()
for column in categorical_col:
    data[column]=l1.fit_transform(data[column])
```

In [13]:
```
from sklearn.model_selection import train_test_split
x=data.drop("Attrition",axis=1)
y=data.Attrition
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
```

In [14]:
```
from sklearn.tree import DecisionTreeClassifier
d1=DecisionTreeClassifier(random_state=10)
d1.fit(x_train,y_train)
```

Out[14]:
```
DecisionTreeClassifier(random_state=10)
```

In [15]:
```
y_pred=d1.predict(x_test)
y_pred
```

Out[15]:
```
array([0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0], dtype=int8)
```

In [16]:
```
from sklearn.metrics import accuracy_score
ac=accuracy_score(y_pred,y_test)*100
print(ac)
```

```
79.59183673469387
```

Thank You

In [ ]:
```
```