Jennifer Dhont
Vrije Universiteit Brussel

How to start a machine learning project – a practical example

Goal

What? **Why?** **How?**

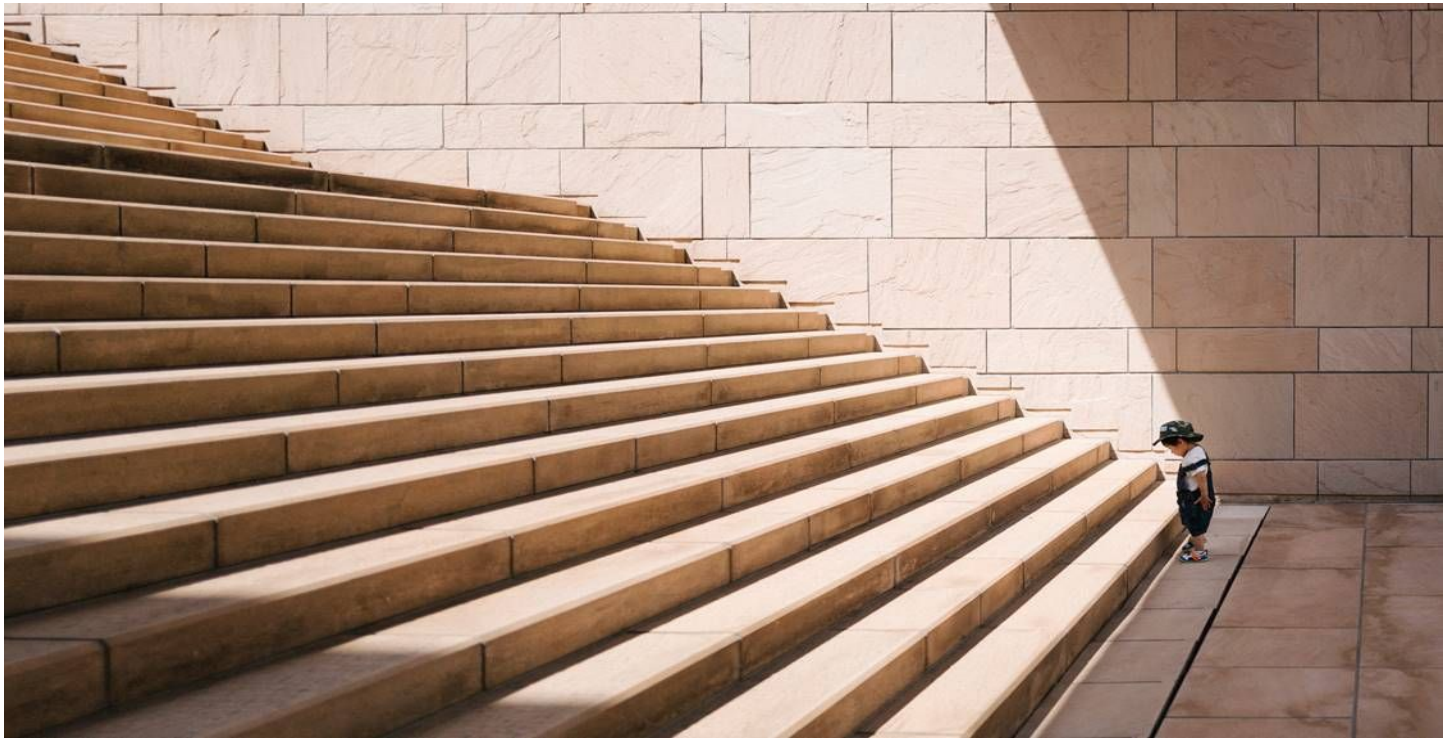
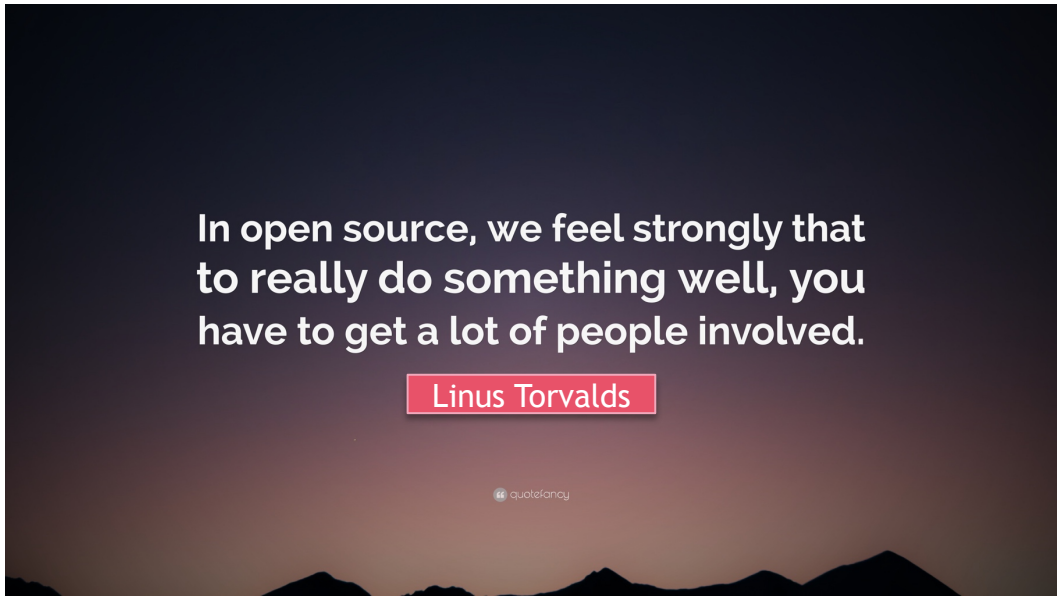


Image source: unknown

Conflicts of interest

None



Outline

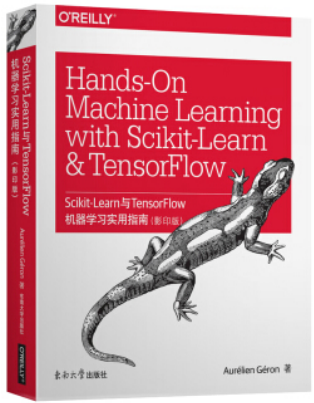
- End-to-end machine learning project
- Only one of many (many!) approaches…
- Snippets of code

Outline

```
# To support both python 2 and python 3
from __future__ import division, print_function, unicode_literals

# Common imports
import numpy as np
import os
import pandas as pd

# To make this notebook's output stable across multiple runs
np.random.seed(42)

# To plot pretty figures
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)
```

Outline

- End-to-end machine learning project
- Only one of many (many!) approaches…
- Snippets of code, available on…



github.com/JDhont/ESTRO38_premeeting

Python.org
Jupyter.org
Scikit-learn.org

An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 1: Frame the problem

In-house experience, confirmed by literature

4DCT significantly underestimates breathing-induced tumour motion during treatment

Margins Determined Using 4DCT Often Underestimate Tumor Motion in Thoracic Tumors

X. Shi, S. Chen, W.D. D'Souza, N.N. Mistry
University of Maryland School of Medicine, Baltimore, MD

The long- and short-term variability of breathing induced tumor motion in lung and liver over the course of a radiotherapy treatment

Jennifer Dhont [a,b,c,d,*], Jef Vandemeulebroucke [a,b], Manuela Burghelea [d], Kenneth Poels [e], Tom Depuydt [e], Robbe Van Den Begin [d], Cyril Jaudet [d], Christine Collen [d], Benedikt Engels [d], Truus Reynders [d], Marlies Boussaer [d], Thierry Gevaert [d], Mark De Ridder [d], Dirk Verellen [c,f]

Both four-dimensional computed tomography and four-dimensional cone beam computed tomography under-predict lung target motion during radiotherapy

Elisabeth Steiner[a,*], Chun-Chien Shieh[a], Vincent Caillet[a,b], Jeremy Booth[b,c], Ricky O'Brien[a], Adam Briggs[b], Nicholas Hardcastle[d,e], Dasantha Jayamanne[b], Kathryn Szymura[b], Thomas Eade[b,f], Paul Keall[a]

Implications using ITV technique?

An end-to-end machine learning project

Step 1: Frame the problem

HYPOTHESIS

Local tumour control probability of thoracic tumours is related to motion amplitude and type of motion management

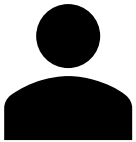
GOAL

***Two-fold** (1) To predict local recurrence, (2) evaluate relevance of motion amplitude and type of motion management.*

Machine learning

GOAL

Two-fold *(1) To predict local recurrence, (2) evaluate relevance of motion amplitude and type of motion management.*



Characteristics
(a,b,c,d,e,f,…)

MODEL
(w,p)

Local control
or
Local recurrence

An end-to-end machine learning project

Step 1: Frame the problem & look at the big picture

What type of problem is this?

1. Supervised, unsupervised, or reinforcement learning?
2. Is it a classification task or a regression task?

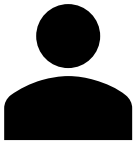
Supervised: fit a model on available input and output data

Classification: will this patient have a local recurrence or not?

Machine learning

Supervised learning

in a classification task



Characteristics
(a,b,c,d,e,f,…)

?
(w,p)

Local control
or
Local recurrence

An end-to-end machine learning project

Step 1: Frame the problem & look at the big picture

What type of problem is this?

