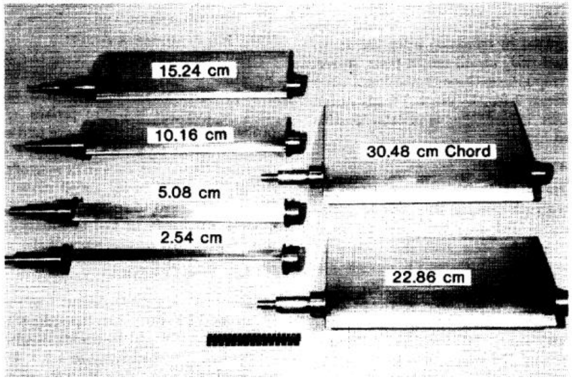
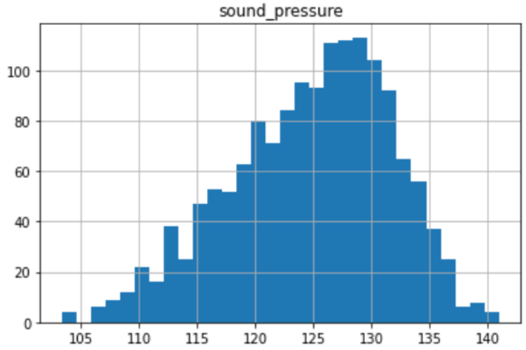
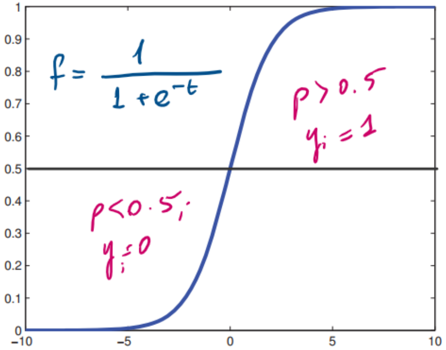
Data Driven Engineering I: Machine Learning for Dynamical Systems

Analysis of Static Datasets I: Classification

Institute of Thermal Turbomachinery
Prof. Dr.-Ing. Hans-Jörg Bauer







Today's Agenda

Basic Steps to Follow:

0.) Understand the business/task.

1.) Understand the data.

2.) Explore & prepare the data.

3.) Shortlist candidate models.

4.) Training the model

5.) Evaluate the model predictions.

6.) "Serve" the model!

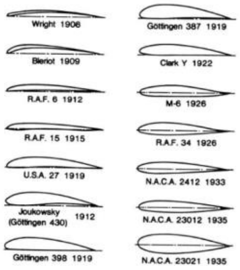
"Classification"

#0 Understanding the task

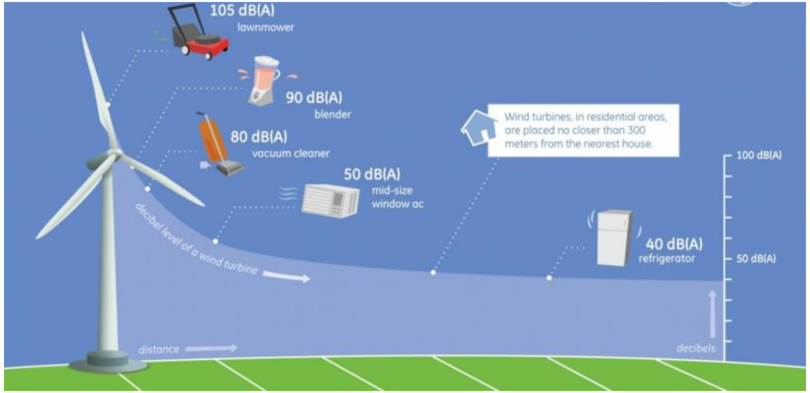
- **Problem**: NACA 0012 Airfoil Noise Prediction based on Wind Tunnel Testing

- **Noise** generated by an aircraft is an **economic** (efficiency) and **enviromental** issue.

- One component of the noise the **self-noise of the airfoil**: interaction of the airfoil with its own boundary layer



1917, the NACA Technical Report No. 18 titled "Aerofoils and Aerofoil Structural Combinations," was released.

