University of Luxembourg

Multilingual. Personalised. Connected.

Machine Learning – Notions of Statistical Learning Theory

Martin GUBRI, 19/10/2021

uni.lu UNIVERSITÉ DU LUXEMBOURG

Who I am?

Martin Gubri

PhD candidate @Serval, SnT

Working on Machine Learning Security, with a focus on adversarial examples.

Session 1 & 2

1. Machine Learning Basics
    1. Recalls
    2. Short introduction to Statistical Learning Theory
    3. Interactive quiz

2. ML project life cycle
    1. Problem and goal definition
    2. Data collection and preparation
    3. Feature engineering
    4. Model buiding
    5. Model evaluation

Machine Learning Basics

Recalls

Model Elements

Hyperparameters & Parameters

Parameters

Internal values of the model learned during training

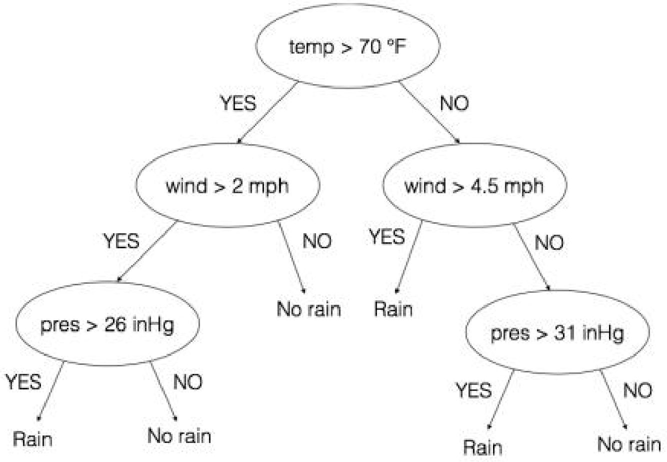
Learned values are called “Parameters Estimate”

E.g. feature and threshold of each node in a decision tree

Hyperparameters

Values that influence the model and cannot be learn

E.g depth of a decision tree



Example of a Decision Tree

Source image:http://ebubekirsezer.com/en/decision-treekarar

Machine Learning Basics

Some considerations of Statistical Learning Theory

Empirical risk minimization (ERM)

How to learn from data?

$f(x_i)$: the model’s prediction for i-th example

Loss: a measure of how far a model's predictions are from its label

In binary classification: $L(y_i\ ,\ f(x_i)) = \mathbb{I}(y_i\ \neq f(x_i))$

Empirical Risk: Average of the loss across the dataset’s examples

$$R_{emp}((X, y), f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i\ ,\ f(x_i))$$

Empirical Risk Minimization: finding a model that minimize the empirical risk

$$\min_{f} R_{emp}((X, y), f)$$

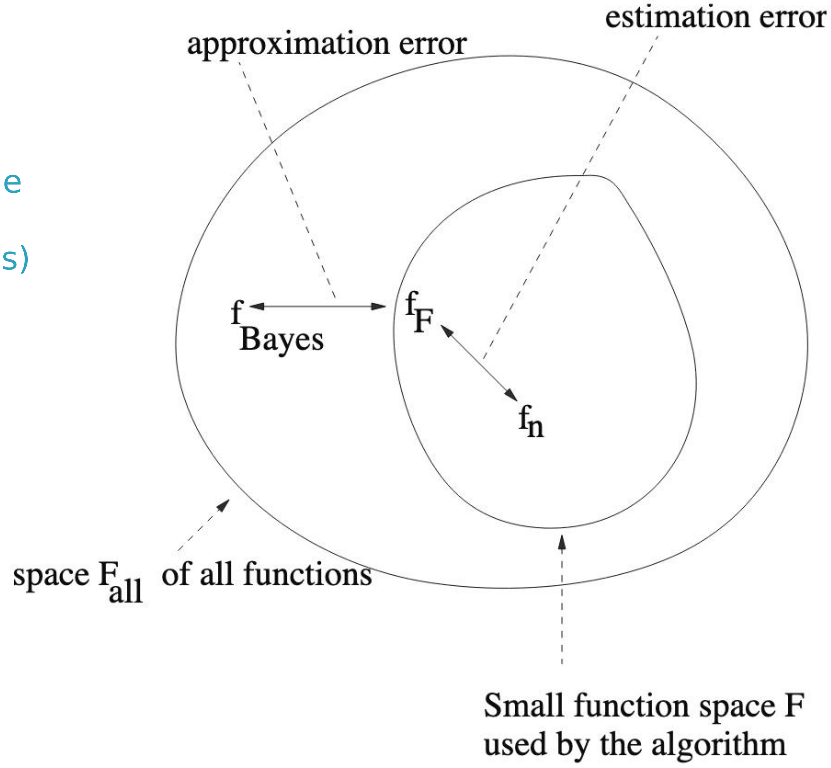
Decomposition of the error

$$R(f_n) - R(f_{Bayes}) = \underbrace{\Big(R(f_n) - R(f_{\mathcal{F}})\Big)}_{\text{estimation error}} + \underbrace{\Big(R(f_{\mathcal{F}}) - R(f_{Bayes})\Big)}_{\text{approximation error}}$$

your trained model

“best” unknown model
possible with the current set of features

best unknown model possible
with the chosen algorithm
(and the current set of features)



Source image:
U. von Luxburg and B. Scholkopf. “Statistical Learning Theory: Models, Concepts, and Results”,2011.

Overfitting & Underfitting

Decomposition of the error

$$R(f_n) - R(f_{Bayes}) = \underbrace{\Big(R(f_n) - R(f_{\mathcal{F}})\Big)}_{\text{estimation error}} + \underbrace{\Big(R(f_{\mathcal{F}}) - R(f_{Bayes})\Big)}_{\text{approximation error}}$$

VARIANCE ↔ BIAS

Source image:
U. von Luxburg and B. Scholkopf. “Statistical Learning Theory: Models, Concepts, and Results”,2011.

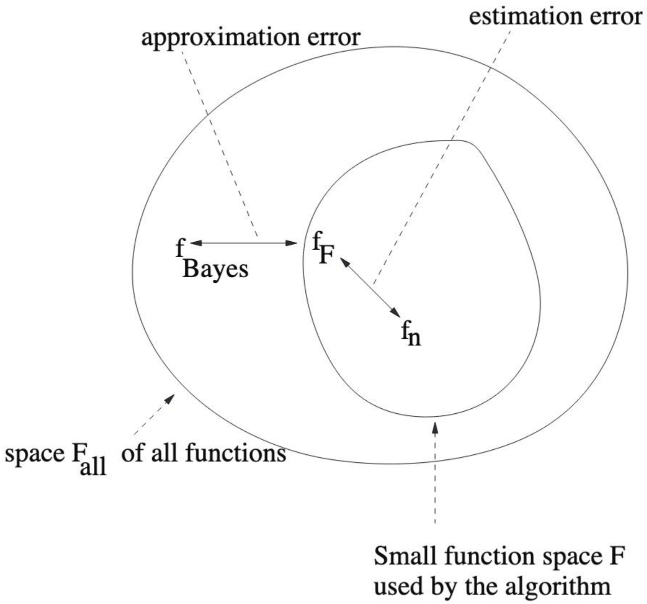
Bias-Variance tradeoff

Biais

Large approximation error, small estimation error → **Underfit**

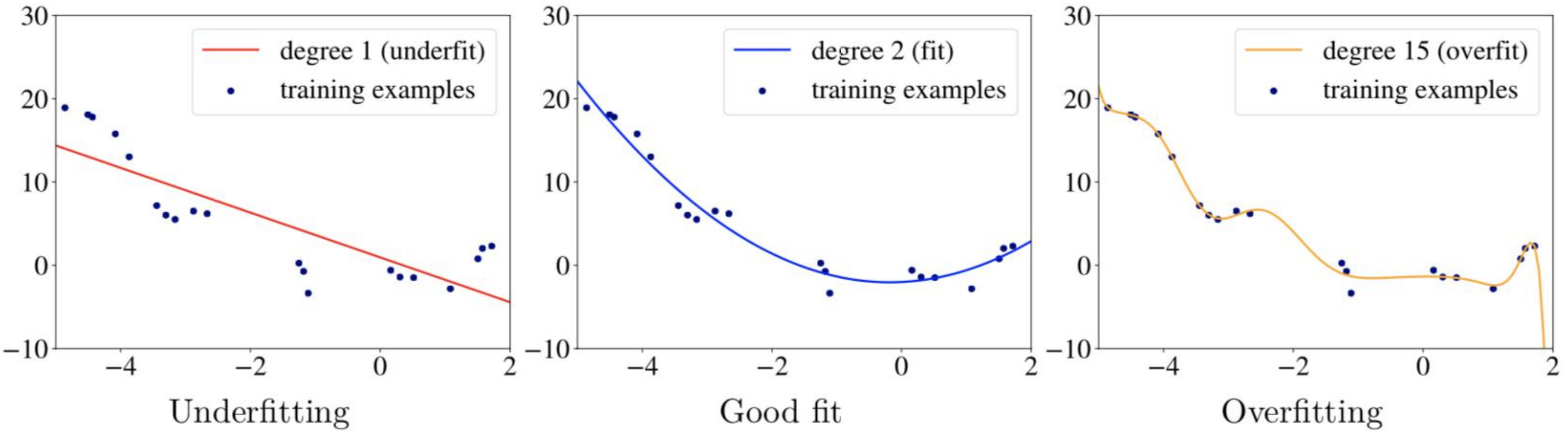
Variance

Large estimation error, small approximation error → **Overfit**


approximation error
estimation error
f_{Bayes}
f_F
f_n
space F_{all} of all functions
Small function space F used by the algorithm

Overfitting & Underfitting

Bias-Variance tradeoff



Underfitting

Good fit

Overfitting

Oversimplification of the data

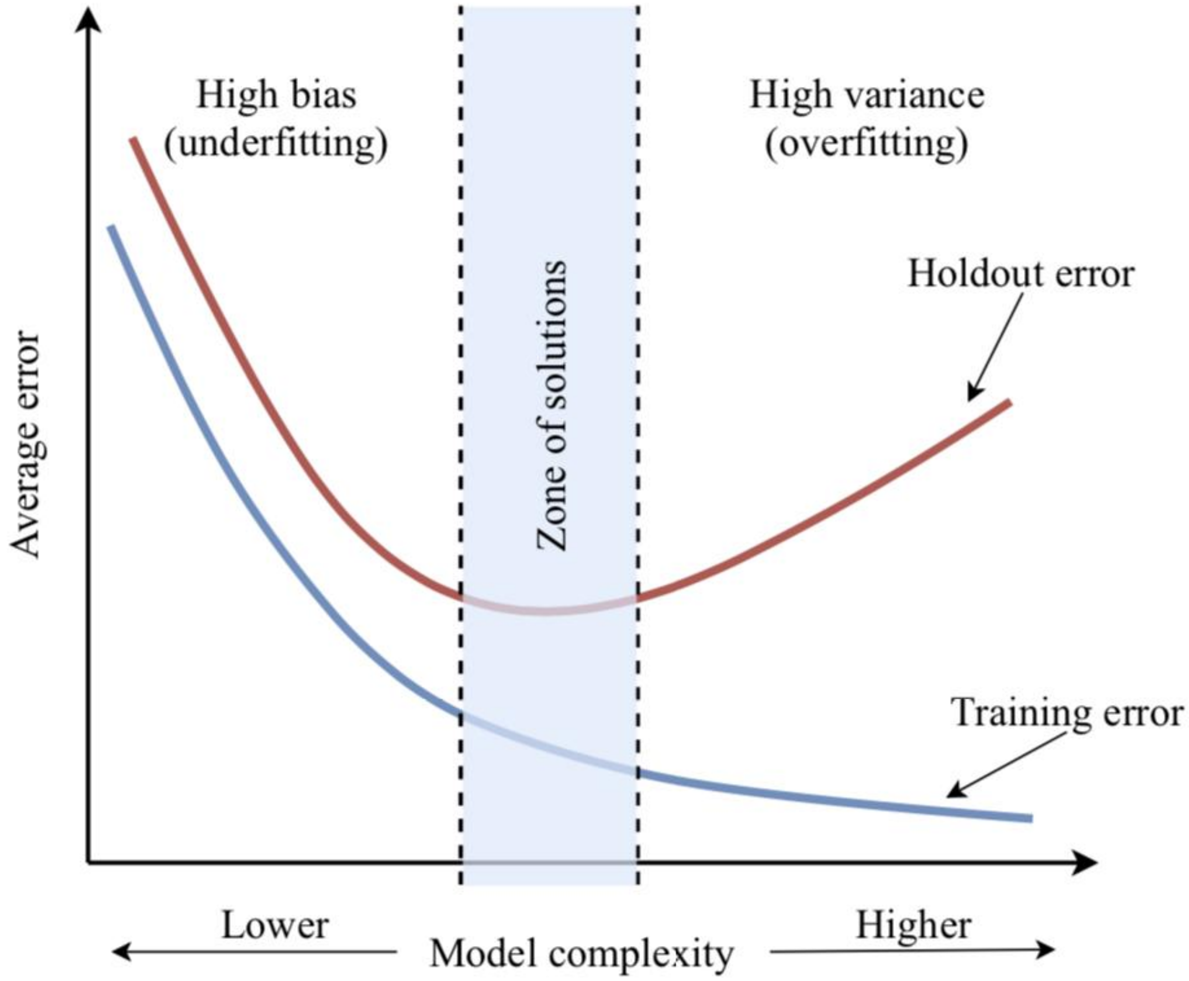
Learning the true signal

Fitting the noise of the data

Source image:
A. Burkov, “Machine Learning Engineering”.

