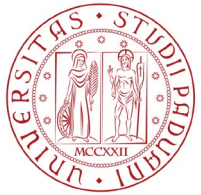


UNIVERSITÀ DEGLI STUDI DI PADOVA

DEPARTMENT OF INDUSTRIAL ENGINEERING

Machine Learning Laboratory #4

Prof. Pierantonio Facco

CAPE-Lab, Computer-Aided Process Engineering Laboratory

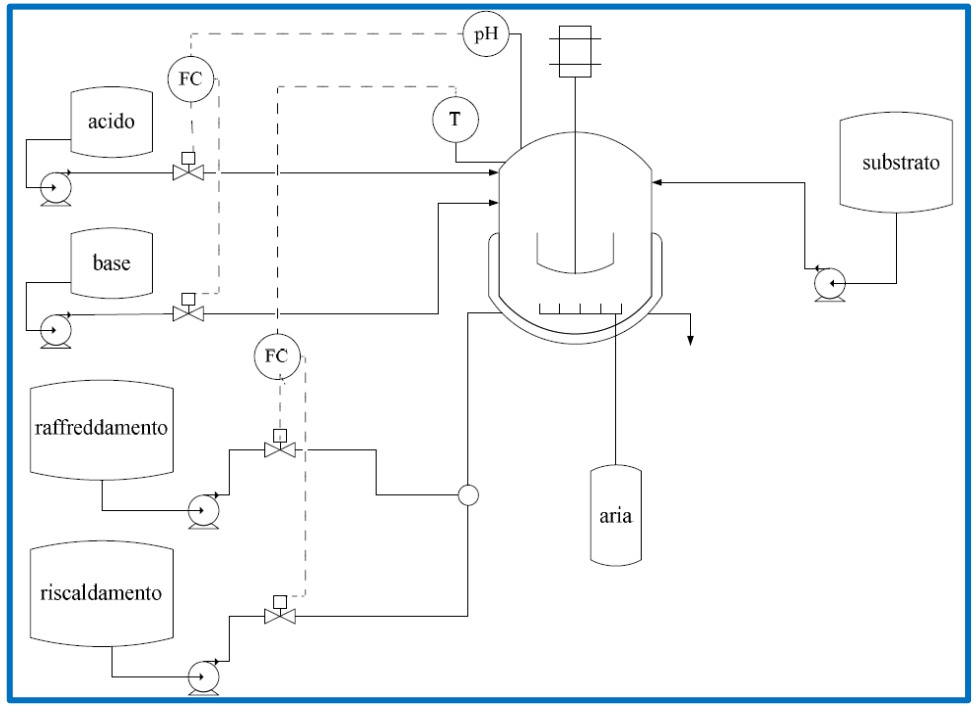
Email: pierantonio.facco@unipd.it

URL: https://research.dii.unipd.it/capelab/

Example #4: penicillin product quality prediction

Penicillin production

- Batch process for the culture cultivation + fed-batch phase of penicillin production
- Objective: **prediction of the final penicillin concentration**



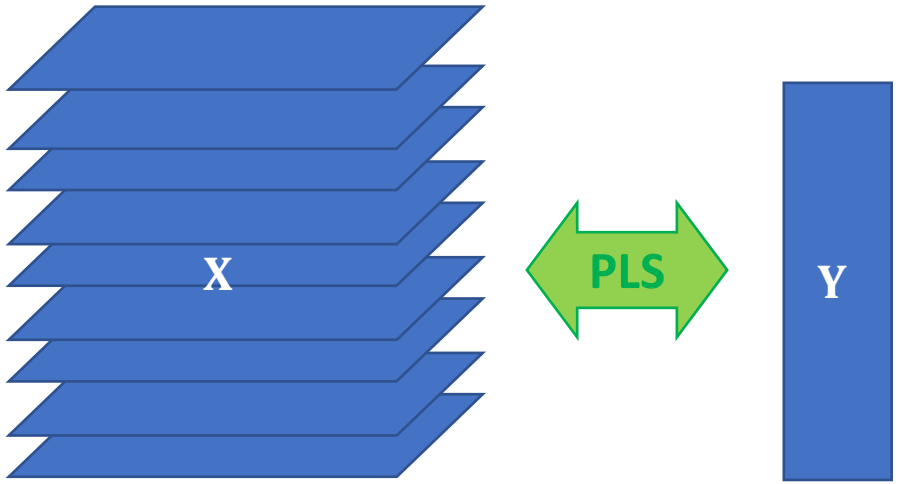
variable	name	Units
1	aeration rate	L/h
2	agitation power set point	W
3	substrate feed rate	L/h
4	substrate feed temperature	K
5	substrate concentration	g/L
6	dissolved O_2/saturation O_2	L
7	culture volume	L
8	CO_2 concentration	mmole/L
9	pH	-
10	temperature	K
11	generated heat	kcal/h
12	acid flow rate	mL/h
13	base flow rate	L/h
14	heating water flow rate	L/h

Birol, G., Ündey, C., Çinar, A., 2002. A modular simulation package for fed-batch fermentation: penicillin production. Comput. Chem. Eng. 26, 1553–1565

Process issues and available data

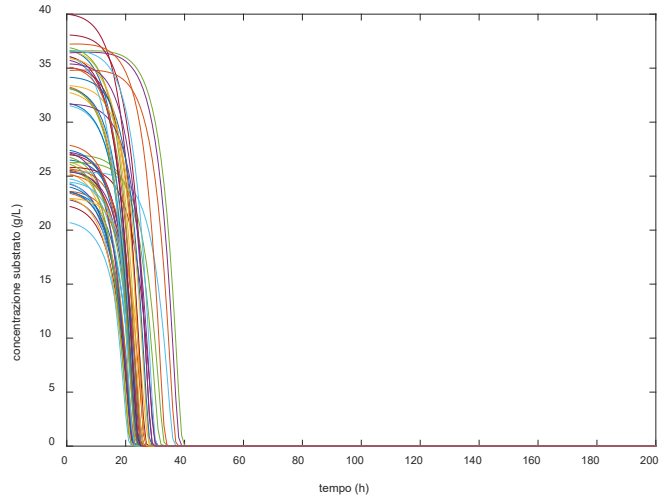
- Data:
  - 60 batches
  - $\mathbf{X}$ matrix:
    - 14 recorded variables (1observation/h)
    - batch duration 200h
  - $\mathbf{Y}$ matrix: final penicillin concentration

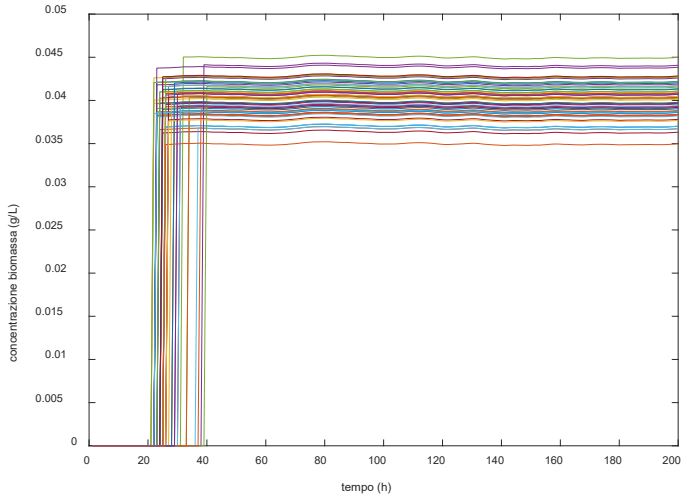
- **High variability of the initial conditions**:
  - substrate initial concentration: 25-35 g/L
  - aeration system fouling
  - agitation power: 30-35-40 W
  - pH sensor fouling

