- NAME: Prathamesh Mahesh Bhangale
- EMAIL: prathamesh.mahesh2021@vitstudent.ac.in
- Branch: Computer Science and Engineering
- Campus: VIT Vellore


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

```
import warnings
warnings.filterwarnings("ignore")
```

```
dataset = pd.read_csv("../Dataset/WA_Fn-UseC_-HR-Employee-Attrition.csv")
dataset.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

```
dataset.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
```

```
dataset.shape
```

```
(1470, 35)
```

```
dataset.describe
```

```
<bound method NDFrame.describe of       Age Attrition     BusinessTravel  DailyRate              Department  \
0      41       Yes      Travel_Rarely       1102                   Sales
1      49        No  Travel_Frequently        279  Research & Development
2      37       Yes      Travel_Rarely       1373  Research & Development
3      33        No  Travel_Frequently       1392  Research & Development
4      27        No      Travel_Rarely        591  Research & Development
...   ...       ...                 ...        ...                     ...
1465   36        No  Travel_Frequently        884  Research & Development
1466   39        No      Travel_Rarely        613  Research & Development
1467   27        No      Travel_Rarely        155  Research & Development
1468   49        No  Travel_Frequently       1023                   Sales
1469   34        No      Travel_Rarely        628  Research & Development

      DistanceFromHome  Education EducationField  EmployeeCount  \
0                    1          2  Life Sciences              1
1                    8          1  Life Sciences              1
2                    2          2          Other              1
3                    3          4  Life Sciences              1
4                    2          1        Medical              1
...                ...        ...            ...            ...
1465                23          2        Medical              1
1466                 6          1        Medical              1
1467                 4          3  Life Sciences              1
1468                 2          3        Medical              1
1469                 8          3        Medical              1

      EmployeeNumber  ...  RelationshipSatisfaction StandardHours  \
0                  1  ...                         1            80
1                  2  ...                         4            80
2                  4  ...                         2            80
3                  5  ...                         3            80
4                  7  ...                         4            80
...              ...  ...                       ...           ...
1465            2061  ...                         3            80
1466            2062  ...                         1            80
1467            2064  ...                         2            80
1468            2065  ...                         4            80
1469            2068  ...                         1            80

      StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                    0                  8                      0
1                    1                 10                      3
2                    0                  7                      3
3                    0                  8                      3
4                    1                  6                      3
...                ...                ...                    ...
1465                 1                 17                      3
1466                 1                  9                      5
1467                 1                  6                      0
1468                 0                 17                      3
1469                 0                  6                      3

     WorkLifeBalance  YearsAtCompany YearsInCurrentRole  \
0                  1               6                  4
1                  3              10                  7
2                  3               0                  0
3                  3               8                  7
4                  3               2                  2
```

Handling Null Values

```
df = pd.DataFrame(data=dataset)
df
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research &	23	2	Medical	1	

```
df.isna().any()
```

```
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
```

```
df.isna().sum()
```

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              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
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
```

```
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

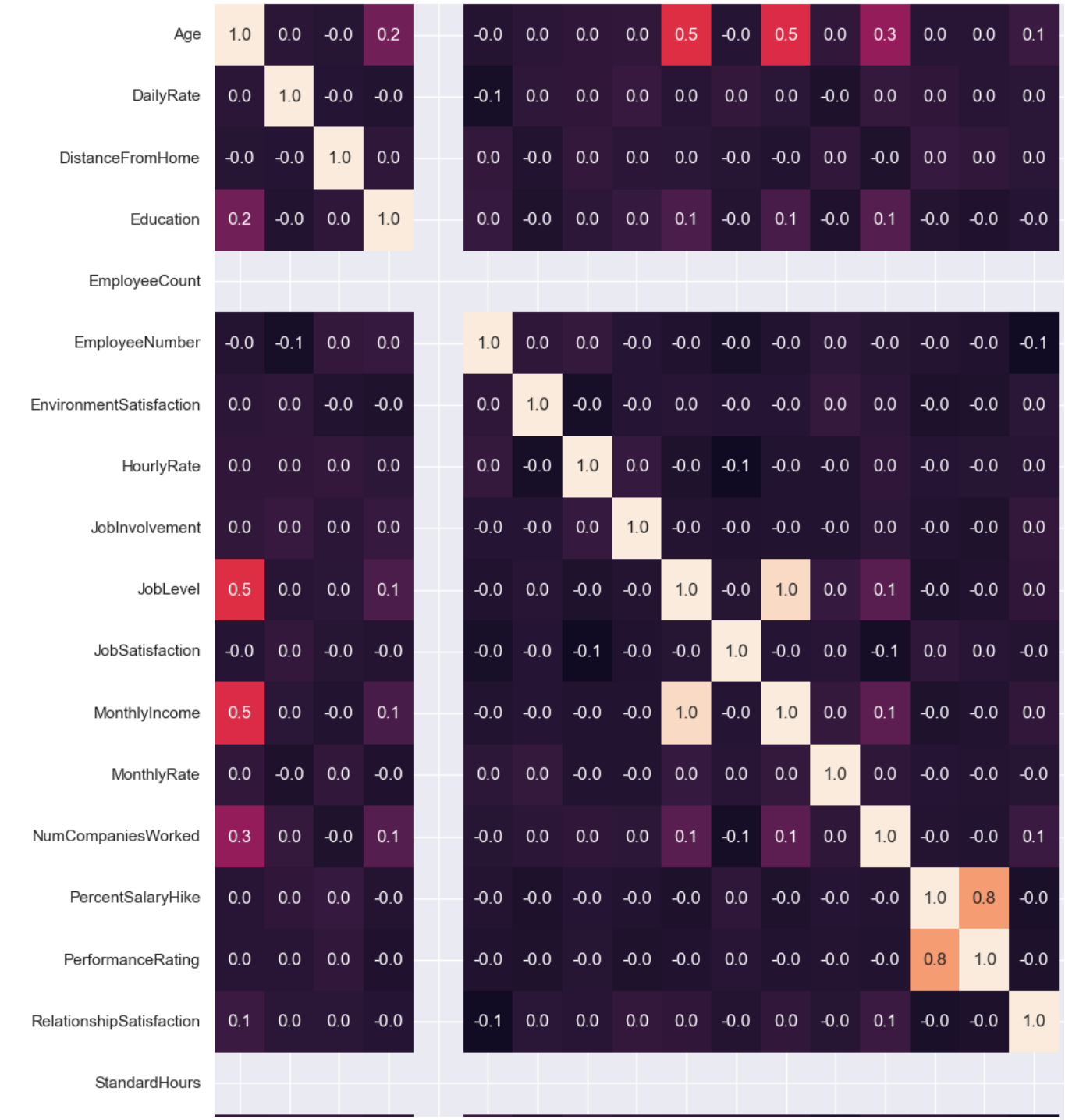
```
corr_mat = df.corr(numeric_only=True)
corr_mat
```

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

26 rows × 26 columns

```
sns.set(rc={'figure.figsize': (20,20)})
sns.heatmap(corr_mat,annot=True,fmt='.1f')
```

```
<Axes: >
```




```
# Drop the blannk area
df.drop(['EmployeeCount','StandardHours'],axis=1,inplace=True)
```

```
df['Over18'].nunique()
```

```
1
```