1. Supervised, unsupervised, or reinforcement learning?
2. Is it a classification task or a regression task?
3. Performance measure?

An end-to-end machine learning project

Step 1: Frame the problem & look at the big picture

Performance measure

Regression task

- RMSE
- MAE

Classification task

- Sensitivity – % of patients with local recurrence identified as such
- Specificity - % of patients without local recurrence identified as such

An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 2: Feature engineering

PREDICT TCP

Model input?

Tumor biology, tumor microenvironment, radiation dosimetry, patient characteristics, imaging features…

“Coming up with features is difficult, time-consuming and requires expert knowledge.” - Andrew Ng

An end-to-end machine learning project

Step 2: Feature engineering - Ml vs. DL

Feature formulation

- Tumor biology
- Tumor microenvironment
- Radiation dosimetry
- Patient characteristics
- Imaging features…

Feature selection

What is available ???

➡ Storing raw data

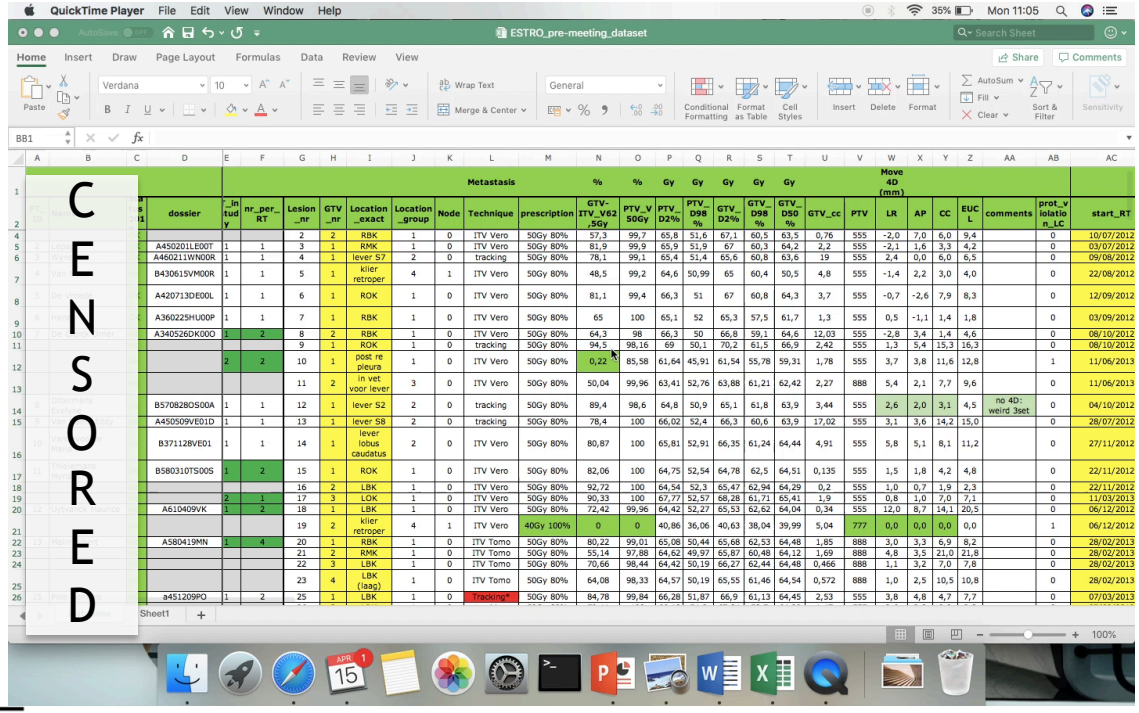
An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 3: Get the data



- Tumor characteristics
- 4DCT motion amplitude
- Treatment data
- Dose parameters
- Date of FU and local recurrence
- ~~Patient characteristics~~
- ~~Imaging data~~
- ~~Biological data~~

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse

1. GTV volume (cc)
2. Tumor location
3. Motion Management (ITV vs. tracking)
4. 4DCT motion amplitude (LR, AP, CC)
5. Six dose parameters (3 PTV, 3 GTV)
6. Start of treatment date
7. Dates of FU
8. Presence of local recurrence [0,1]

Input vs. Output
Features vs. Labels

Features
Categorical or continuous

Size ??
Total: 109, 28% LR [1]

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse

```
# To read the data from an Excel file and illustrate the first few rows;
# Change the location to the Excel file location on your PC!
data = pd.read_excel(r'/Users/jd/Documents/Work/ESTRO/ESTRO 2019/pre-meeting/ESTRO_premeeting_dataset.xlsx',target='LR.
data.head()
```

	Location_exact	Technique	PTV_V50Gy	PTV_D2%	PTV_D98%	GTV_D2%	GTV_D98%	GTV_D50%	GTV_cc	LR	AP	CC	EUCL	LR.1
0	LBK	Tracking	100.0	64.6	51.3	65.9	57.9	62.5	1.06	1.0	3.0	5.7	6.518435	0
1	RBK	ITV	99.7	65.8	51.6	67.1	60.5	63.5	0.76	-2.0	7.0	6.0	9.433981	0
2	RMK	ITV	99.9	65.9	51.9	67.0	60.3	64.2	2.20	-2.1	1.6	3.3	4.226109	1
3	ROK	ITV	99.4	66.3	51.0	67.0	60.8	64.3	3.70	-0.7	-2.6	7.9	8.346257	0
4	RBK	ITV	100.0	65.1	52.0	65.3	57.5	61.7	1.30	0.5	-1.1	1.4	1.849324	0

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse

```
# To show some info on the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 109 entries, 0 to 108
Data columns (total 14 columns):
Location_exact     109 non-null object
Technique          109 non-null object
PTV_V50Gy          109 non-null float64
PTV_D2%            109 non-null float64
PTV_D98%           109 non-null float64
GTV_D2%            109 non-null float64
GTV_D98%           109 non-null float64
GTV_D50%           109 non-null float64
GTV_cc             109 non-null float64
LR                 109 non-null float64
AP                 109 non-null float64
CC                 109 non-null float64
EUCL               109 non-null float64
LR.1               109 non-null int64
dtypes: float64(11), int64(1), object(2)
memory usage: 12.0+ KB
```

Categorical features

Numerical features

Label

An end-to-end machine learning project

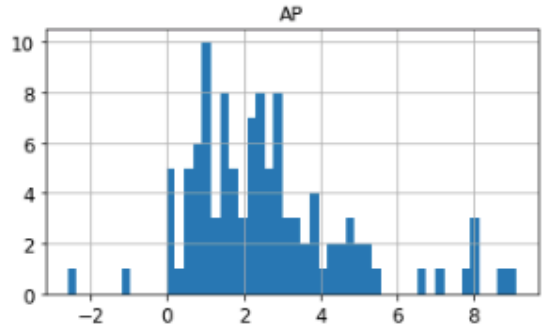
Step 3: Get the data... and have a first glimpse

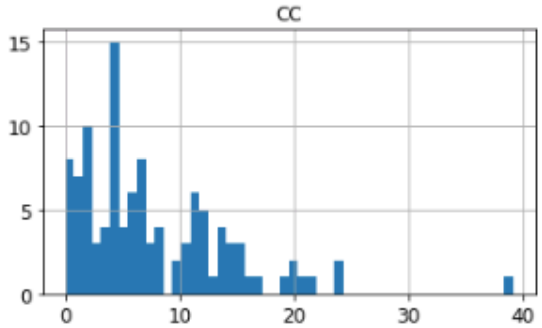
```
# To give some statistics on the numerical features
data.describe()
```

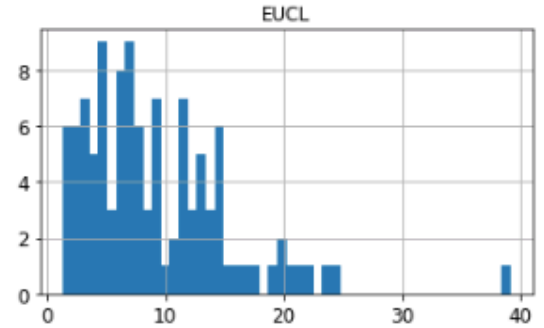
	PTV_V50Gy	PTV_D2%	PTV_D98%	GTV_D2%	GTV_D98%	GTV_D50%	GTV_cc	LR	AP	CC	EUCL	LR.1
count	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000	109.000000
mean	98.774587	65.781284	51.757064	66.475321	60.640826	64.188624	4.216156	1.781284	2.610000	7.665688	9.047311	0.293578
std	4.420853	1.933504	1.934911	3.192854	4.173766	2.921886	7.219889	1.976544	2.098946	6.522449	6.134108	0.457504
min	65.760000	54.570000	43.110000	43.120000	31.210000	41.590000	0.100000	-2.800000	-2.600000	0.000000	1.277576	0.000000
25%	99.390000	64.800000	51.130000	65.700000	60.230000	63.850000	0.600000	0.800000	1.100000	3.100000	4.736032	0.000000
50%	99.870000	66.140000	51.820000	67.000000	61.210000	64.550000	1.300000	1.300000	2.300000	5.900000	7.502666	0.000000
75%	99.990000	66.980000	52.590000	67.980000	62.450000	65.240000	3.700000	2.600000	3.400000	11.600000	12.287392	1.000000
max	100.000000	69.000000	56.180000	71.110000	65.580000	67.600000	31.590000	12.000000	9.100000	39.100000	39.141538	1.000000