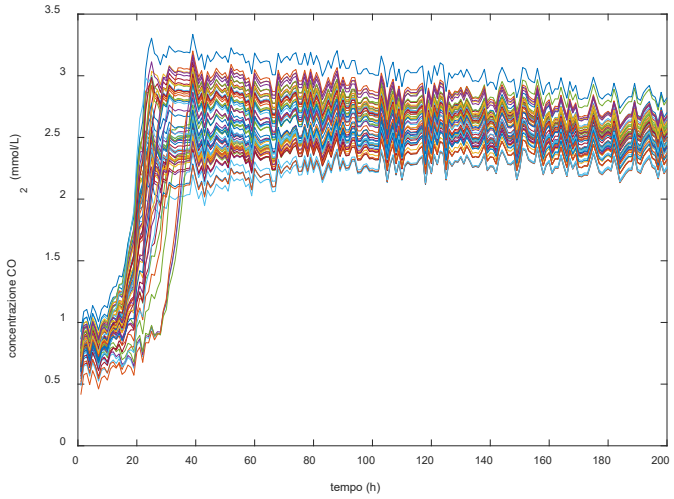


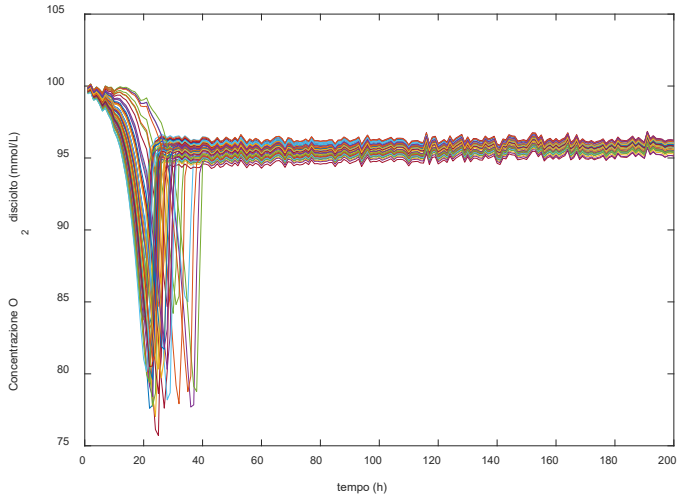
Variable time profiles visualization

- Key variables time trajectories seem to be regular





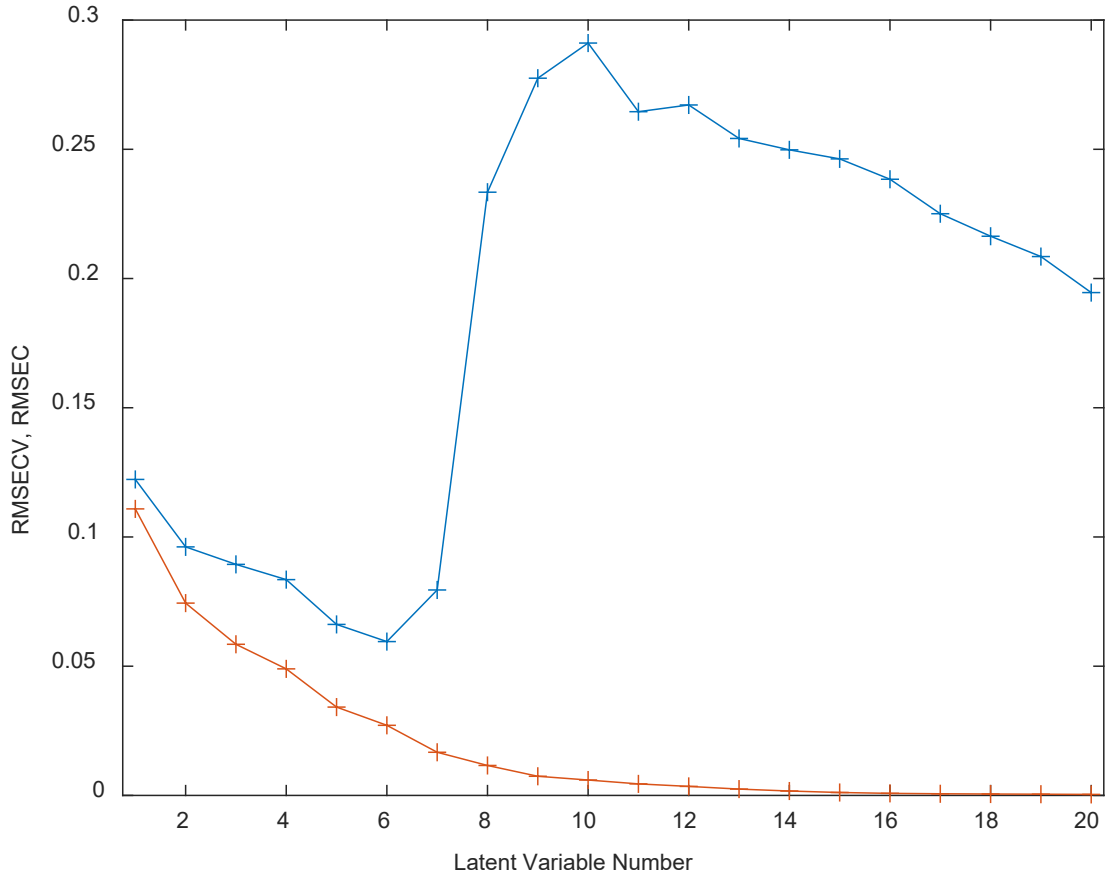




PLS modelling

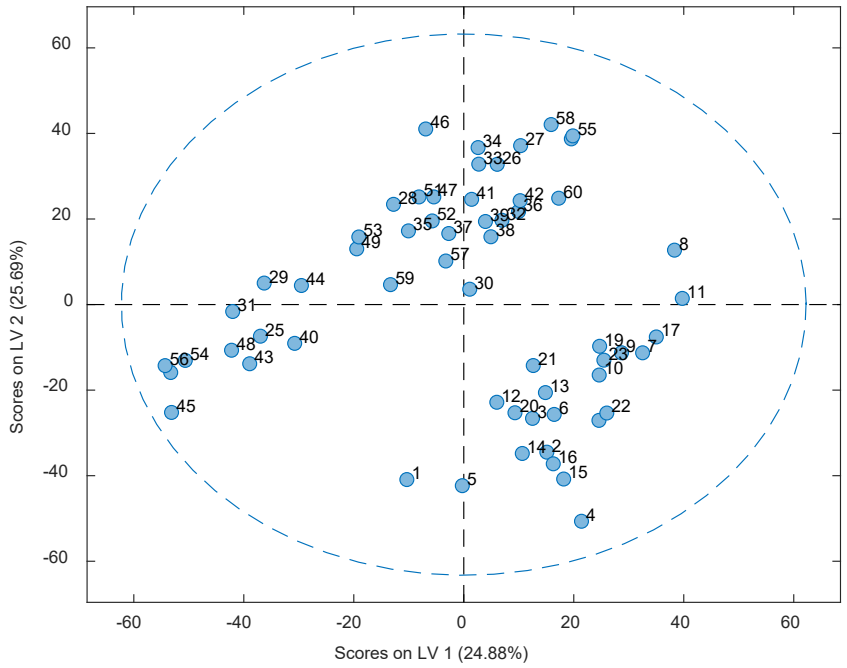
- Load: **`penicillin_fermentation.mat`**
  - select PLS model
  - calibration data: **`Xcal`**, **`Ycal`**
  - pretreatment: autoscaling
  - select 6 LVs

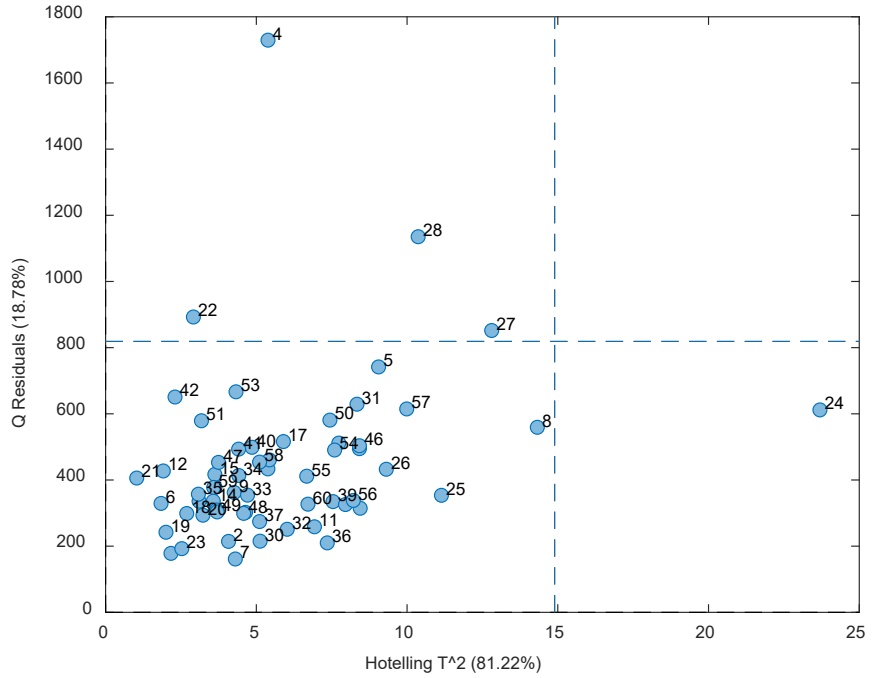
LV	X exp. var.	X cum. exp. var.	Y exp. var.	Y cum. exp. var.
1	24.88	24.88	58.24	58.24
2	25.69	50.58	22.95	81.19
3	10.20	60.78	7.19	88.38
4	9.19	69.97	3.47	91.85
5	4.93	74.90	4.18	96.03
6	6.33	81.2	1.47	97.50
7	2.61	83.83	1.55	99.05
8	1.59	85.42	0.49	99.54
9	1.10	86.52	0.27	99.81
10	2.02	88.54	0.07	99.88



