Brain Stroke Prediction Using Machine Learning and Data Science.

Importing Necessary Libraries.

In [250]:

```
#pip install autoviz
#pip install pandas
#pip install matplotlib.pyplot
#pip install seaborn
#pip install numpy
#pip install sklearn
#pip install collections
#pip install ipywidgets
#pip install imblearn
#pip install statsmodels
#pip install warnings
```

In [251]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('dark_background')
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,classification_report, accuracy_score, precision_recall_curv
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from imblearn.over_sampling import RandomOverSampler,SMOTE
from imblearn.combine import SMOTETomek
from collections import Counter
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
import warnings
warnings.filterwarnings(action='ignore')
from sklearn import tree
import autoviz
from autoviz.AutoViz_Class import AutoViz_Class
#for interactive console
import ipywidgets
import ipywidgets as widgets
from ipywidgets import interact
from ipywidgets import interact_manual
```

Importing and Skimming the Data Set.

The Data set consists of 40000+ entries of Patients Regarding Brain Stroke symptoms. There are total of 12 columns including target_column.

1. id
2. gender
3. age
4. hypertension
5. heart_disease
6. ever_married
7. work_type
8. Residence_type
9. avg_glucose_level
10. bmi
11. smoking_status
12. stroke(target_column)

In [252]:

```
dodge = pd.read_csv('train_strokes.csv')
```

In [253]:

```
# head() helps us to view the first 5 entries in our dataset.

dodge.head()
```

Out[253]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	strok
0	30669	Male	3.0	0	0	No	children	Rural	95.12	18.0	NaN	
1	30468	Male	58.0	1	0	Yes	Private	Urban	87.96	39.2	never smoked	
2	16523	Female	8.0	0	0	No	Private	Urban	110.89	17.6	NaN	
3	56543	Female	70.0	0	0	Yes	Private	Rural	69.04	35.9	formerly smoked	
4	46136	Male	14.0	0	0	No	Never_worked	Rural	161.28	19.1	NaN	

In [254]:

```
# info() gives us the count and dtype, also helps us to identify whether there are any null values or no

dodge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43400 entries, 0 to 43399
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 43400 non-null  int64  
 1   gender             43400 non-null  object 
 2   age                43400 non-null  float64
 3   hypertension       43400 non-null  int64  
 4   heart_disease      43400 non-null  int64  
 5   ever_married       43400 non-null  object 
 6   work_type          43400 non-null  object 
 7   Residence_type     43400 non-null  object 
 8   avg_glucose_level  43400 non-null  float64
 9   bmi                41938 non-null  float64
 10  smoking_status     30108 non-null  object 
 11  stroke             43400 non-null  int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 4.0+ MB
```

In [255]:

```
# describe() gives us a breif description about the columns(count, min, max, mean, median etc)

dodge.describe()
```

Out[255]:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	43400.000000	43400.000000	43400.000000	43400.000000	43400.000000	41938.000000	43400.000000
mean	36326.142350	42.217894	0.093571	0.047512	104.482750	28.605038	0.018041
std	21072.134879	22.519649	0.291235	0.212733	43.111751	7.770020	0.133103
min	1.000000	0.080000	0.000000	0.000000	55.000000	10.100000	0.000000
25%	18038.500000	24.000000	0.000000	0.000000	77.540000	23.200000	0.000000
50%	36351.500000	44.000000	0.000000	0.000000	91.580000	27.700000	0.000000
75%	54514.250000	60.000000	0.000000	0.000000	112.070000	32.900000	0.000000
max	72943.000000	82.000000	1.000000	1.000000	291.050000	97.600000	1.000000

In [256]:

```
# In the case of object columns we get(count, unique values, top, freq)
dodge.describe(include = 'object')
```

Out[256]:

	gender	ever_married	work_type	Residence_type	smoking_status
count	43400	43400	43400	43400	30108
unique	3	2	5	2	3
top	Female	Yes	Private	Urban	never smoked
freq	25665	27938	24834	21756	16053

Exploring Target Variable.

In [257]:

```
dodge['stroke'].value_counts()
```

Out[257]:

```
0    42617
1      783
Name: stroke, dtype: int64
```

In [258]:

```
# There arent any null values, but
dodge['stroke'].isnull().sum()
```

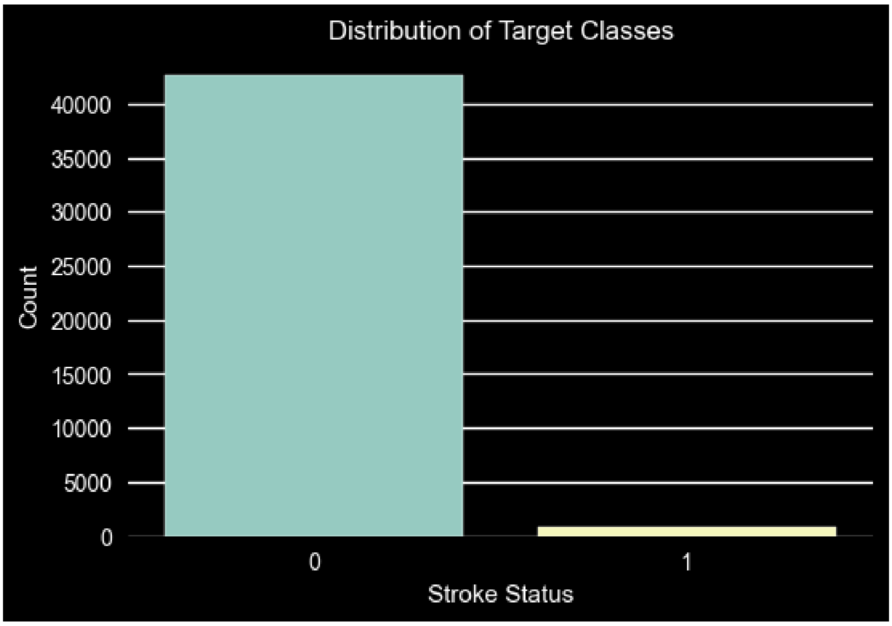
Out[258]:

```
0
```

In [259]:

```
# This plot tell's about, how the distribution of target class is spreaded.
# we can see that the target classes are highly imbalanced with 0->42617, 1->783, so we need to balance
# countplot() helps us to visualize the count the classes.

plt.figure(figsize = (6,4), dpi = 100)
sns.countplot(dodge['stroke'])
plt.xlabel('Stroke Status')
plt.ylabel('Count')
plt.title('Distribution of Target Classes')
plt.show()
```



Exploring Independent Numerical Columns.

1. Cleaning
2. Treating Missing values
3. Anamoly Detection and Reduction

In [260]:

```
numerical = ['age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'bmi']
#dodge[numerical[0]]
```

Treating missing values present in the column dodge['bmi'], no other numerical columns has missing values.

In [261]:

```
dodge['bmi'].isnull().sum()
```

Out[261]:

```
1462
```

In [262]:

```
dodge['bmi'] = dodge['bmi'].fillna(dodge['bmi'].mean())
```

In [263]:

```
dodge['bmi'].isnull().sum()
```

Out[263]:

```
0
```

Exploring each numerical column using describe()

In [264]:

```
for i in numerical:
  print(dodge[i].describe())
```

```
count    43400.000000
mean        42.217894
std         22.519649
min          0.080000
25%         24.000000
50%         44.000000
75%         60.000000
max         82.000000
Name: age, dtype: float64
count    43400.000000
mean         0.093571
std          0.291235
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
Name: hypertension, dtype: float64
count    43400.000000
mean         0.047512
std          0.212733
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
Name: heart_disease, dtype: float64
count    43400.000000
mean       104.482750
std         43.111751
min         55.000000
25%         77.540000
50%         91.580000
75%        112.070000
max        291.050000
Name: avg_glucose_level, dtype: float64
count    43400.000000
mean        28.605038
std          7.638023
min         10.100000
25%         23.400000
50%         28.100000
75%         32.600000
max         97.600000
Name: bmi, dtype: float64
```

In [ ]:

Anamoly Detection and Reduction in Numericals.

1. age

In [265]:

```
dodge['age'].describe()
```

Out[265]:

```
count    43400.000000
mean        42.217894
std         22.519649
min          0.080000
25%         24.000000
50%         44.000000
75%         60.000000
max         82.000000
Name: age, dtype: float64
```

In [266]:

```
dodge['age'].value_counts()
```

Out[266]:

```
51.00    738
52.00    721
53.00    701
78.00    698
50.00    694
        ...
0.48      37
0.40      35
1.00      34
0.16      26
0.08      17
Name: age, Length: 104, dtype: int64
```

Function to check the Anamolies in the column using upper_limit and lower_limit.

1. If the upper_limit > max(df['col']), then we replace the upper_limit with the max value.
2. Similarly, if the lower_limit < min(df['col']), we replace the lower_limit with the min value.

In [267]:

```
anamolies = []
def outliers(data):
  random_state_mean = np.mean(data)
  random_state_std = np.std(data)
  anamoly = random_state_std * 3

  upper_limit = random_state_mean + anamoly
  lower_limit = random_state_mean - anamoly
  lp_lower_limit = 1.00
  up_upper_limit = max(dodge['age'])
  print(upper_limit)
  print(lower_limit)

  print(lp_lower_limit)
  print(up_upper_limit)

  for i in data:
      if i < lp_lower_limit or i > up_upper_limit:
        anamolies.append(i)
```

In [268]:

```
outliers(dodge['age'])
print(len(anamolies))
```

```
109.7760617173718
-25.340273698938617
1.0
82.0
496
```

In [269]:

```
dodge.shape
```

Out[269]:

```
(43400, 12)
```

Here all the values below 1 are termed as outliers, although in rarest of cases Intrauterine stroke occur to unborn childre in the womb.

But in this project we drop those values, but in future we can even work on these values.

In [270]:

```
dodge[dodge['age'] < 1.00]
```

Out[270]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	str
116	7559	Female	0.64	0	0	No	children	Urban	83.82	24.9	NaN	
129	22706	Female	0.88	0	0	No	children	Rural	88.11	15.5	NaN	
323	61511	Female	0.32	0	0	No	children	Rural	73.71	16.2	NaN	
746	54747	Male	0.88	0	0	No	children	Rural	157.57	19.2	NaN	
761	53279	Male	0.24	0	0	No	children	Rural	118.87	16.3	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
43031	2698	Female	0.32	0	0	No	children	Urban	91.86	17.6	NaN	
43106	51999	Male	0.32	0	0	No	children	Urban	90.38	16.1	NaN	
43220	36634	Female	0.08	0	0	No	children	Rural	125.11	12.1	NaN	
43296	52578	Male	0.56	0	0	No	children	Rural	78.07	21.9	NaN	
43330	18634	Female	0.72	0	0	No	children	Urban	87.74	16.6	NaN	

496 rows × 12 columns

In [271]:

```
dodge[dodge['age'] < 1.00].index
```

Out[271]:

```
Int64Index([  116,   129,   323,   746,   761,   861,   975,  1087,  1375,
             1389,
            ...
            42637, 42862, 42880, 42881, 42982, 43031, 43106, 43220, 43296,
            43330],
           dtype='int64', length=496)
```

In [272]:

```
chevy = dodge.drop(index = dodge[dodge['age'] < 1.00].index, axis = 0, inplace=True)
```

In [273]:

```
dodge.drop(index = dodge[(dodge.age > 1.0) & (dodge.age < 2.0)].index, axis = 0, inplace = True)
```

In [274]:

```
dodge.shape
```

Out[274]:

```
(42309, 12)
```

2. avg_glucose_level(Average Glucose Level)

In [275]:

```
anamolies = []
def outliers(data):
  random_state_mean = np.mean(data)
  random_state_std = np.std(data)
  anamoly = random_state_std * 3

  upper_limit = random_state_mean + anamoly
  lower_limit = random_state_mean - anamoly
  ll_p = min(dodge['avg_glucose_level'])

  print(upper_limit)
  print(lower_limit)
  print(ll_p)
  for i in data:
    if i < ll_p or i > upper_limit:
      anamolies.append(i)
```

In [276]:

```
outliers(dodge['avg_glucose_level'])
print(len(anamolies))
```

```
235.13454455171652
-25.55617162869774
55.0
575
```

In [277]:

```
dodge['avg_glucose_level'].describe()
```

Out[277]:

```
count    42309.000000
mean       104.789186
std         43.448966
min         55.000000
25%         77.570000
50%         91.650000
75%        112.260000
max        291.050000
Name: avg_glucose_level, dtype: float64
```

In [278]:

```
dodge[dodge['avg_glucose_level'] > 234.40827023316058 ]
```

Out[278]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_statu
7	41413	Female	75.0	0	1	Yes	Self-employed	Rural	243.53	27.000000	never smoke
54	18518	Male	66.0	0	0	Yes	Private	Rural	242.30	35.300000	smoke
77	4480	Male	76.0	0	0	Yes	Private	Rural	234.58	34.300000	formerly smoke
78	2982	Female	57.0	1	0	Yes	Private	Rural	235.85	40.100000	never smoke
83	59368	Female	78.0	0	0	Yes	Private	Urban	243.50	26.100000	never smoke
...	...	...	...	...	...	...	...	...	...	...	..
43228	27207	Male	41.0	1	0	Yes	Private	Rural	271.01	25.800000	NaN
43279	49997	Male	67.0	0	0	Yes	Self-employed	Rural	242.61	47.000000	NaN
43283	29575	Female	30.0	0	0	No	Self-employed	Urban	258.24	28.605038	never smoke
43287	22198	Male	66.0	0	0	Yes	Private	Rural	238.23	33.300000	formerly smoke
43358	40203	Male	78.0	0	0	Yes	Self-employed	Rural	248.93	21.600000	formerly smoke

617 rows × 12 columns

In [279]:

```
dodge[dodge['avg_glucose_level'] > 234.40827023316058].index
```

Out[279]:

```
Int64Index([    7,    54,    77,    78,    83,    96,   139,   310,   322,
              469,
            ...
            43140, 43144, 43155, 43175, 43188, 43228, 43279, 43283, 43287,
            43358],
           dtype='int64', length=617)
```

In [280]:

```
dodge.drop(index = dodge[dodge['avg_glucose_level'] > 234.40827023316058].index, axis = 0, inplace = True
```

In [281]:

```
dodge.shape
```

Out[281]:

```
(41692, 12)
```

3. bmi(Body Mass Index)

In [282]:

```
anamolies = []
```

```
def outliers(data):
  random_state_mean = np.mean(data)
  random_state_std = np.std(data)
  anamoly = random_state_std * 3

  upper_limit = random_state_mean + anamoly
  lower_limit = random_state_mean - anamoly
  lll_p = min(dodge['bmi'])

  print(upper_limit)
  print(lower_limit)
  print(lll_p)
  for i in data:
    if i < lll_p or i > upper_limit:
      anamolies.append(i)
```

In [283]:

```
outliers(dodge['bmi'])
print(len(anamolies))
```

```
51.34051370653113
6.268167446955189
10.1
431
```

In [284]:

```
dodge['bmi'].describe()
```

Out[284]:

```
count    41692.000000
mean        28.804341
std          7.512148
min         10.100000
25%         23.700000
50%         28.200000
75%         32.700000
max         97.600000
Name: bmi, dtype: float64
```

In [285]:

```
dodge[dodge['bmi'] > 51.35486554902225]
```

Out[285]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	str
9	28674	Female	74.0	1	0	Yes	Self-employed	Urban	205.84	54.6	never smoked	
21	72911	Female	57.0	1	0	Yes	Private	Rural	129.54	60.9	smokes	
86	1703	Female	52.0	0	0	Yes	Private	Urban	82.24	54.7	formerly smoked	
111	66333	Male	52.0	0	0	Yes	Self-employed	Urban	78.40	64.8	never smoked	
184	53144	Female	52.0	0	1	Yes	Private	Urban	72.79	54.7	never smoked	
...	...	...	...	...	...	...	...	...	...	...	...	
43025	14846	Male	50.0	1	0	Yes	Govt_job	Rural	75.29	52.0	never smoked	
43087	70198	Male	78.0	1	0	Yes	Private	Rural	135.73	89.0	never smoked	
43239	36167	Male	21.0	0	0	No	Private	Urban	83.78	54.9	never smoked	
43355	57237	Female	46.0	0	0	Yes	Private	Rural	99.81	53.2	NaN	
43396	5450	Female	56.0	0	0	Yes	Govt_job	Urban	213.61	55.4	formerly smoked	

431 rows × 12 columns

In [286]:

```
dodge[dodge['bmi'] > 51.35486554902225].index
```

Out[286]:

```
Int64Index([    9,    21,    86,   111,   184,   220,   297,   302,   396,
              422,
            ...
            42560, 42589, 42604, 42831, 42977, 43025, 43087, 43239, 43355,
            43396],
           dtype='int64', length=431)
```

In [287]:

```
dodge.drop(index = dodge[dodge['bmi'] > 51.35486554902225].index, axis = 0, inplace = True)
```

In [288]:

```
dodge.shape
```

Out[288]:

```
(41261, 12)
```

Exploring Independent Categorical(Object/String) Columns.

1. Cleaning
2. Treating Missing values

In [289]:

```
dodge.isnull().sum()
```

Out[289]:

```
id                       0
gender                   0
age                      0
hypertension             0
heart_disease            0
ever_married             0
work_type                0
Residence_type           0
avg_glucose_level        0
bmi                      0
smoking_status       12015
stroke                   0
dtype: int64
```

In [290]:

```
categorical = ['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status']
```

In [291]:

```
for i in categorical:
  print(dodge[i].describe())
```

```
count      41261
unique         3
top       Female
freq       24498
Name: gender, dtype: object
count     41261
unique        2
top         Yes
freq      27051
Name: ever_married, dtype: object
count       41261
unique          5
top       Private
freq        24195
Name: work_type, dtype: object
count     41261
unique        2
top       Urban
freq      20664
Name: Residence_type, dtype: object
count            29246
unique               3
top       never smoked
freq             15655
Name: smoking_status, dtype: object
```

In [292]:

```
dodge.describe(include = 'object')
```

Out[292]:

	gender	ever_married	work_type	Residence_type	smoking_status
count	41261	41261	41261	41261	29246
unique	3	2	5	2	3
top	Female	Yes	Private	Urban	never smoked
freq	24498	27051	24195	20664	15655

In [293]:

```
dodge['smoking_status'].value_counts()
```

Out[293]:

```
never smoked       15655
formerly smoked     7222
smokes              6369
Name: smoking_status, dtype: int64
```

In [294]:

```
dodge.describe(include = 'all')
```

Out[294]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	
count	41261.000000	41261	41261.000000	41261.000000	41261.000000	41261	41261	41261	41261.000000	41261.0
unique	NaN	3	NaN	NaN	NaN	2	5	2	NaN	
top	NaN	Female	NaN	NaN	NaN	Yes	Private	Urban	NaN	
freq	NaN	24498	NaN	NaN	NaN	27051	24195	20664	NaN	
mean	36315.893992	NaN	42.998134	0.092533	0.046945	NaN	NaN	NaN	102.504529	28.5
std	21080.388177	NaN	21.848829	0.289780	0.211524	NaN	NaN	NaN	39.968402	6.9
min	1.000000	NaN	1.000000	0.000000	0.000000	NaN	NaN	NaN	55.000000	10.1
25%	18007.000000	NaN	25.000000	0.000000	0.000000	NaN	NaN	NaN	77.370000	23.7
50%	36315.000000	NaN	44.000000	0.000000	0.000000	NaN	NaN	NaN	91.170000	28.1
75%	54539.000000	NaN	60.000000	0.000000	0.000000	NaN	NaN	NaN	110.770000	32.5
max	72943.000000	NaN	82.000000	1.000000	1.000000	NaN	NaN	NaN	234.380000	51.3

Treating Missing values in Object columns using,

1. mean/median/mode
2. Based on frequency Distribution.

In [295]:

```
dodge['smoking_status'].mode()
```

Out[295]:

```
0    never smoked
dtype: object
```

In [296]:

```
dodge['smoking_status'].fillna('never smoked',inplace = True)
```

In [297]:

```
dodge['smoking_status'].isnull().sum()
```

Out[297]:

```
0
```

In [298]:

```
dodge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 41261 entries, 0 to 43399
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 41261 non-null  int64  
 1   gender             41261 non-null  object 
 2   age                41261 non-null  float64
 3   hypertension       41261 non-null  int64  
 4   heart_disease      41261 non-null  int64  
 5   ever_married       41261 non-null  object 
 6   work_type          41261 non-null  object 
 7   Residence_type     41261 non-null  object 
 8   avg_glucose_level  41261 non-null  float64
 9   bmi                41261 non-null  float64
 10  smoking_status     41261 non-null  object 
 11  stroke             41261 non-null  int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 4.1+ MB
```

In [299]:

```
dodge.head()
```

Out[299]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	strok
0	30669	Male	3.0	0	0	No	children	Rural	95.12	18.0	never smoked	
1	30468	Male	58.0	1	0	Yes	Private	Urban	87.96	39.2	never smoked	
2	16523	Female	8.0	0	0	No	Private	Urban	110.89	17.6	never smoked	
3	56543	Female	70.0	0	0	Yes	Private	Rural	69.04	35.9	formerly smoked	
4	46136	Male	14.0	0	0	No	Never_worked	Rural	161.28	19.1	never smoked	

Exploratory Data Analysis.

Exploratory Data Analysis helps us the understand the insights and extract the patterns from the dataset, which might be helpful to explain about the problem statement given to our clients. This can also be done by using traditional python code, But Visualizing the data looks more eye catching than looking at some numbers and letters. so, hence we are going to use various plots and graphs to visualize, which comes from the libraries such as, seaborn and matplotlib.pyplot.

1. bar
2. countplot
3. piechart
4. hist
5. box
6. scatterplot
7. pairplot

Apart from this we have also used and auto visualization tool, "autoviz"

In [300]:

```
dodge.isnull().sum()
```

Out[300]:

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
dtype: int64
```

In [301]:

```
dodge.drop(columns = 'id', inplace=True)
```

In [302]:

```
dodge.head()
```

Out[302]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	3.0	0	0	No	children	Rural	95.12	18.0	never smoked	0
1	Male	58.0	1	0	Yes	Private	Urban	87.96	39.2	never smoked	0
2	Female	8.0	0	0	No	Private	Urban	110.89	17.6	never smoked	0
3	Female	70.0	0	0	Yes	Private	Rural	69.04	35.9	formerly smoked	0
4	Male	14.0	0	0	No	Never_worked	Rural	161.28	19.1	never smoked	0

-> pd.crosstab() function is a very useful and most advanced fuction in the python dataframe, it helps us to compare 2 variables, due to which we can plot the distribution of thsoe variables.

1. Bar plot for crosstab distribution between gender and stroke.

In [303]:

```
plt.figure(figsize = (8,6))
x = pd.crosstab(dodge['gender'], dodge['stroke'])
x.plot(kind = 'bar')
#x.div(x.sum(1).astype(float), axis = 0).plot(kind='bar', stacked = False)
plt.xlabel('Gender_distribution')
plt.ylabel('Count')
plt.title('Gender Distribution over Target Class')
plt.show()
```

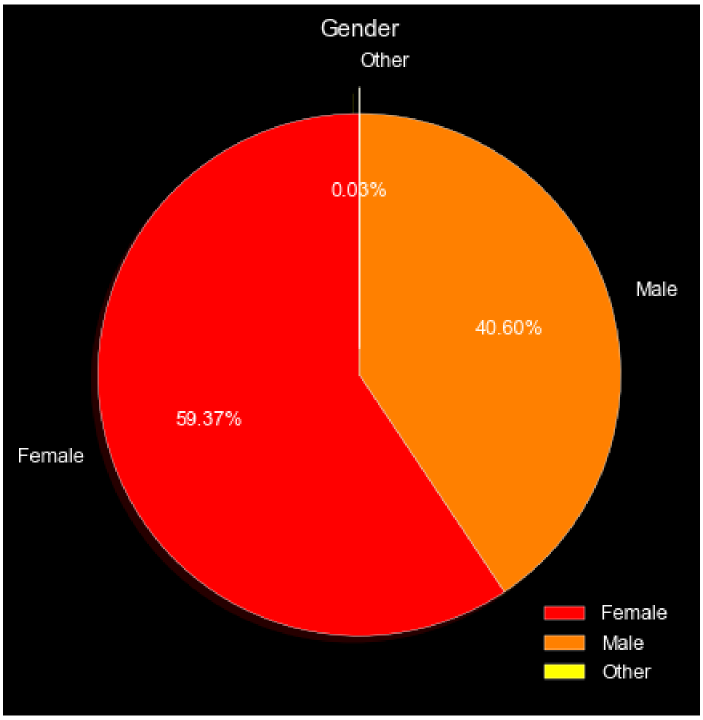
```
<Figure size 576x432 with 0 Axes>
```

2. Pie Chart for distribution of gender.

In [304]:

```
# PIE CHART for dodge['gender'] column.

plt.figure(figsize = (8,6), dpi = 90)
labels = dodge['gender'].value_counts().index
sizes = dodge['gender'].value_counts()
explode = [0,0,0.1]
colors = plt.cm.autumn(np.linspace(0,1,3))
plt.pie(sizes, colors=colors, labels=labels, explode=explode, shadow =True, startangle=90, autopct = '%.2f
plt.title('Gender',fontsize=12)
plt.legend()
plt.show()
```

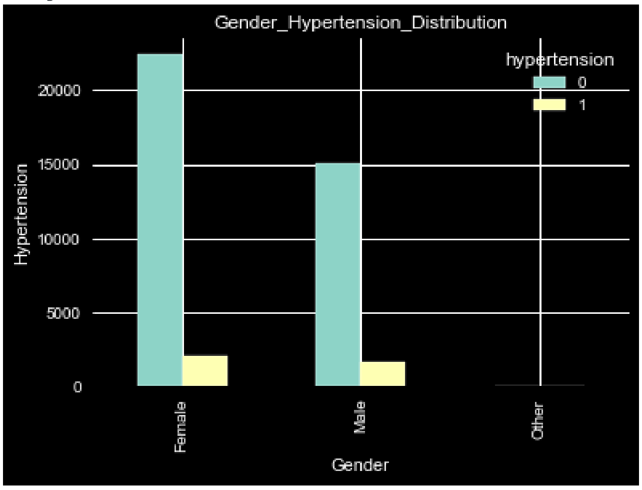


3. Bar chart for gender-hypertentsion distribution.

In [305]:

```
plt.figure(figsize = (8,6), dpi = 90)
x = pd.crosstab(dodge['gender'],dodge['hypertension'])
x.plot(kind = 'bar')
plt.xlabel('Gender')
plt.ylabel('Hypertension')
plt.title("Gender_Hypertension_Distribution")
plt.show()
```

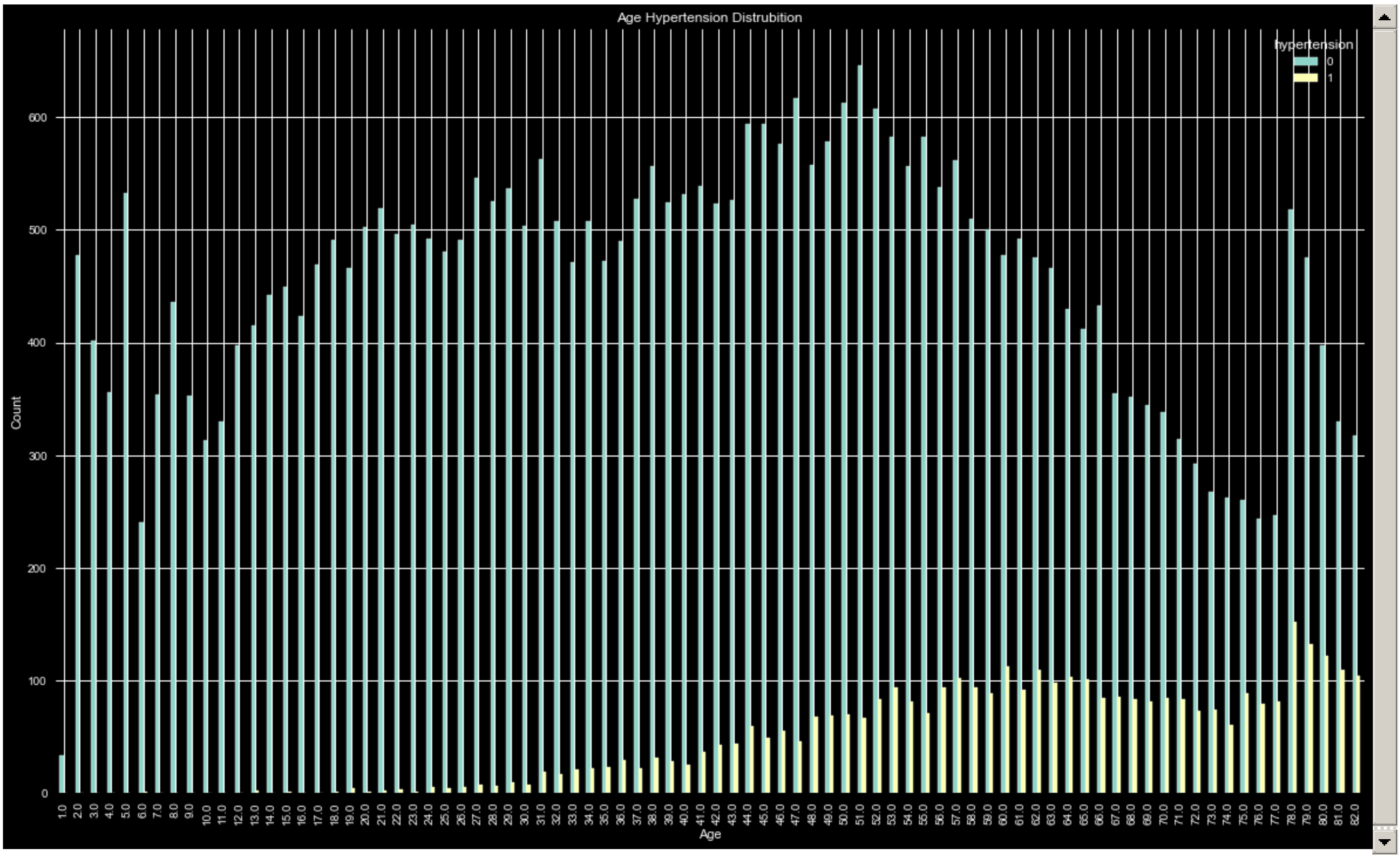
```
<Figure size 720x540 with 0 Axes>
```



4. Bar Chart for age-hypertension distribution

In [306]:

```
plt.rcParams['figure.figsize'] = (20,12)
x = pd.crosstab(dodge['age'], dodge['hypertension'])
x.plot(kind = 'bar')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title("Age Hypertension Distrubition")
plt.show()
```