Overfitting & Underfitting

Bias-Variance tradeoff



Source image:
A. Burkov, “Machine Learning Engineering”.

UNIVERSITÉ DU LUXEMBOURG

Interactive quiz

Question 1

You trained a binary classifier (for example a spam filter) and you expect the same number of examples from both classes. You obtained the following accuracies:

- accuracy computed on the train set (the examples that were used to train the model): **99%**
- accuracy computed on the holdout set (the examples that where **not** used to train the model): **70%**

Do you think that the trained model is overfitting?

- True
- False

Question 2

Same context, but now you obtained the following accuracies:

- accuracy computed on the train set (the examples that were used to train the model): **88%**
- accuracy computed on the holdout set (the examples that where **not** used to train the model): **87%**

Do you think that the trained model is overfitting?

- True
- False

Question 3

Same context, but now you obtained the following accuracies:

- accuracy computed on the train set (the examples that were used to train the model): **61%**
- accuracy computed on the holdout set (the examples that where **not** used to train the model): **60%**

Do you think that the trained model is overfitting?

- True
- False

Question 4

You want to classify images of cats and dogs. Your model takes as inputs pixels values of images, and outputs if the image corresponds to a cat (0) or a dog (1).

For this task you choose a decision tree. You set up a maximum depth of the tree of 8 (which means that for each image to predict your model will use at most 8 conditions to recognize a cat or a dog).

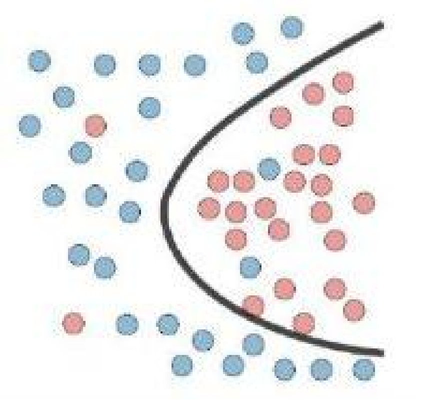
Is this model underfitting?

- True
- False

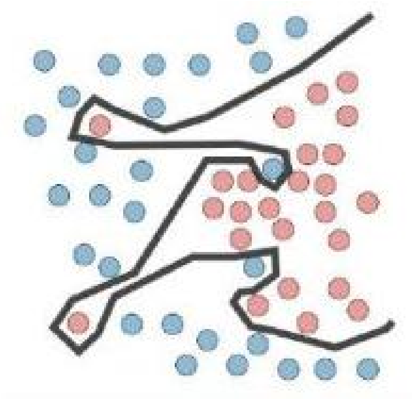
Question 5

Example of classification

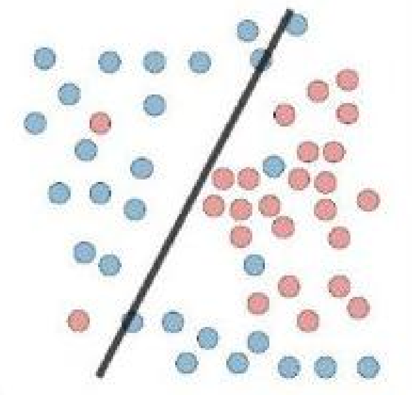
Which one is overfitting, underfitting, or just a good fit?



a)



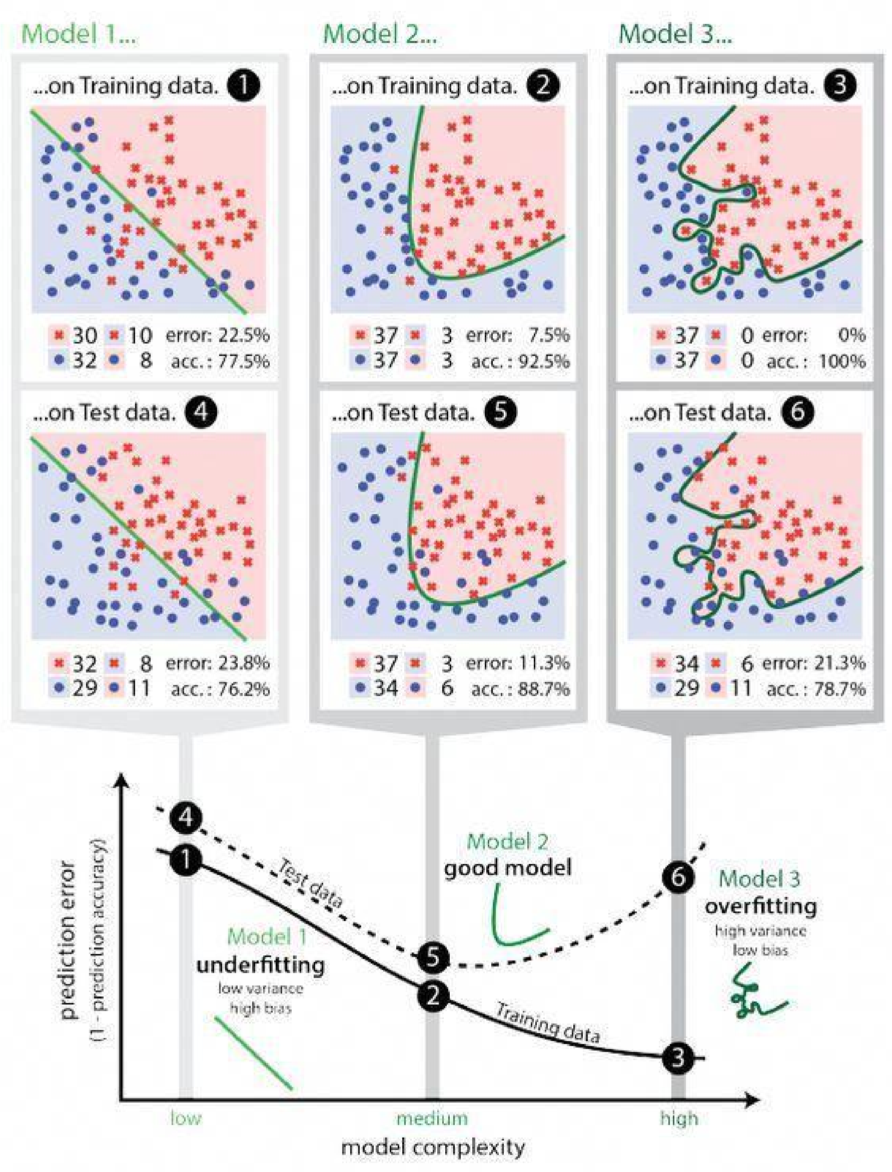
b)



c)

Source image.

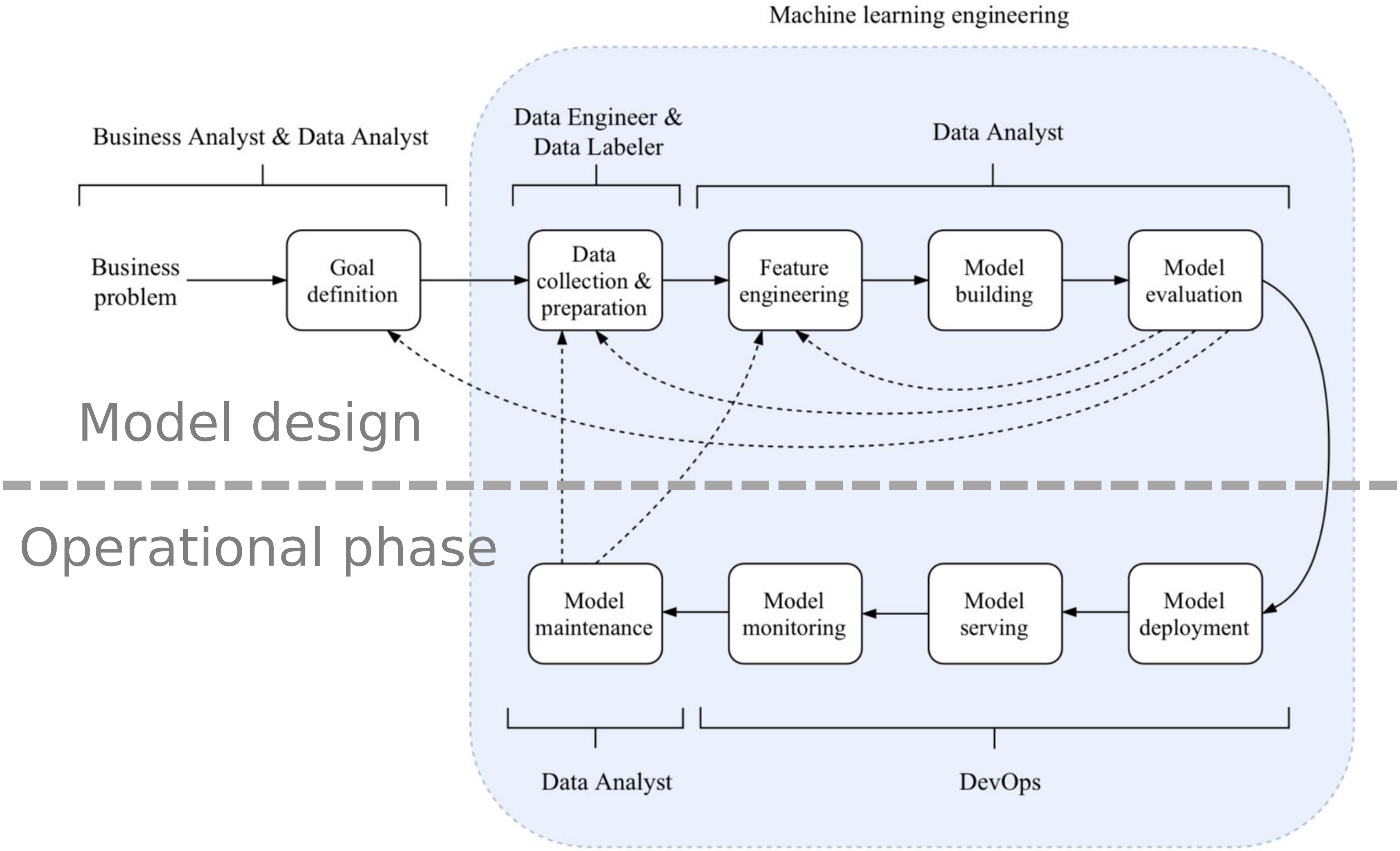
Summary



Source image.

Overview of the life cycle of ML project

ML project life cycle



Model design

Operational phase

Figure 3: Machine learning project life cycle.

Source image:
A. Burkov, “Machine Learning Engineering”.

ML project lifecycle

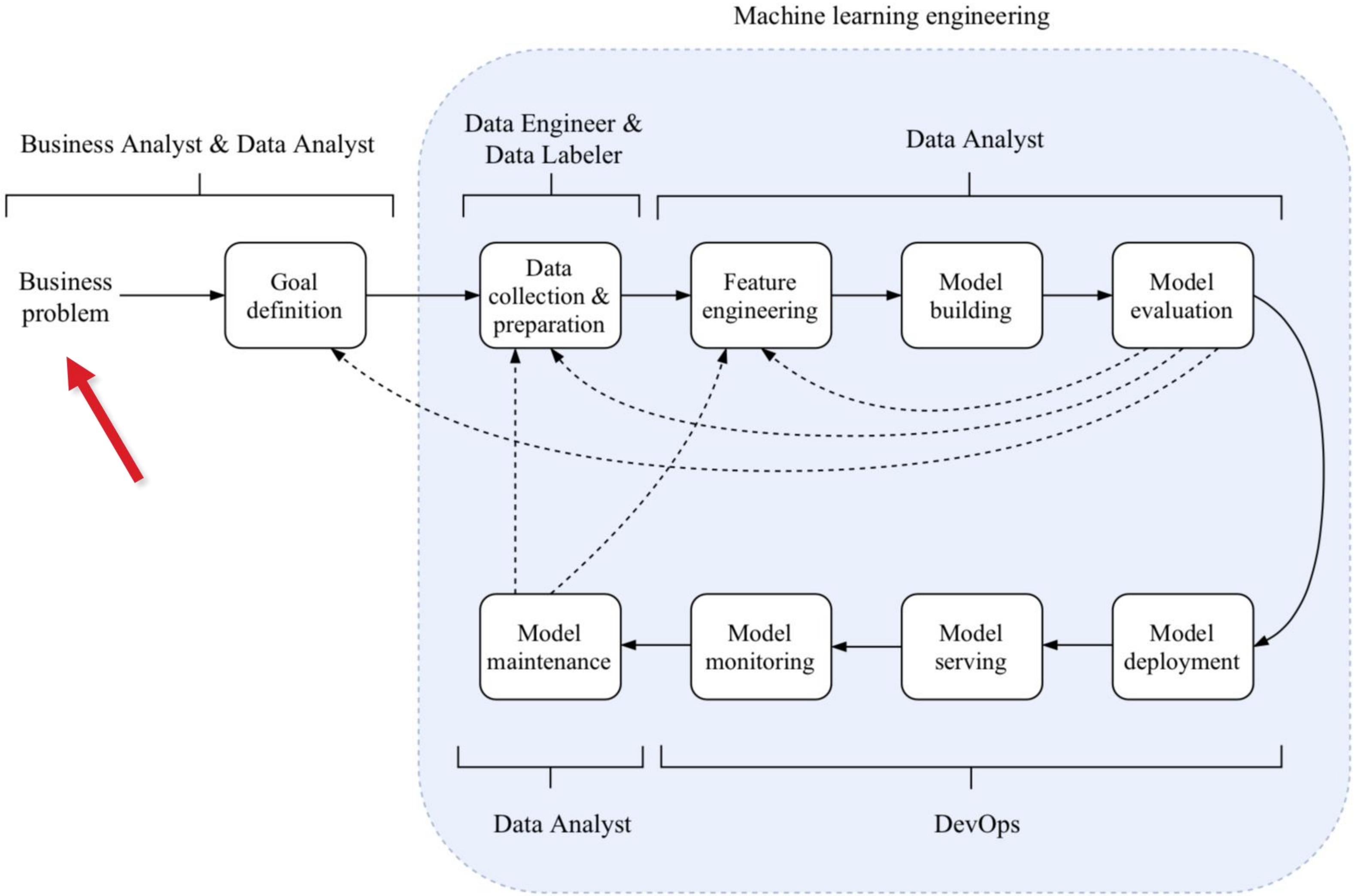


Figure 3: Machine learning project life cycle.

