LECTURE 2: MAIN CONCEPTS IN DATA SCIENCE

EKATERINA MURAVLEVA

OVERVIEW OF THE COURSE

Lecture 1: General course information, CRISP-DM methodology

Lecture 2: Supervised learning/unsupervised learning. Classification/regression problems. Accuracy metrics (precision, recall, ROC-AUC scores). Concept of loss functions, overfitting / underfitting.

Lecture 3: Classical ML: Linear regression, logistic regression, support vector machine

Lecture 4: Classical ML: Decision trees, random forests, boosting.

Lecture 5: Classical ML: Dimensionality reduction: linear, non-linear methods.

Lecture 6: Classical ML: Clustering methods

Lecture 7: Basic neural networks

Lecture 8: Scalable algorithms

RECAP OF LECTURE 1

- Introduction to CRISP-DM Methodology
- Overview of CRISP-DM: 6 stages (Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment)

ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is everywhere now: face detection, autonomous driving, medical imaging, natural language processing (NLP), reinforcement learning (chess, go, …), weather prediction, text mining, etc

Applications: medicine, agriculture, production, social networks, security

Most of the AI uses **machine learning**

MACHINE LEARNING

Machine learning is learning to predict using certain observed data and its statistical properties

The simplest task: **supervised learning**

SUPERVISED LEARNING

In supervised learning we have the **data**. The data consists of **objects.**

Objects are described by **features.**

In supervised learning the goal is to predict the properties of objects using the features.

EXAMPLE 1: OPENING A RESTAURANT

Object: Location of a new restaurant

Prediction: revenue in the next year

Example of features:

Demographics (age, income)

Price of the property

Availability of the offices

Specifics:

Non-uniform features, outliers

EXAMPLE 2: SORTING THE SEARCH RESULTS

Object: (text request, document)

Prediction: Relevant pair / not relevant pair

Example of features:

text of the document,

frequency of the words,

number of references to the document

EXAMPLE 3: COMPLEX STRUCTURE

Machine translation

Object: text in one language

Prediction: text in another language

Image classification

Object: image (pixels)

Prediction: image class

SUPERVISED LEARNING

In supervised learning, we are given the **training set:**

Features $x_i, i = 1,\ldots,N$ of N objects and true answers $y_i,\ i = 1,\ldots,N$

Features can be **binary, nominal (number), ordering, number**

TYPE OF FEATURES (1)

Binary feature: takes value 0 or 1 (example: married or not)

We can also use this type of variable as the one we predict,

i.e. predict default (binary classification).

TYPE OF FEATURES (2)

Nominal features: takes finite set of values $0,1,\ldots,M$

Example: education

TYPE OF FEATURES (3)

Ranking: takes ordered set of variables.

Example: you can get score A, B, C, D, E, F for this course,

and if someone gets A and someone gets B,

this information should be included into the model itself

TYPE OF FEATURES (4)

Numerical: takes arbitrary real values. Example: age, temperature, weight…

In most cases, we can reduce binary/categorical/ranking features to numerical features using **feature transformation**.

For example, for categorial features, the standard approach is **one-hot encoding:**

$i \in \{0,\ldots,M-1\} \rightarrow y \in \mathbb{R}^M, \quad y_j = \delta_{ij}$ (i.e. one only in position i).

Then we can use those features in a single large vector.

SUPERVISED LEARNING

Now, lets move

Given the **training dataset** $(x_i, y_i), \ldots$ we want to learn a model

$y = f(x, \theta)$ such that $\hat{y}_i = f(x_i, \theta) \approx y_i, \quad i = 1, \ldots, N$

The accuracy of predictions is measured on an unseen **test dataset** which is not seen during training phase.

