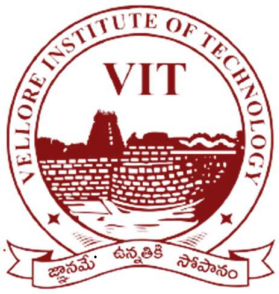


VIT-AP UNIVERSITY

SMARTBRIDGE
Let's Bridge the Gap

Smart Internz

VIT – Foundation – SmartBridge – Artificial Intelligence & Machine Learning in collaboration with Google (Applied Data Science)

Assignment-4

Name: Muni Aswanth Prasad A

Registration number: 21BCE8854

Tutor: Sri Tulasi (Smart Internz)

```
ASSIGNMENT 4 : MORINING SLOT
NAME : MUNI ASWANTH PRASAD A
REG.NO: 21BCE8854
```

In [ ]:
```
#Import the Libraries.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [135]:
```
#Importing the dataset.
df=pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

In [136]:
```
df.head()
```

Out[136]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipS
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 rows × 35 columns

In [137]:
```
df.shape
```

Out[137]:
```
(1470, 35)
```

In [138]:
```
df.Attrition.value_counts()
```

Out[138]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

In [139]:
```
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 [140]:
```
df.describe()
```

Out[140]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	14
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	

8 rows × 26 columns

In [141]:
```
#Checking for Null Values.
df.isnull().any()
```

Out[141]:
```
Age                         False
Attrition                   False
BusinessTravel              False
DailyRate                   False
Department                  False
DistanceFromHome            False
Education                   False
EducationField              False
EmployeeCount               False
EmployeeNumber              False
EnvironmentSatisfaction     False
Gender                      False
HourlyRate                  False
JobInvolvement              False
JobLevel                    False
JobRole                     False
JobSatisfaction             False
MaritalStatus               False
MonthlyIncome               False
MonthlyRate                 False
NumCompaniesWorked          False
Over18                      False
OverTime                    False
PercentSalaryHike           False
PerformanceRating           False
RelationshipSatisfaction    False
StandardHours               False
StockOptionLevel            False
TotalWorkingYears           False
TrainingTimesLastYear       False
WorkLifeBalance             False
YearsAtCompany              False
YearsInCurrentRole          False
YearsSinceLastPromotion     False
YearsWithCurrManager        False
dtype: bool
```

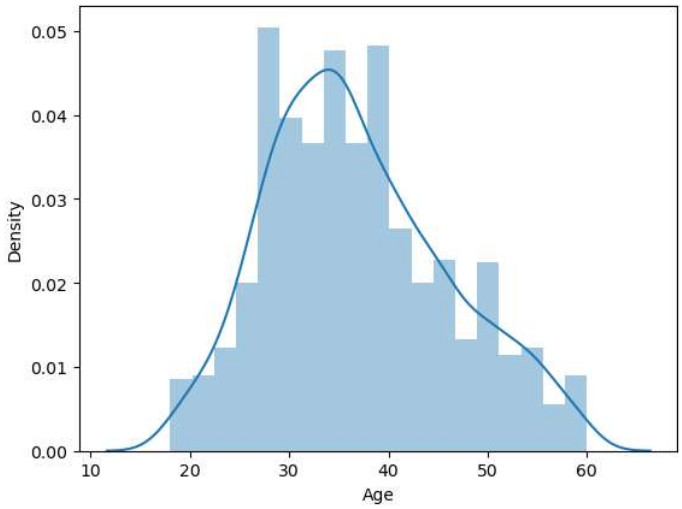
In [142]:
```
df.isnull().sum()
```

Out[142]:
```
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 [143]:
```
#Data Visualization.
sns.distplot(df["Age"])
```

```
C:\Users\ayyam\AppData\Local\Temp\ipykernel_17088\2400079689.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df["Age"])
```

Out[143]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



In [144]:
```
df.corr()
```

```
C:\Users\ayyam\AppData\Local\Temp\ipykernel_17088\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFram
e.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  df.corr()
```

Out[144]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvol
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145	0.010146	0.024287	0
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990	0.018355	0.023381	0
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916	-0.016075	0.031131	0
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070	-0.027128	0.016775	0
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000	0.017621	0.035179	-0
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621	1.000000	-0.049857	-0
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179	-0.049857	1.000000	0
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888	-0.008278	0.042861	1
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519	0.001212	-0.027853	-0
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247	-0.006784	-0.071335	-0
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829	-0.006259	-0.015794	-0
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648	0.037600	-0.015297	-0
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251	0.012594	0.022157	0
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944	-0.031701	-0.009062	-0
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359	-0.029548	-0.002172	-0
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861	0.007665	0.001330	0
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227	0.003432	0.050263	0
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365	-0.002693	-0.002334	-0
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603	-0.019359	-0.008548	-0
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	0.010309	0.027627	-0.004607	-0
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	-0.011240	0.001458	-0.019582	-0
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	-0.008416	0.018007	-0.024106	0
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	-0.009019	0.016194	-0.026716	-0
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.009197	-0.004999	-0.020123	0

