МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №7

по дисциплине «Машинное обучение»

Тема: Классификация (Байесовские методы, деревья)

Студент гр. 6307 _____ Золотухин М. А.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Байесовские методы

Проведем классификацию наблюдений наивным байесовским методом.

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
print((y_test ≠ y_pred).sum()) #количество наблюдений, который были неправильно определены
```

```
4
```

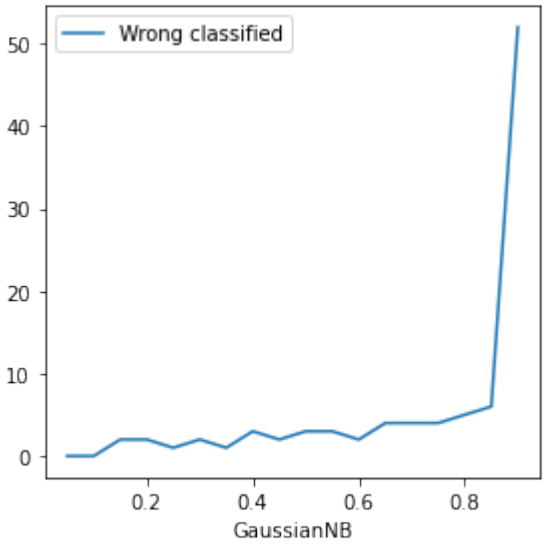
Атрибуты данного классификатора:

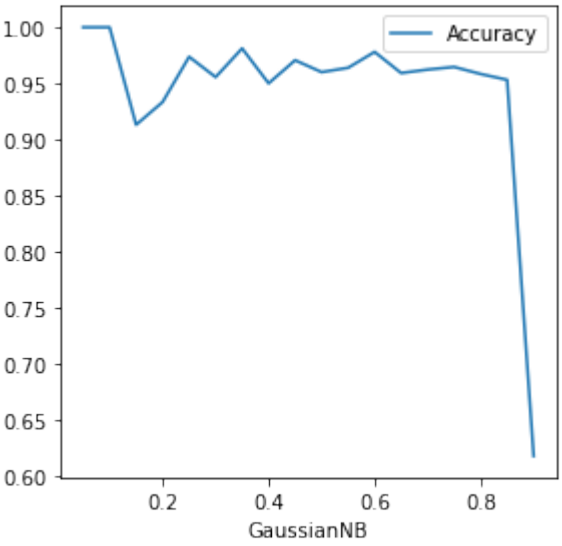
- class*count* - количество тренировочных сэмплов, которые наблюдаются в каждом классе.
- class*prior*- вероятность каждого из классов.
- classes_- метки классов, известные классификатору.
- epsilon_- абсолютное добавочное значение для дисперсий.
- sigma_- дисперсия каждой фичи в каждом классе.
- theta_- среднее кажой фичи каждого класса.

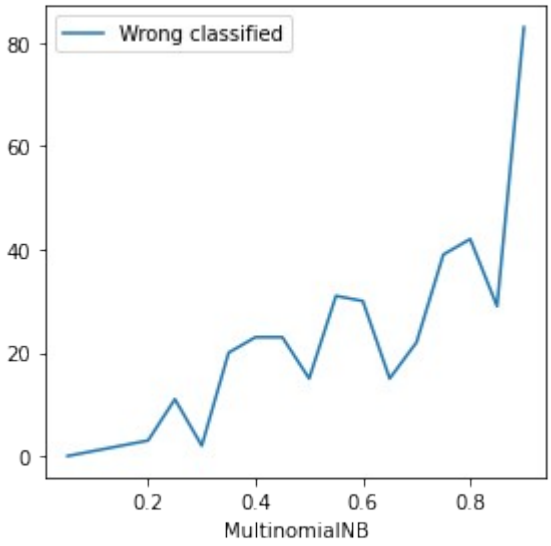
Точность классификации:

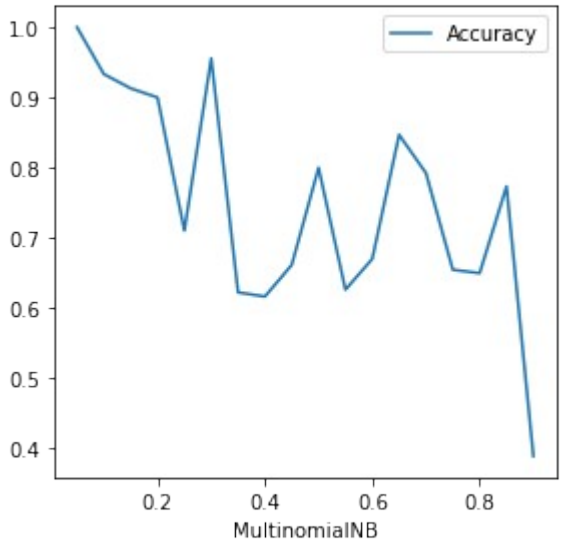
```
print(gnb.score(X_test, y_test))
0.9466666666666667
```

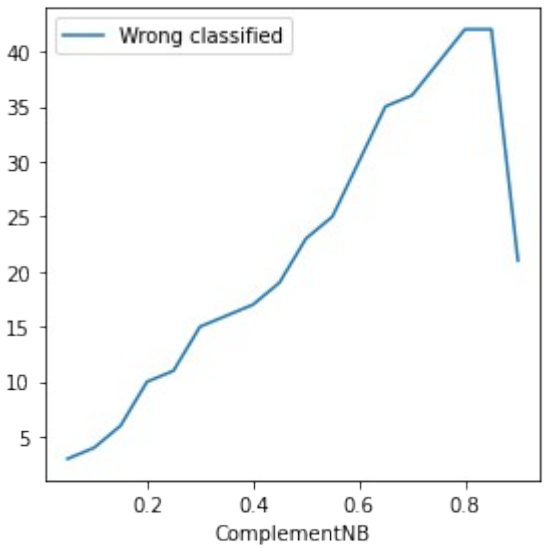
Построим график зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. Размер тестовой выборки изменяйте от 0.05 до 0.95 с шагом 0.05. Параметр random_state сделаем равным номеру своей зачетной книжки. Проведём также классификацию используя MultinomialNB, ComplementNB, BernoulliNB.