TRAIN/TEST/VALIDATION SPLIT

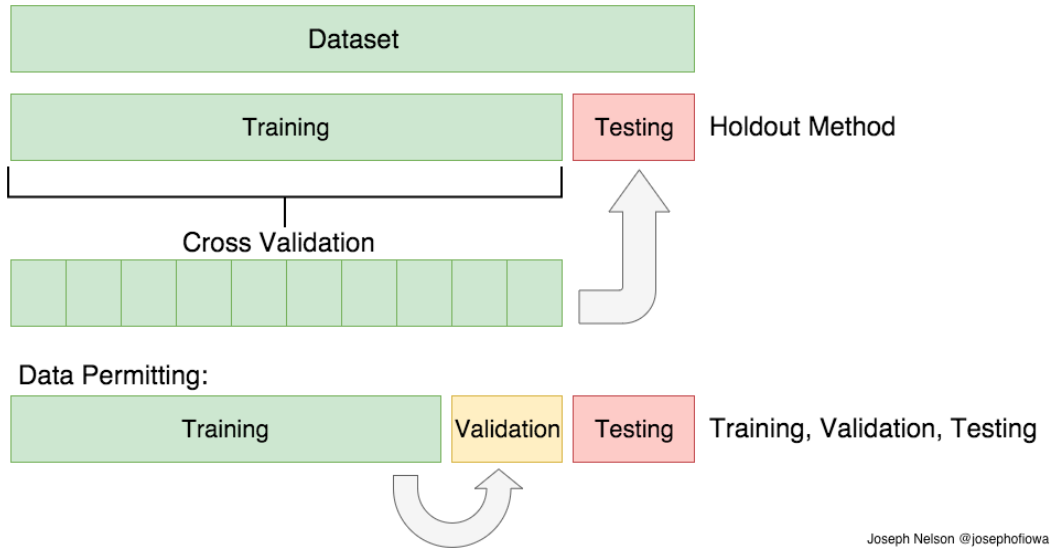
It is very important to split the data into train, test and validation sets.

The model parameters are selected to minimize the error on the **training set,**

The error is measured on the **test set** which should not be seen during training.

If the error is small we say that the model **generalizes well** to unseen data

The **validation set** is not used to update training parameters, but it is used to estimate generalization during training (i.e., for stopping the training process)



MANY WAY TO MAKE AN ERROR IN TRAIN/TEST SPLITS

Even experienced data scientists can make train/test leakage!

Example 1:

Predict crops from satellite images.

Data: Satellite images of California.

Split: random in time.

Problem: California in winter and summer has no snow. The model does not generalize to other locations!

MORE ADVANCED EXAMPLE

Even experienced data scientists can make train/test leakage!

Example 2:

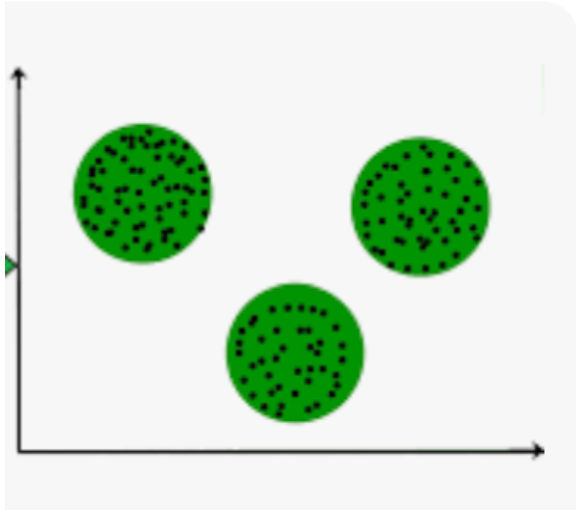
Predict docking of proteins and ligands (small molecules)

Data: Protein / ligand complexes with known energies.

Split: random .

Problem: The data is highly clustered (i.e., similar data points), so train and test data are very similar.

Correct split: cluster the data first and sample test from other clusters.



Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction

J Li, X Guan, O Zhang, K Sun, Y Wang, D Bagni… - arXiv preprint arXiv …, 2023 - arxiv.org

TYPE OF PROBLEMS

Now let's discuss in more details

what types problems are available in machine learning

and how they can be structured.

