Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

Отчет

по лабораторной работе №1

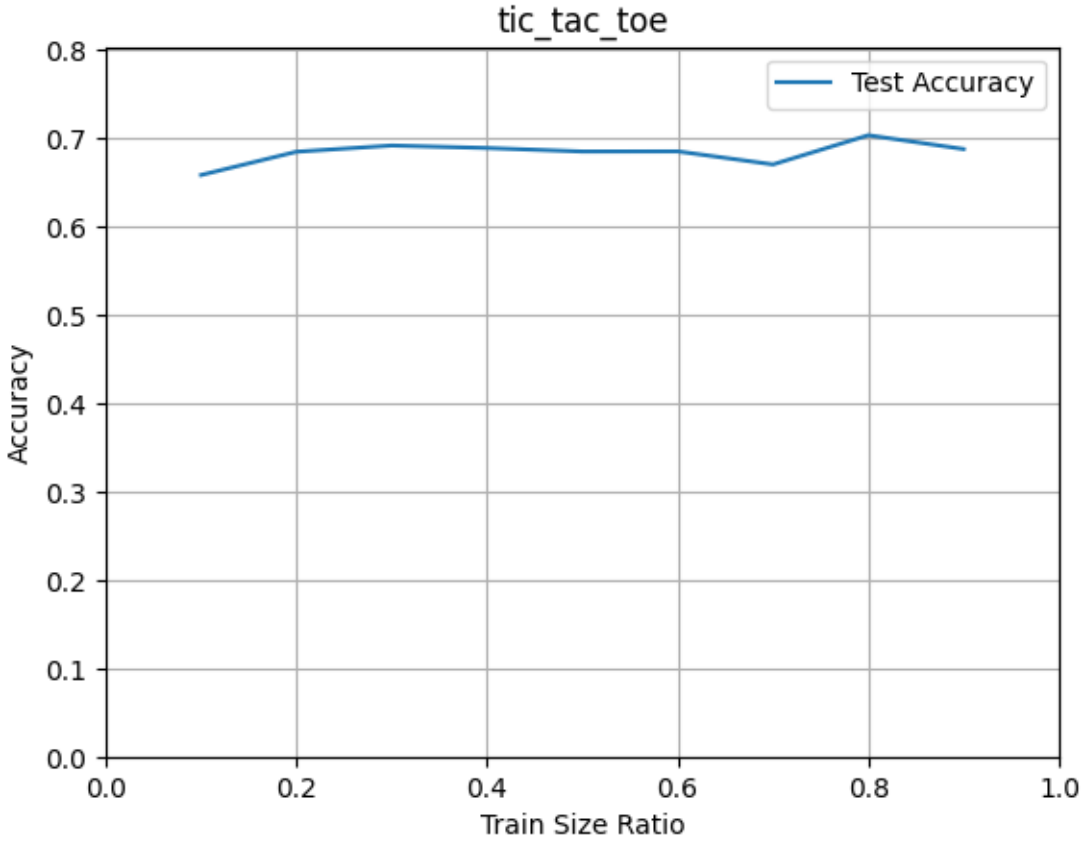
по дисциплине «Введение в машинное обучение»

Выполнила
Студентка гр. 5130904/10101 Никифорова Е. А.

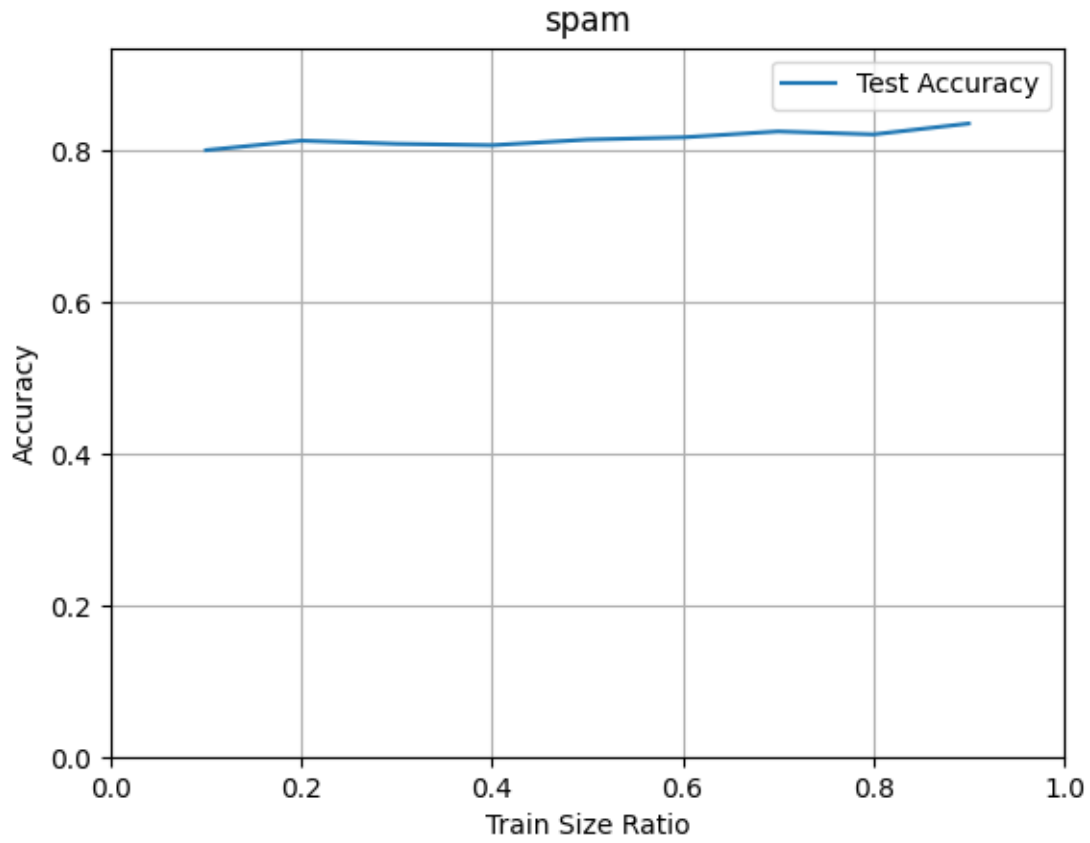
Руководитель Селин И. А

Санкт-Петербург
2024

1. Исследуйте, как объем обучающей выборки и количество тестовых данных, влияет на точность классификации в датасетах про крестики-нолики (tic_tac_toe.txt) и о спаме e-mail сообщений (spam.csv) с помощью наивного Байесовского классификатора. Постройте графики зависимостей точности на обучающей и тестовой выборках в зависимости от их соотношения.



