CMU Portugal Academy

CMU Portugal
Advanced Training Program

Foundations of Data Science

DAVID SEMEDO

RAFAEL FERREIRA

NOVA SCHOOL OF SCIENCE AND TECHNOLOGY

Carnegie Mellon Portugal · TÉCNICO LISBOA · Ciências ULisboa · NOVA School of Science & Technology

Email: df.semedo@fct.unl.pt

CMU Portugal Academy

Today's Topics

1. INTRO TO MACHINE LEARNING

2. FEATURE ENGINEERING

3. MODEL EVALUATION

4. SUPERVISED LEARNING
   - Linear Models
   - Classification
   - Regression

Use-Case Details

Requirements - One operation of each of the following:

- Dataset Descriptive Statistics
- Data Cleaning (e.g. checking for NaNs, column removal, etc.)
- Model Selection, Feature Engineering, and Normalization
- Plotting (frequency, correlation between feature pairs)
- Supervised Learning:
  - Training a linear classifier
  - Evaluate its performance over multiple metrics

Use-Case Discussion Session

Let's simulate a Data Science team discussion:

- Bring your expertise and point of view!

We Invite all groups to present and discuss their use-case with the class:

- Show and discuss your notebook to the class
- 5 to 7 minutes per presentation

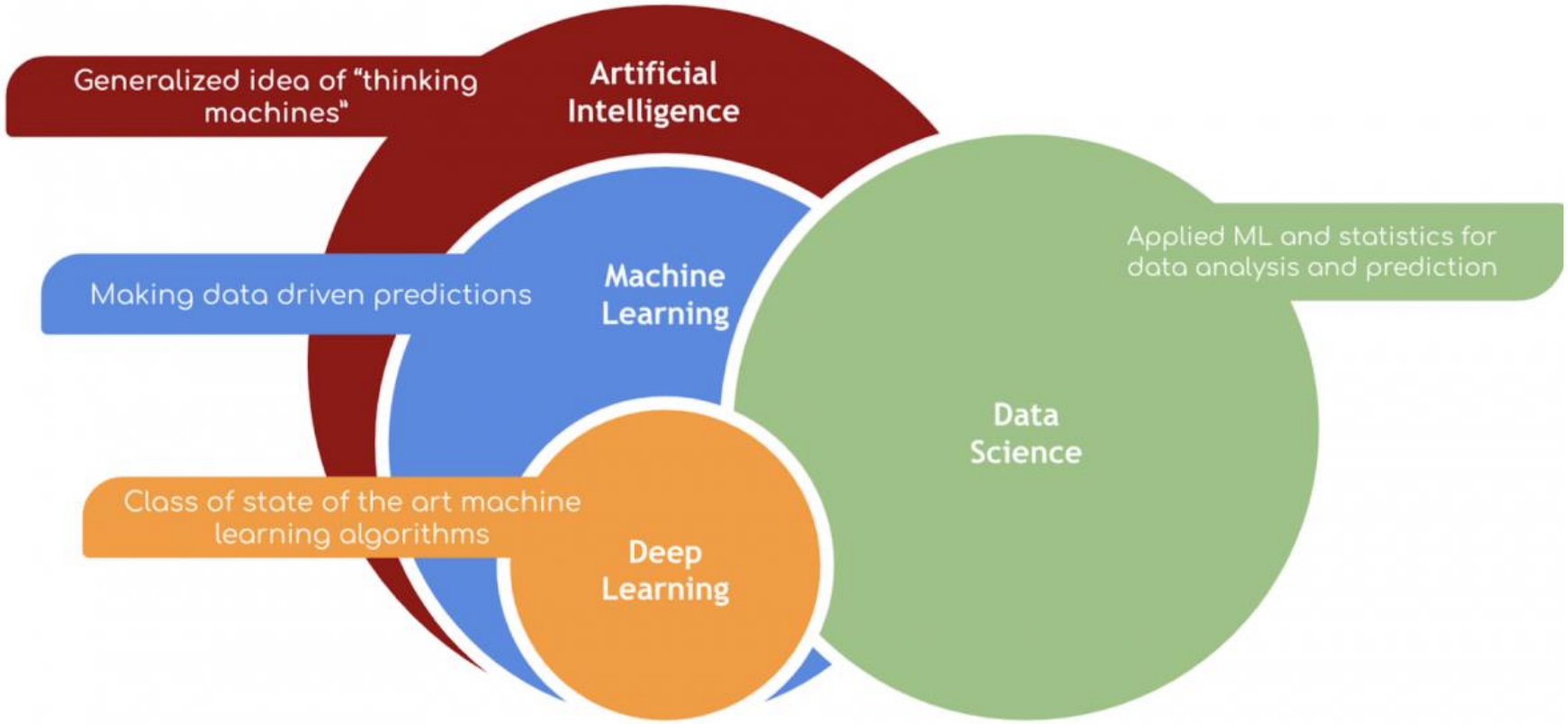
CMU Portugal Academy

01

Intro to Machine Learning

Carnegie Mellon Portugal · TÉCNICO LISBOA · Ciências ULisboa · NOVA SCHOOL OF SCIENCE & TECHNOLOGY

What is machine learning?



What is Machine Learning?

- [Arthur Samuel, 1959]
  - Field of study that gives computers the ability to learn without being explicitly programmed

- [Kevin Murphy] algorithms that
  - automatically detect patterns in data
  - use the uncovered patterns to predict future data or other outcomes of interest

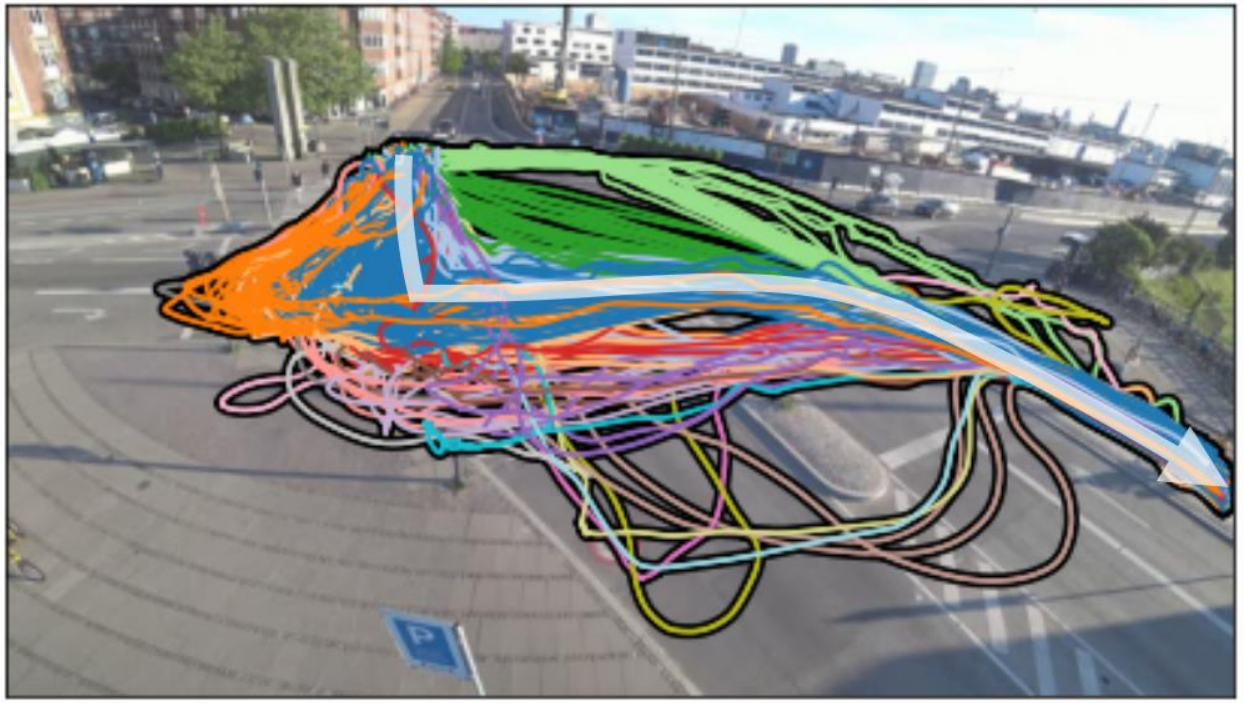
- [Tom Mitchell] algorithms that
  - improve their performance (P)
  - at some task (T)
  - with experience (E)

CMU Portugal Academy

Why do we need learning?

Design

Reality



Breum, Kostic & Szell. Computational Desire Line Analysis of Cyclists on the Dybbølsbro Intersection in Copenhagen, Transport Findings 56683 (2022)

ML in a (tiny)Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year

- Decades of ML research oversimplified:
  - All of Machine Learning:
  - Learn a mapping from input to output f: X → Y
  - X: emails, Y: {spam, notspam}

ML in a Nutshell

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)
- (Unknown) Target Function
  - f: X → Y (the "true" mapping / reality)
- Data
  - $(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)$
- Model / Hypothesis Class
  - g: X → Y
  - $y = g(x) = \text{sign}(w^T x)$

Machine Learning Components

Machine Learning Models

- Decision trees
- Sets of rules / Logic programs
- Instance-based Models
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- And many others

You will cover these in detail in the **Machine Learning** Module

Optimization

- Discrete/Combinatorial optimization
  - Greedy search
  - Graph algorithms (cuts, flows, etc)

- Continuous optimization
  - Convex/Non-convex optimization (gradient descent)
  - Linear programming

Evaluation / Objective Function

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Source: Pedro Domingos

Learning Settings

- Supervised learning
  - Training data includes desired outputs
- Unsupervised learning
  - Training data does not include desired outputs
- Weakly or Semi-supervised learning
  - Training data includes a few desired outputs
- Self-Supervised Learning
  - Model supervises itself
- Reinforcement learning
  - Rewards from sequence of actions

CMU Portugal Academy

Supervised Learning Example

Credit Card Fraud Detection

Binary classification (Fraud, Not Fraud)

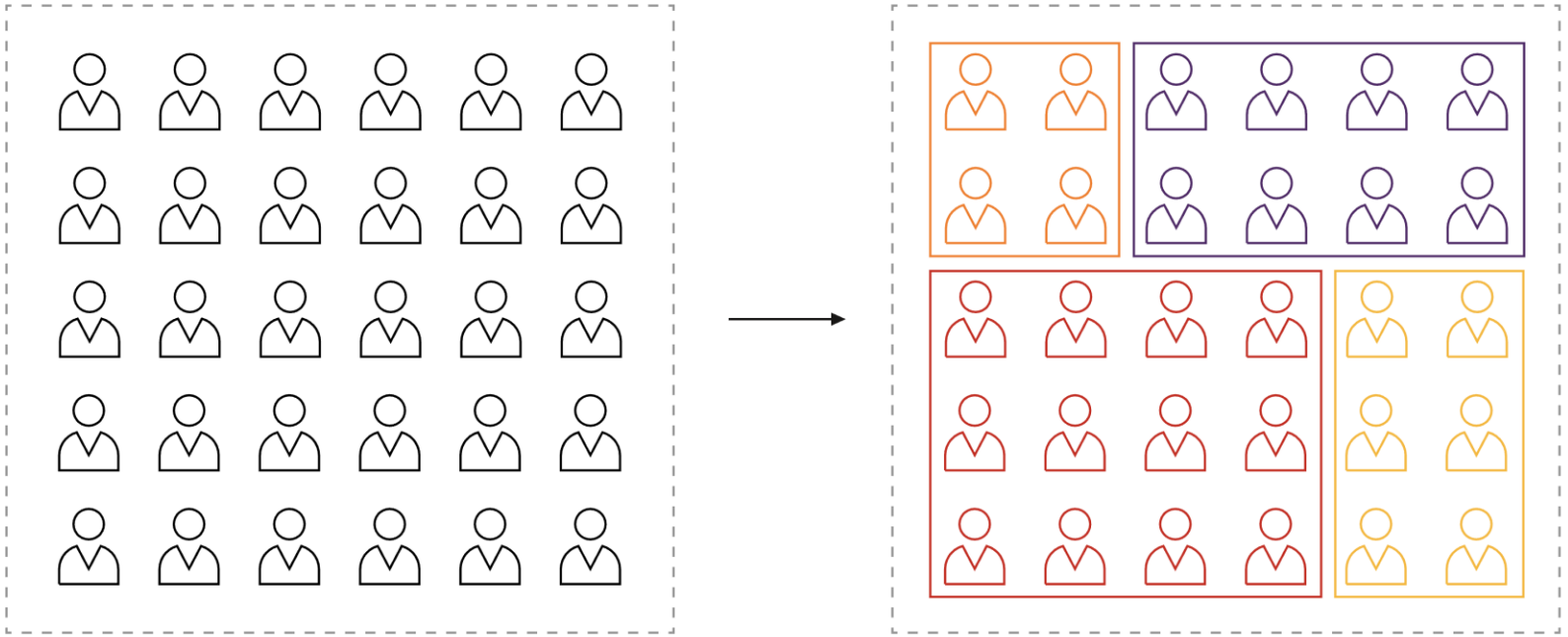
trans_date_trans...	merchant	category	amt	gender	street	zip	unix_time	merch_lat	merch_long
2020-06-21 2020-12-31	693 unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	1 22.8k	F 55% M 45%	924 unique values	1257 99.9k	1375829067.16 - 1376163417.34 Count: 11,371 1.37b 1.39b	19 66.7	-167 -67
2020-06-21 12:14:25	fraud_Kirlin and Sons	personal_care	2.86	M	351 Darlene Green	29209	1371816865	33.986391	-81.200714
2020-06-21 12:14:33	fraud_Sporer-Keebler	personal_care	29.84	F	3638 Marsh Union	84002	1371816873	39.450497999999996	-109.960431
2020-06-21 12:14:53	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	9333 Valentine Point	11710	1371816893	40.49581	-74.196111
2020-06-21 12:15:15	fraud_Haley Group	misc_pos	60.05	M	32941 Krystal Mill Apt. 552	32780	1371816915	28.812397999999998	-80.883061
2020-06-21 12:15:17	fraud_Johnston-Casper	travel	3.19	M	5783 Evan Roads Apt. 465	49632	1371816917	44.959148	-85.884734

Dataset source: https://www.kaggle.com/datasets/kartik2112/fraud-detection

Unsupervised Learning Example

Market Segmentation

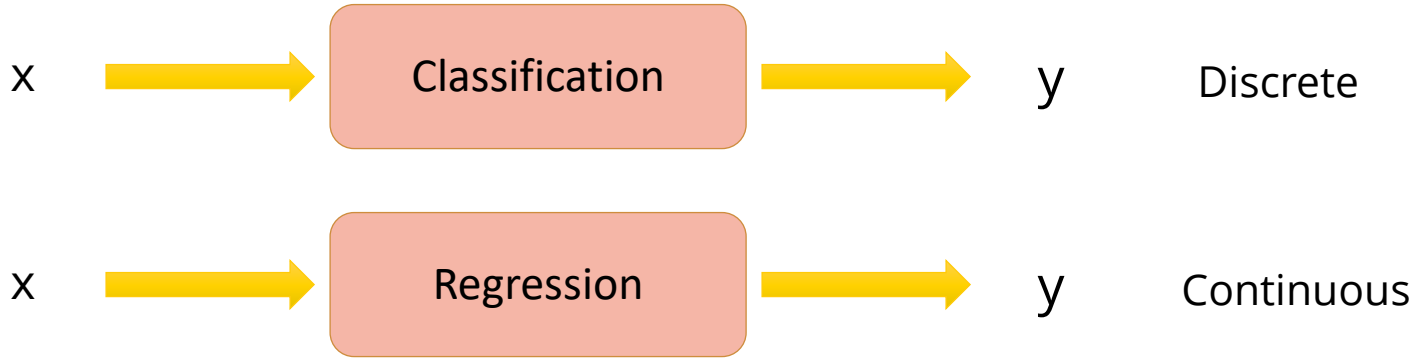
Learning from data without guidance



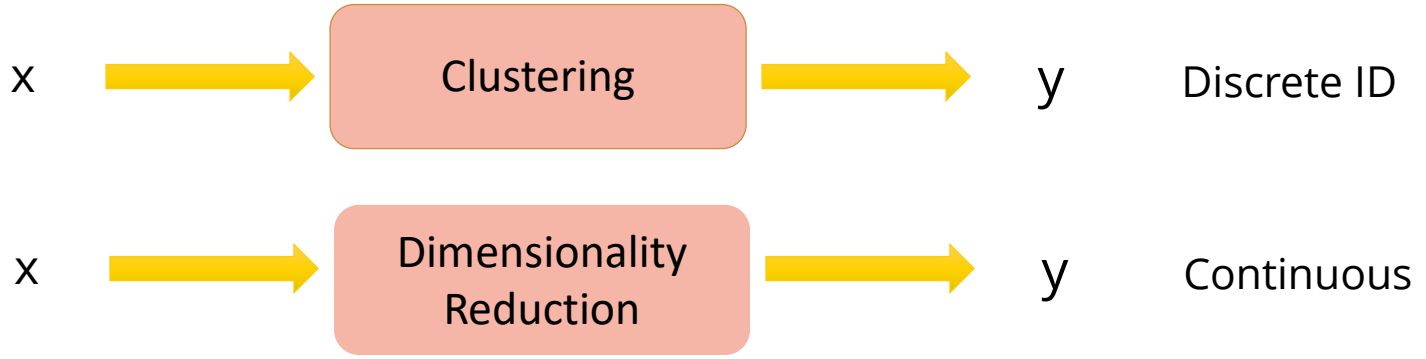
Source: https://www.univio.com/blog/machine-learning-and-customer-segmentation-meet-the-perfect-couple/

Tasks

Supervised Learning



Unsupervised Learning

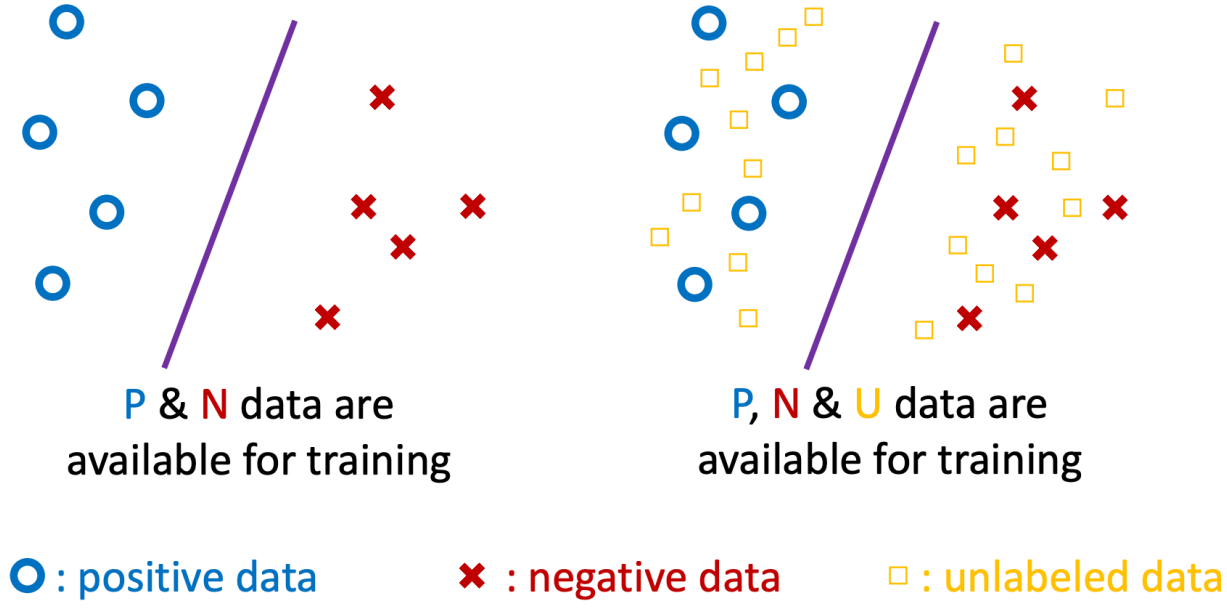


Weakly-Supervised Learning Example

Breast Cancer Tumor Prediction - Scarce labeled data!

