Lecture #5: Machine learning for materials sience pt. 1

Slides by Artem Dembitskiy

Course roadmap

- Class #1: What is materials informatics
  - Trial-and-error paradigm
  - Python crash course
- Class #2: Python libraries for atomistic modelling of materials
  - ASE, Pymatgen
  - NN list
- Class #3: Data in materials science
  - The Materials project
  - Phase diagrams
- Class #4: Data exploration, visualization, and fitting
  - E_g vs. electronegativity
  - Ionic conductivity

Goals

- Overview of materials science problems solved with ML
- Overview of selected regression models
- Learn how to train and evaluate the model

Lecture overview

- ML tasks
- Why and when use ML?
- Property and descriptor
- Linear regression
- Cost function
- Ridge regression
- Gradient descent
- Decision tree and Random Forest
- Model evaluation

Why use ML?

- "Machine learning algorithms aim to optimize the performance of a certain task by using examples and/or past experience."*
- "Study of statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions" (wiki)
- Machine learning accelerates materials design by efficiently learning the structure-property relationship from data to predict unknown properties with minimal experimental or computational effort

*Recent advances and applications of machine learning in solid-state materials science

Example: Band gap estimation

- Experiment (UV-Vis absorption) ~1 week
  - synthesis
  - several measurements
  - equipment required
- DFT HSE ~ 1-10 days
  - crystal structure required
  - computationally expensive

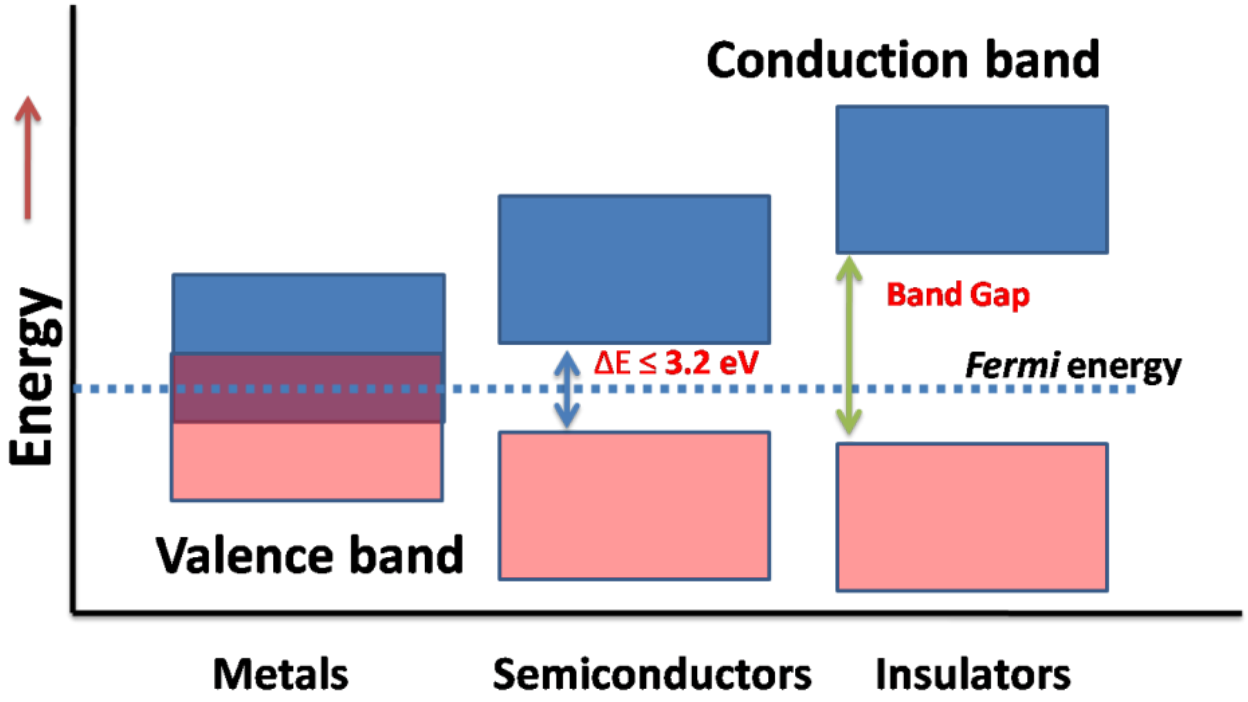


Image source

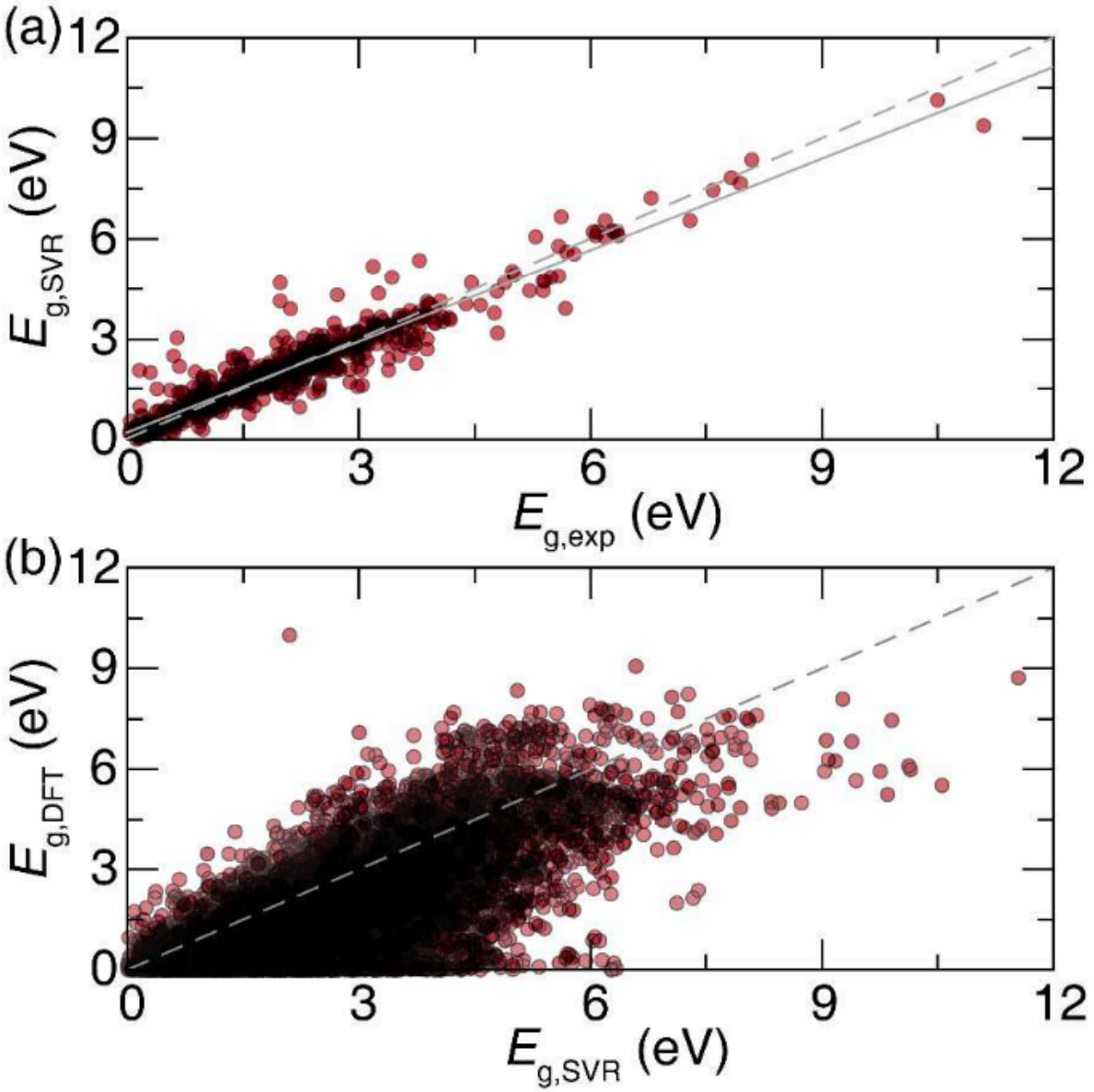
What if we have 10,000 candidates?

How many years would it take to find the best material?

Surrogate model

- takes < 1 second to predict the property
- trained on existing data
- chemical composition only
- simple laptop is sufficient

SVR - Support Vector Regression



Limitations

- High quality data is required to fit the model
- Accuracy is good for screening stages (RMSE = 0.45 eV)
    - but poor for a more specific tasks

There is a trade off between speed and accuracy

(Classical) ML tasks

CLASSICAL MACHINE LEARNING

Data is pre-categorized or numerical

SUPERVISED

Predict a category

CLASSIFICATION

«Divide the socks by color»

Predict a number

REGRESSION

«Divide the ties by length»

Data is not labeled in any way

UNSUPERVISED

Divide by similarity

CLUSTERING

«Split up similar clothing into stacks»

Find hidden dependencies

DIMENSION REDUCTION (generalization)

«Make the best outfits from the given clothes»

Identify sequences

ASSOCIATION

«Find what clothes I often wear together»

Machine Learning for Everyone. In simple words. With real-world examples. Yes, again

Supervised learning

"Tries to find the unknown function that connects known inputs to unknown outputs"