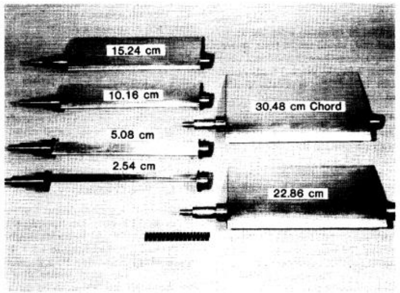


#0 Understanding the task

- ❑ Engineering: semi-emprical models (Brooks)

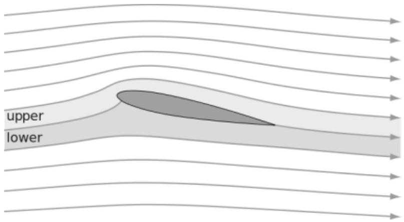
- ❑ Five self-noise mechanisms due to specific boundary-layer phenomena have been identified

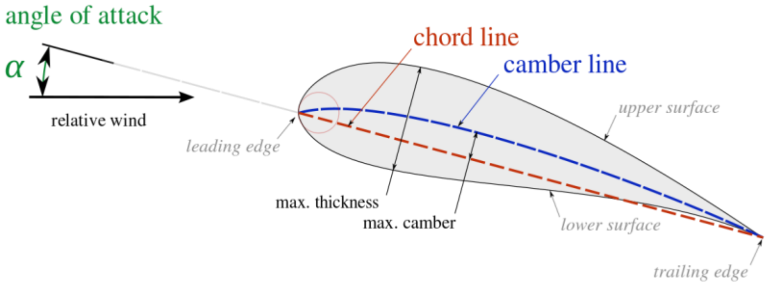
- ❑ The database is from seven NACA0012 airfoil blade sections of different sizes tested at wind tunnel speeds up to Mach 0.21 and at angles of attack from 0°to 25.2°.
  - ✓ Freq. of noise
  - ✓ Angle of attack
  - ✓ Free stream velocity
  - ✓ Geometry of the airfoil



L-82-4573

Figure 2. Two-dimensional NACA 0012 airfoil blade models.





#1 Understanding the data

- ❑ Check the data source: understand what the data refers to

- ❑ Objective: understand the characteristics of the data

- ❑ Look at the feature columns:
  - ❑ Any missing values?
  - ❑ Any features with NaN values?
  - ❑ Uniqueness of the dataset? (“cardinality”)

=> Colab

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1503 entries, 0 to 1502
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   frequency               1503 non-null   int64
 1   angle_attack            1503 non-null   float64
 2   chord_length            1503 non-null   float64
 3   Free-stream_velocity    1503 non-null   float64
 4   displacement_thickness  1503 non-null   float64
 5   sound_pressure          1503 non-null   float64
dtypes: float64(5), int64(1)
memory usage: 70.6 KB
```