Scenarios:

- Low-quality labels
- Proxy process to label data

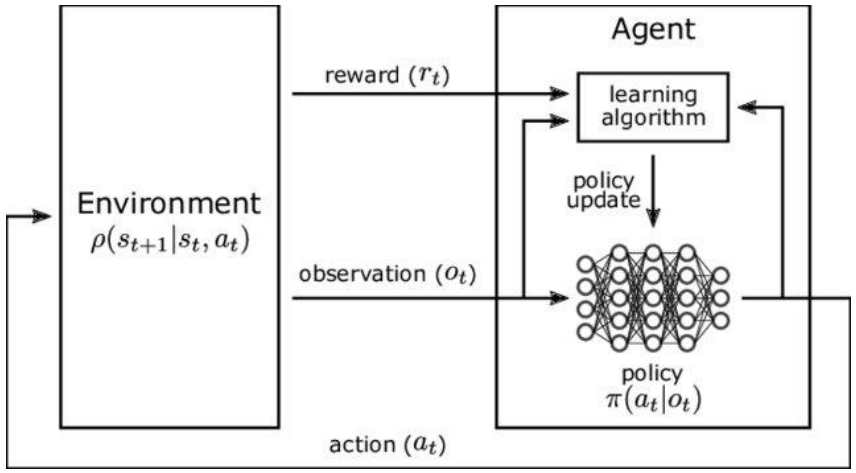


Source: https://niug1984.github.io/paper/niu_tdlw2018.pdf

Reinforcement Learning

An agent:

- Interacts with an Environment
- Learns by Trial-and-Error
- Receives rewards from actions



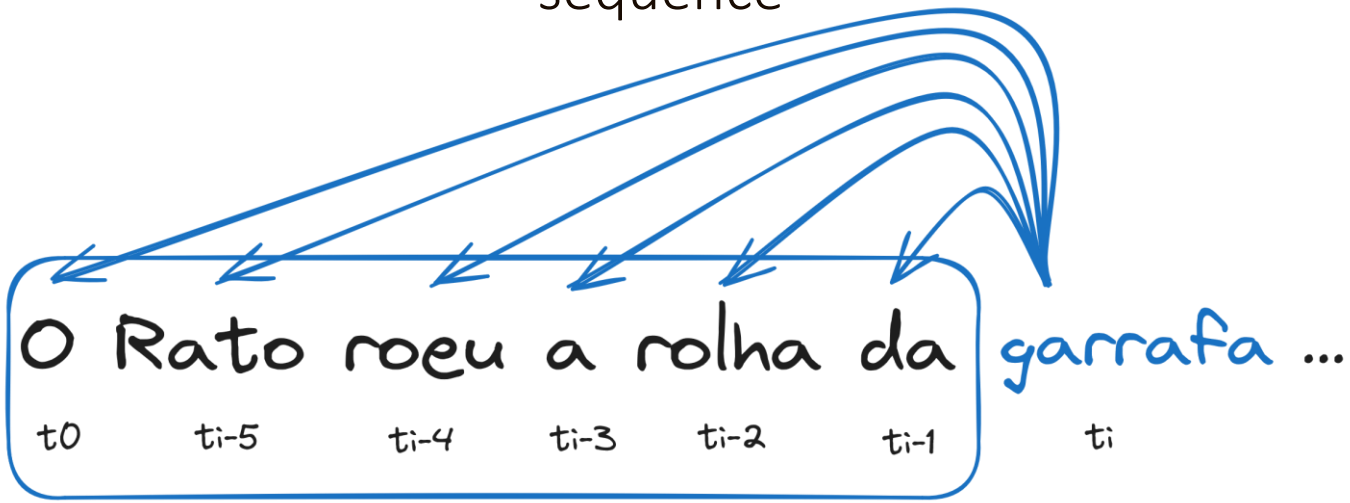
Source: https://bernardmarr.com/how-tesla-is-using-artificial-intelligence-to-create-the-autonomous-cars-of-the-future/

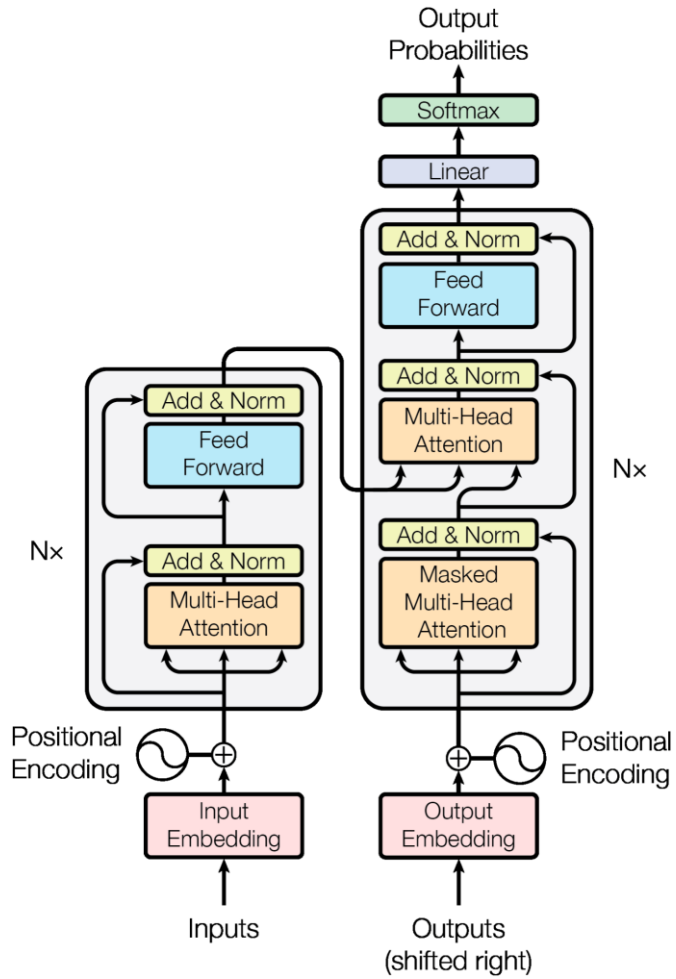
Self-supervised Learning

Causal Language Modeling Objective

$$P(t_1,t_2,\ldots,t_W) = P(t_1) \cdot P(t_2 \mid t_1) \cdot P(t_3 \mid t_2,t_1) \cdot \ldots \cdot P(t_W \mid t_{W-1},\ldots,t_1),$$

t_i is the *i*-th word, and **W** is the total amount of words in a sequence

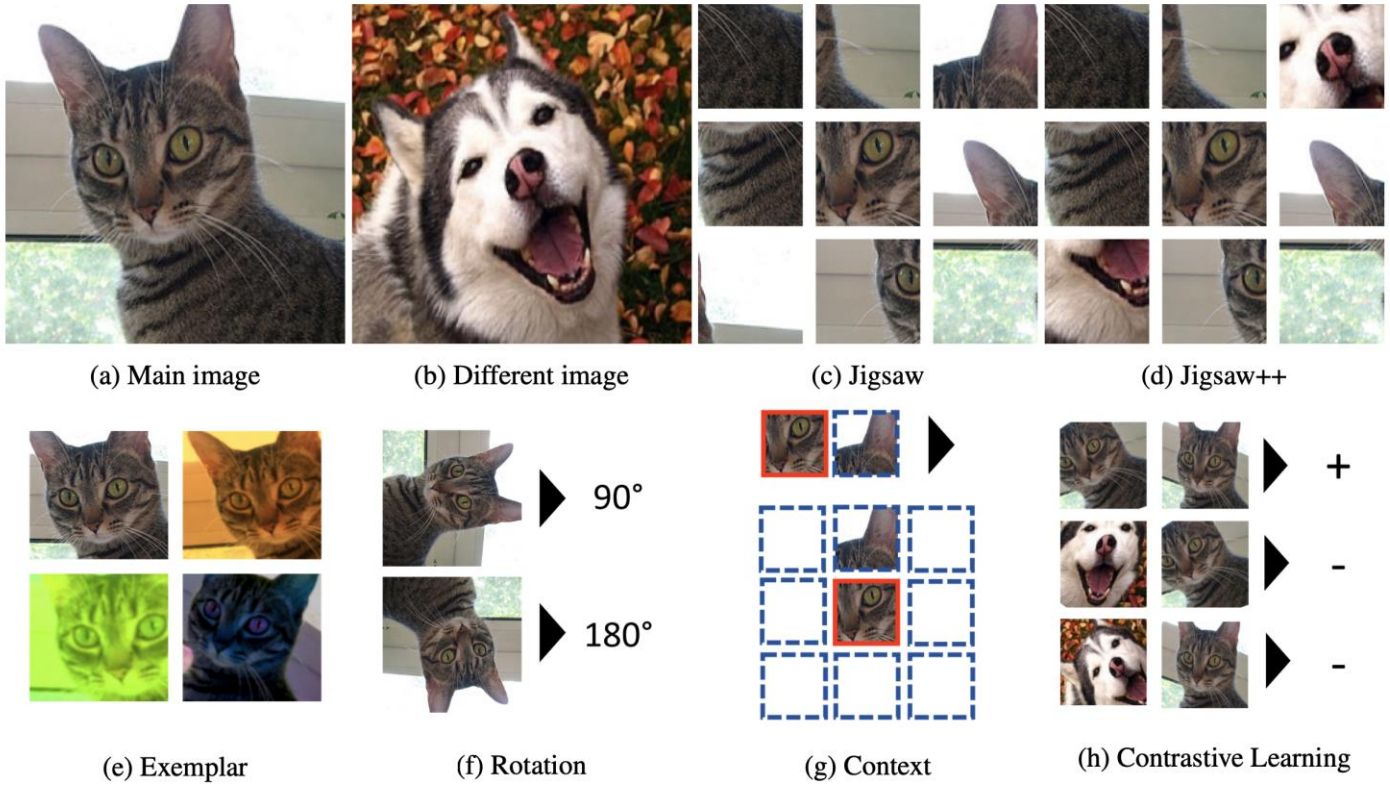




Self-supervised Learning

Pretext Tasks

Learning data representations



Schmarje, L., Santarossa, M., Schroder, S., & Koch, R. A Survey on Semi-, Self- and Unsupervised Learning for Image Classification. *IEEE Access, 2020.*

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)

- (Unknown) Target Function
  - f: X → Y (the “true” mapping / reality)

- Data
  - $\{ (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \}$

- Model / Hypothesis Class
  - H = {h: X → Y}
  - e.g. $y = h(x) = \text{sign}(w^T x)$

- Loss Function
  - How good is a model w.r.t. my data D?

- Learning = Search in hypothesis space
  - Find best h in model class.

Source: Pedro Domingos

Appropriate Applications for Supervised Learning

- **Situations where there is no human expert**

  $\mathbf{x}$: Bond graph for a new molecule.

  $f(\mathbf{x})$: Predicted binding strength to AIDS protease molecule.

- **Situations where humans can perform the task but can't describe how they do it.**

  $\mathbf{x}$: Bitmap picture of hand-written character

  $f(\mathbf{x})$: Ascii code of the character

- **Situations where the desired function is changing frequently**

  $\mathbf{x}$: Description of stock prices and trades for last 10 days.

  $f(\mathbf{x})$: Recommended stock transactions

- **Situations where each user needs a customized function f**

  $\mathbf{x}$: Incoming email message.

  $f(\mathbf{x})$: Importance score for presenting to user (or deleting without presenting).

Source: Pedro Domingos, Tom Mitchel, Tom Dietterich

CMU Portugal Academy

Feature Extraction and Engineering

Obtaining feature vectors:

- Numeric representation of a sample
- Implies the application of translation function

trans_date_trans...	merchant	category	amt	gender	street	zip	unix_time	merch_lat	merch_long
2020-06-21 2020-12-31	693 unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	1 22.8k	F 55% M 45%	924 unique values	1257 99.9k	1.37b 1.39b	19 66.7	-167 -67
2020-06-21 12:14:25	fraud_Kirlin and Sons	personal_care	2.86	M	351 Darlene Green	29209	1371816865	33.986391	-81.200714
2020-06-21 12:14:33	fraud_Sporer-Keebler	personal_care	29.84	F	3638 Marsh Union	84002	1371816873	39.450497999999996	-109.960431
2020-06-21 12:14:53	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	9333 Valentine Point	11710	1371816893	40.49581	-74.196111
2020-06-21 12:15:15	fraud_Haley Group	misc_pos	60.05	M	32941 Krystal Mill Apt. 552	32780	1371816915	28.812397999999998	-80.883061
2020-06-21 12:15:17	fraud_Johnston-Casper	travel	3.19	M	5783 Evan Roads Apt. 465	49632	1371816917	44.959148	-85.884734

CMU Portugal Academy

Feature Extraction and Engineering

Obtaining feature vectors:

- Numeric representation of a sample
- Implies the application of translation function

$$t(x_i) \rightarrow x_i'$$

Might introduce noise or an inaccurate approximation

trans_date_trans...	merchant	category	amt	gender	street	zip	unix_time	merch_lat	merch_long
2020-06-21 2020-12-31	693 unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	1 22.8k	F 55% M 45%	924 unique values	1257 99.9k	1375829067.16 - 1376163417.34 Count: 11,371 1.37b 1.39b	19 66.7	-167 -67
2020-06-21 12:14:25	fraud_Kirlin and Sons	personal_care	2.86	M	351 Darlene Green	29209	1371816865	33.986391	-81.200714
2020-06-21 12:14:33	fraud_Sporer-Keebler	personal_care	29.84	F	3638 Marsh Union	84002	1371816873	39.450497999999996	-109.960431
2020-06-21 12:14:53	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	9333 Valentine Point	11710	1371816893	40.49581	-74.196111
2020-06-21 12:15:15	fraud_Haley Group	misc_pos	60.05	M	32941 Krystal Mill Apt. 552	32780	1371816915	28.812397999999998	-80.883061
2020-06-21 12:15:17	fraud_Johnston-Casper	travel	3.19	M	5783 Evan Roads Apt. 465	49632	1371816917	44.959148	-85.884734

CMU Portugal Academy

Feature Extraction and Engineering

Feature Transformations $t(x_i) \rightarrow x_i'$

Gender: {F, M} -> {0, 1}

trans_date_trans...	merchant	category	amt	gender	street	zip	unix_time	merch_lat	merch_long
2020-06-21 2020-12-31	693 unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	1 22.8k	F 55% M 45%	924 unique values	1257 99.9k	1375829067.16 - 1376163417.34 Count: 11,371 1.37b 1.39b	19 66.7	-167 -67
2020-06-21 12:14:25	fraud_Kirlin and Sons	personal_care	2.86	M	351 Darlene Green	29209	1371816865	33.986391	-81.200714
2020-06-21 12:14:33	fraud_Sporer-Keebler	personal_care	29.84	F	3638 Marsh Union	84002	1371816873	39.450497999999996	-109.960431
2020-06-21 12:14:53	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	9333 Valentine Point	11710	1371816893	40.49581	-74.196111
2020-06-21 12:15:15	fraud_Haley Group	misc_pos	60.05	M	32941 Krystal Mill Apt. 552	32780	1371816915	28.812397999999998	-80.883061
2020-06-21 12:15:17	fraud_Johnston-Casper	travel	3.19	M	5783 Evan Roads Apt. 465	49632	1371816917	44.959148	-85.884734

CMU Portugal Academy

Feature Extraction and Engineering

Feature Transformations $t(x_i) \rightarrow x_i'$

Gender: {F, M} -> {0, 1}

Category: {travel, personal_care, ...} -> {0, 1, 2, ... }

trans_date_trans...	merchant	category	amt	gender	street	zip	unix_time	merch_lat	merch_long
2020-06-21 2020-12-31	693 unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	1 22.8k	F 55% M 45%	924 unique values	1257 99.9k	1375829067.16 - 1376163417.34 Count: 11,371 1.37b 1.39b	19 66.7	-167 -67
2020-06-21 12:14:25	fraud_Kirlin and Sons	personal_care	2.86	M	351 Darlene Green	29209	1371816865	33.986391	-81.200714
2020-06-21 12:14:33	fraud_Sporer-Keebler	personal_care	29.84	F	3638 Marsh Union	84002	1371816873	39.450497999999996	-109.960431
2020-06-21 12:14:53	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	9333 Valentine Point	11710	1371816893	40.49581	-74.196111
2020-06-21 12:15:15	fraud_Haley Group	misc_pos	60.05	M	32941 Krystal Mill Apt. 552	32780	1371816915	28.812397999999998	-80.883061
2020-06-21 12:15:17	fraud_Johnston-Casper	travel	3.19	M	5783 Evan Roads Apt. 465	49632	1371816917	44.959148	-85.884734

CMU Portugal Academy

Feature Extraction and Engineering

Feature Transformations $t(x_i) \rightarrow x_i'$

Unix Timestamp -> Seconds since January 01, 1970.

Unix_time: [0, 2147483647] -> ???

Assuming 32bit integers.

trans_date_trans...	merchant	category	amt	gender	street	zip	unix_time	merch_lat	merch_long
2020-06-21 2020-12-31	693 unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	1 22.8k	F 55% M 45%	924 unique values	1257 99.9k	1.37b 1.39b	19 66.7	-167 -67
2020-06-21 12:14:25	fraud_Kirlin and Sons	personal_care	2.86	M	351 Darlene Green	29209	1371816865	33.986391	-81.200714
2020-06-21 12:14:33	fraud_Sporer-Keebler	personal_care	29.84	F	3638 Marsh Union	84002	1371816873	39.450497999999996	-109.960431
2020-06-21 12:14:53	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	F	9333 Valentine Point	11710	1371816893	40.49581	-74.196111
2020-06-21 12:15:15	fraud_Haley Group	misc_pos	60.05	M	32941 Krystal Mill Apt. 552	32780	1371816915	28.812397999999998	-80.883061
2020-06-21 12:15:17	fraud_Johnston-Casper	travel	3.19	M	5783 Evan Roads Apt. 465	49632	1371816917	44.959148	-85.884734

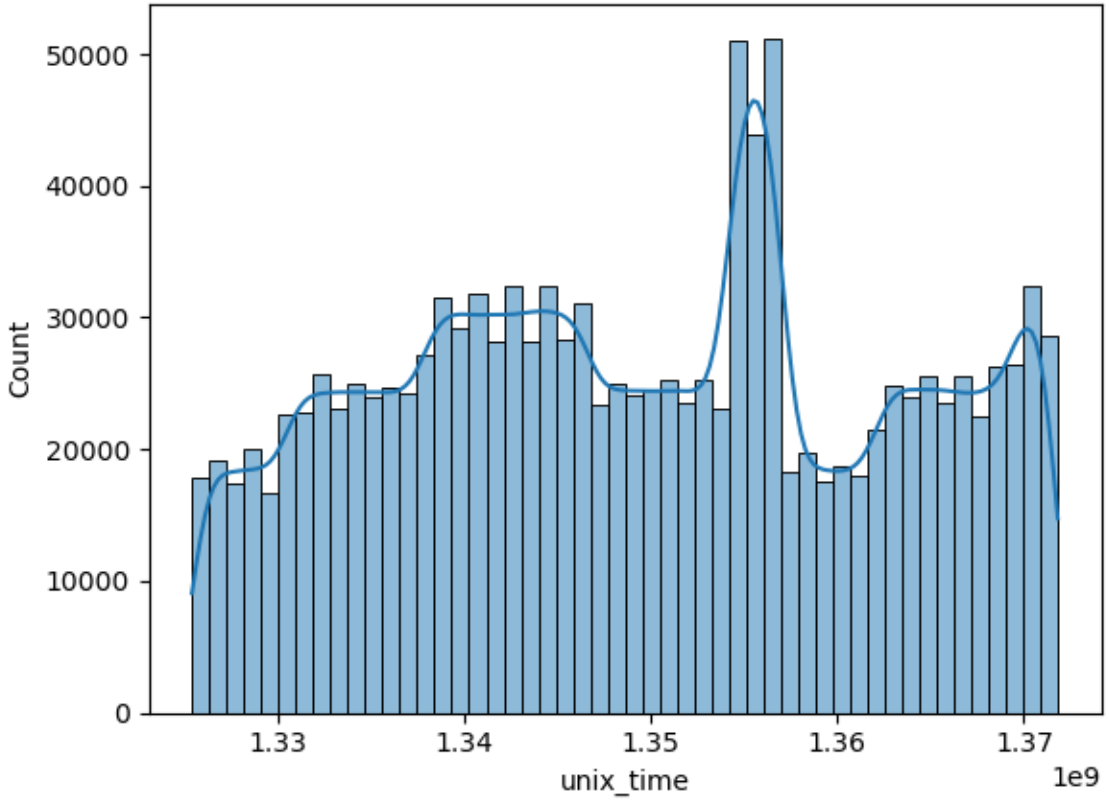
Feature Extraction and Engineering

Unix Timestamp -> Seconds since January 01, 1970

Unix_time: [0, 2147483647] -> ???

Assuming 32bit integers.

What should we do?



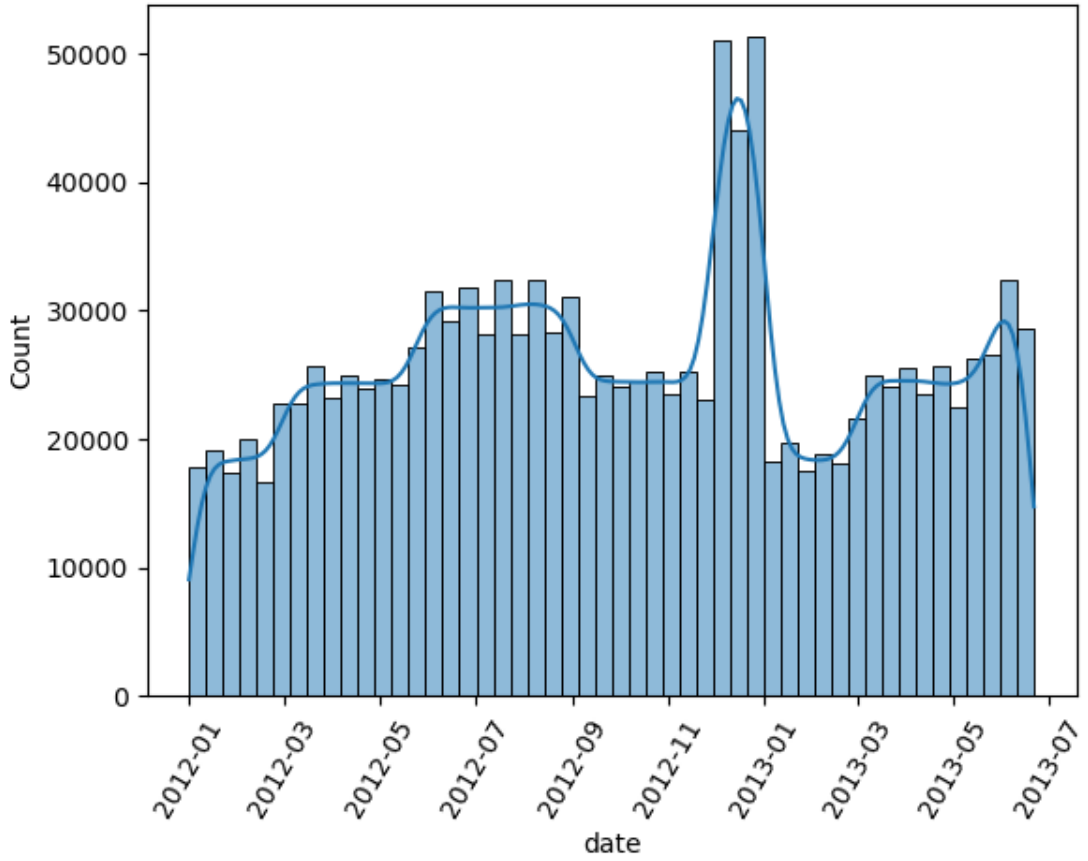
Feature Extraction and Engineering

Unix Timestamp -> Seconds since January 01, 1970.

Unix_time: [0, 2147483647] -> ???

Assuming 32bit integers.

The domain is actually quite narrow!



