```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

import kagglehub

# Download latest version
path = kagglehub.dataset_download("janiobachmann/bank-marketing-
dataset")

print("Path to dataset files:", path)
```

```
Using Colab cache for faster access to the 'bank-marketing-dataset'
dataset.
Path to dataset files: /kaggle/input/bank-marketing-dataset
```

```
df = pd.read_csv('/kaggle/input/bank-marketing-dataset/bank.csv')
print(df)
```

```
       age          job  marital  education default  balance housing
loan  \
0       59       admin.  married  secondary      no     2343     yes
no
1       56       admin.  married  secondary      no       45      no
no
2       41   technician  married  secondary      no     1270     yes
no
3       55     services  married  secondary      no     2476     yes
no
4       54       admin.  married   tertiary      no      184      no
no
...    ...          ...      ...        ...     ...      ...     ...
...
11157   33  blue-collar   single    primary      no        1     yes
no
11158   39     services  married  secondary      no      733      no
no
11159   32   technician   single  secondary      no       29      no
no
11160   43   technician  married  secondary      no        0      no
yes
11161   34   technician  married  secondary      no        0      no
no

        contact  day month  duration  campaign  pdays  previous
poutcome  \
0       unknown    5   may      1042         1     -1         0
unknown
1       unknown    5   may      1467         1     -1         0
unknown
```

```
2       unknown    5   may      1389         1     -1         0
unknown
3       unknown    5   may       579         1     -1         0
unknown
4       unknown    5   may       673         2     -1         0
unknown
...         ...  ...   ...       ...       ...    ...       ...       .
..
11157  cellular   20   apr       257         1     -1         0
unknown
11158   unknown   16   jun        83         4     -1         0
unknown
11159  cellular   19   aug       156         2     -1         0
unknown
11160  cellular    8   may         9         2    172         5
failure
11161  cellular    9   jul       628         1     -1         0
unknown

      deposit
0         yes
1         yes
2         yes
3         yes
4         yes
...       ...
11157      no
11158      no
11159      no
11160      no
11161      no

[11162 rows x 17 columns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11162 entries, 0 to 11161
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        11162 non-null  int64
 1   job        11162 non-null  object
 2   marital    11162 non-null  object
 3   education  11162 non-null  object
 4   default    11162 non-null  object
 5   balance    11162 non-null  int64
 6   housing    11162 non-null  object
 7   loan       11162 non-null  object
 8   contact    11162 non-null  object
 9   day        11162 non-null  int64
 10  month      11162 non-null  object
 11  duration   11162 non-null  int64
 12  campaign   11162 non-null  int64
 13  pdays      11162 non-null  int64
 14  previous   11162 non-null  int64
 15  poutcome   11162 non-null  object
 16  deposit    11162 non-null  object
dtypes: int64(7), object(10)
memory usage: 1.4+ MB
```

```
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11162,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 11,\n        \"min\": 18,\n
\"max\": 95,\n        \"num_unique_values\": 76,\n        \"samples\":
[\n          54,\n          47,\n          30\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 12,\n
\"samples\": [\n          \"self-employed\",\n          \"unknown\",\n
\"admin.\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"married\",\n          \"single\",\n
\"divorced\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"education\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"tertiary\",\n          \"unknown\",\n
\"secondary\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"balance\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3225,\n
\"min\": -6847,\n        \"max\": 81204,\n
\"num_unique_values\": 3805,\n        \"samples\": [\n          3026,\
