UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

KECERDASAN BUATAN

2| DATA DALAM MACHINE LEARNING

Dr. Prima Dewi Purnamasari
Program Studi Teknik Komputer FTUI

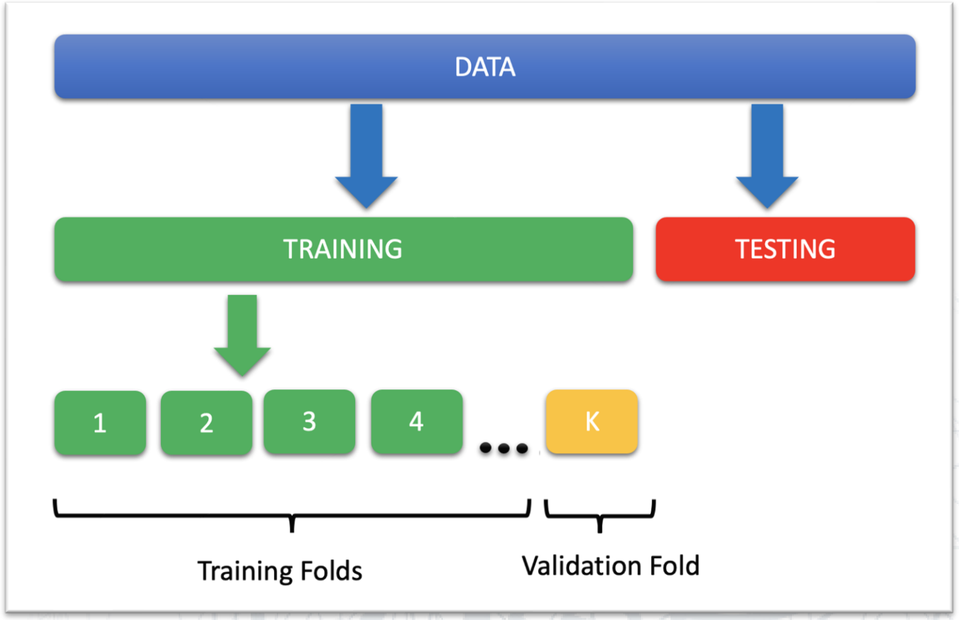
2| DATA DALAM MACHINE LEARNING

Sub CPMK 1.2 Mampu menjelaskan tahapan dalam machine learning (C3)

Materi Persiapan data, modelling, correctness, bias-variance, feature extraction & selection, training-testing

Referensi Theobald, ch. 2, 3, 4 & Gruss, ch 11

Indikator:
Mampu menjelaskan tahap persiapan data
Mampu menjelaskan tahap memodelkan machine learning
Mampu menjelaskan bias & variance
Mampu menjelaskan correctness
Mampu menjelaskan feature extraction & selection
Mampu menjelaskan strategi training, testing dan validation



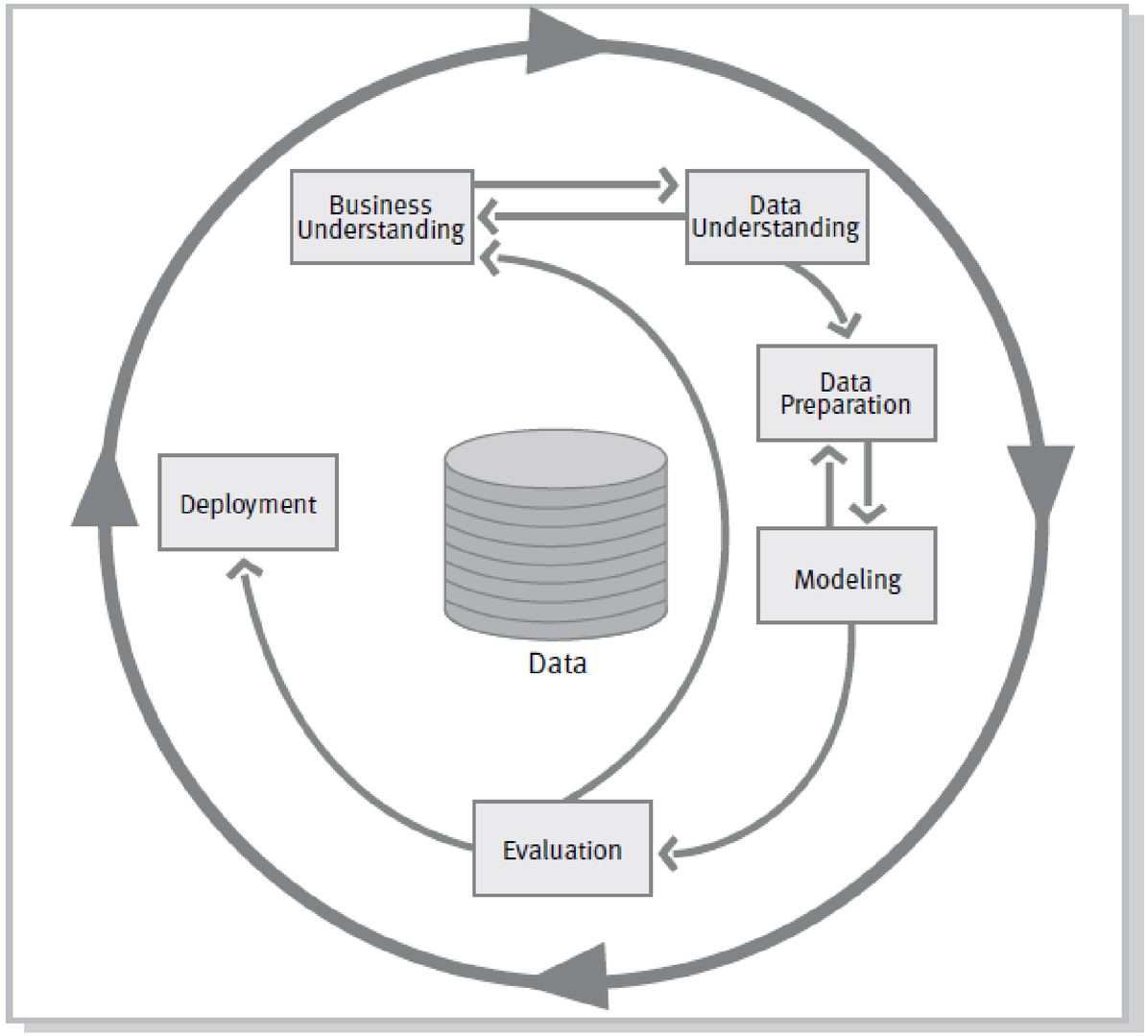
	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Contoh kasus, tabel dan gambar yang digunakan dalam materi ini diambil dari buku teks yang dijadikan referensi di mata kuliah ini, yaitu:

Oliver Theobald, **Machine Learning for Absolute Beginners: A Plain English Introduction,** Independently published, 2018, Chapter 4, 5, 6.

