Name Of The Intern : Sarthak Irappa Gadge

Title Of The Project : Employee Attrition Analysis

Technologies : Data Science

Domain : Human Resource

Importing the Dependencies

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import seaborn as sns
import matplotlib.pyplot as plt
```

Data Collection and Processing

```
# Loading the csv data to Pandas DataFrame
df = pd.read_csv('/content/Attrition data.csv')
```

```
# Print the first 5 rows of the dataset
df.head()
```

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education
0	1	51	No	Travel_Rarely	Sales	6	2
1	2	31	Yes	Travel_Frequently	Research & Development	10	1
2	3	32	No	Travel_Frequently	Research & Development	17	4
3	4	38	No	Non-Travel	Research & Development	2	5
4	5	32	No	Travel_Rarely	Research & Development	10	1

5 rows × 29 columns

```
#Number of rows and colums in the dataset
df.shape
```

```
(4410, 29)
```

```
#statistical measures about the data
df.describe()
```

```
              EmployeeID        Age  DistanceFromHome   Education  EmployeeCount     JobLevel  MonthlyIncom
```

```
# Checking the missing values
df.isnull().sum()
```

```
     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
```

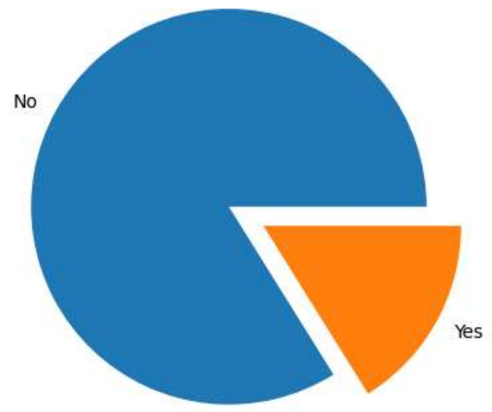
```
attrition_count = pd.DataFrame(df['Attrition'].value_counts())
```

```
attrition_count
```

	Attrition
No	3699
Yes	711

```
plt.pie(attrition_count['Attrition'] , labels = ['No' , 'Yes'] , explode = (0.2,0))
```

```
     ([<matplotlib.patches.Wedge at 0x7fa6d61a9490>,
       <matplotlib.patches.Wedge at 0x7fa6d61a96a0>],
      [Text(-1.136781068348268, 0.6306574368426737, 'No'),
       Text(0.961891673217765, -0.5336332157899548, 'Yes')])
```



Splitting

```
df.drop(['EmployeeID' , 'Age'] , axis = 1)
```

	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCou
0	No	Travel_Rarely	Sales	6	2	Life Sciences	
1	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	
2	No	Travel_Frequently	Research & Development	17	4	Other	
3	No	Non-Travel	Research & Development	2	5	Life Sciences	
4	No	Travel_Rarely	Research & Development	10	1	Medical	
...	...	...	...	...	...	...	
4405	No	Travel_Rarely	Research & Development	5	4	Medical	
4406	No	Travel_Rarely	Research & Development	2	4	Medical	
4407	No	Travel_Rarely	Research & Development	25	2	Life Sciences	
4408	No	Travel_Rarely	Sales	18	2	Medical	
4409	No	Travel_Rarely	Research & Development	28	3	Medical	

4410 rows × 27 columns

```
attrition_dummies = pd.get_dummies(df['Attrition'])
attrition_dummies.head()
```

	No	Yes
0	1	0
1	0	1
2	1	0
3	1	0
4	1	0

Concaneting The DataFrame & attrition_dummies

```
df = pd.concat([df, attrition_dummies] , axis = 1)
df.head
```

```
<bound method NDFrame.head of       EmployeeID  Age Attrition     BusinessTravel              Department  \
0              1   51        No      Travel_Rarely                   Sales
1              2   31       Yes  Travel_Frequently  Research & Development
2              3   32        No  Travel_Frequently  Research & Development
3              4   38        No         Non-Travel  Research & Development
4              5   32        No      Travel_Rarely  Research & Development
...          ...  ...       ...                ...                     ...
4405        4406   42        No      Travel_Rarely  Research & Development
4406        4407   29        No      Travel_Rarely  Research & Development
4407        4408   25        No      Travel_Rarely  Research & Development
4408        4409   42        No      Travel_Rarely                   Sales
4409        4410   40        No      Travel_Rarely  Research & Development

      DistanceFromHome  Education EducationField  EmployeeCount  Gender  ...  \
0                    6          2  Life Sciences              1  Female  ...
1                   10          1  Life Sciences              1  Female  ...
2                   17          4          Other              1    Male  ...
3                    2          5  Life Sciences              1    Male  ...
4                   10          1        Medical              1    Male  ...
...                ...        ...            ...            ...     ...  ...
4405                 5          4        Medical              1  Female  ...
4406                 2          4        Medical              1    Male  ...
```

```
4407              25          2  Life Sciences             1    Male  ...
4408              18          2        Medical             1    Male  ...
4409              28          3        Medical             1    Male  ...