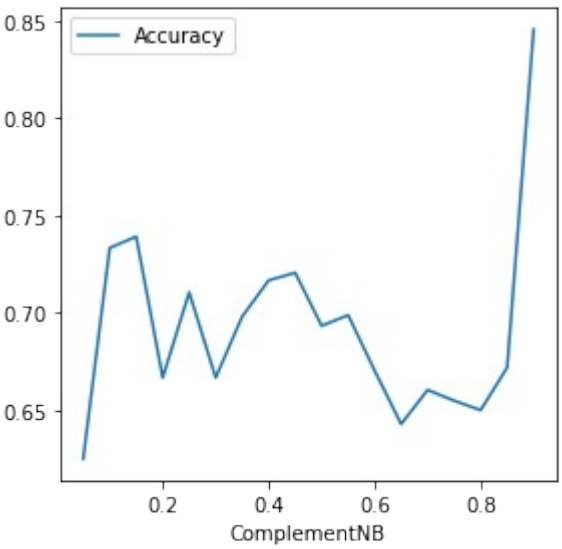

Wrong classified
50
40
30
20
10
0
0.2
0.4
0.6
0.8
GaussianNB

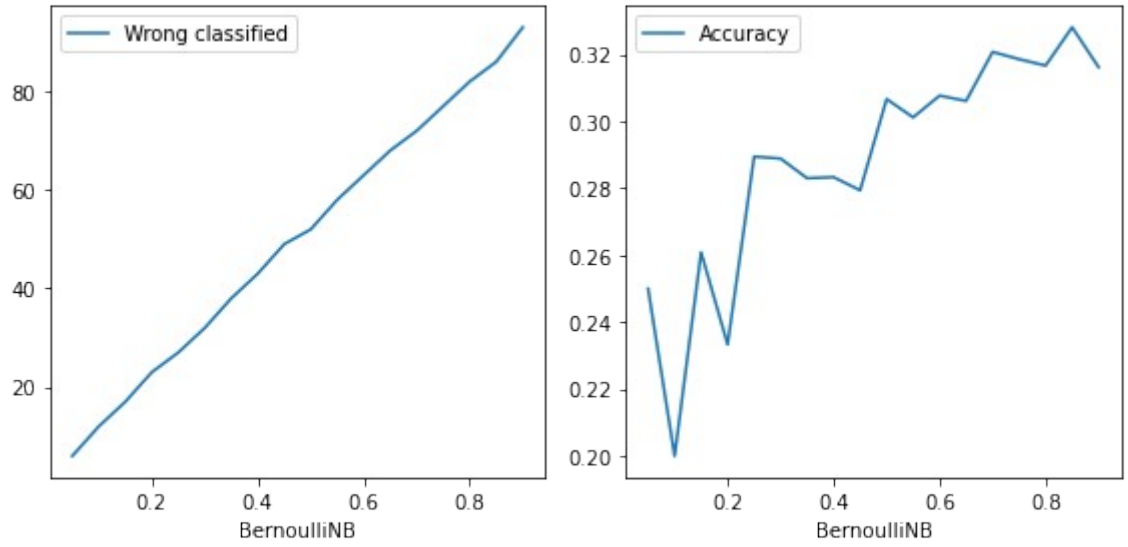

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.2
0.4
0.6
0.8
GaussianNB


Wrong classified
80
60
40
20
0
0.2
0.4
0.6
0.8
MultinomialNB


Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.2
0.4
0.6
0.8
MultinomialNB


Wrong classified
40
35
30
25
20
15
10
5
0.2
0.4
0.6
0.8
ComplementNB


Accuracy
0.85
0.80
0.75
0.70
0.65
0.2
0.4
0.6
0.8
ComplementNB



Gaussian — стандартный алгоритм, подразумевает, что фичи распределяются нормально.

Multinomial — алгоритм подрузумевает, что данные подчиняются мультиномиальному распределению.

Complement — алгоритм, который используется для не сбалансированных датасетов.

Bernoulli — алгоритм подрузумевает, что данные подчиняются распределению Бернулли.

Классифицирующие деревья

Проведем классификацию при помощи деревьев на тех же данных.

```
from sklearn import tree

clf = tree.DecisionTreeClassifier()

y_pred = clf.fit(X_train, y_train).predict(X_test)
print((y_test ≠ y_pred).sum())
4
```

Используя функцию score() выведите точность классификации

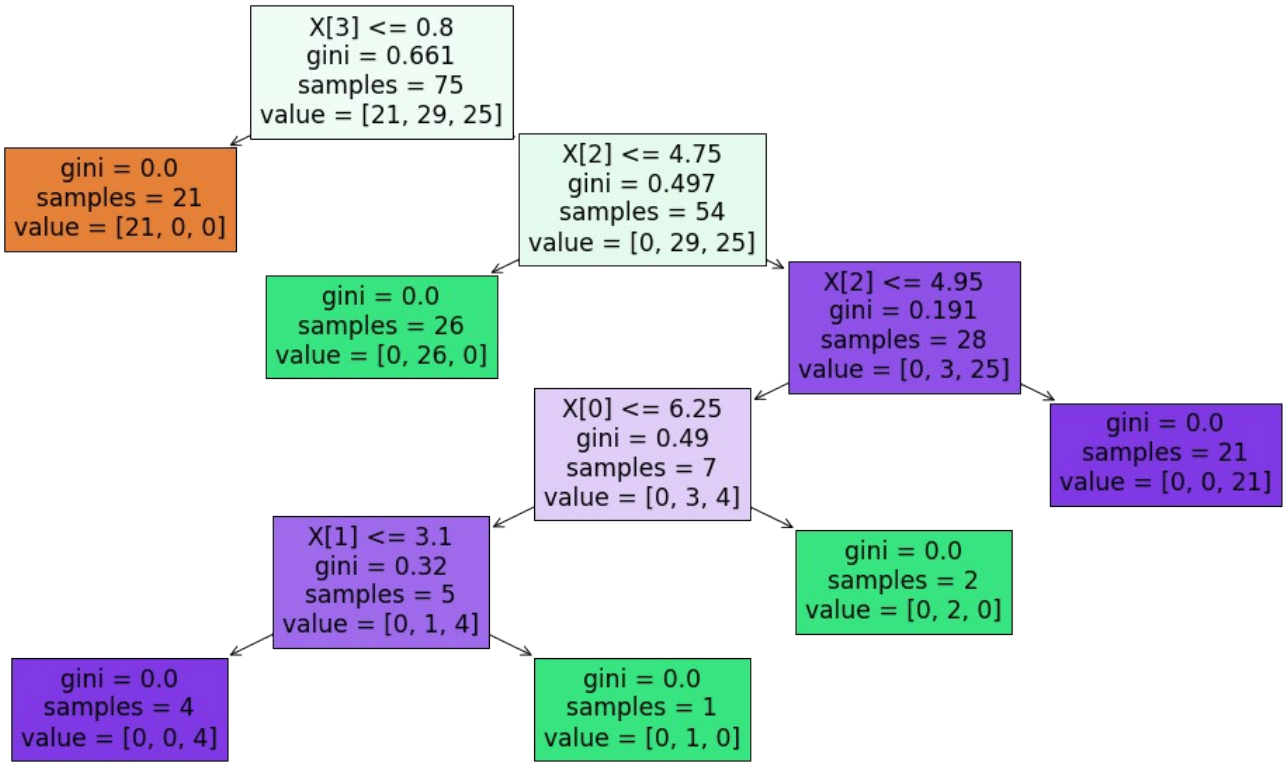
```
print("Score: ", clf.score(X_test, y_test))
Score:  0.9466666666666667
```

Выведите характеристики дерева, количество листьев и глубину, используя функции get_n_leaves и get_depth

```
print('Num of leaves: ', clf.get_n_leaves())
print('Depth: ', clf.get_depth())
Num of leaves:  6
Depth:  5
```

Выведем изображение полученного дерева

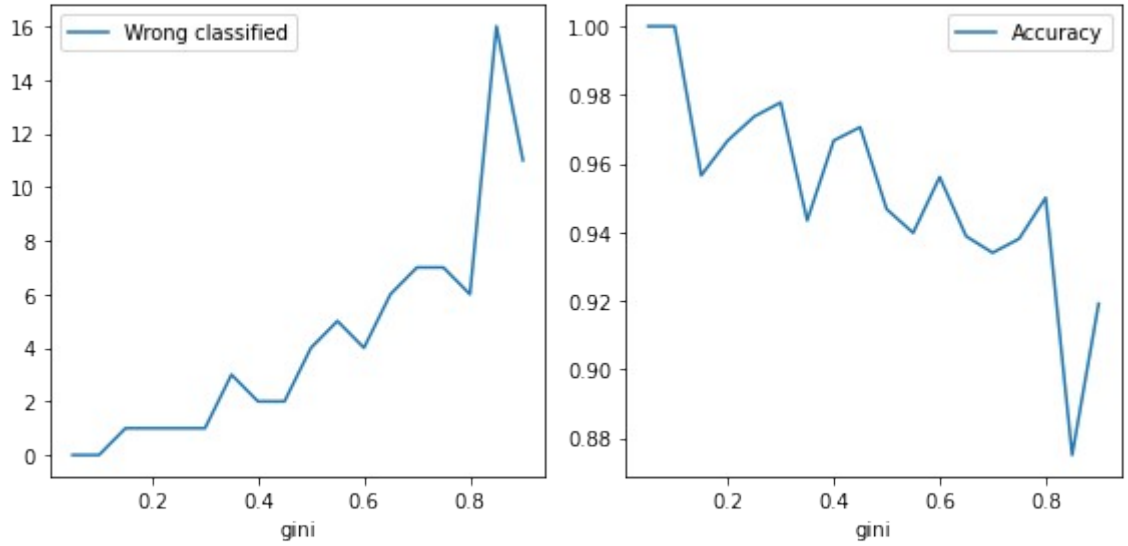
```
import matplotlib.pyplot as plt

plt.subplots(1,1,figsize = (20,10))
tree.plot_tree(clf, filled = True)
plt.show()
```



На рисунке изображено дерево. Каждый узел — вопрос, который представлен первой строкой. В зависимости от ответа на вопрос (да или нет), данное идет по одной ветке или по другой, пока не дойдет до листа. Samples — количество измерений, которые прошли через узел. Gini — примесь gini. Value — количества узлов в кажом из классов (в данном случае класса 3, как и элементов массива).

