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 _____ Новиков Б.М.

Преподаватель _____ Жангиров Т.Р.

2020

ЗАГРУЗКА ДАННЫХ

1-2. Загрузить данные в датафрейм
```
data = pd.read_csv('data/iris.data', header=None)
data.head()
```

	0	1	2	3	4
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

3. Выделить данные и их метки
```
X = data.iloc[:,:4].to_numpy()
labels = data.iloc[:,4].to_numpy()
```

4. Преобразовать тексты меток к числам
```
le = preprocessing.LabelEncoder()
Y = le.fit_transform(labels)
```

5. Разбить выборку на обучающую и тестовую
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5)
```

БАЙЕСОВСКИЕ МЕТОДЫ

1. Провести классификацию наблюдений наивным байесовским методом
```
gnb = GaussianNB()
gnb.fit(X_train, y_train)

y_pred = gnb.predict(X_test)

print((y_test != y_pred).sum())
```

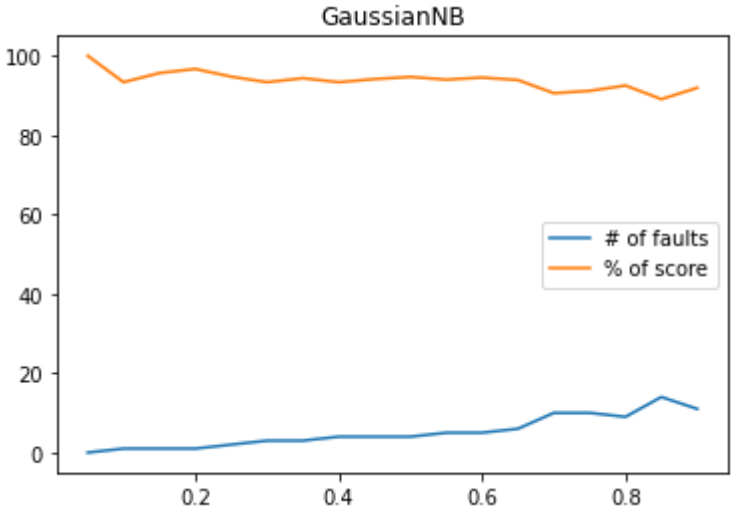
Неправильно предсказано: 3

Описать атрибуты данного классификатора.

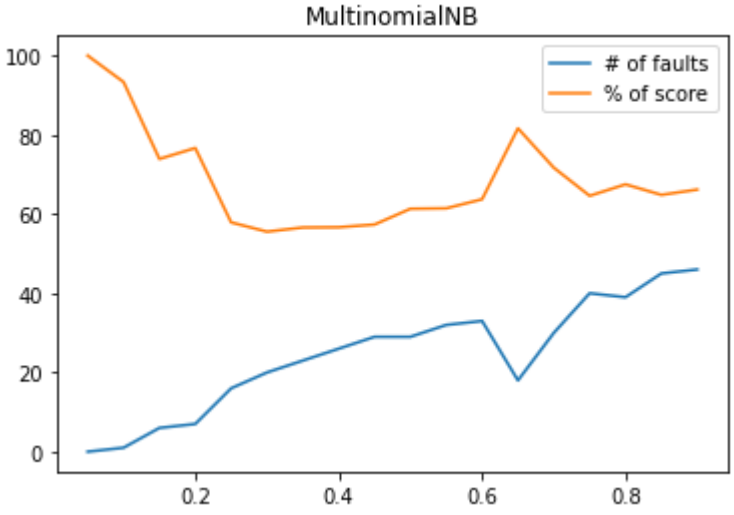
Атрибут	Описание
class_count_	Количество тренировочных семплов наблюдаемых в одном классе
class_prior_	Вероятность каждого класса
classes_	Метки классов, известные классификатору
epsilon_	Абсолютное аддитивное значение к дисперсии
sigma_	Дисперсия каждого признака для класса
theta_	Среднее каждого признака для класса

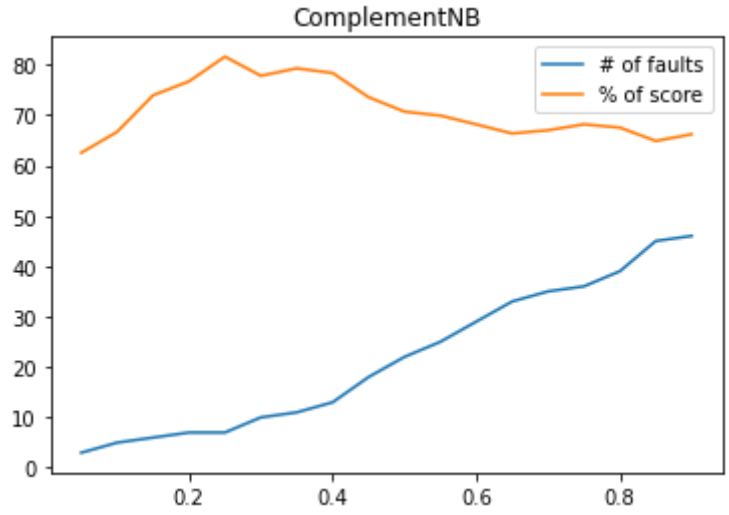
2. Использовать функцию score() для выведения точности классификации
```
gnb.score(X_test, y_test)
```
Точность : 0.96

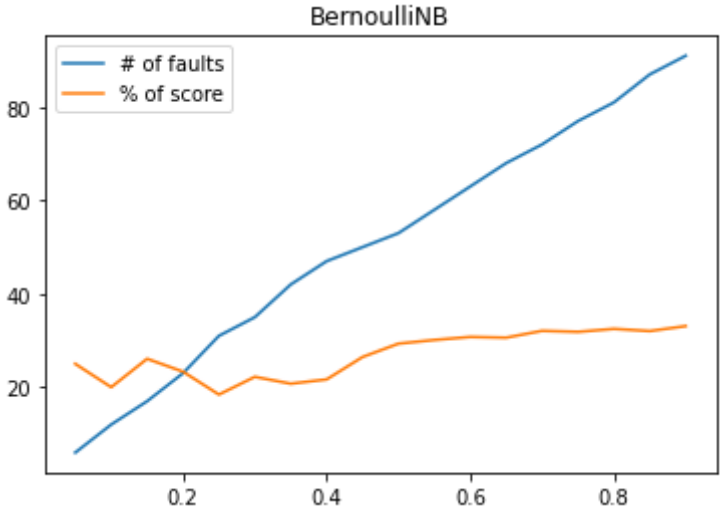
3. Построить график зависимости неправильно класс. наблюдений и точности классификации от размера выборки.



4. Провести классификацию методами MultinomialNB, ComplementNB, BernoulliNB







Описать особенности методов

Метод	Особенность
MultinomialNB	Полиномиальный наивный байесовский классификатор подходит для классификации с дискретными функциями (например, подсчетом слов для классификации текста). Полиномиальное распределение обычно требует целочисленного подсчета признаков. Однако на практике дробные подсчеты, такие как tf-idf, также могут работать.
ComplementNB	Дополнительный наивный байесовский классификатор был разработан для исправления «серьезных допущений», сделанных стандартным полиномиальным наивным байесовским классификатором. Он особенно подходит для несбалансированных наборов данных.
BernoulliNB	Как и MultinomialNB, этот классификатор подходит для дискретных данных. Разница в том, что в то время как MultinomialNB работает с подсчетом вхождений, BernoulliNB предназначен для двоичных / логических функций.

Классифицирующие деревья

1. Классификация тех же данных при помощи деревьев

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5)
clf = tree.DecisionTreeClassifier()
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

print((y_test != y_pred).sum())
```

Количество неправльных предсказаний: 4

2. Вывести точность классификации

```
clf.score(X_test, y_test)
```

Точность: 0.9466666666666667

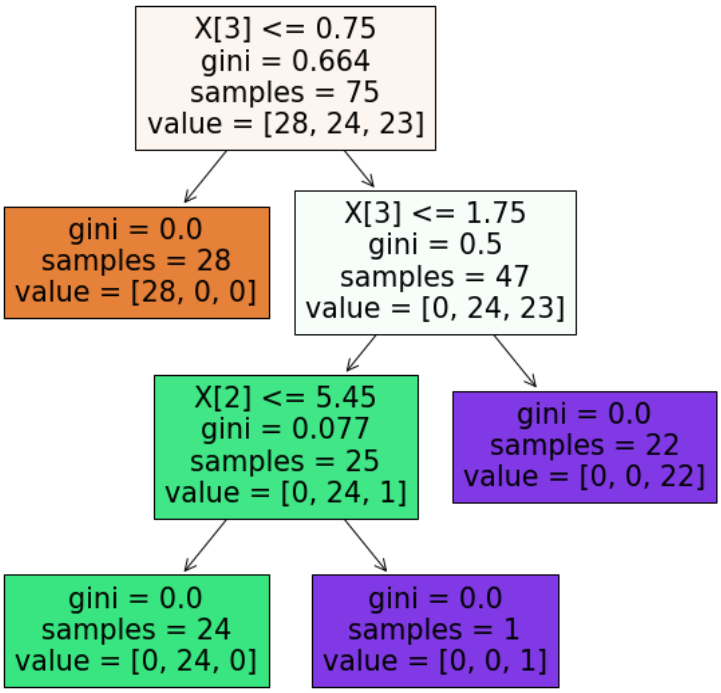
3. Вывести характеристики дерева

```
print(clf.get_n_leaves(), clf.get_depth())
```