n          1792\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"housing\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"no\",\n          \"yes\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"loan\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"contact\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"unknown\",\n
\"cellular\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"day\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 8,\n        \"min\": 1,\n        \"max\": 31,\n
\"num_unique_values\": 31,\n        \"samples\": [\n          10,\n
27\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"month\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 12,\n        \"samples\": [\n
\"apr\",\n          \"mar\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"duration\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 347,\n        \"min\": 2,\n
\"max\": 3881,\n        \"num_unique_values\": 1428,\n
\"samples\": [\n          597,\n          346\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"campaign\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 2,\n
\"min\": 1,\n        \"max\": 63,\n        \"num_unique_values\": 36,\
n        \"samples\": [\n          31,\n          7\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"pdays\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 108,\n
\"min\": -1,\n        \"max\": 854,\n        \"num_unique_values\":
472,\n        \"samples\": [\n          294,\n          148\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"previous\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2,\n        \"min\": 0,\n
\"max\": 58,\n        \"num_unique_values\": 34,\n        \"samples\":
[\n          30,\n          14\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"poutcome\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"other\",\n          \"success\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"deposit\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"no\",\n          \"yes\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```python
df = df.drop(["contact", "day", "month"], axis=1)
df = df[df["job"] != "unknown"]
df = df[df["marital"] != "unknown"]
df = df[df["education"] != "unknown"]

from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
df["deposit"] = LE.fit_transform(df["deposit"])

X_train, X_test, y_train, y_test = train_test_split(
    df.drop("deposit", axis = 1),
    df["deposit"],
    test_size = 0.2
    )

X_train = pd.get_dummies(X_train, drop_first=True)#converts
categorical (text) columns into numerical columns
X_test = pd.get_dummies(X_test, drop_first=True)

Dt = DecisionTreeClassifier()
Dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier()
```

```
from sklearn.tree import plot_tree
plot_tree(Dt, filled = True, max_depth=5)
```

```
[Text(0.5199468085106383, 0.9285714285714286, 'x[2] <= 365.5\ngini =
0.498\nsamples = 8507\nvalue = [4506, 4001]'),
 Text(0.27393617021276595, 0.7857142857142857, 'x[24] <= 0.5\ngini =
0.426\nsamples = 5499\nvalue = [3807, 1692]'),
 Text(0.39694148936170215, 0.8571428571428572, 'True  '),
 Text(0.15691489361702127, 0.6428571428571429, 'x[21] <= 0.5\ngini =
0.368\nsamples = 4951\nvalue = [3746, 1205]'),
 Text(0.0851063829787234, 0.5, 'x[2] <= 124.5\ngini = 0.465\nsamples =
2507\nvalue = [1584, 923]'),
 Text(0.0425531914893617, 0.35714285714285715, 'x[2] <= 77.5\ngini =
0.225\nsamples = 803\nvalue = [699.0, 104.0]'),
 Text(0.02127659574468085, 0.21428571428571427, 'x[0] <= 23.5\ngini =
0.051\nsamples = 342\nvalue = [333, 9]'),
 Text(0.010638297872340425, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.031914893617021274, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.06382978723404255, 0.21428571428571427, 'x[0] <= 28.5\ngini =
0.327\nsamples = 461\nvalue = [366, 95]'),
 Text(0.05319148936170213, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.07446808510638298, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.1276595744680851, 0.35714285714285715, 'x[1] <= 229.5\ngini =
0.499\nsamples = 1704\nvalue = [885, 819]'),
 Text(0.10638297872340426, 0.21428571428571427, 'x[0] <= 26.5\ngini =
0.415\nsamples = 530\nvalue = [374, 156]'),