```
data.head(5)
```

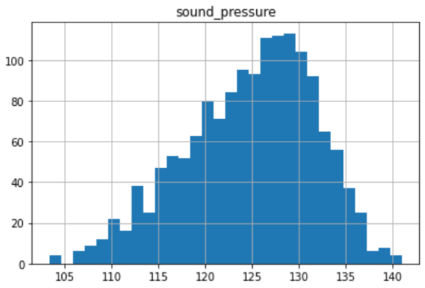
	frequency	angle_attack	chord_length	Free-stream_velocity	displacement_thickness
0	800	0.0	0.3048	71.3	0.002663
1	1000	0.0	0.3048	71.3	0.002663
2	1250	0.0	0.3048	71.3	0.002663
3	1600	0.0	0.3048	71.3	0.002663
4	2000	0.0	0.3048	71.3	0.002663

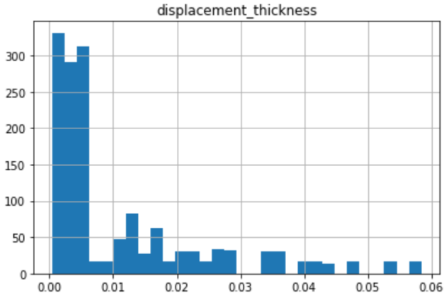
#2 Exploring the data

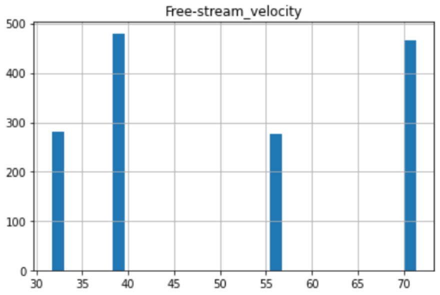
- ❑ **Objective**: generate a data quality report

- ❑ Using standard statistical measures of central tendency and variation
  - ❑ Tabular data and visual plots
  - ❑ mean, mode, and median
  - ❑ standard deviation and percentiles
  - ❑ Bars, histograms, box and violin plots

- ✓ Missing values,
- ✓ Irregular cardinality problems,
  - ▪ 1 or comparably small
- ✓ Outliers
  - ▪ invalid outliers and valid outliers







#2 Exporing the data: Correlation Matrix

❑ Shows the correlation between each pair of features

$$\underbrace{Cov(a,b)}_{\text{Features}} = \frac{1}{n-1}\sum_{i=1}^{n}\left[(a_i - \underbrace{\bar{a}}_{\text{mean}}) \times (b_i - \underbrace{\bar{b}}_{\text{mean}})\right]$$

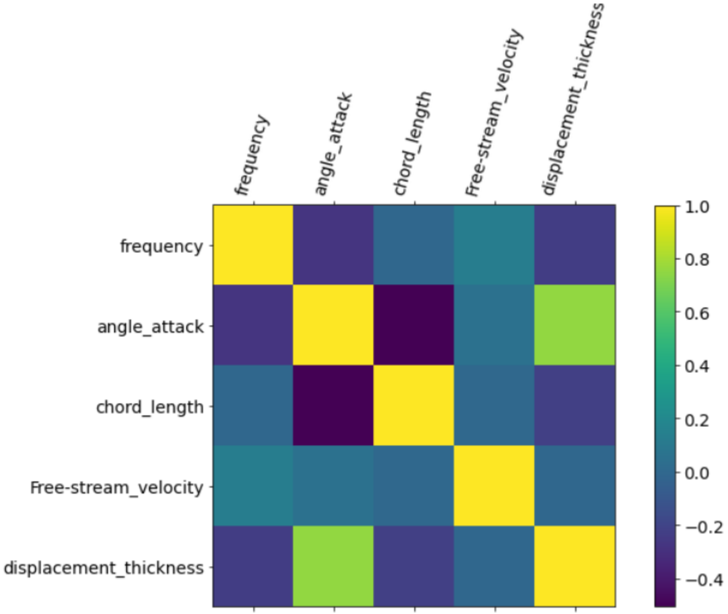
instance

❑ Normalized form of “covariance”

$$Corr(a,b) = \frac{Cov(a,b)}{SD(a) \times SD(b)}$$

* Normalized
* Dimensionless

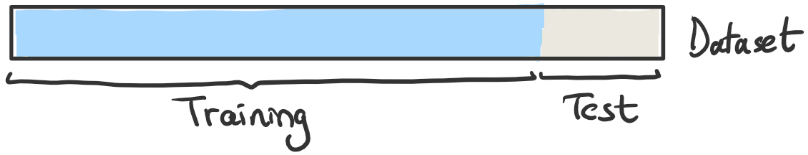
Easy to interpret



