Practical AI: regression practicum

Stanislav Protasov for
Harbour.Space University

Reading

https://scikit-learn.org/stable/modules/feature_selection.html

https://scikit-learn.org/stable/modules/linear_model.html

Agenda

1. ML framework and problems
2. Features
3. Linear models: OLS, Ridge, Lasso, [S]GD
4. Polynomial features with linear models
5. Binary classification using logistic regression

Important steps towards solution

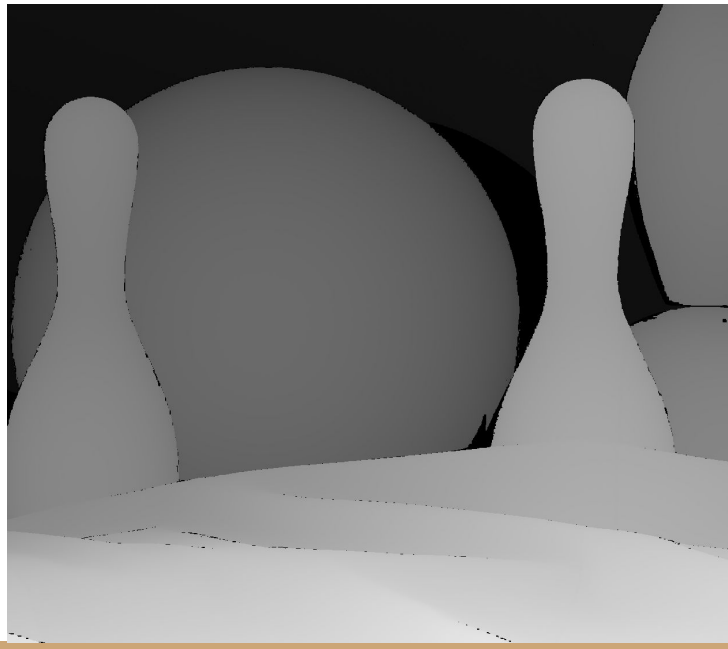
Top-level overview of how ML models are created

1. Find data
2. Prepare data
3. Prepare dataset
4. Train, validate, test
5. Measure quality
6. Save, deploy
7. Improve

1. Find your dataset

Dataset = samples + target (or ground truth)

1) Collect data for your task
2) Take the data from customer
3) Download publicly available dataset



1.1. Dataset and quality

Before you start training the model, be sure you understand:

- How do you measure the quality?
- CAN YOU?
- What are the values that will satisfy you?

Lab #1: Explaining the model, measuring quality

1) Explore naive-ml example.
    a) Consider difference between matrix inverse and LSA.
    b) Compute RMSE for both solutions
        i) Which of solutions is more accurate?
2) Find an approximation for GPD. Compute RMSE

2.1. Clean your data

- Clean
- Restore nulls
- Normalize
- Extend
- Augment
- Bootstrap
- ....

2. Split your data for training

1. **Train**
2. **Validate**
3. Test

Firstly your model is trained to **minimize error** on **training set**.

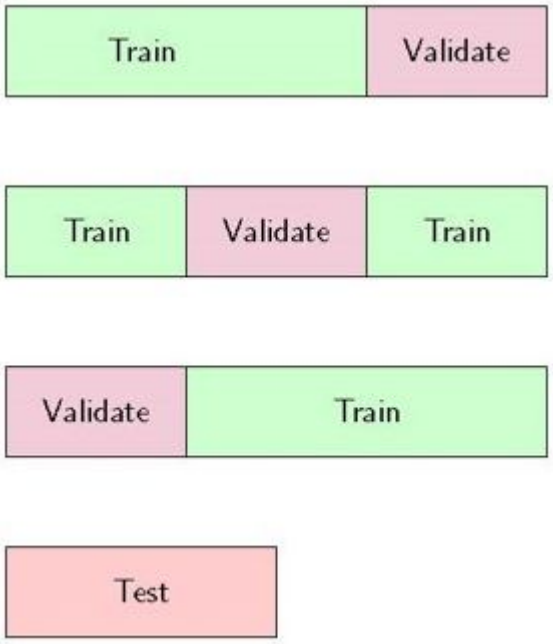
Validation data is used to (1) prevent overfitting (2) tune hyperparameters.