Построим график зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. Размер тестовой выборки будет изменяться от 0.05 до 0.95 с шагом 0.05. Параметр random_state равен 630706.

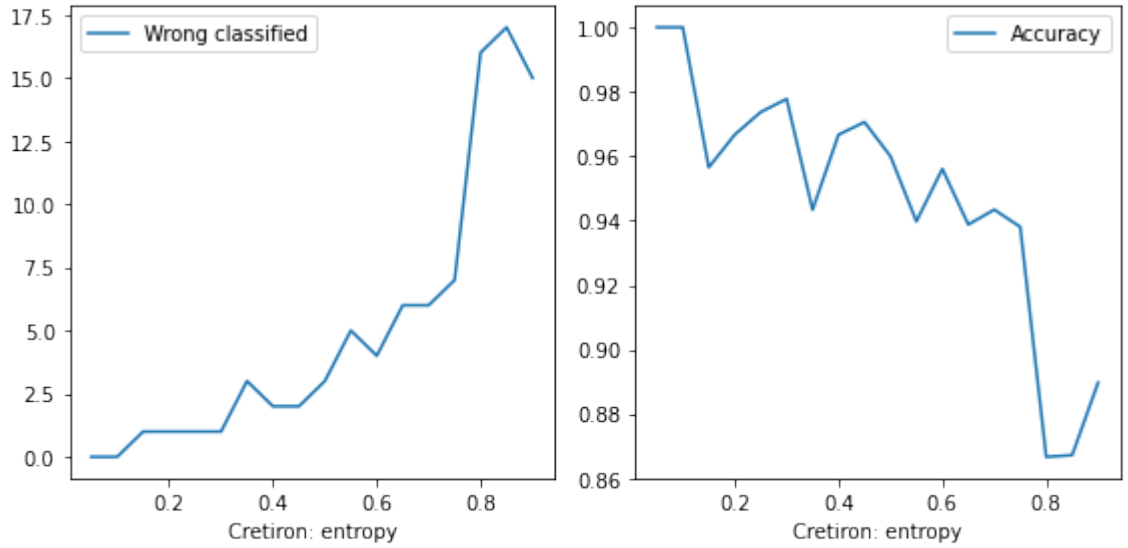
```
plot_clf(clf, "gini")
```



Исследуем работу классифицирующего дерева при различных параметрах criterion, splitter, max_depth, min_samples_split, min_samples_leaf.

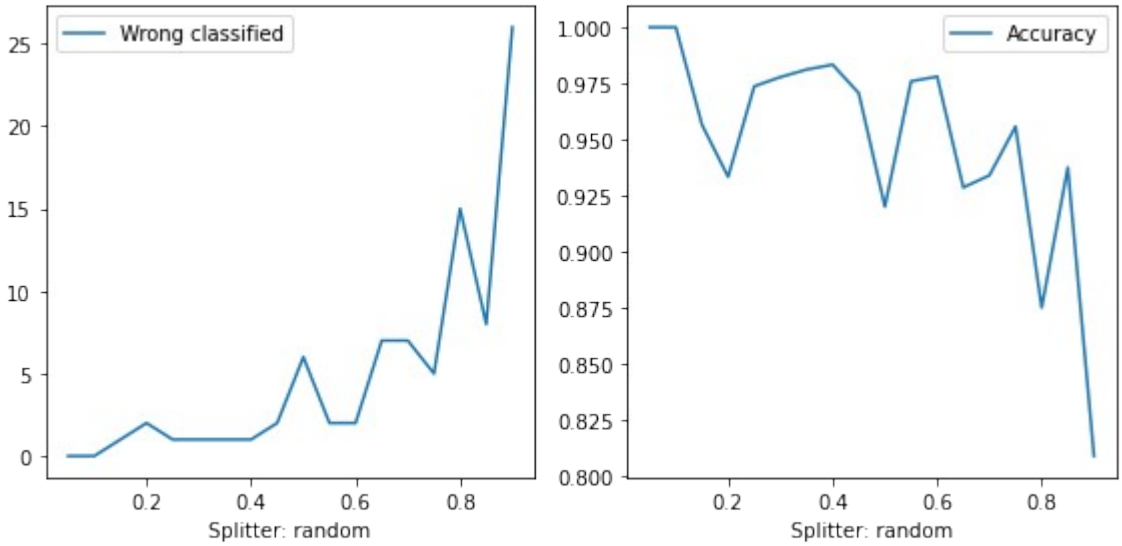
Entropy criterion

```
clf = tree.DecisionTreeClassifier(criterion="entropy")
plot_clf(clf, "Cretiron: entropy")
```



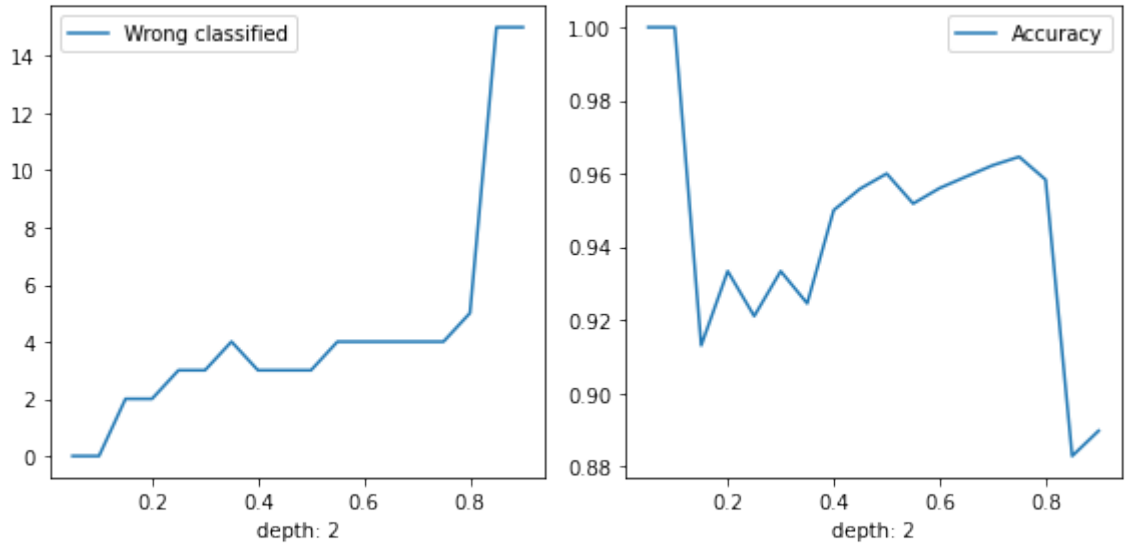
Random Splitter

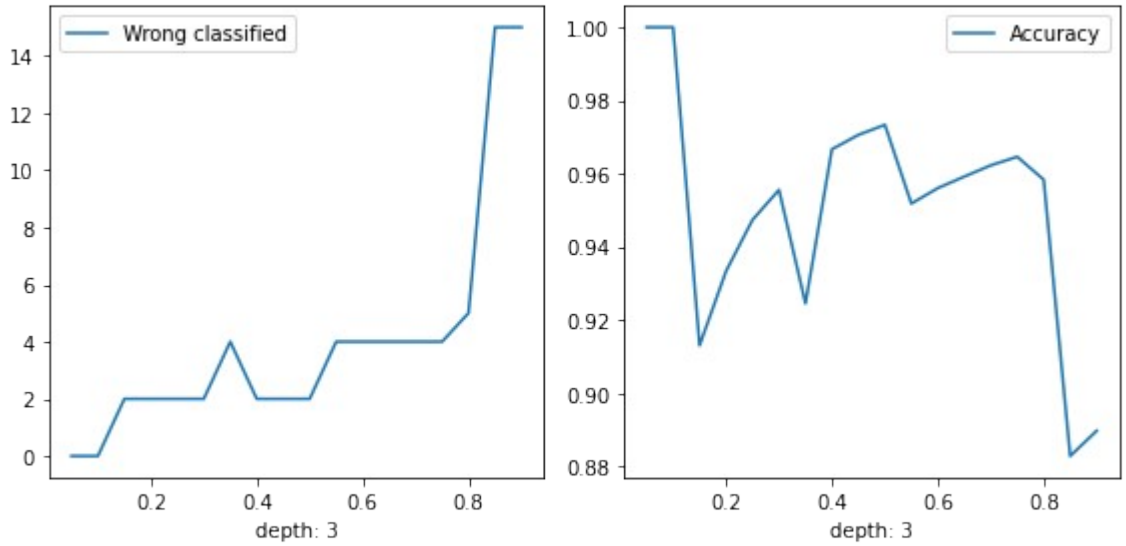
```
clf = tree.DecisionTreeClassifier(splitter="random")

