In [1]:
```
## Decision Tree
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
#### import Data
data = pd.read_csv(r"C:\Users\shubham lokare\Downloads\DT Model with Flask (1)\credit.csv")
```

In [4]:
```
data
```

Out[4]:

	checking_balance	months_loan_duration	credit_history	purpose	amount	savings_balance	employment_duration	percent_of_i
0	< 0 DM	6	critical	furniture/appliances	1169	unknown	> 7 years	
1	1 - 200 DM	48	good	furniture/appliances	5951	< 100 DM	1 - 4 years	
2	unknown	12	critical	education	2096	< 100 DM	4 - 7 years	
3	< 0 DM	42	good	furniture/appliances	7882	< 100 DM	4 - 7 years	
4	< 0 DM	24	poor	car	4870	< 100 DM	1 - 4 years	
...	...	...	...	...	...	...	...	
995	unknown	12	good	furniture/appliances	1736	< 100 DM	4 - 7 years	
996	< 0 DM	30	good	car	3857	< 100 DM	1 - 4 years	
997	unknown	12	good	furniture/appliances	804	< 100 DM	> 7 years	
998	< 0 DM	45	good	furniture/appliances	1845	< 100 DM	1 - 4 years	
999	1 - 200 DM	45	critical	car	4576	100 - 500 DM	unemployed	

1000 rows × 17 columns

In [5]:
```
data.head(10)
```

Out[5]:

	checking_balance	months_loan_duration	credit_history	purpose	amount	savings_balance	employment_duration	percent_of_inc
0	< 0 DM	6	critical	furniture/appliances	1169	unknown	> 7 years	
1	1 - 200 DM	48	good	furniture/appliances	5951	< 100 DM	1 - 4 years	
2	unknown	12	critical	education	2096	< 100 DM	4 - 7 years	
3	< 0 DM	42	good	furniture/appliances	7882	< 100 DM	4 - 7 years	
4	< 0 DM	24	poor	car	4870	< 100 DM	1 - 4 years	
5	unknown	36	good	education	9055	unknown	1 - 4 years	
6	unknown	24	good	furniture/appliances	2835	500 - 1000 DM	> 7 years	
7	1 - 200 DM	36	good	car	6948	< 100 DM	1 - 4 years	
8	unknown	12	good	furniture/appliances	3059	> 1000 DM	4 - 7 years	
9	1 - 200 DM	30	critical	car	5234	< 100 DM	unemployed	

In [6]:
```
data = data.drop(["phone"], axis = 1) # Unwanted columns are removed.
```

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   checking_balance      1000 non-null   object 
 1   months_loan_duration  1000 non-null   int64  
 2   credit_history        1000 non-null   object 
 3   purpose               1000 non-null   object 
 4   amount                1000 non-null   int64  
 5   savings_balance       1000 non-null   object 
 6   employment_duration   1000 non-null   object 
 7   percent_of_income     1000 non-null   int64  
 8   years_at_residence    1000 non-null   int64  
 9   age                   997 non-null    float64
 10  other_credit          1000 non-null   object 
 11  housing               1000 non-null   object 
 12  existing_loans_count  1000 non-null   int64  
 13  job                   1000 non-null   object 
 14  dependents            1000 non-null   int64  
 15  default               1000 non-null   object 
dtypes: float64(1), int64(6), object(9)
memory usage: 125.1+ KB
```

```
In [8]: ### check missing values
data.isna().sum()
```

```
Out[8]:
checking_balance        0
months_loan_duration    0
credit_history          0
purpose                 0
amount                  0
savings_balance         0
employment_duration     0
percent_of_income       0
years_at_residence      0
age                     3
other_credit            0
housing                 0
existing_loans_count    0
job                     0
dependents              0
default                 0
dtype: int64
```

```
In [9]: #### split the input variable and output variable

X = pd.DataFrame(data.iloc[: , 0:15])   ### input
Y = pd.DataFrame(data.iloc[: ,-1])      ### target
```

```
In [10]: X
```

Out[10]:

	checking_balance	months_loan_duration	credit_history	purpose	amount	savings_balance	employment_duration	percent_of_i
0	< 0 DM	6	critical	furniture/appliances	1169	unknown	> 7 years	
1	1 - 200 DM	48	good	furniture/appliances	5951	< 100 DM	1 - 4 years	
2	unknown	12	critical	education	2096	< 100 DM	4 - 7 years	
3	< 0 DM	42	good	furniture/appliances	7882	< 100 DM	4 - 7 years	
4	< 0 DM	24	poor	car	4870	< 100 DM	1 - 4 years	
...	...	...	...	...	...	...	...	
995	unknown	12	good	furniture/appliances	1736	< 100 DM	4 - 7 years	
996	< 0 DM	30	good	car	3857	< 100 DM	1 - 4 years	
997	unknown	12	good	furniture/appliances	804	< 100 DM	> 7 years	
998	< 0 DM	45	good	furniture/appliances	1845	< 100 DM	1 - 4 years	
999	1 - 200 DM	45	critical	car	4576	100 - 500 DM	unemployed	

1000 rows × 15 columns

```
In [11]: Y
```

Out[11]:

	default
0	no
1	yes
2	no
3	no
4	yes
...	...
995	no
996	no
997	no
998	yes
999	no

1000 rows × 1 columns

```
In [12]: # #### Separating Numeric and Non-Numeric columns

numeric_features = X.select_dtypes(exclude = ['object']).columns
numeric_features

categorical_features = X.select_dtypes(include=['object']).columns
categorical_features
```

```
Out[12]: Index(['checking_balance', 'credit_history', 'purpose', 'savings_balance',
       'employment_duration', 'other_credit', 'housing', 'job'],
      dtype='object')
```

In [13]:
```python
# ### Data Preprocessing

# Numeric_features
# ### Imputation to handle missing values
# ### MinMaxScaler to convert the magnitude of the columns to a range of 0 to 1

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline
```

In [14]:
```python
num_pipeline = Pipeline(steps = [('impute', SimpleImputer(strategy = 'mean')),
                                 ('scale', MinMaxScaler())])
```

In [18]:
```python
# ### Encoding - One Hot Encoder to convert Categorical data to Numeric values
# Categorical features
from sklearn.preprocessing import OneHotEncoder