Feature Extraction and Engineering

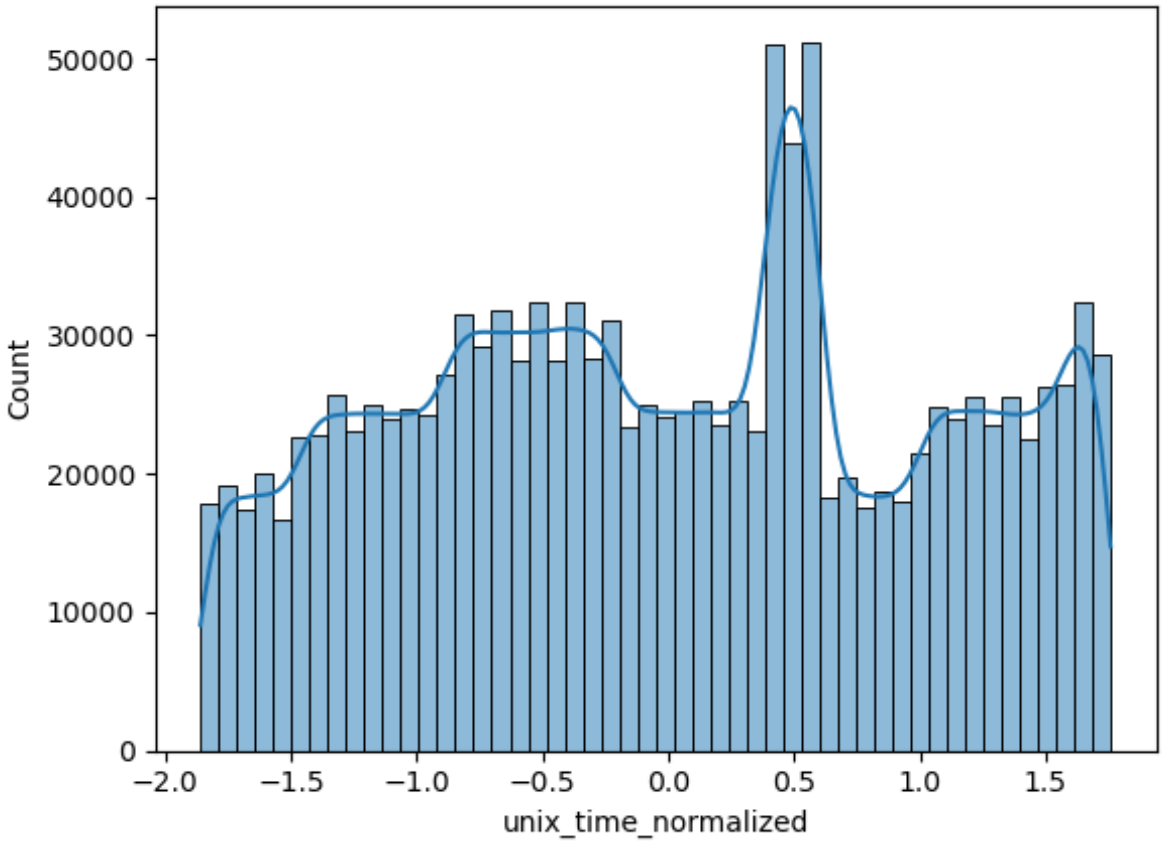
Normalization:

- Standard Scaling (z-scoring):

$$x_i' = \frac{x_i - \mu}{\sigma}$$ Maps to the range $]-\infty, +\infty[$

- Min-max Normalization:

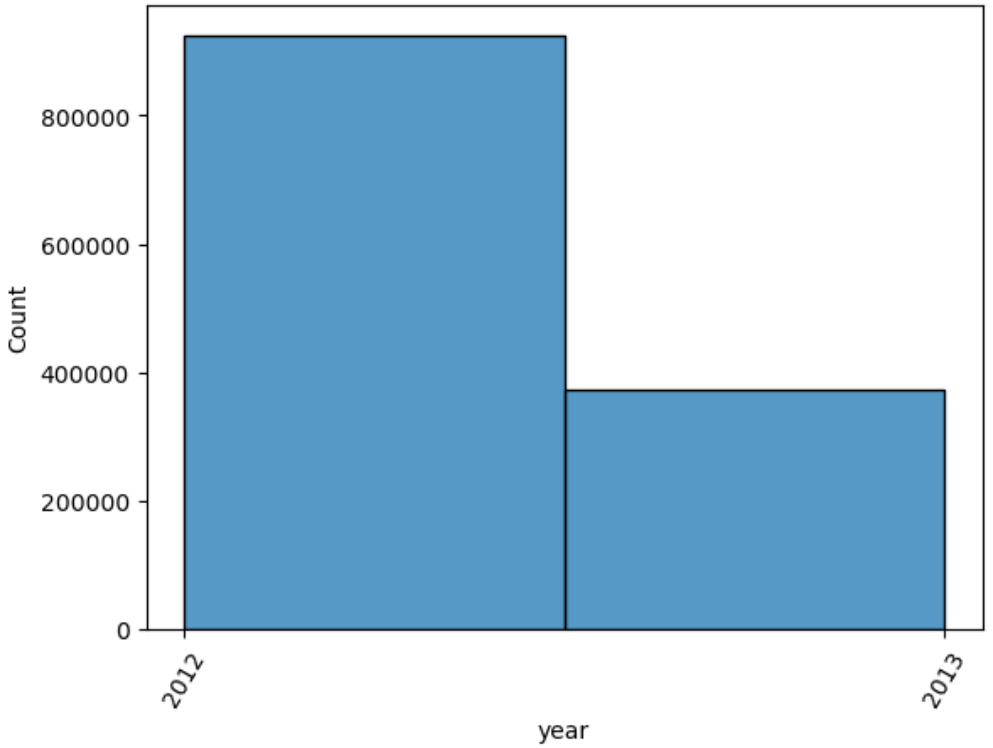
$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$ Maps to the range $[0, 1]$



Feature Extraction and Engineering

Other Approaches:

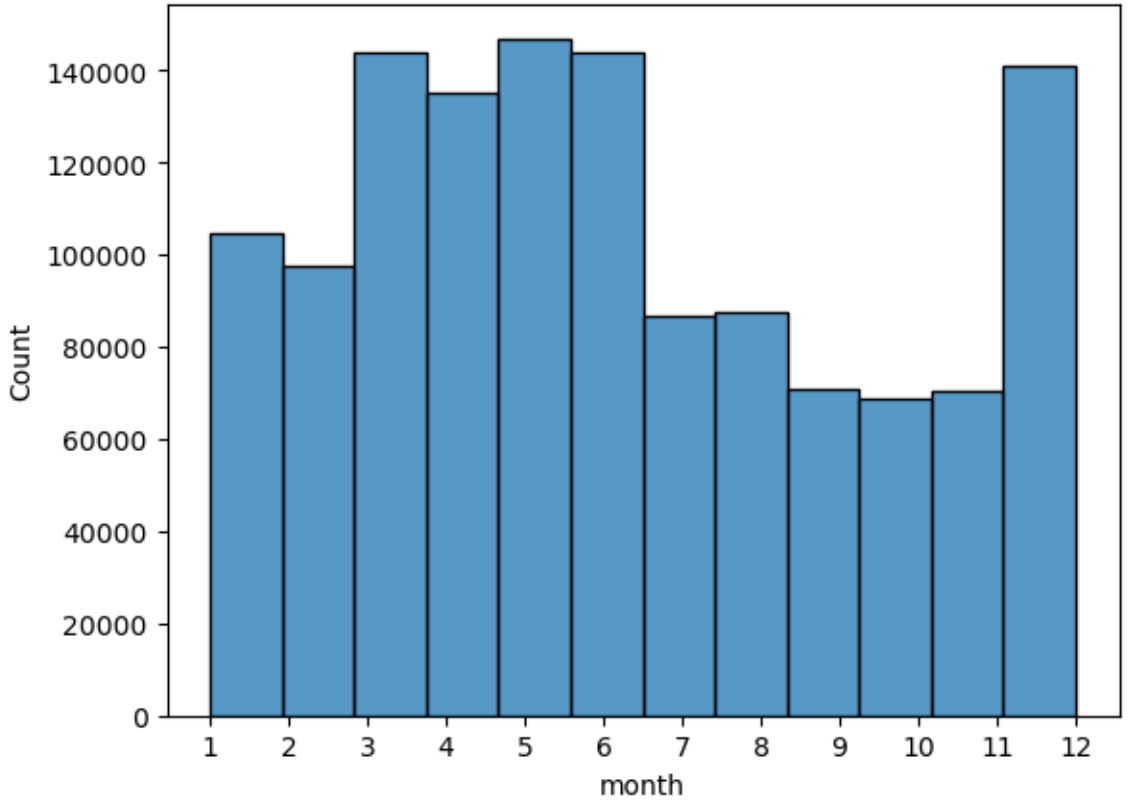
- Discretization through binning
- Discretization through clustering
- Granularity / Precision changes (e.g. use only the date year)



Feature Extraction and Engineering

Other Approaches:

- Discretization through binning
- Discretization through clustering
- Granularity / Precision changes (e.g. use only the date year)
- New features derivation (e.g. difference between locations in kilometers)



Linear Models – Linear Regression

We assume that the relationship between features x and target y is approximately linear

- The conditional mean $E[Y|X = x]$ can be expressed as a weighted sum of the features x.
- We assume there will some well behaved noise (e.g. following a Gaussian distribution)

CMU Portugal Academy

Linear Models – Linear Regression

For example, the price can be defined as a weighted sum of features:

$$price = w_{area} * area + w_{age} * age + b$$

Bias/intercept/offset

Generalizing: $\hat{y} = w_1 x_1 + \cdots + w_d x_d + b$

If we represent all features into a vector $\boldsymbol{x} \in \mathbb{R}^d$ and all weights into a vector $\boldsymbol{w} \in \mathbb{R}^d$:

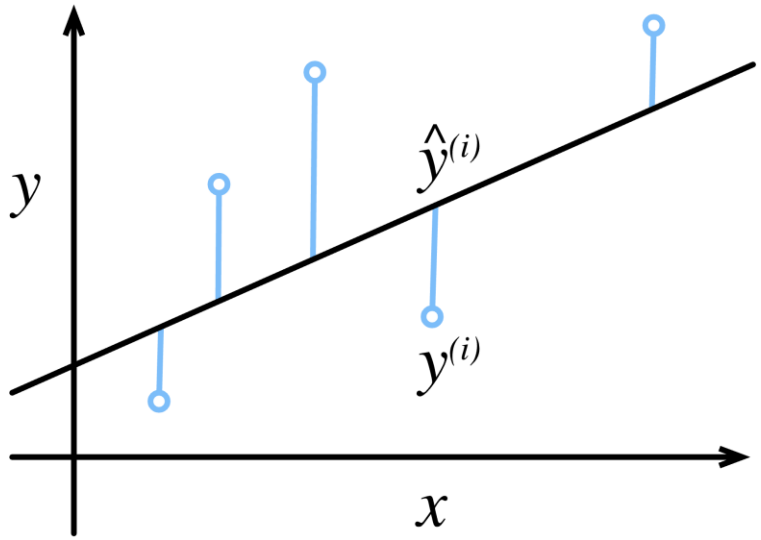
$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b$$

Usually, we represent the whole dataset of n example as a design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
Then, it becomes:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$$

Linear Models – Linear Regression

Regression Loss function: $l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$



Generalizing to the whole dataset:

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

Find the optimal weights:

$$\mathbf{w}^*, b^* = \underset{\mathbf{w}, b}{\operatorname{argmin}}\ L(\mathbf{w}, b)$$

Linear Regression – Closed Form Solution

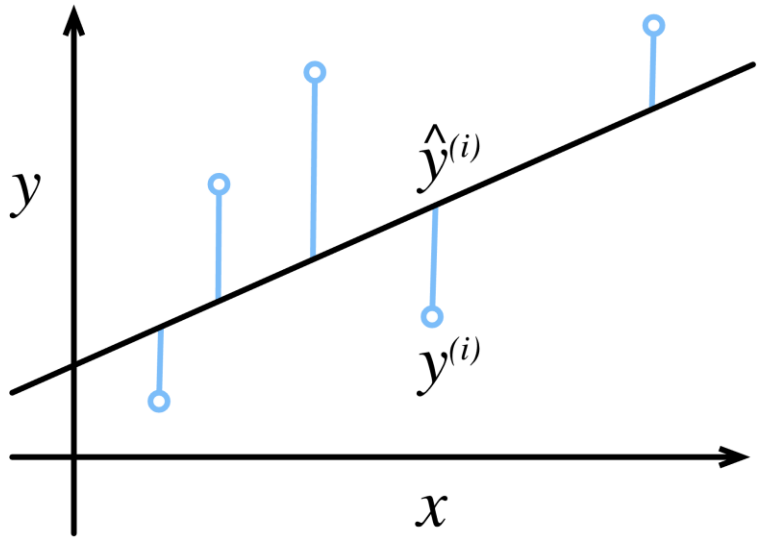
Represent minimization problem as: $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$ ➡ Bias is added to **w** and a 1s column is added to X

We take the derivative, w.r.t. *w*, to find the minimum:

$$\partial_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 = 2\mathbf{X}^\top(\mathbf{X}\mathbf{w} - \mathbf{y}) = 0 \text{ and hence } \mathbf{X}^\top\mathbf{y} = \mathbf{X}^\top\mathbf{X}\mathbf{w}$$

Solving for *w*:

$$\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$



The solution is unique when the matrix X^TX is invertible, i.e., the columns of the design matrix are linearly independent (Golub and Van Loan, 1996).

Golub, G. H., & Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins University Press.

CMU Portugal Academy

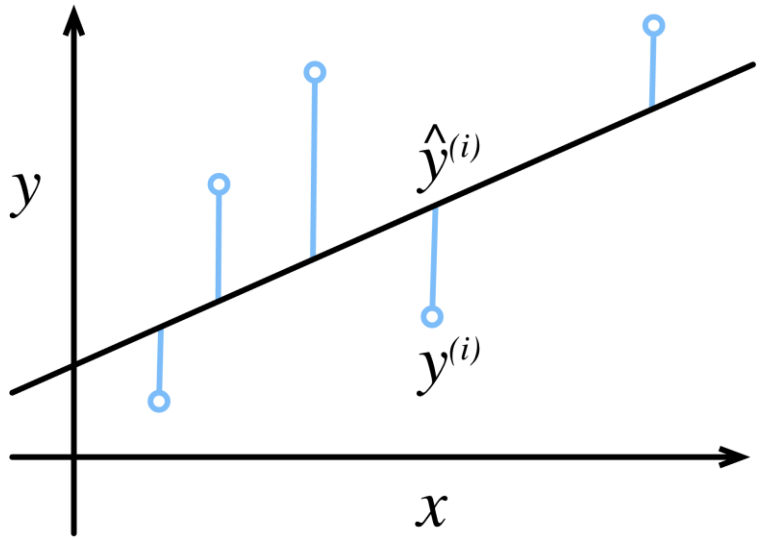
Linear Regression – Closed-Form Solution

Represent minimization problem as: $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$

Bias is added to **w** and a 1s column is added to X

Solving for w: $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$

- Linear regression has a closed-form solution.
- More complex models (e.g. Neural Networks) don't.
- In the Machine Learning module you will learn about other optimization approaches, like Stochastic Gradient Descent.



Logistic Regression

Vanilla linear regression is not so suitable to classification problems:

- Targets are continuous values, along a straight line
- Decision boundary too sensitive to outliers

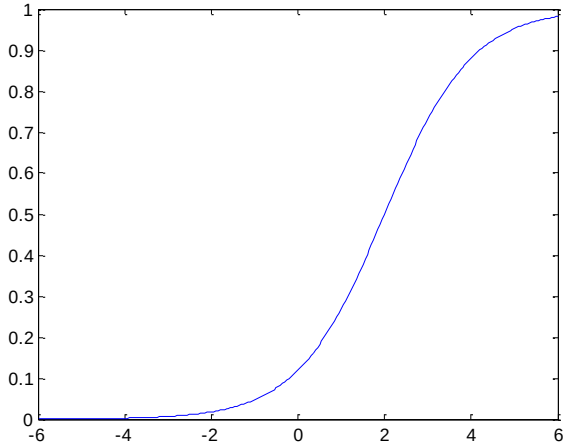
For a classification problem, we want to obtain a probability $P(Y = label|X = x)$.

We can squash the linear regression outputs to the [0,1] range with the sigmoid function!

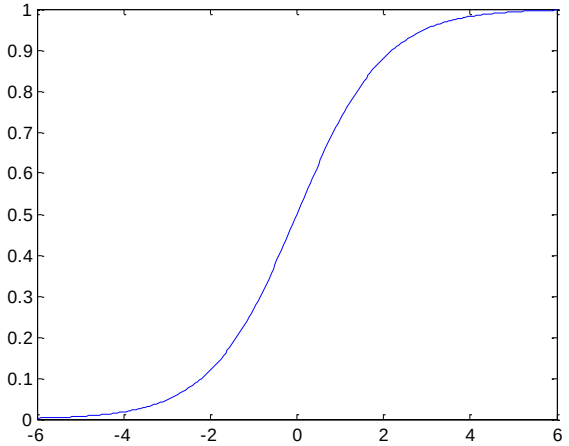
Logistic Regression – Sigmoid

$$\sigma(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{-w_0 - \sum_i w_i x_i}} = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{X}}}$$

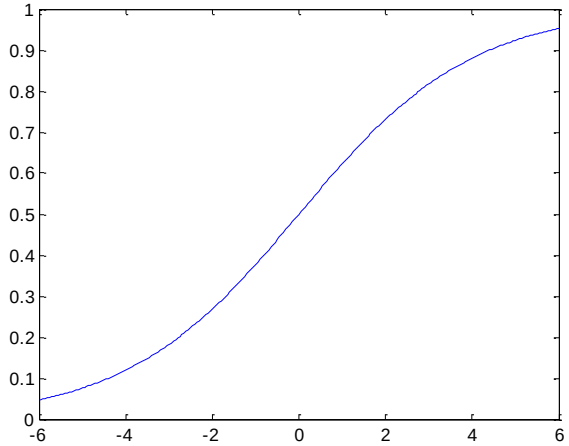
w_0=2, w_1=1



w_0=0, w_1=1



w_0=0, w_1=0.5



Source: Carlos Guestrin

Logistic Regression – Binary Classification

We can assume the following:

- Class 1 corresponds to y = 1
- Class 2 corresponds to y = 0

$$\boldsymbol{P(Y = 1|X, w)} = \sigma(\boldsymbol{w}^T * \boldsymbol{X}) = \frac{1}{1 + e^{-\boldsymbol{w}^T * \boldsymbol{X}}}$$

$$\boldsymbol{P(Y = 0|X, w)} = 1 - P(Y = 1|X, w)$$

For a classification problem, we want to obtain a probability $\boldsymbol{P(Y = label|X = x)}$.

We can squash the linear regression outputs to the [0,1] range with the sigmoid function!

Logistic Regression – Binary Classification

We can assume the following:

- Class 1 corresponds to y = 1
- Class 2 corresponds to y = 0

$$\boldsymbol{P(Y = 1|X, w) = \sigma(w^T * X) = \frac{1}{1 + e^{-w^T * X}}}$$

$$\boldsymbol{P(Y = 0|X, w) = 1 - P(Y = 1|X, w)}$$

To find w, logistic regressions uses negative log-likelihood (or cross-entropy):

$$L(\mathbf{w}) = \sum_j -ln\ P(y^j|x^j, \mathbf{w})$$

$$L(\mathbf{w}) = \sum_j -y^j \cdot ln\ P(y^j = 1|x^j, \mathbf{w}) - (1 - y^j) \cdot ln\ P(y^j = 0|x^j, \mathbf{w})$$

Logistic Regression – Binary Classification

To find w, logistic regression uses negative log-likelihood (or cross-entropy):

$$L(\mathbf{w}) = \sum_j -y^j \cdot ln\ P(y^j = 1|x^j, \mathbf{w}) - (1 - y^j) \cdot ln\ P(y^j = 0|x^j, \mathbf{w})$$

Substituting now with the sigmoid, it becomes:

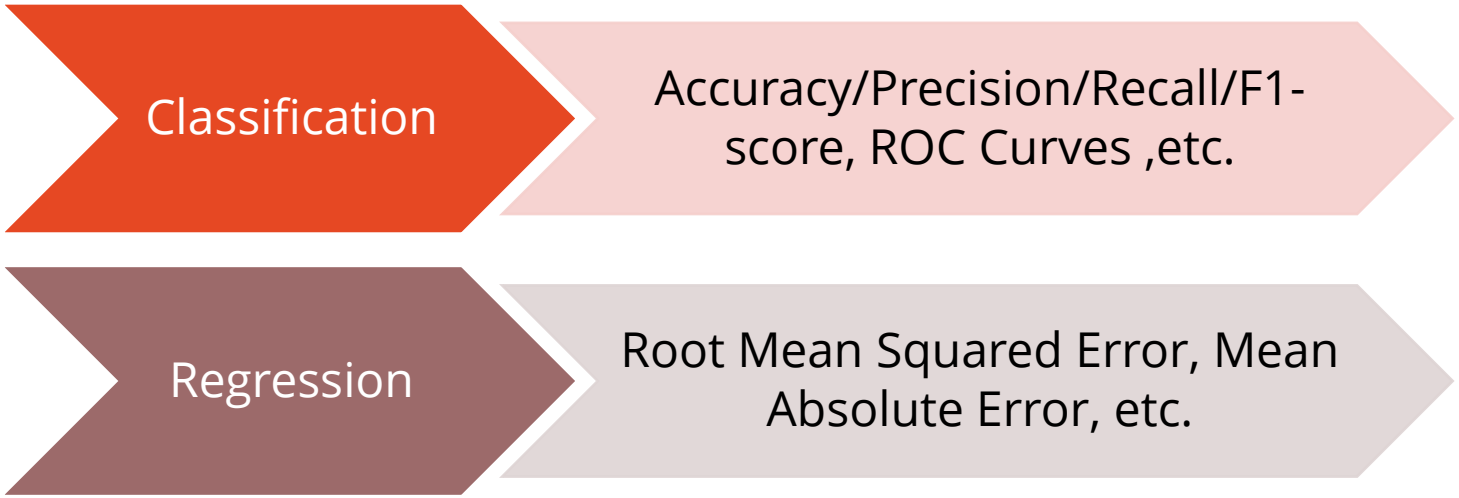
$$L(\mathbf{w}) = \sum_j -y^j \cdot ln\ \sigma(\mathbf{w}^T x^j) - (1 - y^j) \cdot ln\ (1 - \sigma(\mathbf{w}^T x^j))$$

This loss does not have a closed-form solution, but is concave
-> Can be optimized with Gradient Descent

Assessing Model Performance - Metrics

It is critical to use quantitative metrics to evaluate machine learning models

- The loss function is not enough
- Different metrics provide different perspectives of models' performance!