Parameters and hyperparameters that minimize error for **validation set** are desired result.

Test set is used to compute **quality results**. (Consider this as blind **acceptance** by customer).

... or

1) Split you data into train+validate and test sets.
2) Use **cross-validation** for tuning **parameters**
3) Use grid/random/... **search** for tuning **hyperparameters**.



3. Train your model and save results

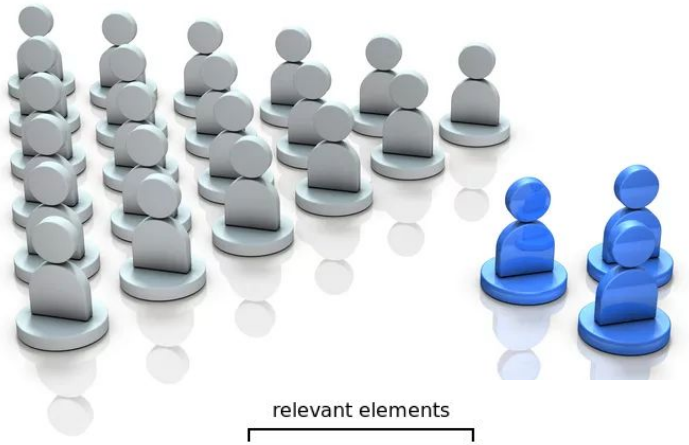
The results of your training (the most valuable thing!):

- Model type (ANN, SVM, CNN, R-CNN, ...)
- Hyperparameters
- Parameters (weights)

SAVE THEM IF YOU LIKE THEM

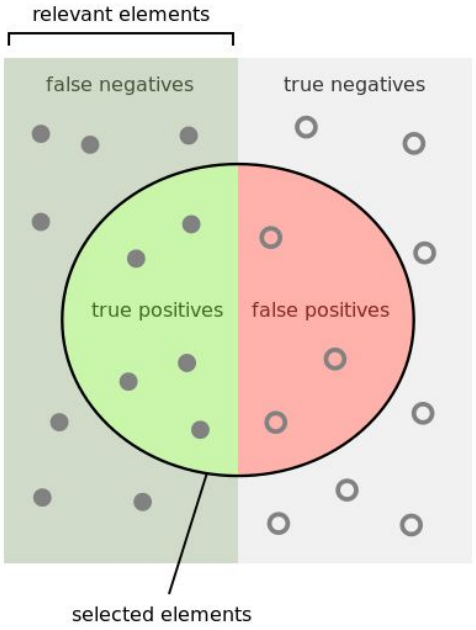
False fiends of ML

Biased data



WRONG:

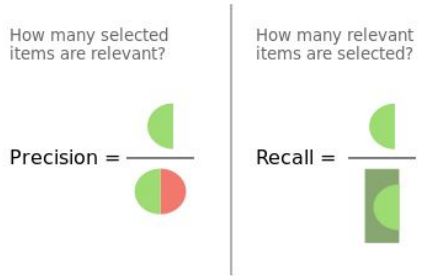
Quality = Accuracy = (TP+TN)/(P+N)



BETTER:

Precision, Recall

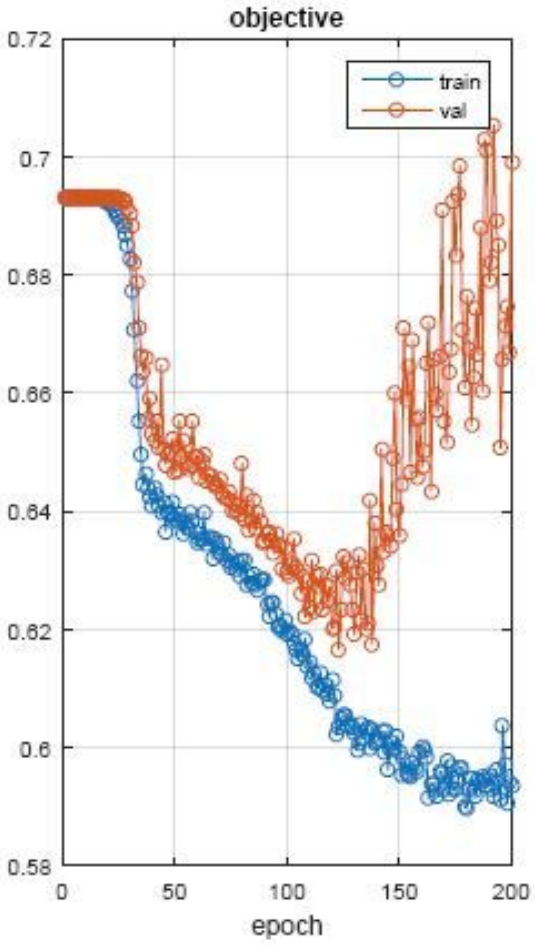
Quality = $F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

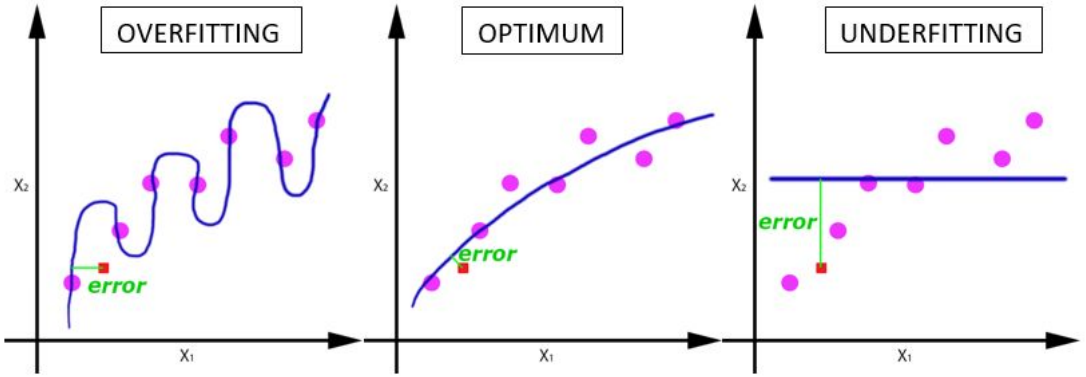


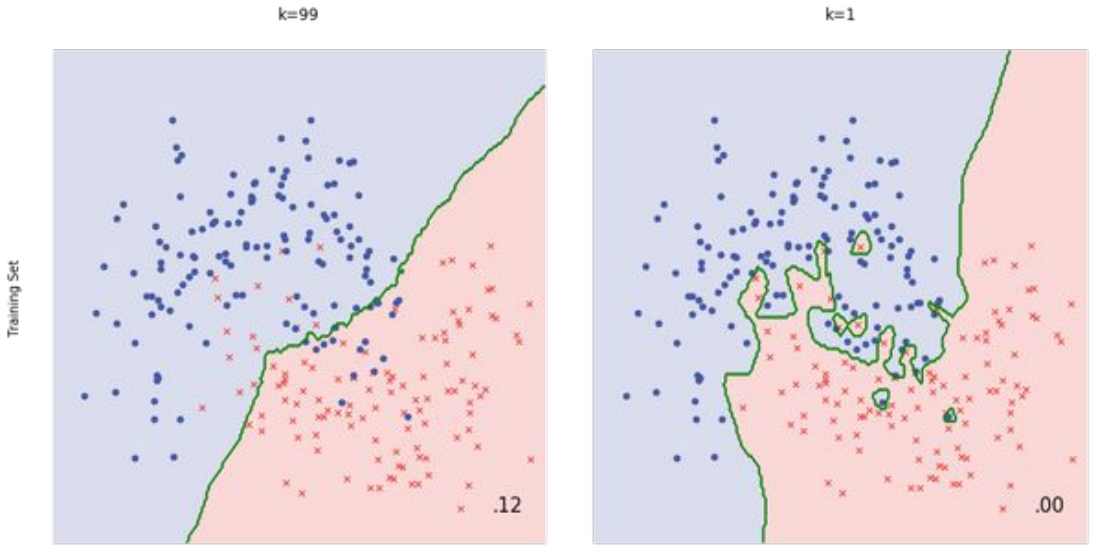
EVEN BETTER:

Normalize your data distribution
(find examples, augment, or at least clone)

Overfitting







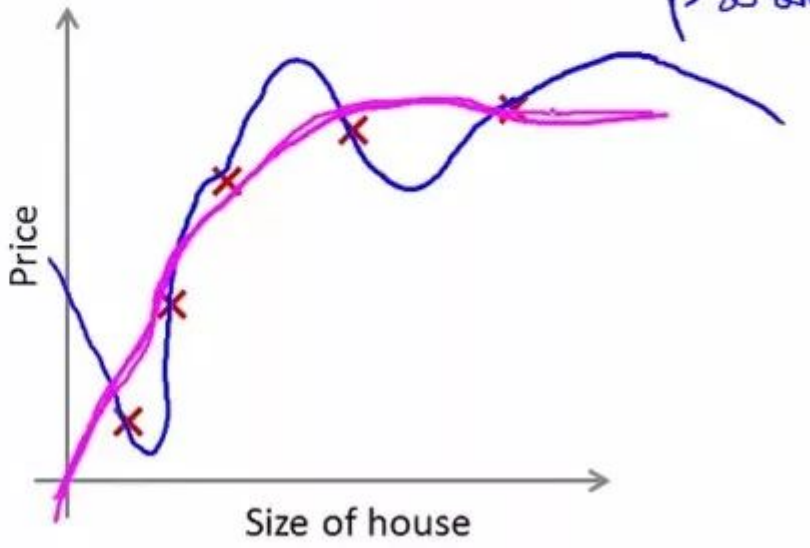
Overfitting?