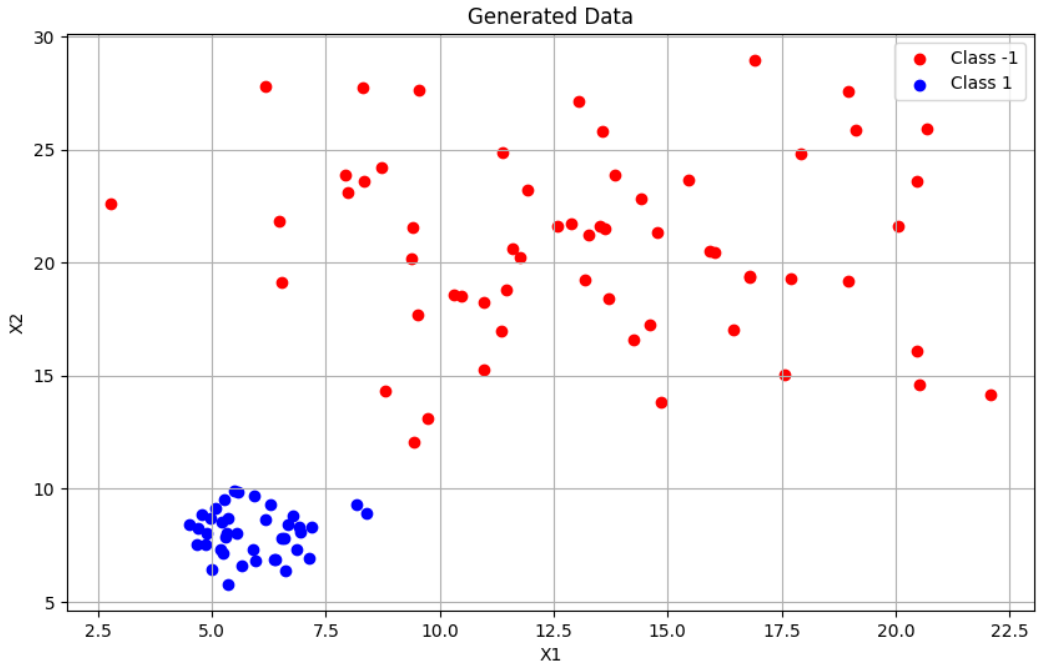
Исходя из полученного графика, можно сделать вывод, что при увеличении объема обучающей выборки точность классификации остается примерно одинаковой, за исключением небольшого колебания при объеме 0.8-0.9.



Исходя из полученного графика, можно сделать вывод, что при увеличении объема обучающей выборки точность классификации повышается.

2. Сгенерируйте 100 точек с двумя признаками X1 и X2 в соответствии с нормальным распределением так, что одна и вторая часть точек (класс -1 и класс 1) имеют параметры: мат. ожидание X1, мат. ожидание X2, среднеквадратические отклонения для обеих переменных, соответствующие вашему варианту (указан в таблице). Построить диаграммы, иллюстрирующие данные. Построить Байесовский классификатор и оценить качество классификации с помощью различных методов (точность, матрица ошибок, ROC и PR-кривые). Является ли построенный классификатор «хорошим»?

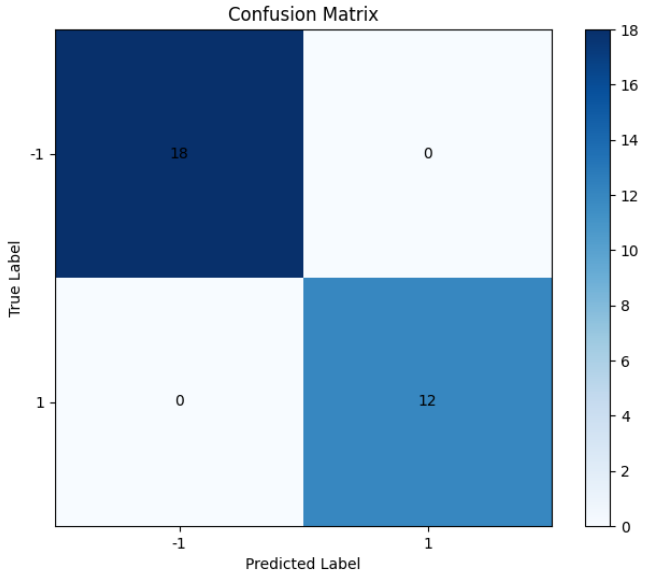
Матем. ожид. X1 (класс -1)	Матем. ожид. X2 (класс -1)	СКО (класс -1)	Матем. ожид. X1 (класс 1)	Матем. ожид. X2 (класс 1)	СКО (класс 1)	Количество элементов (класс -1)	Количество элементов (класс 1)
13	20	4	6	8	1	60	40



Точность = 1.0

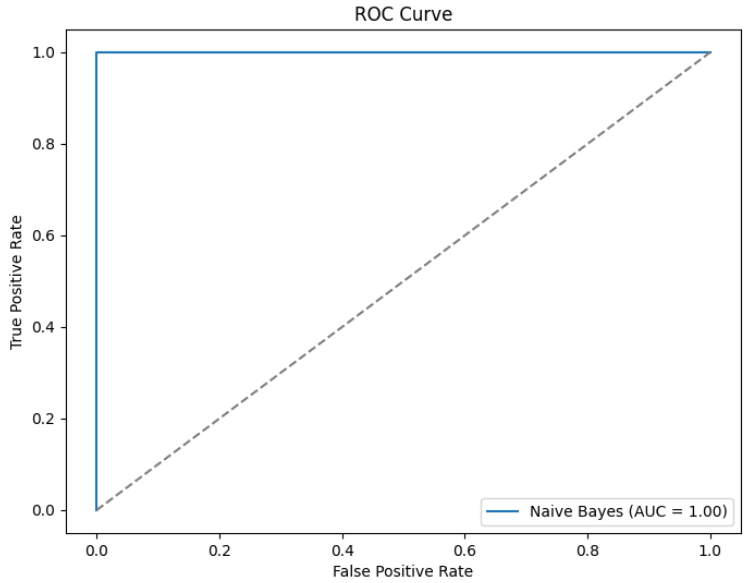
Матрица ошибок:

Все точки распознаны верно.



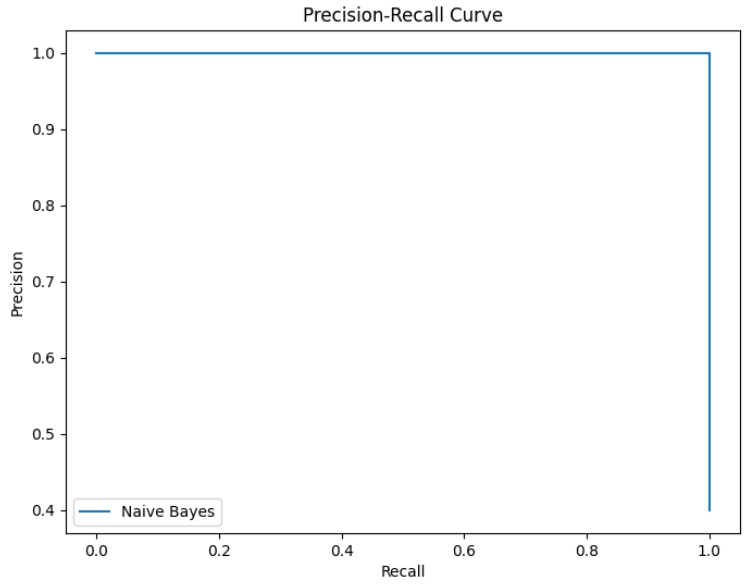
ROC-кривая

Идеальная ROC-кривая, проходит через верхний левый угол графика, что означает, что классификатор имеет высокую чувствительность и низкую специфичность при всех значениях порога