encoding_pipeline = Pipeline([('onehot', OneHotEncoder(drop = 'first'))])
```

In [21]:
```python
# Creating a transformation of variable with ColumnTransformer()
from sklearn.compose import ColumnTransformer
process = ColumnTransformer(transformers = [('num', num_pipeline, numeric_features),
                                            ('categorical', encoding_pipeline, categorical_features)])
```

In [22]:
```python
### fit
data1 = process.fit(X)
```

In [23]:
```python
clean_data = pd.DataFrame(process.transform(X) , columns = data1.get_feature_names_out())
```

In [24]:
```python
clean_data
```

Out[24]:

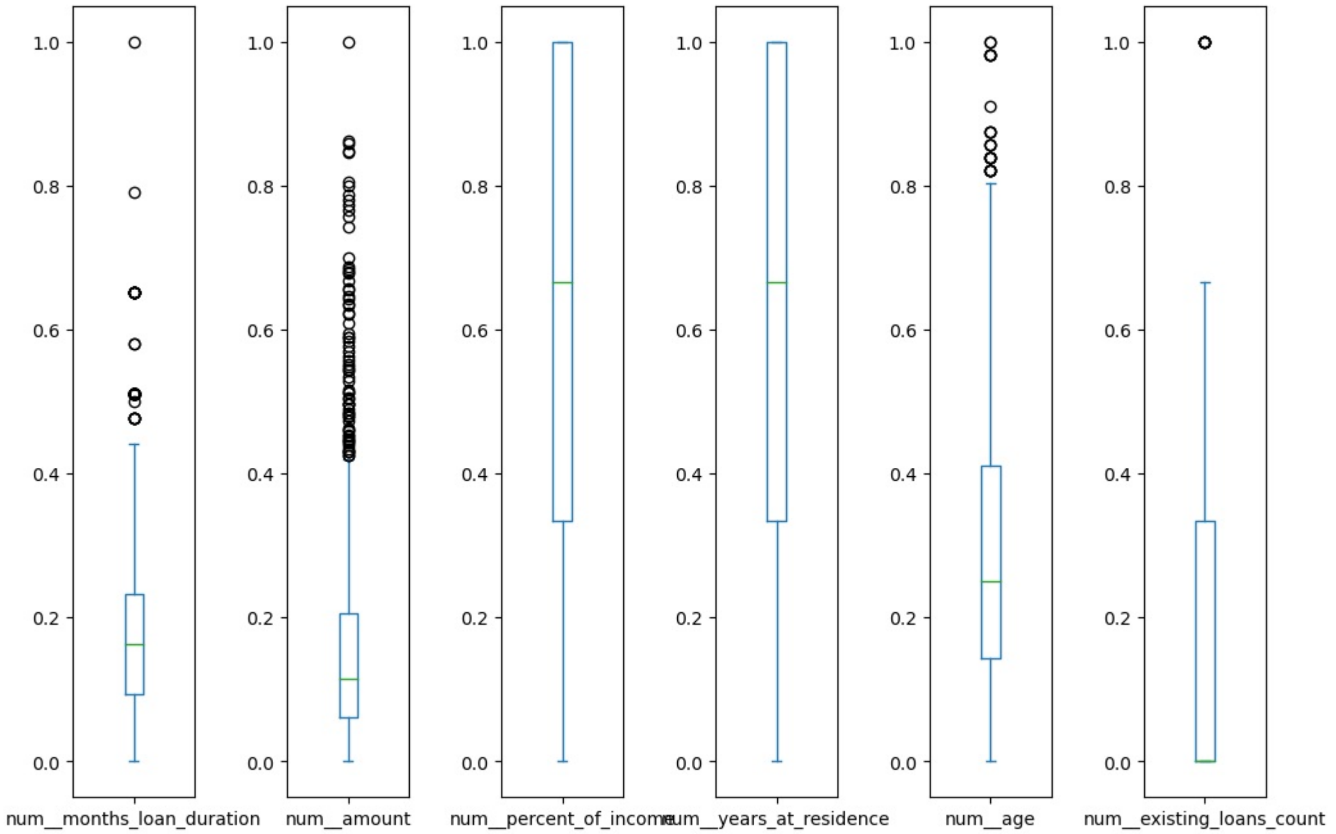
	num__months_loan_duration	num__amount	num__percent_of_income	num__years_at_residence	num__age	num__existing_loans_count
0	0.023256	0.050567	1.000000	1.000000	0.857143	0.333333
1	0.511628	0.313690	0.333333	0.333333	0.053571	0.000000
2	0.093023	0.101574	0.333333	0.666667	0.535714	0.000000
3	0.441860	0.419941	0.333333	1.000000	0.464286	0.000000
4	0.232558	0.254209	0.666667	1.000000	0.607143	0.333333
...	...	...	...	...	...	...
995	0.093023	0.081765	0.666667	1.000000	0.214286	0.000000
996	0.302326	0.198470	1.000000	1.000000	0.375000	0.000000
997	0.093023	0.030483	1.000000	1.000000	0.339286	0.000000
998	0.476744	0.087763	1.000000	1.000000	0.071429	0.000000
999	0.476744	0.238032	0.666667	1.000000	0.142857	0.000000

1000 rows × 34 columns

In [25]:
```python
### check missing values
clean_data.isna().sum()
```

Out[25]:
```
num__months_loan_duration                   0
num__amount                                 0
num__percent_of_income                      0
num__years_at_residence                     0
num__age                                    0
num__existing_loans_count                   0
num__dependents                             0
categorical__checking_balance_< 0 DM        0
categorical__checking_balance_> 200 DM      0
categorical__checking_balance_unknown       0
categorical__credit_history_good            0
categorical__credit_history_perfect         0
categorical__credit_history_poor            0
categorical__credit_history_very good       0
categorical__purpose_car                    0
categorical__purpose_car0                   0
categorical__purpose_education              0
categorical__purpose_furniture/appliances   0
categorical__purpose_renovations            0
categorical__savings_balance_500 - 1000 DM  0
categorical__savings_balance_< 100 DM       0
categorical__savings_balance_> 1000 DM      0
categorical__savings_balance_unknown        0
categorical__employment_duration_4 - 7 years 0
categorical__employment_duration_< 1 year   0
categorical__employment_duration_> 7 years  0
categorical__employment_duration_unemployed 0
categorical__other_credit_none              0
categorical__other_credit_store             0
categorical__housing_own                    0
categorical__housing_rent                   0
categorical__job_skilled                    0
categorical__job_unemployed                 0
categorical__job_unskilled                  0
dtype: int64
```

In [26]:
```
### check outliers
clean_data.iloc[: , 0:6].plot(kind = 'box' , subplots = True , figsize= (12,8))
plt.subplots_adjust(wspace = 0.75) # ws is the width of the padding between subplots, as a fraction of the aver
plt.show()
```



In [27]:
```
#### to Remove outliers use Winsorizer
from feature_engine.outliers import Winsorizer

winsor = Winsorizer(capping_method = 'iqr', # choose  IQR rule boundaries or gaussian for mean and std
                          tail = 'both', # cap left, right or both tails
                          fold = 1.5,
                          variables = ['num__months_loan_duration', 'num__amount',
                                       'num__percent_of_income',
                                       'num__years_at_residence',
                                       'num__age', 'num__existing_loans_count'])
```

In [28]:
```
outliers =winsor.fit(clean_data[['num__months_loan_duration', 'num__amount',
```

