k-Nearest Neighbors Classifier

# Task

Study the [*KNeighborsClassifier*](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html)class from the [*scikit-learn*](https://scikit-learn.org/stable/index.html) library. For your dataset, find the best combination of parameters (*n\_neighbors*, *weights* and *metric*) for the *kNN* classifier according to its [*F*1-score](https://en.wikipedia.org/wiki/F1_score) of [cross-validation](https://scikit-learn.org/stable/modules/cross_validation.html). You do not have to check each combination manually. Use lists and for-loops. Also you can use the [GridSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) to perform this subtask.

For the best combination of parameters (*weights* and *metric*) found, build a plot of quality of classification as a function of the number of nearest neighbors.

# Datasets

Visit [OpenML website](https://www.openml.org/search?type=data) and select a dataset for classification task (the target variable should be nominal). Try to select data sets evenly. Do not take all the same dataset.

Selected dataset should contain at least 100 instances. If it contains too many instances, you can take a subset of it. If one or more features (excluding the target variable) in your dataset are nominal, you should use [OneHotEncoder](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html).

Don't forget to use preprocessing to make features more consistent. You should use [StandardScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) or [MinMaxScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html).

Tip. Almost every dataset after the description contains a section where you can find reports on various runs and compare your results with it.