

Московский Государственный Университет им. М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчёт по заданию «Метод опорных векторов»

М. М. Шамшиев
317 группа
17 декабря 2017

Аннотация

В данном отчете отражены основные результаты, полученные в ходе выполнения практического задания «Метод опорных векторов». Целью задания было написание собственных реализаций различных методов для решения задачи SVM, включающих в себя методы внутренней точки для решения прямой и двойственной задач, методы субградиентного и стохастического субградиентного спусков для решения безусловной задачи SVM, метод PEGASOS.

Также был проведён ряд экспериментов на модельных данных различной сложности, по результатам которых были сделаны соответствующие выводы.

Содержание

1	Исследование времени работы различных реализаций линейного SVM	2
1.1	В зависимости от размерности признакового пространства	2
1.2	В зависимости от размера обучающей выборки	3
2	Исследование времени работы различных реализаций нелинейного SVM	4
3	Сравнение точности различных подходов при решении задачи классификации	5
4	Поиск оптимальных значений параметров	6
4.1	Параметр C	7
4.2	Степень полиномиального ядра	8
4.3	Ширина RBF ядра	9
5	Исследование скорости сходимости различных стратегий выбора длины шага в субградиентных методах	9
6	Исследование влияния размера подвыборки на скорость сходимости и точность решения	11
7	Демонстрация работы некоторых методов для двумерного случая	12
7.1	Линейно разделимые данные	13
7.2	Хорошо разделимые, но линейно неразделимые данные	14
7.3	Данные с несбалансированными классами	15
7.4	Плохо разделимые данные с выбросами	16
8	Заключение	16

1 Исследование времени работы различных реализаций линейного SVM

1.1 В зависимости от размерности признакового пространства

В данном и последующих экспериментах в качестве линейно разделимых данных использовалась выборка, сгенерированная с помощью двух многомерных нормальных распределений с единичными ковариационными матрицами. При этом к объектам первого класса относились векторы, матожидание каждой координаты которых равнялось 2, а к объектам второго класса — векторы, матожидание каждой координаты которых равнялось -2 . Также выборка была сбалансирована: по 200 объектов в каждом классе.

Все полученные в ходе выполнения эксперимента результаты отображены в Таб.1.

Время работы метода полного субградиентного спуска (GD в Таб.1) в зависимости от размерности признакового пространства измерялось на модели с установленным максимальным числом итераций равным 10 000. Время работы алгоритма, ожидаемо, возрастает с увеличением размерности признакового пространства, поскольку время, затрачиваемое на вычисление градиента, напрямую зависит от количества признаков.

Время работы стохастического субградиентного спуска измерялось на модели с максимальным числом итераций равным 100 000. Также, как и в случае полного субградиентного спуска, время работы модели растёт с увеличением количества признаков (SGD в Таб. 1). Однако необходимо отметить, что при увеличении размерности признакового пространства в 40 раз (с 25 до 1000) время работы алгоритма возросло немногим более, чем на 1 секунду. Это обусловлено тем, что вычисление градиента производится лишь по малой части исходной выборки, что значительно сокращает время выполнения данной операции.

	GD	SGD	PEGASOS	primal	dual
$d = 25$	1.12	3.3	1.12	0.94	0.57
$d = 100$	1.22	3.5	1.16	1.09	0.59
$d = 500$	2.14	4.14	1.5	3.62	0.52
$d = 1000$	5.2	4.42	2.5	6.16	0.55

Таблица 1: Время (с) работы модели в зависимости от размерности признакового пространства d на линейно разделимых данных

Для модели, решающей безусловную задачу SVM методом PEGASOS, максимальное число итераций было установлено равным 10 000. Учитывая, что максимальное количество итераций в 10 раз меньше соответствующего значения параметра метода стохастического субградиентного спуска, модель PEGASOS заметно уступает модели SGD по времени работы. Это может быть обусловлено, например, отсутствием в методе PEGASOS какого-либо досрочного критерия остановки (например, выход при значении нормы разности весов на соседних итерациях меньше заданного параметра), вследствие чего модель вынуждена продолжать работать даже в том случае, если оптимизируемый функционал нисколько не уменьшается.

Отметим, что время работы модели, решающей двойственную задачу, оказалось примерно одинаковым для рассматриваемых признаков пространств. Данный факт может быть объяснен тем, что функционал, соответствующий данной задаче, максимизируется по переменным, количество и значения которых напрямую не зависят от размерности признакового пространства. Также время вычисления скалярных произведений $\langle x_i, x_j \rangle$, фигурирующих в оптимизируемом функционале, хоть и зависит от количества признаков, но может быть произведено лишь единожды.

Такое же исследование было проведено на «плохих» данных, сгенерированных с помощью функции *make_classification* библиотеки *sklearn*. В данном случае только половина всех признаков была информативной, четверть признаков представляла собой различные линейные комбинации информативных признаков, а оставшаяся часть дублировала некоторые уже созданные признаки. Параметры всех моделей остались прежними. Полученные результаты приведены в Таб.2.

В целом, общая картина осталась прежней. Но, сравнивая результаты с Таб.1, можно заметить, что на этом наборе данных все модели затратили больше времени. Таким образом, время работы алгоритмов также напрямую зависит от сложности (с точки зрения разделимости) и информативности признаков исходной выборки.

	GD	SGD	PEGASOS	primal	dual
$d = 25$	1.25	4.47	1.35	1.10	0.73
$d = 100$	1.47	4.54	1.36	1.48	0.70
$d = 500$	2.51	4.79	1.94	3.34	0.72
$d = 1000$	6.05	5.13	3.99	8.37	0.79

Таблица 2: Время (с) работы модели в зависимости от размерности признакового пространства d на «плохих» данных

1.2 В зависимости от размера обучающей выборки

В данном эксперименте размерность признакового пространства была зафиксирована равной 200 признакам. Результаты эксперимента для случая линейно разделимых данных приведены в Таб.3.

	GD	SGD	PEGASOS	primal	dual
$l = 10$	0.69	2.48	1.03	0.05	0.01
$l = 100$	0.90	4.22	1.13	0.13	0.02
$l = 500$	1.55	4.33	1.24	2.27	1.22
$l = 1000$	2.33	4.36	1.5	10.9	8.80

Таблица 3: Время (с) работы модели в зависимости от размера обучающей выборки l на линейно разделимых данных

Время работы метода полного субградиентного спуска ожидаемо увеличивалось с ростом размера выборки. В это же время, стохастический субградиентный спуск показал примерно одинаковые результаты при размерах выборки $l = 100, 500, 1000$, что является известным достоинством данного метода. Модель PEGASOS, являясь модификацией метода стохастического субградиентного спуска, также показала хорошие результаты.

В это же время, модели, решающие прямую и двойственную задачу, показали сильные зависимости продолжительности процесса обучения от размера обучающей выборки. Это обусловлено наличием в функционалах обеих задач параметров, по которым проводится оптимизация, количество которых напрямую зависит от размера исходной выборки.

В заключение была исследована скорость сходимости некоторых методов для случая линейно разделимой выборки длины 1000 при количестве признаков равном 1000. Темп обучения (или же длина шага субградиентного спуска) задается формулой $step = \alpha/k^\beta$, где k — номер итерации. В данном эксперименте для модели субградиентного спуска были взяты параметры $\alpha = 0.01$, $\beta = 0$, а для модели стохастического субградиентного спуска — $\alpha = 1$, $\beta = 0.5$. Полученные графики приведены на Рис.1-3.

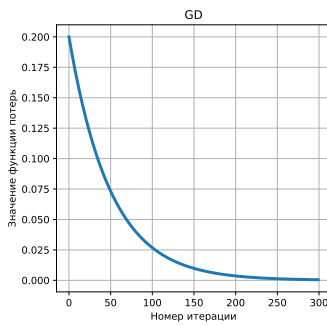


Рис. 1: Субградиентный спуск

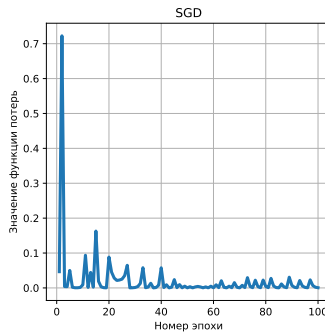


Рис. 2: Стохастический субградиентный спуск

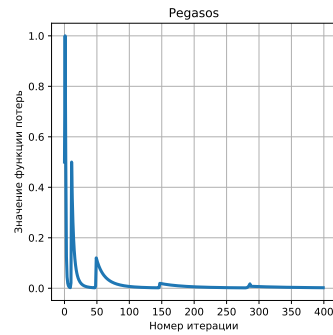


Рис. 3: PEGASOS

Все три модели показали хорошие результаты с точки зрения минимизации функционала. Модель субградиентного спуска достигла значения целевой функции равного 0.000135, стохастическая модель — 0.00017, а модель PEGASOS — 0.00155. Метод внутренней точки для двойственной задачи показал финальное значение целевой функции равное 0.00013. Приняв это решение за точное, можно сказать, что все три метода показали отличные результаты, но максимально близкого к точному решению достигла модель полного субградиентного спуска.

2 Исследование времени работы различных реализаций нелинейного SVM

В данном эксперименте исследованию подверглись модели SVM с полиномиальным ядром степени 2 и RBF ядром с параметром $\gamma = 1$. Эксперимент проводился на линейно неразделимых данных, в качестве которых были взяты пересекающиеся многомерные шары, точки которых принадлежат многомерным нормальным распределениям. Выборка генерировалась с помощью функции `make_blobs` библиотеки `sklearn`. Размер выборки был зафиксирован равным 400.

Как видно из Таб.4, время работы методов оказалось примерно одинаковым для разных значений размерности признакового пространства. Это объясняется тем, что данные модели являются обобщением метода, решающего двойственную задачу, а,

	polynomial	rbf
$d = 25$	0.65	0.50
$d = 100$	0.45	0.47
$d = 500$	0.30	0.39
$d = 1000$	0.30	0.42

Таблица 4: Время (с) работы модели в зависимости от размерности признакового пространства d на линейно неразделимых данных

	polynomial	rbf
$l = 10$	0.002	0.003
$l = 100$	0.01	0.01
$l = 500$	0.58	0.88
$l = 1000$	3.27	4.74

Таблица 5: Время (с) работы модели в зависимости от размера обучающей выборки l на линейно неразделимых данных

как отмечалось ранее, затраты времени таких моделей устойчивы к увеличению количества признаков. Также отметим, что в некоторых ситуациях в случае признаковых пространств с большей размерностью модели затрачивали меньшее количество времени, так как по той или иной причине чаще выходили по досрочному критерию остановки.

Для исследования зависимости времени работы от размера обучающей выборки использовались данные, сгенерированные описанным в начале этого раздела способом. Количество признаков было зафиксировано равным 200. Полученные результаты приведены в Таб.5.

Также как и в случае модели, решающей двойственную задачу с линейным ядром, время работы данных алгоритмов напрямую зависит от размера обучающей выборки: продолжительность процесса обучения растёт при увеличении количества объектов.

Также из приведенной таблицы можно заметить, что время работы модели SVM с RBF ядром превосходит время работы модели с полиномиальным ядром степени 2. Это может быть связано с тем, что операции, необходимые для вычисления значения функции RBF ядра, требуют больше временных затрат по сравнению с операциями для случая полиномиального ядра (например, возведение экспоненты в степень).

3 Сравнение точности различных подходов при решении задачи классификации

Сперва эксперимент проводился на легко линейно разделимых данных, сгенерированных с помощью многомерных нормальных распределений. В этом случае все модели ожидаемо показали 100%-ю точность на отложенной выборке.

Далее эксперимент проводился на линейно неразделимых данных, представляющих собой многомерные пересекающиеся шары, сгенерированные с помощью функции *make_blobs* описанным в предыдущем эксперименте способом. Размер выборки был зафиксирован равным 1000, при чем 30% объектов была отнесена к отложенной выборке для последующего измерения точности. Размерность признакового пространства была установлена равной 20 признакам.

Полученные значения точности различных моделей приведены в Таб.6.

Все линейные модели показали точность, примерно равную 80%, при чем модель PEGASOS показала наибольшую точность среди данного класса моделей. Полученное значение целевой функции данной модели равнялось 0.4. В это же время модель

	GD	SGD	PEGASOS	primal	polynomial	rbf
<i>accuracy</i>	0.8	0.8067	0.81	0.8	0.753	0.71

Таблица 6: Точность различных моделей на отложенной выборке

субградиентного спуска достигла значения функционала равного 0.489, а модель стохастического субградиентного спуска — значения 0.490. Итак, модель PEGASOS, лучше прооптимизировавшая целевой функционал, ожидаемо показала более высокую точность. Модель SVM с полиномиальным ядром степени 2 и модель SVM с RBF ядром при значении параметра $\gamma = 0.01$ показали более низкие точности по сравнению с линейными моделями. Однако отметим, что в данном эксперименте для моделей не производился поиск оптимальных параметров (это будет сделано в следующем эксперименте), поэтому данные значения точностей соответствуют методам со значениями параметров равными значениям по умолчанию.

Также для модели полного субградиентного спуска было проведено дополнительное исследование зависимости точности от значения минимизируемого функционала. Для этого после каждой итерации алгоритма производился подсчет точности текущей модели на отложенной выборке. Полученный график изображен на Рис.4.

Как видно из графика, при уменьшении функции потерь до какого-то порога точность на отложенной выборке возрастает. Однако при достижении этого порога (в данном случае порог — значение функционала, примерно равное 0.5) минимизация целевой функции не приносит вклада в увеличение точности, а даже наоборот, уменьшает её.

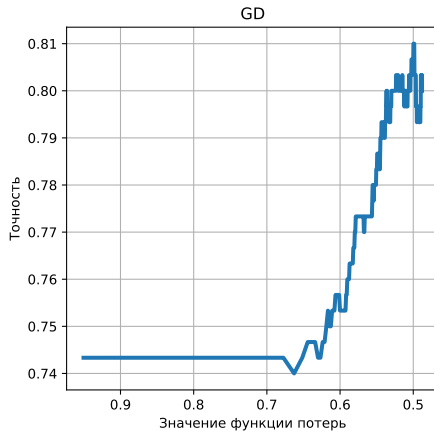


Рис. 4: Точность в зависимости от значения целевой функции

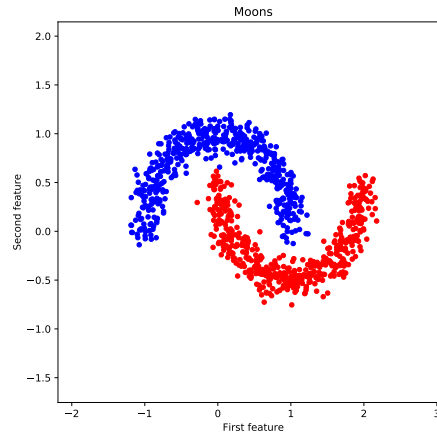


Рис. 5: Выборка «месяцы»

4 Поиск оптимальных значений параметров

В ходе выполнения данного эксперимента была реализована процедура поиска оптимальных значений параметров алгоритма с помощью кросс-валидации. Всюду в данном разделе количество фолдов, на которые делилась выборка при скользящем контроле, было установлено равным 3, а под «точностью по кросс-валидации» понимается среднее значение полученных точностей.

Поиск оптимальных значений параметров проводился для двух различных выборок:

1) для описанной в предыдущих экспериментах выборки, представляющей собой многомерные пересекающиеся шары в качестве трудно разделимой (эту выборку будем условно называть первой);

2) также для того, чтобы иметь возможность визуализировать полученные результаты, в качестве хорошо разделимых, но линейно неразделимых данных была взята выборка, представляющая собой два месяца, сгенерированная с помощью функции *make_moons* библиотеки *sklearn*. Данная выборка изображена на Рис. 5.

4.1 Параметр C

Для модели, решающую прямую задачу SVM, а также моделей субградиентного и стохастического субградиентного спусков был произведен подбор параметра C , отвечающий за то, насколько сильный накладывается штраф за ошибки классификации. В свою очередь для модели, реализующей метод PEGASOS, был подобран параметр λ . Результаты эксперимента для первой выборки приведены в Таб.7 и Таб.8.

	GD	SGD	primal
$C = 0.1$	0.74	0.756	0.843
$C = 1$	0.845	0.847	0.848
$C = 10$	0.847	0.846	0.844
$C = 100$	0.823	0.834	0.85

Таблица 7: Точность различных моделей (первая выборка)

	PEGASOS
$\lambda = 0.1$	0.83
$\lambda = 1$	0.784
$\lambda = 10$	0.742
$\lambda = 100$	0.737

Таблица 8: Точность модели PEGASOS (первая выборка)

Из полученных результатов можно сделать вывод, что подбор оптимальных параметров C и λ сильно повышает точность моделей, решающих безусловную задачу SVM субградиентными методами. В то же время модель, решающая прямую задачу SVM, на данной выборке показала примерно одинаковые значения точности для различных C .

Также заметим, что параметр λ модели PEGASOS обратно пропорционален параметру C других рассматриваемых моделей. Поэтому неудивительно, что наивысшие значения точностей, например, моделей GD и PEGASOS достигаются при $C = 1/\lambda = 10$.

Полученные значения точности моделей на выборке «месяцы» отображены в Таб.9 и Таб.10.

	GD	SGD	primal
$C = 0.1$	0.792	0.684	0.684
$C = 1$	0.795	0.794	0.787
$C = 10$	0.856	0.859	0.856
$C = 100$	0.878	0.880	0.878

Таблица 9: Точность различных моделей («месяцы»)

	PEGASOS
$\lambda = 0.1$	0.858
$\lambda = 1$	0.775
$\lambda = 10$	0.766
$\lambda = 100$	0.766

Таблица 10: Точность модели PEGASOS («месяцы»)

Все методы показали наилучшие точности при наибольших значениях C (для модели PEGASOS — наоборот, наименьшем). Поскольку для данной выборки характерно отсутствие линейной разделимости, для построения наилучшей разделяющей плоскости линейными моделями необходимо сильнее штрафовать ошибки классификации. Разделяющая плоскость, построенная моделью, решающей прямую задачу SVM, при $C = 100$ изображена на Рис.6.

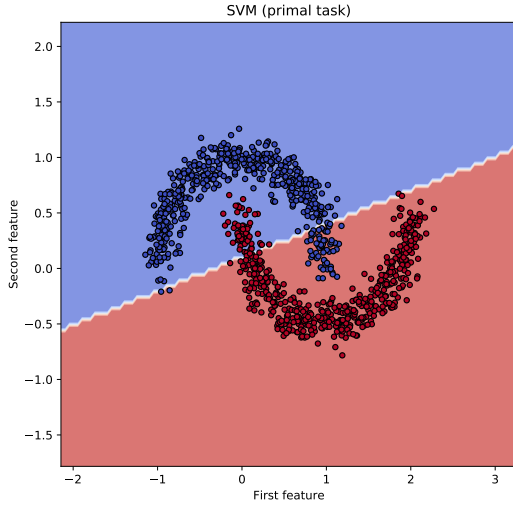


Рис. 6: Разделяющая плоскость модели *primal*, $C = 100$

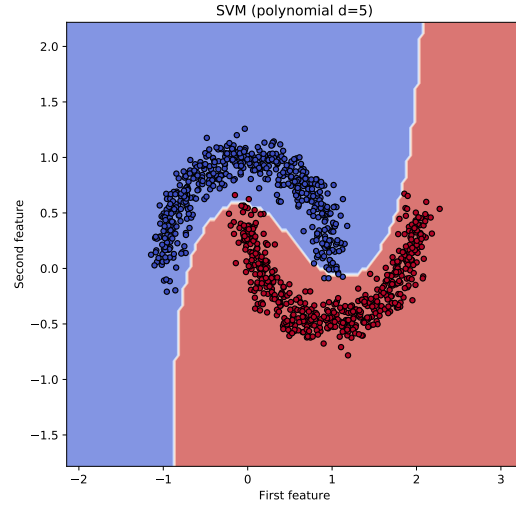


Рис. 7: Разделяющая плоскость модели с полиномиальным ядром, $d = 5$

4.2 Степень полиномиального ядра

Для модели, решающей двойственную задачу с полиномиальным ядром, был произведен поиск оптимального значения степени. Полученные результаты для двух рассматриваемых выборок приведены в Таб.8.

	$d = 2$	$d = 3$	$d = 4$	$d = 5$
Первая выборка	0.791	0.785	0.762	0.80
«Месяцы»	0.834	0.888	0.959	0.995

Таблица 11: Точность моделей с полиномиальными ядрами разных степеней

Для первой выборки результаты моделей с полиномиальными ядрами оказались несколько хуже, чем результаты линейных моделей. Вероятно, переход в новые спрямляющие пространства, соответствующие рассматриваемым степеням ядер, не привнес каких-либо преимуществ с точки зрения построения разделяющих гиперплоскостей.

Для выборки «месяцы» результаты диаметрально противоположные. При увеличении параметра d точность на кросс-валидации росла и достигла значения 99.5% при $d = 5$. Результат работы такой модели изображен на Рис.7. Таким образом, данная модель действительно отлично справляется с данной задачей классификации.

4.3 Ширина RBF ядра

Также для модели с RBF ядром был произведен поиск оптимального значения ширины ядра γ . Результаты приведены в Таб.12 и Таб.13.

	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$
Первая выборка	0.854	0.846	0.843

Таблица 12: Точность моделей с RBF ядрами разной ширины

	$\gamma = 0.1$	$\gamma = 1$	$\gamma = 10$
«Месяцы»	0.834	0.888	0.959

Таблица 13: Точность моделей с RBF ядрами разной ширины

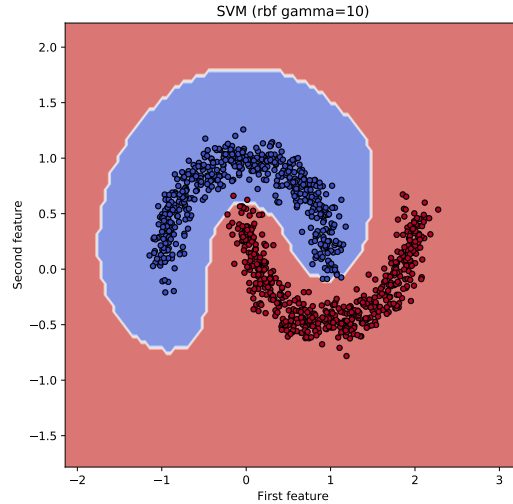


Рис. 8: Разделяющая плоскость модели *rbf*, $\gamma = 10$

Выбор проверяемых значений ширины ядра в случае первой выборки был обусловлен тем фактором, что при увеличении параметра γ точность модели стремительно падала (например, при $\gamma = 1$ точность составляла немногим более 50%). В целом, модели с RBF ядром показали хорошие результаты, полученные точности превосходят результаты моделей с полиномиальными ядрами и сопоставимы с результатами линейных моделей.

Для выборки «месяцы» данные модели также продемонстрировали отличные результаты. С увеличением параметра γ разделяющие плоскости модели чем-то напоминали результат работы алгоритма k-ближайших соседей. Результат работы модели со значением $\gamma = 10$ приведен на Рис.8.

5 Исследование скорости сходимости различных стратегий выбора длины шага в субградиентных методах

Темп обучения (или же длина шага субградиентного спуска) задается формулой $step = \alpha/k^\beta$, где k — номер итерации. Целью данного эксперимента было проведение сравнения влияния некоторых стратегий выбора параметров α и β на скорость сходимости и точность решения. В качестве точного было взято решение, полученное

с помощью метода внутренней точки для прямой задачи SVM. Сравнивая различные стратегии, под точностью решения будем понимать норму разности полученного вектора весов и точного решения. Исследование проводилось на уже известной выборке, представляющей собой многомерные пересекающиеся шары. Для устойчивости процесса обучения выборка была предварительно отмасштабирована.

Для моделей субградиентного и полного субградиентного спуска были исследованы по три различные стратегии выбора параметров α и β . Полученные графики зависимости функции потерь от номера итерации (эпохи) приведены на Рис.9-14.

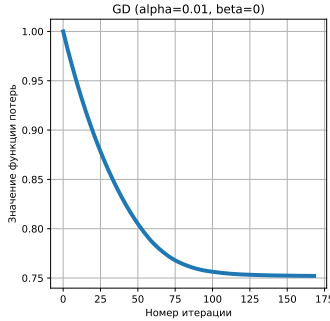


Рис. 9: GD $\alpha = 0.01$, $\beta = 0$

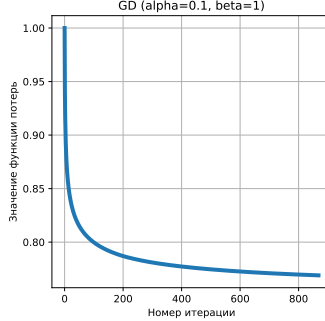


Рис. 10: GD $\alpha = 0.1$, $\beta = 1$

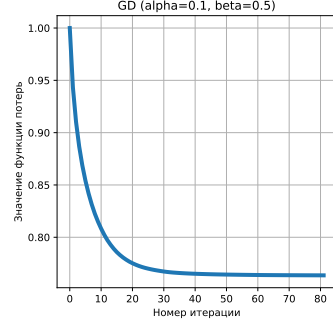


Рис. 11: GD $\alpha = 0.1$, $\beta = 0.5$

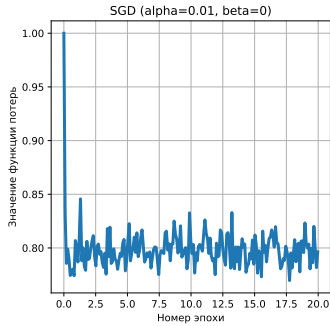


Рис. 12: SGD $\alpha = 0.01$, $\beta = 0$

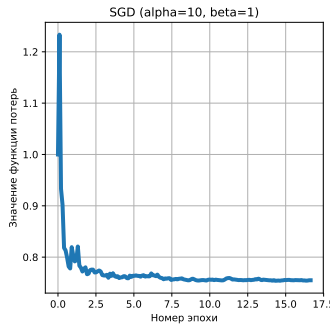


Рис. 13: SGD $\alpha = 10$, $\beta = 1$

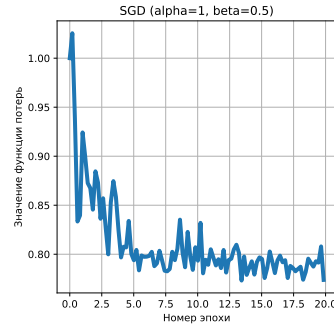


Рис. 14: SGD $\alpha = 1$, $\beta = 0.5$

	$\alpha = 0.01$, $\beta = 0$	$\alpha = 0.1$, $\beta = 1$	$\alpha = 0.1$, $\beta = 0.5$
GD	0.0178	0.1243	0.021

Таблица 14: Точность решения

	$\alpha = 0.01$, $\beta = 0$	$\alpha = 10$, $\beta = 1$	$\alpha = 1$, $\beta = 0.5$
SGD	0.1864	0.070	0.1749

Таблица 15: Точность решения

Модель субградиентного спуска с константной длиной шага (Рис.9) показала отличные результаты: модель максимально близко приблизилась к точному решению (Таб.14). В то же время модель с параметрами $\alpha = 0.1$, $\beta = 0.5$ (Рис.11) также хорошо справилась с поставленной задачей: для минимизации функционала модели потребовалось наименьшее количество итераций. При этом данная модель не сильно уступает по точности решения модели с константной длиной шага. Модель с параметрами $\alpha = 0.1$, $\beta = 1$ также приблизилась к точке минимума функционала, но по скорости сходимости и точности решения явно уступает двум другим моделям.

В случае стохастического субградиентного спуска модель с параметрами длины шага $\alpha = 0.01$, $\beta = 0$ (Рис.12), приблизившись к точке минимума, начала «прыгать»

вокруг неё. Это связано с тем, что длина шага оставалась постоянной с течением времени, и поэтому данной модели не удалось вплотную приблизиться к точному решению. Два другие набора параметров показали себя лучшим образом. Особенно хороших результатов достигла модель с параметрами $\alpha = 10$, $\beta = 1$ (Рис.14): за наименьшее количество эпох данная модель максимально приблизилась к точному решению.

На Рис.15 изображен график функции потерь метода PEGASOS в зависимости от номера итерации. График для данного метода похож на график зависимости функции потерь для стохастической модели с параметрами $\alpha = 10$, $\beta = 1$ (Рис.14). Это неудивительно, поскольку метод PEGASOS, хотя и минимизирует несколько отличный функционал, все равно остается некоторой модификацией метода стохастического субградиентного спуска, в котором параметр β равен 1.

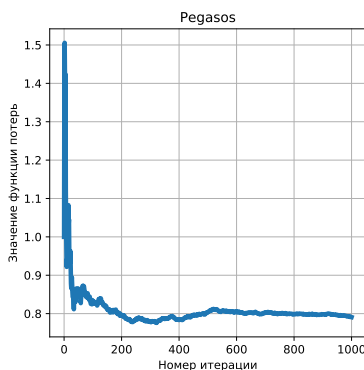


Рис. 15: Модель PEGASOS

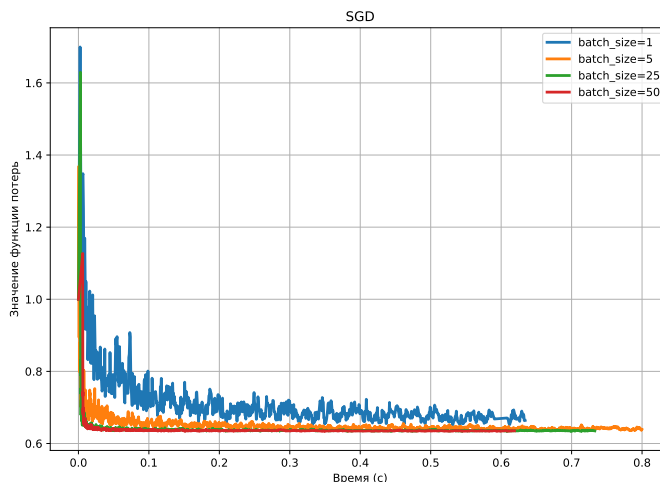


Рис. 16: Значения функции потерь в зависимости от времени работы

6 Исследование влияния размера подвыборки на скорость сходимости и точность решения

В ходе выполнения данного эксперимента необходимо было установить, как размер подвыборки, по которой считается субградиент в методе стохастического субградиентного спуска, влияет на процесс сходимости и точность решения. Максимальное количество итераций моделей было установлено равным 10 000. Напомним, что под точностью решения понимается норма разности полученного вектора весов и решения, принятого за точное (в данном случае, решение с помощью метода внутренней точки для прямой задачи SVM). Значения функции потерь с течением времени для различных размеров подвыборки приведены на Рис. 16.

Несмотря на то, длина шага уменьшалась с увеличением номера итерации, значения минимизируемого функционала для модели с $batch_size = 1$ сильно отличаются на соседних замерах, что ухудшает скорость сходимости данной модели.

Из приведенного графика наглядно видно, что наиболее быструю сходимость в данном случае обеспечивает модель с размером подвыборки равным 50: за отве-

денное количество итераций модель достигла наименьшего значения функционала. При этом данной модели потребовалось наименьшее количество времени, поскольку минимизация функционала была закончена досрочно, в связи с выходом по критерию остановки. Модели с бóльшим значением параметра *batch_size* чаще выходят по критерию остановки, поскольку такие модели всё более приближаются к модели полного субградиентного спуска, и график значений оптимизируемого функционала становится всё более гладким.

Также модель с *batch_size* = 50 показала наилучшую точность решения: 0.1469. Для сравнения, значения точности решений трех других моделей с *batch_size* = 1, 5, 25 составили соответственно 0.2013, 0.1527, 0.1591.

7 Демонстрация работы некоторых методов для двумерного случая

Данный эксперимент проводился на четырех различных типах данных:

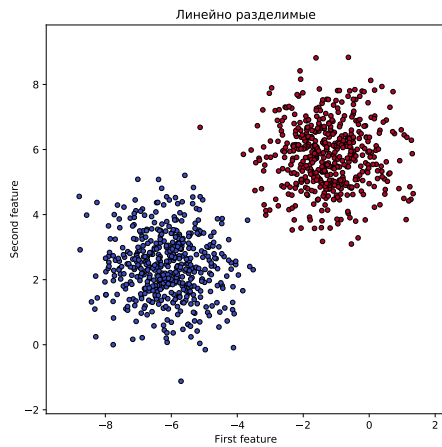


Рис. 17: Линейно разделимые

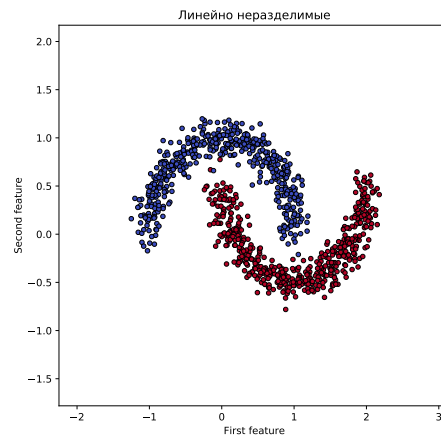


Рис. 18: Линейно неразделимые

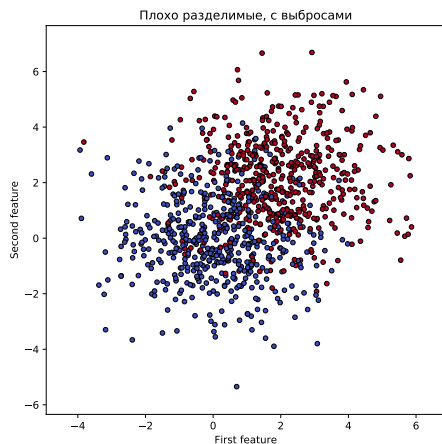


Рис. 19: Плохо разделимые

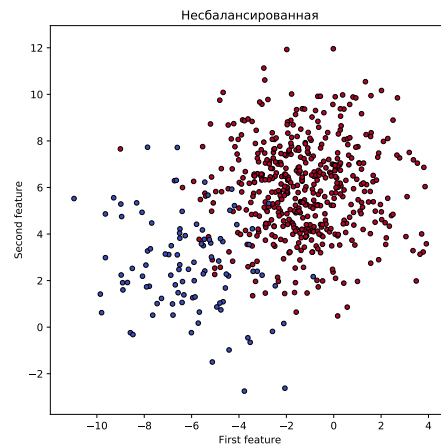


Рис. 20: Выборка с несбалансированными классами

Далее приведены требуемые в задании изображения, визуализирующие результаты работы различных моделей. Всюду в этом эксперименте опорные векторы подсвечены оранжевым цветом.

7.1 Линейно разделимые данные

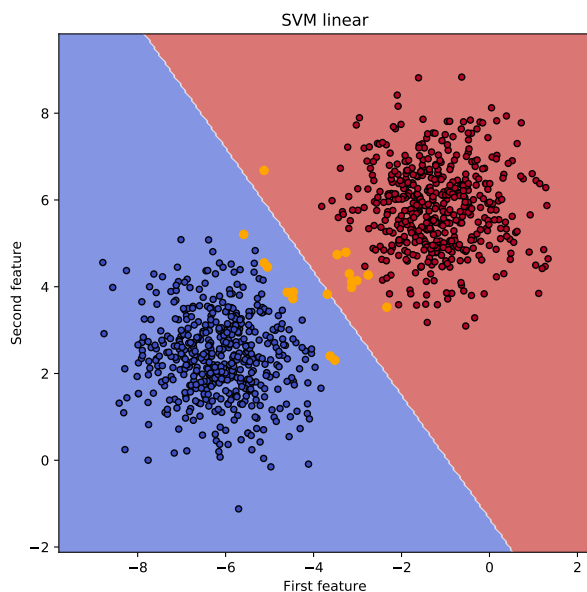


Рис. 21: Линейное ядро

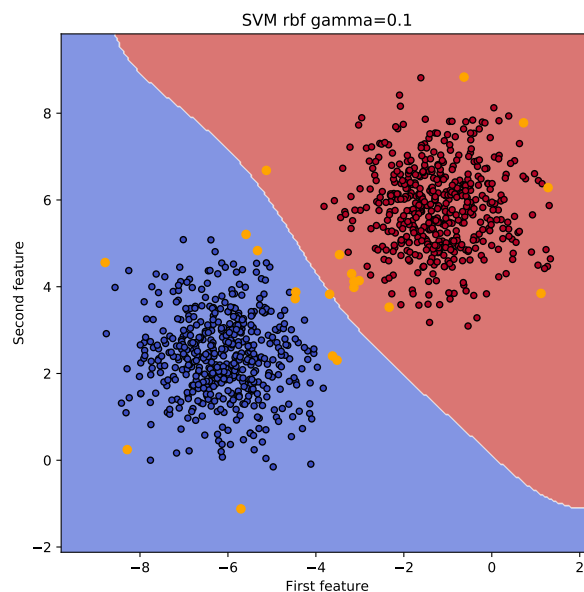


Рис. 22: RBF ядро, $\gamma = 0.1$

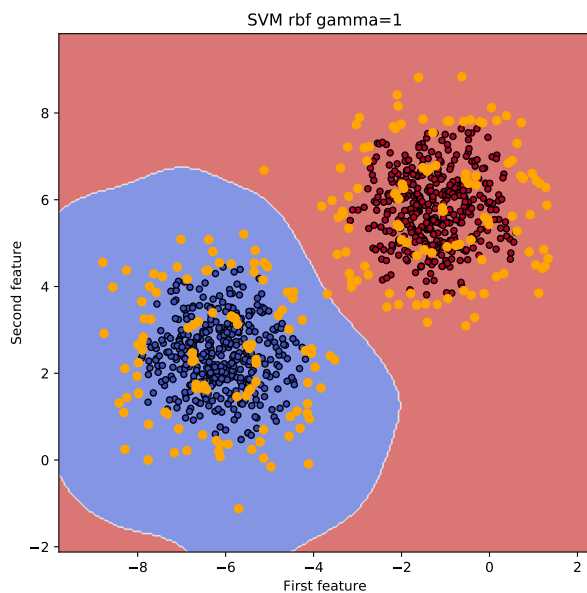


Рис. 23: RBF ядро, $\gamma = 1$

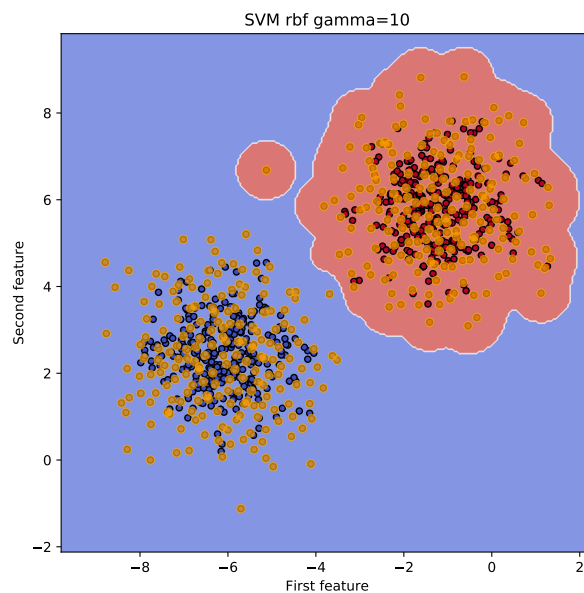


Рис. 24: RBF ядро, $\gamma = 10$

7.2 Хорошо разделимые, но линейно неразделимые данные

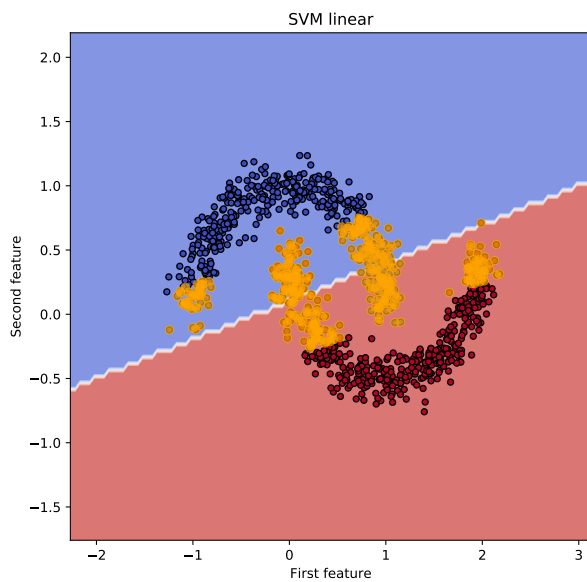


Рис. 25: Линейное ядро

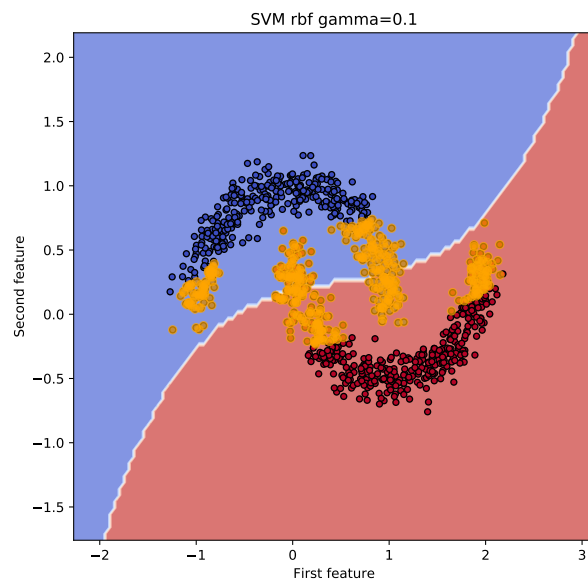


Рис. 26: RBF ядро, $\gamma = 0.1$

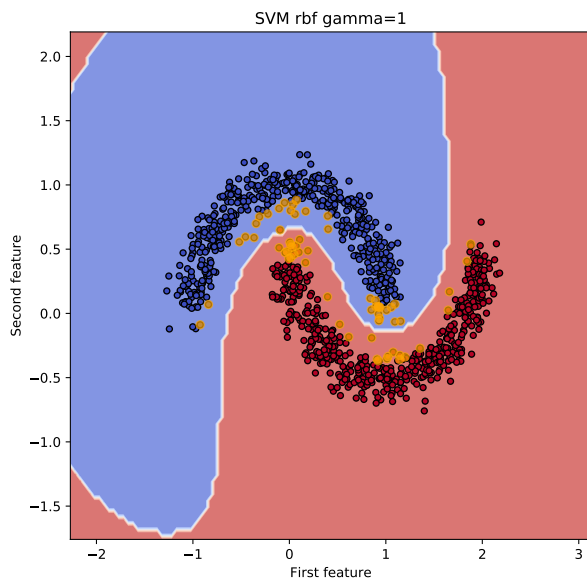


Рис. 27: RBF ядро, $\gamma = 1$

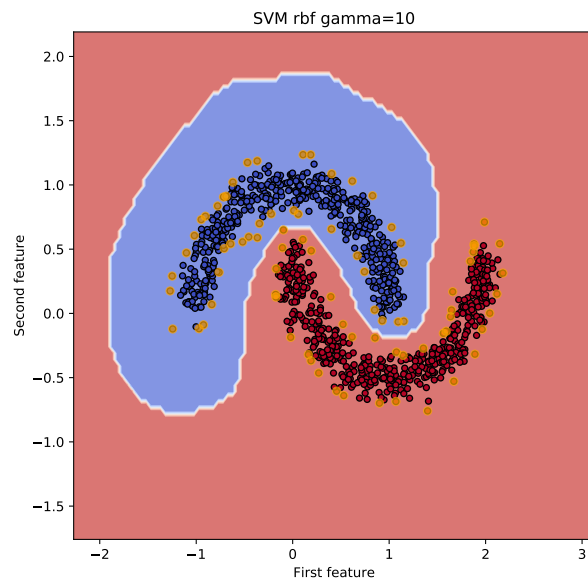


Рис. 28: RBF ядро, $\gamma = 10$

7.3 Данные с несбалансированными классами

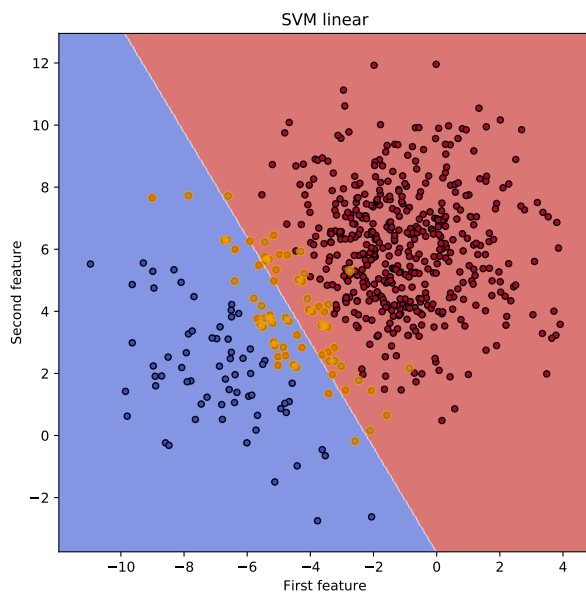


Рис. 29: Линейное ядро

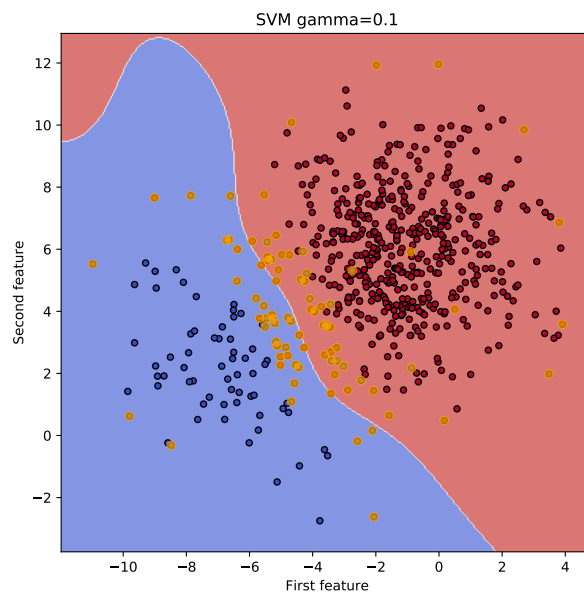


Рис. 30: RBF ядро, $\gamma = 0.1$

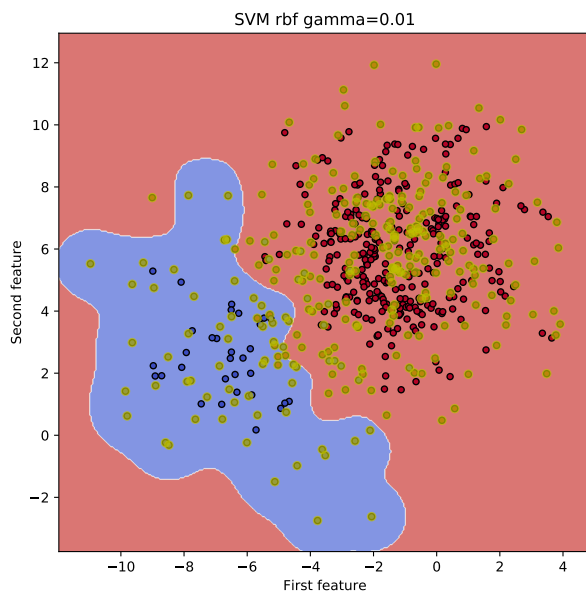


Рис. 31: RBF ядро, $\gamma = 1$

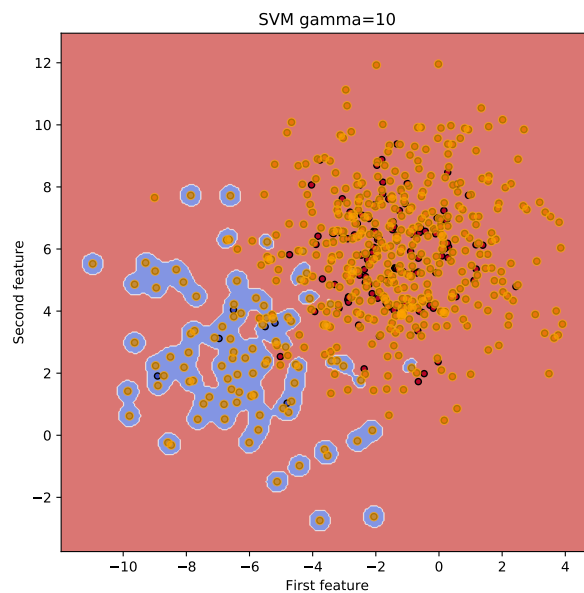


Рис. 32: RBF ядро, $\gamma = 10$

Как уже отмечалось ранее, модель SVM с RBF ядром при увеличении значения γ всё более напоминает модель, реализующую алгоритм k-ближайших соседей. На рассматриваемой выборке этот факт ещё более заметен.

7.4 Плохо разделимые данные с выбросами

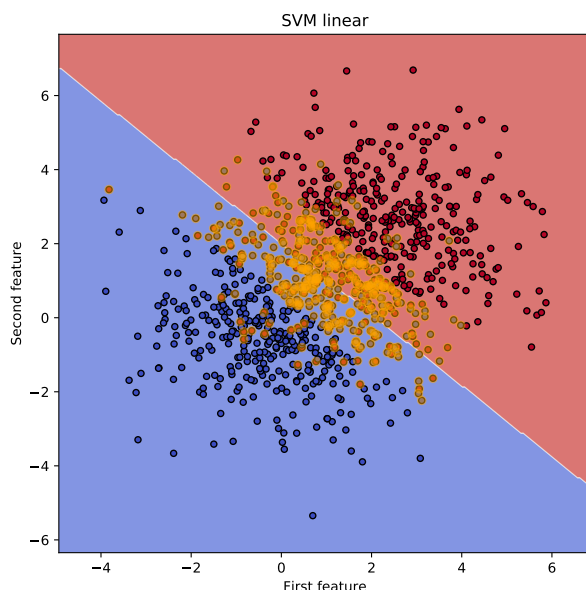


Рис. 33: Линейное ядро

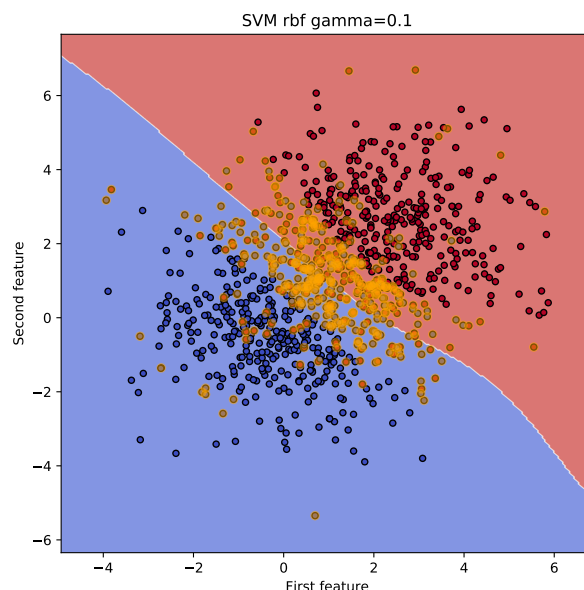


Рис. 34: RBF ядро, $\gamma = 0.1$

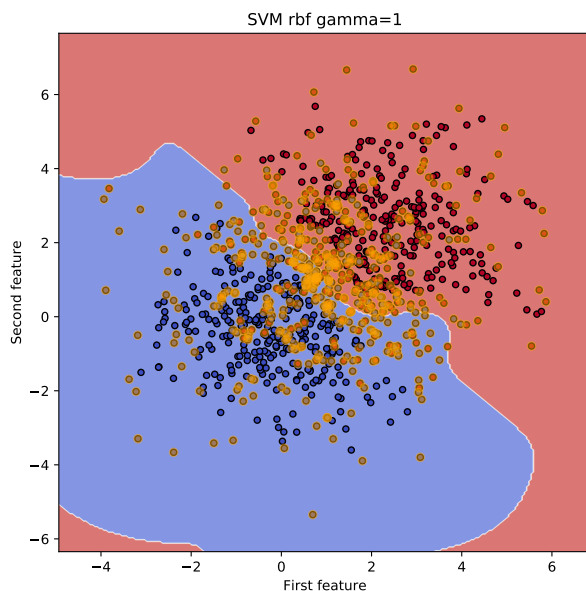


Рис. 35: RBF ядро, $\gamma = 1$

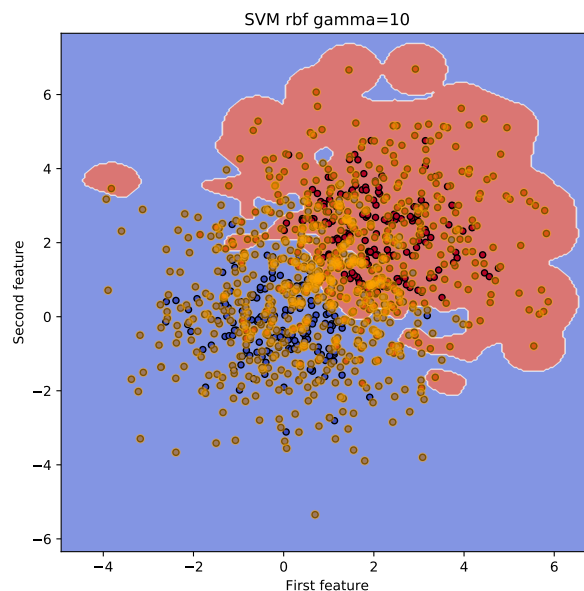


Рис. 36: RBF ядро, $\gamma = 10$

8 Заключение

Таким образом, в ходе выполнения практического задания «Метод опорных векторов» были написаны собственные реализации различных методов для решения задачи SVM, включающих в себя методы внутренней точки для решения прямой и двойственной задач, методы субградиентного и стохастического субградиентного

спусков для решения безусловной задачи SVM, а также метод PEGASOS. Реализованные методы были протестированы на разных по сложности разделения данных, были выявлены некоторые особенности моделей, а также произведен подбор различных параметров, влияющих на скорость сходимости и качество предсказания моделей.