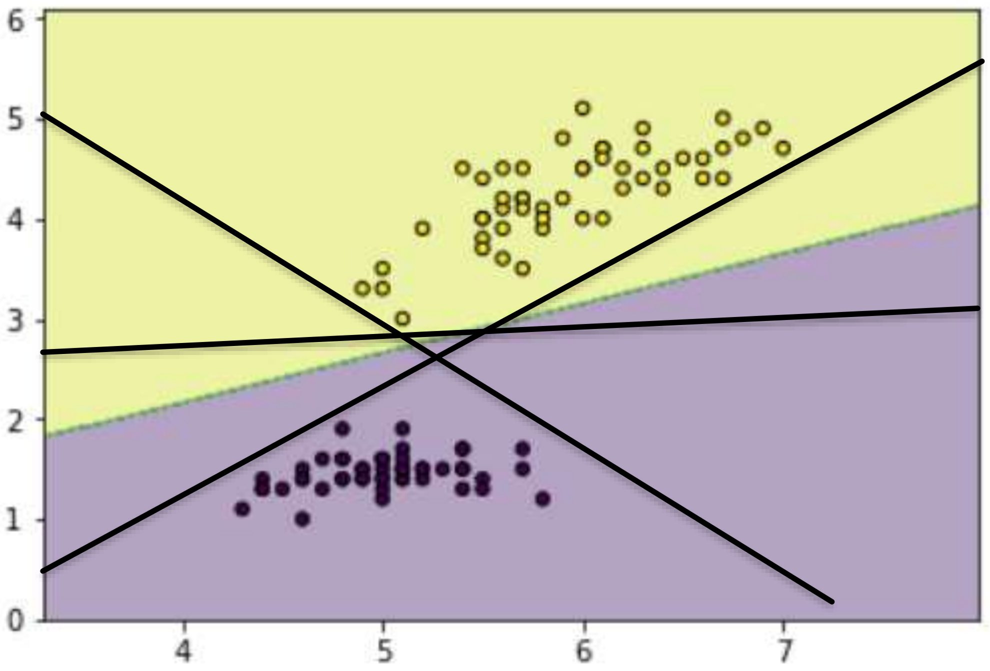
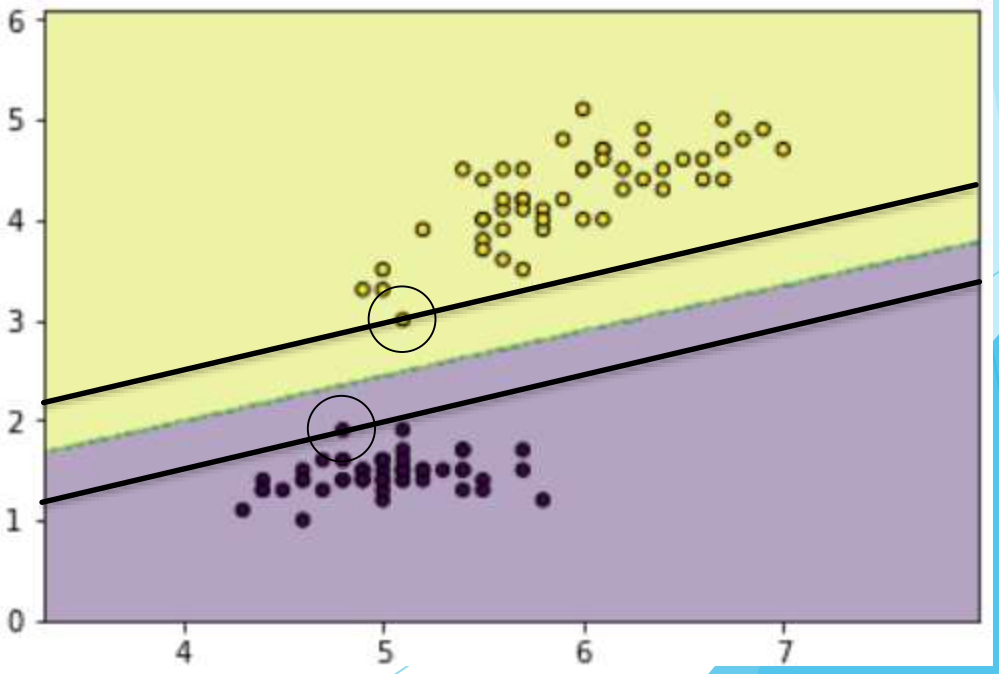
Метод опорных векторов