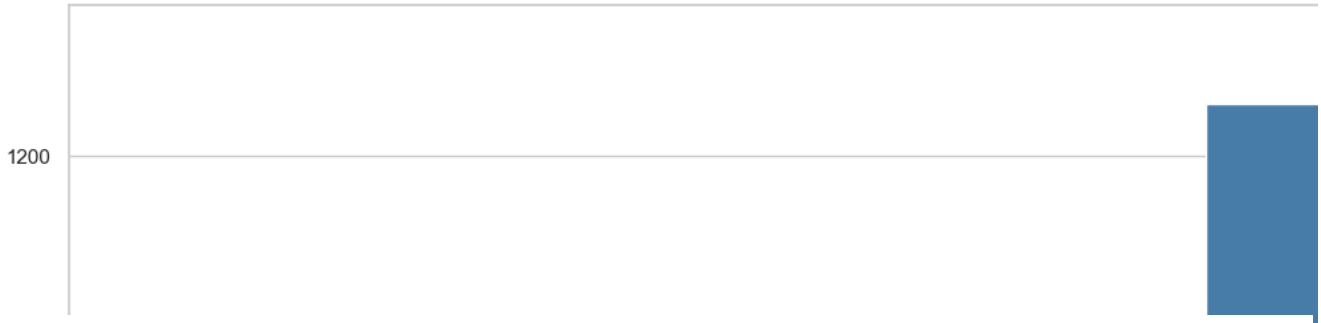
```
#remove Over18 as have repeating value
df.drop(columns=['Over18'],axis=1,inplace=True)
```

```
df.head()
```

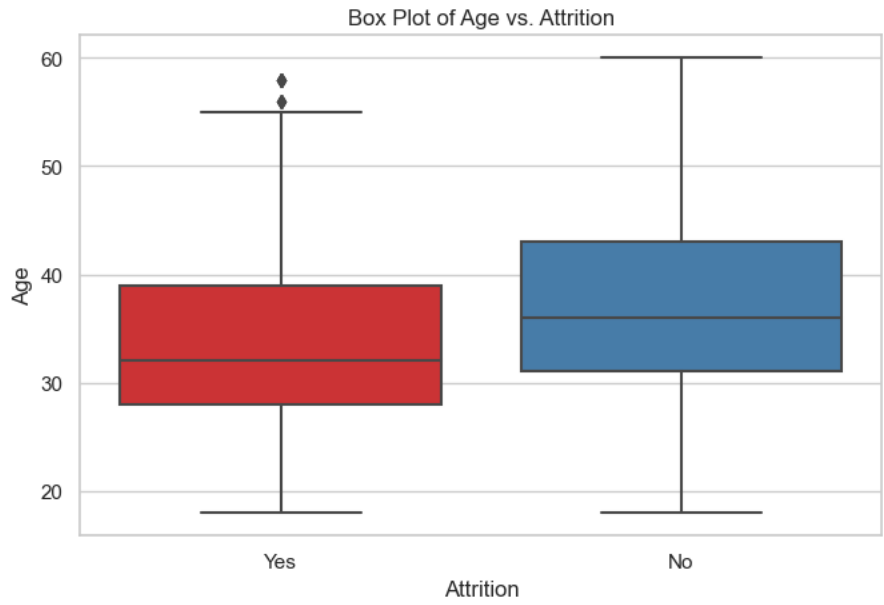
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeNumber	EnvironmentSa
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	2	
2	37	Yes	Travel_Rarely	1373	Research &	2	2	Other	4	

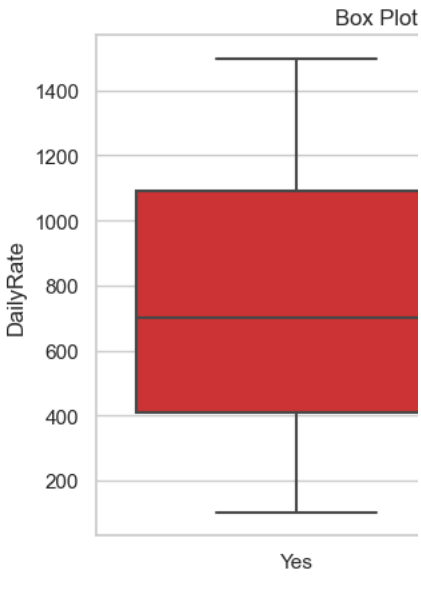
Data Visualization

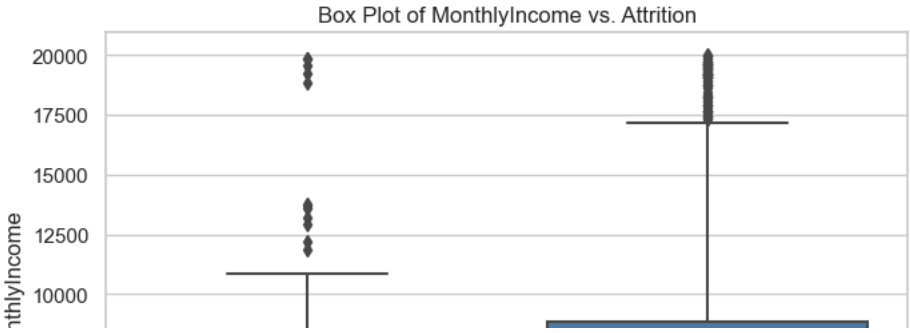
```
# Create the countplot
sns.set(style="whitegrid", palette='Set1')
ax = sns.countplot(x='Attrition', data=df, label='Count')

# Annotate the bars with the count values
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center', xytext=(0, 5), text

plt.show()
```

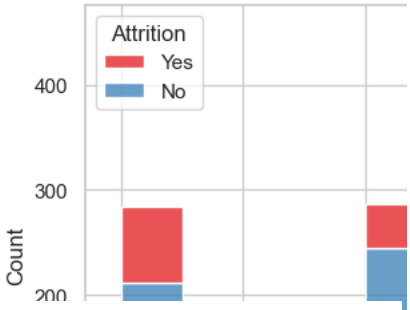



```
# A 3x2 grid of subplots
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(14, 14), constrained_layout=True)

# Subplot 1
sns.boxplot(x='Attrition', y='Age', data=dataset, palette='Set1', ax=axes[0, 0])
axes[0, 0].set_title("Box Plot of Age vs. Attrition", fontsize=12)

# Subplot 2
sns.boxplot(x='Attrition', y='DailyRate', data=dataset, palette='Set1', ax=axes[0, 1])
axes[0, 1].set_title("Box Plot of DailyRate vs. Attrition", fontsize=12)

# Subplot 3
sns.boxplot(x='Attrition', y='MonthlyIncome', data=dataset, palette='Set1', ax=axes[1, 0])
axes[1, 0].set_title("Box Plot of MonthlyIncome vs. Attrition", fontsize=12)

# Subplot 4
sns.histplot(data=dataset, x='EnvironmentSatisfaction', hue='Attrition', multiple='stack', palette='Set1', ax=axes[1, 1])
plt.title("Histogram of EnvironmentSatisfaction vs. Attrition", fontsize=12)

# Subplot 5
sns.histplot(data=dataset, x='StockOptionLevel', hue='Attrition', multiple='stack', palette='Set1', ax=axes[2, 0])
plt.title("Histogram of StockOptionLevel vs. Attrition", fontsize=12)

# Subplot 6
sns.boxplot(x='Attrition', y='DistanceFromHome', data=dataset, palette='Set1', ax=axes[2, 1])
axes[2, 1].set_title("Box Plot of DistanceFromHome vs. Attrition", fontsize=12)

# Adjust layout with vertical padding
plt.tight_layout()

# Show the plot
plt.show()
```