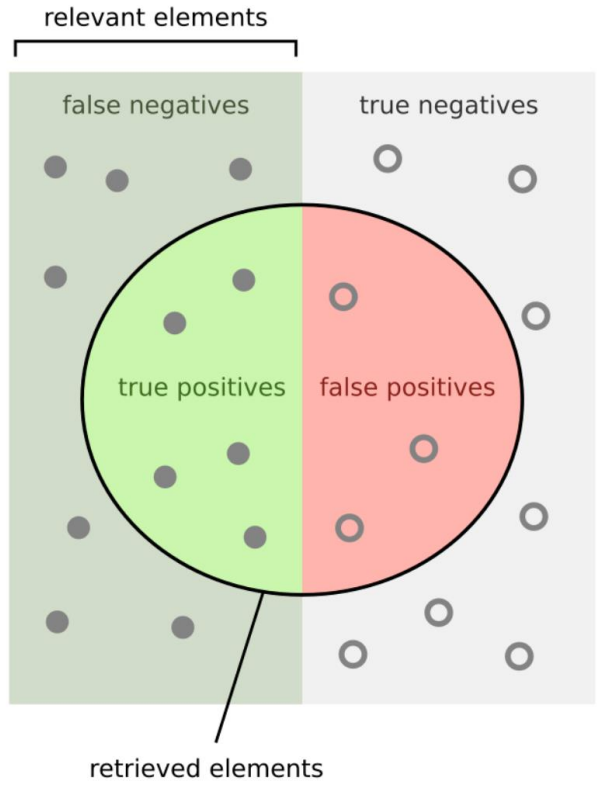
Assessing Model Performance - Metrics

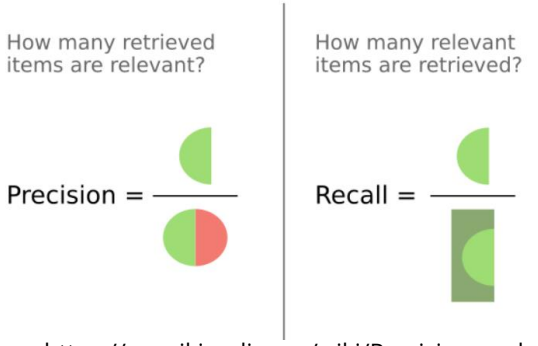
$$\text{Accuracy} = \frac{correct\ classifications}{total\ classifications} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Assessing Model Performance - Metrics

$$Precision = \frac{True\ Positives}{Predicted\ Positives} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$Recall/Sensitivity = \frac{True\ Positives}{Actual\ Positives} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$





Source: https://en.wikipedia.org/wiki/Precision_and_recall

CMU Portugal Academy

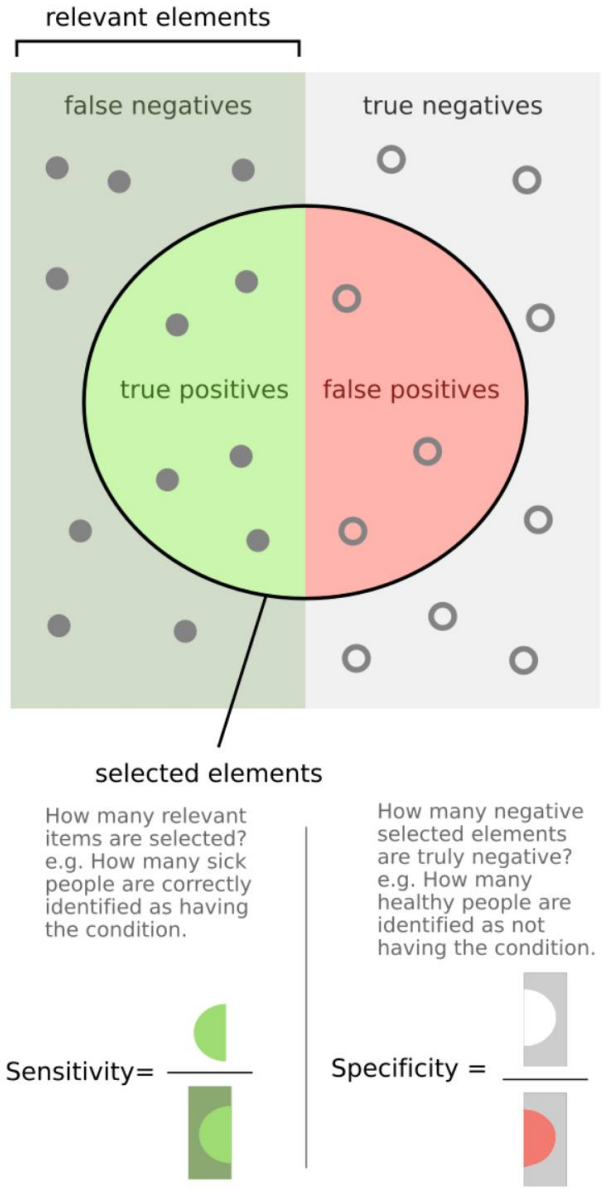
Assessing Model Performance - Metrics

$$Recall/Sensitivity = \frac{True\ Positives}{Total\ Positives} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

True **Positive** Rate (TPR)

$$Specificity = \frac{True\ Negatives}{Total\ Negatives} = \frac{TN}{TN + \mathrm{FP}}$$

True **Negative** Rate (TNR)



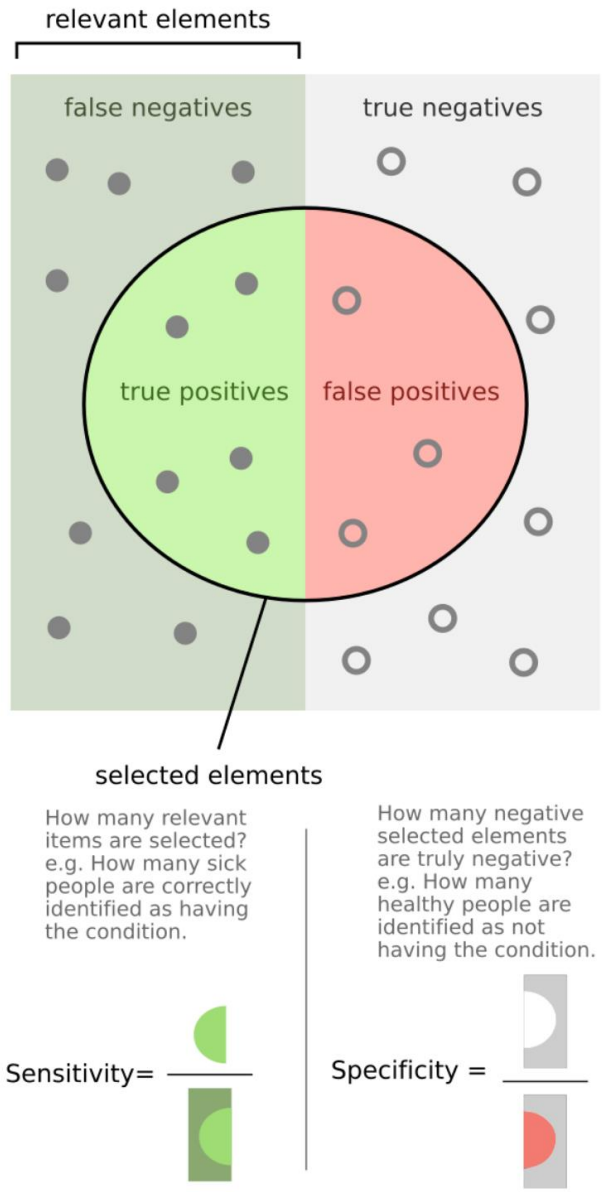
Source: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Assessing Model Performance – Metrics

F-score - Harmonic mean between precision and recall:

$$F_1 = \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Aggregates (symmetrically) information from both metrics.



Assessing Model Performance - Metrics

Confusion Matrix

Provides a more detailed view of a classification model performance

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Why multiple perspectives?

Example 1:

- Imbalanced dataset, 99% of examples are Spam, 1% are not.
- Our model always predicts Spam. What is the accuracy?

It is okay if a Spam email goes to our inbox -> Low Sensitivity
It is not okay if a Non-spam email gets filtered! -> High Specificity

Precision!

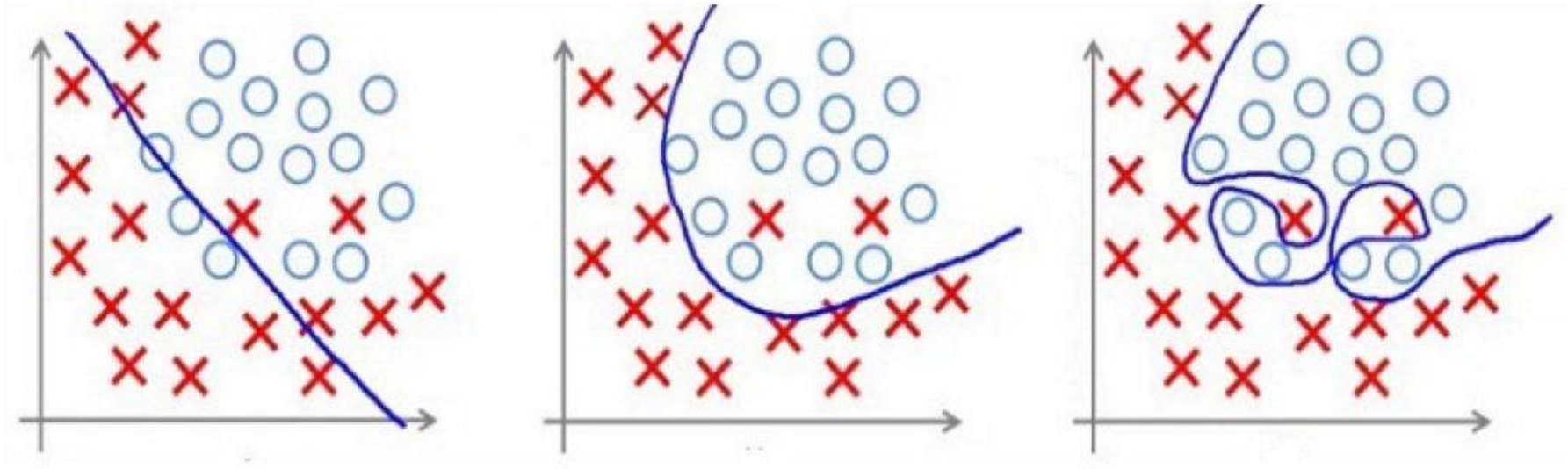
Why multiple perspectives?

Example 2:

- Automatic Border Control Checking

Terrorists shouldn’t go through -> High Sensitivity
False alarms (False positives) are not a big deal -> Low Specificity

Overfitting vs. Underfitting



Underfitting

Too simple to explain variance

High Loss, High Bias

Appropriate

Overfitting

Low generalization

Low Loss, High Variance

Hands-On Session!

Course Shared Folder

CMU Portugal Academy

CMU Portugal
Advanced Training Program

Foundations of Data Science

DAVID SEMEDO
RAFAEL FERREIRA
NOVA SCHOOL OF SCIENCE AND TECHNOLOGY

Carnegie Mellon Portugal · TÉCNICO LISBOA · Ciências ULisboa · NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Email: rah.ferreira@campus.fct.unl.pt

Sources

These slides contain adapted materials from the following sources:

- Stefan Lee, Introduction to Machine Learning, Virginia Tech
- Dive into Deep Learning Book.

CMU Portugal Academy

CMU Portugal
Advanced Training Program

Foundations of Data Science

DAVID SEMEDO
RAFAEL FERREIRA
NOVA SCHOOL OF SCIENCE AND TECHNOLOGY

Carnegie Mellon Portugal · TÉCNICO LISBOA · Ciências ULisboa · NOVA School of Science & Technology

Email: rah.ferreira@campus.fct.unl.pt

Today's Topics

1. MODEL SELECTION
2. CROSS-VALIDATION
3. HYPERPARAMETER TUNING
4. DATA PREPROCESSING
5. FEATURE SELECTION
6. UNSUPERVISED LEARNING
7. DIMENSIONALITY REDUCTION
8. CLUSTERING

CMU Portugal Academy

02

Model Selection

Carnegie Mellon Portugal | TÉCNICO LISBOA | Ciências ULisboa | NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Model Selection

Sports Car
Fast but struggles on rough roads.

Winning a Race

Motorcycle
Agile but risky for harsh weather.

Weaving Through Traffic

Family SUV:
Balanced option.
Not as fast or as agile.

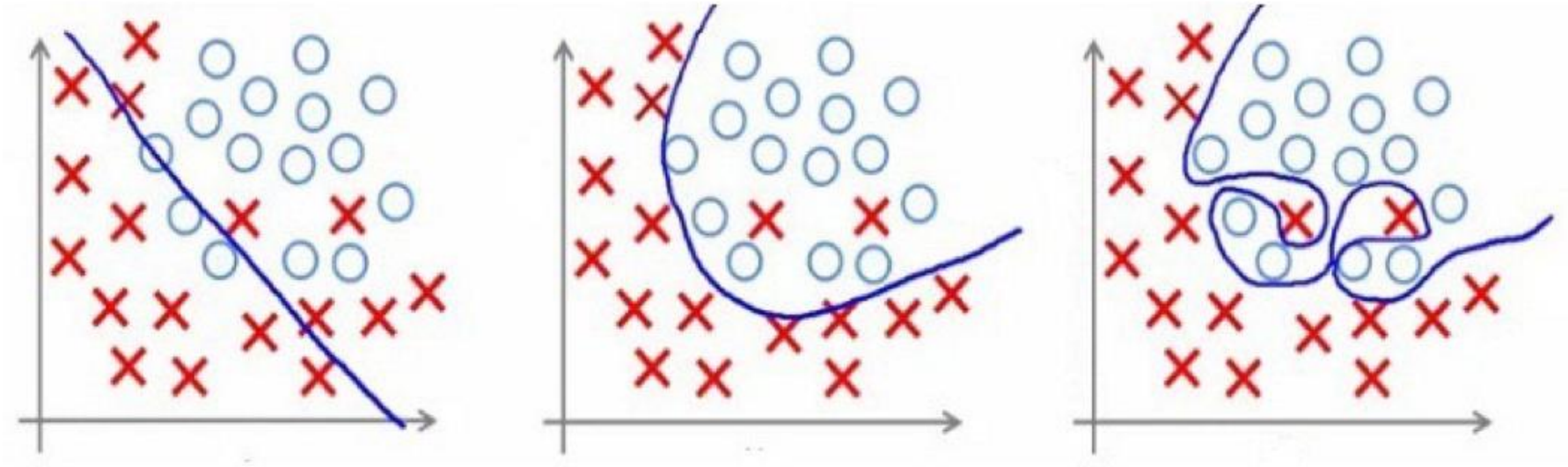
Taking Grandma for a Ride

We need to perform a selection!

Why Model Selection?

- **Performance**: Choosing the model that maximizes metrics and generalizes well to unseen data.
- **Efficiency:** Balance complexity and performance and computation costs.
- **Problem-Specific**: Different models excel in different scenarios (e.g., linear models for simple patterns, non-linear models for complex data).
- **Model Interpretability**: Clearer insights into the underlying data.

Overfitting vs. Underfitting



Underfitting

Too simple to explain variance

High Loss, High Bias

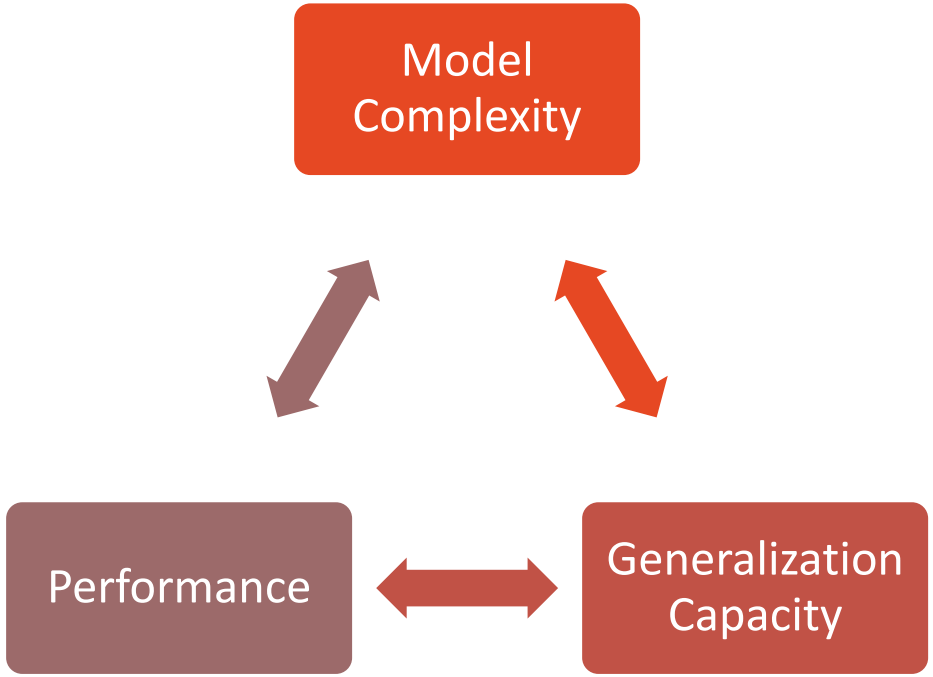
Appropriate

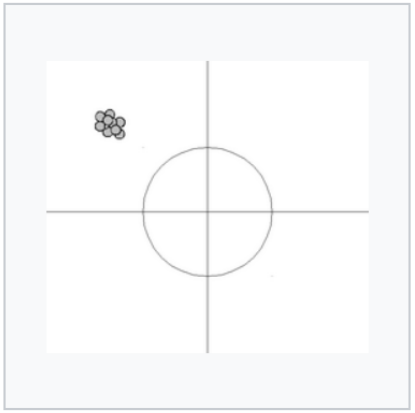
Overfitting

Low generalization

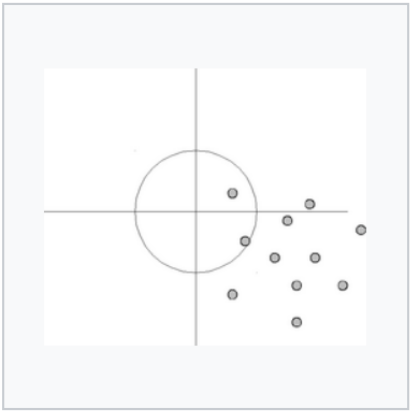
Low Loss, High Variance

Bias-Variance Trade-off



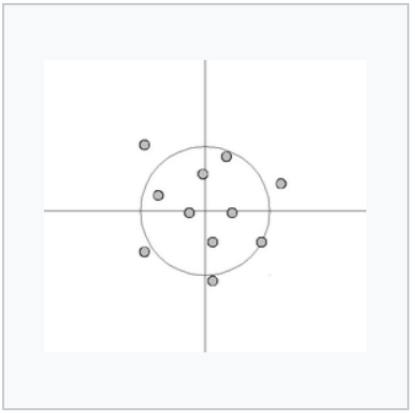


High bias, low variance



High bias, high variance

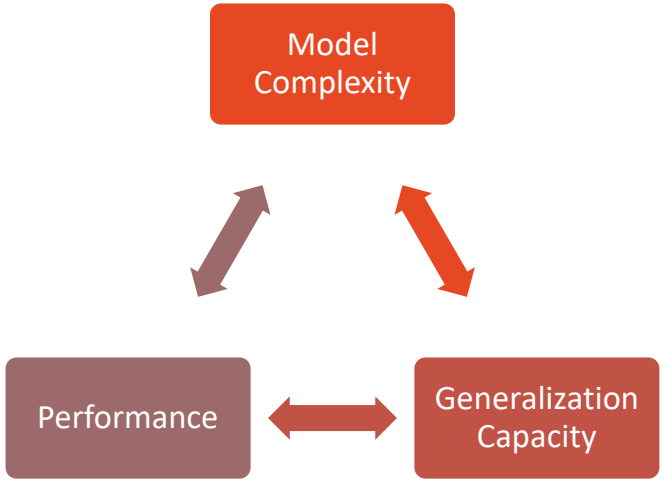
Low bias, low variance

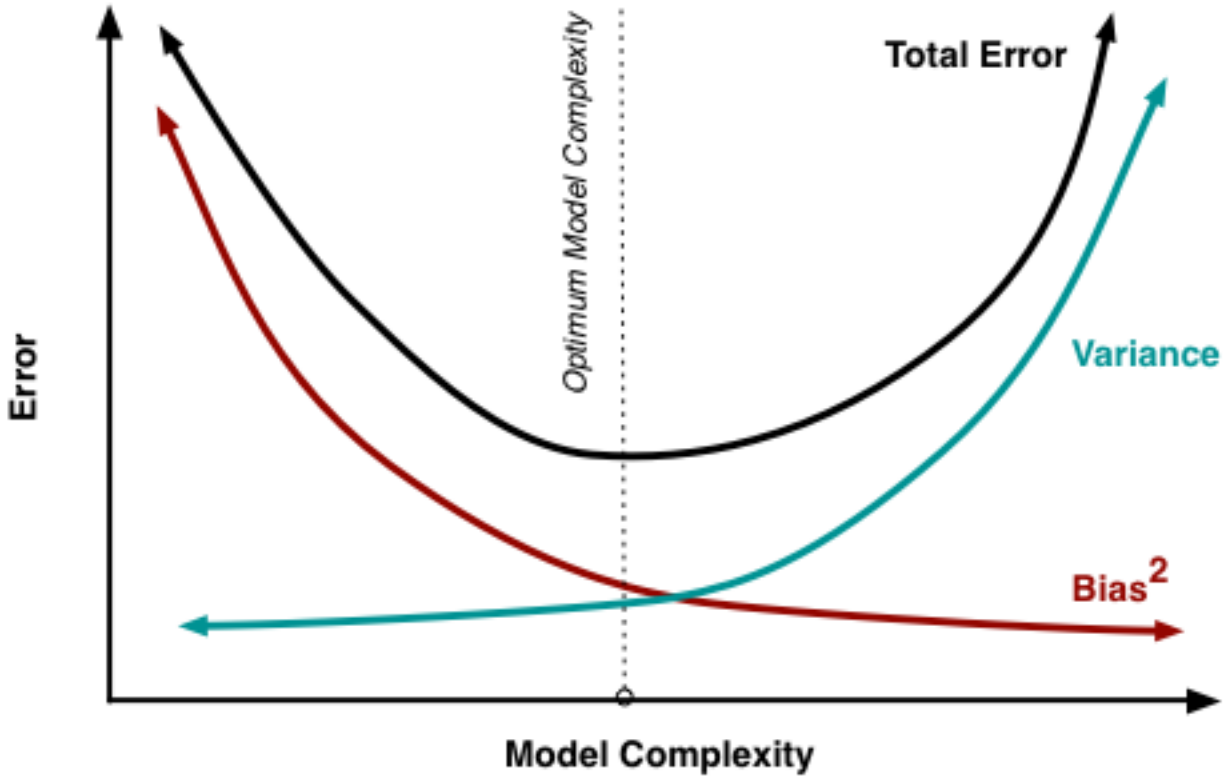


Low bias, high variance

Bias-Variance Trade-off

- Find a **balance** between **bias** (simplicity) and **variance** (complexity) to build a model that generalizes well to new data.





Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

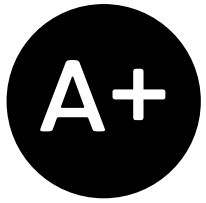
Train, Validation, Test Split

Training Set – Learn the task

- **Train** the model to learns patterns and features.

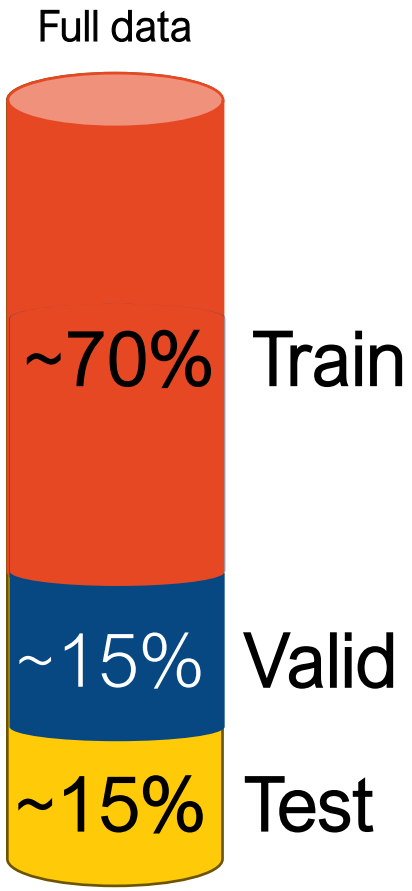
Validation Set – Select the model

- **Tune** hyperparameters.
- Prevents overfitting by analyzing on unseen data.



Test Set – Evaluate the model

- Evaluate the **final** model's performance.
- **Unbiased** assessment of the model.



Common Mistakes

- **Using Test Set for Tuning:** This is cheating!
  - Causes overfitting. Keep the test set separate for final evaluation.
- **Ignoring Data Leakage:** This is cheating!
  - Allows test data to influence training. Maintain strict separation of data.
- **Reusing Validation Data at Test Time:** This is cheating!
  - Biases test evaluations. Keep a distinct test set unused during training.
  - However, you can reuse validation for training time.

Avoid these to have an accurate view of the performance!

Common Mistakes

- **Non-Random Splitting:**
  - Leads to biased samples - Always shuffle data before splitting.
- **Insufficient Sample Size**:
  - Small splits yield unreliable results. Ensure adequate data in each split.
- **Lack of Stratification:**
  - Results in imbalanced splits. Use stratified sampling for class distribution.
- **Neglecting Validation Monitoring:**
  - Miss signs of overfitting. Track validation performance regularly.

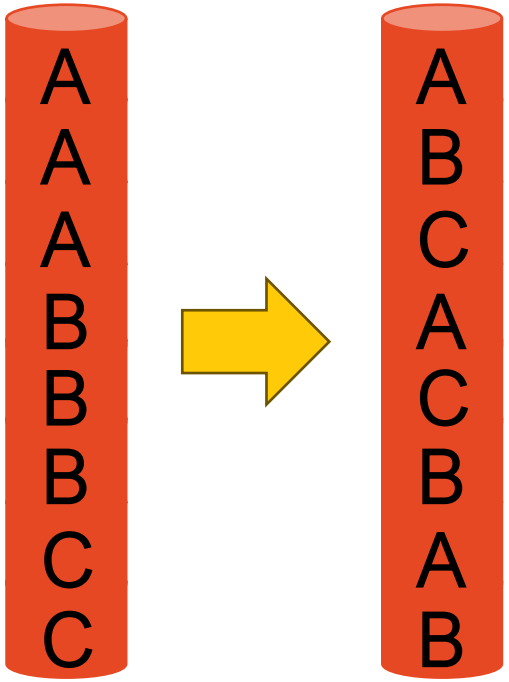
Avoid these to have an accurate view of the performance!

Shuffling (Random Splitting)

- **Randomly reorder** data before splitting:
  - Prevents unwanted data patterns.
  - Reduces cases where some segments may dominate others.

- Always recommended **before splitting data**.

Except: temporal order matters (e.g. time-series forecasting).



Stratified

- Random splitting **does not** guarantee same **class distribution**:
  - e.g. a low occurring class can be completely absent from testing

- **Stratification**:
  - Ensures that splits maintain the same proportion of each class:
    - e.g. A=50%, B=30% and C=20% the splits will also have the same proportions.
  - Prevents bias caused by under-representation

Especially important in the case of class imbalance.

Cross Validation

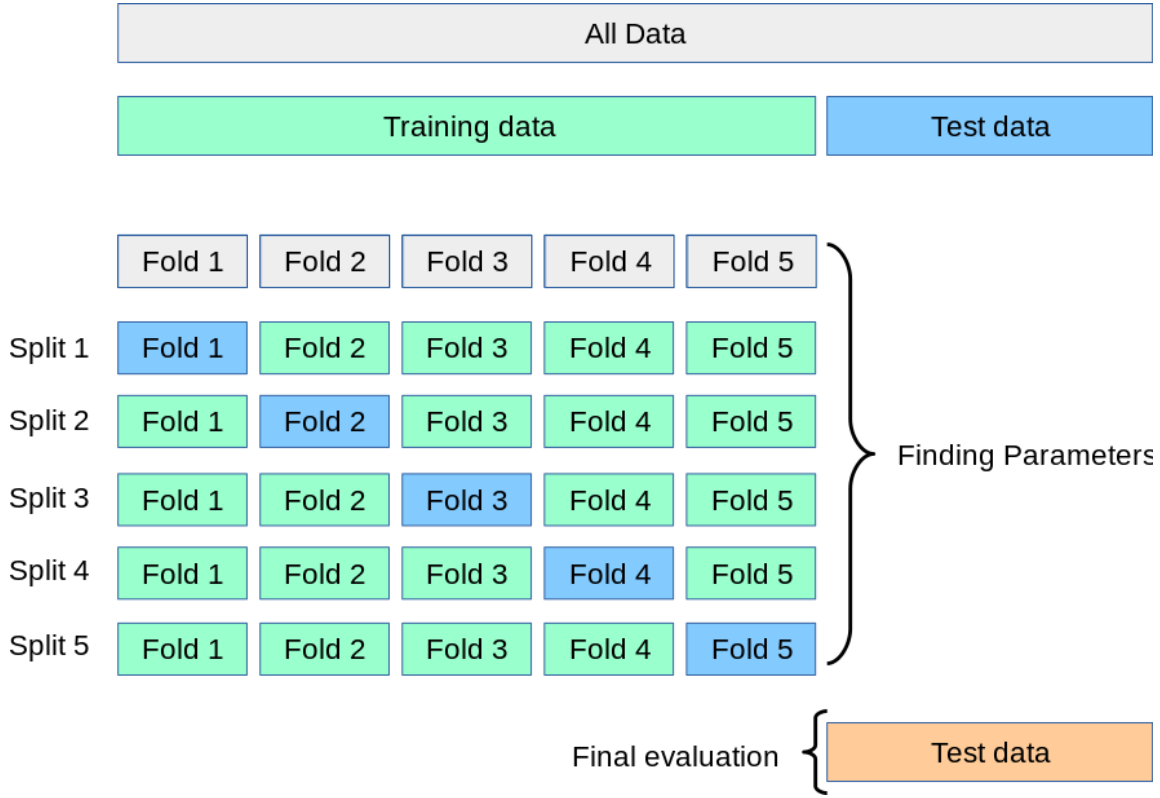
- Evaluate model performance by splitting the data into **multiple "folds"**.

What is a “fold”?

k-Fold Cross Validation

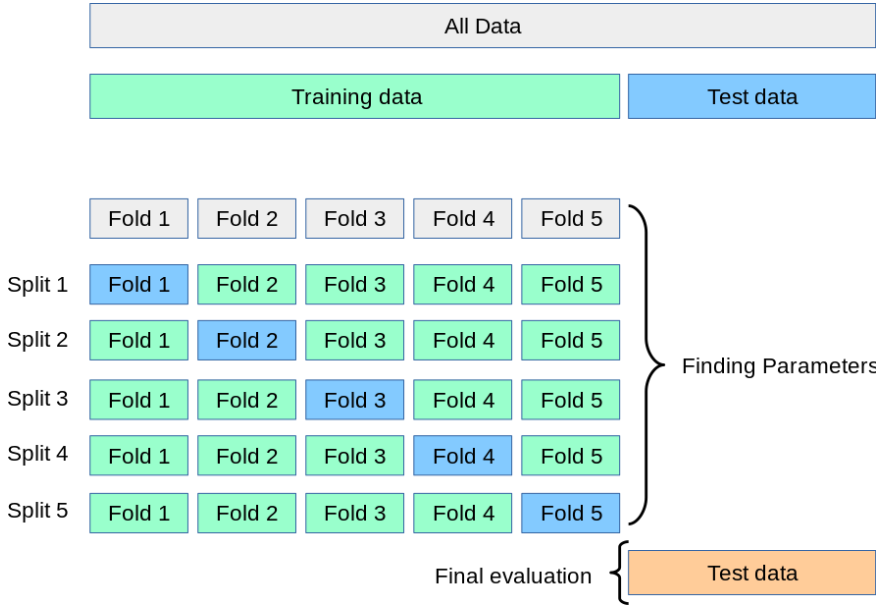
1. Split data into k folds.
2. For each iteration, k-1 folds are used for training, and the remaining used for validation.
3. Process repeats k times, ensuring each fold is used for validation once.
4. The final performance is the average of all k iterations.

Commonly, k is set to 5 or 10 folds.



Leave-One-Out Cross Validation

- Special case of K-Fold Cross-Validation where ***k* equals the number of data points.**



1. For each iteration, one data point is used for validation, and the rest for training.
2. Repeats for every data point
3. Final performance is the average of all iterations.

Maximizes training data

Computationally Expensive

Might Have High Variance

Cross Validation

- Benefits:
  - Reduces overfitting.
  - Ensures more stability and generalizability.
  - Provides a more robust estimate of model performance.

Hyperparameters (What makes them so Hyper?)

- **Hyperparameters** are settings that control the learning process and structure.
  - They are **set before training**.
  - They are **NOT learned** from data.

But can be tuned!

- Example hyperparameters:
  - **Batch Size** - Number of training examples used in one iteration – e.g. 2, 8, 16...
  - **Learning Rate** - Determines how fast a model learns – e.g 0.1, 0.01, 10^{-5} ...
  - **Number of Layers** - Defines the architecture of neural networks – e.g. 1, 2, 4 ...

Hyperparameter Tuning

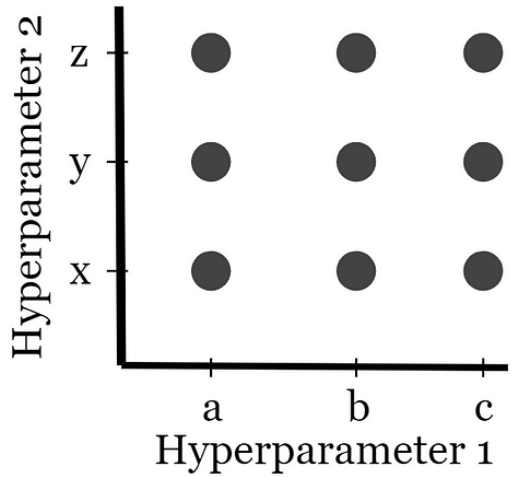
- **Select the optimal hyperparameters** that maximize a model's performance on unseen data.

- Can significantly **improve performance:**
  - Sometimes everything is correct except the hyperparameter choice.

- How to **select the values** for each hyperparameter?
  - Create a **range of values** for each and **test various combinations**.

Grid Search vs Random Search

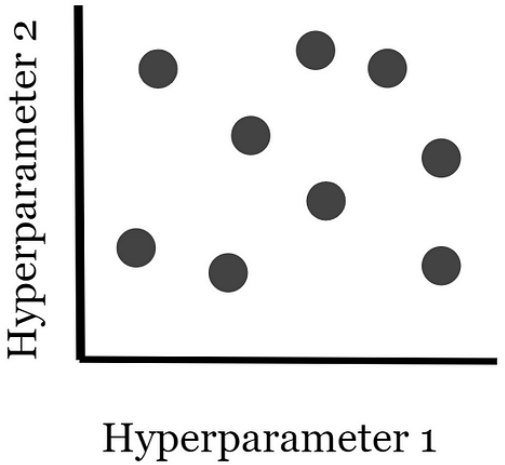
Grid Search



Explores **all combinations**

- Exhaustive method.
- Computationally expensive

Random Search



Samples from the search space

- Faster and more efficient.
- May miss the best possible combination.

As always, we do this outside the test set!

Data Cleaning

- Handling **missing values**: Old friend NaN
  - Removing them.
  - Replacing with mean, median, mode, or KNN imputation.
- **Removing duplicates**:
  - Eliminating duplicate records to avoid biased results.
- **Handling outliers**:
  - Remove via a threshold.
  - Transform to some minimum or maximum value.

Data Transformation:

- Converts **raw data** into a **suitable format.**
  - Improves model performance by **uniformization** of the features.

- **Feature scaling:**
  - Rescales features to a specific range (e.g., 0 to 1), preventing from dominating.
- **Log Transformation:**
  - Reduces the impact of large values or skewed data.
- **Binning:**
  - Converts continuous variables into categorical "bins" or intervals.
  - e.g. ages 10-20, 21-30, >30

Feature Scaling

- **Standardization (Standard Scaler)**
  - Transforms the data to have a **mean of 0** and a **standard deviation of 1** (unit variance).

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma}$$

- **Normalization (MinMaxScaler)**
  - Transforms the data by scaling it to a specific range, typically [0, 1]

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

- **MaxAbs Scaler**
  - Scales data by dividing by the maximum absolute value, ensuring that all values are in the range [-1,1].

$$x_{\text{scaled}} = \frac{x}{|x_{\max}|}$$

Encoding Categorical Data

- Converting categorical variables into **numerical formats** that machine learning algorithms can process.

- One-Hot Encoding:
  - Creates binary columns for each category.
- Label Encoding:
  - Assigning numerical values to categories.
- Ordinal Encoding:
  - Categories have an **inherent order**.

"Red", "Green", "Blue" → [1,0,0], [0,1,0], [0,0,1]

"Red"=1, "Green"=2, "Blue"=3

"Low"=1, "Medium"=2, "High"=3

Feature Engineering

- Creating **new features from existing ones:**
  - **Combining features** – e.g. adding or multiplying
  - **Decomposing** – e.g. extracting day, month, and year from a date
  - **Binning** – creating bins for continuous data

Reduces Noise

More Informative Features

Capture More Complex Patterns

Feature Selection

Features

Data Scientist

"Gardener" - Identifies and selects a subset of relevant features!

Why Feature Selection?

- Not all features are **equally useful** – there is noisy data.
- Predicting House Prices features:
  - Square footage, number of room, location ...
  - Owners favorite color Irrelevant or is it?

- Why?
  - Reduces overfitting by eliminating irrelevant or redundant features.
  - Improves model performance and accuracy.
  - Decreases training time by reducing the dimensionality of the dataset.

Feature Selection Methods:

- **Filter Methods:**
  - Evaluate features based on statistical tests.
- **Wrapper Methods:**
  - Use a specific model to evaluate subsets of features.
- **Embedded Methods:**
  - Perform feature selection during the model training process.

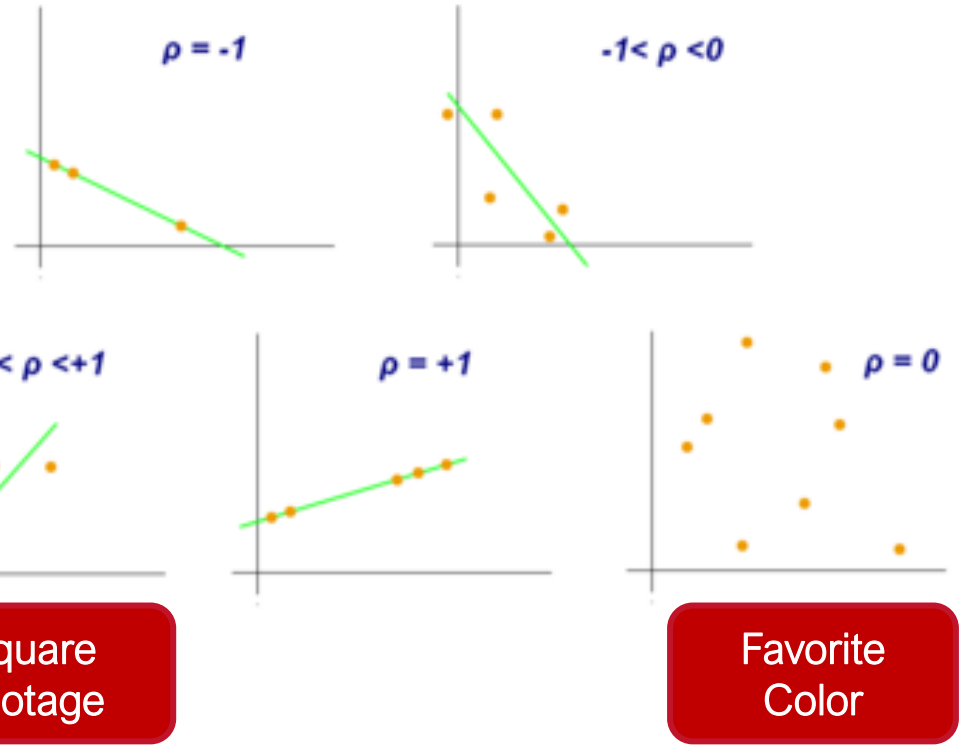
As always, we do this outside the test set!

Filter Methods - Correlation

- Evaluates the **strength** and **direction** of the linear relationship:
  - Select by applying a threshold to remove low correlation features.

- Easy to interpret and apply.
- Only captures linear relationships.
- Does not account for interaction between features.

Person Correlation



Square Footage

Favorite Color

Filter Methods – Chi-Squared (χ^2)

- Evaluates the **independence of categorical features** from the target variable.

$$\chi^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}$$

Assumes independence between categories

  - where O_i is the observed frequency and E_i is the expected frequency.
  - Higher Chi-Squared values indicate a stronger association between the feature and the target variable.

- Good for categorical datasets.
- Does not work well with many categories.

Filter – Chi-Squared (χ^2)

$$\chi^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}$$

- 1 Million residents from neighborhoods A to D.
- Let's check if their occupation influences their residence neighborhoods.

1. Take a random sample:

	A	*B*	*C*	*D*	Total
White collar	90	60	104	95	349
Blue collar	30	50	51	20	151
No collar	30	40	45	35	150
Total	**150**	**150**	**200**	**150**	**650**

2. Calculate Expectation and χ^2 for each one:

$$150 \times \frac{349}{650} \approx 80.54$$

$$\frac{(\text{observed} - \text{expected})^2}{\text{expected}} = \frac{(90 - 80.54)^2}{80.54} \approx 1.11$$

3. Sum all values

$$\approx 24.57.$$

Value is quite large (according to degrees of freedom table), and therefore reject null hypothesis of not being correlated.

https://en.wikipedia.org/wiki/Chi-squared_test

Filter – ANOVA F-Test

- Evaluates if there are significant differences between the **means of three or more groups**:

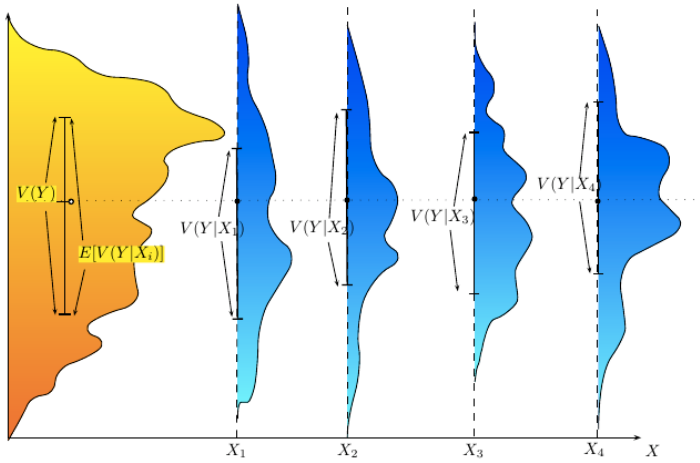
$$F = \frac{\text{between-group variability}}{\text{within-group variability}}. \qquad \frac{\sum_{i=1}^{K} n_i (\bar{Y}_{i\cdot} - \bar{Y})^2 / (K-1)}{\sum_{i=1}^{K} \sum_{j=1}^{n_i} \left(Y_{ij} - \bar{Y}_{i\cdot}\right)^2 / (N-K)}$$

  - where Y_i denotes the sample mean in the i-th group, n_i is the number of observations in the i-th group, Y denotes the overall mean of the data, and K denotes the number of groups.

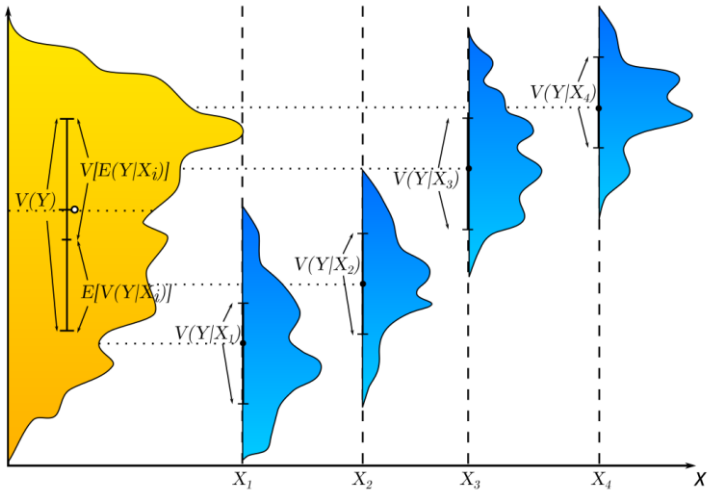
- A higher F-statistic indicates that the group means are significantly different.