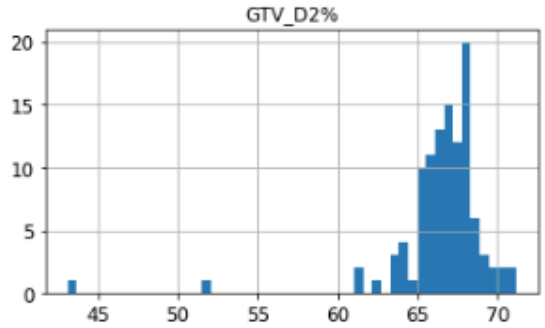
An end-to-end machine learning project

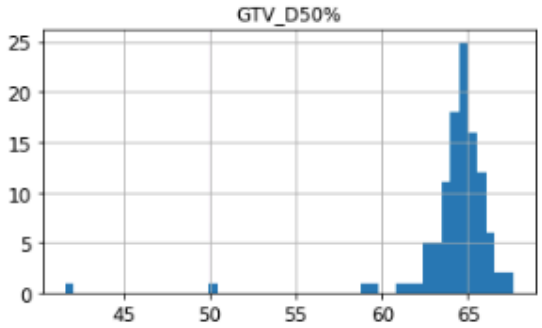
```
# To plot the histograms of all numerical features
%matplotlib inline
data.hist(bins=50, figsize=(20,15))
plt.show()
```

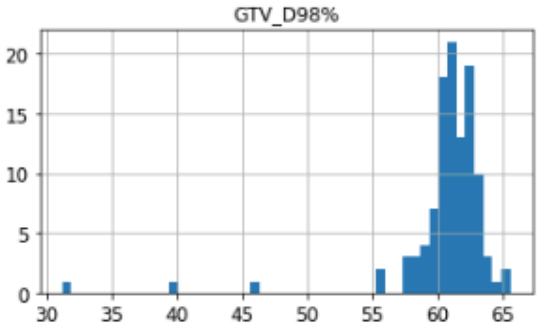












An end-to-end machine learning project

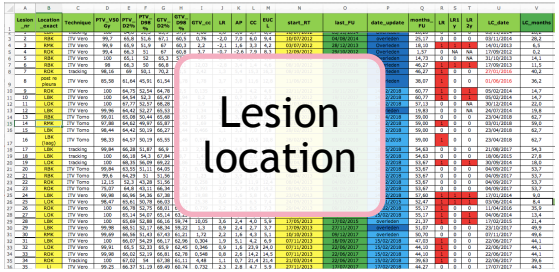
Step 3: Get the data... and have a first glimpse

```
patient_data["Location_exact"].value_counts()
```

```
LBK                                    24
RBK                                    17
LOK                                    15
ROK                                    14
RMK                                     5
LOK ant                                 2
LBK medial                              2
Li apex                                 1
LBK (laag)                              1
ROK, craniaal                           1
post re pleura                          1
Re                                      1
Li                                      1
RMK perifeer                            1
LBK sup                                 1
LOk                                     1
 LBK lateral                            1
ROK cranial                             1
LBK caudal                              1
li hilus                                1
ROK, anterieur                          1
Lingula                                 1
LOK top (cave: LBK was al gereseceerd)  1
LBK inf                                 1
```

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse



Preliminary pre-processing

RBK
RMK
ROK
LBK
LOK

Post re pleura
LBK (laag)
Li
Re
ROK, caudaal
ROK caudal
ant ROK
LBK medial
LBK ant

ROK, craniaal
ROK cranial
ROK, anterieur
RBK ant
RBK med
RBK post
LBK cranial
RE LONG

RMK perifeer
LOK top (...)
LBK inf
LBK sup
LBK caudal
RMK pleura
LBK lateral
LONG

1 dr. dataset

Lingula
Li apex
LOK ant

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse

Features

Lesion location	Treatment technique	GTV_D50%	GTV volume
AP motion	LR motion	CC motion	GTV_D2%
PTV V50Gy	PTV D2%	PTV D98%	GTV_D98%

Labels

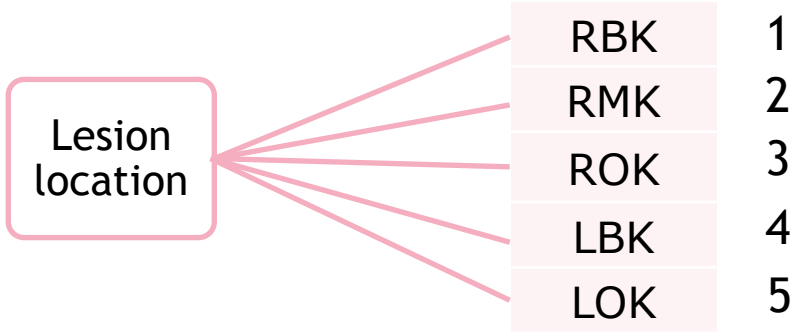
Local control at 6 months

Preliminary pre-processing

- Standardization of continuous features
- Change categorical to binary

An end-to-end machine learning project

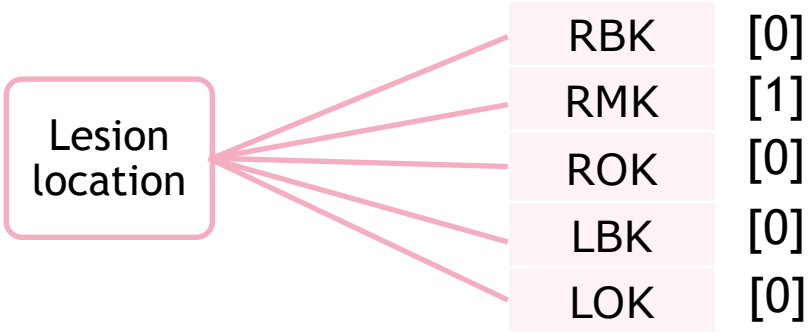
Step 3: Get the data... and have a first glimpse



Nominal vs. **Ordinal** features

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse



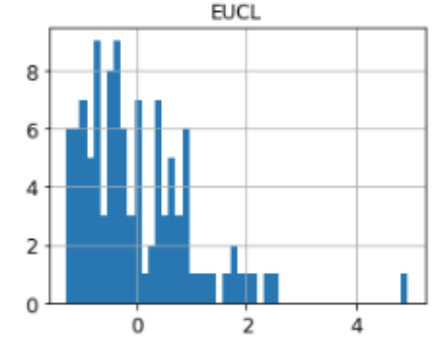
Dimensionality increase!

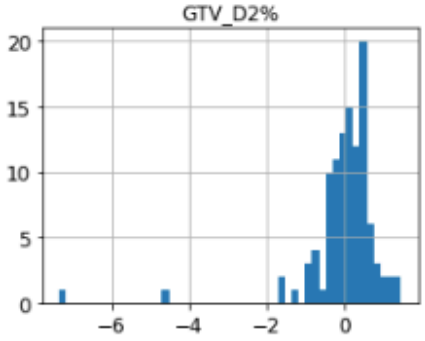
An end-to-end machine learning project

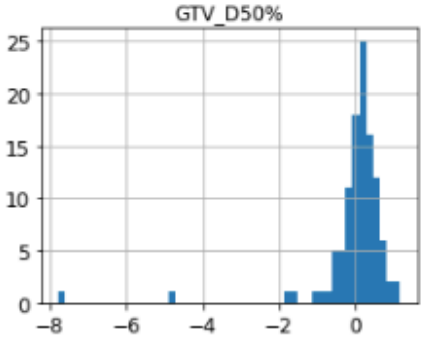
```
# To drop irrelevant features and change categorical features to binary features
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler

onehot = OneHotEncoder(dtype=np.int, sparse=True)

nominals = pd.DataFrame(
    onehot.fit_transform(data[['Location_exact', 'Technique']])\
    .toarray(),
    columns=['LBK', 'LOK', 'RBK', 'RMK','ROK','ITV', 'tracking'])

data_encoded = data.drop(["Location_exact","Technique","AP","LR","CC","PTV_V50Gy"],axis=1)
data_encoded = nominals.join(data_encoded)

# To standardize all continuous features
scaler = StandardScaler()
data_encoded["PTV_D98%"] = scaler.fit_transform(data_encoded["PTV_D98%"].values.reshape(-1,1))
data_encoded["PTV_D2%"] = scaler.fit_transform(data_encoded["PTV_D2%"].values.reshape(-1,1))
data_encoded["GTV_D2%"] = scaler.fit_transform(data_encoded["GTV_D2%"].values.reshape(-1,1))
data_encoded["GTV_D98%"] = scaler.fit_transform(data_encoded["GTV_D98%"].values.reshape(-1,1))
data_encoded["GTV_D50%"] = scaler.fit_transform(data_encoded["GTV_D50%"].values.reshape(-1,1))
data_encoded["GTV_cc"] = scaler.fit_transform(data_encoded["GTV_cc"].values.reshape(-1,1))
data_encoded["EUCL"] = scaler.fit_transform(data_encoded["EUCL"].values.reshape(-1,1))

%matplotlib inline
data_encoded.hist(bins=50, figsize=(20,15))
plt.show()
```

