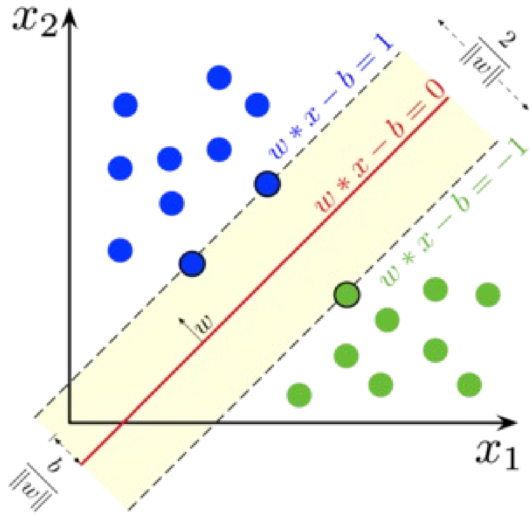
Дополнительное задание 2

Рассмотрим метод опорных векторов (SVM). Это алгоритм машинного обучения, применяемый для задач линейной и нелинейной классификации и регрессии. Идея метода состоит в создании гиперплоскости, которая наилучшим образом разделяет объекты выборки. Алгоритм строится из следующего предположения: что чем больше расстояние (margin) между разделяющей гиперплоскостью и объектами различных классов, тем менее возможна ошибка классификации.

Рассмотрим бинарную задачу классификации: X→Y, где X = R^n, Y = {-1, 1}. Пусть задана обучающая выборка пар: S=(x⃗ _i,y_i). Необходимо построить алгоритм классификации a(x⃗):X→Y.

Любая гиперплоскость может быть записана как множество точек, удовлетворяющих уравнению w^T*x - b = 0, где w - вектор нормали к гиперплоскости.
Пусть у нас есть гиперплоскость, которая делит данные на 2 класса C1 и C2, тогда эти объекты располагаются по разные стороны от гиперплоскости, то есть

$$\begin{cases} \langle \vec{w}, \vec{x} \rangle - b > 0, & \forall x \in C_1 \\ \langle \vec{w}, \vec{x} \rangle - b < 0, & \forall x \in C_2 \end{cases}$$



Отступ (зазор, margin) — свойство, показывающее насколько объект "погружён" в свой класс, насколько типичным представителем класса он является.
Чем меньше отступ, тем ближе объекты находятся к гиперплоскости и тем самым вероятность ошибки тоже больше. Если отступ отрицательный, то классификатор допустил ошибку. Для линейного классификатора отступ определяется уравнением:
M(w⃗) = yi(⟨w⃗, x⃗_i⟩−b)

Линейный SVM:
Предположим что выборка линейно разделима, тогда в качестве алгоритма классификации будем использовать линейный пороговый классификатор a : X → Y вида
a(x) = sign((w⃗, x) − c)
Так как линейная выборка разделима, то существует гиперплоскость, у которой отступ отступ до каждого объекта положителен и тогда построим такую гиперплоскость, что объекты обучающей выборки находились на наибольшем расстоянии от неё. Эти краевые гиперплоскости можно описать уравнениями: w^T*x - b = 1 и w^T*x - b = -1.
Для того, чтобы определить ширину полосы, введём x+ и x− (произвольных опорных объекта классов 1, -1). Тогда ширина полосы выражается как проекция вектора x+, x- на нормаль к гиперплоскости w.

$$\frac{\langle \vec{x}_+ - \vec{x}_-, \vec{w} \rangle}{\|w\|} = \frac{\langle \vec{x}_+, \vec{w} \rangle - \langle \vec{x}_-, \vec{w} \rangle - b + b}{\|w\|} = \frac{(+1)\Big(\langle \vec{x}_+, \vec{w} \rangle - b\Big) + (-1)\Big(\langle \vec{x}_-, \vec{w} \rangle - b\Big)}{\|w\|} =$$

$$= \frac{M_+(\vec{w}, b) + M_-(\vec{w}, b)}{\|w\|} = \frac{2}{\|w\|} \to \max \;\Rightarrow\; \|w\| \to \min$$

Задача в терминах квадратичного программирования:
$(w, w) \to \min$
$M_i(x_i) >= 1$

Нелинейный SVM:
В жизни линейный SVM не встречается особо, поэтому немного изменим немного правила игры и позволим объектам попадать на область другого класса и введем некий штраф за ошибку. Тогда можно записать задачу квадратичного программирования таким образом:
$0.5 * (w, w) + C * \mathrm{sum}(\varepsilon_i) \to \min$
$M_i(x_i) >= 1 — \varepsilon_i$
$\varepsilon_i >= 0$

В качестве примера нелинейного SVM можно рассмотреть следующую окружность:

Как видно провести линию не получится, тогда воспользуемся нелинейными функциями для преобразования данных в более высокоразмерное пространство. В данном случае 2D → 3D, и тогда уже можем поделить.