# Задачи для второкурсников МФТИ

Бурнаев Е.В., burnaevevgeny@gmail.com

Приведенные в данном документе задачи дают некотрое представление о том, какие задачи решают в анализе данных и машинном обучении. Предполагается, что студент сможет на собеседовании изложить выбранную задачу и результаты, полученные в процессе ее решения. Все мысли, возникающие по представленным ниже задачам, в том числе замеченные неточности и неясности, отправляйте на почту burnaevevgeny@gmail.com или likzet@gmail.com.

**Задача 1.**

1. Ознакомиться с простыми методами классификации и регрессии (k-ближайших соседей, деревья решений, линейная регрессия, гребневая регрессия, логистическая регрессия). Лекции 1-4 по адресу https://habrahabr.ru/company/ods/blog/322626/ содержат всю необходимую информацию как о задачах классификации и регрессии, так и о методах решения перечисленных выше задач.

2. По ссылке https://archive.ics.uci.edu/ml/datasets/Energy+efficiency можно найти выборку данных для задачи регрессии. К этой выборке нужно применить несколько методов из перечисленных выше. Реализации методов есть в библиотеке для Python scikit-learn http://scikit-learn.org/stable/

3. Оценить качество работы выбранных алгоритмов с помощью скользящего контроля https://en.wikipedia.org/wiki/Cross-validation_(statistics)). В качестве критерия качества используйте среднеквадратичную ошибку аппроксимации https://en.wikipedia.org/wiki/Mean_squared_error.

**Задача 2.**

1. Изучить алгоритм кластеризации k-means https://en.wikipedia.org/wiki/K-means_clustering

2. Реализовать алгоритм k-means на вашем любимом языке программирования (наш любимый — Python).

3. Используя метод make_blobs из библиотеки scikit-learn http://scikit-learn.org, сгенерировать обучающую выборку размера 500 с количеством признаков 2 и количеством кластеров 5.

4. Применить алгоритм k-means. Посмотреть, визуализировать результаты для случаев, когда количество кластеров меньше истинного количество кластеров данных, равно и больше.

**Задача 3.** В машинном обучении часто возникают задачи снижения размерности: необходимо построить процедуру снижения размерности, то есть отображение $f : \mathbb{R}^p \to \mathbb{R}^q$ из исходного пространства $\mathbb{R}^p$ в пространство меньшей размерности $\mathbb{R}^q$, $p > q$, и процедуру восстановления, обратную сжатию $g : \mathbb{R}^q \to \mathbb{R}^p$. Тогда если сжать, а потом восстановить вектор $\mathbf{x}$, то получится вектор $g(f(\mathbf{x}))$. Качество построенных процедур снижения размерности и восстановления можно оценить на выборке $D = \{\mathbf{x}_i\}_{i=1}^n$ с помощью квадратичной ошибки $Q(f, g, D) = \sum_{i=1}^n \|g(f(\mathbf{x}_i)) - \mathbf{x}_i\|^2$.

В качестве процедур сжатия и восстановления часто используют линейные модели:

$$f(\mathbf{x}) = A\mathbf{x} + \overline{f}, A \in \mathbb{R}^{q\times p}, \overline{f} \in \mathbb{R}^q$$
$$g(\mathbf{x}) = B\mathbf{y} + \overline{g}, B \in \mathbb{R}^{p\times q}, \overline{g} \in \mathbb{R}^p.$$

Исторически первым методом построения такой пары линейных моделей является метод главных компонент, предложенный К.Пирсоном в 1901 году.

Необходимо доказать, что процедуры $f$ и $q$, построенные с помощью метода главных компонент с использованием выборки $D$, минимизируют среднеквадратичную ошибку $Q(f, g, D)$ для заданной размерности $q$.

Метод главных компонент описан, например, в книге Хасти и Тибширани [1] в разделе 14.5.1 (см. также книгу Бишопа по байесовским методам машинного обучения [3], разделы 12.1 и 12.2).

**Задача 4.** Задача классификации заключается в том, что по выборке данных $\{(\mathbf{x}_i, y_i = y(\mathbf{x}_i))\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{0, 1\}$ необходимо построить модель зависимости $\hat{y}(\mathbf{x})$, такую, что $\hat{y}(\mathbf{x}) \in \{0, 1\}$ и для большинства значений $\mathbf{x}$ прогноз метки класса $\hat{y}(\mathbf{x})$ совпадает с настоящей меткой класса $y(\mathbf{x})$.

В книге Хасти и Тибширани [1] (см. также книгу Мори [2], раздел 4.2) в разделе 4.5.2 рассматривается модель линейного дискриминантного анализа. Рассматривается модель линейной разделяющей гиперплоскости $\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{x}_i^T\mathbf{w} + w_0 > 0)$. Таким образом, вектор параметров модели $\mathbf{w}, w_0$ задает гиперплоскость, разделяющую два класса.

Для оценки вектора параметров $\mathbf{w}, w_0$ максимизируется отступ разделяющей гиперплоскости от объектов обучающей выборки:

$$\max_{\mathbf{w},w_0,\|\mathbf{w}\|=1} M,$$
$$\text{s.t. } y_i(\mathbf{x}_i^T\mathbf{w} + w_0) \geq M, i = 1, \ldots, n.$$

Нужно описать, как решать такую задачу оптимизации и какими свойствами обладает ее решение.

## Список литературы


[1] Jerome Friedman, Trevor Hastie, Robert Tibshirani. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition, Springer, 2009. http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf

[2] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning. MIT Press, 2012.

[3] Christopher Bishop. Pattern Recognition and Machine Learning // Springer, 2006.