PR-кривая

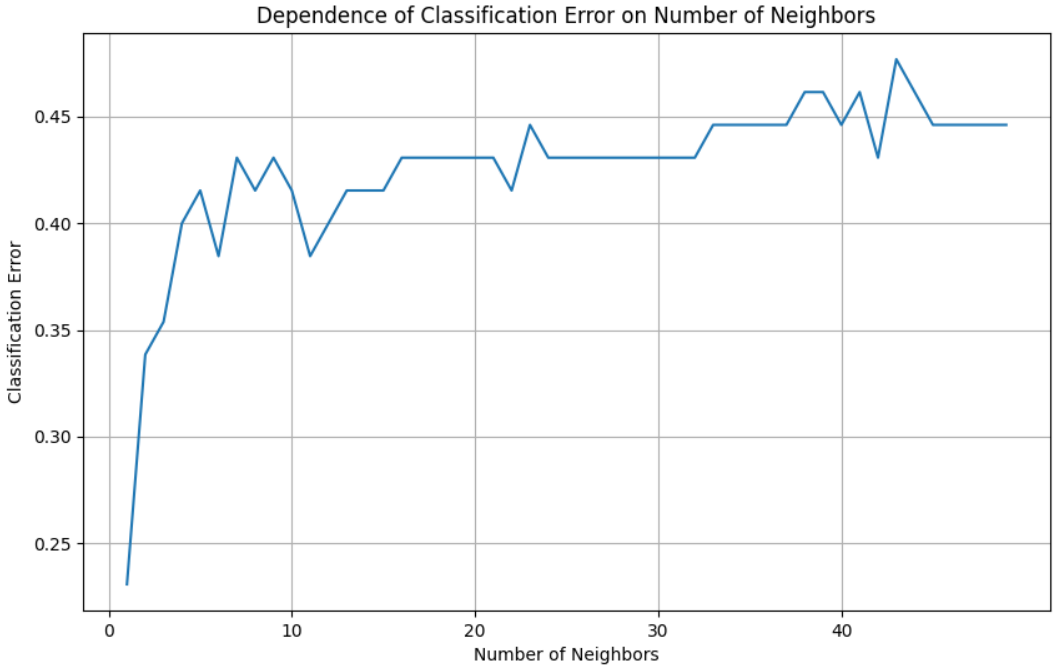
Идеальная PR-кривая, проходит через верхний правый угол графика, что означает, что классификатор имеет высокую точность и полноту при всех значениях порога.



Из-за того, что в данных существует разделение по двум классам с относительно большим различием в средних значениях и дисперсиях, наивный байесовский классификатор может эффективно разделить эти классы. В данном случае классы разделяются четко, что приводит к высокой точности классификации.

3. Постройте классификатор на основе метода k ближайших соседей для обучающего множества Glass (glass.csv). Посмотрите заголовки признаков и классов. Перед построением классификатора необходимо также удалить первый признак Id number, который не несет никакой информационной нагрузки.

a. Постройте графики зависимости ошибки классификации от количества ближайших соседей.



Исходя из графика зависимости ошибки от количества ближайших соседей можно сделать вывод, что при увеличении количества соседей увеличивается ошибка.

b. Определите подходящие метрики расстояния и исследуйте, как тип метрики расстояния влияет на точность классификации.

Для 5 соседей посчитаем метрики расстояния и соответствующие точности:

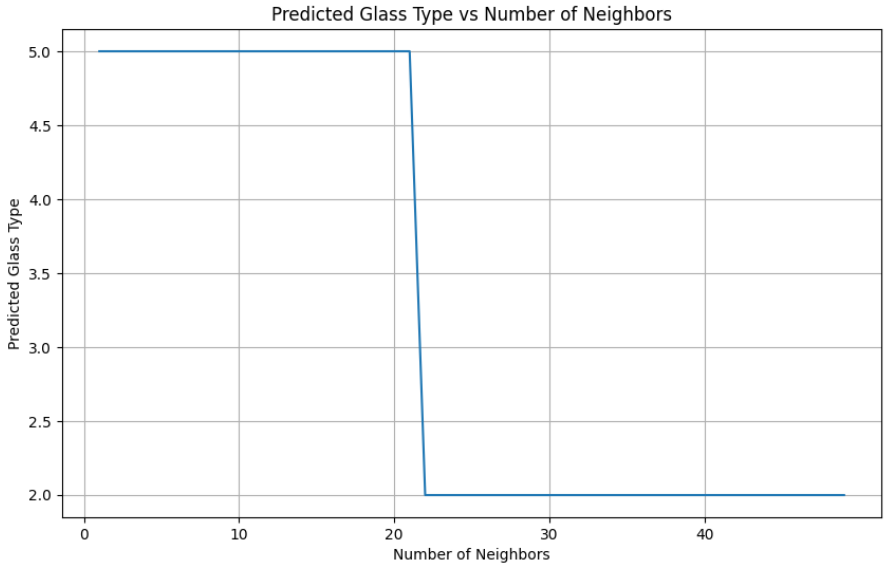
Accuracy with euclidean distance metric: 0.5846153846153846

Accuracy with manhattan distance metric: 0.6307692307692307

Accuracy with chebyshev distance metric: 0.5846153846153846

Наибольшая точность получается при использовании Манхэттенской метрики, при использовании метрики Чебышева и Евклидовой метрики точность получается хуже на ~0.05

c. Определите, к какому типу стекла относится экземпляр с характеристиками:
RI =1.516 Na =11.7 Mg =1.01 Al =1.19 Si =72.59 K=0.43 Ca =11.44 Ba =0.02 Fe =0.1



Предполагаемый тип стекла зависит от выбора количества соседей, при k < 21 тип – 5, иначе – 2. Так как при увеличении количества соседей уменьшается точность, данный экземпляр стекла относится к 5-му типу.

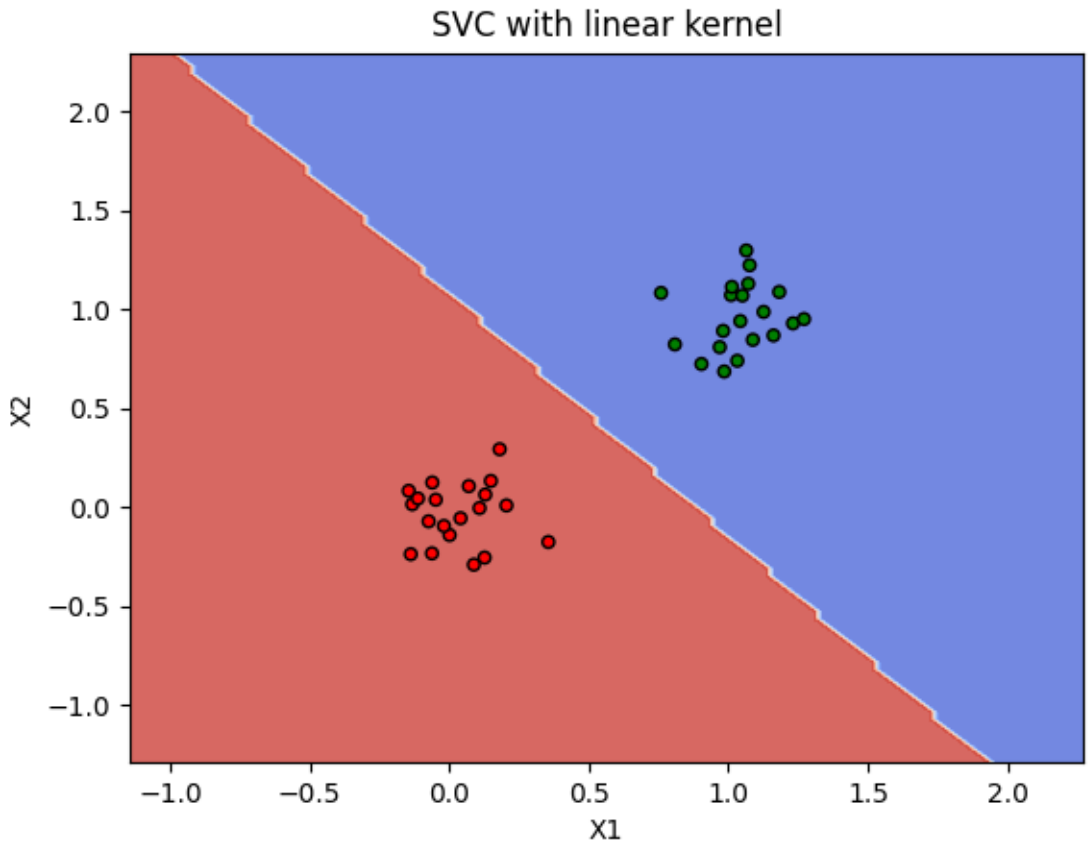
4. Постройте классификаторы на основе метода опорных векторов для наборов данных из файлов svmdataN.txt и svmdataNtest.txt, где N – индекс задания:

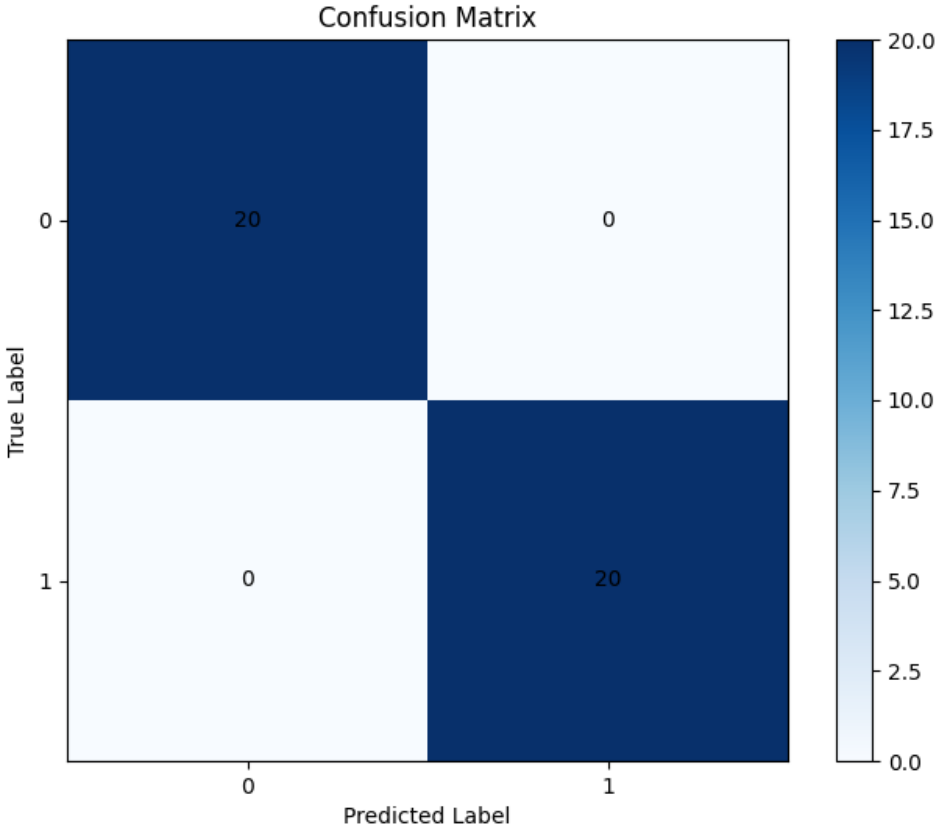
    a. Постройте алгоритм метода опорных векторов с линейным ядром. Визуализируйте разбиение пространства признаков на области с помощью полученной модели (пример визуализации). Выведите количество полученных опорных векторов, а также матрицу ошибок классификации на обучающей и тестовой выборках.

    Accuracy: 1.0

    Number of support vectors: 6

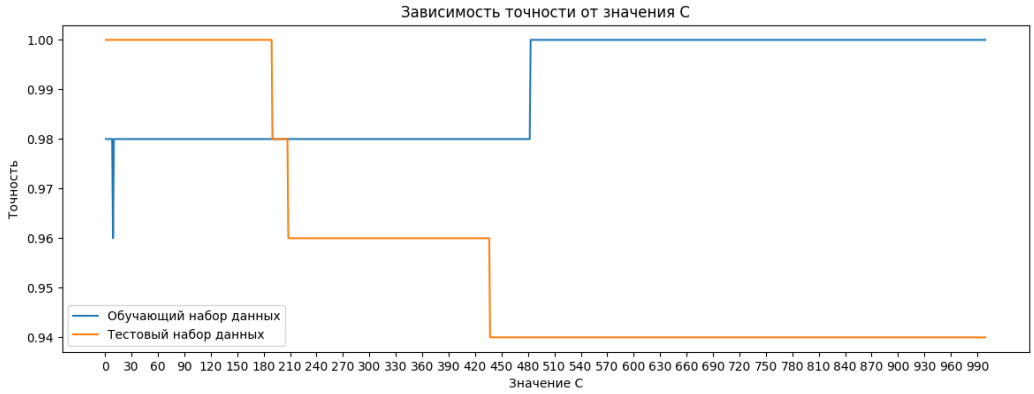
    Матрица ошибок также показывает, что все точки определены верно, точность = 1





b. Постройте алгоритм метода опорных векторов с линейным ядром. Добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения

штрафного параметра. Выберите оптимальное значение данного параметра и объясните свой выбор. Всегда ли нужно добиваться минимизации ошибки на обучающей выборке?



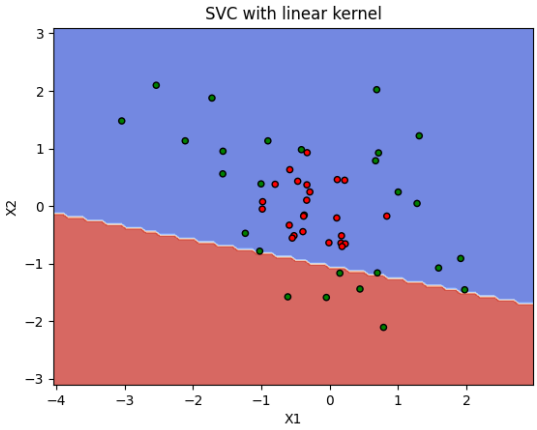
Нулевая ошибка на обучающей выборке получается при штрафном параметре > 482.

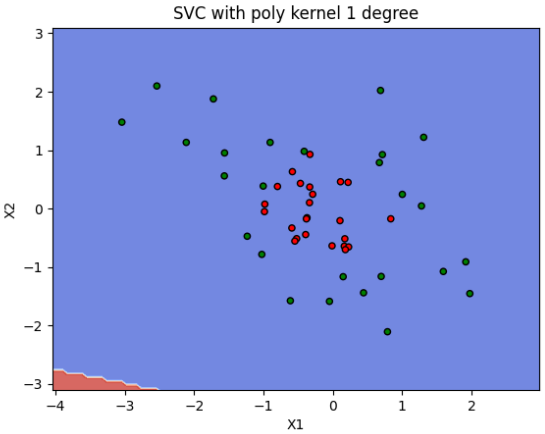
Нулевая ошибка на тестовой выборке получается при штрафном параметре < 190.

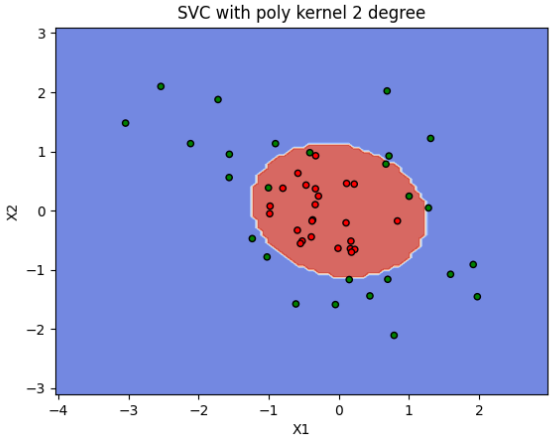
Таким образом, увеличении штрафного параметра повышает точность обучающей выборки, но уменьшает точность тестовой. Следовательно, происходит переобучение и не следует добиваться нулевой ошибки на обучающей выборке.

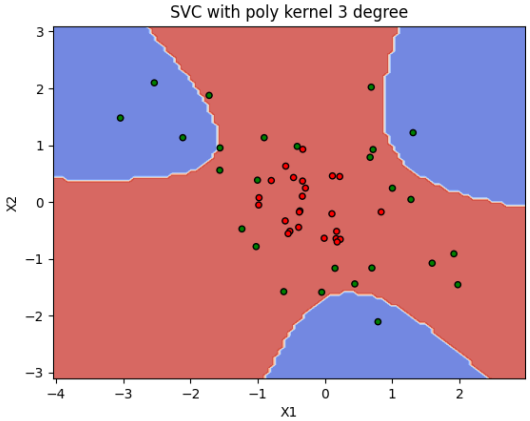
c. Постройте алгоритм метода опорных векторов, используя различные ядра (линейное, полиномиальное степеней 1-5, сигмоидальная функция, гауссово). Визуализируйте разбиение пространства признаков на области с помощью полученных моделей. Сделайте выводы.