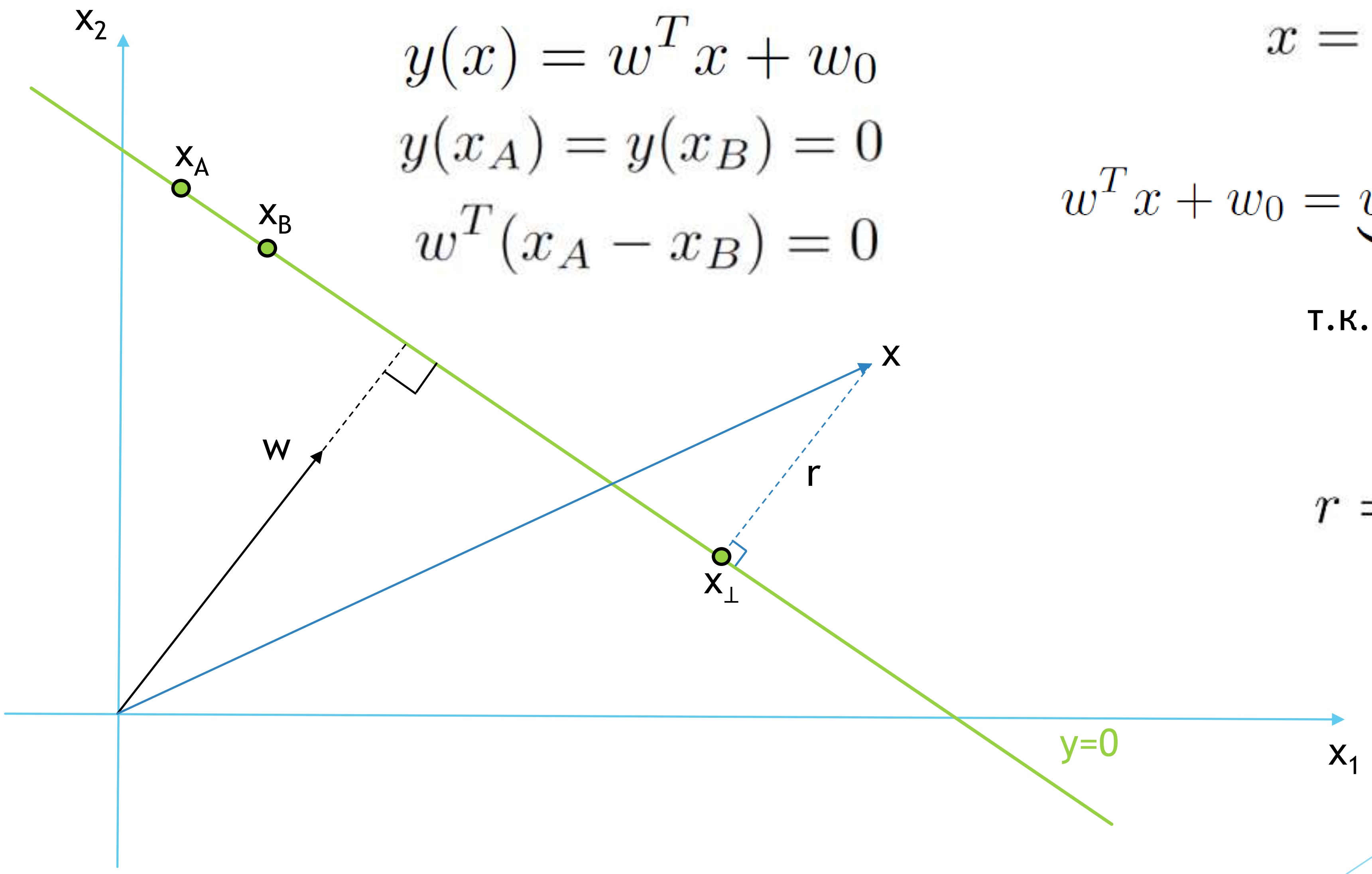
Метод опорных векторов (SVM – support vector machine)

- Персептрон – минимизация ошибок классификации
- SVM – максимизация зазора
- Зазор – расстояние между разделяющей гиперплоскостью и самыми близкими тренировочными образцами