```
                    'num__percent_of_income', 'num__years_at_residence',
                    'num__age', 'num__existing_loans_count']])
```

In [29]:
```
clean_data[['num__months_loan_duration', 'num__amount',
           'num__percent_of_income',
           'num__years_at_residence',
           'num__age', 'num__existing_loans_count']] = outliers.transform(clean_data[['num__months_loan_duratio
                                                                                     'num__amount', 'num__percen
                                                                                     'num__years_at_residence',
                                                                                     'num__existing_loans_count'
```

In [30]:
```
clean_data
```

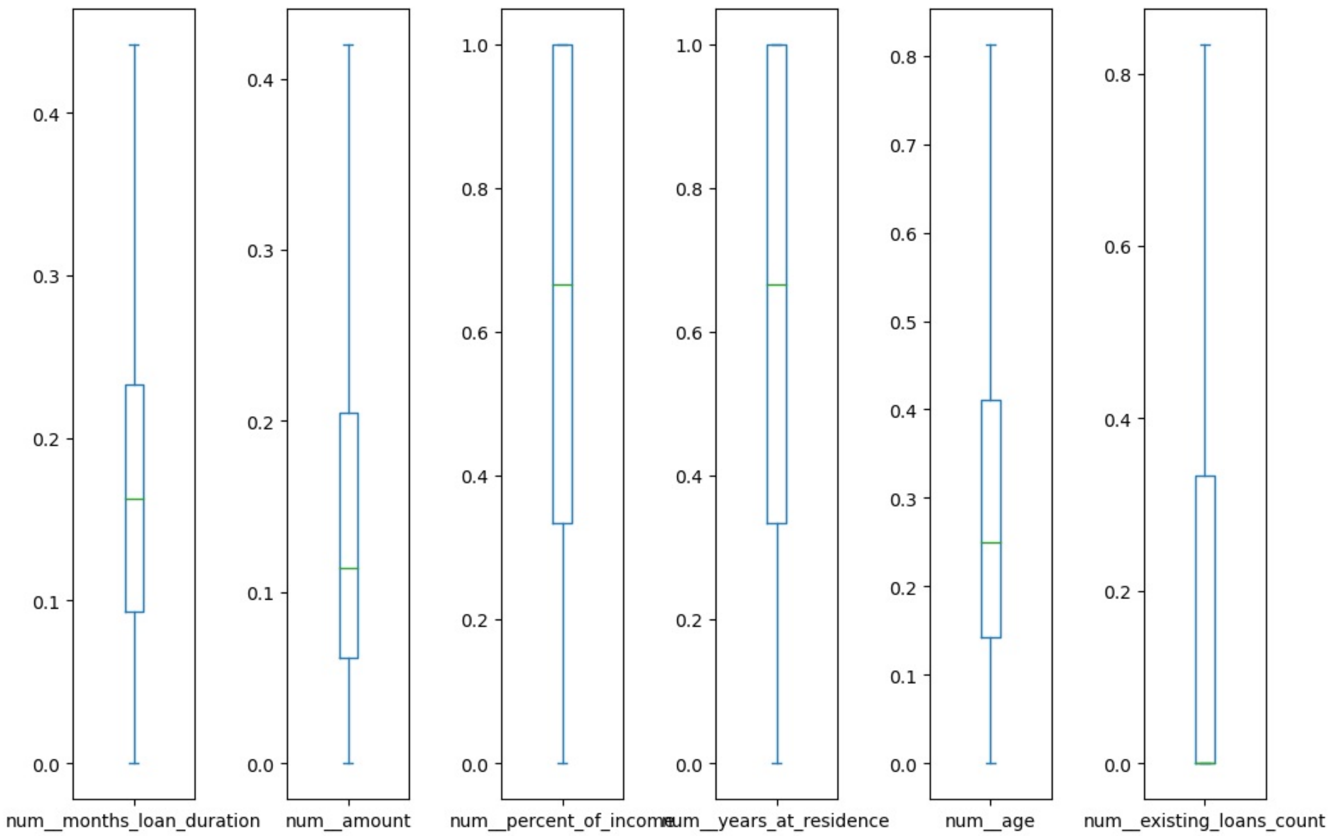
Out[30]:

	num__months_loan_duration	num__amount	num__percent_of_income	num__years_at_residence	num__age	num__existing_loans_count
0	0.023256	0.050567	1.000000	1.000000	0.812500	0.333333
1	0.441860	0.313690	0.333333	0.333333	0.053571	0.000000
2	0.093023	0.101574	0.333333	0.666667	0.535714	0.000000
3	0.441860	0.419941	0.333333	1.000000	0.464286	0.000000
4	0.232558	0.254209	0.666667	1.000000	0.607143	0.333333
...	...	...	...	...	...	...
995	0.093023	0.081765	0.666667	1.000000	0.214286	0.000000
996	0.302326	0.198470	1.000000	1.000000	0.375000	0.000000
997	0.093023	0.030483	1.000000	1.000000	0.339286	0.000000
998	0.441860	0.087763	1.000000	1.000000	0.071429	0.000000
999	0.441860	0.238032	0.666667	1.000000	0.142857	0.000000

1000 rows × 34 columns

In [31]:
```
### again check outliers

clean_data.iloc[: , 0:6].plot(kind = 'box' , subplots = True , figsize= (12,8))
plt.subplots_adjust(wspace = 0.75) # ws is the width of the padding between subplots, as a fraction of the aver
plt.show()
```



In [63]:
```
# Split data into train and test with Stratified Sample technique
# from sklearn.model_selection import train_test_split

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(clean_data, Y,
                                                    test_size = 0.2,
                                                    stratify = Y, random_state = 0)
```

In [64]:
```python
# Proportion of Target variable categories are consistent across train and test
print(Y_train.value_counts()/800)
print("\n")
print(Y_test.value_counts()/200)
```

```
default
no         0.7
yes        0.3
dtype: float64


default
no         0.7
yes        0.3
dtype: float64
```

In [66]:
```python
### Decision Tree Model
from sklearn.tree import DecisionTreeClassifier

model =DecisionTreeClassifier(criterion = 'entropy')
model.fit(X_train , Y_train)
```

Out[66]:
```
▾           DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy')
```

In [35]:
```python
# Prediction on Test Data

pred = model.predict(X_test)
```

In [36]:
```python
print(pred)
```

```
['no' 'yes' 'no' 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no'
 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no'
 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no' 'no' 'yes'
 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'yes' 'no' 'yes' 'no' 'no' 'yes' 'yes' 'no' 'no' 'yes' 'no' 'yes' 'no'
 'yes' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'yes' 'yes' 'yes' 'no'
 'yes' 'yes' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'yes' 'no' 'yes' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no'
 'yes' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes'
 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no' 'yes'
 'yes' 'no' 'yes' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'yes' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no' 'no'
 'yes' 'no' 'no' 'yes' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no']
```

In [37]:
```python
# Accuracy
from sklearn.metrics import accuracy_score , classification_report

print("Accuracy :" , accuracy_score(Y_test , pred))
```

```
Accuracy : 0.715
```

In [38]:
```python
### Classification

classification = classification_report(Y_test , pred)
```

In [39]:
```python
print(classification)
```

```
              precision    recall  f1-score   support

          no       0.78      0.82      0.80       140
         yes       0.53      0.47      0.50        60

    accuracy                           0.71       200
   macro avg       0.66      0.64      0.65       200
weighted avg       0.71      0.71      0.71       200
```

In [40]:
```python
##### ### Hyperparameter Optimization
from sklearn.model_selection import GridSearchCV

param_grid = { 'criterion':['gini', 'entropy'], 'max_depth': np.arange(3, 15)}
```

In [41]:
```python
tree = DecisionTreeClassifier()
```

In [42]:
```python
# GridsearchCV with cross-validation to perform experiments with parameters set
grid = GridSearchCV(tree, param_grid, cv = 5, scoring = 'accuracy',
                          return_train_score = False, verbose = 1)
```

In [53]:
```python
# Train the model with Grid search optimization technique
model =grid.fit(X_train , Y_train)
```

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
```

In [54]:
```python
model.best_params_
```
Out[54]:
```
{'criterion': 'entropy', 'max_depth': 5}
```

In [55]:
```python
preds =model.predict(X_test)
```

In [56]:
```python
print(preds)
```
```
['no' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no'
 'no' 'no' 'no' 'yes' 'no' 'no' 'yes' 'yes' 'yes' 'no' 'no' 'yes' 'yes'
 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'yes' 'no' 'yes' 'no' 'yes' 'yes' 'no' 'no' 'no' 'yes' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes' 'no' 'yes'
 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes' 'no' 'no'
 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no'
 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no'
 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes'
 'no' 'no' 'no' 'no' 'no' 'no']
```