Joel Gruss, **Data Science from Scratch**, O'Reilly, 2015, Chapter 10

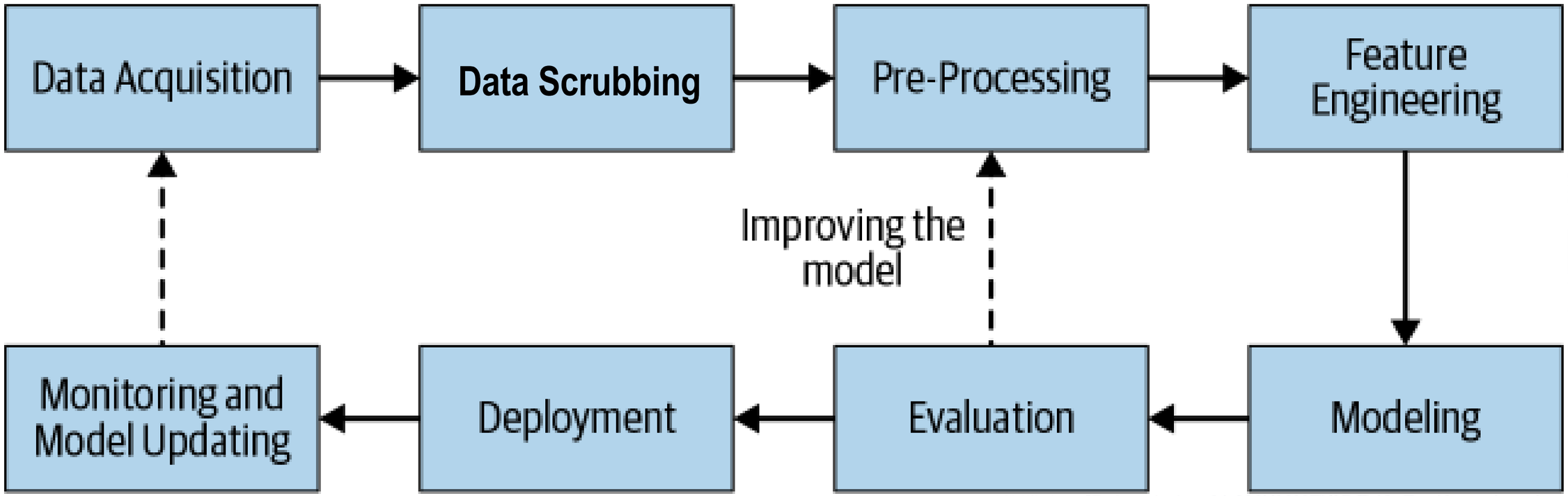
CRISP-DM Model



Source: Pete Chapman (NCR), Julian Clinton (SPSS), Randy Kerber (NCR),
Thomas Khabaza (SPSS), Thomas Reinartz (DaimlerChrysler),
Colin Shearer (SPSS) and Rüdiger Wirth (DaimlerChrysler), **CRISP-DM 1.0, Step-by-step data mining guide**

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives *Background* *Business Objectives* *Business Success Criteria* **Assess Situation** *Inventory of Resources* *Requirements, Assumptions, and Constraints* *Risks and Contingencies* *Terminology* *Costs and Benefits* **Determine Data Mining Goals** *Data Mining Goals* *Data Mining Success Criteria* **Produce Project Plan** *Project Plan* *Initial Assessment of Tools and Techniques*	**Collect Initial Data** *Initial Data Collection Report* **Describe Data** *Data Description Report* **Explore Data** *Data Exploration Report* **Verify Data Quality** *Data Quality Report*	**Select Data** *Rationale for Inclusion/ Exclusion* **Clean Data** *Data Cleaning Report* **Construct Data** *Derived Attributes* *Generated Records* **Integrate Data** *Merged Data* **Format Data** *Reformatted Data* *Dataset* *Dataset Description*	**Select Modeling Techniques** *Modeling Technique* *Modeling Assumptions* **Generate Test Design** *Test Design* **Build Model** *Parameter Settings* *Models* *Model Descriptions* **Assess Model** *Model Assessment* *Revised Parameter Settings*	**Evaluate Results** *Assessment of Data Mining Results w.r.t. Business Success Criteria* *Approved Models* **Review Process** *Review of Process* **Determine Next Steps** *List of Possible Actions* *Decision*	**Plan Deployment** *Deployment Plan* **Plan Monitoring and Maintenance** *Monitoring and Maintenance Plan* **Produce Final Report** *Final Report* *Final Presentation* **Review Project** *Experience Documentation*

Pipeline


Data Acquisition
Data Scrubbing
Pre-Processing
Feature Engineering
Improving the model
Monitoring and Model Updating
Deployment
Evaluation
Modeling

Data scrubbing

DATA SCRUBBING = DATA CLEANING

- Like most varieties of fruit, datasets need upfront cleaning and human manipulation before they are ready for consumption.
- The technical process of refining your dataset to make it more workable.
- Modifying and removing incomplete, incorrectly formatted, irrelevant or duplicated data, converting text-based data to numeric values and the redesigning of features.
- For data practitioners, data scrubbing typically demands the greatest application of time and effort. ➔ 79 – 90%

FEATURE SELECTION

- Identify which variables are most relevant to your hypothesis or objective.
- Selective in choosing the variables you include in your model.
  - Rather than creating a four-dimensional scatterplot with four features in your model, an opportunity may present to select two highly relevant features and build a two-dimensional plot that is easier to interpret and visualize.
- Preserving features that don’t correlate strongly with the output value can manipulate and derail the model’s accuracy.

Example: identify variables that contribute to a language becoming endangered

Name in English	Name in Spanish	Countries	Country Code	Num. of Speakers
South Italian	Napolitano -calabres	Italy	ITA	7500000
Sicilian	Siciliano	Italy	ITA	5000000
Low Saxon	Bajo Sajón	Germany, Denmark, Netherlands, Poland, Russian Federation	DEU, DNK, NLD, POL, RUS	4800000
Belarusian	Bielorruso	Belarus, Latvia, Lithuania, Poland, Russian Federation, Ukraine	BRB, LVA, LTU, POL, RUS, UKR	4000000
Lombard	Lombardo	Italy, Switzerland	ITA, CHE	3500000
Romani	Romaní	Albania, Germany, Austria, Belarus, Bosnia and Herzegovina, Bulgaria, Croatia, Estonia, Finland, France, Greece, Hungary, Italy, Latvia, Lithuania, The former Yugoslav Republic of Macedonia, Netherlands, Poland, Romania, United Kingdom of Great Britain and Northern Ireland, Russian Federation, Slovakia, Slovenia, Switzerland, Czech Republic, Turkey, Ukraine, Serbia, Montenegro	ALB, DEU, AUT, BRB, BIH, BGR, HRV, EST, FIN, FRA, GRC, HUN, ITA, LVA, LTU, MKD, NLD, POL, ROU, GBR, RUS, SVK, SVN, CHE, CZE, TUR, UKR, SRB, MNE	3500000
Yiddish	Yiddish	Israel	ISR	3000000
Gondi	Gondi	India	IND	2713790

Table 3: Endangered languages, database: https://www.kaggle.com/the-guardian/extinct- languages

Scrubbing the data

- The language's "Name in Spanish" may not contribute to any insight → delete this vector (column) from the dataset.