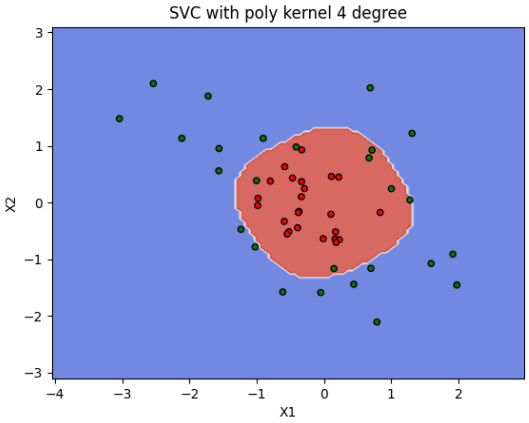
По полученным графикам можно сделать вывод, что алгоритм метода опорных векторов с линейным ядром, полиномиальным степени 1 наименее точные. Наибольшая точность у метода с полиномиальным ядром 2 и 4 степени.

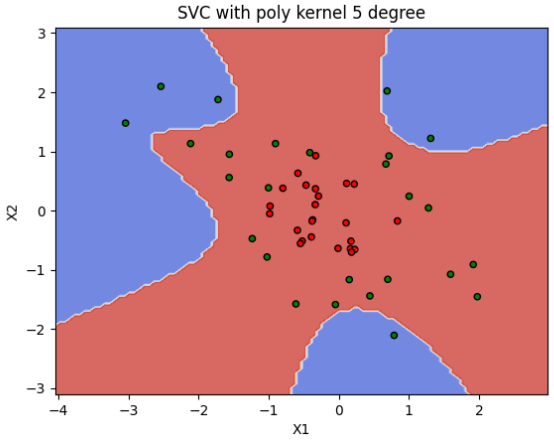

SVC with linear kernel
X1
X2

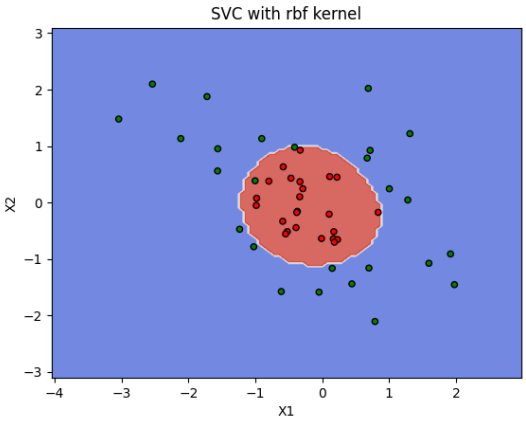

SVC with poly kernel 1 degree
X1
X2

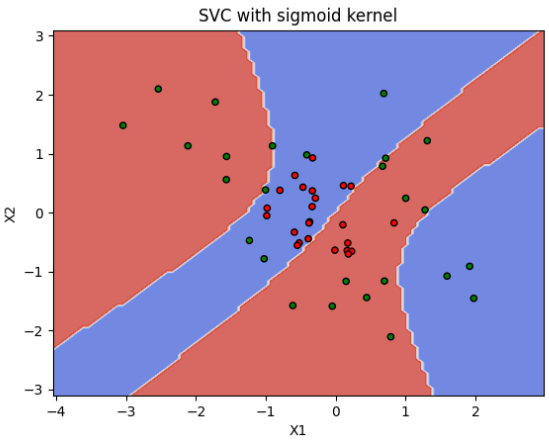

SVC with poly kernel 2 degree
X1
X2


SVC with poly kernel 3 degree
X1
X2


SVC with poly kernel 4 degree
X1
X2

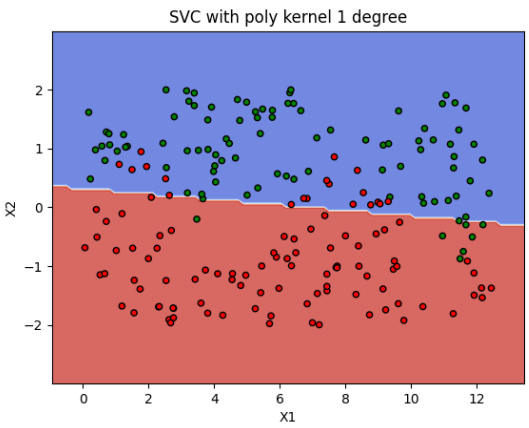

SVC with poly kernel 5 degree
X1
X2

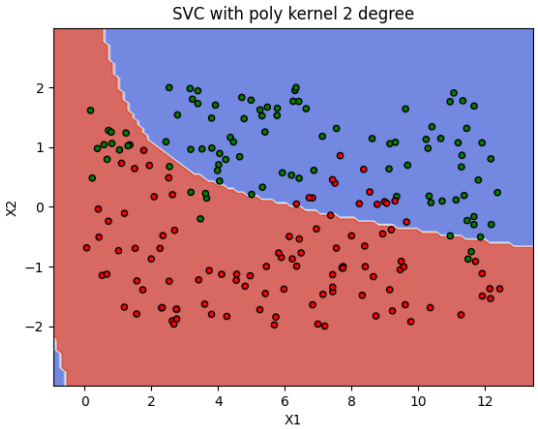

SVC with rbf kernel
X1
X2

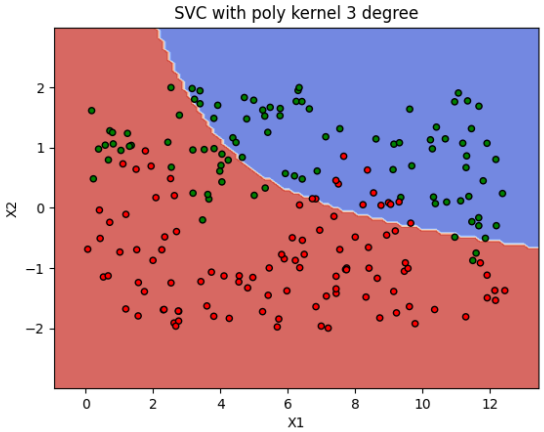

SVC with sigmoid kernel
X1
X2

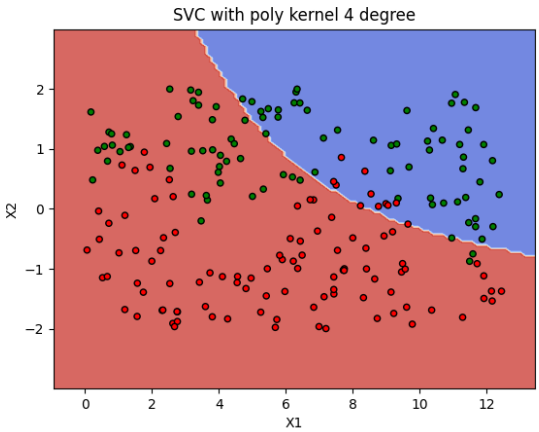
d. Постройте алгоритм метода опорных векторов, используя различные ядра (полиномиальное степеней 1-5, сигмоидальная функция, гауссово). Визуализируйте разбиение пространства признаков на области с помощью полученных моделей. Сделайте выводы.