Relations among batches

- Two **clusters** are shown in the score space
- Both the clusters are well represented into the model

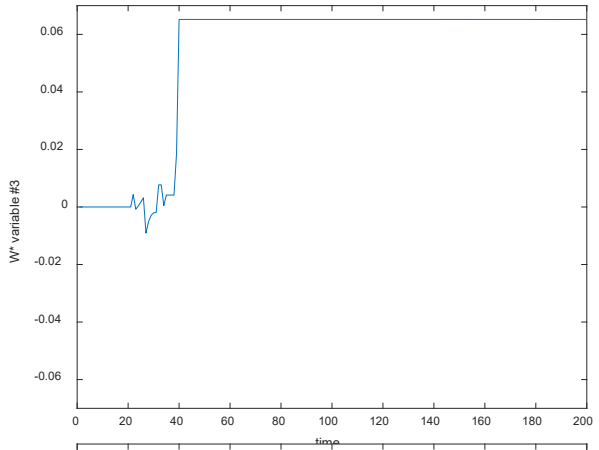


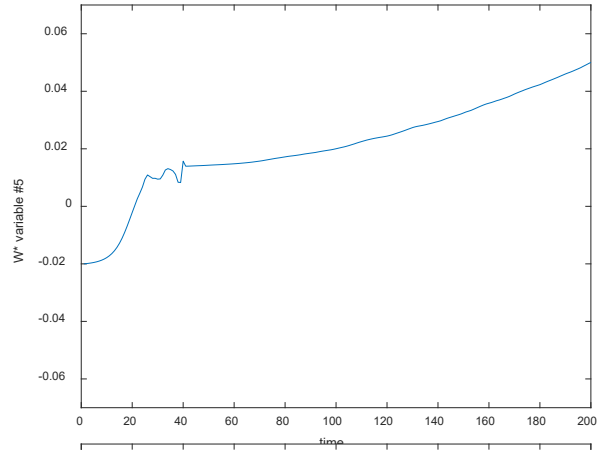


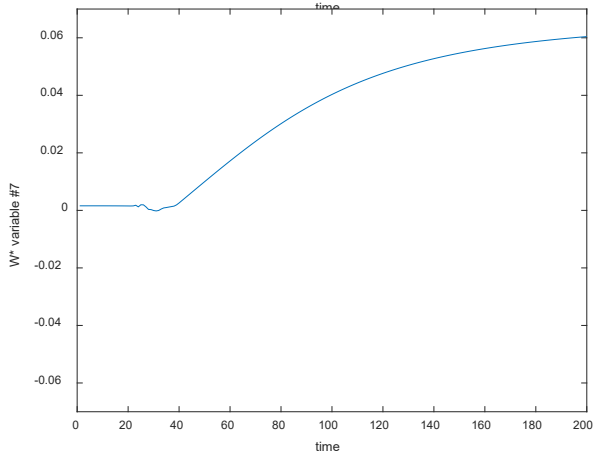
Variables mostly related to product quality (1/2)

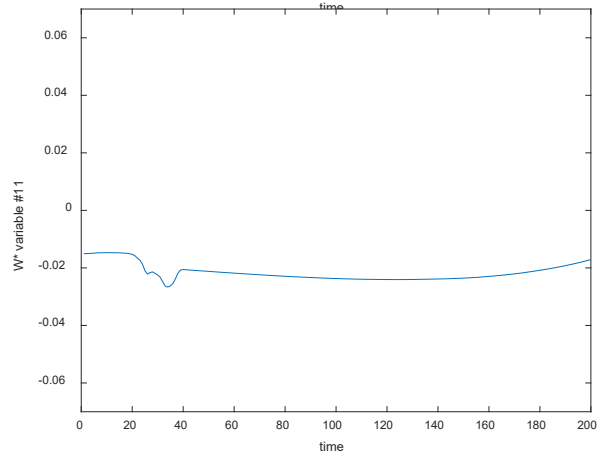
- Weights $\mathbf{W}^*$ analysis for one latent variable at a time
  - LV1 (58% of $\mathbf{Y}$ variance)

variable	name
1	aeration rate
2	agitation power set point
3	substrate feed rate
4	substrate feed temperature
5	substrate concentration
6	dissolved O_2/saturation O_2
7	culture volume
8	CO_2 concentration
9	pH
10	temperature
11	generated heat
12	acid flow rate
13	base flow rate
14	heating water flow rate





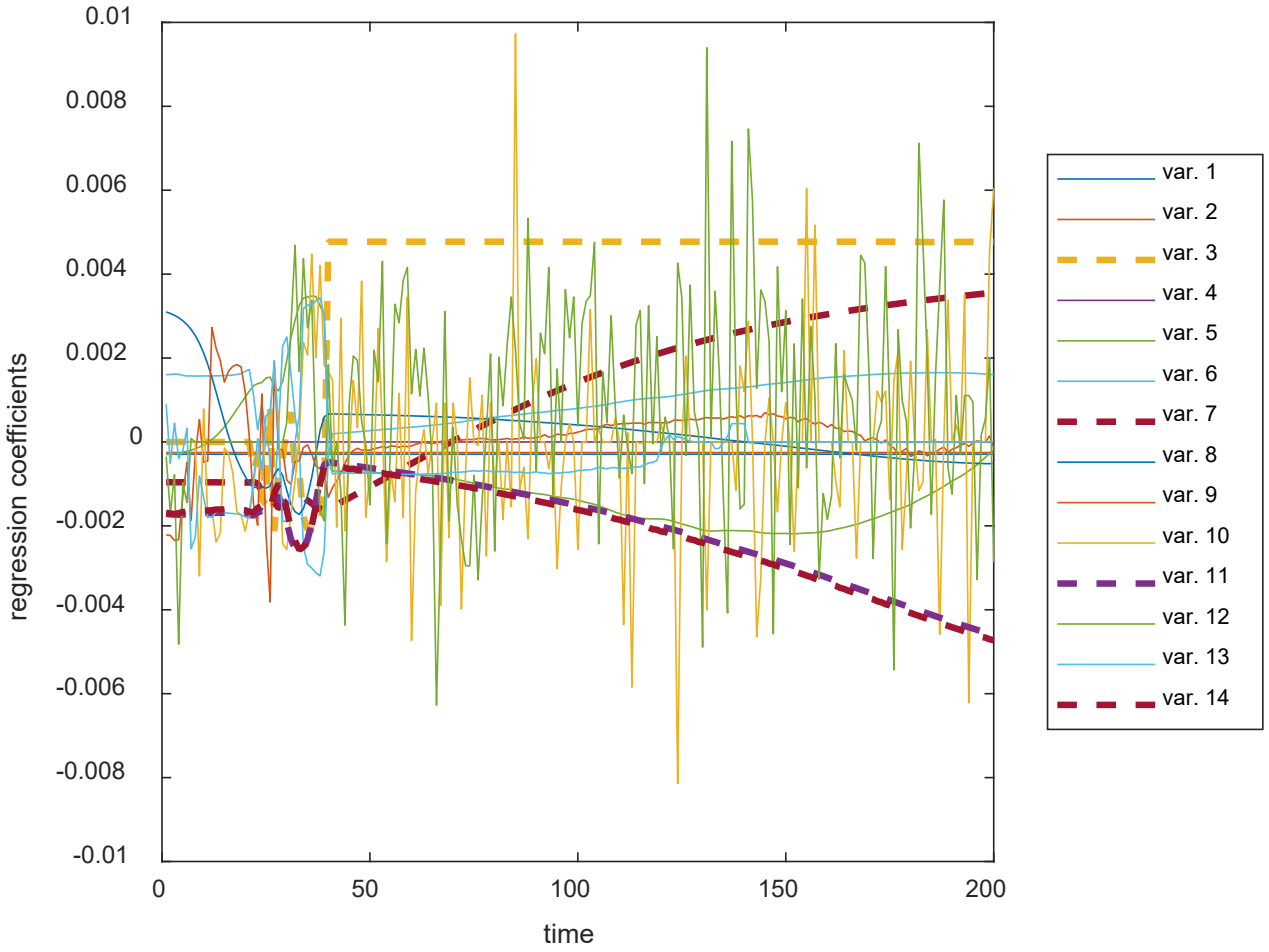




Variables mostly related to product quality (2/2)