26 rows × 26 columns

In [145]:
```
df.head()
```

Out[145]:

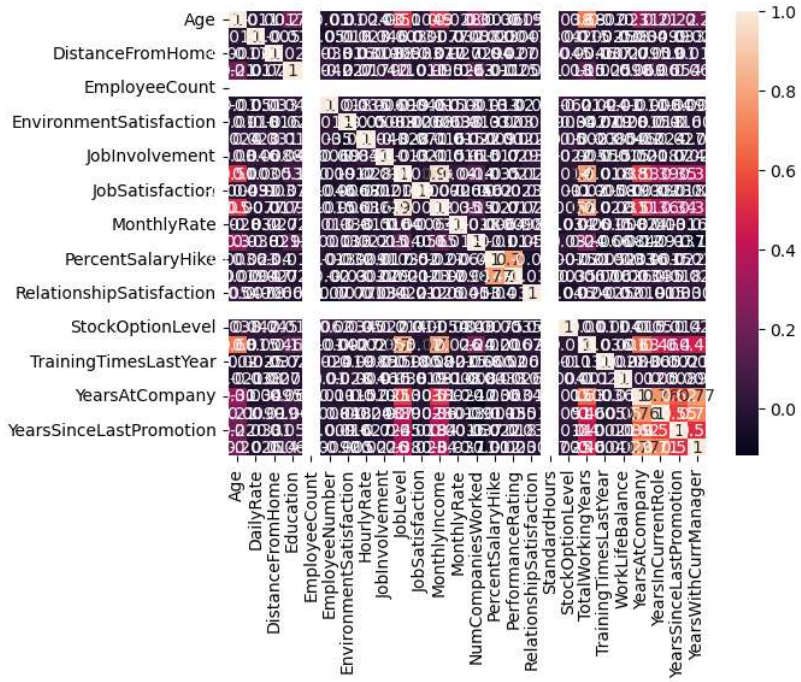
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipS
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	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 rows × 35 columns

In [146]:
```
sns.heatmap(df.corr(),annot=True)
```

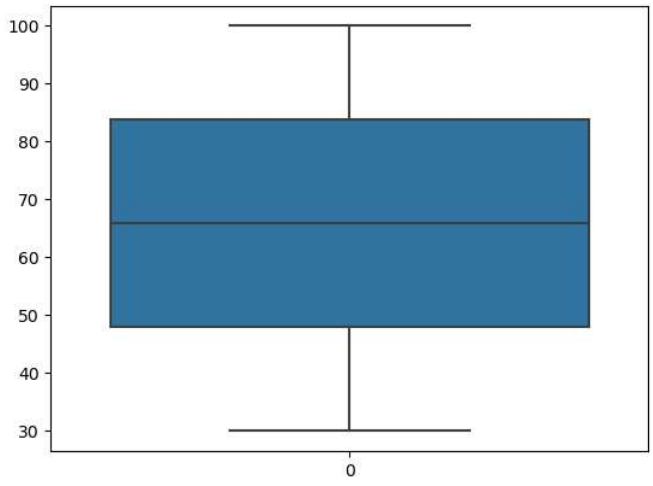
```
C:\Users\ayyam\AppData\Local\Temp\ipykernel_17088\4277794465.py:1: FutureWarning: The default value of numeric_only in DataFram
e.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

Out[146]: <Axes: >

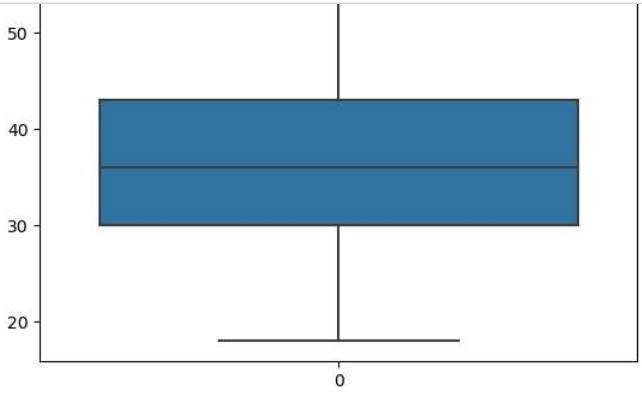


In [147]:
```
sns.boxplot(df.HourlyRate)
```

Out[147]: <Axes: >



In [148]:
```
sns.boxplot(df.Age)
```



In [149]:
```
df.head()
```

Out[149]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipS
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 rows × 35 columns

In [150]:
```
# Split the dataset into training and testing sets
X = df.drop('Attrition', axis=1)  # Features
y = df['Attrition']  # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [151]:
```
x_test
```