In [48]:
```python
print("Accuracy :" , accuracy_score(Y_test , preds))
```
```
Accuracy : 0.72
```

In [49]:
```python
### Classification

classification = classification_report(Y_test , pred)
print(classification)
```
```
              precision    recall  f1-score   support

          no       0.78      0.82      0.80       140
         yes       0.53      0.47      0.50        60

    accuracy                           0.71       200
   macro avg       0.66      0.64      0.65       200
weighted avg       0.71      0.71      0.71       200
```

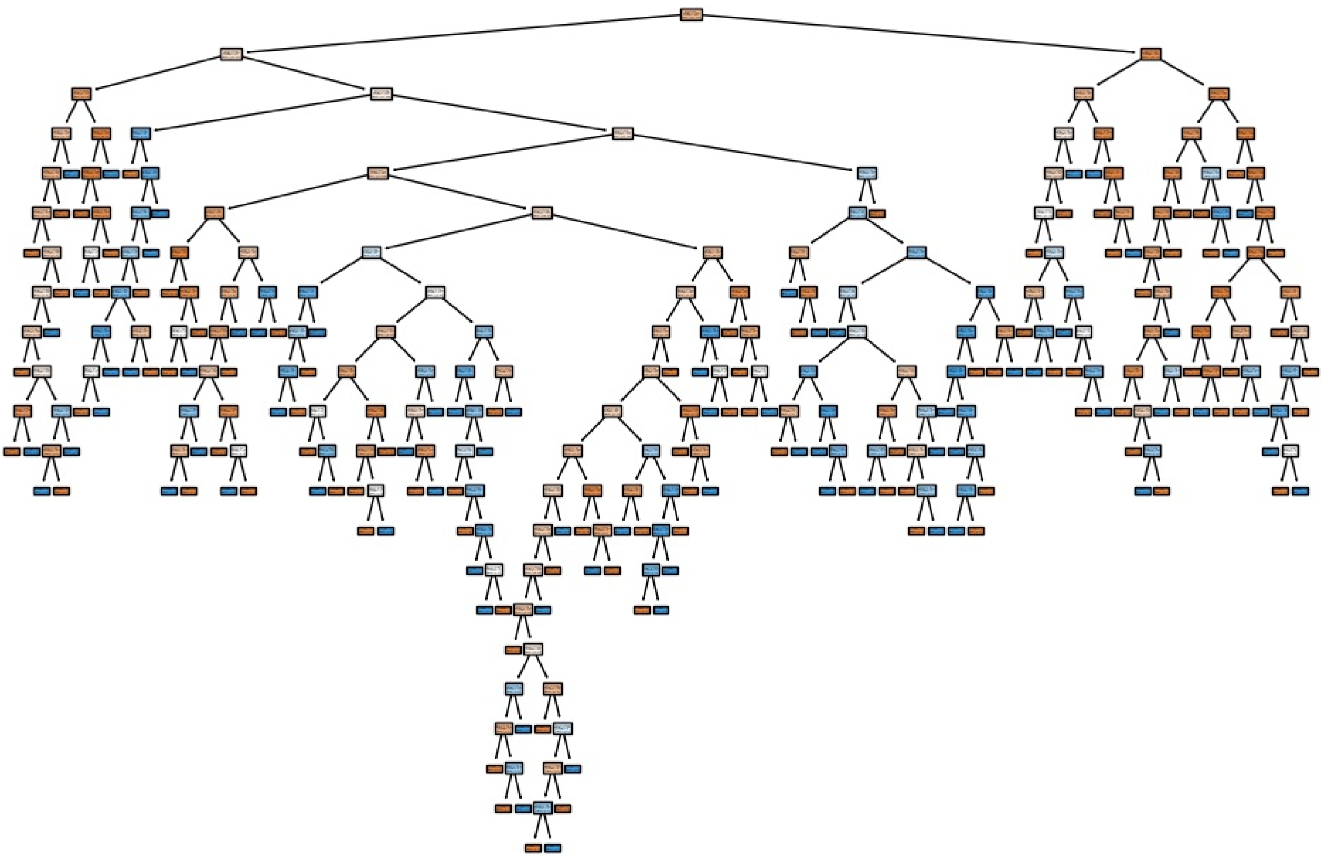
In [68]:
```python
### plot tree
from sklearn import tree
plt.figure(figsize=(12,8))
tree.plot_tree(model, filled=True)
```
Out[68]:
```
[Text(0.5229419309701493, 0.9772727272727273, 'x[9] <= 0.5\nentropy = 0.881\nsamples = 800\nvalue = [560, 240]'
),
 Text(0.17368236940298507, 0.9318181818181818, 'x[0] <= 0.087\nentropy = 0.982\nsamples = 477\nvalue = [276, 20
1]'),
 Text(0.05970149253731343, 0.8863636363636364, 'x[4] <= 0.295\nentropy = 0.727\nsamples = 74\nvalue = [59, 15]'
),
 Text(0.04477611940298507, 0.8409090909090909, 'x[16] <= 0.5\nentropy = 0.935\nsamples = 37\nvalue = [24, 13]')
,
 Text(0.03731343283582089, 0.7954545454545454, 'x[5] <= 0.167\nentropy = 0.845\nsamples = 33\nvalue = [24, 9]')
,
 Text(0.029850746268656716, 0.75, 'x[1] <= 0.02\nentropy = 0.918\nsamples = 27\nvalue = [18, 9]'),
 Text(0.022388059701492536, 0.7045454545454546, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.03731343283582089, 0.7045454545454546, 'x[4] <= 0.241\nentropy = 0.966\nsamples = 23\nvalue = [14, 9]')
,
 Text(0.029850746268656716, 0.6590909090909091, 'x[4] <= 0.188\nentropy = 0.993\nsamples = 20\nvalue = [11, 9]'
),
 Text(0.022388059701492536, 0.6136363636363636, 'x[27] <= 0.5\nentropy = 0.964\nsamples = 18\nvalue = [11, 7]')
,
 Text(0.014925373134328358, 0.5681818181818182, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.029850746268656716, 0.5681818181818182, 'x[0] <= 0.052\nentropy = 0.997\nsamples = 15\nvalue = [8, 7]')
,
 Text(0.014925373134328358, 0.5227272727272727, 'x[1] <= 0.244\nentropy = 0.65\nsamples = 6\nvalue = [5, 1]'),
 Text(0.007462686567164179, 0.4772727272727273, 'entropy = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.022388059701492536, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.04477611940298507, 0.5227272727272727, 'x[4] <= 0.125\nentropy = 0.918\nsamples = 9\nvalue = [3, 6]'),
 Text(0.03731343283582089, 0.4772727272727273, 'x[29] <= 0.5\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
 Text(0.029850746268656716, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.04477611940298507, 0.4318181818181818, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.05223880597014925, 0.4772727272727273, 'entropy = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.03731343283582089, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.04477611940298507, 0.6590909090909091, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.04477611940298507, 0.75, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.05223880597014925, 0.7954545454545454, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.07462686567164178, 0.8409090909090909, 'x[1] <= 0.404\nentropy = 0.303\nsamples = 37\nvalue = [35, 2]')
,
 Text(0.06716417910447761, 0.7954545454545454, 'x[8] <= 0.5\nentropy = 0.183\nsamples = 36\nvalue = [35, 1]'),
 Text(0.05970149253731343, 0.75, 'entropy = 0.0\nsamples = 27\nvalue = [27, 0]'),
 Text(0.07462686567164178, 0.75, 'x[29] <= 0.5\nentropy = 0.503\nsamples = 9\nvalue = [8, 1]'),
 Text(0.06716417910447761, 0.7045454545454546, 'x[5] <= 0.167\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.05970149253731343, 0.6590909090909091, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.07462686567164178, 0.6590909090909091, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
```

```
 Text(0.08208955223880597, 0.7045454545454546, 'entropy = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.08208955223880597, 0.7954545454545454, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2876632462686567, 0.8863636363636364, 'x[1] <= 0.039\nentropy = 0.996\nsamples = 403\nvalue = [217, 186
]'),
 Text(0.1044776119402985, 0.8409090909090909, 'x[1] <= 0.019\nentropy = 0.784\nsamples = 30\nvalue = [7, 23]'),
 Text(0.09701492537313433, 0.7954545454545454, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.11194029850746269, 0.7954545454545454, 'x[14] <= 0.5\nentropy = 0.677\nsamples = 28\nvalue = [5, 23]'),
 Text(0.1044776119402985, 0.75, 'x[4] <= 0.357\nentropy = 0.852\nsamples = 18\nvalue = [5, 13]'),
 Text(0.09701492537313433, 0.7045454545454546, 'x[4] <= 0.33\nentropy = 0.94\nsamples = 14\nvalue = [5, 9]'),
 Text(0.08955223880597014, 0.6590909090909091, 'x[33] <= 0.5\nentropy = 0.811\nsamples = 12\nvalue = [3, 9]'),
 Text(0.07462686567164178, 0.6136363636363636, 'x[10] <= 0.5\nentropy = 0.503\nsamples = 9\nvalue = [1, 8]'),
 Text(0.06716417910447761, 0.5681818181818182, 'x[8] <= 0.5\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.05970149253731343, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.07462686567164178, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.08208955223880597, 0.5681818181818182, 'entropy = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.1044776119402985, 0.6136363636363636, 'x[10] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
 Text(0.09701492537313433, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.11194029850746269, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.1044776119402985, 0.6590909090909091, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.11194029850746269, 0.7045454545454546, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.11940298507462686, 0.75, 'entropy = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(0.4708488805970149, 0.8409090909090909, 'x[0] <= 0.32\nentropy = 0.989\nsamples = 373\nvalue = [210, 163]
'),
 Text(0.2849813432835821, 0.7954545454545454, 'x[0] <= 0.11\nentropy = 0.96\nsamples = 282\nvalue = [174, 108]'
),
 Text(0.16044776119402984, 0.75, 'x[31] <= 0.5\nentropy = 0.784\nsamples = 60\nvalue = [46, 14]'),
 Text(0.13432835820895522, 0.7045454545454546, 'x[2] <= 0.833\nentropy = 0.242\nsamples = 25\nvalue = [24, 1]')
,
 Text(0.12686567164179105, 0.6590909090909091, 'entropy = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(0.1417910447761194, 0.6590909090909091, 'x[7] <= 0.5\nentropy = 0.544\nsamples = 8\nvalue = [7, 1]'),
