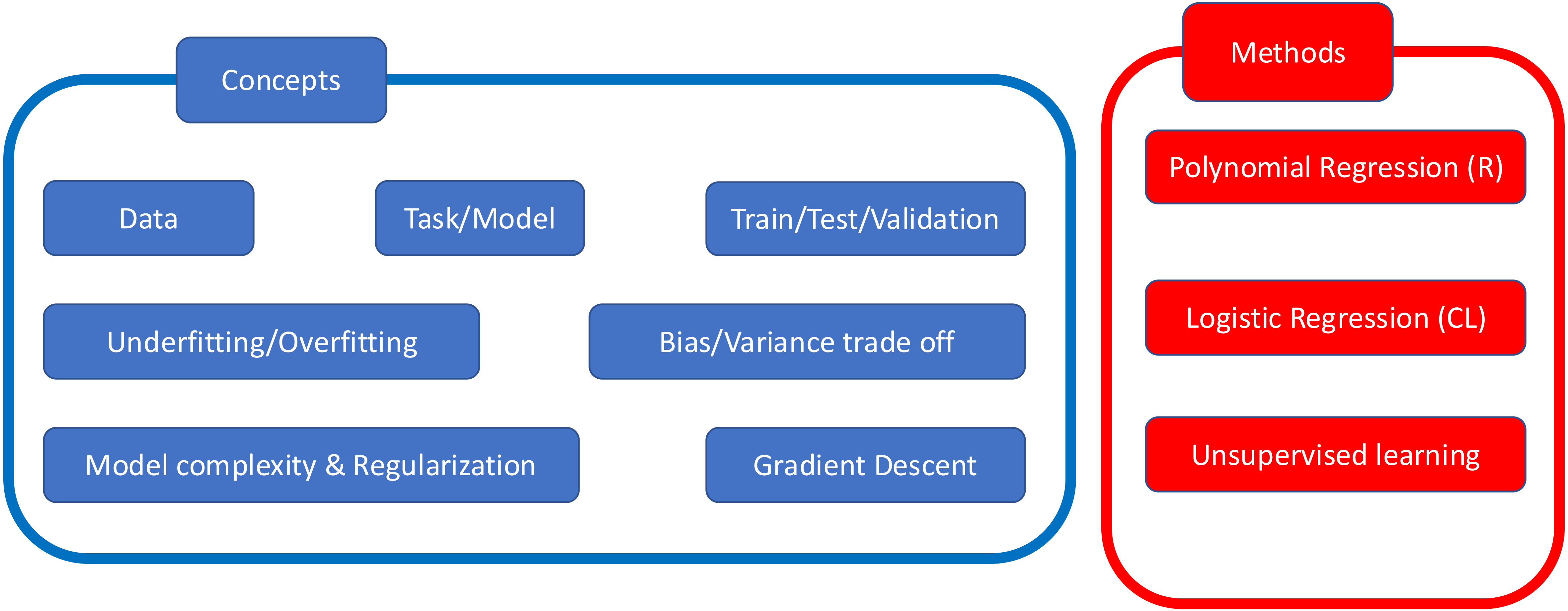
Course content

Main idea: get to know most common techniques (and theory) used in ML together with (some) coding skills.

Intro

- Basic notions (dataset, task…) + Regression and classification models (A+F)
- Model assessment (A)
- Unsupervised learning: dimensionality reduction, etc (A)
- Supervised learning: K-Nearest Neighbor, Trees (A)
- Kernels (F)
- Artificial Neural Networks (F)
- (Probabilistic formulation, recommender systems, RL) (L)

Overview of the introduction



Concepts

Data

Task/Model

Train/Test/Validation

Underfitting/Overfitting

Bias/Variance trade off

Model complexity & Regularization

Gradient Descent

Methods

Polynomial Regression (R)

Logistic Regression (CL)

Unsupervised learning

Regression

Linear Regression:

- $\hat{y} = X^T w$
- $\mathcal{L} = \|\hat{y} - y\|^2 = \|X^T w - y\|^2$
- $w = (XX^T)^{-1} Xy$
- Convex solution ($\mathbb{H} = 2XX^T$) and possible problems.
- GD solution and considerations about the stepsize
- Offset
- From linear to polynomial

Logistic Regression:

- $p(y|x) = \frac{1}{1+e^{w^T x}}$
- $\mathcal{L} = -\frac{1}{N} \sum_i y_i (w^T x_i) - \log\left(1 + e^{w^T x_i}\right)$ derived both from ML and entropy approaches.
- No closed form, numerical minimization needed (SGD).
- Extension to multinomial.

Regression

- **Poisson Regression**
  - $p(y|x) = \frac{e^{-\lambda}\lambda^{y}}{y!}$
  - Cases of application
- **Generalized Linear Models**
  - $p(y|x) = f(w^{T}x)$
  - The one parameter exponential family
  - $p(y;w) \propto e^{[a(y)b(w)+c(w)+d(y)]}$
  - Linear, Poisson and Logistic Regressions as special cases of GLM’s
  - The link function

- Derivation of the **bias-variance tradeoff**
- **Regularization**:
  - Ridge (L_2), Lasso (L_1) and Elastic Nets (mixed $L_1 \& L_2$)
  - Closed solution for ridge regularization
  - Least Angle Regression algorithm for solving Lasso
  - Gradient Descent for Lasso and elastic nets (this was given as exercise).

Two hints for the exercise

$$\begin{aligned}
\nabla_w[\|y - X^T w\|_2^2 + \lambda\|w\|_2^2] &= \nabla_w[(y - X^T w)^T(y - X^T w) + \lambda w^T w] \\
&= 2\nabla_w[(y - X^T w)](y - X^T w)] + 2\lambda w \\
&= 2X(y - X^T w) + 2\lambda w
\end{aligned}$$

$$|f(x)|' = \frac{f(x)}{|f(x)|} f'(x)$$

SUPERVISED LEARNING MODEL SELECTION & ASSESSMENT

Measuring the quality of our model

- Can we use the loss for assessing the quality of our model?
- Let’s see what happens in a simple linear regression

$$\mathcal{L} = \sum_i (\hat{y}_i - y_i)^2$$

- Can you guess a problem in using that expression?
- (Hint: What happens if I multiply by a constant the values of y_i and repeat the learning?)
- We need a reference!

R^2

- Let's define $\bar{y} = \frac{1}{N}\sum_i^N y_i$
- So the variance of our data would be proportional to:

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

- We will use the SS_{tot} as reference for our quality parameter.

$$R^2 = 1 - \frac{\mathcal{L}}{SS_{tot}} = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2}$$

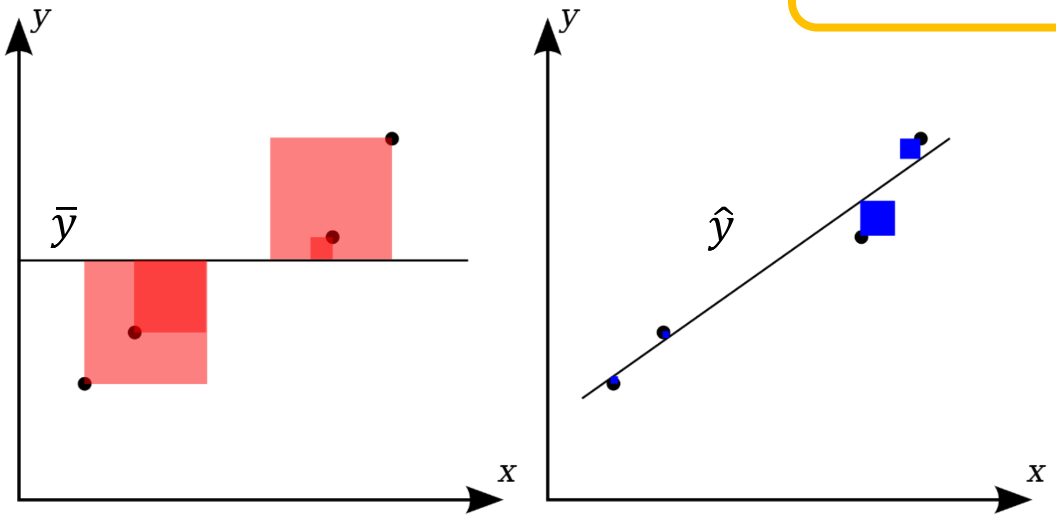
- This is equivalent to measure the Pearson correlation coefficient between the predicted values ($\hat{y}_i$) and the real data (y_i) (*Prove it as exercise*).
- $R^2$is related with the fraction of unexplained variance $FUV = 1 - R^2$

R^2

$$R^2 = 1 - \frac{\mathcal{L}}{SS_{tot}} = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- $R^2 = 1$ Perfect fit
- $0 \leq R^2 \leq 1$ A measure of the goodness of fit
- $R^2 < 0$ The model performs worse than a baseline model

Baseline model: *In linear regression, a model that always predicts the average. In general, a model with a unique constant value.*



Adjusted R^2

- Let’s define $\bar{y} = \frac{1}{N}\sum_i^N y_i$
- So the variance of our data would be proportional to:

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

- We will use the SS_{tot} as reference for our quality parameter.

$$R^2 = 1 - \frac{\mathcal{L}}{SS_{tot}} = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- This is equivalent to measure the Pearson correlation coefficient between the predicted values ($\hat{y}_i$) and the real data (y_i) (*Prove it as exercise*).
- $R^2$is related with the fraction of unexplained variance $FUV = 1 - R^2$

Adjusted R^2

- Variance in the data y :

$$Var_{tot} = \frac{1}{N}\sum_i (y_i - \bar{y})^2 = \frac{1}{N} SS_{tot}$$

- Variance explained by the model:

$$Var_{exp} = \frac{1}{N}\sum_i (y_i - \hat{y}_i)^2$$

- So $R^2$is:

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2} = R^2 = 1 - \frac{\frac{1}{N}\sum_i (\hat{y}_i - y_i)^2}{\frac{1}{N}\sum_i (y_i - \bar{y})^2} = 1 - \frac{Var_{exp}}{Var_{tot}}$$

- Let’s substitute the variances in this equation by their unbiased estimators*.