Source image:
A. Burkov, “Machine Learning Engineering”.

When to use Machine Learning?

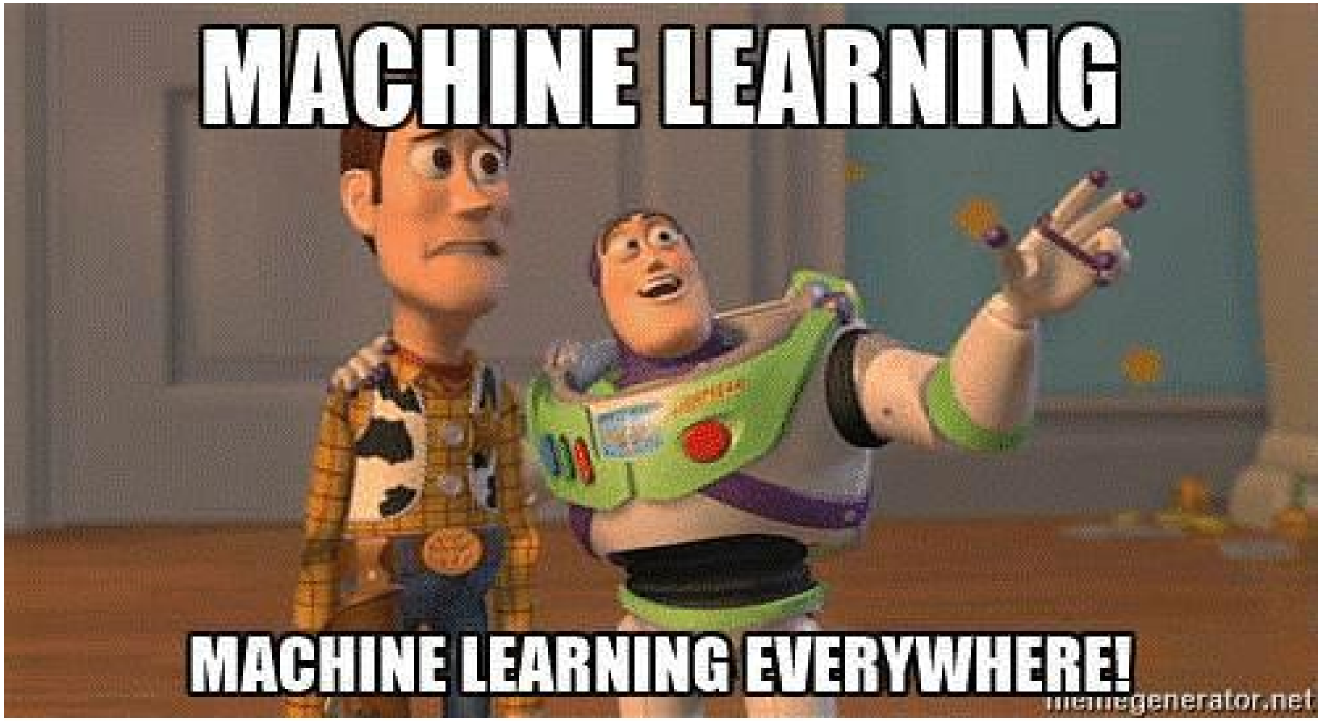
Consider using ML in those situations:

- the problem is too complex for coding
- the problem is constantly changing
- it is a perceptive problem (e.g. image, speech, and video recognition)
- it is an unstudied phenomenon
- the problem has a simple objective (e.g. yes/no decision, single number target)
- when it is cost-effective
  - 3 main costs: building the model, infrastructure to serve the model, model maintenance

Source content:
A. Burkov, “Machine Learning Engineering”.

Initial issue

When to **not** use Machine Learning?



When to **not** use Machine Learning?

Probably better to not use ML when:

- getting right data is too complex or impossible
  - no relation between features and labels
  - the phenomenon has too many outcomes while you cannot get enough examples to represent those outcomes
- you know how to solve the problem using traditional software development at a lower cost
  - e.g. you can fill an exhaustive lookup table manually by providing the expected output for any input
- every decision made by the system has to be explainable
  - can be a GDPR requirement
  - idem for decision changes in similar situations over time
- the cost of an error made by the system is too high

Source content:
A. Burkov, “Machine Learning Engineering”.

Exercice

ML or not ML

ML or not ML

Among the following situations choose the ones where you would use Machine Learning, and ones where it's much easier to use traditional software development.

- Compute the best play in a Tic-tac-toe game: https://en.wikipedia.org/wiki/Tic-tac-toe
- Identify the human language of audio recordings (English, French, etc.)
- Develop the guidance system of a space rocket
- Identify where are forests using Satellite imagery
- Develop a word processing software, like Word
- Extract content from thousands of websites which all have different structures

ML project lifecycle

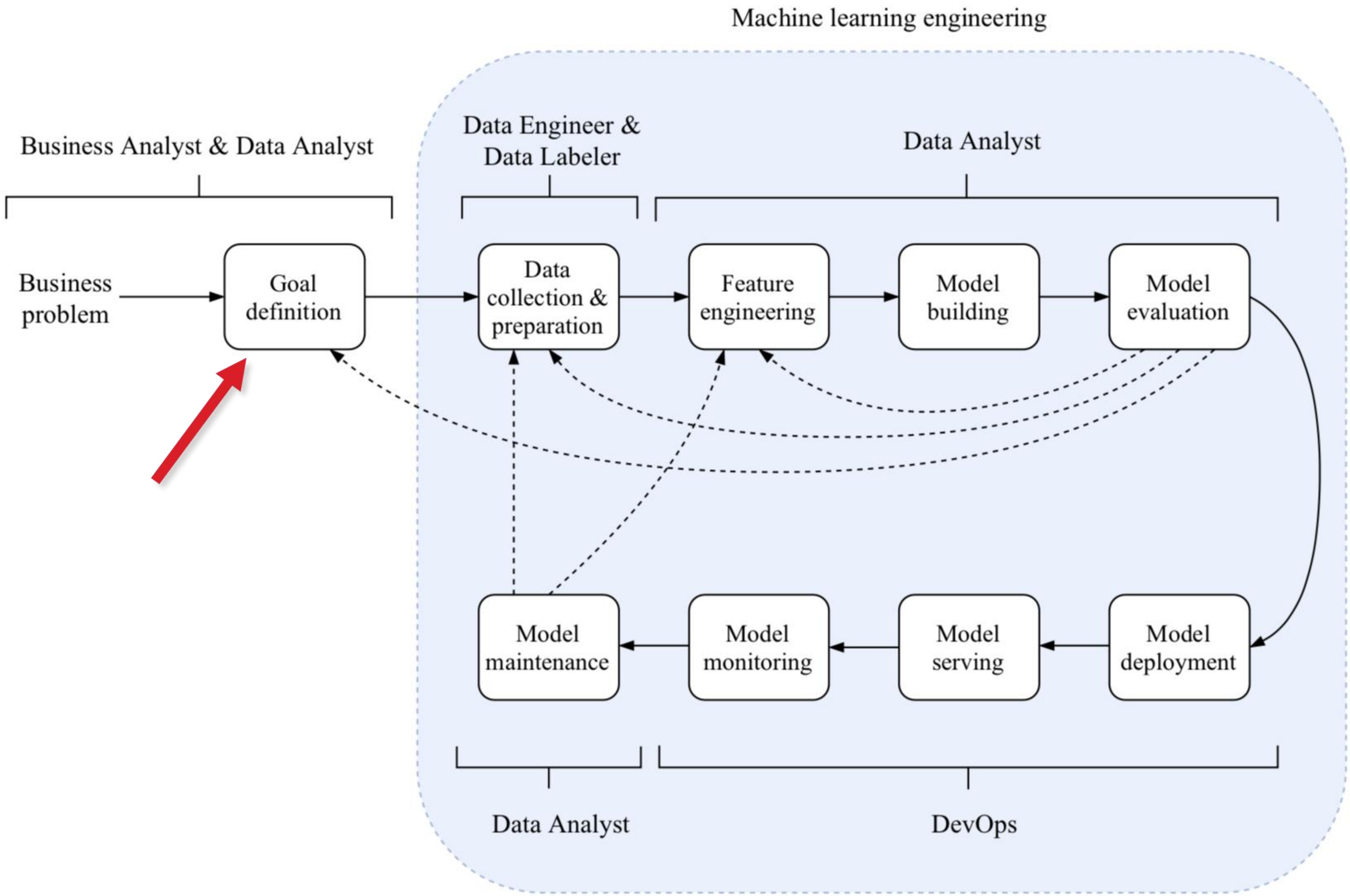


Figure 3: Machine learning project life cycle.

Source image:
A. Burkov, "Machine Learning Engineering".

Goal definition

Write down the elements of your problem

- What is my objective?
- What is my task?
- What is an example?
- What is the expected output?
- What are acceptable behaviors? Unacceptable behaviors?
  - Think of performances, security, discriminations, privacy, ethical issues, etc.
- How to measure them?
- **Lastly**, do I have available data? If not, can I build a dataset?

Adapted from:
A. Burkov, “Machine Learning Engineering”.

Goal definition

Example

Objective: be the best email provider

Task: automatically detect spam

Example: an email

Output: spam / ham

Acceptable behaviors:

- classify most of the spam, measured by probability of detection (sensitivity or recall) of at least 75%

Unacceptable behaviors:

- (performances) should only classify a very small amount of ham as spam, measured by specificity (true negative rate) of at least 95%
- (security) spammer should not fond easy way to fool the spam filter
- (privacy) email content should not be leaked from the trained model

Goal definition

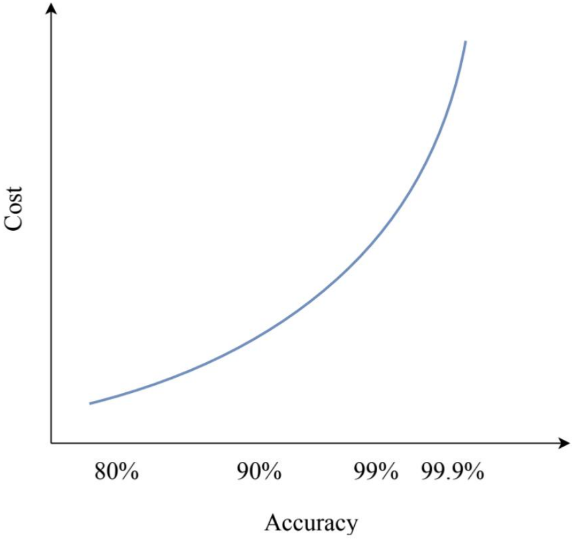
Content & figure from: A. Burkov, “Machine Learning Engineering”.