plot_clf(clf, "Splitter: random")
```

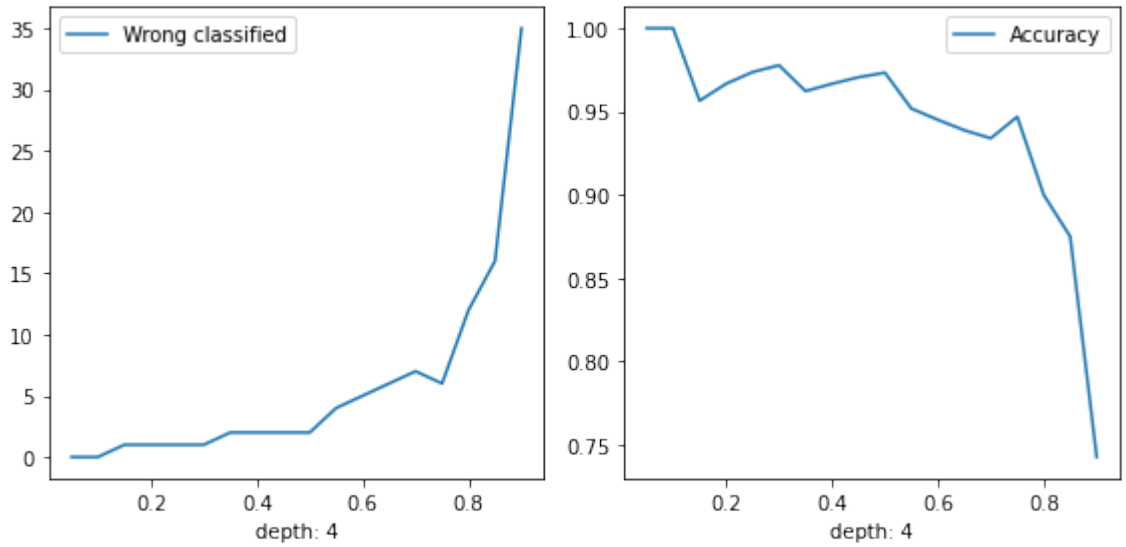


Depth

```
depths = np.arange(2, 5, 1)
for depth in depths:
    clf = tree.DecisionTreeClassifier(max_depth=depth)
    plot_clf(clf, f'depth: {depth}')
```

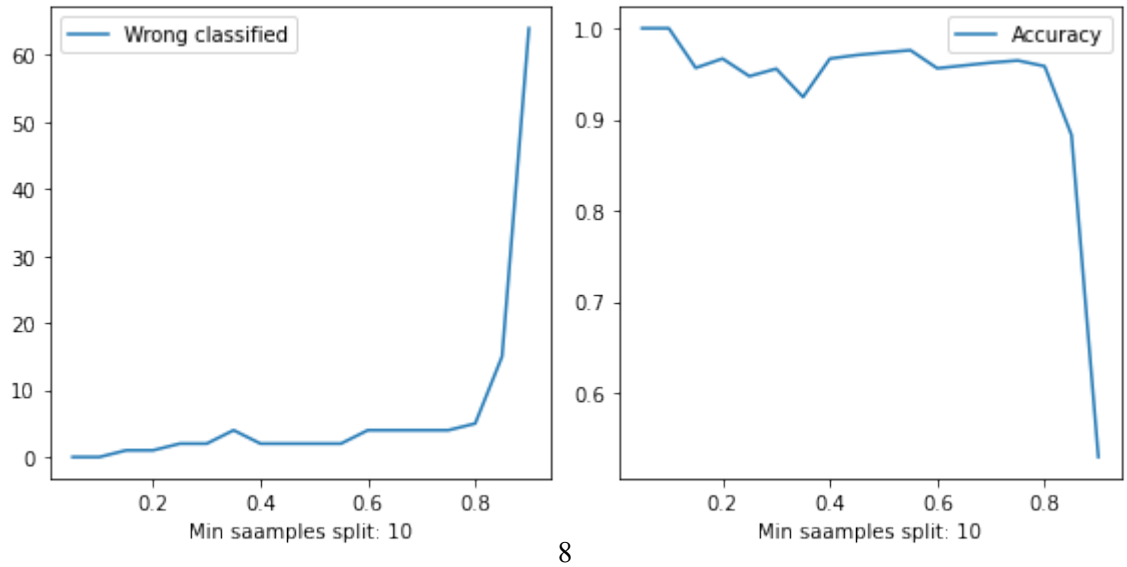


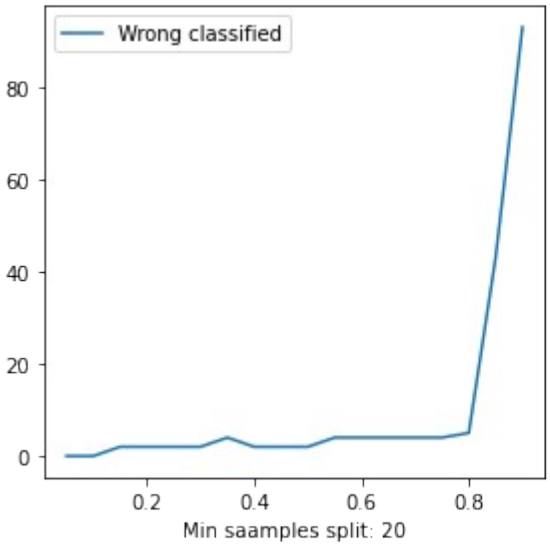


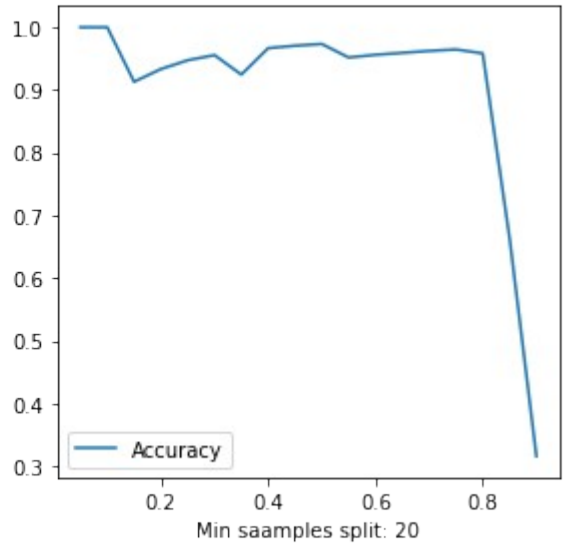


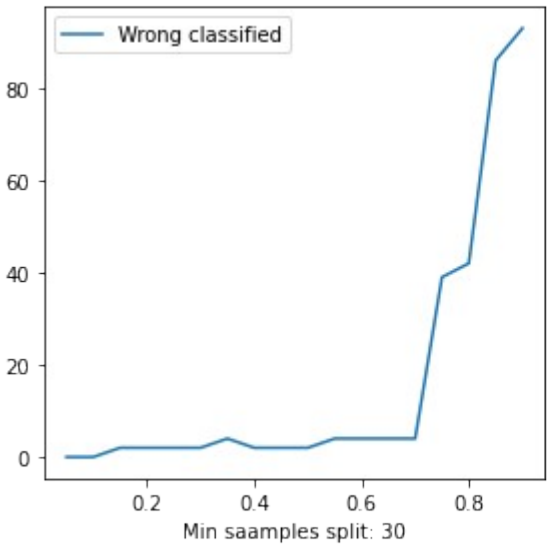
Min Samples Split

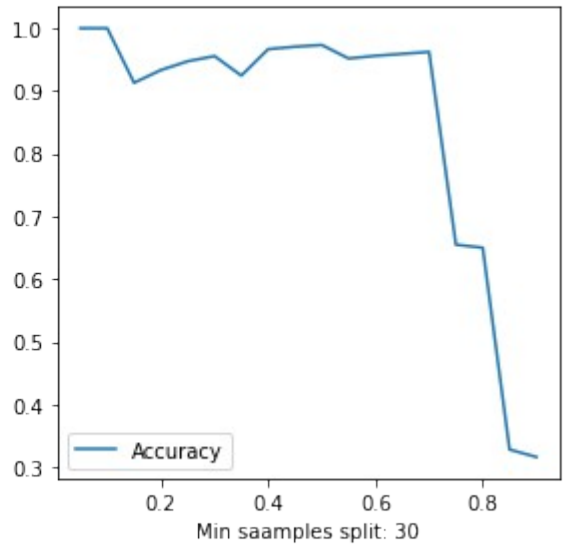
```
min_samples = np.arange(10, 90, 10)
for min_sample in min_samples:
    clf = tree.DecisionTreeClassifier(min_samples_split=min_sample)
    plot_clf(clf, f'Min saamples split: {min_sample}')
```

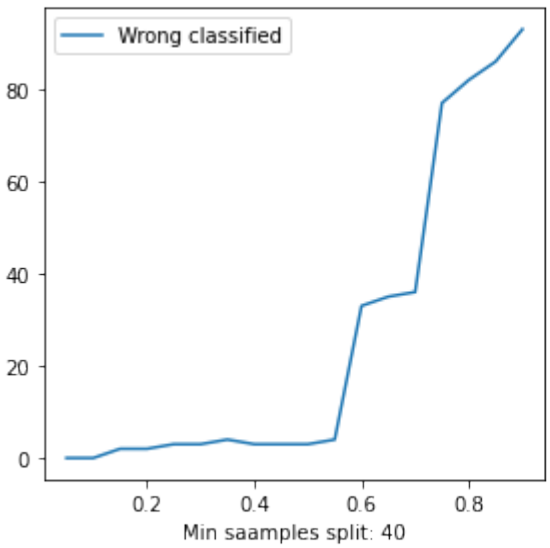


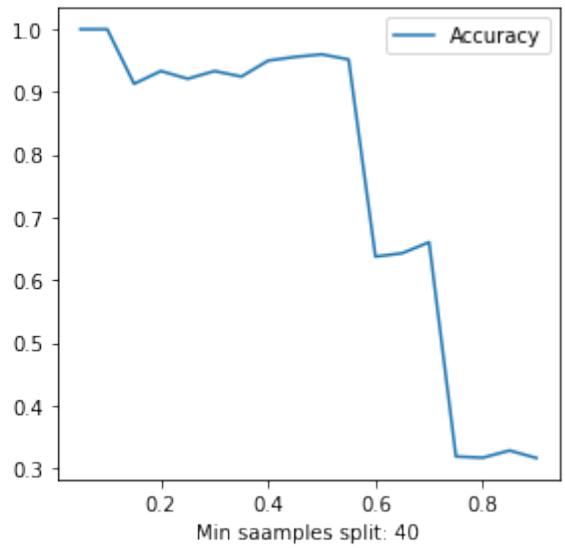