❑ Ranges between −1 and +1

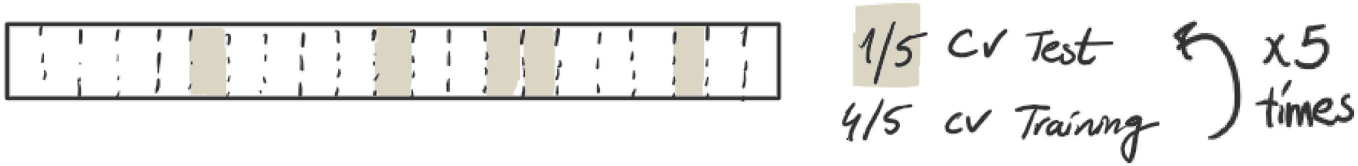
#2 Preparing the Data

- Classification >> supervised >> **training & test split**



- Reducing overfitting via **cross-validation**: take **random portions** of the data to build a model

- **k-fold** method: k = 5; (typically 10)



#3 Candidate Model Selection 1

Logistic Regression (probability)

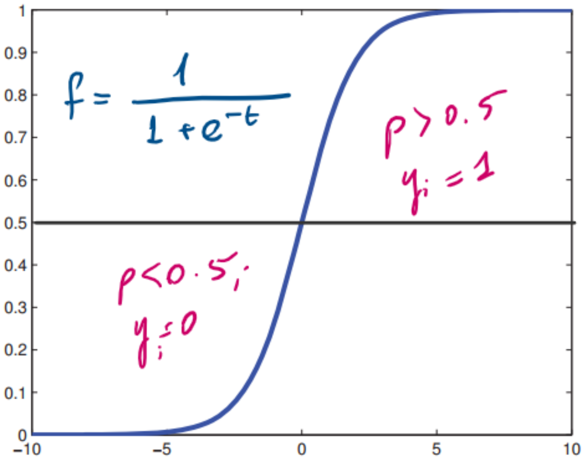
❑ A “derivative” of linear regression

>> probability func := Bernoulli distribution

>> pass the inputs through a function: **sigm**

>> maps the whole real line to [0, 1]

>> necessary for the output to be interpreted as a probability



$$p(y|\mathbf{x}, \mathbf{w}) = \mathrm{Ber}(y|\mathrm{sigm}(\mathbf{w}^T\mathbf{x}))$$

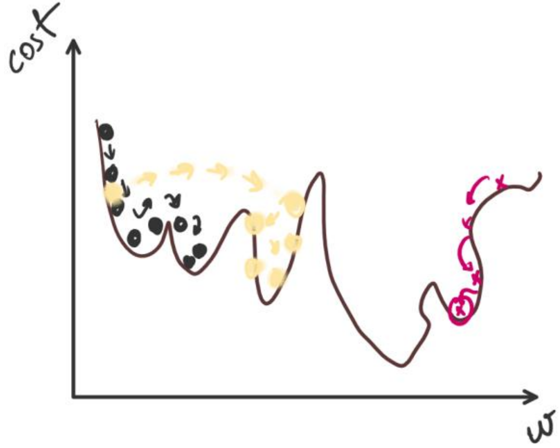
We apply regularization >>

$$\min_{w,c} \frac{1-\rho}{2} w^T w + \rho \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1)$$

#3 Candidate Model Selection 2

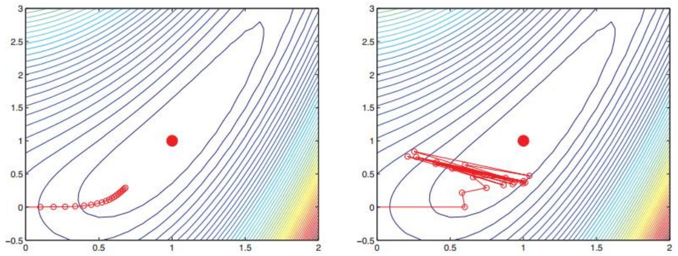
Gradient Decent (error based)

- ❑ Optimization technique under convex loss functions
- ❑ Measures the local gradient of the error function and goes in the direction of descending gradient (partial derivatives)



- ❑ “a way to train a model”
- ❑ Efficient and many tuning options

- ❑ An important parameter is the **learning rate**
- ❑ **Types** >> batch, stochastic, mini-batch