 Text(0.13432835820895522, 0.6136363636363636, 'x[27] <= 0.5\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.12686567164179105, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.1417910447761194, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.14925373134328357, 0.6136363636363636, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.1865671641791045, 0.7045454545454546, 'x[25] <= 0.5\nentropy = 0.952\nsamples = 35\nvalue = [22, 13]'),
 Text(0.17164179104477612, 0.6590909090909091, 'x[11] <= 0.5\nentropy = 0.85\nsamples = 29\nvalue = [21, 8]'),
 Text(0.16417910447761194, 0.6136363636363636, 'x[2] <= 0.833\nentropy = 0.764\nsamples = 27\nvalue = [21, 6]')
,
 Text(0.15671641791044777, 0.5681818181818182, 'x[1] <= 0.072\nentropy = 0.954\nsamples = 16\nvalue = [10, 6]')
,
 Text(0.1417910447761194, 0.5227272727272727, 'x[1] <= 0.059\nentropy = 0.863\nsamples = 7\nvalue = [2, 5]'),
 Text(0.13432835820895522, 0.4772727272727273, 'x[17] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
 Text(0.12686567164179105, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1417910447761194, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.14925373134328357, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.17164179104477612, 0.5227272727272727, 'x[1] <= 0.165\nentropy = 0.503\nsamples = 9\nvalue = [8, 1]'),
 Text(0.16417910447761194, 0.4772727272727273, 'entropy = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.1791044776119403, 0.4772727272727273, 'x[6] <= 0.5\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.17164179104477612, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1865671641791045, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.17164179104477612, 0.5681818181818182, 'entropy = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(0.1791044776119403, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.20149253731343283, 0.6590909090909091, 'x[2] <= 0.833\nentropy = 0.65\nsamples = 6\nvalue = [1, 5]'),
 Text(0.19402985074626866, 0.6136363636363636, 'entropy = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.208955223880597, 0.6136363636363636, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.40951492537313433, 0.75, 'x[1] <= 0.114\nentropy = 0.983\nsamples = 222\nvalue = [128, 94]'),
 Text(0.2798507462686567, 0.7045454545454546, 'x[27] <= 0.5\nentropy = 0.989\nsamples = 91\nvalue = [40, 51]'),
 Text(0.23134328358208955, 0.6590909090909091, 'x[0] <= 0.174\nentropy = 0.702\nsamples = 21\nvalue = [4, 17]')
,
 Text(0.22388059701492538, 0.6136363636363636, 'x[22] <= 0.5\nentropy = 0.946\nsamples = 11\nvalue = [4, 7]'),
 Text(0.21641791044776118, 0.5681818181818182, 'x[24] <= 0.5\nentropy = 0.764\nsamples = 9\nvalue = [2, 7]'),
 Text(0.208955223880597, 0.5227272727272727, 'entropy = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.22388059701492538, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.23134328358208955, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.23880597014925373, 0.6136363636363636, 'entropy = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(0.3283582089552239, 0.6590909090909091, 'x[14] <= 0.5\nentropy = 0.999\nsamples = 70\nvalue = [36, 34]'),
 Text(0.291044776119403, 0.6136363636363636, 'x[1] <= 0.092\nentropy = 0.946\nsamples = 44\nvalue = [28, 16]'),
 Text(0.26119402985074625, 0.5681818181818182, 'x[1] <= 0.057\nentropy = 0.811\nsamples = 32\nvalue = [24, 8]')
,
 Text(0.23880597014925373, 0.5227272727272727, 'x[1] <= 0.048\nentropy = 1.0\nsamples = 12\nvalue = [6, 6]'),
 Text(0.23134328358208955, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.2462686567164179, 0.4772727272727273, 'x[4] <= 0.589\nentropy = 0.811\nsamples = 8\nvalue = [2, 6]'),
 Text(0.23880597014925373, 0.4318181818181818, 'entropy = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.2537313432835821, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.2835820895522388, 0.5227272727272727, 'x[4] <= 0.205\nentropy = 0.469\nsamples = 20\nvalue = [18, 2]'),
 Text(0.27611940298507465, 0.4772727272727273, 'x[1] <= 0.075\nentropy = 0.764\nsamples = 9\nvalue = [7, 2]'),
 Text(0.26865671641791045, 0.4318181818181818, 'entropy = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.2835820895522388, 0.4318181818181818, 'x[3] <= 0.167\nentropy = 1.0\nsamples = 4\nvalue = [2, 2]'),
 Text(0.27611940298507465, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.291044776119403, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.291044776119403, 0.4772727272727273, 'entropy = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(0.3208955223880597, 0.5681818181818182, 'x[24] <= 0.5\nentropy = 0.918\nsamples = 12\nvalue = [4, 8]'),
 Text(0.31343283582089554, 0.5227272727272727, 'x[10] <= 0.5\nentropy = 0.985\nsamples = 7\nvalue = [4, 3]'),
 Text(0.30597014925373134, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.3208955223880597, 0.4772727272727273, 'x[6] <= 0.5\nentropy = 0.722\nsamples = 5\nvalue = [4, 1]'),
 Text(0.31343283582089554, 0.4318181818181818, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.3283582089552239, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3283582089552239, 0.5227272727272727, 'entropy = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.3656716417910448, 0.6136363636363636, 'x[1] <= 0.089\nentropy = 0.89\nsamples = 26\nvalue = [8, 18]'),
```

```
 Text(0.35074626865671643, 0.5681818181818182, 'x[4] <= 0.116\nentropy = 0.722\nsamples = 20\nvalue = [4, 16]')
,
 Text(0.34328358208955223, 0.5227272727272727, 'entropy = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.3582089552238806, 0.5227272727272727, 'x[22] <= 0.5\nentropy = 0.89\nsamples = 13\nvalue = [4, 9]'),
 Text(0.35074626865671643, 0.4772727272727273, 'x[5] <= 0.167\nentropy = 0.991\nsamples = 9\nvalue = [4, 5]'),
 Text(0.34328358208955223, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3582089552238806, 0.4318181818181818, 'x[1] <= 0.051\nentropy = 0.863\nsamples = 7\nvalue = [2, 5]'),
 Text(0.35074626865671643, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3656716417910448, 0.38636363636363635, 'x[25] <= 0.5\nentropy = 0.65\nsamples = 6\nvalue = [1, 5]'),
 Text(0.3582089552238806, 0.3409090909090909, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.373134328358209, 0.3409090909090909, 'x[1] <= 0.063\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.3656716417910448, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3805970149253731, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3656716417910448, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.3805970149253731, 0.5681818181818182, 'x[1] <= 0.111\nentropy = 0.918\nsamples = 6\nvalue = [4, 2]'),