Regularization.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

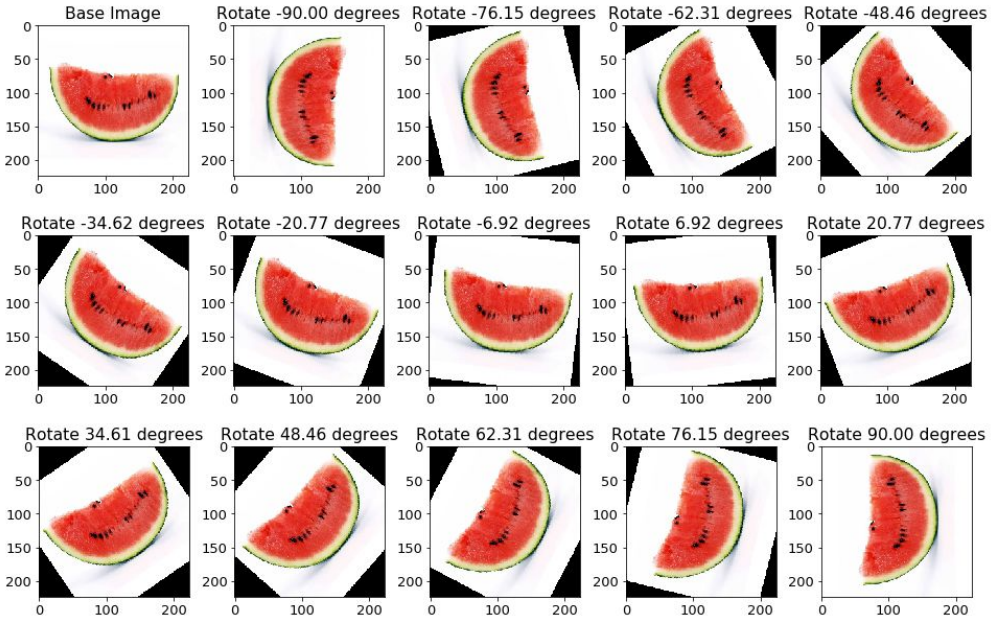
regularization parameter

$$\min_\theta J(\theta)$$

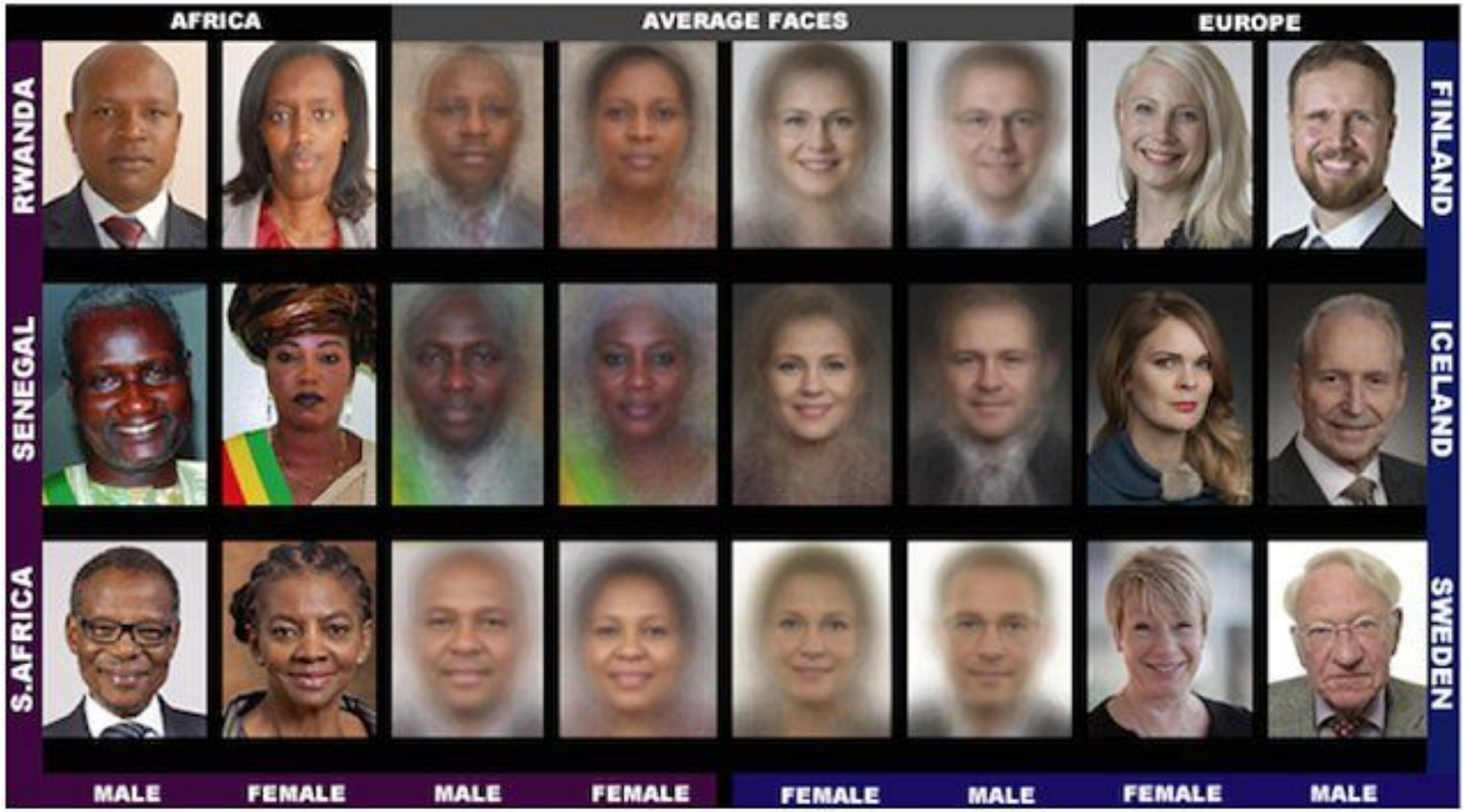


Small dataset and complex model

1) Don’t use complex model for small dataset
   a) Rule of thumb: number of **parameters** should be comparable with **dataset size**
2) Data **augmentation**
3) Data generation



Biased conditions (datasets)



Features

Variance and Features

Variance is the expectation of the squared deviation of a random variable from its mean.

Correlation - any statistical relationship, **whether causal or not**, between two random variables or bivariate data.

- A **correlation coefficient** is a numerical measure of some type of correlation.
- **Covariance** is a measure of the joint variability of two random variables.

$$\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\operatorname{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