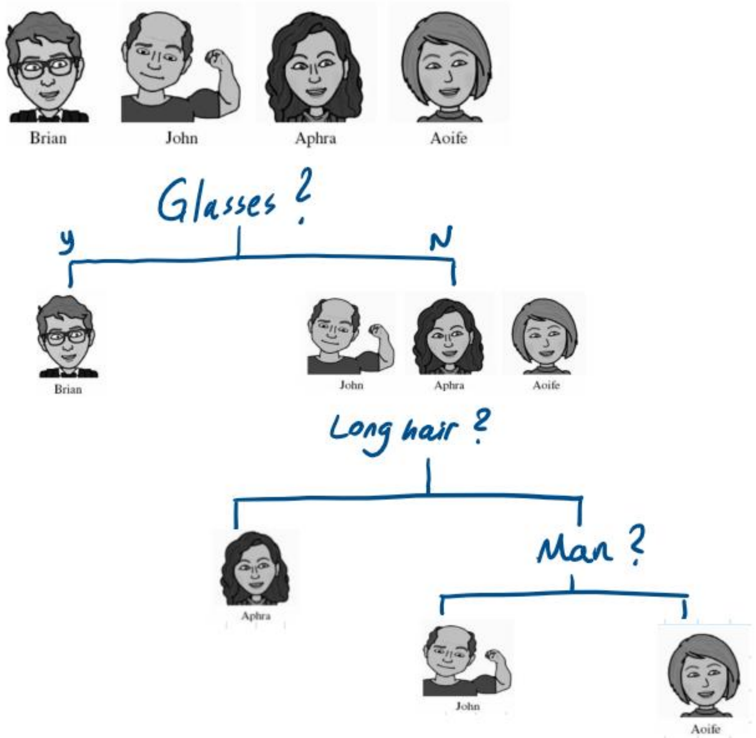
#3 Candidate Model Selection 3

Random Forest (information based)

- Predicts the value of a target variable by learning simple decision rules inferred from the data

- **Decision Trees** are the fundamental components of **Random Forests**
- **Train** >> Classification and Regression Tree (CART) algorithms (entropy)

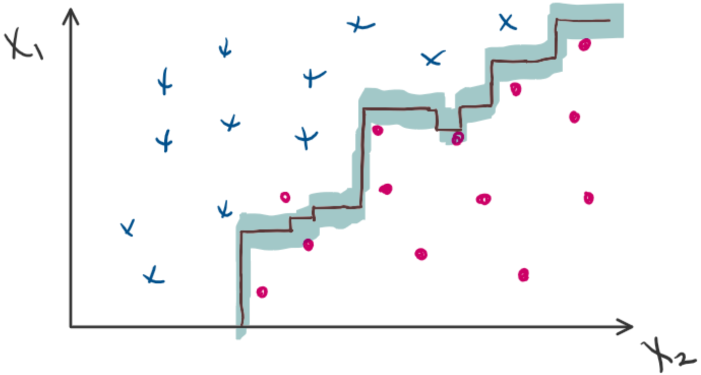
- it requires O(exp(m)) time, making the problem intractable even for small training sets (*reasonably good solutions*)



#3 Candidate Model Selection 3

Random Forest (information based)

- To avoid overfitting the training data, you need to restrict the Decision Tree’s freedom during training
  - maximum depth of the tree
  - pruning

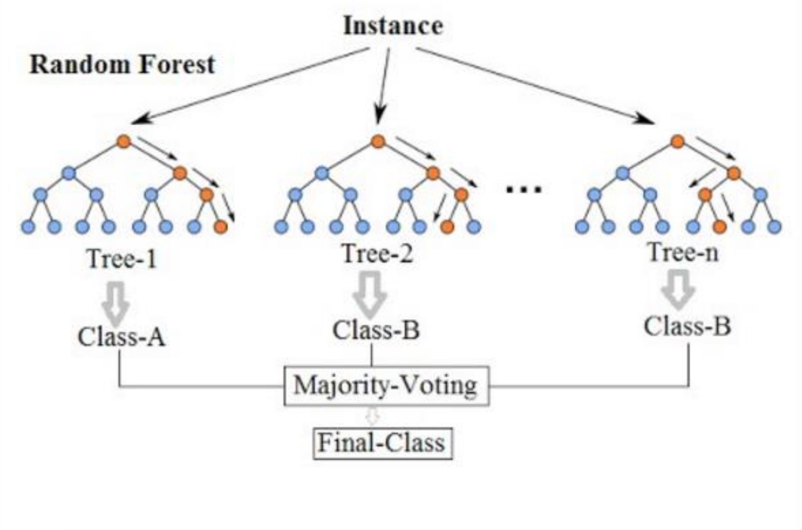


- **Unstable**: small variations in the data might result in a completely different trees
- **Orthogonal decision** boundaries creates problems

#3 Candidate Model Selection 3

Random Forest (information based)

- ❑ **Tree** >> **Forest** : “wisdom of the crowd”
  - A group of predictors is called an ensemble: ensemble learning
  - Decision Trees :each on a different random subset of the training set
- ❑ Searches for the best feature among a random subset of features, not all training set
- ❑ Easy to measure the relative importance of each feature