- Regression coefficients $\mathbf{B}$ can be studied

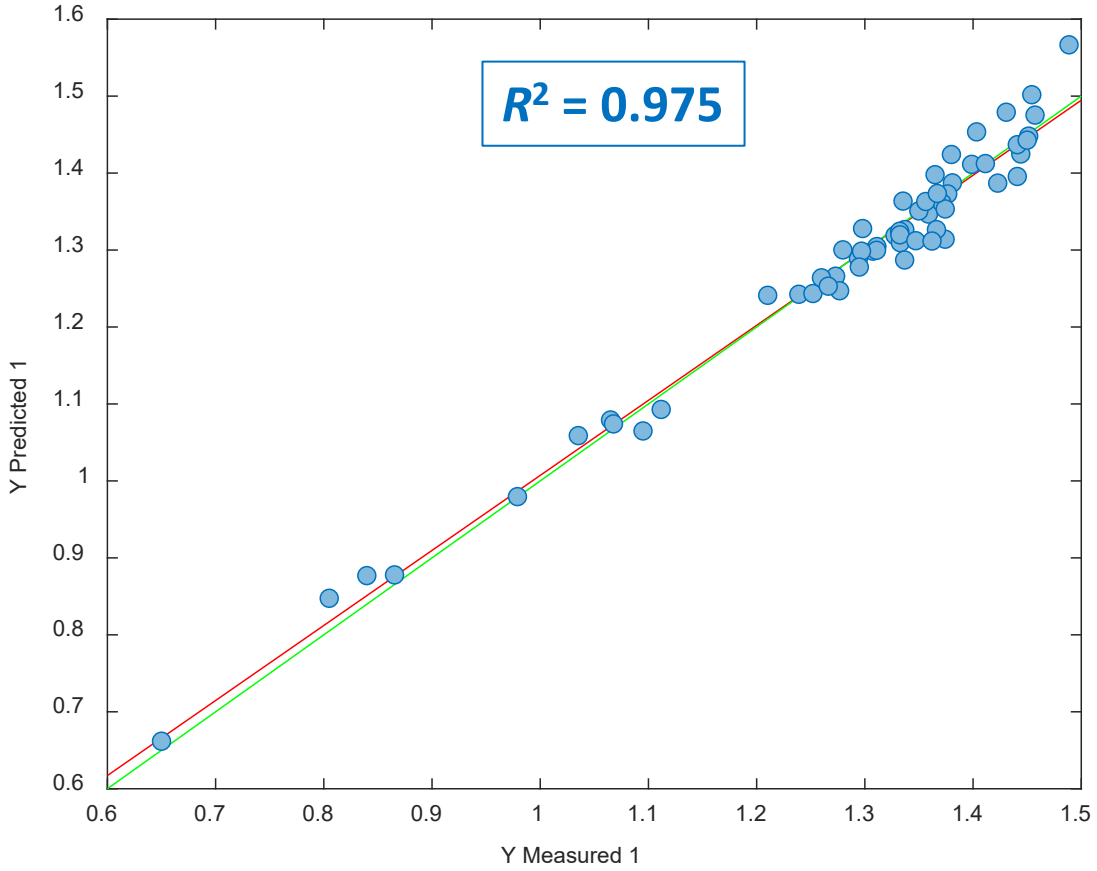
variable	name
1	aeration rate
2	agitation power set point
3	substrate feed rate
4	substrate feed temperature
5	substrate concentration
6	dissolved O_2/saturation O_2
7	culture volume
8	CO_2 concentration
9	pH
10	temperature
11	generated heat
12	acid flow rate
13	base flow rate
14	heating water flow rate



Calibration prediction performance

- **Goodness of fitting** in calibration:
  - **very high determination coefficient**

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y}_n)^2}$$



Prediction of new observations

- Load validation data: `Xval`, `Yval`
- Pretreatment: autoscaling **on mean and variance of the calibration dataset**
- Projection on the LV space:

$$\mathbf{t}_{\mathrm{NEW}} = \mathbf{x}_{\mathrm{NEW}}\mathbf{P}$$

- Estimation of the final penicillin concentration:

$$\hat{y} = \mathbf{b}\mathbf{t}_{\mathrm{NEW}}\mathbf{Q}^{\mathrm{T}}$$

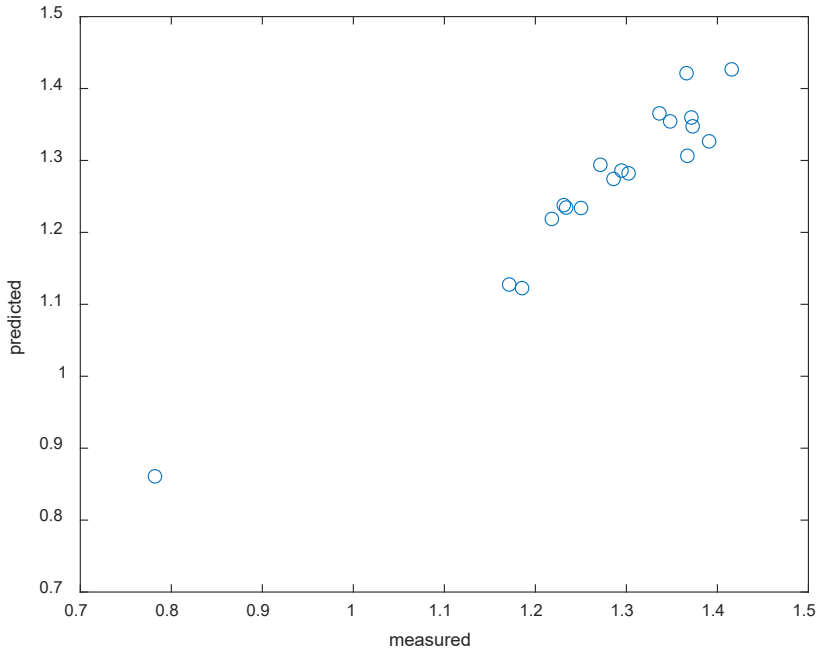
- Command: `yp=modlpred(Xval,plsm1);`

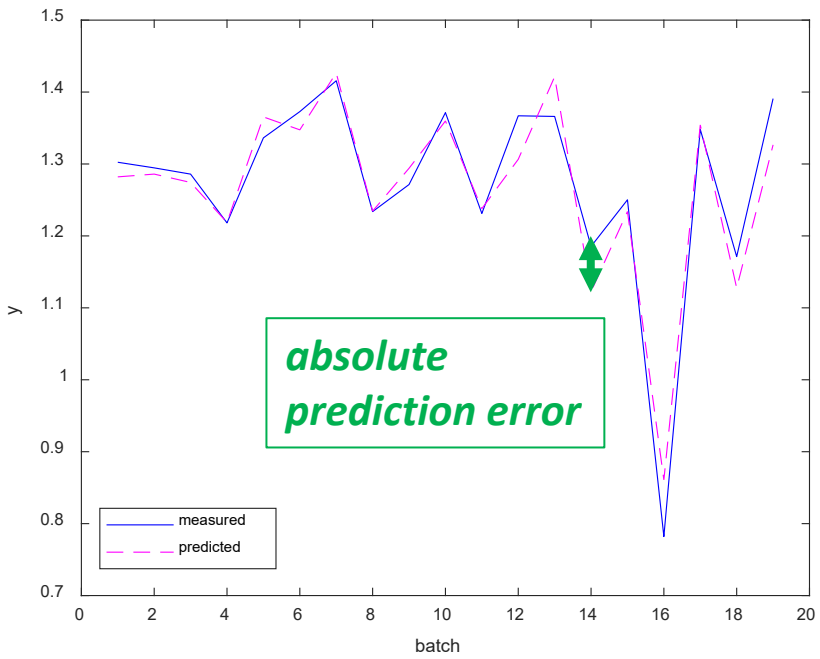
Validation estimation error

- Validation determination coefficient: $\boldsymbol{R^2 = 0.925}$
  - average absolute prediction error: $AAPE = \mathbf{0.0282\%}$
  - mean relative error: $MRE = 2.38\%$