* Unbiased estimator of a number: The expected value of the estimator is the number itself

Adjusted R^2

- Variance in the data y :

$$Var^*_{tot} = \frac{1}{N-1}\sum_i (y_i - \bar{y})^2$$

- Variance explained by the model:

$$Var^*_{exp} = \frac{1}{N-p-1}\sum_i (y_i - \hat{y}_i)^2$$

- So $R^2_{adj}$is:

$$R^2_{adj} = 1 - \frac{Var^*_{exp}}{Var^*_{tot}} = 1 - \frac{\frac{1}{N-p-1}\sum_i (\hat{y}_i - y_i)^2}{\frac{1}{N-1}\sum_i (y_i - \bar{y})^2}$$

Generalizing R^2 to other types of regression

- Which of these definitions can be applied to, for instance, the logistic regression?
- Comparison with a baseline model:

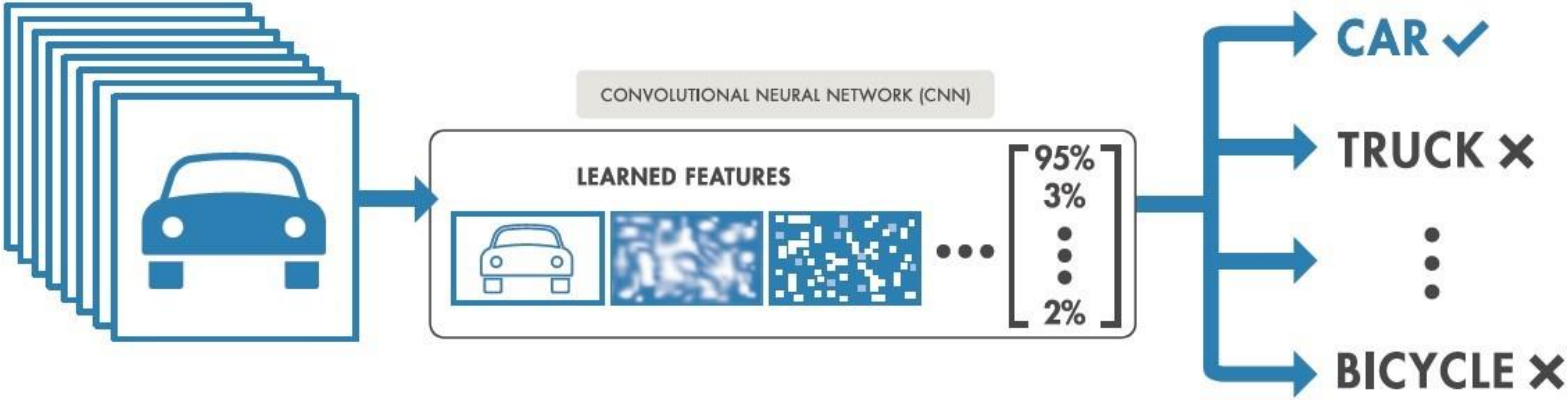
$$R_{gen}^2 = 1 - \left(\frac{\mathcal{L}(0)}{\mathcal{L}(\widehat{w})}\right)^{\frac{2}{N}}$$

- $\mathcal{L}(0)$is the likelihood of the baseline model.
- $\mathcal{L}(\widehat{w})$ is the likelihood of the optimized model.

Generalizing R^2 to other types of regression

- It is consistent with the classical R^2 when both can be computed;
- Its value is maximised by the maximum likelihood estimation of a model;
- It is asymptotically independent of the sample size;
- The interpretation is the proportion of the variation explained by the model;
- The values are between 0 and 1, with 0 denoting that model does not explain any variation and 1 denoting that it perfectly explains the observed variation;
- It does not have any unit.
- The maximum value for logistic regression is $R_{max}^2 = 1 - \left(\mathcal{L}(0)\right)^{\frac{2}{N}}$ so it is suggested to use $R^2 = R_{gen}^2 / R_{max}^2$

What's the purpose of supervised learning?



What's the purpose of learning?

- Getting better at future predictions, what is called generalization

Training Data

We want to choose the model that generalizes better!

Testing Data

For applications,
this is UNKNOWN

Generalization Error vs Training Error

How well will the model perform in a future prediction task?

Let’s define some concepts:

1. True data generating process: π_{XY}
2. Generalization error: $\varepsilon(y, \hat{y}) = \mathbb{E}_{\pi_{XY}}\left[\mathcal{L}\left(y, \hat{y}(x)\right)\right]$
3. Dataset $\mathcal{D} = \left((x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\right) \sim \pi_{XY}$
4. Estimated gener. error based on $\mathcal{D}$: $\varepsilon_{\mathcal{D}}(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(y_i, \hat{y}(x_i)\right)$
5. Training in $\mathcal{D}'$ means $\hat{y} = arg\min_{w} \sum_{(x,y)\in\mathcal{D}'} \mathcal{L}\left(y, f(w; x)\right)$
6. If $\mathcal{D}' = \mathcal{D}$ then $\varepsilon_{\mathcal{D}}(\widehat{y, y})$ is known as the training error $\varepsilon_{train}(y, \hat{y})$.
7. It is clear than $\varepsilon_{train}(y, \hat{y}) \leq \varepsilon_{\mathcal{D}}(y, \hat{y})$for a generic $\mathcal{D}$, but how can be this quantified?

Optimism of the training error

Define a data set y generated by the model:

$$y = X^T w + \epsilon$$

and another data set y'generated with the same data but with different noise:

$$y' = X^T w + \epsilon'$$

We learn our parameters $\hat{w}$ on the dataset y, so $\hat{y}_i = \hat{w}^T x_i$. The training error (in-sample error) will be:

$$\varepsilon_{tr}(y, \hat{y}) = \frac{1}{n} \sum_i (\hat{y}_i - y_i)^2$$

And the out-of sample error :

$$\varepsilon_o(y', \hat{y}) = \frac{1}{n} \sum_i (\hat{y}_i - y_i')^2$$

Optimism of the training error

$$\mathbb{E}\left((\hat{y}_i - y_i)^2\right) = Var\left((\hat{y}_i - y_i)\right) + \left(\mathbb{E}(\hat{y}_i - y_i)\right)^2 = Var(\hat{y}_i) + Var(y_i) - 2Cov(\hat{y}_i, y_i) + \left(\mathbb{E}(\hat{y}_i) - \mathbb{E}(y_i)\right)^2$$

$$\mathbb{E}\left((\hat{y}_i - y_i')^2\right) = Var\left((\hat{y}_i - y_i')\right) + \left(\mathbb{E}(\hat{y}_i - y_i')\right)^2 = Var(\hat{y}_i) + Var(y_i') - 2Cov(\hat{y}_i, y_i') + \left(\mathbb{E}(\hat{y}_i) - \mathbb{E}(y_i')\right)^2$$

From the definitions of the models ($y = X^T w + \epsilon$ & $y' = X^T w + \epsilon'$):

1. $\mathbb{E}(y_i') = \mathbb{E}(y_i)$
2. $Var(y_i') = Var(y_i)$
3. $Cov(\hat{y}_i, y_i') = 0$

So…

$$\mathbb{E}\left((\hat{y}_i - y_i')^2\right) = \mathbb{E}\left((\hat{y}_i - y_i)^2\right) + 2Cov(\hat{y}_i, y_i)$$

$$\mathbb{E}\left(\frac{1}{n}\sum_i (\hat{y}_i - y_i')^2\right) = \mathbb{E}\left(\frac{1}{n}\sum_i (\hat{y}_i - y_i)^2\right) + \frac{2}{n}\sum_i Cov(\hat{y}_i, y_i)$$

Optimism of the training error

Using that for a linear regression:

$$\hat{y} = X^T(XX^T)^{-1}Xy$$

if we define the hat matrix as

$$M = X^T(XX^T)^{-1}X$$

Then, it is easy to show that

$$Cov(\hat{y}_i, y_i) = \sigma^2 M_{ii}$$

And therefore:

$$\mathbb{E}\big(\varepsilon_o(y', \hat{y})\big) = \mathbb{E}\big(\varepsilon_{tr}(y, \hat{y})\big) + \frac{2}{n}\sum_i \sigma^2 M_{ii} = \mathbb{E}\big(\varepsilon_{tr}(y, \hat{y})\big) + \frac{2\sigma^2}{n} tr(M)$$

But

$$tr(M) = tr(X^T(XX^T)^{-1}X) = tr\big((XX^T)^{-1}XX^T\big) = tr(I) = p + 1$$

So

$$\underset{\text{OFS error}}{\boxed{\mathbb{E}\big(\varepsilon_o(y', \hat{y})\big)}} = \underset{\text{Training error}}{\boxed{\mathbb{E}\big(\varepsilon_{tr}(y, \hat{y})\big)}} + \underset{\text{Optimism}}{\boxed{\frac{2\sigma^2}{n}(p + 1)}}$$

Properties of the optimism

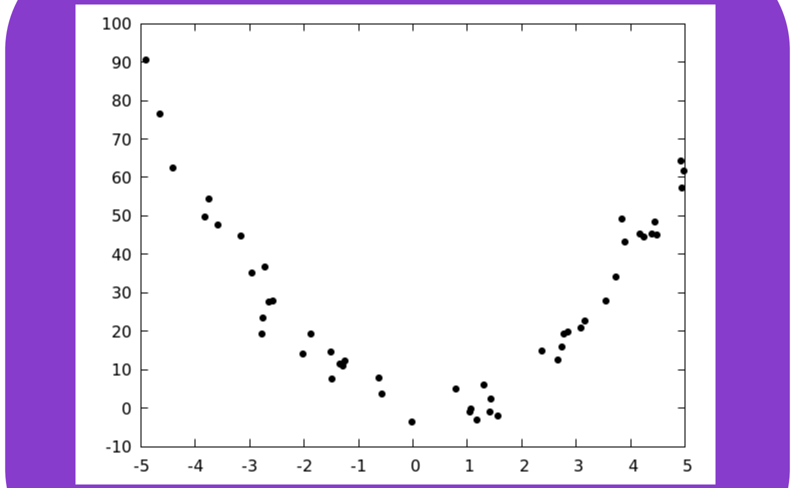
$$\frac{2\sigma^2}{n}(p+1)$$

- **Grows with $\boldsymbol{\sigma^2}$** : more noise gives the model more opportunities to seem to fit well by capitalizing on chance.
- **Shrinks with $\boldsymbol{n}$** : at any fixed level of noise, more data makes it harder to pretend the fit is better than it really is.
- **Grows with $\boldsymbol{p}$** : every extra parameter is another control which can be adjusted to fit to the noise.

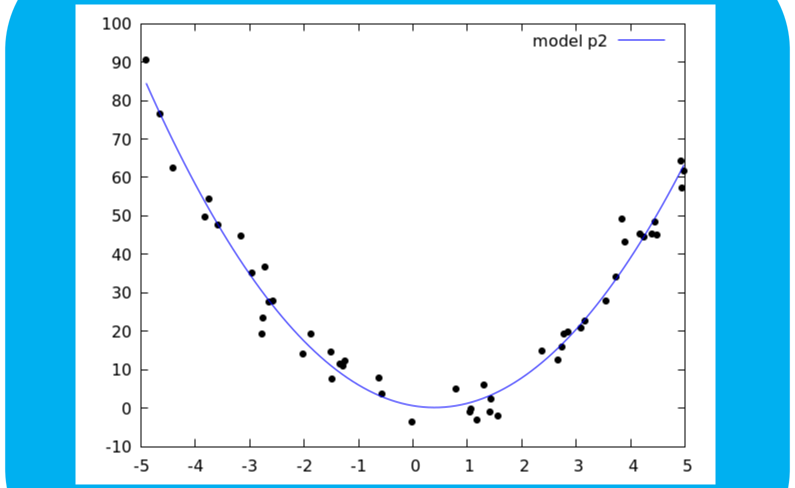
Mallow’s C_p statistic: Using the optimism for model selection

- We want to choose between two models, for instance: the model P2 is a polynomial of order 2, while the model P3 is a polynomial of order 3.
- Which of them is better?
- The one with lower **generalization** error
- Mallow’s test uses the estimate of the optimism that we had just shown: $\mathbb{E}\left(\varepsilon_o(y', \hat{y})\right) = \mathbb{E}\left(\varepsilon_{tr}(y, \hat{y})\right) + \frac{2\sigma^2}{n}(p+1)$
- This is known as ***in-sample prediction plus penalty*** model selection.