Grain size of alloys

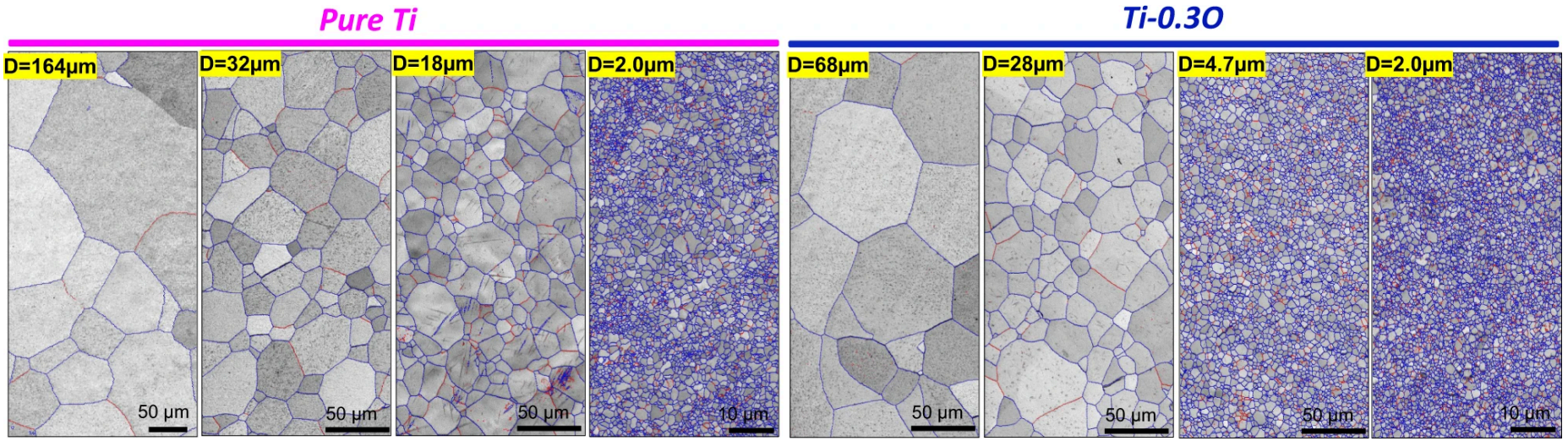
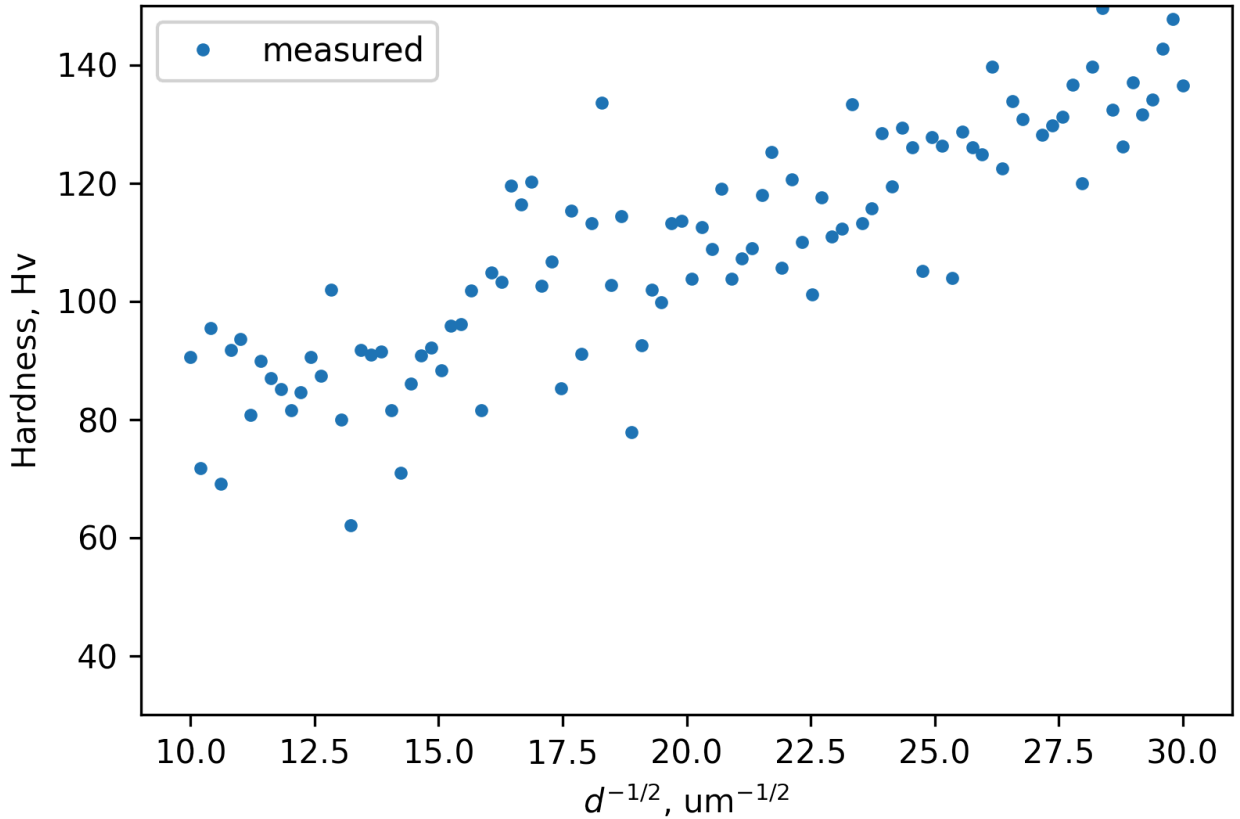


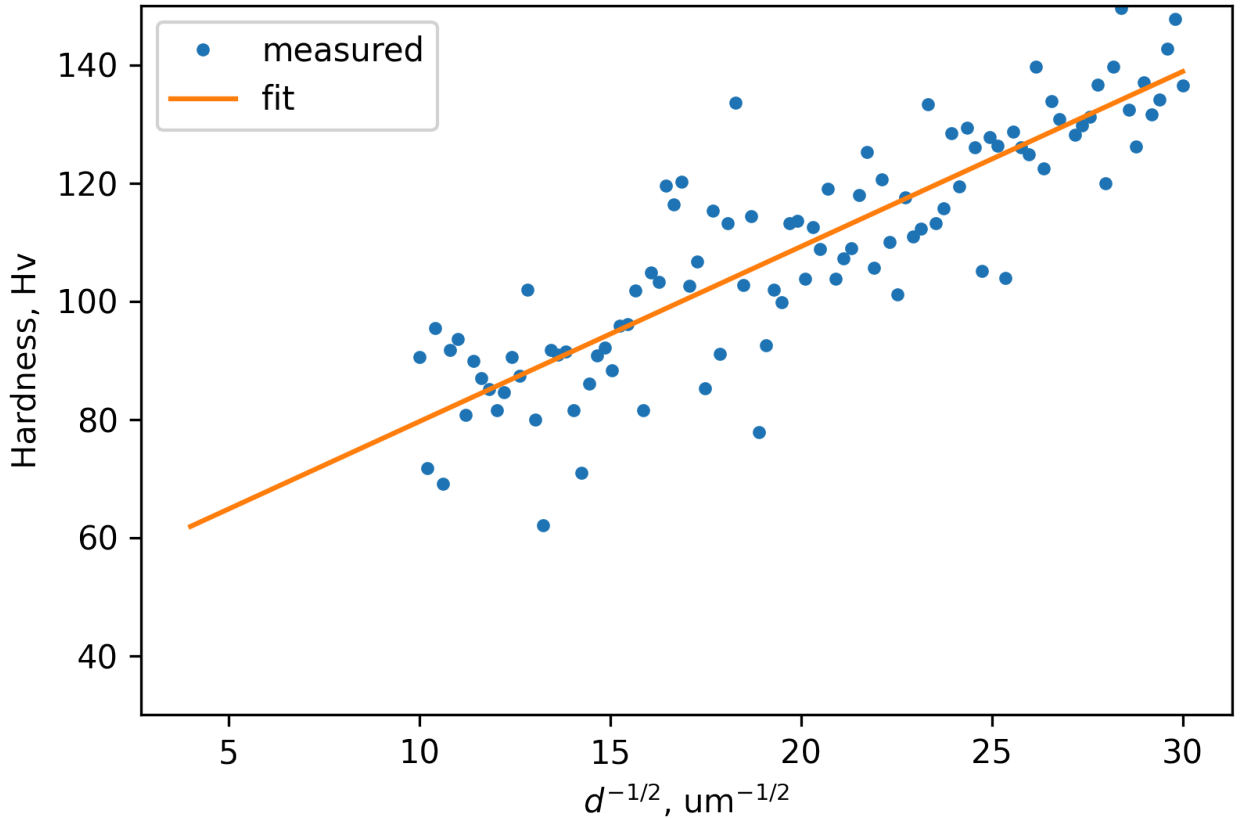
Image source

- input: grain size
- output: hardness



Model fit:

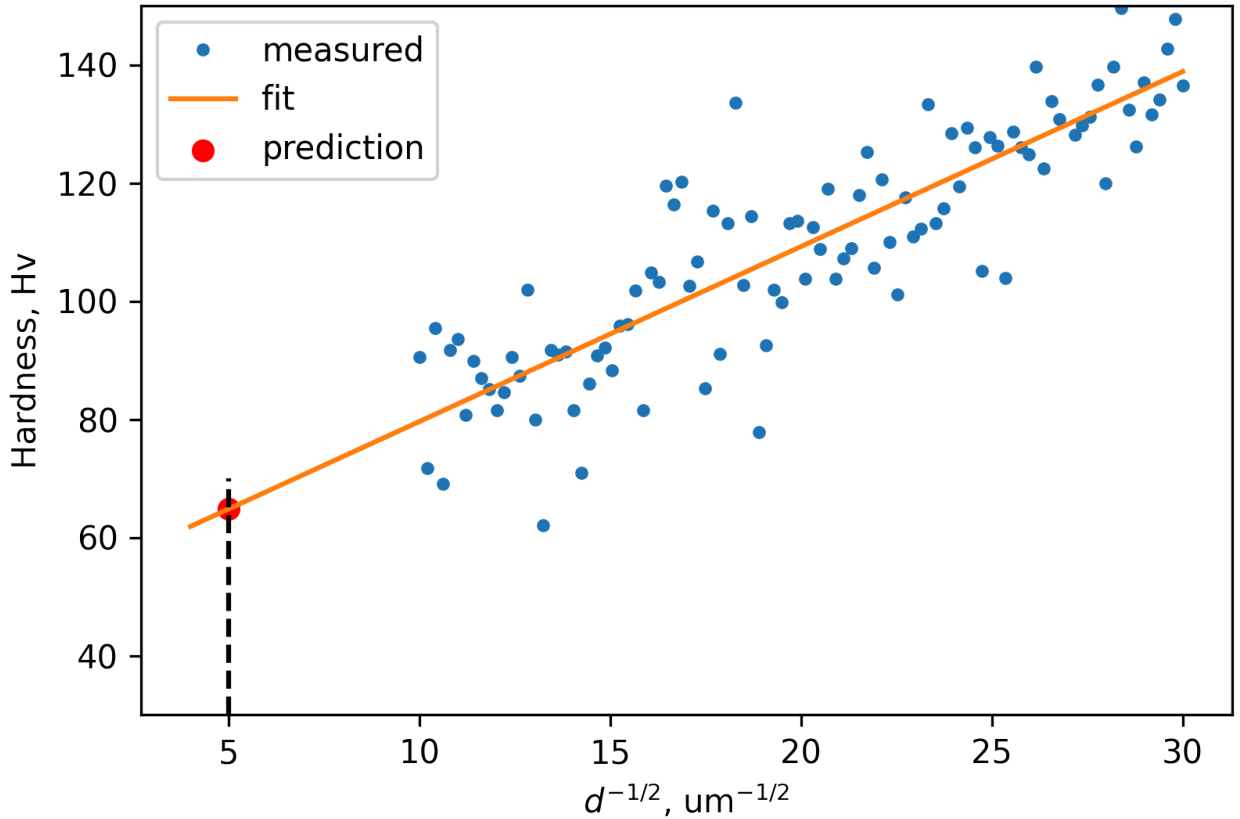
- input (features): grain size (known)
- output (labels): hardness (known)



Model prediction:

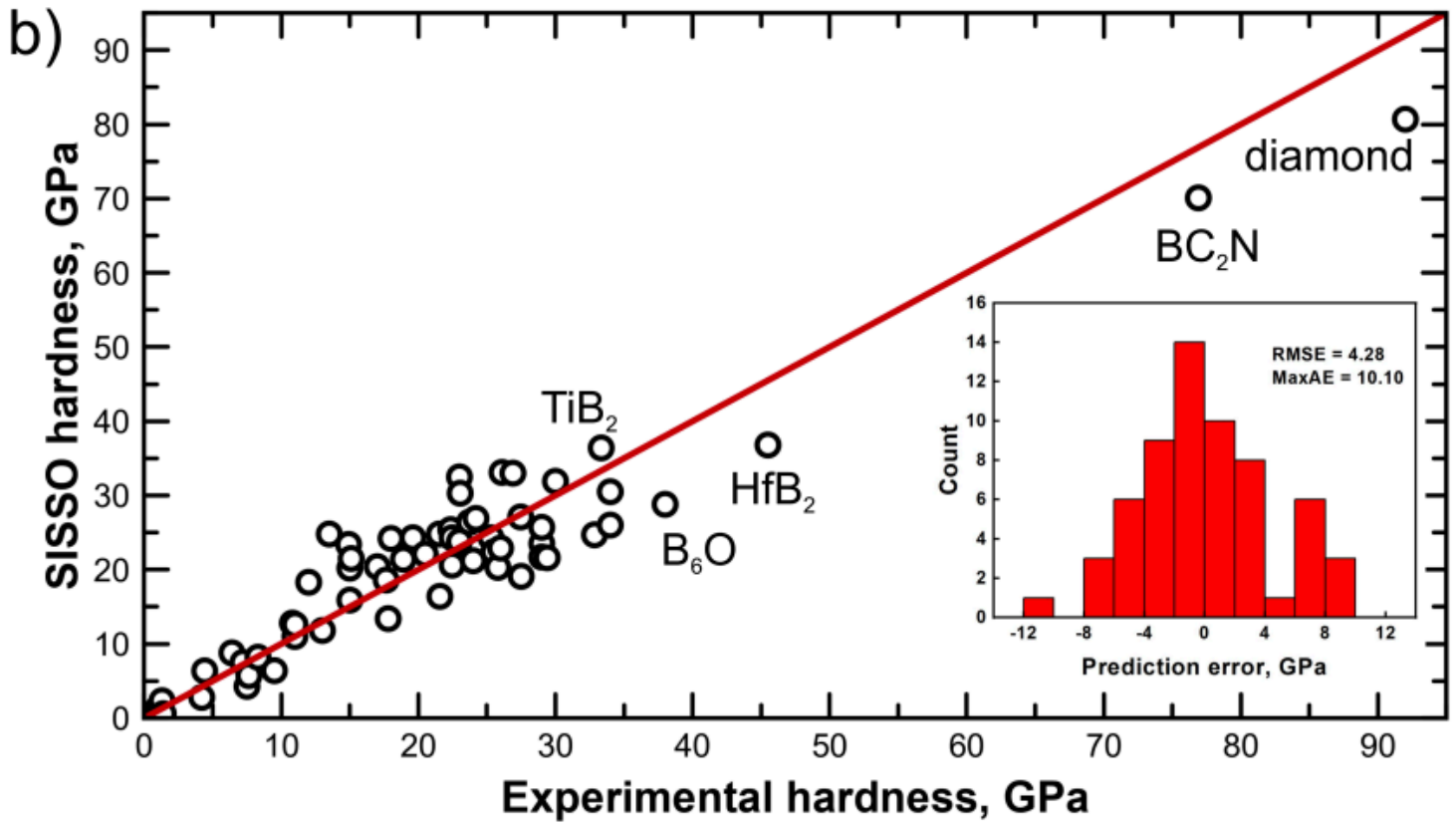
Hardness($d^{-1/2}$ = 5 um$^{-1/2}$)?

- input: grain size (known)
- output: hardness (unkown)



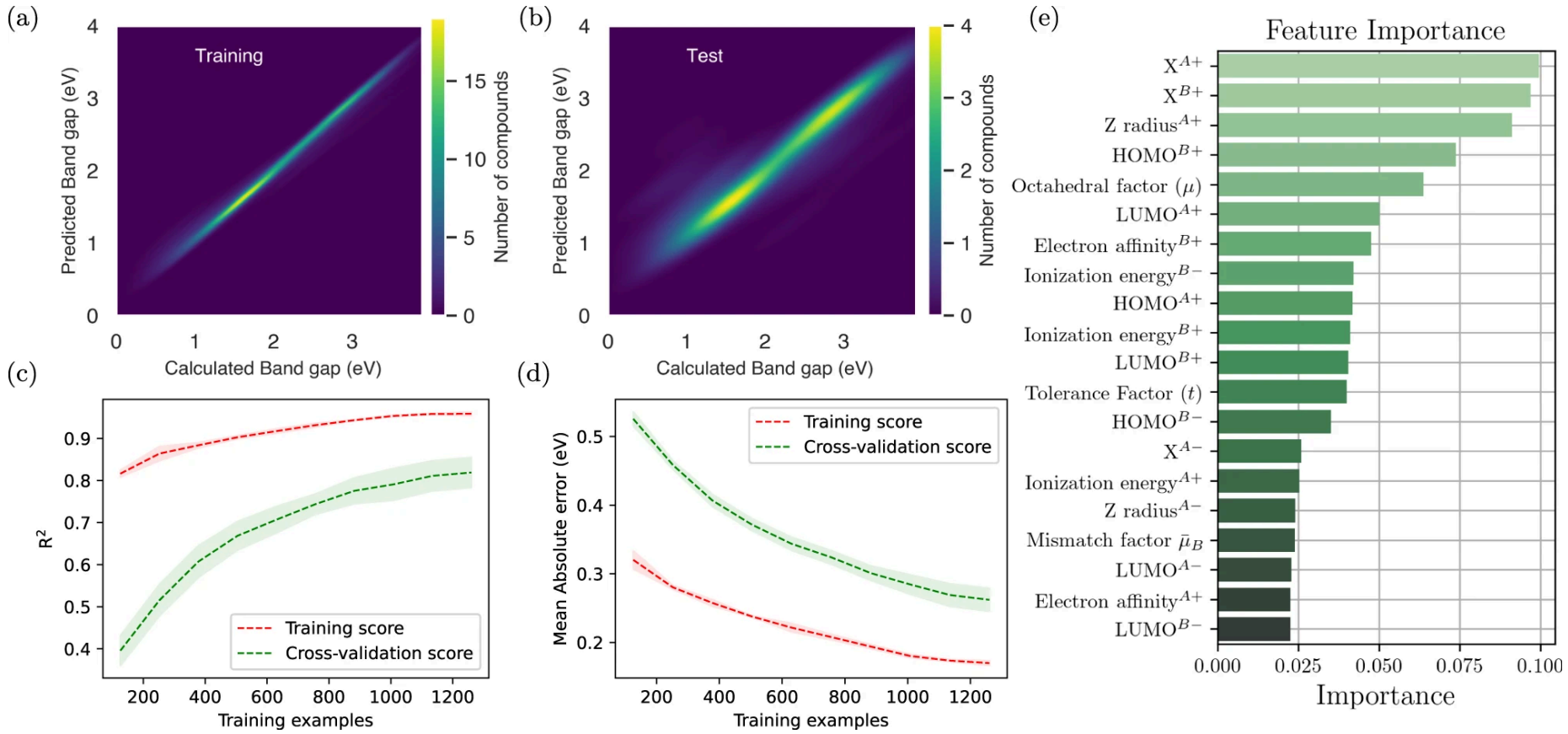
Supervised ML for materials science (examples)

Hardness



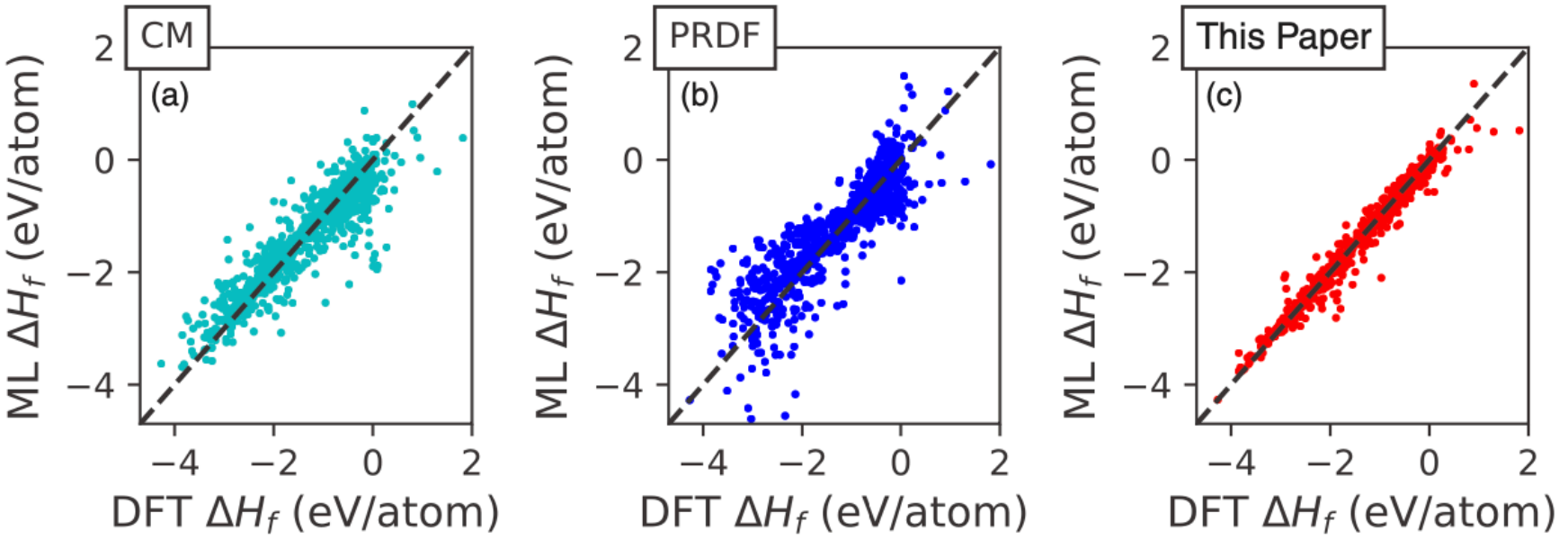
Material Hardness Descriptor Derived by Symbolic Regression

Band gap



Band gap predictions of double perovskite oxides using machine learning

Formation energy



R:	0.923	R:	0.858	R:	0.988
MAE:	0.25 eV/atom	MAE:	0.37 eV/atom	MAE:	0.09 eV/atom
RMSE:	0.38 eV/atom	RMSE:	0.50 eV/atom	RMSE:	0.15 eV/atom
Max Error:	2.03 eV/atom	Max Error:	2.21 eV/atom	Max Error:	1.30 eV/atom

Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations

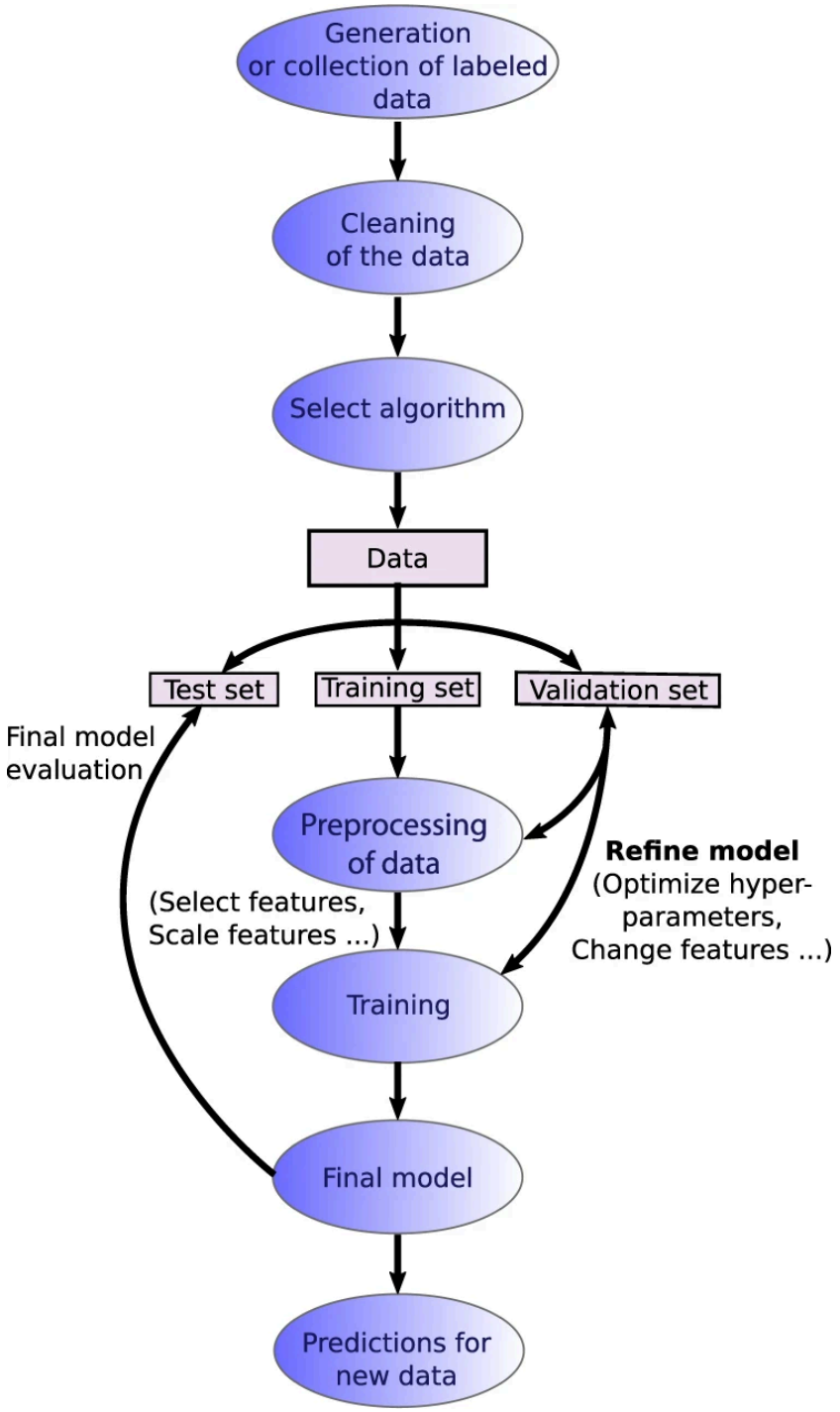
Self-supervised learning

... is about "finding patterns in unlabeled data"

(not taught on course)

Blog: Clustering by Deniz Gunay

Basic supervised learning workflow



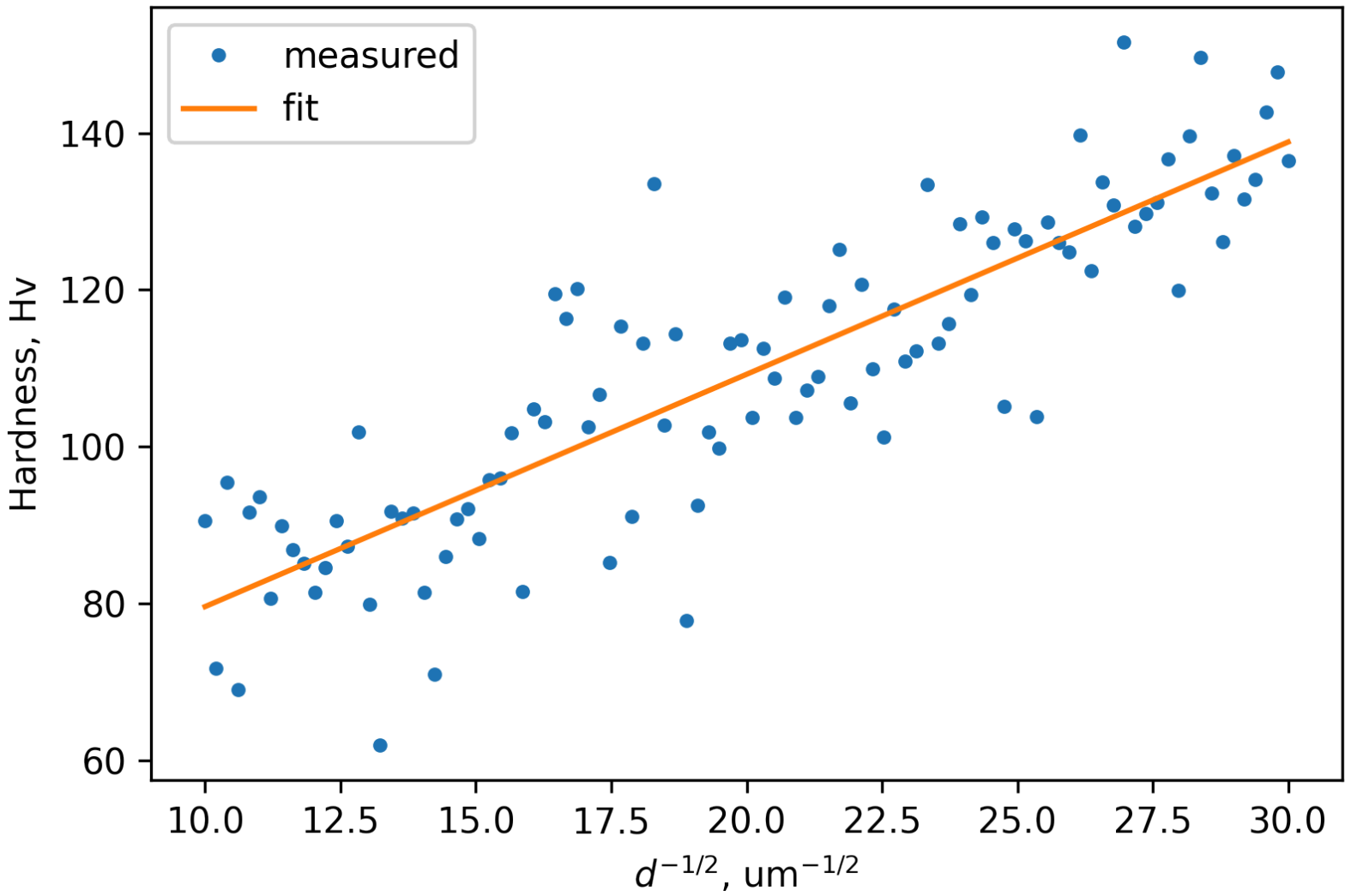
Attribution

- Parts of the next slides are adopted from the Machine Learning (MA060018) course given by prof. Evgeny Burnaev at Skolkovo Institute of Science and Technology
- Consider enrolling in this course if you are interested in fundamentals of ML

Materials:

- https://github.com/dzisandy/Machine-Learning/blob/master/Lecture Slides/Lecture_2_Regression_handout.pdf
- https://github.com/dzisandy/Machine-Learning/blob/master/Lecture Slides/Lecture_5_Decision_Trees_handout.pdf

(Multiple) Linear regression



Data samples: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_n, y_n)\}$

y_i - response variable (regressand), also called label/target/property

$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})$ - 1 x d vector of input features (regressors), also called features/descriptors

Assume that relationship between y and $\boldsymbol{x}$ is linear:

$$\hat{f} = b + w_0 x_0 + w_1 x_1 + \ldots w_d x_d$$
$$\hat{f} = \boldsymbol{w}\boldsymbol{x}^T + b$$

$\boldsymbol{w}$ and b are unknown parameters of the model

Task:

Find parameters of $\hat{f}$ that will minimize the error = $\frac{1}{n} \sum_i^n ||y_i - \hat{f}(\boldsymbol{x}_i)||^2$

