# Nama : Rangga Pebrianto

# NIM : G6601231006

# PERTEMUAN 8

**Machine Learning berbasis Algoritma Tree**

**TUJUAN PRAKTIKUM**

Pada praktikum Pereteman 8 akan menerapkan beberapa konsep antara lain :

1. Decision Tree
2. CART
3. Bagging dan Random Forest

Keempat konsep tersebut diterapkan pada sebuah permasalahan dengan menggunakan tools / bahasa pemrograman Python.

1. **Decision Tree**

#Import Library   
import pandas as pd

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

#Melakukan pembacaaan dataset

col\_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']

# load dataset

pima = pd.read\_csv("pima-indians-diabetes.csv", header=None, names=col\_names)

#print(pima)

#split dataset in features and target variable

feature\_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']

x = pima[feature\_cols] # Features

y = pima.label # Target variable

# Split dataset into training set and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.25, random\_state=3)

# Membuat objek DT

# Dapat dioptimalkan dengan menghitung Entropy

clf = DecisionTreeClassifier()

clf = DecisionTreeClassifier(criterion="entropy", max\_depth=3)

# Melakukan Pelatihan DT

clf = clf.fit(X\_train,y\_train)

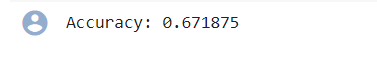
# Memprediksi

y\_pred = clf.predict(X\_test)

# Menghitung akurasi model

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

**Hasil**

****

1. **CART (Classification And Regression Tree)**

# Load Dataset

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.datasets import load\_boston

boston\_dataset = load\_boston()

boston = pd.DataFrame(boston\_dataset.data, columns=boston\_dataset.feature\_names)

boston['MEDV'] = boston\_dataset.target

names = boston\_dataset.feature\_names

#Library CART pada python  
from sklearn.tree import DecisionTreeRegressor

array = boston.values

X = array[:,0:13]

Y = array[:,13]

#print(X)

#print(Y)

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state=1234)

#model = DecisionTreeRegressor(max\_leaf\_nodes = 20)

model = DecisionTreeRegressor(criterion='mse', max\_depth=None, max\_features=None,

max\_leaf\_nodes=50, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

random\_state=None, splitter='best')

#Evaluasi

rt = model.fit(X\_train, Y\_train)

rt

import random as rnd

rnd.seed(123458)

X\_new = X[rnd.randrange(X.shape[0])]

X\_new = X\_new.reshape(1,13)

#Prediksi Model

YHat = model.predict(X\_new)

df = pd.DataFrame(X\_new, columns = names)

df["Predicted Price"] = YHat

df.head(1)

from sklearn.metrics import r2\_score

YHat = model.predict(X\_test)

print(YHat)

#Menghitung Rata-rata Kuadrat

r2 = r2\_score(Y\_test, YHat)

print("R-Squared = ", r2)

**Hasil**

****

1. **Bagging**

#Impor Libraryimport numpy as np

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.pipeline import make\_pipeline

from sklearn.ensemble import BaggingClassifier

from sklearn.model\_selection import GridSearchCV

#Load cancer dataset

bc = datasets.load\_breast\_cancer()

X = bc.data

y = bc.target

#membagi dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=1, stratify=y)

#Melakukan pipelining   
pipeline = make\_pipeline(StandardScaler(),

LogisticRegression(random\_state=1))

**#Skema bagging**bgclassifier = BaggingClassifier(base\_estimator=pipeline, n\_estimators=100,

max\_features=8,

max\_samples=80,

random\_state=1, n\_jobs=5)

bgclassifier.fit(X\_train, y\_train)