```
frequency 0.5078899615845973
angle_attack 0.10084945043187156
chord_length 0.09809369981835218
Free-stream_velocity 0.05777680855346639
displacement_thickness 0.2353900796117127
```

#3 Candidate Model Selection 4

Boosting (ensemble learning): LightGBM

- ❑ combine several weak learners into a strong learner.
- ❑ train predictors sequentially, each trying to correct its predecessor

- ❑ Gradient boosting := “Gradient” + “Boosting”

- **Boosting:** instances hard to predict correctly are focused on during the iterative learning process >> the model learns from past mistakes

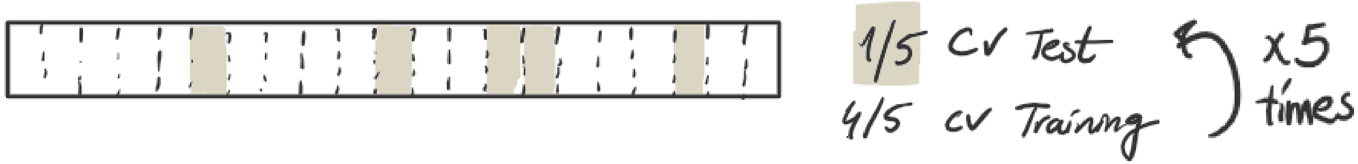
- **Gradient:** second partial derivatives of the loss function + advanced regularization

#4 Training the model

- Classification >> supervised >> **training & test split**

- Reducing overfitting via **cross-validation**: take **random portions** of the data to build a model

- **k-fold** method: k = 5; (typically 10)



#5 Evaluation of the predictions

Log loss (binary classification):

- Cross-entropy between the true labels and the model-based predictions

- Average loss function for classes A and B:

$$\underset{[A,B]}{\text{Log Loss}} = -\frac{1}{N}\sum_{i=1}^{N} \overbrace{y_i}^{\text{A Class}} \log(p(y_i)) + \overbrace{(1-y_i)}^{\text{B Class}} \log(1-p(y_i))$$

$y_i \rightarrow$ label; $p(y_i) \rightarrow$ prob. predicted

~~Eg.~~

$$\underset{(\text{lin.})}{\text{cost}} = \begin{cases} -\log(p) & ; y=1 \\ -\log(1-p) & ; y=0 \end{cases}$$

$$y_i=1 \Rightarrow p_i = 10^{-4} \Rightarrow \left.\begin{array}{l} -\log(p_i) = 4 \\ -\log(1-p_i) \cong 10^{-7} \end{array}\right\} \text{large error!}$$

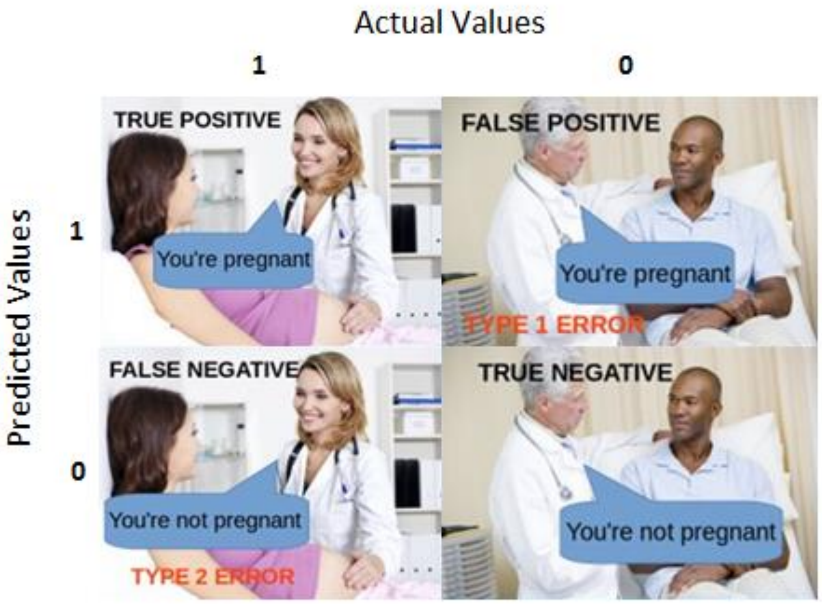
$$\searrow p_i = 0.95 \Rightarrow -\log(p_i) = 0.02 \} \text{ small error!}$$

- Cost function is convex >> global minimum exists!