Conduce a **Cost-Benefit Analysis**

- Learn more: https://en.wikipedia.org/wiki/Cost%E2%80%93benefit_analysis

To estimate the **cost** consider:

- the difficulty of the problem
  - existing library
  - computation to train & serve the model
- the cost of data
  - can data be generated automatically?
  - cost of manual annotation
    labelling cats & dogs images ≠ doctors labelling X-ray scans
  - how many examples are approximately needed
- the needed accuracy
  - how costly is each wrong prediction?
  - lowest accuracy level below which the model is imp



ML project lifecycle

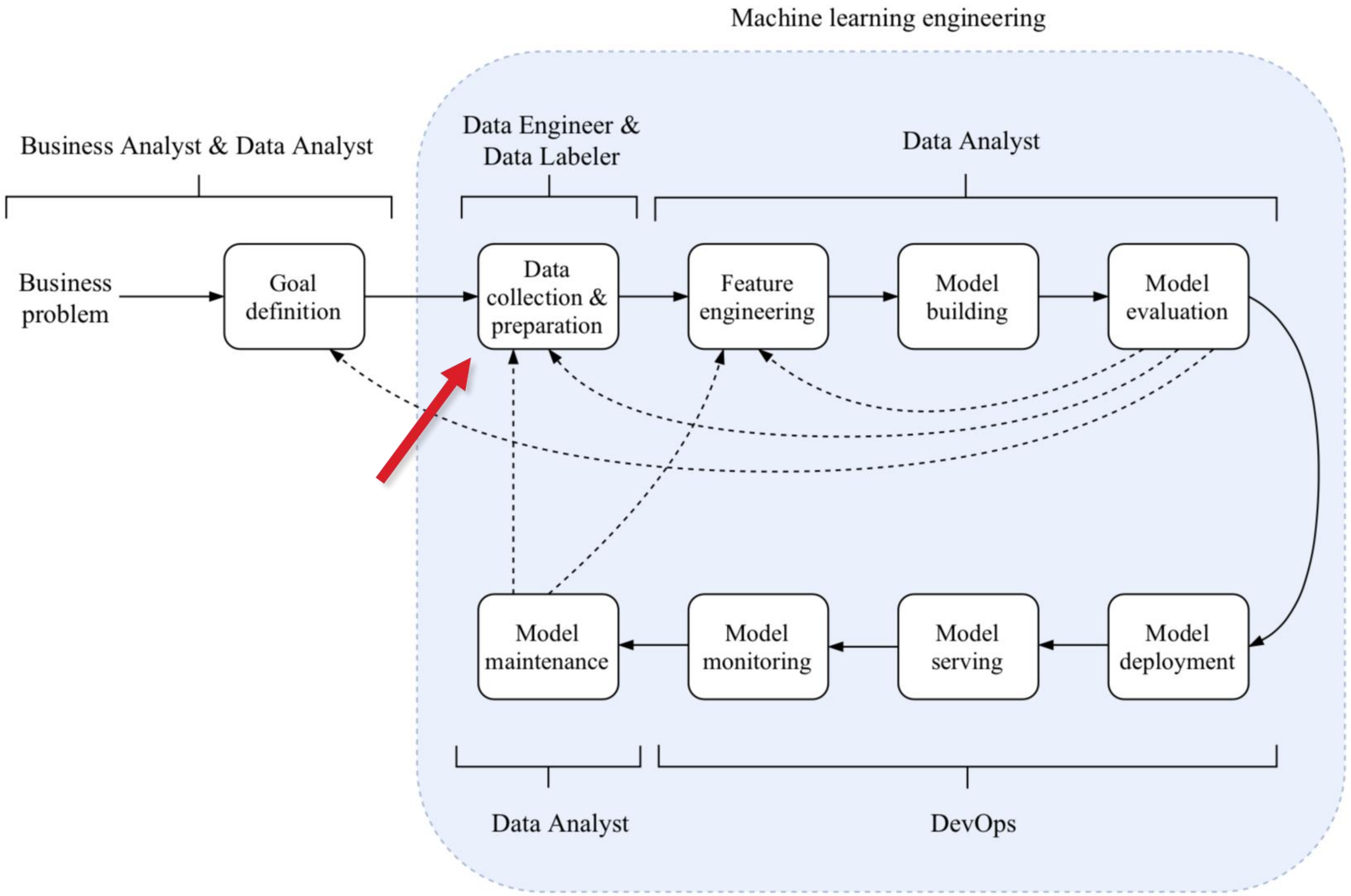


Figure 3: Machine learning project life cycle.

Source image:
A. Burkov, "Machine Learning Engineering".

Data collection & preparation

In industry: 80% building dataset, 20% playing with it

Steps:

- Existing data:
  - access to existing data (negotiation, paperwork, anonymization, etc.), database work, merging datasets, etc.
- Creating a dataset
  - ⚠ can be very long
  - collecting data, statistical sampling, scraping, survey, labelling data, etc.
- Data cleaning
  - Domain validity, outliers, errors, duplicates, expired data, etc.

Data collection & preparation

Advices

Think of **reproducibility** from the beginning

- you might have to extract the same data again in the future: bugs, adding features, update data, etc.
- avoid manual labor on data in Excel: error-prone & difficult to traceback

Always check raw data and processed data

- Don’t trust your implementation
- Keep backups of both datasets

Don’t assume that your data is valid

- Check domain validity (e.g. postal code, age)
- Properly encode missing values (e.g. “-1”, “99999” → NaN)

ML project lifecycle

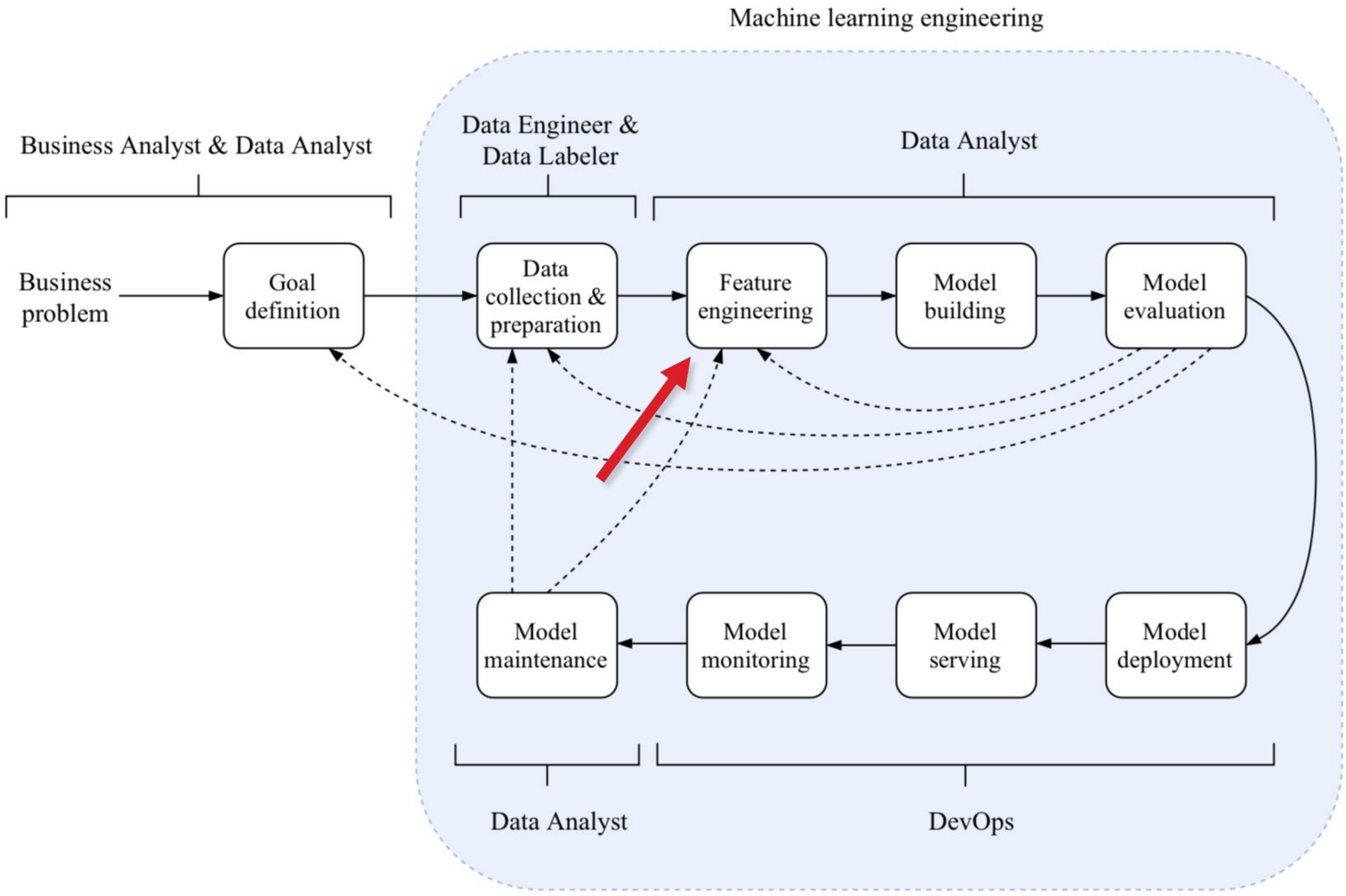


Figure 3: Machine learning project life cycle.

Source image:
A. Burkov, "Machine Learning Engineering".

Feature engineering

Goal

Transform raw data into tidy data with numerical or categorical features

Example:

Plain text email → Frequencies of “$”, “!”, “#”, “viagra”, “buy”, “mail”, etc.
Average length of uninterrupted sequences of capital letters.
Etc.

Tidy data

Follow principles of **tidy data**

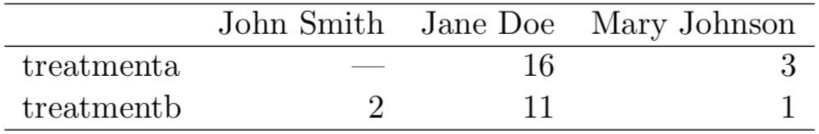
- 1 row ←→ 1 example
- 1 column ←→ 1 feature
- single dataset

Sources of messiness:

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of experimental unit stored in the same table
- One type of experimental unit stored in multiple tables

Source content:
Wickham, H. (2014). Tidy Data.
Journal of Statistical Software,
59(10).

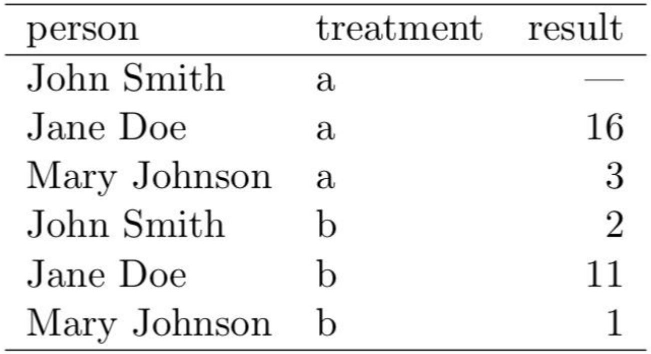
Tidy data



	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

×

→



person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

✓

See more examples of bad practices: https://www.jstatsoft.org/article/view/v059i10

Source images:
Wickham, H. (2014). Tidy Data.
Journal of Statistical Software,
59(10).

Steps

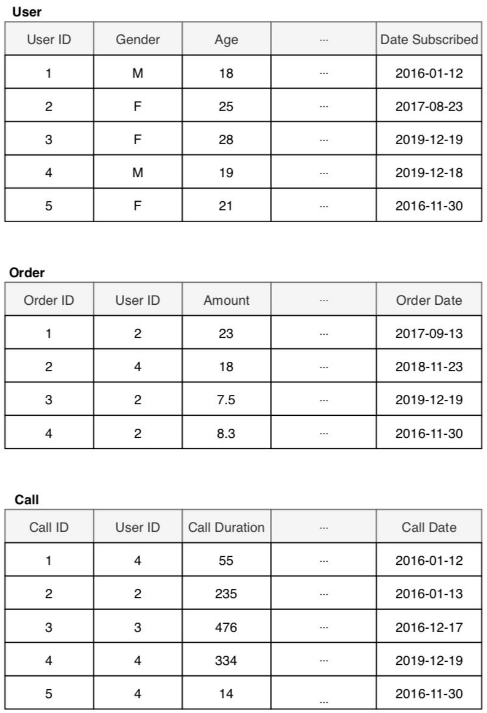
1. Defining the features to represent well your examples
   - Usually involve domain experts
   - Document everything into a schema file (spreadsheet or Json with name, type, definition, missing values, source, etc.)

2. Implement their computation
   - reproducibility + checks
   - data manipulation: filter, transform, aggregate, sort

Summarizing data

You can compute **mean & standard deviation** for each feature to aggregate

Example of Churn Analysis



User

User ID	Gender	Age	...	Date Subscribed
1	M	18	...	2016-01-12
2	F	25	...	2017-08-23
3	F	28	...	2019-12-19
4	M	19	...	2019-12-18
5	F	21	...	2016-11-30

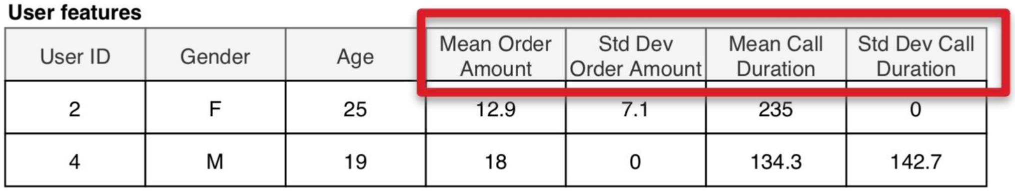
Order

Order ID	User ID	Amount	...	Order Date
1	2	23	...	2017-09-13
2	4	18	...	2018-11-23
3	2	7.5	...	2019-12-19
4	2	8.3	...	2016-11-30

Call

Call ID	User ID	Call Duration	...	Call Date
1	4	55	...	2016-01-12
2	2	235	...	2016-01-13
3	3	476	...	2016-12-17
4	4	334	...	2019-12-19
5	4	14	...	2016-11-30

Figure 24: Relational data for churn analysis.



User features

User ID	Gender	Age	Mean Order Amount	Std Dev Order Amount	Mean Call Duration	Std Dev Call Duration
2	F	25	12.9	7.1	235	0
4	M	19	18	0	134.3	142.7

Figure 25: Synthetic features based on sample mean and standard deviation.

Steps

3. Encoding categorical features
   - Models accept **only numerical** values
   - Categorial features should be encoded as integers (**one-hot encoded**):
     - “male”, “female”, “female” → 0, 1, 1
     - “bachelor”, “master”, “bachelor”, “high school”, “master” →

0	1	0
0	0	1
0	1	0
1	0	0
0	0	1

4. Discretization of continuous variable *(optional):*
   - Age: 8, 65, 42, 15 → “-18”, “50+”, “18-50”, “-18” →

1	0	0
0	0	1
0	1	0
1	0	0

Steps

4. **Missing data treatment**
   - Most models don’t accept missing data
   - Solutions: imputation, categorization, etc.