CLASSIFICATION

$y = f(x, \theta)$ such that $\hat{y}_i = f(x_i, \theta) \approx y_i, \quad i = 1, \ldots, N$

Let $y_i \in Y$.

Classification

$Y = \{0,1\}$ — binary classification (cat/dog)

$Y = \{1, \ldots, M\}$ — M-class classification (ImageNet with 1000 classes)

$Y = \{0,1\}^M$ — each object can be described by several classes (many classes on 1 picture, still discrete)

REGRESSION

$y = f(x, \theta)$ such that $\hat{y}_i = f(x_i, \theta) \approx y_i, \quad i = 1, \ldots, N$

Let $y_i \in Y$.

Regression

$Y = \mathbb{R}$ (predict scalar quantity)

$Y = \mathbb{R}^d$ (predict vector quantity)

LEARNING TO RANK

$y = f(x, \theta)$ such that $\hat{y}_i = f(x_i, \theta) \approx y_i, \quad i = 1, \ldots, N$

Let $y_i \in Y$. Depending on Y, we have the following problems:

Ranking (learning to rank)

Y — finite ordered set

Example: search results, recommender systems

UNSUPERVISED LEARNING

In Unsupervised Learning, we do not have outputs, only inputs

So we need to define what we want to learn.

Some options:

- Clustering (i.e. map input to a number of the cluster)
- Generative models (advanced, learn probability distribution of samples)
- Dimensionality reduction
- Self-supervised learning (advanced & recent).

STATISTICAL MACHINE LEARNING

= Learning from precedents

= Recovering dependencies from empirical data

= Predictive modelling

= Approximation of functions from samples

STATISTICAL MACHINE LEARNING

= Learning from precedents

= Recovering dependencies from empirical data

= Predictive modelling

= Approximation of functions from samples!

LOSS FUNCTION(1)

We need to measure the quality of predictions.

This is done by using the **loss function**

For a single object with features x

we have the prediction $\hat{y} = f(x, \theta)$ and we have the true value y

We use a measure $l(y, \hat{y})$ for the discrepancy.

LOSS FUNCTION(2)

For a single object with features x

we have the prediction $\hat{y} = f(x, \theta)$ and we have the true value y

we use a measure $l(y, \hat{y})$ for the discrepancy.

Examples: mean squared error $\|y - \hat{y}\|^2$ for regression

$[y \neq \hat{y}]$ for classification (accuracy).

Averaging over the whole training set gives the loss function, known as **empirical risk:**

$$Q(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, \hat{y}_i)$$

We are minimizing the empirical risk!

MODEL SELECTION

How to parametrize the function $y = f(x, \theta)$?

Of course, it depends on the application!

We will describe the simplest models (linear models) in the next lecture.

Today, we will discuss how we measure the accuracy.

CLASSIFICATION TASK: METRICS

Suppose we have a binary classification task.

How we measure the quality of the model?

CLASSIFICATION TASK: METRICS (1)

Suppose we have a binary classification task.

How we measure the quality of the model?

Simple approach: **accuracy** (our model has predicted the correct class in 90% of the cases).

When this metrics is good? When it is bad?

CLASSIFICATION TASK: METRICS (2)

Simple approach: **accuracy** (our model has predicted the correct class in 90% of the cases).

Accuracy is a bad metric for classification when classes are **unbalanced**

Example: Predict the litho type during drilling. 90% of the type there is sand,
so the model that always predicts sand (i.e., constant function) gives 90% of the accuracy.

What to do in the unbalanced classes?