Wrong classified
Min saamples split: 20

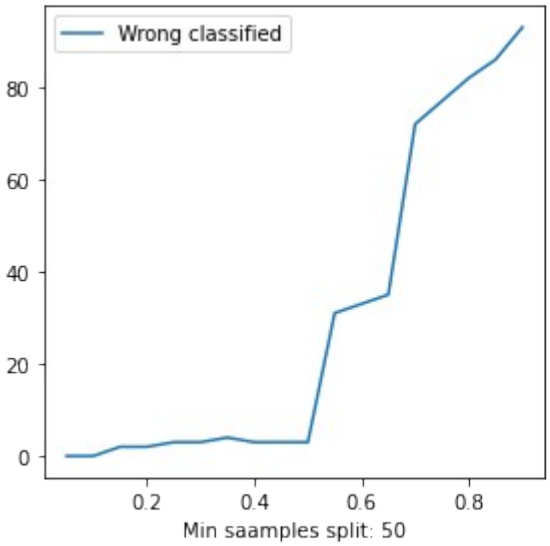

Accuracy
Min saamples split: 20

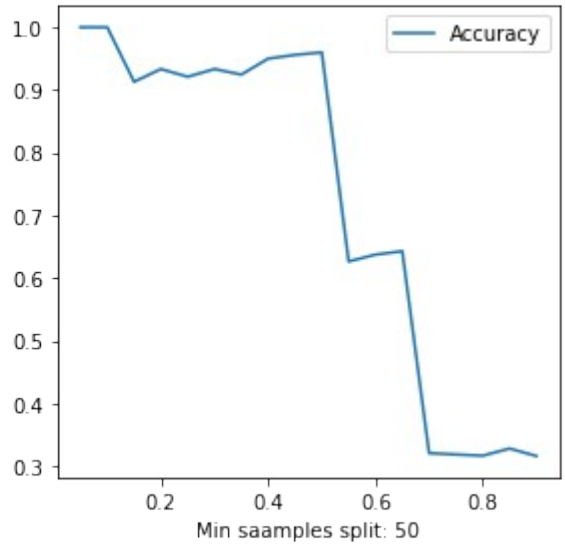

Wrong classified
Min saamples split: 30

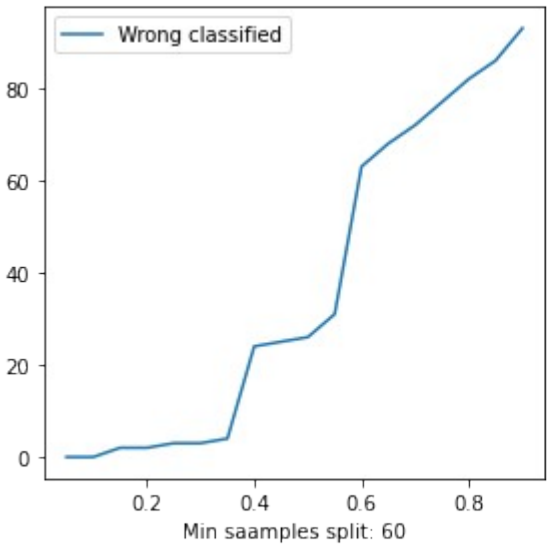

Accuracy
Min saamples split: 30

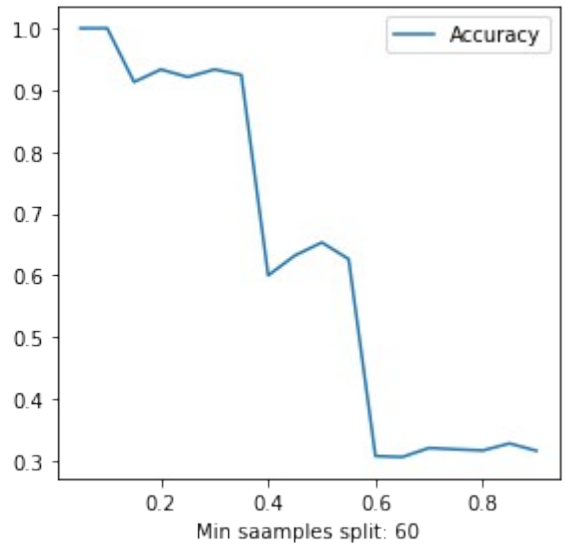

Wrong classified
Min saamples split: 40

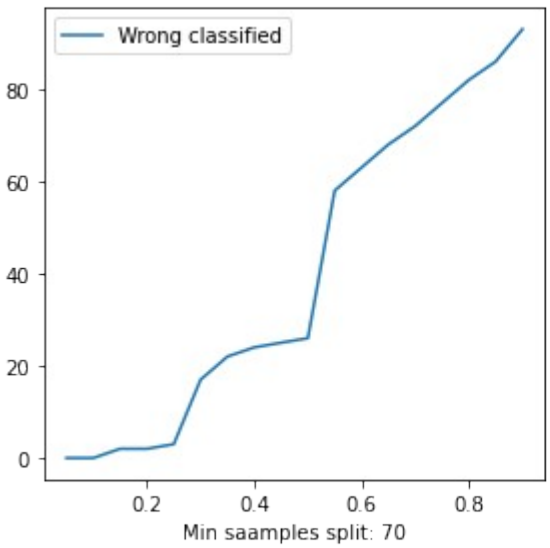

Accuracy
Min saamples split: 40

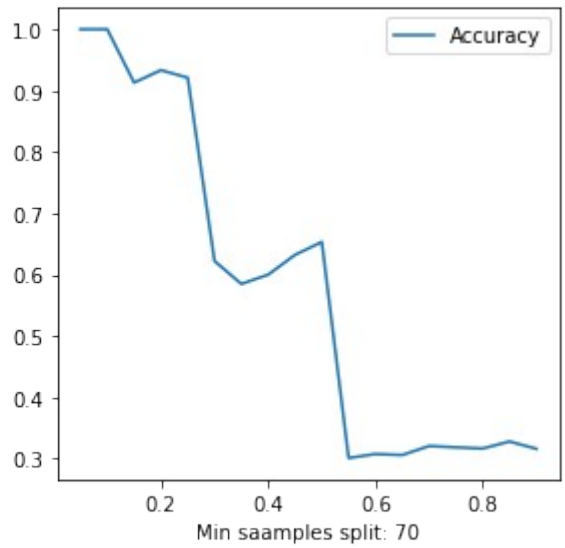

Wrong classified
Min saamples split: 50

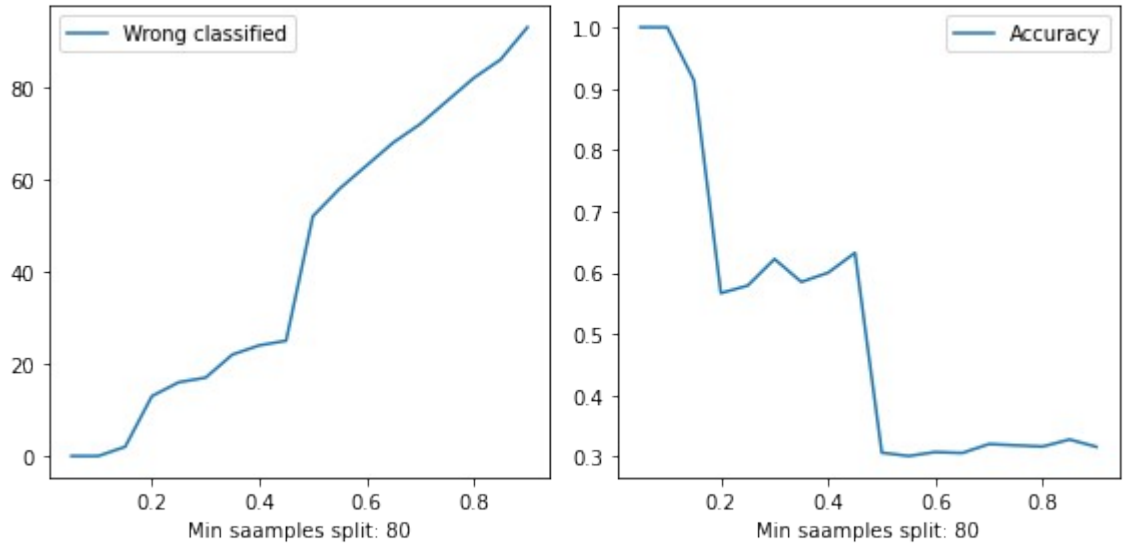

Accuracy
Min saamples split: 50


Wrong classified
Min saamples split: 60


Accuracy
Min saamples split: 60

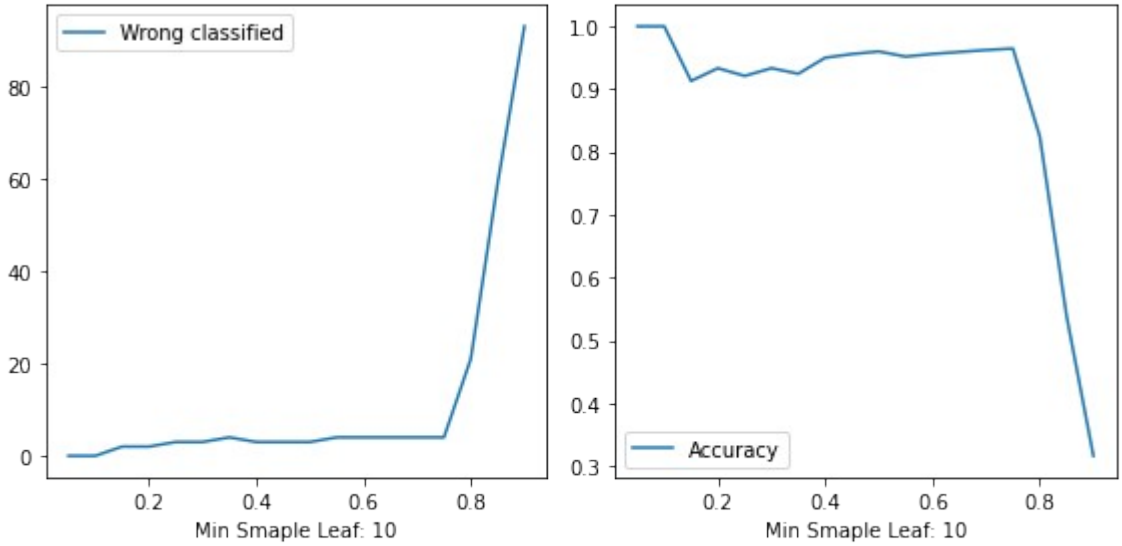