5. **Data Normalization of continuous features**
   - I.e. mean removal and variance scaling
   - Almost all models performs better with normalized data

6. Other features transformation
   - E.g, polynomial features: (X_1, X_2) to $(1, X_1, X_2, X_1^2, X_1X_2, X_2^2)$

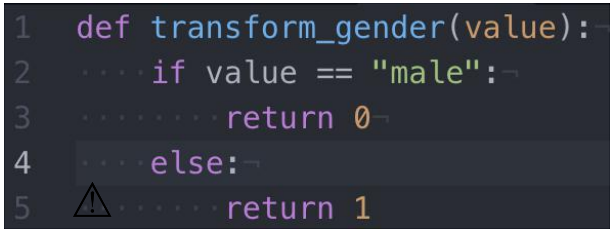
Feature engineering

Practical advices

Informative features (high predictive power)

- Constant data are useless
- Totally unrelated features are not useful
- Duplicated features are not useful (e.g. weight in kg and in pounds)

Always check the presence of missing data



```python
def transform_gender(value):
    if value == "male":
        return 0
    else:
        return 1
```

Use all data analysis tools to help you

- descriptive statistics (mean, standard deviation, min, max, quantiles, absolute and relative frequencies, etc.), data visualization, etc.

Use small sample to develop and debug

- Unit test for each data extractor

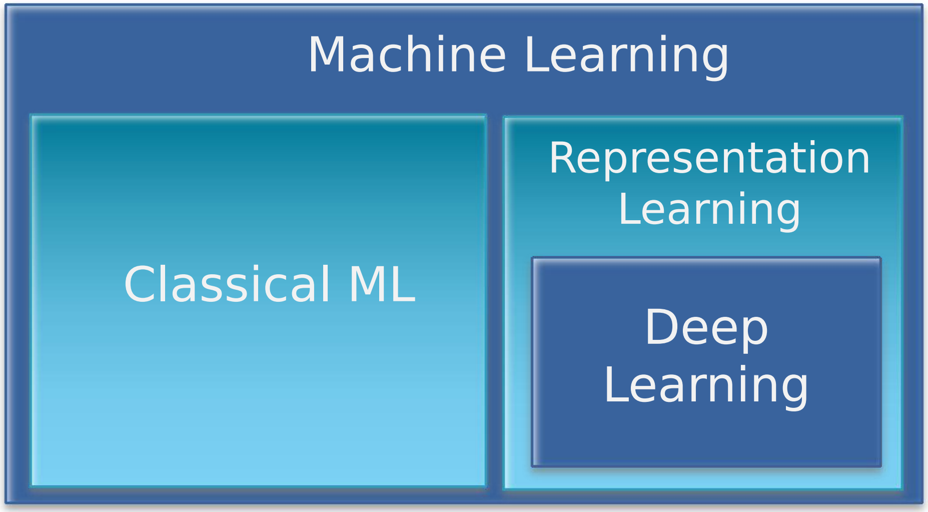
Feature engineering

Deep Learning

No feature engineering in Deep Learning

But has a cost

- number of examples
(in millions)
- complexity



When to **not** use Deep Learning?

ML project lifecycle

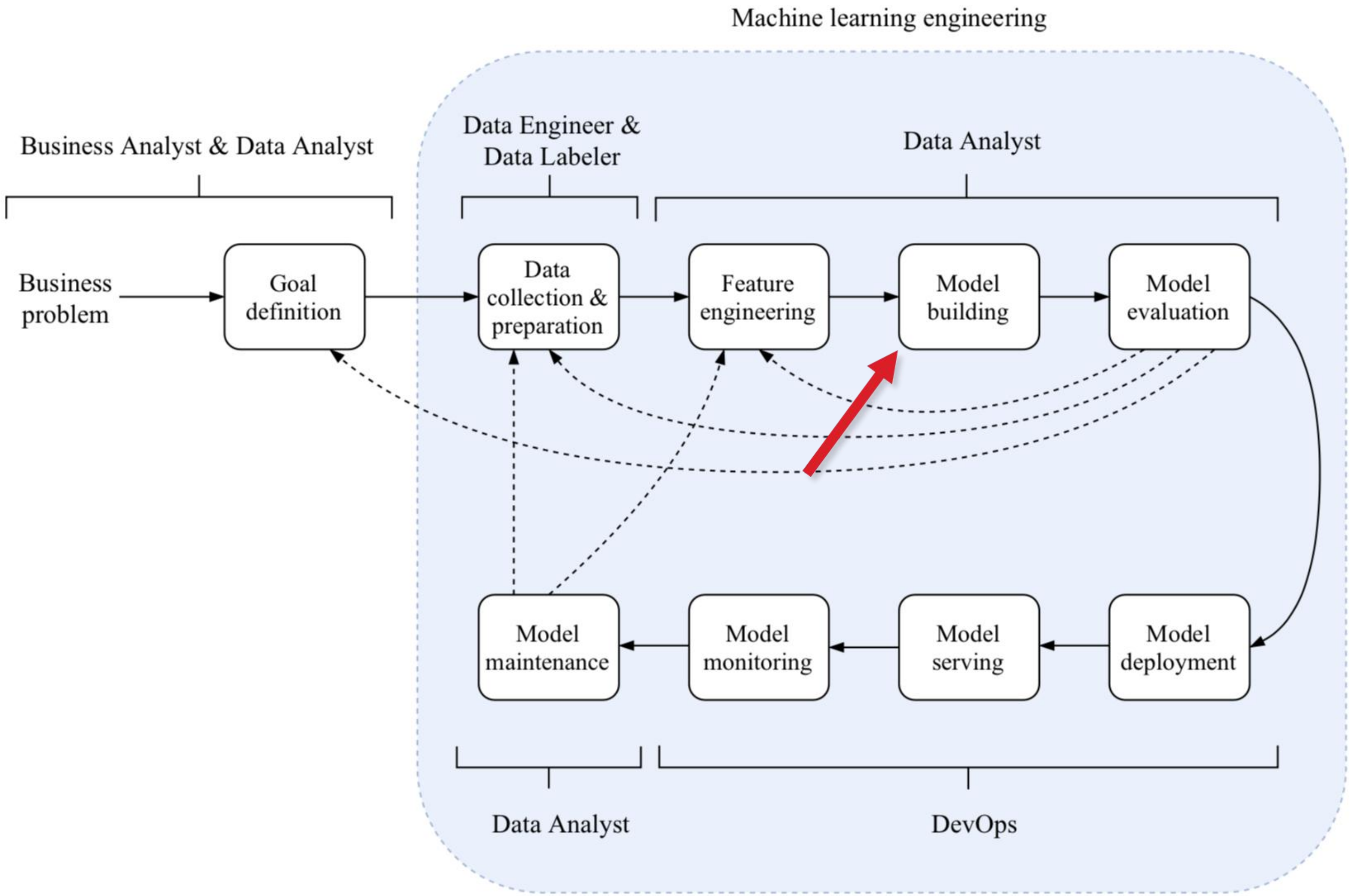


Figure 3: Machine learning project life cycle.

Source image:
A. Burkov, "Machine Learning Engineering".

Choosing a model

No free lunch theorem

Theory said that there is no universally best model (Wolpert 1996)
Learning is impossible unless we make assumptions on the underlying distribution.

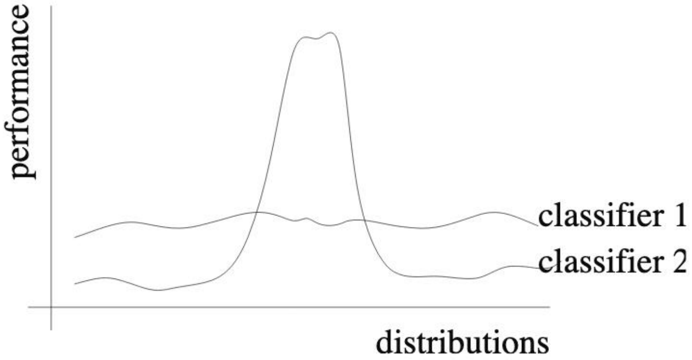


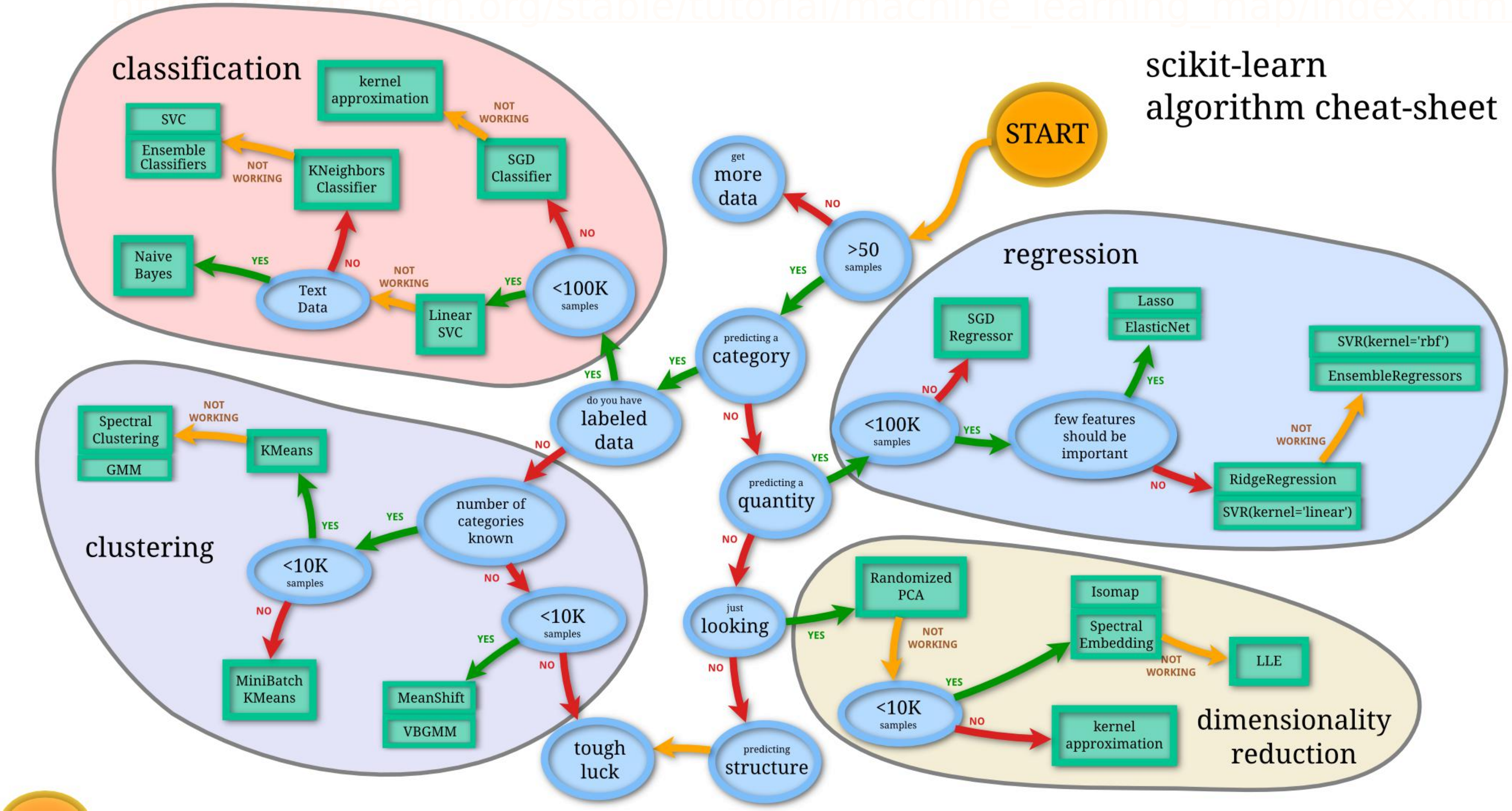
Figure 7. No free lunch theorem: Classifier 1 depicts a general purpose classifier. It performs moderately on all kinds of distributions. Classifier 2 depicts a more specialized classifier which has been tailored towards particular distributions. It behaves very good on those distributions, but worse on other distributions. According to the no free lunch theorem, the average performance of all classifiers over all distributions, that is the area under the curves, is identical for all classifiers.

Source image:
U. von Luxburg and B. Scholkopf. “Statistical Learning Theory: Models, Concepts, and Results”,2011.