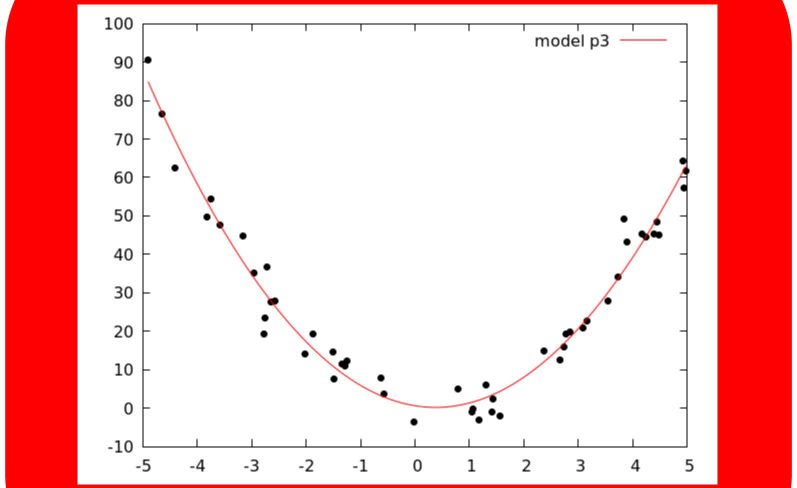
Mallow's C_p statistic: Using the optimism for model selection



Original data. n=50



p= 2; $\varepsilon_{tr}(y, \hat{y})$= 19.03



p= 3; $\varepsilon_{tr}(y, \hat{y})$= 19.01

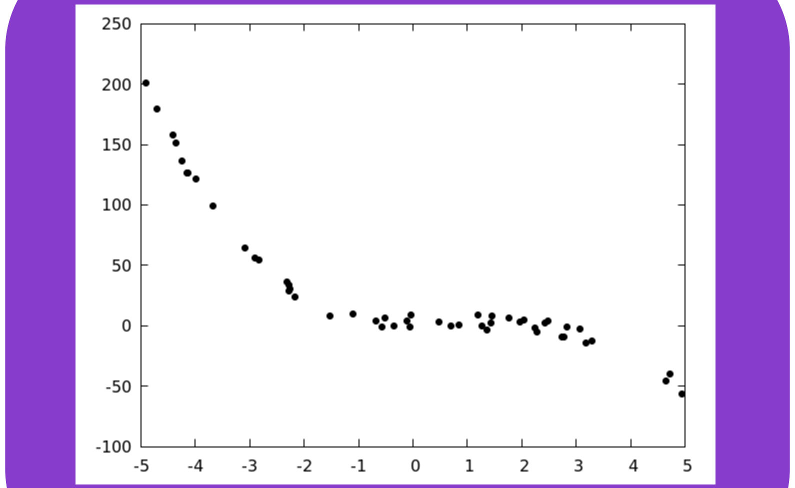
$$\mathbb{E}\big(\varepsilon_o(y', \hat{y})\big) = \mathbb{E}\big(\varepsilon_{tr}(y, \hat{y})\big) + \frac{2\sigma^2}{n}(p+1)$$

We estimate σ^2 from the largest model: In this case P3. Could you guess why?

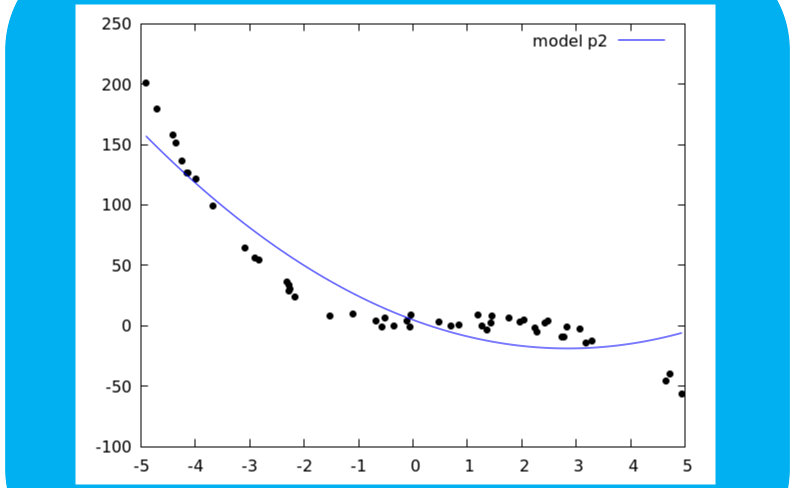
$$\big(\varepsilon_o(P2)\big) = 19.03 + \frac{2\sigma^2}{n}(p+1) = 19.03 + \frac{2(19.01)}{50}(2+1) = 21.31$$

$$\big(\varepsilon_o(P3)\big) = 19.01 + \frac{2\sigma^2}{n}(p+1) = 19.01 + \frac{2(19.01)}{50}(3+1) = 22.06$$

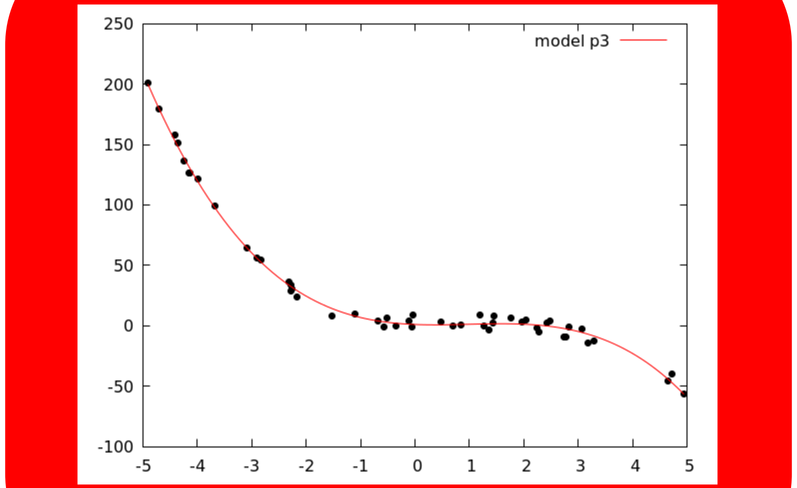
Mallow’s C_p statistic: Using the optimism for model selection



Original data. n=50



p= 2; $\varepsilon_{tr}(y,\hat{y})$= 408.19



p= 3; $\varepsilon_{tr}(y,\hat{y})$= 15.13

$$\mathbb{E}\big(\varepsilon_o(y',\hat{y})\big) = \mathbb{E}\big(\varepsilon_{tr}(y,\hat{y})\big) + \frac{2\sigma^2}{n}(p+1)$$

$$\big(\varepsilon_o(P2)\big) = 408.19 + \frac{2\sigma^2}{n}(p+1) = 408.19 + \frac{2(15.13)}{50}(2+1) = 410.01$$

$$\big(\varepsilon_o(P3)\big) = 15.13 + \frac{2\sigma^2}{n}(p+1) = 15.13 + \frac{2(15.13)}{50}(3+1) = 17.55$$

Akaike & Bayes Information Criteria: Another *in-sample plus penalty* criteria

- Can we extend Mallow’s criterion?
- We will consider two cases, both applied to the Likelihood:
  - Akaike Information Criterion (AIC):

$$AIC(M) = \mathcal{L}(M) - dim(M)$$

  - Bayes Information Criterion (BIC):

$$BIC(M) = \mathcal{L}(M) - \frac{\log N}{2} dim(M)$$

- What happens if we apply it to a Linear Model with Gaussian Noise (exercise)?

Considerations about *in-sample* criteria

- Mallow’s criterion is usually good, but limited to linear predictors.
- Akaike criterion does not predict the asymptotically correct model.
- Akaike criterion penalty is not enough when used for variable selection.
- Bayes criterion predicts the asymptotically correct model performs worst when used for predicting the performance of several models.

Out-of-sample Validation

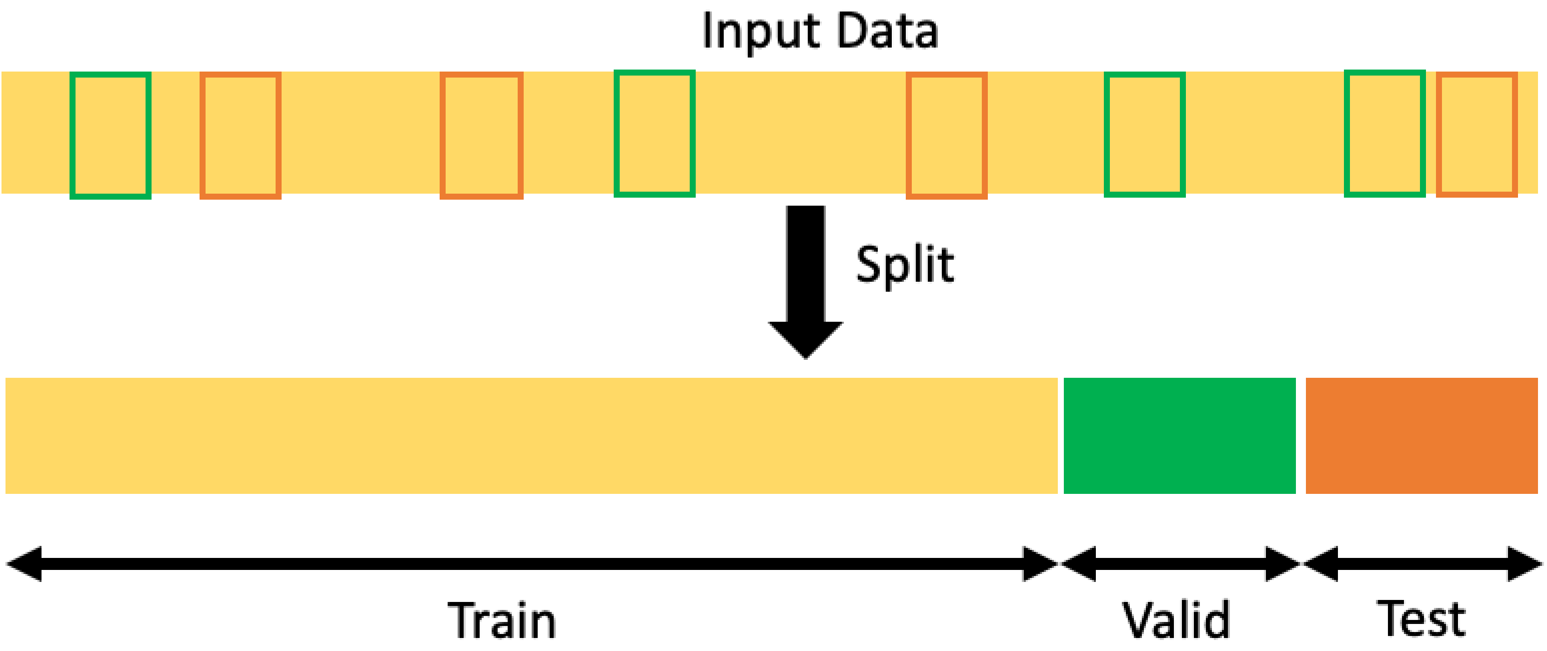
Restraining the learning set…

Last lecture

- Use of R^2 for quality validation (Also $R^2_{gen}$and R^2_{adj}).
- Optimism of the training error ($\frac{2\sigma^2}{n}(p+1)$)
- In-sample criteria for model selection:
    - Mallow's (Optimism, σ^2 evaluated on the "bigger" model)
    - Akaike Information Criterion (AIC)
    - Bayes Information Criterion (BIC)