Out[151]:

	Attrition	BusinessTravel	DailyRate
442	0.0	0.0	0.385311
1091	0.0	1.0	0.338983
981	1.0	0.5	0.407910
785	0.0	1.0	0.995480
1332	1.0	0.5	0.249718
...	...	...	...
1439	0.0	1.0	0.323164
481	0.0	1.0	0.112994
124	1.0	1.0	0.108475
198	0.0	1.0	0.822599
1229	0.0	1.0	0.190960

294 rows × 3 columns

In [152]:
```
y_train
```

Out[152]:
```
1097     No
727      No
254      No
1175     No
1341     No
        ...
1130     No
1294     No
860     Yes
1459     No
1126     No
Name: Attrition, Length: 1176, dtype: object
```

Model Building using Logistic Regression and Decision Tree

In [153]:
```
# Train and evaluate a Logistic Regression model
lr_model = LogisticRegression()
lr_model.fit(x_train, y_train)
lr_predictions = lr_model.predict(x_test)
```

In [154]:
```
# Evaluate the performance of the Logistic Regression model
lr_accuracy = accuracy_score(y_test, lr_predictions)
lr_classification_report = classification_report(y_test, lr_predictions)
lr_confusion_matrix = confusion_matrix(y_test, lr_predictions)
```

```
C:\Users\ayyam\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ayyam\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ayyam\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [155]:
```
print("Logistic Regression Performance:")
print(f"Accuracy: {lr_accuracy:.2f}")
print("Classification Report:")
print(lr_classification_report)
print("Confusion Matrix:")
print(lr_confusion_matrix)
```

```
Logistic Regression Performance:
Accuracy: 0.87
Classification Report:
              precision    recall  f1-score   support

          No       0.87      1.00      0.93       255
         Yes       0.00      0.00      0.00        39

    accuracy                           0.87       294
   macro avg       0.43      0.50      0.46       294
weighted avg       0.75      0.87      0.81       294

Confusion Matrix:
[[255   0]
 [ 39   0]]
```

In [156]:
```
X_train
```

Out[156]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction
1097	24	Travel_Rarely	350	Research & Development	21	2	Technical Degree	1	1551	3
727	18	Non-Travel	287	Research & Development	5	2	Life Sciences	1	1012	2
254	29	Travel_Rarely	1247	Sales	20	2	Marketing	1	349	4
1175	39	Travel_Rarely	492	Research & Development	12	3	Medical	1	1654	4
1341	31	Travel_Rarely	311	Research & Development	20	3	Life Sciences	1	1881	2
...	...	...	...	...	...	...	...	...	...	...
1130	35	Travel_Rarely	750	Research & Development	28	3	Life Sciences	1	1596	2
1294	41	Travel_Rarely	447	Research & Development	5	3	Life Sciences	1	1814	2
860	22	Travel_Frequently	1256	Research & Development	3	4	Life Sciences	1	1203	3
1459	29	Travel_Rarely	1378	Research & Development	13	2	Other	1	2053	4
1126	50	Travel_Rarely	264	Sales	9	3	Marketing	1	1591	3

1176 rows × 34 columns

In [157]:
```
y_train
```

Out[157]:
```
1097     No
727      No
254      No
1175     No
1341     No
       ...
1130     No
1294     No
860     Yes
1459     No
1126     No
Name: Attrition, Length: 1176, dtype: object
```

In [ ]:
```
```

In [164]:
```
# Train and evaluate a Decision Tree model
dt_model = DecisionTreeClassifier()
dt_model.fit(x_train, y_train)
dt_predictions = dt_model.predict(x_test)
```

In [166]:
```
# Evaluate the performance of the Decision Tree model
dt_accuracy = accuracy_score(y_test, dt_predictions)
dt_classification_report = classification_report(y_test, dt_predictions)
dt_confusion_matrix = confusion_matrix(y_test, dt_predictions)
```

In [167]:
```
print("\nDecision Tree Performance:")
print(f"Accuracy: {dt_accuracy:.2f}")
print("Classification Report:")
print(dt_classification_report)
print("Confusion Matrix:")
print(dt_confusion_matrix)
```

```
Decision Tree Performance:
Accuracy: 0.78
Classification Report:
              precision    recall  f1-score   support

          No       0.88      0.86      0.87       255
         Yes       0.20      0.23      0.21        39

    accuracy                           0.78       294
   macro avg       0.54      0.54      0.54       294
weighted avg       0.79      0.78      0.78       294

Confusion Matrix:
[[219  36]
 [ 30   9]]
```

In [ ]: