# The exercise 1

Instruction

In this task, you need to select the optimal value of k for the kNN algorithm. We will use the Wine data set, where we need to predict the grape variety from which the wine is made using the results of chemical analyzes.

Do the following steps:

* Download the Wine sample at <https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>
* Extract attributes and classes from the data. The class is written in the first column (three variants), the signs are in the columns from the second to the last. More information about the essence of the signs can be found at <https://archive.ics.uci.edu/ml/datasets/Wine> (see also the wine.names file attached to the task)
* Quality assessment should be carried out by cross-validation in 5 blocks (5-fold). Create a splitter generator that mixes the selection before forming the blocks (shuffle = True). To reproduce the result, create a KFold generator with a fixed parameter random state = 42. As a measure of quality, use the proportion of correct answers (accuracy).
* Find the classification accuracy for cross-validation for the nearest neighbor method (sklearn.neighbors.KNeighborsClassifier), for k from 1 to 50. At which k did the optimal quality get? What is it equal to (a number in the range from 0 to 1)? These results will be answers to questions 1 and 2.
* Scale the characteristics using the function sklearn.preprocessing.scale. Find the optimal k for cross-validation again.
* What is the optimal value of k after bringing the characteristics to the same scale? Answer the questions 3 and 4. Did the scaling of the signs help?
* If the answer is a non-integer number, then the whole and fractional part must be delimited by a point, for example, 0.5. If necessary, round off the fractional part to two characters.

The answer to each task is a text file containing the answer in the first line. Note that the files sent must not contain an empty string at the end. This nuance is a limitation of the Coursera platform. We are working to remove this restriction.

# The exercise 2

Instruction

We will use the Boston data set in this task, where we need to predict the cost of housing based on various characteristics of location (air pollution, proximity to roads, etc.). More details about the features can be read at <https://archive.ics.uci.edu/ml/machine-learning-databases/housing/>

Download the Boston sample using the sklearn.datasets.load\_boston () function. The result of calling this function is an object whose attributes are written in the data field, and the target vector in the target field.

Bring the characteristics in the sample to the same scale using the function sklearn.preprocessing.scale.

Go through various options for the metric p on the grid from 1 to 10 in such a way that only 200 variants are tested (use the numpy.linspace function). Use KNeighborsRegressor with n\_neighbors = 5 and weights = 'distance' - this option adds weights to the algorithm, depending on the distance to the nearest neighbors. As the quality metric, use the root-mean-square error (scoring = 'mean\_squared\_error' parameter for cross\_val\_score; when using scikit-learn version 0.18.1 and higher, scoring = 'neg\_mean\_squared\_error' should be specified. Evaluate the quality as in the previous task, using cross-validation for 5 blocks with random\_state = 42, do not forget to enable shuffle = True.

Determine for which p the quality of the cross-validation was optimal. Note that cross\_val\_score returns an array of quality indicators by blocks; it is necessary to maximize the average of these indicators. This value of the parameter is the answer to the problem.

If the answer is not an integer, then the integer and fractional part must be delimited by a point, for example, 0.4. If necessary, round off the fractional part to one character.

# The Exercise 3

Instruction

* Download the training and test samples from the files perceptron-train.csv and perceptron-test.csv. The target variable is written in the first column, the characteristics are in the second and third.
* Train the perceptron with the standard parameters and random\_state = 241.
* Calculate the quality (the proportion of correctly classified objects, accuracy) of the obtained classifier on the test sample.
* Normalize the training and test sample using the StandardScaler class.
* Train the perceptron on a new sample. Find the proportion of correct answers to the test sample.
* Find the difference between the quality on the test sample after the normalization and the quality before it. This number is the answer to the task.

If the answer is not an integer, then the integer and fractional part must be delimited by a point, for example, 0.421. If necessary, round off the fractional part to three characters.