CLASSIFICATION TASK: METRICS (2)

The correct approach is more complicated. Suppose we have two classes, Positive and Negative. We need to measure:

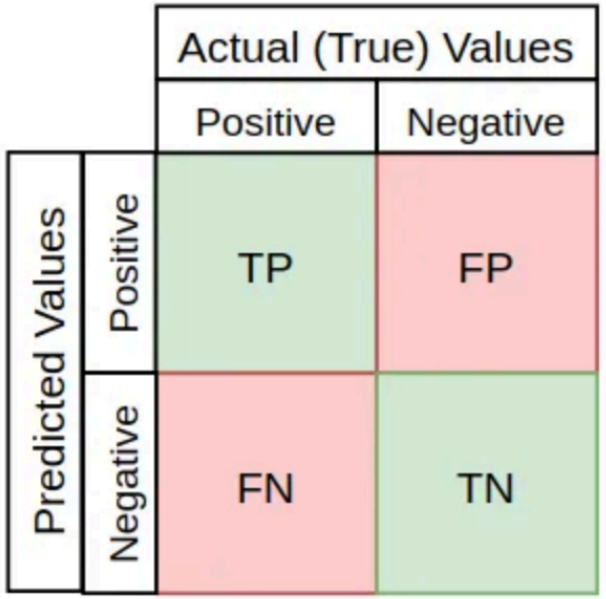
True Positive: model predicted positive and true label is positive

True Negative: model predicted negative and true label was negative

False Positive: model predicted positive and true label was negative

False Negative: model predicted negative and true label was positive

This can be visualized in the form of confusion matrix



CLASSIFICATION TASK: METRICS (3)

What can we estimate from this matrix?

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = 70\,\%$$

Precision measures how many positive corrections are made correct:

$$\text{Precision} = \frac{TP}{TP + FP} = 71.4\,\%$$

Recall measure how many positive cases have been predicted from total positive in the data (that we don't miss)

$$\text{Recall} = \frac{TP}{TP + FN} = 78.9\,\%$$

F1-score combines precision and recall into one number by taking harmonic mean:

$$\text{F1} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

		Actual (True) Values	
		Cancer	No Cancer
Predicted Values	Cancer	45	18
	No Cancer	12	25

CLASSIFICATION TASK: F1-SCORE

F1-score combines precision and recall into one number by taking harmonic mean:

$$F1 = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Two get high F1 score, we need to have high precision and high recall. For the cancer example, it will be 75%

If Precision is 0.01 and Recall is 1.0, the arithmetic mean will be around 50%, while F1-score will be approx 2%

CLASSIFICATION TASK: ROC CURVE

More advanced metric uses ROC (Receiver Operating Characteristics)

The model predicts not the class, but the probability of the class, say 0.6 for being Positive.

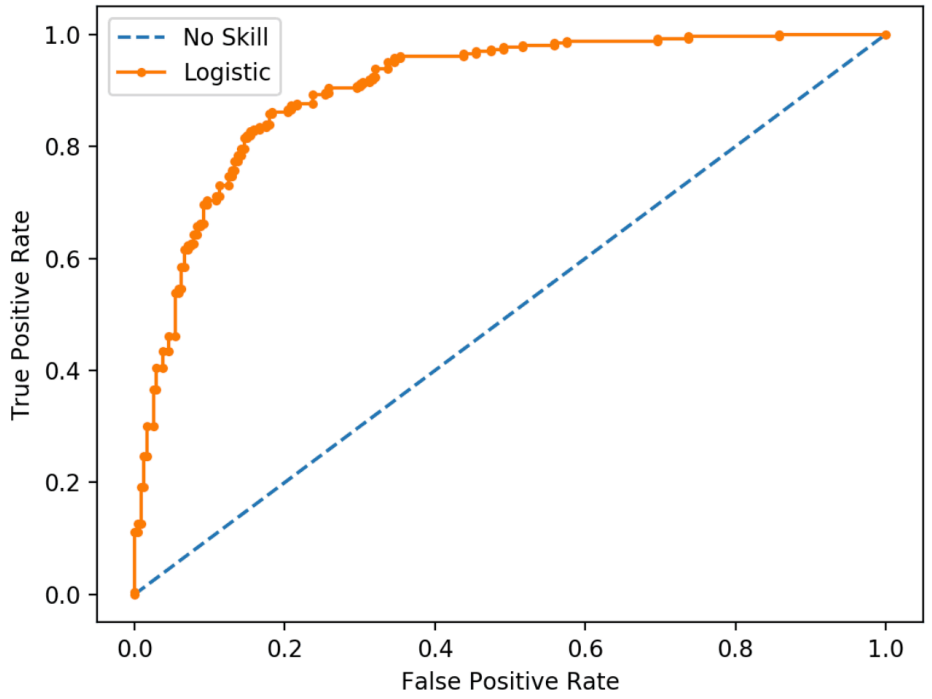
Then, we can select a threshold: if output is greater than t, we say the output is positive.