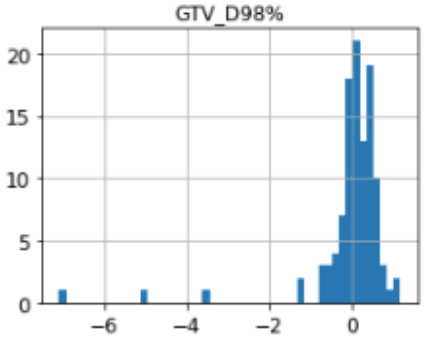
An end-to-end machine learning project

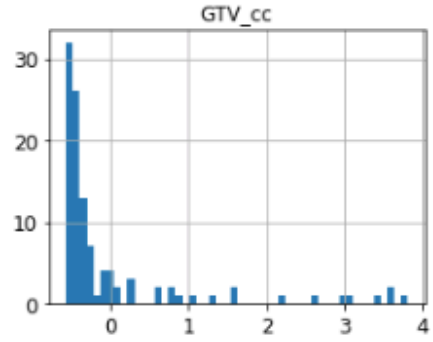
Step 3: Explore the data to get insights

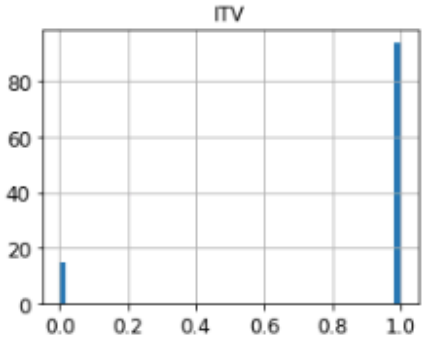


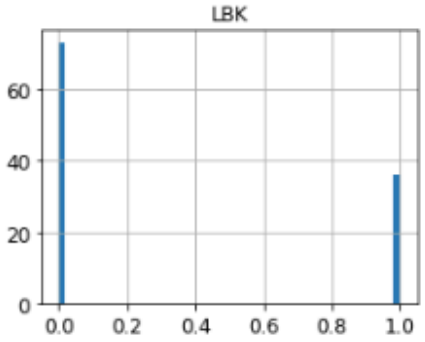


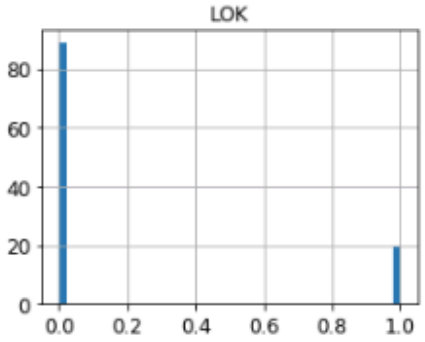












An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse

Features

Lesion location	Treatment technique	GTV_D50%	GTV volume
AP motion	LR motion	CC motion	GTV_D2%
PTV V50Gy	PTV D2%	PTV D98%	GTV_D98%

Labels

Local control at 6 months

Preliminary pre-processing

- Standardization of continuous features
- Change categorical to binary

An end-to-end machine learning project

Step 3: Get the data... and have a first glimpse

Features

Tracking	ITV	GTV volume	LBK
LR motion	AP motion	CC motion	LOK
GTV_D2%	GTV_D98%	GTV_D50%	ROK
PTV D98%	PTV D2%	PTV V50Gy	RMK

Labels

Local control at 6 months

RBK

An end-to-end machine learning project

Some take-home messages on data

Your application will only ever be as good as your data.

With machine learning, you assume nothing a priori, you learn from the data.

→ Data from old treatment techniques will not be accurate now.

An end-to-end machine learning project

Some take-home messages on data

A matter of quality and quantity

The unreasonable effectiveness of data[1].

"Significant benefit of training data available in the wild"

Small and medium-sized datasets are still common…

So don't abandon algorithms just yet...

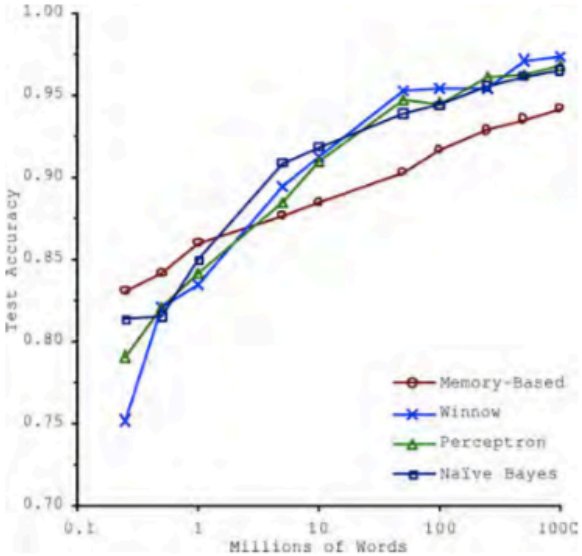


Figure 1-20. The importance of data versus algorithms[9]

[1] "The Unreasonable Effectiveness of Data," Peter Norvig et al. (2009).

An end-to-end machine learning project

OUTLINE

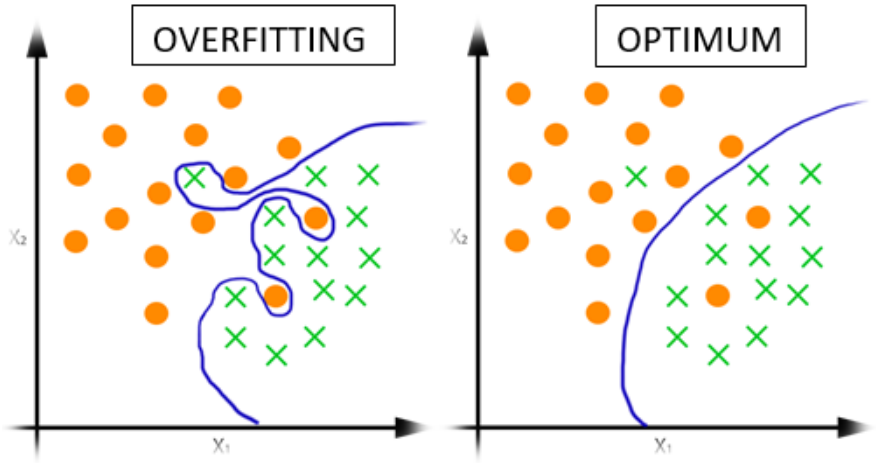
1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 4: Creation of a training, (validation) and test set

GOAL in ML, DL, …

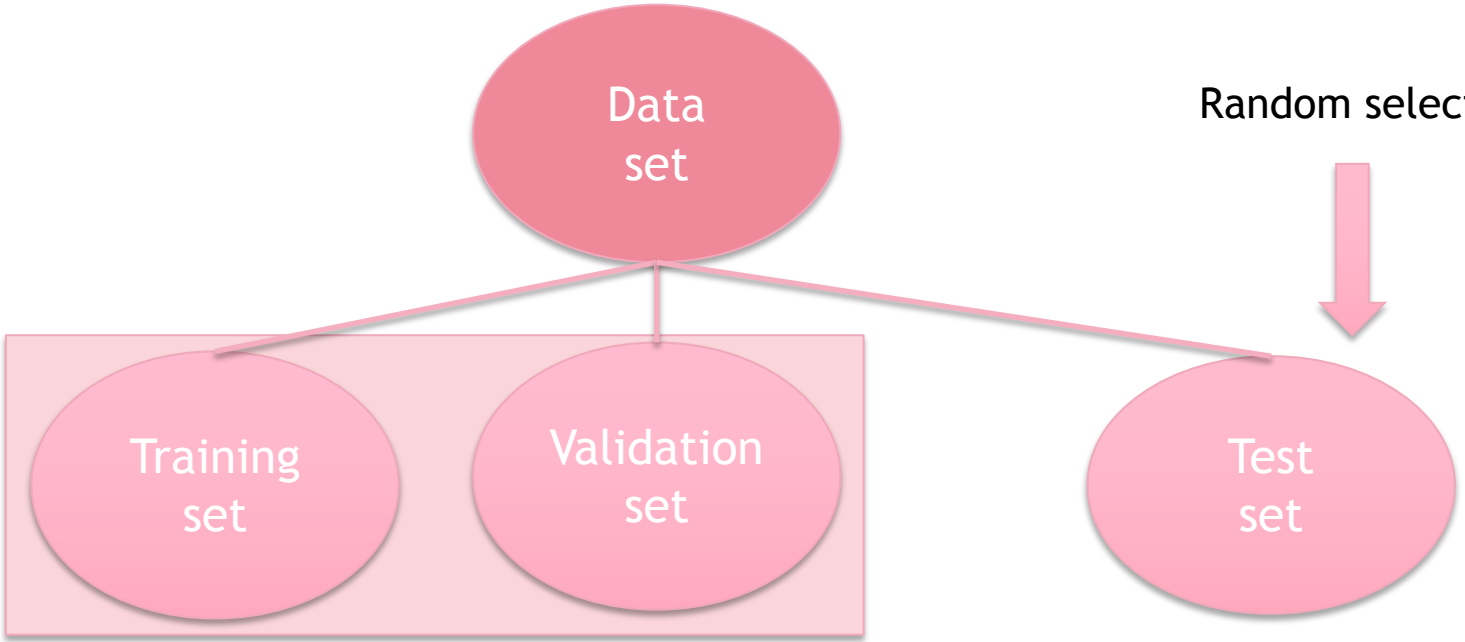
Model generalization vs. Overfitting



Source: Overfitting and human behaviour, Medium (2018)