- The dataset contains duplicated information in the form of separate vectors for "Countries" and "Country Code." Analyzing both of these vectors doesn't provide any additional insight; hence, we can choose to delete one and retain the other.

Name in English	Name in Spanish	Countries	Country Code	Num. of Speakers
South Italian	Napolitano - Calabrés	Italy	ITA	7500000
Sicilian	Siciliano	Italy	ITA	5000000
Low Saxon	Bajo Sajón	Germany, Denmark, Netherlands, Poland, Russian Federation	DEU, DNK, NLD, POL, RUS	4800000
Belarusian	Bielorruso	Belarus, Latvia, Lithuania, Poland, Russian Federation, Ukraine	BRB, LVA, LTU, POL, RUS, UKR	4000000
Lombard	Lombardo	Italy, Switzerland	ITA, CHE	3500000
Romani	Romaní	Albania, Germany, Austria, Belarus, Bosnia and Herzegovina, Bulgaria, Croatia, Estonia, Finland, France, Greece, Hungary, Italy, Latvia, Lithuania, The former Yugoslav Republic of Macedonia, Netherlands, Poland, Romania, United Kingdom of Great Britain and Northern Ireland, Russian Federation, Slovakia, Slovenia, Switzerland, Czech Republic, Turkey, Ukraine, Serbia, Montenegro	ALB, DEU, AUT, BRB, BIH, BGR, HRV, EST, FIN, FRA, GRC, HUN, ITA, LVA, LTU, MKD, NLD, POL, ROU, GBR, RUS, SVK, SVN, CHE, CZE, TUR, UKR, SRB, MNE	3500000
Yiddish	Yiddish	Israel	ISR	3000000
Gondi	Gondi	India	IND	2713790

FEATURE REDUCTION (DIMENSION REDUCTION)

- Another method: roll multiple features into one

	Protein Shake	Nike Sneakers	Adidas Boots	Fitbit	Powerade	Protein Bar	Fitness Watch	Vitamins
Buyer 1	1	1	0	1	0	5	1	0
Buyer 2	0	0	0	0	0	0	0	1
Buyer 3	3	0	1	0	5	0	0	0
Buyer 4	1	1	0	0	10	1	0	0

Table 4: Sample product inventory

MERGING SIMILAR FEATURES

- Reduce the number of columns by merging similar features into fewer columns.
- Remove individual product names and replace the eight product items with a lower number of categories or subtypes.
- As all product items fall under the category of “fitness,” we can sort by product subtype and compress the columns from eight to three: “Health Food,” “Apparel,” and “Digital.”

	Protein Shake	Nike Sneakers	Adidas Boots	Fitbit	Powerade	Protein Bar	Fitness Watch	Vitamins
Buyer 1	1	1	0	1	0	5	1	0
Buyer 2	0	0	0	0	0	0	0	1
Buyer 3	3	0	1	0	5	0	0	0
Buyer 4	1	1	0	0	10	1	0	0

	Health Food	Apparel	Digital
Buyer 1	6	1	2
Buyer 2	1	0	0
Buyer 3	8	1	0
Buyer 4	12	1	0

Buyers will be recommended health food when they buy other health food or when they buy apparel (depending on the degree of correlation)

Synthesized product inventory

- **Pros:** transform the dataset in a way that preserves and captures information using fewer variables.
- **Cons:** less information about the relationships between specific products.

- Trade-off between convenience and the overall precision of the model.

Row Compression

- Reduce the number of rows and thereby compress the total number of data points.
- This may involve merging two or more rows into one,

Before

Animal	Meat Eater	Legs	Tail	Race Time
Tiger	Yes	4	Yes	2:01 mins
Lion	Yes	4	Yes	2:05 mins
Tortoise	No	4	No	55:02 mins

After

Animal	Meat Eater	Legs	Tail	Race Time
Carnivore	Yes	4	Yes	2:03 mins
Tortoise	No	4	No	55:02 mins

Table 6: Example of row merge

it's possible to merge the two rows because they possess the same categorical values for all features except Race Time—which can be easily aggregated.
The race time of the Tiger and the Lion can be added and divided by two.

- Numeric values are normally easy to aggregate given they are not categorical.
  - Beware: it would be impossible to aggregate an animal with four legs and an animal with two legs! We obviously can't merge these two animals and set "three" as the aggregate number of legs.
- Row compression can also be challenging to implement in cases where numeric values aren't available.
- Row compression is usually less attainable than feature compression and especially for datasets with a high number of features.

ONE HOT ENCODING

- Aside from set text-based values such as True/False (that automatically convert to “1” and “0” respectively), most algorithms are not compatible with non- numeric data.
- Transforms values into binary form,
  - “1” or “0”—“True” or “False.”
  - “0,” representing False, means that the value does not belong to this particular feature,
  - “1”—True or “hot”— confirms that the value does belong to this feature.

Name in English	Speakers	Degree of Endangerment
South Italian	7500000	Vulnerable
Sicilian	5000000	Vulnerable
Low Saxon	4800000	Vulnerable
Belarusian	4000000	Vulnerable
Lombard	3500000	Definitely endangered
Romani	3500000	Definitely endangered
Yiddish	3000000	Definitely endangered
Gondi	2713790	Vulnerable
Picard	700000	Severely endangered

Table 7: Endangered languages

Name in English	Speakers	Vulnerable	Definitely Endangered	Severely Endangered
South Italian	7500000	1	0	0
Sicilian	5000000	1	0	0
Low Saxon	4800000	1	0	0
Belarusian	4000000	1	0	0
Lombard	3500000	0	1	0
Romani	3500000	0	1	0
Yiddish	3000000	0	1	0
Gondi	2713790	1	0	0
Picard	700000	0	0	1

Table 8: Example of one-hot encoding

- **Pros:** Many of machine learning algorithms “likes” this kind of input.
- **Cons:** more dataset features, which may slightly extend processing time
- One hack to minimize the total number of features is to restrict binary cases to a single column.
  - Aspeed dating dataset on kaggle.com lists “Gender” in a single column using one-hot encoding. Rather than create discrete columns for both “Male” and “Female,” they merged these two features into one. According to the dataset’s key, females are denoted as “0” and males as “1.” The creator of the dataset also used this technique for “Same Race” and “Match.”

