Отчет по практическому заданию: Метод опорных векторов

Выполнено студентом 317 группы, Филимоновым Владиславом

1 Замечание о выполненной работе

Задание было выполнено не полностью: часть экспериментов не было проведена и соответственно не отражена в отчете. При этом код написан и отлажен, эксперименты были выбраны так, чтобы продемонстрировать работоспособность всех алгоритмов.

2 Введение

Данный отчет имеет три части:

- особенности реализации (тут описана моя реализация генерации хорошо разделимых, но линейно не разделимых данных);
- описание используемых датасетов и их визуализация для 2-мерного признакового пространства;
- эксперименты;

3 Особенности реализации

Для исследования необходимо было придумать датасеты различной сложности, так чтобы в них можно было менять количество признаков и объектов, а сложность не менялась. Такие датасеты легко придумать для линейно разделимых и плохо разделимых данных (облака точек из нормального распределения с удаленными и близкими центрами соответственно). Можно заметить, что при увеличении размерности все признаки в таких моделях являются важными. Это свойство(все признаки информативные) кажется мне важным, так как в реальных задач мы сначала проведем отбор признаков и оставим только информативные, так что исследовать свойства алгоритма на датасете с большим числом не информативных(а тем более случайных) признаков кажется неправильным. Придумать датасет, который разделим, но не линейно, и обладает такими свойствами(можно менять размерность, все признаки важные) оказалось нетривиальной задачей.

Для его генерации сначала генерируется выборка из многомерного нормального распределения с нулевым вектором мат. ожиданий и единичной матрицей ковариаций.

$$x_i \sim N(E, \sigma^2), E = (\underbrace{0, ..., 0}_{n}), \sigma^2 = I \in R^{n \times n},$$

где x_i - i-тый объект выборки. Рассмотрим следующее решающее правило для классов:

$$\varphi : x \to y, x \in R^{n \times 1}, y \in R, \varphi(x) = 2[\sum_{i=1}^{n} x_i^2 < R] - 1,$$

где x - любой объект выборки. Таким образом, те объекты, которые попадают в круг некоторого радиуса буду иметь класс 1, а остальные -1. Но необходимо, чтобы выборка была сбалансирована и нужна устойчивость, то есть если сгенерировать еще одну выборку из того же распределения и использовать то же решающее правило, то примерно такое же количество объектов должно иметь класс 1. Для этого используется свойство, что каждый признак объекта из описанного многомерного нормального имеет стандартное нормальное распределение($x_i \sim N(0, 1)$, для любого x - объекта выборки). А сумма квадратов признаков для каждого объекта будет иметь распределение ($\sum_{i=1}^{n} x_i^2 \sim \chi^2(n)$, для любого x - объекта выборки). Таким образом, достаточно положить:

$$R = mediana(\chi^2(n)) \approx n - \frac{2}{3}$$

4 Описание используемых датасетов

Для создания датасетов для разной размерности и количества объектов была написана функция create data, ее можно найти в приложенном модуле data.py В данной работе использовались следующие датасеты:

- линейно разделимые данные, которые были сгенерированы с помощью функции make blobs из sklearn, пример такого датасета для 2 признаков и 100 объектов можно найти на рис. 1. Такой датасет в дальнейшем называется "датасет 0";

- хорошо разделимые, но не линейно разделимые данные были сгенерированы с помошью функции my circles, реализацию которой можно найти приложенном модуле data.py (О реализации этой функции см 3). Пример такого датасета приведен на рис. 2. Такой датасет в дальнейшем называется "датасет 1";

- плохо разделимые данные были сгенерированы с помощью функции make blobs, так что центр точек, которые имеют класс 1, находится в начале координат, а центр второго шара генерируется случайно, а затем нормируется так, что расстояние от начала координат до этого шара равно 1. При этом мы получаем плохо разделяемые данные для любой размерности, так как make blobs использует многомерное нормальное распределение, в котором вектор матожиданий совпадает с центром шара (для каждого шара=класса), а матрица ковариаций единична. Таким образом, если расстояние расстояние между центрами шаров равно 1, что совпадает с стандартным отклонением от центра по каждой оси. Это гарантирует то, что данные будут очень плохо разделимы. Пример такого датасета можно найти на рис.3. Такой датасет в дальнейшем называется "датасет 2";