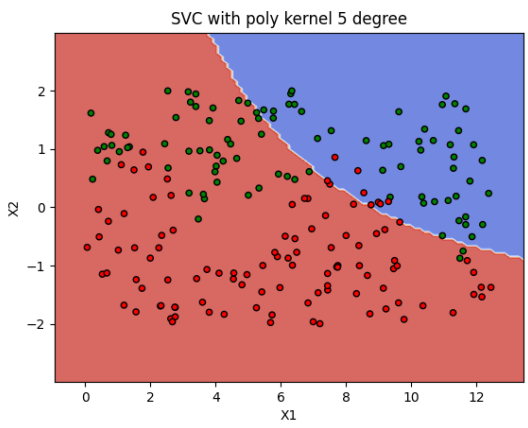
По полученным графикам можно сделать вывод, что алгоритм метода опорных векторов с сигмоидальным ядром наименее точный. Наибольшая точность у метода с гауссовым ядром. Менее точный с полиномиальным ядром 1, 2 степеней, метод с остальными ядрами еще менее точен.

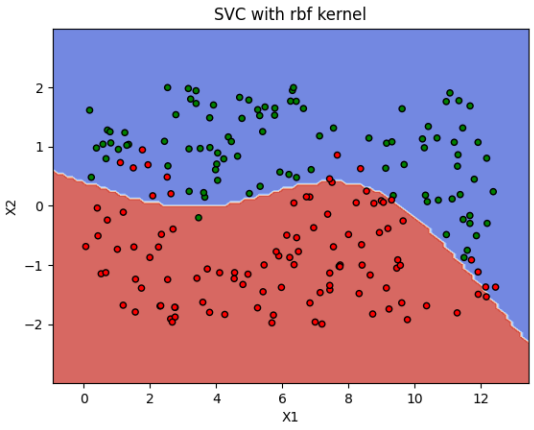


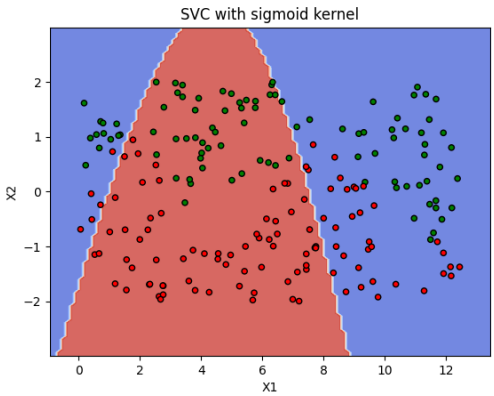






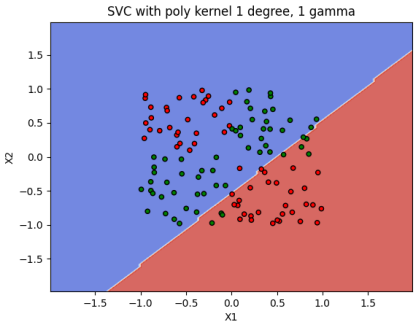


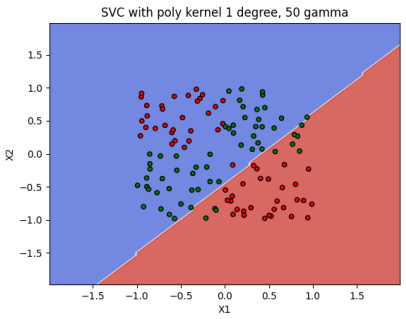


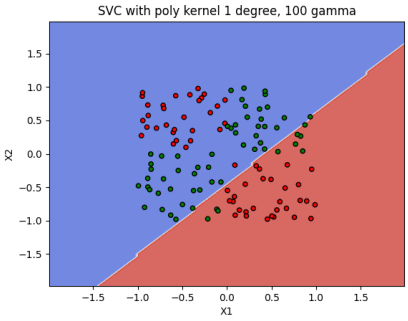


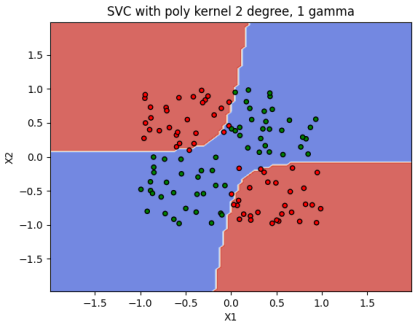
e. Постройте алгоритм метода опорных векторов, используя различные ядра (полиномиальное степеней 1-5, сигмоидальная функция, гауссово). Изменяя значение параметра ядра (гамма), продемонстрируйте эффект переобучения, выполните при этом визуализацию разбиения пространства признаков на области.

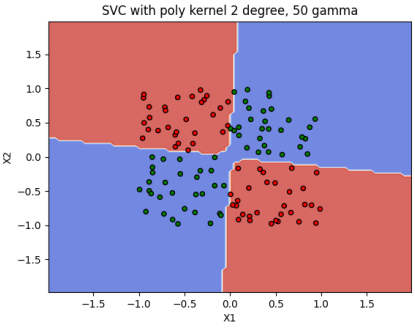
По полученным графикам можно сделать вывод, что сначала при увеличении параметра ядра гамма увеличивается точность классификации, однако при слишком большом значении параметра наступает переобучение модели, что негативно скажется на точности классификации тестовых данных.