Filter – ANOVA F-Test

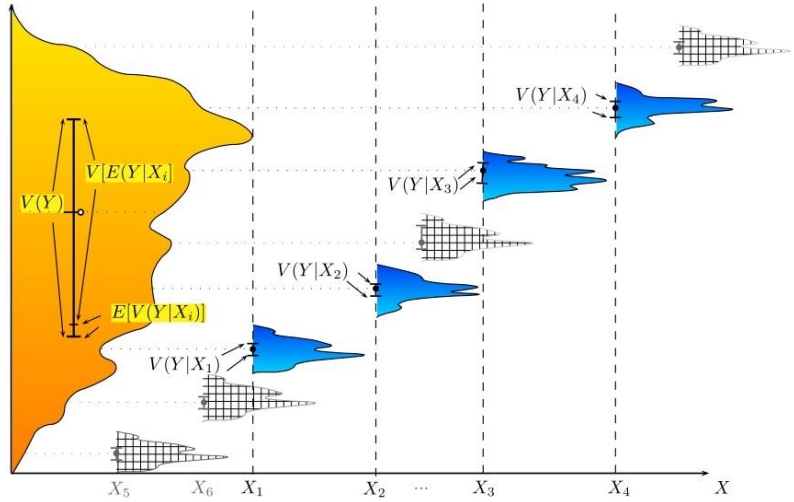
- Relationship between dog weight and various features:



No fit: Short-haired vs long-haired



Fair fit: Pet vs Working breed

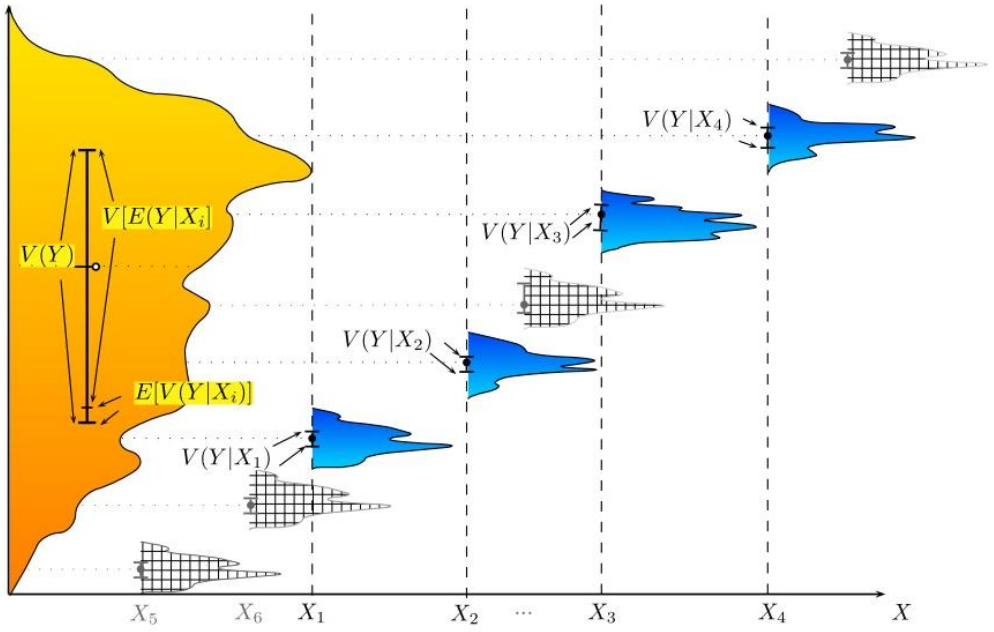


Very good fit: Weight by breed

Filter – ANOVA F-Test

- Advantages:
  - Compares **multiple groups** simultaneously.
  - Helps **identify features** that contribute to differences in target variable means.

- Disadvantages:
  - Assumes **normally distributed** groups and **equal variances** (homoscedasticity).
  - **Sensitive to outliers**, which can skew results.



Very good fit: Weight by breed

Wrapper Methods

- Evaluates the **usefulness of subsets of features** based on their impact on a model's performance.

- Process:
  1. Consider a set of features
  2. Train model with those features
  3. Evaluate performance
  4. Change the feature set based on the performance
  5. Repeat step 1 until some criteria is reached

As always, we do this outside the test set!

Wrapper Methods

- **Forward Selection:**
  - **Adds features one at a time** based on performance improvement.
- **Backward Elimination:**
  - **Removes the least significant** features iteratively from a full set.
- **Recursive Feature Elimination** (RFE):
  - Trains the model repeatedly, removing the least important
    - For Tree-Based Models (e.g., Decision Trees, Random Forests): Gini impurity or information gain.
    - For Linear Models (e.g., Linear Regression, Logistic Regression): Absolute value of the coefficients.

0,1,2...n

n, n-1, n-2, ...

Model Specific

Wrapper Methods

- Advantages:
  - Captures **interactions between features**, improving model performance.
  - **Tailored** to a specific model for better accuracy.

- Disadvantages:
  - **Computationally intensive**, especially with large datasets.
  - **Risk of overfitting** due to dependence on the chosen model.

Embedded (Intrinsic) Methods

- Performs feature selection as **part of the model training**.
- Examples:
  - **Lasso Regression**: Can shrink some coefficients to zero, effectively performing feature selection.
  - **Decision Trees**: Splits based on feature importance.

More on this in the future!

Conclusion

- Use the **right model** – model selection.
- Use the **right hyperparameters** – hyperparameter tunning.
- Use the **right data** – data preprocessing and feature selection.

- **Key Takeaway:**

Combine these to have more accurate and efficient models.

Scikit-learn

- Python library for **machine learning and data analysis**.

- Provides easy-to-use tools for:
  - classification
  - regression
  - clustering
  - dimensionality reduction
  - model selection
  - and many more...

Follow Along!

Course Shared Folder - Google Drive
session_6a_datascience_cmu.ipynb

Use-Case Details

Requirements - One operation of each of the following:

- Dataset Descriptive Statistics
- Data Cleaning (e.g. checking for NaNs, column removal, etc.)
- Model Selection, Feature Engineering, and Normalization
- Plotting (frequency, correlation between feature pairs)
- Supervised Learning:
  - Training a linear classifier
  - Evaluate its performance over multiple metrics

Use-Case Discussion Session

Let's simulate a Data Science team discussion:

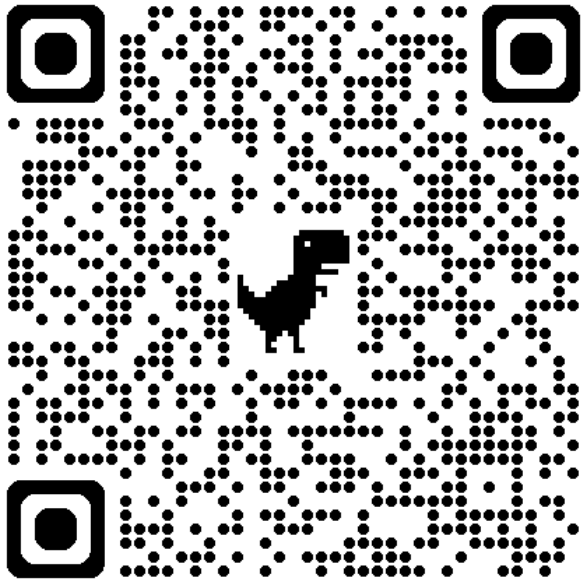
- Bring your expertise and point of view!

We Invite all groups to present and discuss their use-case with the class:

- Show and discuss your notebook to the class
- 5 to 7 minutes per presentation

Use Case – Form Filling

- Fill in this form with your group's information.
- Only one person from the team needs to fill in the information.
- Presentations on the 18th of October (next Friday).



CMU Portugal Academy

03

Unsupervised Learning

Carnegie Mellon Portugal | TÉCNICO LISBOA | Ciências ULisboa | NOVA SCHOOL OF SCIENCE & TECHNOLOGY

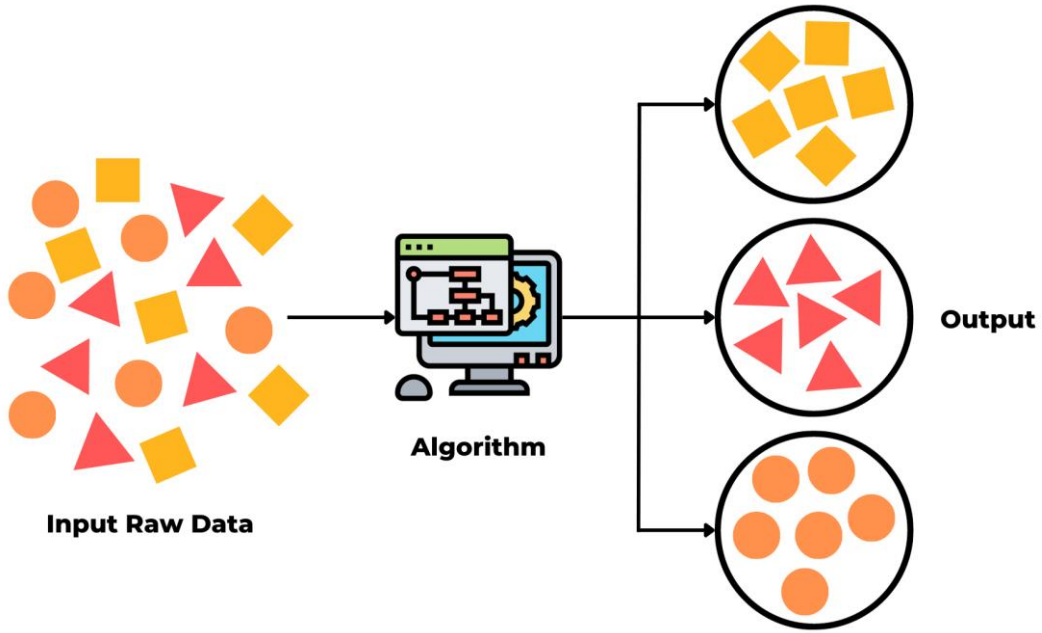
Unsupervised Learning

Assorted Legos

Neatly Organized

What is Unsupervised Learning?

- A type of machine learning where the model is trained on data **without labeled responses**.
- The goal is to **identify patterns or groupings** within the data.



https://eastgate-software.com/what-is-unsupervised-learning/

Difference to Supervised Learning

Aspect	Supervised Learning	Unsupervised Learning
Definition	Trains models using **labeled data** (input-output pairs).	Trains models using **unlabeled data** to find patterns.
Objective	Predict outcomes based on **input features**.	Discover **hidden structures** or groupings in data.
Data Requirement	Requires a **labeled dataset** for training.	Does **not require labeled data**; works with raw data.
Common Algorithms	Linear regression, logistic regression, decision trees, support vector machines.	K-means, hierarchical clustering, PCA, t-SNE.
Applications	Spam detection, image classification, predictive analytics.	Customer segmentation, anomaly detection, topic modeling.

Why Unsupervised Learning?

- Discovers **hidden patterns**.
- Identifies **clusters**.
- **No labeled data** required.

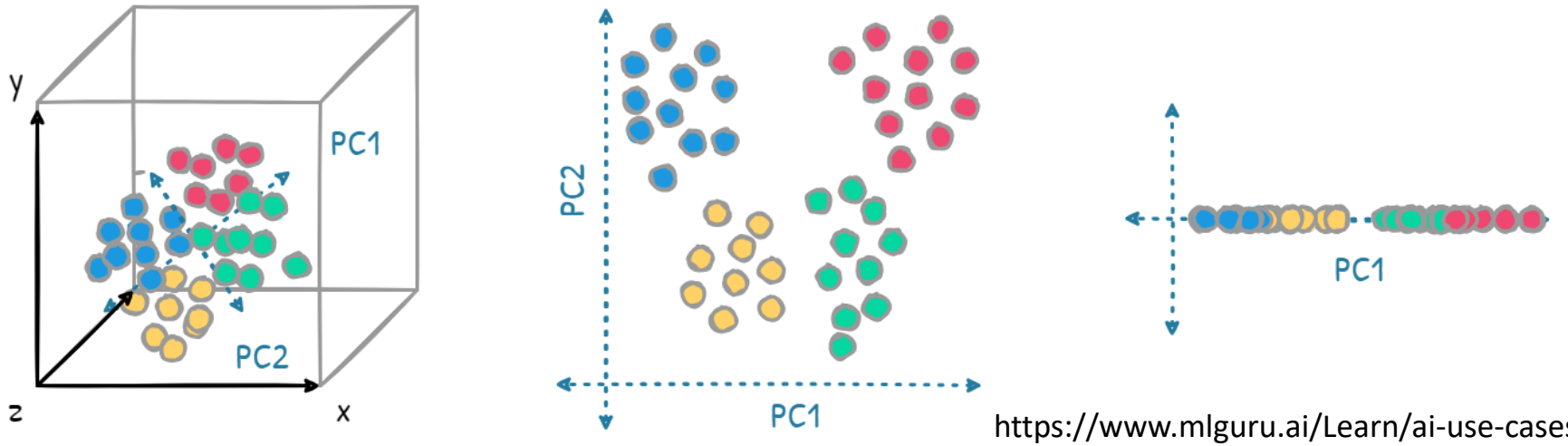
Market Segmentation

- Detects **outliers**.

Bank Fraud Detection

What is Dimensionality Reduction?

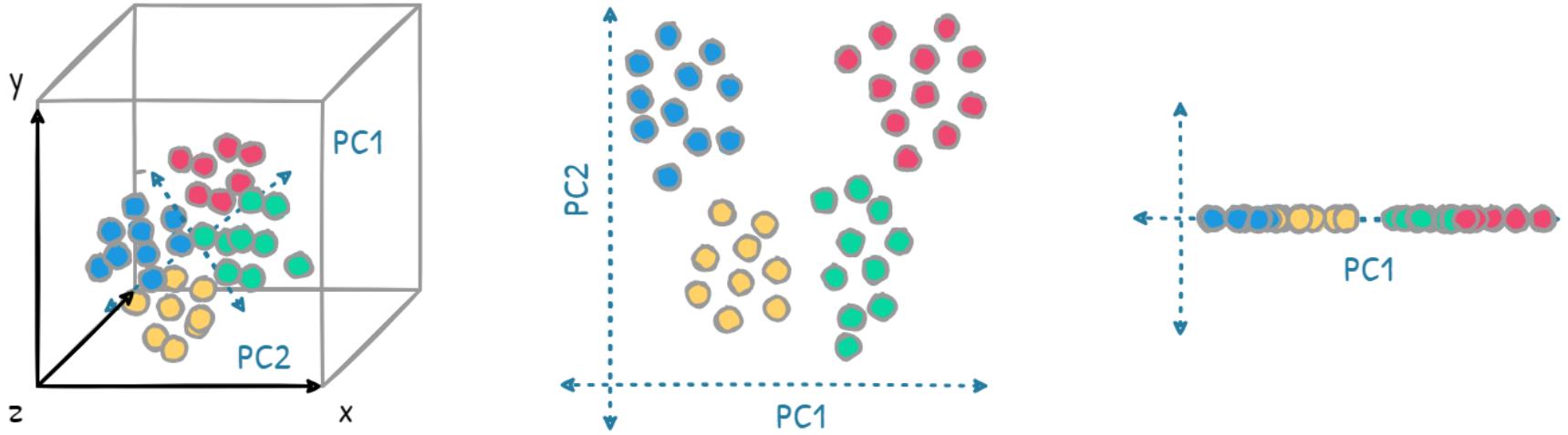
- The process of **reducing the number of input features** in a dataset while preserving important information.



https://www.mlguru.ai/Learn/ai-use-cases-dimensionality-reduction

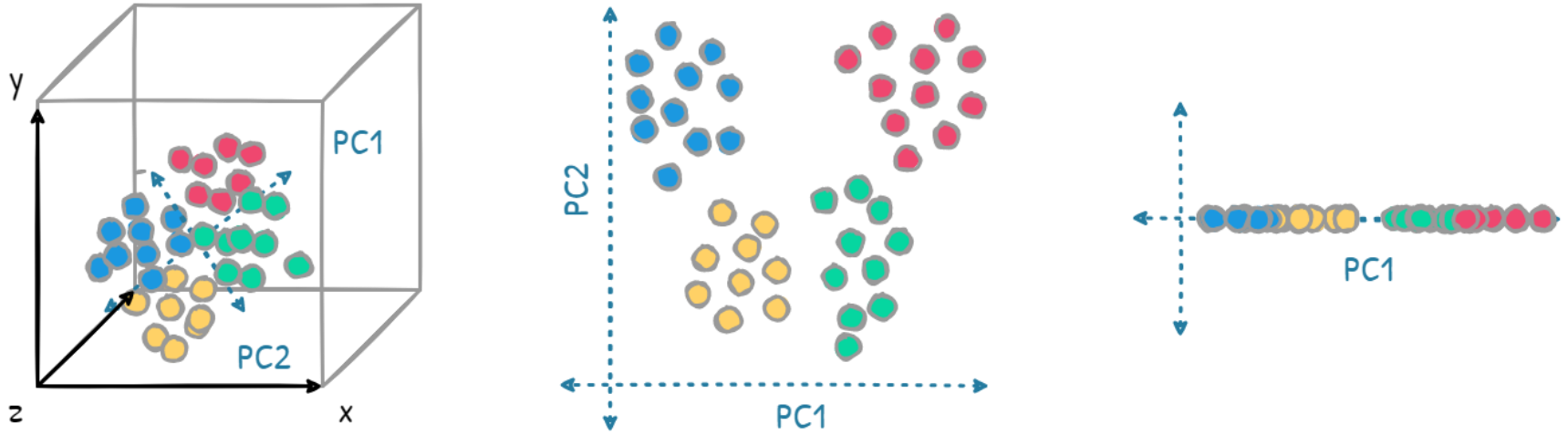
Why use Dimensionality Reduction?

- Reduces **computational complexity**.
- Helps **prevent overfitting** by removing irrelevant features.
- Makes **visualization** of high-dimensional data possible.



Principal Component Analysis (PCA)

- Transforms original features into a smaller set of new, uncorrelated features called **principal components.**
- Tries to preserve as much variance between features as possible.



Principal Component Analysis (PCA)

1. **Standardize** the data:

$$x_{\text{standardized}} = \frac{x - \mu}{\sigma}$$

2. Compute the **Covariance Matrix**:
   - X is the matrix of standardized features, and n is the number of points.

$$\text{Cov}(X) = \frac{1}{n-1} X^T X$$

3. Calculate the **Eigenvectors and Eigenvalues**:
   - where v is an eigenvector and λ is the corresponding eigenvalue.

$$\text{Cov}(X) \cdot v = \lambda v$$

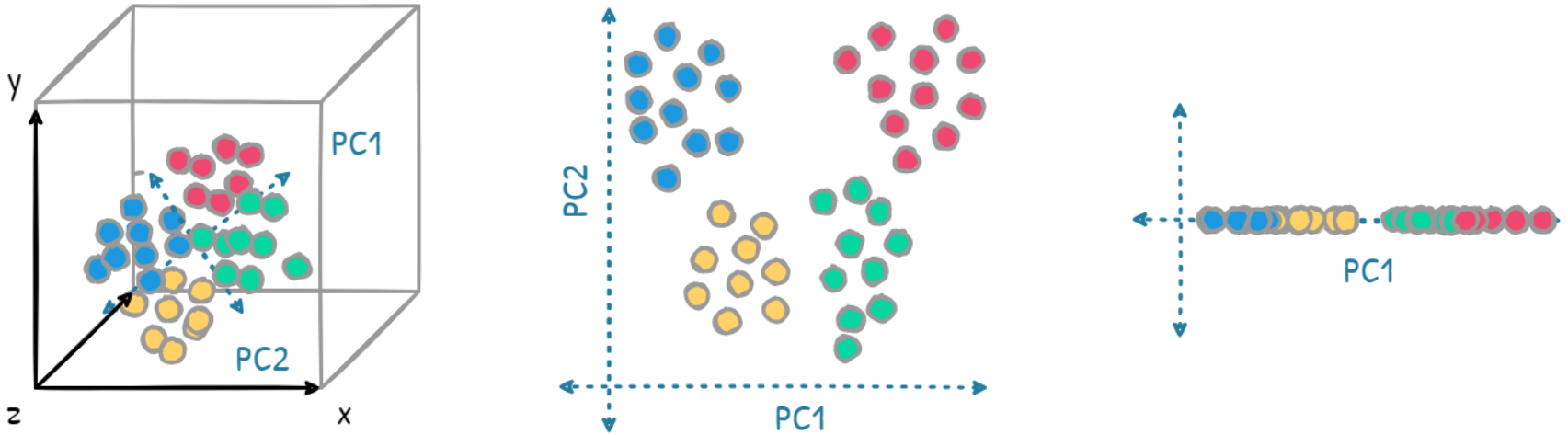
4. **Order the eigenvalues** in descending order and select the top k eigenvectors.
5. **Project the data** onto the new feature space:
   - Where Z is the transformed data, X is the original data matrix, and V_k is the matrix of the top k eigenvectors.

$$Z = X \cdot V_k$$

More on this in the future!

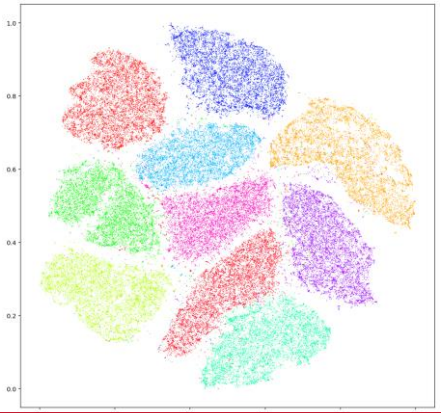
Principal Component Analysis (PCA)

- Project the data onto the new feature space:



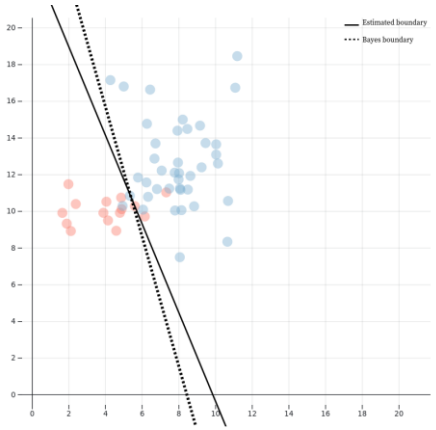
Other Dimensionality Reduction Methods

- t-Distributed Stochastic Neighbor Embedding (**t-SNE**):
  - Nonlinear dimensionality reduction.
  - Visualizes high-dimensional data in 2D or 3D.
  - Primarily used for data exploration.