        YearsAtCompany YearsSinceLastPromotion YearsWithCurrManager  \
0                    1                       0                    0
1                    5                       1                    4
2                    5                       0                    3
3                    8                       7                    5
4                    6                       0                    4
...                ...                     ...                  ...
4405                 3                       0                    2
4406                 3                       0                    2
4407                 4                       1                    2
4408                 9                       7                    8
4409                21                       3                    9

        EnvironmentSatisfaction  JobSatisfaction WorkLifeBalance  \
0                           3.0              4.0             2.0
1                           3.0              2.0             4.0
2                           2.0              2.0             1.0
3                           4.0              4.0             3.0
4                           4.0              1.0             3.0
...                         ...              ...             ...
4405                        4.0              1.0             3.0
4406                        4.0              4.0             3.0
4407                        1.0              3.0             3.0
4408                        4.0              1.0             3.0
4409                        1.0              3.0             NaN

        JobInvolvement  PerformanceRating  No  Yes
0                    3                  3   1    0
1                    2                  4   0    1
2                    3                  3   1    0
3                    2                  3   1    0
```

Splitting

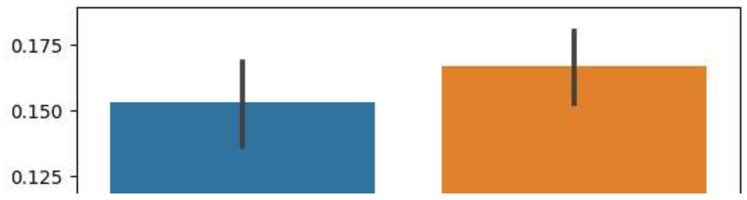
```
df = df.drop(['Attrition' , 'No'] , axis = 1)
df.head()
```

	EmployeeID	Age	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeC
0	1	51	Travel_Rarely	Sales	6	2	Life Sciences	
1	2	31	Travel_Frequently	Research & Development	10	1	Life Sciences	
2	3	32	Travel_Frequently	Research & Development	17	4	Other	
3	4	38	Non-Travel	Research & Development	2	5	Life Sciences	
4	5	32	Travel_Rarely	Research & Development	10	1	Medical	

5 rows × 29 columns

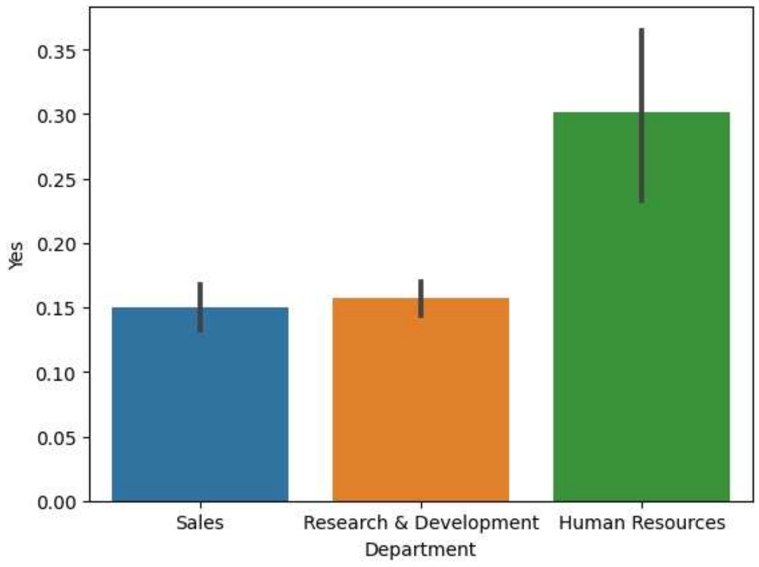
```
sns.barplot(x = 'Gender' , y = 'Yes', data = df)
```