Wrong classified
Min saamples split: 70

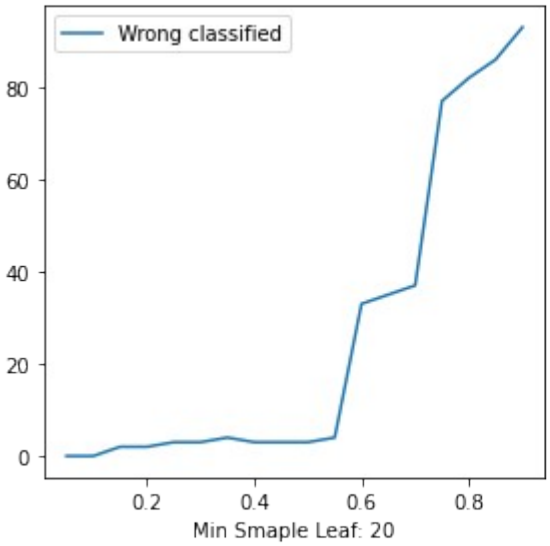

Accuracy
Min saamples split: 70

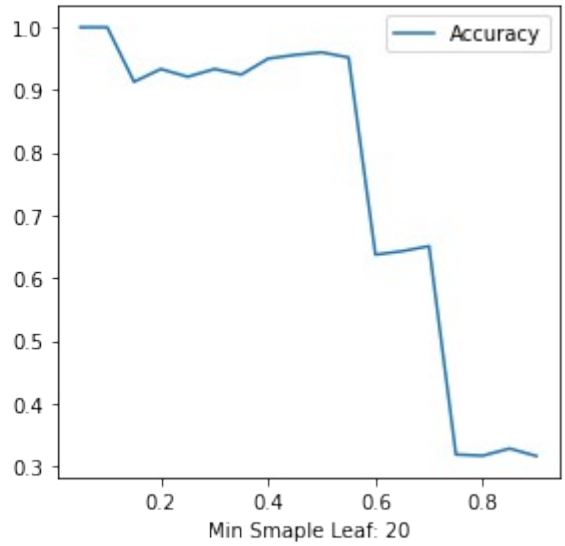


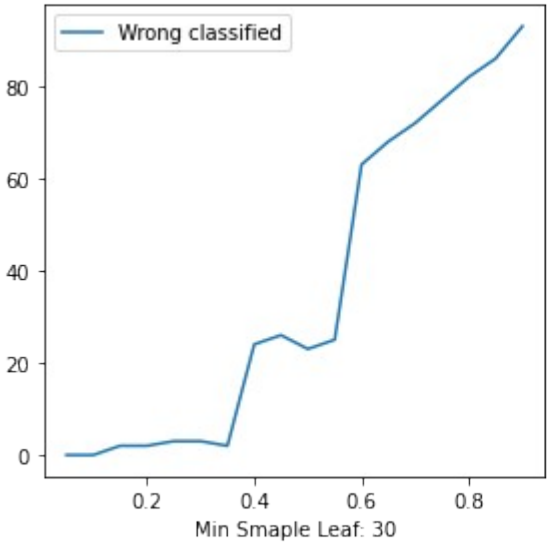
Min sample leaf

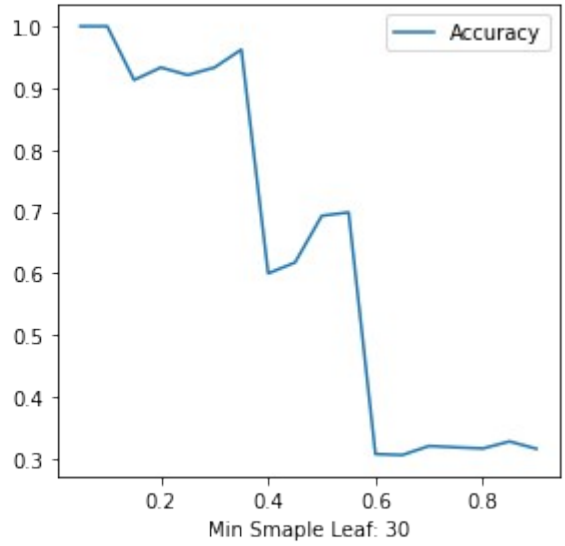
```
min_samples = np.arange(10, 90, 10)
for min_sample in min_samples:
    clf = tree.DecisionTreeClassifier(min_samples_leaf=min_sample)
    plot_clf(clf, f'Min Smaple Leaf: {min_sample}')
```

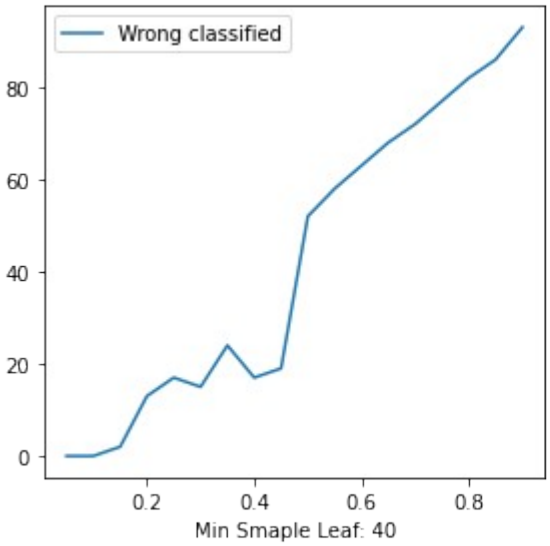


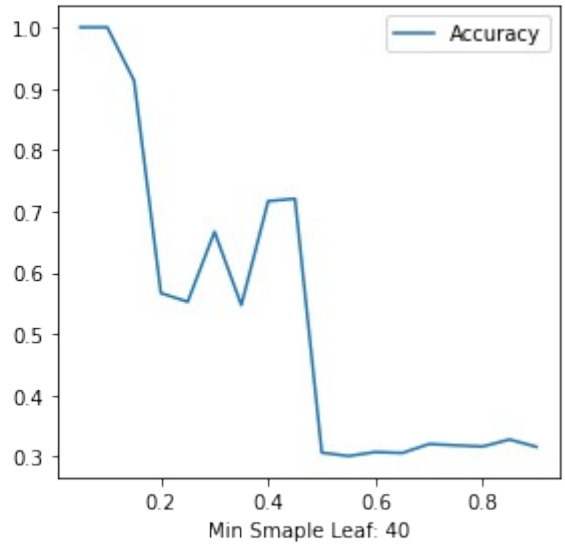

Wrong classified
Min Smaple Leaf: 20

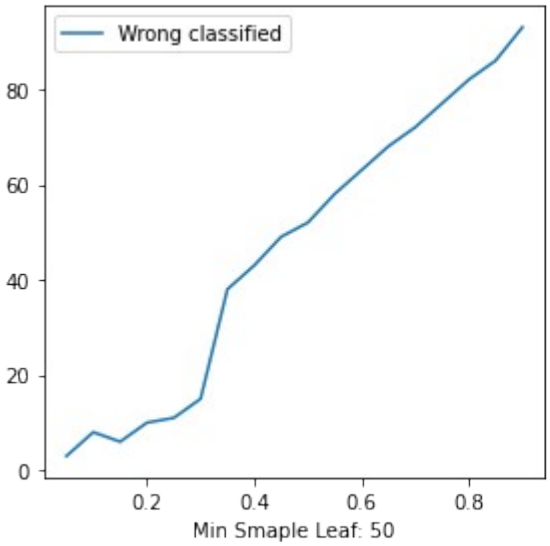

Accuracy
Min Smaple Leaf: 20

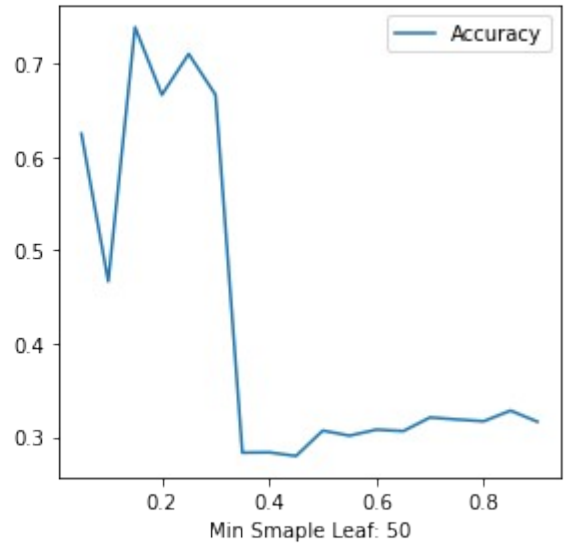

Wrong classified
Min Smaple Leaf: 30