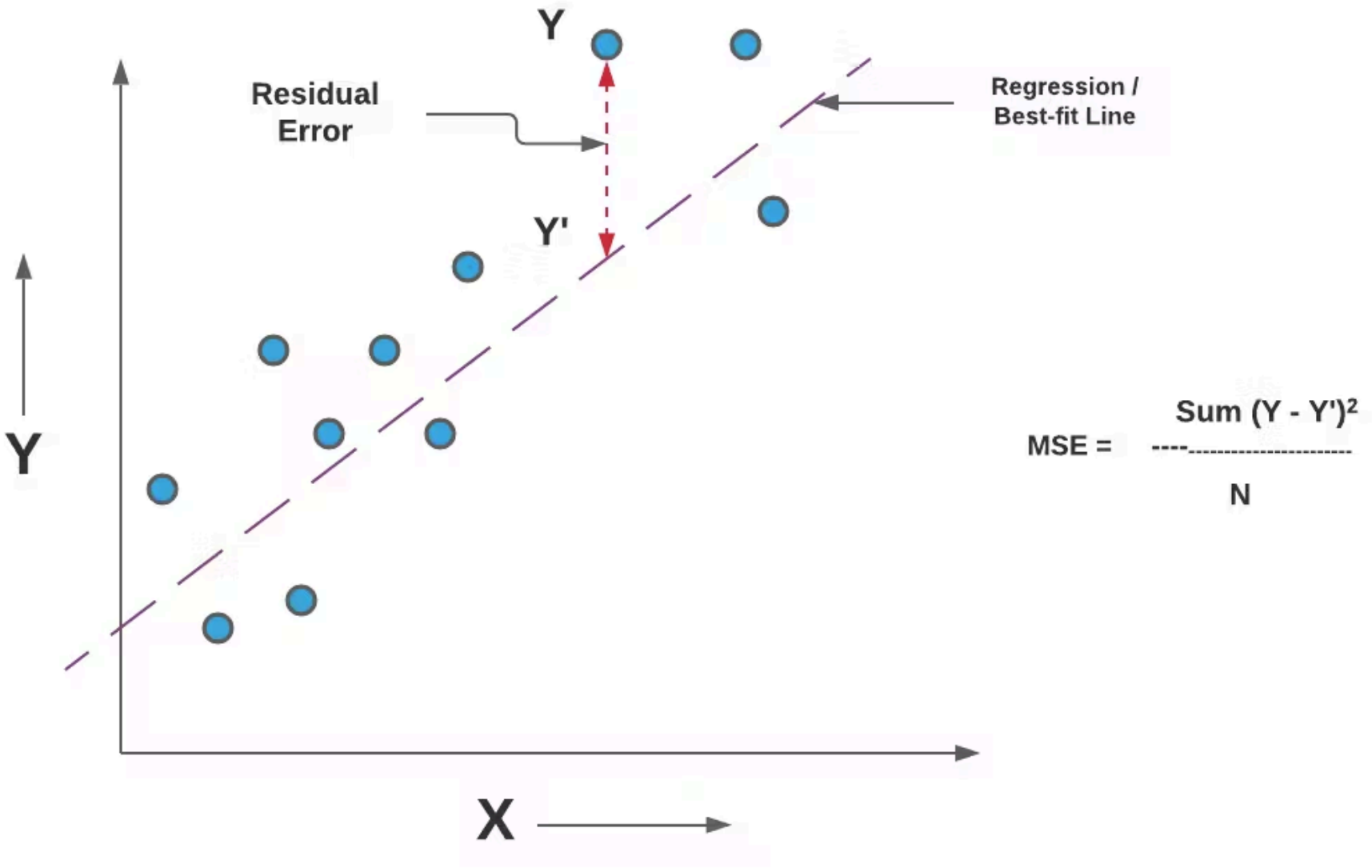


Image source

Loss function

$$J(\boldsymbol{w}, b) = \frac{1}{n} \sum_{i}^{n} (\boldsymbol{w}\boldsymbol{x}_{\boldsymbol{i}}^{T} + b - y_i)^2$$

Optimization problem

$$J(\boldsymbol{w}, b) \rightarrow \min_{w,b}$$

Matrix form

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \dots & & & & \\ x_{n1} & x_{n2} & \dots & x_{nd} & 1 \end{bmatrix}, W = \begin{bmatrix} w_1 \\ \dots \\ w_d \\ b \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix}$$

$$J(W) = \frac{1}{n}||XW - Y||^2 \rightarrow \min_W$$

Solution

$$J(W) = \frac{1}{n}||XW - Y||^2 \rightarrow \min_W$$

$$\nabla J(W) = 0$$

$$X^T XW = X^T Y$$

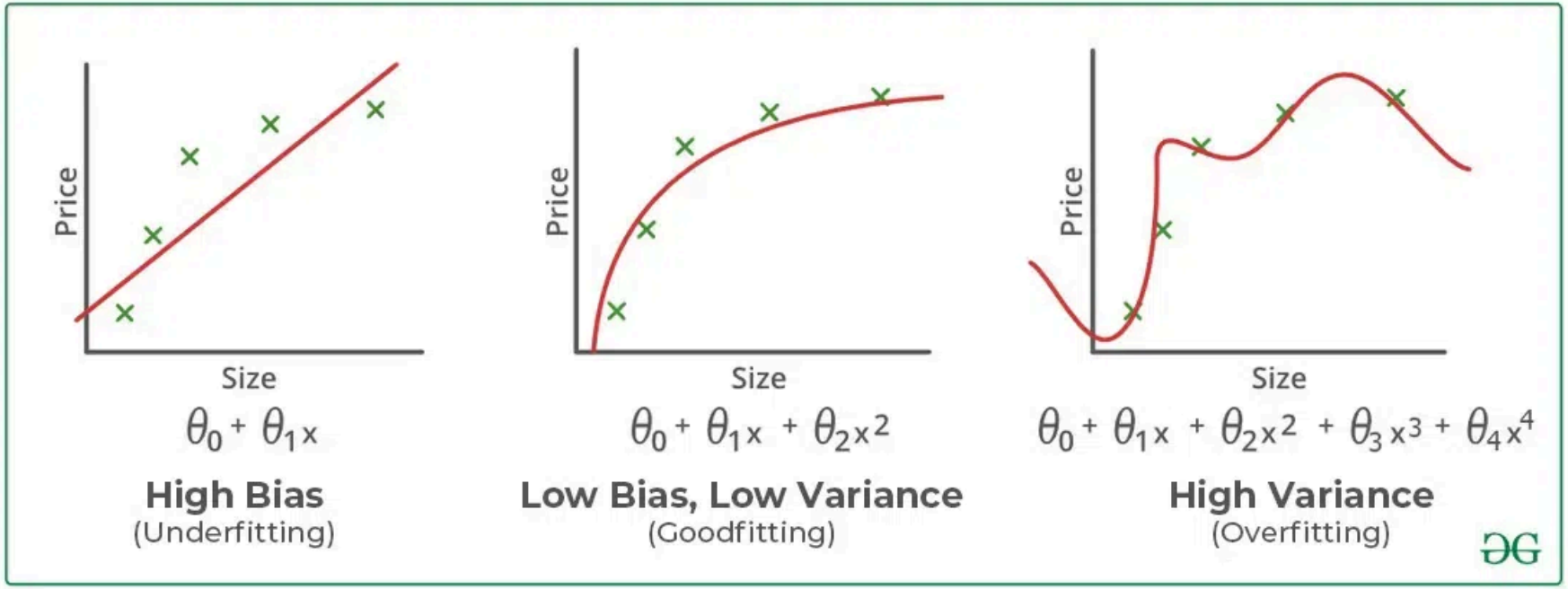
W = $(X^T X)^{-1} X^T Y$, if $X^T X$ invertible

$W = (X^T X)^{-1} X^T Y$, if $X^T X$ invertible

- Will fail if there are linearly dependent $\boldsymbol{x_i}$ in X
- Computationally expensive if $d >> 1, n >> 1$
- No regularization
    - model is not robust to noise and outliers
    - model can overfit

Generalization of a model

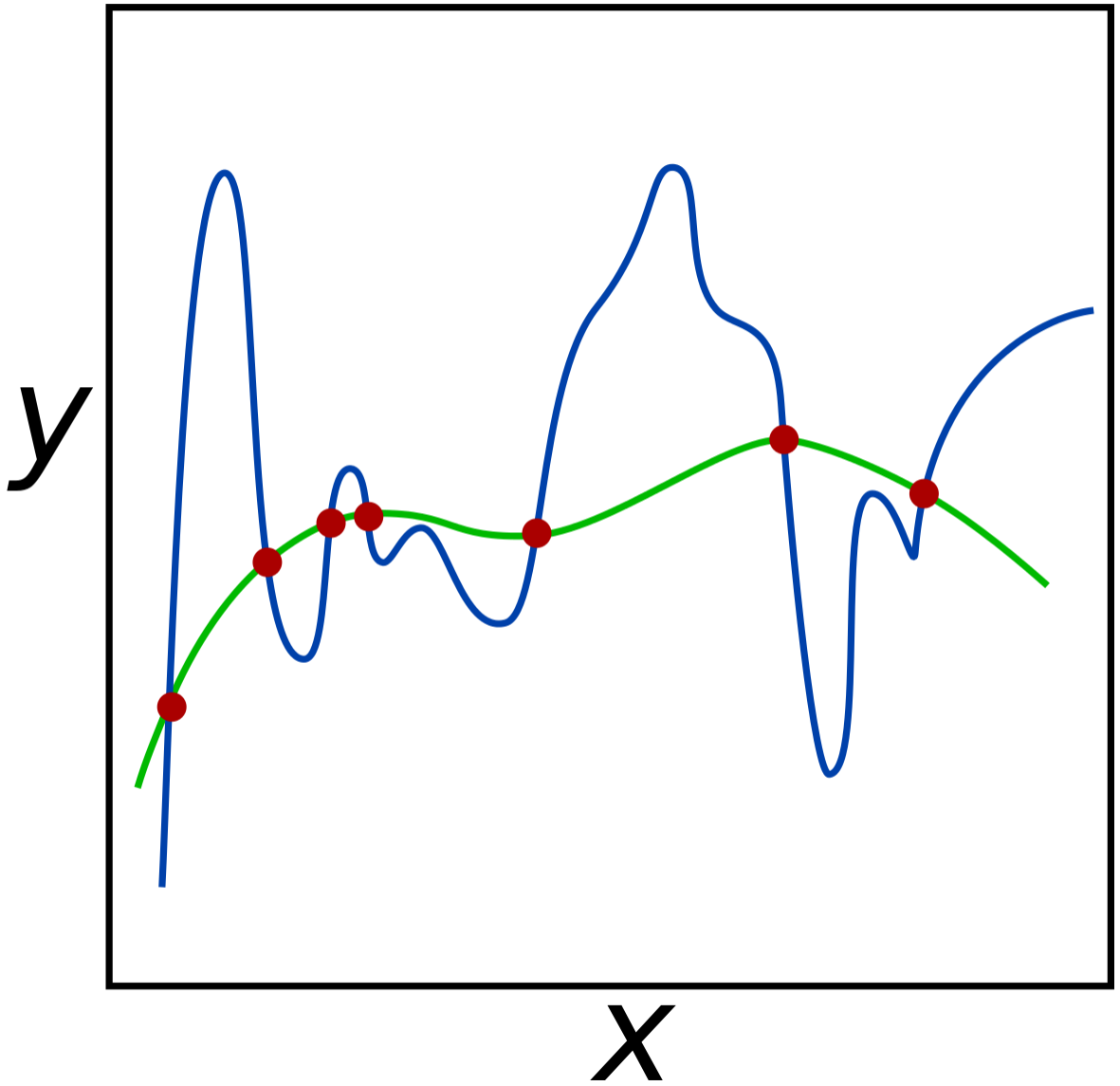
source



Regularization

Which model may generalize better to more points drawn from the underlying unknown distribution?

wiki



Ridge regression

$$J(\boldsymbol{w}, b) = \frac{1}{n} \sum_{i}^{n} (\boldsymbol{w}\boldsymbol{x}_{\boldsymbol{i}}^{T} + b - y_i)^2 + \lambda ||w||^2$$

$$J(\boldsymbol{w}, b) \rightarrow \min_{w,b}$$

Solution

$W = (X^T X + \lambda I)^{-1} X^T Y$, $(X^T X + \lambda I)^{-1}$ - always invertible

- we penalize large weights to correct overfitting
- model is more robust to noise and outliers
- the problem of near-singular $(X^T X)^{-1}$ is solved
- penalty defines the model’s bias-variance tradeoff