An end-to-end machine learning project

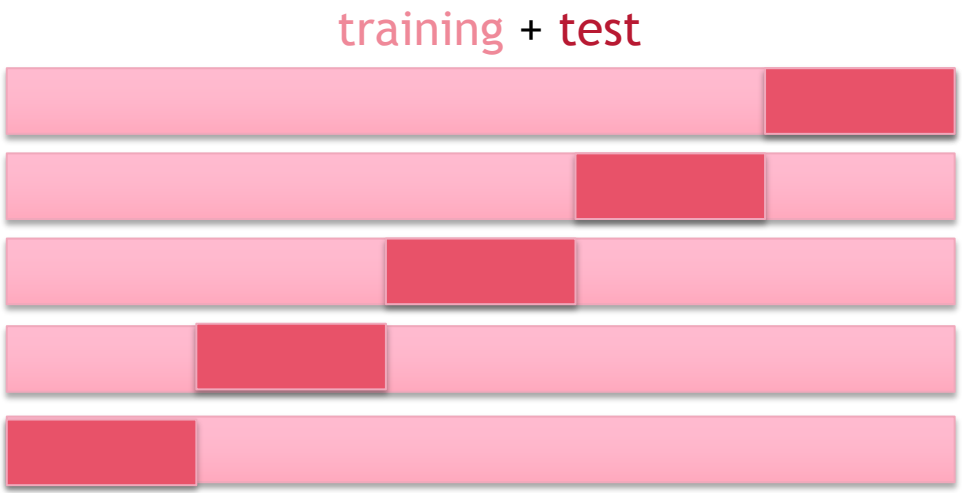
Step 4: Creation of a training, (validation) and test set



An end-to-end machine learning project

Step 4: Creation of a training, (validation) and test set

In case of smaller datasets; n-fold cross-validation



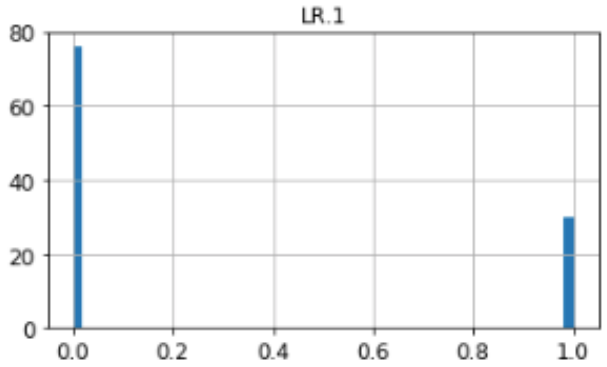
An end-to-end machine learning project

Step 4: Creation of a training, (validation) and test set

"Random sampler" - 80% training set

Large datasets: Train & test sets will represent population distributions

Data imbalance




```
0     77
1     32
Name: LR.1, dtype: int64
```

Sensitivity – % of patients with local recurrence identified as such

Specificity - % of patients without local recurrence identified as such

An end-to-end machine learning project

Step 4: Creation of a training, (validation) and test set

- Undersampling

  65 - 15 → 15 - 15

- Synthetic minority oversampling

Small datasets: Imbalanced AND not representative

```
# To show number of "Technique" categories in the dataset
data["Technique"].value_counts()
```

```
ITV         94
Tracking    15
Name: Technique, dtype: int64
```

imbalanced-learn.org

+ stratified sampling *(= population representation)*

An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 5: Feature selection

Feature formulation

- Tumor biology
- Tumor microenvironment
- Radiation dosimetry
- Patient characteristics
- Imaging features…

→ Storing raw data

Feature extraction

- Patient records
- Pathology
- Planning software
- Radiology…

→ Consistency

Feature selection

→ Relevant feature selection

- The curse of dimensionality (data sparsity)
- Reduces chance of overfitting
- Improved model accuracy,
- Decreases training time /complexity
- Better understanding

An end-to-end machine learning project

Step 5: Feature selection

Curse of dimensionality

Multiple representations of every option

Ex. 2 binary features (gender, cancer diagnosis)

4 combinations, 40 training samples

Ex. 5 binary features (gender, cancer diagnosis, ….)

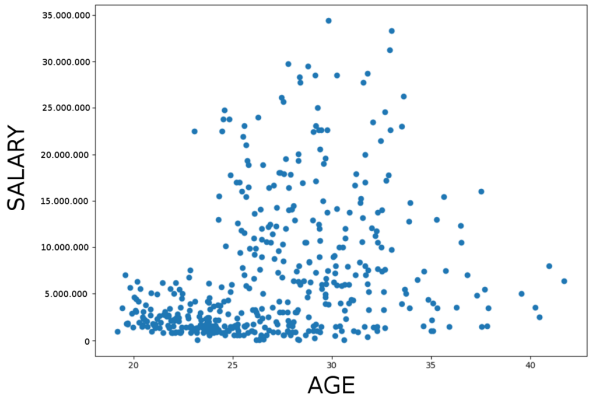
32 combinations, 320 training samples

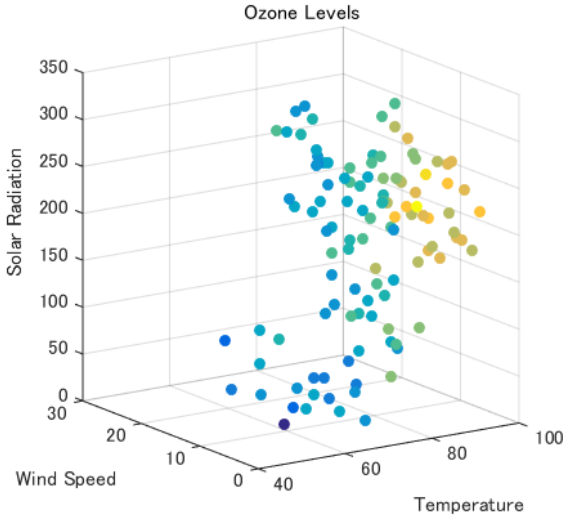
An end-to-end machine learning project

Step 5: Feature selection

Curse of dimensionality

Multiple representations of every option





An end-to-end machine learning project

Step 5: Feature selection

Feature formulation

- Tumor biology
- Tumor microenvironment
- Radiation dosimetry
- Patient characteristics
- Imaging features…

→ Storing raw data

Feature extraction

- Patient records
- Pathology
- Planning software
- Radiology…

→ Consistency

Feature selection

→ Relevant feature selection

- The curse of dimensionality (data sparsity)
- Reduces chance of overfitting
- Improved model accuracy
- Decreases training time /complexity
- Better understanding

An end-to-end machine learning project

Step 5: Feature selection **How?**

A matter of balance, enough but not too much

Evaluate all possible feature combinations…
but nobody has time for that, so a suboptimal methodology has to be applied.

1. Start with all of them.
2. *Look for redundant features*
3. *Dimensionality reduction*
4. *Automated selection methods*

[1] A survey on feature selection methods. Girish Chandrashekar, et al. (2014)

An end-to-end machine learning project

Step 5: Feature selection

How?

1. Start with all of them.
2. ***Look for redundant features***
3. *Dimensionality reduction*
4. *Automated selection methods*

Features

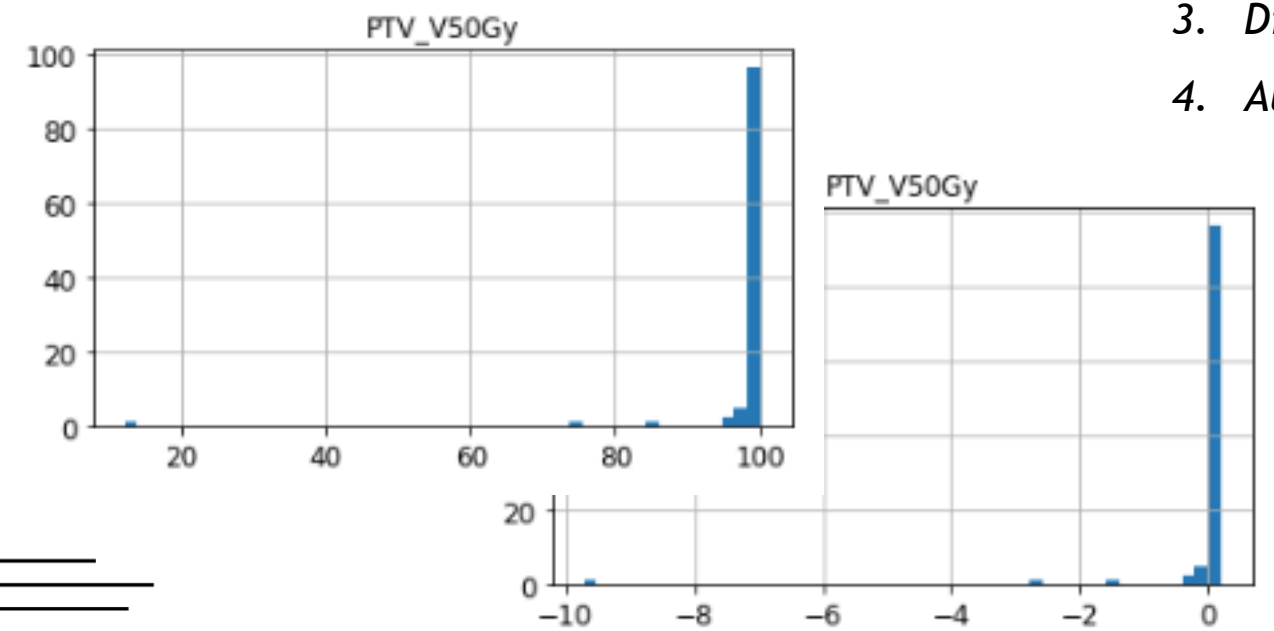
Tracking	ITV	GTV volume	LBK	
LR motion	AP motion	CC motion	LOK	
GTV_D2%	GTV_D98%	GTV_D50%	ROK	
PTV D98%	PTV D2%	PTV V50Gy	RMK	RBK

