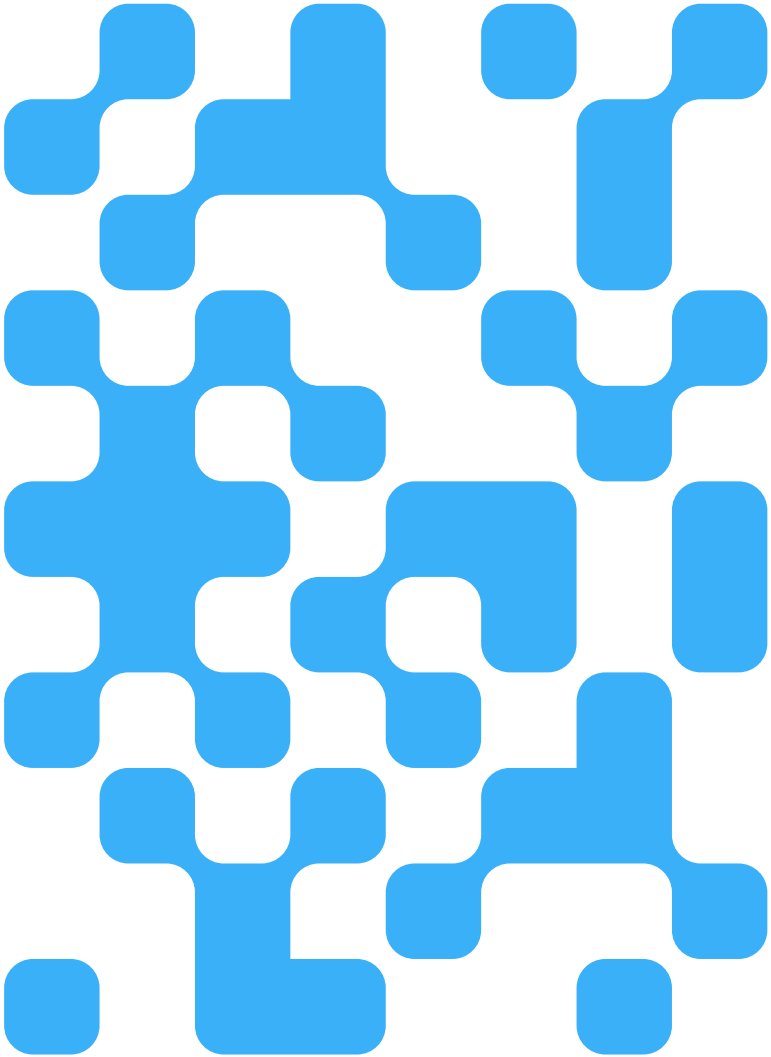
Machine Learning

2024 (ML-2024)

Lecture 4. Quality metrics. Tips & tricks in ML: data splitting, cross-validation, regularization, batches

by Alexei Valerievich Kornaev, Dr. habil. in Eng. Sc.,
Researcher at the RC for AI, Assoc. Prof. of the Robotics and CV Master's Program, Innopolis University
Researcher at the RC for AI, National RC for Oncology n.a. NN Blohin
Professor at the Dept. of Mechatronics, Mechanics, and Robotics, Orel State University



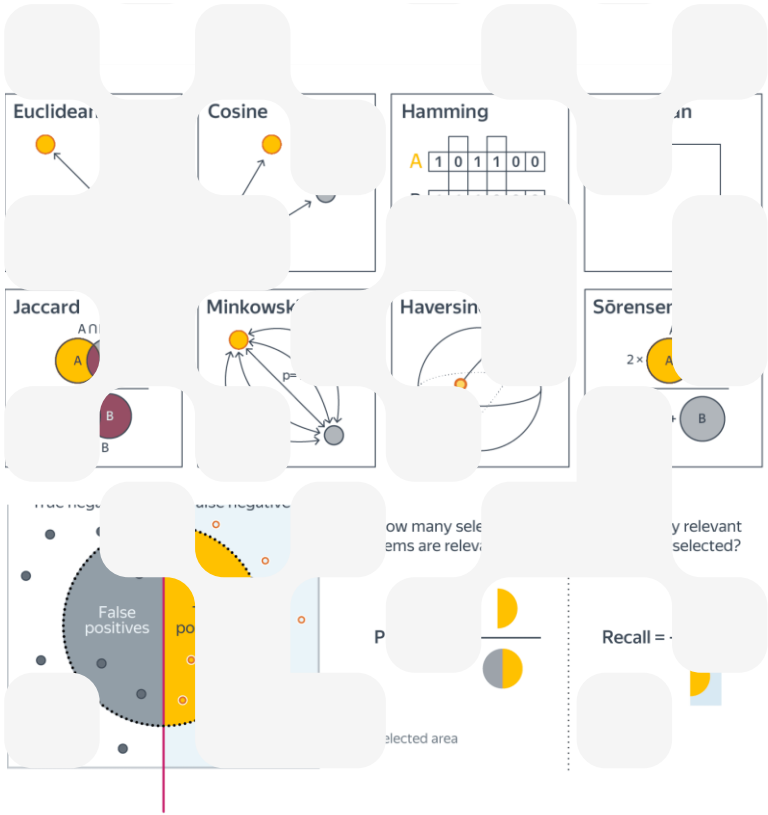
Agenda

I. Quality metrics in ML

II. Data splitting, cross-validation

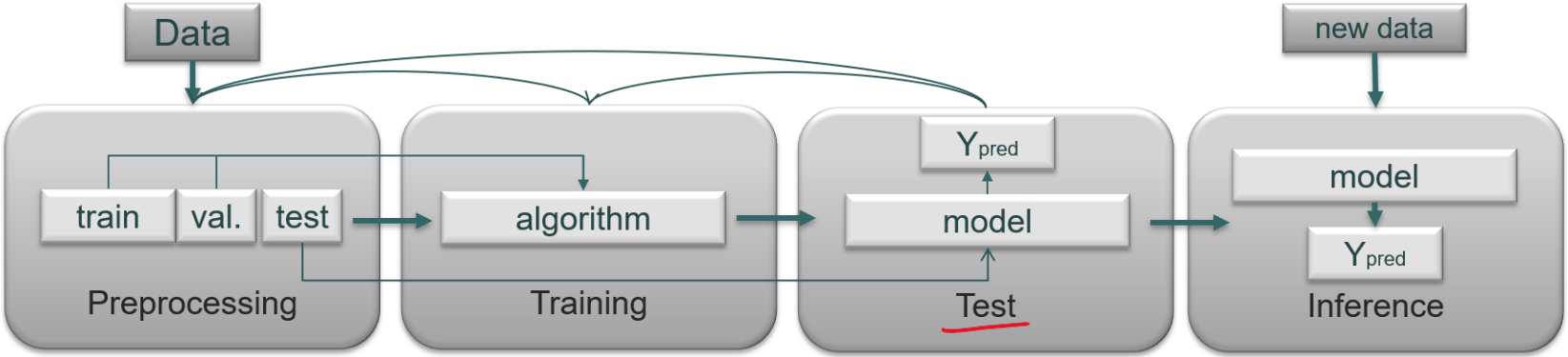
III. Regularization

IV. Batches



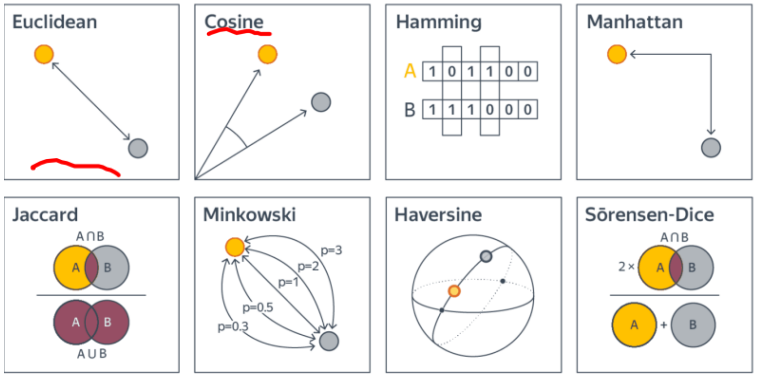
All models are wrong, but some are useful.
/George Box/

Flowchart for an ML model design

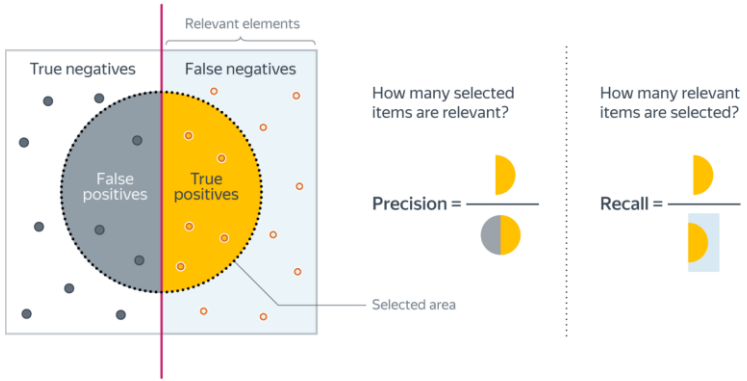


Metrics can be used both in learning (metric learning) and in estimating a model's quality

Model predicts output ***h*** given input ***x***

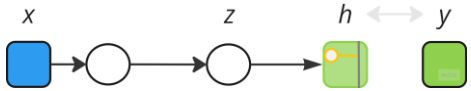


[Metrics from the Yandex Handbook]



[Classification metrics from the Yandex Handbook]

Metrics that estimate model's quality. Binary classification



Model predicts output *h* given input *x*

1. Accuracy: $acc(\boldsymbol{h}, \boldsymbol{y}) = \frac{1}{m}\sum_{i=1}^{m}\left(h^{(i)} == y^{(i)}\right)$, or (and) error rate: $error\ rate = 1 - \text{acc}(\boldsymbol{h}, \boldsymbol{y})$

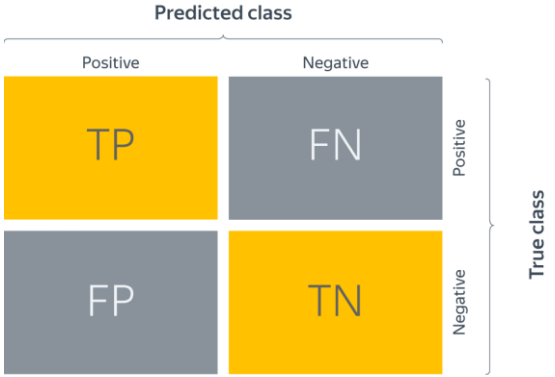
$$acc(\boldsymbol{h}, \boldsymbol{y}) = \frac{TP+TN}{TP+TN+FP+FN}.$$

$acc =$ 0.8

Predicted	True
1	0
0	0
1	1
0	0
1	0
0	0
0	0
1	1
0	0
0	0

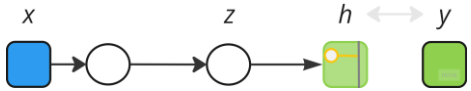
$acc =$ 0.8

Predicted	True
0	1
0	0
0	0
0	0
0	1
0	0
0	0
0	0
0	0
0	0



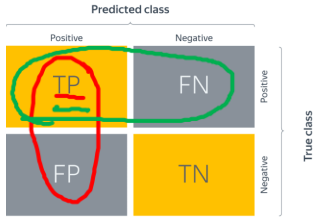
Confusion matrix

Metrics that estimate model's quality. Binary classification



Model predicts output *h* given input *x*

1 - rare class



2. $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $F_1 = \frac{Recall \cdot Precision}{Recall+Precision}$.

F1 score

$pr =$ $re =$ $F_1 =$

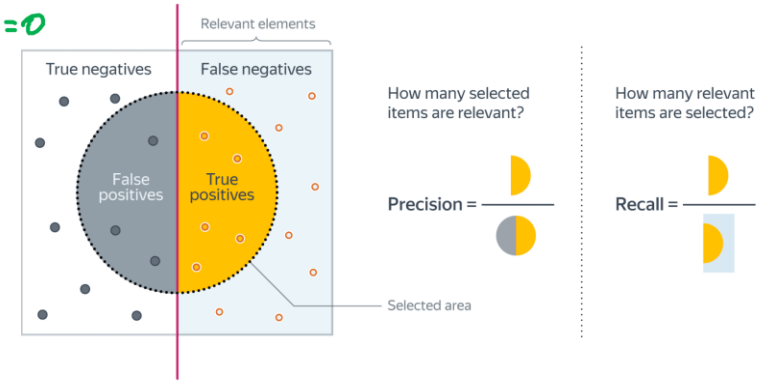
Predicted	True
1	0
0	0
1	1
0	0
1	0
0	0
0	0
1	1
0	0
0	0

FP

$pr = 0$ $re = 0$ $F_1 = 0$

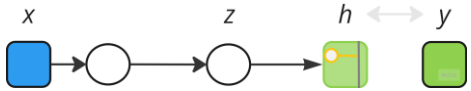
Predicted	True
0	1
0	0
0	0
0	0
0	1
0	0
0	0
0	0
0	0
0	0

TP=0



Precision, recall, F1-score intuition

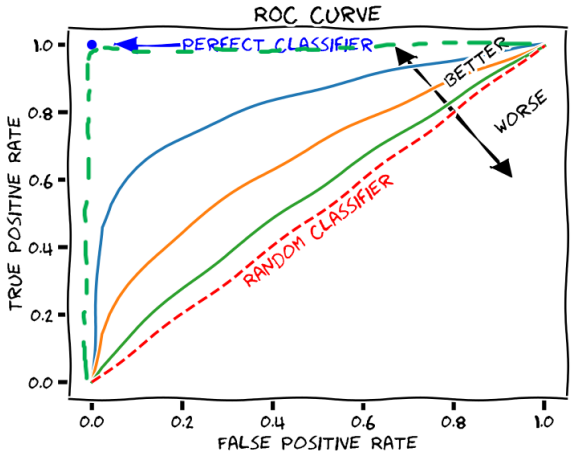
Metrics that estimate model's quality. Binary classification



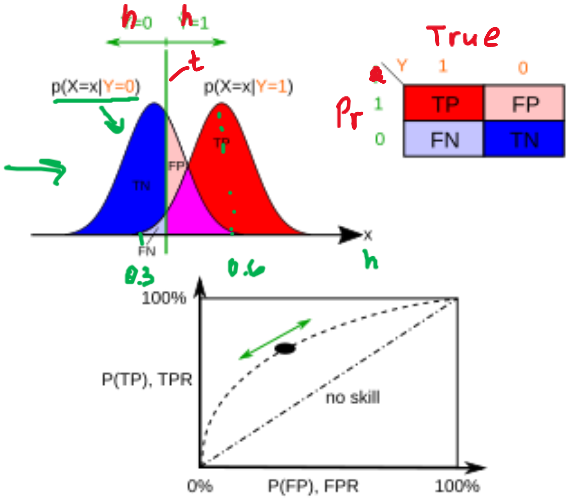
$h < t \to 0$; t is threshold - a tunable param.
$h \geq t \to 1$;

Model predicts output *h* given input *x*

1. True positive rate $TPR = Recall = \frac{TP}{TP+FN}$, false positive rate $FPR = \frac{FP}{FP+TN}$, Receiver operating characteristic (*ROC*), area under curve AUC.

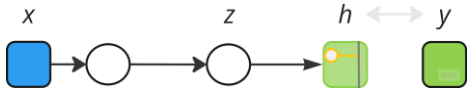


[ROC curve intuition]



Overall, the optimization of precision and recall proceeds as follows:
- train the model on a loss function;
- obtain metric graphs depending on the threshold using real predictions on the validation set, by iterating over different thresholds from 0 to 1;
- select the desired combination of precision and recall.

Metrics that estimate model's quality. Fitting



Model predicts output *h* given input *x*

1. Mean squared error (MSE): $$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(h^{(i)} - y^{(i)}\right)^2;$$
2. Mean absolute error (MAE): $$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|h^{(i)} - y^{(i)}\right|;$$
3. Root MSE (RMSE): $$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h^{(i)} - y^{(i)})^2}.$$

Agenda

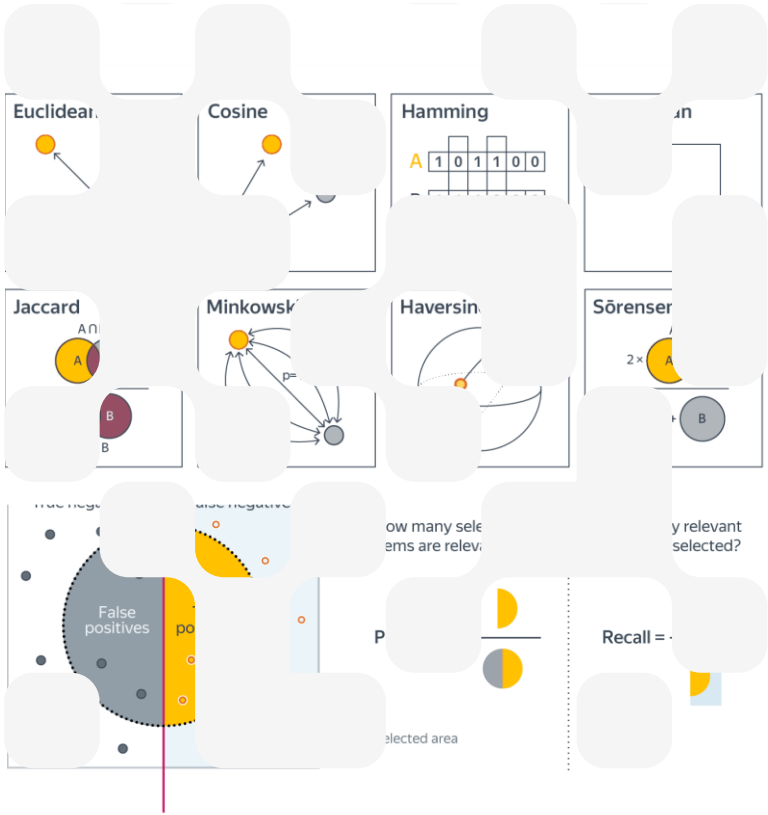
I. Quality metrics in ML

II. Data splitting, cross-validation

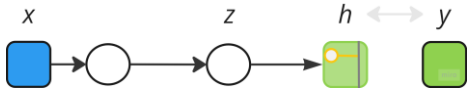
III. Regularization

IV. Batches

All models are wrong, but some are useful.
/George Box/



Overfitting intuition and data splitting

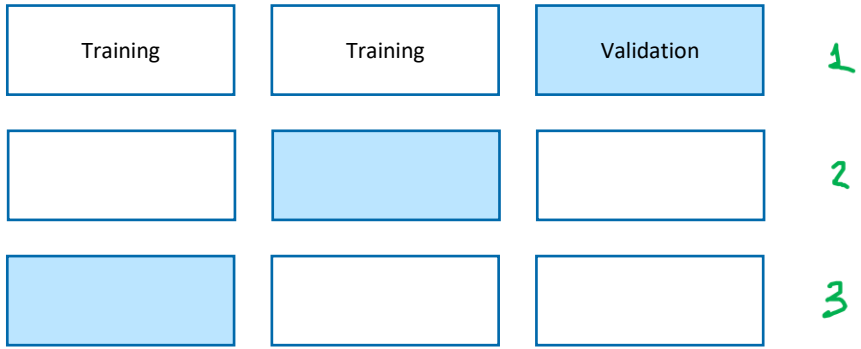


Model predicts output *h* given input *x*

Model *parameters* are determined during the solution of the ML problem, e.g. model weights. *Hyperparameters* are set by the user, usually not in a single way, and their values affect the values of the sought parameters.

Training data splitting by default: training set, validation set

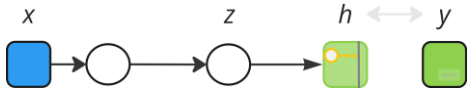
K-Fold splitting (normally used when the dataset is small)



Algorithm. The dataset is divided into *k* equal parts. Next, *k* iterations occur, during each of which one fold serves as the validation set, and the union of the remaining folds serves as the training set. The model is trained on *k-1* folds and tested on the remaining one. The final score of the model is obtained either by averaging the resulting test results or by measuring it on a held-out test set that did not participate in cross-validation.

Testing procedure: the accuracy of the selected trained model is checked on a new (test set)!!!

Overfitting intuition and data splitting

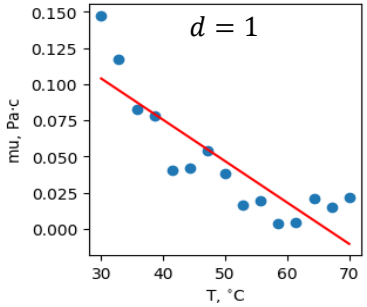


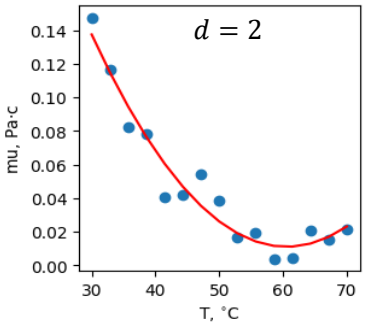
Model predicts output *h* given input *x*

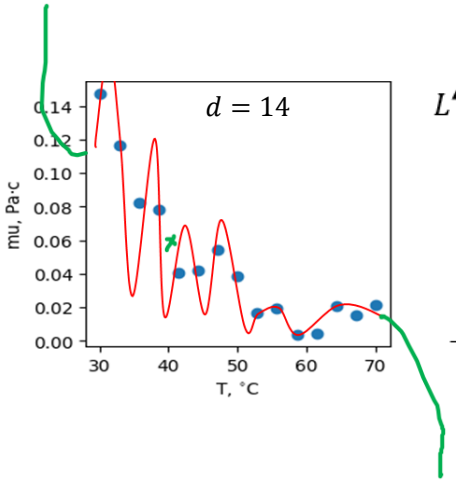
$$h = \theta_j x^j, (j = 0,..d)$$

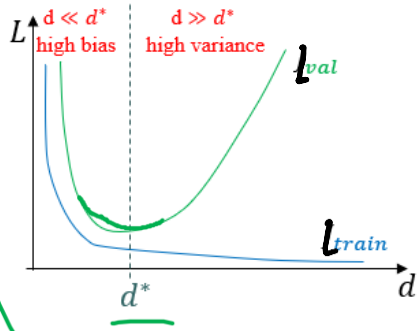
Model *parameters* are determined during the solution of the ML problem, e.g. model weights. *Hyperparameters* are set by the user, usually not in a single way, and their values affect the values of the sought parameters.

$$L = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h^{(i)} - y^{(i)}\right)^2\right] \Rightarrow \min.$$









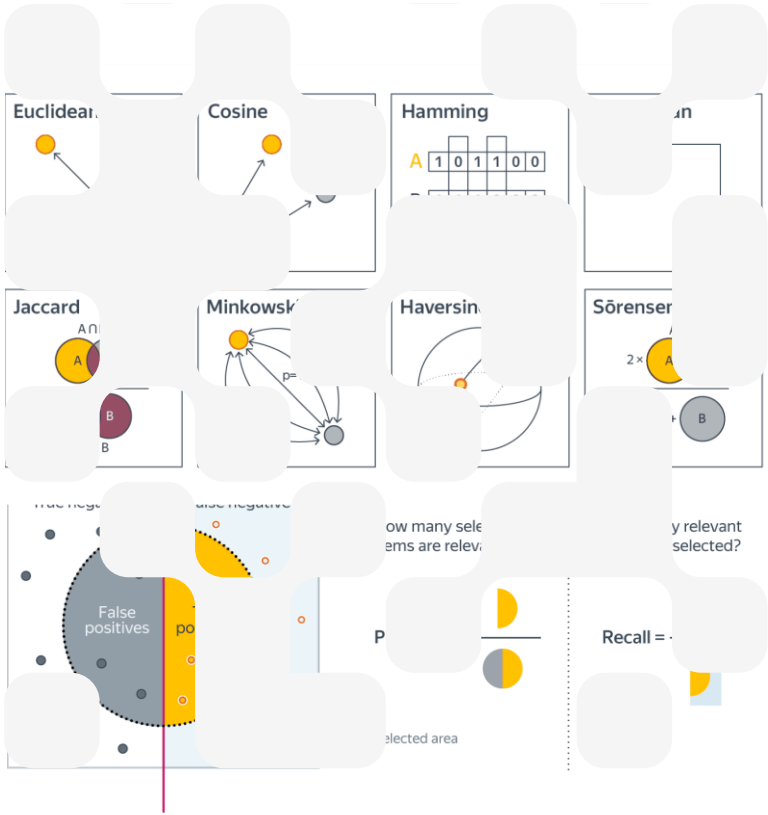
Agenda

I. Quality metrics in ML

II. Data splitting, cross-validation

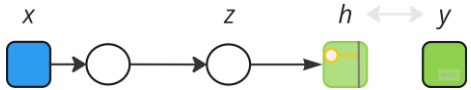
III. Regularization

IV. Batches



All models are wrong, but some are useful.
/George Box/

Overfitting intuition and regularization (L2)



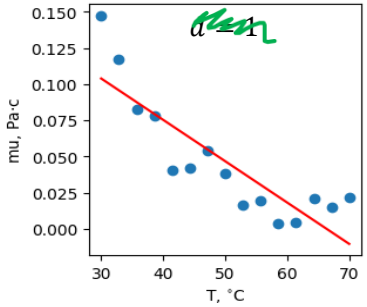
Model predicts output *h* given input *x*

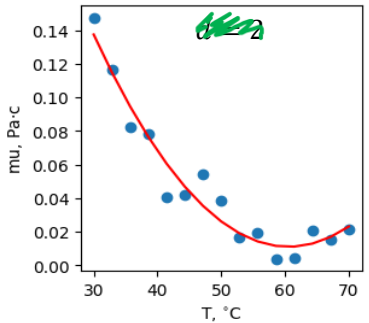
1. Feature Scaling
2. Learning Rate
3. Error and # of iterations
4. **Regularization (L2)**

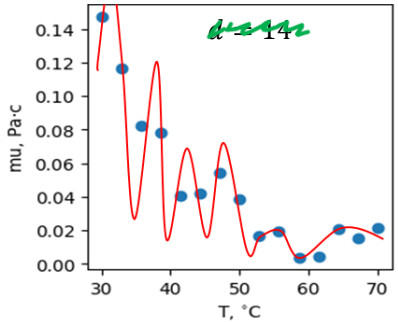
Model *parameters* are determined during the solution of the ML problem. For example, in regression problems, the parameters are the components of the matrix of weights $\boldsymbol{\phi}$. *Hyperparameters* are set by the user, usually not in a single way, and their values affect the values of the sought parameters.

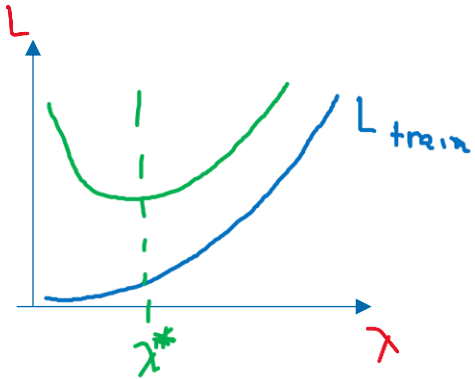
$$L = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h^{(i)} - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \phi_j^2\right] \Rightarrow \min.$$

$$h(x) = \theta_j x^j \quad , (j = 0, .. d)$$









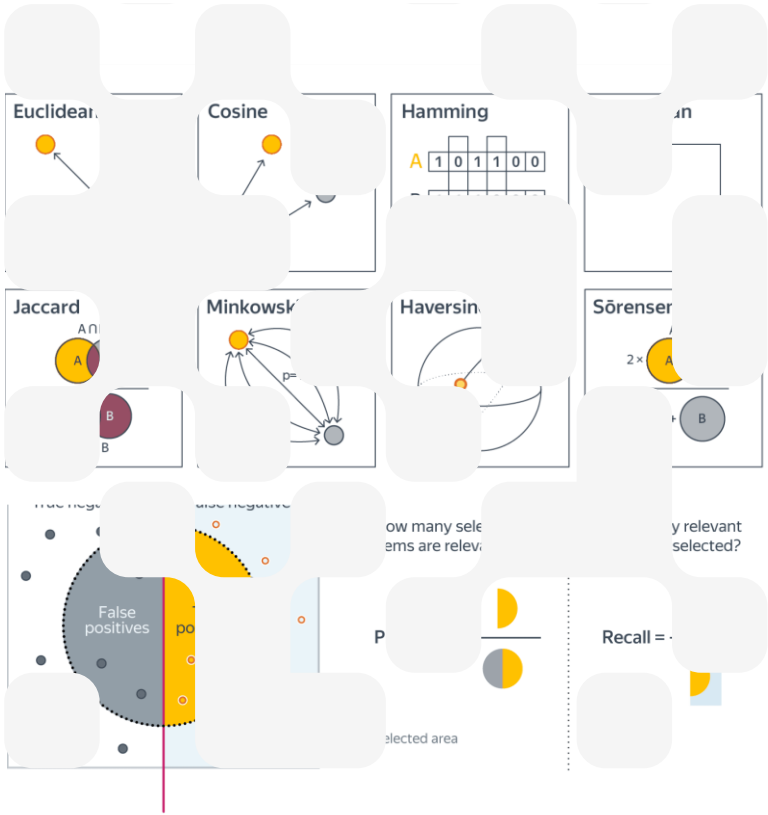
Agenda

I. Quality metrics in ML

II. Data splitting, cross-validation

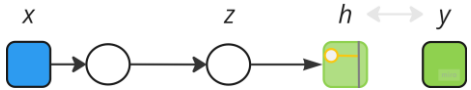
III. Regularization

IV. Batches



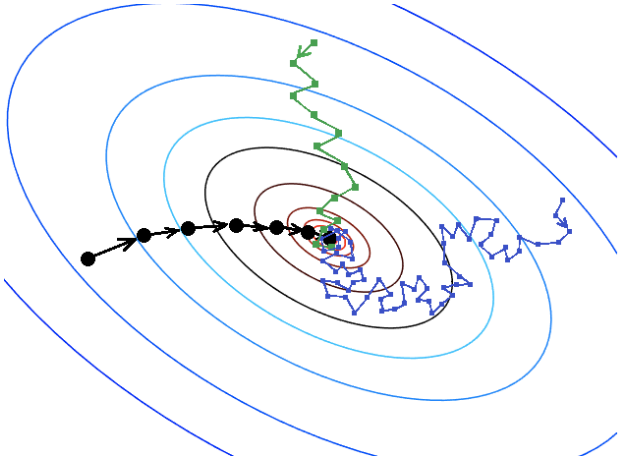
All models are wrong, but some are useful.
/George Box/

Overfitting intuition and data splitting



Model predicts output *h* given input *x*

Model *parameters* are determined during the solution of the ML problem, e.g. model weights. *Hyperparameters* are set by the user, usually not in a single way, and their values affect the values of the sought parameters.



Batch GD
- Slowest
- Perfect gradient

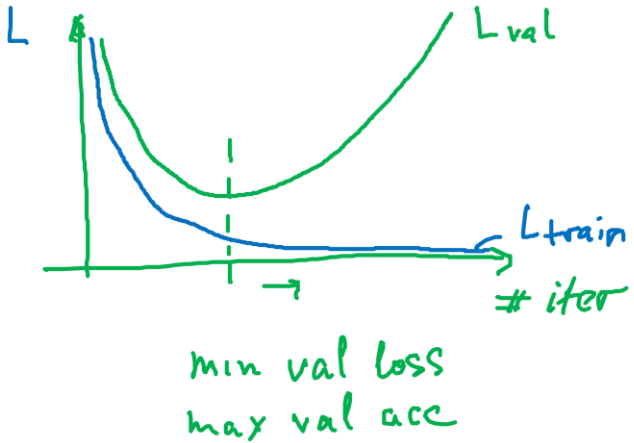
Stochastic GD
- Fastest
- Rough-estimate grad

Mini-batch GD
- Compromise

https://dragonnotes.org/MachineLearning/Optimization

Thank you for your attention!

a.kornaev@innopolis.ru, @avkornaev



$$L = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h^{(i)} - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\phi_j^2\right] \Rightarrow \min.$$