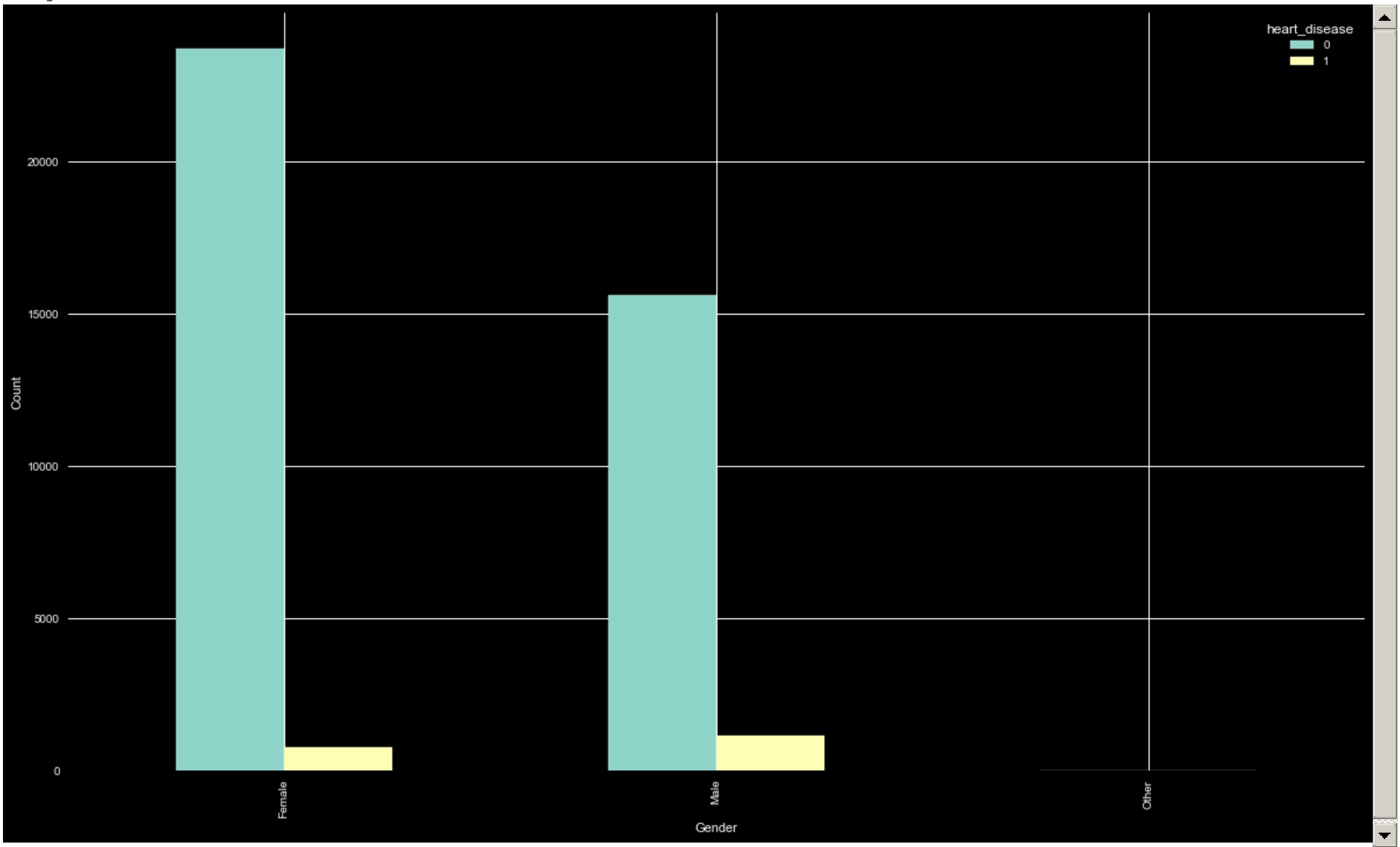


5. Bar Chart for gender-heart_disease distribution

In [307]:

```
plt.figure(figsize=(12,10))
ab = pd.crosstab(dodge['gender'], dodge['heart_disease'])
ab.plot(kind = 'bar')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

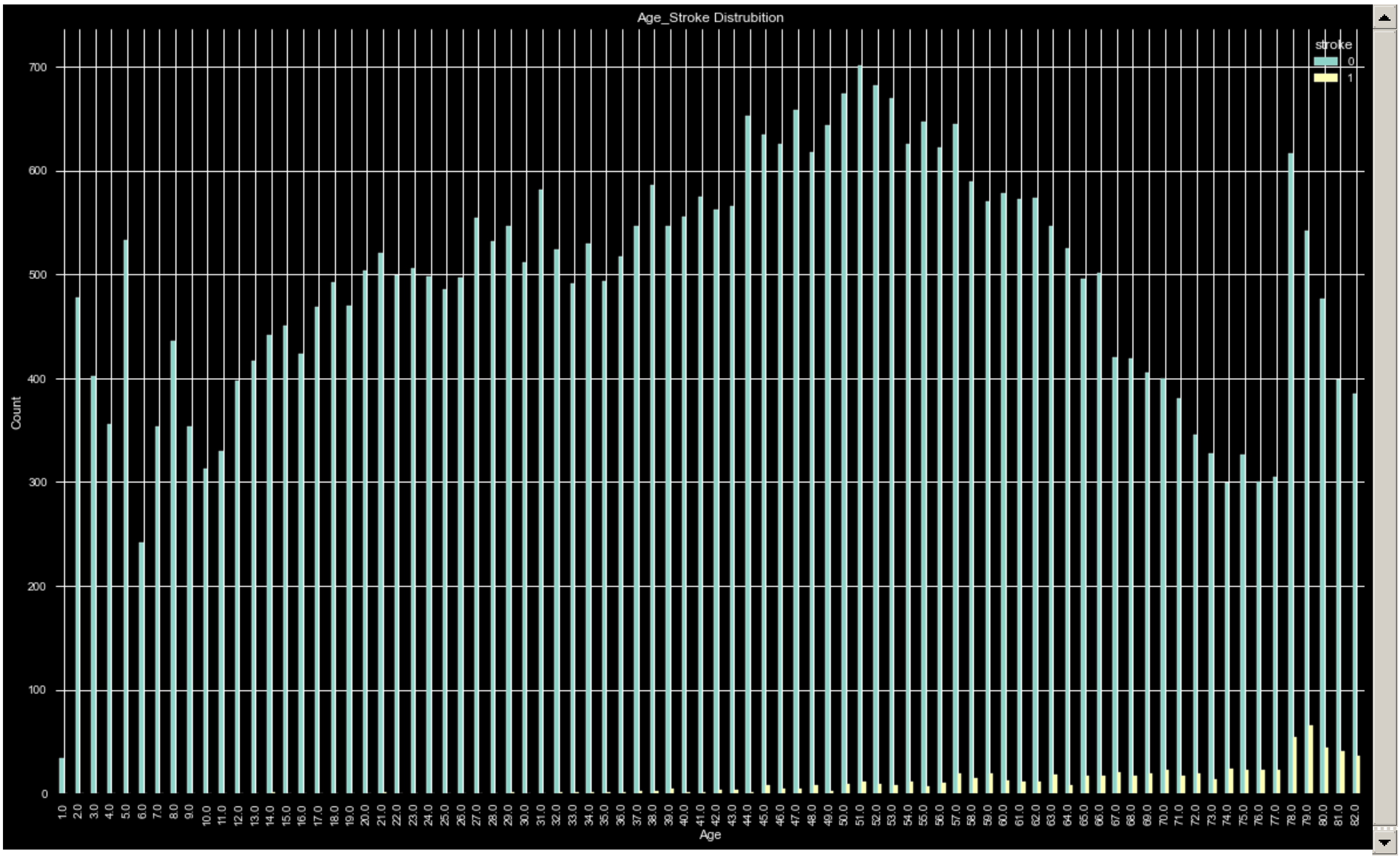
<Figure size 864x720 with 0 Axes>



6. age-stroke distribution

In [308]:

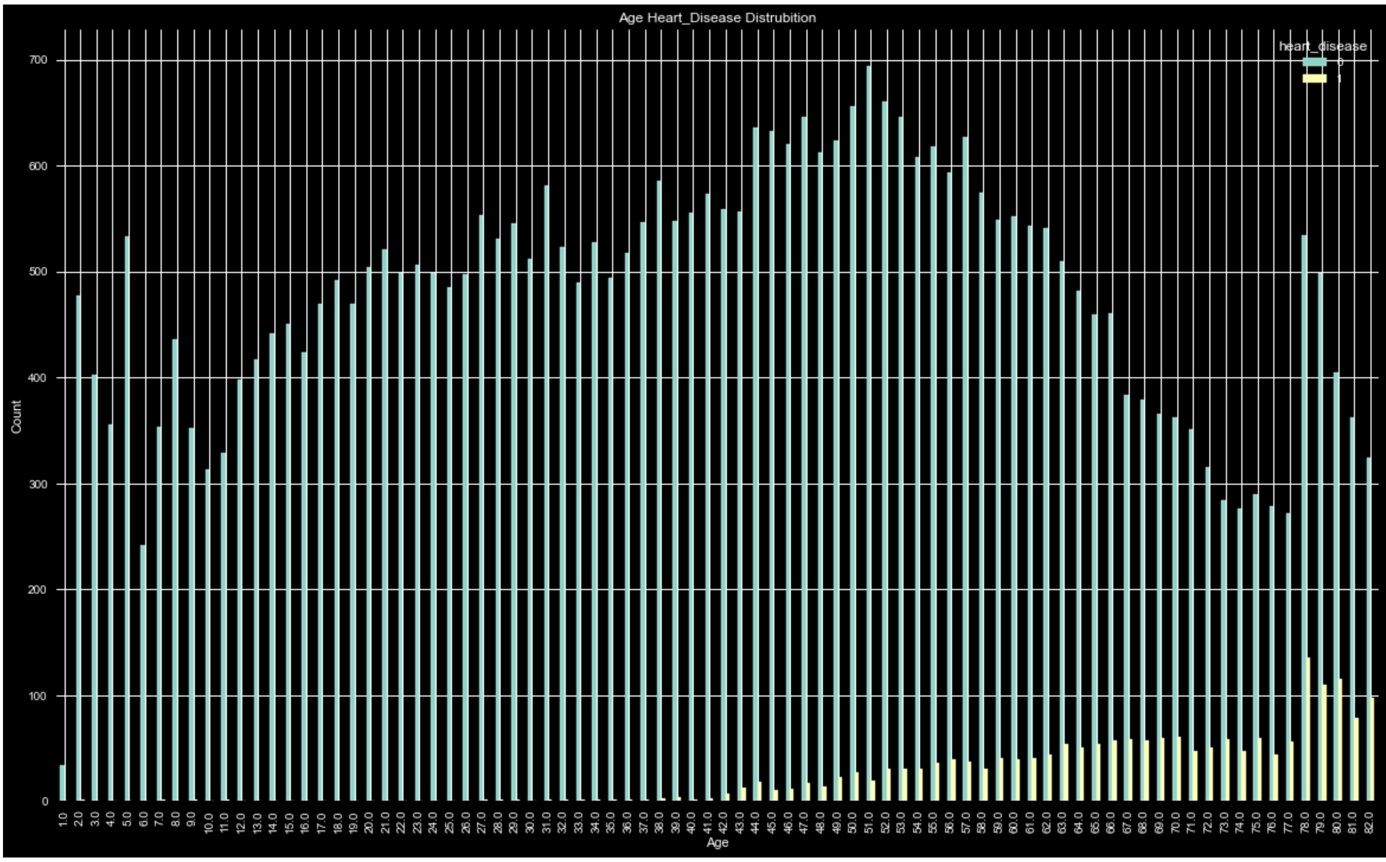
```
plt.rcParams['figure.figsize'] = (20,12)
x = pd.crosstab(dodge['age'], dodge['stroke'])
x.plot(kind = 'bar')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title("Age_Stroke Distrubition")
plt.show()
```



7. age-heart_disease distribution.

In [309]:

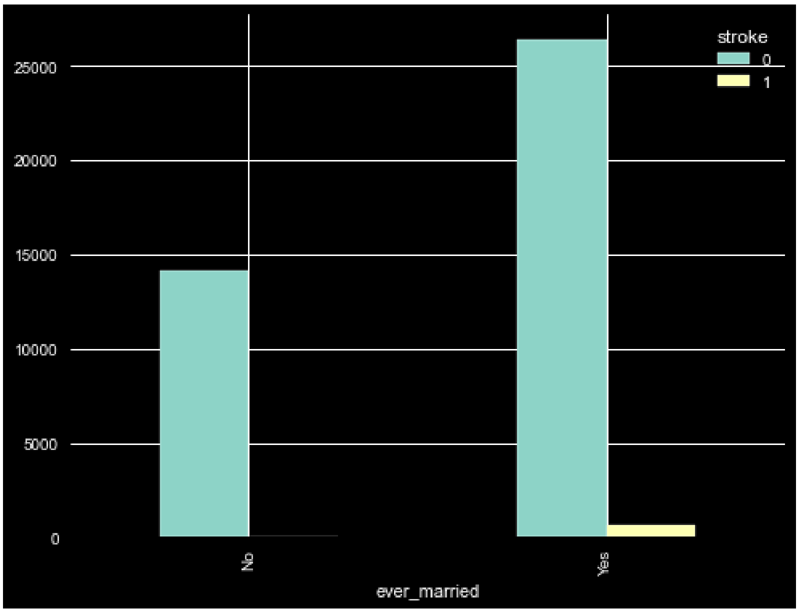
```
plt.rcParams['figure.figsize'] = (20,12)
#plt.figure(figsize =(13,6))
x = pd.crosstab(dodge['age'], dodge['heart_disease'])
x.plot(kind = 'bar')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title("Age Heart_Disease Distrubition")
plt.show()
```



8. Distribution of people getting stroke with respect to whether they are married or not.

In [310]:

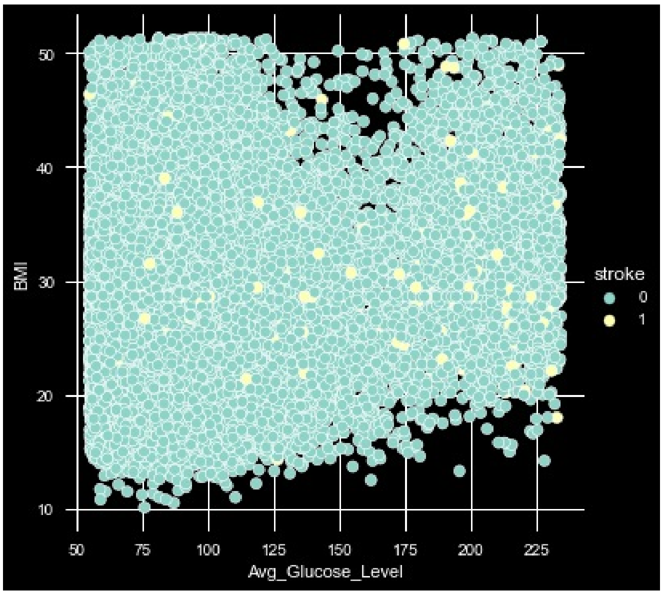
```
plt.rcParams['figure.figsize'] = (8,6)
h = pd.crosstab(dodge['ever_married'], dodge['stroke'])
h.plot(kind ='bar')
plt.show()
```



9. Scatterplot for avg_glucose level and bmi with hue as stroke, hue is an additional parameter which seperates the classes using different colors.

In [311]:

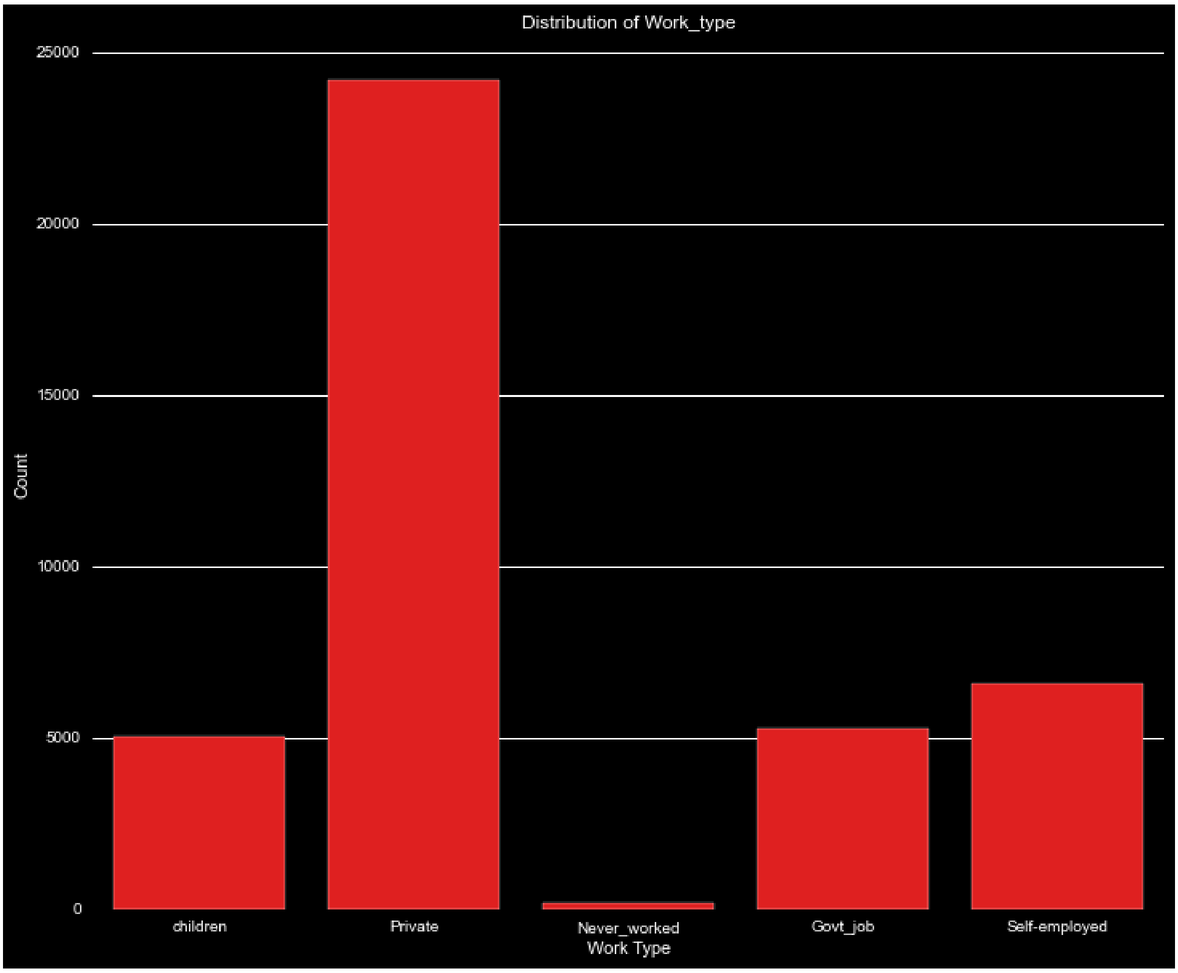
```
plt.rcParams['figure.figsize'] = (20,12)
sns.relplot(dodge['avg_glucose_level'], dodge['bmi'], hue = dodge['stroke'], kind = 'scatter')
plt.xlabel('Avg_Glucose_Level')
plt.ylabel('BMI')
plt.show()
```



10. Countplot() for checking distribution of work_type.

In [312]:

```
plt.figure(figsize = (12,10))
sns.countplot(dodge['work_type'], color ='red')
plt.xlabel("Work Type")
plt.ylabel('Count')
plt.title("Distribution of Work_type")
plt.show()
```

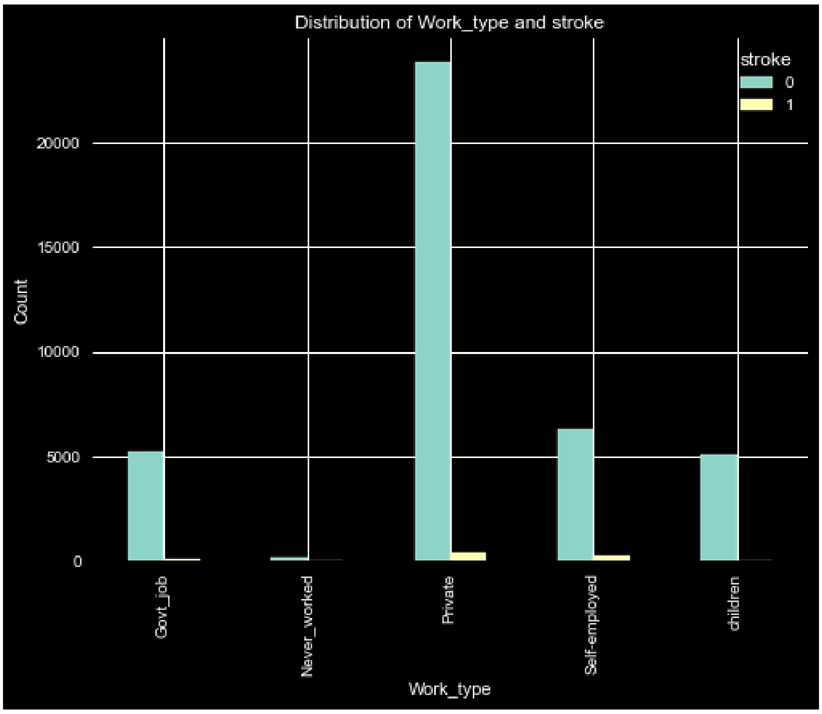


11. Distribution of work_type with respect to stroke occurence.

In [313]:

```
plt.rcParams['figure.figsize'] = (8,6)
h = pd.crosstab(dodge['work_type'], dodge['stroke'])
h.plot(kind ='bar')
```

```
plt.xlabel("Work_type")
plt.ylabel("Count")
plt.title("Distribution of Work_type and stroke")
plt.show()
```



autoviz -> An AutoVisualization tool, which helps to visualize the features in the dataset more in depth.

In [314]:

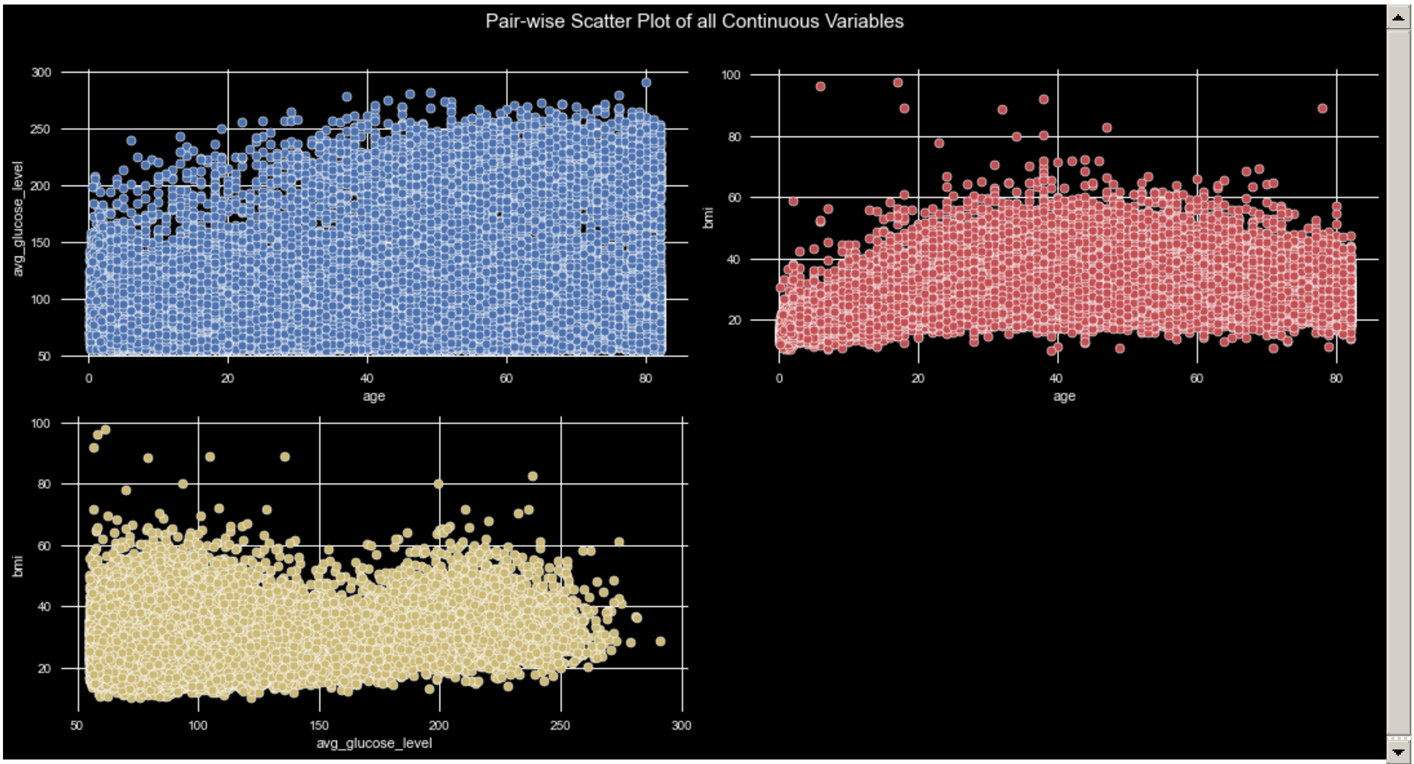
```
AV = AutoViz_Class()
autovis = AV.AutoViz(filename = 'train_strokes.csv', sep=',', depVar='', dfte=None, header=0, verbose=2,
                        lowess=False,chart_format='svg',max_rows_analyzed=150000,max_cols_analyzed=3(
autovis
```

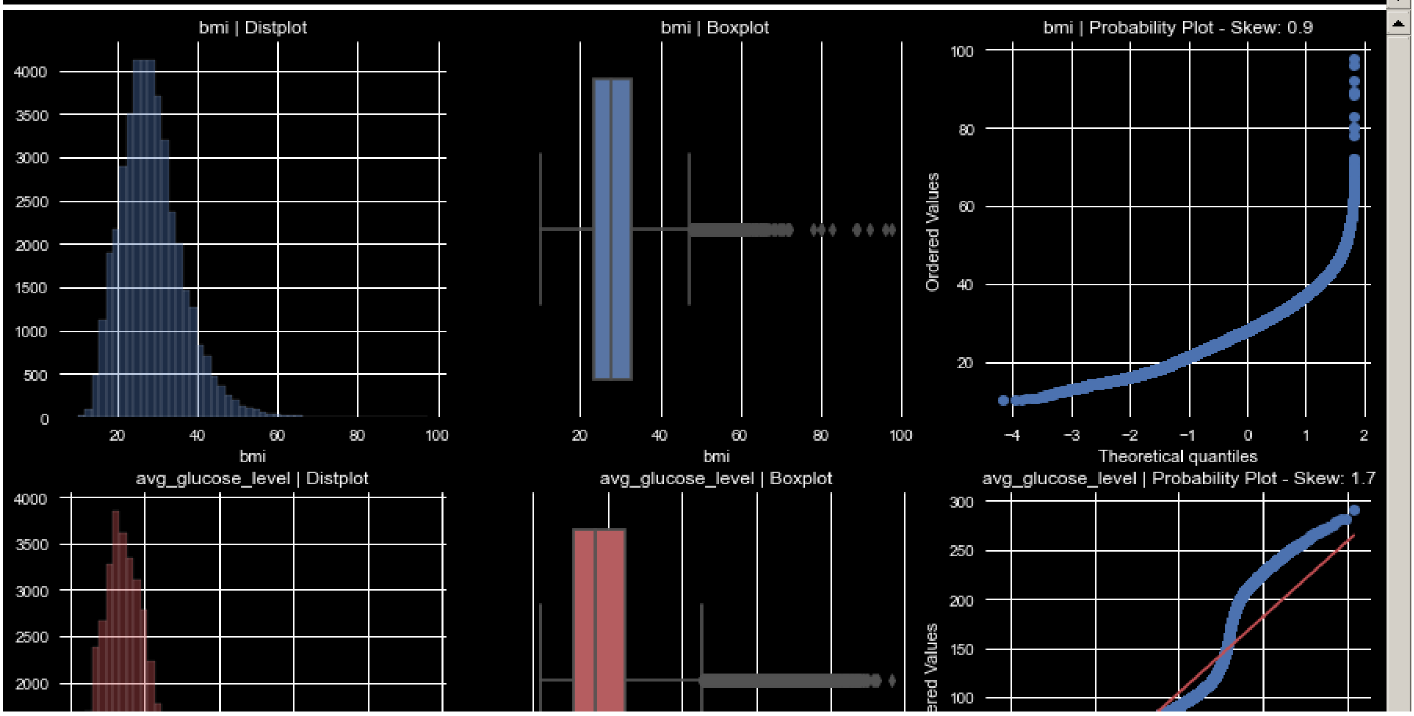
```
Shape of your Data Set: (43400, 12)
############## C L A S S I F Y I N G  V A R I A B L E S  ####################
Classifying variables in data set...
Data Set Shape: 43400 rows, 12 cols
Data Set columns info:
* id: 0 nulls, 43400 unique vals, most common: {2047: 1, 42270: 1}
* gender: 0 nulls, 3 unique vals, most common: {'Female': 25665, 'Male': 17724}
* age: 0 nulls, 104 unique vals, most common: {51.0: 738, 52.0: 721}
* hypertension: 0 nulls, 2 unique vals, most common: {0: 39339, 1: 4061}
* heart_disease: 0 nulls, 2 unique vals, most common: {0: 41338, 1: 2062}
* ever_married: 0 nulls, 2 unique vals, most common: {'Yes': 27938, 'No': 15462}
* work_type: 0 nulls, 5 unique vals, most common: {'Private': 24834, 'Self-employed': 6793}
* Residence_type: 0 nulls, 2 unique vals, most common: {'Urban': 21756, 'Rural': 21644}
* avg_glucose_level: 0 nulls, 12543 unique vals, most common: {82.71: 19, 87.07: 18}
* bmi: 1462 nulls, 555 unique vals, most common: {27.7: 271, 27.6: 267}
* smoking_status: 13292 nulls, 3 unique vals, most common: {'never smoked': 16053, 'formerly smoked': 
7493}
* stroke: 0 nulls, 2 unique vals, most common: {0: 42617, 1: 783}
--------------------------------------------------------------------
    Numeric Columns: ['age', 'avg_glucose_level', 'bmi']
    Integer-Categorical Columns: []
    String-Categorical Columns: ['gender', 'work_type', 'smoking_status']
    Factor-Categorical Columns: []
    String-Boolean Columns: ['ever_married', 'Residence_type']
    Numeric-Boolean Columns: ['hypertension', 'heart_disease', 'stroke']
    Discrete String Columns: []
    NLP text Columns: []
    Date Time Columns: []
    ID Columns: ['id']
    Columns that will not be considered in modeling: []
    12 Predictors classified...
        This does not include the Target column(s)
        1 variables removed since they were ID or low-information variables
    List of variables removed: ['id']
Number of All Scatter Plots = 6
Time to run AutoViz (in seconds) = 9.392
```

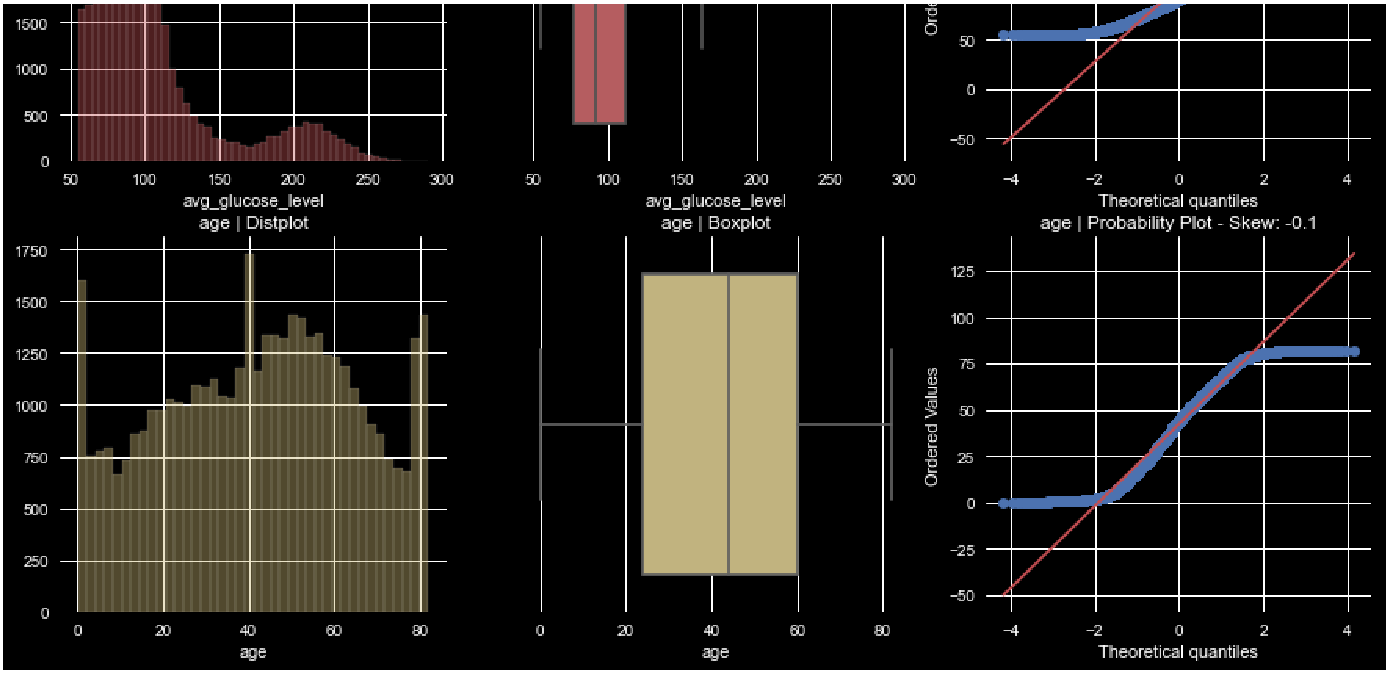
Out[314]:

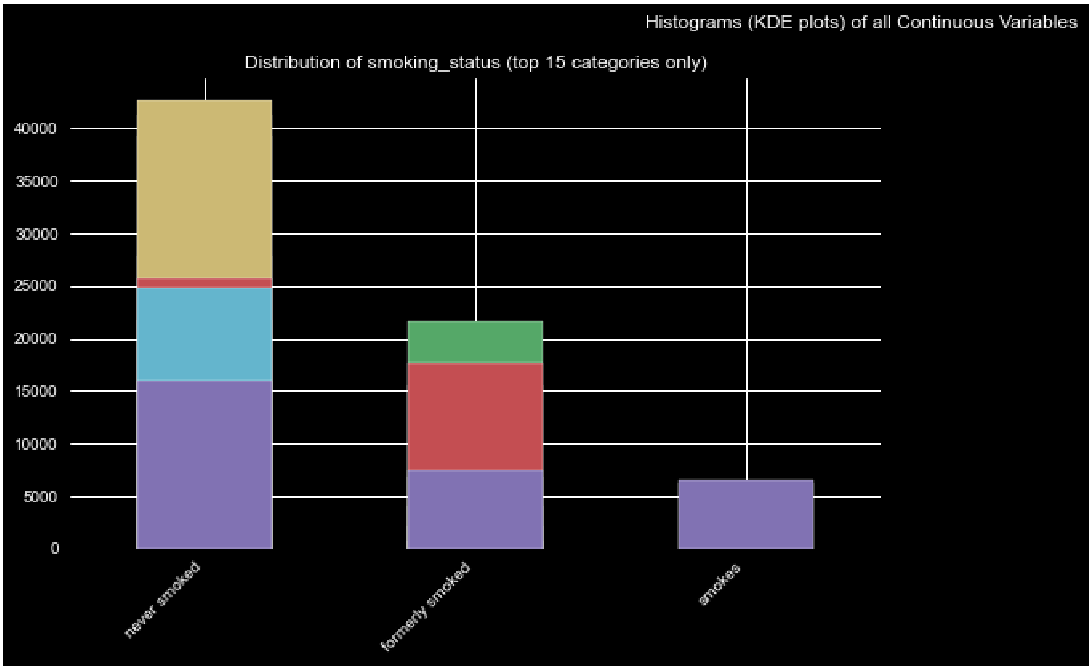
	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	3.0	0	0	No	children	Rural	95.12	18.0	NaN	0
1	Male	58.0	1	0	Yes	Private	Urban	87.96	39.2	never smoked	0
2	Female	8.0	0	0	No	Private	Urban	110.89	17.6	NaN	0
3	Female	70.0	0	0	Yes	Private	Rural	69.04	35.9	formerly smoked	0
4	Male	14.0	0	0	No	Never_worked	Rural	161.28	19.1	NaN	0
...	...	...	...	...	...	...	...	...	...	...	...
43395	Female	10.0	0	0	No	children	Urban	58.64	20.4	never smoked	0
43396	Female	56.0	0	0	Yes	Govt_job	Urban	213.61	55.4	formerly smoked	0
43397	Female	82.0	1	0	Yes	Private	Urban	91.94	28.9	formerly smoked	0
43398	Male	40.0	0	0	Yes	Private	Urban	99.16	33.2	never smoked	0
43399	Female	82.0	0	0	Yes	Private	Urban	79.48	20.6	never smoked	0

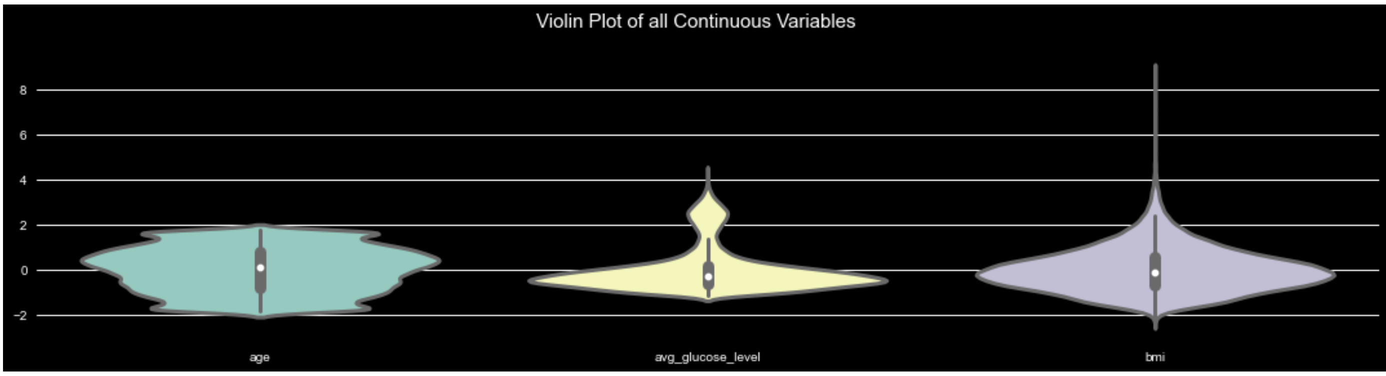
43400 rows × 11 columns

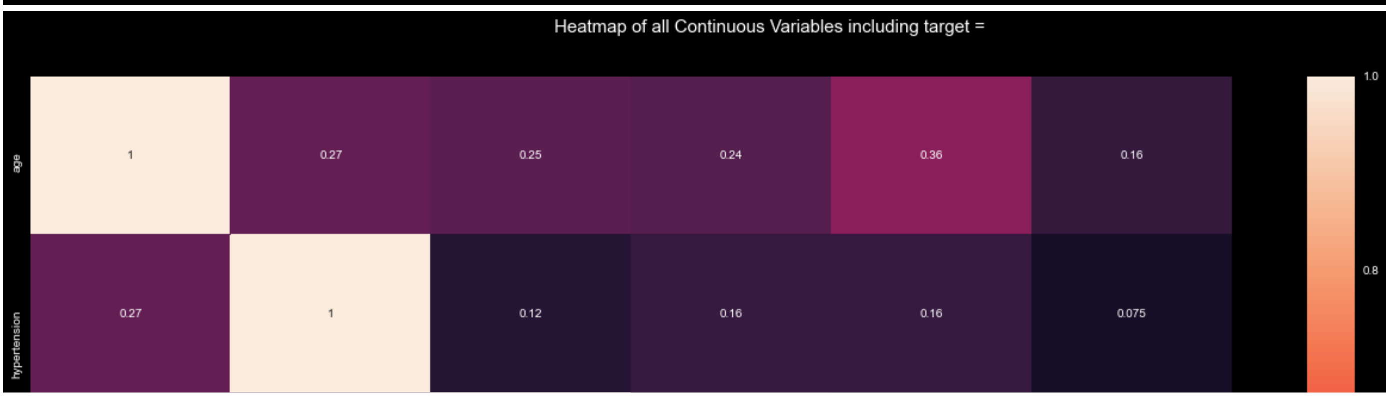




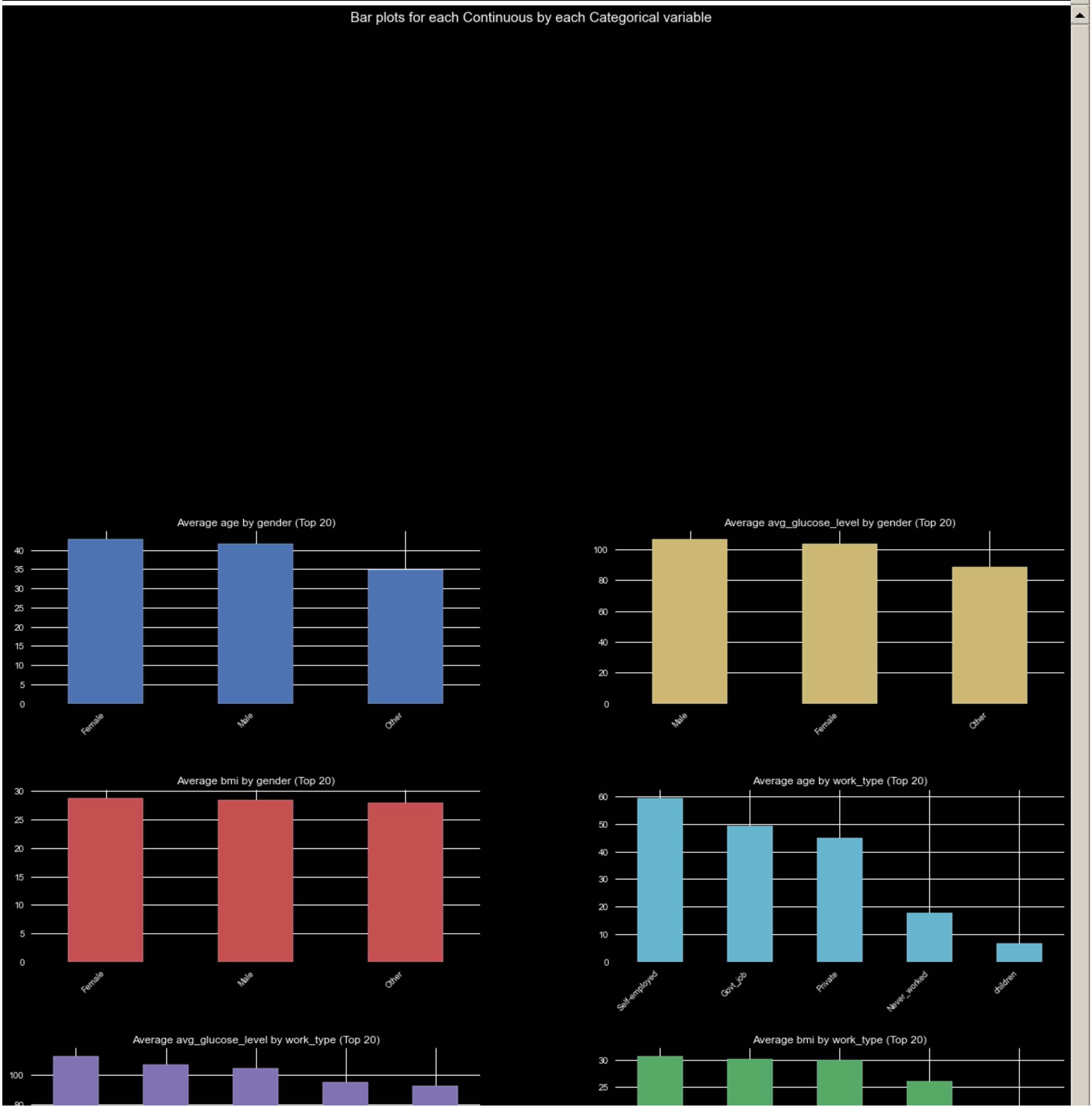

avg_glucose_level
age | Distplot
avg_glucose_level
age | Boxplot
Theoretical quantiles
age | Probability Plot - Skew: -0.1
Ordered Values
age
age
Theoretical quantiles

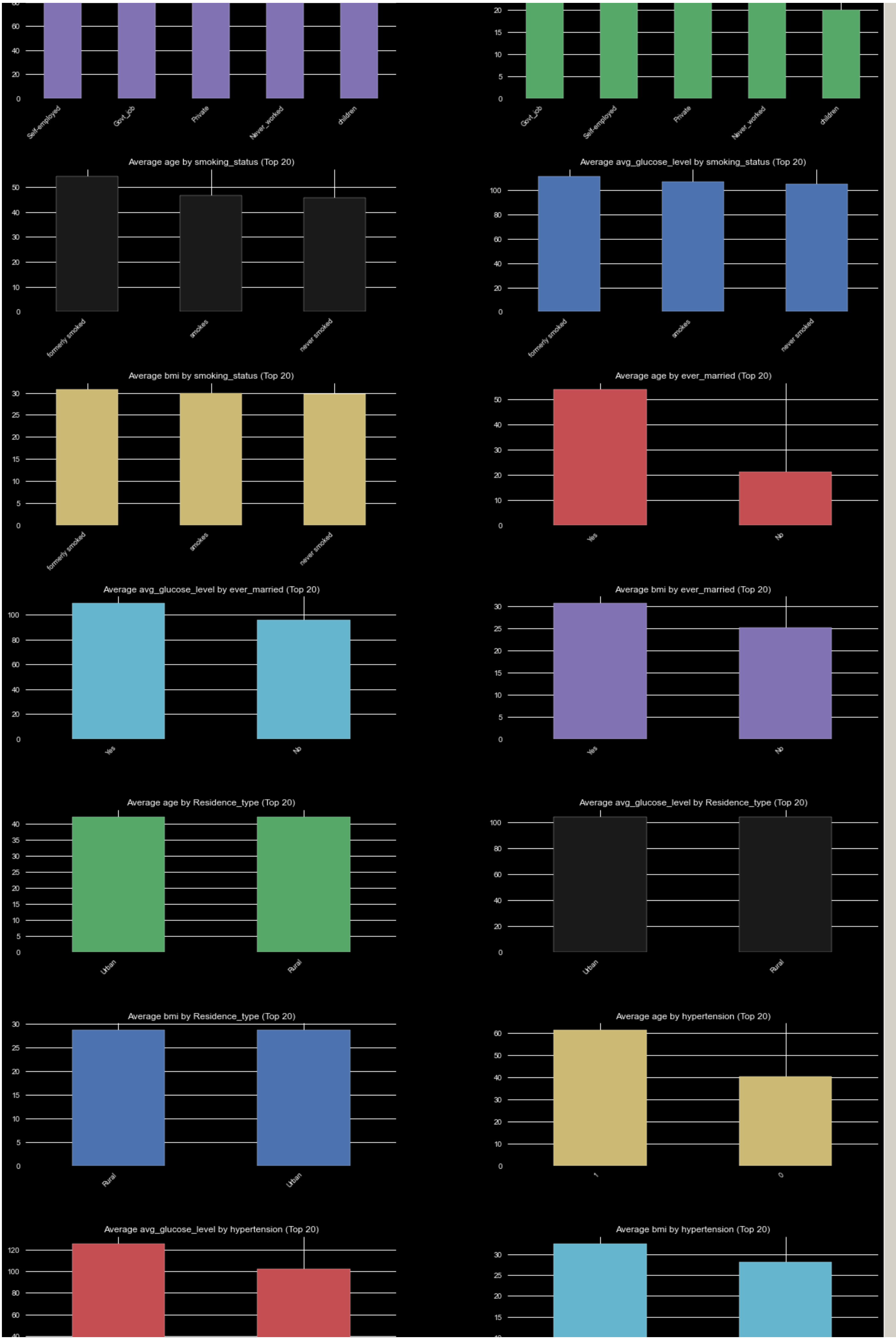

Histograms (KDE plots) of all Continuous Variables
Distribution of smoking_status (top 15 categories only)
never smoked
formerly smoked
smokes


Violin Plot of all Continuous Variables
age
avg_glucose_level
bmi

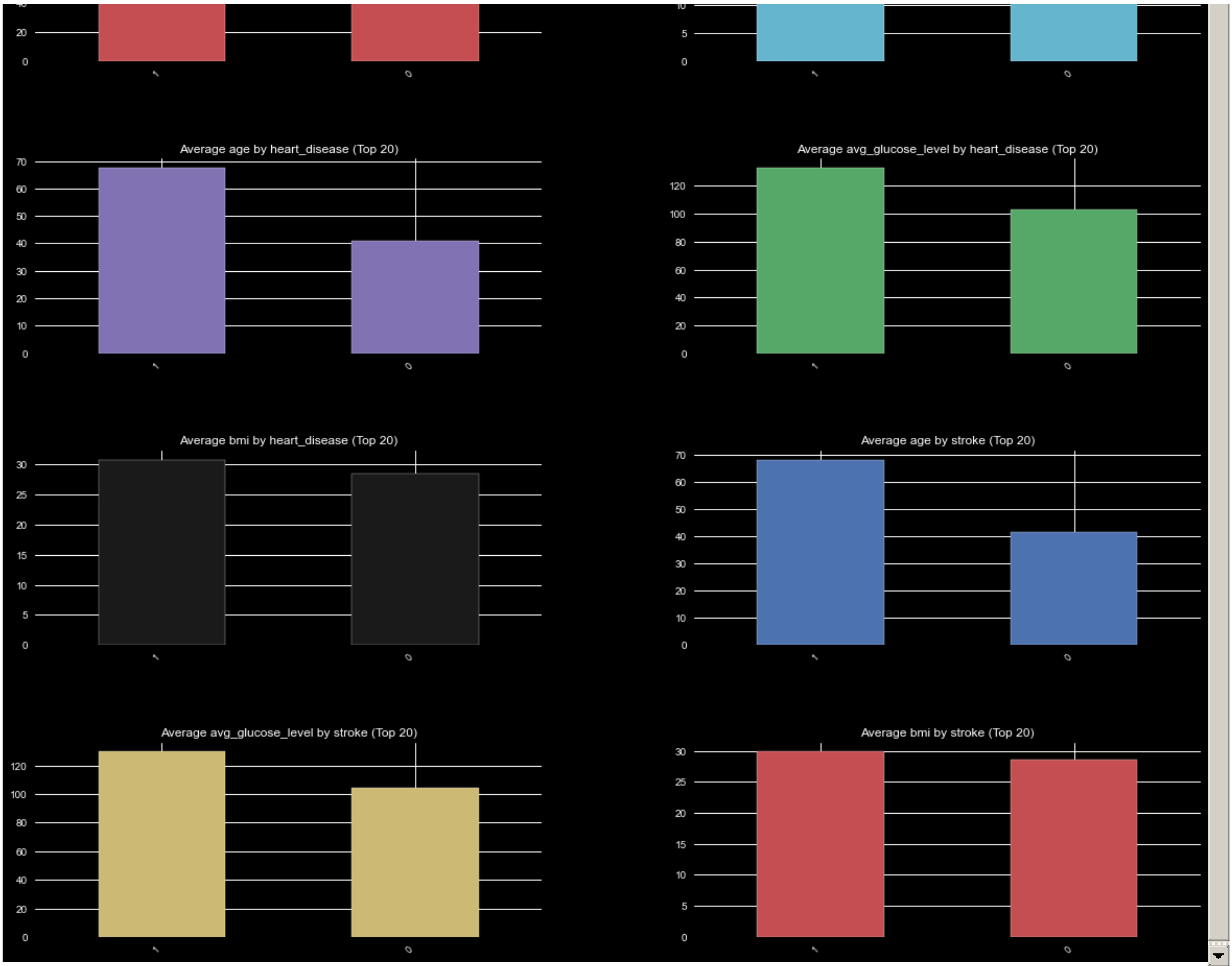

Heatmap of all Continuous Variables including target =
age
hypertension
1
0.27
0.25
0.24
0.36
0.16
0.27
1
0.12
0.16
0.16
0.075


heart_disease
avg_glucose_level
bmi
stroke
0.25 0.12 1 0.15 0.058 0.11
0.24 0.16 0.15 1 0.19 0.079
0.36 0.16 0.058 0.19 1 0.02
0.16 0.075 0.11 0.079 0.02 1
age
hypertension
heart_disease
avg_glucose_level
bmi
stroke
0.6
0.4
0.2


Bar plots for each Continuous by each Categorical variable
Average age by gender (Top 20)
Female
Male
Other
Average avg_glucose_level by gender (Top 20)
Male
Female
Other
Average bmi by gender (Top 20)
Female
Male
Other
Average age by work_type (Top 20)
Self-employed
Govt_job
Private
Never_worked
children
Average avg_glucose_level by work_type (Top 20)
Average bmi by work_type (Top 20)



Self-employed
Govt_job
Private
Never_worked
children
Govt_job
Self-employed
Private
Never_worked
children
Average age by smoking_status (Top 20)
formerly smoked
smokes
never smoked
Average avg_glucose_level by smoking_status (Top 20)
formerly smoked
smokes
never smoked
Average bmi by smoking_status (Top 20)
formerly smoked
smokes
never smoked
Average age by ever_married (Top 20)
Yes
No
Average avg_glucose_level by ever_married (Top 20)
Yes
No
Average bmi by ever_married (Top 20)
Yes
No
Average age by Residence_type (Top 20)
Urban
Rural
Average avg_glucose_level by Residence_type (Top 20)
Urban
Rural
Average bmi by Residence_type (Top 20)
Rural
Urban
Average age by hypertension (Top 20)
1
0
Average avg_glucose_level by hypertension (Top 20)
Average bmi by hypertension (Top 20)



Exploring data using Traditional python code, with the help of interactive widgets.

In [315]:

```
abg = dodge[['hypertension', 'heart_disease']].groupby(['hypertension']).count().style.background_gradier
```

Sum of Heart Disease values with respect to hypertension, This can be easily eaxplained by crosstab()

In [316]:

```
abg
```

Out[316]:

	heart_disease
hypertension	
0	37443
1	3818

In [317]:

```
dre = pd.crosstab(dodge['hypertension'], dodge['heart_disease'])
dre
```

Out[317]:

heart_disease	0	1
hypertension		
0	35984	1459
1	3340	478

@interact:

The interact function (ipywidgets.interact) automatically creates user interface (UI) controls for exploring code and data interactively.

The function gets called each time the slider is moved.

In [318]:

```
@interact
def abc(x = 50):
    y = dodge[dodge['avg_glucose_level'] > x]
    return y['stroke'].value_counts()
abc()
```

Out[318]:

```
0    40517
1      744
Name: stroke, dtype: int64
```

In [319]:

```
@interact
def hyp_heart(x=0, y=0):
    g = dodge[(dodge['hypertension'] == x) & (dodge['heart_disease'] == y)]
    return g['stroke'].value_counts()
hyp_heart()
```

Out[319]:

```
0    35541
1      443
Name: stroke, dtype: int64
```

In [320]:

```
@interact
def hy_he_eve(x=0,y=0,z='No'):
    j = dodge[(dodge['hypertension'] == x) & (dodge['heart_disease'] == y) & (dodge['ever_married'] == z)]
    return j['stroke'].value_counts(), j['smoking_status'].value_counts()
hy_he_eve()
```

Out[320]:

```
(0    13690
 1       44
 Name: stroke, dtype: int64,
 never smoked       11124
 smokes              1437
 formerly smoked     1173
 Name: smoking_status, dtype: int64)
```

Feature Transformation.

Feature Transformation is the technique of transforming the variable into other form like Strings -> Numeric, splitting the Date Column in to pieces etc.

Types of encoding.

1. Nominal Encoding.

- one hot encoding -> Creating Dummy variables.
- one hot encoding with multi categories(more than 20 categories)
- mean encoding

2. Ordinal Encoding.

- Label Encoder
- target_guided_encoding

-> For the columns with less than 5 categories we can manually perform encoding, usinf map().

-> For Columns with more than 20 Categories we can perform one hot encoding with multi categories, where we tend to select the top categories based on their value_counts().

In [321]:

```
dodge.head()
```

Out[321]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	3.0	0	0	No	children	Rural	95.12	18.0	never smoked	0
1	Male	58.0	1	0	Yes	Private	Urban	87.96	39.2	never smoked	0
2	Female	8.0	0	0	No	Private	Urban	110.89	17.6	never smoked	0
3	Female	70.0	0	0	Yes	Private	Rural	69.04	35.9	formerly smoked	0
4	Male	14.0	0	0	No	Never_worked	Rural	161.28	19.1	never smoked	0

In [322]:

```
dodge['smoking_status'].unique()
```

Out[322]:

```
array(['never smoked', 'formerly smoked', 'smokes'], dtype=object)
```

In [323]:

```
mapping = {'Male':2, 'Female':1, 'Other':0}
mapping1 = {'No':0, 'Yes':1}
mapping2 = {'never smoked':0, 'formerly smoked':1, 'smokes':2}
```

In [324]:

```
dodge['gender'] = dodge['gender'].map(mapping)
```

In [325]:

```
dodge['ever_married'] = dodge['ever_married'].map(mapping1)
```

In [326]:

```
dodge['smoking_status'] = dodge['smoking_status'].map(mapping2)
```

In [327]:

```
dodge[['gender', 'smoking_status', 'ever_married']].head()
```

Out[327]:

	gender	smoking_status	ever_married
0	2	0	0
1	2	0	1
2	1	0	0
3	1	1	1
4	2	0	0

In [328]:

```
dodge['work_type'].unique()
```

Out[328]:

```
array(['children', 'Private', 'Never_worked', 'Govt_job', 'Self-employed'],
      dtype=object)
```

In [329]:

```
dodge['Residence_type'].unique()
```

Out[329]:

```
array(['Rural', 'Urban'], dtype=object)
```

In [330]:

```
dodge['home_town'] = pd.get_dummies(dodge['Residence_type'], drop_first = True)
```

Creating a new dataframe with respect to work_type.

In [331]:

```
f150 = pd.get_dummies(dodge['work_type'], drop_first = True)
```

Merging 2 DataFrames(dodge,f150) with the default join.

In [332]:

```
camero = pd.concat([dodge,f150], axis = 1)
```

In [333]:

```
camero.head()
```

Out[333]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke	hom
0	2	3.0	0	0	0	children	Rural	95.12	18.0	0	0	
1	2	58.0	1	0	1	Private	Urban	87.96	39.2	0	0	
2	1	8.0	0	0	0	Private	Urban	110.89	17.6	0	0	
3	1	70.0	0	0	1	Private	Rural	69.04	35.9	1	0	
4	2	14.0	0	0	0	Never_worked	Rural	161.28	19.1	0	0	

In [334]:

```
camero.rename(columns = {'Never_worked':'w_t_n_w', 'Private':'w_t_p', 'Self-employed':'w_t_s_e', 'childre
```

Droping the columns ['work_type', 'Residence_type'], as we have already created dummy variables for them.

In [335]:

```
camero.drop(columns = ['work_type','Residence_type'], inplace = True)
```

In [336]:

```
camero.head()
```

Out[336]:

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	smoking_status	stroke	home_town	w_t_n_w	w_t_p	w_t
0	2	3.0	0	0	0	95.12	18.0	0	0	0	0	0	
1	2	58.0	1	0	1	87.96	39.2	0	0	1	0	1	
2	1	8.0	0	0	0	110.89	17.6	0	0	1	0	1	
3	1	70.0	0	0	1	69.04	35.9	1	0	0	0	1	
4	2	14.0	0	0	0	161.28	19.1	0	0	0	1	0	

In [337]:

```
camero.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 41261 entries, 0 to 43399
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             41261 non-null  int64  
 1   age                41261 non-null  float64
 2   hypertension       41261 non-null  int64  
 3   heart_disease      41261 non-null  int64  
 4   ever_married       41261 non-null  int64  
 5   avg_glucose_level  41261 non-null  float64
 6   bmi                41261 non-null  float64
 7   smoking_status     41261 non-null  int64  
 8   stroke             41261 non-null  int64  
 9   home_town          41261 non-null  uint8  
 10  w_t_n_w            41261 non-null  uint8  
 11  w_t_p              41261 non-null  uint8  
 12  w_t_s_e            41261 non-null  uint8  
 13  w_t_c              41261 non-null  uint8  
dtypes: float64(3), int64(6), uint8(5)
memory usage: 4.6 MB
```

Feature Scaling

Feature Scaling is the technique to scale down all the values in the datset to same level, so that there will be no partiality while we train the model like bmi -> 56 getting high priority than heart_disease -> 0, so in order to remove this error, feature scaling is done.

Feature Scaling Tools.

1. Standardisation (values are centered around the mean with unit standard deviation.)
2. Normalisation/min_max scaling.(values range from 0 to 1)

StandardScaler()

In [338]:

```
se = StandardScaler()
abh = se.fit_transform(camero.drop(columns=['stroke']))
mercury = pd.DataFrame(data = abh, columns = camero.drop(columns = ['stroke']).columns)
mercury.head()
```

Out[338]:

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	smoking_status	home_town	w_t_n_w	w_t_
0	1.208899	-1.830699	-0.319325	-0.22194	-1.379732	-0.184761	-1.513815	-0.647332	-1.001625	-0.065637	1.19068
1	1.208899	0.686629	3.131608	-0.22194	0.724779	-0.363905	1.539898	-0.647332	0.998378	-0.065637	0.83985
2	-0.825374	-1.601851	-0.319325	-0.22194	-1.379732	0.209805	-1.571433	-0.647332	0.998378	-0.065637	0.83985
3	-0.825374	1.235865	-0.319325	-0.22194	0.724779	-0.837285	1.064556	0.690823	-1.001625	-0.065637	0.83985
4	1.208899	-1.327233	-0.319325	-0.22194	-1.379732	1.470566	-1.355368	-0.647332	-1.001625	15.235255	1.19068

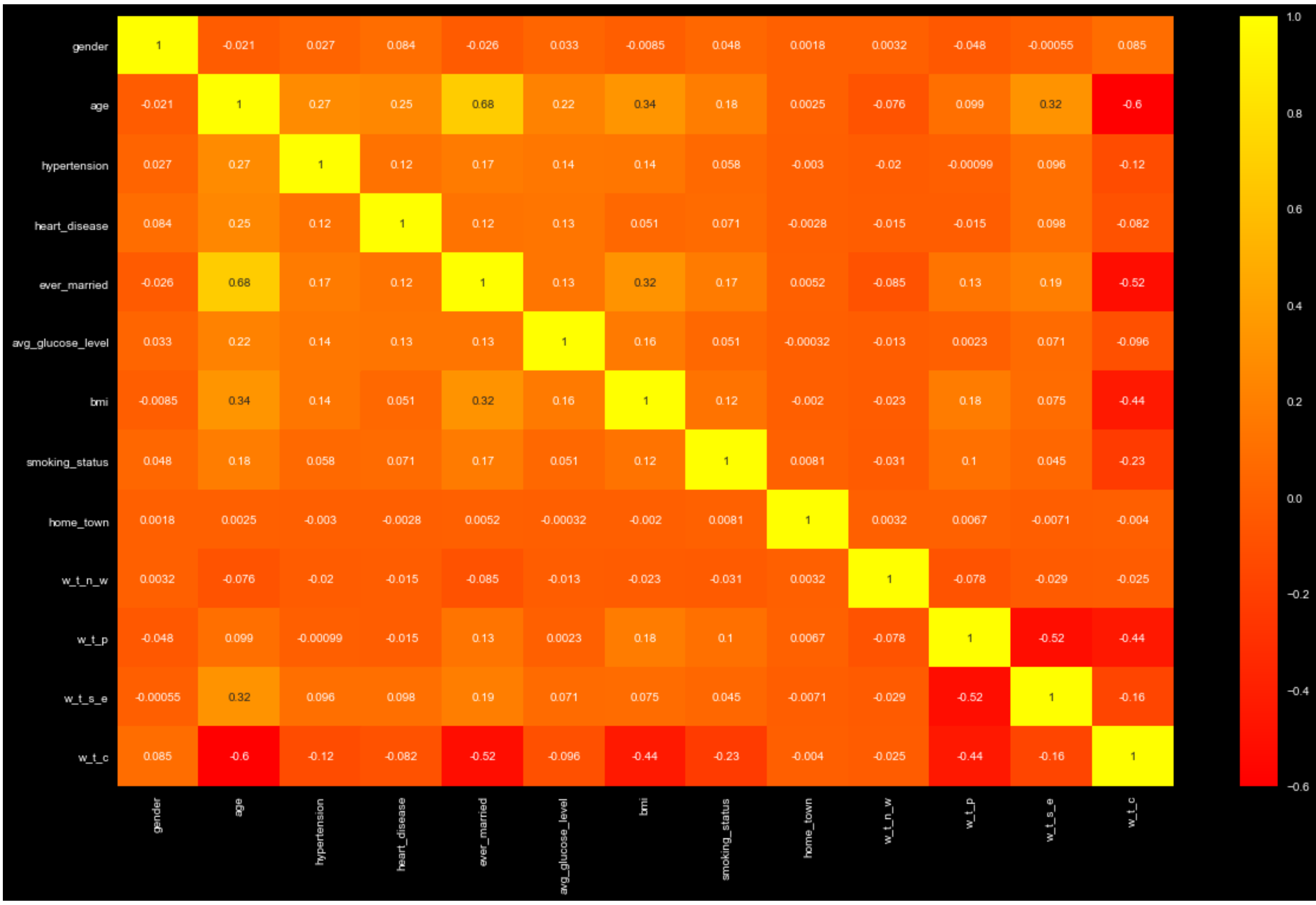
Feature Selection

Selecting the best features which best contribute to our model.

Correlation Diagram.

In [339]:

```
plt.rcParams['figure.figsize'] = (20,12)
corr = mercury.corr()
sns.heatmap(corr, annot=True, cmap='autumn')
plt.show()
```



Function to select the best features with some threshold value.

In [340]:

```
def correlation(dataset,threshold):
    corr_list = []
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i,j] > threshold):
                column = corr_matrix.columns[[i,j]]
                corr_list.append(column)
    print(len(corr_list))
    return corr_list
correlation(mercury,0.6)
```

```
1
```

Out[340]:

```
[Index(['ever_married', 'age'], dtype='object')]
```

Although, we can see ['ever_married', 'age'] are somewhat correlated, but where as if we use 'Variance Inflation Factor", we ended up with fixing the Multicollinearirty.

variance_inflation_factor -> it is used to remove multicollinearity between variables by removing as few variables as possible.

VIF->Variance Inflation Factor

In [341]:

```
vif = variance_inflation_factor
earth1 = pd.Series([vif(mercury.values, i) for i in range(mercury.shape[1])], index = mercury.columns)
earth1
```

Out[341]:

```
gender               1.022118
age                  2.637361
hypertension         1.098485
heart_disease        1.096197
ever_married         1.950928
avg_glucose_level    1.081189
bmi                  1.287565
smoking_status       1.069008
home_town            1.000213
w_t_n_w              1.051573
w_t_p                2.336830
w_t_s_e              1.949642
w_t_c                2.712860
dtype: float64
```

Function to check and remove multicollinearity between independent variables.

In [342]:

```
def mc(data):
  earth = pd.Series([vif(data.values, i) for i in range(data.shape[1])], index = data.columns)
  if earth.max() > 6:
    print(earth[earth == earth.max()].index[0], 'Has Been Removed.')
    data = data.drop(columns = earth[earth == earth.max()].index[0])
  else:
    print("MultiCollinearity Has Been Removed.")
    return data
```

In [343]:

```
for i in range(5):
  mercury = mc(mercury)
mercury.head()
```

```
MultiCollinearity Has Been Removed.
MultiCollinearity Has Been Removed.
MultiCollinearity Has Been Removed.
MultiCollinearity Has Been Removed.
MultiCollinearity Has Been Removed.
```

Out[343]:

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	smoking_status	home_town	w_t_n_w	w_t_
0	1.208899	-1.830699	-0.319325	-0.22194	-1.379732	-0.184761	-1.513815	-0.647332	-1.001625	-0.065637	1.19068
1	1.208899	0.686629	3.131608	-0.22194	0.724779	-0.363905	1.539898	-0.647332	0.998378	-0.065637	0.83985
2	-0.825374	-1.601851	-0.319325	-0.22194	-1.379732	0.209805	-1.571433	-0.647332	0.998378	-0.065637	0.83985
3	-0.825374	1.235865	-0.319325	-0.22194	0.724779	-0.837285	1.064556	0.690823	-1.001625	-0.065637	0.83985
4	1.208899	-1.327233	-0.319325	-0.22194	-1.379732	1.470566	-1.355368	-0.647332	-1.001625	15.235255	1.19068

Splitting Data

Splitting the dataset

1. target_var
2. Independent_var

In [344]:

```
target_var = camero['stroke']
inde_vars = camero.drop(columns=['stroke'], axis = 1)
```

In [345]:

```
target_var
```

Out[345]:

```
0        0
1        0
2        0
3        0
4        0
        ..
43394    0
43395    0
43397    0
43398    0
43399    0
Name: stroke, Length: 41261, dtype: int64
```

In [346]:

```
inde_vars.head()
```

Out[346]:

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	smoking_status	home_town	w_t_n_w	w_t_p	w_t_s_e	w
0	2	3.0	0	0	0	95.12	18.0	0	0	0	0	0	
1	2	58.0	1	0	1	87.96	39.2	0	1	0	1	0	
2	1	8.0	0	0	0	110.89	17.6	0	1	0	1	0	
3	1	70.0	0	0	1	69.04	35.9	1	0	0	1	0	
4	2	14.0	0	0	0	161.28	19.1	0	0	1	0	0	

Handling Imbalanced Dataset.

As we saw the target_calss was highly imbalanced, so we try to balance the target_class using Oversampling method, using "SMOTETomek" tool.

In [347]:

```
camero.head()
```

Out[347]:

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	smoking_status	stroke	home_town	w_t_n_w	w_t_p	w_t
0	2	3.0	0	0	0	95.12	18.0	0	0	0	0	0	
1	2	58.0	1	0	1	87.96	39.2	0	0	1	0	1	
2	1	8.0	0	0	0	110.89	17.6	0	0	1	0	1	
3	1	70.0	0	0	1	69.04	35.9	1	0	0	0	1	
4	2	14.0	0	0	0	161.28	19.1	0	0	0	1	0	

SMOTETomek Tool

In [348]:

```
so = SMOTETomek()
x_resample,y_resample = so.fit_sample(inde_vars, target_var.values.ravel())
brad = pd.DataFrame(data=x_resample, columns = inde_vars.columns)
```

In [349]:

```
#Before resampling
print("Before Resampling Target_Variable: ")
print(target_var.value_counts())

# After resampling
y_resample = pd.DataFrame(y_resample)
print("After Resampling Target_Variable:")
print(y_resample[0].value_counts())
```

```
Before Resampling Target_Variable:
0    40517
1      744
Name: stroke, dtype: int64
After Resampling Target_Variable:
1    40470
0    40470
Name: 0, dtype: int64
```

Train Test Split.

Splitting the data into train and test datasets.

In [350]:

```
x_train,x_test,y_train,y_test = train_test_split(x_resample, y_resample, test_size = 0.3, random_state =
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(56658, 13)
(56658, 1)
(24282, 13)
(24282, 1)
```

Feature Scaling Balanced Data.

Now, as we have balanced our data, we need to perform feature scaling to the banlanced data.

In [351]:

```
x_train_ss = se.fit_transform(x_train)
x_test_ss = se.transform(x_test)
```

Creating Test Data.

In [352]:

```
ford = pd.read_csv("healthcare-dataset-stroke-data.csv")
```

In [353]:

```
ford.head()
```

Out[353]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

In [354]:

```
ford.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

In [355]:

```
ford.drop(index = ford[(ford.age > 1.0) & (ford.age < 2.0)].index, axis = 0, inplace = True)
```

In [356]:

```
ford.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5038 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5038 non-null   int64  
 1   gender             5038 non-null   object 
 2   age                5038 non-null   float64
 3   hypertension       5038 non-null   int64  
 4   heart_disease      5038 non-null   int64  
 5   ever_married       5038 non-null   object 
 6   work_type          5038 non-null   object 
 7   Residence_type     5038 non-null   object 
 8   avg_glucose_level  5038 non-null   float64
 9   bmi                4842 non-null   float64
 10  smoking_status     5038 non-null   object 
 11  stroke             5038 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 511.7+ KB
```

In [357]:

```
ford.shape
```

Out[357]:

```
(5038, 12)
```

In [358]:

```
anamolies = []
def outliers(data):
  random_state_mean = np.mean(data)
  random_state_std = np.std(data)
  anamoly = random_state_std * 3

  upper_limit = random_state_mean + anamoly
  lower_limit = random_state_mean - anamoly
  uu =  max(ford['avg_glucose_level'])
  ll = min(ford['avg_glucose_level'])

  print(upper_limit)
  print(lower_limit)
  for i in data:
    if i < ll or i > uu:
      anamolies.append(i)
```

In [359]:

```
outliers(ford['avg_glucose_level'])
print(len(anamolies))
```

```
242.64307272917313
-30.033838906227473
0
```

In [360]:

```
dodge['avg_glucose_level'].describe()
```

Out[360]:

```
count    41261.000000
mean       102.504529
std         39.968402
min         55.000000
25%         77.370000
50%         91.170000
75%        110.770000
max        234.380000
Name: avg_glucose_level, dtype: float64
```

In [361]:

```
anamolies = []
def outliers(data):
  random_state_mean = np.mean(data)
  random_state_std = np.std(data)
  anamoly = random_state_std * 3

  upper_limit = random_state_mean + anamoly
  lower_limit = random_state_mean - anamoly
  ll = min(ford['bmi'])

  print(upper_limit)
  print(lower_limit)
  for i in data:
    if i < ll or i > upper_limit:
      anamolies.append(i)
```

In [362]:

```
outliers(ford['bmi'])
print(len(anamolies))
```

```
52.45615973942819
5.616289661645759
58
```

In [363]:

```
ford[ford['bmi'] > 52.45615973942819]
```

Out[363]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stro
113	41069	Female	45.0	0	0	Yes	Private	Rural	224.10	56.6	never smoked	
258	28674	Female	74.0	1	0	Yes	Self-employed	Urban	205.84	54.6	never smoked	
270	72911	Female	57.0	1	0	Yes	Private	Rural	129.54	60.9	smokes	
333	1703	Female	52.0	0	0	Yes	Private	Urban	82.24	54.7	formerly smoked	
358	66333	Male	52.0	0	0	Yes	Self-employed	Urban	78.40	64.8	never smoked	
430	53144	Female	52.0	0	1	Yes	Private	Urban	72.79	54.7	never smoked	
466	1307	Female	61.0	1	0	Yes	Private	Rural	170.05	60.2	smokes	
544	545	Male	42.0	0	0	Yes	Private	Rural	210.48	71.9	never smoked	
637	3130	Female	56.0	0	0	Yes	Private	Rural	112.43	54.6	never smoked	
662	23551	Male	28.0	0	0	Yes	Private	Urban	87.43	55.7	Unknown	
672	31145	Female	17.0	0	0	No	Private	Urban	67.81	55.7	never smoked	
715	3590	Female	28.0	1	0	No	Private	Rural	80.40	57.5	never smoked	
761	4169	Female	37.0	0	0	No	Private	Rural	92.78	54.2	never smoked	
928	41097	Female	23.0	1	0	No	Private	Urban	70.03	78.0	smokes	
1061	8332	Female	50.0	0	0	Yes	Private	Rural	206.25	53.4	formerly smoked	
1077	15220	Female	53.0	1	0	Yes	Private	Urban	87.03	55.2	formerly smoked	

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stro
[illegible]	[illegible]	Female	[illegible]	1	0	Yes	Private	Urban	[illegible]	[illegible]	formerly smoked	
1304	6040	Female	46.0	0	0	No	Private	Rural	79.63	55.0	Unknown	
1322	35913	Female	55.0	1	0	Yes	Private	Urban	206.40	54.8	never smoked	
1532	19735	Female	59.0	0	0	Yes	Private	Rural	79.18	52.8	formerly smoked	
1559	37759	Female	53.0	0	0	Yes	Private	Rural	72.63	66.8	Unknown	
1564	3178	Female	25.0	0	0	Yes	Private	Rural	68.78	55.1	formerly smoked	
1584	6372	Female	32.0	0	0	Yes	Private	Urban	97.14	55.9	never smoked	
1595	2898	Male	46.0	0	0	Yes	Private	Urban	87.66	57.3	never smoked	
1660	8960	Female	42.0	0	0	No	Self-employed	Rural	73.41	56.0	smokes	
1898	25405	Male	62.0	0	0	Yes	Govt_job	Urban	187.52	57.7	never smoked	
2071	61242	Female	41.0	1	0	Yes	Govt_job	Rural	107.50	54.0	never smoked	
2081	5355	Male	63.0	0	0	Yes	Govt_job	Rural	231.69	56.1	formerly smoked	
2128	56420	Male	17.0	1	0	No	Private	Rural	61.67	97.6	Unknown	
2136	59745	Female	27.0	0	0	Yes	Private	Urban	76.74	53.9	Unknown	
2330	32365	Male	42.0	0	0	Yes	Private	Rural	89.22	53.8	Unknown	
2441	3668	Female	65.0	0	0	Yes	Govt_job	Urban	84.47	52.7	smokes	
2545	19504	Female	66.0	0	0	Yes	Private	Rural	87.84	52.8	Unknown	
2555	7171	Female	56.0	0	0	Yes	Govt_job	Urban	102.51	55.7	Unknown	
2567	65564	Female	48.0	0	0	Yes	Private	Urban	57.43	53.5	formerly smoked	
2764	20292	Female	24.0	0	0	Yes	Private	Urban	85.55	63.3	never smoked	
2815	50215	Male	42.0	0	0	No	Govt_job	Rural	59.83	52.8	never smoked	
2840	65895	Female	52.0	0	0	Yes	Private	Urban	98.27	61.2	Unknown	
3060	32604	Male	49.0	0	0	Yes	Self-employed	Rural	215.81	58.1	never smoked	
3243	11111	Female	66.0	1	0	Yes	Govt_job	Urban	205.01	52.7	formerly smoked	
3508	65154	Female	30.0	0	0	Yes	Private	Urban	112.19	53.4	never smoked	
3588	23047	Male	43.0	0	0	Yes	Private	Urban	100.16	59.7	never smoked	
3606	14872	Male	45.0	1	0	Yes	Self-employed	Rural	239.19	52.5	Unknown	
3688	38575	Male	58.0	1	0	Yes	Self-employed	Rural	209.15	52.9	formerly smoked	
3702	7730	Male	31.0	0	0	No	Private	Rural	94.96	54.7	smokes	
3825	72784	Female	52.0	0	0	Yes	Private	Rural	118.46	61.6	smokes	
3909	4077	Male	49.0	0	0	Yes	Private	Urban	219.70	53.8	Unknown	
3931	27660	Female	73.0	1	0	No	Self-employed	Rural	198.30	54.3	formerly smoked	
3980	11192	Female	45.0	0	0	Yes	Private	Rural	218.10	55.0	smokes	
4154	47668	Female	49.0	0	0	Yes	Private	Rural	125.63	57.2	Unknown	
4188	70670	Female	27.0	0	0	Yes	Private	Rural	57.96	64.4	never smoked	
4209	51856	Male	38.0	1	0	Yes	Private	Rural	56.90	92.0	never smoked	
4225	14658	Female	37.0	0	0	Yes	Private	Rural	77.10	55.9	Unknown	
4351	63915	Female	39.0	0	0	Yes	Private	Urban	87.39	57.9	never smoked	
4407	49277	Female	34.0	0	0	No	Private	Urban	70.87	55.7	formerly smoked	
4475	60675	Female	48.0	1	0	Yes	Govt_job	Rural	221.08	57.2	never smoked	
4838	5131	Female	51.0	0	0	Yes	Private	Urban	107.72	60.9	Unknown	
4906	72696	Female	53.0	0	0	Yes	Private	Urban	70.51	54.1	never smoked	
4952	16245	Male	51.0	1	0	Yes	Self-employed	Rural	211.83	56.6	never smoked	

In [364]:

```
ford.drop(index = ford[ford['bmi'] > 52.45615973942819].index, axis = 0, inplace = True)
```

In [365]:

```
ford.shape
```

Out[365]:

```
(4980, 12)
```

In [366]:

```
ford.isnull().sum()
```

Out[366]:

```
id                     0
gender                 0
age                    0
hypertension           0
heart_disease          0
ever_married           0
work_type              0
Residence_type         0
avg_glucose_level      0
bmi                  196
smoking_status         0
stroke                 0
dtype: int64
```

In [367]:

```
ford['bmi'].mean()
```

Out[367]:

```
28.681291806020063
```

In [368]:

```
ford['bmi'].fillna(ford['bmi'].mean(), inplace = True)
```

In [369]:

```
ford['bmi'].isnull().sum()
```

Out[369]:

```
0
```

In [370]:

```
ford['smoking_status'].replace('Unknown', 'never smoked')
```

Out[370]:

```
0       formerly smoked
1          never smoked
2          never smoked
3                smokes
4          never smoked
              ...
5105       never smoked
5106       never smoked
5107       never smoked
5108    formerly smoked
5109       never smoked
Name: smoking_status, Length: 4980, dtype: object
```

In [371]:

```
ford.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4980 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 4980 non-null   int64  
 1   gender             4980 non-null   object 
 2   age                4980 non-null   float64
 3   hypertension       4980 non-null   int64  
 4   heart_disease      4980 non-null   int64  
 5   ever_married       4980 non-null   object 
 6   work_type          4980 non-null   object 
 7   Residence_type     4980 non-null   object 
 8   avg_glucose_level  4980 non-null   float64
 9   bmi                4980 non-null   float64
 10  smoking_status     4980 non-null   object 
 11  stroke             4980 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 505.8+ KB
```

In [372]:

```
ford.drop(columns = ['id'], axis=1, inplace = True)
```

In [373]:

```
ford['smoking_status'].replace({'Unknown':'never smoked'}, inplace = True)
```

In [374]:

```
ford['gender'] = ford['gender'].map(mapping)
```

In [375]:

```
ford['ever_married'] = ford['ever_married'].map(mapping1)
```

In [376]:

```
ford['smoking_status'] = ford['smoking_status'].map(mapping2)
```

In [377]:

```
ford[['gender', 'smoking_status', 'ever_married']].head()
```

Out[377]:

	gender	smoking_status	ever_married
0	2	1	1
1	1	0	1
2	2	0	1
3	1	2	1
4	1	0	1

In [378]:

```
ford['work_type'].unique()
```

Out[378]:

```
array(['Private', 'Self-employed', 'Govt_job', 'children', 'Never_worked'],
      dtype=object)
```

In [379]:

```
ford['Residence_type'].unique()
```

Out[379]:

```
array(['Urban', 'Rural'], dtype=object)
```

In [380]:

```
ford['home_town'] = pd.get_dummies(ford['Residence_type'], drop_first = True)
```

In [381]:

```
rap = pd.get_dummies(ford['work_type'], drop_first = True)
```

In [382]:

```
cam = pd.concat([ford,rap], axis = 1)
```

In [383]:

```
cam.head()
```

Out[383]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke	h
0	2	67.0	0	1	1	Private	Urban	228.69	36.600000	1	1	
1	1	61.0	0	0	1	Self-employed	Rural	202.21	28.681292	0	1	
2	2	80.0	0	1	1	Private	Rural	105.92	32.500000	0	1	
3	1	49.0	0	0	1	Private	Urban	171.23	34.400000	2	1	
4	1	79.0	1	0	1	Self-employed	Rural	174.12	24.000000	0	1	

In [384]:

```
cam.rename(columns = {'Never_worked':'w_t_n_w', 'Private':'w_t_p', 'Self-employed':'w_t_s_e', 'children':
```

In [385]:

```
cam.drop(columns = ['work_type','Residence_type'], inplace = True)
```

In [386]:

```
target = cam['stroke']
original = cam.drop(columns = ['stroke'])
```

In [387]:

```
resampled_x,resampled_y = so.fit_resample(original,target.values.ravel())
pitt = pd.DataFrame(data = resampled_x, columns=original.columns)
```

In [388]:

```
#Before resampling
print("Before Resampling Target_Variable: ")
print(target.value_counts())

# After resampling
resampled_y = pd.DataFrame(resampled_y)
print("After Resampling Target_Variable:")
print(resampled_y[0].value_counts())
```

```
Before Resampling Target_Variable:
0    4733
1     247
Name: stroke, dtype: int64
After Resampling Target_Variable:
1    4685
0    4685
Name: 0, dtype: int64
```

In [389]:

```
fish = se.fit_transform(resampled_x)
lucas = pd.DataFrame(data = fish, columns = original.columns)
```

In [390]:

```
lucas.head()
```

Out[390]:

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	smoking_status	home_town	w_t_n_w	w_t_p
0	1.439490	0.527117	-0.308237	4.467882	0.613451	1.981290	1.207034	0.887229	1.201633	-0.048512	1.046323
1	-0.694015	0.251840	-0.308237	-0.223820	0.613451	1.502107	-0.094405	-0.619629	-0.832201	-0.048512	-0.955728
2	1.439490	1.123550	-0.308237	4.467882	0.613451	-0.240362	0.533199	-0.619629	-0.832201	-0.048512	1.046323
3	-0.694015	-0.298714	-0.308237	-0.223820	0.613451	0.941491	0.845464	2.394087	1.201633	-0.048512	1.046323
4	-0.694015	1.077670	3.244259	-0.223820	0.613451	0.993789	-0.863775	-0.619629	-0.832201	-0.048512	-0.955728

In [391]:

```
lucas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9370 entries, 0 to 9369
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             9370 non-null   float64
 1   age                9370 non-null   float64
 2   hypertension       9370 non-null   float64
 3   heart_disease      9370 non-null   float64
 4   ever_married       9370 non-null   float64
 5   avg_glucose_level  9370 non-null   float64
 6   bmi                9370 non-null   float64
 7   smoking_status     9370 non-null   float64
 8   home_town          9370 non-null   float64
 9   w_t_n_w            9370 non-null   float64
 10  w_t_p              9370 non-null   float64
 11  w_t_s_e            9370 non-null   float64
 12  w_t_c              9370 non-null   float64
dtypes: float64(13)
memory usage: 951.8 KB
```

Building Predictive Models.