 Text(0.373134328358209, 0.5227272727272727, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.3880597014925373, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.539179104477612, 0.7045454545454546, 'x[22] <= 0.5\nentropy = 0.913\nsamples = 131\nvalue = [88, 43]'),
 Text(0.5186567164179104, 0.6590909090909091, 'x[1] <= 0.417\nentropy = 0.958\nsamples = 108\nvalue = [67, 41]'
),
 Text(0.5, 0.6136363636363636, 'x[21] <= 0.5\nentropy = 0.918\nsamples = 99\nvalue = [66, 33]'),
 Text(0.4925373134328358, 0.5681818181818182, 'x[23] <= 0.5\nentropy = 0.938\nsamples = 93\nvalue = [60, 33]'),
 Text(0.4626865671641791, 0.5227272727272727, 'x[30] <= 0.5\nentropy = 0.965\nsamples = 82\nvalue = [50, 32]'),
 Text(0.43283582089552236, 0.4772727272727273, 'x[25] <= 0.5\nentropy = 0.908\nsamples = 65\nvalue = [44, 21]')
,
 Text(0.417910447761194, 0.4318181818181818, 'x[1] <= 0.357\nentropy = 0.971\nsamples = 50\nvalue = [30, 20]'),
 Text(0.41044776119402987, 0.38636363636363635, 'x[1] <= 0.29\nentropy = 0.954\nsamples = 48\nvalue = [30, 18]'
),
 Text(0.40298507462686567, 0.3409090909090909, 'x[1] <= 0.237\nentropy = 0.981\nsamples = 43\nvalue = [25, 18]'
),
 Text(0.39552238805970147, 0.29545454545454547, 'x[27] <= 0.5\nentropy = 0.942\nsamples = 39\nvalue = [25, 14]'
),
 Text(0.3880597014925373, 0.25, 'entropy = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.40298507462686567, 0.25, 'x[1] <= 0.174\nentropy = 0.997\nsamples = 30\nvalue = [16, 14]'),
 Text(0.3880597014925373, 0.20454545454545456, 'x[7] <= 0.5\nentropy = 0.94\nsamples = 14\nvalue = [5, 9]'),
 Text(0.3805970149253731, 0.1590909090909091, 'x[17] <= 0.5\nentropy = 0.863\nsamples = 7\nvalue = [5, 2]'),
 Text(0.373134328358209, 0.11363636363636363, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.3880597014925373, 0.11363636363636363, 'x[20] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.3805970149253731, 0.06818181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.39552238805970147, 0.06818181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.39552238805970147, 0.1590909090909091, 'entropy = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.417910447761194, 0.20454545454545456, 'x[1] <= 0.201\nentropy = 0.896\nsamples = 16\nvalue = [11, 5]'),
 Text(0.41044776119402987, 0.1590909090909091, 'entropy = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.4253731343283582, 0.1590909090909091, 'x[4] <= 0.295\nentropy = 0.991\nsamples = 9\nvalue = [4, 5]'),
 Text(0.417910447761194, 0.11363636363636363, 'x[3] <= 0.667\nentropy = 0.918\nsamples = 6\nvalue = [4, 2]'),
 Text(0.41044776119402987, 0.06818181818181818, 'x[14] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.40298507462686567, 0.022727272727272728, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.417910447761194, 0.022727272727272728, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.4253731343283582, 0.06818181818181818, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.43283582089552236, 0.11363636363636363, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.41044776119402987, 0.29545454545454547, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.417910447761194, 0.3409090909090909, 'entropy = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.4253731343283582, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.44776119402985076, 0.4318181818181818, 'x[12] <= 0.5\nentropy = 0.353\nsamples = 15\nvalue = [14, 1]'),
 Text(0.44029850746268656, 0.38636363636363635, 'entropy = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(0.4552238805970149, 0.38636363636363635, 'x[5] <= 0.167\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
 Text(0.44776119402985076, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4626865671641791, 0.3409090909090909, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.4925373134328358, 0.4772727272727273, 'x[2] <= 0.167\nentropy = 0.937\nsamples = 17\nvalue = [6, 11]'),
 Text(0.47761194029850745, 0.4318181818181818, 'x[10] <= 0.5\nentropy = 0.722\nsamples = 5\nvalue = [4, 1]'),
 Text(0.4701492537313433, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.48507462686567165, 0.38636363636363635, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.5074626865671642, 0.4318181818181818, 'x[4] <= 0.777\nentropy = 0.65\nsamples = 12\nvalue = [2, 10]'),
 Text(0.5, 0.38636363636363635, 'x[3] <= 0.667\nentropy = 0.439\nsamples = 11\nvalue = [1, 10]'),
 Text(0.4925373134328358, 0.3409090909090909, 'x[4] <= 0.152\nentropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
 Text(0.48507462686567165, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5, 0.29545454545454547, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.5074626865671642, 0.3409090909090909, 'entropy = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.5149253731343284, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5223880597014925, 0.5227272727272727, 'x[1] <= 0.219\nentropy = 0.439\nsamples = 11\nvalue = [10, 1]'),
 Text(0.5149253731343284, 0.4772727272727273, 'entropy = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.5298507462686567, 0.4772727272727273, 'x[7] <= 0.5\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
 Text(0.5223880597014925, 0.4318181818181818, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.5373134328358209, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5074626865671642, 0.5681818181818182, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.5373134328358209, 0.6136363636363636, 'x[24] <= 0.5\nentropy = 0.503\nsamples = 9\nvalue = [1, 8]'),
 Text(0.5298507462686567, 0.5681818181818182, 'entropy = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.5447761194029851, 0.5681818181818182, 'x[3] <= 0.333\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.5373134328358209, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5522388059701493, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5597014925373134, 0.6590909090909091, 'x[7] <= 0.5\nentropy = 0.426\nsamples = 23\nvalue = [21, 2]'),
 Text(0.5522388059701493, 0.6136363636363636, 'entropy = 0.0\nsamples = 15\nvalue = [15, 0]'),
 Text(0.5671641791044776, 0.6136363636363636, 'x[3] <= 0.833\nentropy = 0.811\nsamples = 8\nvalue = [6, 2]'),
 Text(0.5597014925373134, 0.5681818181818182, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.5746268656716418, 0.5681818181818182, 'x[1] <= 0.15\nentropy = 1.0\nsamples = 4\nvalue = [2, 2]'),