print('Model test Score: %.3f, ' %bgclassifier.score(X\_test, y\_test),

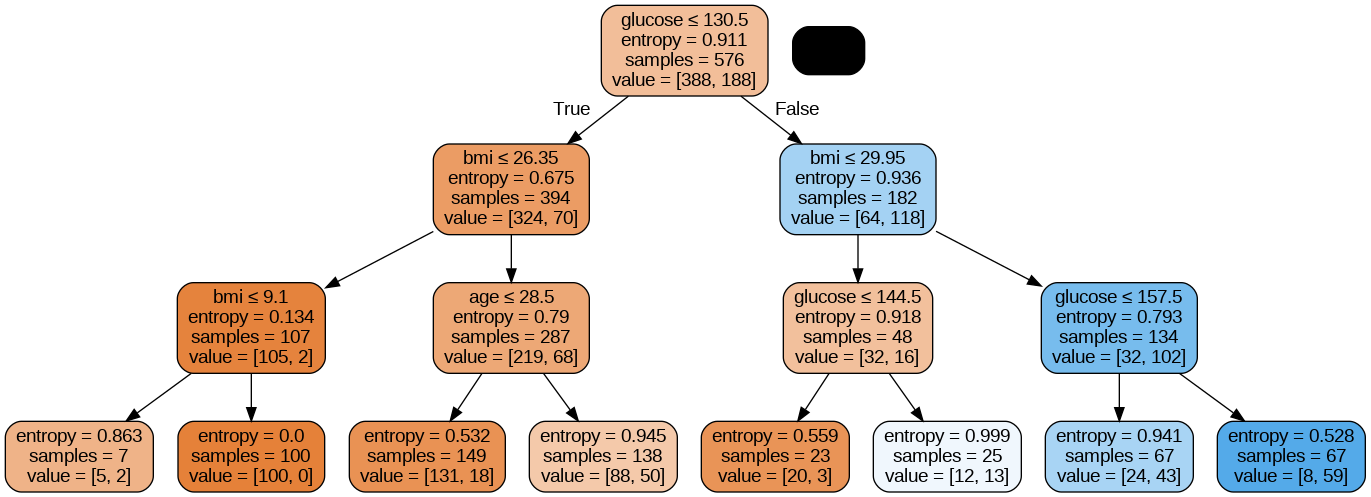
'Model training Score: %.3f' %bgclassifier.score(X\_train, y\_train))

**Hasil**

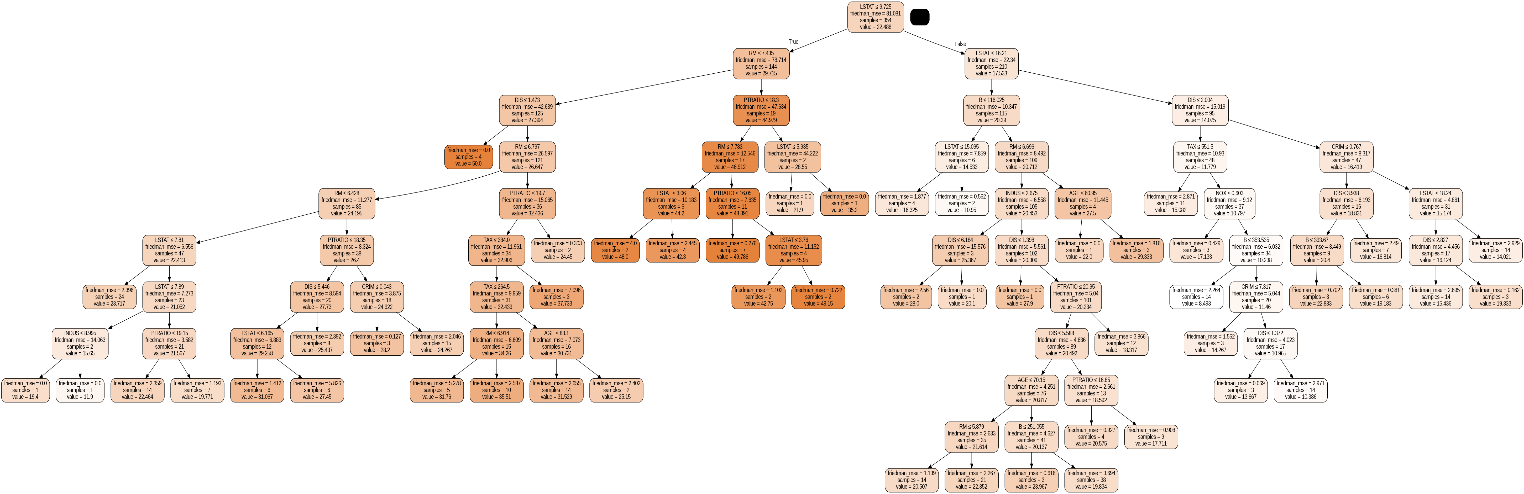
****

**TUGAS PRAKTIKUM**

1. Melakukan visualisasi Tree
   1. Decision Tree



* 1. CART



1. Melakukan tanpa skema bagging
   1. Pada Bagging



**DAFTAR PUSTAKA**

1. Richert W & Coelho LP. *Builidng Machine Learning System with Python*. 2013. Packt Publising. Birmingham, UK.