1. Decision Tree
2. Random Forest
3. Logistic Regression

Decision Tree Classifier

In [392]:

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(x_train_ss,y_train)
predictions = dt.predict(x_test_ss)

print('The Training Accuracy of x_train and y_train is', dt.score(x_train_ss,y_train))
print("The Testing Accuracy of x_test and y_test is", dt.score(x_test_ss,y_test))
```

```
The Training Accuracy of x_train and y_train is 1.0
The Testing Accuracy of x_test and y_test is 0.953751750267688
```

In [393]:

```
print(confusion_matrix(predictions,y_test))
```

```
[[11463   459]
 [  664 11696]]
```

In [394]:

```
print(classification_report(predictions,y_test))
```

```
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     11922
           1       0.96      0.95      0.95     12360

    accuracy                           0.95     24282
   macro avg       0.95      0.95      0.95     24282
weighted avg       0.95      0.95      0.95     24282
```

In [395]:

```
print(accuracy_score(predictions, y_test))
```

```
0.953751750267688
```

Tree Plot.

In [396]:

```
plt.figure(figsize = (15,10))
tree.plot_tree(dt, filled = True)
```

Out[396]:

```
[Text(266.3129478253746, 535.3636363636364, 'X[1] <= -0.253\ngini = 0.5\nsamples = 56658\nvalue = [28343, 
28315]'),
 Text(41.14472276558815, 518.8909090909091, 'X[1] <= -0.991\ngini = 0.211\nsamples = 19561\nvalue = [1721
5, 2346]'),
 Text(13.035640610795724, 502.41818181818184, 'X[5] <= -0.732\ngini = 0.025\nsamples = 10723\nvalue = 
[10587, 136]'),
 Text(8.873999542284635, 485.9454545454546, 'X[6] <= -0.55\ngini = 0.081\nsamples = 2926\nvalue = [2802, 
124]'),
 Text(6.156328868489475, 469.4727272727273, 'X[1] <= -1.028\ngini = 0.001\nsamples = 1627\nvalue = 
[1626, 1]'),
 Text(5.815788637216073, 453.0, 'gini = 0.0\nsamples = 1603\nvalue = [1603, 0]'),
 Text(6.496869099762877, 453.0, 'X[1] <= -1.005\ngini = 0.08\nsamples = 24\nvalue = [23, 1]'),
 Text(6.156328868489475, 436.52727272727276, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(6.83740933103628, 436.52727272727276, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
 Text(11.591670216079795, 469.4727272727273, 'X[8] <= 0.228\ngini = 0.171\nsamples = 1299\nvalue = 
[1176, 123]'),
 Text(11.251129984806392, 453.0, 'X[6] <= 0.332\ngini = 0.289\nsamples = 701\nvalue = [578, 123]'),
 Text(7.518489793583084, 436.52727272727276, 'X[1] <= -1.128\ngini = 0.394\nsamples = 411\nvalue = [300, 
111]'),
 Text(4.820772648964101, 420.05454545454546, 'X[5] <= -0.903\ngini = 0.351\nsamples = 344\nvalue = [266, 
78]'),
 Text(2.319930325550053, 403.5818181818182, 'X[5] <= -1.08\ngini = 0.442\nsamples = 191\nvalue = [128, 63
]'),
 Text(0.6810804625468045, 387.1090909090909, 'X[6] <= 0.185\ngini = 0.19\nsamples = 47\nvalue = [42, 
5]'),
 Text(0.34054023127340227, 370.6363636363636, 'gini = 0.0\nsamples = 37\nvalue = [37, 0]'),
 Text(1.0216206938202068, 370.6363636363636, 'X[10] <= 0.084\ngini = 0.5\nsamples = 10\nvalue = [5, 
5]'),
 Text(0.6810804625468045, 354.1636363636364, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(1.362160925093609, 354.1636363636364, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(3.9587801885533014, 387.1090909090909, 'X[1] <= -1.728\ngini = 0.481\nsamples = 144\nvalue = [86, 
58]'),
 Text(2.383781618913816, 370.6363636363636, 'X[5] <= -1.079\ngini = 0.191\nsamples = 28\nvalue = [25, 
3]'),
 Text(2.0432413876404136, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(2.724321850187218, 354.1636363636364, 'X[5] <= -1.016\ngini = 0.137\nsamples = 27\nvalue = [25, 
2]'),
 Text(2.383781618913816, 337.69090909090914, 'X[10] <= 0.084\ngini = 0.375\nsamples = 8\nvalue = [6, 2]'
),
 Text(2.0432413876404136, 321.21818181818185, 'X[5] <= -1.054\ngini = 0.444\nsamples = 3\nvalue = [1, 2]
'),
 Text(1.7027011563670114, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(2.383781618913816, 304.74545454545455, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(2.724321850187218, 321.21818181818185, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(3.0648620814606202, 337.69090909090914, 'gini = 0.0\nsamples = 19\nvalue = [19, 0]'),
 Text(5.533778758192787, 370.6363636363636, 'X[7] <= 1.643\ngini = 0.499\nsamples = 116\nvalue = [61, 
55]'),
 Text(5.193238526919385, 354.1636363636364, 'X[1] <= -1.13\ngini = 0.499\nsamples = 105\nvalue = [50, 
55]'),
 Text(4.8526982956459825, 337.69090909090914, 'X[1] <= -1.22\ngini = 0.491\nsamples = 97\nvalue = [42, 5
5]'),
 Text(4.5121580643725805, 321.21818181818185, 'X[4] <= -0.534\ngini = 0.499\nsamples = 87\nvalue = [42, 
45]'),
 Text(3.0648620814606202, 304.74545454545455, 'X[10] <= 0.084\ngini = 0.482\nsamples = 74\nvalue = [30, 
```

```
44]'),
 Text(2.2135115032771147, 288.2727272727273, 'X[0] <= 0.411\ngini = 0.34\nsamples = 23\nvalue = [5,
18]'),
 Text(1.8729712720037126, 271.8, 'X[1] <= -1.329\ngini = 0.1\nsamples = 19\nvalue = [1, 18]'),
 Text(1.5324310407303101, 255.32727272727277, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(2.2135115032771147, 255.32727272727277, 'X[1] <= -1.305\ngini = 0.444\nsamples = 3\nvalue = [1, 2]
'),
 Text(1.8729712720037126, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(2.554051734550517, 238.85454545454547, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(2.554051734550517, 271.8, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(3.9162126596441262, 288.2727272727273, 'X[0] <= 0.411\ngini = 0.5\nsamples = 51\nvalue = [25,
26]'),
 Text(3.2351321970973217, 271.8, 'X[5] <= -0.909\ngini = 0.124\nsamples = 15\nvalue = [14, 1]'),
 Text(2.894591965823919, 255.32727272727277, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(3.5756724283707237, 255.32727272727277, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(4.597293122190931, 271.8, 'X[1] <= -1.614\ngini = 0.424\nsamples = 36\nvalue = [11, 25]'),
 Text(4.256752890917529, 255.32727272727277, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(4.937833353464333, 255.32727272727277, 'X[5] <= -1.06\ngini = 0.367\nsamples = 33\nvalue = [8,
25]'),
 Text(4.597293122190931, 238.85454545454547, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(5.278373584737735, 238.85454545454547, 'X[6] <= -0.01\ngini = 0.278\nsamples = 30\nvalue = [5,
25]'),
 Text(4.937833353464333, 222.38181818181823, 'X[1] <= -1.544\ngini = 0.191\nsamples = 28\nvalue = [3,
25]'),
 Text(4.256752890917529, 205.90909090909093, 'X[6] <= -0.483\ngini = 0.48\nsamples = 5\nvalue = [2,
3]'),
 Text(3.9162126596441262, 189.43636363636364, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(4.597293122190931, 189.43636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(5.618913816011138, 205.90909090909093, 'X[6] <= -0.491\ngini = 0.083\nsamples = 23\nvalue = [1,
22]'),
 Text(5.278373584737735, 189.43636363636364, 'X[1] <= -1.49\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(4.937833353464333, 172.9636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(5.618913816011138, 172.9636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(5.95945404728454, 189.43636363636364, 'gini = 0.0\nsamples = 21\nvalue = [0, 21]'),
 Text(5.618913816011138, 222.38181818181823, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(5.95945404728454, 304.74545454545455, 'X[5] <= -1.061\ngini = 0.142\nsamples = 13\nvalue = [12,
1]'),
 Text(5.618913816011138, 288.2727272727273, 'X[7] <= 0.129\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(5.278373584737735, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(5.95945404728454, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(6.299994278557942, 288.2727272727273, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(5.193238526919385, 321.21818181818185, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(5.533778758192787, 337.69090909090914, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(5.87431898946619, 354.1636363636364, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(7.321614972378149, 403.5818181818182, 'X[1] <= -1.401\ngini = 0.177\nsamples = 153\nvalue = [138, 1
5]'),
 Text(6.981074741104747, 387.1090909090909, 'gini = 0.0\nsamples = 86\nvalue = [86, 0]'),
 Text(7.662155203651551, 387.1090909090909, 'X[1] <= -1.365\ngini = 0.348\nsamples = 67\nvalue = [52,
15]'),
 Text(7.321614972378149, 370.6363636363636, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(8.002695434924954, 370.6363636363636, 'X[4] <= -0.534\ngini = 0.231\nsamples = 60\nvalue = [52,
8]'),
 Text(7.662155203651551, 354.1636363636364, 'X[1] <= -1.182\ngini = 0.391\nsamples = 30\nvalue = [22,
8]'),
 Text(7.321614972378149, 337.69090909090914, 'X[5] <= -0.774\ngini = 0.48\nsamples = 20\nvalue = [12,
8]'),
 Text(6.981074741104747, 321.21818181818185, 'X[5] <= -0.86\ngini = 0.473\nsamples = 13\nvalue = [5,
8]'),
 Text(6.640534509831344, 304.74545454545455, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(7.321614972378149, 304.74545454545455, 'X[1] <= -1.337\ngini = 0.32\nsamples = 10\nvalue = [2,
8]'),
 Text(6.981074741104747, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(7.662155203651551, 288.2727272727273, 'X[6] <= 0.074\ngini = 0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(7.321614972378149, 271.8, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(8.002695434924954, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(7.662155203651551, 321.21818181818185, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(8.002695434924954, 337.69090909090914, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(8.343235666198355, 354.1636363636364, 'gini = 0.0\nsamples = 30\nvalue = [30, 0]'),
 Text(10.216206938202069, 420.05454545454546, 'X[4] <= -0.534\ngini = 0.5\nsamples = 67\nvalue = [34,
33]'),
 Text(9.364856360018562, 403.5818181818182, 'X[10] <= 0.084\ngini = 0.358\nsamples = 30\nvalue = [7,
23]'),
 Text(9.024316128745161, 387.1090909090909, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(9.705396591291965, 387.1090909090909, 'X[1] <= -0.991\ngini = 0.252\nsamples = 27\nvalue = [4,
23]'),
 Text(9.364856360018562, 370.6363636363636, 'X[7] <= 1.643\ngini = 0.147\nsamples = 25\nvalue = [2,
23]'),
```

```
 Text(9.024316128745161, 354.1636363636364, 'X[6] <= -0.383\ngini = 0.08\nsamples = 24\nvalue = [1, 23]')
,
 Text(8.683775897471758, 337.69090909090914, 'X[5] <= -0.988\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(8.343235666198355, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(9.024316128745161, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(9.364856360018562, 337.69090909090914, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(9.705396591291965, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(10.045936822565366, 370.6363636363636, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(11.067557516385573, 403.5818181818182, 'X[1] <= -1.087\ngini = 0.394\nsamples = 37\nvalue = [27,
10]'),
 Text(10.727017285112172, 387.1090909090909, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(11.408097747658976, 387.1090909090909, 'X[6] <= 0.052\ngini = 0.298\nsamples = 33\nvalue = [27,
6]'),
 Text(10.727017285112172, 370.6363636363636, 'X[6] <= -0.077\ngini = 0.08\nsamples = 24\nvalue = [23,
1]'),
 Text(10.38647705383877, 354.1636363636364, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),
 Text(11.067557516385573, 354.1636363636364, 'X[6] <= -0.052\ngini = 0.278\nsamples = 6\nvalue = [5,
1]'),
 Text(10.727017285112172, 337.69090909090914, 'X[2] <= 1.529\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(10.38647705383877, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(11.067557516385573, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(11.408097747658976, 337.69090909090914, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(12.08917821020578, 370.6363636363636, 'X[6] <= 0.131\ngini = 0.494\nsamples = 9\nvalue = [4, 5]'),
 Text(11.74863797893238, 354.1636363636364, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(12.429718441479183, 354.1636363636364, 'X[1] <= -1.06\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(12.08917821020578, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(12.770258672752584, 337.69090909090914, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(14.9837701760297, 436.52727272727276, 'X[5] <= -0.737\ngini = 0.079\nsamples = 290\nvalue = [278,
12]'),
 Text(14.132419597846194, 420.05454545454546, 'X[5] <= -1.115\ngini = 0.061\nsamples = 285\nvalue =
[276, 9]'),
 Text(13.45133913529939, 403.5818181818182, 'X[5] <= -1.12\ngini = 0.287\nsamples = 46\nvalue = [38,
8]'),
 Text(13.110798904025987, 387.1090909090909, 'gini = 0.0\nsamples = 35\nvalue = [35, 0]'),
 Text(13.791879366572791, 387.1090909090909, 'X[10] <= 0.084\ngini = 0.397\nsamples = 11\nvalue = [3, 8]
'),
 Text(13.45133913529939, 370.6363636363636, 'X[0] <= 0.411\ngini = 0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(13.110798904025987, 354.1636363636364, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(13.791879366572791, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(14.132419597846194, 370.6363636363636, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(14.813500060392998, 403.5818181818182, 'X[5] <= -0.742\ngini = 0.008\nsamples = 239\nvalue = [238,
1]'),
 Text(14.472959829119597, 387.1090909090909, 'gini = 0.0\nsamples = 230\nvalue = [230, 0]'),
 Text(15.154040291666401, 387.1090909090909, 'X[5] <= -0.742\ngini = 0.198\nsamples = 9\nvalue = [8,
1]'),
 Text(14.813500060392998, 370.6363636363636, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(15.494580522939803, 370.6363636363636, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(15.835120754213206, 420.05454545454546, 'X[5] <= -0.736\ngini = 0.48\nsamples = 5\nvalue = [2,
3]'),
 Text(15.494580522939803, 403.5818181818182, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(16.17566098548661, 403.5818181818182, 'X[5] <= -0.734\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(15.835120754213206, 387.1090909090909, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(16.51620121676001, 387.1090909090909, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(11.932210447353198, 453.0, 'gini = 0.0\nsamples = 598\nvalue = [598, 0]'),
 Text(17.197281679306816, 485.9454545454546, 'X[1] <= -1.036\ngini = 0.003\nsamples = 7797\nvalue =
[7785, 12]'),
 Text(16.17566098548661, 469.4727272727273, 'X[1] <= -1.078\ngini = 0.001\nsamples = 7524\nvalue =
[7521, 3]'),
 Text(15.835120754213206, 453.0, 'gini = 0.0\nsamples = 7252\nvalue = [7252, 0]'),
 Text(16.51620121676001, 453.0, 'X[1] <= -1.046\ngini = 0.022\nsamples = 272\nvalue = [269, 3]'),
 Text(16.17566098548661, 436.52727272727276, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(16.85674144803341, 436.52727272727276, 'X[5] <= 1.474\ngini = 0.007\nsamples = 270\nvalue = [269,
1]'),
 Text(16.51620121676001, 420.05454545454546, 'gini = 0.0\nsamples = 253\nvalue = [253, 0]'),
 Text(17.197281679306816, 420.05454545454546, 'X[6] <= -0.516\ngini = 0.111\nsamples = 17\nvalue = [16,
1]'),
 Text(16.85674144803341, 403.5818181818182, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(17.537821910580217, 403.5818181818182, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(18.218902373127023, 469.4727272727273, 'X[1] <= -0.991\ngini = 0.064\nsamples = 273\nvalue = [264,
9]'),
 Text(17.878362141853618, 453.0, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(18.559442604400424, 453.0, 'X[5] <= -0.561\ngini = 0.008\nsamples = 265\nvalue = [264, 1]'),
 Text(18.218902373127023, 436.52727272727276, 'X[5] <= -0.565\ngini = 0.034\nsamples = 57\nvalue = [56,
1]'),
 Text(17.878362141853618, 420.05454545454546, 'gini = 0.0\nsamples = 56\nvalue = [56, 0]'),
 Text(18.559442604400424, 420.05454545454546, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
```

```
 Text(18.899982835673825, 436.52727272727276, 'gini = 0.0\nsamples = 208\nvalue = [208, 0]'),
 Text(69.25380492038057, 502.41818181818184, 'X[1] <= -0.53\ngini = 0.375\nsamples = 8838\nvalue =
[6628, 2210]'),
 Text(36.96989885761874, 485.9454545454546, 'X[4] <= -0.534\ngini = 0.284\nsamples = 4725\nvalue =
[3916, 809]'),
 Text(27.732745084327696, 469.4727272727273, 'X[5] <= -0.533\ngini = 0.472\nsamples = 1050\nvalue =
[649, 401]'),
 Text(23.369573371137232, 453.0, 'X[7] <= 1.643\ngini = 0.49\nsamples = 652\nvalue = [279, 373]'),
 Text(23.029033139863827, 436.52727272727276, 'X[5] <= -0.64\ngini = 0.467\nsamples = 594\nvalue = [221,
373]'),
 Text(19.24052306694723, 420.05454545454546, 'X[11] <= 1.02\ngini = 0.5\nsamples = 364\nvalue = [180,
184]'),
 Text(18.899982835673825, 403.5818181818182, 'X[1] <= -0.531\ngini = 0.496\nsamples = 339\nvalue = [155,
184]'),
 Text(18.559442604400424, 387.1090909090909, 'X[5] <= -1.08\ngini = 0.49\nsamples = 322\nvalue = [138, 18
4]'),
 Text(17.537821910580217, 370.6363636363636, 'X[7] <= 0.129\ngini = 0.236\nsamples = 22\nvalue = [19,
3]'),
 Text(17.197281679306816, 354.1636363636364, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(17.878362141853618, 354.1636363636364, 'X[5] <= -1.123\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(17.537821910580217, 337.69090909090914, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(18.218902373127023, 337.69090909090914, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(19.58106329822063, 370.6363636363636, 'X[2] <= 1.529\ngini = 0.479\nsamples = 300\nvalue = [119, 18
1]'),
 Text(19.24052306694723, 354.1636363636364, 'X[6] <= -1.313\ngini = 0.47\nsamples = 291\nvalue = [110, 18
1]'),
 Text(18.899982835673825, 337.69090909090914, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(19.58106329822063, 337.69090909090914, 'X[1] <= -0.9\ngini = 0.461\nsamples = 283\nvalue = [102,
181]'),
 Text(17.112146621488463, 321.21818181818185, 'X[5] <= -0.76\ngini = 0.317\nsamples = 71\nvalue = [14, 5
7]'),
 Text(16.43106615894166, 304.74545454545455, 'X[1] <= -0.94\ngini = 0.455\nsamples = 20\nvalue = [13,
7]'),
 Text(16.090525927668256, 288.2727272727273, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(16.771606390215062, 288.2727272727273, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(17.79322708403527, 304.74545454545455, 'X[5] <= -0.664\ngini = 0.038\nsamples = 51\nvalue = [1,
50]'),
 Text(17.452686852761868, 288.2727272727273, 'gini = 0.0\nsamples = 45\nvalue = [0, 45]'),
 Text(18.13376731530867, 288.2727272727273, 'X[5] <= -0.661\ngini = 0.278\nsamples = 6\nvalue = [1,
5]'),
 Text(17.79322708403527, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(18.47430754658207, 271.8, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(22.049979974952798, 321.21818181818185, 'X[5] <= -0.906\ngini = 0.486\nsamples = 212\nvalue = [88,
124]'),
 Text(20.347278818585785, 304.74545454545455, 'X[1] <= -0.852\ngini = 0.334\nsamples = 104\nvalue = [22,
82]'),
 Text(19.49592824040228, 288.2727272727273, 'X[5] <= -0.913\ngini = 0.32\nsamples = 15\nvalue = [12, 3]')
,
 Text(19.155388009128878, 271.8, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
 Text(19.836468471675683, 271.8, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(21.19862939676929, 288.2727272727273, 'X[6] <= 0.426\ngini = 0.199\nsamples = 89\nvalue = [10,
79]'),
 Text(20.517548934222486, 271.8, 'X[1] <= -0.705\ngini = 0.121\nsamples = 77\nvalue = [5, 72]'),
 Text(20.177008702949085, 255.32727272727277, 'X[6] <= 0.1\ngini = 0.224\nsamples = 39\nvalue = [5,
34]'),
 Text(19.836468471675683, 238.85454545454547, 'X[5] <= -1.003\ngini = 0.149\nsamples = 37\nvalue = [3, 3
4]'),
 Text(19.49592824040228, 222.38181818181823, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(20.177008702949085, 222.38181818181823, 'X[7] <= 0.129\ngini = 0.105\nsamples = 36\nvalue = [2, 34
]'),
 Text(19.836468471675683, 205.90909090909093, 'X[1] <= -0.81\ngini = 0.32\nsamples = 10\nvalue = [2,
8]'),
 Text(19.49592824040228, 189.43636363636364, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(20.177008702949085, 189.43636363636364, 'X[5] <= -0.96\ngini = 0.48\nsamples = 5\nvalue = [2,
3]'),
 Text(19.836468471675683, 172.9636363636364, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(20.517548934222486, 172.9636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(20.517548934222486, 205.90909090909093, 'gini = 0.0\nsamples = 26\nvalue = [0, 26]'),
 Text(20.517548934222486, 238.85454545454547, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(20.85808916549589, 255.32727272727277, 'gini = 0.0\nsamples = 38\nvalue = [0, 38]'),
 Text(21.879709859316097, 271.8, 'X[5] <= -1.055\ngini = 0.486\nsamples = 12\nvalue = [5, 7]'),
 Text(21.539169628042693, 255.32727272727277, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(22.2202500905895, 255.32727272727277, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(23.752681131319807, 304.74545454545455, 'X[1] <= -0.854\ngini = 0.475\nsamples = 108\nvalue = [66,
42]'),
 Text(22.901330553136305, 288.2727272727273, 'X[1] <= -0.899\ngini = 0.403\nsamples = 25\nvalue = [7,
18]'),
```

```
 Text(22.5607903218629, 271.8, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(23.241870784409706, 271.8, 'gini = 0.0\nsamples = 18\nvalue = [0, 18]'),
 Text(24.604031709503314, 288.2727272727273, 'X[6] <= 1.585\ngini = 0.411\nsamples = 83\nvalue = [59,
24]'),
 Text(23.922951246956508, 271.8, 'X[1] <= -0.572\ngini = 0.313\nsamples = 67\nvalue = [54, 13]'),
 Text(23.582411015683107, 255.32727272727277, 'X[10] <= 0.084\ngini = 0.264\nsamples = 64\nvalue = [54,
10]'),
 Text(23.241870784409706, 238.85454545454547, 'X[6] <= 0.07\ngini = 0.491\nsamples = 23\nvalue = [13,
10]'),
 Text(22.2202500905895, 222.38181818181823, 'X[7] <= 0.129\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),
 Text(21.879709859316097, 205.90909090909093, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(22.5607903218629, 205.90909090909093, 'X[5] <= -0.745\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(22.2202500905895, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(22.901330553136305, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(24.263491478229913, 222.38181818181823, 'X[7] <= 0.129\ngini = 0.426\nsamples = 13\nvalue = [4, 9]
'),
 Text(23.922951246956508, 205.90909090909093, 'X[6] <= 0.499\ngini = 0.298\nsamples = 11\nvalue = [2, 9]
'),
 Text(23.582411015683107, 189.43636363636364, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(24.263491478229913, 189.43636363636364, 'X[6] <= 0.785\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'
),
 Text(23.922951246956508, 172.9636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(24.604031709503314, 172.9636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(24.604031709503314, 205.90909090909093, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(23.922951246956508, 238.85454545454547, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),
 Text(24.263491478229913, 255.32727272727277, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(25.28511217205012, 271.8, 'X[5] <= -0.704\ngini = 0.43\nsamples = 16\nvalue = [5, 11]'),
 Text(24.944571940776715, 255.32727272727277, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(25.62565240332352, 255.32727272727277, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
 Text(19.921603529494032, 354.1636363636364, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(19.24052306694723, 387.1090909090909, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(19.58106329822063, 403.5818181818182, 'gini = 0.0\nsamples = 25\nvalue = [25, 0]'),
 Text(26.81754321278043, 420.05454545454546, 'X[1] <= -0.579\ngini = 0.293\nsamples = 230\nvalue = [41,
189]'),
 Text(25.11484205641342, 403.5818181818182, 'X[6] <= -0.857\ngini = 0.244\nsamples = 218\nvalue = [31, 18
7]'),
 Text(24.774301825140014, 387.1090909090909, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(25.45538228768682, 387.1090909090909, 'X[6] <= -0.277\ngini = 0.208\nsamples = 212\nvalue = [25, 18
7]'),
 Text(24.093221362593212, 370.6363636363636, 'X[1] <= -0.626\ngini = 0.049\nsamples = 119\nvalue = [3,
116]'),
 Text(23.752681131319807, 354.1636363636364, 'X[6] <= -0.816\ngini = 0.033\nsamples = 118\nvalue = [2,
116]'),
 Text(23.071600668773005, 337.69090909090914, 'X[11] <= 1.02\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(22.7310604374996, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(23.412140900046406, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(24.433761593866613, 337.69090909090914, 'X[0] <= 0.411\ngini = 0.017\nsamples = 116\nvalue = [1, 1
15]'),
 Text(24.093221362593212, 321.21818181818185, 'gini = 0.0\nsamples = 113\nvalue = [0, 113]'),
 Text(24.774301825140014, 321.21818181818185, 'X[6] <= -0.432\ngini = 0.444\nsamples = 3\nvalue = [1, 2]
'),
 Text(24.433761593866613, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(25.11484205641342, 304.74545454545455, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(24.433761593866613, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(26.81754321278043, 370.6363636363636, 'X[6] <= 1.527\ngini = 0.361\nsamples = 93\nvalue = [22,
71]'),
 Text(26.136462750233623, 354.1636363636364, 'X[6] <= -0.253\ngini = 0.1\nsamples = 19\nvalue = [18,
1]'),
 Text(25.79592251896022, 337.69090909090914, 'X[5] <= -0.584\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(25.45538228768682, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(26.136462750233623, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(26.477002981507027, 337.69090909090914, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(27.498623675327234, 354.1636363636364, 'X[6] <= 2.579\ngini = 0.102\nsamples = 74\nvalue = [4,
70]'),
 Text(27.15808344405383, 337.69090909090914, 'X[1] <= -0.924\ngini = 0.054\nsamples = 72\nvalue = [2,
70]'),
 Text(26.81754321278043, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(27.498623675327234, 321.21818181818185, 'X[8] <= 0.228\ngini = 0.028\nsamples = 71\nvalue = [1, 70
]'),
 Text(27.15808344405383, 304.74545454545455, 'X[5] <= -0.608\ngini = 0.375\nsamples = 4\nvalue = [1,
3]'),
 Text(26.81754321278043, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(27.498623675327234, 288.2727272727273, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(27.839163906600636, 304.74545454545455, 'gini = 0.0\nsamples = 67\nvalue = [0, 67]'),
 Text(27.839163906600636, 337.69090909090914, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(28.52024436914744, 403.5818181818182, 'X[5] <= -0.535\ngini = 0.278\nsamples = 12\nvalue = [10,
2]'),
```

```
 Text(28.179704137874037, 387.1090909090909, 'X[6] <= 1.016\ngini = 0.165\nsamples = 11\nvalue = [10,
1]'),
 Text(27.839163906600636, 370.6363636363636, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(28.52024436914744, 370.6363636363636, 'X[6] <= 1.97\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(28.179704137874037, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(28.860784600420843, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(28.860784600420843, 387.1090909090909, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(23.710113602410633, 436.52727272727276, 'gini = 0.0\nsamples = 58\nvalue = [58, 0]'),
 Text(32.095916797518164, 453.0, 'X[6] <= 2.292\ngini = 0.131\nsamples = 398\nvalue = [370, 28]'),
 Text(30.563485756787856, 436.52727272727276, 'X[5] <= 1.264\ngini = 0.034\nsamples = 348\nvalue = [342,
6]'),
 Text(29.54186506296765, 420.05454545454546, 'X[1] <= -0.563\ngini = 0.006\nsamples = 318\nvalue = [317,
1]'),
 Text(29.201324831694244, 403.5818181818182, 'gini = 0.0\nsamples = 287\nvalue = [287, 0]'),
 Text(29.88240529424105, 403.5818181818182, 'X[1] <= -0.54\ngini = 0.062\nsamples = 31\nvalue = [30,
1]'),
 Text(29.54186506296765, 387.1090909090909, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(30.22294552551445, 387.1090909090909, 'gini = 0.0\nsamples = 30\nvalue = [30, 0]'),
 Text(31.58510645060806, 420.05454545454546, 'X[5] <= 1.494\ngini = 0.278\nsamples = 30\nvalue = [25,
5]'),
 Text(31.244566219334658, 403.5818181818182, 'X[2] <= 1.529\ngini = 0.278\nsamples = 6\nvalue = [1,
5]'),
 Text(30.904025988061257, 387.1090909090909, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(31.58510645060806, 387.1090909090909, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(31.925646681881464, 403.5818181818182, 'gini = 0.0\nsamples = 24\nvalue = [24, 0]'),
 Text(33.62834783824847, 436.52727272727276, 'X[5] <= -0.313\ngini = 0.493\nsamples = 50\nvalue = [28,
22]'),
 Text(33.28780760697507, 420.05454545454546, 'X[11] <= 1.02\ngini = 0.26\nsamples = 26\nvalue = [4,
22]'),
 Text(32.94726737570167, 403.5818181818182, 'X[5] <= -0.353\ngini = 0.153\nsamples = 24\nvalue = [2,
22]'),
 Text(32.26618691315487, 387.1090909090909, 'X[5] <= -0.471\ngini = 0.091\nsamples = 21\nvalue = [1,
20]'),
 Text(31.925646681881464, 370.6363636363636, 'X[6] <= 2.868\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(31.58510645060806, 354.1636363636364, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(32.26618691315487, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(32.60672714442827, 370.6363636363636, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(33.62834783824847, 387.1090909090909, 'X[6] <= 3.364\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(33.28780760697507, 370.6363636363636, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(33.968888069521874, 370.6363636363636, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(33.62834783824847, 403.5818181818182, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(33.968888069521874, 420.05454545454546, 'gini = 0.0\nsamples = 24\nvalue = [24, 0]'),
 Text(46.20705263090977, 469.4727272727273, 'X[1] <= -0.945\ngini = 0.197\nsamples = 3675\nvalue =
[3267, 408]'),
 Text(45.86651239963637, 453.0, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(46.54759286218317, 453.0, 'X[8] <= 0.228\ngini = 0.191\nsamples = 3659\nvalue = [3267, 392]'),
 Text(39.46009929880549, 436.52727272727276, 'X[1] <= -0.853\ngini = 0.263\nsamples = 1917\nvalue =
[1619, 298]'),
 Text(37.11888520880085, 420.05454545454546, 'X[5] <= -0.846\ngini = 0.096\nsamples = 457\nvalue = [434,
23]'),
 Text(34.649968532068684, 403.5818181818182, 'X[1] <= -0.944\ngini = 0.329\nsamples = 77\nvalue = [61,
16]'),
 Text(34.309428300795275, 387.1090909090909, 'gini = 0.0\nsamples = 24\nvalue = [24, 0]'),
 Text(34.990508763342085, 387.1090909090909, 'X[1] <= -0.906\ngini = 0.422\nsamples = 53\nvalue = [37,
16]'),
 Text(34.649968532068684, 370.6363636363636, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(35.331048994615486, 370.6363636363636, 'X[5] <= -0.922\ngini = 0.315\nsamples = 46\nvalue = [37,
9]'),
 Text(34.309428300795275, 354.1636363636364, 'X[7] <= 0.129\ngini = 0.071\nsamples = 27\nvalue = [26,
1]'),
 Text(33.968888069521874, 337.69090909090914, 'gini = 0.0\nsamples = 22\nvalue = [22, 0]'),
 Text(34.649968532068684, 337.69090909090914, 'X[6] <= -0.424\ngini = 0.32\nsamples = 5\nvalue = [4,
1]'),
 Text(34.309428300795275, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(34.990508763342085, 321.21818181818185, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(36.35266968843569, 354.1636363636364, 'X[5] <= -0.882\ngini = 0.488\nsamples = 19\nvalue = [11,
8]'),
 Text(36.01212945716229, 337.69090909090914, 'X[7] <= 0.886\ngini = 0.397\nsamples = 11\nvalue = [3,
8]'),
 Text(35.67158922588889, 321.21818181818185, 'X[6] <= -0.185\ngini = 0.198\nsamples = 9\nvalue = [1,
8]'),
 Text(35.331048994615486, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(36.01212945716229, 304.74545454545455, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(36.35266968843569, 321.21818181818185, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(36.6932099197091, 337.69090909090914, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(39.58780188553301, 403.5818181818182, 'X[5] <= 1.478\ngini = 0.036\nsamples = 380\nvalue = [373, 7]
'),
```

```
 Text(38.73645130734951, 387.1090909090909, 'X[6] <= -1.229\ngini = 0.016\nsamples = 366\nvalue = [363, 3
]'),
 Text(38.0553708448027, 370.6363636363636, 'X[6] <= -1.237\ngini = 0.18\nsamples = 20\nvalue = [18, 2]'),
 Text(37.7148306135293, 354.1636363636364, 'X[5] <= 0.109\ngini = 0.1\nsamples = 19\nvalue = [18, 1]'),
 Text(37.3742903822559, 337.69090909090914, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(38.0553708448027, 337.69090909090914, 'X[5] <= 0.426\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(37.7148306135293, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(38.3959110760761, 321.21818181818185, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(38.3959110760761, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(39.417531769896314, 370.6363636363636, 'X[5] <= -0.641\ngini = 0.006\nsamples = 346\nvalue = [345,
1]'),
 Text(39.07699153862291, 354.1636363636364, 'X[5] <= -0.642\ngini = 0.024\nsamples = 81\nvalue = [80,
1]'),
 Text(38.73645130734951, 337.69090909090914, 'gini = 0.0\nsamples = 80\nvalue = [80, 0]'),
 Text(39.417531769896314, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(39.758072001169715, 354.1636363636364, 'gini = 0.0\nsamples = 265\nvalue = [265, 0]'),
 Text(40.43915246371652, 387.1090909090909, 'X[5] <= 1.554\ngini = 0.408\nsamples = 14\nvalue = [10,
4]'),
 Text(40.09861223244312, 370.6363636363636, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(40.77969269498992, 370.6363636363636, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(41.80131338881013, 420.05454545454546, 'X[1] <= -0.807\ngini = 0.306\nsamples = 1460\nvalue =
[1185, 275]'),
 Text(41.46077315753673, 403.5818181818182, 'gini = 0.0\nsamples = 32\nvalue = [0, 32]'),
 Text(42.14185362008353, 403.5818181818182, 'X[1] <= -0.53\ngini = 0.282\nsamples = 1428\nvalue = [1185,
243]'),
 Text(41.80131338881013, 387.1090909090909, 'X[1] <= -0.575\ngini = 0.317\nsamples = 1231\nvalue = [988,
243]'),
 Text(41.46077315753673, 370.6363636363636, 'X[1] <= -0.806\ngini = 0.283\nsamples = 1191\nvalue = [988,
203]'),
 Text(40.48171999262569, 354.1636363636364, 'X[5] <= -0.232\ngini = 0.011\nsamples = 183\nvalue = [182, 1
]'),
 Text(40.14117976135229, 337.69090909090914, 'gini = 0.0\nsamples = 117\nvalue = [117, 0]'),
 Text(40.8222602238991, 337.69090909090914, 'X[5] <= -0.231\ngini = 0.03\nsamples = 66\nvalue = [65,
1]'),
 Text(40.48171999262569, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(41.1628004551725, 321.21818181818185, 'gini = 0.0\nsamples = 65\nvalue = [65, 0]'),
 Text(42.43982632244776, 354.1636363636364, 'X[1] <= -0.761\ngini = 0.32\nsamples = 1008\nvalue = [806, 2
02]'),
 Text(42.099286091174356, 337.69090909090914, 'gini = 0.0\nsamples = 38\nvalue = [0, 38]'),
 Text(42.78036655372116, 337.69090909090914, 'X[1] <= -0.668\ngini = 0.281\nsamples = 970\nvalue = [806,
164]'),
 Text(41.843880917719304, 321.21818181818185, 'X[1] <= -0.668\ngini = 0.187\nsamples = 536\nvalue =
[480, 56]'),
 Text(41.5033406864459, 304.74545454545455, 'X[1] <= -0.714\ngini = 0.264\nsamples = 358\nvalue = [302,
56]'),
 Text(41.1628004551725, 288.2727272727273, 'X[1] <= -0.716\ngini = 0.136\nsamples = 326\nvalue = [302, 24
]'),
 Text(40.8222602238991, 271.8, 'X[1] <= -0.76\ngini = 0.234\nsamples = 177\nvalue = [153, 24]'),
 Text(40.48171999262569, 255.32727272727277, 'X[5] <= 2.061\ngini = 0.013\nsamples = 154\nvalue = [153,
1]'),
 Text(40.14117976135229, 238.85454545454547, 'gini = 0.0\nsamples = 150\nvalue = [150, 0]'),
 Text(40.8222602238991, 238.85454545454547, 'X[5] <= 2.098\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(40.48171999262569, 222.38181818181823, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(41.1628004551725, 222.38181818181823, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(41.1628004551725, 255.32727272727277, 'gini = 0.0\nsamples = 23\nvalue = [0, 23]'),
 Text(41.5033406864459, 271.8, 'gini = 0.0\nsamples = 149\nvalue = [149, 0]'),
 Text(41.843880917719304, 288.2727272727273, 'gini = 0.0\nsamples = 32\nvalue = [0, 32]'),
 Text(42.184421148992705, 304.74545454545455, 'gini = 0.0\nsamples = 178\nvalue = [178, 0]'),
 Text(43.716852189723014, 321.21818181818185, 'X[1] <= -0.623\ngini = 0.374\nsamples = 434\nvalue =
[326, 108]'),
 Text(43.37631195844961, 304.74545454545455, 'gini = 0.0\nsamples = 41\nvalue = [0, 41]'),
 Text(44.05739242099642, 304.74545454545455, 'X[6] <= 2.534\ngini = 0.283\nsamples = 393\nvalue = [326,
67]'),
 Text(42.865501611539514, 288.2727272727273, 'X[1] <= -0.622\ngini = 0.22\nsamples = 358\nvalue = [313, 4
5]'),
 Text(42.184421148992705, 271.8, 'X[0] <= 0.411\ngini = 0.025\nsamples = 156\nvalue = [154, 2]'),
 Text(41.843880917719304, 255.32727272727277, 'gini = 0.0\nsamples = 100\nvalue = [100, 0]'),
 Text(42.524961380266106, 255.32727272727277, 'X[5] <= -0.579\ngini = 0.069\nsamples = 56\nvalue = [54,
2]'),
 Text(42.184421148992705, 238.85454545454547, 'X[5] <= -0.606\ngini = 0.133\nsamples = 28\nvalue = [26,
2]'),
 Text(41.843880917719304, 222.38181818181823, 'X[5] <= -0.754\ngini = 0.071\nsamples = 27\nvalue = [26,
1]'),
 Text(41.5033406864459, 205.90909090909093, 'gini = 0.0\nsamples = 19\nvalue = [19, 0]'),
 Text(42.184421148992705, 205.90909090909093, 'X[5] <= -0.736\ngini = 0.219\nsamples = 8\nvalue = [7, 1]
'),
 Text(41.843880917719304, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
```

```
 Text(42.524961380266106, 189.43636363636364, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(42.524961380266106, 222.38181818181823, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(42.865501611539514, 238.85454545454547, 'gini = 0.0\nsamples = 28\nvalue = [28, 0]'),
 Text(43.54658207408632, 271.8, 'X[1] <= -0.576\ngini = 0.335\nsamples = 202\nvalue = [159, 43]'),
 Text(43.206041842812915, 255.32727272727277, 'gini = 0.0\nsamples = 39\nvalue = [0, 39]'),
 Text(43.88712230535972, 255.32727272727277, 'X[6] <= 2.131\ngini = 0.048\nsamples = 163\nvalue = [159,
4]'),
 Text(43.54658207408632, 238.85454545454547, 'gini = 0.0\nsamples = 151\nvalue = [151, 0]'),
 Text(44.22766253663312, 238.85454545454547, 'X[10] <= 0.084\ngini = 0.444\nsamples = 12\nvalue = [8, 4]
'),
 Text(43.88712230535972, 222.38181818181823, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(44.56820276790652, 222.38181818181823, 'X[5] <= -0.847\ngini = 0.49\nsamples = 7\nvalue = [3,
4]'),
 Text(44.22766253663312, 205.90909090909093, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(44.90874299917992, 205.90909090909093, 'X[7] <= 0.886\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(44.56820276790652, 189.43636363636364, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(45.24928323045333, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(45.24928323045333, 288.2727272727273, 'X[5] <= -0.652\ngini = 0.467\nsamples = 35\nvalue = [13,
22]'),
 Text(44.90874299917992, 271.8, 'X[5] <= -0.856\ngini = 0.337\nsamples = 28\nvalue = [6, 22]'),
 Text(44.56820276790652, 255.32727272727277, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(45.24928323045333, 255.32727272727277, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(45.58982346172673, 271.8, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(42.14185362008353, 370.6363636363636, 'gini = 0.0\nsamples = 40\nvalue = [0, 40]'),
 Text(42.48239385135693, 387.1090909090909, 'gini = 0.0\nsamples = 197\nvalue = [197, 0]'),
 Text(53.63508642556086, 436.52727272727276, 'X[1] <= -0.622\ngini = 0.102\nsamples = 1742\nvalue =
[1648, 94]'),
 Text(51.46414245119292, 420.05454545454546, 'X[5] <= 0.152\ngini = 0.048\nsamples = 1307\nvalue =
[1275, 32]'),
 Text(50.01684646828096, 403.5818181818182, 'X[3] <= 2.06\ngini = 0.02\nsamples = 1066\nvalue = [1055, 11
]'),
 Text(49.33576600573416, 387.1090909090909, 'X[6] <= 0.098\ngini = 0.019\nsamples = 1060\nvalue = [1050,
10]'),
 Text(48.99522577446075, 370.6363636363636, 'X[6] <= 0.09\ngini = 0.036\nsamples = 545\nvalue = [535, 10]
'),
 Text(48.65468554318735, 354.1636363636364, 'X[6] <= 0.043\ngini = 0.033\nsamples = 544\nvalue = [535, 9]
'),
 Text(47.633064849367145, 337.69090909090914, 'X[5] <= -0.956\ngini = 0.019\nsamples = 519\nvalue =
[514, 5]'),
 Text(46.611444155546934, 321.21818181818185, 'X[5] <= -0.957\ngini = 0.115\nsamples = 49\nvalue = [46,
3]'),
 Text(46.27090392427353, 304.74545454545455, 'X[5] <= -1.014\ngini = 0.08\nsamples = 48\nvalue = [46,
2]'),
 Text(45.93036369300013, 288.2727272727273, 'gini = 0.0\nsamples = 33\nvalue = [33, 0]'),
 Text(46.611444155546934, 288.2727272727273, 'X[1] <= -0.652\ngini = 0.231\nsamples = 15\nvalue = [13,
2]'),
 Text(46.27090392427353, 271.8, 'X[6] <= -0.911\ngini = 0.133\nsamples = 14\nvalue = [13, 1]'),
 Text(45.93036369300013, 255.32727272727277, 'X[6] <= -1.193\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(45.58982346172673, 238.85454545454547, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(46.27090392427353, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(46.611444155546934, 255.32727272727277, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(46.951984386820335, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(46.951984386820335, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(48.65468554318735, 321.21818181818185, 'X[5] <= -0.592\ngini = 0.008\nsamples = 470\nvalue = [468,
2]'),
 Text(48.31414531191395, 304.74545454545455, 'X[5] <= -0.593\ngini = 0.021\nsamples = 192\nvalue = [190,
2]'),
 Text(47.973605080640546, 288.2727272727273, 'X[5] <= -0.603\ngini = 0.01\nsamples = 191\nvalue = [190, 1
]'),
 Text(47.633064849367145, 271.8, 'gini = 0.0\nsamples = 184\nvalue = [184, 0]'),
 Text(48.31414531191395, 271.8, 'X[1] <= -0.784\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(47.973605080640546, 255.32727272727277, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(48.65468554318735, 255.32727272727277, 'X[1] <= -0.714\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(48.31414531191395, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(48.99522577446075, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(48.65468554318735, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(48.99522577446075, 304.74545454545455, 'gini = 0.0\nsamples = 278\nvalue = [278, 0]'),
 Text(49.67630623700756, 337.69090909090914, 'X[7] <= 1.643\ngini = 0.269\nsamples = 25\nvalue = [21,
4]'),
 Text(49.33576600573416, 321.21818181818185, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(50.01684646828096, 321.21818181818185, 'X[0] <= 0.411\ngini = 0.494\nsamples = 9\nvalue = [5,
4]'),
 Text(49.67630623700756, 304.74545454545455, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(50.35738669955436, 304.74545454545455, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(49.33576600573416, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(49.67630623700756, 370.6363636363636, 'gini = 0.0\nsamples = 515\nvalue = [515, 0]'),
```