```
<Axes: xlabel='Gender', ylabel='Yes'>
```



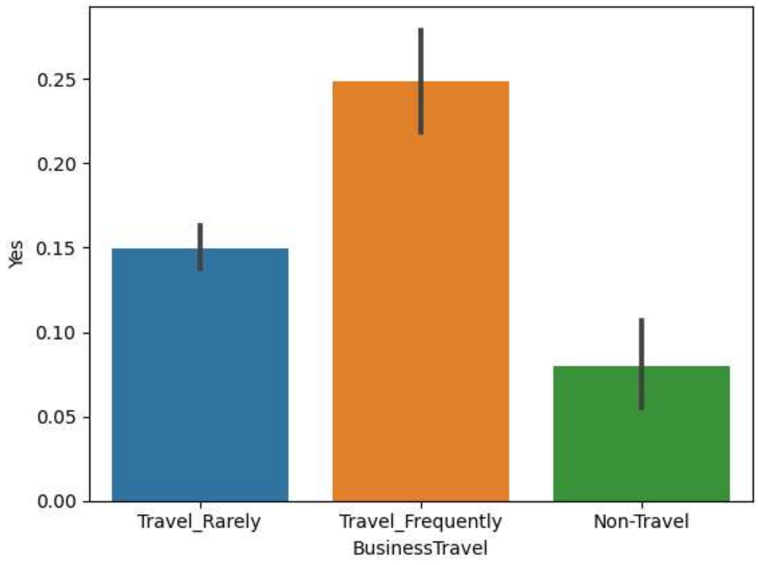
```
sns.barplot(x = 'Department', y = 'Yes', data = df)
```

```
<Axes: xlabel='Department', ylabel='Yes'>
```



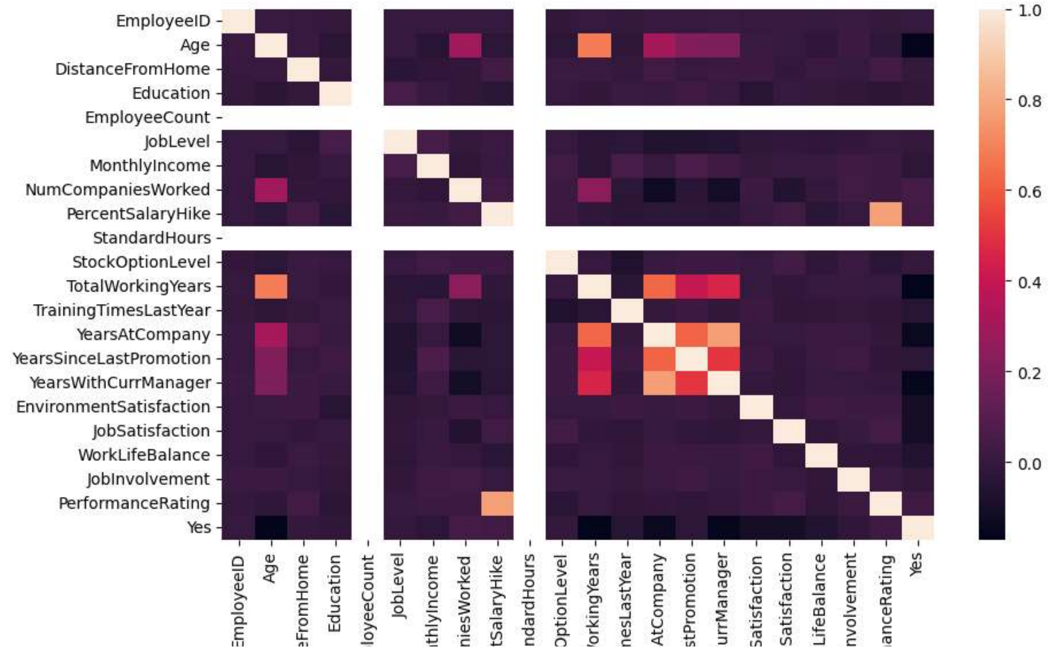
```
sns.barplot(x = 'BusinessTravel', y = 'Yes', data = df)
```

```
<Axes: xlabel='BusinessTravel', ylabel='Yes'>
```



```
plt.figure(figsize = (10,6))
sns.heatmap(df.corr())
```

```
<Axes: >
```



```
df = df.drop(['Age' , 'JobLevel'], axis = 1)
```

```
from sklearn.preprocessing import LabelEncoder
for column in df.columns:
    if df[column].dtype==np.number:
        continue
    else:
        df[column]=LabelEncoder().fit_transform(df[column])
```

```
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
```

```
  if df[column].dtype==np.number:
<ipython-input-23-b2b38dba6098>:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecated. The current re
  if df[column].dtype==np.number:
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
```

```
x  = df.drop(['Yes'], axis = 1)
y = df['Yes']
```

Splitting The Data into Training Data & Test Data

```
x_train, x_test , y_train, y_test = train_test_split(x,y, test_size = 0.3, random_state = 0)
```

```
x_train.head()
```

	EmployeeID	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCoun
1087	1087	2	2	1	0	5	
1621	1621	1	2	9	2	2	
1804	1804	2	1	5	3	4	
744	744	1	1	5	3	3	
686	686	0	1	0	2	3	

5 rows × 26 columns