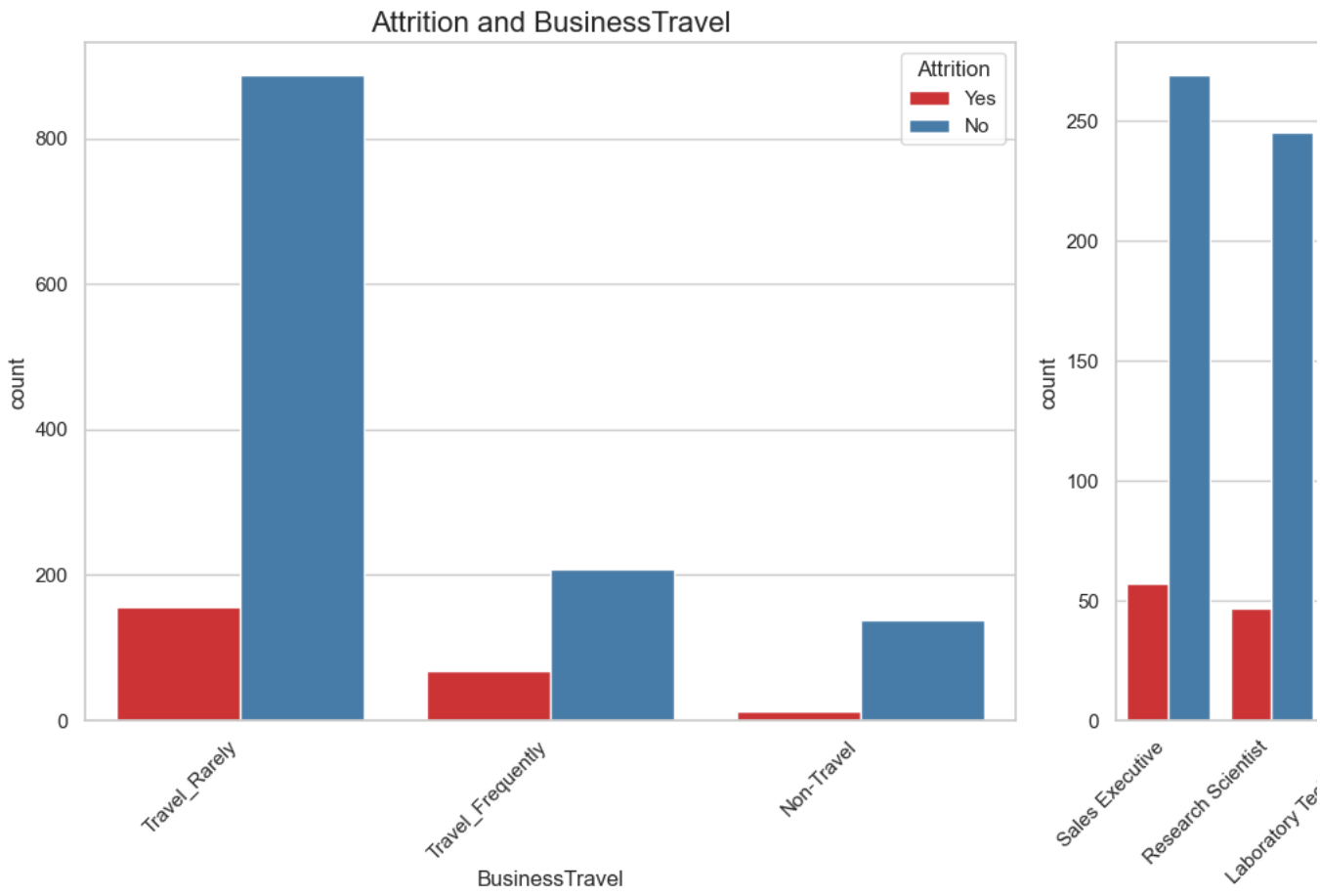


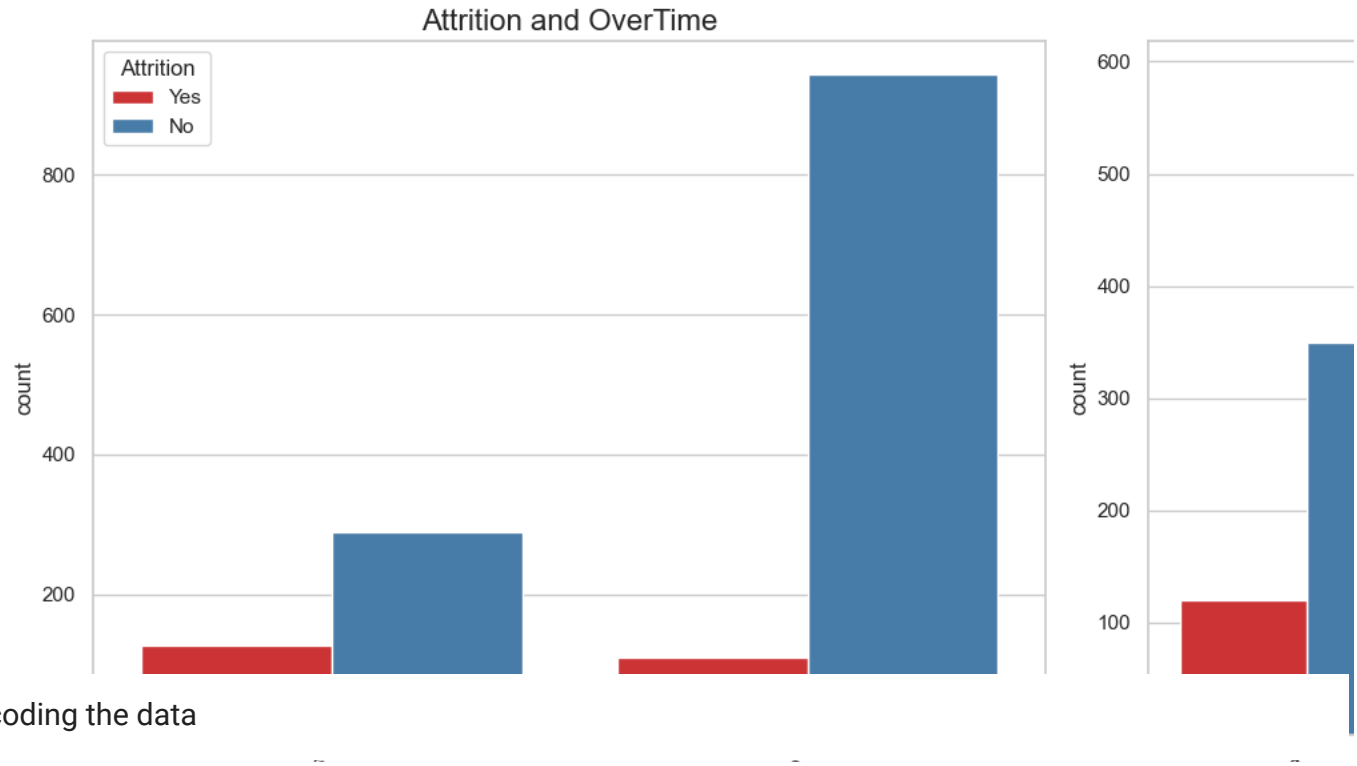
Inferences:

The graphs show that these employees are more prone to attrition:

- Younger age
- With a lower daily rate
- With a lower monthly income
- Working farther from home
- Having a lower Stock Option Level

```
# Visualizing categorical variables vs Attrition variable

# A 2x2 grid of subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(16, 14))

# Subplot 1
ax1 = sns.countplot(x='BusinessTravel', hue='Attrition', data=dataset, palette='Set1', ax=axes[0, 0])
ax1.set_title("Attrition and BusinessTravel", fontsize=16)
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=45, ha="right")

# Subplot 2
ax2 = sns.countplot(x='JobRole', hue='Attrition', data=dataset, palette='Set1', ax=axes[0, 1])
ax2.set_title("Attrition and JobRole", fontsize=16)
ax2.set_xticklabels(ax2.get_xticklabels(), rotation=45, ha="right")

# Subplot 3
ax3 = sns.countplot(x='OverTime', hue='Attrition', data=dataset, palette='Set1', ax=axes[1, 0])
ax3.set_title("Attrition and OverTime", fontsize=16)
ax3.set_xticklabels(ax3.get_xticklabels(), rotation=45, ha="right")

# Subplot 4
ax4 = sns.countplot(x='MaritalStatus', hue='Attrition', data=dataset, palette='Set1', ax=axes[1, 1])
ax4.set_title("Attrition and MaritalStatus", fontsize=16)
ax4.set_xticklabels(ax4.get_xticklabels(), rotation=45, ha="right")

# Adjust layout and show the plot
plt.tight_layout()
plt.show()
```





Encoding the data

```
dataset = pd.get_dummies(data=df,drop_first=True)
```

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 46 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Age                                1470 non-null   int64
 1   DailyRate                          1470 non-null   int64
 2   DistanceFromHome                   1470 non-null   int64
```

```
 3   Education                          1470 non-null   int64
 4   EmployeeNumber                     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  Attrition_Yes                      1470 non-null   bool
 25  BusinessTravel_Travel_Frequently   1470 non-null   bool
 26  BusinessTravel_Travel_Rarely       1470 non-null   bool
 27  Department_Research & Development  1470 non-null   bool
 28  Department_Sales                   1470 non-null   bool
 29  EducationField_Life Sciences       1470 non-null   bool
 30  EducationField_Marketing           1470 non-null   bool
 31  EducationField_Medical             1470 non-null   bool
 32  EducationField_Other               1470 non-null   bool
 33  EducationField_Technical Degree    1470 non-null   bool
 34  Gender_Male                        1470 non-null   bool
 35  JobRole_Human Resources            1470 non-null   bool
 36  JobRole_Laboratory Technician      1470 non-null   bool
 37  JobRole_Manager                    1470 non-null   bool
 38  JobRole_Manufacturing Director     1470 non-null   bool
 39  JobRole_Research Director          1470 non-null   bool
 40  JobRole_Research Scientist         1470 non-null   bool
 41  JobRole_Sales Executive            1470 non-null   bool
 42  JobRole_Sales Representative       1470 non-null   bool
 43  MaritalStatus_Married              1470 non-null   bool
 44  MaritalStatus_Single               1470 non-null   bool
 45  OverTime_Yes                       1470 non-null   bool
dtypes: bool(22), int64(24)
memory usage: 307.3 KB
```

```
# We see very high correlation between MonthlyIncome and JobLevel

correlation = dataset['MonthlyIncome'].corr(dataset['JobLevel'])

print(f"Correlation between MonthlyRate and JobLevel: {correlation}")
```

```
Correlation between MonthlyRate and JobLevel: 0.950299913479846
```

Sepration of Dependent and Independent Variables

```
X = dataset.drop(columns=['Attrition_Yes'],axis=1)
y = dataset['Attrition_Yes']
```

```
X
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	Job
0	41	1102	1	2	1	2	94	3	2	

```
y
```

```
0        True
1       False
2        True
3       False
4       False
        ...
1465    False
1466    False
1467    False
1468    False
1469    False
Name: Attrition_Yes, Length: 1470, dtype: bool
```

1468	49	1023	2	3	2065	4	63	2	2	

Splitting in Testing and Training Data

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

```
X_train.shape
```

```
(1176, 45)
```

```
X_test.shape
```

```
(294, 45)
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

MODEL BUILDING

1) Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr_clf = LogisticRegression(random_state=0)
```

```
lr_clf.fit(X_train,y_train)
```

```
LogisticRegression(random_state=0)
```

```
# Making prediction
y_pred = lr_clf.predict(X_test)
```

Load the metrics and Check the results

```
from sklearn.metrics import accuracy_score,confusion_matrix,f1_score,precision_score,recall_score
```

```
acc_lr = accuracy_score(y_test,y_pred)
f1_lr = f1_score(y_test,y_pred)
prec_lr = precision_score(y_test,y_pred)
rec_lr = recall_score(y_test,y_pred)
```

```
#Printing the result
results = pd.DataFrame([['LogisticRegression',acc_lr,f1_lr,prec_lr,rec_lr]],
                       columns=["Model","Accuracy","f1","Precision","Recall"])
results
```

	Model	Accuracy	f1	Precision	Recall
0	LogisticRegression	0.87415	0.519481	0.714286	0.408163

```
# Take a look at confusionn matrix
cm = confusion_matrix(y_test,y_pred)
cm
```

```
array([[237,   8],
       [ 29,  20]], dtype=int64)
```

```
# Cross validation Score
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=lr_clf,X =X_train,y=y_train,cv=10)
accuracies.mean()*100
```

```
87.41851368970013
```

```
accuracies.std()*100
```

```
2.4445072177051452
```