```
 Text(50.69792693082776, 387.1090909090909, 'X[6] <= -0.824\ngini = 0.278\nsamples = 6\nvalue = [5,
1]'),
 Text(50.35738669955436, 370.6363636363636, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(51.03846716210116, 370.6363636363636, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(52.91143843410488, 403.5818181818182, 'X[5] <= 0.175\ngini = 0.159\nsamples = 241\nvalue = [220, 21
]'),
 Text(52.060087855921374, 387.1090909090909, 'X[6] <= -0.741\ngini = 0.388\nsamples = 19\nvalue = [5,
14]'),
 Text(51.71954762464797, 370.6363636363636, 'gini = 0.0\nsamples = 14\nvalue = [0, 14]'),
 Text(52.400628087194775, 370.6363636363636, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(53.76278901228839, 387.1090909090909, 'X[1] <= -0.658\ngini = 0.061\nsamples = 222\nvalue = [215, 7
]'),
 Text(53.08170854974158, 370.6363636363636, 'X[11] <= 1.02\ngini = 0.011\nsamples = 187\nvalue = [186, 1]
'),
 Text(52.741168318468176, 354.1636363636364, 'gini = 0.0\nsamples = 166\nvalue = [166, 0]'),
 Text(53.42224878101498, 354.1636363636364, 'X[1] <= -0.691\ngini = 0.091\nsamples = 21\nvalue = [20,
1]'),
 Text(53.08170854974158, 337.69090909090914, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(53.76278901228839, 337.69090909090914, 'X[6] <= 0.91\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(53.42224878101498, 321.21818181818185, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(54.10332924356179, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(54.44386947483519, 370.6363636363636, 'X[1] <= -0.624\ngini = 0.284\nsamples = 35\nvalue = [29,
6]'),
 Text(54.10332924356179, 354.1636363636364, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(54.78440970610859, 354.1636363636364, 'gini = 0.0\nsamples = 29\nvalue = [29, 0]'),
 Text(55.806030399928794, 420.05454545454546, 'X[1] <= -0.576\ngini = 0.244\nsamples = 435\nvalue =
[373, 62]'),
 Text(55.46549016865539, 403.5818181818182, 'gini = 0.0\nsamples = 36\nvalue = [0, 36]'),
 Text(56.1465706312022, 403.5818181818182, 'X[1] <= -0.576\ngini = 0.122\nsamples = 399\nvalue = [373, 26
]'),
 Text(55.46549016865539, 387.1090909090909, 'X[2] <= 1.529\ngini = 0.01\nsamples = 192\nvalue = [191, 1]'
),
 Text(55.12494993738199, 370.6363636363636, 'gini = 0.0\nsamples = 177\nvalue = [177, 0]'),
 Text(55.806030399928794, 370.6363636363636, 'X[6] <= 1.822\ngini = 0.124\nsamples = 15\nvalue = [14,
1]'),
 Text(55.46549016865539, 354.1636363636364, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(56.1465706312022, 354.1636363636364, 'X[7] <= 0.129\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(55.806030399928794, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(56.4871108624756, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(56.827651093749004, 387.1090909090909, 'X[1] <= -0.53\ngini = 0.212\nsamples = 207\nvalue = [182,
25]'),
 Text(56.4871108624756, 370.6363636363636, 'gini = 0.0\nsamples = 24\nvalue = [0, 24]'),
 Text(57.168191325022406, 370.6363636363636, 'X[7] <= 1.643\ngini = 0.011\nsamples = 183\nvalue = [182,
1]'),
 Text(56.827651093749004, 354.1636363636364, 'gini = 0.0\nsamples = 139\nvalue = [139, 0]'),
 Text(57.50873155629581, 354.1636363636364, 'X[5] <= -0.301\ngini = 0.044\nsamples = 44\nvalue = [43,
1]'),
 Text(57.168191325022406, 337.69090909090914, 'gini = 0.0\nsamples = 30\nvalue = [30, 0]'),
 Text(57.84927178756921, 337.69090909090914, 'X[5] <= -0.267\ngini = 0.133\nsamples = 14\nvalue = [13,
1]'),
 Text(57.50873155629581, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(58.189812018842616, 321.21818181818185, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(101.5377109831424, 485.9454545454546, 'X[0] <= 0.411\ngini = 0.449\nsamples = 4113\nvalue = [2712,
1401]'),
 Text(82.57738950663686, 469.4727272727273, 'X[6] <= -0.541\ngini = 0.488\nsamples = 2830\nvalue =
[1635, 1195]'),
 Text(64.4046712395822, 453.0, 'X[5] <= -0.775\ngini = 0.278\nsamples = 617\nvalue = [514, 103]'),
 Text(61.08440398466653, 436.52727272727276, 'X[5] <= -0.847\ngini = 0.436\nsamples = 193\nvalue = [131,
62]'),
 Text(59.55197294393622, 420.05454545454546, 'X[5] <= -1.052\ngini = 0.251\nsamples = 129\nvalue = [110,
19]'),
 Text(59.21143271266282, 403.5818181818182, 'X[6] <= -0.842\ngini = 0.471\nsamples = 50\nvalue = [31,
19]'),
 Text(58.87089248138942, 387.1090909090909, 'gini = 0.0\nsamples = 22\nvalue = [22, 0]'),
 Text(59.55197294393622, 387.1090909090909, 'X[5] <= -1.119\ngini = 0.436\nsamples = 28\nvalue = [9,
19]'),
 Text(59.21143271266282, 370.6363636363636, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(59.89251317520962, 370.6363636363636, 'X[11] <= 1.02\ngini = 0.287\nsamples = 23\nvalue = [4,
19]'),
 Text(59.55197294393622, 354.1636363636364, 'X[1] <= -0.255\ngini = 0.172\nsamples = 21\nvalue = [2,
19]'),
 Text(59.21143271266282, 337.69090909090914, 'X[7] <= 1.643\ngini = 0.095\nsamples = 20\nvalue = [1,
19]'),
 Text(58.87089248138942, 321.21818181818185, 'gini = 0.0\nsamples = 19\nvalue = [0, 19]'),
 Text(59.55197294393622, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(59.89251317520962, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(60.23305340648303, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
```

```
 Text(59.89251317520962, 403.5818181818182, 'gini = 0.0\nsamples = 79\nvalue = [79, 0]'),
 Text(62.616835025396846, 420.05454545454546, 'X[1] <= -0.422\ngini = 0.441\nsamples = 64\nvalue = [21,
43]'),
 Text(61.254674100303234, 403.5818181818182, 'X[1] <= -0.494\ngini = 0.375\nsamples = 16\nvalue = [12,
4]'),
 Text(60.91413386902983, 387.1090909090909, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(61.595214331576635, 387.1090909090909, 'X[5] <= -0.812\ngini = 0.142\nsamples = 13\nvalue = [12,
1]'),
 Text(61.254674100303234, 370.6363636363636, 'X[4] <= -0.534\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(60.91413386902983, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(61.595214331576635, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(61.935754562850036, 370.6363636363636, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(63.97899595049045, 403.5818181818182, 'X[1] <= -0.255\ngini = 0.305\nsamples = 48\nvalue = [9,
39]'),
 Text(63.63845571921705, 387.1090909090909, 'X[8] <= 0.228\ngini = 0.231\nsamples = 45\nvalue = [6,
39]'),
 Text(62.616835025396846, 370.6363636363636, 'X[10] <= 0.084\ngini = 0.463\nsamples = 11\nvalue = [4, 7]
'),
 Text(62.27629479412344, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(62.95737525667025, 354.1636363636364, 'X[1] <= -0.342\ngini = 0.346\nsamples = 9\nvalue = [2,
7]'),
 Text(62.616835025396846, 337.69090909090914, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(63.29791548794365, 337.69090909090914, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(64.66007641303726, 370.6363636363636, 'X[5] <= -0.827\ngini = 0.111\nsamples = 34\nvalue = [2,
32]'),
 Text(64.31953618176385, 354.1636363636364, 'X[10] <= 0.084\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(63.97899595049045, 337.69090909090914, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(64.66007641303726, 337.69090909090914, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(65.00061664431065, 354.1636363636364, 'gini = 0.0\nsamples = 30\nvalue = [0, 30]'),
 Text(64.31953618176385, 387.1090909090909, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(67.72493849449788, 436.52727272727276, 'X[5] <= 0.267\ngini = 0.175\nsamples = 424\nvalue = [383,
41]'),
 Text(65.68169710685747, 420.05454545454546, 'X[6] <= -0.79\ngini = 0.072\nsamples = 350\nvalue = [337,
13]'),
 Text(65.34115687558406, 403.5818181818182, 'gini = 0.0\nsamples = 234\nvalue = [234, 0]'),
 Text(66.02223733813086, 403.5818181818182, 'X[5] <= -0.461\ngini = 0.199\nsamples = 116\nvalue = [103, 1
3]'),
 Text(65.68169710685747, 387.1090909090909, 'X[5] <= -0.557\ngini = 0.357\nsamples = 56\nvalue = [43,
13]'),
 Text(65.34115687558406, 370.6363636363636, 'gini = 0.0\nsamples = 33\nvalue = [33, 0]'),
 Text(66.02223733813086, 370.6363636363636, 'X[8] <= 0.228\ngini = 0.491\nsamples = 23\nvalue = [10,
13]'),
 Text(65.68169710685747, 354.1636363636364, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(66.36277756940427, 354.1636363636364, 'X[1] <= -0.265\ngini = 0.401\nsamples = 18\nvalue = [5,
13]'),
 Text(66.02223733813086, 337.69090909090914, 'X[11] <= 1.02\ngini = 0.305\nsamples = 16\nvalue = [3,
13]'),
 Text(65.68169710685747, 321.21818181818185, 'X[6] <= -0.56\ngini = 0.231\nsamples = 15\nvalue = [2,
13]'),
 Text(65.34115687558406, 304.74545454545455, 'X[6] <= -0.723\ngini = 0.133\nsamples = 14\nvalue = [1,
13]'),
 Text(65.00061664431065, 288.2727272727273, 'X[5] <= -0.508\ngini = 0.375\nsamples = 4\nvalue = [1,
3]'),
 Text(64.66007641303726, 271.8, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(65.34115687558406, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(65.68169710685747, 288.2727272727273, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(66.02223733813086, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(66.36277756940427, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(66.70331780067767, 337.69090909090914, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(66.36277756940427, 387.1090909090909, 'gini = 0.0\nsamples = 60\nvalue = [60, 0]'),
 Text(69.76817988213828, 420.05454545454546, 'X[6] <= -0.763\ngini = 0.47\nsamples = 74\nvalue = [46,
28]'),
 Text(68.40601895704468, 403.5818181818182, 'X[1] <= -0.399\ngini = 0.3\nsamples = 49\nvalue = [40, 9]'),
 Text(68.06547872577129, 387.1090909090909, 'X[5] <= 1.416\ngini = 0.483\nsamples = 22\nvalue = [13,
9]'),
 Text(67.38439826322447, 370.6363636363636, 'X[5] <= 0.34\ngini = 0.142\nsamples = 13\nvalue = [12, 1]'),
 Text(67.04385803195107, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(67.72493849449788, 354.1636363636364, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
 Text(68.74655918831809, 370.6363636363636, 'X[11] <= 1.02\ngini = 0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(68.40601895704468, 354.1636363636364, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(69.08709941959148, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(68.74655918831809, 387.1090909090909, 'gini = 0.0\nsamples = 27\nvalue = [27, 0]'),
 Text(71.1303408072319, 403.5818181818182, 'X[11] <= 1.02\ngini = 0.365\nsamples = 25\nvalue = [6,
19]'),
 Text(70.7898005759585, 387.1090909090909, 'X[8] <= 0.228\ngini = 0.287\nsamples = 23\nvalue = [4,
19]'),
```

```
 Text(70.10872011341169, 370.6363636363636, 'X[5] <= 0.685\ngini = 0.105\nsamples = 18\nvalue = [1,
17]'),
 Text(69.76817988213828, 354.1636363636364, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
 Text(70.4492603446851, 354.1636363636364, 'X[5] <= 1.222\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),
 Text(70.10872011341169, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(70.7898005759585, 337.69090909090914, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(71.4708810385053, 370.6363636363636, 'X[5] <= 1.026\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
 Text(71.1303408072319, 354.1636363636364, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(71.8114212697787, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(71.4708810385053, 387.1090909090909, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(100.75010777369154, 453.0, 'X[6] <= 0.918\ngini = 0.5\nsamples = 2213\nvalue = [1121, 1092]'),
 Text(95.23037962562697, 436.52727272727276, 'X[1] <= -0.253\ngini = 0.486\nsamples = 1715\nvalue =
[712, 1003]'),
 Text(91.08686301278424, 420.05454545454546, 'X[7] <= 1.643\ngini = 0.467\nsamples = 1559\nvalue = [578,
981]'),
 Text(86.7160424467429, 403.5818181818182, 'X[11] <= 1.02\ngini = 0.439\nsamples = 1394\nvalue = [454,
940]'),
 Text(80.69872316484745, 387.1090909090909, 'X[10] <= 0.084\ngini = 0.419\nsamples = 1324\nvalue = [395,
929]'),
 Text(76.26238851134434, 370.6363636363636, 'X[1] <= -0.299\ngini = 0.291\nsamples = 627\nvalue = [111, 5
16]'),
 Text(75.92184828007095, 354.1636363636364, 'X[1] <= -0.299\ngini = 0.323\nsamples = 547\nvalue = [111, 4
36]'),
 Text(75.58130804879754, 337.69090909090914, 'X[1] <= -0.345\ngini = 0.286\nsamples = 527\nvalue = [91,
436]'),
 Text(75.24076781752414, 321.21818181818185, 'X[1] <= -0.346\ngini = 0.328\nsamples = 441\nvalue = [91,
350]'),
 Text(72.56167396680291, 304.74545454545455, 'X[5] <= -0.258\ngini = 0.279\nsamples = 399\nvalue = [67,
332]'),
 Text(69.5872678842743, 288.2727272727273, 'X[6] <= 0.271\ngini = 0.214\nsamples = 312\nvalue = [38,
274]'),
 Text(66.02223733813086, 271.8, 'X[5] <= -1.064\ngini = 0.161\nsamples = 261\nvalue = [23, 238]'),
 Text(64.29825241730927, 255.32727272727277, 'X[1] <= -0.409\ngini = 0.375\nsamples = 4\nvalue = [3,
1]'),
 Text(63.95771218603586, 238.85454545454547, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(64.63879264858267, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(67.74622225895247, 255.32727272727277, 'X[1] <= -0.484\ngini = 0.144\nsamples = 257\nvalue = [20,
237]'),
 Text(65.31987311112947, 238.85454545454547, 'X[6] <= -0.139\ngini = 0.389\nsamples = 34\nvalue = [9,
25]'),
 Text(64.97933287985607, 222.38181818181823, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(65.66041334240288, 222.38181818181823, 'X[5] <= -0.466\ngini = 0.238\nsamples = 29\nvalue = [4,
25]'),
 Text(64.97933287985607, 205.90909090909093, 'X[1] <= -0.485\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(64.63879264858267, 189.43636363636364, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(65.31987311112947, 189.43636363636364, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(66.34149380494968, 205.90909090909093, 'X[4] <= -0.534\ngini = 0.083\nsamples = 23\nvalue = [1,
22]'),
 Text(66.00095357367628, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(66.68203403622309, 189.43636363636364, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(70.17257140677546, 238.85454545454547, 'X[5] <= -0.385\ngini = 0.094\nsamples = 223\nvalue = [11,
212]'),
 Text(68.89554553950019, 222.38181818181823, 'X[5] <= -0.587\ngini = 0.066\nsamples = 206\nvalue = [7,
199]'),
 Text(68.04419496131669, 205.90909090909093, 'X[1] <= -0.437\ngini = 0.12\nsamples = 94\nvalue = [6,
88]'),
 Text(67.36311449876989, 189.43636363636364, 'X[1] <= -0.447\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(67.02257426749648, 172.9636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(67.70365473004328, 172.9636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(68.7252754238635, 189.43636363636364, 'X[5] <= -0.742\ngini = 0.084\nsamples = 91\nvalue = [4,
87]'),
 Text(68.3847351925901, 172.9636363636364, 'gini = 0.0\nsamples = 65\nvalue = [0, 65]'),
 Text(69.0658156551369, 172.9636363636364, 'X[6] <= 0.09\ngini = 0.26\nsamples = 26\nvalue = [4, 22]'),
 Text(68.7252754238635, 156.4909090909091, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(69.4063558864103, 156.4909090909091, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(69.7468961176837, 205.90909090909093, 'X[5] <= -0.413\ngini = 0.018\nsamples = 112\nvalue = [1,
111]'),
 Text(69.4063558864103, 189.43636363636364, 'gini = 0.0\nsamples = 101\nvalue = [0, 101]'),
 Text(70.0874363489571, 189.43636363636364, 'X[6] <= -0.144\ngini = 0.165\nsamples = 11\nvalue = [1,
10]'),
 Text(69.7468961176837, 172.9636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(70.4279765802305, 172.9636363636364, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(71.44959727405072, 222.38181818181823, 'X[5] <= -0.295\ngini = 0.36\nsamples = 17\nvalue = [4,
13]'),
 Text(71.10905704277731, 205.90909090909093, 'X[1] <= -0.424\ngini = 0.32\nsamples = 5\nvalue = [4,
1]'),
```

```
 Text(70.76851681150391, 189.43636363636364, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(71.44959727405072, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(71.79013750532411, 205.90909090909093, 'gini = 0.0\nsamples = 12\nvalue = [0, 12]'),
 Text(73.15229843041773, 271.8, 'X[5] <= -0.381\ngini = 0.415\nsamples = 51\nvalue = [15, 36]'),
 Text(72.81175819914432, 255.32727272727277, 'X[5] <= -0.659\ngini = 0.494\nsamples = 27\nvalue = [15,
12]'),
 Text(72.47121796787093, 238.85454545454547, 'X[5] <= -0.869\ngini = 0.465\nsamples = 19\nvalue = [7,
12]'),
 Text(72.13067773659752, 222.38181818181823, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(72.81175819914432, 222.38181818181823, 'X[8] <= 0.228\ngini = 0.32\nsamples = 15\nvalue = [3,
12]'),
 Text(72.47121796787093, 205.90909090909093, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(73.15229843041773, 205.90909090909093, 'X[1] <= -0.437\ngini = 0.444\nsamples = 9\nvalue = [3,
6]'),
 Text(72.81175819914432, 189.43636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(73.49283866169112, 189.43636363636364, 'X[7] <= 0.129\ngini = 0.245\nsamples = 7\nvalue = [1,
6]'),
 Text(73.15229843041773, 172.9636363636364, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(73.83337889296453, 172.9636363636364, 'X[5] <= -0.78\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(73.49283866169112, 156.4909090909091, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(74.17391912423793, 156.4909090909091, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(73.15229843041773, 238.85454545454547, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(73.49283866169112, 255.32727272727277, 'gini = 0.0\nsamples = 24\nvalue = [0, 24]'),
 Text(75.53608004933155, 288.2727272727273, 'X[5] <= 0.952\ngini = 0.444\nsamples = 87\nvalue = [29,
58]'),
 Text(74.51445935551133, 271.8, 'X[1] <= -0.492\ngini = 0.422\nsamples = 33\nvalue = [23, 10]'),
 Text(74.17391912423793, 255.32727272727277, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(74.85499958678474, 255.32727272727277, 'X[1] <= -0.391\ngini = 0.328\nsamples = 29\nvalue = [23,
6]'),
 Text(74.51445935551133, 238.85454545454547, 'X[5] <= 0.026\ngini = 0.252\nsamples = 27\nvalue = [23,
4]'),
 Text(74.17391912423793, 222.38181818181823, 'X[5] <= -0.072\ngini = 0.426\nsamples = 13\nvalue = [9,
4]'),
 Text(73.83337889296453, 205.90909090909093, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(74.51445935551133, 205.90909090909093, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(74.85499958678474, 222.38181818181823, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(75.19553981805814, 238.85454545454547, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(76.55770074315174, 271.8, 'X[5] <= 1.997\ngini = 0.198\nsamples = 54\nvalue = [6, 48]'),
 Text(76.21716051187835, 255.32727272727277, 'X[2] <= 1.529\ngini = 0.111\nsamples = 51\nvalue = [3,
48]'),
 Text(75.87662028060494, 238.85454545454547, 'X[4] <= -0.534\ngini = 0.04\nsamples = 49\nvalue = [1,
48]'),
 Text(75.53608004933155, 222.38181818181823, 'X[1] <= -0.397\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(75.19553981805814, 205.90909090909093, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(75.87662028060494, 205.90909090909093, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(76.21716051187835, 222.38181818181823, 'gini = 0.0\nsamples = 47\nvalue = [0, 47]'),
 Text(76.55770074315174, 238.85454545454547, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(76.89824097442515, 255.32727272727277, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(77.91986166824536, 304.74545454545455, 'X[5] <= -0.891\ngini = 0.49\nsamples = 42\nvalue = [24,
18]'),
 Text(77.57932143697195, 288.2727272727273, 'X[7] <= 0.129\ngini = 0.1\nsamples = 19\nvalue = [1, 18]'),
 Text(77.23878120569856, 271.8, 'gini = 0.0\nsamples = 18\nvalue = [0, 18]'),
 Text(77.91986166824536, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(78.26040189951875, 288.2727272727273, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
 Text(75.92184828007095, 321.21818181818185, 'gini = 0.0\nsamples = 86\nvalue = [0, 86]'),
 Text(76.26238851134434, 337.69090909090914, 'gini = 0.0\nsamples = 20\nvalue = [20, 0]'),
 Text(76.60292874261775, 354.1636363636364, 'gini = 0.0\nsamples = 80\nvalue = [0, 80]'),
 Text(85.13505781835057, 370.6363636363636, 'X[5] <= -0.967\ngini = 0.483\nsamples = 697\nvalue = [284, 4
13]'),
 Text(80.98472374970598, 354.1636363636364, 'X[5] <= -1.126\ngini = 0.244\nsamples = 176\nvalue = [25, 15
1]'),
 Text(80.64418351843257, 337.69090909090914, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(81.32526398097939, 337.69090909090914, 'X[7] <= 0.129\ngini = 0.164\nsamples = 166\nvalue = [15,
151]'),
 Text(80.64418351843257, 321.21818181818185, 'X[2] <= 1.529\ngini = 0.127\nsamples = 161\nvalue = [11,
150]'),
 Text(80.30364328715918, 304.74545454545455, 'X[8] <= 0.228\ngini = 0.107\nsamples = 159\nvalue = [9,
150]'),
 Text(79.28202259333897, 288.2727272727273, 'X[6] <= -0.243\ngini = 0.043\nsamples = 135\nvalue = [3,
132]'),
 Text(78.60094213079216, 271.8, 'X[1] <= -0.332\ngini = 0.18\nsamples = 10\nvalue = [1, 9]'),
 Text(78.26040189951875, 255.32727272727277, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
 Text(78.94148236206557, 255.32727272727277, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(79.96310305588577, 271.8, 'X[1] <= -0.388\ngini = 0.031\nsamples = 125\nvalue = [2, 123]'),
 Text(79.62256282461237, 255.32727272727277, 'X[1] <= -0.392\ngini = 0.108\nsamples = 35\nvalue = [2,
33]'),
 Text(79.28202259333897, 238.85454545454547, 'X[6] <= 0.502\ngini = 0.057\nsamples = 34\nvalue = [1,
```

```
33]'),
 Text(78.94148236206557, 222.38181818181823, 'gini = 0.0\nsamples = 24\nvalue = [0, 24]'),
 Text(79.62256282461237, 222.38181818181823, 'X[5] <= -1.037\ngini = 0.18\nsamples = 10\nvalue = [1,
9]'),
 Text(79.28202259333897, 205.90909090909093, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
 Text(79.96310305588577, 205.90909090909093, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(79.96310305588577, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(80.30364328715918, 255.32727272727277, 'gini = 0.0\nsamples = 90\nvalue = [0, 90]'),
 Text(81.32526398097939, 288.2727272727273, 'X[1] <= -0.471\ngini = 0.375\nsamples = 24\nvalue = [6,
18]'),
 Text(80.98472374970598, 271.8, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(81.66580421225278, 271.8, 'X[5] <= -1.076\ngini = 0.245\nsamples = 21\nvalue = [3, 18]'),
 Text(80.98472374970598, 255.32727272727277, 'X[6] <= -0.176\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(80.64418351843257, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(81.32526398097939, 238.85454545454547, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(82.34688467479958, 255.32727272727277, 'X[6] <= -0.353\ngini = 0.105\nsamples = 18\nvalue = [1,
17]'),
 Text(82.00634444352619, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(82.68742490607299, 238.85454545454547, 'gini = 0.0\nsamples = 17\nvalue = [0, 17]'),
 Text(80.98472374970598, 304.74545454545455, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(82.00634444352619, 321.21818181818185, 'X[1] <= -0.287\ngini = 0.32\nsamples = 5\nvalue = [4,
1]'),
 Text(81.66580421225278, 304.74545454545455, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(82.34688467479958, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(89.28539188699516, 354.1636363636364, 'X[1] <= -0.299\ngini = 0.5\nsamples = 521\nvalue = [259, 262
]'),
 Text(88.94485165572175, 337.69090909090914, 'X[1] <= -0.301\ngini = 0.497\nsamples = 479\nvalue = [259,
220]'),
 Text(88.60431142444835, 321.21818181818185, 'X[5] <= -0.278\ngini = 0.499\nsamples = 422\nvalue = [202,
220]'),
 Text(86.34823239226206, 304.74545454545455, 'X[5] <= -0.542\ngini = 0.474\nsamples = 313\nvalue = [121,
192]'),
 Text(84.39012606244, 288.2727272727273, 'X[1] <= -0.436\ngini = 0.489\nsamples = 139\nvalue = [80, 59]')
,
 Text(83.3685053686198, 271.8, 'X[6] <= 0.255\ngini = 0.043\nsamples = 46\nvalue = [45, 1]'),
 Text(83.02796513734638, 255.32727272727277, 'gini = 0.0\nsamples = 34\nvalue = [34, 0]'),
 Text(83.7090455998932, 255.32727272727277, 'X[6] <= 0.321\ngini = 0.153\nsamples = 12\nvalue = [11, 1]'
),
 Text(83.3685053686198, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(84.0495858311666, 238.85454545454547, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(85.41174675626021, 271.8, 'X[5] <= -0.71\ngini = 0.469\nsamples = 93\nvalue = [35, 58]'),
 Text(85.0712065249868, 255.32727272727277, 'X[5] <= -0.859\ngini = 0.351\nsamples = 75\nvalue = [17,
58]'),
 Text(84.7306662937134, 238.85454545454547, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(85.41174675626021, 238.85454545454547, 'X[2] <= 1.529\ngini = 0.233\nsamples = 67\nvalue = [9,
58]'),
 Text(85.0712065249868, 222.38181818181823, 'X[6] <= 0.352\ngini = 0.192\nsamples = 65\nvalue = [7,
58]'),
 Text(84.7306662937134, 205.90909090909093, 'X[5] <= -0.775\ngini = 0.146\nsamples = 63\nvalue = [5,
58]'),
 Text(84.39012606244, 189.43636363636364, 'gini = 0.0\nsamples = 45\nvalue = [0, 45]'),
 Text(85.0712065249868, 189.43636363636364, 'X[1] <= -0.397\ngini = 0.401\nsamples = 18\nvalue = [5,
13]'),
 Text(84.7306662937134, 172.9636363636364, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
 Text(85.41174675626021, 172.9636363636364, 'X[6] <= 0.25\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
 Text(85.0712065249868, 156.4909090909091, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(85.75228698753361, 156.4909090909091, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(85.41174675626021, 205.90909090909093, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(85.75228698753361, 222.38181818181823, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(85.75228698753361, 255.32727272727277, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),
 Text(88.30633872208412, 288.2727272727273, 'X[6] <= 0.595\ngini = 0.36\nsamples = 174\nvalue = [41, 133]
'),
 Text(87.96579849081073, 271.8, 'X[6] <= -0.087\ngini = 0.307\nsamples = 164\nvalue = [31, 133]'),
 Text(87.11444791262721, 255.32727272727277, 'X[1] <= -0.398\ngini = 0.498\nsamples = 32\nvalue = [17,
15]'),
 Text(86.77390768135382, 238.85454545454547, 'X[5] <= -0.478\ngini = 0.454\nsamples = 23\nvalue = [8,
15]'),
 Text(86.43336745008041, 222.38181818181823, 'X[1] <= -0.478\ngini = 0.278\nsamples = 18\nvalue = [3,
15]'),
 Text(86.09282721880702, 205.90909090909093, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(86.77390768135382, 205.90909090909093, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),
 Text(87.11444791262721, 222.38181818181823, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(87.45498814390062, 238.85454545454547, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(88.81714906899423, 255.32727272727277, 'X[1] <= -0.438\ngini = 0.19\nsamples = 132\nvalue = [14, 11
8]'),
 Text(88.13606860644742, 238.85454545454547, 'X[5] <= -0.447\ngini = 0.051\nsamples = 76\nvalue = [2,
```

```
74]'),
 Text(87.79552837517403, 222.38181818181823, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(88.47660883772083, 222.38181818181823, 'gini = 0.0\nsamples = 74\nvalue = [0, 74]'),
 Text(89.49822953154103, 238.85454545454547, 'X[5] <= -0.42\ngini = 0.337\nsamples = 56\nvalue = [12,
44]'),
 Text(89.15768930026763, 222.38181818181823, 'X[4] <= -0.534\ngini = 0.153\nsamples = 48\nvalue = [4,
44]'),
 Text(88.81714906899423, 205.90909090909093, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(89.49822953154103, 205.90909090909093, 'X[6] <= 0.053\ngini = 0.12\nsamples = 47\nvalue = [3,
44]'),
 Text(89.15768930026763, 189.43636363636364, 'X[6] <= -0.033\ngini = 0.083\nsamples = 46\nvalue = [2,
44]'),
 Text(88.81714906899423, 172.9636363636364, 'gini = 0.0\nsamples = 38\nvalue = [0, 38]'),
 Text(89.49822953154103, 172.9636363636364, 'X[1] <= -0.398\ngini = 0.375\nsamples = 8\nvalue = [2,
6]'),
 Text(89.15768930026763, 156.4909090909091, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(89.83876976281444, 156.4909090909091, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(89.83876976281444, 189.43636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(89.83876976281444, 222.38181818181823, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(88.64687895335753, 271.8, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(90.86039045663465, 304.74545454545455, 'X[5] <= 1.181\ngini = 0.382\nsamples = 109\nvalue = [81,
28]'),
 Text(89.66849964717774, 288.2727272727273, 'X[1] <= -0.498\ngini = 0.076\nsamples = 76\nvalue = [73,
3]'),
 Text(89.32795941590433, 271.8, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(90.00903987845113, 271.8, 'X[1] <= -0.333\ngini = 0.027\nsamples = 74\nvalue = [73, 1]'),
 Text(89.66849964717774, 255.32727272727277, 'gini = 0.0\nsamples = 73\nvalue = [73, 0]'),
 Text(90.34958010972454, 255.32727272727277, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(92.05228126609155, 288.2727272727273, 'X[5] <= 1.574\ngini = 0.367\nsamples = 33\nvalue = [8,
25]'),
 Text(91.37120080354475, 271.8, 'X[4] <= -0.534\ngini = 0.077\nsamples = 25\nvalue = [1, 24]'),
 Text(91.03066057227134, 255.32727272727277, 'gini = 0.0\nsamples = 21\nvalue = [0, 21]'),
 Text(91.71174103481815, 255.32727272727277, 'X[5] <= 1.377\ngini = 0.375\nsamples = 4\nvalue = [1,
3]'),
 Text(91.37120080354475, 238.85454545454547, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(92.05228126609155, 238.85454545454547, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(92.73336172863836, 271.8, 'X[7] <= 0.129\ngini = 0.219\nsamples = 8\nvalue = [7, 1]'),
 Text(92.39282149736495, 255.32727272727277, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(93.07390195991177, 255.32727272727277, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(89.28539188699516, 321.21818181818185, 'gini = 0.0\nsamples = 57\nvalue = [57, 0]'),
 Text(89.62593211826857, 337.69090909090914, 'gini = 0.0\nsamples = 42\nvalue = [0, 42]'),
 Text(92.73336172863836, 387.1090909090909, 'X[5] <= -0.091\ngini = 0.265\nsamples = 70\nvalue = [59,
11]'),
 Text(92.39282149736495, 370.6363636363636, 'gini = 0.0\nsamples = 44\nvalue = [44, 0]'),
 Text(93.07390195991177, 370.6363636363636, 'X[8] <= 0.228\ngini = 0.488\nsamples = 26\nvalue = [15,
11]'),
 Text(92.73336172863836, 354.1636363636364, 'X[6] <= -0.077\ngini = 0.475\nsamples = 18\nvalue = [7,
11]'),
 Text(92.39282149736495, 337.69090909090914, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(93.07390195991177, 337.69090909090914, 'X[7] <= 0.129\ngini = 0.391\nsamples = 15\nvalue = [4,
11]'),
 Text(92.73336172863836, 321.21818181818185, 'X[1] <= -0.357\ngini = 0.26\nsamples = 13\nvalue = [2,
11]'),
 Text(92.39282149736495, 304.74545454545455, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(93.07390195991177, 304.74545454545455, 'X[1] <= -0.323\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(92.73336172863836, 288.2727272727273, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(93.41444219118516, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(93.41444219118516, 321.21818181818185, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(93.41444219118516, 354.1636363636364, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(95.45768357882558, 403.5818181818182, 'X[1] <= -0.295\ngini = 0.373\nsamples = 165\nvalue = [124, 4
1]'),
 Text(95.11714334755217, 387.1090909090909, 'X[1] <= -0.486\ngini = 0.279\nsamples = 149\nvalue = [124, 2
5]'),
 Text(94.77660311627876, 370.6363636363636, 'gini = 0.0\nsamples = 12\nvalue = [0, 12]'),
 Text(95.45768357882558, 370.6363636363636, 'X[1] <= -0.439\ngini = 0.172\nsamples = 137\nvalue = [124, 1
3]'),
 Text(94.77660311627876, 354.1636363636364, 'X[1] <= -0.478\ngini = 0.466\nsamples = 27\nvalue = [17,
10]'),
 Text(94.43606288500537, 337.69090909090914, 'X[5] <= 1.806\ngini = 0.105\nsamples = 18\nvalue = [17,
1]'),
 Text(94.09552265373196, 321.21818181818185, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(94.77660311627876, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(95.11714334755217, 337.69090909090914, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
 Text(96.13876404137238, 354.1636363636364, 'X[6] <= 0.877\ngini = 0.053\nsamples = 110\nvalue = [107, 3]
'),
 Text(95.79822381009897, 337.69090909090914, 'X[1] <= -0.398\ngini = 0.036\nsamples = 109\nvalue = [107,
```

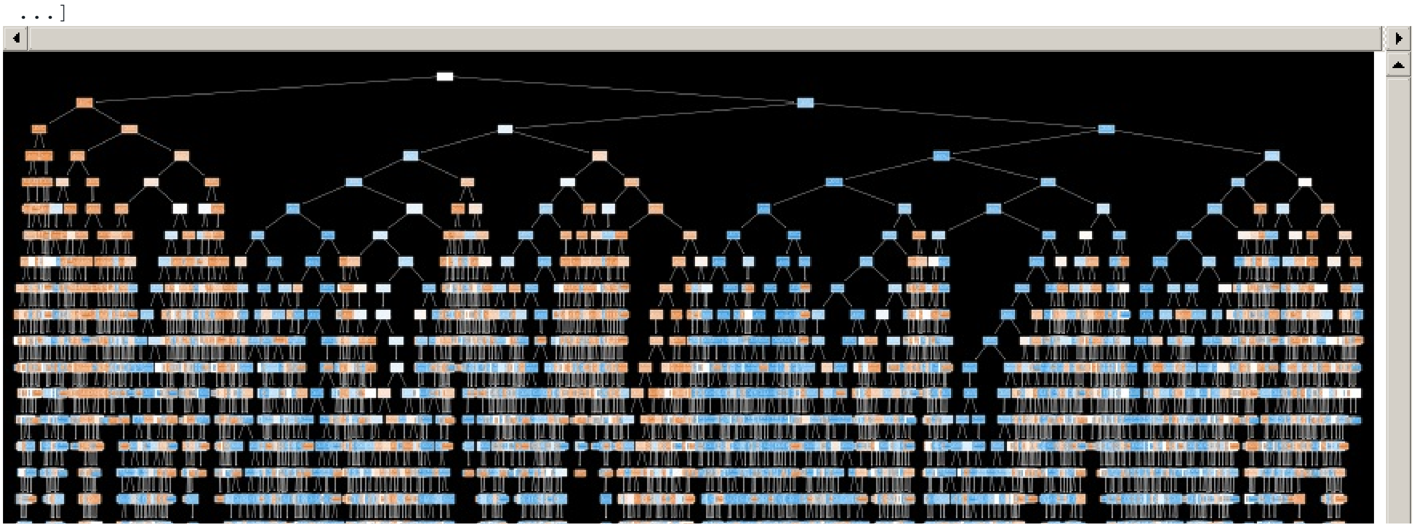
```
 Text(95.79822381009897, 337.69090909090914, 'X[1] <= -0.398\ngini = 0.036\nsamples = 109\nvalue = [107,
2]'),
 Text(95.45768357882558, 321.21818181818185, 'X[1] <= -0.421\ngini = 0.142\nsamples = 26\nvalue = [24,
2]'),
 Text(95.11714334755217, 304.74545454545455, 'X[5] <= -0.5\ngini = 0.077\nsamples = 25\nvalue = [24,
1]'),
 Text(94.77660311627876, 288.2727272727273, 'X[5] <= -0.532\ngini = 0.219\nsamples = 8\nvalue = [7,
1]'),
 Text(94.43606288500537, 271.8, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(95.11714334755217, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(95.45768357882558, 288.2727272727273, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(95.79822381009897, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(96.13876404137238, 321.21818181818185, 'gini = 0.0\nsamples = 83\nvalue = [83, 0]'),
 Text(96.47930427264578, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(95.79822381009897, 387.1090909090909, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(99.3738962384697, 420.05454545454546, 'X[6] <= 0.672\ngini = 0.242\nsamples = 156\nvalue = [134,
22]'),
 Text(97.8414651977394, 403.5818181818182, 'X[5] <= 1.405\ngini = 0.117\nsamples = 128\nvalue = [120,
8]'),
 Text(96.81984450391919, 387.1090909090909, 'X[6] <= 0.549\ngini = 0.018\nsamples = 113\nvalue = [112, 1]
'),
 Text(96.47930427264578, 370.6363636363636, 'gini = 0.0\nsamples = 100\nvalue = [100, 0]'),
 Text(97.16038473519258, 370.6363636363636, 'X[6] <= 0.565\ngini = 0.142\nsamples = 13\nvalue = [12,
1]'),
 Text(96.81984450391919, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(97.50092496646599, 354.1636363636364, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
 Text(98.86308589155959, 387.1090909090909, 'X[5] <= 1.483\ngini = 0.498\nsamples = 15\nvalue = [8,
7]'),
 Text(98.5225456602862, 370.6363636363636, 'X[8] <= 0.228\ngini = 0.219\nsamples = 8\nvalue = [1, 7]'),
 Text(98.18200542901279, 354.1636363636364, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(98.86308589155959, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(99.203626122833, 370.6363636363636, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(100.90632727920001, 403.5818181818182, 'X[8] <= 0.228\ngini = 0.5\nsamples = 28\nvalue = [14,
14]'),
 Text(100.5657870479266, 387.1090909090909, 'X[10] <= 0.084\ngini = 0.444\nsamples = 21\nvalue = [7,
14]'),
 Text(99.8847065853798, 370.6363636363636, 'X[11] <= 1.02\ngini = 0.165\nsamples = 11\nvalue = [1,
10]'),
 Text(99.5441663541064, 354.1636363636364, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(100.22524681665321, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(101.24686751047341, 370.6363636363636, 'X[5] <= -0.723\ngini = 0.48\nsamples = 10\nvalue = [6,
4]'),
 Text(100.90632727920001, 354.1636363636364, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(101.58740774174682, 354.1636363636364, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(101.24686751047341, 387.1090909090909, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(106.2698359217561, 436.52727272727276, 'X[1] <= -0.485\ngini = 0.294\nsamples = 498\nvalue = [409,
89]'),
 Text(105.92929569048269, 420.05454545454546, 'gini = 0.0\nsamples = 23\nvalue = [0, 23]'),
 Text(106.61037615302949, 420.05454545454546, 'X[5] <= -0.808\ngini = 0.239\nsamples = 475\nvalue =
[409, 66]'),
 Text(103.46037901375053, 403.5818181818182, 'X[5] <= -0.886\ngini = 0.389\nsamples = 102\nvalue = [75,
27]'),
 Text(102.26848820429362, 387.1090909090909, 'X[6] <= 2.771\ngini = 0.061\nsamples = 63\nvalue = [61,
2]'),
 Text(101.92794797302022, 370.6363636363636, 'gini = 0.0\nsamples = 56\nvalue = [56, 0]'),
 Text(102.60902843556703, 370.6363636363636, 'X[6] <= 2.878\ngini = 0.408\nsamples = 7\nvalue = [5,
2]'),
 Text(102.26848820429362, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(102.94956866684042, 354.1636363636364, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(104.65226982320743, 387.1090909090909, 'X[6] <= 1.712\ngini = 0.46\nsamples = 39\nvalue = [14,
25]'),
 Text(104.31172959193404, 370.6363636363636, 'X[10] <= 0.084\ngini = 0.312\nsamples = 31\nvalue = [6, 25
]'),
 Text(103.63064912938722, 354.1636363636364, 'X[1] <= -0.482\ngini = 0.153\nsamples = 24\nvalue = [2,
22]'),
 Text(103.29010889811383, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(103.97118936066063, 337.69090909090914, 'X[2] <= 1.529\ngini = 0.083\nsamples = 23\nvalue = [1, 22
]'),
 Text(103.63064912938722, 321.21818181818185, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(104.31172959193404, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(104.99281005448084, 354.1636363636364, 'X[5] <= -0.826\ngini = 0.49\nsamples = 7\nvalue = [4,
3]'),
 Text(104.65226982320743, 337.69090909090914, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(105.33335028575424, 337.69090909090914, 'X[8] <= 0.228\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'
),
 Text(104.99281005448084, 321.21818181818185, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(105.67389051702764, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(104.99281005448084, 370.6363636363636, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
```

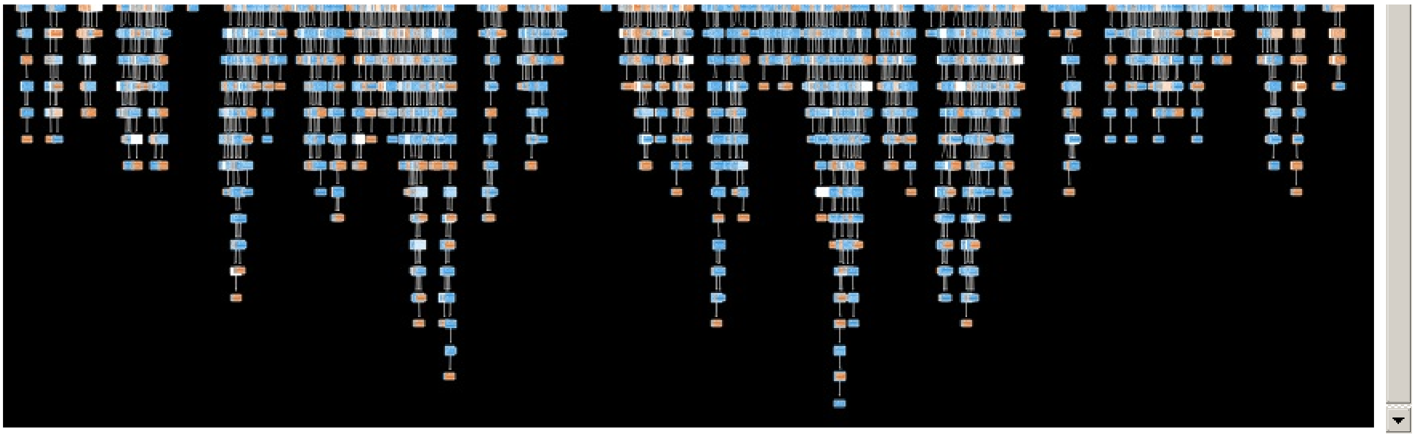
```
 Text(104.99281005448084, 370.6363636363636, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(109.76037329230847, 403.5818181818182, 'X[5] <= 2.009\ngini = 0.187\nsamples = 373\nvalue = [334,
39]'),
 Text(108.82388765630661, 387.1090909090909, 'X[5] <= -0.289\ngini = 0.154\nsamples = 344\nvalue = [315,
29]'),
 Text(107.97253707812311, 370.6363636363636, 'X[5] <= -0.3\ngini = 0.235\nsamples = 206\nvalue = [178,
28]'),
 Text(107.29145661557631, 354.1636363636364, 'X[5] <= -0.597\ngini = 0.192\nsamples = 195\nvalue = [174,
21]'),
 Text(106.9509163843029, 337.69090909090914, 'gini = 0.0\nsamples = 86\nvalue = [86, 0]'),
 Text(107.6319968468497, 337.69090909090914, 'X[5] <= -0.507\ngini = 0.311\nsamples = 109\nvalue = [88,
21]'),
 Text(106.35497097957445, 321.21818181818185, 'X[6] <= 1.894\ngini = 0.483\nsamples = 49\nvalue = [29, 2
0]'),
 Text(105.16308017011754, 304.74545454545455, 'X[2] <= 1.529\ngini = 0.26\nsamples = 26\nvalue = [22,
4]'),
 Text(104.82253993884413, 288.2727272727273, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(105.50362040139095, 288.2727272727273, 'X[5] <= -0.552\ngini = 0.32\nsamples = 5\nvalue = [1,
4]'),
 Text(105.16308017011754, 271.8, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(105.84416063266434, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(107.54686178903135, 304.74545454545455, 'X[1] <= -0.35\ngini = 0.423\nsamples = 23\nvalue = [7, 16
]'),
 Text(106.86578132648455, 288.2727272727273, 'X[1] <= -0.482\ngini = 0.142\nsamples = 13\nvalue = [1,
12]'),
 Text(106.52524109521114, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(107.20632155775796, 271.8, 'gini = 0.0\nsamples = 12\nvalue = [0, 12]'),
 Text(108.22794225157816, 288.2727272727273, 'X[5] <= -0.519\ngini = 0.48\nsamples = 10\nvalue = [6,
4]'),
 Text(107.88740202030476, 271.8, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(108.56848248285156, 271.8, 'X[6] <= 2.226\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(108.22794225157816, 255.32727272727277, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(108.90902271412496, 255.32727272727277, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(108.90902271412496, 321.21818181818185, 'X[6] <= 3.24\ngini = 0.033\nsamples = 60\nvalue = [59,
1]'),
 Text(108.56848248285156, 304.74545454545455, 'gini = 0.0\nsamples = 57\nvalue = [57, 0]'),
 Text(109.24956294539837, 304.74545454545455, 'X[4] <= -0.534\ngini = 0.444\nsamples = 3\nvalue = [2, 1]
'),
 Text(108.90902271412496, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(109.59010317667178, 288.2727272727273, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(108.65361754066991, 354.1636363636364, 'X[1] <= -0.298\ngini = 0.463\nsamples = 11\nvalue = [4,
7]'),
 Text(108.3130773093965, 337.69090909090914, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(108.99415777194332, 337.69090909090914, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(109.67523823449012, 370.6363636363636, 'X[5] <= 1.913\ngini = 0.014\nsamples = 138\nvalue = [137,
1]'),
 Text(109.33469800321672, 354.1636363636364, 'gini = 0.0\nsamples = 132\nvalue = [132, 0]'),
 Text(110.01577846576352, 354.1636363636364, 'X[5] <= 1.918\ngini = 0.278\nsamples = 6\nvalue = [5,
1]'),
 Text(109.67523823449012, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(110.35631869703693, 337.69090909090914, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(110.69685892831032, 387.1090909090909, 'X[7] <= 1.643\ngini = 0.452\nsamples = 29\nvalue = [19,
10]'),
 Text(110.35631869703693, 370.6363636363636, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(111.03739915958373, 370.6363636363636, 'X[8] <= 0.228\ngini = 0.278\nsamples = 12\nvalue = [2,
10]'),
 Text(110.69685892831032, 354.1636363636364, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(111.37793939085714, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(120.49803245964793, 469.4727272727273, 'X[4] <= -0.534\ngini = 0.27\nsamples = 1283\nvalue =
[1077, 206]'),
 Text(116.48604285995816, 453.0, 'X[7] <= 0.129\ngini = 0.5\nsamples = 204\nvalue = [99, 105]'),
 Text(114.44280147231775, 436.52727272727276, 'X[11] <= 1.02\ngini = 0.467\nsamples = 145\nvalue = [54,
91]'),
 Text(114.10226124104435, 420.05454545454546, 'X[10] <= 0.084\ngini = 0.436\nsamples = 134\nvalue = [43,
91]'),
 Text(112.39956008467733, 403.5818181818182, 'X[5] <= -0.621\ngini = 0.178\nsamples = 71\nvalue = [7,
64]'),
 Text(112.05901985340394, 387.1090909090909, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(112.74010031595074, 387.1090909090909, 'X[1] <= -0.254\ngini = 0.111\nsamples = 68\nvalue = [4,
64]'),
 Text(112.39956008467733, 370.6363636363636, 'X[6] <= -0.828\ngini = 0.086\nsamples = 67\nvalue = [3,
64]'),
 Text(112.05901985340394, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(112.74010031595074, 354.1636363636364, 'X[6] <= 0.779\ngini = 0.059\nsamples = 66\nvalue = [2,
64]'),
 Text(112.39956008467733, 337.69090909090914, 'gini = 0.0\nsamples = 48\nvalue = [0, 48]'),
 Text(113.08064054722414, 337.69090909090914, 'X[8] <= 0.228\ngini = 0.198\nsamples = 18\nvalue = [2, 16
]'),
```