Subject Number ID	Gender	Same Race	Age	Match
1	0	0	27	0
1	0	0	22	0
1	0	1	22	1
1	0	0	23	1
1	0	0	24	1
1	0	0	25	0
1	0	0	30	0

Table 9: Speed dating results, database: https://www.kaggle.com/annavictoria/speed-dating- experiment

Gender:	Same Race:	Match:
Female = 0	No = 0	No = 0
Male = 1	Yes = 1	Yes = 1

BINNING

- Binning is another method of feature engineering but is used to convert numeric values into a category.
- Whoa, hold on! Aren't numeric values a good thing? Yes, in most cases numeric values are preferred as they are compatible with a broader selection of algorithms. Where numeric values are not ideal, is in situations where they list variations irrelevant to the goals of your analysis.
- Let's take house price evaluation as an example. The exact measurements of a tennis court might not matter greatly when evaluating house prices; the relevant information is whether the house has a tennis court. This logic probably also applies to the garage and the swimming pool, where the existence or non-existence of the variable is generally more influential than their specific measurements.
- The solution here is to replace the numeric measurements of the tennis court with a True/False feature or a categorical value such as "small," "medium," and "large." Another alternative would be to apply one-hot encoding with "0" for homes that do not have a tennis court and "1" for homes that do have a tennis court.

Missing Data

- Missing values in your dataset can be equally frustrating and interfere with your analysis and the model's predictions.
- Approximate the missing value by mode value or median value
- **Mode value:** represents the single most common variable value available in the dataset.
  - This works best with categorical and binary variable types, such as one to five-star rating systems and positive/negative drug tests respectively.
- **Median value:** adopts the value(s) located in the middle of the dataset.
  - This works best with continuous variables, which have an infinite number of possible values, such as house prices.
- As a last resort, rows with missing values can be removed altogether.

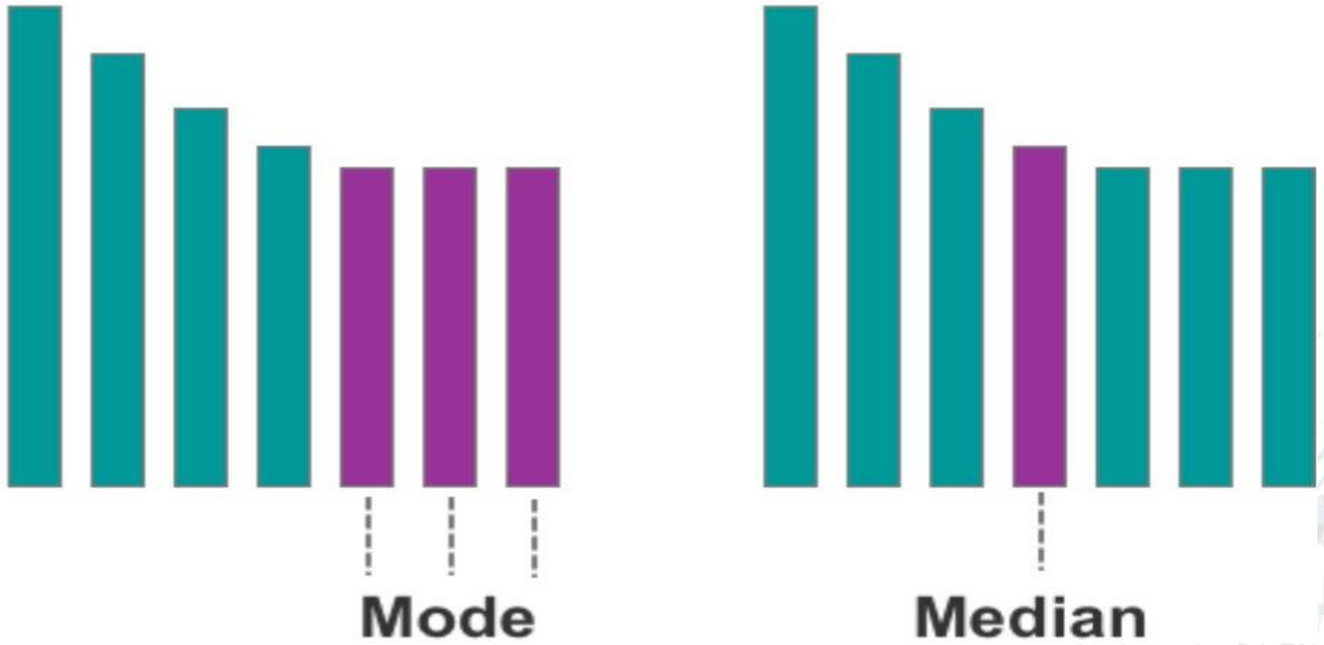


Figure 10: A visual example of the mode and median respectively

UNIVERSITAS
INDONESIA
Veritas, Probitas, Iustitia

Setting up data

Split validation

	Variable 1	Variable 2	Variable 3
Row 1			
Row 2			
Row 3			
Row 4			
Row 5			
Row 6			
Row 7			
Row 8			
Row 9			
Row 10			

Training Data

Test Data

- The ratio of the two splits should be approximately 70/30 or 80/20.
- First: randomize the row order
  - This helps to avoid bias in your model, as your original dataset might be arranged alphabetically or sequentially according to when the data was collected.

Figure 11: 70/30 partitioning of training and test data

Cross Validation

- Exhaustive cross validation
  - Finding and testing all possible combinations to divide the original sample into a training set and a test set.
- K-fold validation.
  - Splitting data into k assigned buckets and reserving one of those buckets for testing the training model at each round.
  - Data are randomly assigned .
  - One bucket is reserved as the test bucket and is used to measure and evaluate the performance of the remaining (k-1) buckets.