$$AAPE = \frac{\sum_{n=1}^{N} |y_n - \hat{y}_n|}{N}$$

$$MRE = \frac{1}{N} \sum_{n=1}^{N} \frac{|y_n - \hat{y}_n|}{y_n}$$





Take-home message

- PLS is a powerful tool to correlate two blocks of multivariate data
- PLS can be used to perform accurate:
  - **estimations**
  - **predictions**
  - **process understanding**
  - **correlative analysis**
- PLS is a **flexible technique** to deal with:
  - hardware sensor data
  - chemical, physical, mechanical measurements
  - panel judgement
  - classification
  - quantitative and qualitative data
  - images
  - spectra
  - internet data
  - etc…

Batch quality estimation code (1/4)

```
close all
clear all
clc
% import data of penicillin fermentation
load(['C:\Courses\MachineLearningForProcessEngineering\penicilin_fermentation.mat']);
% data visualization
for v=1:14
        a(:,:)=X3d_cal(:,v,:);
        figure;
        plot(a');
        xlabel('time');
        ylabel(['variable #' num2str(v)]);
        box on;
end
boxplot(Ycal);
figure;plot(Ycal);xlabel('observation');ylabel('penicilin concentration')
for m=1:2
        figure;
        plot(Ycal_p(:,m:2:end)');
        xlabel('time');
        ylabel(['quality variable #' num2str(m)]);
end
```

Batch quality estimation code (2/4)

```
% PLS model building + autoscaling
o.display='off';
o.plots='none';
o.preprocessing={'autoscale' 'autoscale'};
plsm=pls(Xcal,Ycal,6,o);

T=plsm.loads{1,1};
P=plsm.loads{2,1};
U=plsm.loads{1,2};
Q=plsm.loads{2,2};

E=auto(Xcal)-T*P';
normplot(E(:,1374))
hist(E(:,1374))
figure;
scatter(T(:,1),T(:,2));xlabel('PC1 T scores');ylabel('PC2 T scores');
figure;
scatter(T(:,1),U(:,1));xlabel('PC1 T scores');ylabel('PC1 U scores');
```

Batch quality estimation code (3/4)

```
% weights
W=plsm.wts;

for v=1:14
        figure;
        plot(W(v:14:end,1));
        xlabel('time');
        ylabel(['PC1 weights of variable #' num2str(v)]);
        ylim([-0.07 0.07])
end

% regression coefficients
B=plsm.reg;
for v=1:14
        figure;
        plot(B(v:14:end,1));
        xlabel('time');
        ylabel(['regr. coef. of variable' num2str(v)]);
        ylim([-0.01 0.01])
end
```

Batch quality estimation code (4/4)

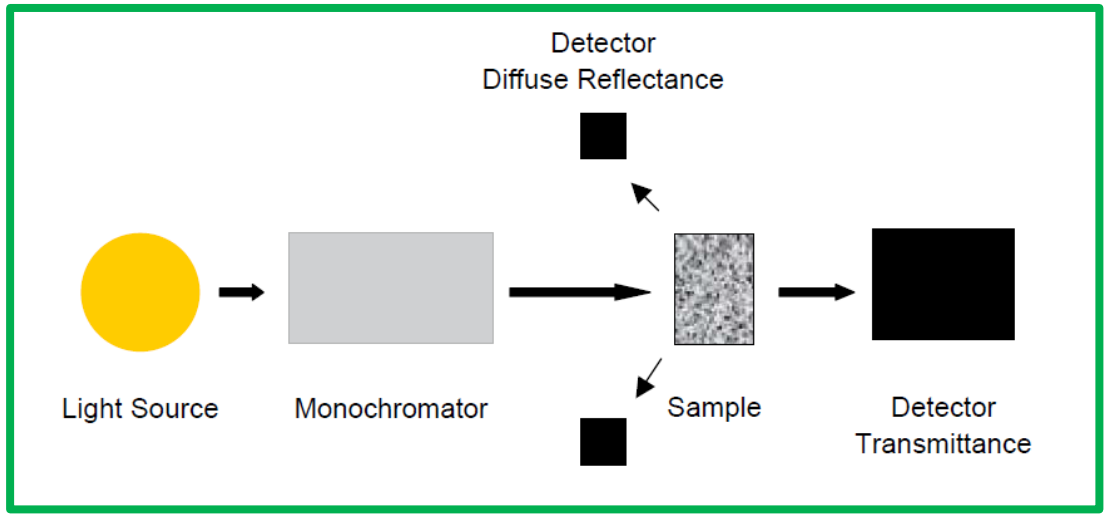
```
% monitoring charts
yhat_cal=modlpred(Xcal,plsm,0);
ec=Ycal-yhat_cal;
MAEc=mean(abs(ec));
MAEc/std(yhat_cal)
rec=(Ycal-yhat_cal)./Ycal;
MREc=mean(abs(rec));
R2c=1-sum((Ycal-yhat_cal).^2)/sum((Ycal-repmat(mean(Ycal),size(Ycal))).^2);
figure;plot(ec);xlabel('observation');ylabel('calibration error');
figure;scatter(Ycal,yhat_cal);xlabel('calibration measured y');ylabel('predicted y');box on;xlim([0.7 1.6]);ylim([0.7 1.6]);

yhat_val=modlpred(Xval,plsm,0);
ev=Yval-yhat_val;
MAEv=mean(abs(ev));
rev=(Yval-yhat_val)./Yval;
MREv=mean(abs(rev));
R2v=1-sum((Yval-yhat_val).^2)/sum((Yval-repmat(mean(Yval),size(Yval))).^2);
figure;plot(ev);xlabel('observation');ylabel('validation error');
figure;scatter(Yval,yhat_val);xlabel('validation measured y');ylabel('predicted y');box on;xlim([0.7 1.6]);ylim([0.7 1.6]);
```

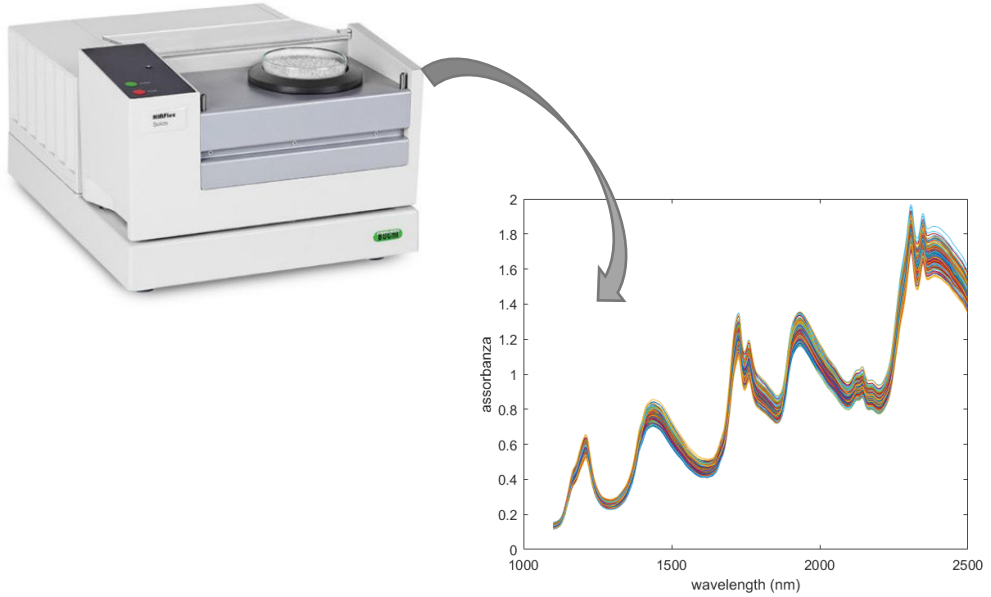
Example #5: NIR characterization of pig fat

