# ML (theory and practice)

*Author: Daria Shutina*

## 23-09-08

### Data representation

**Continuous** variables are those that can take any real-numbered value within a certain range (e.g. height, temperature, weight).

**Binary** variables are categorical variables that can take one of two possible values.

**Embeddings** are a technique used to represent categorical or discrete data (e.g. words, categories, IDs) in a continuous vector space.

### Datasets

**Datasets** are structured collections of data used for training, testing, and evaluating models.

Datasets consist of **instances**. Each instance represents a single data entry. **Features** (or attributes) are characteristics or properties associated with each instance. **Labels** (or target values) represent the desired output that the model should learn to make for each corresponding instance.

The **training dataset** is used to train the ML model. It consists of a large number of labeled examples, allowing the model to learn the relationships between the features and labels.

The **validation dataset** is a separate portion of the data used during the training process to assess the model's performance and make decisions about hyperparameters or model selection. It helps prevent overfitting.

The **testing dataset** is used to evaluate the final performance of the trained model. It should be distinct from the training and validation datasets and provide an unbiased assessment of how well the model works with new data.

## Train, validation, test

Hyperparameters are like settings for the ML algorithm. At the beginning, hyperparam values are set based on what you believe would result in a good performance. This can be based on research or experience. After the hyperparams are set and the data is ready, we **train** the model on the training dataset.

The model's performance depends on the dataset it is being trained on. One of the main aims is to make the model robust, meaning the model gives consistent results that are correct most of the time.

After the training is done, results of the model might be not good enough, because hyperparameters do not have an optimal value. On the next step -- **tuning** -- different combinations of hyperparams are tested.

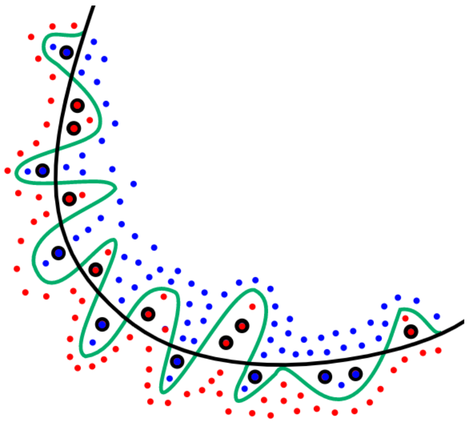
So, the main steps of preparing the model:

- **train**: find the model params. The model is trained with the training dataset.
- **validation**: tune hyperparameters. We tune hyperparams and train again, also checking for the performance. If needed, the tuning can be done again. This way the model is being optimized to show the best performance on the validation data.
- **test**: run the model on data the model has never seen.

We need to split data between train, validation and test datasets. It can be done randomly or sequentially.

## Overfitting

In simple words, the model "overfits" when it is correct on every instance of the training dataset but has poor fit with new, unknown datasets.



Here, blue and red dots represent different classes, and the model is learning to distinguish between them. The green line is an overfitted curve, which means the model gives a correct answer on every instance from the training dataset. On the other hand, the black line is more general.

As a result, the model with the black curve has a higher probability of being correct on a new data, since the green curve is too fluctuating.

## Loss functions

A **loss function** (or cost function) quantifies the difference between the predicted values generated by a model and the target values in the training dataset.

The goal in training a model is to minimize this loss function. The aim is to find the model's parameters that result in the lowest possible loss.

#### Example

We want know whether the user clicks the video. Lets use a **log loss function**:

$$L = -\log p \cdot y + \log(1-p) \cdot (1-y)$$

Here $p$ is a prediction, it is calculated by our model, and $y$ is a label ($1$ if click, $0$ otherwise).

- $L \to \min$
- $y = 1 \;\Rightarrow\; -\log p \to \min$
    - $-\log 0 = +\infty$
    - $-\log 1 = 0$
- $y = 0 \;\Rightarrow\; \log(1-p) \to \min$
    - $\log(1-0) = 0$
    - $\log(1-1) = +\infty$

## 23-09-15

## Linear models

Jupiter notebook practice

We have $n$ features $(x_0, \dots, x_{n-1})$ and want to find coefficients $a_0, \dots, a_{n-1}$:

$$y_{pred} = f(A, X) = a_0 x_0 + \dots + a_{n-1} x_{n-1}$$

The loss function will be the distance from real and predicted results:

$$L = (y_{pred} - y_{real})^2 = (\sum_0^{n-1} a_i x_i - y_{real})^2$$
$$L \to \min$$

$\frac{\Delta L}{\Delta a_i} = L'_{a_i} = 2(a_0 x_0 + \dots + a_{n-1} x_{n-1} - y_{real}) x_i$

## Decision trees

### Boosting forest

The implementation is `catboost`

### Random forest

$y = avg(tree_0(x), tree_1(x), \dots, tree_{n-1}(x))$