- An optimization algorithm to compute it

#5 Evaluation of the predictions

Confusion Matrix

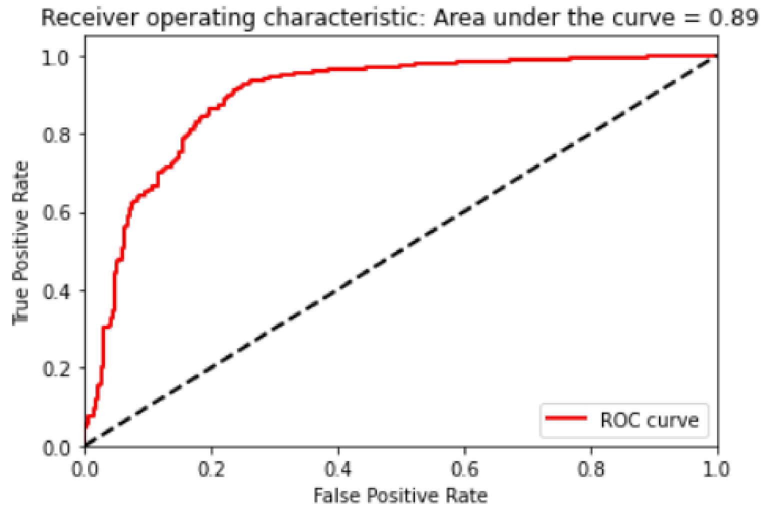
- Convenient way to fully describe the performance
- basis for different performance measures
- Good for balanced classes (# TP~ #TN)
- Imbalanced data sets: may overpredict the model outcomes



#5 Evaluation of the predictions

ROC Curve

- ❑ "Receiver operating characteristic curve"
- ❑ Confusion matrix based on a prediction score threshold of 0.5.
- ❑ For every possible value of the threshold, in the range [0, 1], there are corresponding TP and TN values.
- ❑ ROC curve is drawn by plotting a point for every feasible threshold value and joining them.
- ❑ The closer the curve is to the top left corner of the plot, the better the solution



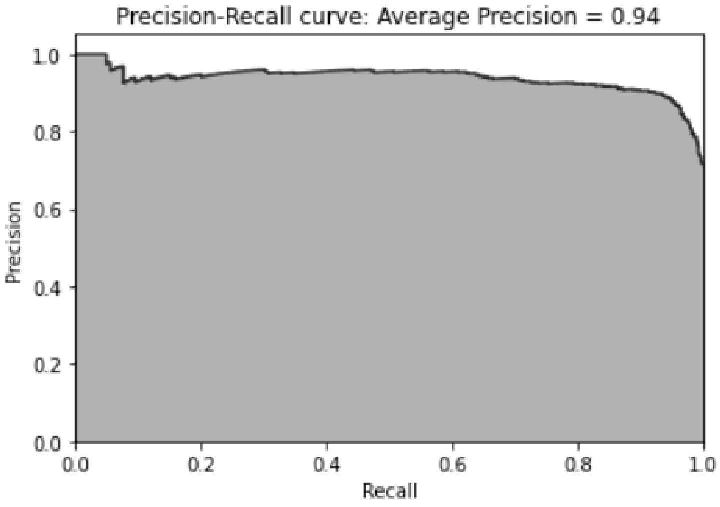
#5 Evaluation of the predictions

Karlsruhe Institute of Technology

Precision Recall Curve (for imbalanced data)

$$\text{Precision} := \frac{\text{True Positive}}{\text{TP} + \text{False Positive}} \Rightarrow \frac{\text{It is positive}}{\text{"It is positive"}}$$

$$\text{Recall} := \frac{\text{True Positive}}{\text{TP} + \text{False Negative}} \Rightarrow \frac{\#\text{Correct Predict.}}{\#\text{True Cases}}$$



- **Precision** captures how often, when a model makes a positive prediction, this prediction turns out to be correct.
- **Recall** tells us how confident we can be that all the instances with the positive target level have been found by the model.


Ateliers & Saveurs in Montreal

Additional Notes

Preparing the Data: Bootstrapping

- Bootstrapping approaches are preferred over CV in the case of very small datasets (< 300 instances).
- using slightly different training and test sets each time to evaluate the expected performance
- k is set to values greater than or equal to 200