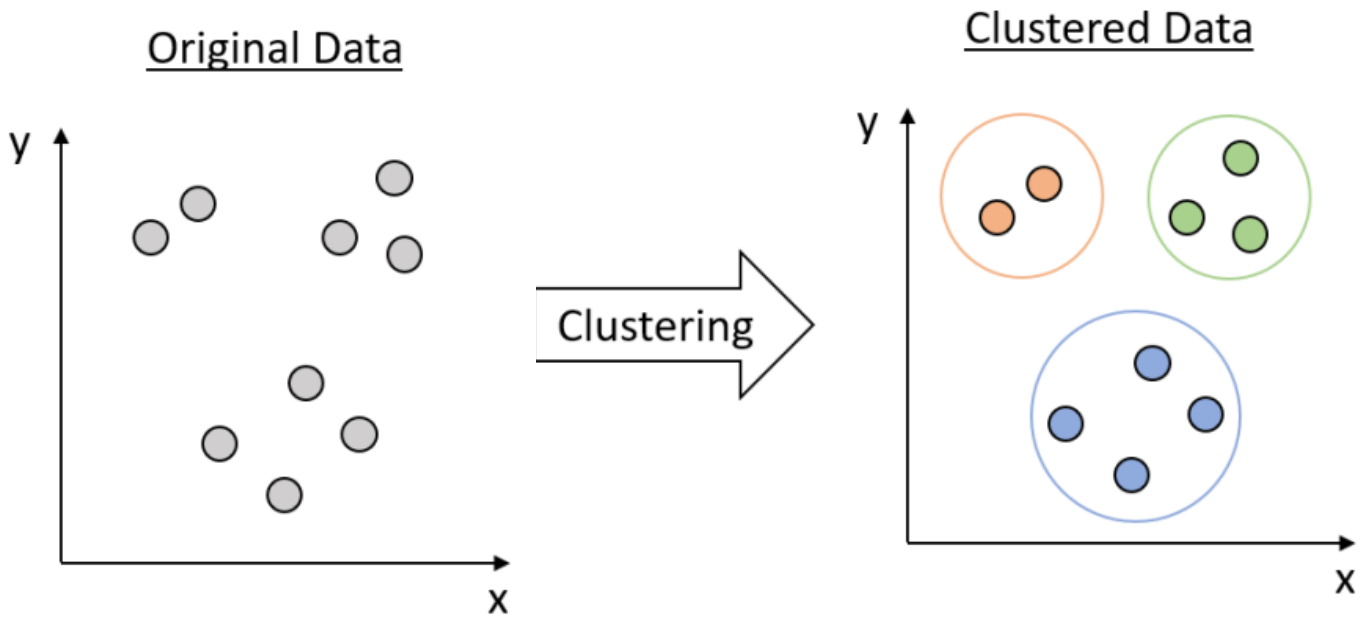
More on this in the future!

- Linear Discriminant Analysis (**LDA**):
  - Supervised dimensionality reduction.
  - Finds linear combinations of features to separate classes.



Clustering

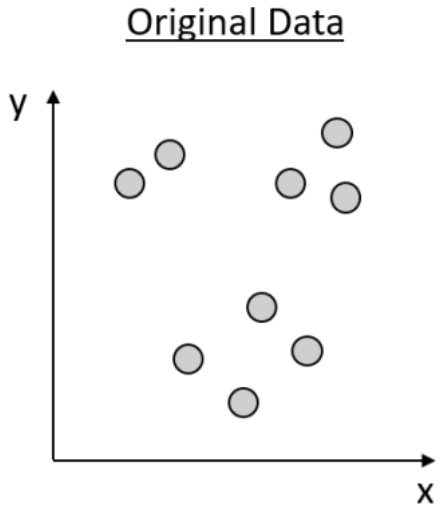
- Unsupervised technique that **groups similar data points** together.

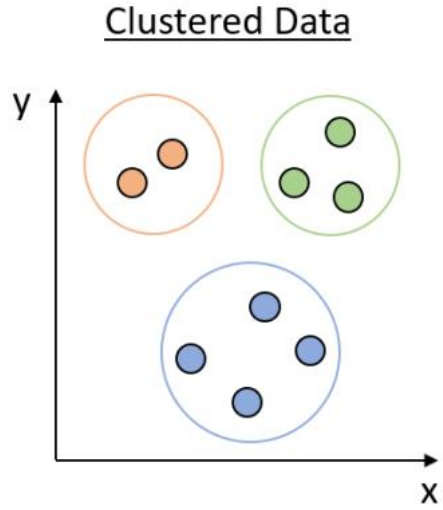


Clustering

- Data points in the same cluster are **more similar** to each other than to those in other clusters.

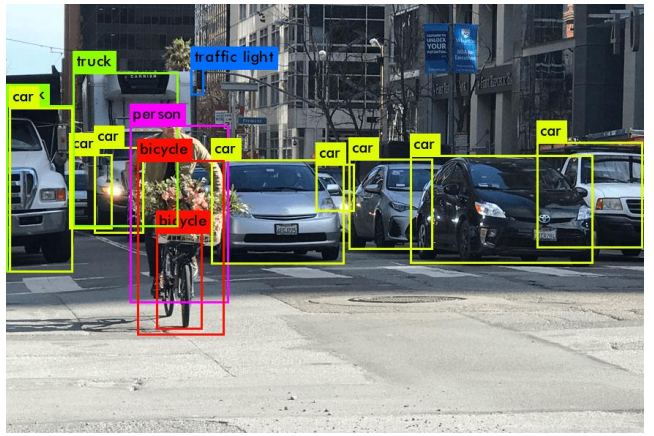
- Operates on **unlabeled data** to discover inherent groupings.

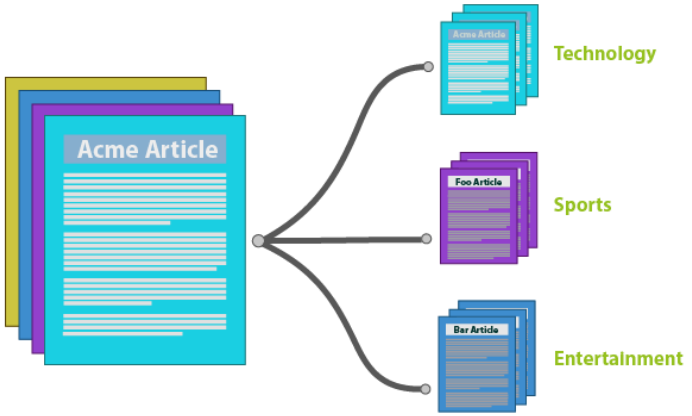




Clustering Applications

- **Market Segmentation**: Identifying distinct customer segments for targeted marketing.
- **Image Segmentation:** Grouping pixels in images for object detection.
- **Document Clustering:** Organizing text documents into topic-based clusters.





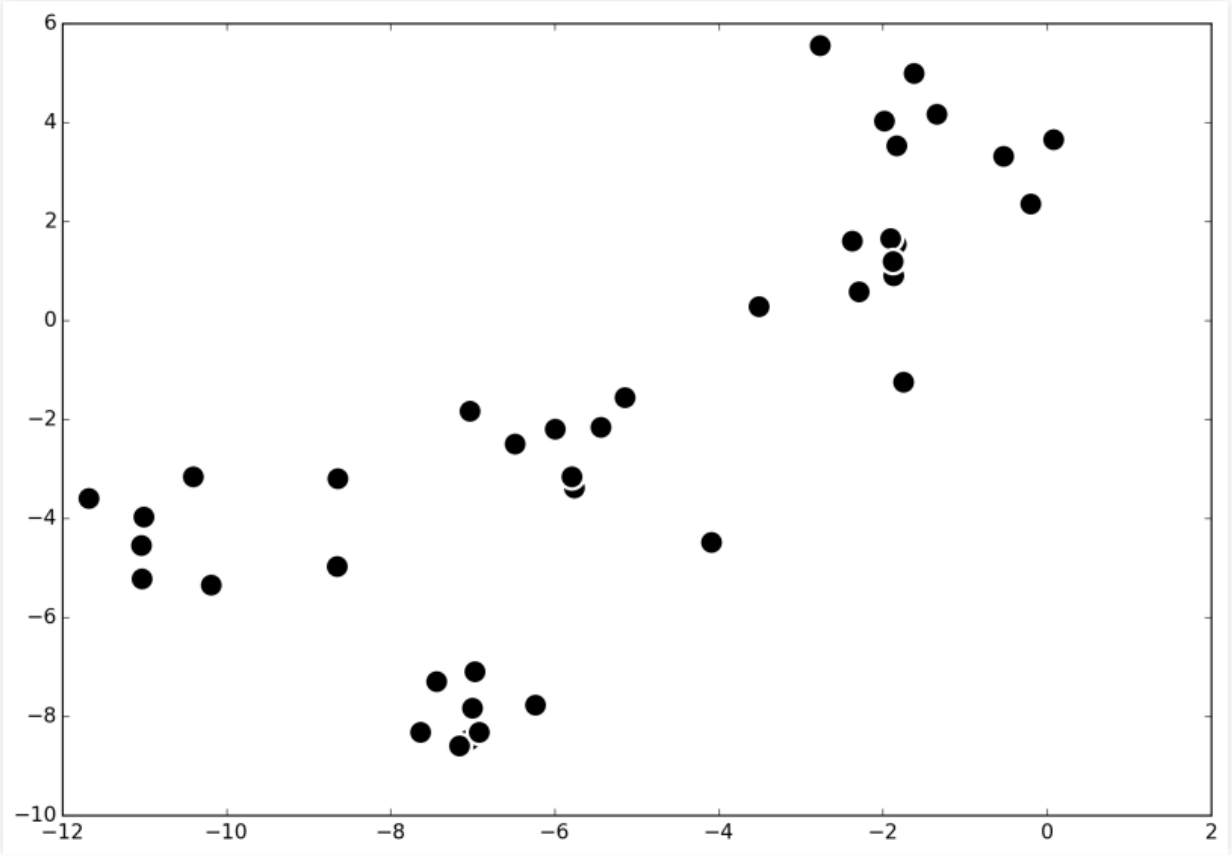
K-Means

- Partitions data into **K distinct clusters** based on feature similarity.

1. **Initialization**: Select K initial **centroids** randomly from the dataset.
2. **Assignment Step**: Assign each data point to the **nearest centroid**, forming K clusters.
3. **Update Step**: Calculate the **new centroids** by taking the mean of all points in each cluster.
4. **Repeat**: Iterate the assignment and update steps until the centroids stabilize (i.e., no significant change).

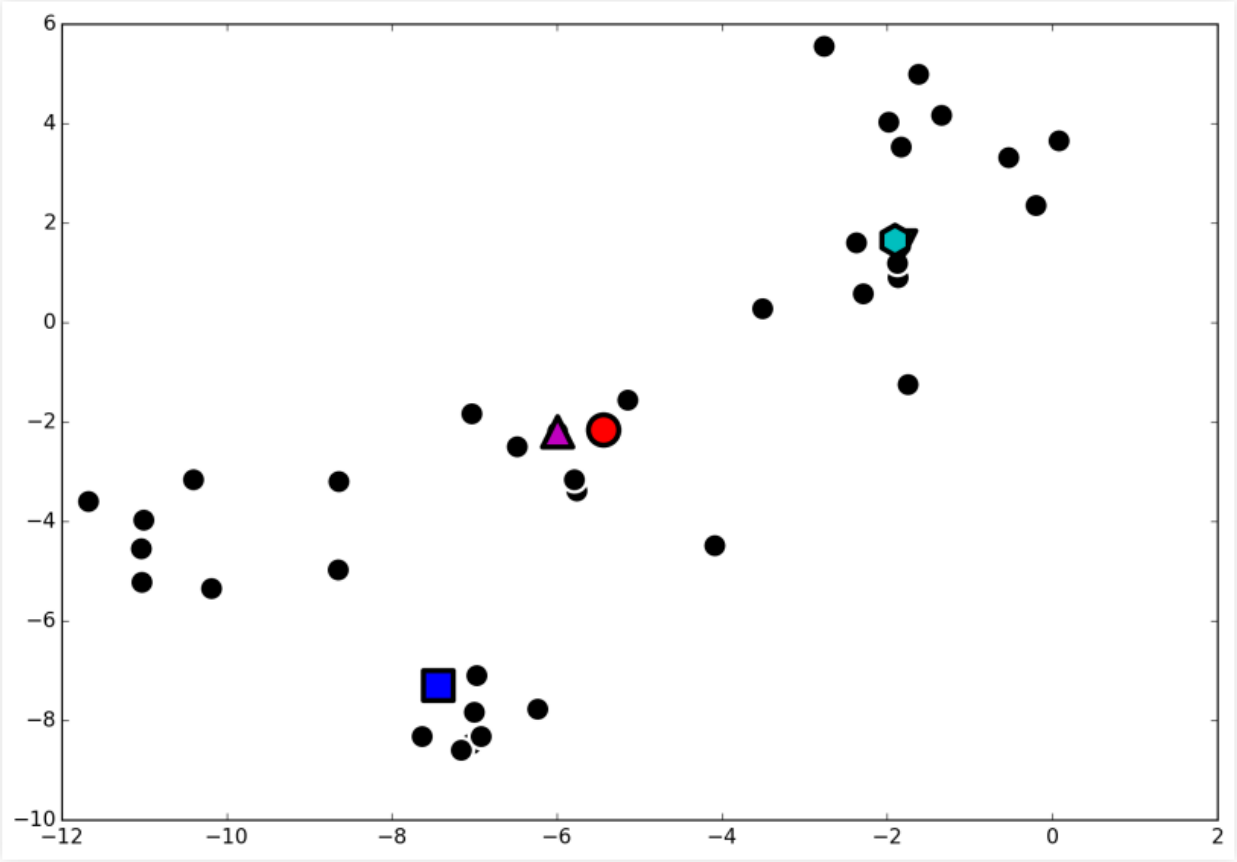
K-Means

- Start with a random set of examples.



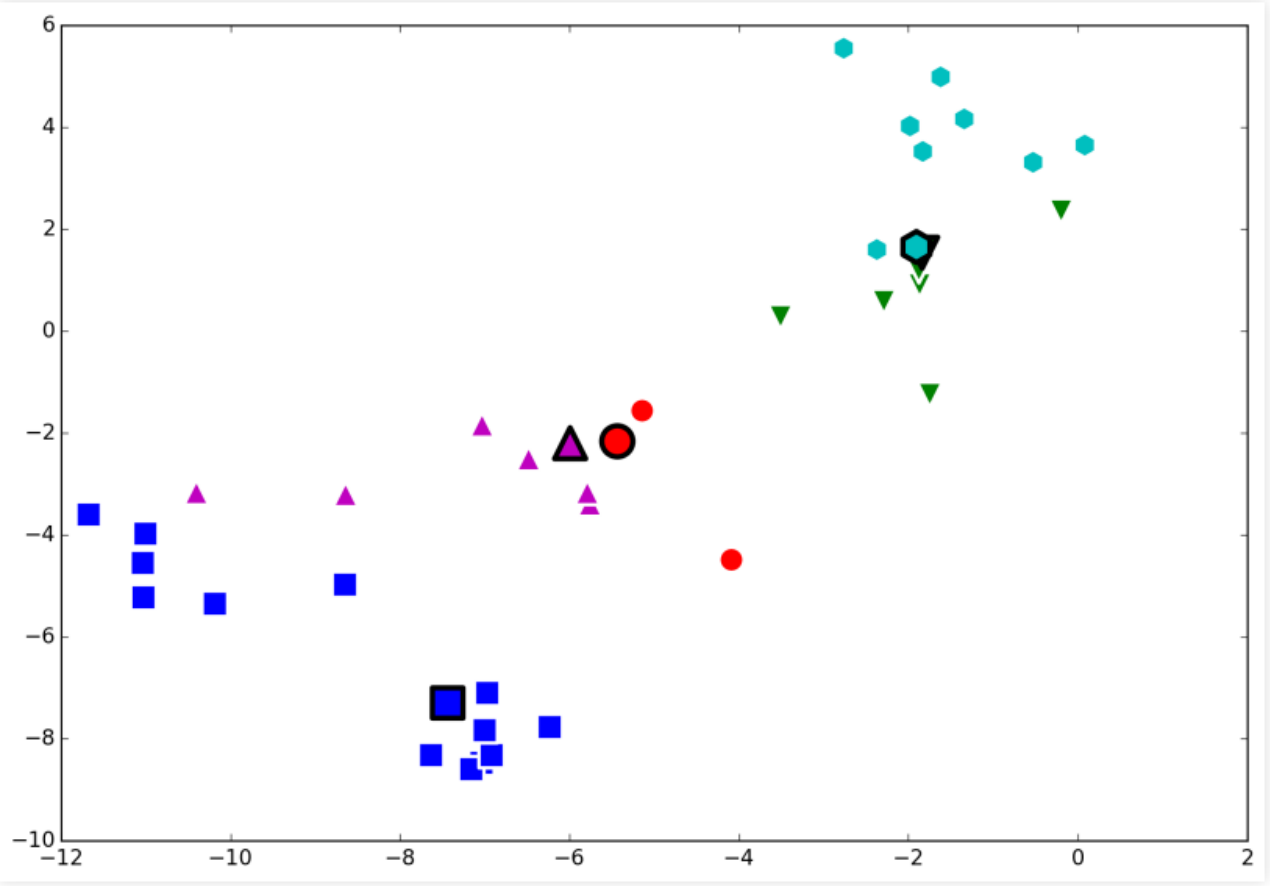
K-Means

1. **Initialization**: Select K initial **centroids** randomly from the dataset.



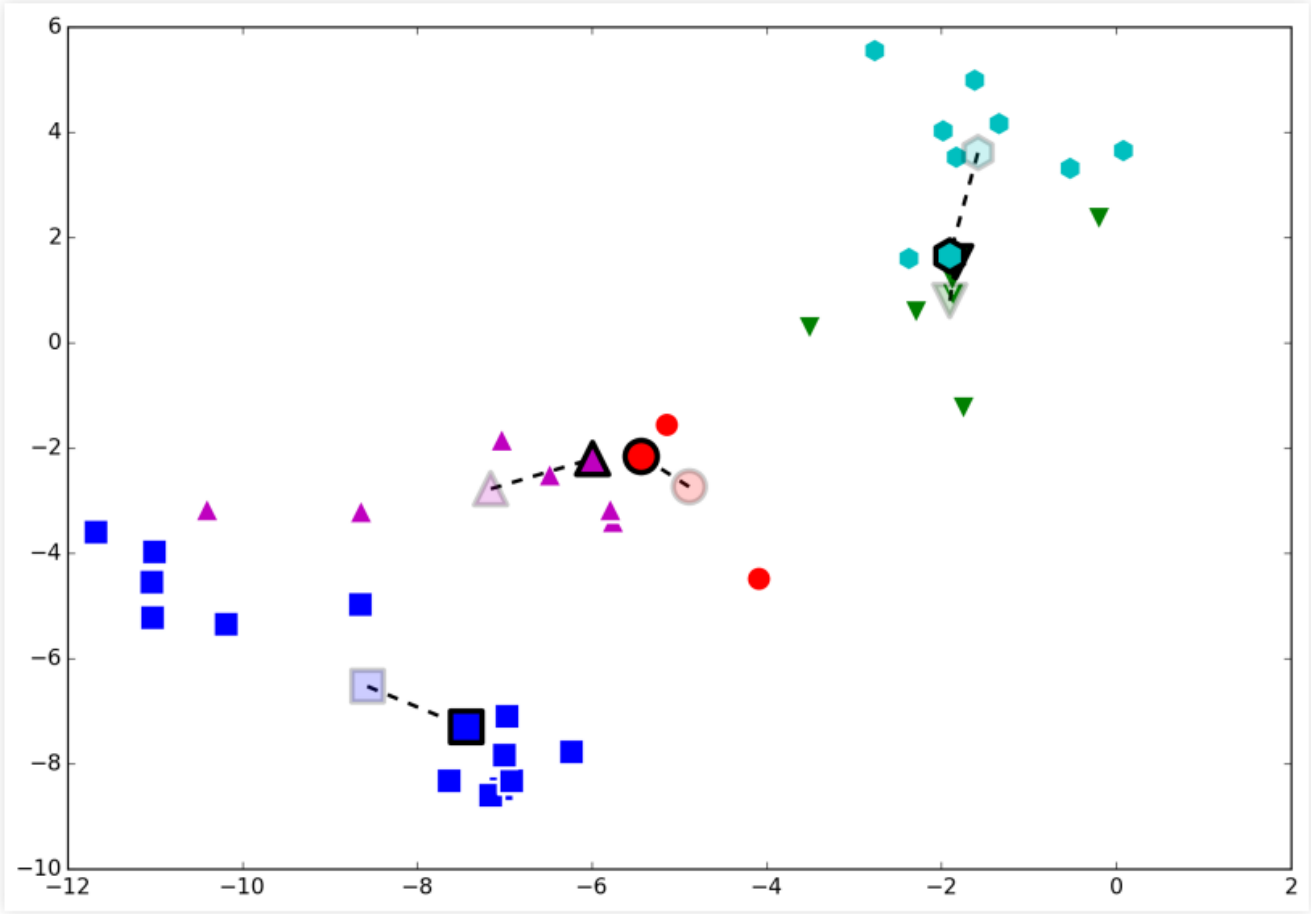
K-Means

2. **Assignment Step**:
Assign each data point to the nearest centroid, forming K clusters.