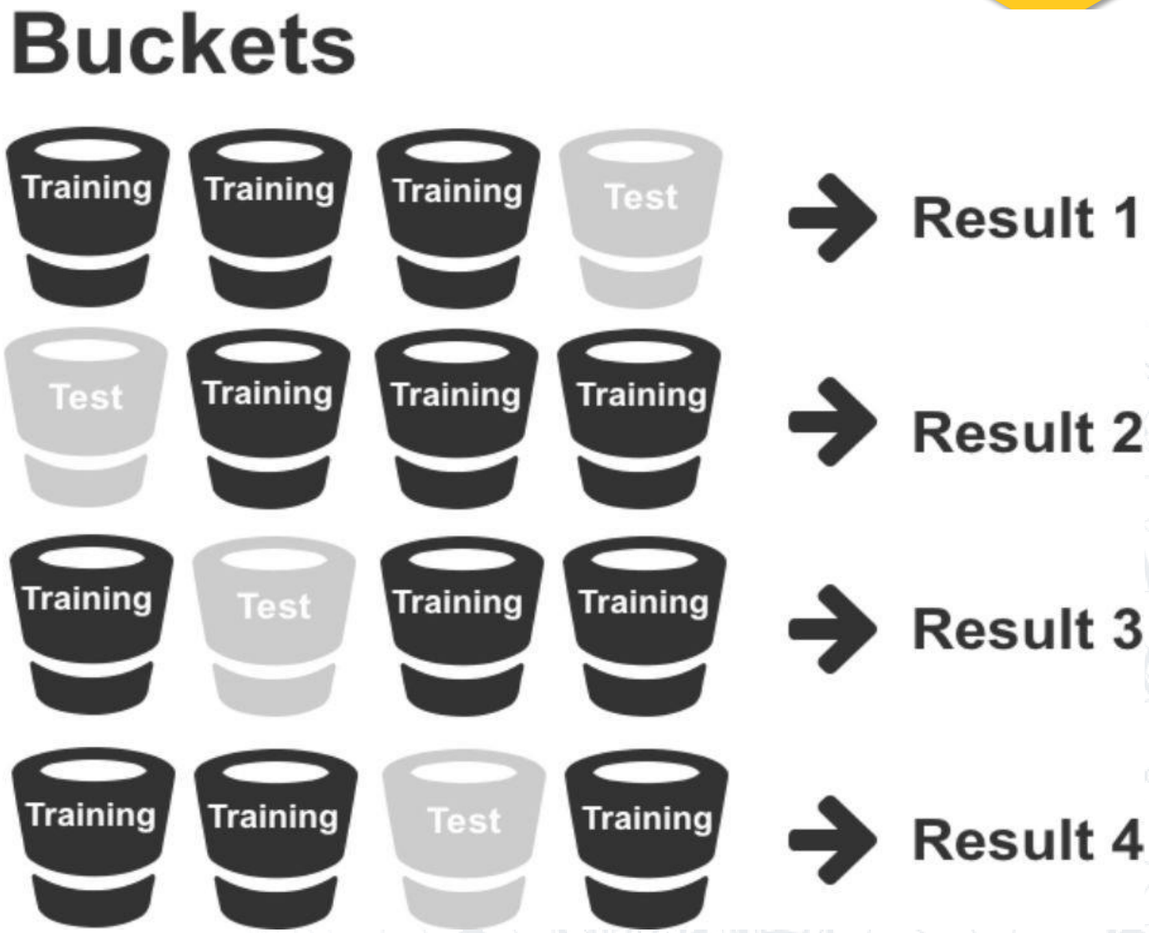


Figure 12: *k*-fold validation

How Much Data Do I Need?

- At minimum ∑samples ≥10 x features
  - E.g. for a small dataset with 5 features, the training data should ideally have at least 50 rows.
- Generally, the more relevant data you have available as training data, the more combinations you can incorporate into your prediction model
- In general, machine learning works best when your training dataset includes a full range of feature combinations.

What does a full range of feature combinations look like?

- Imagine you have a dataset about data scientists. The features :
  - University degree (X)
  - 5+ years of professional experience (X)
  - Children (X)
  - Salary (y)
- We need to know the salary for data scientists with a university degree and 5+ years of professional experience who don’t have children, as well as data scientists with a university degree and 5+ years of professional experience who do have children.

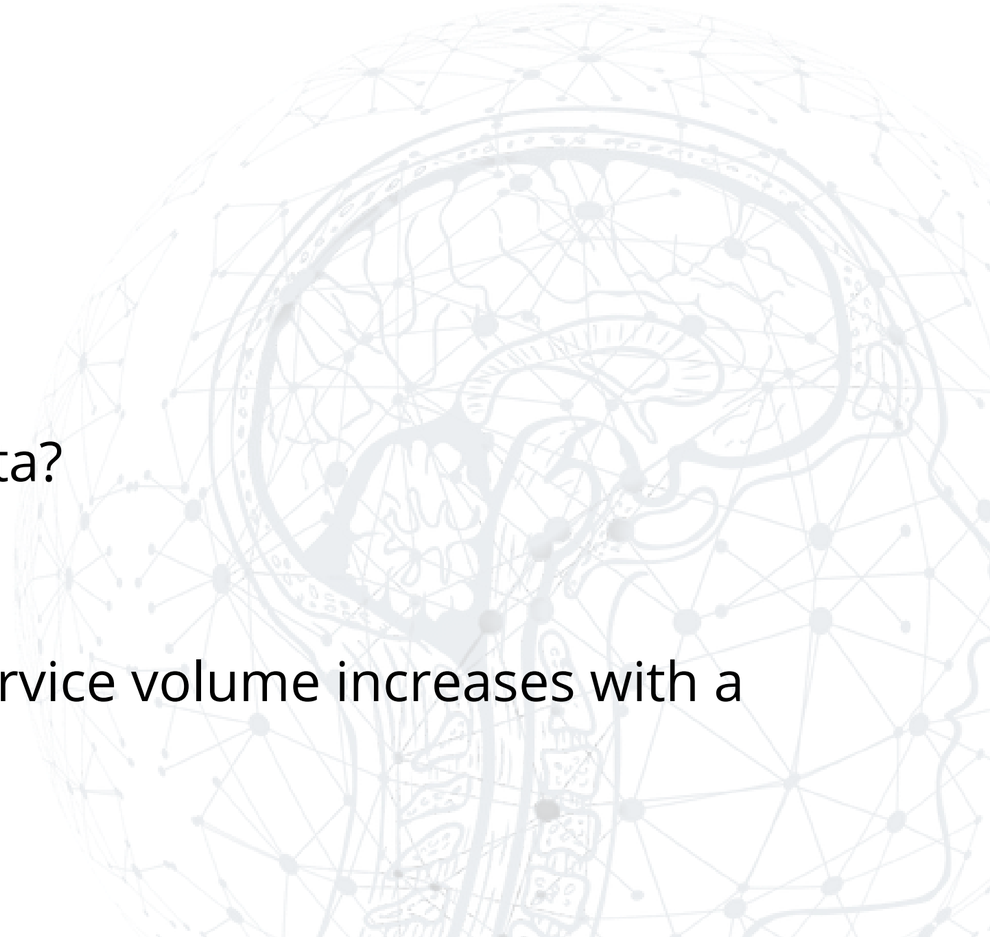
Matching data to an algorithm

- For datasets with less than 10,000 samples, clustering and dimensionality reduction algorithms can be highly effective
- Regression analysis and classification algorithms are more suitable for datasets with less than 100,000 samples.
- Neural networks require even more samples to run effectively and are more cost-effective and time-efficient for working with massive quantities of data.

Post analysis data

What is a good model?

- Generalization capability
  - Can it accurately predict the actual service data?
- Interpretability
  - Is the prediction result easy to interpret?
- Prediction speed
  - How long does it take to predict each piece of data?
- Practicability
  - Is the prediction rate still acceptable when the service volume increases with a huge data volume?