With threshold 0.55 the class will be predicted as positive, with 0.7 — as Negative
(remember the cancer example!)

CLASSIFICATION TASK: ROC CURVE

ROC curve plots True Positive vs False Positive values for different thresholds.

A random guess will give a straight line, more advanced models look like an orange line.

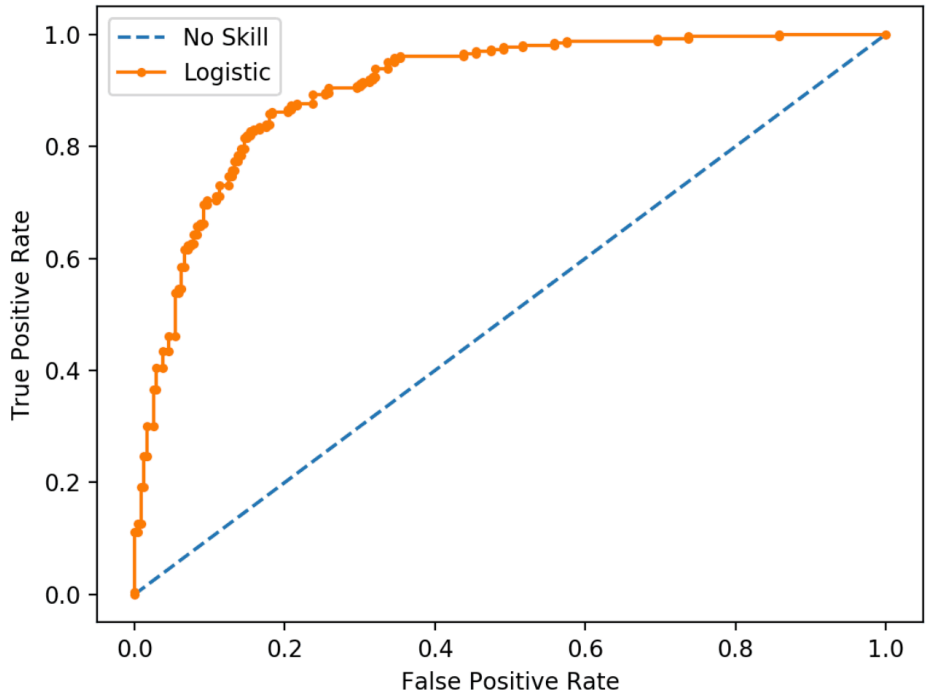


CLASSIFICATION TASK: ROC-AUC

AUC (Area under the Curve) is literally the area under the curve.

Sometimes referred to as ROC-AUC.