Near Infra-Red (NIR) spectroscopy

- NIR spectroscopy is a powerful **analytical technology**
- Working principle:
  - *electromagnetic waves* (780-2500 nm) are absorbed/transmitted in different ways depending on the chemical composition of the analyzed samples(the molecular vibrations of the sample composition)
  - indirect measurement of the chemical composition

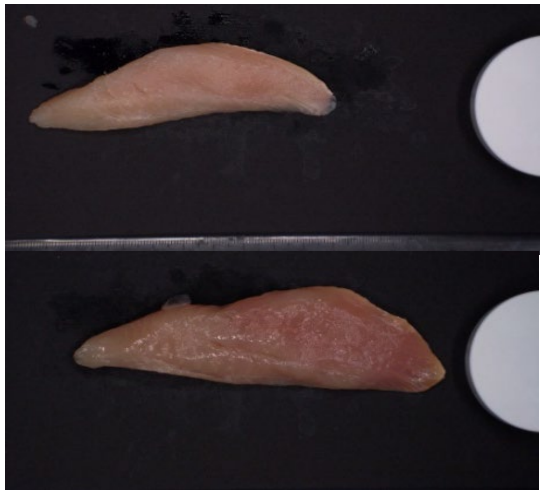
NIR spectrometer configuration





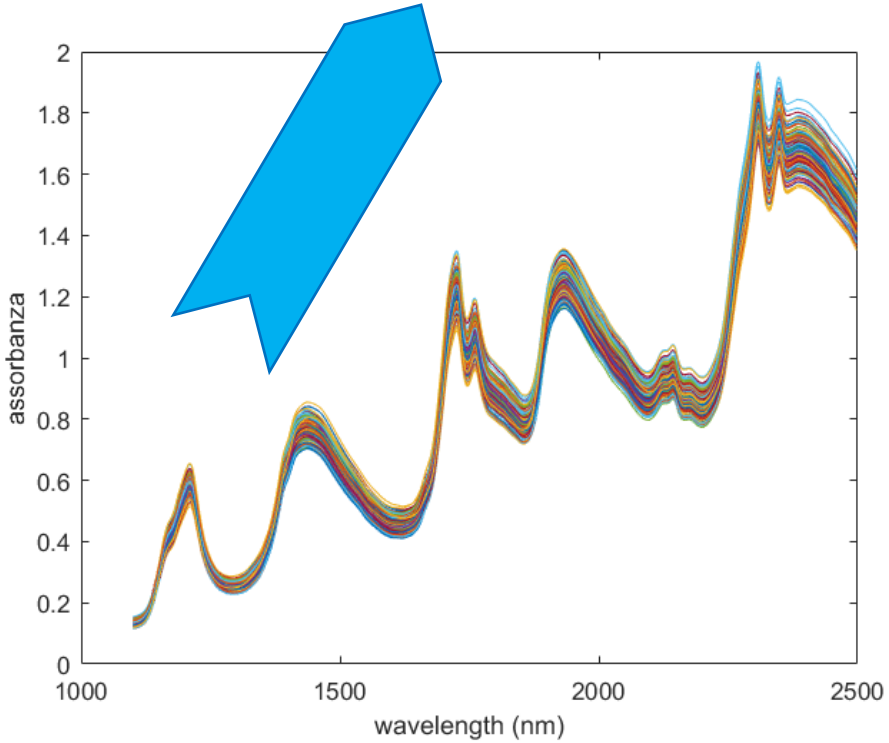
Food industry applications

- Applications to **food industry**
  - *fresh-thawed fish classification (mullets)*
  - *cheese origin classification (Asiago d'allevo cheese)*
  - *fish origin classification (goat fish, seabass, swordfish, etc…)*
  - *speck characterization (% fat, nutritional features, etc…)*
  - *sausage additives characterization*
  - *shellfish healthyness (cozze of the Venice lagoon)*



Pig fat characterization

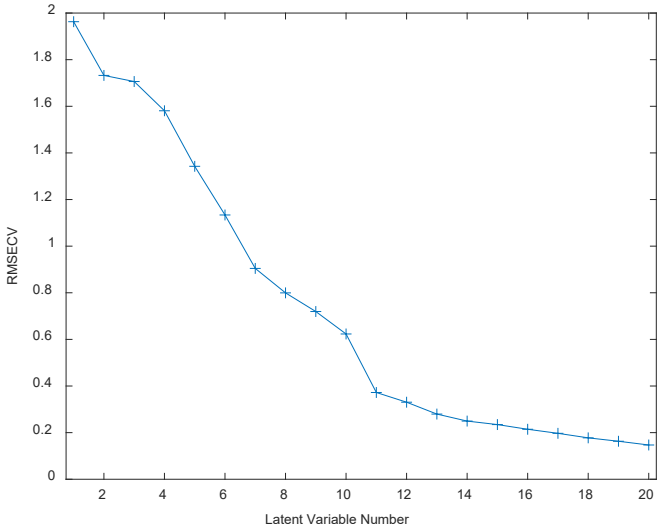
- NIR spectra of pig fat
- Objective: **iodine content estimation**
  - without expensive chemical analysis
- Load: `NIR_pigfat.mat`
- Data:
  - observations
    - 150 calibration samples of pig fat
    - 49 validation samples
  - **X** variables:
    - 700 wavelengths
    - NIR wavelengths:
      - 1000-2500 nm
  - **Y** variables:
    - iodine content

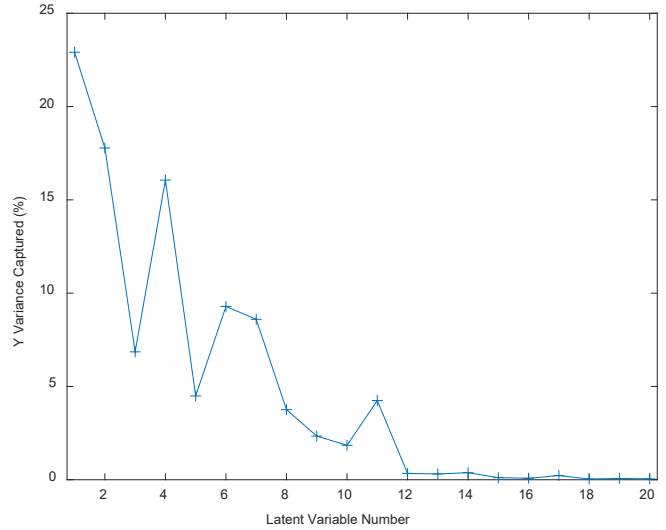


PLS modelling

- Pretreatment: autoscaling on **X**, autoscaling on **Y**
- Select 11 LVs
- Load calibration data: `Xcal`, `Y1cal`
- Load validation data `Xval`, `Y1val`