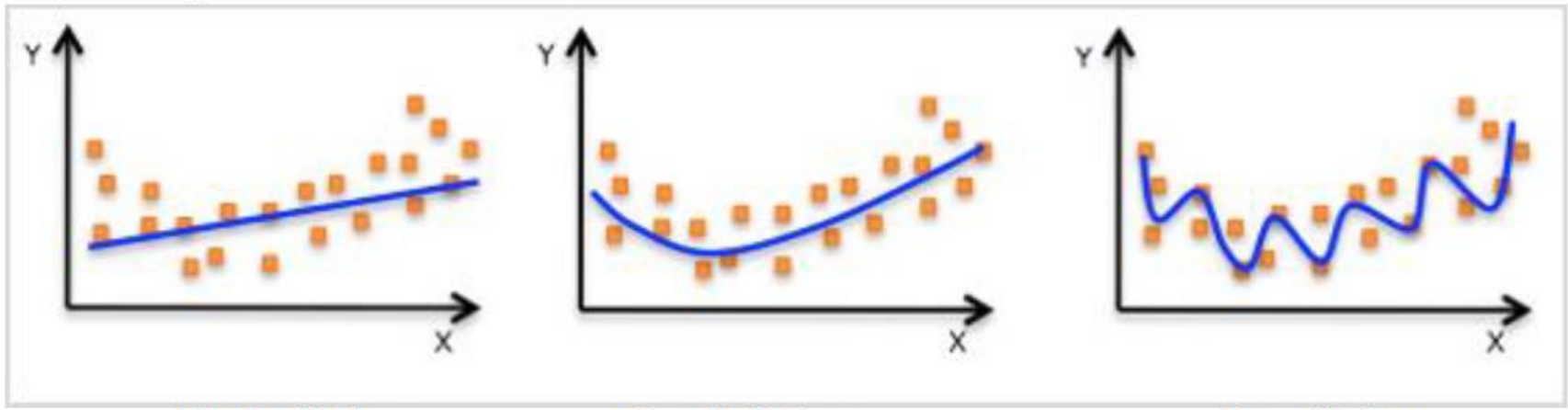


Error

- Difference between the sample result predicted by the model obtained after learning and the actual sample result.
- **Training error:** error that you get when you run the model on the training data.
- **Generalization error (testing error):** error that you get when you run the model on new samples. Obviously, we prefer a model with a smaller generalization error.

Underfitting vs Overfitting

- Underfitting: occurs when the model or the algorithm does not fit the data well enough.
- Overfitting: occurs when the training error is small but the generalization error is large (poor generalization capability).



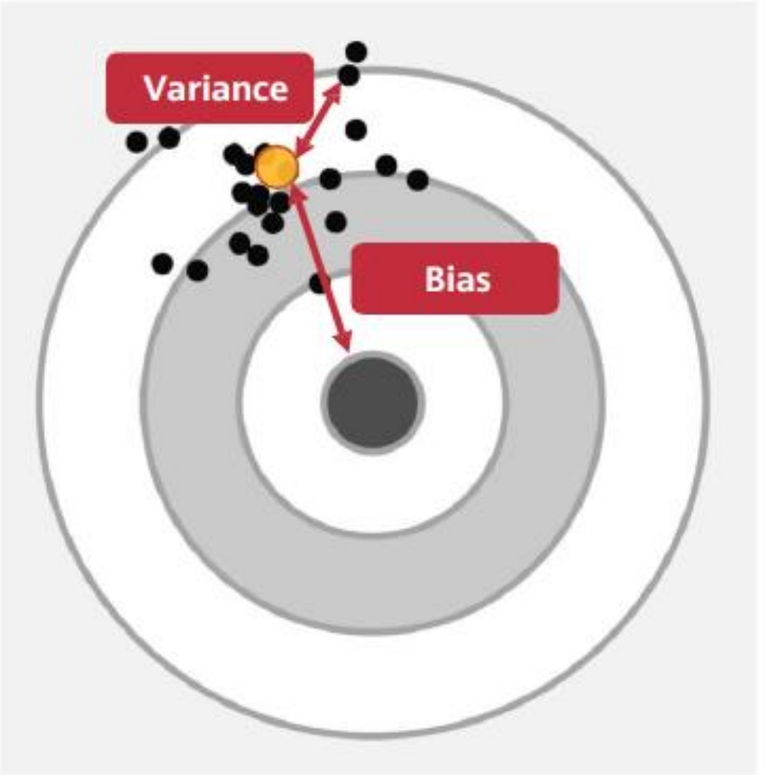
Underfitting
Not all features are learned.

Good fitting

Overfitting
Noises are learned.

Overfitting Cause —Error

- Generally,the prediction error can be divided into two types:
  - Error caused by "bias"
  - Error caused by "variance"
- Variance:
  - Offset of the prediction result from the average value
  - Error caused by the model's sensitivity to small fluctuations in the training set
- Bias:
  - Difference between the expected (or average) prediction value and the correct value we are trying to predict.



Bias – Variance

- If your model has high bias (which means it performs poorly even on your training data) then one thing to try is adding more features. Going from the degree 0 model in to the degree 1 model was a big improvement.
- If your model has high variance, then you can similarly remove features. But another solution is to obtain more data (if you can).

