## Диагностика и кривые обучения

### Цель работы

Познакомиться на практике с явлениями недо и переобучения, научиться их выявлять путем диагностики, составлять и применять рекомендации по улучшению эффективности систем машинного обучения

### Задания для выполнения

1. Загрузите датасет breast\_cancer.
2. Выберите два первых признака. Таким образом мы уменьшим размерность задачи. У нас была задача множественной классификации, а стала - бинарная.
3. Разделите датасет на обучающую и тестовую выборки в пропорции 80-20.
4. Постройте классификатор на опорных векторах с гауссовым ядром с параметром регуляризации C=0.01. Выведите на экран значение точности на тестовой и обучающей выборке. Сделайте предварительный вывод о степени обученности модели.
5. Постройте графически зависимость тестовой и обучающей точности от размера обучающей выборки. Прокомментируйте получившийся результат. Сделайте вывод о том, недообучается или переобучается модель? В зависимости от этого выберите варианты, которые могут привести к увеличению точности модели (обычно подходит больше одного варианта):
   1. собрать больше объектов наблюдения;
   2. провести обучение на меньшем объеме данных;
   3. исключить часть признаков из модели;
   4. добавить новые признаки в модель;
   5. добавить полиномиальные признаки;
   6. попробовать более сложные модели;
   7. попробовать более простые модели;
   8. увеличить регуляризацию модели;
   9. уменьшить регуляризацию модели;
6. Проверьте выдвинутую гипотезу и попробуйте улучшить модель в соответствии с выбранными рекомендациями. Проверьте, увеличивается ли тестовая точность. Сделайте вывод.
7. Повторите предыдущий пункт для различных значений C в диапазоне от 0,0001 до 1 000 000. Обратите внимание на шкалу по вертикальной оси. Для каких значений тестовая точность оптимальна? Какие значения дают недообучение? Какие - переобучение?
8. Постройте график зависимости тестовой и обучающей точности от значения регуляризационного параметра. Сделайте вывод о зависимости степени регуляризации на точность модели.
9. Найдите методом подбора оптимальное значение параметра регуляризации (примерное).
10. Повторите два предыдущих пункта для метода опорных векторов с линейным ядром. Сравните получившиеся результаты. Сделайте вывод о сравнительной эффективности двух этих моделей.

### Методические указания

Для начала работы нам потребуется импортировать необходимые библиотеки:

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import load\_breast\_cancer

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

В первую очередь загрузим датасет, выберем нужные данные и выделим тестовую выборку для оценки точности модели:

breast\_cancer = load\_breast\_cancer()

X = breast\_cancer.data

y = breast\_cancer.target

X\_2d = X[:, :2]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_2d, y)

Затем построим первый классификатор. Оценка его точности даст нам приблизительную оценку уровня общей эффективности простых неоптимизированных моделей:

clf = SVC(C=0.01)

clf.fit(X\_train, y\_train)

print(clf.score(X\_train, y\_train))

print(clf.score(X\_test, y\_test))

Сами по себе данные значения мало о чем могут нам сказать. Понятно, что хотелось бы получить как можно более высокие значения. Но скорее всего вы не получите идеальную классификацию, а оценка точности будет лежать в пределах от 0 до 1. Для разных задач разная точность может считаться приемлемой. Для того, чтобы понять, можно ли повысить точность усовершенствованием модели нужно понять, какова степень обученности данной модели. Какую-то информацию может нам дать разница между этими значениями.

Если тестовая точность близка к обучающей, это скорее всего означает, что модель недообучается и при использовании более сложных моделей (или при снижении регуляризации, что эффективно то же самое) есть надежда улучшить данный показатель.

Если тестовая точность значительно ниже обучающей, это свидетельствует о потенциальном переобучении и потере способности к переобучению. Тогда для совершенствования модели нужно действовать в обратном направлении: использовать более простые модели или увеличивать регуляризацию данной.

Если же тестовая точность выше обучающей, это говорит о наличии случайных ошибок выборки тестовой модели. Такие ошибки неизбежно присутствуют и могут зашумлять результаты диагностики. Поэтому чаще всего для таких работ производится перекрестная проверка, которую мы будем проходить позже.

Для более точного анализа попробуем выяснить, что происходит с точностью при обучении на части обучающей выборки в зависимости от количества используемых точек. Для этого организуем обучение в цикле (обратите внимание на использование функции linspace):

training\_scores = []

test\_scores = []

numbers = []

for i in np.linspace(20, len(X\_train), 20):

clf = SVC(C=10, kernel='rbf')

X = X\_train[:int(i)]

y = y\_train[:int(i)]

clf.fit(X, y)

numbers.append(int(i))

training\_scores.append(clf.score(X\_train, y\_train))

test\_scores.append(clf.score(X\_test, y\_test))

plt.figure(figsize=(10, 5))

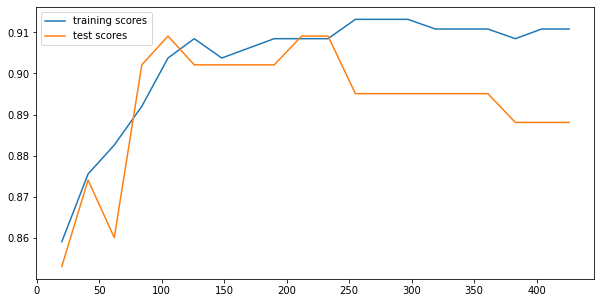
plt.plot(numbers, training\_scores, label="training scores")

plt.plot(numbers, test\_scores, label="test scores")

plt.legend(loc="best")

Проанализировав получившийся график мы может понять более подробную картину. В частности можно видеть, что с ростом объема обучающей выборки точность на ней растет монотонно, а вот точность на тестовой резко возрастает, а затем начинает снижаться.

При анализе таких графиков следует понимать, что на реальных данных всегда будут присутствовать случайные отклонения, что приводит к зашумленности графиков. Это означает, что нужно обращать внимание на общую тенденцию, а мелкие вариации скорее всего не несут никакой полезной информации. Причем, чем меньше выборка в целом, тем больше будет вклад случайных факторов. Поэтому, в общем случае, увеличение объема данных приводит и к более точным моделям и облегчает их анализ.



Для лучшей читаемости результатов рекомендуется зафиксировать минимальное и максимальное значение по вертикальной оси, чтобы все графики, построенные в рамках данной работы были сопоставимы. Например, можно строить все графики в диапазоне y от 0,8 до 1,0.

### Контрольные вопросы

1. Как влияет параметр С на степень регуляризации моделей классификации?
2. Что показывает соотношение точности (ошибки) на тестовой и обучающей выборке?
3. Чем характеризуется ситуация недообучения? Как ее распознать на кривых обучения? А переобучение?
4. Каковы рекомендации по улучшению точности модели в случае недообучения? А переобучения?
5. Какова вычислительная сложность подбора гиперпараметров модели?

### Дополнительные задания

1. Повторите такой же анализ для полиномиального ядра и степени полинома как гиперпараметра. Прокомментируйте кривые обучения для случаев недо-и переобучения.
2. Исследуйте зависимость степени обученности модели от значения других гиперпараметров различных классификаторов, реализованных в sklearn. Постройте кривые обучения в зависимости от значения этих параметров.
3. Повторите данное исследование на полном датасете. Сделайте выводы об оптимальной регуляризации моделей.
4. Повторите данное исследование на другом датасете. Предпочтительно выбирать более объемный набор данных для классификации (несколько тысяч примеров на каждый класс).
5. Напишите функцию, которая автоматизирует процесс подбора гиперпараметров данной модели путем проверки ошибок на тестовой выборке моделей с разными значениями гиперпараметра. Сравните вашу реализацию с функцией GridSearch.