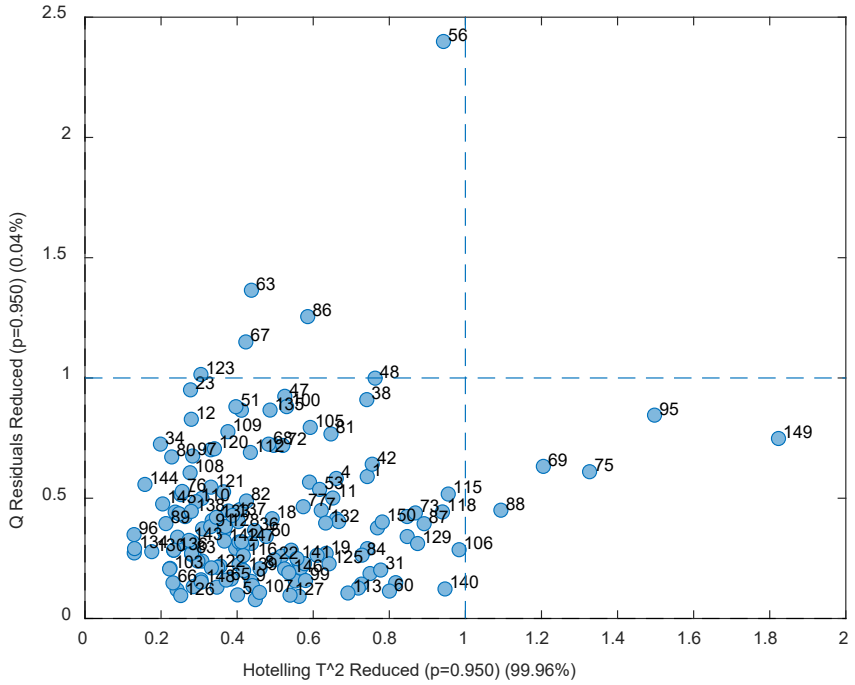
LV	X exp. var.	X cum. exp. var.	Y exp. var.	Y cum. exp. var.
1	73.11	73.11	22.91	22.91
2	22.29	95.4	17.78	40.7
3	0.94	96.34	6.85	47.55
4	0.62	96.95	16.06	63.61
5	2.34	99.3	4.49	68.1
6	0.33	99.63	9.29	77.39
7	0.12	99.74	8.6	85.99
8	0.17	99.91	3.76	89.75
9	0.02	99.94	2.34	92.09
10	0.02	99.96	1.85	93.93
11	0	99.96	4.25	98.18
12	0.01	99.97	0.34	98.52
13	0.01	99.98	0.31	98.83
14	0	99.98	0.38	99.21
15	0.01	99.99	0.11	99.32

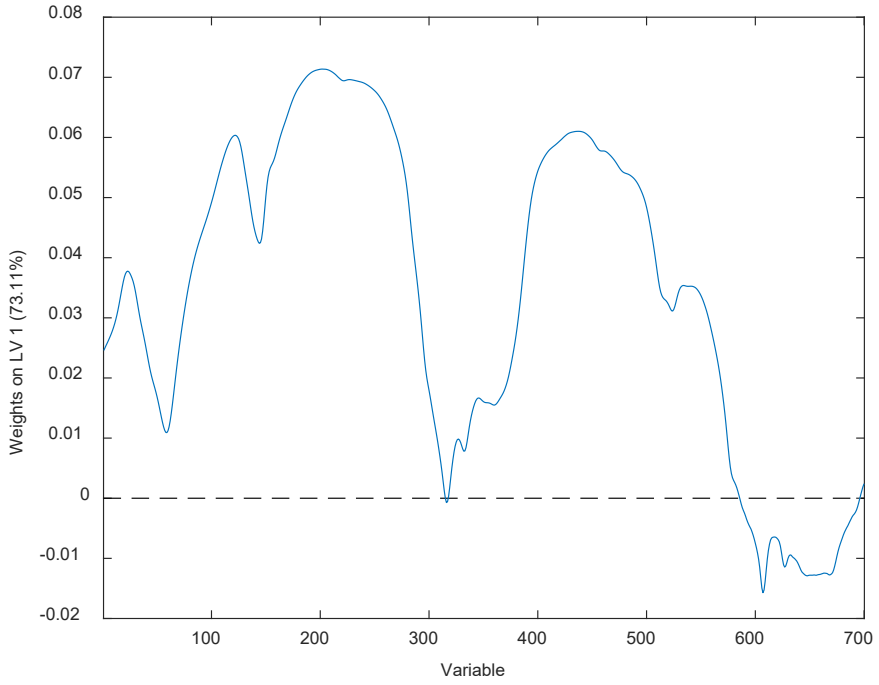




Inspection of scores, Q e T^2

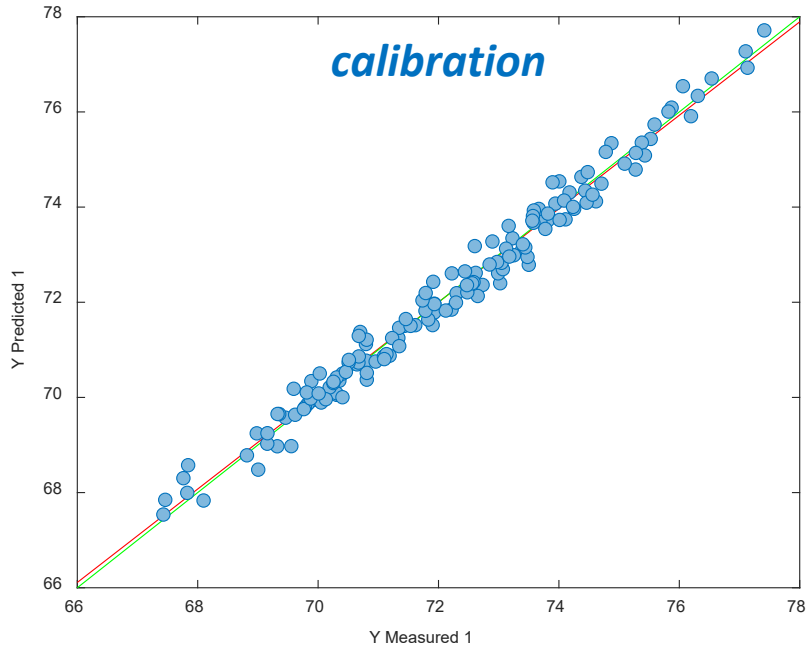
- The score plot allows detecting outliers, measurement errors, effects of different operators, instruments drifts, etc…
- The loadings/weights allow understanding what chemical structures are related to the food quality

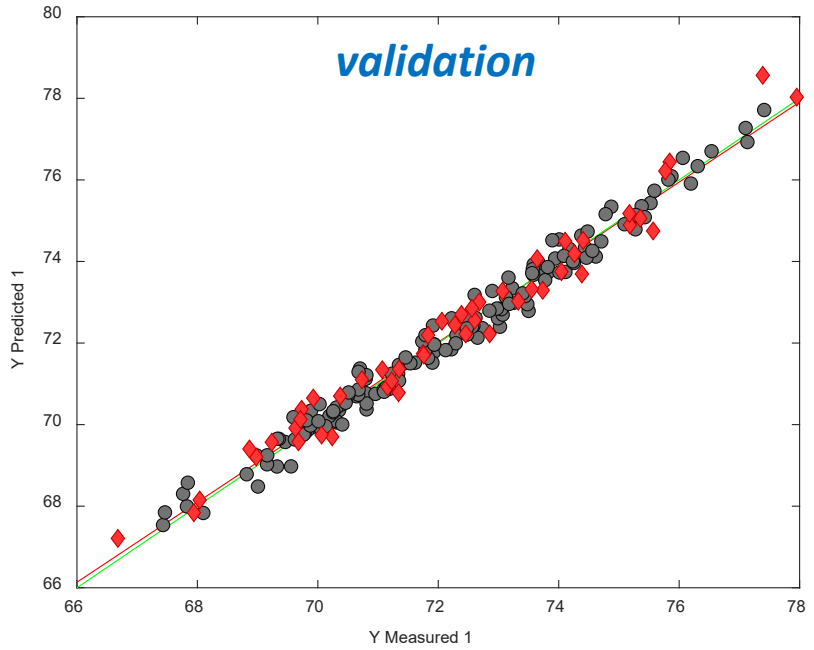




Estimation performance

- Very accurate estimations:
  - R^2 =0.9818 in calibration
  - R^2 = 0.9725 in validation
  - AAPE = 0.3382
  - MRE = 0.0047