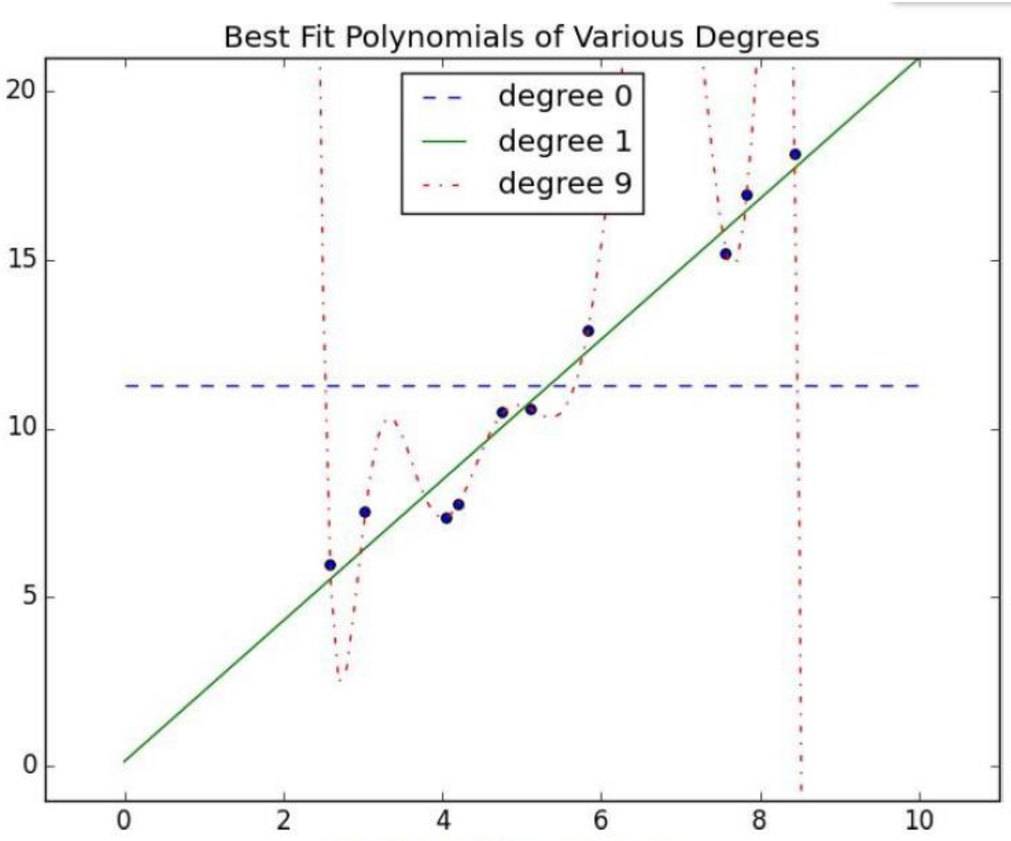
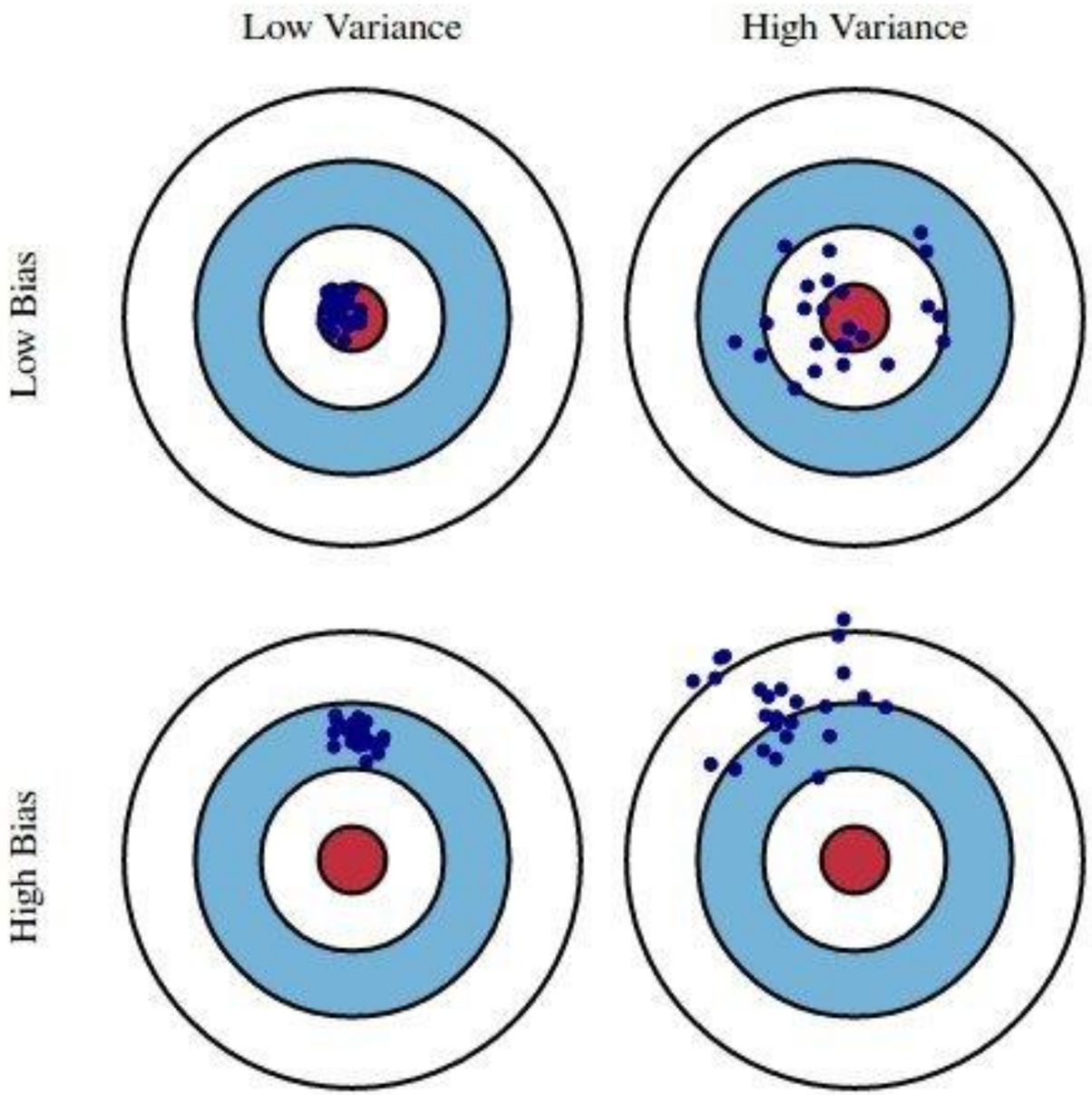
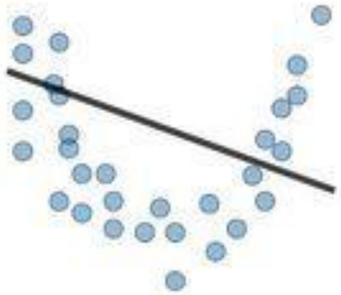
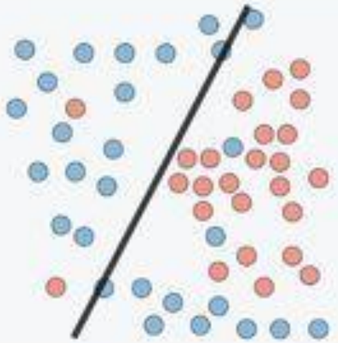
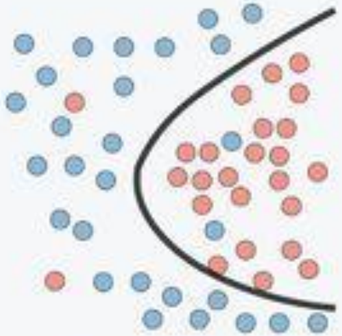
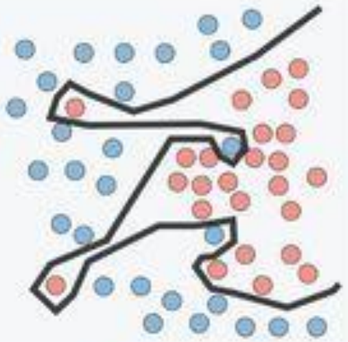
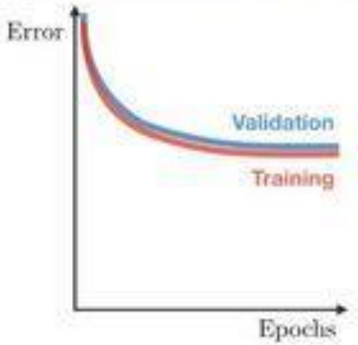
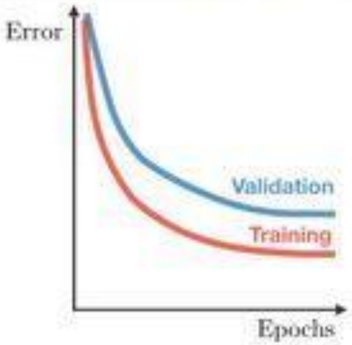
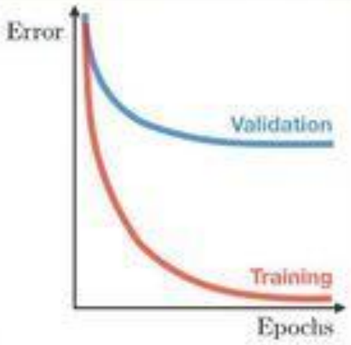