Choosing a model

Practical advices

- Some empirical guide can help you, e.g this sklearn flowchart: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html



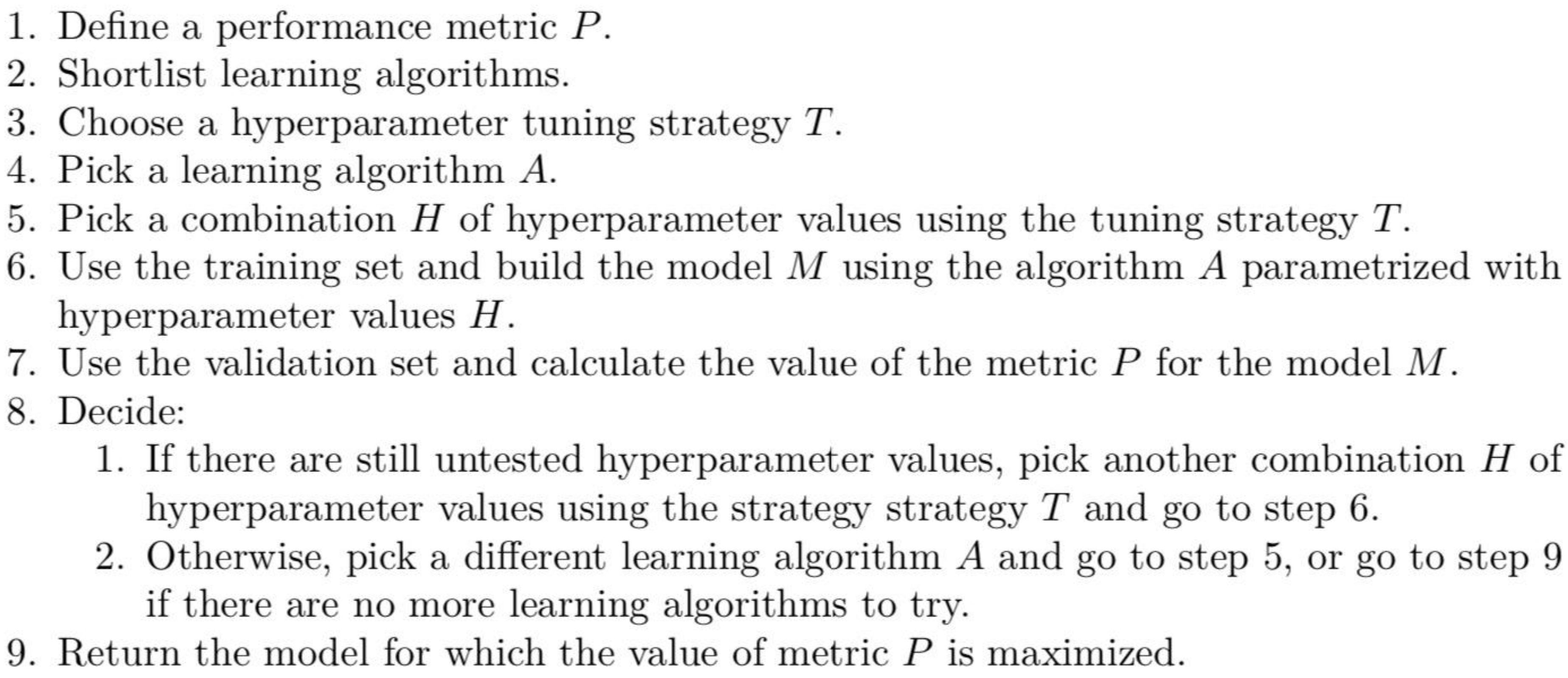
Source image:
Sklearn documentation

Choosing a model

Practical advices

- Select models with the following criterion:
  - Explainability
  - Number of examples and features
  - Non-linearity of the data
  - Computational complexity
  - Prediction speed
- Choose model with best performances always evaluated with Cross Validation
- Don’t forget to tune hyperparameters of model
  - Different strategies: grid search, random search, etc.

Model Building Strategy



1. Define a performance metric P.
2. Shortlist learning algorithms.
3. Choose a hyperparameter tuning strategy T.
4. Pick a learning algorithm A.
5. Pick a combination H of hyperparameter values using the tuning strategy T.
6. Use the training set and build the model M using the algorithm A parametrized with hyperparameter values H.
7. Use the validation set and calculate the value of the metric P for the model M.
8. Decide:
    1. If there are still untested hyperparameter values, pick another combination H of hyperparameter values using the strategy strategy T and go to step 6.
    2. Otherwise, pick a different learning algorithm A and go to step 5, or go to step 9 if there are no more learning algorithms to try.
9. Return the model for which the value of metric P is maximized.

Source content:
A. Burkov, “Machine Learning Engineering”.

ML project lifecycle

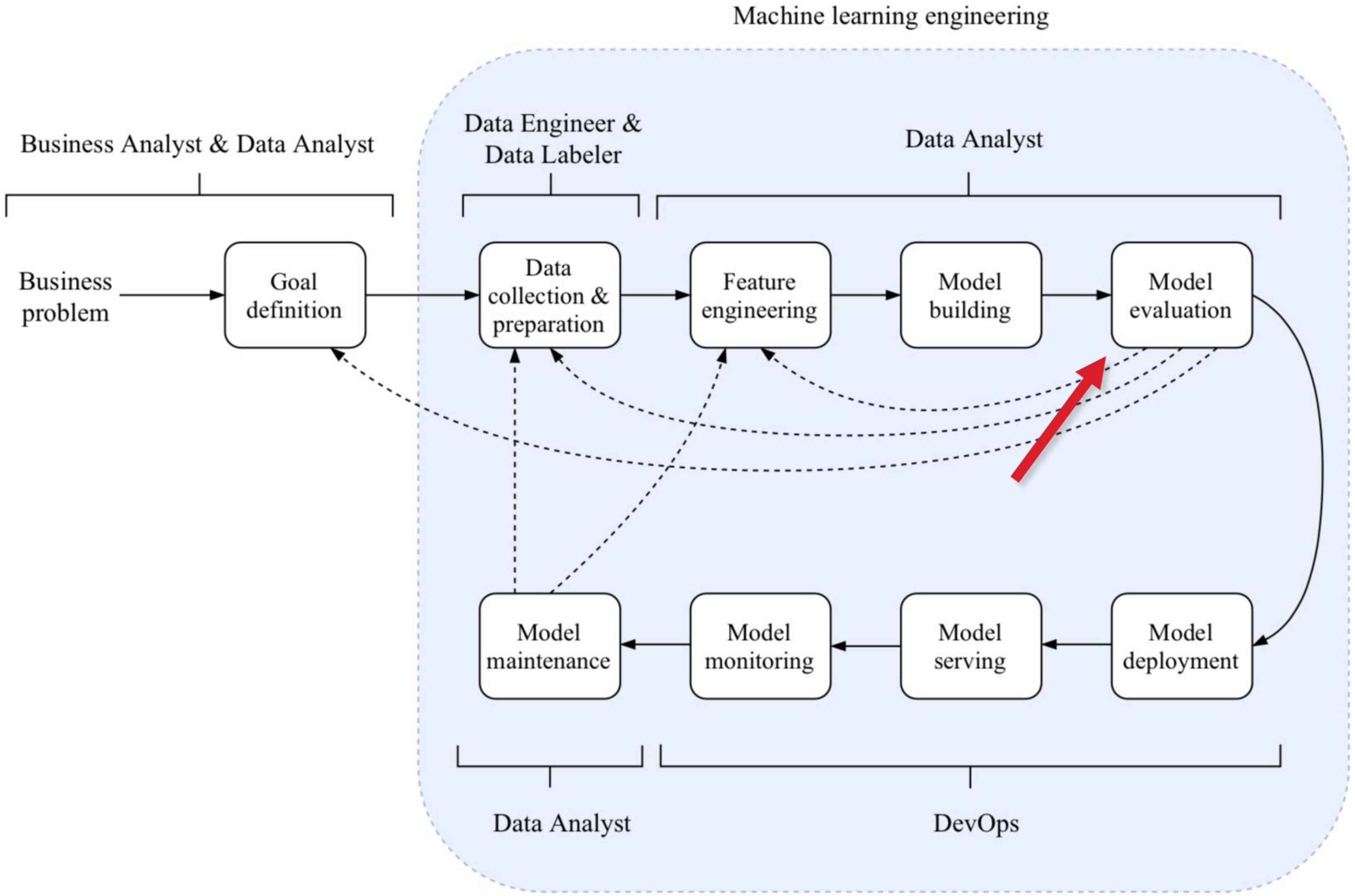


Figure 3: Machine learning project life cycle.

Source image: A. Burkov, “Machine Learning Engineering”.

Model evaluation

Overfitting ⚠

Always use **Cross-Validation!**

- Shuffle the examples
- Split them into 3 distinct sets:
  1. Training set → for model training
  2. Validation set → for tuning hyperparamers & choosing models
  3. Test set → for independent evaluation of your final model
- Good practice: split before at the beginning of data preparation step

Extreme case:

- Your model simply memorizes its training examples but returns random labels for new examples.
- Need to evaluate on “unseen” examples.

Model evaluation

Choose an appropriate metric for your problem

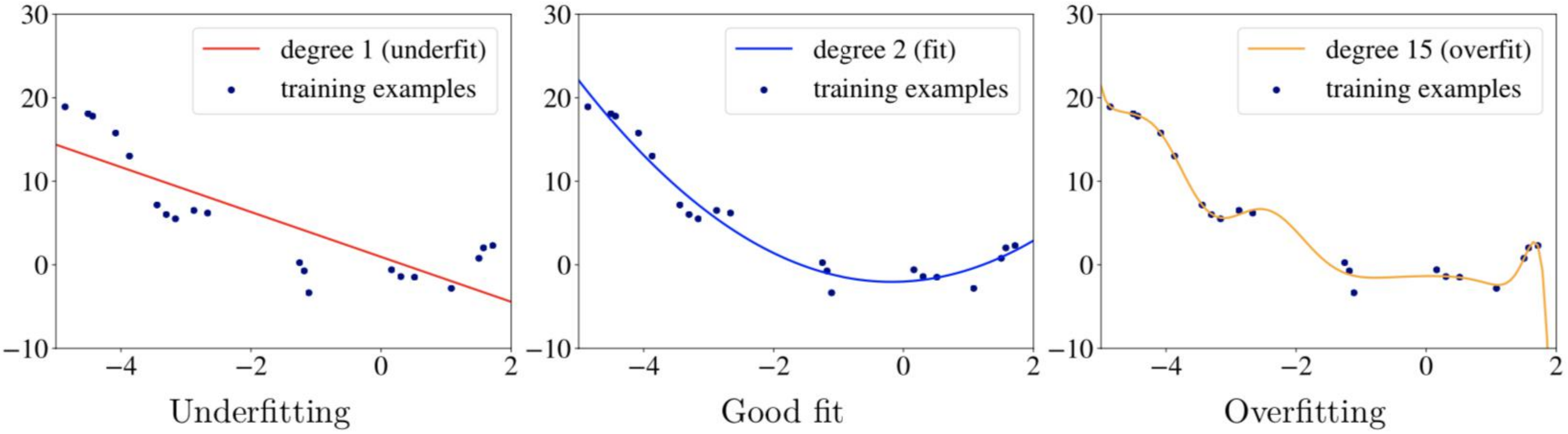
- Report it computed on the test set
- Baseline:
  - performance of the simplest algorithm, usually random labelling
  - human performance

Be careful to class imbalance problem

- Consider credit card fraud detection
- Probably have 1% of fraud among all transactions
- Accuracy isn’t appropriate metric here: you can achieve 99% accuracy, just by classifying all transactions as genuine, which has 0 practical use
- In that case, metrics

Detecting Underfitting / Overfitting

Example of Regression



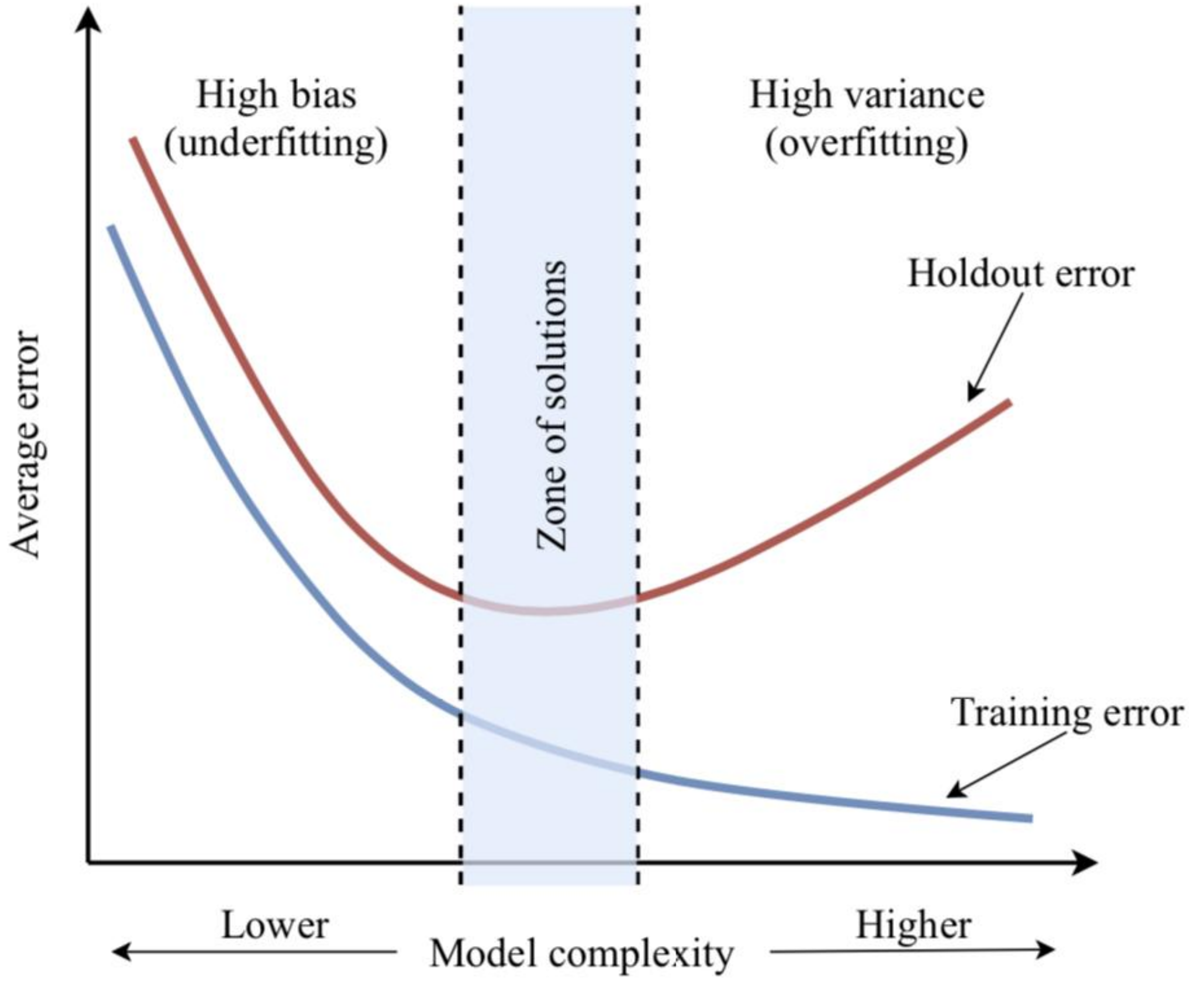
Oversimplification of the data

Learning the true signal

Fitting the noise of the data

Source image:
A. Burkov, “Machine Learning Engineering”.

