S R Rupa Krishna

21BIT0275

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

```
data = pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
data.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educatic
0	41	Yes	Travel_Rarely	1102	Sales	1	
1	49	No	Travel_Frequently	279	Research & Development	8	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	
4	27	No	Travel_Rarely	591	Research & Development	2	

5 rows × 35 columns

```
1 data.shape
```

```
(1470, 35)
```

```
1 data.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	J
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	

8 rows × 26 columns

```
1 data.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
```

```
1 data.isnull().sum()
```

```
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
```

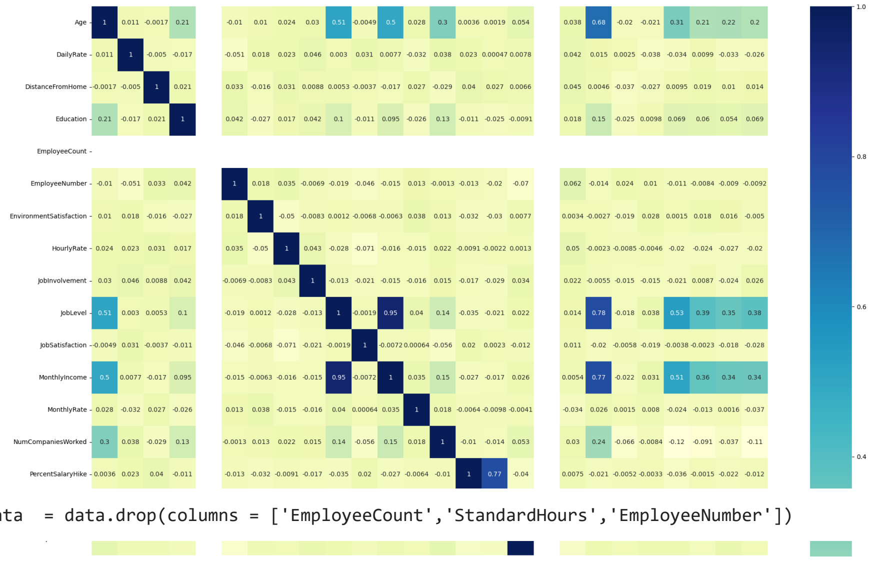
```
1 data.corr()
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_8564\2627137660.py:1: FutureWarning: The c
  data.corr()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCo
Age	1.000000	0.010661	-0.001686	0.208034	[illegible]
DailyRate	0.010661	1.000000	-0.004985	-0.016806	[illegible]
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	[illegible]
Education	0.208034	-0.016806	0.021042	1.000000	[illegible]
EmployeeCount	NaN	NaN	NaN	NaN	[illegible]
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	[illegible]
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	[illegible]
HourlyRate	0.024287	0.023381	0.031131	0.016775	[illegible]
JobInvolvement	0.029820	0.046135	0.008783	0.042438	[illegible]
JobLevel	0.509604	0.002966	0.005303	0.101589	[illegible]
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	[illegible]
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	[illegible]
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

```
plt.figure(figsize =(24,24))
sns.heatmap(data.corr(),annot = True,cmap = "YlGnBu")
plt.show()
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_8564\2009265951.py:2: FutureWarning: The c
  sns.heatmap(data.corr(),annot = True,cmap = "YlGnBu")
```




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

```
data.head()
```

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

5 rows × 32 columns

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

	Attrition
No	1233
Yes	237

```
plt.pie(attrition_count['Attrition'] , labels = ['No' , 'Yes'] , explode = (0.15,0))
```

```
([<matplotlib.patches.Wedge at 0x270888b5c10>,
  <matplotlib.patches.Wedge at 0x27088980990>],
 [Text(-1.0930587195656423, 0.606401381579494, 'No'),
  Text(0.961891673217765, -0.5336332157899547, 'Yes')])
```

```
attrition_dummies = pd.get_dummies(data['Attrition'])
data = pd.concat([data, attrition_dummies] , axis = 1)
```

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

```
data.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	Hourly
0	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female	
1	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male	
2	37	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male	
3	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female	
4	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male	

5 rows × 32 columns

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
from sklearn.preprocessing import LabelEncoder
l = LabelEncoder()
for columns in data.columns:
    if data[columns].dtype == [np.number,np.float64,np.int64]:
        continue
    else:
        data[columns] = l.fit_transform(data[columns])
```

Spliting dependent and independent variables

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

```
y.head()
```

```
0    1
1    0
2    1
3    0
4    0
Name: Yes, dtype: int64
```

```
x.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Education
0	23	2	624	2	0	1	
1	31	1	113	1	7	0	
2	19	2	805	1	1	1	
3	15	1	820	1	2	3	
4	9	2	312	1	1	0	