Figure 11-1. Overfitting and underfitting


Low Variance
High Variance
Low Bias
High Bias

	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

Correctness

Correctness

- True positive:
  - “This message is spam, and we correctly predicted spam.”
- False positive (Type 1 Error):
  - “This message is not spam, but we predicted spam.”
- False negative (Type 2 Error):
  - “This message is spam, but we predicted not spam.” True negative: “This message is not spam, and we correctly predicted not spam.”

	Spam	not Spam
predict “Spam”	True Positive	False Positive
predict “Not Spam”	False Negative	True Negative

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Confusion Matrix

Classifier metric (for binary classification)

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

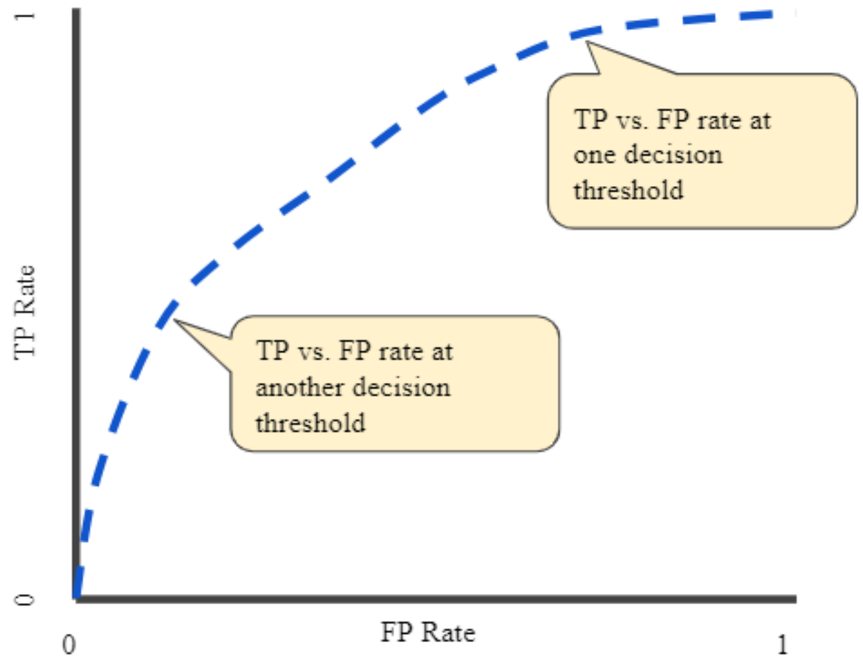
$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$specificity = \frac{TN}{TN + FP}$$

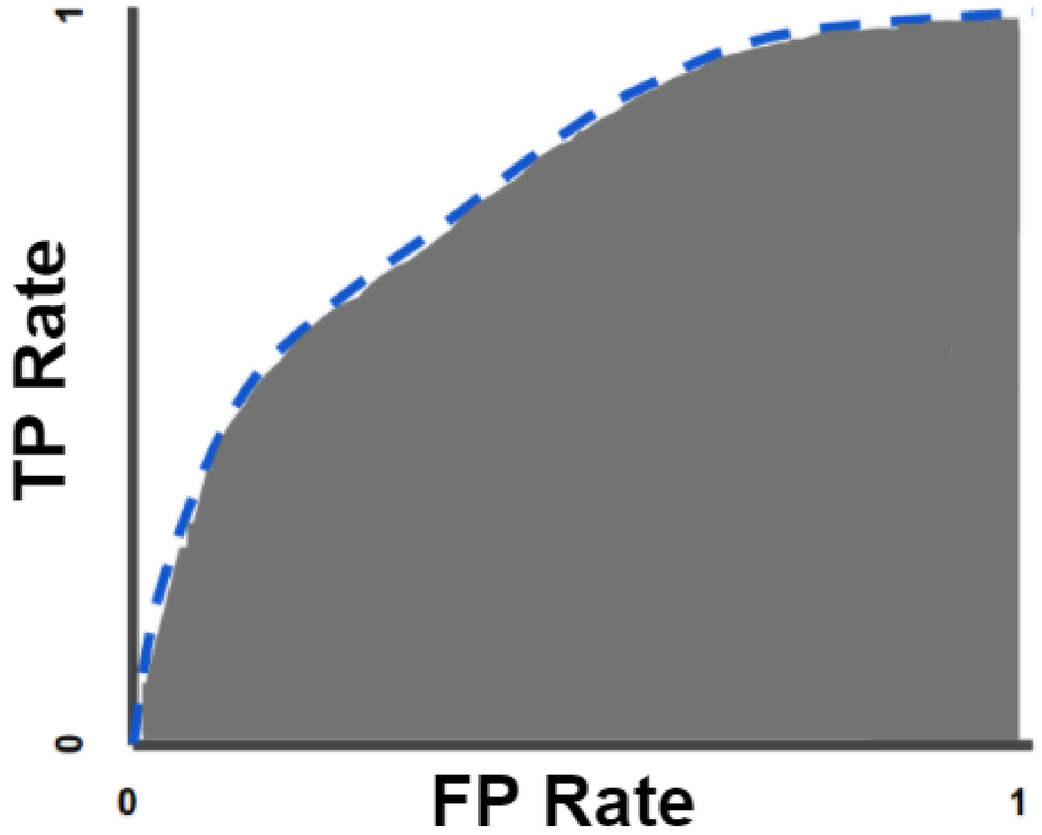
ROC curve

- An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds.
-



AUC: Area Under the ROC Curve

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example



Today’s hands on

https://www.kaggle.com/learn/data-cleaning