 Text(0.5671641791044776, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.582089552238806, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6567164179104478, 0.7954545454545454, 'x[26] <= 0.5\nentropy = 0.968\nsamples = 91\nvalue = [36, 55]'),
 Text(0.6492537313432836, 0.75, 'x[3] <= 0.167\nentropy = 0.93\nsamples = 84\nvalue = [29, 55]'),
```

```
 Text(0.6044776119402985, 0.7045454545454546, 'x[4] <= 0.143\nentropy = 0.845\nsamples = 11\nvalue = [8, 3]'),
 Text(0.5970149253731343, 0.6590909090909091, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6119402985074627, 0.6590909090909091, 'x[30] <= 0.5\nentropy = 0.503\nsamples = 9\nvalue = [8, 1]'),
 Text(0.6044776119402985, 0.6136363636363636, 'entropy = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.6194029850746269, 0.6136363636363636, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6940298507462687, 0.7045454545454546, 'x[0] <= 0.424\nentropy = 0.866\nsamples = 73\nvalue = [21, 52]')
,
 Text(0.6417910447761194, 0.6590909090909091, 'x[1] <= 0.111\nentropy = 0.98\nsamples = 36\nvalue = [15, 21]'),
 Text(0.6343283582089553, 0.6136363636363636, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.6492537313432836, 0.6136363636363636, 'x[4] <= 0.188\nentropy = 0.997\nsamples = 32\nvalue = [15, 17]')
,
 Text(0.6119402985074627, 0.5681818181818182, 'x[1] <= 0.21\nentropy = 0.779\nsamples = 13\nvalue = [3, 10]'),
 Text(0.5970149253731343, 0.5227272727272727, 'x[4] <= 0.116\nentropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
 Text(0.5895522388059702, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.6044776119402985, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6268656716417911, 0.5227272727272727, 'x[4] <= 0.143\nentropy = 0.469\nsamples = 10\nvalue = [1, 9]'),
 Text(0.6194029850746269, 0.4772727272727273, 'entropy = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.6343283582089553, 0.4772727272727273, 'x[5] <= 0.167\nentropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
 Text(0.6268656716417911, 0.4318181818181818, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.6417910447761194, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6865671641791045, 0.5681818181818182, 'x[4] <= 0.375\nentropy = 0.949\nsamples = 19\nvalue = [12, 7]'),
 Text(0.6716417910447762, 0.5227272727272727, 'x[27] <= 0.5\nentropy = 0.684\nsamples = 11\nvalue = [9, 2]'),
 Text(0.664179104477612, 0.4772727272727273, 'x[4] <= 0.286\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.6567164179104478, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6716417910447762, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6791044776119403, 0.4772727272727273, 'entropy = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.7014925373134329, 0.5227272727272727, 'x[1] <= 0.408\nentropy = 0.954\nsamples = 8\nvalue = [3, 5]'),
 Text(0.6940298507462687, 0.4772727272727273, 'x[29] <= 0.5\nentropy = 0.971\nsamples = 5\nvalue = [3, 2]'),
 Text(0.6865671641791045, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.7014925373134329, 0.4318181818181818, 'x[25] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.6940298507462687, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7089552238805971, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.7089552238805971, 0.4772727272727273, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.746268656716418, 0.6590909090909091, 'x[22] <= 0.5\nentropy = 0.639\nsamples = 37\nvalue = [6, 31]'),
 Text(0.7313432835820896, 0.6136363636363636, 'x[8] <= 0.5\nentropy = 0.439\nsamples = 33\nvalue = [3, 30]'),
 Text(0.7238805970149254, 0.5681818181818182, 'x[23] <= 0.5\nentropy = 0.337\nsamples = 32\nvalue = [2, 30]'),
 Text(0.7164179104477612, 0.5227272727272727, 'entropy = 0.0\nsamples = 21\nvalue = [0, 21]'),
 Text(0.7313432835820896, 0.5227272727272727, 'x[4] <= 0.179\nentropy = 0.684\nsamples = 11\nvalue = [2, 9]'),
 Text(0.7238805970149254, 0.4772727272727273, 'entropy = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.7388059701492538, 0.4772727272727273, 'x[29] <= 0.5\nentropy = 0.918\nsamples = 6\nvalue = [2, 4]'),
 Text(0.7313432835820896, 0.4318181818181818, 'x[17] <= 0.5\nentropy = 0.722\nsamples = 5\nvalue = [1, 4]'),
 Text(0.7238805970149254, 0.38636363636363635, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.7388059701492538, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.746268656716418, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7388059701492538, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7611940298507462, 0.6136363636363636, 'x[7] <= 0.5\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
 Text(0.753731343283582, 0.5681818181818182, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.7686567164179104, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.664179104477612, 0.75, 'entropy = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.8722014925373134, 0.9318181818181818, 'x[27] <= 0.5\nentropy = 0.531\nsamples = 323\nvalue = [284, 39]'
),
 Text(0.8208955223880597, 0.8863636363636364, 'x[17] <= 0.5\nentropy = 0.892\nsamples = 55\nvalue = [38, 17]'),
 Text(0.8059701492537313, 0.8409090909090909, 'x[26] <= 0.5\nentropy = 0.994\nsamples = 33\nvalue = [18, 15]'),
 Text(0.7985074626865671, 0.7954545454545454, 'x[4] <= 0.455\nentropy = 0.971\nsamples = 30\nvalue = [18, 12]')
,
 Text(0.7910447761194029, 0.75, 'x[3] <= 0.167\nentropy = 1.0\nsamples = 24\nvalue = [12, 12]'),
 Text(0.7835820895522388, 0.7045454545454546, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.7985074626865671, 0.7045454545454546, 'x[31] <= 0.5\nentropy = 0.985\nsamples = 21\nvalue = [9, 12]'),
 Text(0.7835820895522388, 0.6590909090909091, 'x[6] <= 0.5\nentropy = 0.918\nsamples = 9\nvalue = [6, 3]'),
 Text(0.7761194029850746, 0.6136363636363636, 'entropy = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.7910447761194029, 0.6136363636363636, 'x[23] <= 0.5\nentropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
 Text(0.7835820895522388, 0.5681818181818182, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.7985074626865671, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8134328358208955, 0.6590909090909091, 'x[1] <= 0.292\nentropy = 0.811\nsamples = 12\nvalue = [3, 9]'),
 Text(0.8059701492537313, 0.6136363636363636, 'entropy = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.8208955223880597, 0.6136363636363636, 'x[1] <= 0.418\nentropy = 1.0\nsamples = 6\nvalue = [3, 3]'),