K-Means

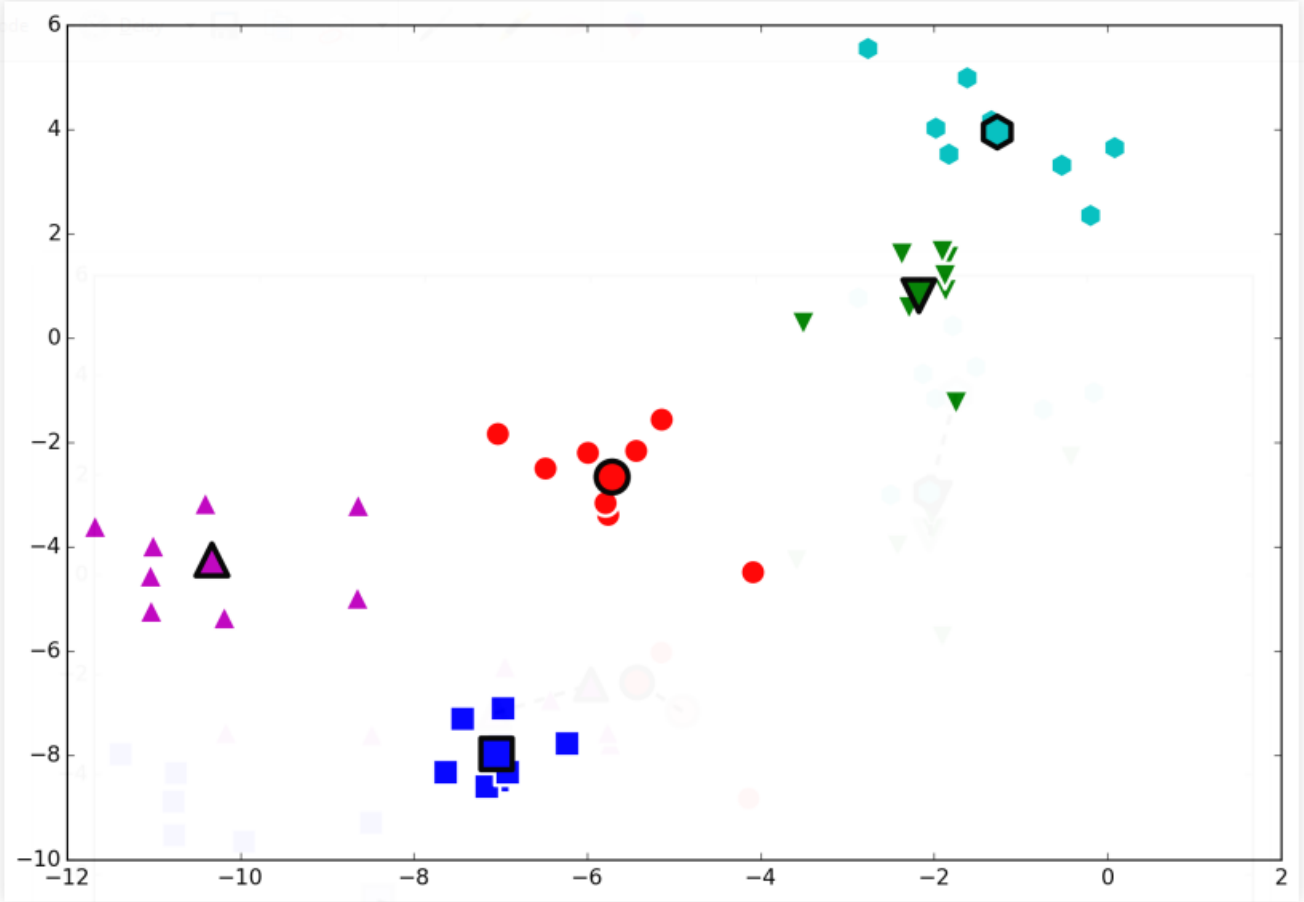
3. Update Step: Calculate the new centroids by taking the mean of all points in each cluster.



K-Means

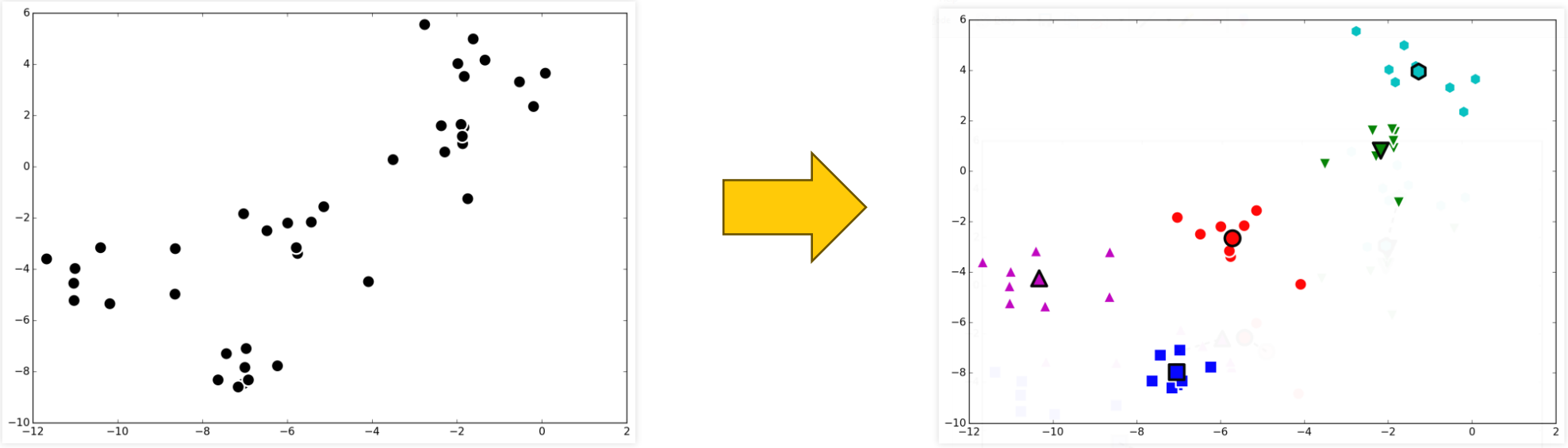
4. Repeat: Iterate the assignment and update steps until the centroids stabilize (i.e., no significant change).

- Interactive example here
- Video example here



K-Means

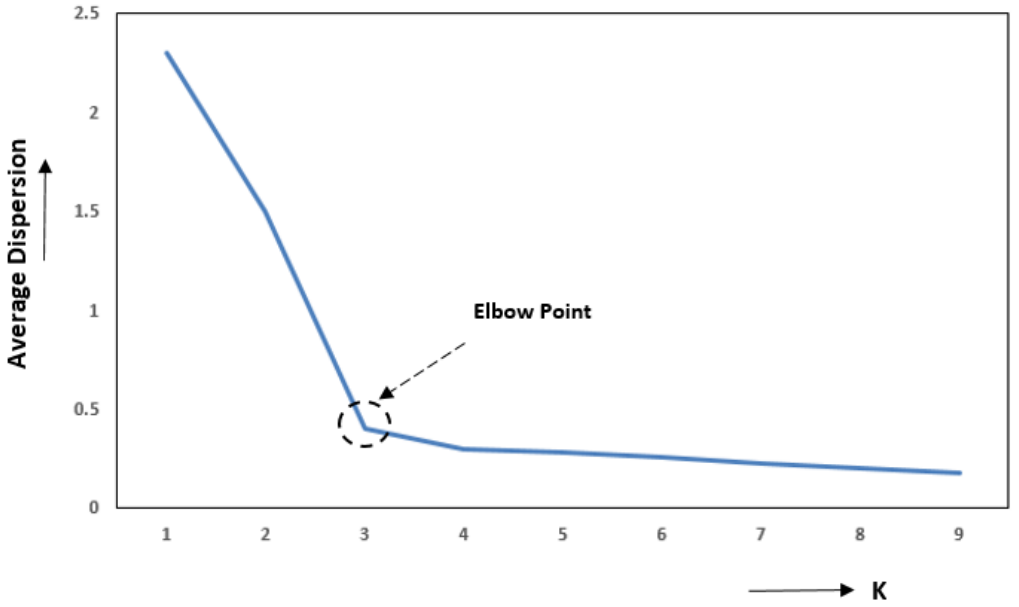
- Requires the number of clusters K to be **specified beforehand**.
- Sensitive to the **initial placement** of centroids and outliers.



Elbow Method

- K needs to be specified beforehand...
- The **Elbow Method** helps in selecting the optimal number of clusters K:
  1. Perform K-Means with various K (e.g., from 1 to 10).
  2. For each K, calculate the within-cluster sum of squares (WCSS), which represents the total variance within each cluster.
  3. Plot the number of clusters K on the x-axis and WCSS on the y-axis.
  4. Look for a "bend" or "elbow" in the curve where the rate of decrease in WCSS slows significantly.

How to select this value?

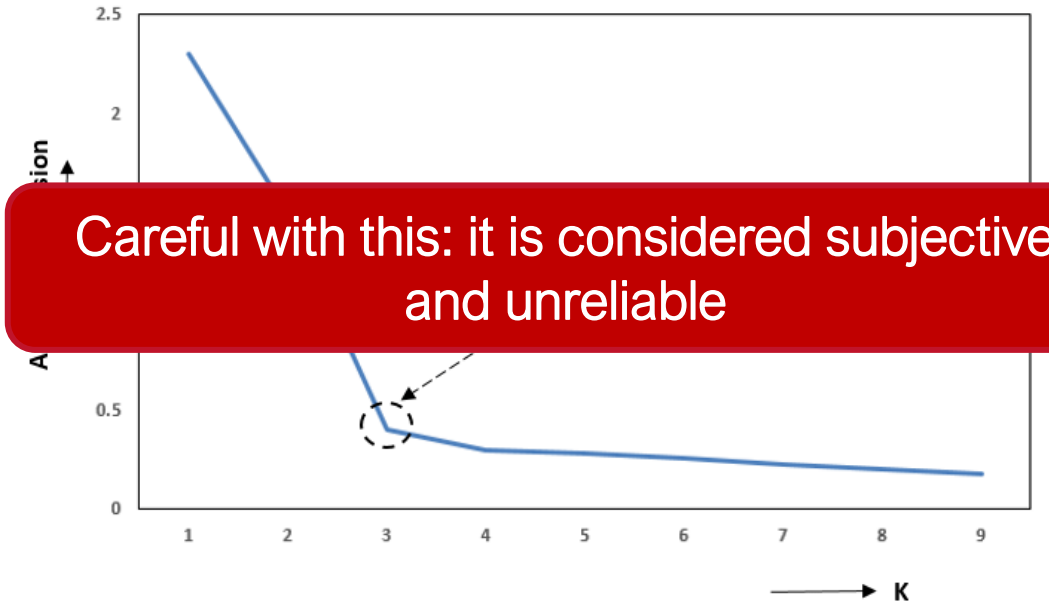


Balancing model complexity and fit

Elbow Method

- K needs to be specified beforehand...
- The **Elbow Method** helps in selecting the optimal number of clusters K:
  1. Perform K-Means with various K (e.g., from 1 to 10).
  2. For each K, calculate the within-cluster sum of squares (WCSS), which represents the total variance within each cluster.
  3. Plot the number of clusters K on the x-axis and WCSS on the y-axis.
  4. Look for a "bend" or "elbow" in the curve where the rate of decrease in WCSS slows significantly.

How to select this value?



Careful with this: it is considered subjective and unreliable

Balancing model complexity and fit

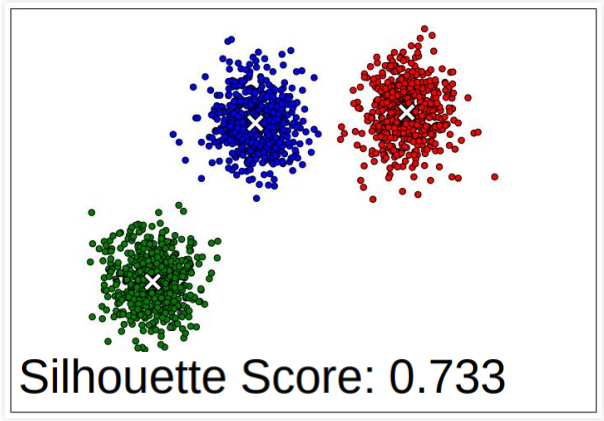
Clustering Validation

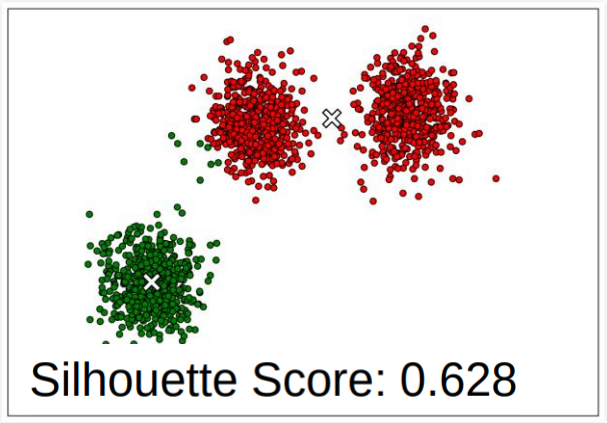
- How to evaluate how good the clustering method performed?
  - Silhouette Score
  - Davies-Bouldin Index (DBI)
  - Within-Cluster Sum of Squares (WCSS)

Clustering Validation

- Silhouette Score: $S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$

  - a(i): Average distance between point i and all other points in the same cluster.
  - $b(i)$: Average distance between point i and points in the nearest neighboring cluster.
  - Range: -1 (poor clustering) to 1 (good clustering).





Clustering Validation

- **Davies-Bouldin Index:**
  - σ_i: Average distance of points in cluster i to centroid c_i
  - $d(ci,cj)$: Distance between centroids c_i and c_j.
  - Lower values indicate better clustering.

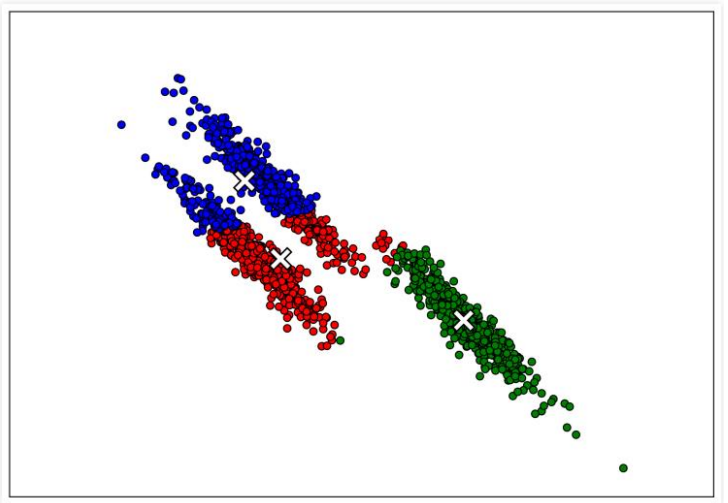
$$DBI = \frac{1}{K}\sum_{i=1}^{K}\max_{i\neq j}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

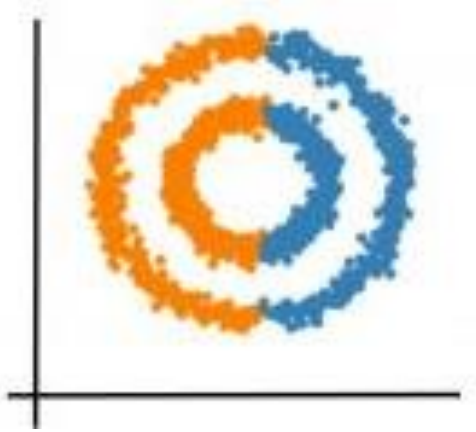
- **Within-Cluster Sum of Squares:**
  - Measures how compact the clusters are.
  - Lower WCSS means tighter clusters.

$$WCSS = \sum_{k=1}^{K}\sum_{i\in C_k}\|x_i - c_k\|^2$$

K-Means Deficiencies

- Fixed number of clusters
- Sensitivity to initialization and outliers.
- Assumes spherical clusters

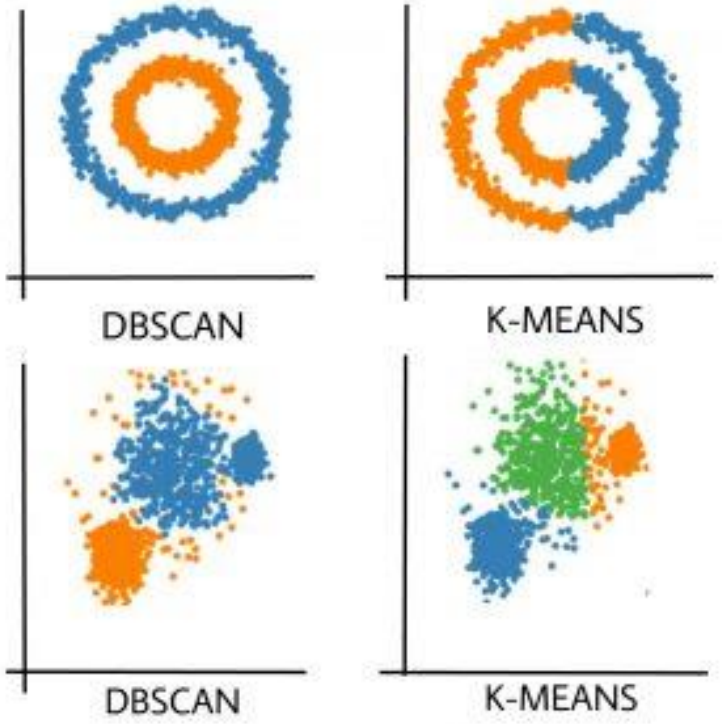




Other Clustering Methods

- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise)
  - Groups together points that are closely packed together.
  - Marks points in low-density regions as outliers (noise).
- Interactive example here

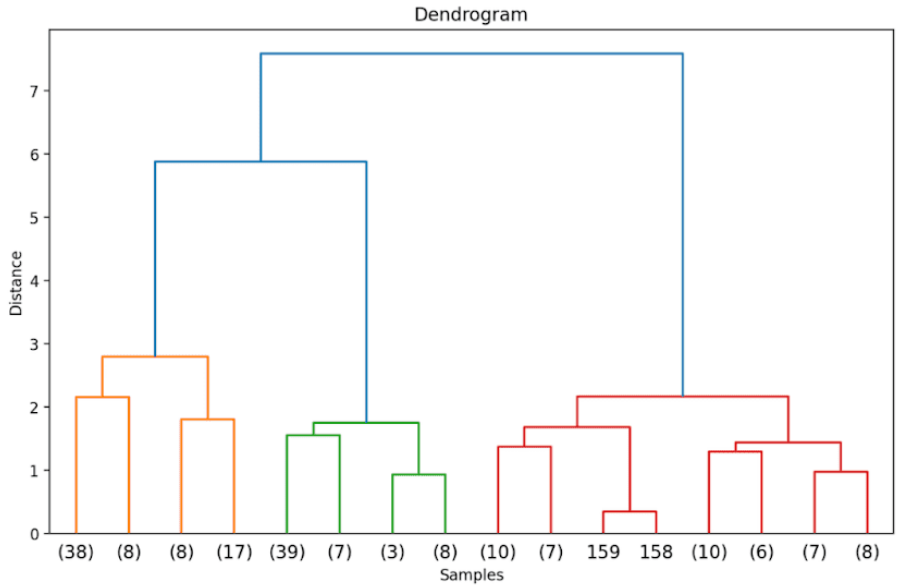
More on this in future modules!

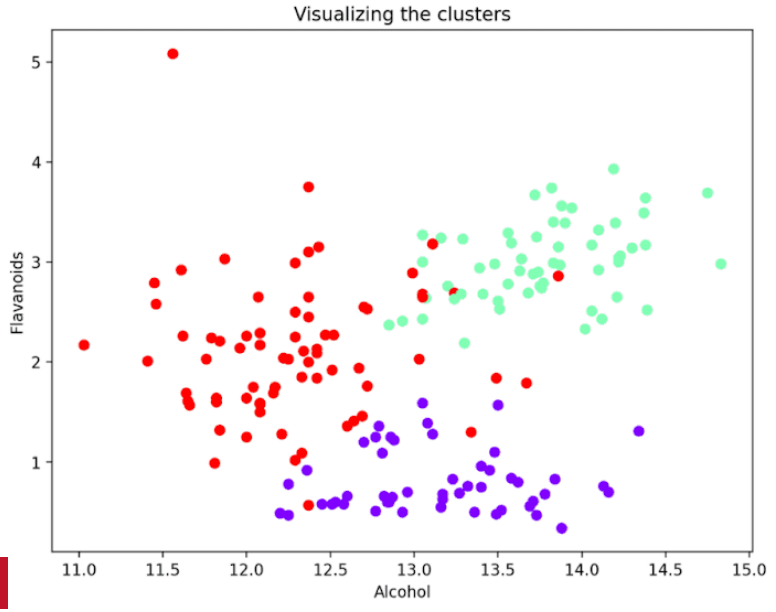


Other Clustering Methods

- Hierarchical Clustering:
  - **Agglomerative (Bottom-Up):**
    - Starts with **each point as its own cluster** and merges clusters step by step until all points belong to a single cluster.
  - **Divisive (Top-Down):**
    - Starts with **all points in one cluster** and splits them step by step into smaller clusters.

More on this in future modules!





https://www.kdnuggets.com/unveiling-hidden-patterns-an-introduction-to-hierarchical-clustering

Conclusion

- **Unsupervised Learning:**
  - Finds hidden patterns in data without labels.
  - Useful for segmentation, anomaly detection, and exploration.
- **Dimensionality Reduction:**
  - PCA which reduces data complexity while preserving important information.
  - Helps visualize high-dimensional data.
- **Clustering:**
  - K-Means groups data into K clusters based on similarity.
  - Simple and fast but limited to spherical clusters.
  - DBSCAN or Hierarchical clustering are other options.

Follow Along!

Course Shared Folder - Google Drive
session_6a_datascience_cmu.ipynb

Use-Case Details

Requirements - One operation of each of the following:

- Dataset Descriptive Statistics
- Data Cleaning (e.g. checking for NaNs, column removal, etc.)
- Model Selection, Feature Engineering, and Normalization
- Plotting (frequency, correlation between feature pairs)
- Supervised Learning:
  - Training a linear classifier
  - Evaluate its performance over multiple metrics

Use-Case Discussion Session

Let's simulate a Data Science team discussion:

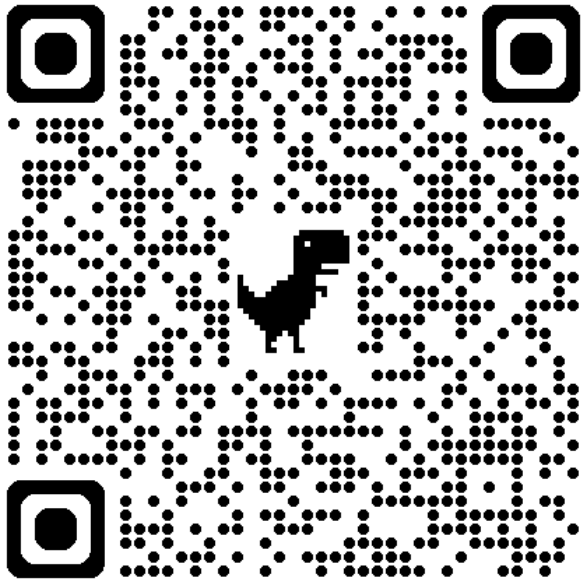
- Bring your expertise and point of view!

We Invite all groups to present and discuss their use-case with the class:

- Show and discuss your notebook to the class
- 5 to 7 minutes per presentation

Use Case – Form Filling

- Fill in this form with your group's information.
- Only one person from the team needs to fill in the information.
- Presentations on the 18th of October (next Friday).



CMU Portugal Academy

Carnegie Mellon Portugal · TÉCNICO LISBOA · Ciências ULisboa · NOVA SCHOOL OF SCIENCE & TECHNOLOGY