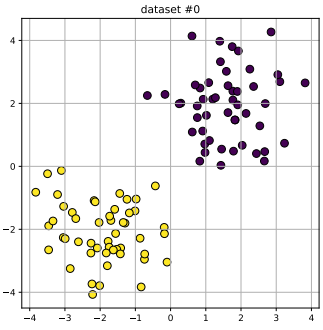

Рис. 1: Датасет 0

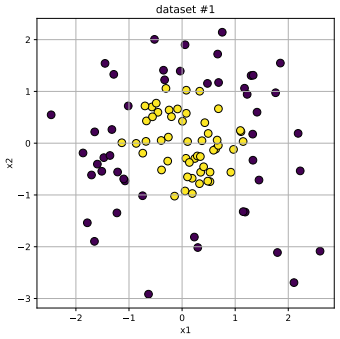

Рис. 2: Датасет 1

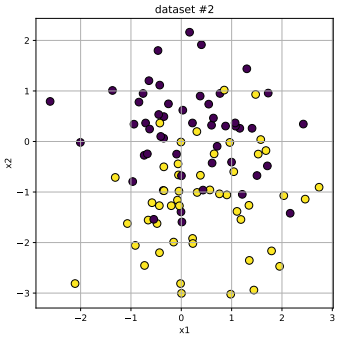

Рис. 3: Датасет 2

- линейно разделимые данные с несбалансированным числом классов, эти данные генерируются с помошью многомерного нормального распределения.Для объектов первого класса используется вектор мат. ожиданий $(2, \ldots, 2)$, а для второго $(-2, \ldots, -2)$. Матрица ковариаций единичная. Эти векторы были выбраны так, что при n=2, объект первого класса будет находится ближе к центру второго класса с очень маленькой вероятностью(порядка 10^{-2}), а при увеличении числа признаков расстояние между центрами будет только расти. При этом объекты первого класса составляют 10% от всей выборки. Такой датасет в дальнейшем называется "датасет 3"

- данные с выбросами, для генерации которых используется многомерное нормальное распределение, описанное в предыдущем пункте (вектора мат. ожиданий и ковариационные матрицы те же самые для объектов первого и второго класса). Для генерации выбросов был придуман следующий алгоритм:

  1. Для каждого признака создается равномерная сетка от минимального до максимального значения этого признака, так что число элементов сетки составляет 30% от полного датасета.
  2. Этот вектор становится одним из признаков датасета-выбросов
  3. После проделанной операции остается выбрать классы для заданных объектов выбросов. Класс назначается случайно.

На рис.5 приведен пример такого датасета. На картинке видно, что данные-выбросы попадают на одну прямую, но это не является существенным недостатком, так как модель не является настолько сложной, чтобы это заметить. Такой датасет в дальнейшем называется "датасет 4"

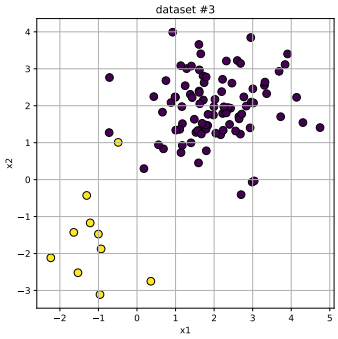

Рис. 4: Датасет 3

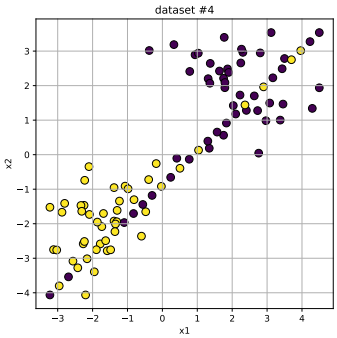

Рис. 5: Датасет 4

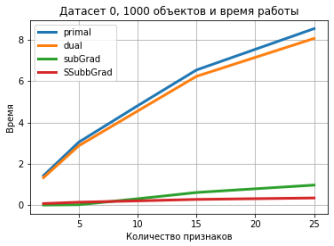

Рис. 6: Время от количества признаков (датасет 0)

5 Эксперименты

5.1 Исследование времени работы от числа признаков и объектов и значений целевой функции для SVM с линейным ядром

Данное исследование будет проводится на датасетах 0, 1, 2. Также алгоритм Pegasos здесь не рассматривается, так как он имеет фиксированное число итераций. На рис. 6 показана зависимость времени обучения от числа признаков для линейно разделимых данных. Можно заметить, что субградиентные методы значительно быстрее метода внутренней точки(объяснение не было найдено, т.к. неизвестна реализация метода внутренней точки). На рис. 7, 8 показана аналогичные графики для других датасетов. Из этих графиков можно сделать вывод, что время обучения растет значительно быстрее при росте размерности пространства для методов внутренней точки. Метод внутренней точки имеет наибольшее время обучения на линейно неразделимых данных, а наименьшее на не разделимых данных. Время обучения субградиентных методов для всех датасетов оказалось значительно меньшим.

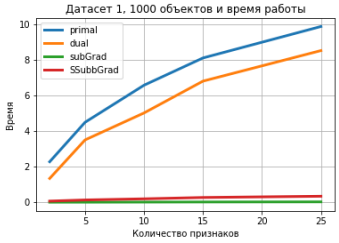

Рис. 7: Время от количества признаков (датасет 1)

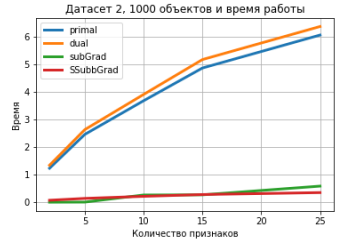

Рис. 8: Время от количества признаков (датасет 2)

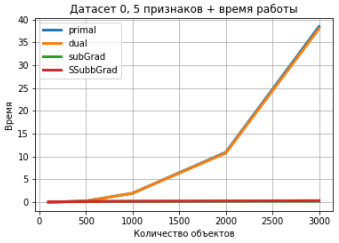

Рис. 9: Время от количества объектов (датасет 0

На рис. 9,10, 11 показана зависимость времени обучения различных методов для различных датасетов. Из этих графиков можно сделать аналогичные выводы.

Теперь рассмотрим значение целевой функции для различных методов и датасетов (при этом датасет всегда имеет размер (1000,5)). В таблице 1 приведены соответствующие данные. Можно заметить, что метод dual, единственный, кто имеет отрицательные значения целевой функции. Это можно объяснить тем, что математически двойственная задача определяется как задача максимизации, а в моей реализации она решалась как прямая задача(задача минимизации), поэтому

Рис. 10: Время от количества объектов (датасет 1)

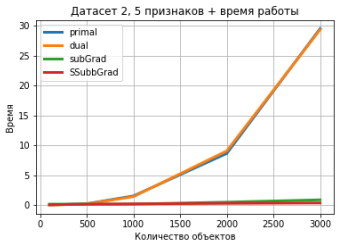

Рис. 11: Время от количества объектов (датасет 2)

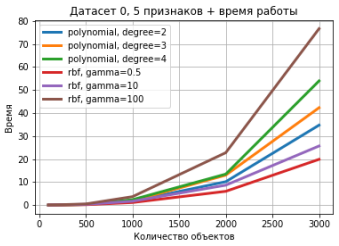

Рис. 12: Время от количества объектов(датасет 0)

соответствующий функционал был умножен на -1. Из этой таблице видно, что и прямая, обратная задача дают одинаковые значения целевого функционала SVM с точностью до знака(что подтверждает наличие сильной двойственности). Так же можно заметить, что методы внутренней точки имеют значительно меньшие значения целевой функции для датасета с выбросами. Значение целевых функций имеют один порядок для всех методов для линейно не разделимых данных и плохо разделимых данных (это можно объяснить тем, что для линейных методов эти два типа данных одинаково плохо классифицируются)

Таблица 1: Значение целевой функции для различных методов

Метод	датасет 0	датасет 1	датасет 2	датасет 3	датасет 4
primal	0.06	1.00	0.88	0.05	0.49
dual	-0.06	-1.00	-0.88	-0.05	-0.49
SubGrad	0.07	1.00	0.84	0.05	1.66
SSubGrad	0.05	1.00	0.81	0.03	2.19

5.2 Исследование времени работы от числа признаков и объектов и значений целевой функции для SVM с нелинейным ядром

В данном разделе будут рассмотрены классификаторы с rbf и полинимальными ядрами, при этом $\gamma \in \{0.5, 10, 100\}, d \in \{2, 3, 4\}$. Как и в предыдущем эксперимент, для измерения времени от количества признаков было фиксировано количество объектов(1000), и наоборот(количество признаков-5). На рис.12, 13,13 показана зависимость времени от числа объектов на датасетах 0,1,2 (кол-во признаков 5). Можно заметить, что наибольшее время обучения имеет алгоритм с rbf ядром и $\gamma = 100$.

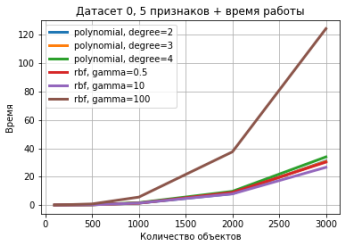

Рис. 13: Время от количества объектов (датасет 1)

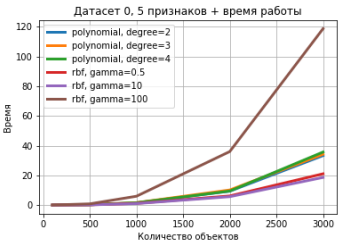

Рис. 14: Время от количества объектов (датасет 2)

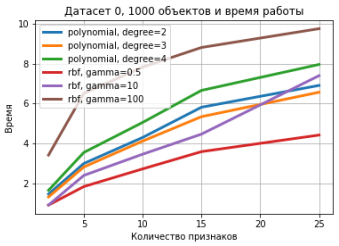

Рис. 15: Время от количества признаков(датасет 0)

Интуитивно понятно, что большие значения γ и d приводят к более сложным моделям. Поэтому логично, что время обучения занимает больше времени при большом γ. Так же можно заметить,

что алгоритмы с rbf ядром либо имеют похожее время обучения, либо время обучения ранжируется по сложности(по степени ядра). Из рисунков 15, 16, 17, где показана зависимость времени от количества объектов, можно сделать аналогичные выводы.

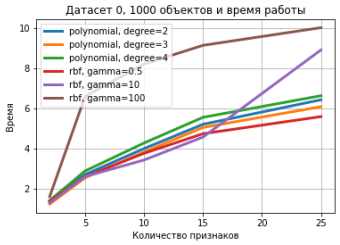

Рис. 16: Время от количества признаков (датасет 1)

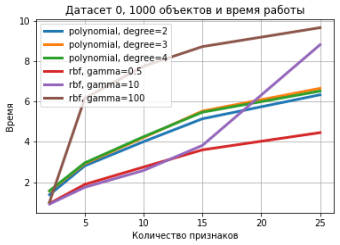

Рис. 17: Время от количества признаков (датасет 2)

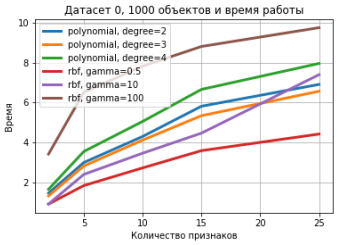

Рис. 18: CHANGE

Рассмотрим теперь значения целевой функции, напомним, что они будут отрицательными в силу реализации. В таблице 2 приведены соответствующие данные. К сожалению эти данные являются не слишком интерпретируемыми, так как производится некоторое преобразование целевого функционала. Например, для линейно разделимых данных rbf ядро имеет единичный целевой функционал, в то время как полиномимальное ядро имеет на данном датасете 0 целевой функционал.

Таблица 2: Значение целевой функции для различных методов с нелинейным ядром

Метод	датасет 0	датасет 1	датасет 2	датасет 3	датасет 4
poli, degree=2	-0.04	-0.62	-0.80	-0.03	-0.44
poli, degree=3	-0.00	-0.43	-0.72	-0.00	-0.32
poli, degree=4	-0.00	-0.17	-0.64	-0.00	-0.30
rbf, gamma=0.5	-0.98	-0.96	-1.00	-0.20	-0.98
rbf, gamma=10	-1.00	-0.97	-1.00	-0.20	-0.99
rbf, gamma=100	-1.00	-0.97	-1.00	-0.20	-0.99

5.3 Визуализация в 2-мерном пространстве

В данном эксперименте будет дополнительно рассмотрен датасет сгенерированный с помошью make moons из sklearn. А так же датасет 0 и 1. На графиках ниже для каждого датасета приведены различные алгоритмы, а так же визуализация всей выборки, и опорныз векторов. Из рисунков ниже можно сделать вывод, что большие значение γ и d приводят к усложнению модели и могут приводить к переобучению.

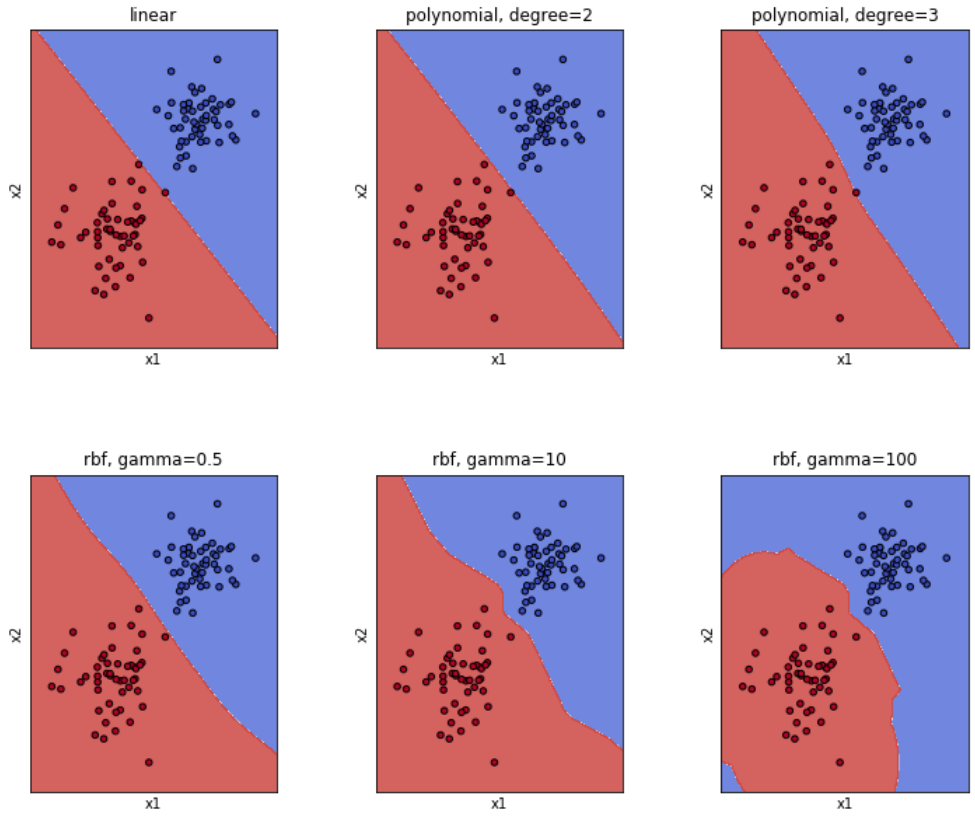

Рис. 19: Датасет 0 с полностью визулизированной выборкой

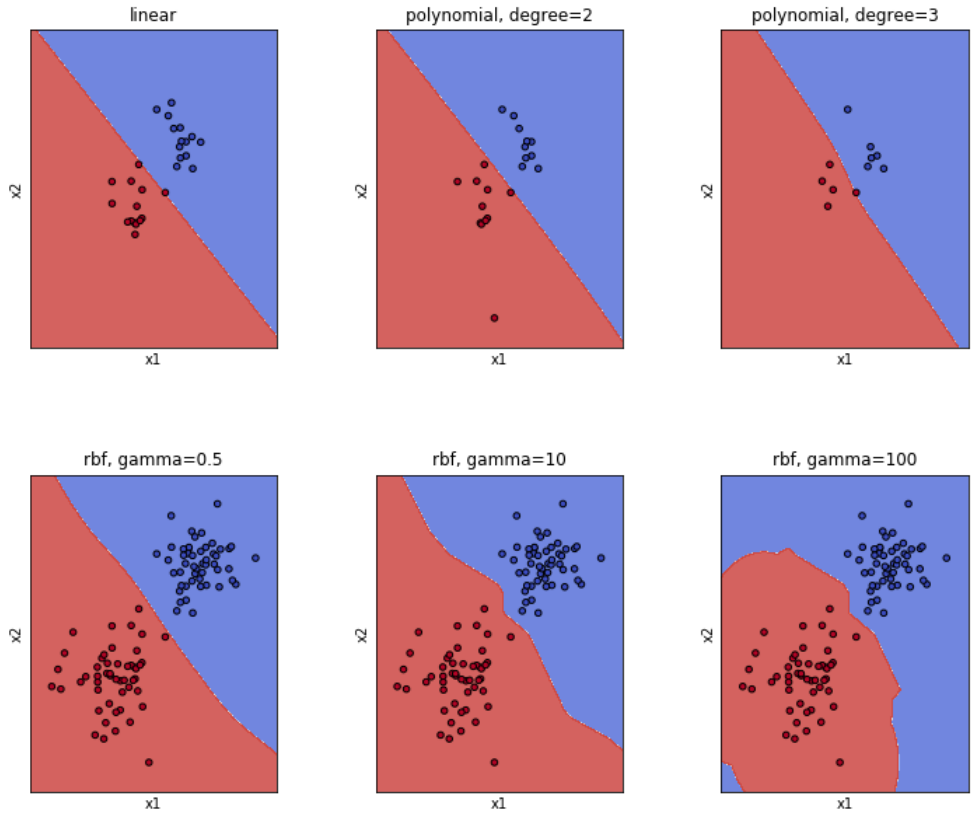

Рис. 20: Датасет 0 с визуализацией только опорных векторов

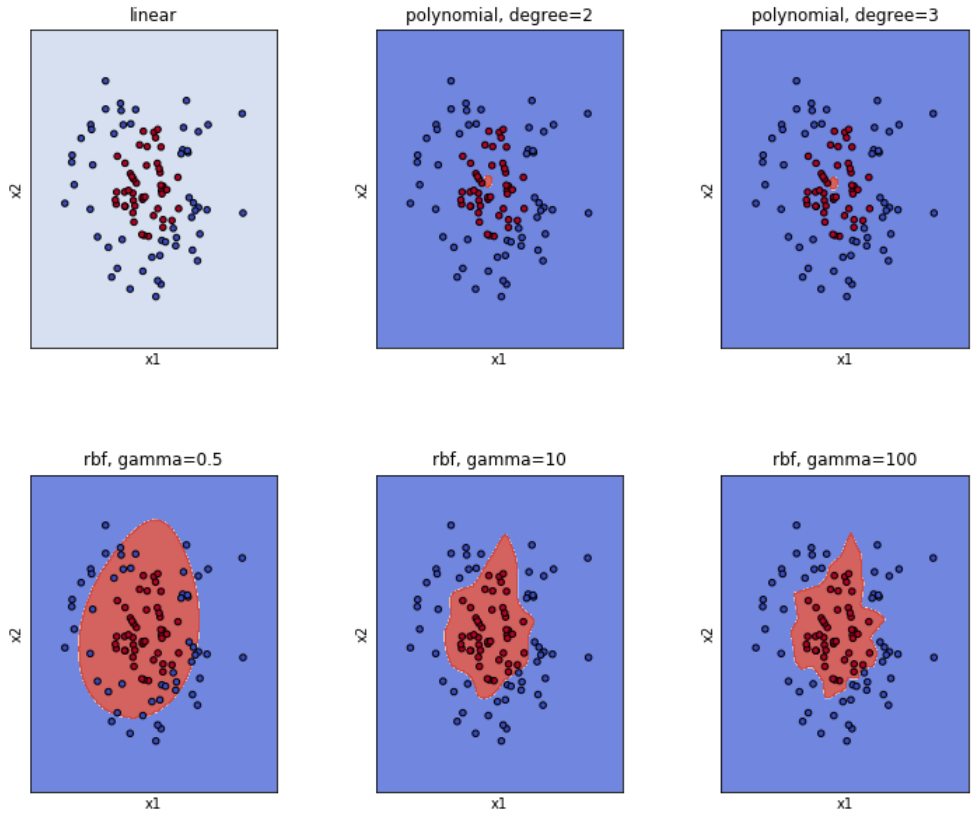

Рис. 21: Датасет 1 с полностью визулизированной выборкой