 Text(0.8134328358208955, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8283582089552238, 0.5681818181818182, 'x[2] <= 0.167\nentropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
 Text(0.8208955223880597, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.835820895522388, 0.5227272727272727, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8059701492537313, 0.75, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.8134328358208955, 0.7954545454545454, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.835820895522388, 0.8409090909090909, 'x[4] <= 0.125\nentropy = 0.439\nsamples = 22\nvalue = [20, 2]'),
 Text(0.8283582089552238, 0.7954545454545454, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8432835820895522, 0.7954545454545454, 'x[19] <= 0.5\nentropy = 0.276\nsamples = 21\nvalue = [20, 1]'),
 Text(0.835820895522388, 0.75, 'entropy = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(0.8507462686567164, 0.75, 'x[1] <= 0.116\nentropy = 0.722\nsamples = 5\nvalue = [4, 1]'),
 Text(0.8432835820895522, 0.7045454545454546, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.8582089552238806, 0.7045454545454546, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9235074626865671, 0.8863636363636364, 'x[4] <= 0.17\nentropy = 0.409\nsamples = 268\nvalue = [246, 22]'
),
 Text(0.9029850746268657, 0.8409090909090909, 'x[0] <= 0.355\nentropy = 0.665\nsamples = 75\nvalue = [62, 13]')
,
 Text(0.8880597014925373, 0.7954545454545454, 'x[5] <= 0.167\nentropy = 0.493\nsamples = 65\nvalue = [58, 7]'),
 Text(0.8805970149253731, 0.75, 'x[1] <= 0.129\nentropy = 0.641\nsamples = 43\nvalue = [36, 7]'),
 Text(0.8731343283582089, 0.7045454545454546, 'x[0] <= 0.064\nentropy = 0.758\nsamples = 32\nvalue = [25, 7]'),
 Text(0.8656716417910447, 0.6590909090909091, 'entropy = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.8805970149253731, 0.6590909090909091, 'x[12] <= 0.5\nentropy = 0.887\nsamples = 23\nvalue = [16, 7]'),
 Text(0.8731343283582089, 0.6136363636363636, 'x[14] <= 0.5\nentropy = 0.792\nsamples = 21\nvalue = [16, 5]'),
```

```
 Text(0.8582089552238806, 0.5681818181818182, 'x[3] <= 0.833\nentropy = 0.544\nsamples = 16\nvalue = [14, 2]'),
 Text(0.8507462686567164, 0.5227272727272727, 'entropy = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(0.8656716417910447, 0.5227272727272727, 'x[20] <= 0.5\nentropy = 0.971\nsamples = 5\nvalue = [3, 2]'),
 Text(0.8582089552238806, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8731343283582089, 0.4772727272727273, 'x[0] <= 0.18\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.8656716417910447, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8805970149253731, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8880597014925373, 0.5681818181818182, 'x[24] <= 0.5\nentropy = 0.971\nsamples = 5\nvalue = [2, 3]'),
 Text(0.8805970149253731, 0.5227272727272727, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8955223880597015, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8880597014925373, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8880597014925373, 0.7045454545454546, 'entropy = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(0.8955223880597015, 0.75, 'entropy = 0.0\nsamples = 22\nvalue = [22, 0]'),
 Text(0.917910447761194, 0.7954545454545454, 'x[1] <= 0.15\nentropy = 0.971\nsamples = 10\nvalue = [4, 6]'),
 Text(0.9104477611940298, 0.75, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.9253731343283582, 0.75, 'x[4] <= 0.054\nentropy = 0.592\nsamples = 7\nvalue = [1, 6]'),
 Text(0.917910447761194, 0.7045454545454546, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9328358208955224, 0.7045454545454546, 'entropy = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.9440298507462687, 0.8409090909090909, 'x[1] <= 0.071\nentropy = 0.272\nsamples = 193\nvalue = [184, 9]'
),
 Text(0.9365671641791045, 0.7954545454545454, 'entropy = 0.0\nsamples = 55\nvalue = [55, 0]'),
 Text(0.9514925373134329, 0.7954545454545454, 'x[1] <= 0.071\nentropy = 0.348\nsamples = 138\nvalue = [129, 9]'
),
 Text(0.9440298507462687, 0.75, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9589552238805971, 0.75, 'x[0] <= 0.25\nentropy = 0.321\nsamples = 137\nvalue = [129, 8]'),
 Text(0.9514925373134329, 0.7045454545454546, 'x[1] <= 0.184\nentropy = 0.386\nsamples = 106\nvalue = [98, 8]')
,
 Text(0.9253731343283582, 0.6590909090909091, 'x[12] <= 0.5\nentropy = 0.233\nsamples = 79\nvalue = [76, 3]'),
 Text(0.9104477611940298, 0.6136363636363636, 'x[16] <= 0.5\nentropy = 0.106\nsamples = 72\nvalue = [71, 1]'),
 Text(0.9029850746268657, 0.5681818181818182, 'entropy = 0.0\nsamples = 64\nvalue = [64, 0]'),
 Text(0.917910447761194, 0.5681818181818182, 'x[4] <= 0.223\nentropy = 0.544\nsamples = 8\nvalue = [7, 1]'),
 Text(0.9104477611940298, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9253731343283582, 0.5227272727272727, 'entropy = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.9402985074626866, 0.6136363636363636, 'x[4] <= 0.384\nentropy = 0.863\nsamples = 7\nvalue = [5, 2]'),
 Text(0.9328358208955224, 0.5681818181818182, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.9477611940298507, 0.5681818181818182, 'x[6] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.9402985074626866, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9552238805970149, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9776119402985075, 0.6590909090909091, 'x[10] <= 0.5\nentropy = 0.691\nsamples = 27\nvalue = [22, 5]'),
 Text(0.9701492537313433, 0.6136363636363636, 'entropy = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(0.9850746268656716, 0.6136363636363636, 'x[14] <= 0.5\nentropy = 0.961\nsamples = 13\nvalue = [8, 5]'),
 Text(0.9776119402985075, 0.5681818181818182, 'x[33] <= 0.5\nentropy = 0.863\nsamples = 7\nvalue = [2, 5]'),
 Text(0.9701492537313433, 0.5227272727272727, 'x[20] <= 0.5\nentropy = 0.65\nsamples = 6\nvalue = [1, 5]'),
 Text(0.9626865671641791, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.9776119402985075, 0.4772727272727273, 'x[3] <= 0.667\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.9701492537313433, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9850746268656716, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9850746268656716, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9925373134328358, 0.5681818181818182, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.9664179104477612, 0.7045454545454546, 'entropy = 0.0\nsamples = 31\nvalue = [31, 0]')]
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js