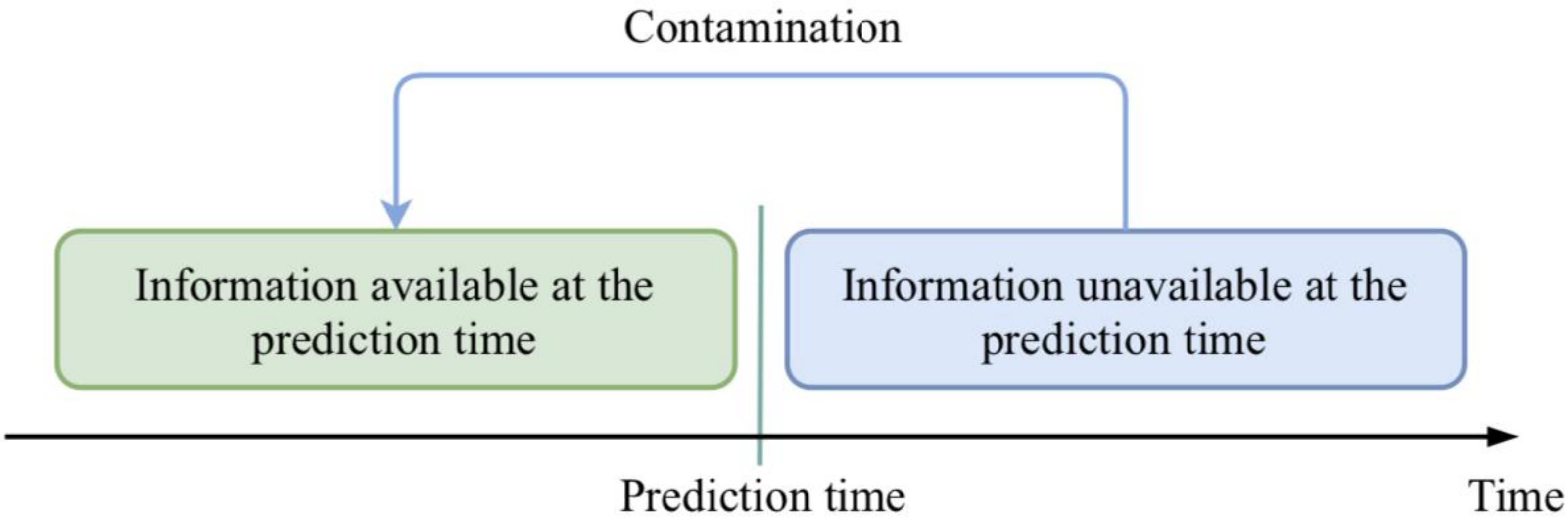
Detecting Underfitting / Overfitting



Source image:
A. Burkov, “Machine Learning Engineering”.

Data Leakage

Be careful to **data leakage** if your examples are not independent


Contamination
Information available at the prediction time
Information unavailable at the prediction time
Prediction time
Time

Source image:
A. Burkov, “Machine Learning Engineering”.

More information

- List of datasets: https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research

- Getting started with Sklearn: https://scikit-learn.org/stable/getting_started.html

- Sklearn cheat sheet: https://www.datacamp.com/community/blog/scikit-learn-cheat-sheet

- The hundred-page Machine Learning Book: http://themlbook.com

References

- A. Burkov. “Machine Learning Engineering”. Draft available at mlebook.com

- K. Murphy. “Machine Learning: a Probabilistic Perspective”. MIT Press, 2012

- U. von Luxburg and B. Scholkopf. “Statistical Learning Theory: Models, Concepts, and Results”. In Handbook of the History of Logic, Vol. 10: Inductive Logic, volume 10, pages 651–706, 2011

- H. Wickham. “Tidy Data”. Journal of Statistical Software, 59(10), 2014

Quiz

ML or not ML

Question 1

Cost to increase accuracy

In general, the cost to train a model (ie. developers' time + computational needs) grows linearly with its accuracy

- True
- False

Question 2

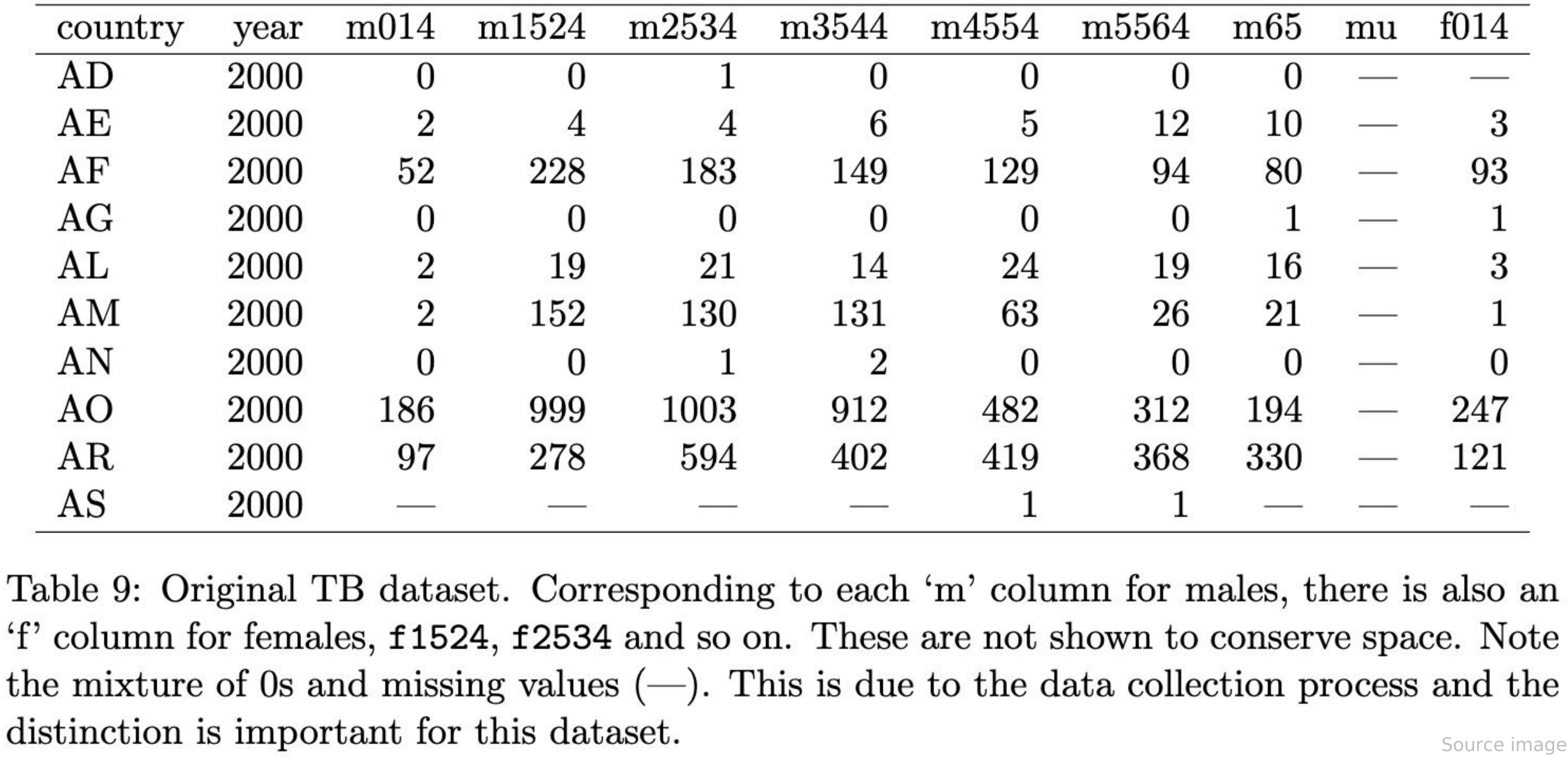
Cross-validation

We want to predict the GDP of countries from its economics caracteristics. We collect these variables for each year since 1900 and each countries. We shuffle all the data, keep-out 20% for the test set, and train a time-series model on the 80%.

What are the issues with this model?

Question 2

Is it tidy data?



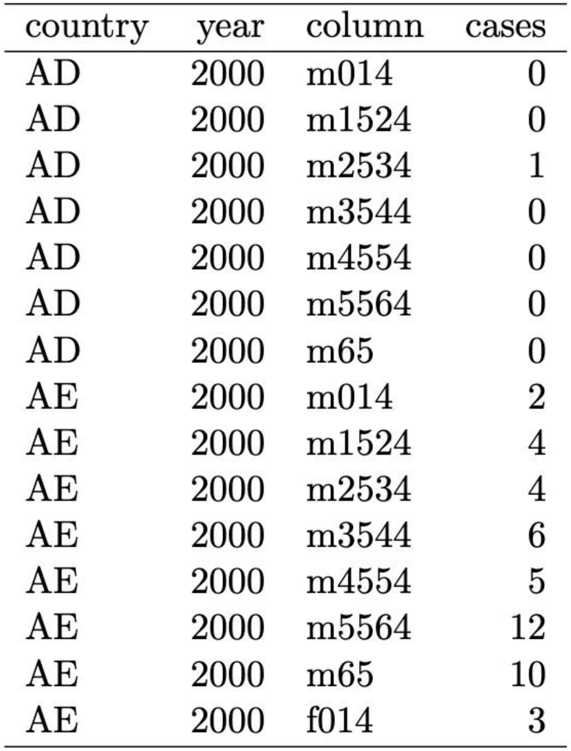
country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Table 9: Original TB dataset. Corresponding to each ‘m’ column for males, there is also an ‘f’ column for females, `f1524`, `f2534` and so on. These are not shown to conserve space. Note the mixture of 0s and missing values (—). This is due to the data collection process and the distinction is important for this dataset.

Source images:
Wickham, H. (2014). Tidy Data.
Journal of Statistical Software,
59(10)

Question 2

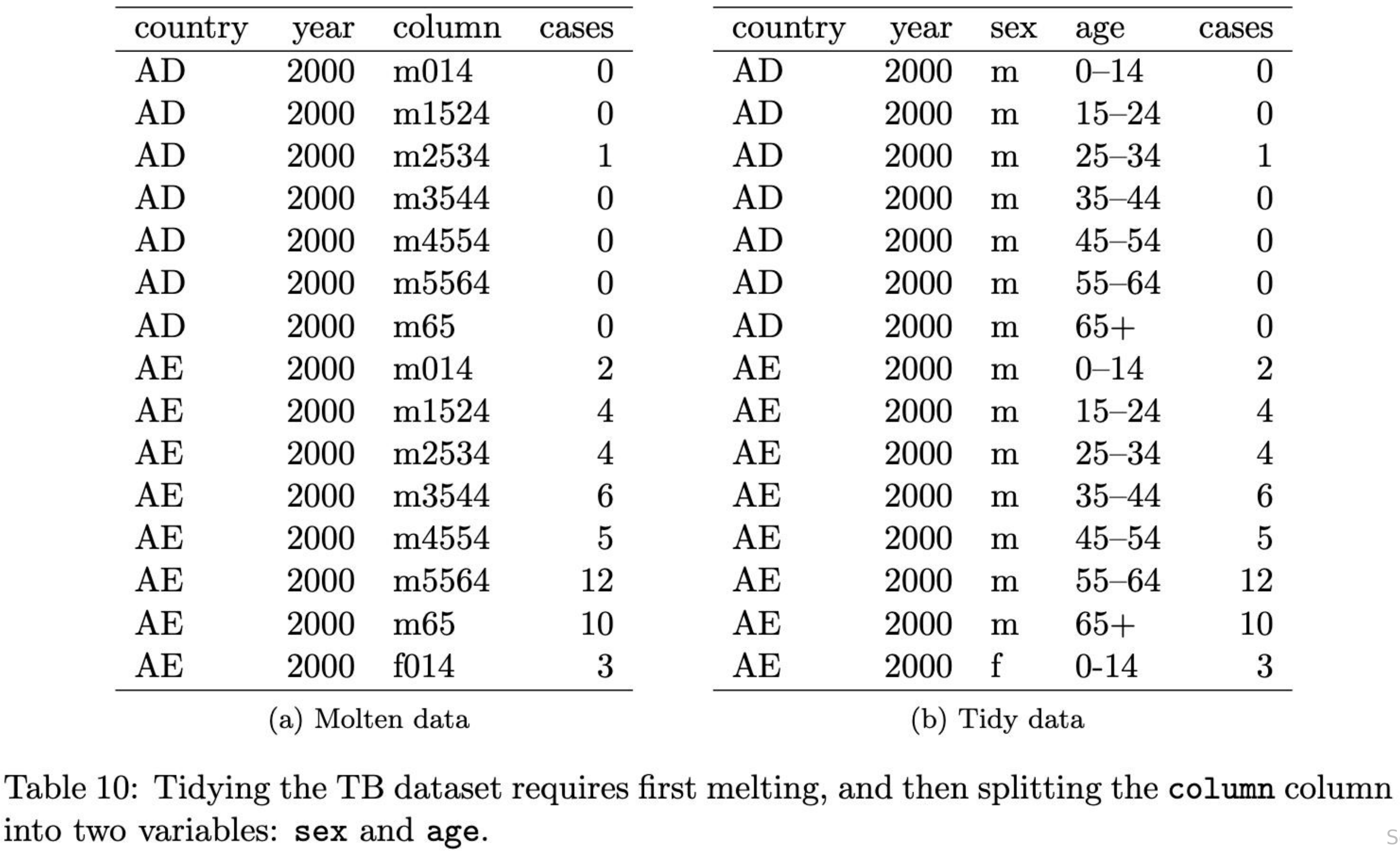
Is it tidy data?