Correlation, variance and PCA

Please refer to this tutorial

https://github.com/hsu-ai-course/hsu.ai/blob/master/code/11.%20Features.ipynb

Implement “Nutrition” lab

- Which features are redundant?
- Which will you keep to represent food better?

Non-Linear features and feature engineering

Problem of XOR

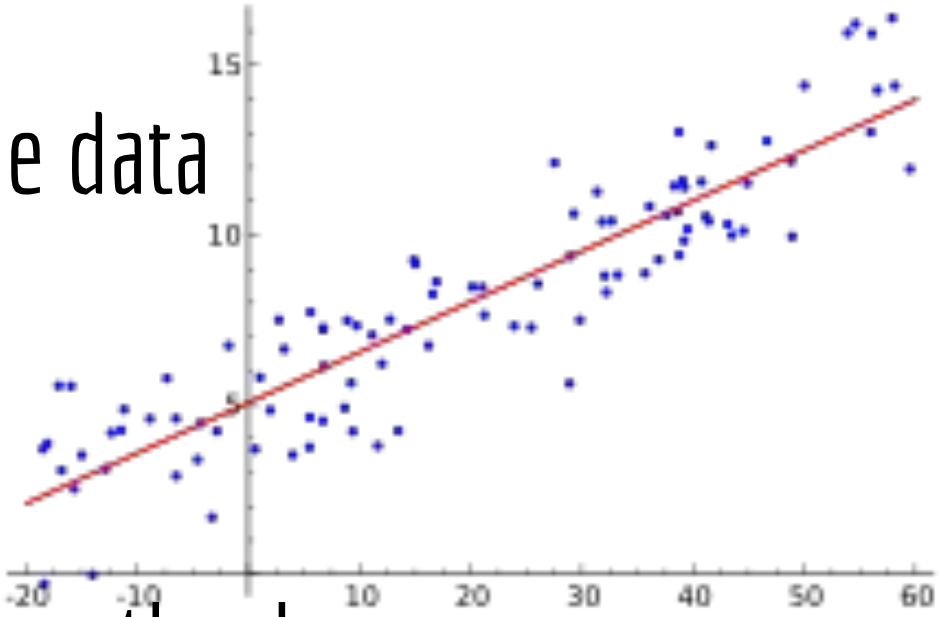
Sometimes there’s not enough information in data

Linear Regression

Linear model to explain the data

Ordinary Least Squares

$$\bar{x} = \left[(A^T A)^{-1} A^T\right] b$$



Lasso - “keep it simple” method

Ridge - “keep numbers small” (regularization)

SGD - “fit in memory” method

Lab #2. Implement GDP lab

Updated GDP lab version

1) Train-Validate split
2) Fit linear models, explore their properties
3) Measure RMSE for these models

Homework.

Predict **calories** of the food **from other factors**.

1) Select features
2) Try Linear model
3) Try mode complex models: MLP (2 layers), SVM. Estimate their quality. Which one is the best?