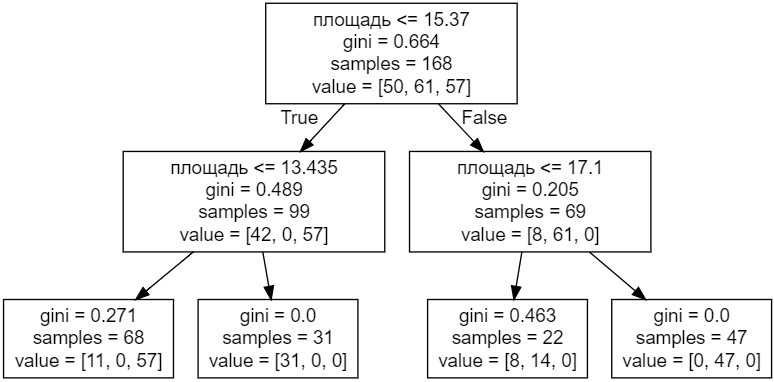
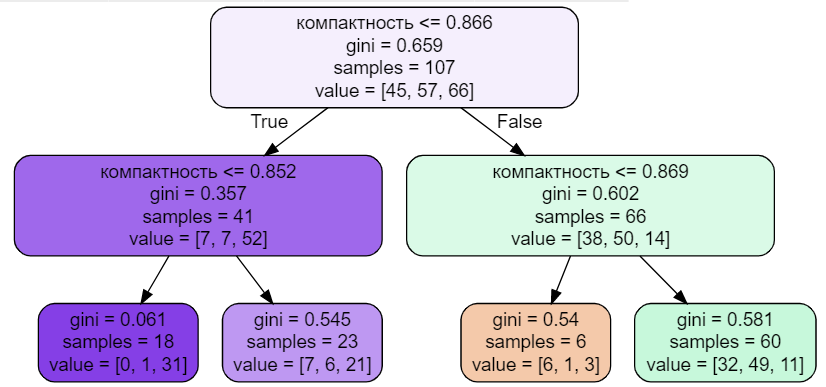
DecisionTreeClassifier:



gini

Score: 0.814

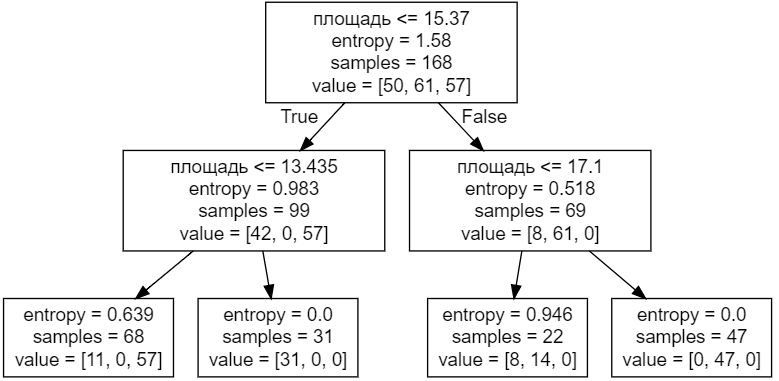
RandomForestClassifier:



gini

Score: 0.809

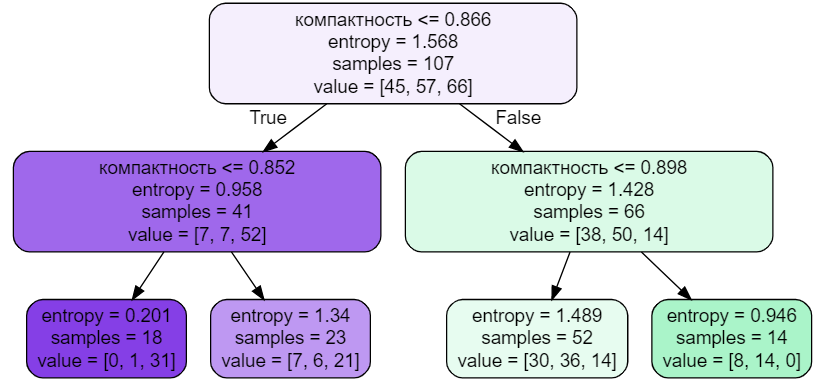
DecisionTreeClassifier:



entropy

Score: 0.809

RandomForestClassifier:



entropy

Score: 0.8

Из оценок точности можно сказать, что критерий gini с глубиной 2 лучше подходит для данных, которые я использовал. При увеличении глубины точность снижается в случае с DecisionTreeClassifier, но с RandomForestClassifier незначительно растет.

|  |  |
| --- | --- |
| Глубина 4. entropy | Глубина 4. entropy |
| Глубина 4. gini | Глубина 4.giny |

**Код программы**

import numpy as np

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.model\_selection import cross\_val\_score

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

from mlxtend.plotting import plot\_decision\_regions

# from sklearn import datasets

from sklearn.svm import SVC

import pandas as pd

#деревья

url = "C:/Users/fox/Desktop/ii/data/seeds\_dataset.data"

names = [

    "площадь",

    "периметр",

    "компактность",

    "длина",

    "ширина",

    "асимметрия",

    "Длина канавки ядра",

    "Сорт"

] #название атрибутов

dataset = pd.read\_csv(url, names=names) #чтение файла с данными

dataset.head()

X = dataset[["площадь","компактность"]]

X = X.values

Y = dataset[["Сорт"]]

Y = Y.values

#только сорт

#print(dataset.shape)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(

    X, Y, test\_size=0.20, random\_state=1

)

X\_combined=np.vstack((X\_train,X\_test))

Y\_combined=np.vstack((Y\_train,Y\_test))

#создание

from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(criterion="gini",

                              max\_depth=2,

                              random\_state=0) #или entropy"

tree.fit(X\_train, Y\_train)

scores = cross\_val\_score(tree, X, Y, cv=5)

print('DecisionTreeClassifier:')

print('Score: ',scores.mean())

#отрисовка

from sklearn.tree import export\_graphviz

export\_graphviz(

    tree, out\_file="treeDecision.dot", feature\_names=["площадь", "компактность"]

)

from sklearn.ensemble import RandomForestClassifier

forest = RandomForestClassifier(

                                max\_depth=2,

                                random\_state=0)

forest.fit(X\_train, Y\_train)

estimator = forest.estimators\_[5]

scores = cross\_val\_score(forest, X, Y, cv=5)

print("RandomForestClassifier:")

print('Score: ',scores.mean())

dot\_data = export\_graphviz(estimator, out\_file="treeRandom.dot", max\_depth=2,feature\_names=["площадь", "компактность"], filled=True, rounded=True,)