country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

Question 2

Is it tidy data?



country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

(a) Molten data

country	year	sex	age	cases
AD	2000	m	0–14	0
AD	2000	m	15–24	0
AD	2000	m	25–34	1
AD	2000	m	35–44	0
AD	2000	m	45–54	0
AD	2000	m	55–64	0
AD	2000	m	65+	0
AE	2000	m	0–14	2
AE	2000	m	15–24	4
AE	2000	m	25–34	4
AE	2000	m	35–44	6
AE	2000	m	45–54	5
AE	2000	m	55–64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

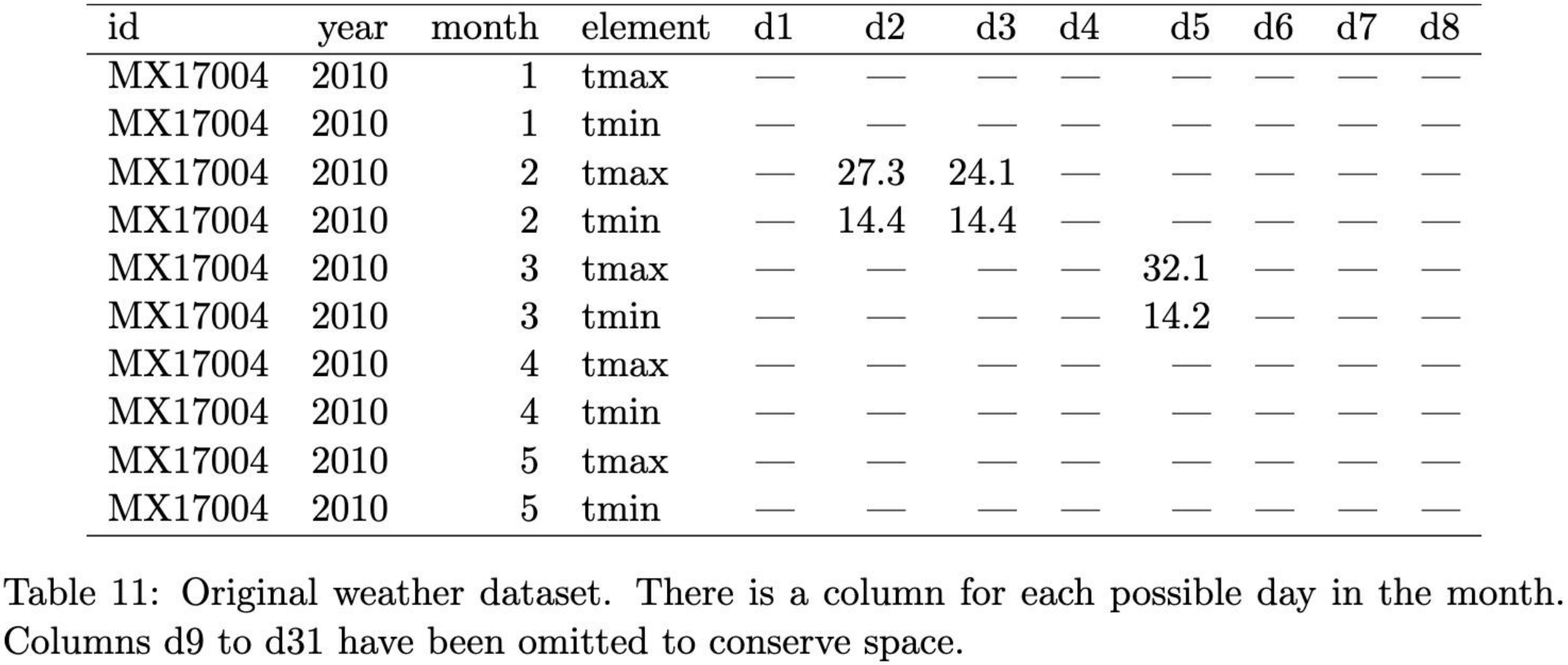
(b) Tidy data

Table 10: Tidying the TB dataset requires first melting, and then splitting the `column` column into two variables: `sex` and `age`.

Source images:
Wickham, H. (2014). Tidy Data.
Journal of Statistical Software,
59(10)

Question 3

Is it tidy data?



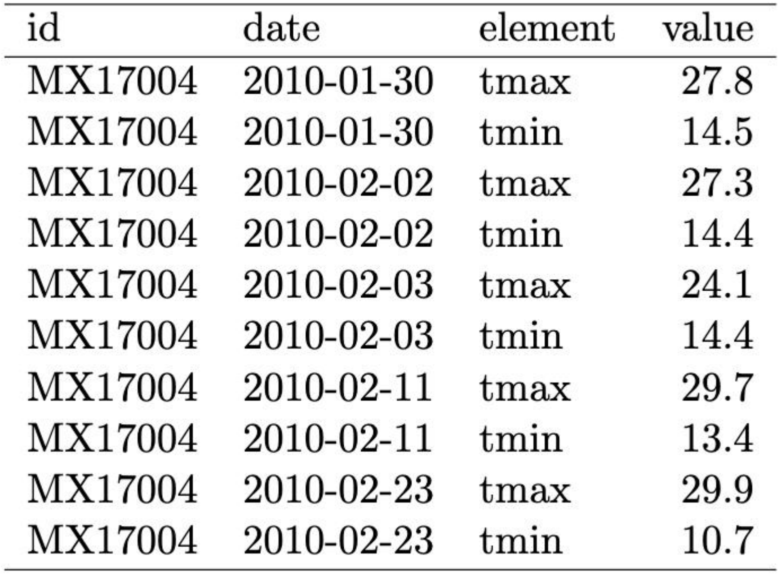
id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

Table 11: Original weather dataset. There is a column for each possible day in the month. Columns d9 to d31 have been omitted to conserve space.

Source images:
Wickham, H. (2014). Tidy Data.
Journal of Statistical Software,
50(10).

Question 3

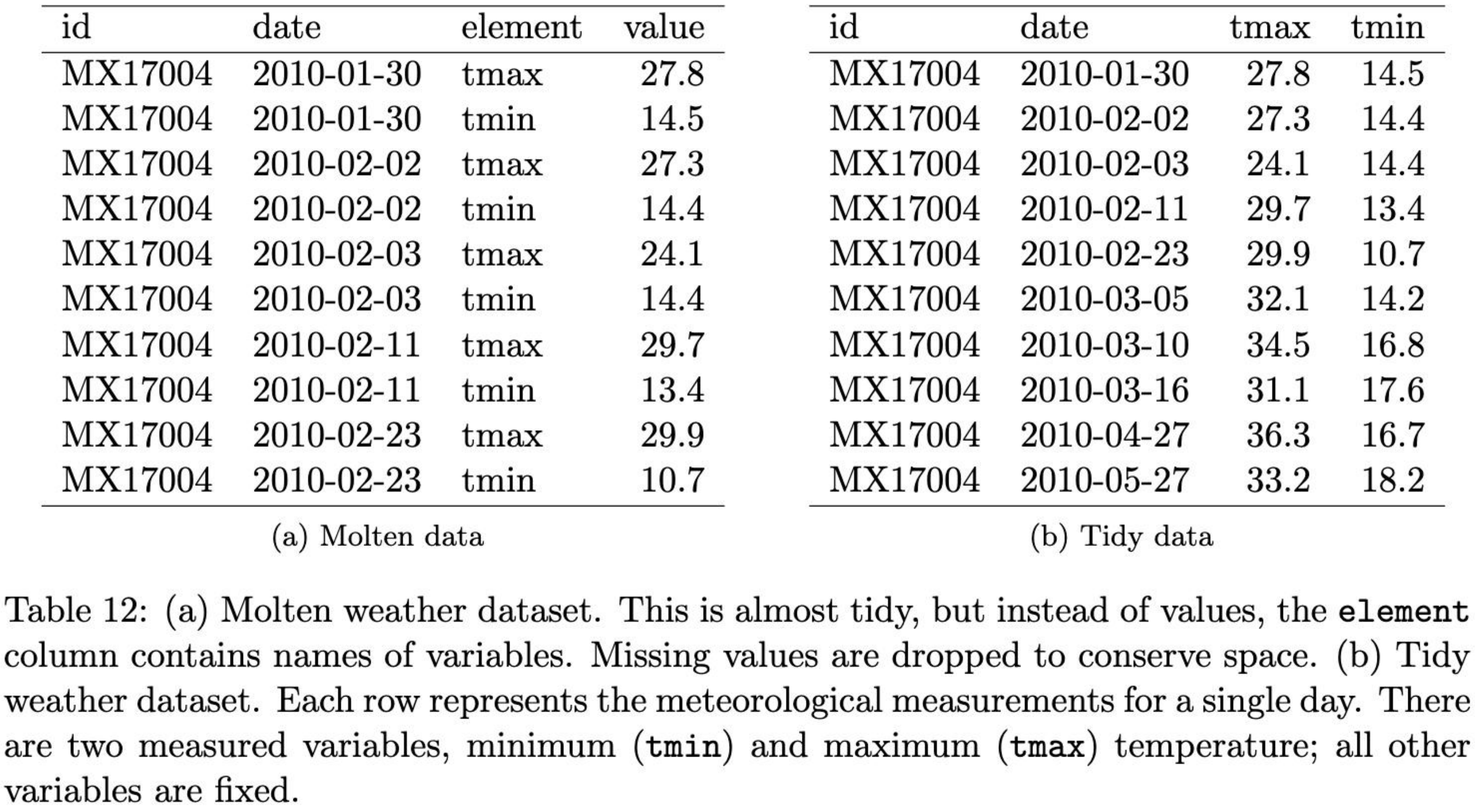
Is it tidy data?



id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

Question 3

Is it tidy data?



id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

(a) Molten data

id	date	tmax	tmin
MX17004	2010-01-30	27.8	14.5
MX17004	2010-02-02	27.3	14.4
MX17004	2010-02-03	24.1	14.4
MX17004	2010-02-11	29.7	13.4
MX17004	2010-02-23	29.9	10.7
MX17004	2010-03-05	32.1	14.2
MX17004	2010-03-10	34.5	16.8
MX17004	2010-03-16	31.1	17.6
MX17004	2010-04-27	36.3	16.7
MX17004	2010-05-27	33.2	18.2

(b) Tidy data

Table 12: (a) Molten weather dataset. This is almost tidy, but instead of values, the `element` column contains names of variables. Missing values are dropped to conserve space. (b) Tidy weather dataset. Each row represents the meteorological measurements for a single day. There are two measured variables, minimum (`tmin`) and maximum (`tmax`) temperature; all other variables are fixed.

Source images:
Wickham, H. (2014). Tidy Data.
Journal of Statistical Software,

Exercice

Define your ML project

Define your ML project

First, choose an application of Machine Learning. You can either choose:

- one covered in the course (except spam filtering)
- one that you are interested in (be creative!)
- to take inspiration from a list of applications: https://en.wikipedia.org/wiki/Machine_learning#Applications

Don't worry, you can be creative and try (almost) anything! There is no "wrong" answers.

Adapted from:
A. Burkov, “Machine Learning Engineering”.

Goal definition

Then, write down the **elements of your problem.**

Element	Description
Objective	The general goal of your system
Task	The task of the sub-system considered
Example	The unit of data fed one per one to the model
Output	The prediction of one exemple
Acceptable behaviors	Regarding the acceptable and unacceptable behaviors, think in terms of **performances, fairness, security, privacy, ethical issues, etc.** Explain only the topics that make sense for your problem. One or two sentences for each topic that makes sense.
How to measure these acceptable behaviors	
Unacceptable behaviors	
How to measure these unacceptable behaviors	

Goal definition

Example

Objective: be the best email provider

Task: automatically detect spam

Example: an email

Output: spam / ham

Acceptable behaviors:

- classify most of the spam, measured by probability of detection (sensitivity or recall) of at least 75%

Unacceptable behaviors:

- (performances) should only classify a very small amount of ham as spam, measured by specificity (true negative rate) of at least 95%
- (security) spammer should not fond easy way to fool the spam filter
- (privacy) email content should not be leaked from the trained model