ROC has to be used, when the classes are **balanced**

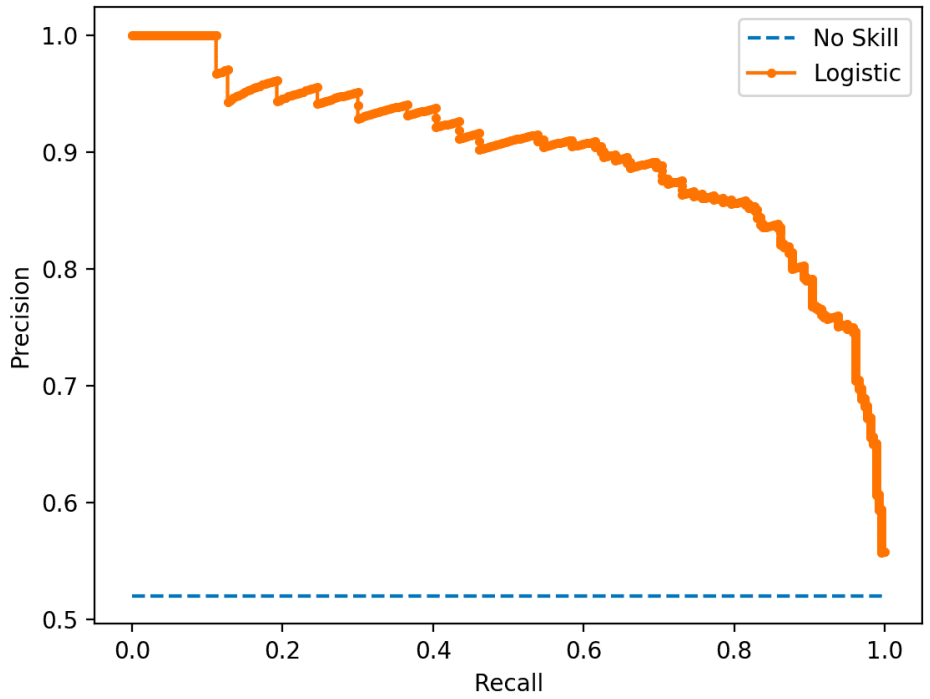


CLASSIFICATION TASK: PRECISION-RECALL

ROC has to be used, when the classes are **balanced.**

If the classes are unbalanced, it is better to use precision-recall curves

Q: Why the blue line (random guess) has the following form?

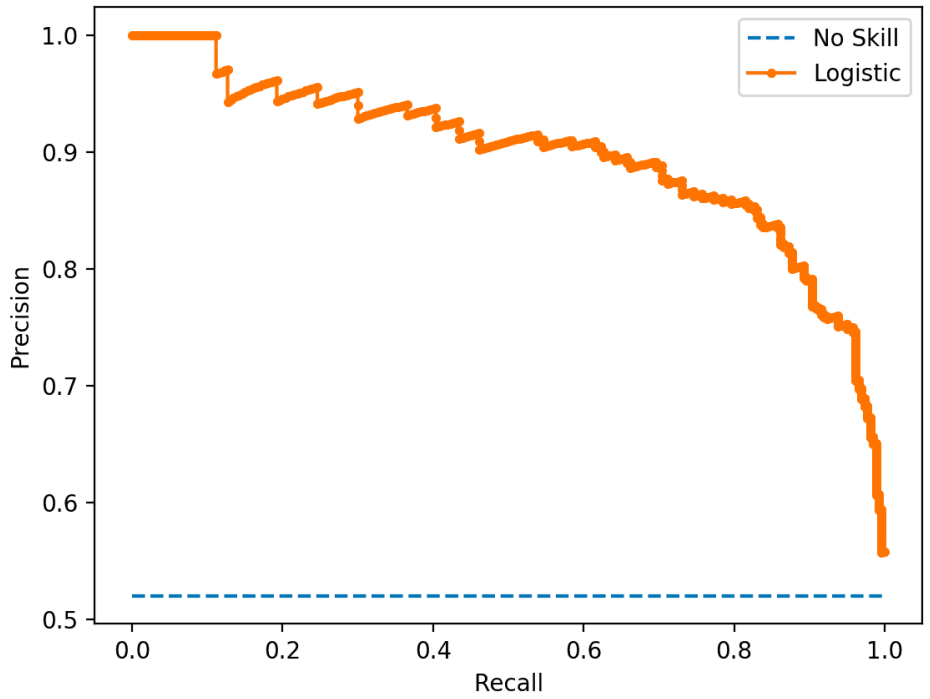


CLASSIFICATION TASK: PRECISION-RECALL

ROC has to be used, when the classes are **balanced.**

If the classes are unbalanced, it is better to use precision-recall curves

Q: Why the blue line (random guess) has the following form?



METRICS FOR REGRESSION

For regression tasks, the situation is simpler.

Typically, **Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error** are

Question: What is more robust to outliers in the data?