2) Decision Tree

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

```
dtc.fit(X_train,y_train)
```

```
DecisionTreeClassifier()
```

```
pred_dtc = dtc.predict(X_test)
pred_dtc
```

```
array([False, False,  True, False, False,  True,  True, False, False,
       False, False, False, False,  True, False, False, False,  True,
       False,  True, False, False, False, False,  True, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False,  True, False,  True, False, False,
       False, False, False, False, False, False, False, False,  True,
       False, False, False, False, False, False, False, False, False,
       False, False, False,  True,  True, False, False, False, False,
       False, False, False, False, False,  True, False, False, False,
       False, False, False,  True, False, False, False, False, False,
       False, False,  True,  True, False, False, False, False,  True,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False,  True,  True, False,
       False, False, False, False, False, False,  True,  True,  True,
       False, False, False,  True, False, False, False,  True, False,
        True, False, False,  True, False, False, False, False, False,
       False, False, False, False, False, False, False,  True, False,
       False,  True, False, False, False, False, False, False, False,
       False, False, False, False, False, False,  True,  True, False,
       False, False, False,  True, False, False,  True, False,  True,
       False, False,  True, False, False, False, False, False, False,
       False, False, False, False,  True, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False,  True, False, False,
       False, False,  True,  True, False, False,  True, False, False,
       False, False, False, False, False, False,  True, False, False,
       False, False, False, False, False, False, False, False, False,
        True, False, False, False, False,  True, False, False, False,
       False, False, False, False,  True, False,  True, False, False,
       False, False, False, False, False, False, False, False, False,
        True, False, False,  True, False, False,  True, False, False,
       False, False, False, False, False, False])
```

```
y_test
```

```
442     False
1091    False
981      True
785     False
1332     True
        ...
1439    False
481     False
124      True
198     False
1229    False
Name: Attrition_Yes, Length: 294, dtype: bool
```

```
acc_dtc = accuracy_score(y_test,pred_dtc)
f1_dtc = f1_score(y_test,pred_dtc)
prec_dtc = precision_score(y_test,pred_dtc)
rec_dtc = recall_score(y_test,pred_dtc)
```

```
result_dtc = pd.DataFrame([['Decision Tree',acc_dtc,f1_dtc,prec_dtc,rec_dtc]],
                      columns=["Model","Accuracy","f1","Precision","Recall"])
results = pd.concat([results,result_dtc],ignore_index=True)
results
```

	Model	Accuracy	f1	Precision	Recall
0	LogisticRegression	0.874150	0.519481	0.714286	0.408163
1	Decision Tree	0.812925	0.432990	0.437500	0.428571

```
cm_dtc = confusion_matrix(y_test,pred_dtc)
cm_dtc
```

```
array([[218,  27],
       [ 28,  21]], dtype=int64)
```

```
# Cross validation Score
from sklearn.model_selection import cross_val_score
accuracies_dtc = cross_val_score(estimator=dtc,X =X_train,y=y_train,cv=10)
accuracies_dtc.mean()*100
```

```
77.30189772562653
```

HyperParameter Tuning

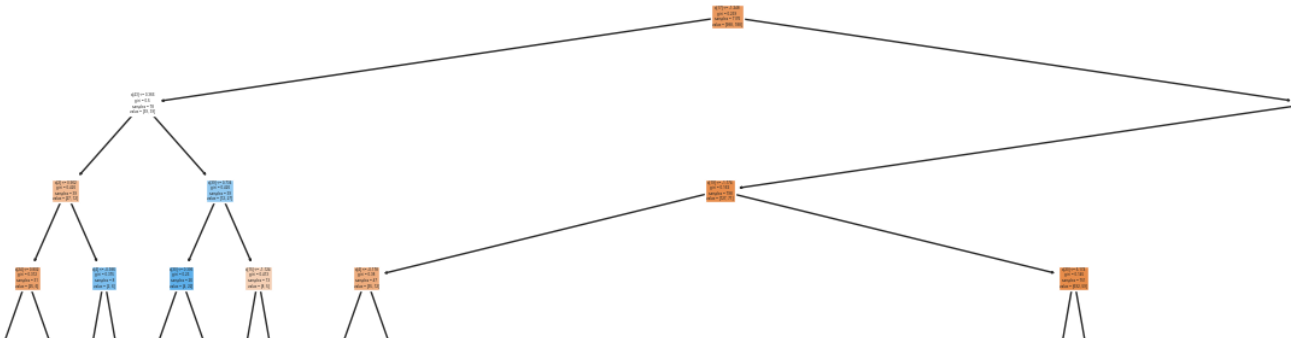
```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

