

Decision Tree (ID3)

```

import numpy as np
import pandas as pd
df = pd.read_csv("ml")
df.head()
df.info()
df.describe()
def find_entropy(df):
    target = df.keys()[-1]
    entropy = 0
    values = df[target].unique()
    for value in values:
        fraction = df[df[target] == value].count() / len(df[target])
        entropy += fraction * np.log2(fraction)
    return entropy

def BuildTree(df, tree=None):
    target = df.keys()[-1]
    mode = find_winner(df)
    ctt = np.unique(df[node])
    if tree is None:
        tree = {}
    if len(ctt) == 1:
        tree[target] = mode
    else:
        sub = get_subtable(df, node, value)
        dvalue, count = np.unique(sub[target])
        if len(count) == 1:
            tree[target] = Value[0]
        else:
            tree[target] = buildtree(sub)
    return tree

```

tree = buildtree(df)

import pprint

pprint.print(tree)

⇒ Decision tree (sklearn) -

import pandas as pd

import numpy as np

import sklearn.model_selection import decisiontree
from sklearn.tree import plot_tree

df = pd.read_csv('iris')

df.head()

df.info()

df.isnull().sum()

cols = df.columns[0:-1]

for i in cols

sns.boxplot(y = df[i])

plt.show()

x = df.drop(['species'], axis=1)

y = df['species']

X-train, X-test, y-train =

Xtrain-test-split(X,y,test_size=0.3)

dt = DecisionTreeClassifier(max_depth=3)

dt = fit(X,y)

y-pred.train = dt.predict(X-train)

y-pred = dt.predict(X-test)

accuracy-score(y-pred, y-test)