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{\text{real}} - y_i^{\text{pred}}\right|$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{real}} - y_i^{\text{pred}}\right)^2$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{real}} - y_i^{\text{pred}}\right)^2}$$

METRICS FOR REGRESSION (2)

For regression tasks the situation is simpler.

MSE/MAE/RMSE are not scale invariant: if we multiply the output by 10, the error is multiplied. We can measure **relative error.**

In statistics, the standard metrics is R^2 score, which is just linear correlation between output and predicted values.

The baseline: predict mean value.

If the R^2 is negative, we are worse than this baseline!

Mean value: $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Sum of squares of the residuals: ($\hat{y}_i$ are predictions)

$$SS_{\text{res}} = \sum_i \left(y_i - \hat{y}_i\right)^2 = \sum_i e_i^2$$

Variance:

$$SS_{\text{tot}} = \sum \left(y_i - \bar{y}\right)^2$$

R2-score:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

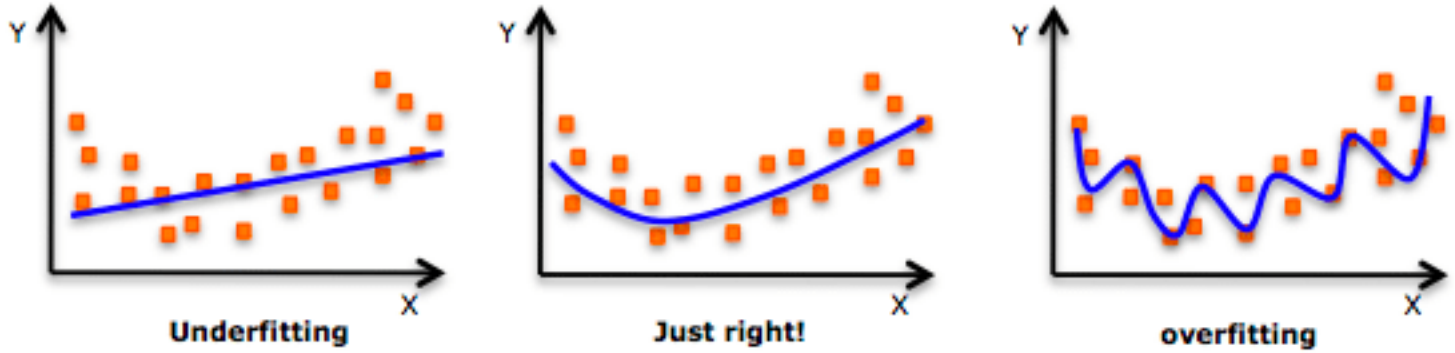
HOW MUCH DATA DO WE NEED?

There are several regimes in training.

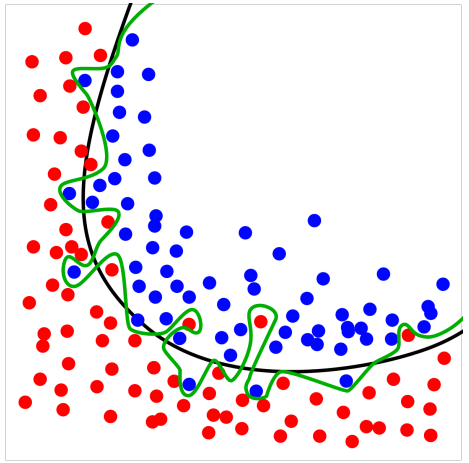
Underfitting: the number of parameters is not enough to represent the data

Normal: the model trained well, the train and test accuracy is ok

Overfitting: train accuracy is ok, but test accuracy is bad.



An example of overfitting, underfitting and a model that's "just right!"



RECAP OF LECTURE 2

- Supervised learning/unsupervised learning.
- Classification/regression problems. Accuracy metrics (precision, recall, ROC-AUC scores)
- Concept of loss functions
- Overfitting / underfitting

NEXT LECTURE

Classical ML:

Linear regression

Logistic regression

Support vector machine