```
]'),
 Text(112.74010031595074, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(113.42118077849754, 321.21818181818185, 'X[5] <= 0.407\ngini = 0.111\nsamples = 17\nvalue = [1, 16
]'),
 Text(113.08064054722414, 304.74545454545455, 'gini = 0.0\nsamples = 14\nvalue = [0, 14]'),
 Text(113.76172100977095, 304.74545454545455, 'X[6] <= 1.038\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'
),
 Text(113.42118077849754, 288.2727272727273, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(114.10226124104435, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(113.08064054722414, 370.6363636363636, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(115.80496239741136, 403.5818181818182, 'X[1] <= -0.438\ngini = 0.49\nsamples = 63\nvalue = [36,
27]'),
 Text(115.12388193486456, 387.1090909090909, 'X[6] <= 0.126\ngini = 0.389\nsamples = 34\nvalue = [9,
25]'),
 Text(114.78334170359115, 370.6363636363636, 'X[5] <= -1.083\ngini = 0.191\nsamples = 28\nvalue = [3,
25]'),
 Text(114.44280147231775, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(115.12388193486456, 354.1636363636364, 'X[6] <= -1.014\ngini = 0.137\nsamples = 27\nvalue = [2,
25]'),
 Text(114.44280147231775, 337.69090909090914, 'X[8] <= 0.228\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(114.10226124104435, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(114.78334170359115, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(115.80496239741136, 337.69090909090914, 'X[5] <= 0.385\ngini = 0.077\nsamples = 25\nvalue = [1, 24
]'),
 Text(115.46442216613796, 321.21818181818185, 'X[5] <= -0.761\ngini = 0.198\nsamples = 9\nvalue = [1, 8]
'),
 Text(115.12388193486456, 304.74545454545455, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(115.80496239741136, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(116.14550262868477, 321.21818181818185, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(115.46442216613796, 370.6363636363636, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(116.48604285995816, 387.1090909090909, 'X[5] <= -0.948\ngini = 0.128\nsamples = 29\nvalue = [27,
2]'),
 Text(116.14550262868477, 370.6363636363636, 'X[5] <= -1.002\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(115.80496239741136, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(116.48604285995816, 354.1636363636364, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(116.82658309123157, 370.6363636363636, 'gini = 0.0\nsamples = 25\nvalue = [25, 0]'),
 Text(114.78334170359115, 420.05454545454546, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(118.52928424759858, 436.52727272727276, 'X[6] <= -0.068\ngini = 0.362\nsamples = 59\nvalue = [45,
14]'),
 Text(118.18874401632517, 420.05454545454546, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(118.86982447887198, 420.05454545454546, 'X[6] <= 0.524\ngini = 0.465\nsamples = 38\nvalue = [24, 1
4]'),
 Text(118.52928424759858, 403.5818181818182, 'X[8] <= 0.228\ngini = 0.493\nsamples = 25\nvalue = [11,
14]'),
 Text(117.84820378505178, 387.1090909090909, 'X[1] <= -0.358\ngini = 0.36\nsamples = 17\nvalue = [4,
13]'),
 Text(117.50766355377837, 370.6363636363636, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(118.18874401632517, 370.6363636363636, 'X[1] <= -0.254\ngini = 0.133\nsamples = 14\nvalue = [1,
13]'),
 Text(117.84820378505178, 354.1636363636364, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(118.52928424759858, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(119.21036471014538, 387.1090909090909, 'X[1] <= -0.461\ngini = 0.219\nsamples = 8\nvalue = [7,
1]'),
 Text(118.86982447887198, 370.6363636363636, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(119.55090494141878, 370.6363636363636, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(119.21036471014538, 403.5818181818182, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(124.5100220593377, 453.0, 'X[1] <= -0.484\ngini = 0.17\nsamples = 1079\nvalue = [978, 101]'),
 Text(124.1694818280643, 436.52727272727276, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(124.85056229061111, 436.52727272727276, 'X[6] <= -0.042\ngini = 0.151\nsamples = 1066\nvalue =
[978, 88]'),
 Text(121.93468656033261, 420.05454545454546, 'X[1] <= -0.447\ngini = 0.021\nsamples = 376\nvalue =
[372, 4]'),
 Text(120.9130658665124, 403.5818181818182, 'X[1] <= -0.474\ngini = 0.103\nsamples = 55\nvalue = [52,
3]'),
 Text(120.57252563523899, 387.1090909090909, 'X[5] <= 1.017\ngini = 0.037\nsamples = 53\nvalue = [52,
1]'),
 Text(120.2319854039656, 370.6363636363636, 'gini = 0.0\nsamples = 48\nvalue = [48, 0]'),
 Text(120.9130658665124, 370.6363636363636, 'X[10] <= 0.084\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(120.57252563523899, 354.1636363636364, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(121.25360609778579, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(121.25360609778579, 387.1090909090909, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(122.9563072541528, 403.5818181818182, 'X[1] <= -0.415\ngini = 0.006\nsamples = 321\nvalue = [320, 1
]'),
 Text(122.61576702287941, 387.1090909090909, 'X[5] <= -0.544\ngini = 0.032\nsamples = 62\nvalue = [61,
1]'),
 Text(122.275226791606, 370.6363636363636, 'X[5] <= -0.566\ngini = 0.091\nsamples = 21\nvalue = [20,
1]'),
 Text(121.93468656033261, 354.1636363636364, 'gini = 0.0\nsamples = 20\nvalue = [20, 0]'),
```

```
 Text(121.93468656033261, 354.1636363636364, 'gini = 0.0\nsamples = 20\nvalue = [20, 0]'),
 Text(122.61576702287941, 354.1636363636364, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(122.9563072541528, 370.6363636363636, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),
 Text(123.29684748542621, 387.1090909090909, 'gini = 0.0\nsamples = 259\nvalue = [259, 0]'),
 Text(127.76643802088961, 420.05454545454546, 'X[6] <= 0.088\ngini = 0.214\nsamples = 690\nvalue = [606,
84]'),
 Text(124.65900841051982, 403.5818181818182, 'X[5] <= -0.763\ngini = 0.444\nsamples = 78\nvalue = [52,
26]'),
 Text(123.97792794797301, 387.1090909090909, 'X[5] <= -0.872\ngini = 0.311\nsamples = 26\nvalue = [5,
21]'),
 Text(123.6373877166996, 370.6363636363636, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(124.31846817924642, 370.6363636363636, 'X[6] <= -0.019\ngini = 0.087\nsamples = 22\nvalue = [1,
21]'),
 Text(123.97792794797301, 354.1636363636364, 'X[6] <= -0.034\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(123.6373877166996, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(124.31846817924642, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(124.65900841051982, 354.1636363636364, 'gini = 0.0\nsamples = 20\nvalue = [0, 20]'),
 Text(125.34008887306662, 387.1090909090909, 'X[1] <= -0.283\ngini = 0.174\nsamples = 52\nvalue = [47,
5]'),
 Text(124.99954864179323, 370.6363636363636, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),
 Text(125.68062910434003, 370.6363636363636, 'X[1] <= -0.255\ngini = 0.496\nsamples = 11\nvalue = [6,
5]'),
 Text(125.34008887306662, 354.1636363636364, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(126.02116933561342, 354.1636363636364, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(130.8738676312594, 403.5818181818182, 'X[6] <= 1.258\ngini = 0.172\nsamples = 612\nvalue = [554, 58
]'),
 Text(127.72387049198043, 387.1090909090909, 'X[10] <= 0.084\ngini = 0.107\nsamples = 407\nvalue = [384,
23]'),
 Text(127.38333026070704, 370.6363636363636, 'X[6] <= 0.253\ngini = 0.239\nsamples = 166\nvalue = [143,
23]'),
 Text(126.70224979816024, 354.1636363636364, 'X[8] <= 0.228\ngini = 0.49\nsamples = 35\nvalue = [20,
15]'),
 Text(126.36170956688683, 337.69090909090914, 'X[11] <= 1.02\ngini = 0.408\nsamples = 21\nvalue = [6, 15
]'),
 Text(126.02116933561342, 321.21818181818185, 'X[1] <= -0.483\ngini = 0.278\nsamples = 18\nvalue = [3, 1
5]'),
 Text(125.68062910434003, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(126.36170956688683, 304.74545454545455, 'X[7] <= 0.886\ngini = 0.208\nsamples = 17\nvalue = [2, 15
]'),
 Text(126.02116933561342, 288.2727272727273, 'X[1] <= -0.264\ngini = 0.117\nsamples = 16\nvalue = [1,
15]'),
 Text(125.68062910434003, 271.8, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),
 Text(126.36170956688683, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(126.70224979816024, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(126.70224979816024, 321.21818181818185, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(127.04279002943363, 337.69090909090914, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(128.06441072325384, 354.1636363636364, 'X[5] <= 0.618\ngini = 0.115\nsamples = 131\nvalue = [123,
8]'),
 Text(127.72387049198043, 337.69090909090914, 'gini = 0.0\nsamples = 111\nvalue = [111, 0]'),
 Text(128.40495095452724, 337.69090909090914, 'X[5] <= 1.159\ngini = 0.48\nsamples = 20\nvalue = [12,
8]'),
 Text(127.72387049198043, 321.21818181818185, 'X[6] <= 0.341\ngini = 0.346\nsamples = 9\nvalue = [2, 7]'
),
 Text(127.38333026070704, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(128.06441072325384, 304.74545454545455, 'X[2] <= 1.529\ngini = 0.219\nsamples = 8\nvalue = [1, 7]'
),
 Text(127.72387049198043, 288.2727272727273, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(128.40495095452724, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(129.08603141707405, 321.21818181818185, 'X[1] <= -0.276\ngini = 0.165\nsamples = 11\nvalue = [10,
1]'),
 Text(128.74549118580066, 304.74545454545455, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(129.42657164834745, 304.74545454545455, 'X[6] <= 0.968\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(129.08603141707405, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(129.76711187962084, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(128.06441072325384, 370.6363636363636, 'gini = 0.0\nsamples = 241\nvalue = [241, 0]'),
 Text(134.02386477053838, 387.1090909090909, 'X[5] <= -0.527\ngini = 0.283\nsamples = 205\nvalue = [170,
35]'),
 Text(132.83197396108147, 370.6363636363636, 'X[5] <= -0.573\ngini = 0.392\nsamples = 101\nvalue = [74,
27]'),
 Text(132.15089349853466, 354.1636363636364, 'X[8] <= 0.228\ngini = 0.126\nsamples = 74\nvalue = [69,
5]'),
 Text(131.81035326726126, 337.69090909090914, 'X[5] <= -0.797\ngini = 0.229\nsamples = 38\nvalue = [33,
5]'),
 Text(131.46981303598787, 321.21818181818185, 'X[5] <= -0.83\ngini = 0.375\nsamples = 20\nvalue = [15, 5
]'),
 Text(130.78873257344105, 304.74545454545455, 'X[6] <= 2.457\ngini = 0.124\nsamples = 15\nvalue = [14, 1
]'),
```

```
 Text(130.44819234216766, 288.2727272727273, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(131.12927280471447, 288.2727272727273, 'X[1] <= -0.476\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(130.78873257344105, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(131.46981303598787, 271.8, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(132.15089349853466, 304.74545454545455, 'X[1] <= -0.42\ngini = 0.32\nsamples = 5\nvalue = [1,
4]'),
 Text(131.81035326726126, 288.2727272727273, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(132.49143372980808, 288.2727272727273, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(132.15089349853466, 321.21818181818185, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),
 Text(132.49143372980808, 337.69090909090914, 'gini = 0.0\nsamples = 36\nvalue = [36, 0]'),
 Text(133.5130544236283, 354.1636363636364, 'X[8] <= 0.228\ngini = 0.302\nsamples = 27\nvalue = [5,
22]'),
 Text(133.17251419235487, 337.69090909090914, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(133.85359465490168, 337.69090909090914, 'X[10] <= 0.084\ngini = 0.083\nsamples = 23\nvalue = [1, 2
2]'),
 Text(133.5130544236283, 321.21818181818185, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(134.19413488617508, 321.21818181818185, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(135.2157555799953, 370.6363636363636, 'X[10] <= 0.084\ngini = 0.142\nsamples = 104\nvalue = [96,
8]'),
 Text(134.8752153487219, 354.1636363636364, 'X[5] <= 0.608\ngini = 0.339\nsamples = 37\nvalue = [29,
8]'),
 Text(134.53467511744847, 337.69090909090914, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),
 Text(135.2157555799953, 337.69090909090914, 'X[5] <= 0.93\ngini = 0.488\nsamples = 19\nvalue = [11,
8]'),
 Text(134.8752153487219, 321.21818181818185, 'X[1] <= -0.473\ngini = 0.198\nsamples = 9\nvalue = [1,
8]'),
 Text(134.53467511744847, 304.74545454545455, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(135.2157555799953, 304.74545454545455, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(135.55629581126868, 321.21818181818185, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(135.55629581126868, 354.1636363636364, 'gini = 0.0\nsamples = 67\nvalue = [67, 0]'),
 Text(491.4811728851611, 518.8909090909091, 'X[1] <= 0.485\ngini = 0.42\nsamples = 37097\nvalue = [11128,
25969]'),
 Text(303.719537008691, 502.41818181818184, 'X[8] <= 0.228\ngini = 0.497\nsamples = 14526\nvalue =
[6658, 7868]'),
 Text(244.56843254439679, 485.9454545454546, 'X[11] <= 1.02\ngini = 0.465\nsamples = 9118\nvalue =
[3351, 5767]'),
 Text(209.26600439319338, 469.4727272727273, 'X[10] <= 0.084\ngini = 0.439\nsamples = 8078\nvalue =
[2632, 5446]'),
 Text(171.28267410268717, 453.0, 'X[5] <= -0.058\ngini = 0.271\nsamples = 3691\nvalue = [596, 3095]'),
 Text(149.1898771797105, 436.52727272727276, 'X[6] <= -0.94\ngini = 0.362\nsamples = 1749\nvalue = [415,
1334]'),
 Text(138.55730659936555, 420.05454545454546, 'X[7] <= 0.129\ngini = 0.479\nsamples = 68\nvalue = [41, 2
7]'),
 Text(137.02487555863524, 403.5818181818182, 'X[6] <= -0.985\ngini = 0.175\nsamples = 31\nvalue = [28,
3]'),
 Text(136.68433532736185, 387.1090909090909, 'gini = 0.0\nsamples = 24\nvalue = [24, 0]'),
 Text(137.36541578990864, 387.1090909090909, 'X[1] <= -0.131\ngini = 0.49\nsamples = 7\nvalue = [4,
3]'),
 Text(137.02487555863524, 370.6363636363636, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(137.70595602118203, 370.6363636363636, 'X[5] <= -0.177\ngini = 0.32\nsamples = 5\nvalue = [4,
1]'),
 Text(137.36541578990864, 354.1636363636364, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(138.04649625245546, 354.1636363636364, 'X[1] <= 0.22\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(137.70595602118203, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(138.38703648372885, 337.69090909090914, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(140.08973764009585, 403.5818181818182, 'X[1] <= 0.079\ngini = 0.456\nsamples = 37\nvalue = [13,
24]'),
 Text(139.74919740882245, 387.1090909090909, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 ...]
```





Performing Cross Validation on DT Model.

Performing cross validation on the dataset using StratifiedKFold and calculating the mean Accuracy that can be achieved by the model.

In [397]:

```
x = pd.DataFrame(data = x_train_ss, columns = inde_vars.columns)
y = y_train
from sklearn.model_selection import  StratifiedKFold
accuracy = []
skf = StratifiedKFold(n_splits = 10, random_state = None)
skf.get_n_splits(x,y)
for train_index, test_index in skf.split(x,y):
  print('Train:', train_index, 'Validation',test_index)
  x1_train,x1_test = x.iloc[train_index],x.iloc[test_index]
  y1_train,y1_test = y.iloc[train_index],y.iloc[test_index]
  dt.fit(x1_train,y1_train)
  pred = dt.predict(x1_test)
  score = accuracy_score(pred,y1_test)
  accuracy.append(score)
print(accuracy)
```

```
Train: [ 5595  5596  5597 ... 56655 56656 56657] Validation [    0     1     2 ... 5749 5750 5752]
Train: [    0     1     2 ... 56655 56656 56657] Validation [ 5595  5596  5597 ... 11348 11351 11352]
Train: [    0     1     2 ... 56655 56656 56657] Validation [11305 11306 11308 ... 17019 17022 17024]
Train: [    0     1     2 ... 56655 56656 56657] Validation [16980 16983 16984 ... 22681 22685 22689]
Train: [    0     1     2 ... 56655 56656 56657] Validation [22637 22642 22646 ... 28498 28499 28500]
Train: [    0     1     2 ... 56655 56656 56657] Validation [28164 28165 28166 ... 34208 34210 34211]
Train: [    0     1     2 ... 56655 56656 56657] Validation [33785 33786 33788 ... 39838 39839 39840]
Train: [    0     1     2 ... 56655 56656 56657] Validation [39486 39487 39490 ... 45432 45434 45435]
Train: [    0     1     2 ... 56655 56656 56657] Validation [45222 45224 45226 ... 51006 51007 51008]
Train: [    0     1     2 ... 51006 51007 51008] Validation [50979 50980 50981 ... 56655 56656 56657]
[0.9532297917402047, 0.9417578538651606, 0.952347334980586, 0.9528768090363572, 0.9514648782209671,
0.9505824214613484, 0.9511118955171196, 0.9505824214613484, 0.9525154457193292, 0.9526919682259488]
```

In [398]:

```
arr = np.array(accuracy)
```

In [399]:

```
np.mean(arr)
```

Out[399]:

```
0.9509160820228371
```

Hyper Parameter Tuning the model to overcome Overfitting model.

Determining the parameters by plotting f1_score metrics.

1. Function to calculate f1_score.
2. Function to plot the f1_score that we have calculated.
3. Pass the parameter values in the model and call the functions.

In [400]:

```
def cal_score(model, x1,y1,x2,y2):
  model.fit(x1,y1)
  p = model.predict(x1)
  f1 = f1_score(y1, p)
  p1 = model.predict(x2)
  f2 = f1_score(y2,p1)
```

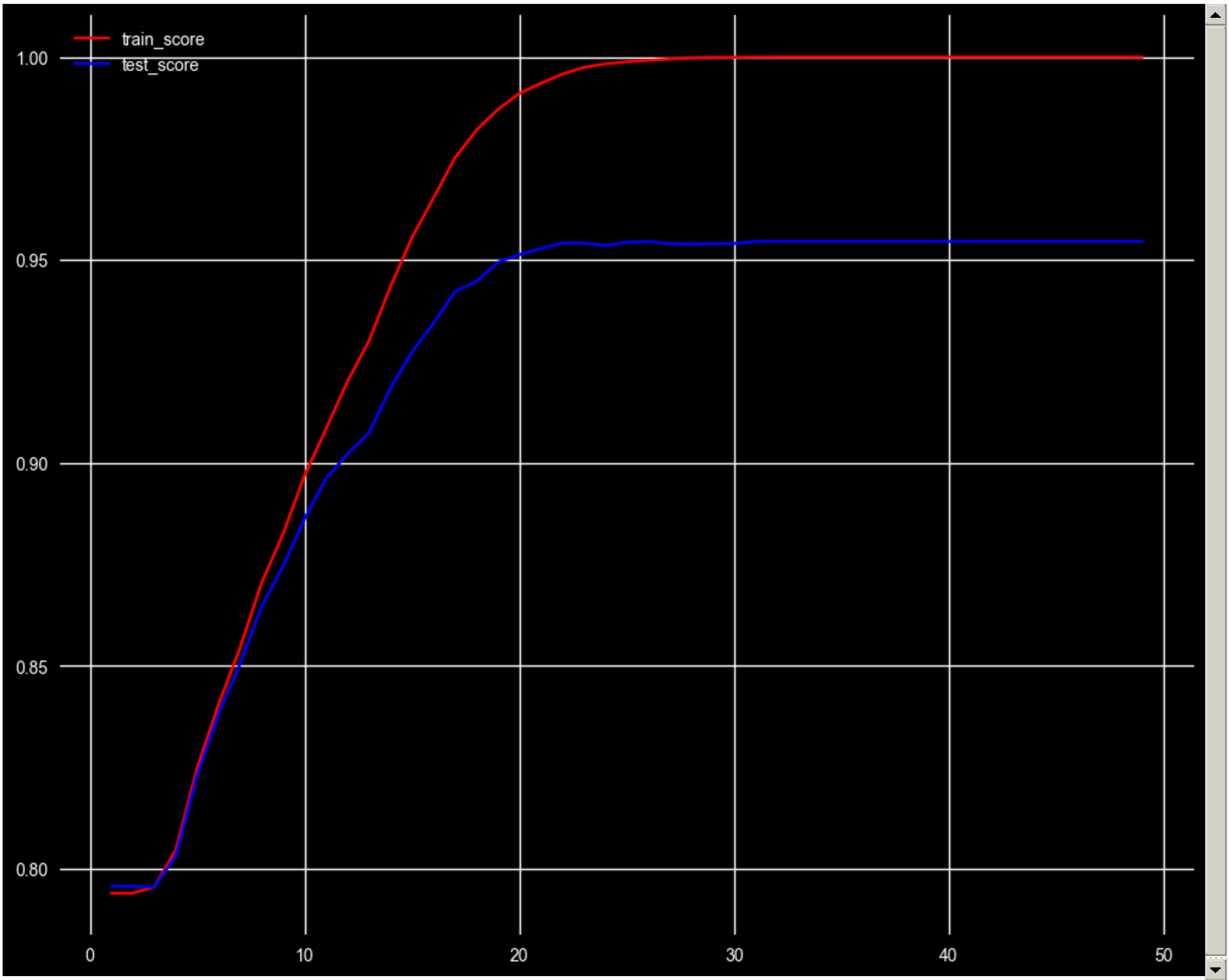
```
  return f1,f2
```

In [401]:

```
def effect(train, test, x_axis, title):
  plt.figure(figsize = (12,10), dpi = 100)
  plt.plot(x_axis, train, color = 'red', label = 'train_score')
  plt.plot(x_axis, test, color = 'blue', label = 'test_score')
  plt.legend()
  plt.show()
```

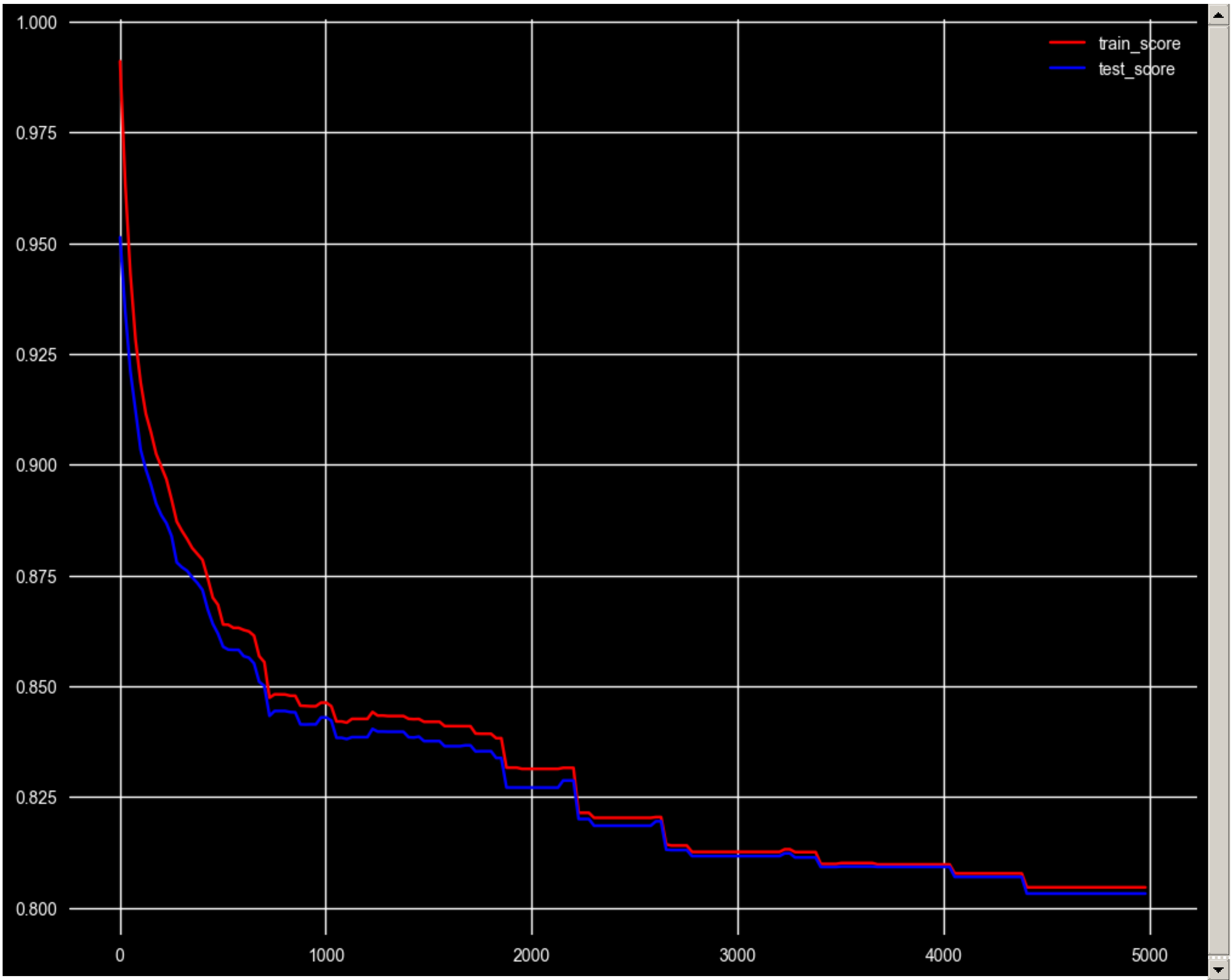
In [402]:

```
max_depth = [i for i in range(1,50)]
train = []
test = []
for i in max_depth:
  model =DecisionTreeClassifier(max_depth=i, random_state=50)
  f1,f2 = cal_score(model, x_train, y_train, x_test, y_test)
  train.append(f1)
  test.append(f2)
effect(train,test, range(1,50), 'Max_Depth')
```



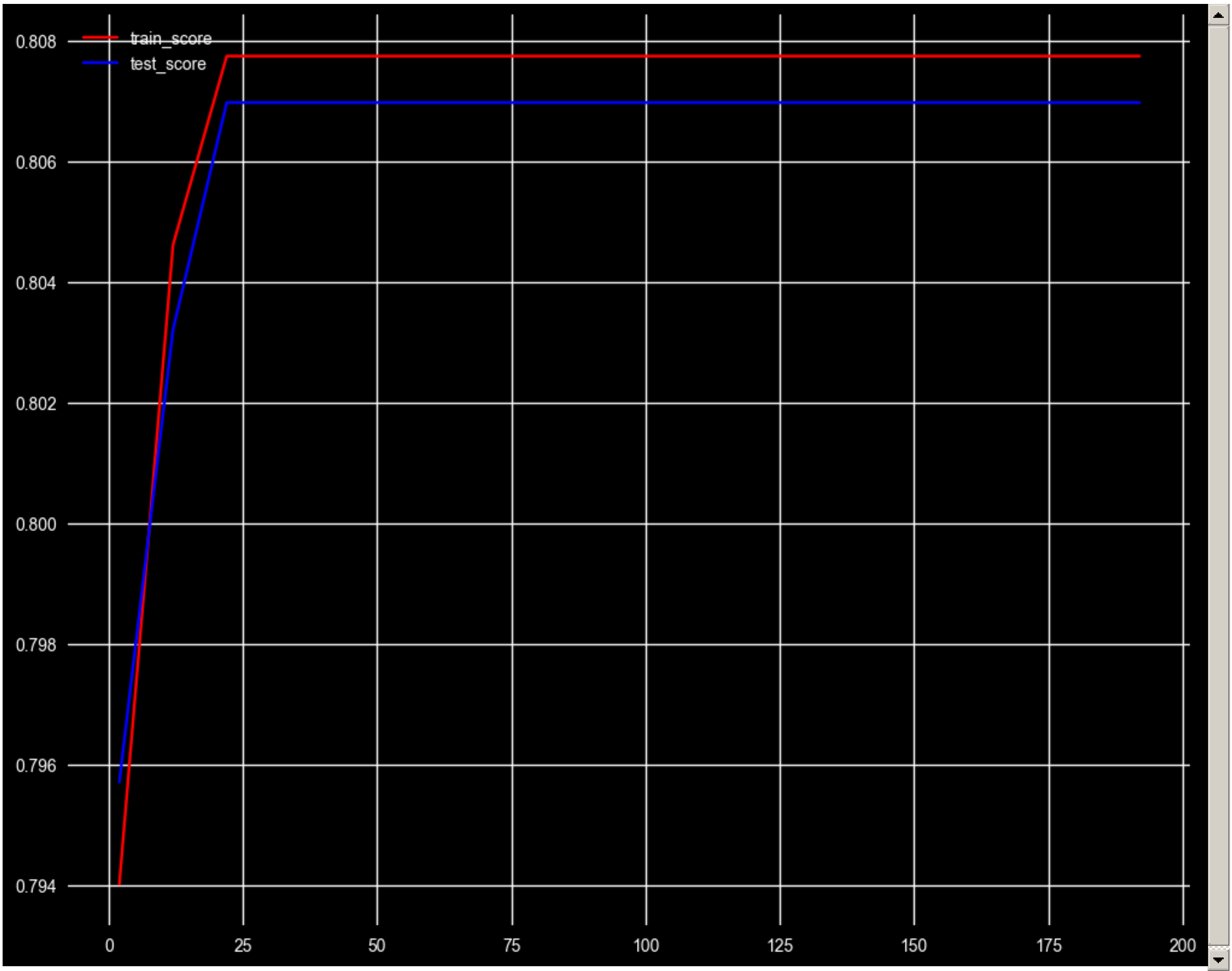
In [403]:

```
min_samples = [i for i in range(2,5000,25)]
train = []
test = []
for i in min_samples:
  model =DecisionTreeClassifier(max_depth=20, random_state=50, min_samples_split=i)
  f1,f2 = cal_score(model, x_train, y_train, x_test, y_test)
  train.append(f1)
  test.append(f2)
effect(train,test, range(2,5000,25), 'Min_Samples_Split')
```



In [404]:

```
max_leaf = [i for i in range(2,200,10)]
train = []
test = []
for i in max_leaf:
  model =DecisionTreeClassifier(max_depth=20,min_samples_split=4250,max_leaf_nodes=i, random_state=50)
  f1,f2 = cal_score(model, x_train, y_train, x_test, y_test)
  train.append(f1)
  test.append(f2)
effect(train,test, range(2,200,10), 'Max_Leaf_Nodes')
```



Hyper Parameter Tuning the model by using roc_auc_curve.

In [405]:

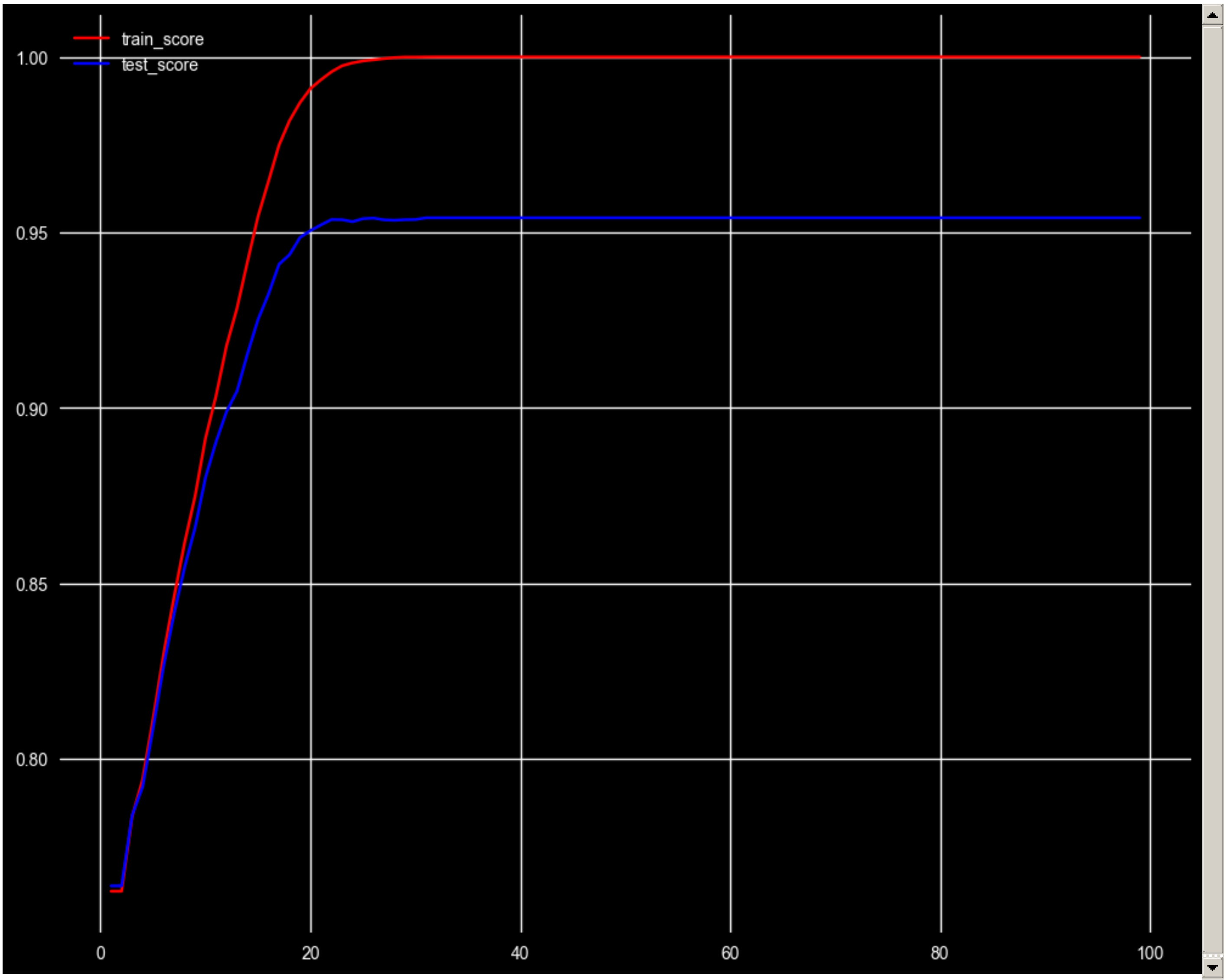
```
def cal_score1(model, x1,y1,x2,y2):
  model.fit(x1,y1)
  p = model.predict(x1)
  false_positive_rate, true_positive_rate, thresholds = roc_curve(y1, p)
  roc_auc_1 = auc(false_positive_rate, true_positive_rate)
  p1 = model.predict(x2)
  false_positive_rate, true_positive_rate, thresholds = roc_curve(y2, p1)
  roc_auc_2 = auc(false_positive_rate, true_positive_rate)
  return roc_auc_1,roc_auc_2
```

In [406]:

```
def effect1(train, test, x_axis, title):
  plt.figure(figsize = (12,10), dpi = 100)
  plt.plot(x_axis, train, color = 'red', label = 'train_score')
  plt.plot(x_axis, test, color = 'blue', label = 'test_score')
  plt.legend()
  plt.show()
```

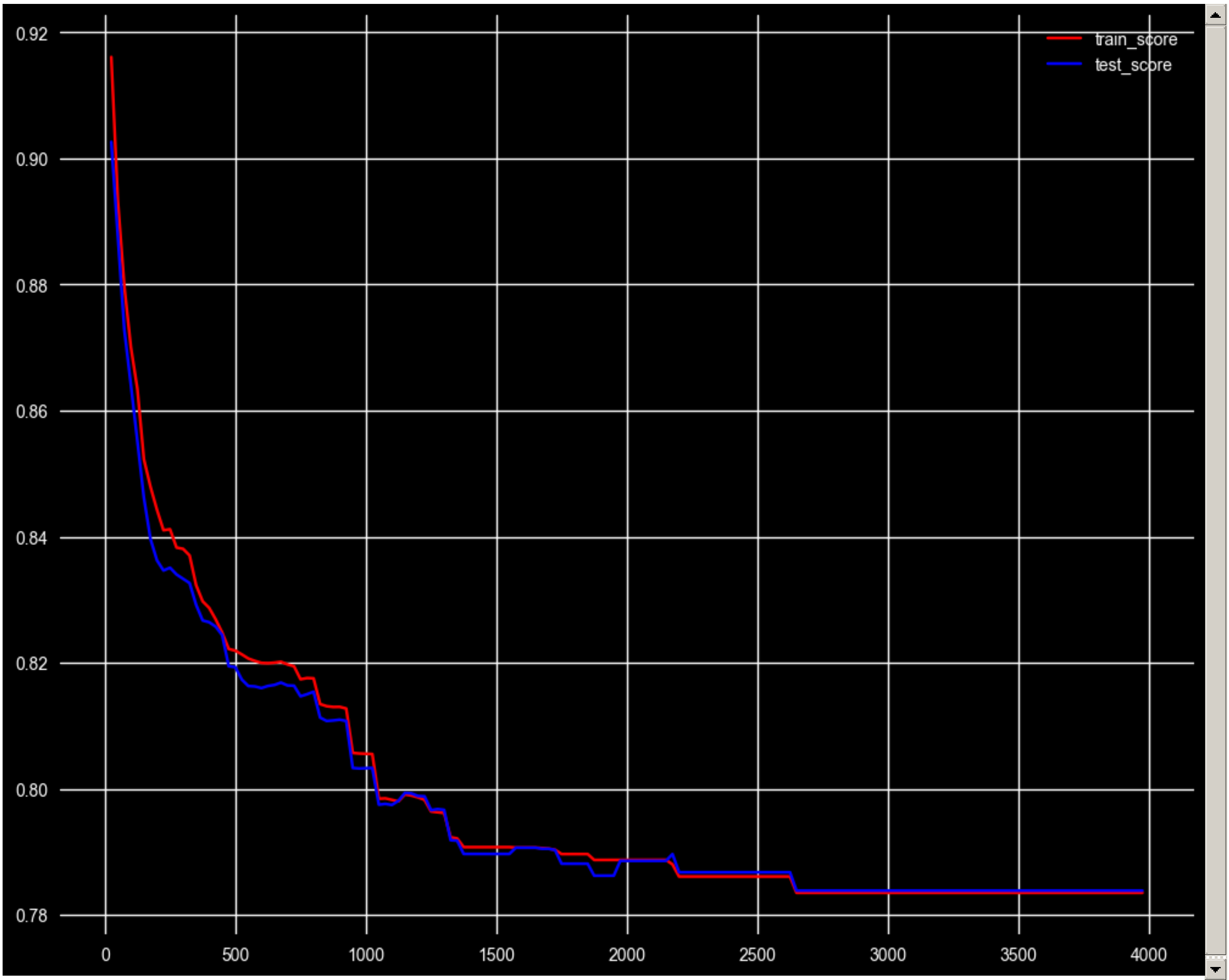
In [407]:

```
max_depth = [i for i in range(1,100)]
train = []
test = []
for i in max_depth:
  roc_auc_model =DecisionTreeClassifier(max_depth=i, random_state=50)
  roc_auc_1,roc_auc_2 = cal_score1(roc_auc_model, x_train, y_train, x_test, y_test)
  train.append(roc_auc_1)
  test.append(roc_auc_2)
effect1(train,test, range(1,100), 'Max_Depth')
```



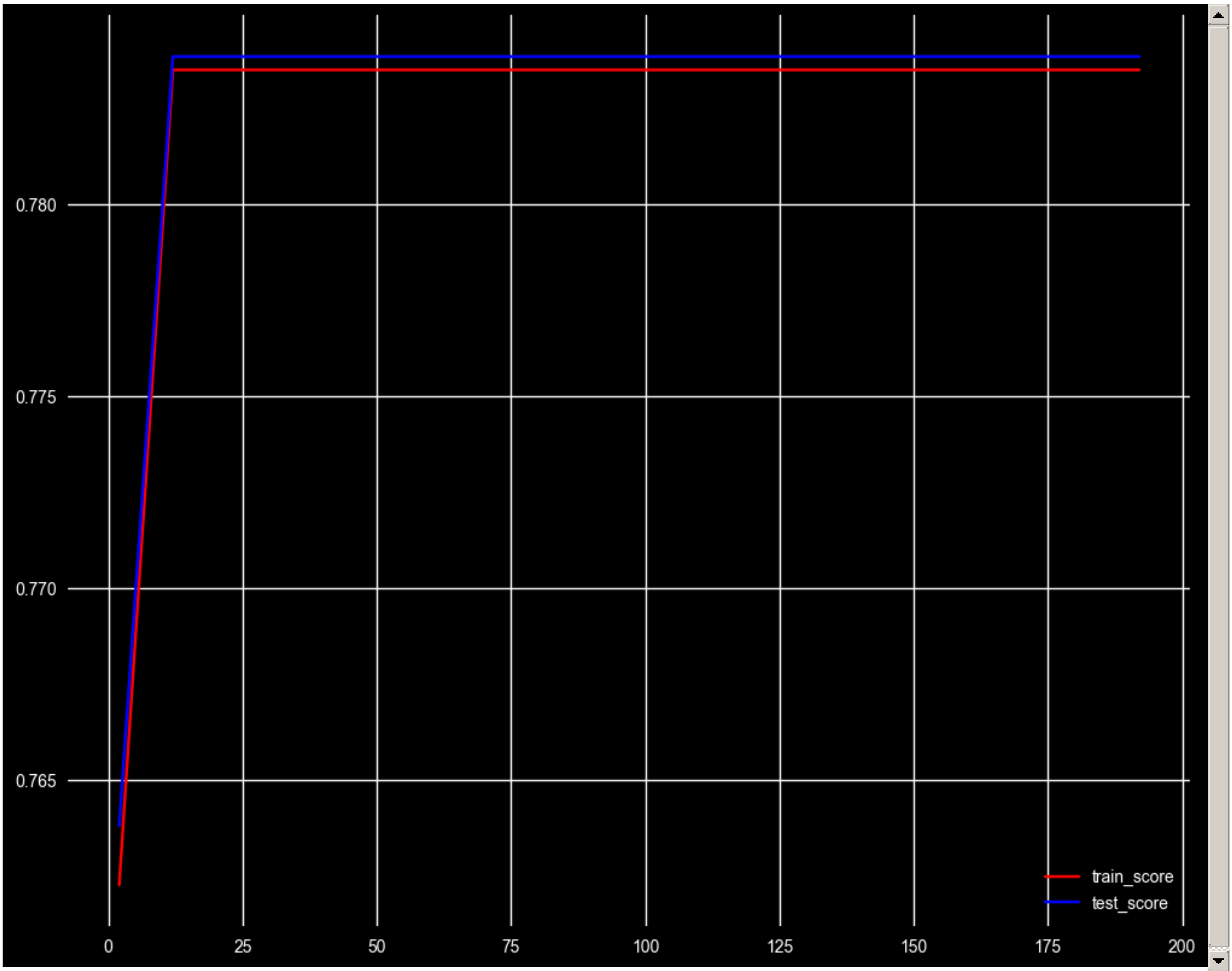
In [408]:

```
min_sample_leaff = [i for i in range(25,4000,25)]
train = []
test = []
for i in min_sample_leaff:
  roc_auc_model =DecisionTreeClassifier(max_depth=20, min_samples_leaf=i, random_state=50)
  roc_auc_1,roc_auc_2 = cal_score1(roc_auc_model, x_train, y_train, x_test, y_test)
  train.append(roc_auc_1)
  test.append(roc_auc_2)
effect1(train,test, range(25,4000,25), 'Min_Samples_Leaf')
```



In [409]:

```
max_leaf_node = [i for i in range(2,200,10)]
train = []
test = []
for i in max_leaf_node:
  roc_auc_model =DecisionTreeClassifier(max_depth=20,max_leaf_nodes=i, min_samples_leaf=3700, random_stat
  roc_auc_1,roc_auc_2 = cal_score1(roc_auc_model, x_train, y_train, x_test, y_test)
  train.append(roc_auc_1)
  test.append(roc_auc_2)
effect1(train,test, range(2,200,10), 'Max_Leaf_Nodes')
```



In [ ]:

Hyper parameter Tuning the model using ccp(cost complexity pruning)

which helps us to select the best values for max_depth and max_samples_leaf parameter for Decision Tree.

In [410]:

```
path = dt.cost_complexity_pruning_path(x_train_ss,y_train)
ccp_alphas, impurities = path.ccp_alphas, path.impurities
```

In [411]:

```
ccp_alphas
```

Out[411]:

```
array([0.00000000e+00, 1.06968232e-05, 1.10310989e-05, ...,
       9.63635249e-03, 2.11399782e-02, 1.52138383e-01])
```

In [412]:

```
clfs = []
for i in ccp_alphas:
   dt = DecisionTreeClassifier(random_state = 0, ccp_alpha=i)
   dt.fit(x_train_ss,y_train)
   clfs.append(dt)
   print('Number of Nodes in the Last Tree is: {} with ccp_alpha: {}'.format(clfs[-1].tree_.node_count, cc
```

```
Number of Nodes in the Last Tree is: 6063 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6063 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6057 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6051 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6033 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6033 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6033 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 6027 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6021 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6015 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6009 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 6003 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5997 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5991 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5985 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5979 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5973 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5967 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5961 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5955 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5949 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5943 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5937 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5931 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5925 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5919 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5907 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5901 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5889 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5889 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5889 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5879 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5869 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5861 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5861 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5855 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5845 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5841 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5831 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5821 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5805 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5805 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5805 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5805 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5777 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5763 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5741 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5725 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5725 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5725 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5725 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5677 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5661 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5661 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5661 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5661 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5649 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5601 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5601 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5601 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5601 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5601 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 5593 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5549 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5541 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5525 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5525 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5525 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5525 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5507 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5463 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5459 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5423 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5391 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5391 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5391 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5371 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5371 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5371 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5371 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5371 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5363 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5349 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5321 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5321 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5321 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5245 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5221 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 5221 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5221 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5221 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5217 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5205 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5201 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5197 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5185 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5185 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5185 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5177 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5177 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5177 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5157 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5149 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5129 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5121 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5121 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5105 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5105 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5105 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5105 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5101 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5093 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5077 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5077 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5073 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5061 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5061 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5061 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5057 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5049 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5041 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5041 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5037 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5029 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5029 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5021 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5013 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4997 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4997 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4997 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4989 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4977 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4969 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4969 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4965 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4957 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4953 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4949 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4945 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4941 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4921 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4921 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4921 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4921 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4921 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4913 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4913 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4909 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4901 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4897 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4885 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4885 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4885 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4877 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4873 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4865 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4849 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4849 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4849 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4845 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 4841 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4837 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4825 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4825 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4817 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4813 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4801 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4801 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4797 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4789 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4781 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4769 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4769 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4769 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4765 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4761 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4757 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4753 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4749 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4741 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4737 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4729 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4725 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4717 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4713 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4705 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4701 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4697 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4689 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4689 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4681 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4677 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4673 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4665 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4665 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4661 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4653 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4649 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4645 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4641 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4637 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4633 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4629 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4621 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4617 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4613 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4605 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4597 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4597 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4593 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4581 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4577 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4573 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4569 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4565 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4559 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4555 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4547 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4543 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4539 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4531 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4527 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4523 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4519 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4515 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4497 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 4493 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4489 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4485 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4481 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4473 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4469 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4465 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4461 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4455 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4433 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4421 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4417 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4405 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4395 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4389 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4387 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4367 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4351 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4351 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4351 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4351 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4345 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4339 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4327 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4321 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4315 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4309 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4303 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4297 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4291 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4279 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4259 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4235 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4229 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4221 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4215 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 4139 with ccp alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 4119 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4117 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4109 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4107 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4107 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4099 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4091 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4079 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4079 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4079 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4071 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4063 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4055 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4033 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4025 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4017 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4009 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 4007 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3955 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3953 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3949 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3943 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3937 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3933 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3933 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3927 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3913 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3909 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3903 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3893 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3881 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3881 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3869 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.1521383836474234
```

```
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3787 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3781 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3769 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3761 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3761 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3757 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3755 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3751 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3751 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3745 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3739 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3713 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3711 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3705 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3693 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3657 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3653 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3651 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3649 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3647 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3645 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3643 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3643 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3639 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 3609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3595 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3595 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3593 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3587 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3587 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3587 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3579 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3579 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3579 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3579 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3579 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3551 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3543 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3535 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3533 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3529 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3527 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3523 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3503 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3499 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3499 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3499 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3495 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3493 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3491 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3485 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3485 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3485 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3479 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3479 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3475 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3469 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3463 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3433 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3431 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3429 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3425 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3421 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3421 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3405 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3397 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3387 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3387 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3387 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 3387 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3387 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3379 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3375 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3365 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3363 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3357 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3357 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3357 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3357 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3347 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3347 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3347 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3347 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3347 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3343 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3339 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3317 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3317 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3315 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3311 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3311 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3307 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3305 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3275 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3267 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3255 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3255 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3255 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3255 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3247 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3229 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3229 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3229 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3229 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3225 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3211 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3207 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3205 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3201 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3193 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3189 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3185 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3179 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3177 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3175 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3163 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3161 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3153 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3153 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3135 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3129 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3129 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3125 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 3125 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3121 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3117 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3113 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3103 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3099 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3091 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3091 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3091 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3087 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3085 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3079 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3077 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3069 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3069 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3069 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3069 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3063 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3063 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3061 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3057 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3053 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3051 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3047 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3043 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3041 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3039 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3035 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3031 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3029 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3029 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3025 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3021 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3019 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3015 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3009 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3003 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 3001 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2997 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2993 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2991 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2989 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2987 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2985 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2981 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2981 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2979 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2977 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2975 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2971 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2971 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2969 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2967 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2963 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2961 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2953 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2949 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2947 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2943 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2933 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2931 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2927 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2925 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2917 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2905 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2903 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2901 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2901 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2897 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2891 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2889 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2887 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2887 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2885 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2881 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2881 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2875 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 2875 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2869 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2863 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2853 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2851 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2849 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2843 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2839 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2835 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2835 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2821 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2819 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2813 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2807 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2807 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2807 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2799 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2791 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2791 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2791 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2789 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2787 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2783 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2767 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2755 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2753 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2751 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2743 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2741 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2731 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2727 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2707 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2705 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2703 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2699 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2699 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2695 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2687 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2687 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2657 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2655 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2643 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2633 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2633 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2633 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2633 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2623 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2613 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2601 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2599 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2591 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2587 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2585 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2583 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2579 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2575 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2571 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2567 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2559 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2557 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2549 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2549 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2545 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2541 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2529 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2527 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2523 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2513 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2513 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2513 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2513 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2513 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2509 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2507 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 2507 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2477 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2473 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2469 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2467 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2457 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2453 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2449 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2445 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2425 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2423 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2421 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2417 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2415 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2397 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2387 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2381 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2379 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2379 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2369 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2367 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2361 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2357 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2353 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2353 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2339 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2331 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2327 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2323 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2317 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2317 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2313 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2311 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2309 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2303 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2303 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2299 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2297 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2295 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2289 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2281 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2271 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2263 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2259 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2249 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2241 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2237 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2229 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2225 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2223 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2205 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2203 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2201 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2197 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2195 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2189 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2189 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2189 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2187 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2181 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2179 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2177 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2175 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2169 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2163 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2161 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2161 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2157 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2157 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2153 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2151 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2149 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2147 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2139 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 2139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2137 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2133 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2133 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2129 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2129 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2125 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2123 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2121 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2119 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2115 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2095 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2093 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2091 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2085 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2083 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2079 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2075 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2075 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2073 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2063 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2061 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2057 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2055 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2051 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2049 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2047 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2045 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2039 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2035 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2033 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2027 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2027 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2027 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2025 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2017 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2013 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2011 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2007 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2005 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2001 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 2001 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1995 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1991 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1991 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1989 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1987 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1985 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1983 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1981 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1981 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1979 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1975 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1969 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1965 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1961 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1957 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1957 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1953 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1951 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1949 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1947 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1945 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1941 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1937 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1935 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1933 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1931 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1921 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1919 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1915 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1911 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1909 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1899 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1887 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1885 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1883 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1875 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 1875 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1869 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1869 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1851 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1849 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1847 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1841 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1839 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1835 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1821 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1819 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1817 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1815 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1811 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1807 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1803 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1801 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1799 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1793 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1791 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1783 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1779 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1773 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1771 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1769 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1767 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1763 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1761 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1759 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1753 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1749 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1749 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1741 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1739 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1733 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1725 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1723 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1715 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1713 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1711 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1699 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1697 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1693 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1691 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1687 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1681 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1679 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1673 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1667 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1667 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1667 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1663 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1659 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1653 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1651 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1649 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1645 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1641 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1635 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1631 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1629 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1627 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1625 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1621 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1619 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1613 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1611 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1609 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1605 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1603 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1601 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1595 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1593 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1589 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1587 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1581 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1575 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1571 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 1571 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1569 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1563 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1561 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1559 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1557 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1555 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1543 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1539 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1535 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1531 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1505 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1503 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1501 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1501 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1495 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1491 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1489 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1487 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1485 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1479 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1475 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1467 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1467 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1465 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1463 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1461 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1459 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1457 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1455 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1443 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1439 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1437 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1425 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1415 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1413 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1405 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1405 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1391 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1389 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1377 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1369 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1367 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1361 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1359 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1357 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1355 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1353 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1351 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1349 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1347 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1345 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1335 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1333 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1331 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1329 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1325 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1323 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1313 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1309 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1305 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1301 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1301 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1297 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1295 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1293 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1291 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1287 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1281 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1279 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1273 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 1269 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1265 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1265 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1263 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1261 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1257 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1253 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1249 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1247 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1245 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1243 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1241 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1237 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1231 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1227 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1225 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1215 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1213 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1211 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1211 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1209 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1207 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1205 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1203 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1199 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1185 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1181 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1179 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1177 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1175 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1171 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1167 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1165 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1163 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1157 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1155 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1153 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1149 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1147 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1143 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1131 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1129 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1125 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1119 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1115 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1113 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1111 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1107 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1105 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1099 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1097 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1091 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1089 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1087 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1087 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1085 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1083 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1081 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1077 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1073 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1067 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1065 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1063 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1055 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1051 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1051 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1049 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1045 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1041 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1039 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1037 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1035 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1021 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1017 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1015 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 1013 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1011 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1009 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1007 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1005 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1003 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1001 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 999 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 997 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 991 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 979 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 977 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 973 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 969 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 967 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 963 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 961 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 959 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 957 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 955 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 953 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 951 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 947 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 943 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 941 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 939 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 937 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 933 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 919 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 899 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 897 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 895 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 891 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 889 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 883 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 881 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 879 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 877 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 873 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 871 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 869 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 865 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 861 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 859 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 857 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 855 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 853 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 847 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 843 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 841 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 839 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 835 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 833 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 829 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 827 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 825 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 823 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 821 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 821 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 817 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 815 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 813 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 807 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 801 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 799 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 787 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 783 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 779 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 765 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 761 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 759 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 757 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 753 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 747 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 745 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 741 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 739 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 737 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 733 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 729 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 729 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 727 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 723 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 719 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 715 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 713 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 711 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 709 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 707 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 705 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 701 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 699 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 693 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 689 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 687 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 685 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 683 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 681 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 679 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 677 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 669 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 665 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 663 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 659 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 657 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 655 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 651 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 647 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 645 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 637 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 635 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 633 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 627 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 605 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 597 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 595 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 587 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 585 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 581 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 575 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 573 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 561 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 557 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 555 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 553 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 545 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 543 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 539 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 533 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 531 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 529 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 523 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 521 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 519 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 517 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 513 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 507 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 505 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 503 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 501 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 499 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 497 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 493 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 491 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 487 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 485 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 483 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 481 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 479 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 477 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 471 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 469 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 461 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 459 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 453 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 445 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 441 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 439 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 437 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 435 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 433 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 427 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 425 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 423 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 421 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 419 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 411 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 407 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 403 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 399 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 397 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 395 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 391 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 389 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 385 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 383 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 379 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 375 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 373 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 371 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 369 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 367 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 363 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 359 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 355 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 353 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 351 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 345 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 341 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 337 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 323 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 321 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 317 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 315 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 311 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 309 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 307 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 305 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 303 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 299 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 297 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 287 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 285 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 283 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 281 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 277 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 273 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 271 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 267 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 263 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 261 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 259 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 257 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 257 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 251 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 247 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 243 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 239 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 235 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 233 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 227 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 225 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 221 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 219 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 213 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 207 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 203 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 201 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 199 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 193 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 189 with ccp_alpha: 0.15213838336474234
```

```
Number of Nodes in the Last Tree is: 187 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 185 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 183 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 159 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 159 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 151 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 149 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 147 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 143 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 141 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 139 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 137 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 135 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 133 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 131 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 127 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 125 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 119 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 117 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 115 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 113 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 111 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 109 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 105 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 101 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 99 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 95 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 91 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 89 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 87 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 75 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 73 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 71 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 69 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 67 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 65 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 61 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 59 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 57 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 55 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 53 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 51 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 49 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 47 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 45 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 43 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 41 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 37 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 35 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 33 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 31 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 27 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 25 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 23 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 19 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 15 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 11 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 9 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 7 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 5 with ccp_alpha: 0.15213838336474234
Number of Nodes in the Last Tree is: 1 with ccp_alpha: 0.15213838336474234
```

Plotting a graph with Respect to Accuracy score and various clfs(classifiers)

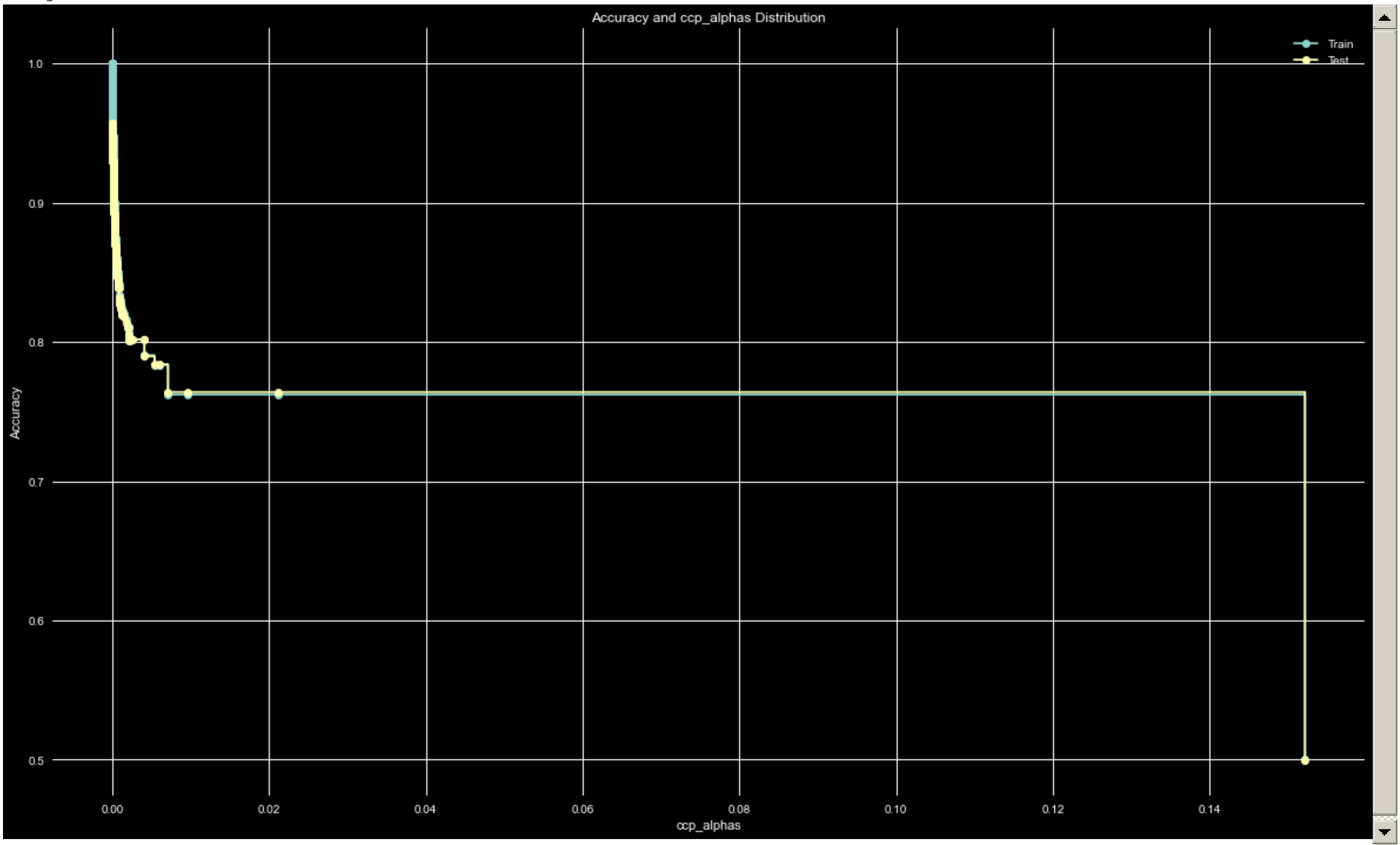
In [413]:

```
train_set = [dt.score(x_train_ss,y_train) for dt in clfs]
test_set = [dt.score(x_test_ss,y_test) for dt in clfs]

plt.figure(figsize = (12,10), dpi = 100)
fig,ax = plt.subplots()
ax.plot(ccp_alphas, train_set, marker = 'o', label = 'Train', drawstyle = 'steps-post')
ax.plot(ccp_alphas, test_set, marker = 'o', label = 'Test', drawstyle = 'steps-post')
ax.set_xlabel('ccp_alphas')
ax.set_ylabel('Accuracy')
ax.set_title("Accuracy and ccp_alphas Distribution")
ax.legend()
```

```
plt.show()
```

```
<Figure size 1200x1000 with 0 Axes>
```



So, After applying Hyper Parameter Tuning **With Respect to Evaluation Metrics**, our model has successfully overcomed the problem of overfitting which has occured earlier.

In [414]:

```
modified_model = DecisionTreeClassifier(max_depth = 18, min_samples_split=4250, min_samples_leaf=3700, ma
modified_model.fit(x_train_ss, y_train)
pr = modified_model.predict(x_test_ss)
```

In [415]:

```
print(modified_model.score(x_train_ss,y_train))
print(modified_model.score(x_test_ss, y_test))
print(accuracy_score(pr,y_test))
```

```
0.7834727664231
0.7839140103780579
0.7839140103780579
```

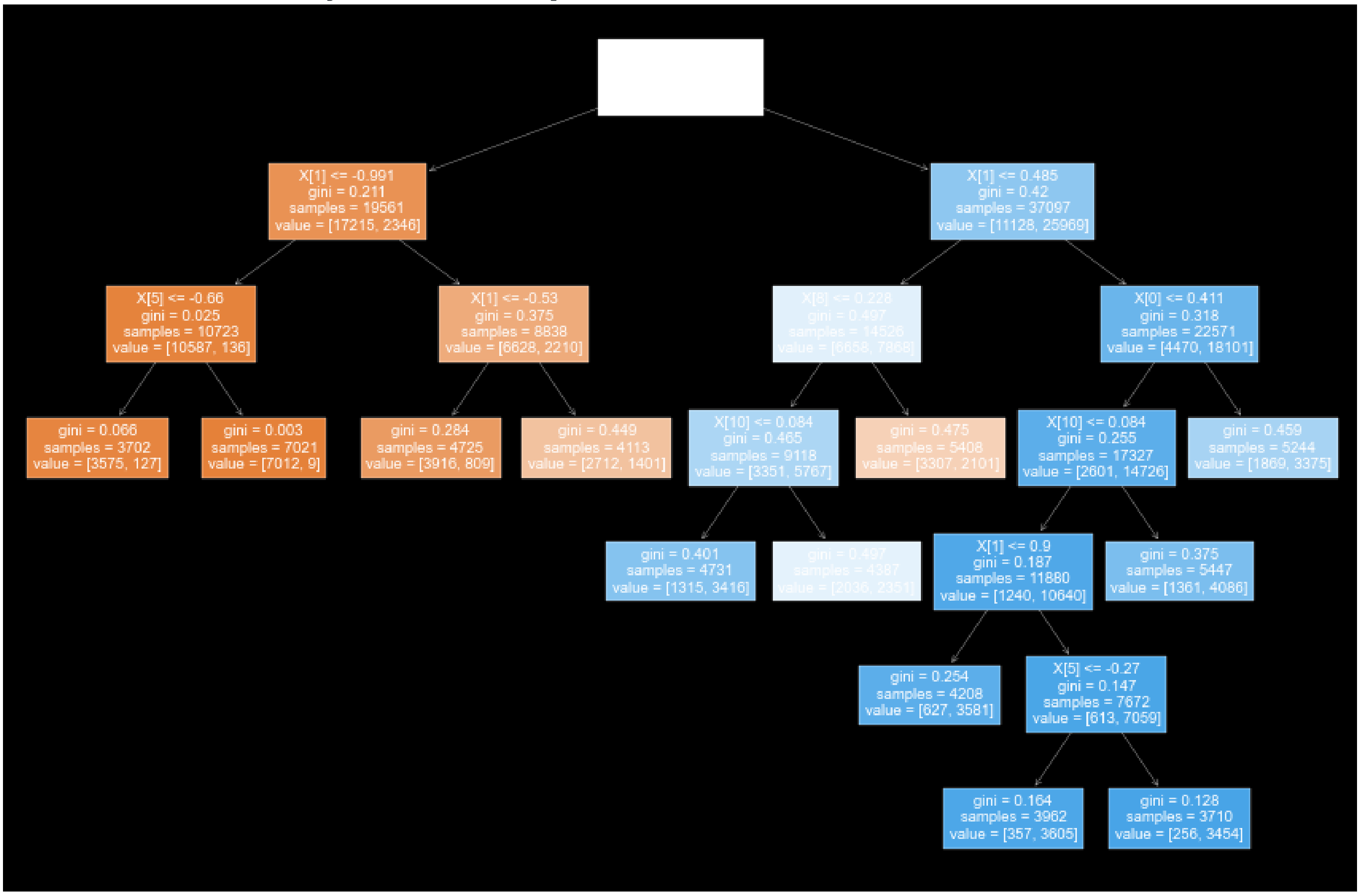
Tree Plot With Respect to Modified Model.

In [416]:

```
plt.figure(figsize = (15,10))
tree.plot_tree(modified_model, filled = True)
```

Out[416]:

```
[Text(418.5, 504.7714285714286, 'X[1] <= -0.253\ngini = 0.5\nsamples = 56658\nvalue = [28343, 28315]'),
 Text(209.25, 427.11428571428576, 'X[1] <= -0.991\ngini = 0.211\nsamples = 19561\nvalue = [17215,
2346]'),
 Text(104.625, 349.4571428571429, 'X[5] <= -0.66\ngini = 0.025\nsamples = 10723\nvalue = [10587, 136]'),
 Text(52.3125, 271.8, 'gini = 0.066\nsamples = 3702\nvalue = [3575, 127]'),
 Text(156.9375, 271.8, 'gini = 0.003\nsamples = 7021\nvalue = [7012, 9]'),
 Text(313.875, 349.4571428571429, 'X[1] <= -0.53\ngini = 0.375\nsamples = 8838\nvalue = [6628, 2210]'),
 Text(261.5625, 271.8, 'gini = 0.284\nsamples = 4725\nvalue = [3916, 809]'),
 Text(366.1875, 271.8, 'gini = 0.449\nsamples = 4113\nvalue = [2712, 1401]'),
 Text(627.75, 427.11428571428576, 'X[1] <= 0.485\ngini = 0.42\nsamples = 37097\nvalue = [11128,
25969]'),
 Text(523.125, 349.4571428571429, 'X[8] <= 0.228\ngini = 0.497\nsamples = 14526\nvalue = [6658, 7868]'),
 Text(470.8125, 271.8, 'X[10] <= 0.084\ngini = 0.465\nsamples = 9118\nvalue = [3351, 5767]'),
 Text(418.5, 194.14285714285717, 'gini = 0.401\nsamples = 4731\nvalue = [1315, 3416]'),
 Text(523.125, 194.14285714285717, 'gini = 0.497\nsamples = 4387\nvalue = [2036, 2351]'),
 Text(575.4375, 271.8, 'gini = 0.475\nsamples = 5408\nvalue = [3307, 2101]'),
 Text(732.375, 349.4571428571429, 'X[0] <= 0.411\ngini = 0.318\nsamples = 22571\nvalue = [4470,
18101]'),
 Text(680.0625, 271.8, 'X[10] <= 0.084\ngini = 0.255\nsamples = 17327\nvalue = [2601, 14726]'),
 Text(627.75, 194.14285714285717, 'X[1] <= 0.9\ngini = 0.187\nsamples = 11880\nvalue = [1240, 10640]'),
 Text(575.4375, 116.48571428571432, 'gini = 0.254\nsamples = 4208\nvalue = [627, 3581]'),
 Text(680.0625, 116.48571428571432, 'X[5] <= -0.27\ngini = 0.147\nsamples = 7672\nvalue = [613, 7059]'),
 Text(627.75, 38.82857142857142, 'gini = 0.164\nsamples = 3962\nvalue = [357, 3605]'),
 Text(732.375, 38.82857142857142, 'gini = 0.128\nsamples = 3710\nvalue = [256, 3454]'),
 Text(732.375, 194.14285714285717, 'gini = 0.375\nsamples = 5447\nvalue = [1361, 4086]'),
 Text(784.6875, 271.8, 'gini = 0.459\nsamples = 5244\nvalue = [1869, 3375]')]
```



Evaluating Tuned Model on Test Data.

In [417]:

```
hash = modified_model.predict(lucas)
```

In [418]:

```
print(accuracy_score(hash,resampled_y))
```

```
0.7828175026680897
```

In [419]:

```
print(classification_report(hash,resampled_y))
```

```
              precision    recall  f1-score   support

           0       0.72      0.82      0.77      4080
           1       0.85      0.75      0.80      5290

    accuracy                           0.78      9370
   macro avg       0.78      0.79      0.78      9370
weighted avg       0.79      0.78      0.78      9370
```

In [420]:

```
print(confusion_matrix(hash,resampled_y))
```

```
[[3365  715]
 [1320 3970]]
```

In [421]:

```
print(precision_score(hash, resampled_y))
```

```
0.847385272145144
```

In [422]:

```
print(recall_score(hash, resampled_y))
```

```
0.7504725897920604
```

In [423]:

```
print(f1_score(hash, resampled_y))
```

```
0.7959899749373432
```

In [ ]:

Verifying With Respect to ccp_alpha value.

In [424]:

```
pathh = dt.cost_complexity_pruning_path(x_train_ss,y_train)
ccp_alphass, impurities = path.ccp_alphas, path.impurities
```

In [425]:

```
clfss = []
for i in ccp_alphass:
  dt = DecisionTreeClassifier(max_depth = 18, min_samples_split=4250, min_samples_leaf=3700, max_leaf_noc
  dt.fit(x_train_ss,y_train)
  clfss.append(dt)
print('Number of Nodes in the Last Tree is: {} with ccp_alpha: {}'.format(clfs[-1].tree_.node_count, ccp_
```

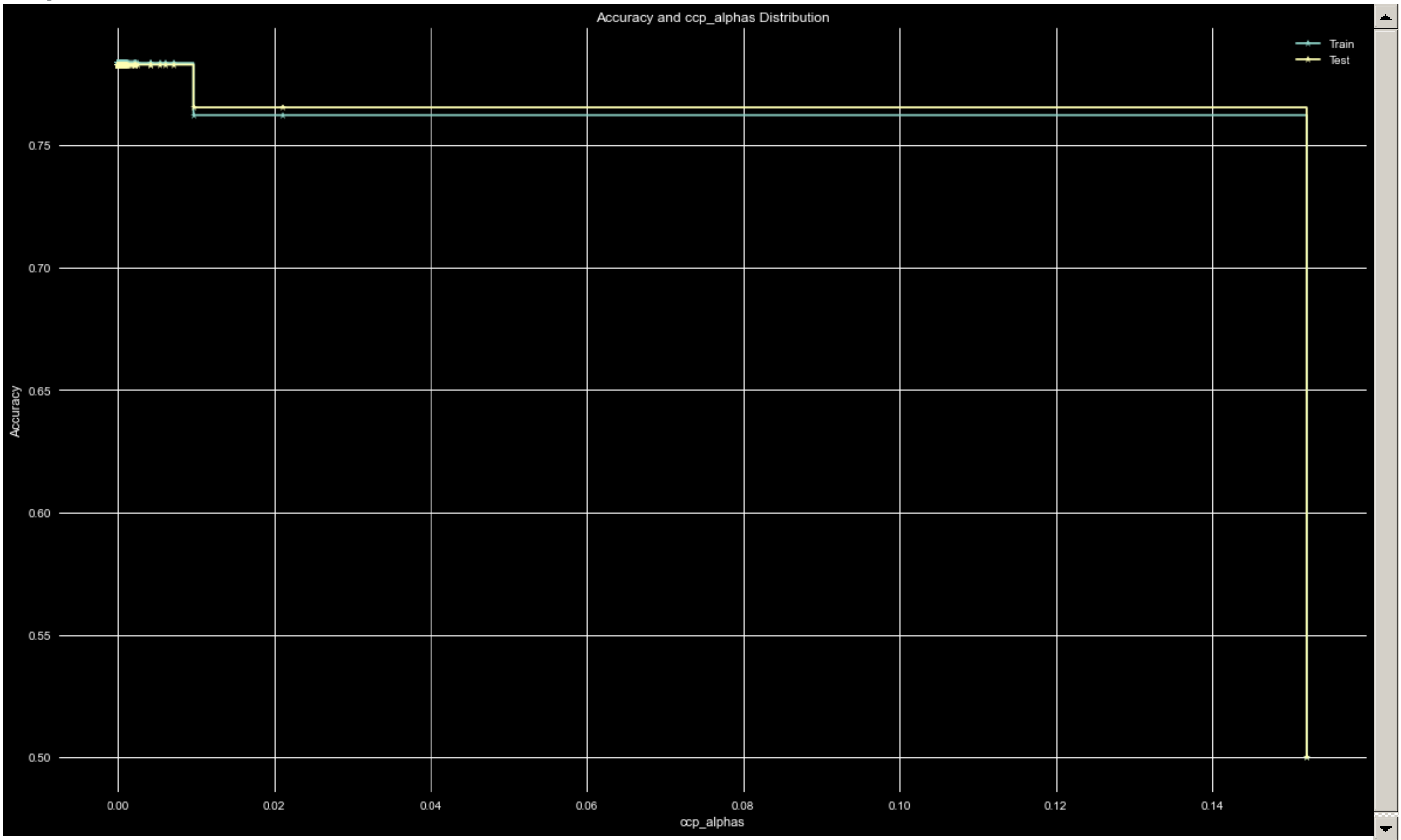
```
Number of Nodes in the Last Tree is: 1 with ccp_alpha: 0.15213838336474234
```

In [426]:

```
train_sett = [dt.score(x_train_ss,y_train) for dt in clfss]
test_sett = [dt.score(lucas,resampled_y) for dt in clfss]

plt.figure(figsize = (12,10), dpi = 100)
fig,ax = plt.subplots()
ax.plot(ccp_alphass, train_sett, marker = '*', label = 'Train', drawstyle = 'steps-post')
ax.plot(ccp_alphass, test_sett, marker = '*', label = 'Test', drawstyle = 'steps-post')
ax.set_xlabel('ccp_alphas')
ax.set_ylabel('Accuracy')
ax.set_title("Accuracy and ccp_alphas Distribution")
ax.legend()
plt.show()
```

<Figure size 1200x1000 with 0 Axes>



In [427]:

```
mod_model_ccp = DecisionTreeClassifier(random_state = 0, ccp_alpha = 0.04)
mod_model_ccp.fit(x_train_ss,y_train)
```

Out[427]:

```
DecisionTreeClassifier(ccp_alpha=0.04, random_state=0)
```

In [504]:

```
print(mod_model_ccp.score(x_train_ss,y_train))
print(mod_model_ccp.score(x_test_ss, y_test))
```

```
0.762187158035935
0.76398155011943
```

In [428]:

```
predicate = mod_model_ccp.predict(lucas)
```

In [429]:

```
print(accuracy_score(predicate,resampled_y))
```

```
0.7653148345784418
```

In [430]:

```
print(precision_score(predicate, resampled_y))
```

```
0.9415154749199574
```

In [431]:

```
print(recall_score(predicate, resampled_y))
```

```
0.6961805555555556
```

In [432]:

```
print(f1_score(predicate, resampled_y))
```

```
0.8004718265130207
```

In [ ]:

In [ ]:

In [ ]:

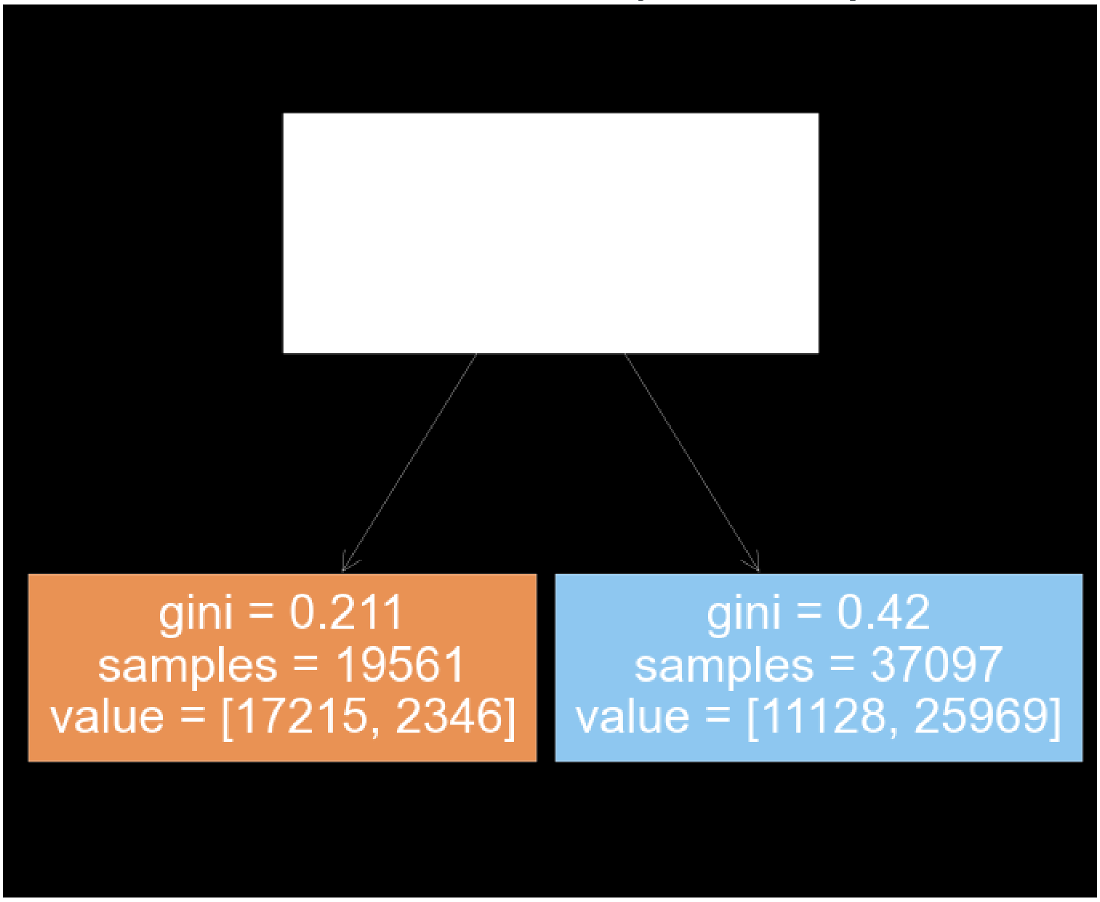
Tree plot with respect to ccp_modified_model

In [433]:

```
plt.figure(figsize = (12,10))
tree.plot_tree(mod_model_ccp, filled=True)
```

Out[433]:

```
[Text(334.8, 407.70000000000005, 'X[1] <= -0.253\ngini = 0.5\nsamples = 56658\nvalue = [28343, 28315]'),
 Text(167.4, 135.89999999999998, 'gini = 0.211\nsamples = 19561\nvalue = [17215, 2346]'),
 Text(502.20000000000005, 135.89999999999998, 'gini = 0.42\nsamples = 37097\nvalue = [11128, 25969]')]
```



Logistic Regression

In [434]:

```
from sklearn.linear_model import LogisticRegression
lg = LogisticRegression()
lg.fit(x_train_ss, y_train)
lg_pred = lg.predict(x_test_ss)
predicted_values = lg.predict_proba(x_test_ss)
```

In [436]:

```
print('Training Accuracy:', lg.score(x_train_ss,y_train))
print('Test Accuracy:', lg.score(x_test_ss,y_test))
```

```
Training Accuracy: 0.8186310847541388
Test Accuracy: 0.8206490404414792
```

In [437]:

```
recall_score(y_test, lg_pred)
```

Out[437]:

```
0.8296174413821472
```

In [438]:

```
precision_score(y_test,lg_pred)
```

Out[438]:

```
0.8153298835705045
```

In [439]:

```
f1_score(y_test,lg_pred)
```

Out[439]:

```
0.8224116135872447
```

In [440]:

```
y_testt = y_test.squeeze()
```

In [441]:

```
precision_points, recall_points, threshold_points = precision_recall_curve(y_testt, predicted_values[:,1]
```

In [442]:

```
precision_points.shape, recall_points.shape, threshold_points.shape
```

Out[442]:

```
((22766,), (22766,), (22765,))
```

In [443]:

```
precision_points
```

Out[443]:

```
array([0.53393367, 0.5339132 , 0.53393665, ..., 1.        , 1.        ,
       1.        ])
```

In [444]:

```
recall_points
```

Out[444]:

```
array([1.00000000e+00, 9.99917729e-01, 9.99917729e-01, ...,
       1.64541341e-04, 8.22706705e-05, 0.00000000e+00])
```

In [445]:

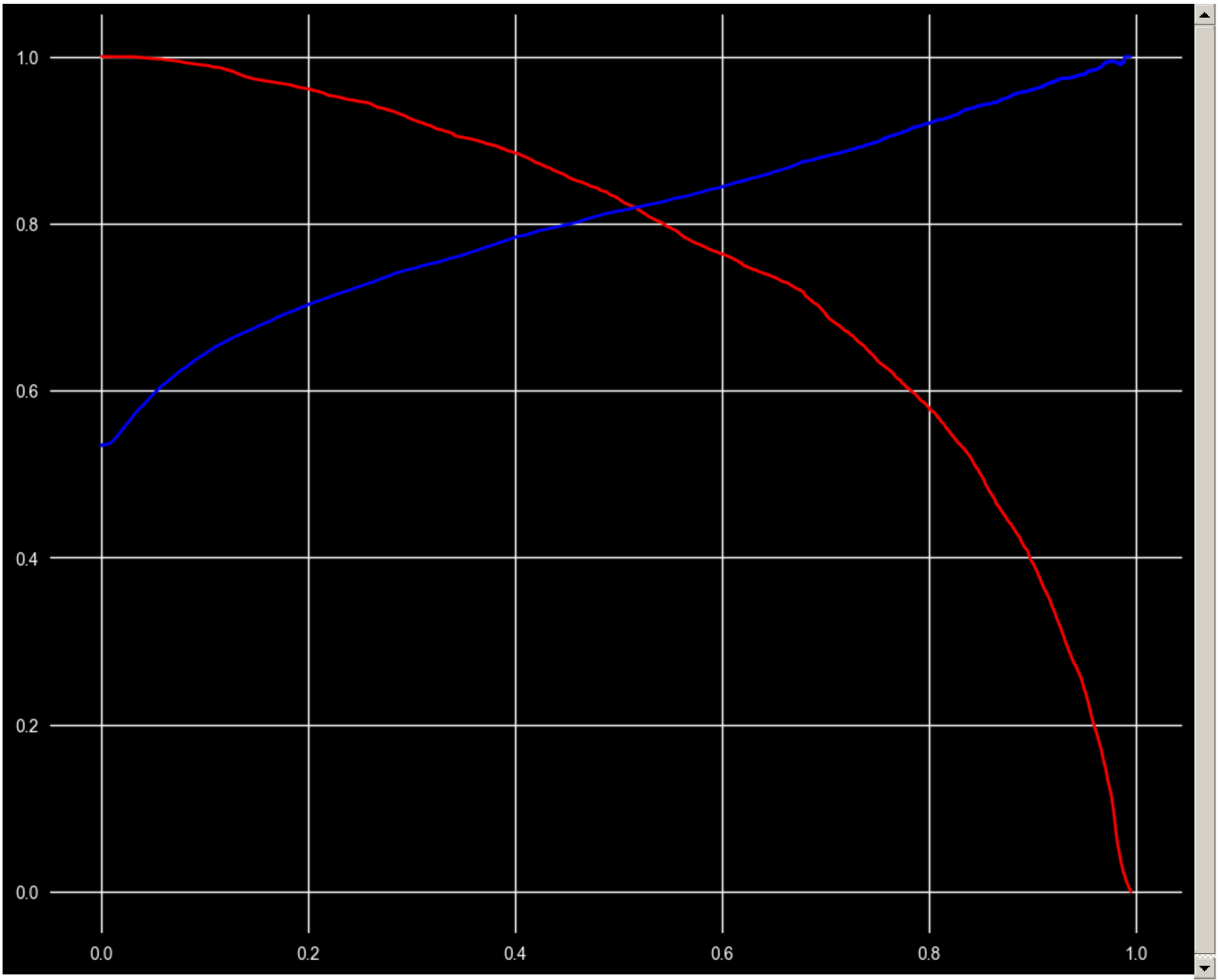
```
threshold_points
```

Out[445]:

```
array([2.90483138e-04, 2.91188966e-04, 2.92204193e-04, ...,
       9.94797111e-01, 9.94891745e-01, 9.94912946e-01])
```

In [446]:

```
plt.figure(figsize = (12,10), dpi = 100)
plt.plot(threshold_points, recall_points[:-1], color = 'red')
plt.plot(threshold_points, precision_points[:-1], color = 'blue')
plt.show()
```



Feature Importance

In [447]:

```
lg.coef_
```

Out[447]:

```
array([[-0.32559288,  2.02658129, -0.332789  , -0.19699277, -0.3458451 ,
         0.23818498,  0.01763683, -0.16767866, -0.43481141, -0.35506599,
        -0.83236205, -0.74291851, -1.56588543]])
```

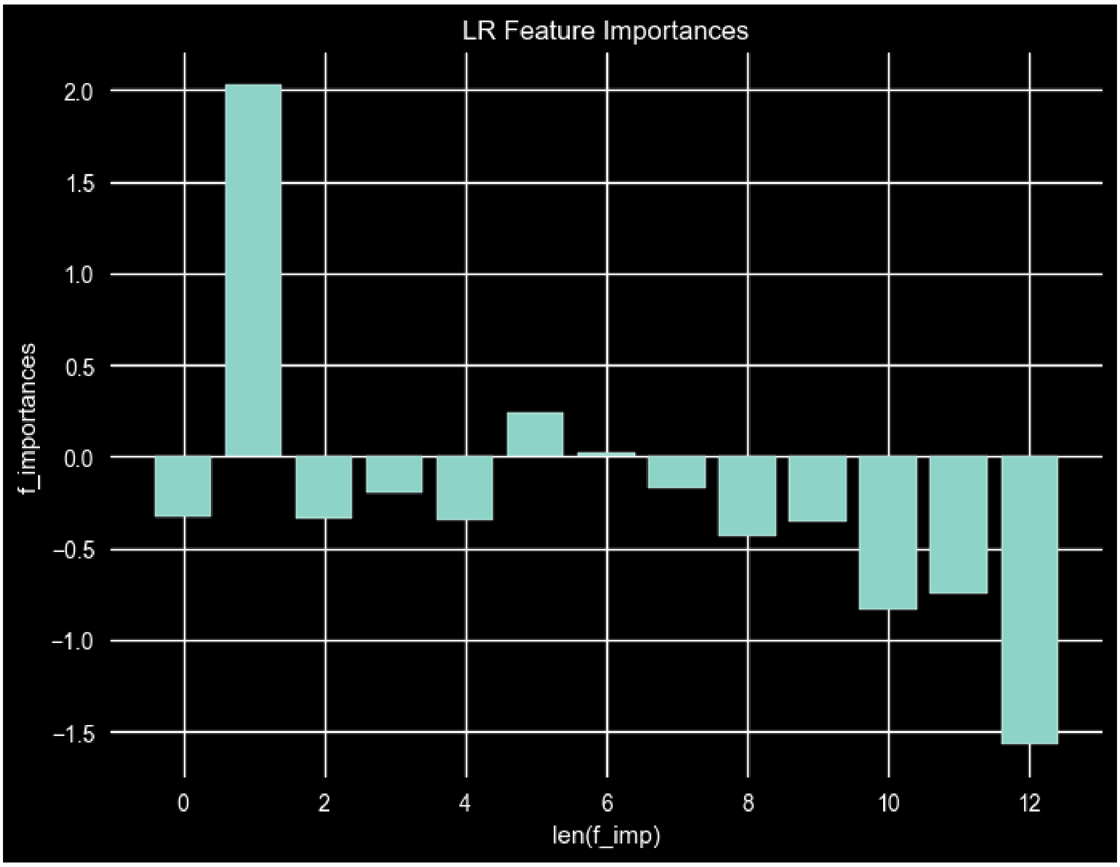
In [448]:

```
f_imp = lg.coef_[0]
print(f_imp)
for i,v in enumerate(f_imp):
    print('Feature: %0d, Score: %.5f' % (i,v))
```

```
[-0.32559288  2.02658129 -0.332789   -0.19699277 -0.3458451   0.23818498
  0.01763683 -0.16767866 -0.43481141 -0.35506599 -0.83236205 -0.74291851
 -1.56588543]
Feature: 0, Score: -0.32559
Feature: 1, Score: 2.02658
Feature: 2, Score: -0.33279
Feature: 3, Score: -0.19699
Feature: 4, Score: -0.34585
Feature: 5, Score: 0.23818
Feature: 6, Score: 0.01764
Feature: 7, Score: -0.16768
Feature: 8, Score: -0.43481
Feature: 9, Score: -0.35507
Feature: 10, Score: -0.83236
Feature: 11, Score: -0.74292
Feature: 12, Score: -1.56589
```

In [449]:

```
plt.figure(figsize =(8,6), dpi = 100)
plt.bar([i for i in range(len(f_imp))], f_imp)
plt.xlabel('len(f_imp)')
plt.ylabel('f_importances')
plt.title('LR Feature Importances')
plt.show()
```



Evaluating LogisticRegresssion using roc_auc_score metric.

In [450]:

```
tpr,fpr, threshold = roc_curve(y_testt, predicted_values[:,1])
tpr.shape, fpr.shape, threshold.shape
```

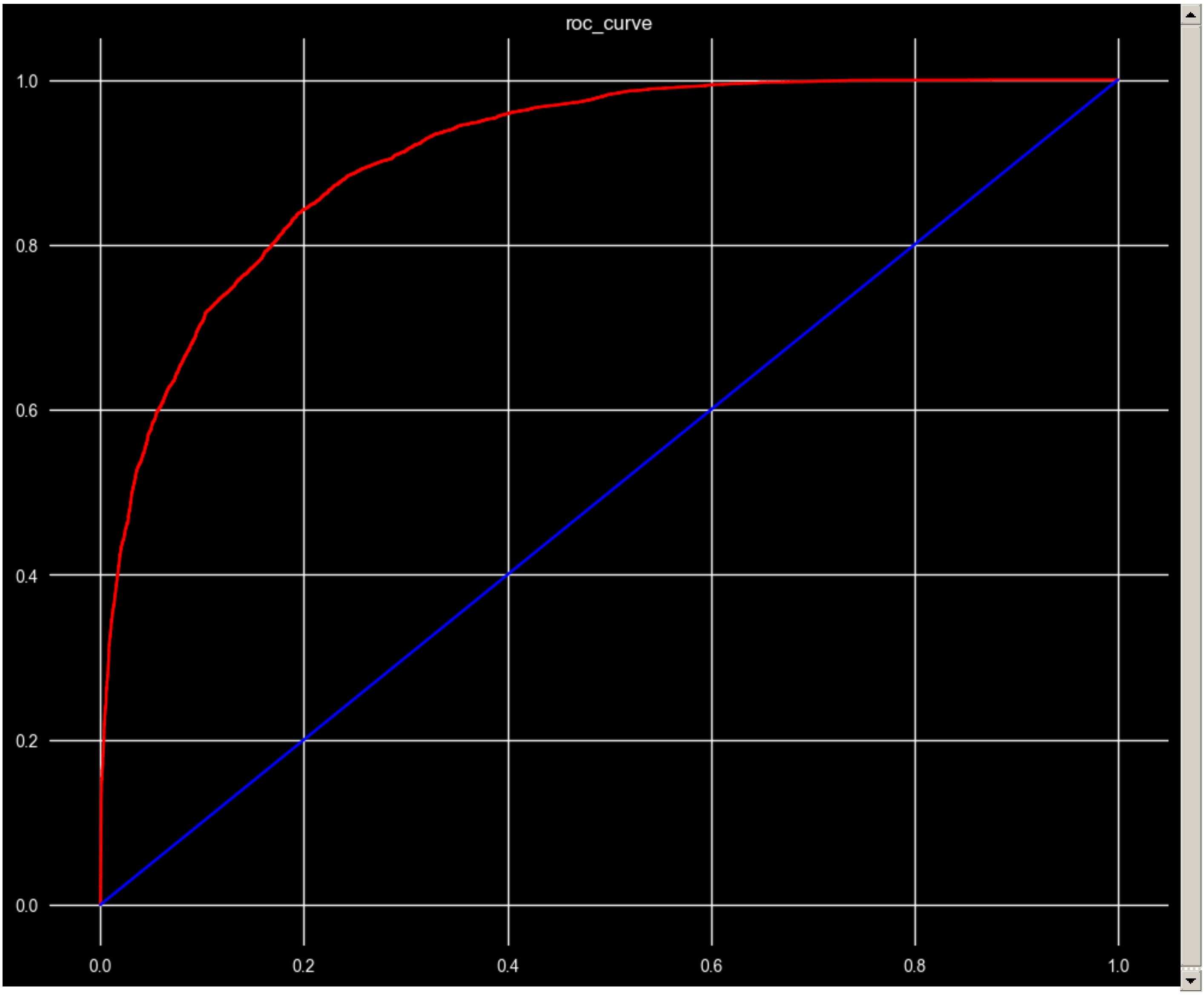
Out[450]:

```
((5860,), (5860,), (5860,))
```

In [451]:

```
plt.figure(figsize = (12,10), dpi = 100)
plt.plot(tpr,fpr, color = 'red')
plt.plot([0,1],[0,1], color = 'blue')
plt.title("roc_curve")
plt.show()

print(roc_auc_score(y_test, predicted_values[:,1]))
```




```
0.909085990624997
```

In [452]:

```
print("Training Accuracy ", lg.score(x_train_ss,y_train))
print("Testing Accuracy ", lg.score(x_test_ss,y_test))
```

```
Training Accuracy  0.8186310847541388
Testing Accuracy  0.8206490404414792
```

In [453]:

```
print(classification_report(lg_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       0.81      0.83      0.82     11914
           1       0.83      0.82      0.82     12368

    accuracy                           0.82     24282
   macro avg       0.82      0.82      0.82     24282
weighted avg       0.82      0.82      0.82     24282
```

In [454]:

```
print(confusion_matrix(lg_pred, y_test))
```

```
[[ 9843  2071]
 [ 2284 10084]]
```

In [455]:

```
print(accuracy_score(lg_pred,y_test))
```

```
0.8206490404414792
```

Performing Cross Validating LR Model.

In [456]:

```
x = pd.DataFrame(data = x_train_ss, columns = inde_vars.columns)
y = y_train
from sklearn.model_selection import  StratifiedKFold
accuracy1 = []
skf = StratifiedKFold(n_splits = 10, random_state = None)
skf.get_n_splits(x,y)
for train_index, test_index in skf.split(x,y):
  print('Train:', train_index, 'Validation',test_index)
  x1_train,x1_test = x.iloc[train_index],x.iloc[test_index]
  y1_train,y1_test = y.iloc[train_index],y.iloc[test_index]
  lg.fit(x1_train,y1_train)
  pred = lg.predict(x1_test)
  score = accuracy_score(pred,y1_test)
  accuracy1.append(score)
print(accuracy1)
```

```
Train: [ 5595  5596  5597 ... 56655 56656 56657] Validation [    0     1     2 ... 5749 5750 5752]
Train: [    0     1     2 ... 56655 56656 56657] Validation [ 5595  5596  5597 ... 11348 11351 11352]
Train: [    0     1     2 ... 56655 56656 56657] Validation [11305 11306 11308 ... 17019 17022 17024]
Train: [    0     1     2 ... 56655 56656 56657] Validation [16980 16983 16984 ... 22681 22685 22689]
Train: [    0     1     2 ... 56655 56656 56657] Validation [22637 22642 22646 ... 28498 28499 28500]
Train: [    0     1     2 ... 56655 56656 56657] Validation [28164 28165 28166 ... 34208 34210 34211]
Train: [    0     1     2 ... 56655 56656 56657] Validation [33785 33786 33788 ... 39838 39839 39840]
Train: [    0     1     2 ... 56655 56656 56657] Validation [39486 39487 39490 ... 45432 45434 45435]
Train: [    0     1     2 ... 56655 56656 56657] Validation [45222 45224 45226 ... 51006 51007 51008]
Train: [    0     1     2 ... 51006 51007 51008] Validation [50979 50980 50981 ... 56655 56656 56657]
[0.816448993999294, 0.8166254853512178, 0.8086833745146488, 0.8206847864454642, 0.8212142605012355,
0.8180374161666079, 0.8242146134839393, 0.8168019767031416, 0.8144748455428067, 0.8275375110326567]
```

Hyper Parameter Tuning Logistic regression model, using RandomizedSearchCV tool.

In [457]:

```
lo = LogisticRegression()
```

In [458]:

```
parameters = {'penalty':['l1','l2','elasticnet','none'],
              'solver':['newton-cg','lbfgs','sag','saga'],
              'max_iter':[i for i in range(100,2000,100)],
              'warm_start':['True','False']}
```

In [459]:

```
print(parameters)
```

```
{'penalty': ['l1', 'l2', 'elasticnet', 'none'], 'solver': ['newton-cg', 'lbfgs', 'sag', 'saga'],
'max_iter': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700,
1800, 1900], 'warm_start': ['True', 'False']}
```

In [460]:

```
lg_tuned_model = RandomizedSearchCV(estimator=lo, param_distributions = parameters, scoring='accuracy', n
```

In [461]:

```
lg_tuned_model.fit(x_train_ss,y_train)
```

```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
```

Out[461]:

```
RandomizedSearchCV(cv=10, estimator=LogisticRegression(), n_jobs=-1,
                   param_distributions={'max_iter': [100, 200, 300, 400, 500,
                                                     600, 700, 800, 900, 1000,
                                                     1100, 1200, 1300, 1400,
                                                     1500, 1600, 1700, 1800,
                                                     1900],
                                        'penalty': ['l1', 'l2', 'elasticnet',
                                                    'none'],
                                        'solver': ['newton-cg', 'lbfgs', 'sag',
                                                   'saga'],
                                        'warm_start': ['True', 'False']},
                   random_state=50, scoring='accuracy', verbose=2)
```

In [462]:

```
lg_tuned_model.best_params_
```

Out[462]:

```
{'warm_start': 'False', 'solver': 'saga', 'penalty': 'none', 'max_iter': 400}
```

In [463]:

```
lg_tuned_model.get_params
```

Out[463]:

```
<bound method BaseEstimator.get_params of RandomizedSearchCV(cv=10, estimator=LogisticRegression(),
n_jobs=-1,
                   param_distributions={'max_iter': [100, 200, 300, 400, 500,
                                                     600, 700, 800, 900, 1000,
                                                     1100, 1200, 1300, 1400,
                                                     1500, 1600, 1700, 1800,
                                                     1900],
                                        'penalty': ['l1', 'l2', 'elasticnet',
                                                    'none'],
                                        'solver': ['newton-cg', 'lbfgs', 'sag',
                                                   'saga'],
                                        'warm_start': ['True', 'False']},
                   random_state=50, scoring='accuracy', verbose=2)>
```

In [464]:

```
lg_tuned_model.best_score_
```

Out[464]:

```
0.8184899755092937
```

Testing the Accuracy using Tuned LR Model.

In [465]:

```
lr = LogisticRegression(max_iter = 1300,
                        penalty='l2',
                        solver= 'newton-cg',
                        warm_start=True)
lr.fit(x_train_ss,y_train)
```

Out[465]:

```
LogisticRegression(max_iter=1300, solver='newton-cg', warm_start=True)
```

In [466]:

```
tuned_pred = lr.predict(x_test_ss)
print(accuracy_score(tuned_pred,y_test))
```

```
0.8206490404414792
```

Evaluating Tuned Model on Test Data.

In [467]:

```
print('Tuned Training Accuracy:', lr.score(x_train_ss, y_train))
```

```
Tuned Training Accuracy: 0.8186310847541388
```

In [468]:

```
jim = lr.predict(lucas)
print(accuracy_score(jim, resampled_y))
```

```
0.8260405549626467
```

In [469]:

```
print(roc_auc_score(jim, resampled_y))
```

```
0.8262995820543876
```

In [470]:

```
print(precision_score(jim, resampled_y))
```

```
0.8401280683030949
```

In [471]:

```
print(recall_score(jim, resampled_y))
```

```
0.8171060826240398
```

In [472]:

```
print(f1_score(jim, resampled_y))
```

```
0.828457166912229
```

Results of LR Model without Tuning.

In [512]:

```
pam = lg.predict(lucas)
print(accuracy_score(pam, resampled_y))
```

```
0.8261472785485592
```

In [513]:

```
print(precision_score(pam, resampled_y))
```

```
0.8390608324439701
```

In [514]:

```
print(recall_score(pam, resampled_y))
```

```
0.8179359134415314
```

In [515]:

```
print(f1_score(pam, resampled_y))
```

```
0.8283637129912549
```

Random Forest Classifier

In [473]:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(x_train_ss, y_train)
rf_pred = rf.predict(x_test_ss)
```

In [474]:

```
print("Training accuracy is", rf.score(x_train_ss, y_train))
print('Testing Accuracy is', rf.score(x_test_ss, y_test))
print(accuracy_score(y_test, rf_pred))
```

```
Training accuracy is 1.0
Testing Accuracy is 0.9719133514537518
0.9719133514537518
```

In [475]:

```
print(confusion_matrix(y_test, rf_pred))
```

```
[[11693   434]
 [  248 11907]]
```

In [476]:

```
print(classification_report(y_test, rf_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     12127
           1       0.96      0.98      0.97     12155

    accuracy                           0.97     24282
   macro avg       0.97      0.97      0.97     24282
weighted avg       0.97      0.97      0.97     24282
```

In [477]:

```
recall_score(y_test, rf_pred)
```

Out[477]:

```
0.9795968737145208
```

In [478]:

```
precision_score(y_test, rf_pred)
```

Out[478]:

```
0.9648326715825298
```

In [479]:

```
f1_score(y_test, rf_pred)
```

Out[479]:

```
0.9721587197909863
```

Feature Importance

In [480]:

```
rf.feature_importances_
```

Out[480]:

```
array([0.02731006, 0.40079617, 0.01583128, 0.00791932, 0.01904344,
       0.18657347, 0.14964531, 0.039123  , 0.03138638, 0.00074875,
       0.05651673, 0.03342756, 0.03167853])
```

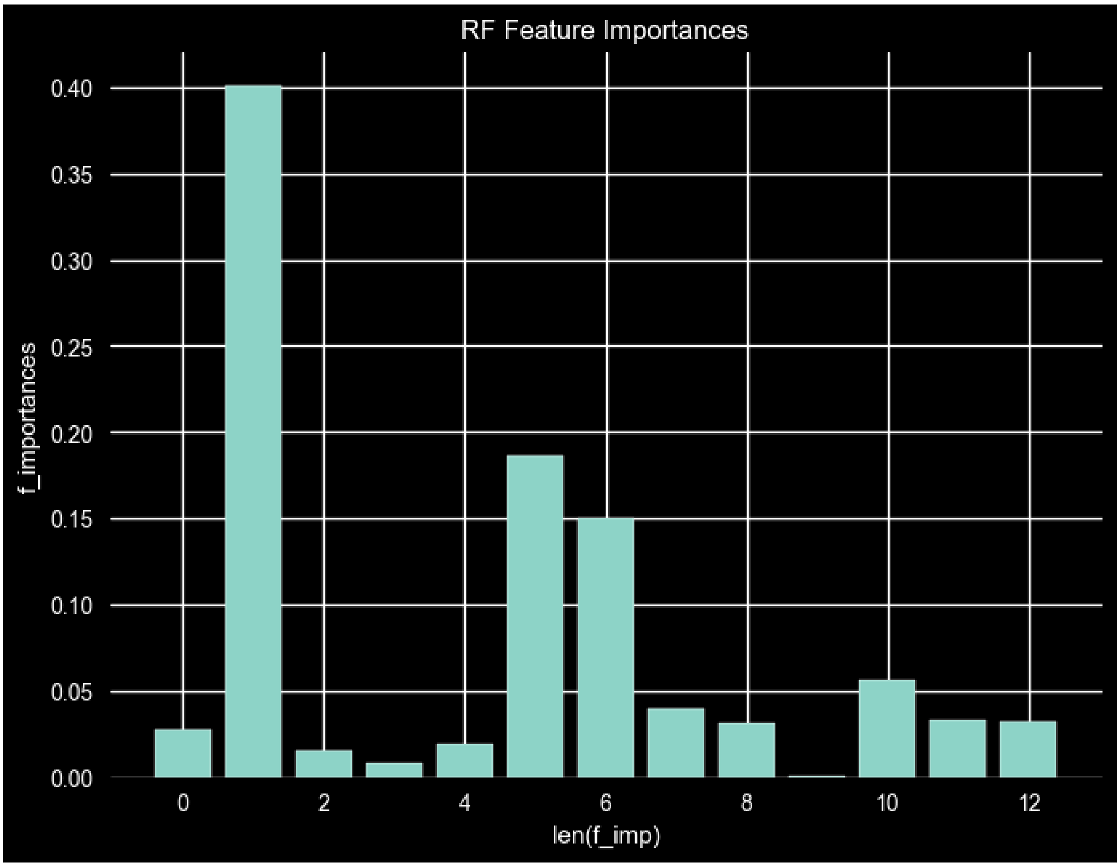
In [481]:

```
fe_imp = rf.feature_importances_
for i,v in enumerate(fe_imp):
  print('Feature: %0d, Score: %.5f' % (i,v))
```

```
Feature: 0, Score: 0.02731
Feature: 1, Score: 0.40080
Feature: 2, Score: 0.01583
Feature: 3, Score: 0.00792
Feature: 4, Score: 0.01904
Feature: 5, Score: 0.18657
Feature: 6, Score: 0.14965
Feature: 7, Score: 0.03912
Feature: 8, Score: 0.03139
Feature: 9, Score: 0.00075
Feature: 10, Score: 0.05652
Feature: 11, Score: 0.03343
Feature: 12, Score: 0.03168
```

In [482]:

```
plt.figure(figsize =(8,6), dpi = 100)
plt.bar([i for i in range(len(fe_imp))], fe_imp)
plt.xlabel('len(f_imp)')
plt.ylabel('f_importances')
plt.title('RF Feature Importances')
plt.show()
```



Performing Cross Validation on RFC Model.

In [483]:

```
x = pd.DataFrame(data = x_train_ss, columns = inde_vars.columns)
y = y_train
from sklearn.model_selection import  StratifiedKFold
accuracy2 = []
skf = StratifiedKFold(n_splits = 10, random_state = None)
skf.get_n_splits(x,y)
for train_index, test_index in skf.split(x,y):
  print('Train:', train_index, 'Validation',test_index)
  x1_train,x1_test = x.iloc[train_index],x.iloc[test_index]
  y1_train,y1_test = y.iloc[train_index],y.iloc[test_index]
  rf.fit(x1_train,y1_train)
  pred = rf.predict(x1_test)
  score = accuracy_score(pred,y1_test)
  accuracy2.append(score)
print(accuracy2)
```

```
Train: [ 5595  5596  5597 ... 56655 56656 56657] Validation [    0     1     2 ... 5749 5750 5752]
Train: [    0     1     2 ... 56655 56656 56657] Validation [ 5595  5596  5597 ... 11348 11351 11352]
Train: [    0     1     2 ... 56655 56656 56657] Validation [11305 11306 11308 ... 17019 17022 17024]
Train: [    0     1     2 ... 56655 56656 56657] Validation [16980 16983 16984 ... 22681 22685 22689]
Train: [    0     1     2 ... 56655 56656 56657] Validation [22637 22642 22646 ... 28498 28499 28500]
Train: [    0     1     2 ... 56655 56656 56657] Validation [28164 28165 28166 ... 34208 34210 34211]
Train: [    0     1     2 ... 56655 56656 56657] Validation [33785 33786 33788 ... 39838 39839 39840]
Train: [    0     1     2 ... 56655 56656 56657] Validation [39486 39487 39490 ... 45432 45434 45435]
Train: [    0     1     2 ... 56655 56656 56657] Validation [45222 45224 45226 ... 51006 51007 51008]
Train: [    0     1     2 ... 51006 51007 51008] Validation [50979 50980 50981 ... 56655 56656 56657]
[0.9699964701729615, 0.9685845393575715, 0.9668196258383339, 0.967525591246029, 0.9714084009883516,
0.9707024355806565, 0.969643487469114, 0.9669961171902577, 0.970873786407767, 0.971756398940865]
```

Hyper Parameter Tuning Random Forest Model Using RandomizedSearchCV.

In [484]:

```
rfc = RandomForestClassifier()
#rfc
```

In [485]:

```
param = {'n_estimators' : [i for i in range(100,1500,100)],
         'max_depth' : [i for i in range(10,100,10)],
         'max_features' : ['auto','sqrt','log2'],
```

```
          'min_samples_split' :  np.linspace(0.1,1.0,10, endpoint = True),
          'min_samples_leaf' : np.linspace(0.1,0.5,5, endpoint =True),
          'warm_start' : ['True', 'False']
}

#param
```

In [486]:

```
rf_tuned_model = RandomizedSearchCV(estimator =rfc, param_distributions=param, scoring = 'roc_auc', verbo
```

In [487]:

```
rf_tuned_model.fit(x_train_ss,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

Out[487]:

```
RandomizedSearchCV(estimator=RandomForestClassifier(), n_jobs=-1,
                   param_distributions={'max_depth': [10, 20, 30, 40, 50, 60,
                                                      70, 80, 90],
                                        'max_features': ['auto', 'sqrt',
                                                         'log2'],
                                        'min_samples_leaf': array([0.1, 0.2, 0.3, 0.4, 0.5]),
                                        'min_samples_split': array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.
, 0.9, 1. ]),
                                        'n_estimators': [100, 200, 300, 400,
                                                         500, 600, 700, 800,
                                                         900, 1000, 1100, 1200,
                                                         1300, 1400],
                                        'warm_start': ['True', 'False']},
                   random_state=50, scoring='roc_auc', verbose=2)
```

In [488]:

```
rf_tuned_model.best_score_
```

Out[488]:

```
0.868225005051485
```

In [489]:

```
rf_tuned_model.get_params
```

Out[489]:

```
<bound method BaseEstimator.get_params of RandomizedSearchCV(estimator=RandomForestClassifier(),
n_jobs=-1,
                   param_distributions={'max_depth': [10, 20, 30, 40, 50, 60,
                                                      70, 80, 90],
                                        'max_features': ['auto', 'sqrt',
                                                         'log2'],
                                        'min_samples_leaf': array([0.1, 0.2, 0.3, 0.4, 0.5]),
                                        'min_samples_split': array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.
, 0.9, 1. ]),
                                        'n_estimators': [100, 200, 300, 400,
                                                         500, 600, 700, 800,
                                                         900, 1000, 1100, 1200,
                                                         1300, 1400],
                                        'warm_start': ['True', 'False']},
                   random_state=50, scoring='roc_auc', verbose=2)>
```

In [490]:

```
rf_tuned_model.best_estimator_
```

Out[490]:

```
RandomForestClassifier(max_depth=40, max_features='sqrt', min_samples_leaf=0.1,
                       min_samples_split=0.1, warm_start='True')
```

Before Tuning the Random Forest Model.

In [491]:

```
dwight = rf.predict(lucas)
print(accuracy_score(dwight, resampled_y))
```

```
0.8453575240128068
```

In [516]:

```
print(precision_score(dwight, resampled_y))
```

```
0.735965848452508
```

In [517]:

```
print(recall_score(resampled_y, dwight))
```

```
0.735965848452508
```

In [518]:

```
print(f1_score(resampled_y, dwight))
```

```
0.8263630916716597
```

Evaluating Tuned Model on Test Data.

In [492]:

```
kite = RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=40, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=0.1, min_samples_split=0.1,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start='True')
```

In [493]:

```
kite.fit(x_train_ss,y_train)
```

Out[493]:

```
RandomForestClassifier(max_depth=40, max_features='sqrt', min_samples_leaf=0.1,
                       min_samples_split=0.1, warm_start='True')
```

In [499]:

```
print('Tuned Training Score:', kite.score(x_train_ss, y_train))
```

```
Tuned Training Score: 0.795721698612729
```

In [500]:

```
lion = kite.predict(lucas)
print(accuracy_score(lion,resampled_y))
```

```
0.7945570971184632
```

In [501]:

```
print(recall_score(resampled_y, lion))
```

```
0.8439701173959445
```

In [502]:

```
print(precision_score(resampled_y, lion))
```

```
0.7680652680652681
```

In [503]:

```
print(f1_score(resampled_y, lion))
```

```
0.8042306518865047
```

In [ ]: