**Задание 1**

**Изображение выглядит как текст

Автоматически созданное описание**

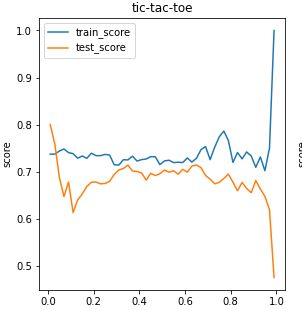
**Решение**

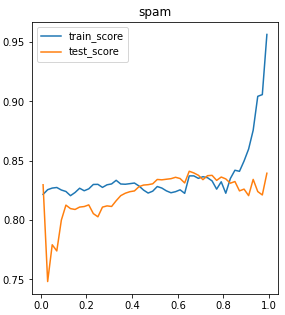
Для начала постоим Байесовский классификатор.

Закодируем для датасета tic\_tac\_toe символ ‘O’ как 0, символ ‘X’ – 1, символ ‘b’ (пустое поле) – 2



Построим графики зависимостей:





**Вывод:**

По графикам можно сделать вывод, что точность классификации в обоих датасетах будем меняться при изменении соотношения обучающей и тестовой выборок.

**Задание 2**

**Изображение выглядит как текст

Автоматически созданное описание**

**Решение:**

****

Отсюда получаем:

*Класс -1:*

X1 = 13

X2 = 11

D = 3

N = 40

*Класс 1:*

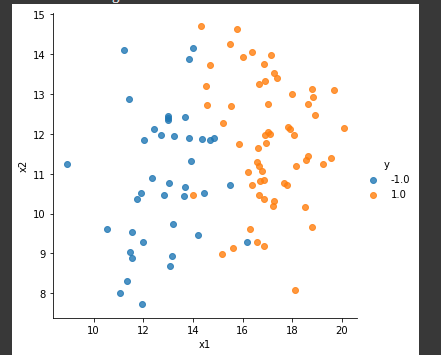
X1 = 17

X2 = 12

D = 2

N = 60

Построим диаграмму:



Строим Байесовский классификатор. Разобьем данные на test и train в соотношении 0.25/0.75:

Изображение выглядит как текст, снимок экрана, экран, несколько

Автоматически созданное описание

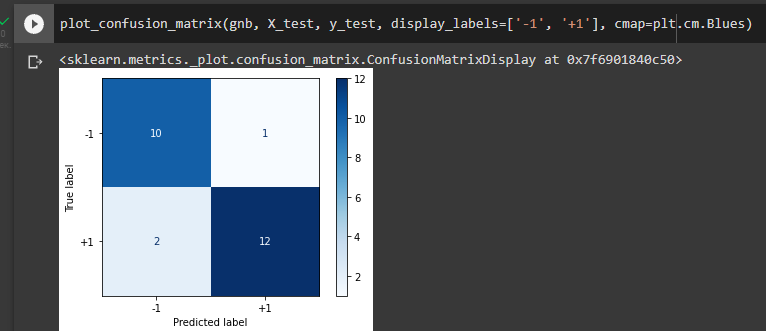
Оценим качество классификации при помощи различных методов:

1. Точность

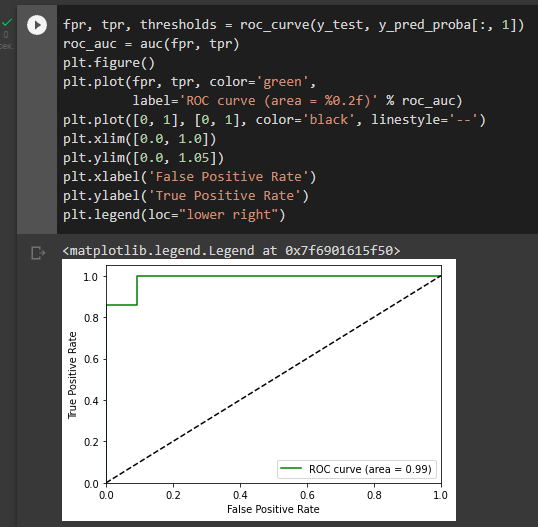
Изображение выглядит как текст

Автоматически созданное описание

1. Матрицы ошибок:



1. Roc-кривая



1. PR-кривая:

Изображение выглядит как текст

Автоматически созданное описание

**Вывод:**

Я не могу утверждать, что мы получили хороший классификатор. Такая точность (=0.88) может быть недостаточна для решения более углубленных задач.

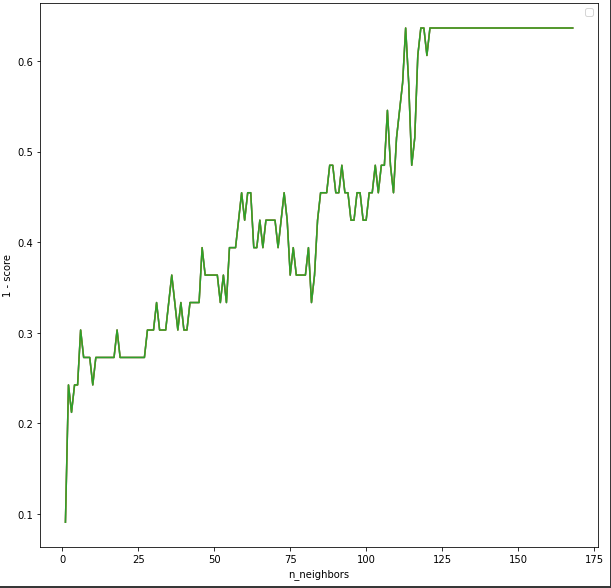
**Задача 3**

**Изображение выглядит как текст

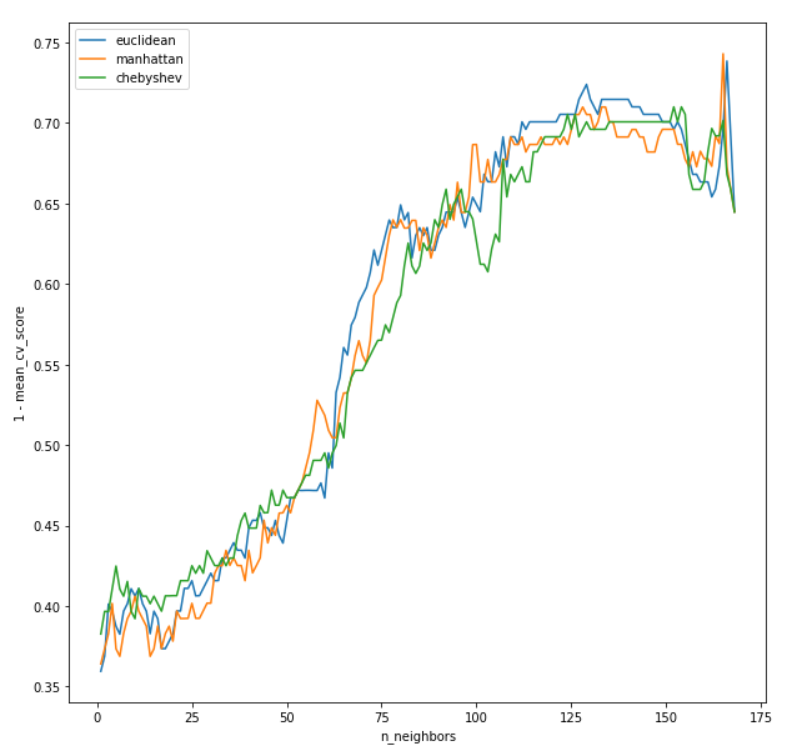
Автоматически созданное описание**

**Решение:**

1. Построим график зависимости ошибки классификации от количества ближайших соседей



1. Построим графики зависимости ошибки классификации от количества ближайших соседей, при разных метриках расстояния



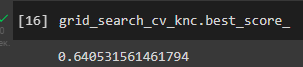
Определим лучшую метрику:

Изображение выглядит как текст, снимок экрана, внутренний

Автоматически созданное описание

Как можем заметить, лучшая метрика – ‘euclidean’

Получим ее точность:





Изображение выглядит как текст, экран, закрыть

Автоматически созданное описание

Данный экземпляр относится к 5 типу

**Вывод:**

Лучшая метрика – ‘euclidean’. С помощью нее мы определили, что искомый экземпляр стекла относится к 5 типу.

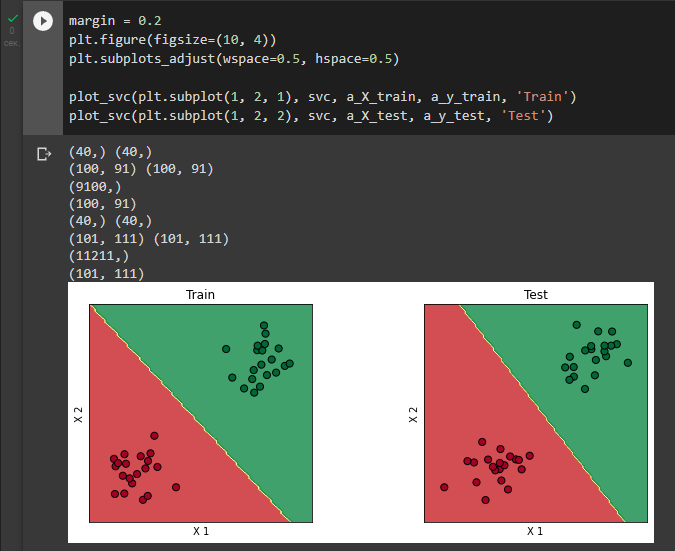
**Задание 4**

**Изображение выглядит как текст

Автоматически созданное описание**

**Решение:**

1. Построим модель опорных векторов с линейным ядром. Визуализируем разбиение пространства признаков на области с помощью полученной модели:

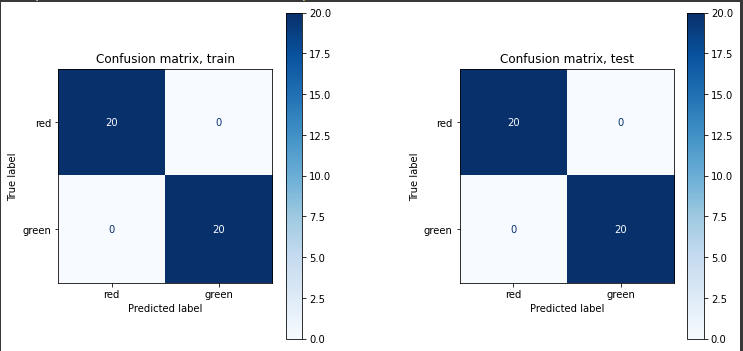


Выведем количество полученных опорных векторов:

Изображение выглядит как текст

Автоматически созданное описание

Выведем матрицу ошибок классификации на обучающей и тестовой выборках:



1. Построим модель метода опорных векторов с линейным ядром. Добьемся нулевой ошибки на обучающей и тестовой выборках путем изменения штрафного параметра:

Изображение выглядит как текст

Автоматически созданное описание

Найдем оптимальное значение:

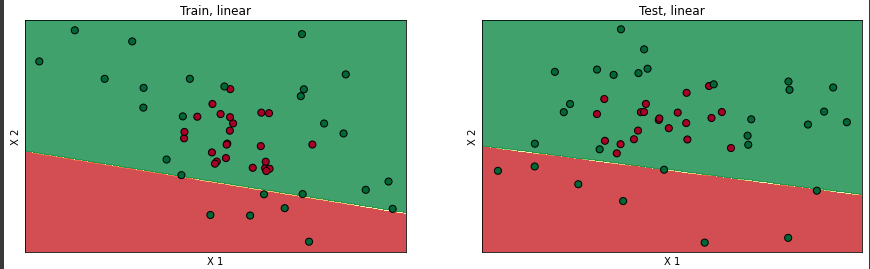
Изображение выглядит как текст

Автоматически созданное описание

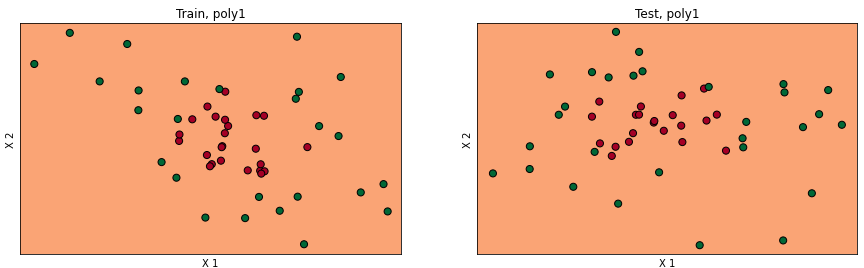
Не всегда нужно добиваться минимизации ошибки на обучающей выборке, потому что можно столкнуться с проблемой переобучения. На тестовых выборках модель будет плохо себя проявлять.

1. Построим модель метода опорных векторов, используя различные ядра:

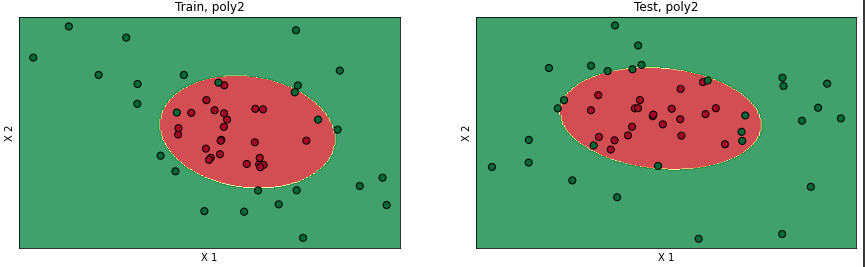
Линейное ядро:



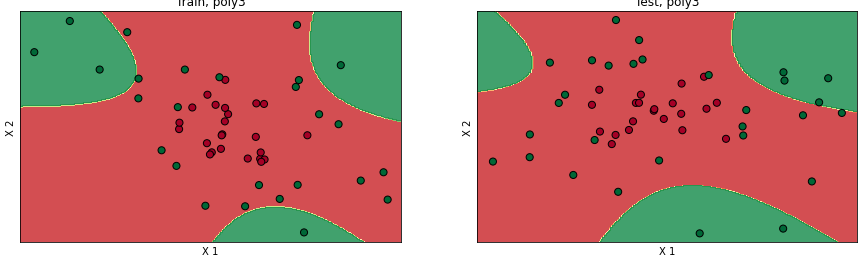
Полиномиальное ядро, степень 1:



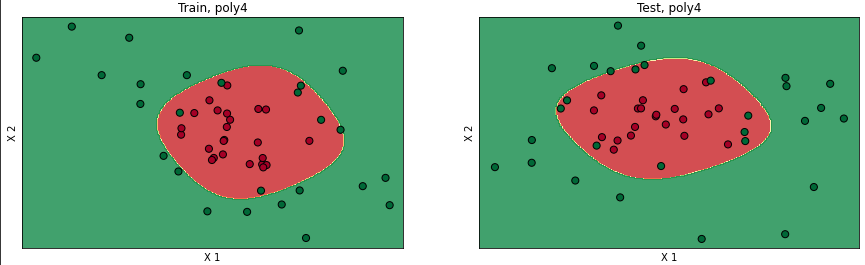
Полиномиальное ядро, степень 2:



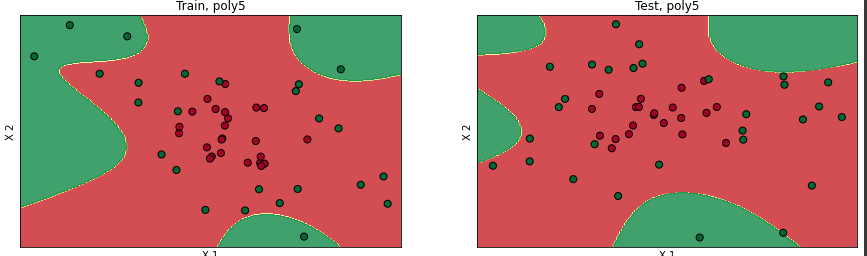
Полиномиальное ядро, степень 3:



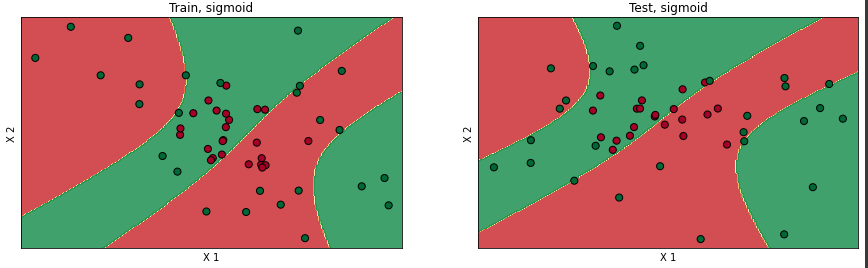
Полиномиальное ядро, степень 4:



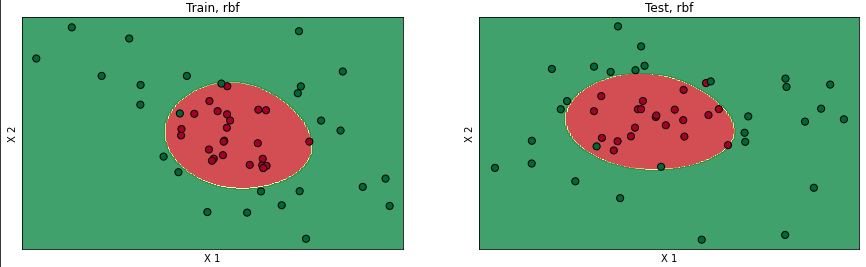
Полиномиальное ядро, степень 5:



Сигмоидальная функция:



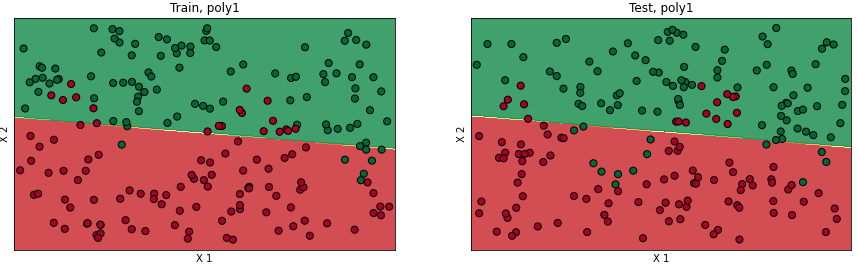
Гауссово:



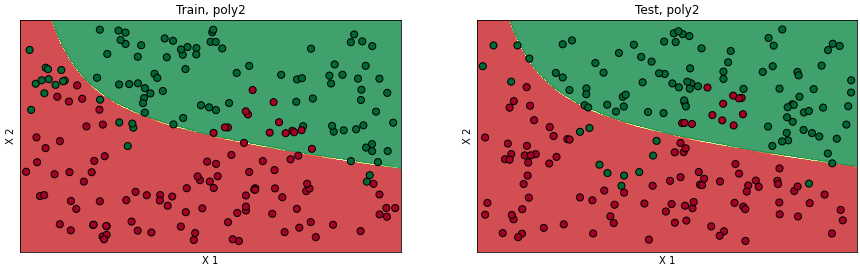
Вывод: Из графиков видно, что лучше всего себя проявили: полиномиальные ядра 2 и 4, Гауссово ядро.

1. Построим модель метода опорных векторов с различными ядрами. Визуализируем разбиение пространства признаков на области для каждой модели.

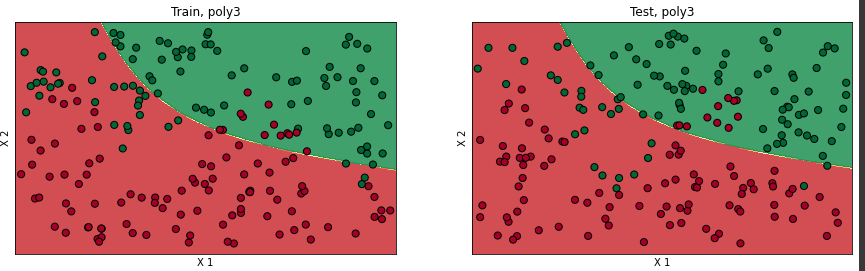
Полиномиальное ядро, степень 1:



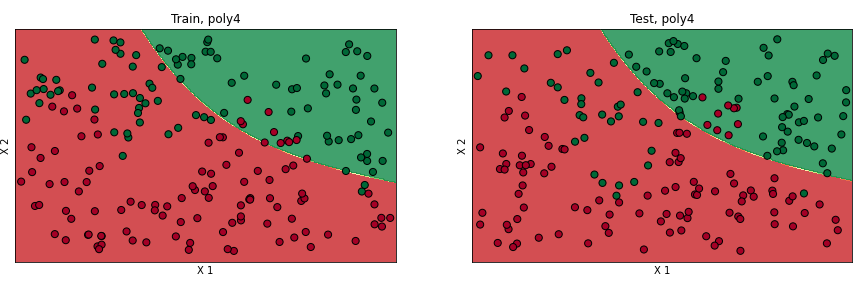
Полиномиальное ядро, степень 2:



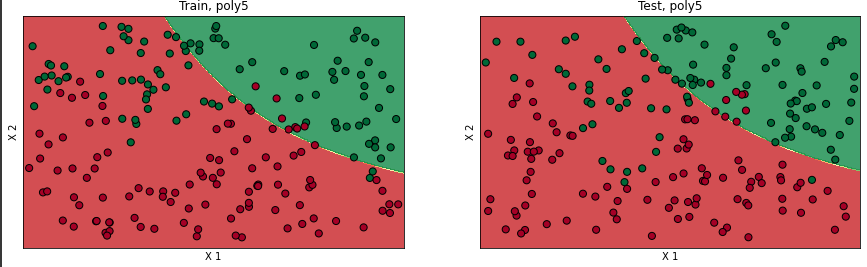
Полиномиальное ядро, степень 3:



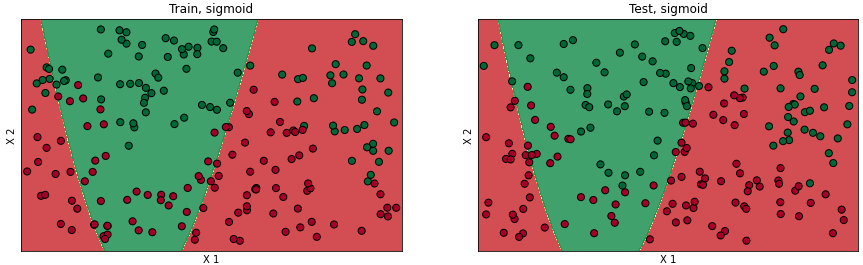
Полиномиальное ядро, степень 4:



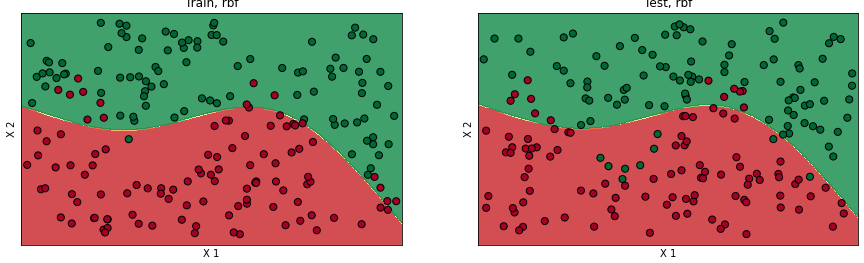
Полиномиальное ядро, степень 5:



Сигмоидальная функция



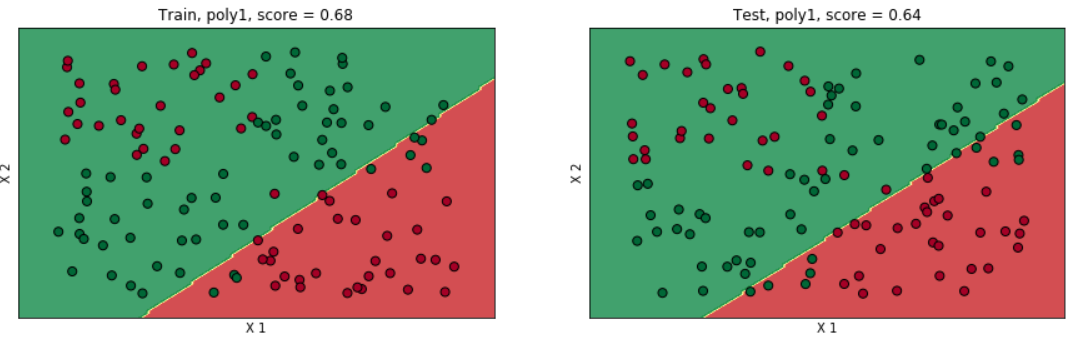
Гауссово:



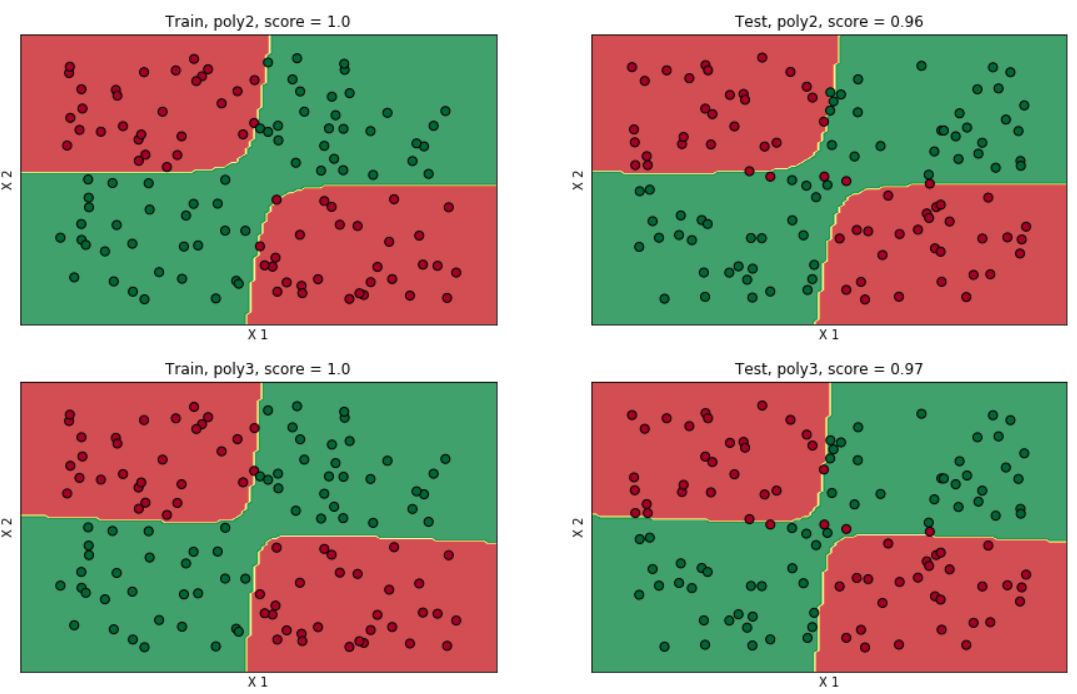
Вывод: Из графиков видим, что лучше всего себя проявили: полиномиальные ядра 1 и 2, Гауссово ядро.

1. Построим модель метода опорных векторов с различными ядрами. Изменяя значение параметра ядра, продемонстрируем эффект переобучения:

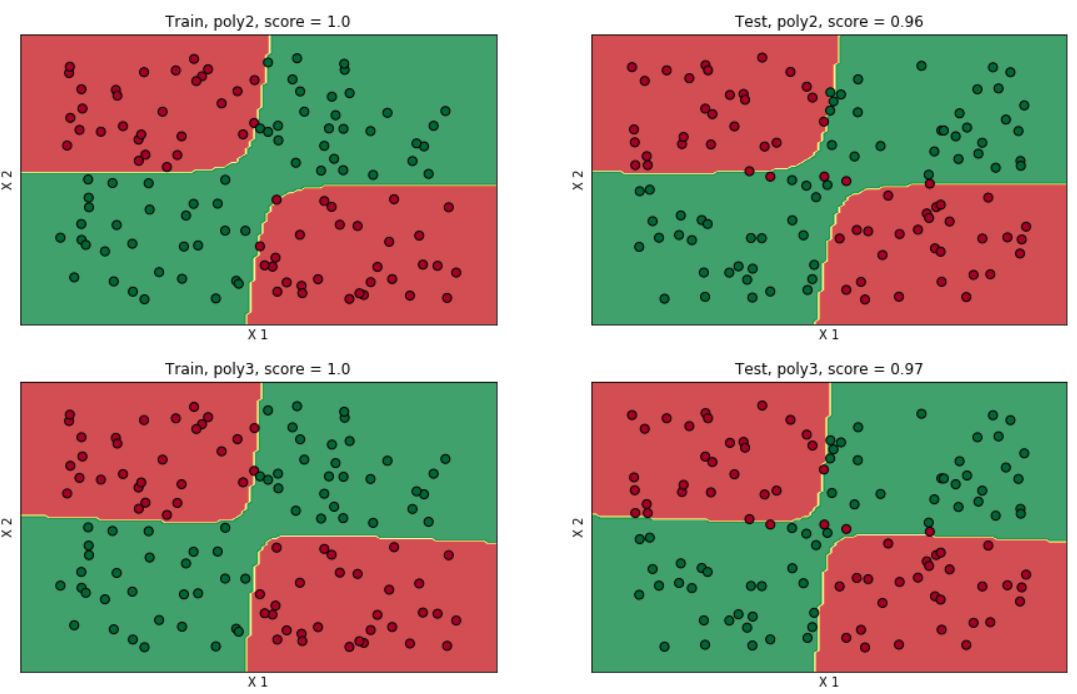
Переобученная модель, полиномиальное ядро, степень 1:



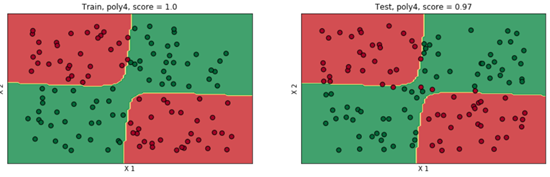
Переобученная модель, полиномиальное ядро, степень 2:



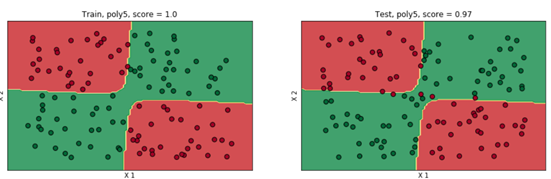
Переобученная модель, полиномиальное ядро, степень 3:



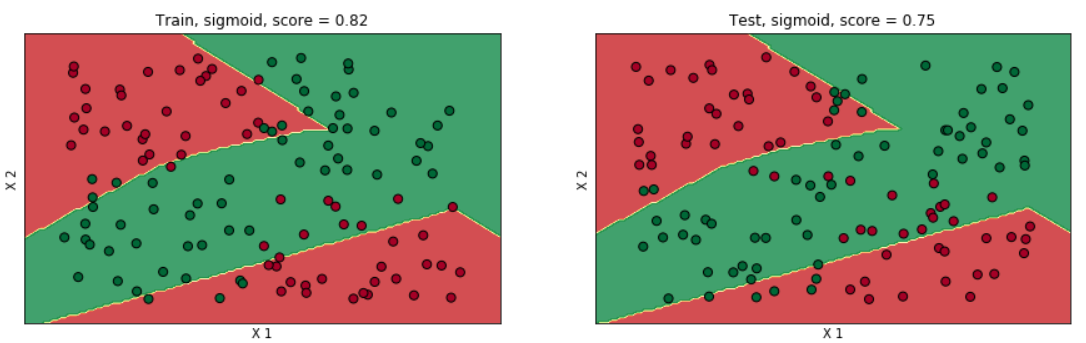
Переобученная модель, полиномиальное ядро, степень 4:



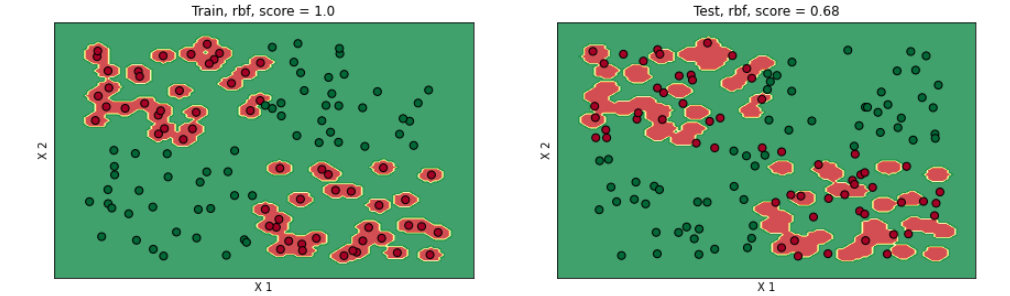
Переобученная модель, полиномиальное ядро, степень 5:



Переобученная модель, сигмоидальная функция:



Переобученная модель, Гауссово:



Вывод:

Мы показали негативный эффект переобучения модели. Лучше всего он виден в модели с Гауссовым ядром.

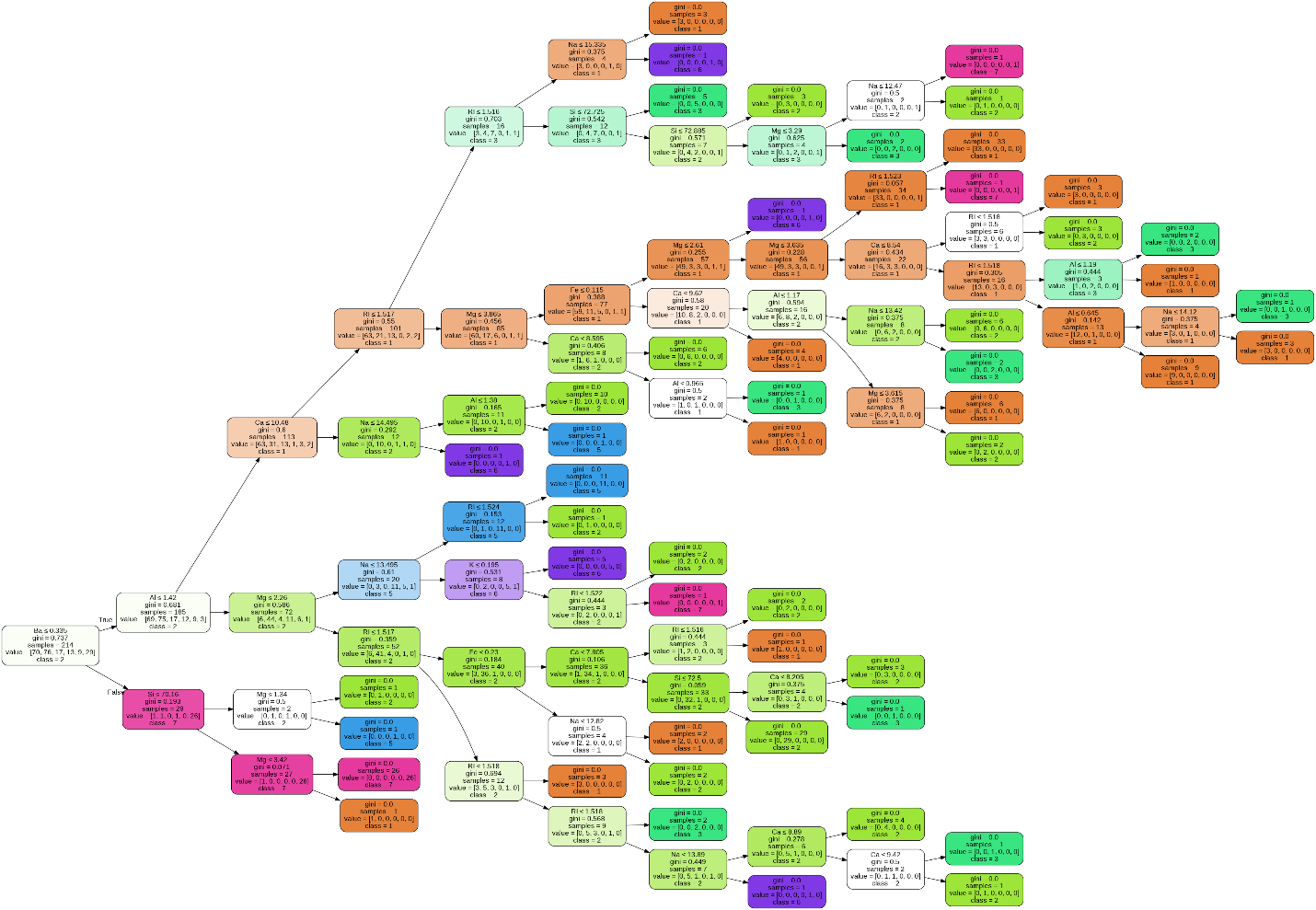
**Задание 5**

**Изображение выглядит как текст

Автоматически созданное описание**

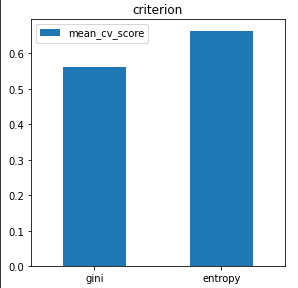
**Решение:**

Дерево решения:

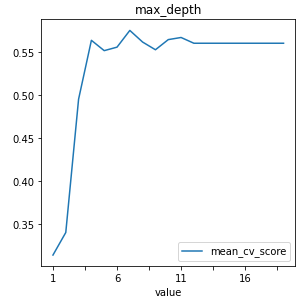


Данное дерево является избыточным, это говорит о наличии эффекта переобучения.

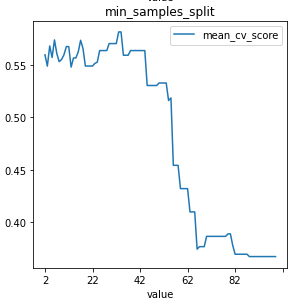
Исследуем зависимость точности классификации от критерия расщепления:



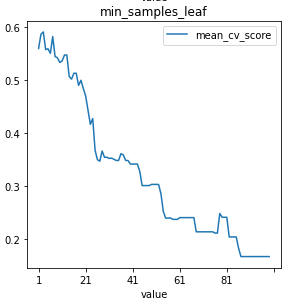
Исследуем зависимость точности классификации от максимальной глубины дерева:



Исследуем зависимость точности от минимального разделения выборки:



Исследуем зависимость точности классификации от минимального количества samples в листе:

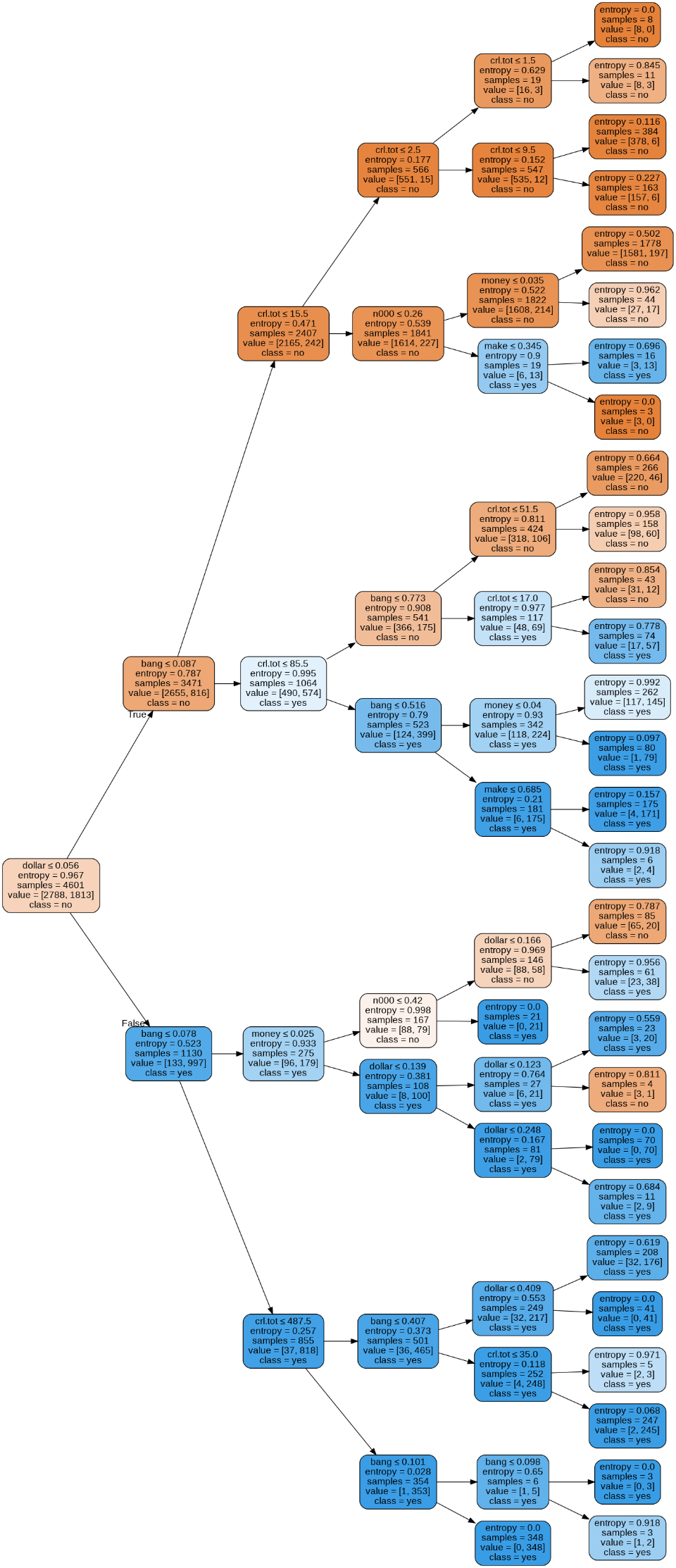


b. Построим дерево классификации для параметра yesno. Нам подобрались параметры (использовал модуль RandomizedSearchCV, для подбора параметров):

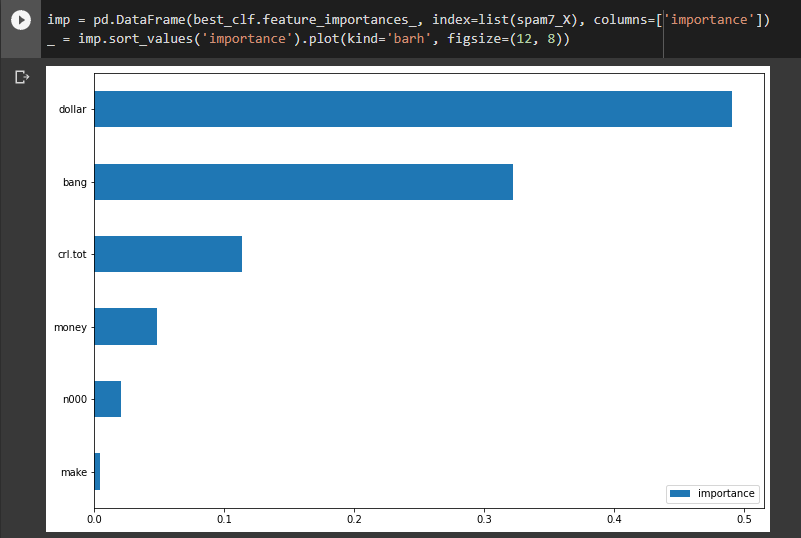
Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

Теперь построим дерево:



Определим наиболее влияющие признаки:



Качество классификации:



Вывод:

Качество классификации достаточно высокое. Самыми влияющими признаки: dollar и bang.

**Задание 6**

**Изображение выглядит как текст

Автоматически созданное описание**

**Решение:**

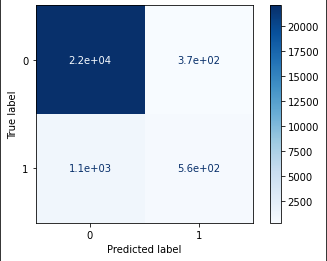
I вариант – решающее дерево:

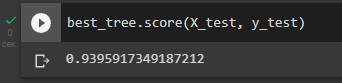
Найдем лучший классификатор:

Изображение выглядит как текст

Автоматически созданное описание

Матрица ошибок и точность:

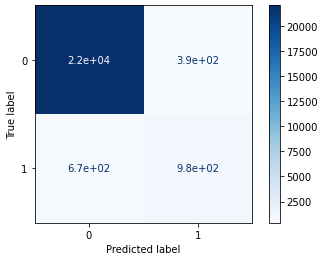


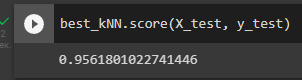


II вариант – k-ближайших соседей:

Изображение выглядит как текст

Автоматически созданное описание





Вывод:

По матрице ошибок, можем заметить, что у метода k-ближайших соседей показатель предсказания выше (5.6e+02<9.8e+02). Но также точность у метода k-ближайших соседей выше (0.93<0.95).

Видим, что принятие решений системой, основанной на методе k-ближайших соседей, лучше.