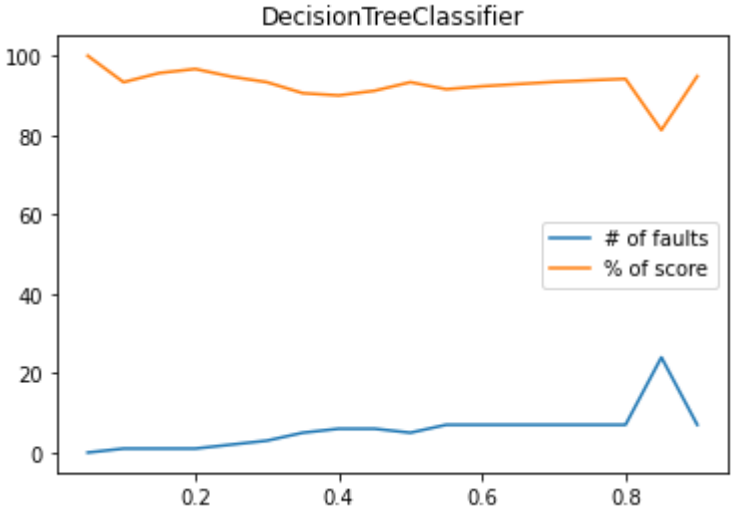
Кол-во листьев: 4

Глубина: 3

4. Изображение дерева

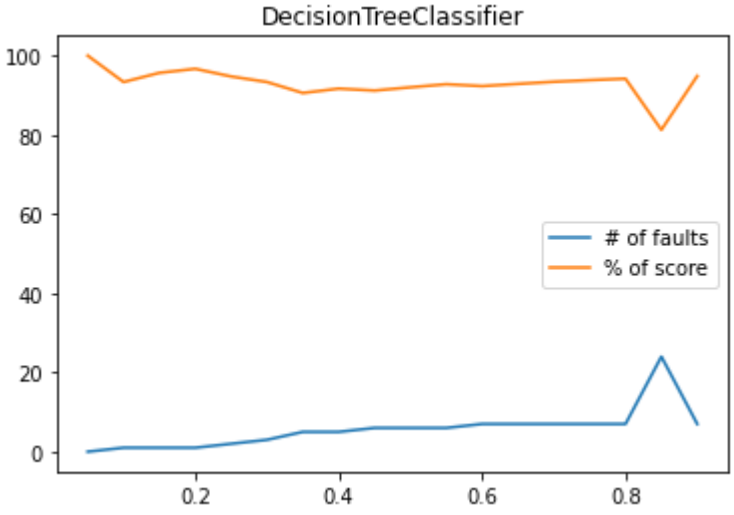


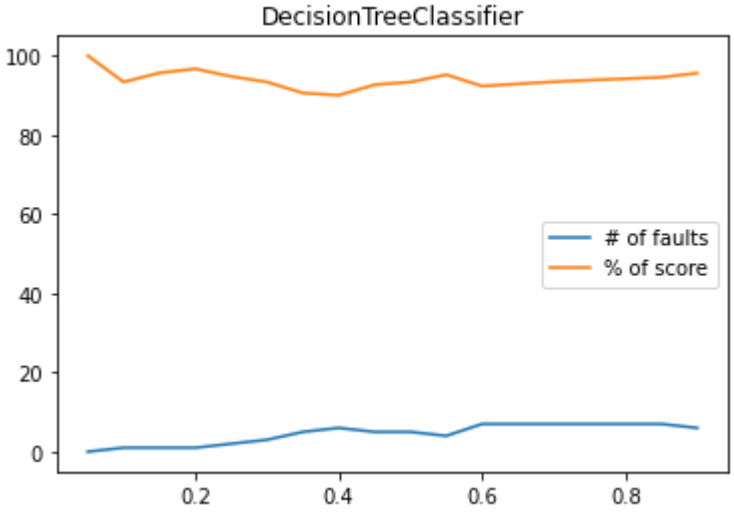
5. Построить график зависимости неправильно класс. наблюдений и точности классификации от размера выборки.



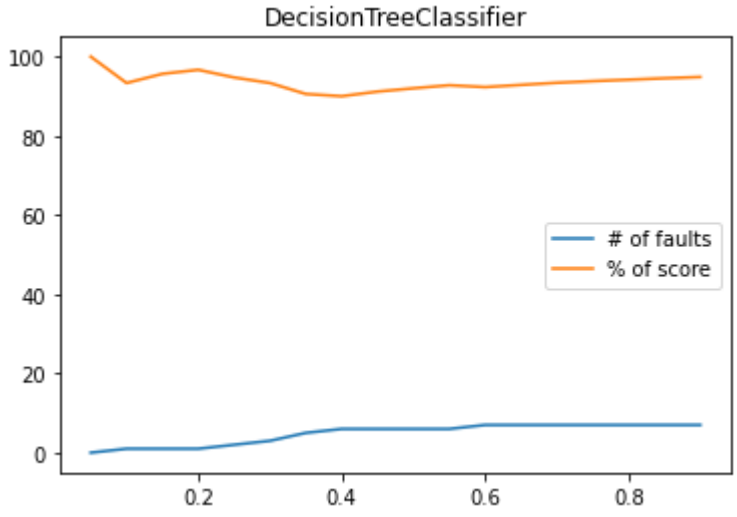
6. Исследовать работу алгоритма при различных параметрах