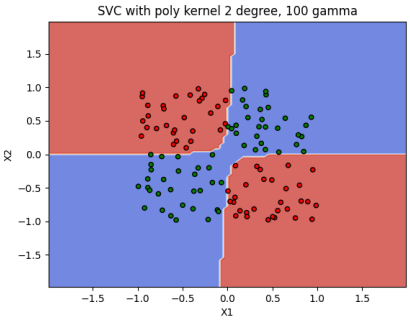

SVC with poly kernel 1 degree, 1 gamma
X1
X2

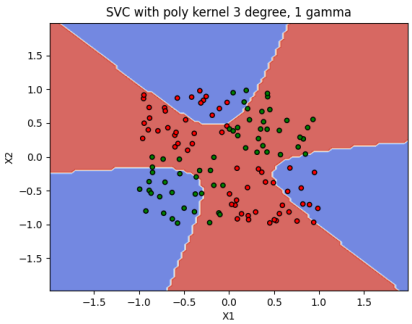

SVC with poly kernel 1 degree, 50 gamma
X1
X2

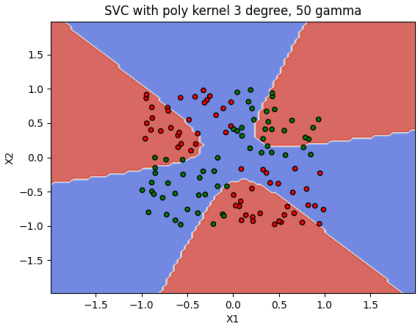

SVC with poly kernel 1 degree, 100 gamma
X1
X2

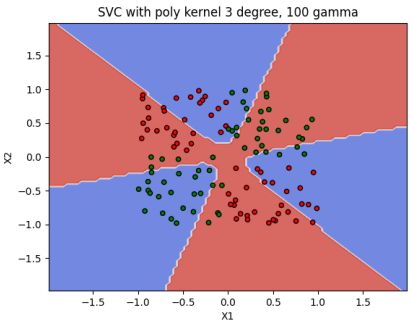

SVC with poly kernel 2 degree, 1 gamma
X1
X2

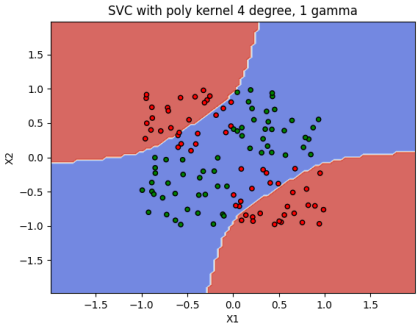

SVC with poly kernel 2 degree, 50 gamma
X1
X2

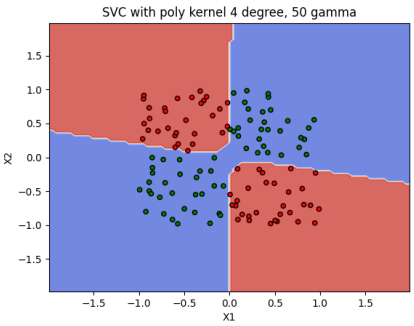

SVC with poly kernel 2 degree, 100 gamma
X1
X2

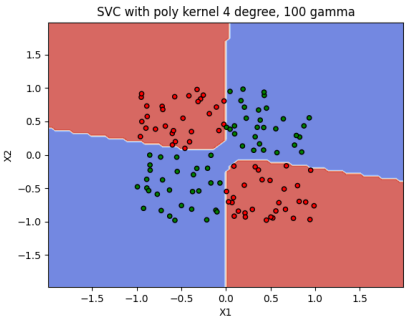

SVC with poly kernel 3 degree, 1 gamma
X1
X2

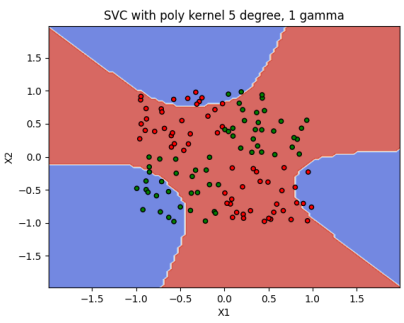

SVC with poly kernel 3 degree, 50 gamma
X1
X2

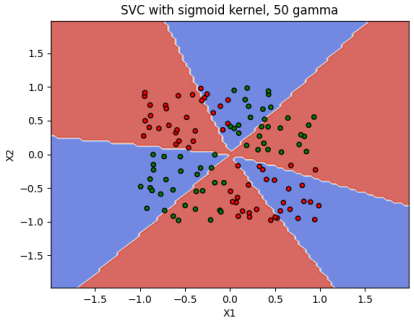

SVC with poly kernel 3 degree, 100 gamma
X1
X2

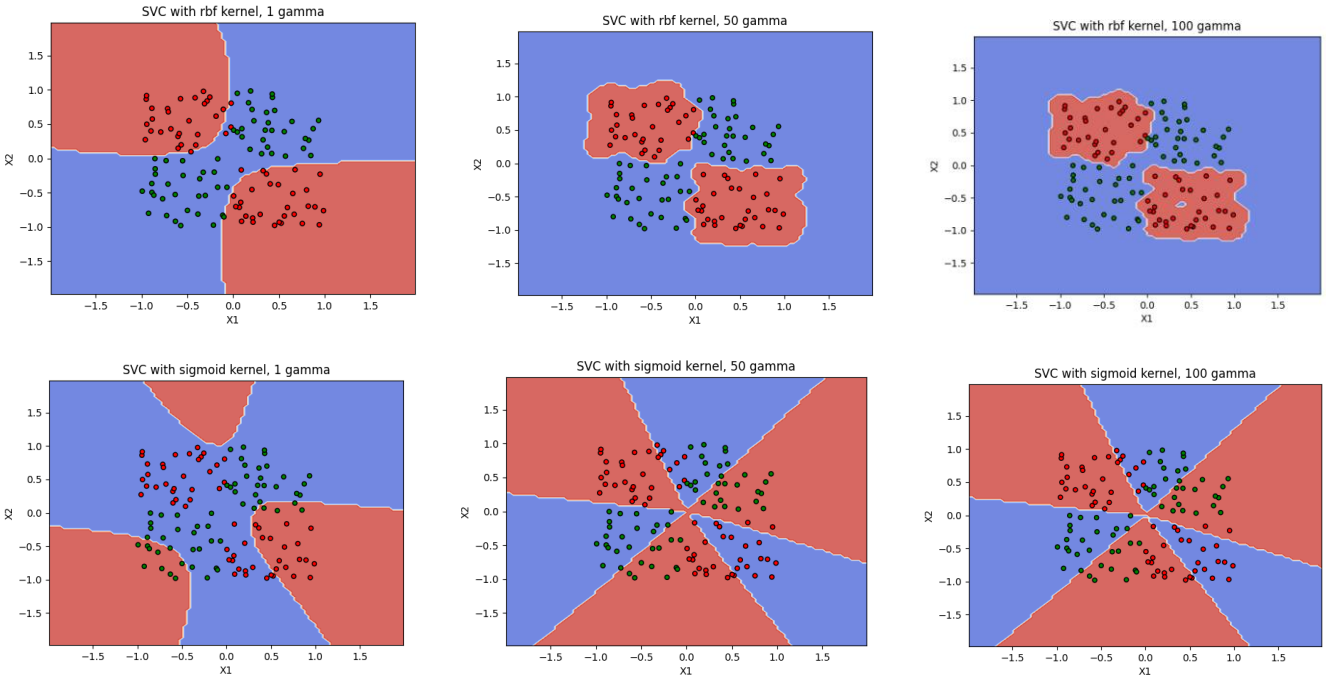

SVC with poly kernel 4 degree, 1 gamma
X1
X2


SVC with poly kernel 4 degree, 50 gamma
X1
X2

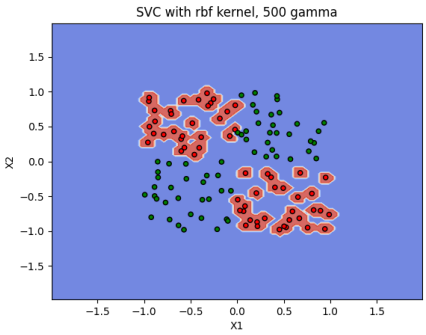

SVC with poly kernel 4 degree, 100 gamma
X1
X2


SVC with poly kernel 5 degree, 1 gamma
X1
X2


SVC with sigmoid kernel, 50 gamma
X1
X2

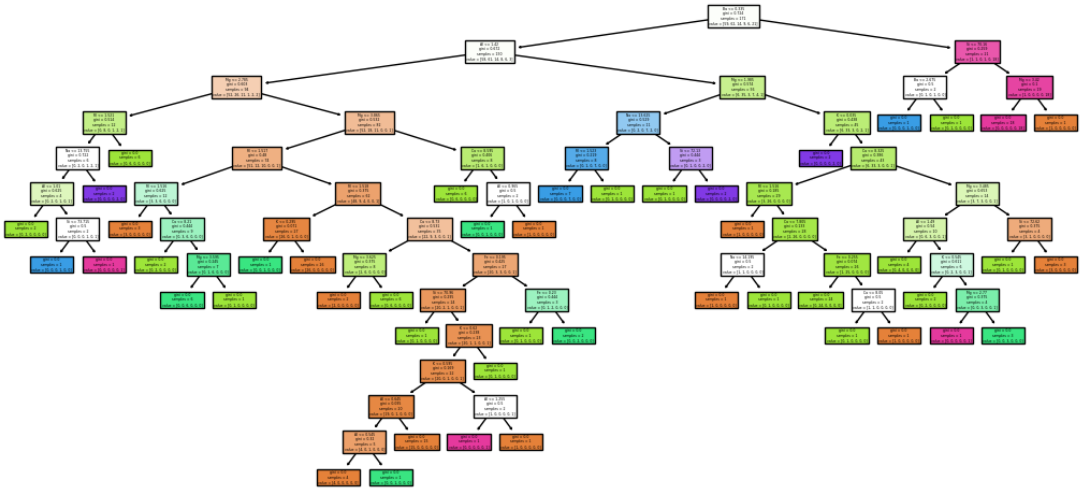


Пример переобучения:



5. Постройте классификаторы для различных данных на основе деревьев решений:

   a. Загрузите набор данных Glass из файла glass.csv.
   Постройте дерево классификации для модели, предсказывающей тип (Type) по остальным признакам. Визуализируйте результирующее дерево решения. Дайте интерпретацию полученным результатам. Является ли построенное дерево избыточным? Исследуйте зависимость точности классификации от критерия расщепления, максимальной глубины дерева и других параметров по вашему усмотрению.