```
[Text(0.33247887469084914, 0.96875, 'x[17] <= -1.248\ngini = 0.269\nsamples = 1176\nvalue = [988, 188]'),
 Text(0.08244023083264633, 0.90625, 'x[43] <= 0.366\ngini = 0.5\nsamples = 78\nvalue = [39, 39]'),
 Text(0.0494641384995878, 0.84375, 'x[2] <= 0.902\ngini = 0.426\nsamples = 39\nvalue = [27, 12]'),
 Text(0.03297609233305853, 0.78125, 'x[24] <= 0.802\ngini = 0.312\nsamples = 31\nvalue = [25, 6]'),
 Text(0.01978565539983512, 0.71875, 'x[9] <= -1.119\ngini = 0.198\nsamples = 27\nvalue = [24, 3]'),
 Text(0.013190436933223413, 0.65625, 'x[44] <= 0.505\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.006595218466611707, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.01978565539983512, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.026380873866446827, 0.65625, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(0.046166529266281946, 0.71875, 'x[0] <= -0.31\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.03957131079967024, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.05276174773289365, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.06595218466611706, 0.78125, 'x[4] <= -0.386\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.05935696619950536, 0.71875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.07254740313272877, 0.71875, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.11541632316570487, 0.84375, 'x[39] <= 0.734\ngini = 0.426\nsamples = 39\nvalue = [12, 27]'),
 Text(0.0989282769991756, 0.78125, 'x[30] <= 0.395\ngini = 0.26\nsamples = 26\nvalue = [4, 22]'),
 Text(0.08573784006595218, 0.71875, 'x[6] <= 1.522\ngini = 0.095\nsamples = 20\nvalue = [1, 19]'),
 Text(0.07914262159934048, 0.65625, 'gini = 0.0\nsamples = 18\nvalue = [0, 18]'),
 Text(0.09233305853256389, 0.65625, 'x[14] <= 0.977\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.08573784006595218, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.0989282769991756, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.11211871393239901, 0.71875, 'x[4] <= -0.41\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.1055234954657873, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.11871393239901072, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.13190436933223412, 0.78125, 'x[15] <= -1.124\ngini = 0.473\nsamples = 13\nvalue = [8, 5]'),
 Text(0.12530915086562242, 0.71875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.13849958779884583, 0.71875, 'x[11] <= 1.11\ngini = 0.32\nsamples = 10\nvalue = [8, 2]'),
 Text(0.13190436933223412, 0.65625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.14509480626545754, 0.65625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.582517518549052, 0.90625, 'x[44] <= 0.505\ngini = 0.235\nsamples = 1098\nvalue = [949, 149]'),
 Text(0.3295032976092333, 0.84375, 'x[19] <= -1.774\ngini = 0.162\nsamples = 798\nvalue = [727, 71]'),
 Text(0.17807089859851608, 0.78125, 'x[4] <= -0.178\ngini = 0.38\nsamples = 47\nvalue = [35, 12]'),
 Text(0.16488046166529266, 0.71875, 'x[43] <= 0.366\ngini = 0.1\nsamples = 19\nvalue = [18, 1]'),
 Text(0.15828524319868095, 0.65625, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),
 Text(0.17147568013190437, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1912613355317395, 0.71875, 'x[10] <= -0.777\ngini = 0.477\nsamples = 28\nvalue = [17, 11]'),
 Text(0.18466611706512778, 0.65625, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.1978565539983512, 0.65625, 'x[4] <= 0.097\ngini = 0.413\nsamples = 24\nvalue = [17, 7]'),
 Text(0.1912613355317395, 0.59375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.2044517724649629, 0.59375, 'x[23] <= 0.393\ngini = 0.351\nsamples = 22\nvalue = [17, 5]'),
 Text(0.1912613355317395, 0.53125, 'x[1] <= -1.638\ngini = 0.133\nsamples = 14\nvalue = [13, 1]'),
 Text(0.18466611706512778, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1978565539983512, 0.46875, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(0.21764220939818632, 0.53125, 'x[1] <= -0.592\ngini = 0.5\nsamples = 8\nvalue = [4, 4]'),
 Text(0.2110469909315746, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.22423742786479803, 0.46875, 'x[22] <= 2.601\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.21764220939818632, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.23083264633140974, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.48093569661995056, 0.78125, 'x[20] <= 5.174\ngini = 0.145\nsamples = 751\nvalue = [692, 59]'),
 Text(0.47434047815333885, 0.71875, 'x[20] <= -0.407\ngini = 0.143\nsamples = 750\nvalue = [692, 58]'),
 Text(0.3322341302555647, 0.65625, 'x[5] <= -1.142\ngini = 0.218\nsamples = 257\nvalue = [225, 32]'),
 Text(0.3017312448474856, 0.59375, 'x[23] <= -0.449\ngini = 0.355\nsamples = 65\nvalue = [50, 15]'),
 Text(0.2802967848309975, 0.53125, 'x[23] <= -1.01\ngini = 0.303\nsamples = 59\nvalue = [48, 11]'),
 Text(0.25721352019785654, 0.46875, 'x[7] <= -0.314\ngini = 0.463\nsamples = 22\nvalue = [14, 8]'),
 Text(0.24402308326463315, 0.40625, 'x[6] <= -1.116\ngini = 0.198\nsamples = 9\nvalue = [8, 1]'),
 Text(0.23742786479802144, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.25061830173124483, 0.34375, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.27040395713107995, 0.40625, 'x[6] <= -0.351\ngini = 0.497\nsamples = 13\nvalue = [6, 7]'),
 Text(0.26380873866446825, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.27699917559769166, 0.34375, 'x[2] <= -0.02\ngini = 0.346\nsamples = 9\nvalue = [2, 7]'),
 Text(0.27040395713107995, 0.28125, 'x[6] <= 0.437\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.26380873866446825, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.27699917559769166, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.28359439406430337, 0.28125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.3033800494641385, 0.46875, 'x[9] <= -1.119\ngini = 0.149\nsamples = 37\nvalue = [34, 3]'),
 Text(0.2967848309975268, 0.40625, 'x[19] <= -0.371\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.2901896125309151, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.3033800494641385, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.3099752679307502, 0.40625, 'gini = 0.0\nsamples = 31\nvalue = [31, 0]'),
 Text(0.3231657048639736, 0.53125, 'x[4] <= -1.5\ngini = 0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.3165704863973619, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3297609233305853, 0.46875, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.36273701566364386, 0.59375, 'x[28] <= 0.169\ngini = 0.161\nsamples = 192\nvalue = [175, 17]'),
 Text(0.35614179719703215, 0.53125, 'x[19] <= -0.371\ngini = 0.24\nsamples = 122\nvalue = [105, 17]'),
 Text(0.34295136026380874, 0.46875, 'x[6] <= 0.437\ngini = 0.463\nsamples = 22\nvalue = [14, 8]'),
 Text(0.33635614179719703, 0.40625, 'x[0] <= -0.144\ngini = 0.444\nsamples = 12\nvalue = [4, 8]'),
 Text(0.3297609233305853, 0.34375, 'x[13] <= 0.643\ngini = 0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(0.3231657048639736, 0.28125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.33635614179719703, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.34295136026380874, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.34954657873042044, 0.40625, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(0.36933223413025557, 0.46875, 'x[1] <= -1.699\ngini = 0.164\nsamples = 100\nvalue = [91, 9]'),
 Text(0.36273701566364386, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3759274525968673, 0.40625, 'x[4] <= -1.651\ngini = 0.149\nsamples = 99\nvalue = [91, 8]'),
 Text(0.35696619950535863, 0.34375, 'x[35] <= 0.839\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.3503709810387469, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.36356141797197034, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3948887056883759, 0.34375, 'x[1] <= 1.52\ngini = 0.118\nsamples = 95\nvalue = [89, 6]'),
 Text(0.37675185490519375, 0.28125, 'x[3] <= 1.567\ngini = 0.086\nsamples = 89\nvalue = [85, 4]'),
```