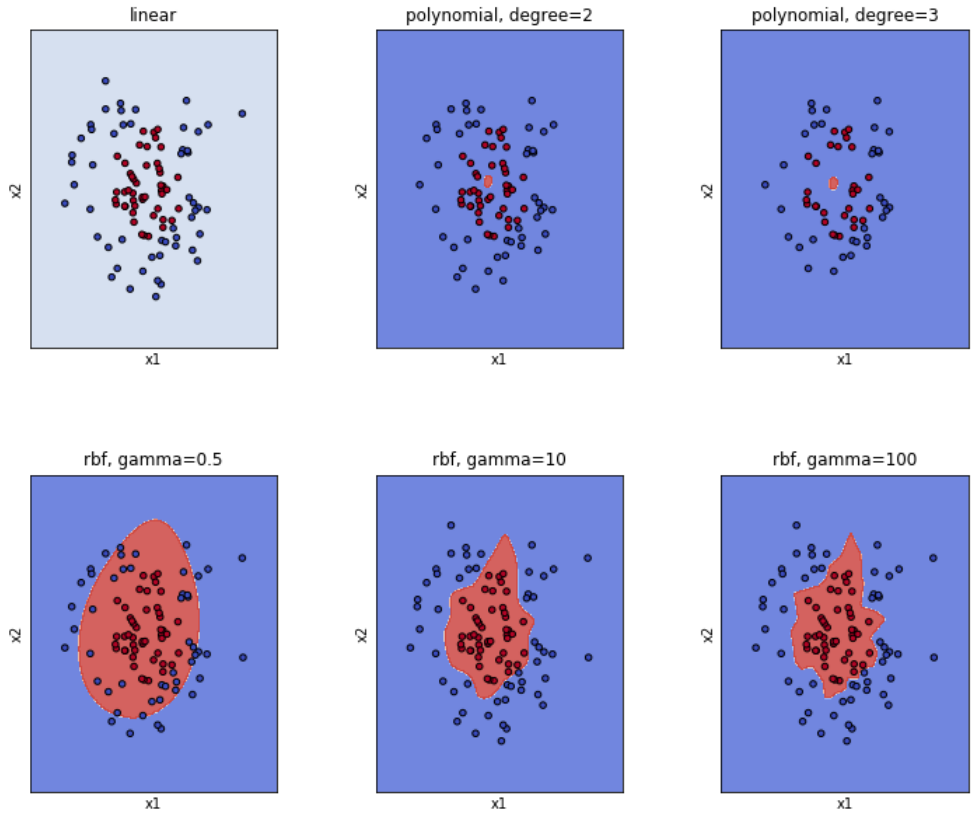

Рис. 22: Датасет 1 с визуализацией только опорных векторов

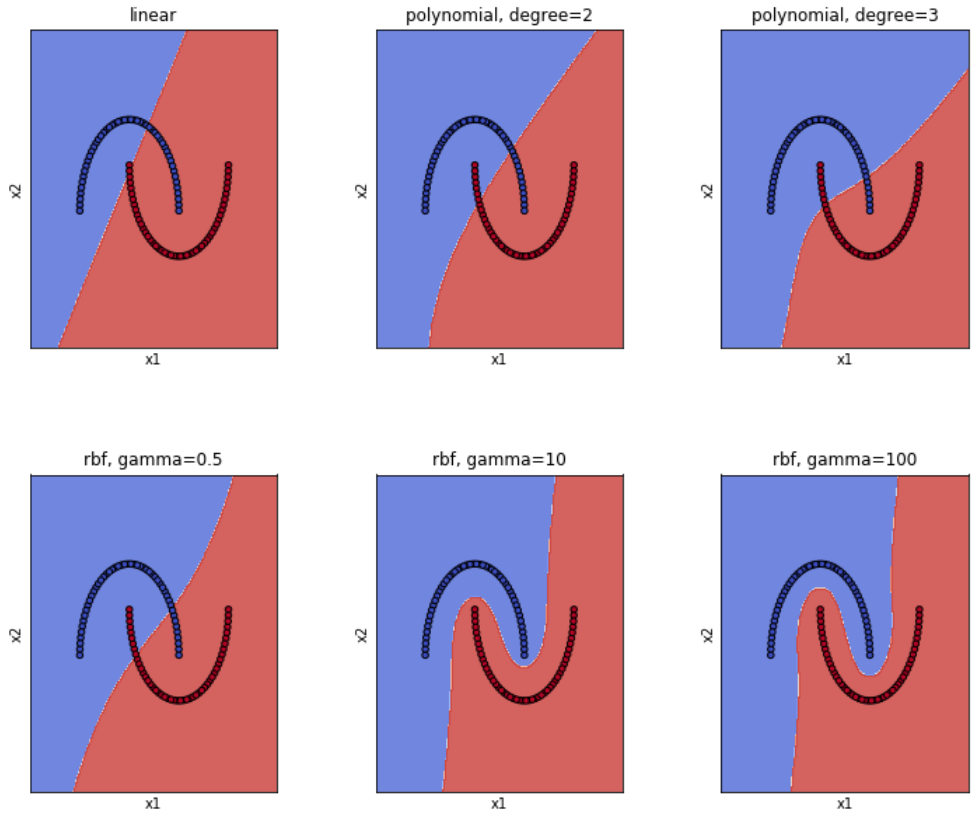

Рис. 23: Датасет make moons с полностью визулизированной выборкой

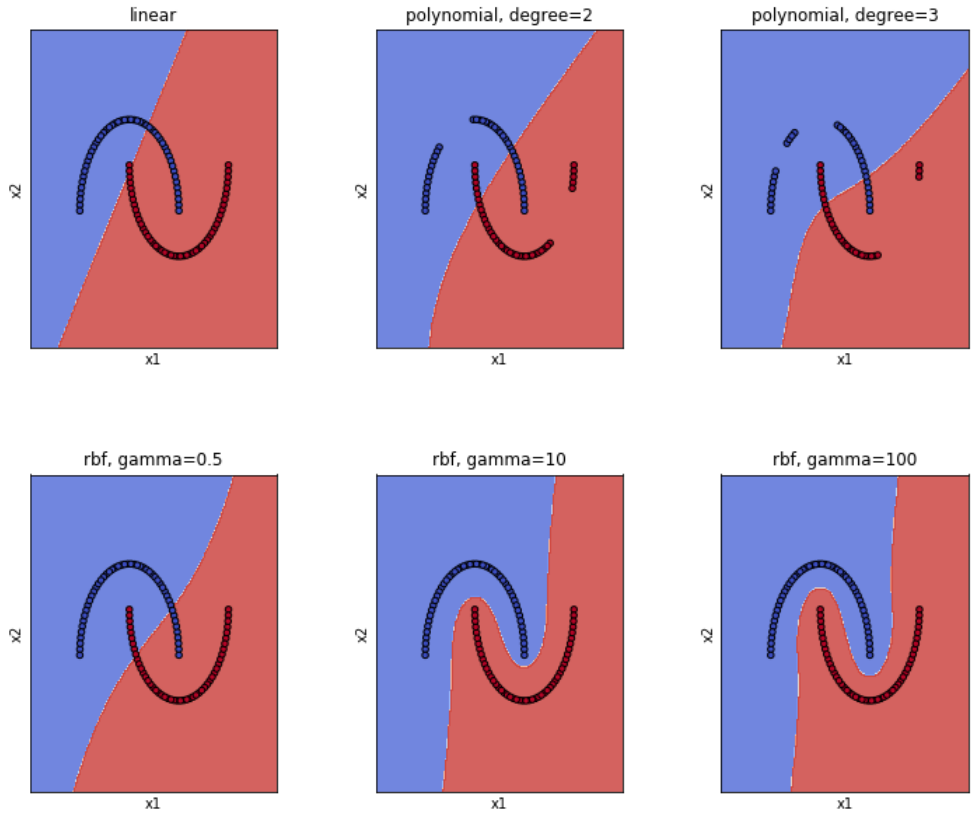

Рис. 24: Датасет make moons с визуализацией только опорных векторов