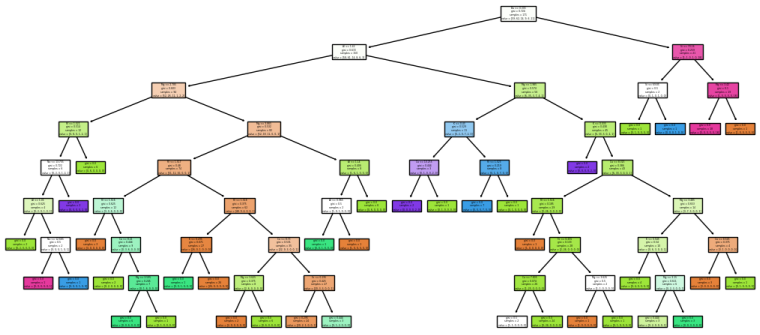


Accuracy: 0.7906976744186046

Depth: 13

Max Depth	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.511628	0.651163	0.740186	0.790698	0.767442	0.720930	0.740186	0.837209	0.767442	0.790698	0.790698	0.767442	0.790698	0.837209	0.813953	0.740186	0.860465	0.740186	0.767442	0.790698

У дерева с глубиной 8 точность выше: 0.8372093023255814, следовательно, дерево, которое строится по исходным данным избыточно



b. Загрузите набор данных spam7 из файла spam7.csv. Постройте оптимальное, по вашему мнению, дерево классификации для параметра yesno. Объясните, как был осуществлён подбор параметров. Визуализируйте результирующее дерево решения. Определите наиболее влияющие признаки. Оцените качество классификации.

По умолчанию строится избыточное дерево:

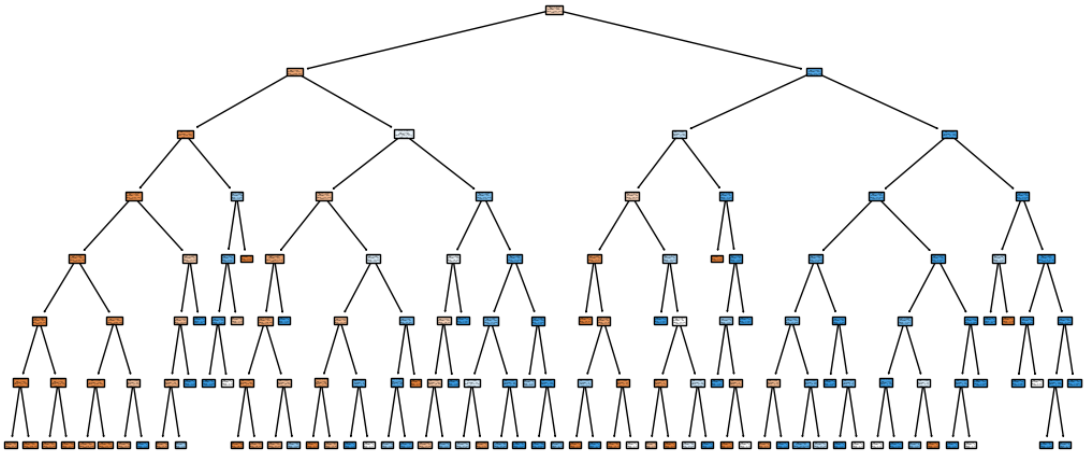
Accuracy: 0.8338762214983714

Depth: 25

С помощью кросс-валидации, получили параметры, при которых дерево не избыточно и точность модели выше, чем по умолчанию.

Наилучшие параметры: {'max_depth': 7, 'min_samples_leaf': 2, 'min_samples_split': 2}

Точность наилучшей модели: 0.8675352877307275

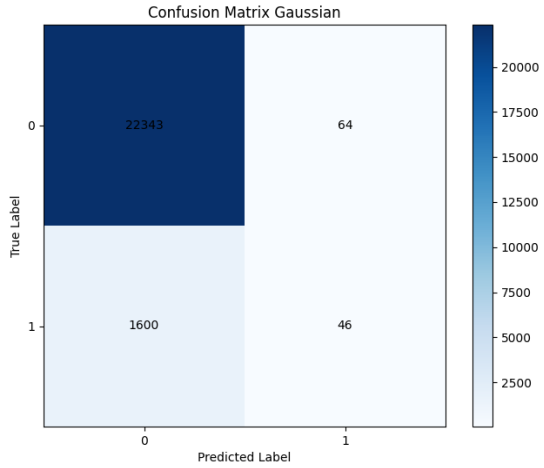


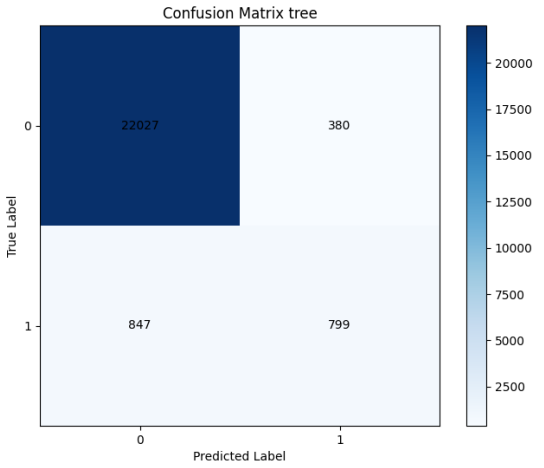
6. Загрузите набор данных из файла bank_scoring_train.csv. Это набор финансовых данных, характеризующий физических лиц. Целевым столбцом является «SeriousDlqin2yrs», означающий, ухудшится ли финансовая ситуация у клиента. Постройте систему по принятию решения о выдаче или невыдаче кредита физическому лицу. Сделайте как минимум 2 варианта системы на основе различных классификаторов. Подберите подходящую метрику качества работы системы исходя из специфики задачи и определите, принятие решения какой системой сработало лучше на bank_scoring_test.csv.

Решим систему с помощью двух классификаторов: наивного байесовского и дерева классификации.

Оценим результаты классификации с помощью матриц ошибок.

Необходимо минимизировать ложноположительные (правый верхний угол) и ложноотрицательные (левый нижний угол) результаты.





Разберем результаты на примере матрицы ошибок байесовского классификатора:

- Верхний левый элемент (22343) представляет собой количество True Negative (TN), то есть количество клиентов, у которых финансовая ситуация не ухудшилась, и они правильно классифицированы как не имеющие задолженности.
- Верхний правый элемент (64) представляет собой количество False Positive (FP), то есть количество клиентов, у которых финансовая ситуация не ухудшилась, но они неправильно классифицированы как имеющие задолженности.
- Нижний левый элемент (1600) представляет собой количество False Negative (FN), то есть количество клиентов, у которых финансовая ситуация ухудшилась, но они неправильно классифицированы как не имеющие задолженности.
- Нижний правый элемент (46) представляет собой количество True Positive (TP), то есть количество клиентов, у которых финансовая ситуация ухудшилась, и они правильно классифицированы как имеющие задолженности.

По полученным матрицам ошибок можно сделать вывод, что при использовании дерева классификации ложноотрицательные результаты в 2 раза меньше, но значительно больше ложноположительных и истинно положительных.

Для банковской системы больший риск несут FN результаты, тогда банк будет выдавать кредит тем, кому не должен из-за ухудшенного финансового состояния.
FP несет меньший риск, т.к. банк не будет выдавать кредит, хотя у клиента не ухудшится финансовое состояние на самом деле.

Так как мы выяснили, что важнее минимизировать ложноотрицательные результаты, то с решением задачи лучше справилась система с деревом классификации.