Задача классификации



$$y(x) = w^T x + w_0$$

$$y(x_A) = y(x_B) = 0$$

$$w^T(x_A - x_B) = 0$$

$$x = x_\perp + r\frac{w}{\|w\|}$$

$$w^T x + w_0 = \underbrace{w^T x_\perp + w_0}_{=0} + r\frac{w^T w}{\|w\|}$$

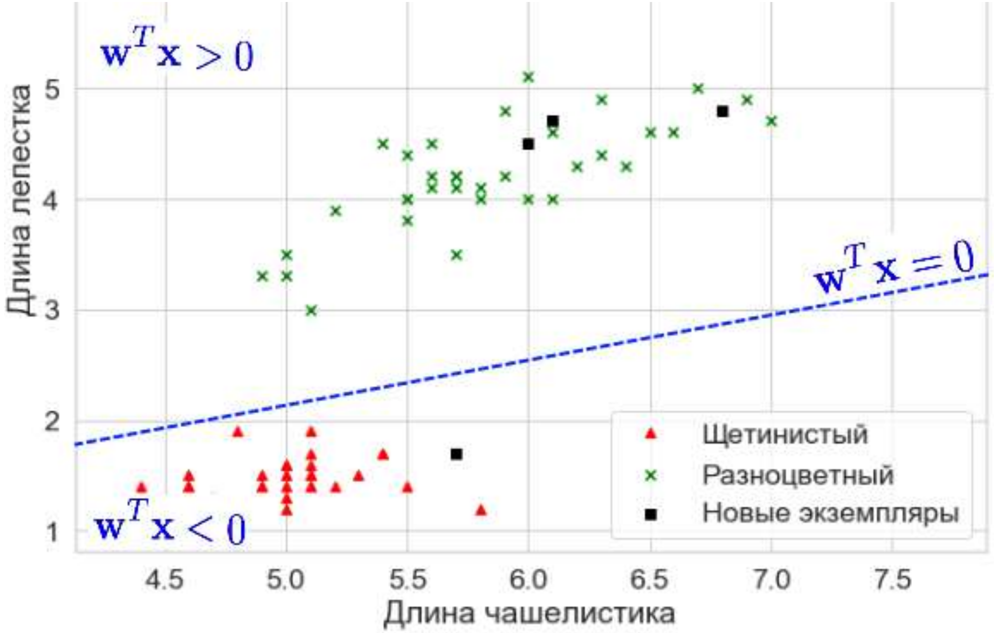
т.к. $x_\perp$ на плоскости y=0

$$r = \frac{y(x)}{\|w\|}$$

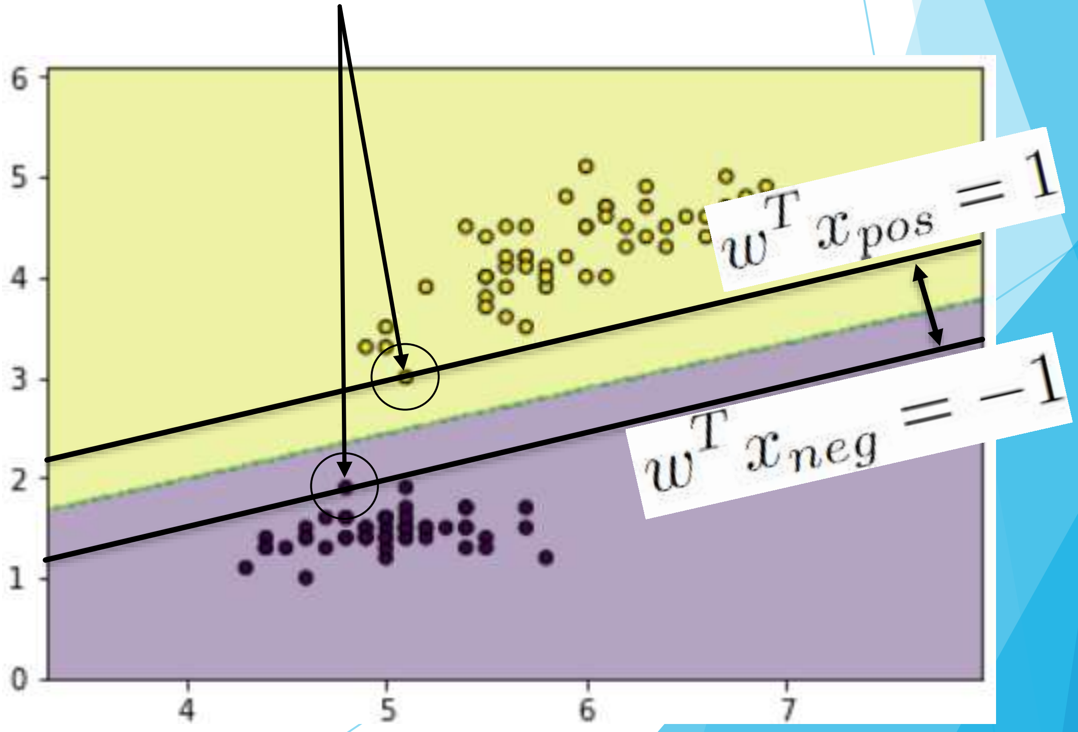
Bishop, Christopher. Pattern Recognition and Machine Learning, 2006

Метод опорных векторов

- Введем понятия положительной и отрицательной гиперплоскости, параллельные границе решения
- Шире граница – меньше ошибка обобщения



Опорные векторы – образцы, через которые проходят гиперплоскости

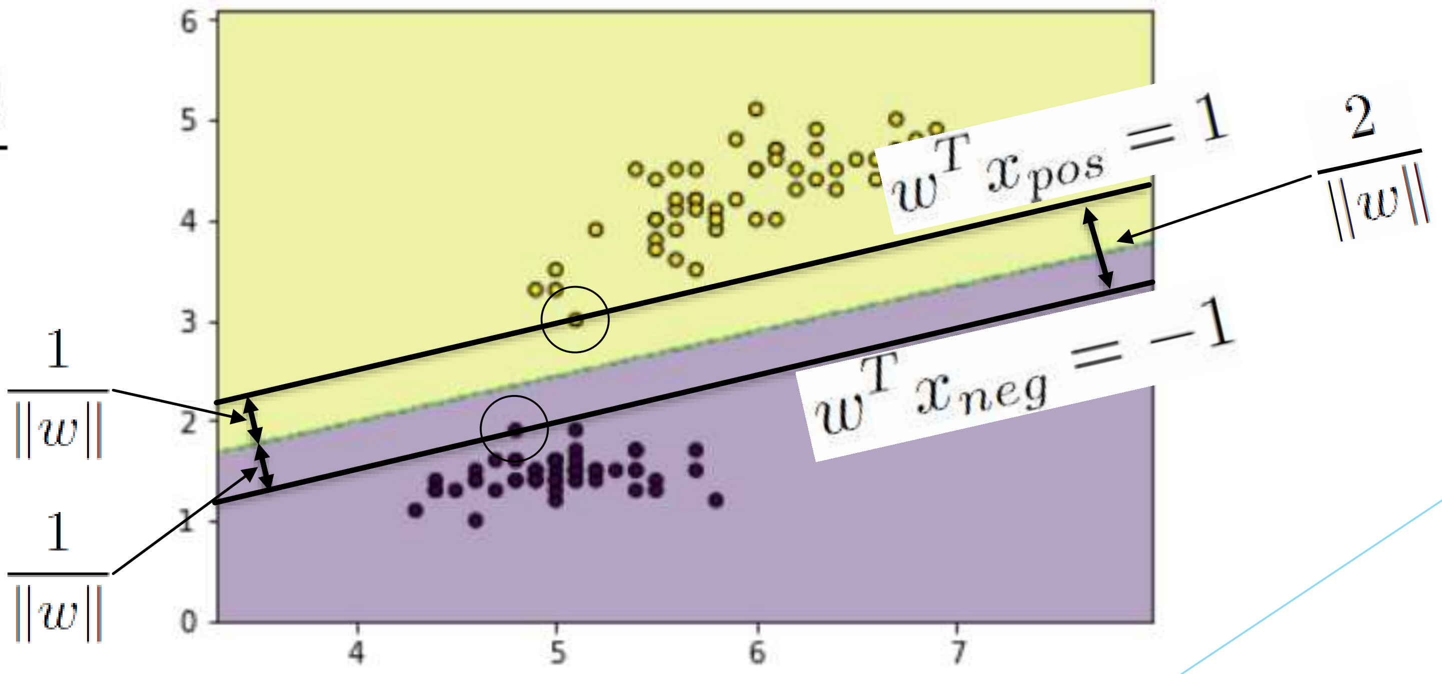


Метод опорных векторов

- Нормированная ширина границы

$$\frac{w^T(x_{pos} - x_{neg})}{\|w\|} = \frac{2}{\|w\|}$$

$$r = \frac{y(x)}{\|w\|}$$



Метод опорных векторов (с жестким зазором)