Iterative approach to find the best fit parameters

Instead of solving problem in one step

$$\theta = (X^T X)^{-1} X^T Y$$

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Image source

We solve it iteratively

Gradient descent steps

1. (Randomly) initialize weights
2. Compute the gradient of the cost function with respect to each parameter (partial derrivatives)
3. Update the weights of the model by taking steps in the opposite direction of the model
4. Repeat steps 2 and 3 iteratively to get the best parameter for the defined model

Cost Function

$$J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[h_\Theta(x_i) - y_i]^2$$

$h_\Theta(x_i)$: Predicted Value; y_i: True Value

Gradient Descent

$$\Theta_j = \Theta_j - \alpha\frac{\partial}{\partial\Theta_j}J(\Theta_0, \Theta_1)$$

α: Learning Rate

Now,

$$\frac{\partial}{\partial\Theta}J_\Theta = \frac{\partial}{\partial\Theta}\frac{1}{2m}\sum_{i=1}^{m}[h_\Theta(x_i) - y]^2$$

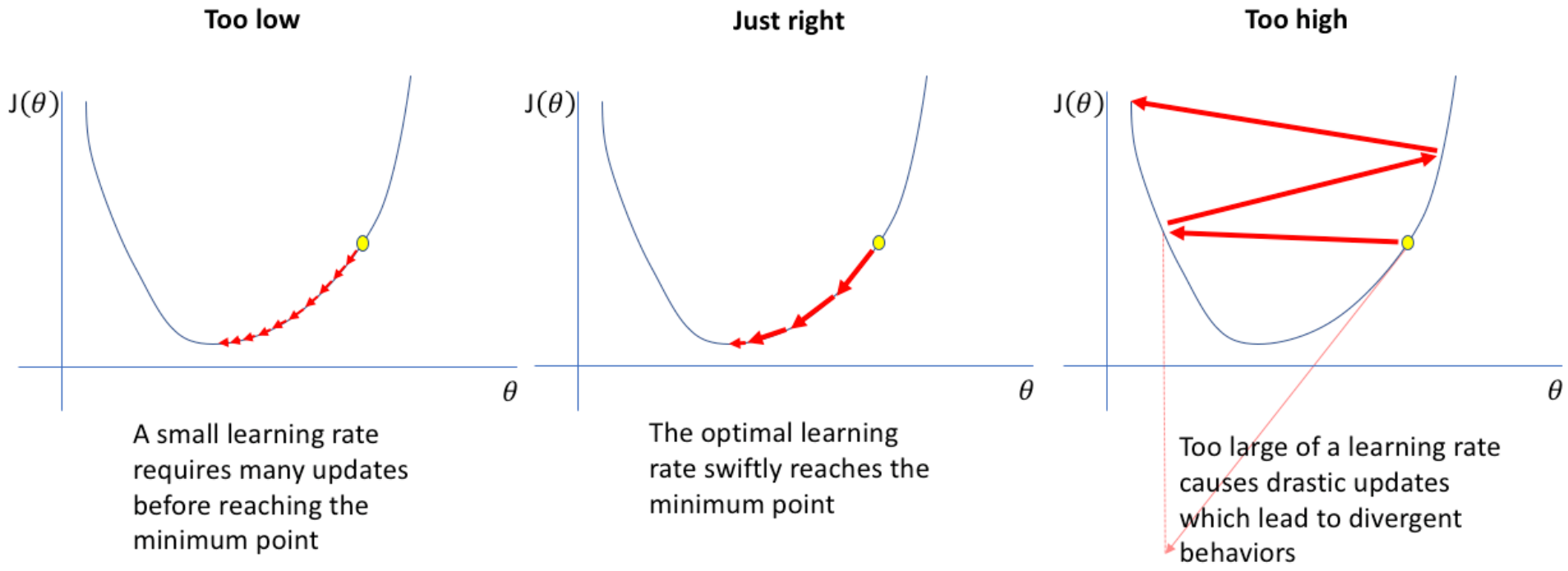
$$= \frac{1}{m}\sum_{i=1}^{m}(h_\Theta(x_i) - y)\frac{\partial}{\partial\Theta_j}(\Theta x_i - y)$$

$$= \frac{1}{m}(h_\Theta(x_i) - y)x_i$$

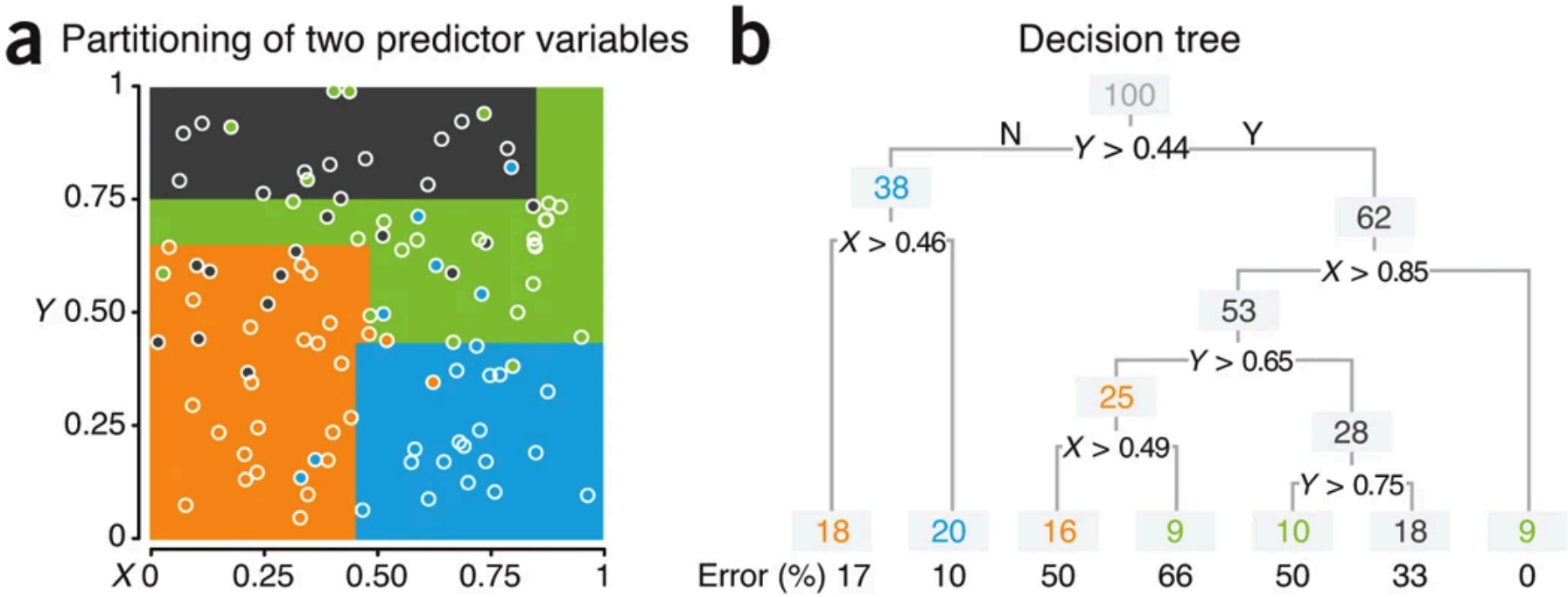
Therefore,

$$\Theta_j := \Theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}[(h_\Theta(x_i) - y)x_i]$$

Learning rate matters



Decision tree



Attributes

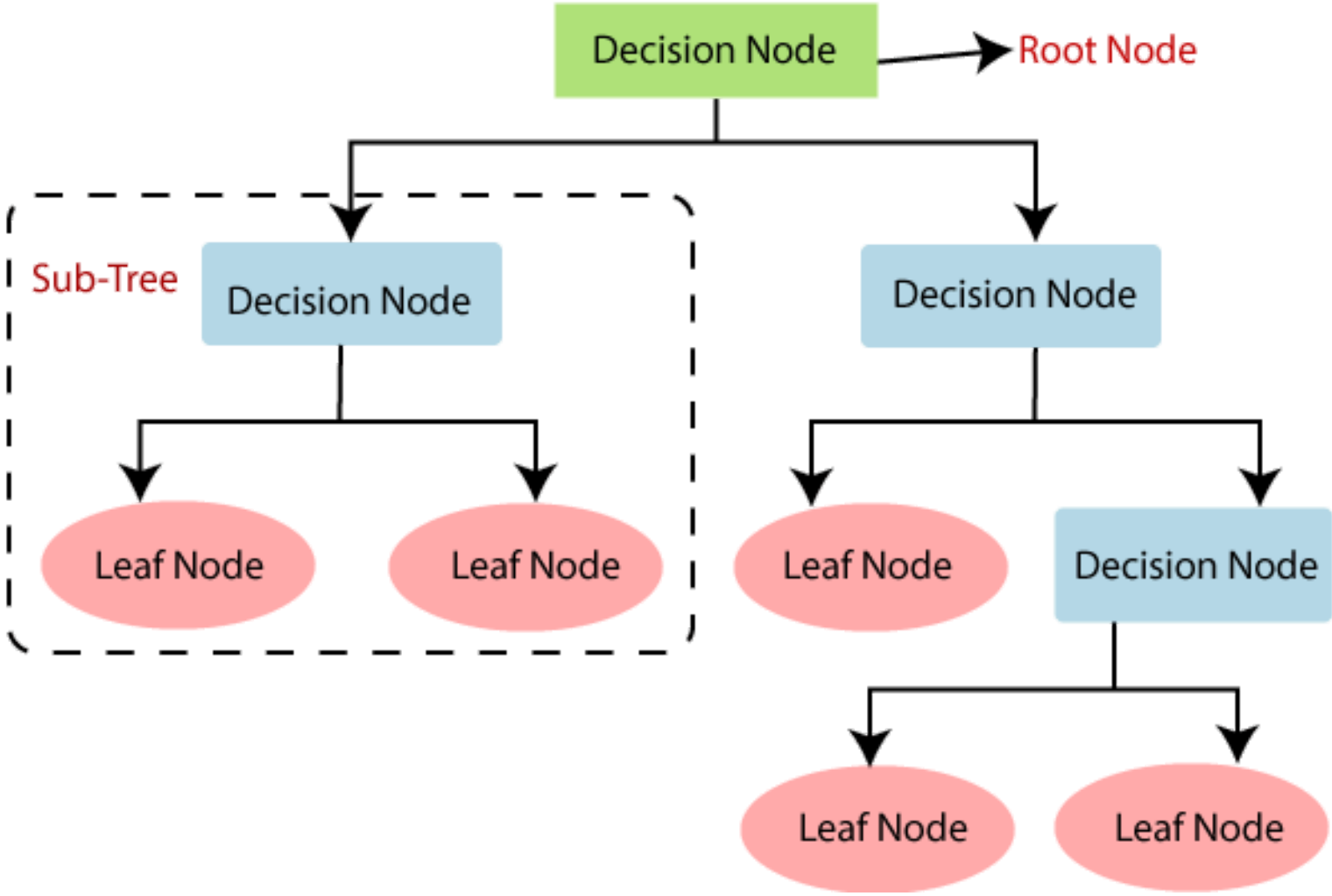


Image source

How to select splits?