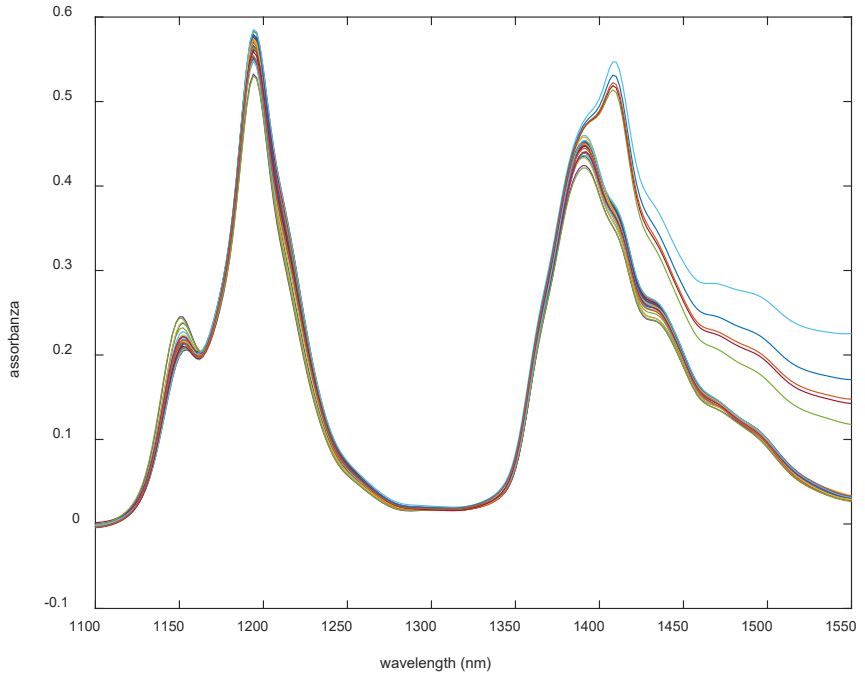




Example #6:
gasoline classification

NIR gasoline classification

- Gasoline classification based on octane number
  - the application requires an octane number >90
- Available data: **`gasoline_octane.mat`**
  - observation:
    - 30 calibrations gasolines
    - 9 validation gasolines
  - **X** variables:
    - 226 wavelengths
    - 1100-1550nm
  - **Y** variables:
    - dummy variable
      - 1 = octane number >90
      - 0 = octane number <90



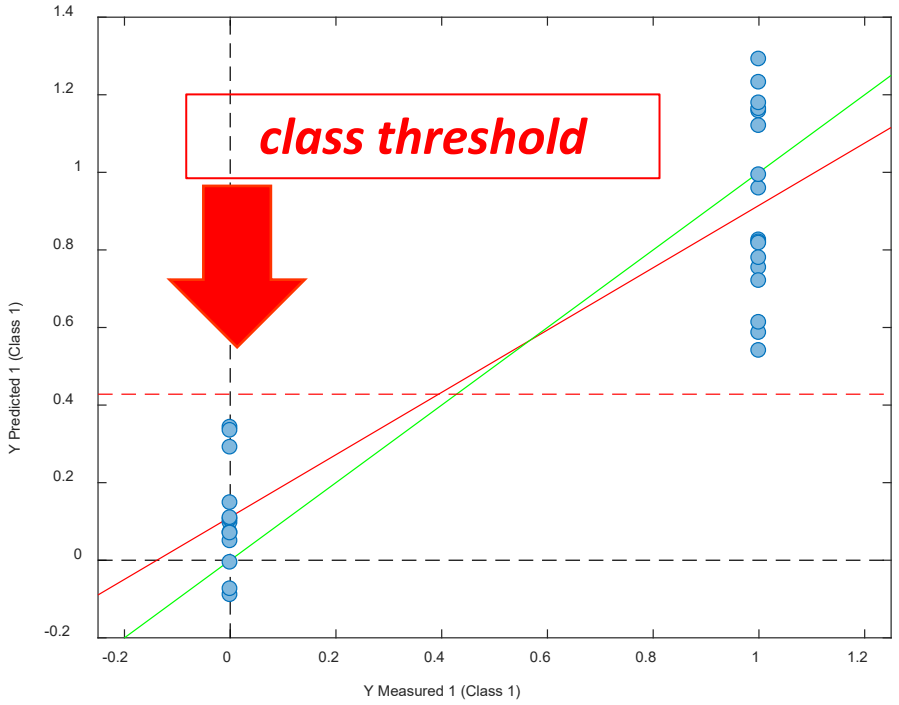
Esbensen, K.H., 2001. Multivariate Data Analysis in Practice, *Camo Process AS* (5th edition)

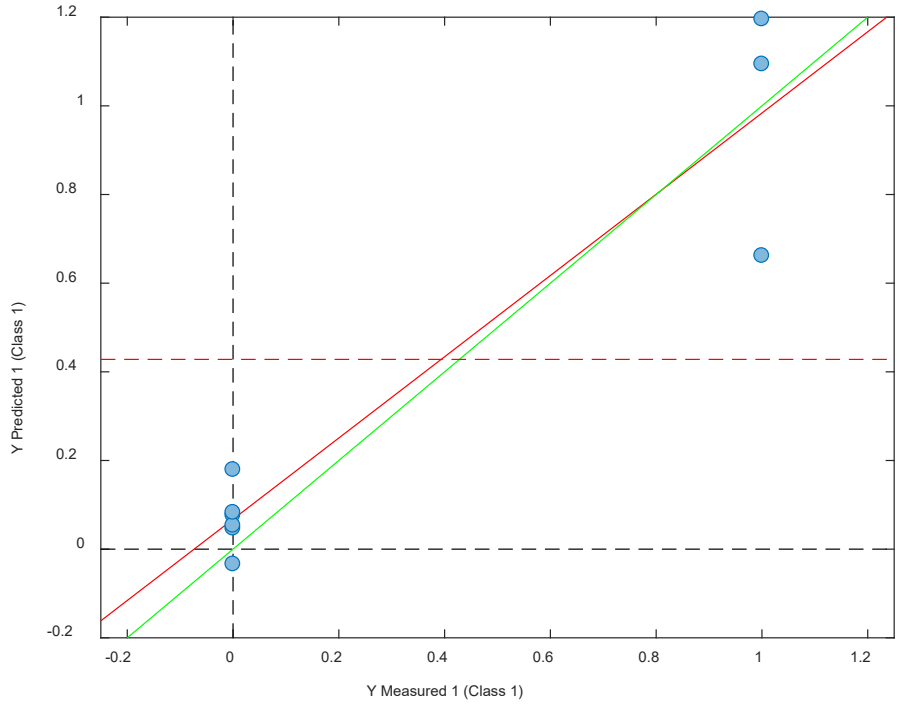
PLS-DA modeling

- PLS-discriminant analysis
  - «supervised» method for **classification**
- Pretreatment:
  - standard normal variate SNV (autoscaling on rows) + autoscaling for $\mathbf{X}$
  - autoscaling for $\mathbf{Y}$
- Number of selected LVs: 2
- Model calibration and validation:
  - calibration on: `Xcal`, `Y1cal`
  - validation on: `Xval`, `Y1val`

Classification performance

- **100% of correct classification** both on calibration and validation!
  - maximum sensitivity and specificity
  - the **threshold** among classes is fixed studying the class distribution dispersion





Take-home message

- PLS is a linear technique that is very effective on multivariate data for:
  - **estimation/prediction**
  - **supervised classification**
- The classification model calibration identifies the threshold among the classes
  - a class attribution probability can be assigned to each class
- Analyzing the relation between the $\mathbf{X}$ weights and the $\mathbf{Y}$ loadings one can understand what are the predictors which are most related to the class discrimination

PLSDA code (1/4)

```
close all
clear all
clc
% gasoline octane number NIR
load('C:\MachineLearningForProcessEngineering\dataset\gasoline_octane.mat')
% data visualization
figure;plot(Xcal');xlabel('wavelength');ylabel('gasoline NIR spectra');
% show y
figure;plot(Ycal);xlabel('observation');ylabel('class identifier');
% PLSDA model calibration
o.display='off';
o.plots='none';
o.preprocessing={'autoscale' 'autoscale'};
plsdam=plsda(Xcal,Ycal,11,o);
yp=plsdam.pred{1,2}(:,1);
plot(yp);
```

PLSDA code (2/4)

```
% show class attributions
i_1=find(yp>0.5);
i_0=find(yp<=0.5);


figure;
plot(-1:0.01:2,normpdf(-1:0.01:2,mean(yp(i_0,1)),std(yp(i_0,1))),'-b');
hold on;
plot(-1:0.01:2,normpdf(-1:0.01:2,mean(yp(i_1,1)),std(yp(i_1,1))),'--m');
hold off
figure;
plot(-1:0.01:2,1-normcdf(-1:0.01:2,mean(yp(i_0,1)),std(yp(i_0,1))),'-b');
hold on;
plot(-1:0.01:2,normcdf(-1:0.01:2,mean(yp(i_1,1)),std(yp(i_1,1))),'--m');
hold off
```

… per sempre a fianco a me!