

1. Загрузите набор данных ZipCode (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Используя метод опорных векторов, исследуйте зависимость тестовой ошибки от параметра γ , используя ядро типа radial.
2. Загрузите набор данных ZipCode (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Используя метод опорных векторов, исследуйте зависимость тестовой ошибки от степени полинома degree, используя ядро типа polynomial
3. Пусть ответ задается в виде аналитической функции **x AND (y XOR z)**, где x, y и z – принимают значение TRUE или FALSE. Постройте дерево решений, предсказывающее ответ с нулевой ошибкой.
4. Загрузите набор данных ZipCode (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Исследуйте значение тестовой ошибки от параметра k (число соседей) в методе K ближайших соседей.
5. Загрузите набор данных ZipCode (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Исследуйте зависимость тестовой ошибки от высоты дерева решений.
6. Загрузите набор данных ZipCode (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Подсчитайте cross-validation ошибку для метода опорных векторов при числе разбиений, равном 5, и сравните её с тестовой ошибкой.
7. Загрузите набор данных Ozone (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Подберите оптимальную регрессионную модель для данных (ozone ~.).
8. Загрузите набор данных Ozone (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на тестовую и обучающую выборку (в пропорции 50 на 50). Подберите оптимальный (в плане тестовой ошибки) параметр lambda в гребневой регрессии.
9. Загрузите набор данных Ozone (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на тестовую и обучающую выборку (в пропорции 50 на 50). Сравните качество обучения с использованием метода опорных векторов и регрессии (по ошибке на тестовой выборке). Параметры моделей выберите на Ваше усмотрение.
10. Загрузите набор данных Ozone (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на тестовую и обучающую выборку (в пропорции 50 на 50). Сравните качество обучения с использованием метода опорных векторов и K ближайших соседей (по ошибке на тестовой выборке). Параметры моделей выберите на Ваше усмотрение.
11. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.traintest). Используя метод опорных векторов, добейтесь минимальной тестовой ошибки путем подбора типа ядра и его параметров.
12. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.traintest). Подберите оптимальное (в смысле тестовой ошибки) значение параметра k (число соседей) в методе K ближайших соседей.
13. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.traintest). Используя процедуру pruning.tree постройте оптимальное (в плане тестовой ошибки) дерево.
14. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.traintest). Подсчитайте cross-validation ошибку для деревьев решений при числе разбиений, равном 5, и сравните её с тестовой ошибкой.

15. Пусть ответ задается в виде аналитической функции $x \text{ XOR } ((y \text{ XOR } z) \text{ OR } w)$, где w, x, y и z – принимают значение TRUE или FALSE. Постройте дерево решений, предсказывающее ответ с нулевой ошибкой.
16. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.train и spam.test). Сравните качество обучения с использованием метода опорных векторов и K ближайших соседей. Параметры моделей выберите на Ваше усмотрение.
17. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.train и spam.test). Сравните качество обучения с использованием деревьев решений и метода K ближайших соседей. Параметры моделей выберите на Ваше усмотрение.
18. Загрузите набор данных Spam (<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>). Разделите данные на обучающую и тестовую выборку (согласно меткам в файле spam.train и spam.test). Сравните качество обучения с использованием деревьев решений и метода опорных векторов. Параметры моделей выберите на Ваше усмотрение.
19. Загрузите набор данных Vehicle из пакета mlbench. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. Подберите оптимальный параметр γ для радиального ядра в методе опорных векторов, используя в качестве оптимизируемого критерия cross-validation (CV) ошибку на тренировочной выборке. Сравните полученную CV ошибку с ошибкой на тестовой выборке.
20. Загрузите набор данных Vehicle из пакета mlbench. Пусть N – число объектов в наборе данных. Для каждого $i = 1, \dots, 9$ разделите данные на тренировочную выборку мощности $(i/10) * N$ и тестовую выборку мощности $((10-i)/10) * N$. Исследуйте зависимость тестовой и cross-validation ошибки от мощности обучающей выборки (для дерева решений и метода опорных векторов с радиальным ядром), дайте интерпретацию полученному результату.
21. Загрузите набор данных Vehicle из пакета mlbench. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. Сравните качество обучения с использованием метода опорных векторов (с радиальным ядром) и K ближайших соседей.
22. Загрузите набор данных Vehicle из пакета mlbench. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. Пусть response (ответ) принимает M различных значений $\{\text{Response}_1, \dots, \text{Response}_M\}$. Для всех $m = 1, \dots, M-1$ выполните следующую операцию: удалите из обучающей и тестовой выборки все прецеденты, в которых ответ принимает значение, равное одному из значений $\{\text{Response}_{m+1}, \dots, \text{Response}_M\}$ (т.о., при $m=M$ никаких прецедентов удалять не требуется), и обучите модель, используя метод опорных векторов с радиальным ядром. Постройте график зависимости тестовой ошибки от m , дайте интерпретацию полученному результату.
23. Загрузите набор данных Vehicle из пакета mlbench. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. Используя метод опорных векторов с радиальным ядром, исследуйте зависимость cross-validation ошибки и тестовой ошибки от параметра γ (изменяйте данный параметр от 1 до 10 с шагом 0.5)
24. Загрузите набор данных Vehicle из пакета mlbench. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. Для алгоритма bagging подберите оптимальное число деревьев (не более 200) а) в плане ошибки на тренировочной выборке, б) в плане ошибки на тестовой выборке. Дайте интерпретацию полученным результатам.

25. Создайте набор данных, мощности 500, где каждый прецедент (y_i, x_i) генерируется по следующему правилу: x_i 10-мерный вещественный вектор, в котором каждая компонента сэмпляется из равномерного распределения на отрезке $[0, 1]$, $y_i = x_{i1} + x_{i2}$ (таким образом, на ответ влияют лишь две первые компоненты x_i). Постройте линейную регрессию с помощью функции `lm`, дайте интерпретацию построенной модели (объясните, какие коэффициенты являются значимыми и почему).
26. Создайте набор данных, мощности 400, где каждый прецедент (y_i, x_i) генерируется по следующему правилу: x_i 2-мерный вещественный вектор, в котором каждая компонента сэмпляется из равномерного распределения на отрезке $[0, 1]$, $y_i = x_{i1} + x_{i2} + x_{i1}x_{i2} + \text{rnorm}(0,1)$ (нормальное распределение с мат. ожиданием 0 и стандартной ошибкой 1). Выберите оптимальную регрессионную модель, реализуйте её с помощью функции `lm`. Используя данные построенной модели, покажите, что она не является избыточной (все коэффициенты являются значимыми).
27. Загрузите набор данных `Glass` из пакета `mlbench`. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. Сравните качество обучения (на тестовой выборке) следующих моделей: метод опорных векторов с радиальным ядром, метод К ближайших соседей, дерево решений, `adaboost`.
28. Загрузите набор данных `Glass` из пакета `mlbench`. Разделите данные на тренировочную и тестовую выборки равной мощности случайным образом. В тренировочной и тестовой выборке присвойте всем прецедентам, 1) которым соответствует `Type = 1,2,3,4`, значение `Type = 1`, 2) которым соответствует `Type = 5,6,7`, значение `Type = 2`. Постройте дерево решений, вычислите ошибку на тестовом множестве. Сравните с тестовой ошибкой, которая получается на исходной тренировочной и тестовой выборках. Объясните полученные результаты.