An end-to-end machine learning project

Step 5: Feature selection **How?**

1. Start
2. *Look* *tures*
3. *Dimer* *n*
4. *Auton* *thods*

PTV V50Gy



	PTV_V50Gy
count	106.000000
mean	98.352358
std	8.915657
min	12.150000
25%	99.382500
50%	99.860000
75%	99.980000
max	100.000000

An end-to-end machine learning project

Step 5: Feature selection

How?

1. Start with all of them.
2. ***Look for redundant features***
3. *Dimensionality reduction*
4. *Automated selection methods*

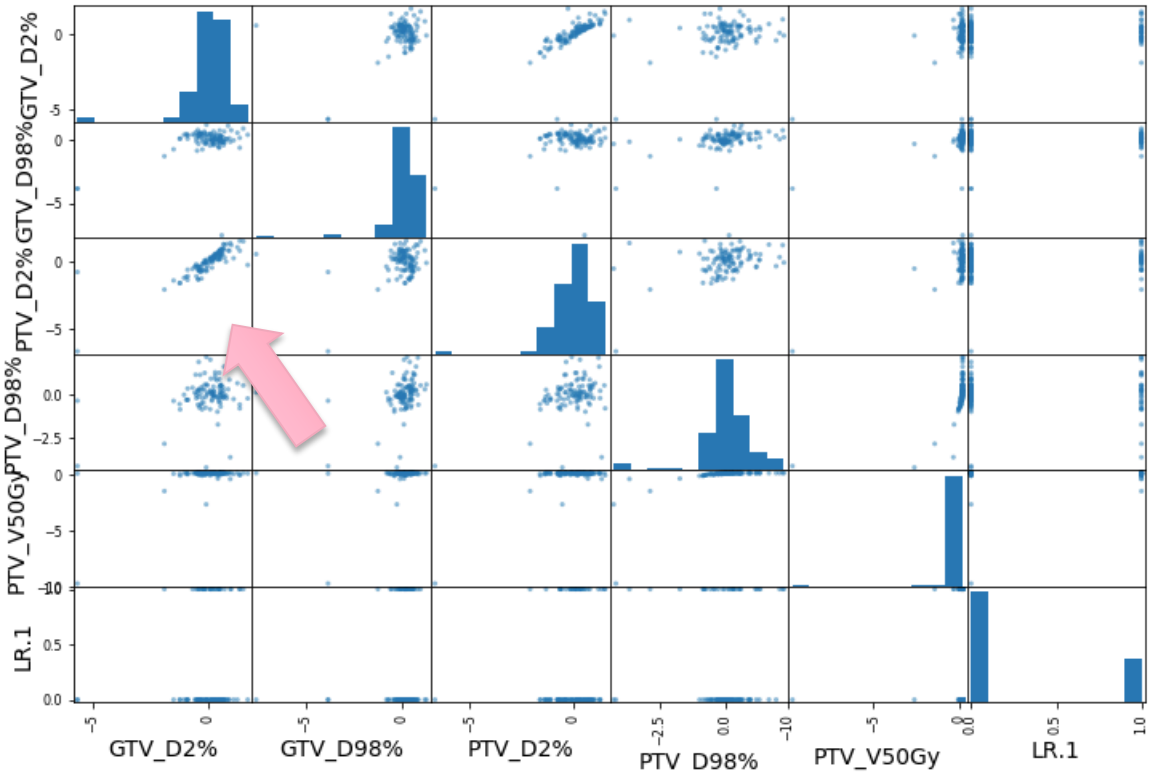
Features

Tracking	ITV	GTV volume	LBK	
LR motion	AP motion	CC motion	LOK	
GTV_D2%	GTV_D98%	GTV_D50%	ROK	
PTV D98%	PTV D2%	PTV V50Gy	RMK	RBK

An end-to-end machine learning project

Step 5: Feature selection

How?



of them.

undant features

ty reduction

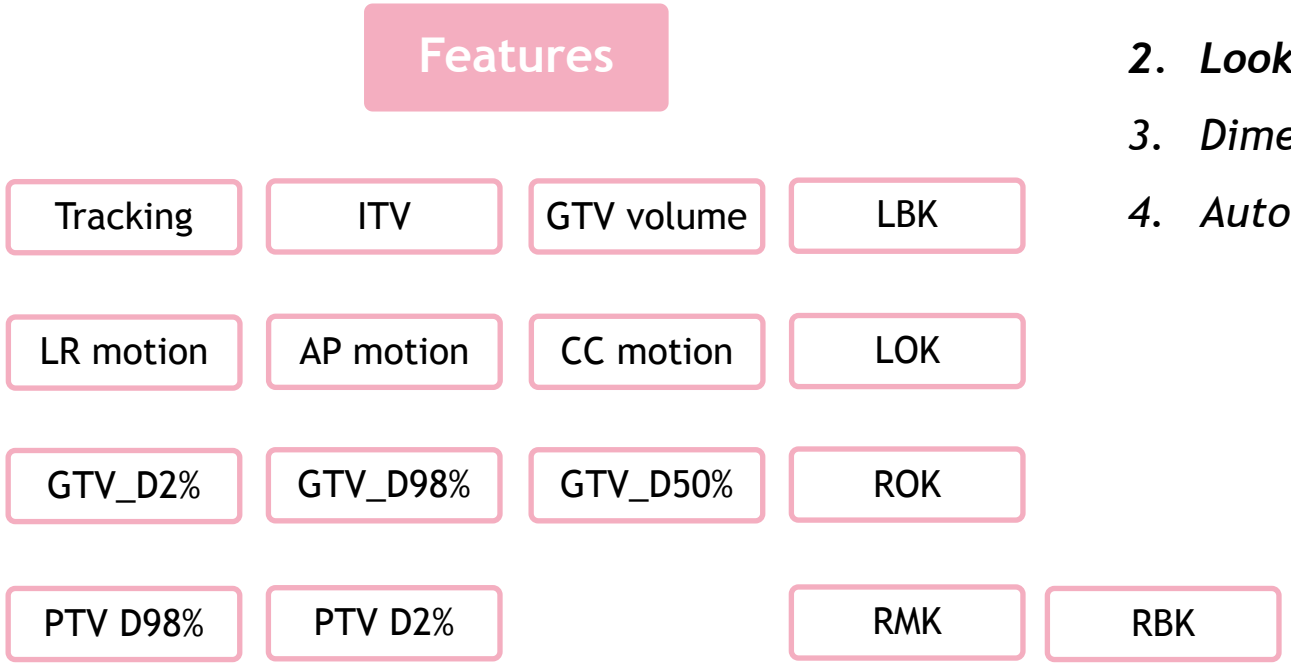
election methods

An end-to-end machine learning project

Step 5: Feature selection

How?

1. Start with all of them.
2. ***Look for redundant features***
3. *Dimensionality reduction*
4. *Automated selection methods*