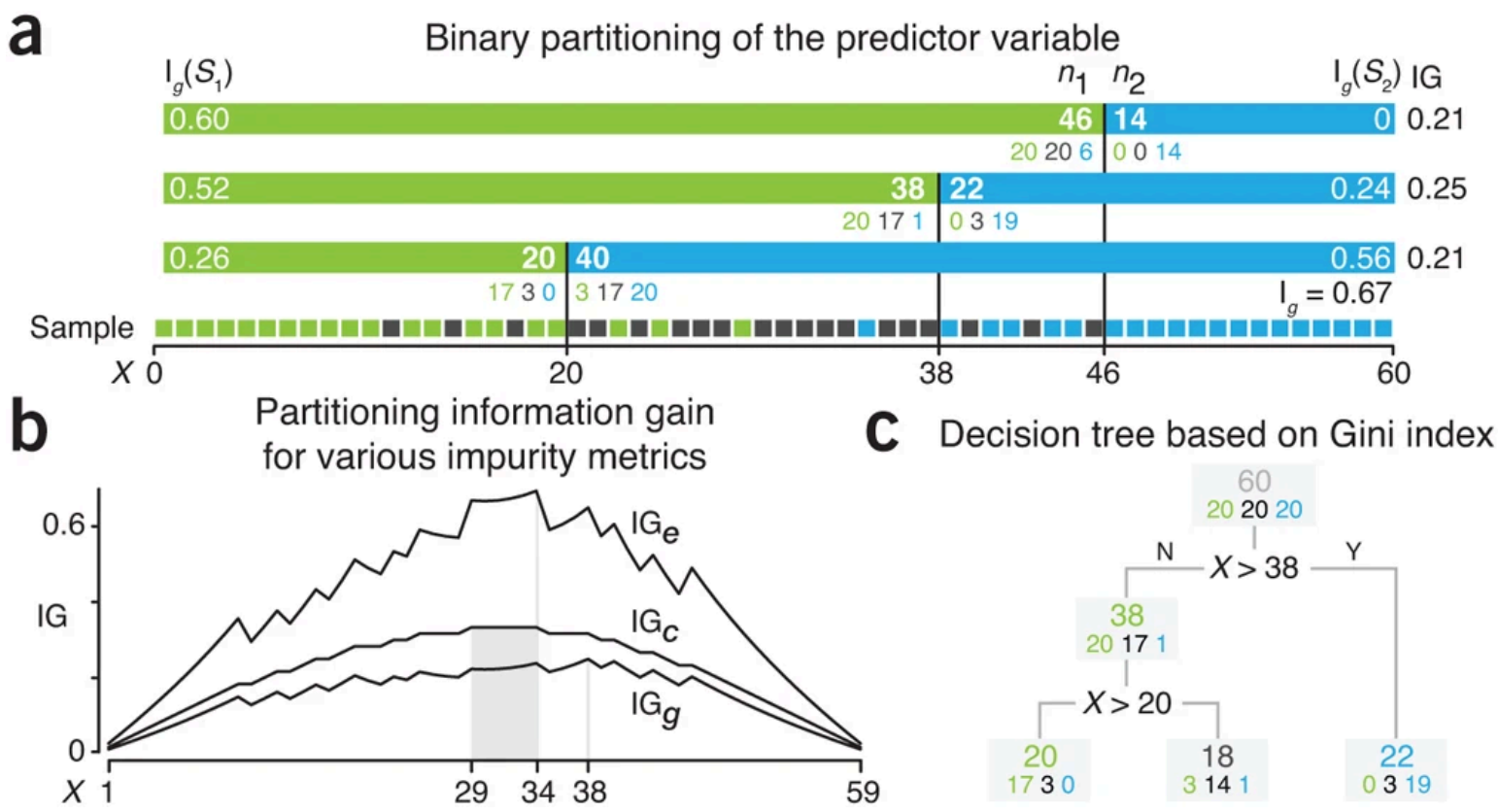
- Gini

$I_G = 1 - \sum p_i^2$, where p_i - proportion of the samples made up class i for a given node

- Entropy

$E = -\sum p_i \log_2 p_i$

for all $p_i ! = 0$
if all samples at the node are from the same class $I_H = 0$



Before split: 5 blue samples and 5 green samples

$$E = -\sum p_i \log_2 p_i = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

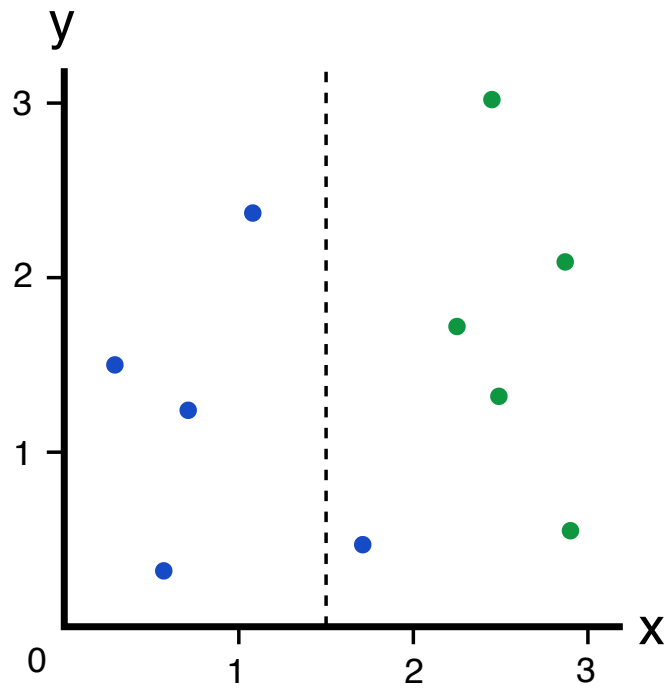
After split at $X = 1.5$

Left branch: $E_{left} = -\log_2 1 = 0$ (all samples are blue)

Right branch: $E_{right} = \frac{1}{6}\log_2 \frac{1}{6} + \frac{5}{6}\log_2 \frac{5}{6} = 0.65$

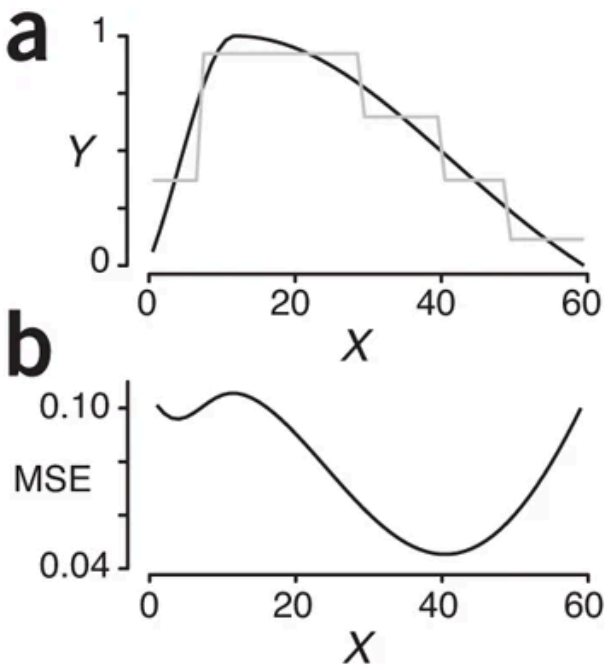
$$E_{split} = 0.4 * 0 + 0.6 * 0.65 = 0.39$$

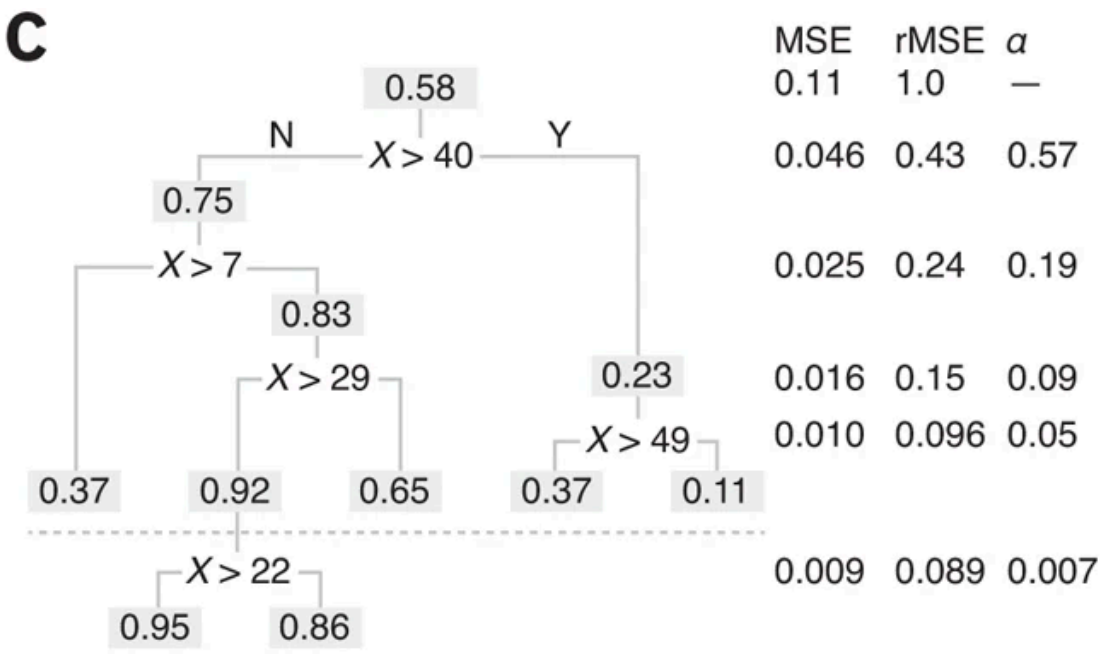
Information gain = $1 - 0.39 = 0.61$



Regression with decision trees

- Piecewise constant approximation





Pros

- Easy to interprete
- Categorical values can be used
- Solves non-linear problems