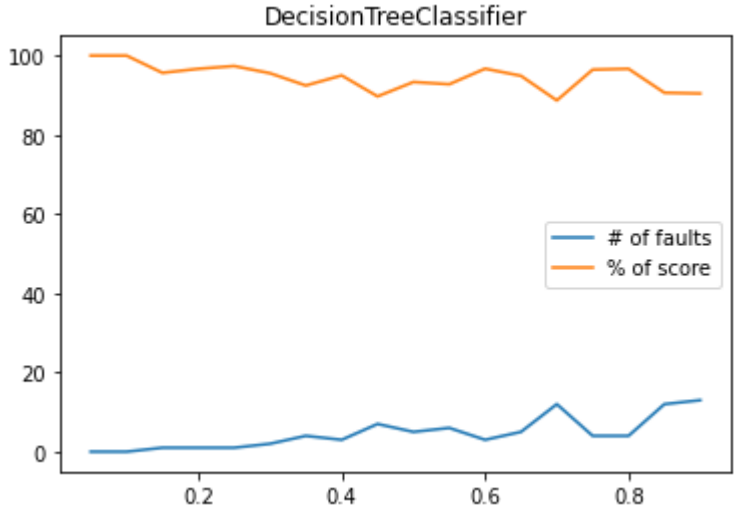
`criterion in ['gini', 'entropy']`
`- Функция измерения качества раскола. Поддерживаемые критерии: «Джини» для примеси Джини и «энтропия» для получения информации.`



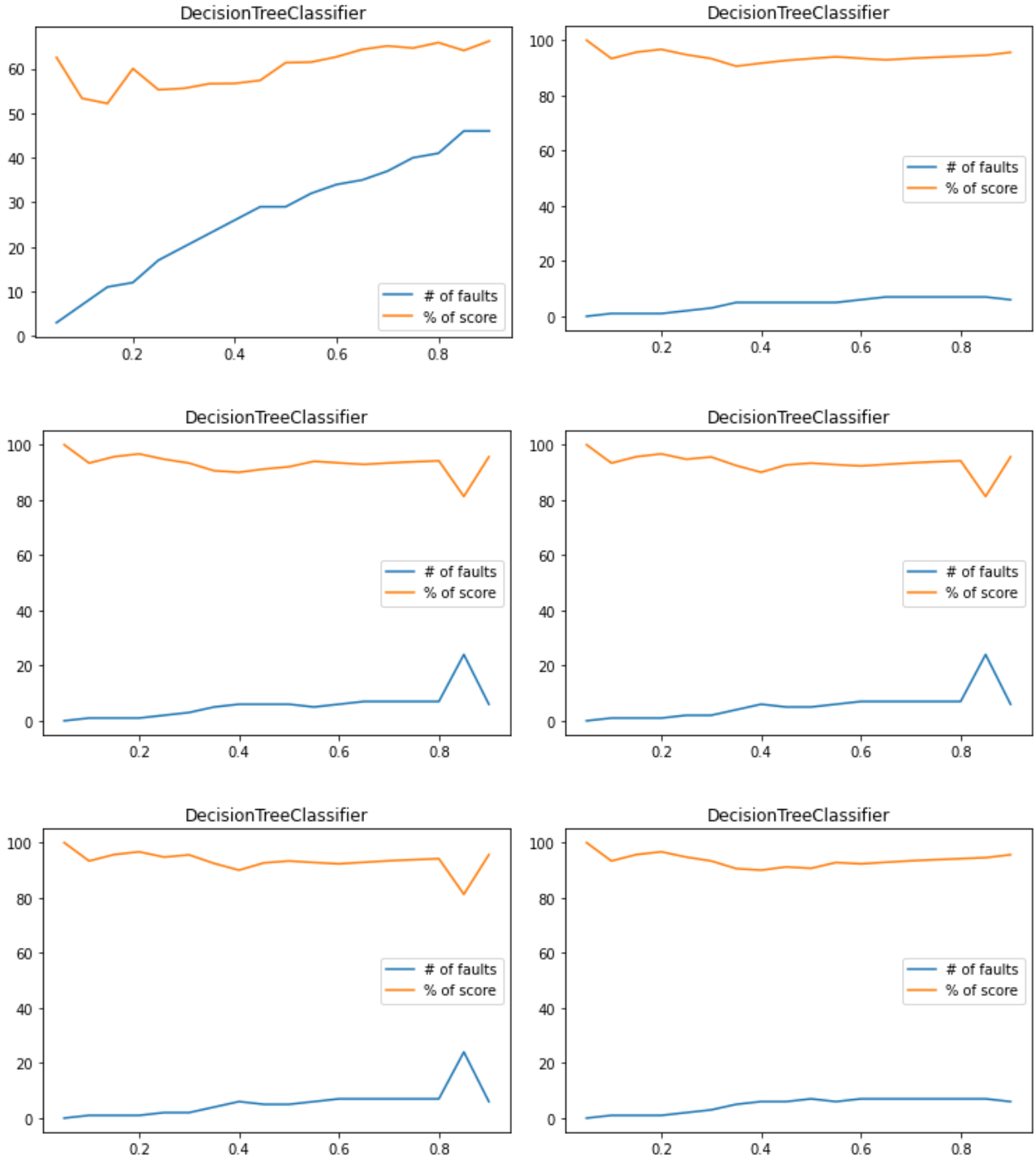


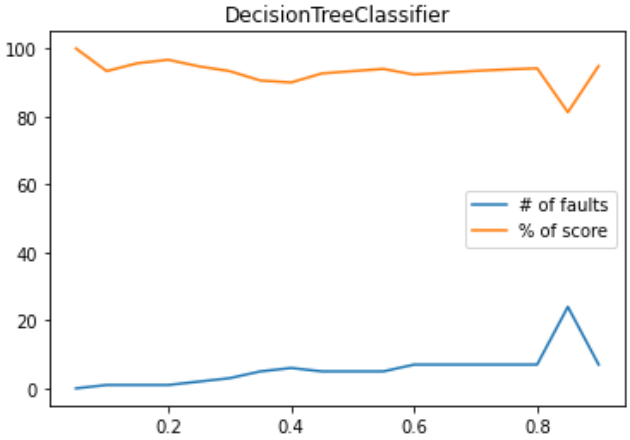
splitter in ['best', 'random']
- Стратегия, используемая для выбора разделения на каждом узле. Поддерживаемые стратегии являются «лучшими» для выбора наилучшего разделения и «случайными» для выбора лучшего случайного разделения.

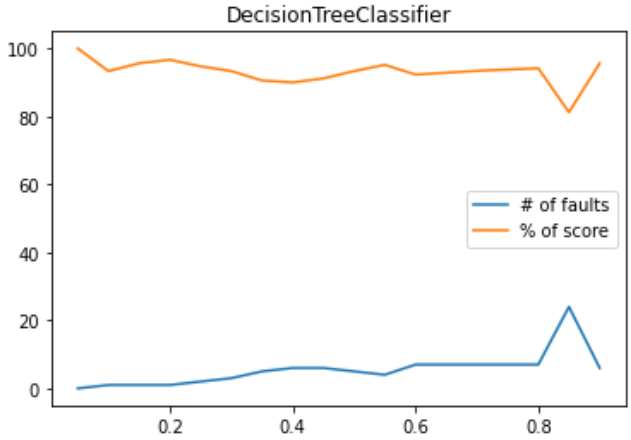


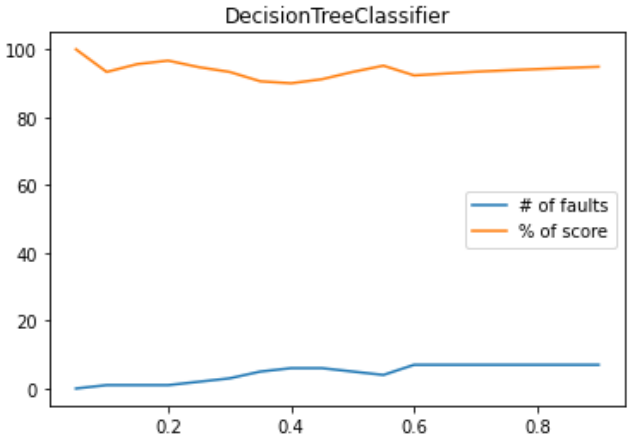


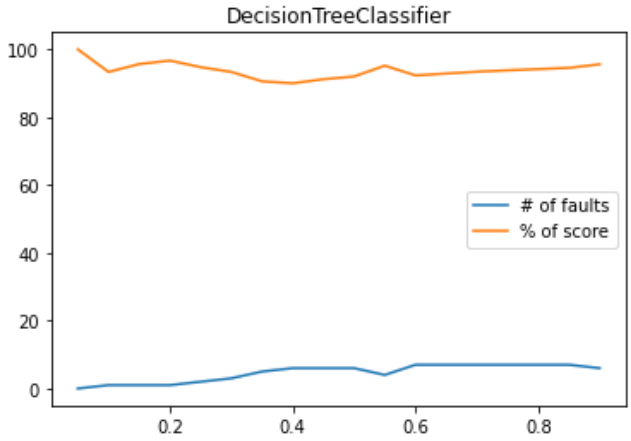
max_depth in range(1, 10)
-Максимальная глубина дерева. Если None, то узлы расширяются до тех пор, пока все листья не станут чистыми или пока все листья не будут содержать менее min_samples_split выборок.



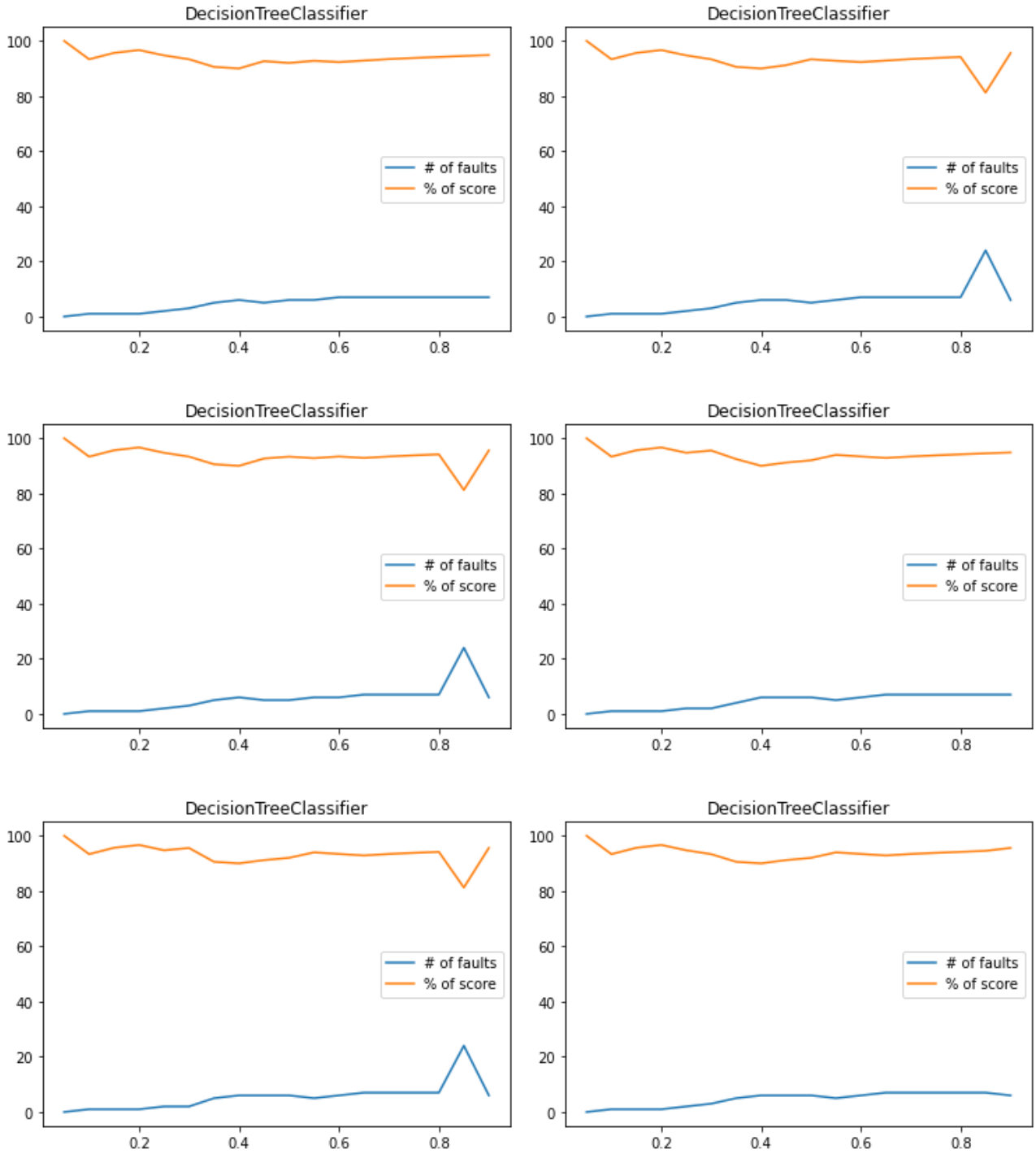

DecisionTreeClassifier
100
80
60
40
20
0
0.2
0.4
0.6
0.8
# of faults
% of score

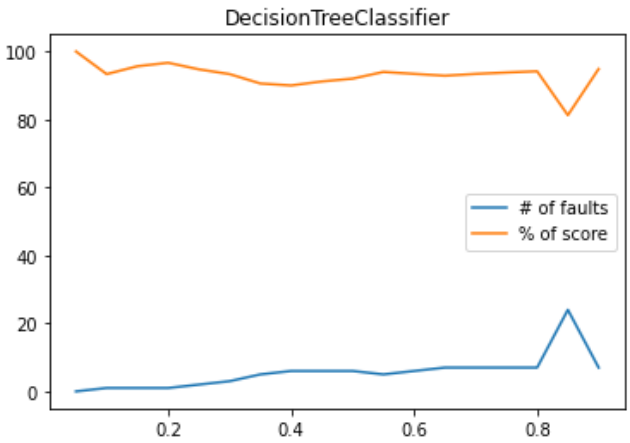

DecisionTreeClassifier
100
80
60
40
20
0
0.2
0.4
0.6
0.8
# of faults
% of score

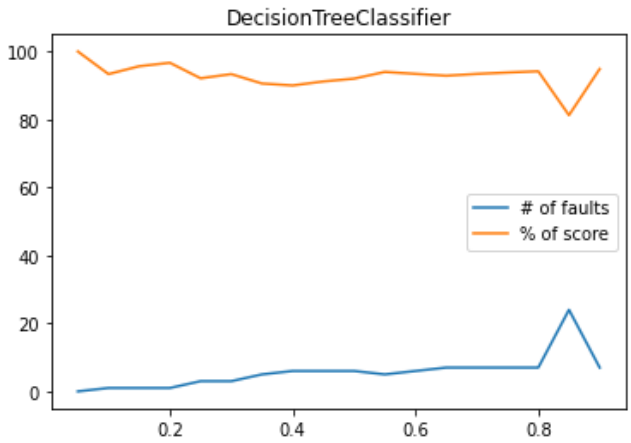

DecisionTreeClassifier
100
80
60
40
20
0
0.2
0.4
0.6
0.8
# of faults
% of score


DecisionTreeClassifier
100
80
60
40
20
0
0.2
0.4
0.6
0.8
# of faults
% of score

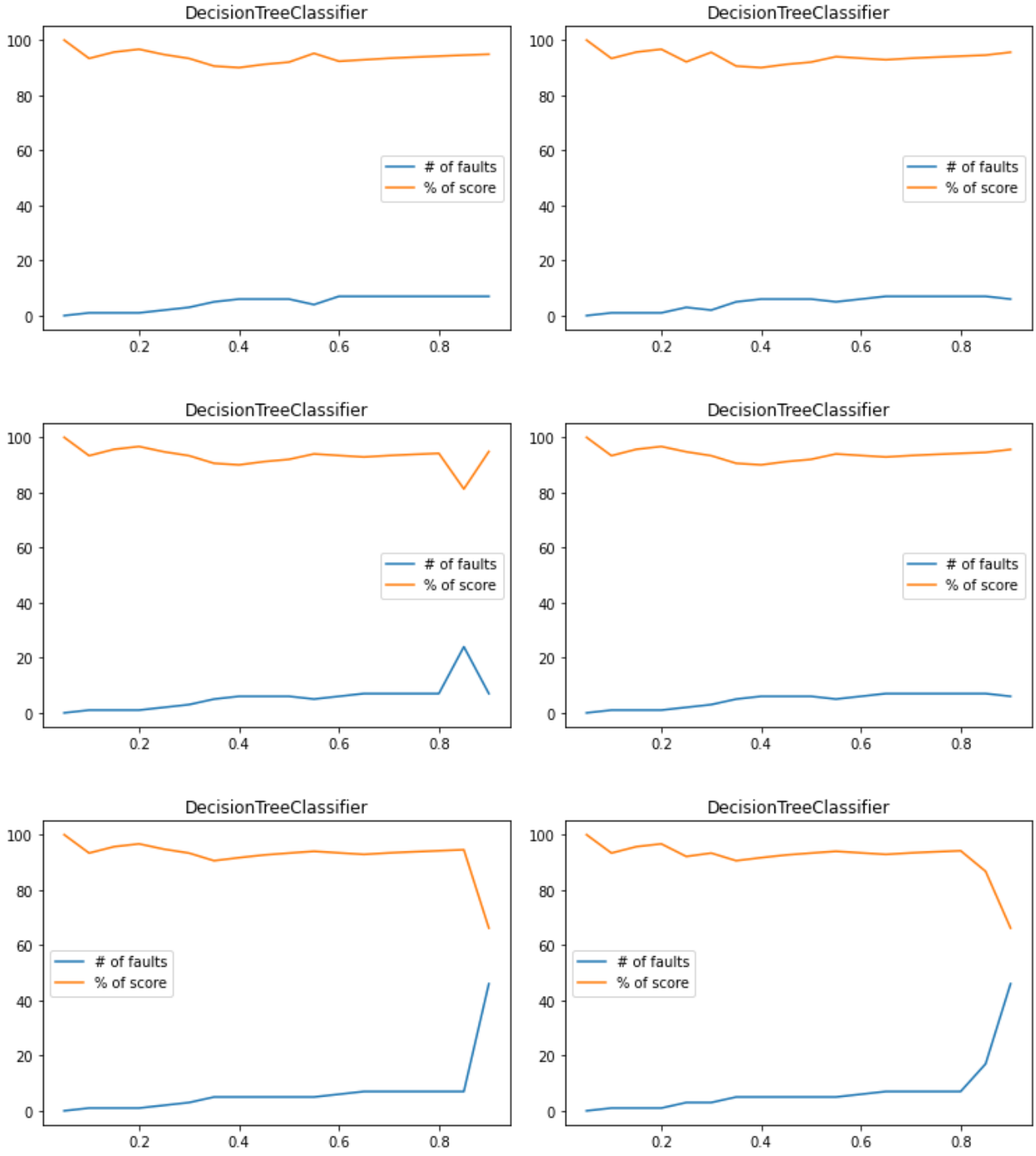
min_samples_split in range(2, 10)
- Минимальное количество выборок, необходимое для разделения внутреннего узла