An end-to-end machine learning project

Step 5: Feature selection

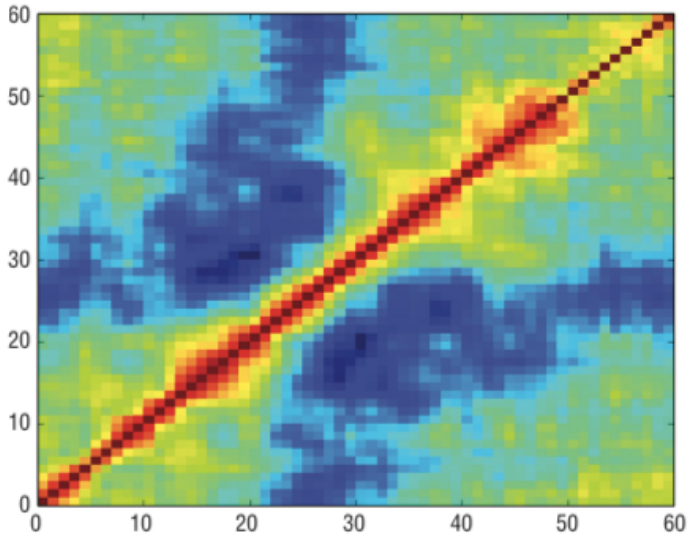


Figure 2-8: Heat map showing attribute cross-correlations

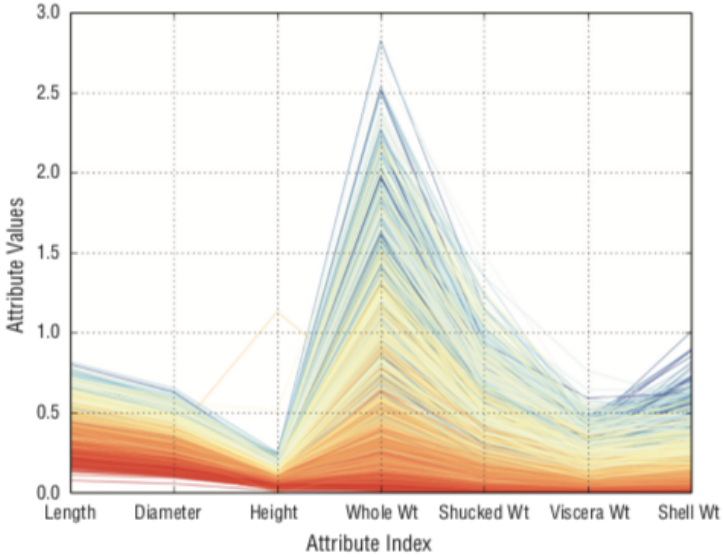


Figure 2-12: Color-coded parallel coordinate plot for abalone

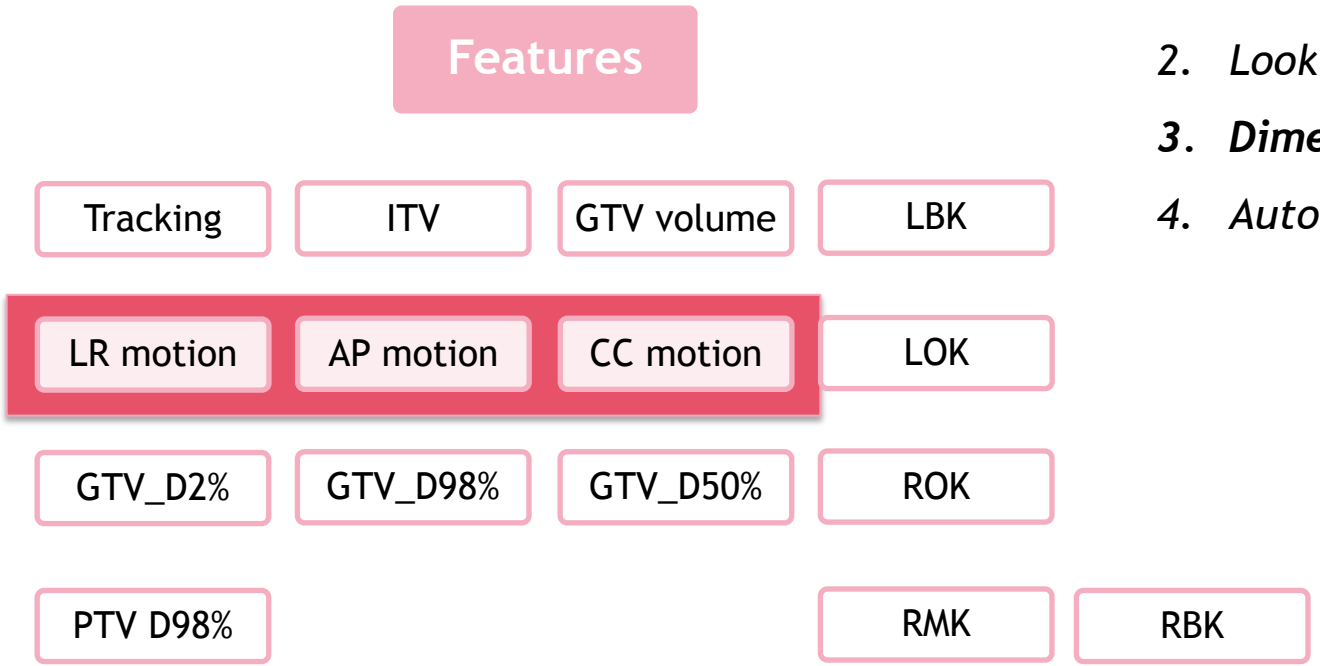
Source: Machine learning in Python, Michael Bowles (2015)

An end-to-end machine learning project

Step 5: Feature selection

How?

1. Start with all of them.
2. *Look for redundant features*
3. ***Dimensionality reduction***
4. *Automated selection methods*

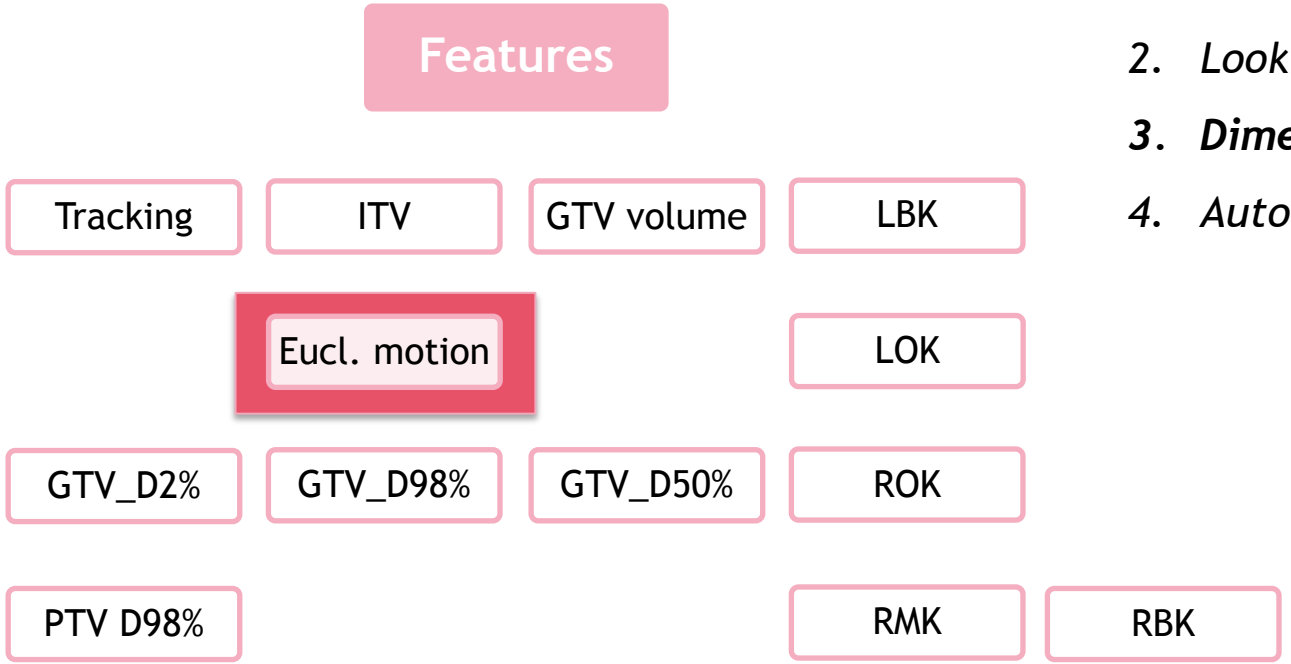


An end-to-end machine learning project

Step 5: Feature selection

How?

1. Start with all of them.
2. *Look for redundant features*
3. ***Dimensionality reduction***
4. *Automated selection methods*



An end-to-end machine learning project

Step 5: Feature selection

4. Automated selection/illimination methods

Filter

Features are ranked according to relevance.

Pearson correlation (linear), Mutual information (MI), …

Only highest ranked features are selected.

- univariate

Wrapper

Search algorithm finds subset of features with highest predictor performance.

Objective function steared maximum performance with minimum of features.

Some: backtracking
take into account correlation between features.

Embedded

Embedded in the training process of the model.