 Text(0.09574468085106383, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.11702127659574468, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.14893617021276595, 0.21428571428571427, 'x[4] <= 24.5\ngini =
0.492\nsamples = 1174\nvalue = [511.0, 663.0]'),
 Text(0.13829787234042554, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.1595744680851064, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.22872340425531915, 0.5, 'x[2] <= 145.5\ngini = 0.204\nsamples
= 2444\nvalue = [2162, 282]'),
 Text(0.20212765957446807, 0.35714285714285715, 'x[0] <= 68.0\ngini =
0.091\nsamples = 1178\nvalue = [1122, 56]'),
 Text(0.19148936170212766, 0.21428571428571427, 'x[4] <= 378.5\ngini =
0.088\nsamples = 1176\nvalue = [1122, 54]'),
 Text(0.18085106382978725, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.20212765957446807, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.2127659574468085, 0.21428571428571427, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.2553191489361702, 0.35714285714285715, 'x[4] <= 3.0\ngini =
0.293\nsamples = 1266\nvalue = [1040, 226]'),
 Text(0.23404255319148937, 0.21428571428571427, 'x[0] <= 60.5\ngini =
0.229\nsamples = 1006\nvalue = [873, 133]'),
 Text(0.22340425531914893, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.24468085106382978, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.2765957446808511, 0.21428571428571427, 'x[4] <= 123.0\ngini =
0.459\nsamples = 260\nvalue = [167, 93]'),
 Text(0.26595744680851063, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.2872340425531915, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.39095744680851063, 0.6428571428571429, 'x[2] <= 119.5\ngini =
0.198\nsamples = 548\nvalue = [61.0, 487.0]'),
 Text(0.32978723404255317, 0.5, 'x[2] <= 62.5\ngini = 0.498\nsamples =
58\nvalue = [27.0, 31.0]'),
 Text(0.3191489361702128, 0.35714285714285715, 'gini = 0.0\nsamples =
8\nvalue = [8, 0]'),
 Text(0.3404255319148936, 0.35714285714285715, 'x[5] <= 6.5\ngini =
0.471\nsamples = 50\nvalue = [19, 31]'),
 Text(0.3191489361702128, 0.21428571428571427, 'x[1] <= 2316.5\ngini =
0.444\nsamples = 45\nvalue = [15, 30]'),
 Text(0.30851063829787234, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.32978723404255317, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.3617021276595745, 0.21428571428571427, 'x[2] <= 70.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.35106382978723405, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.3723404255319149, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.4521276595744681, 0.5, 'x[2] <= 363.0\ngini = 0.129\nsamples =
490\nvalue = [34, 456]'),
 Text(0.425531914893617, 0.35714285714285715, 'x[2] <= 146.5\ngini =
0.123\nsamples = 487\nvalue = [32, 455]'),
 Text(0.40425531914893614, 0.21428571428571427, 'x[19] <= 0.5\ngini =
0.346\nsamples = 36\nvalue = [8, 28]'),
 Text(0.39361702127659576, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.4148936170212766, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.44680851063829785, 0.21428571428571427, 'x[5] <= 21.0\ngini =
0.101\nsamples = 451\nvalue = [24, 427]'),
 Text(0.43617021276595747, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.4574468085106383, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.4787234042553192, 0.35714285714285715, 'x[11] <= 0.5\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.46808510638297873, 0.21428571428571427, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.48936170212765956, 0.21428571428571427, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7659574468085106, 0.7857142857142857, 'x[2] <= 636.5\ngini =
0.357\nsamples = 3008\nvalue = [699, 2309]'),
 Text(0.6429521276595744, 0.8571428571428572, '  False'),
 Text(0.6329787234042553, 0.6428571428571429, 'x[24] <= 0.5\ngini =
0.439\nsamples = 1518\nvalue = [493, 1025]'),
 Text(0.574468085106383, 0.5, 'x[21] <= 0.5\ngini = 0.461\nsamples =
1350\nvalue = [487.0, 863.0]'),
 Text(0.5319148936170213, 0.35714285714285715, 'x[1] <= 242.5\ngini =
0.391\nsamples = 712\nvalue = [190, 522]'),
 Text(0.5106382978723404, 0.21428571428571427, 'x[0] <= 59.5\ngini =
0.46\nsamples = 220\nvalue = [79, 141]'),