```
Text(0.3602638087386645, 0.21875, 'x[0] <= 2.173\ngini = 0.047\nsamples = 83\nvalue = [81, 2]'),
Text(0.34707337180544107, 0.15625, 'x[7] <= -1.713\ngini = 0.024\nsamples = 81\nvalue = [80, 1]'),
Text(0.34047815333882936, 0.09375, 'x[22] <= -0.206\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.33388293487221765, 0.03125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.34707337180544107, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.3536685902720528, 0.09375, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),
Text(0.3734542456718879, 0.15625, 'x[16] <= -0.353\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.3668590272052762, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.3800494641384996, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.393239901071723, 0.21875, 'x[11] <= 0.356\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),
Text(0.3866446826051113, 0.15625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.3998351195383347, 0.15625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.41302555647155814, 0.28125, 'x[13] <= -0.589\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),
Text(0.40643033800494643, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.41962077493816985, 0.21875, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.36933223413025557, 0.53125, 'gini = 0.0\nsamples = 70\nvalue = [70, 0]'),
Text(0.6164468260511129, 0.65625, 'x[20] <= 4.025\ngini = 0.1\nsamples = 493\nvalue = [467, 26]'),
Text(0.5758450123660346, 0.59375, 'x[9] <= -0.211\ngini = 0.094\nsamples = 486\nvalue = [462, 24]'),
Text(0.5144270403957131, 0.53125, 'x[41] <= 1.929\ngini = 0.154\nsamples = 191\nvalue = [175, 16]'),
Text(0.5078318219291014, 0.46875, 'x[11] <= -0.026\ngini = 0.145\nsamples = 190\nvalue = [175, 15]'),
Text(0.48392415498763397, 0.40625, 'x[23] <= 2.639\ngini = 0.221\nsamples = 95\nvalue = [83, 12]'),
Text(0.47732893652102226, 0.34375, 'x[11] <= -0.063\ngini = 0.207\nsamples = 94\nvalue = [83, 11]'),
Text(0.47073371805441055, 0.28125, 'x[24] <= 0.802\ngini = 0.192\nsamples = 93\nvalue = [83, 10]'),
Text(0.4361088211046991, 0.21875, 'x[10] <= 0.446\ngini = 0.124\nsamples = 75\nvalue = [70, 5]'),
Text(0.41302555647155814, 0.15625, 'x[37] <= 1.338\ngini = 0.037\nsamples = 53\nvalue = [52, 1]'),
Text(0.40643033800494643, 0.09375, 'gini = 0.0\nsamples = 45\nvalue = [45, 0]'),
Text(0.41962077493816985, 0.09375, 'x[15] <= -1.124\ngini = 0.219\nsamples = 8\nvalue = [7, 1]'),
Text(0.41302555647155814, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.42621599340478156, 0.03125, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.45919208573784004, 0.15625, 'x[3] <= -0.414\ngini = 0.298\nsamples = 22\nvalue = [18, 4]'),
Text(0.4460016488046167, 0.09375, 'x[29] <= 1.285\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.43940643033800497, 0.03125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.4525968672712284, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.4723825226710634 5, 0.09375, 'x[2] <= -0.942\ngini = 0.105\nsamples = 18\nvalue = [17, 1]'),
Text(0.4657873042044517 5, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4789774113767516, 0.03125, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
Text(0.505358615004122, 0.21875, 'x[43] <= 0.366\ngini = 0.401\nsamples = 18\nvalue = [13, 5]'),
Text(0.4921681780708986, 0.15625, 'x[1] <= -1.326\ngini = 0.142\nsamples = 13\nvalue = [12, 1]'),
Text(0.4855729596042868 7, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4987633965375103, 0.09375, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
Text(0.5185490519373455, 0.15625, 'x[6] <= -0.549\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
Text(0.5119538334707338, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.5251442704039572, 0.09375, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
Text(0.48392415498763397, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4905193734542457, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.5317394888705689, 0.40625, 'x[12] <= 0.713\ngini = 0.061\nsamples = 95\nvalue = [92, 3]'),
Text(0.5251442704039572, 0.34375, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),
Text(0.5383347073371806, 0.34375, 'x[23] <= -0.729\ngini = 0.266\nsamples = 19\nvalue = [16, 3]'),
Text(0.5251442704039572, 0.28125, 'x[10] <= 0.829\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.5185490519373455, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.5317394888705689, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.551525144270404, 0.28125, 'x[10] <= -0.719\ngini = 0.117\nsamples = 16\nvalue = [15, 1]'),
Text(0.5449299258037923, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.5581203627370157, 0.21875, 'gini = 0.0\nsamples = 15\nvalue = [15, 0]'),
Text(0.5210222588623248, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6372629843363561, 0.53125, 'x[13] <= -1.0\ngini = 0.053\nsamples = 295\nvalue = [287, 8]'),
Text(0.6141797197032152, 0.46875, 'x[22] <= 2.601\ngini = 0.159\nsamples = 46\nvalue = [42, 4]'),
Text(0.6075845012366035, 0.40625, 'x[17] <= 2.11\ngini = 0.124\nsamples = 45\nvalue = [42, 3]'),
Text(0.5910964550700742, 0.34375, 'x[7] <= -1.713\ngini = 0.089\nsamples = 43\nvalue = [41, 2]'),
Text(0.5779060181368508, 0.28125, 'x[22] <= 1.821\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.5713107996702391, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.5845012366034625, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6042868920032977, 0.28125, 'x[0] <= 0.684\ngini = 0.048\nsamples = 41\nvalue = [40, 1]'),
Text(0.597691673536686, 0.21875, 'gini = 0.0\nsamples = 33\nvalue = [33, 0]'),
Text(0.6108821104699094, 0.21875, 'x[0] <= 0.849\ngini = 0.219\nsamples = 8\nvalue = [7, 1]'),
Text(0.6042868920032977, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6174773289365211, 0.15625, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.6240725474031328, 0.34375, 'x[30] <= 0.395\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.6174773289365211, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6306677658697445, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6207749381698269, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6603462489694971, 0.46875, 'x[10] <= -0.931\ngini = 0.032\nsamples = 249\nvalue = [245, 4]'),
Text(0.6438582028029678, 0.40625, 'x[19] <= -0.371\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.6372629843363561, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6504534212695795, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.6768342951360263, 0.40625, 'x[1] <= -1.673\ngini = 0.024\nsamples = 244\nvalue = [241, 3]'),
Text(0.6636438582028029, 0.34375, 'x[13] <= 1.602\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.6570486397361912, 0.28125, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.6702390766694146, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6900247320692497, 0.34375, 'x[15] <= -1.124\ngini = 0.017\nsamples = 238\nvalue = [236, 2]'),
Text(0.683429513602638, 0.28125, 'x[19] <= 1.032\ngini = 0.073\nsamples = 53\nvalue = [51, 2]'),
Text(0.6702390766694146, 0.21875, 'x[35] <= 0.839\ngini = 0.041\nsamples = 48\nvalue = [47, 1]'),
Text(0.6636438582028029, 0.15625, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
Text(0.6768342951360263, 0.15625, 'x[23] <= -0.028\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.6702390766694146, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.683429513602638, 0.09375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.6966199505358615, 0.21875, 'x[21] <= 0.909\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.6900247320692497, 0.15625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.7032151690024732, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6966199505358615, 0.28125, 'gini = 0.0\nsamples = 185\nvalue = [185, 0]'),
Text(0.6570486397361912, 0.59375, 'x[33] <= -0.218\ngini = 0.408\nsamples = 7\nvalue = [5, 2]')
```

```
Text(0.6570486397561912, 0.59375, 'x[33] <= -0.218\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
Text(0.6504534212695795, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.6636438582028029, 0.53125, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.48753091508656227, 0.71875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.8355317394888706, 0.84375, 'x[10] <= -0.522\ngini = 0.385\nsamples = 300\nvalue = [222, 78]'),
Text(0.7535037098103875, 0.78125, 'x[16] <= -0.353\ngini = 0.5\nsamples = 96\nvalue = [49, 47]'),
Text(0.7197032151690025, 0.71875, 'x[2] <= -0.45\ngini = 0.459\nsamples = 42\nvalue = [15, 27]'),
Text(0.6966199505358615, 0.65625, 'x[4] <= -0.282\ngini = 0.499\nsamples = 23\nvalue = [12, 11]'),
Text(0.683429513602638, 0.59375, 'x[11] <= 0.254\ngini = 0.355\nsamples = 13\nvalue = [3, 10]'),
Text(0.6768342951360263, 0.53125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
Text(0.6900247320692497, 0.53125, 'x[18] <= 0.586\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
Text(0.683429513602638, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.6966199505358615, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.7098103874690849, 0.59375, 'x[41] <= 1.929\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),
Text(0.7032151690024732, 0.53125, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
Text(0.7164056059356966, 0.53125, 'x[25] <= -0.452\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.7098103874690849, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7230008244023083, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7427864798021434, 0.65625, 'x[17] <= 0.367\ngini = 0.266\nsamples = 19\nvalue = [3, 16]'),
Text(0.7361912613355317, 0.59375, 'x[6] <= -1.042\ngini = 0.198\nsamples = 18\nvalue = [2, 16]'),
Text(0.72959604286892, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7427864798021434, 0.53125, 'x[0] <= 0.684\ngini = 0.111\nsamples = 17\nvalue = [1, 16]'),
Text(0.7361912613355317, 0.46875, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),
Text(0.7493816982687551, 0.46875, 'x[23] <= -0.729\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.7427864798021434, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7559769167353668, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7493816982687551, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7873042044517725, 0.71875, 'x[0] <= -1.137\ngini = 0.466\nsamples = 54\nvalue = [34, 20]'),
Text(0.7691673536685902, 0.65625, 'x[7] <= 1.085\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),
Text(0.7625721352019785, 0.59375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
Text(0.7757625721352019, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.8054410552349547, 0.65625, 'x[1] <= 0.418\ngini = 0.418\nsamples = 47\nvalue = [33, 14]'),
Text(0.7889530090684254, 0.59375, 'x[1] <= -1.227\ngini = 0.482\nsamples = 32\nvalue = [19, 13]'),
Text(0.7757625721352019, 0.53125, 'x[2] <= 1.825\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),
Text(0.7691673536685902, 0.46875, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
Text(0.7823577906018137, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.8021434460016488, 0.53125, 'x[11] <= 1.335\ngini = 0.496\nsamples = 22\nvalue = [10, 12]'),
Text(0.7955482275350371, 0.46875, 'x[28] <= 0.169\ngini = 0.465\nsamples = 19\nvalue = [7, 12]'),
Text(0.7823577906018137, 0.40625, 'x[1] <= 0.121\ngini = 0.298\nsamples = 11\nvalue = [2, 9]'),
Text(0.7757625721352019, 0.34375, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
Text(0.7889530090684254, 0.34375, 'x[10] <= -0.877\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.7823577906018137, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7955482275350371, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.8087386644682605, 0.40625, 'x[35] <= 0.839\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
Text(0.8021434460016488, 0.34375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.8153338829348722, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.8087386644682605, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.8219291014014839, 0.59375, 'x[6] <= -1.51\ngini = 0.124\nsamples = 15\nvalue = [14, 1]'),
Text(0.8153338829348722, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.8285243198680956, 0.53125, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
Text(0.9175597691673537, 0.78125, 'x[43] <= 0.366\ngini = 0.258\nsamples = 204\nvalue = [173, 31]'),
Text(0.8713932399010718, 0.71875, 'x[10] <= 2.836\ngini = 0.138\nsamples = 147\nvalue = [136, 11]'),
Text(0.86479802143446, 0.65625, 'x[2] <= 0.657\ngini = 0.128\nsamples = 146\nvalue = [136, 10]'),
Text(0.8483099752679307, 0.59375, 'x[20] <= -0.736\ngini = 0.038\nsamples = 104\nvalue = [102, 2]'),
Text(0.841714756801319, 0.53125, 'x[6] <= -1.066\ngini = 0.32\nsamples = 10\nvalue = [8, 2]'),
Text(0.8351195383347073, 0.46875, 'x[10] <= 1.253\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.8285243198680956, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.841714756801319, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.8483099752679307, 0.46875, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.8549051937345424, 0.53125, 'gini = 0.0\nsamples = 94\nvalue = [94, 0]'),
Text(0.8812860676009893, 0.59375, 'x[29] <= 1.285\ngini = 0.308\nsamples = 42\nvalue = [34, 8]'),
Text(0.8680956306677658, 0.53125, 'x[5] <= -0.229\ngini = 0.229\nsamples = 38\nvalue = [33, 5]'),
Text(0.8615004122011541, 0.46875, 'x[12] <= 0.911\ngini = 0.486\nsamples = 12\nvalue = [7, 5]'),
Text(0.8549051937345424, 0.40625, 'x[4] <= -1.084\ngini = 0.346\nsamples = 9\nvalue = [7, 2]'),
Text(0.8483099752679307, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.8615004122011541, 0.34375, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.8680956306677658, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.8746908491343776, 0.46875, 'gini = 0.0\nsamples = 26\nvalue = [26, 0]'),
Text(0.8944765045342127, 0.53125, 'x[20] <= 1.07\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.887881286067601, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.9010717230008244, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.8779884583676835, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.9637262984336357, 0.71875, 'x[40] <= 0.678\ngini = 0.456\nsamples = 57\nvalue = [37, 20]'),
Text(0.9406430338004946, 0.65625, 'x[22] <= 1.197\ngini = 0.238\nsamples = 29\nvalue = [25, 4]'),
Text(0.9274525968672712, 0.59375, 'x[4] <= -1.479\ngini = 0.142\nsamples = 26\nvalue = [24, 2]'),
Text(0.9208573784006595, 0.53125, 'x[13] <= -0.726\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.9142621599340478, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.9274525968672712, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.9340478153338829, 0.53125, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
Text(0.953833470733718, 0.59375, 'x[1] <= -0.611\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.9472382522671063, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.9604286892003298, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.9868095630667766, 0.65625, 'x[22] <= -0.206\ngini = 0.49\nsamples = 28\nvalue = [12, 16]'),
Text(0.9802143446001649, 0.59375, 'x[7] <= 1.085\ngini = 0.48\nsamples = 20\nvalue = [12, 8]'),
Text(0.9736191261335532, 0.53125, 'x[2] <= -0.942\ngini = 0.415\nsamples = 17\nvalue = [12, 5]'),
Text(0.9670239076669415, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.9802143446001649, 0.46875, 'x[10] <= 0.312\ngini = 0.32\nsamples = 15\nvalue = [12, 3]'),
Text(0.9736191261335532, 0.40625, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
Text(0.9868095630667766, 0.40625, 'x[15] <= 0.727\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.9802143446001649, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.9934047815333883, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
```

```
Text(0.9868095630667766, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.9934047815333883, 0.59375, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]')]
```



```
from sklearn.model_selection import GridSearchCV
parameter={
 'criterion':['gini','entropy'],
  'splitter':['best','random'],
  'max_depth':[1,2,3,4,5],
  'max_features':['auto', 'sqrt', 'log2']

}
```

```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accuracy")
```

```
grid_search.fit(X_train,y_train)
```

GridSearchCV
estimator: DecisionTreeClassifier
DecisionTreeClassifier

```
grid_search.best_params_
```

```
{'criterion': 'entropy',
 'max_depth': 5,
 'max_features': 'log2',
 'splitter': 'random'}
```

```
dtc_cv=DecisionTreeClassifier(criterion= 'entropy',
 max_depth=3,
 max_features='sqrt',
 splitter='best')
dtc_cv.fit(X_train,y_train)
```

DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features='sqrt')

```
pred1=dtc_cv.predict(X_test)
pred1
```

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False,  True, False,
```

```
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False])
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,pred1))
```

```
              precision    recall  f1-score   support

       False       0.83      1.00      0.91       245
        True       0.00      0.00      0.00        49

    accuracy                           0.83       294
   macro avg       0.42      0.50      0.45       294
weighted avg       0.69      0.83      0.76       294
```

3) Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
```

```
forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(range(0,14))}]
```

```
rfc_cv= GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring="accuracy")
```

```
rfc_cv.fit(X_train,y_train)
```

```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

```
pred2=rfc_cv.predict(X_test)
```

```
print(classification_report(y_test,pred2))
```

```
              precision    recall  f1-score   support

       False       0.86      0.99      0.92       245
        True       0.75      0.18      0.30        49

    accuracy                           0.85       294
   macro avg       0.80      0.59      0.61       294
weighted avg       0.84      0.85      0.81       294
```

```
rfc_cv.best_params_
```

```
{'max_depth': 13, 'max_features': 8}
```