Validation

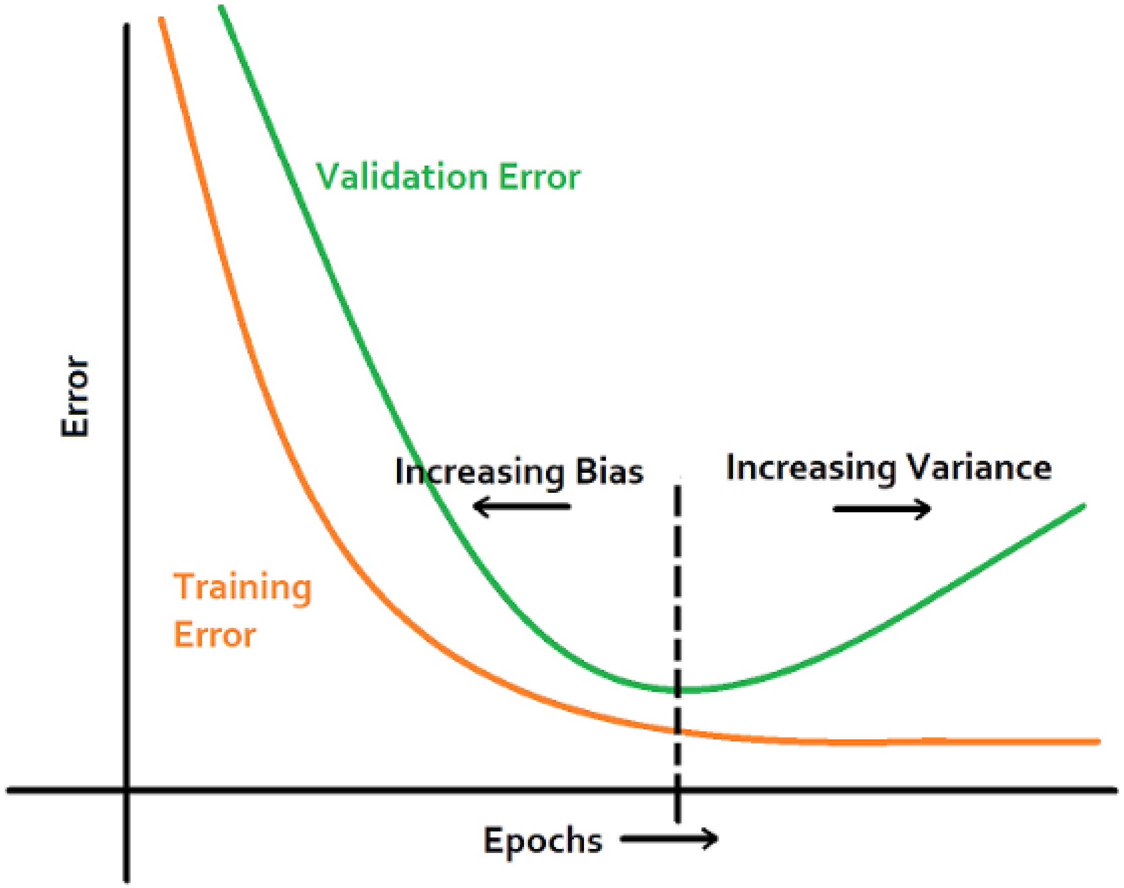
- Direct estimate of the generalization error by dedicating part of the data.



Testing for good models: validation e cross-validation

- Testing if the model has extracted the correct information from the data.
- A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model’s hyperparameters.
- The validation dataset is different from the test dataset that is also held back from the training of the model, but is instead used to give an unbiased estimate of the skill of the final tuned model when comparing or selecting between final models.

Early stopping: A regularization/validation technique.

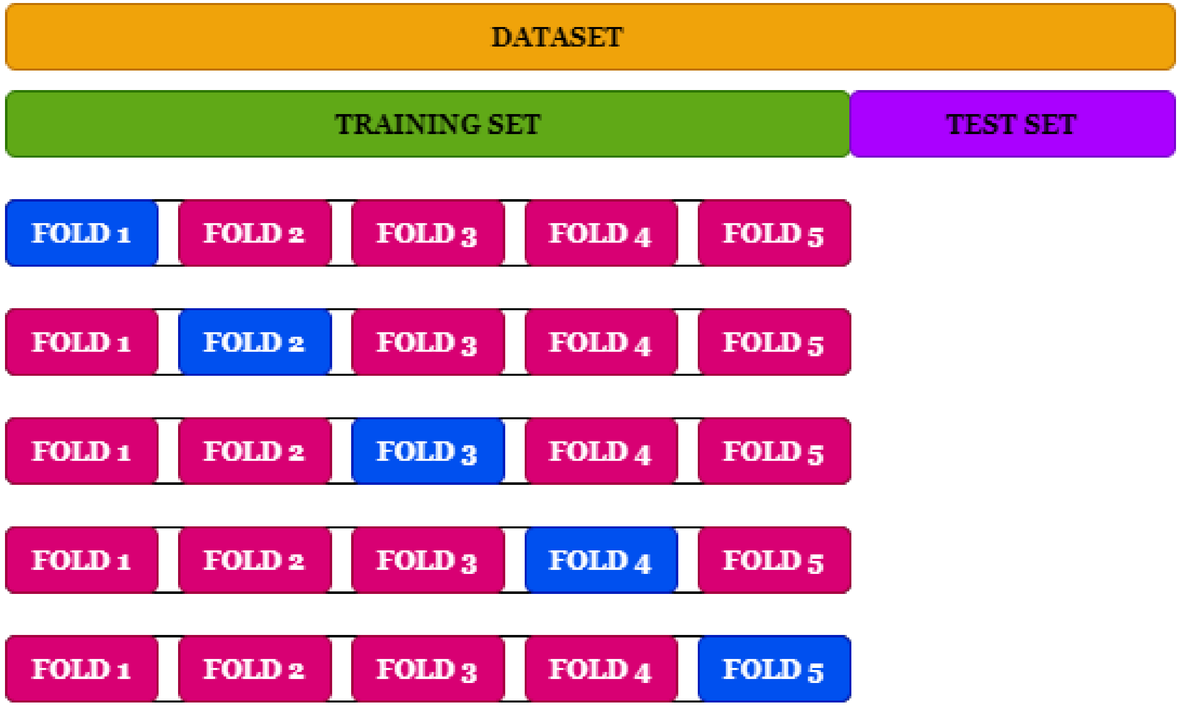


Validation

- The scores for evaluating the model are highly variable depending on the observations that make up in train set and validation set from a single train validation split.

- Reserving much of the data for a single validation set reduces the number of observations we can use to train the model.

K-fold cross-validation



DATASET

TRAINING SET | TEST SET

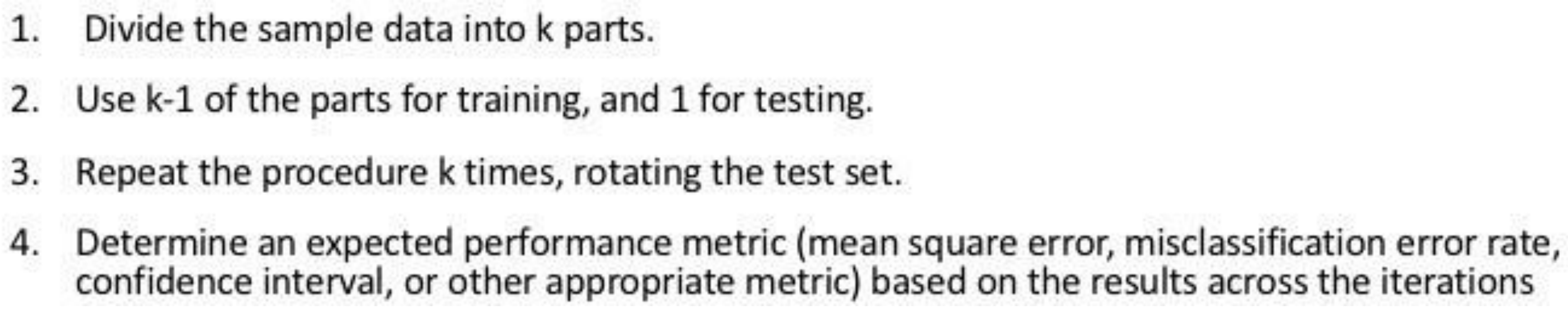
FOLD 1 | FOLD 2 | FOLD 3 | FOLD 4 | FOLD 5

FOLD 1 | FOLD 2 | FOLD 3 | FOLD 4 | FOLD 5

FOLD 1 | FOLD 2 | FOLD 3 | FOLD 4 | FOLD 5

FOLD 1 | FOLD 2 | FOLD 3 | FOLD 4 | FOLD 5

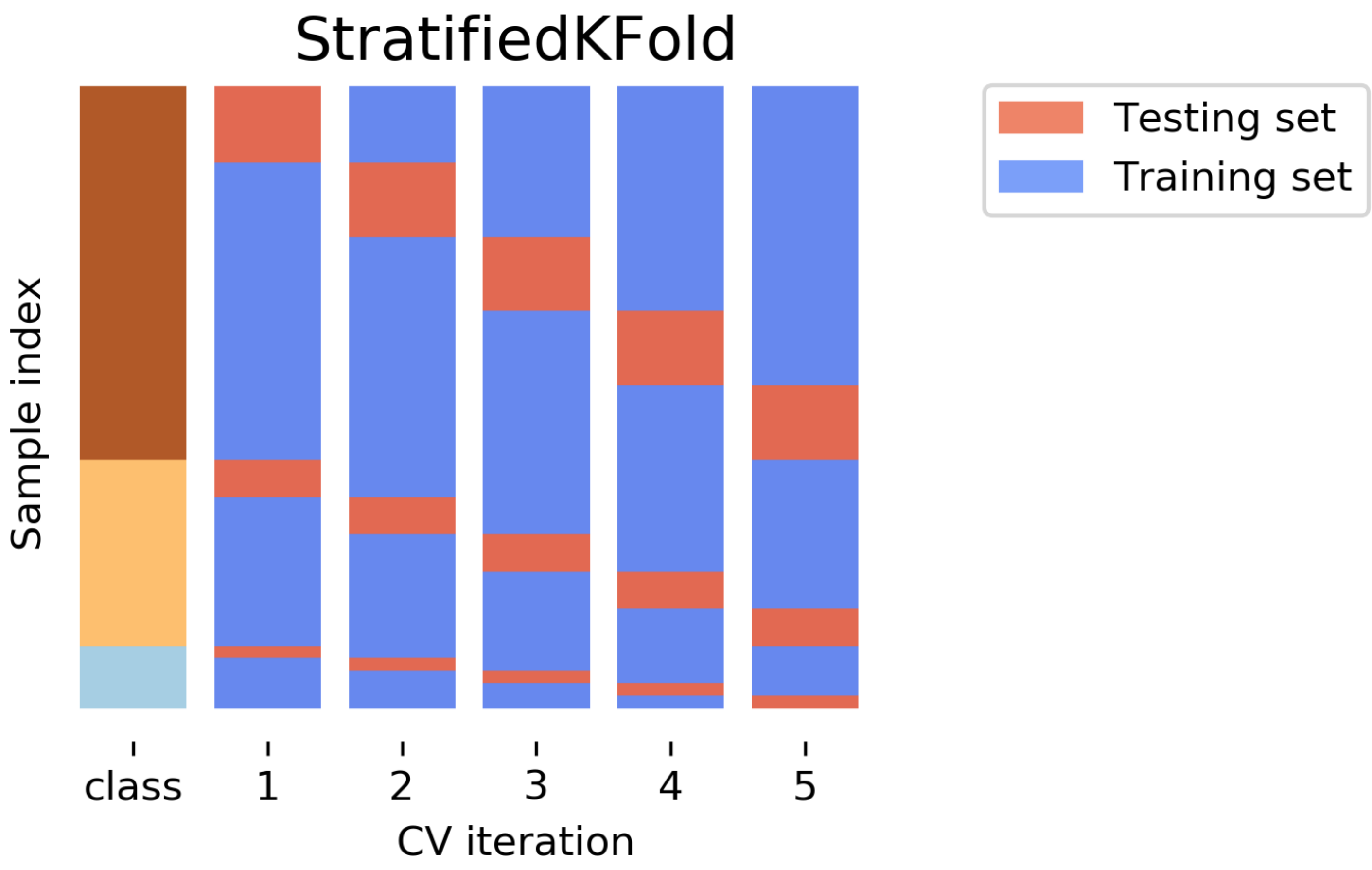
FOLD 1 | FOLD 2 | FOLD 3 | FOLD 4 | FOLD 5



1. Divide the sample data into k parts.
2. Use k-1 of the parts for training, and 1 for testing.
3. Repeat the procedure k times, rotating the test set.
4. Determine an expected performance metric (mean square error, misclassification error rate, confidence interval, or other appropriate metric) based on the results across the iterations

Stratified K-fold cross-validation

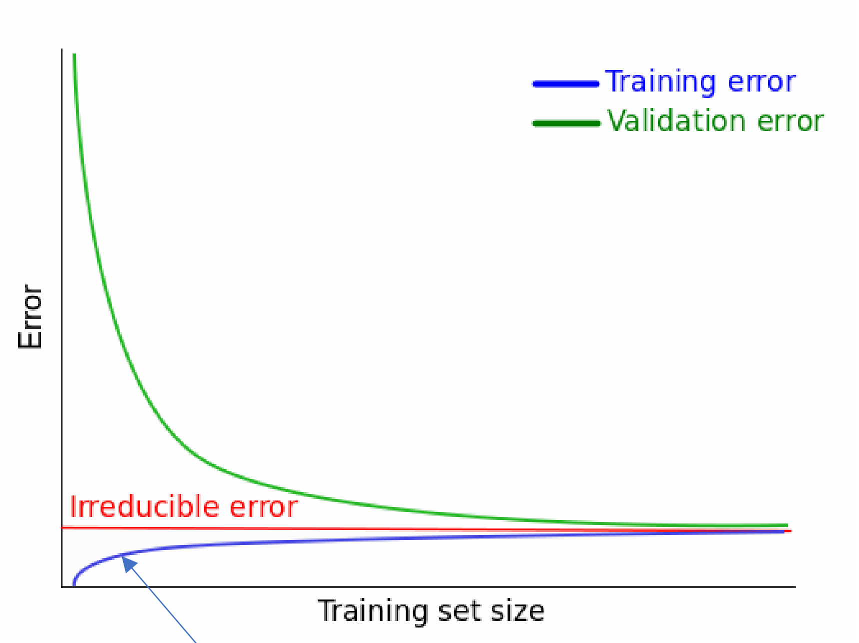
- Needed when the classes are unbalanced
- Aims to maintain data set distribution in each fold.



Choosing K

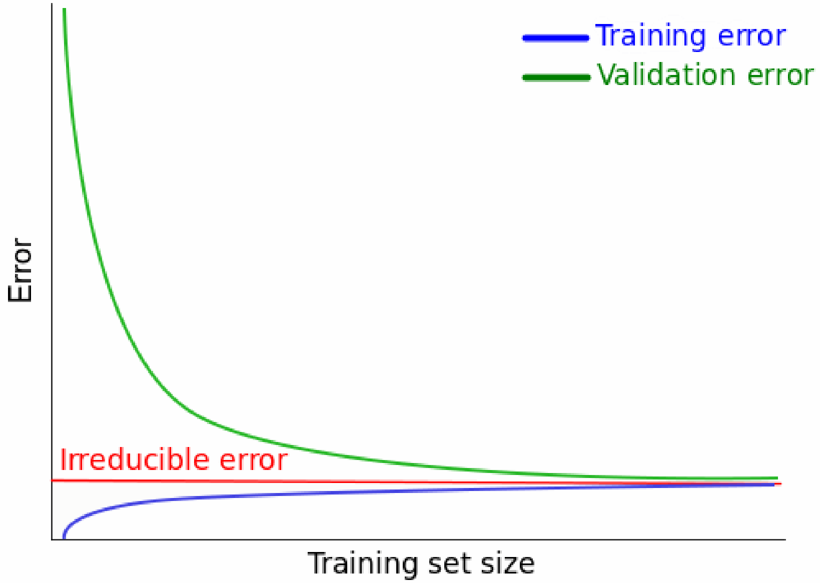
- Two extreme cases:
  - $K = 2$
  - $K = N$ (Leave-one-out)
- With $K = 2$ there's no correlation between the learned models, while with $K > 2$ there is some overlap between the data used for each model.
- Could you foresee a problem with $K = 2$?

Learning curves



If $^{N}/_{2}$ is here, we will overestimate the generalization error!

Choosing K



- Two extreme cases:
  - $K = 2$
  - $K = N$ (Leave-one-out)
- With $K = 2$ there's no correlation between the learned models, while with $K > 2$ there is some overlap between the data used for each model.
- Could you foresee a problem with $K = 2$?
- The correlation is maximum when $K = N$ how many data points will share two models?
- Let's see what happens when we apply Leave-one-out to the linear regression problem.

Leave-one-out cross-validation for linear fit

- The average predicted error will be:

$$\varepsilon_{loo} = \frac{1}{N}\sum_{i}\left(\hat{y}_i^{(-i)} - y_i\right)^2$$

- It can be shown that this is equivalent to:

$$\varepsilon_{loo} = \frac{1}{N}\sum_{i}\left(\frac{\hat{y}_i - y_i}{1 - M_{ii}}\right)^2$$

- Let’s recall $tr(M) = p + 1$ so, $\langle M_{ii}\rangle = \frac{p+1}{N} = \gamma$. We can do:

$$\varepsilon_{loo} \approx \frac{1}{N}\sum_i\left(\frac{\hat{y}_i - y_i}{1-\gamma}\right)^2$$

$M = X^T(XX^T)^{-1}X$

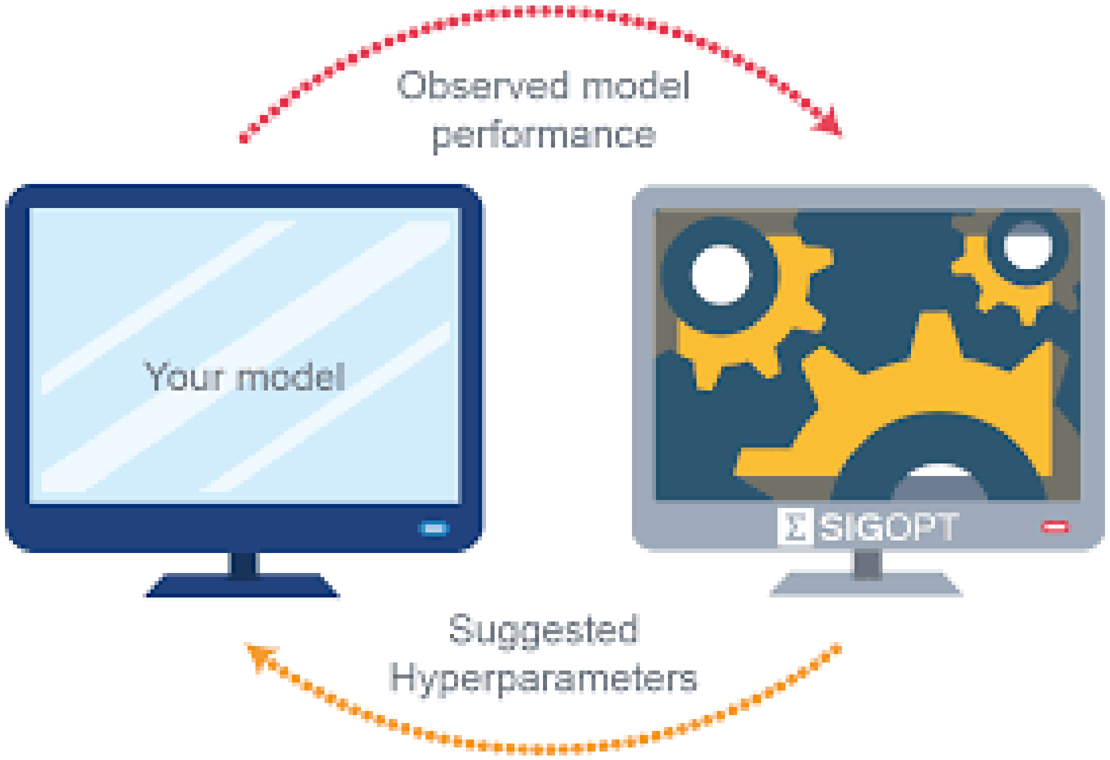
Leave-one-out cross-validation for linear fit

- $\varepsilon_{loo} \approx \frac{1}{N}\sum_i \left(\frac{\hat{y}_i - y_i}{1-\gamma}\right)^2$
- $(1-\gamma)^{-2} \approx 1 + 2\gamma$ (Taylor expansion)
- $\varepsilon_{loo} \approx \frac{1+2\gamma}{N}\sum_i(\hat{y}_i - y_i)^2 = \frac{1}{N}\sum_i(\hat{y}_i - y_i)^2 + 2\gamma\frac{1}{N}\sum_i(\hat{y}_i - y_i)^2 =$ $\frac{1}{N}\sum_i(\hat{y}_i - y_i)^2 + \underset{\text{Optimism}}{\boxed{\frac{p+1}{N}2\sigma^2}}$

Choosing K

- In terms of accuracy, LOO often results in high variance as an estimator for the test error. Intuitively, since $N - 1$ of the Nsamples are used to build each model, models constructed from folds are virtually identical to each other and to the model built from the entire training set.
- However, if the learning curve is steep for the training size in question, then 5- or 10- fold cross validation can overestimate the generalization error.
- As a general rule, most authors, and empirical evidence, suggest that 5- or 10- fold cross validation should be preferred to LOO.

Cross-validation and hyperparameter tuning



Classification Assessment measures

Can we squeeze the classification assessment for obtaining more information?

A test case: Predicting the development of a disease from genetic markers

- We have several models that predict if a person would develop a given disease within its live based on some genetic markers.

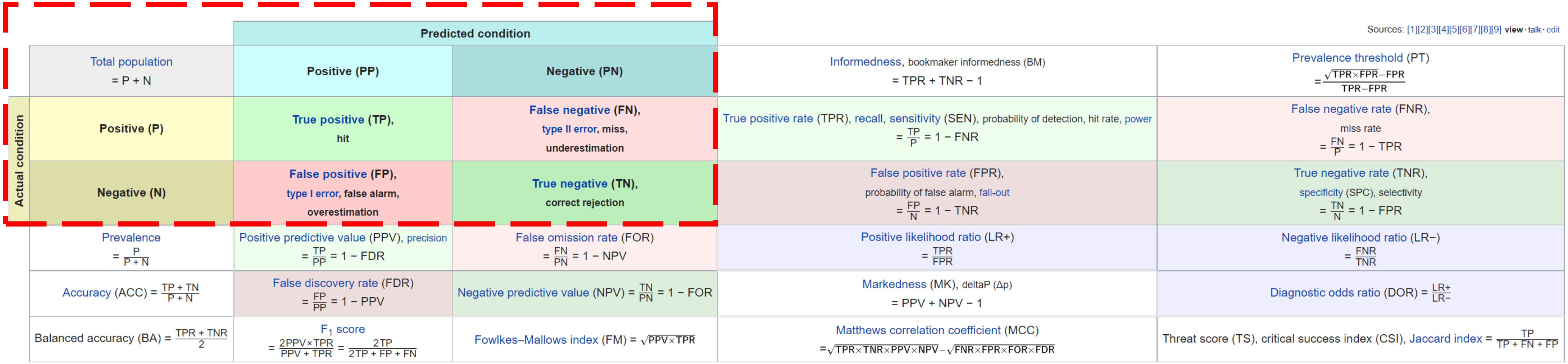
Total number of people in the study: 750
Total number of people that developed the disease: 150

Which is the result of the baseline model in this case?

Baseline model for classification

- The baseline result will assign all the elements to the majority class
- In this case, the baseline prediction will be that no one of the 750 patients will develop the disease
- It is important to specify the baseline prediction any time we report a quality measure, we can obtain extremely good results with a baseline model due to imbalance.

Binary classification metrics

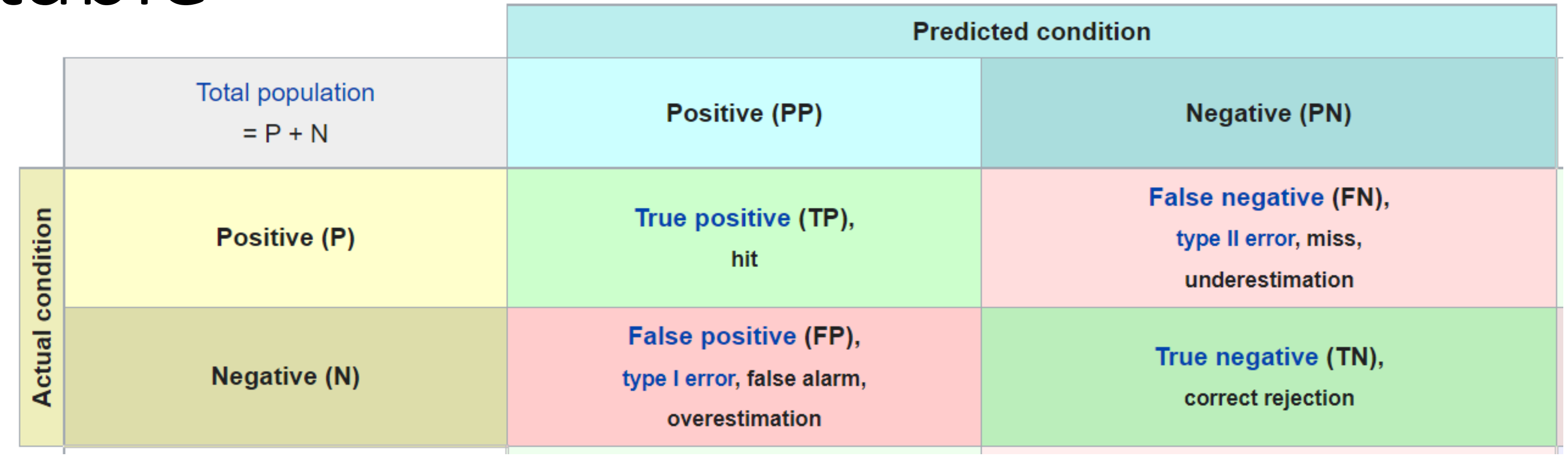


Sources: [1][2][3][4][5][6][7][8][9] **view** · talk · edit

		Predicted condition			
	Total population = P + N	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) = TPR + TNR − 1	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP),** **hit**	**False negative (FN),** **type II error, miss,** **underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P}$ = 1 − FNR	False negative rate (FNR), miss rate $= \frac{FN}{P}$ = 1 − TPR
	Negative (N)	**False positive (FP),** **type I error, false alarm,** **overestimation**	**True negative (TN),** **correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N}$ = 1 − TNR	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N}$ = 1 − FPR
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP}$ = 1 − FDR	False omission rate (FOR) $= \frac{FN}{PN}$ = 1 − NPV	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP}$ = 1 − PPV	Negative predictive value (NPV) $= \frac{TN}{PN}$ = 1 − FOR	Markedness (MK), deltaP (Δp) = PPV + NPV − 1	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2 TP}{2 TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Source: Wikipedia

Summarizing the assessment of two models in a table



		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection

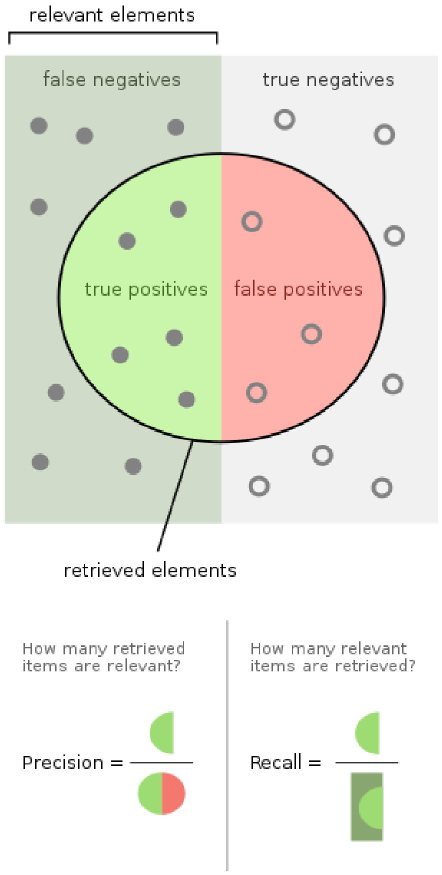
		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

Which is the crucial difference between false positive and false negative?

Simple measurements

- Prevalence: Rate of positive cases in the ground truth.
- True Positive Rate (recall/sensitivity): Conditional probability of predicting 1 given that the true label is 1.
- True Negative Rate (specificity): Conditional probability of predicting 0 given that the true label is 0.
- False Positive Rate: probability of type-I error.
- False Negative Rate : probability of type-II error.
- Positive Predictive Value (precision): Conditional probability of having a true label of 1 given that the prediction is 1.
- Negative Predictive Value: Conditional probability of having a true label of 0 given that the prediction is 0.

Precision and recall



True Positive Rate (recall/sensitivity): Conditional probability of predicting 1 given that the true label is 1. $TPR = \frac{TP}{P}$. Of all the patients that developed the disease, how many we predict?

Positive Predictive Value (precision) : Conditional probability of having a true label of 1 given that the prediction is 1. $PPV = \frac{TP}{PP}$ Of all the patients that we predicted that will develope the disease, how many actually did it?

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$$TPR_{bm} = \frac{TP}{P} = \frac{0}{150} = 0.0 \qquad PPV_{bm} = \frac{TP}{PP} = \frac{0}{0} = 0.0$$

$$TPR_{M1} = \frac{TP}{P} = \frac{90}{150} = 0.6 \qquad PPV_{M1} = \frac{TP}{PP} = \frac{90}{180} = 0.5$$

Exercise

- Let’s compute together the rest of the simple measurements:
  - Prevalence: Rate of positive cases in the ground truth: $Prev = \frac{P}{P+N}$
  - True Negative Rate (specificity): Conditional probability of predicting 0 given that the true label is 0: $TNR = \frac{TN}{N}$
  - False Positive Rate: probability of type-I error. $FPR = \frac{FP}{N}$
  - False Negative Rate : probability of type-II error. $FNR = \frac{FN}{P}$
  - Negative Predictive Value: Conditional probability of having a true label of 0 given that the prediction is 0: $NPV = \frac{TN}{PN}$

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

Accuracy

$$ACC = \frac{TP + TN}{P + N}$$

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$$ACC_{bm} = \frac{0 + 600}{750} = 0.8 \qquad ACC_{M1} = \frac{90 + 510}{750} = 0.8$$

Are these models equivalent?

Derived measures

- Accuracy alone can be misleading in unbalanced data sets
- Other measures aim to improve the accuracy measure:
  - Balanced Accuracy
  - F_1 score
  - Fowlkes-Mallows index
  - Matthews Correlation Coefficient
  - Jaccard Index

Balanced Accuracy

$$BA = \frac{TPR + TNR}{2} = \frac{\frac{TP}{P} + \frac{TN}{N}}{2}$$

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$$BA_{bm} = \frac{0. + 1.0}{2} = 0.5$$

$$BA_{M1} = \frac{0.6 + 0.85}{2} = 0.725$$

F_1 score

$$F1 = \boxed{\frac{TPR \times PPV}{TPR + PPV}} = \frac{2TP}{2TP + FP + FN}$$

Equal importance of precision and recall

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$$F1_{bm} = \frac{2 \times 0}{2 \times 0 + 0 + 150} = 0.0 \quad F1_{M1} = \frac{2 \times 90}{2 \times 90 + 90 + 60} = 0.55$$

Fowlkes-Mallows index

$$FM = \sqrt{TPR \times PPV} = \sqrt{\frac{TP}{P} \times \frac{TP}{PP}}$$

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$$FM_{bm} = \sqrt{\frac{0}{150} \times \frac{0}{0}} = 0.0 \quad FM_{M1} = \sqrt{\frac{90}{150} \times \frac{90}{180}} = 0.55$$

Matthews Correlation Coefficient

$$MCC = \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$$

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$MCC_{bm} = 0.0$ $\qquad$ $MCC_{M1} = ?$

Jaccard Index

$$J = \frac{TP}{TP + FP + FN}$$

		Predicted by baseline model		Predicted by model 1	
	Total pop (P+N) 750	Predicted Positive (PP)	Predicted Negative (PN)	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	0	150	90	60
	Negative (N)	0	600	90	510

$$J_{bm} = \frac{0}{0 + 0 + 150} = 0.0$$

$$J_{M1} = \frac{90}{90 + 90 + 60} = 0.375$$

Assignation

- The prediction of classification methods is usually not a direct assignation, but a probability.
- This means that we have to transform a real number into a class.
- Usually, for binary classification, we have that:

$$g_i(x) = \begin{cases} 0, \hat{f}(x) < 0.5 \\ 1, \hat{f}(x) \geq 0.5 \end{cases}$$

- But, in general, we can define:

$$g_i(x) = \begin{cases} 0, \hat{f}(x) < \alpha \\ 1, \hat{f}(x) \geq \alpha \end{cases}$$

Where α corresponds to the threshold value.

Assignation: Limiting cases

$$g_i(x) = \begin{cases} 0, \hat{f}(x) < \alpha \\ 1, \hat{f}(x) \geq \alpha \end{cases}$$

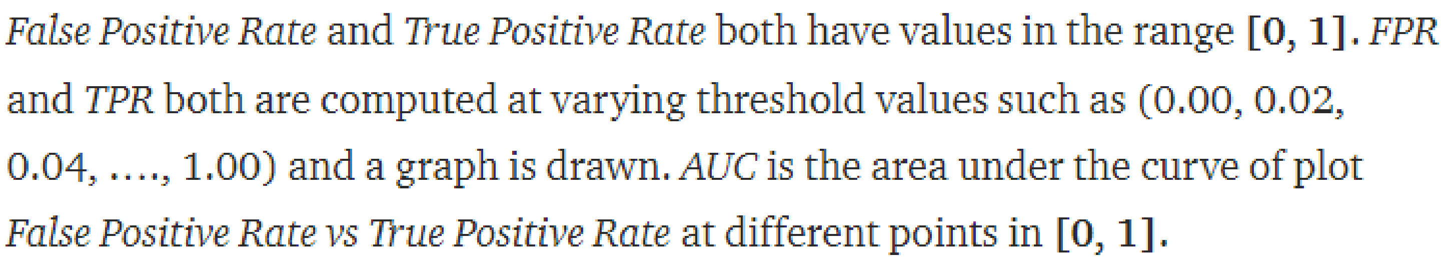
Where α corresponds to the threshold value.

What happens in the limiting cases?

- α=1 → $g_i(x)$=0 → $FPR = \frac{FP}{N} = ?; TPR = \frac{TP}{P} = ?$
- α=0 → $g_i(x)$=1 → FPR=?; TPR=?

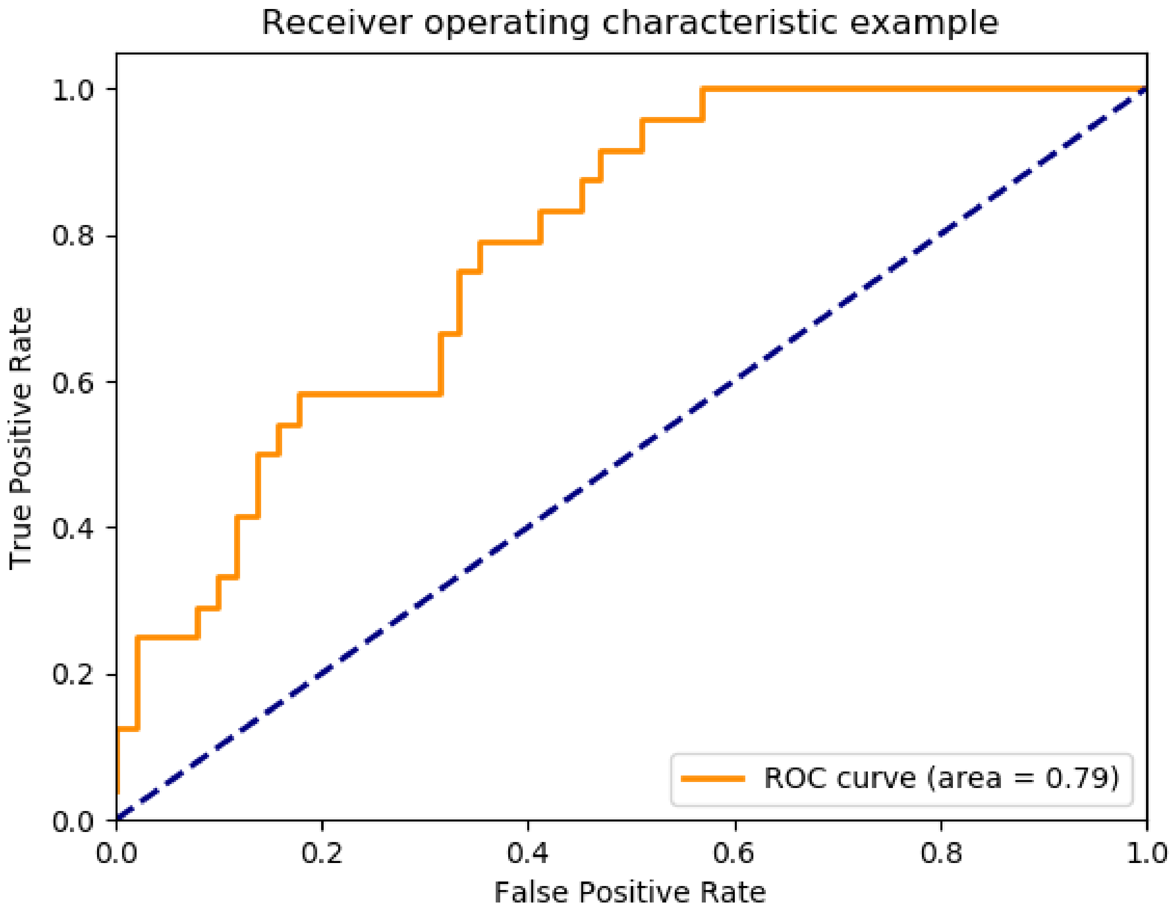
What happens in the middle?

AUC, ROC



False Positive Rate and *True Positive Rate* both have values in the range **[0, 1]**. *FPR* and *TPR* both are computed at varying threshold values such as (0.00, 0.02, 0.04, …., 1.00) and a graph is drawn. *AUC* is the area under the curve of plot *False Positive Rate vs True Positive Rate* at different points in **[0, 1]**.

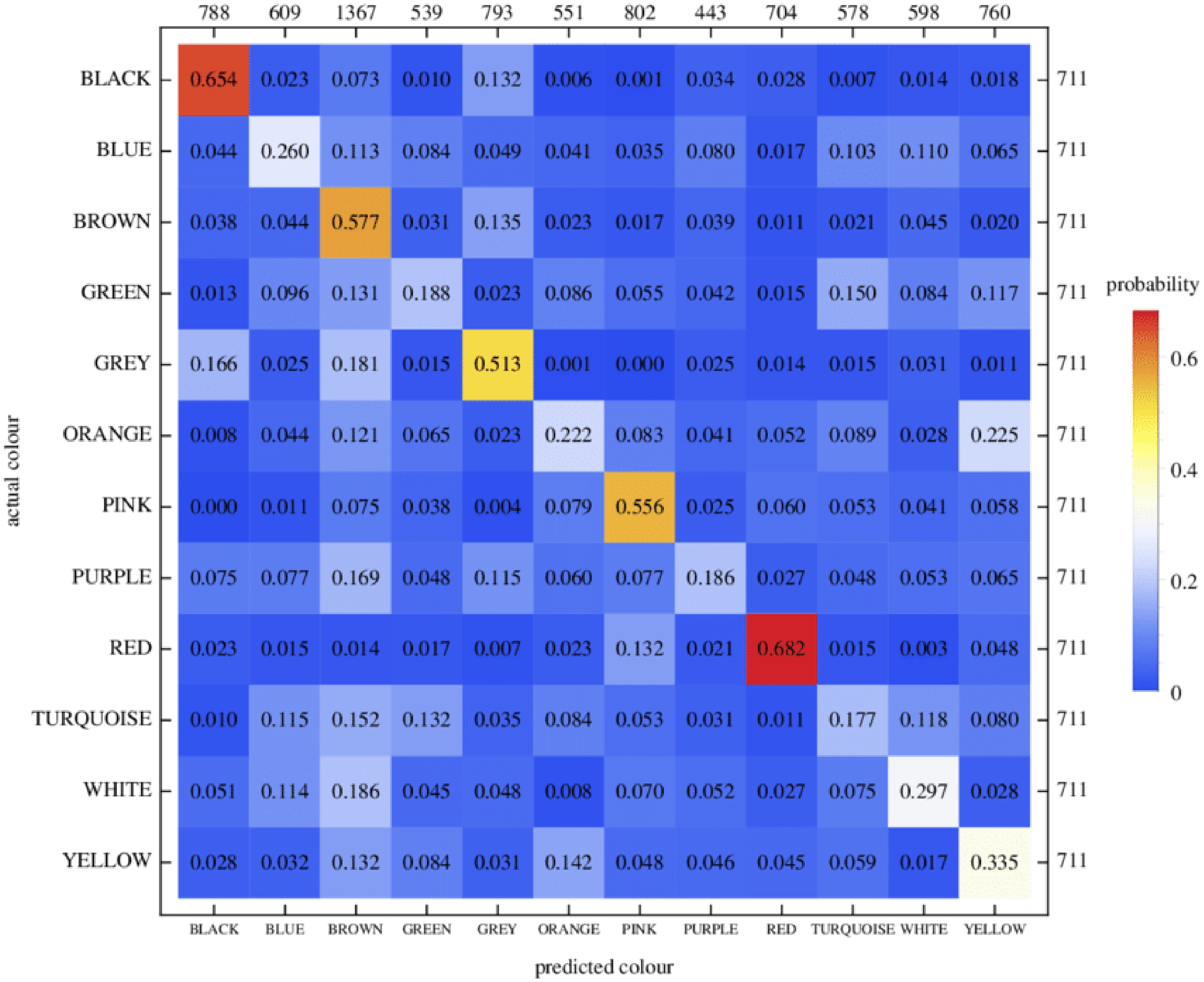
The dashed line is the chance model



Extending quality measures to multi-class problems

- All these measures are for binary classification
- What happens in multiclass problems?
- And if the number of predicted classes and the ground truth are not the same? This problem usually happens in unsupervised learning , but it may happen also in some supervised learning problems.

Confusion matrix



	788	609	1367	539	793	551	802	443	704	578	598	760	
BLACK	0.654	0.023	0.073	0.010	0.132	0.006	0.001	0.034	0.028	0.007	0.014	0.018	711
BLUE	0.044	0.260	0.113	0.084	0.049	0.041	0.035	0.080	0.017	0.103	0.110	0.065	711
BROWN	0.038	0.044	0.577	0.031	0.135	0.023	0.017	0.039	0.011	0.021	0.045	0.020	711
GREEN	0.013	0.096	0.131	0.188	0.023	0.086	0.055	0.042	0.015	0.150	0.084	0.117	711
GREY	0.166	0.025	0.181	0.015	0.513	0.001	0.000	0.025	0.014	0.015	0.031	0.011	711
ORANGE	0.008	0.044	0.121	0.065	0.023	0.222	0.083	0.041	0.052	0.089	0.028	0.225	711
PINK	0.000	0.011	0.075	0.038	0.004	0.079	0.556	0.025	0.060	0.053	0.041	0.058	711
PURPLE	0.075	0.077	0.169	0.048	0.115	0.060	0.077	0.186	0.027	0.048	0.053	0.065	711
RED	0.023	0.015	0.014	0.017	0.007	0.023	0.132	0.021	0.682	0.015	0.003	0.048	711
TURQUOISE	0.010	0.115	0.152	0.132	0.035	0.084	0.053	0.031	0.011	0.177	0.118	0.080	711
WHITE	0.051	0.114	0.186	0.045	0.048	0.008	0.070	0.052	0.027	0.075	0.297	0.028	711
YELLOW	0.028	0.032	0.132	0.084	0.031	0.142	0.048	0.046	0.045	0.059	0.017	0.335	711
	BLACK	BLUE	BROWN	GREEN	GREY	ORANGE	PINK	PURPLE	RED	TURQUOISE	WHITE	YELLOW	

Pair-counting measures

Measure the number of pairs that are in:

Same class **both** in *P* and *G*.

$$a = \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}(n_{ij}-1)$$

Same class in *P* but different in *G*.

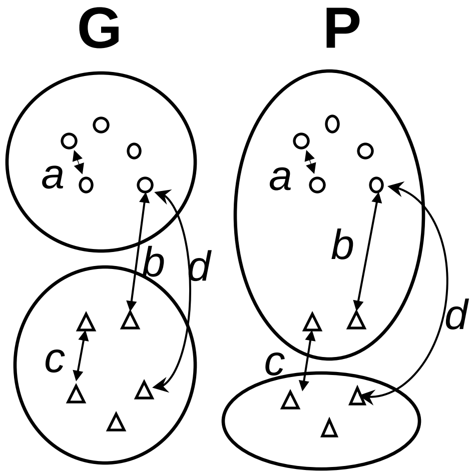
$$b = \frac{1}{2}(\sum_{j=1}^{K'} n_{.j}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2)$$

Different classes in *P* but same in *G*.

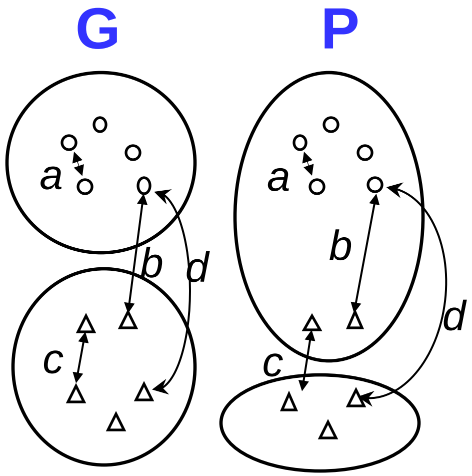
$$c = \frac{1}{2}(\sum_{i=1}^{K} n_{i.}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2)$$

Different classes **both** in *P* and *G*.

$$d = \frac{1}{2}(N^2 + \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2 - (\sum_{i=1}^{K} n_{i.}^2 + \sum_{j=1}^{K'} n_{.j}^2))$$



Rand index



Agreement: *a*, *d*
Disagreement: *b*, *c*

$$RI(P,G) = \frac{a+d}{a+b+c+d}$$

Rand index

(example)

Vectors assigned to:	Same cluster	Different clusters
Same cluster in ground truth	20	24
Different clusters in ground truth	20	72

Rand index = (20+72) / (20+24+20+72) = 92/136 = **0.68**

Mutual information

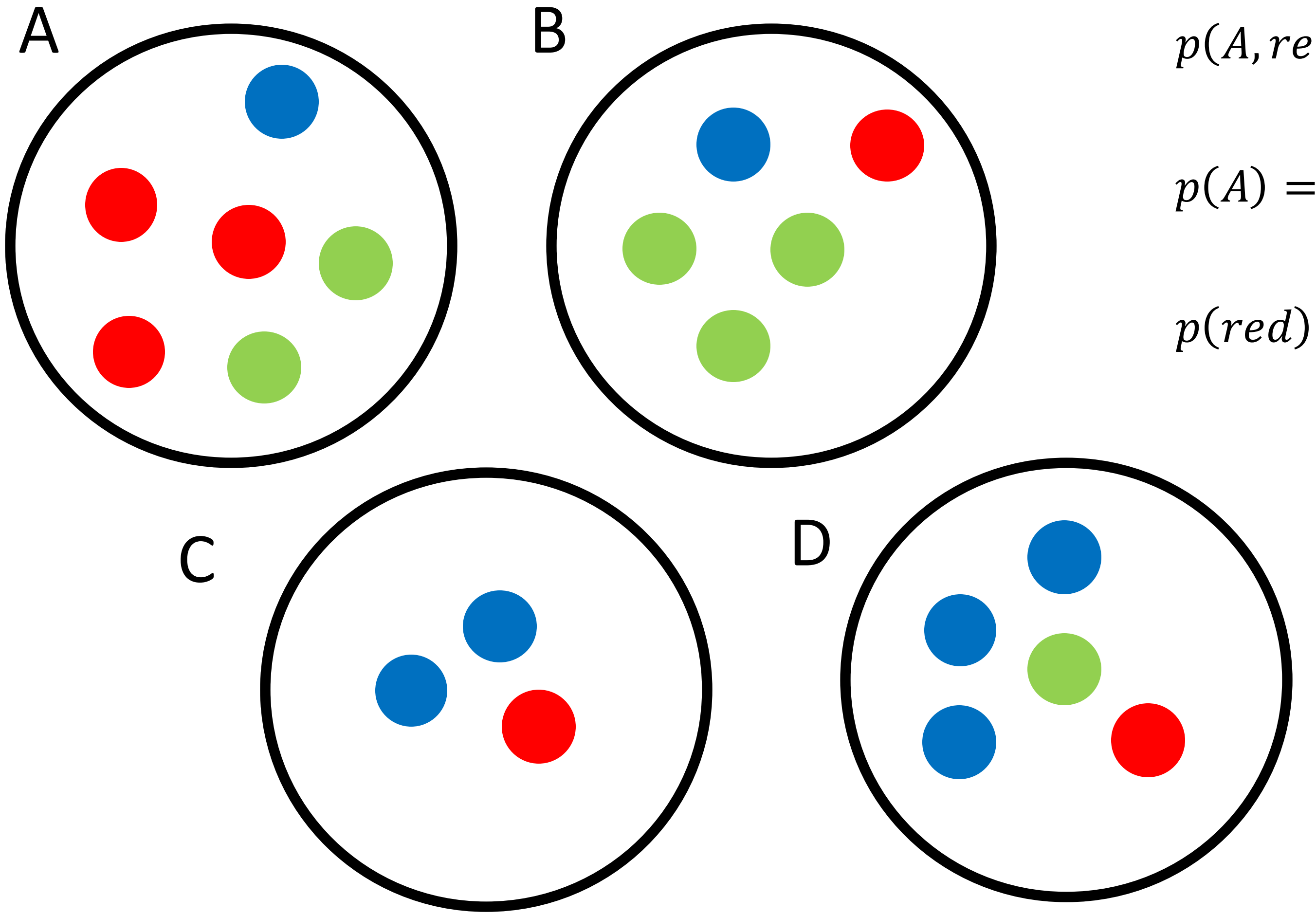
$$MI(k, G) = \sum_{l=1}^{k} \sum_{j=1}^{G} p(l, j) \log \frac{p(l, j)}{p(l)p(j)}$$

- Is a formalization of the intuition that the higher the joint distribution, the higher the mutual information, i.e. the higher should it be in the rank.

Techniques for Feature Selection (1) Information Theoretic Ranking

- The probabilities are estimated from frequency counts.
- Imagine a three class problem (red, green, blue) with a discrete variable that can take 4 values (A,B,C,D).
    - P(y) are 3 frequency counts.
    - P(x) are 4 frequency counts.
    - P(x,y) are 12 frequency counts.

Mutual Information



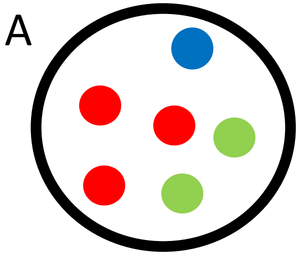
$$p(A, red) = \frac{3}{19}$$

$$p(A) = \frac{6}{19}$$

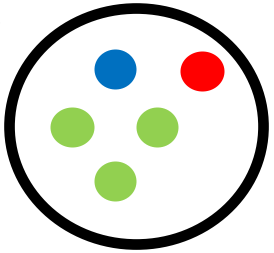
$$p(red) = \frac{6}{19}$$

Mutual Information

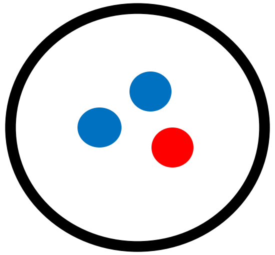
A



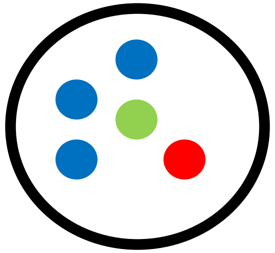
B



C



D



$$
\begin{aligned}
I = {} & \frac{3}{19}\log\left(\frac{3/19}{6/19\,6/19}\right) + \frac{1}{19}\log\left(\frac{1/19}{6/19\,7/19}\right) \\
& + \frac{2}{19}\log\left(\frac{2/19}{6/19\,6/19}\right) + \frac{1}{19}\log\left(\frac{1/19}{5/19\,6/19}\right) + \\
& \frac{1}{19}\log\left(\frac{1/19}{5/19\,7/19}\right) + \frac{3}{19}\log\left(\frac{3/19}{5/19\,6/19}\right) + \\
& \frac{1}{19}\log\left(\frac{1/19}{3/19\,6/19}\right) + \frac{0}{19}\log\left(\frac{0/19}{3/19\,7/19}\right) + \\
& \frac{2}{19}\log\left(\frac{2/19}{3/19\,6/19}\right) + \frac{1}{19}\log\left(\frac{1/19}{5/19\,6/19}\right) + \\
& \frac{1}{19}\log\left(\frac{1/19}{5/19\,7/19}\right) + \frac{3}{19}\log\left(\frac{3/19}{5/19\,6/19}\right) \approx 0.17
\end{aligned}
$$

Normalized Mutual Information

$$MI(k,G) = \sum_{l=1}^{k} \sum_{j=1}^{G} p(l,j) \log \frac{p(l,j)}{p(l)p(j)}$$

However, it does not take into account our intuitive preference for few clusters, so we normalized it:

$$NMI(k,G) = \frac{2MI}{H(k) + H(G)}$$

$$H(k) = -\sum_{l=1}^{k} p(l) \log[p(l)]$$

Summary:

- Use of R^2 for quality validation (Also $R^2_{gen}$and R^2_{adj}).
- Optimism of the training error
- In-sample criteria for model selection:
  - Mallow's
  - Akaike Information Criterion (AIC)
  - Bayes Information Criterion (BIC)
- Validation and Cross Validation:
  - Train-validation-test partition.
    - Early stopping
  - K-fold cross validation:
    - Leave-one out.
    - Choice of k
- Classification Assessment measures:
  - Confusion matrix
  - Simple measures
  - Derived measures
  - ROC, AUC
  - Extension to multiclass (Rand Index, NMI)