5 rows × 31 columns

Feature Scaling

```
from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()
x_Scaled = pd.DataFrame(ms.fit_transform(x),columns=x.columns)
x_Scaled.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educa
0	0.547619	1.0	0.705085	1.0	0.000000	0.25	
1	0.738095	0.5	0.127684	0.5	0.250000	0.00	
2	0.452381	1.0	0.909605	0.5	0.035714	0.25	
3	0.357143	0.5	0.926554	0.5	0.071429	0.75	
4	0.214286	1.0	0.352542	0.5	0.035714	0.00	

5 rows × 31 columns

Train and Test data Split

```
from sklearn.model_selection import train_test_split
```

```
xtrain,xtest,ytrain,ytest = train_test_split(x_Scaled,y,test_size = 0.2,random_state = 0)
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
```

```
lr.fit(xtrain,ytrain)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
lrpred = lr.predict(xtest)
```

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve
```

```
accuracy_score(ytest,lrpred)
```

```
0.8843537414965986
```

```
confusion_matrix(ytest,lrpred)
```

```
array([[242,   3],
       [ 31,  18]], dtype=int64)
```

```
print(classification_report(ytest,lrpred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.99      0.93       245
           1       0.86      0.37      0.51        49

    accuracy                           0.88       294
   macro avg       0.87      0.68      0.72       294
weighted avg       0.88      0.88      0.86       294
```

Decision Tree Classification

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
```

```
dtc.fit(xtrain,ytrain)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
dtcpred = dtc.predict(xtest)
```

```
1 accuracy_score(ytest,dtcpred)
```

```
0.7482993197278912
```

```
1 print(classification_report(ytest,dtcpred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.83      0.85       245
           1       0.28      0.33      0.30        49

    accuracy                           0.75       294
   macro avg       0.57      0.58      0.57       294
weighted avg       0.76      0.75      0.76       294
```

```
1 from sklearn import tree
2 plt.figure(figsize=(35,25))
3 tree.plot_tree(dtc,filled=True)
```

```
[Text(0.31914702868852457, 0.9722222222222222, 'x[24] <= 0.038\ngini =
0.269\nsamples = 1176\nvalue = [988, 188]'),
 Text(0.07213114754098361, 0.9166666666666666, 'x[14] <= 0.75\ngini = 0.5\nsamples
= 78\nvalue = [39, 39]'),
 Text(0.04262295081967213, 0.8611111111111112, 'x[4] <= 0.554\ngini =
0.426\nsamples = 39\nvalue = [27, 12]'),
 Text(0.02622950819672131, 0.8055555555555556, 'x[13] <= 0.167\ngini =
0.312\nsamples = 31\nvalue = [25, 6]'),
 Text(0.013114754098360656, 0.75, 'x[15] <= 0.046\ngini = 0.49\nsamples = 7\nvalue
= [3, 4]'),
 Text(0.006557377049180328, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.019672131147540985, 0.6944444444444444, 'x[14] <= 0.25\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.013114754098360656, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.02622950819672131, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.03934426229508197, 0.75, 'x[17] <= 0.056\ngini = 0.153\nsamples =
24\nvalue = [22, 2]'),
 Text(0.03278688524590164, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.04590163934426229, 0.6944444444444444, 'x[7] <= 0.167\ngini =
0.083\nsamples = 23\nvalue = [22, 1]'),
 Text(0.03934426229508197, 0.6388888888888888, 'x[1] <= 0.75\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.03278688524590164, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.04590163934426229, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.05245901639344262, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0]'),
 Text(0.05901639344262295, 0.8055555555555556, 'x[20] <= 0.679\ngini =
0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.05245901639344262, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.06557377049180328, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.10163934426229508, 0.8611111111111112, 'x[9] <= 0.364\ngini =
0.426\nsamples = 39\nvalue = [12, 27]'),
 Text(0.08524590163934426, 0.8055555555555556, 'x[15] <= 0.231\ngini =
0.133\nsamples = 14\nvalue = [1, 13]'),
 Text(0.07868852459016394, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(0.09180327868852459, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.1180327868852459, 0.8055555555555556, 'x[19] <= 0.5\ngini = 0.493\nsamples
= 25\nvalue = [11, 14]'),
 Text(0.10491803278688525, 0.75, 'x[2] <= 0.108\ngini = 0.484\nsamples = 17\nvalue
= [10, 7]'),
 Text(0.09836065573770492, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.11147540983606558, 0.6944444444444444, 'x[22] <= 0.167\ngini =
0.408\nsamples = 14\nvalue = [10, 4]'),
 Text(0.09836065573770492, 0.6388888888888888, 'x[24] <= 0.013\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.09180327868852459, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.10491803278688525, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.12459016393442623, 0.6388888888888888, 'x[12] <= 0.875\ngini =
```