- Целевая функция – максимизация зазора, с учетом того, чтобы образцы классифицировались правильно

$$\frac{2}{||w||} \rightarrow max \qquad \text{равносильно} \qquad \frac{||w||}{2} \rightarrow min$$

- Задача квадратичного программирования

$$\begin{cases} \frac{||w||^2}{2} \rightarrow min \\ w^T x \geq 1, & \text{если } y^{(i)} = 1 \\ w^T x \leq -1, & \text{если } y^{(i)} = -1 \end{cases}$$

https://ru.wikipedia.org/wiki/Метод_опорных_векторов

Метод опорных векторов (линейно неразделимый случай) – SVM с мягким зазором

- Для каждого образца вводятся ослабленные переменные (slack), позволяющие нарушать зазор $\xi^{(i)}$

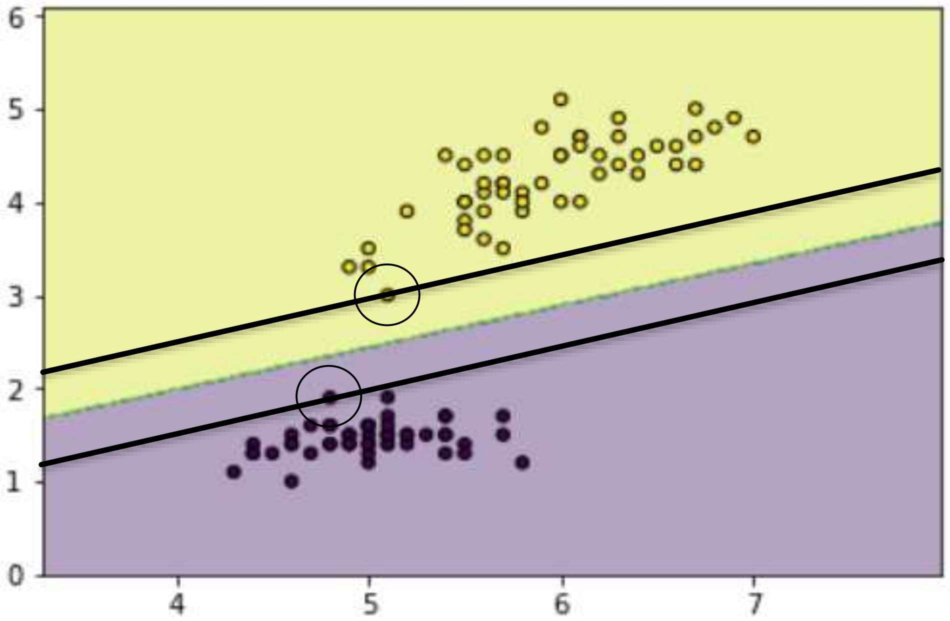
$$w^T x^{(i)} \geq 1 - \xi^{(i)}$$

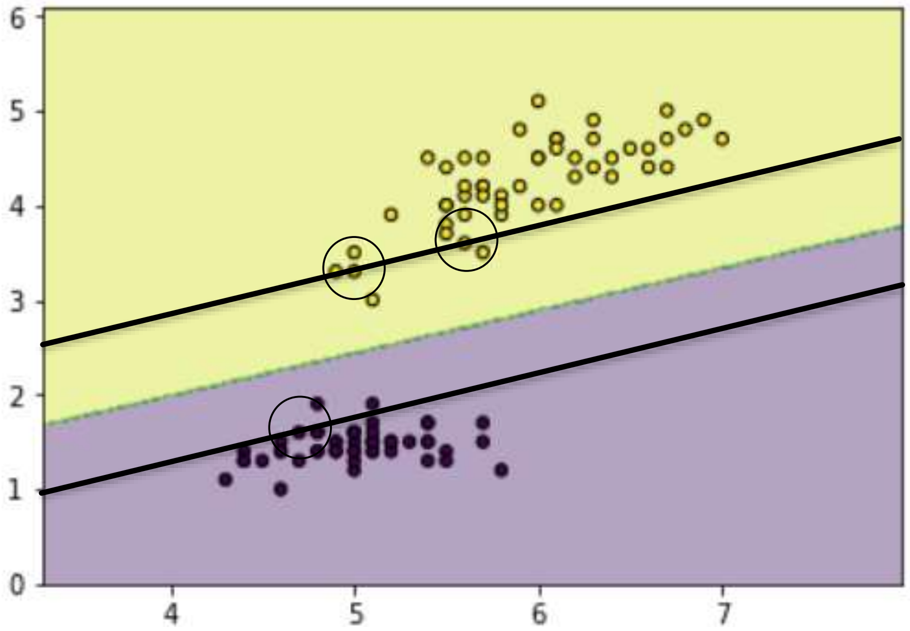
- Задача: расширить зазор минимизируя норму и уменьшить фиктивные переменные, чтобы сократить нарушения зазора

C - гиперпараметр

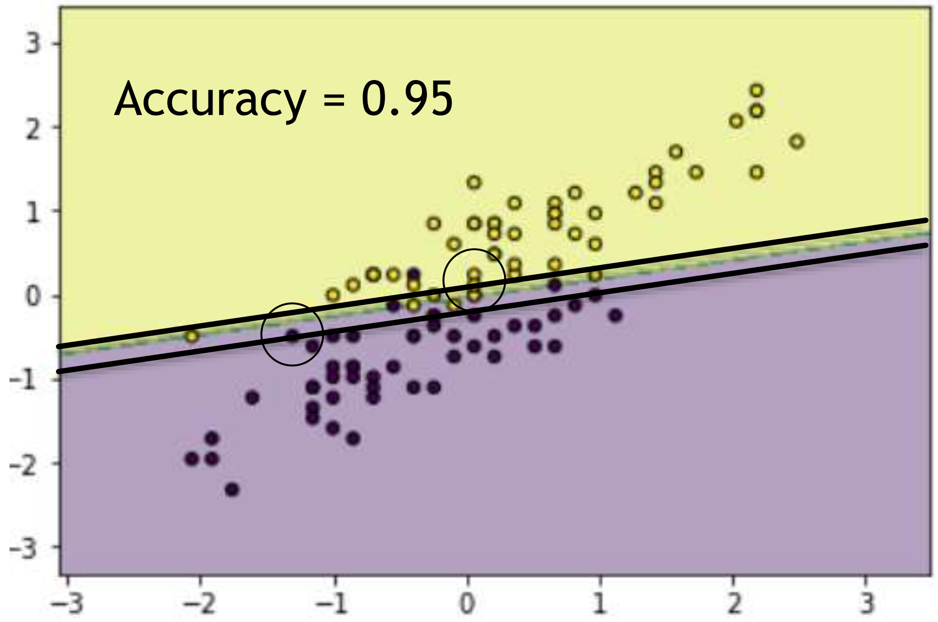
$$\begin{cases} \frac{\|w\|^2}{2} + C\sum \xi^{(i)} \to min \\ w^T x^{(i)} \geq 1 - \xi^{(i)}, & \text{если } y^{(i)} = 1 \\ w^T x^{(i)} \leq -1 + \xi^{(i)}, & \text{если } y^{(i)} = -1 \end{cases}$$

Метод опорных векторов (мягкий зазор)





Линейно нераз-
делимый случай:

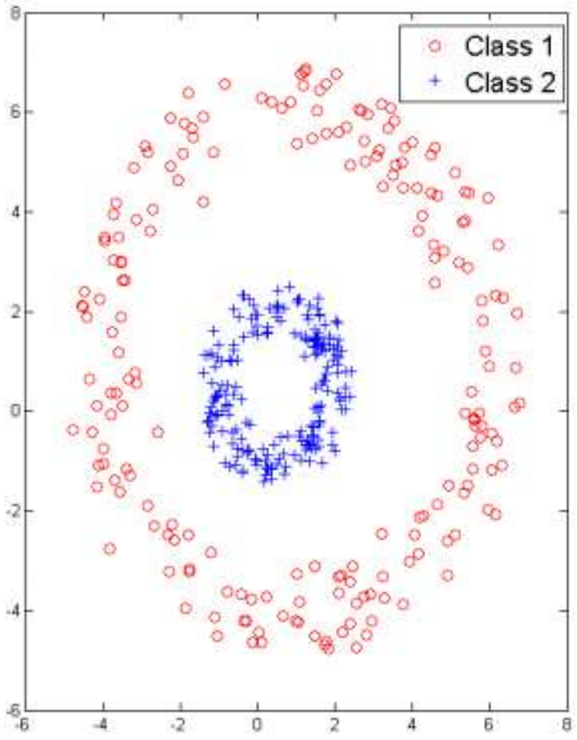


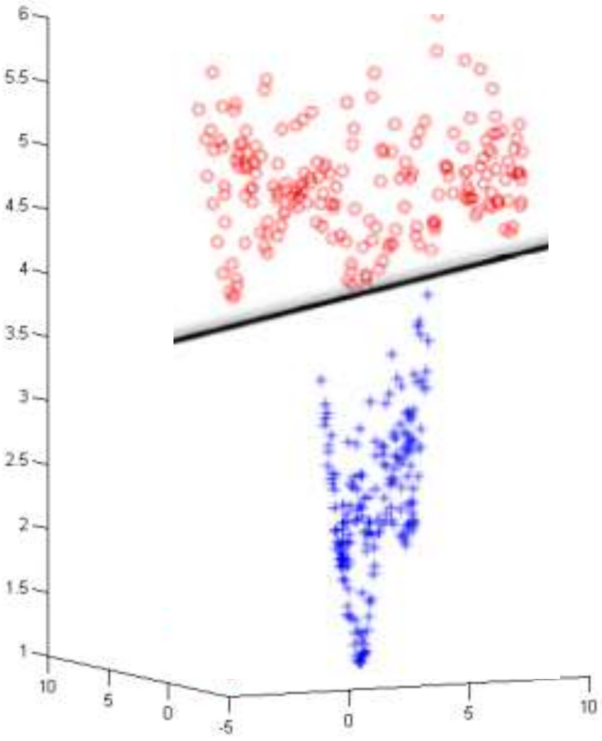
Линейно неразделимое множество

- Сделать нелинейную функцию отображения исходных признаков на пространство более высокой размерности

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$

- Произвести линейное разделение
- Отобразить границу обратным преобразованием





Pham, Trong-Ton, MODELE DE GRAPHE ET MODELE DE LANGUE POUR LA RECONNAISSANCE DE SCENES VISUELLES, 12.2010

Немного подробнее о решении задачи квадратичного программирования

$$\begin{cases} \frac{\|w\|^2}{2} + \mathrm{C}\sum \xi^{(i)} \to min \\ w^T x^{(i)} \geq 1 - \xi^{(i)}, & \text{если } y^{(i)} = 1 \\ w^T x^{(i)} \leq -1 + \xi^{(i)}, & \text{если } y^{(i)} = -1 \end{cases}$$

Запишем приведя ограничения к одному неравенству

$$\begin{cases} \frac{\|w\|^2}{2} + \mathrm{C}\sum \xi^{(i)} \to min \\ y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi^{(i)}, \\ \xi^{(i)} \geq 0 \end{cases}$$

https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM)

Немного подробнее о решении задачи квадратичного программирования

- Метод множителей Лагранжа

$$\begin{cases} f(w) \to min \\ \phi_i(w) = 0 \end{cases}$$

$$L(w, \lambda) = f(w) + \sum \lambda_i \phi_i(w)$$

- Теорема (условия Каруша-Куна-Таккера (обобщение на неравенства))

$$\begin{cases} f(w) \to min \\ \phi_i(w) \leq 0 \\ h_i(w) = 0 \end{cases}$$

Задача минимизации эквивалентна двойственной задаче поиска седловой точки функции Лагранжа

$$L(w, \lambda, \mu) = f(w) + \sum \lambda_i \phi_i(w) + \sum \mu_i h_i(w)$$

https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM)

Для нашей задачи

$$\begin{cases} \frac{\|w\|^2}{2} + \mathrm{C}\sum \xi^{(i)} \to min \\ y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi^{(i)}, \\ \xi^{(i)} \geq 0 \end{cases}$$

Существуют множители (λ, μ), что для функции Лагранжа выполняются условия

$$L(w, w_0, \xi, \lambda, \mu) = \frac{\|w\|^2}{2} - \sum \lambda_i \left(y^{(i)}\left(w^T x^{(i)} + w_0\right) - 1\right) - \sum \xi^{(i)} \left(\lambda_i + \mu_i - C\right)$$

$$\begin{cases} \frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial w_0} = 0, \frac{\partial L}{\partial \xi} = 0 \\ \xi_i \geq 0, \lambda_i \geq 0, \mu_i \geq 0 \\ \lambda_i = 0 \text{ либо } y^{(i)}\left(w^T x^{(i)} + w_0\right) = 1 - \xi^{(i)}, \\ \mu_i = 0 \text{ либо } \xi_i = 0 \end{cases}$$

Исходные и двойственные ограничения

Условия дополняющей нежесткости

Эквивалентная (двойственная) задача

$$\begin{cases} -L(\lambda) = -\sum_i \lambda_i + \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \left(x^{(i)}, x^{(j)}\right) \to \min_\lambda \\ 0 \le \lambda_i \le C \\ \sum_i \lambda_i y_i = 0 \end{cases}$$

Ядерный трюк

$$\left(x^{(i)}, x^{(j)}\right) \to \left(\phi(x^{(i)}), \phi(x^{(j)})\right) = k(x^{(i)}, x^{(j)})$$

Позволяет получить те же самые результаты, что и при добавлении дополнительных нелинейных признаков, но сами признаки не добавляя

Ядра

- Линейное

$$k\left(x^{(i)}, x^{(j)}\right) = \left(x^{(i)}, x^{(j)}\right)$$

- Полиномиальное ядро со степенью p

$$k\left(x^{(i)}, x^{(j)}\right) = \left(1 + \left(x^{(i)}, x^{(j)}\right)\right)^p$$

- Радиальные базисные функции (RBF)

$$k\left(x^{(i)}, x^{(j)}\right) = \exp\left(-\frac{\left(x^{(i)} - x^{(j)}\right)^2}{2\sigma^2}\right)$$

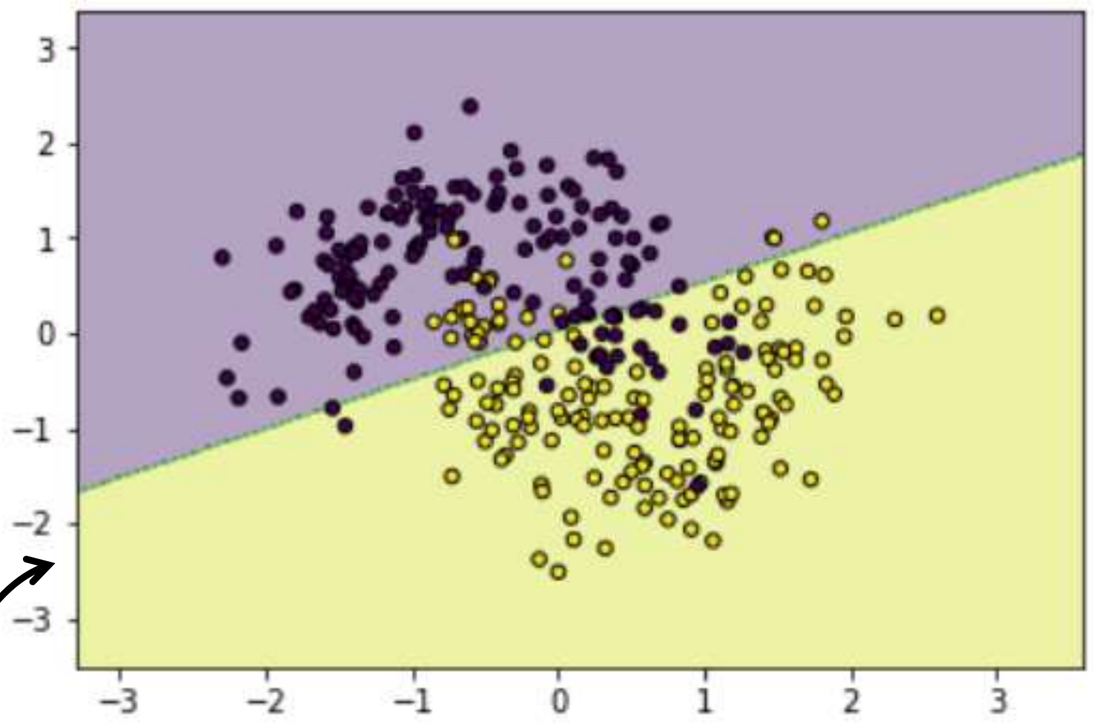
$$k\left(x^{(i)}, x^{(j)}\right) = \exp\left(-\gamma \|x^{(i)} - x^{(j)}\|^2\right) \qquad \gamma = \frac{1}{2\sigma^2}$$

Гипер-
параметр

RBF ядро – функция подобия образцов. 1 – подобны, 0 - нет

Линейно неразделимое множество

```
from sklearn.datasets import make_moons
X,y = make_moons(n_samples=300, noise=0.3)
```



Perceptron = 0.67

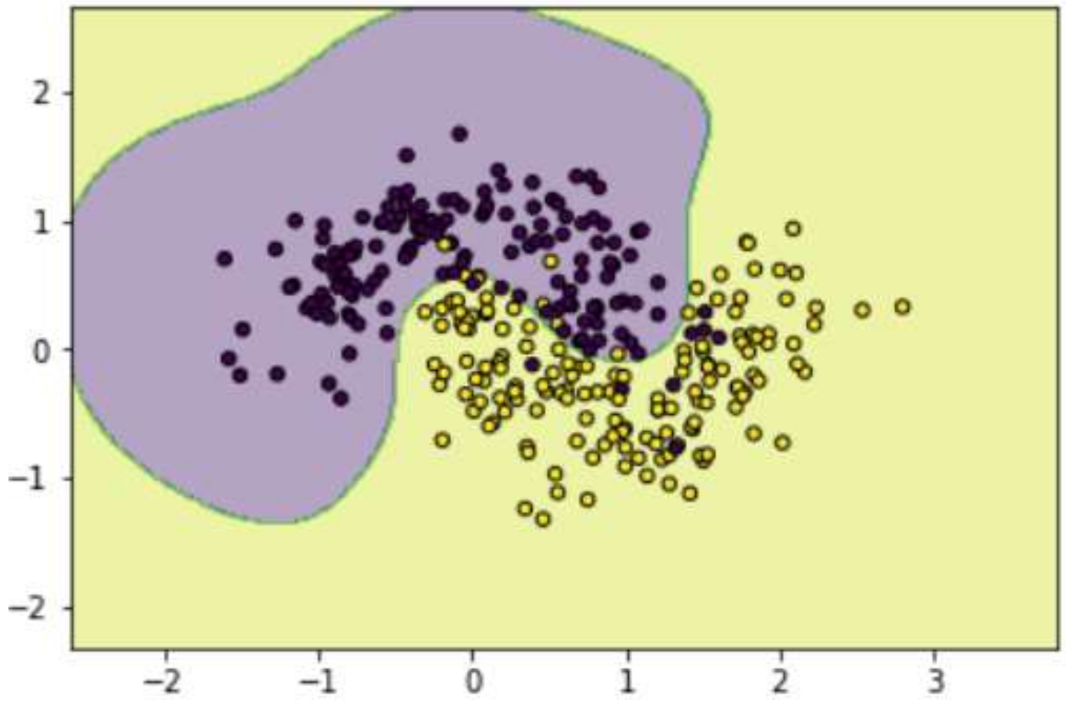
LogisticRegression = 0.84

LinearSVC = 0.84

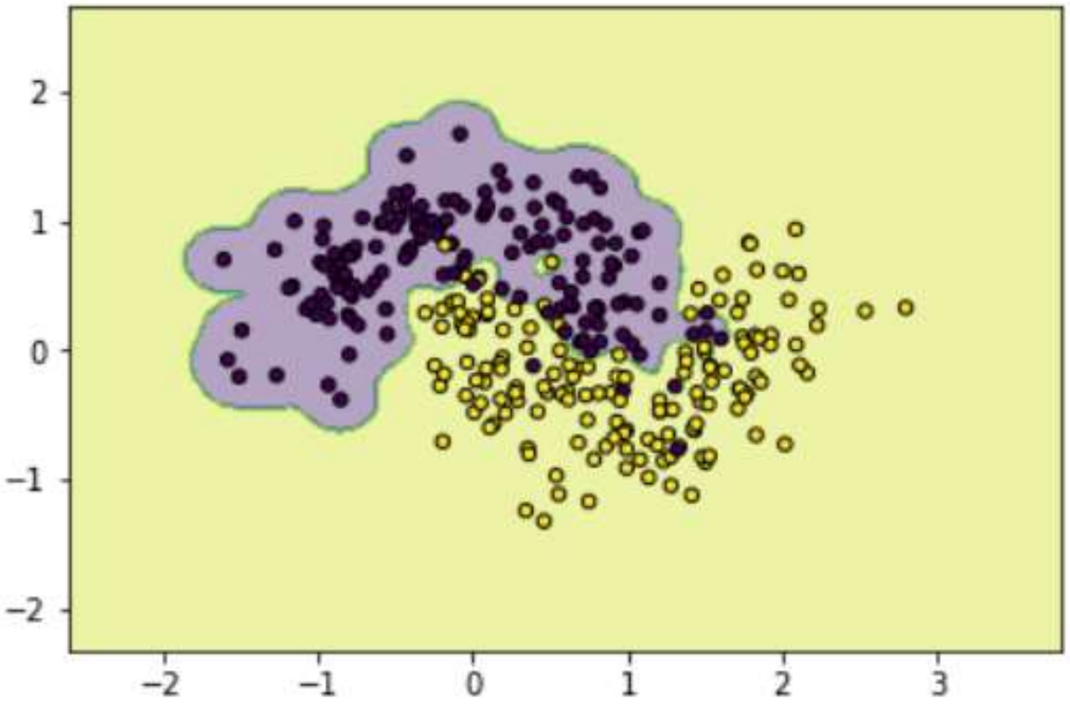
LogisticRegression vs SVM

- Логистическая регрессия и линейный метод опорных векторов дают примерно одинаковые результаты.
- Логистическая регрессия более подвержена выбросам, тогда как SVM сосредоточен на точках, ближайших к границе, хотя если границы SVM будут строиться на шуме – тоже плохо
- Логистическая регрессия имеет более простую модель и ее можно легко обновлять, плюс дает вероятность для каждого класса

Линейно неразделимое множество
Ядро RBF

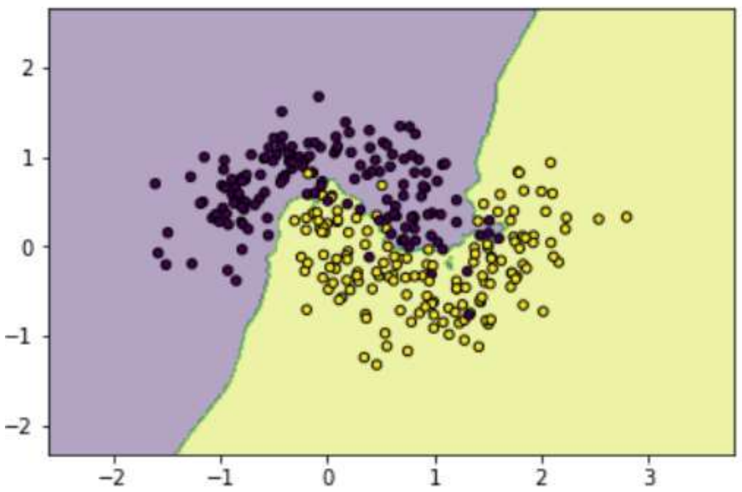


SVC(kernel='rbf', gamma=2) = 0.93

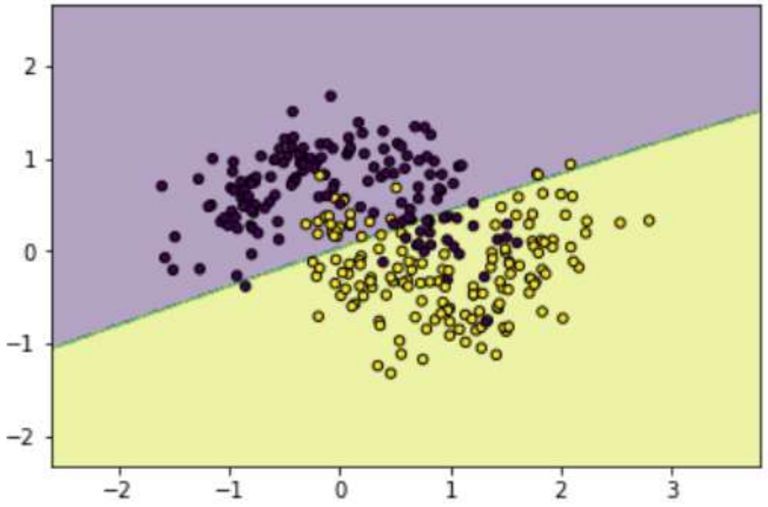


SVC(kernel='rbf', gamma=50) = 0.88

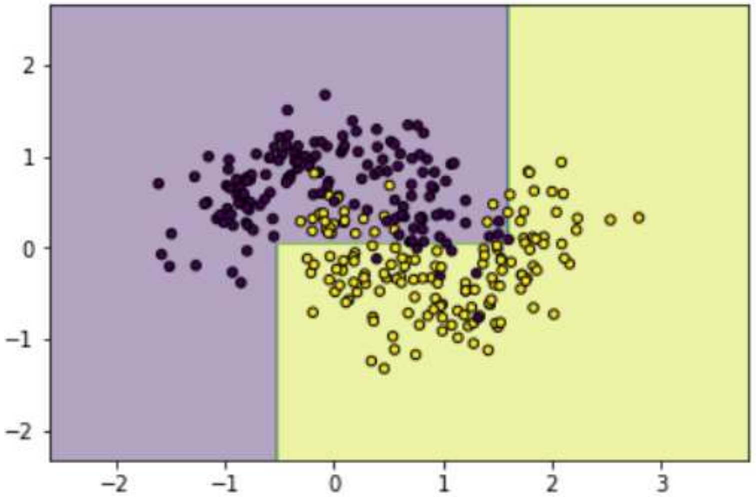
Линейно неразделимое множество
Рассмотренные ранее методы



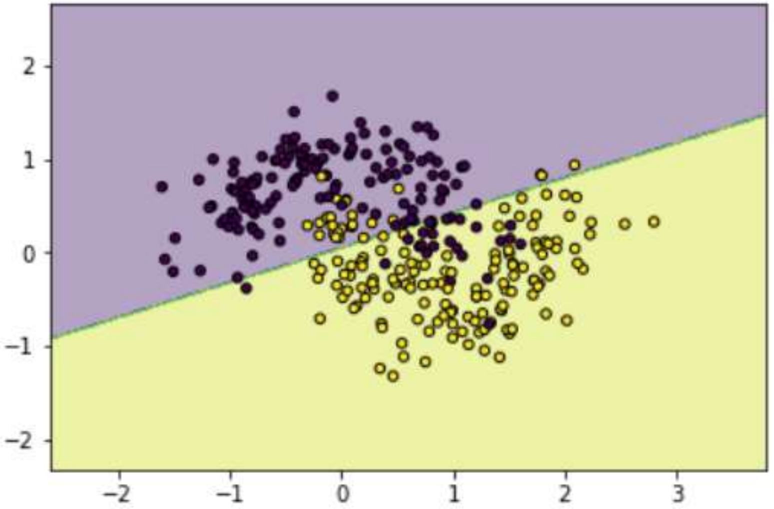
kNN(k=5) = 0.92



GaussNB = 0.83



DecisionTreeClassifier (max_depth=2) = 0.88



LogisticRegression = 0.84

Реализации в sklearn

- linear_model.SGDClassifier с параметрами по умолчанию дает линейный SVM
  - Быстро обучается и поддерживает внешнее обучение
  - Не поддерживает ядерный трюк
- svm.LinearSVC оптимизированный алгоритм для линейного случая
  - Быстро обучается, не поддерживает внешнее обучение
  - Не поддерживает ядерный трюк
- svm.SVC алгоритм, полноценно поддерживающий метод
  - Обучается медленно, не поддерживает внешнее обучение
  - Поддерживает ядерный трюк

Все реализации чувствительны к масштабированию признаков.
Перед работой необходимо провести стандартизацию

Гиперпараметры: параметр регуляризации C и для алгоритма SVC параметр gamma

Вопросы

- Главная идея метода опорных векторов
- Что такое опорный вектор?
- Оказывают ли влияние на формирование классификатора образцы, находящиеся далеко от границы?
- Где хранятся знания, на основе которых алгоритм выдаёт решения?
- Что такое ядерный трюк?