- Artificial neural networks

An end-to-end machine learning project

Step 5: Feature selection

Filter

Wrapper

Cross-validation recursive feature selection on the training set

- Takes into account collinearity
- Only works on linear models

```python
# To perform automatic cross-validation recursive feature selection on the training set

import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import StratifiedKFold
from sklearn.feature_selection import RFECV
from sklearn.datasets import make_classification

# Create the RFE object and compute a cross-validated score.
svc = SVC(kernel="linear")
# The "accuracy" scoring is proportional to the number of correct
# classifications
rfecv = RFECV(estimator=svc, step=1, cv=StratifiedKFold(5),
              scoring='accuracy')
rfecv.fit(train_features, train_labels)

print("Optimal number of features : %d" % rfecv.n_features_)
print(rfecv.support_)

# Plot number of features VS. cross-validation scores
plt.figure()
plt.xlabel("Number of features selected")
plt.ylabel("Cross validation score (nb of correct classifications)")
plt.plot(range(1, len(rfecv.grid_scores_) + 1), rfecv.grid_scores_)
plt.show()
```

An end-to-end machine learning project

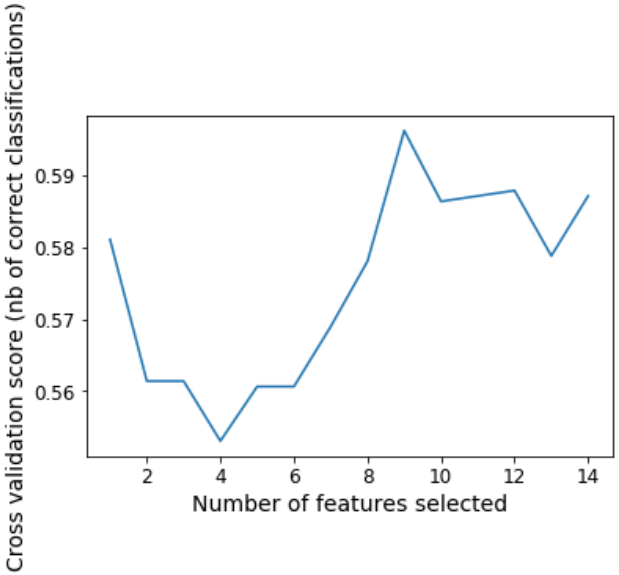
Step 5: Feature selection

Filter

Wrapper

Cross-validation recursive feature selection on the training set

```
Optimal number of features : 9
[ True  True  True  True  True False False False False  True  True  True
  True False]
```



- Takes into account collinearity
- Only works on linear models

An end-to-end machine learning project

Artificial dataset

```
# To show some info on the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 450 entries, 0 to 449
Data columns (total 13 columns):
Location_exact    450 non-null object
Technique         450 non-null object
PTV_V50Gy         450 non-null float64
PTV_D2%           450 non-null float64
PTV_D98%          450 non-null float64
GTV_D2%           450 non-null float64
GTV_D98%          450 non-null float64
GTV_cc            450 non-null float64
LR                450 non-null float64
AP                450 non-null float64
CC                450 non-null float64
EUCL              450 non-null float64
LR.1              450 non-null int64
dtypes: float64(10), int64(1), object(2)
memory usage: 45.8+ KB
```

May or may not have imposed some correlations

An end-to-end machine learning project

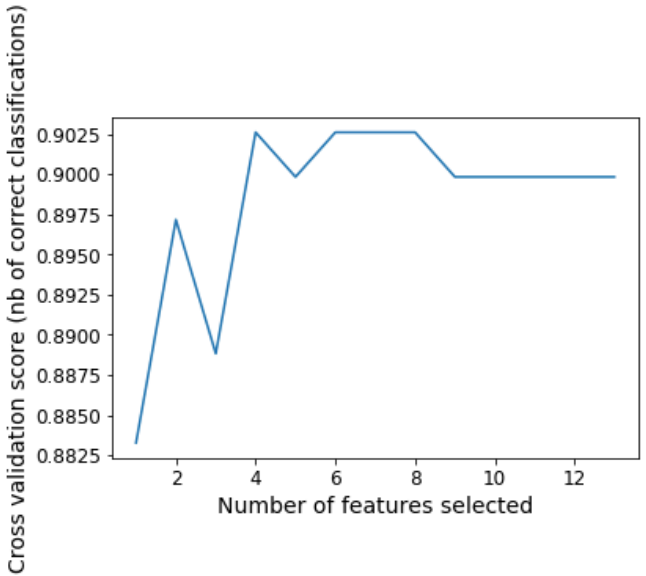
Step 5: Feature selection

Filter

Wrapper

Cross-validation recursive feature selection on the training set

```
Optimal number of features : 4
[False False  True False False False False False False  True False  True
  True]
```



- Euclidean motion
- GTV volume
- GTV_D2%

An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 6: Model selection

- Explore many models and short-list the best once.

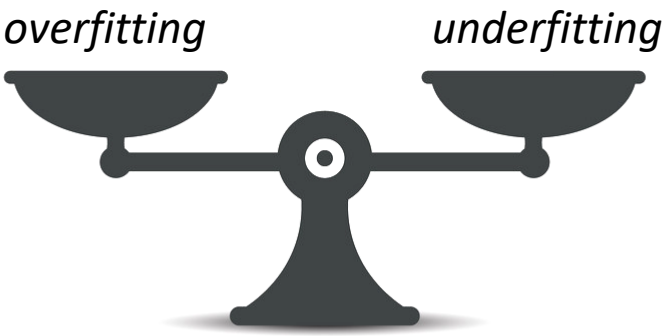
  → Based on problem framing.

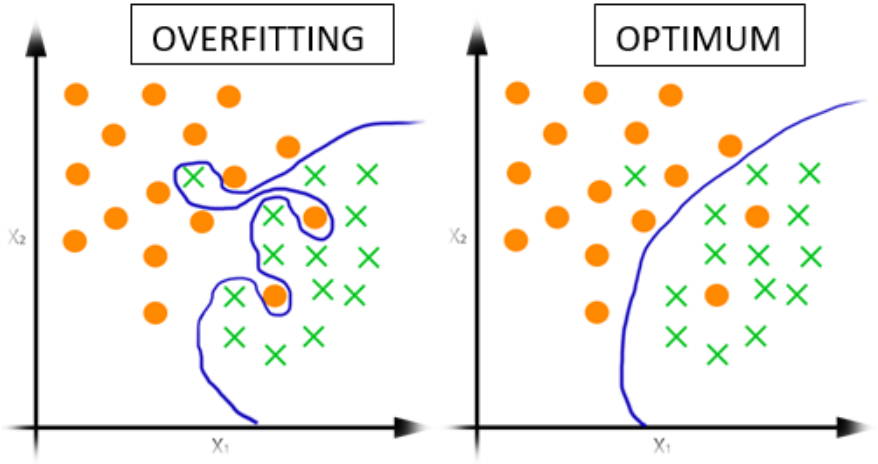
- Have a peak at literature, you don't have to re-invent the wheel, and nobody builds a neural network from scratch! But be critical…

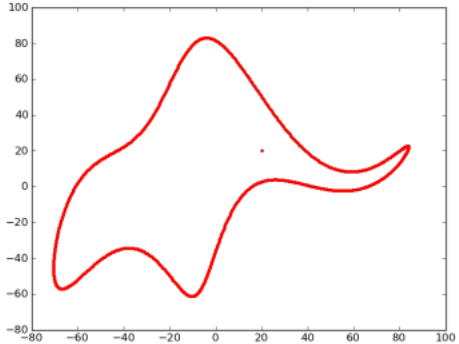
- Try short-list on training - validation set.

An end-to-end machine learning project

Step 6: Model selection







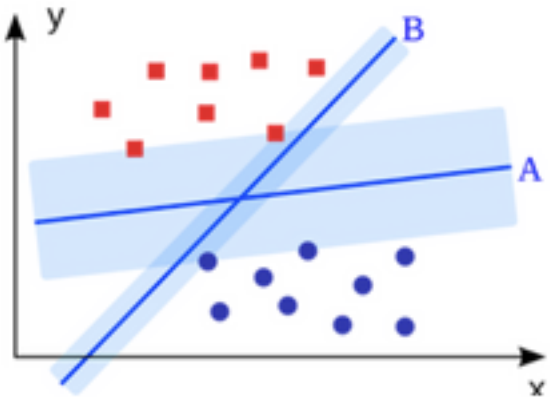
"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." – John von Neumann

Source: Overfitting and human behaviour, Medium (2018)

An end-to-end machine learning project

Step 6: Model selection

Support Vector Machine (SVM) – Support Vector Classifier (SVC)



Separates classes by finding plane between classes…
with maximum margin that ideally does not contain any instances

An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

Step 7: Fitting the model

```
# To select and train the linear SVC (Support-Vector classification)

from sklearn.svm import SVC
clf = SVC(gamma='auto')
clf.fit(train_features_forfit, train_labels)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

```
from sklearn.metrics import classification_report

predictions_train = clf.predict(train_features_forfit)
print(classification_report(train_labels, predictions_train))

predictions_test = clf.predict(test_features_forfit)
print(classification_report(test_labels, predictions_test))
```

	precision	recall	f1-score	support
0	0.90	0.98	0.94	258
1	0.93	0.73	0.81	102
micro avg	0.91	0.91	0.91	360
macro avg	0.91	0.85	0.87	360
weighted avg	0.91	0.91	0.90	360

Training set

	precision	recall	f1-score	support
0	0.88	0.95	0.92	64
1	0.86	0.69	0.77	26
micro avg	0.88	0.88	0.88	90
macro avg	0.87	0.82	0.84	90
weighted avg	0.88	0.88	0.87	90

Test set

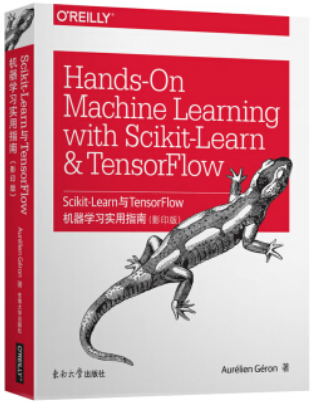
An end-to-end machine learning project

OUTLINE

1. Frame the problem, look at the big picture
2. Feature engineering
3. Get the data – data pre-processing
4. Creating a training, (validation) and test set
5. Feature selection
6. Model selection
7. Fitting the model
8. Model evaluation
9. QA – implement
10. share!

An end-to-end machine learning project

The end



github.com/JDhont/ESTRO38_premeeting

Python.org
Jupyter.org
Scikit-learn.org