 Text(0.5, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.5212765957446809, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.5531914893617021, 0.21428571428571427, 'x[2] <= 433.5\ngini =
0.349\nsamples = 492\nvalue = [111, 381]'),
 Text(0.5425531914893617, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.5638297872340425, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.6170212765957447, 0.35714285714285715, 'x[2] <= 475.5\ngini =
0.498\nsamples = 638\nvalue = [297, 341]'),
 Text(0.5957446808510638, 0.21428571428571427, 'x[4] <= 374.5\ngini =
0.486\nsamples = 300\nvalue = [175, 125]'),
 Text(0.5851063829787234, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.6063829787234043, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.6382978723404256, 0.21428571428571427, 'x[17] <= 0.5\ngini =
0.461\nsamples = 338\nvalue = [122.0, 216.0]'),
 Text(0.6276595744680851, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.648936170212766, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.6914893617021277, 0.5, 'x[13] <= 0.5\ngini = 0.069\nsamples =
168\nvalue = [6, 162]'),
 Text(0.6702127659574468, 0.35714285714285715, 'x[1] <= -207.0\ngini =
0.037\nsamples = 160\nvalue = [3, 157]'),
 Text(0.6595744680851063, 0.21428571428571427, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.6808510638297872, 0.21428571428571427, 'x[4] <= 63.5\ngini =
0.025\nsamples = 159\nvalue = [2, 157]'),
 Text(0.6702127659574468, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.6914893617021277, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.7127659574468085, 0.35714285714285715, 'x[0] <= 25.5\ngini =
0.469\nsamples = 8\nvalue = [3, 5]'),
 Text(0.7021276595744681, 0.21428571428571427, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.723404255319149, 0.21428571428571427, 'x[4] <= 285.5\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.7127659574468085, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.7340425531914894, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.898936170212766, 0.6428571428571429, 'x[2] <= 889.5\ngini =
0.238\nsamples = 1490\nvalue = [206.0, 1284.0]'),
 Text(0.8297872340425532, 0.5, 'x[16] <= 0.5\ngini = 0.286\nsamples =
762\nvalue = [132, 630]'),
 Text(0.7872340425531915, 0.35714285714285715, 'x[0] <= 35.5\ngini =
0.216\nsamples = 341\nvalue = [42, 299]'),
 Text(0.7659574468085106, 0.21428571428571427, 'x[3] <= 6.5\ngini =
0.152\nsamples = 193\nvalue = [16, 177]'),
 Text(0.7553191489361702, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.776595744680851, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.8085106382978723, 0.21428571428571427, 'x[2] <= 649.5\ngini =
0.29\nsamples = 148\nvalue = [26, 122]'),
 Text(0.7978723404255319, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.8191489361702128, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.8723404255319149, 0.35714285714285715, 'x[1] <= 1949.5\ngini =
0.336\nsamples = 421\nvalue = [90, 331]'),
 Text(0.851063829787234, 0.21428571428571427, 'x[5] <= 5.5\ngini =
0.375\nsamples = 312\nvalue = [78, 234]'),
 Text(0.8404255319148937, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.8617021276595744, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.8936170212765957, 0.21428571428571427, 'x[22] <= 0.5\ngini =
0.196\nsamples = 109\nvalue = [12, 97]'),
 Text(0.8829787234042553, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.9042553191489362, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.9680851063829787, 0.5, 'x[0] <= 86.5\ngini = 0.183\nsamples =
728\nvalue = [74, 654]'),
 Text(0.9574468085106383, 0.35714285714285715, 'x[4] <= 213.0\ngini =
0.181\nsamples = 727\nvalue = [73, 654]'),
 Text(0.9361702127659575, 0.21428571428571427, 'x[2] <= 1524.5\ngini =
0.17\nsamples = 682\nvalue = [64, 618]'),
 Text(0.925531914893617, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.9468085106382979, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.9787234042553191, 0.21428571428571427, 'x[4] <= 220.5\ngini =
0.32\nsamples = 45\nvalue = [9, 36]'),
 Text(0.9680851063829787, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.9893617021276596, 0.07142857142857142, '\n  (...)  \n'),
 Text(0.9787234042553191, 0.35714285714285715, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]')]
```