Cons

- Easy to overfit
- Computationally expensive

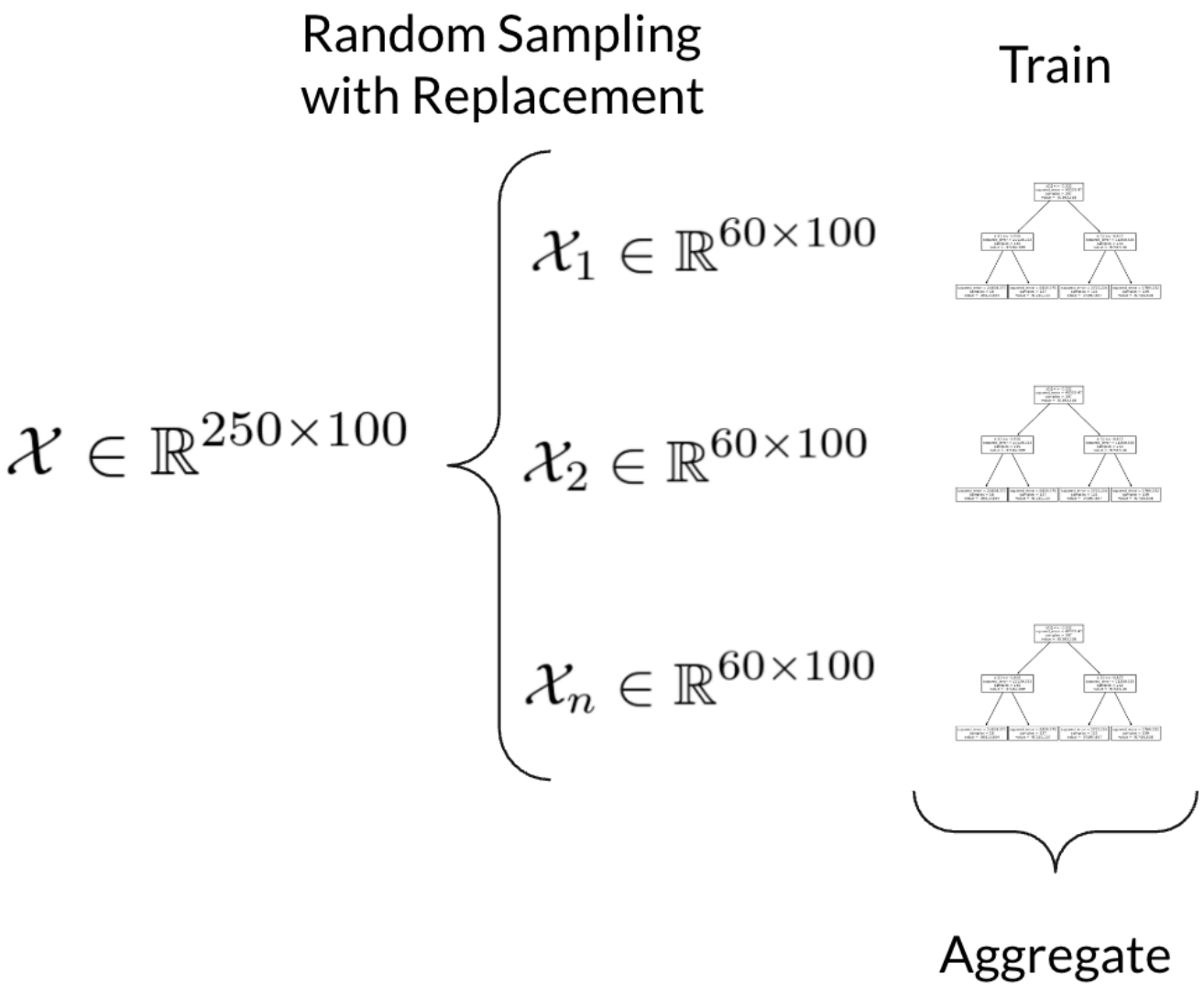
Random forest (Ensemble of weak learners)

Given n samples $\{X_i, y\}$

for b in $1..B$

- Bootstrap (duplicates are possible) n samples from the dataset
- Fit h_b decision tree on a randomly sampled set of features
- Predict unkown label as a mean prediction of B models

$\hat{y} = \frac{\sum h_b(x_b')}{B}$ - prediction



Pros

- Less prone to overfit
- Can compute feature importance (next lecture)

Cons

- High computational cost

Regression model evaluation (metrics)

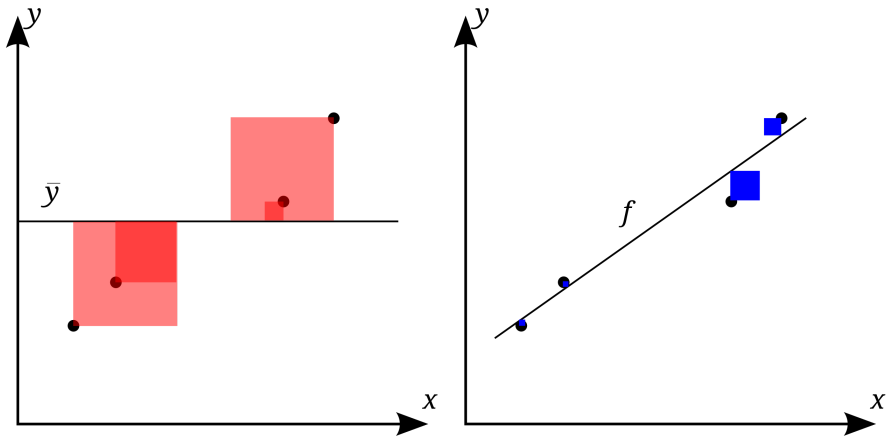
Mean Absolute Error - MAE

Root Mean Squared Error - RMSE

Coefficient of determination - R^2

R^2 score

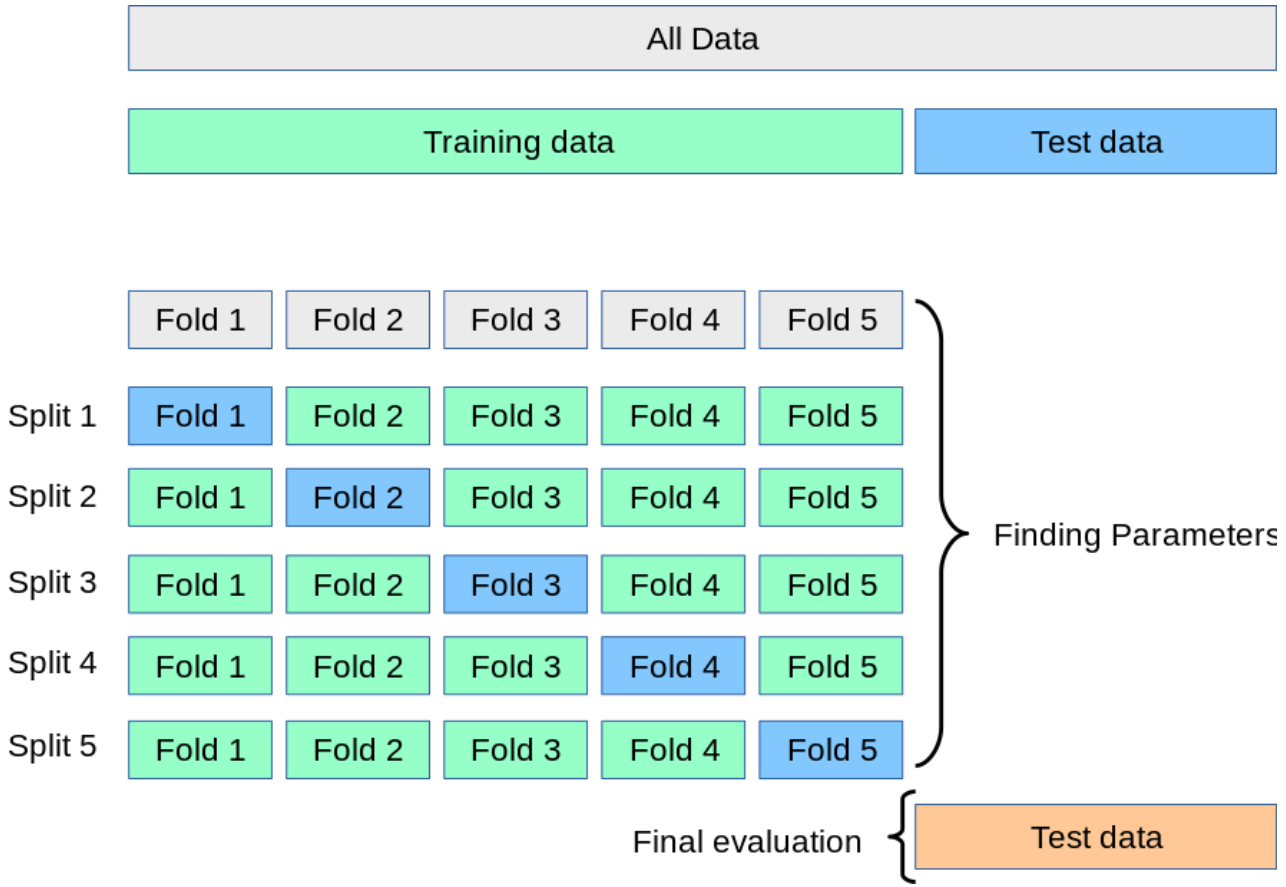
is a statistical measure of how well the regression predictions approximate the real data points. An R^2 of 1 indicates that the regression predictions perfectly fit the data. (wiki)



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

Train/test/val split

- train
  - fit the model
- val
  - tune hyperparameters
- test
  - get final scores



Hyperparameters?

- During the training (learning) process we find the best **parameters** (weights) of our model
- In addition we have **hyperparameters** which are usually fixed before the actual training process begins

Example:

- number of trees in random forest, or depth of a decision tree
- λ in Ridge Regression
- They control the learning process itself rather than being learned from the data.

These parameters should be tuned to get better performance of the model

When use ML?

Case #1

- I have 10 crystal structures for which I need to precisely calculate the band gap value
- The density functional theory (DFT) calculation takes about 2 days per structure

Should I use ML in this case?

Case #2

- I have 10,000 crystal structures for which I need to calculate the band gap value
- The DFT calculation takes about 10 minutes per structure
- I developed a featurization scheme that calculates features for a crystal structure
    - for further fitting of the model to predict the band gap
- The featurization takes about 9 minutes per structure
- I expect that regression model will possess a high accuracy (say, MAE = 0.1 eV)

Should I use ML in this case?

Case #3

- I have 10,000 crystal structures for which I need to calculate the band gap value
- The DFT calculation takes about 10 minutes per structure
- I developed a featurization scheme that calculates features for a crystal structure
    - for further fitting of the model to predict the band gap
- The featurization takes about 0.1 second per structure
- I expect that regression model will possess a moderate accuracy (say, MAE = 0.5 eV)

Should I use ML in this case?

About final projects

Final project (FP) should reflect the ILOs of the course

- Apply python libraries and data science tools to solve materials science problems
- Critically evaluate materials informatics literature
- Collect, generate and analyse materials science datasets, including identification of structure-property relationships

FP description

Write a 3 page article style report including

- Introduction
- Methods
- Results
- Discussion
- Conclusion
- Bibliography

Prepare a 5 minutes oral presentation

Example topics

- Dataset collection and analysis (EDA)
- Prediction of some property of materials
    - band gap, hardness, HOMO/LUMO, formation energy, etc.
- Screening of materials with superior properties (e.g. ionic conductors)
- MLMD study of some process
    - diffusion, adsorption
- Comparative analysis of models (aka benchmark)

Thank you for your attention!