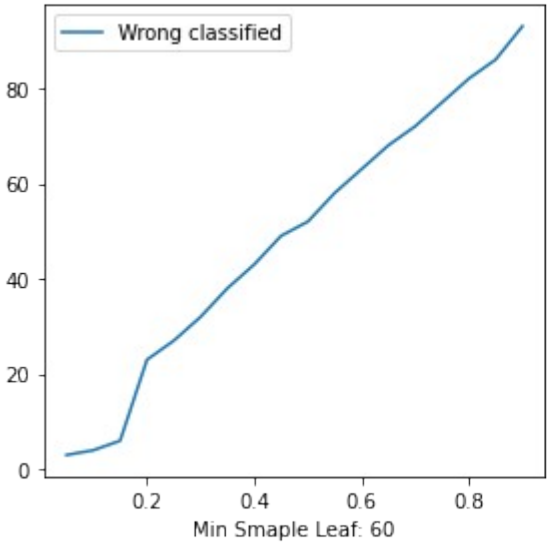

Accuracy
Min Smaple Leaf: 30

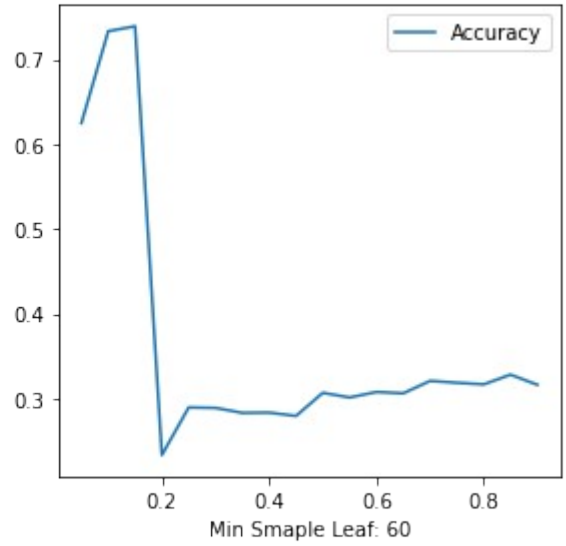

Wrong classified
Min Smaple Leaf: 40

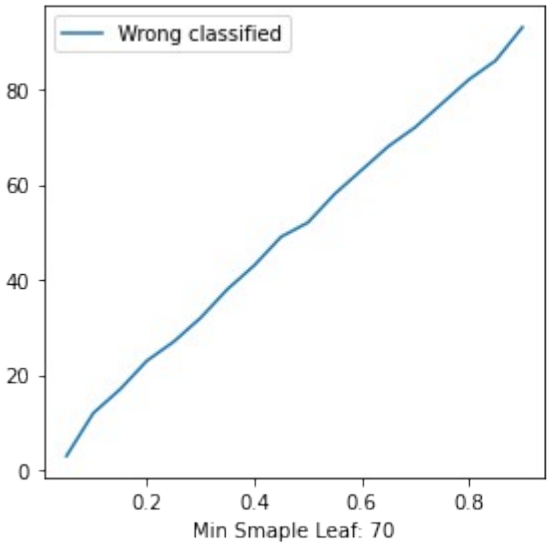

Accuracy
Min Smaple Leaf: 40

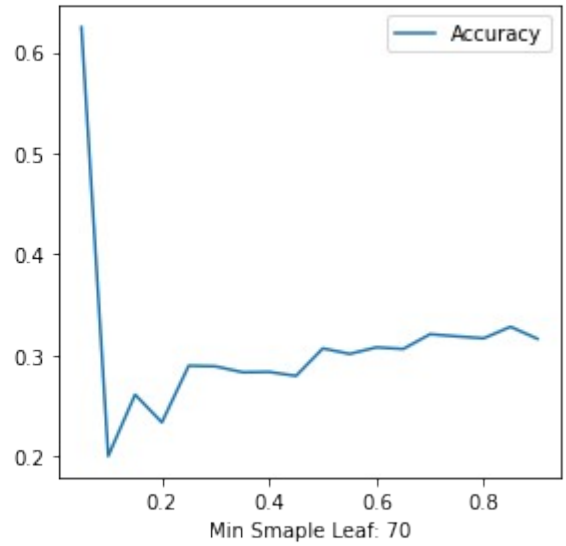

Wrong classified
0
20
40
60
80
0.2
0.4
0.6
0.8
Min Smaple Leaf: 50

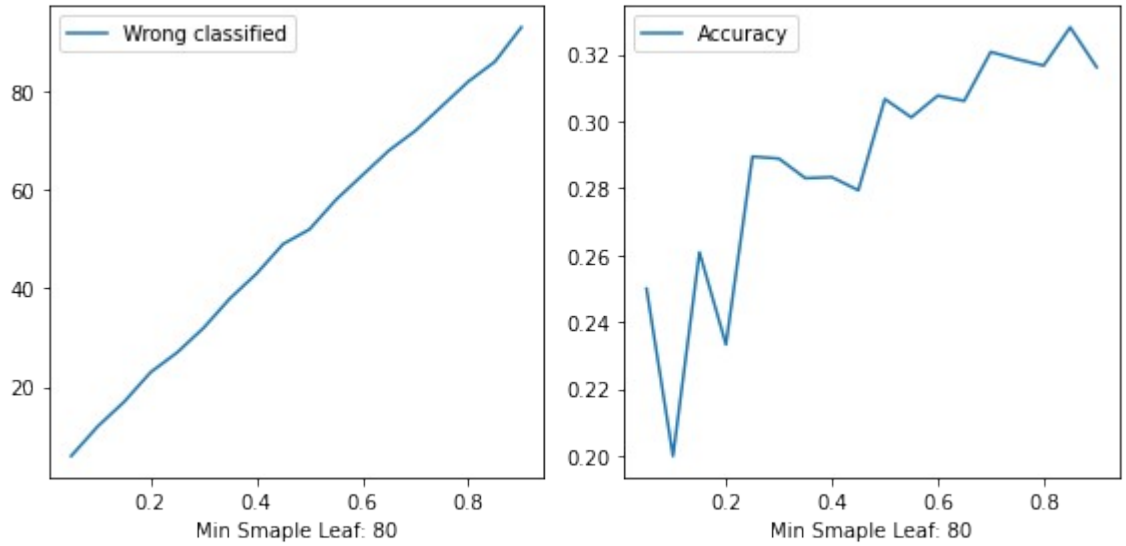

Accuracy
0.3
0.4
0.5
0.6
0.7
0.2
0.4
0.6
0.8
Min Smaple Leaf: 50


Wrong classified
0
20
40
60
80
0.2
0.4
0.6
0.8
Min Smaple Leaf: 60


Accuracy
0.3
0.4
0.5
0.6
0.7
0.2
0.4
0.6
0.8
Min Smaple Leaf: 60


Wrong classified
0
20
40
60
80
0.2
0.4
0.6
0.8
Min Smaple Leaf: 70


Accuracy
0.2
0.3
0.4
0.5
0.6
0.2
0.4
0.6
0.8
Min Smaple Leaf: 70



Объяснение параметров:

- Criterion — функция для измерения качества разбиения. Выбор значения повлиял на результат незначительно, и характер остался тем же.
- Splitter — стратегия выбора разделения. Поскольку один из вариантов — это рандом, то итоговый результат получился прежним, но с некоторым шумом.
- Max_depth — максимальная глубину дерева решений. При увеличении глубины, разумеется, точность повышается, правда, если перестараться, то может возникнуть оверфиттинг.
- Min_samples_split — минимальное число наблюдений необходимых для разбиения узла дерева. С увеличением значения наблюдается ухудшение классификации.
- Min_samples_leaf — минимальное число наблюдений, требующееся для листа. Чем больше параметр, тем хуже классификация.