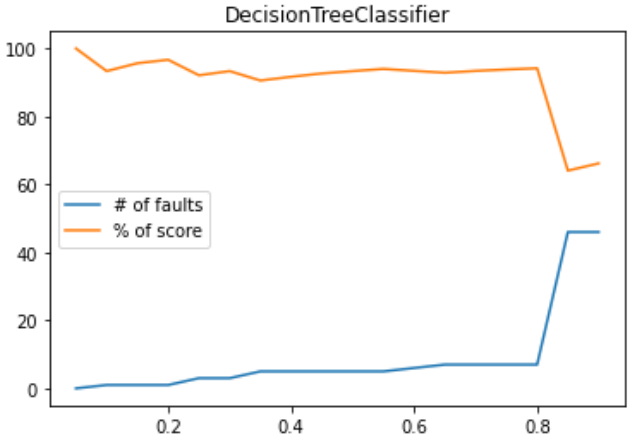


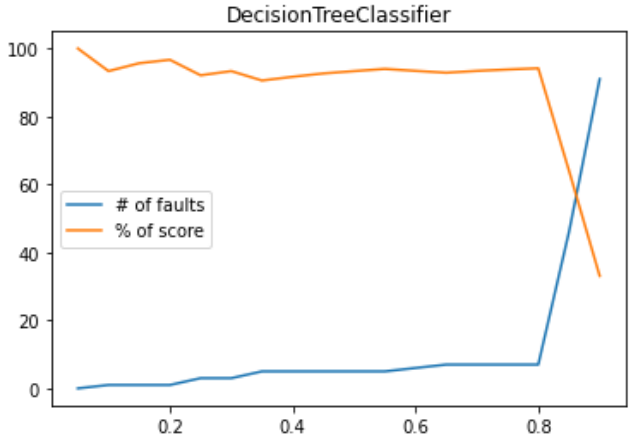

DecisionTreeClassifier
100
80
60
40
20
0
0.2
0.4
0.6
0.8
# of faults
% of score

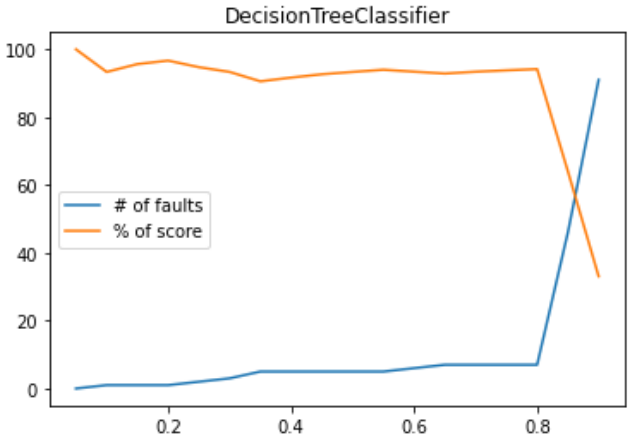

DecisionTreeClassifier
100
80
60
40
20
0
0.2
0.4
0.6
0.8
# of faults
% of score

min_samples_leaf in range(1, 10)
-Минимальное количество выборок, которое требуется для конечного узла. Точка разделения на любой глубине будет учитываться только в том случае, если она оставляет не менее min_samples_leaf обучающих выборок в каждой из левой и правой ветвей. Это может иметь эффект сглаживания модели, особенно при регрессии.




DecisionTreeClassifier
# of faults
% of score
100
80
60
40
20
0
0.2
0.4
0.6
0.8


DecisionTreeClassifier
# of faults
% of score
100
80
60
40
20
0
0.2
0.4
0.6
0.8


DecisionTreeClassifier
# of faults
% of score
100
80
60
40
20
0
0.2
0.4
0.6
0.8