Machine learning basics: **Introduction to ML**

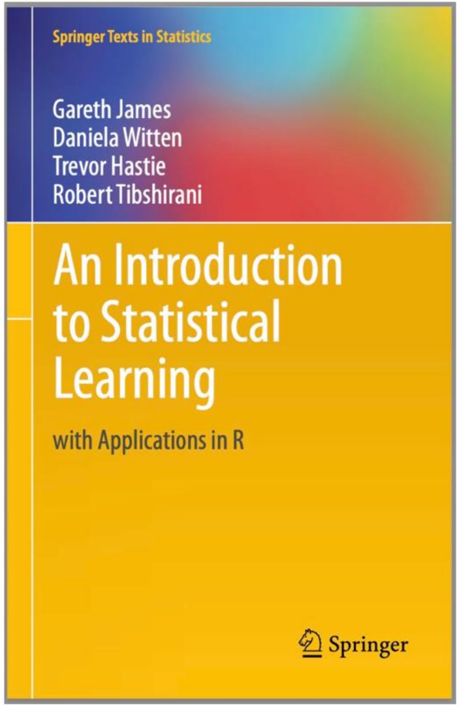
Raquel Pezoa Rivera
raquel.pezoa@usm.cl
Departamento de Informática
Centro Científico Tecnológico de Valparaíso
Universidad Técnica Federico Santa María

Outline

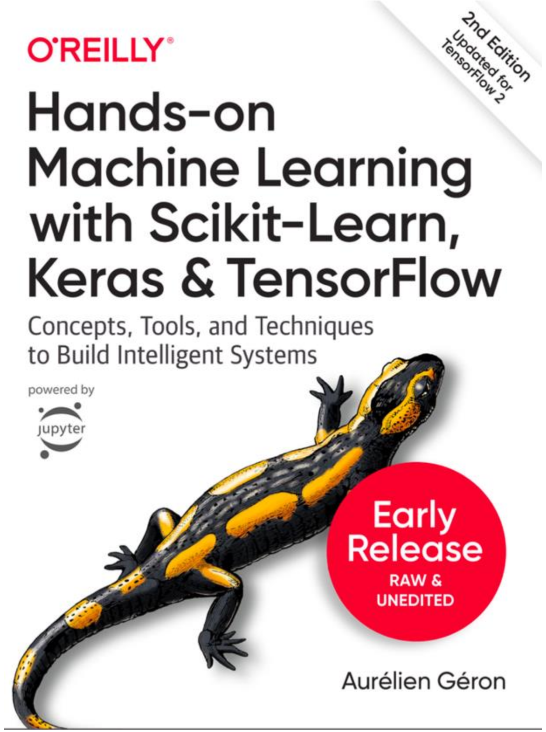
- Definitions of ML
- Basic concepts
- Supervised and Unsupervised Methods

ML Definitions

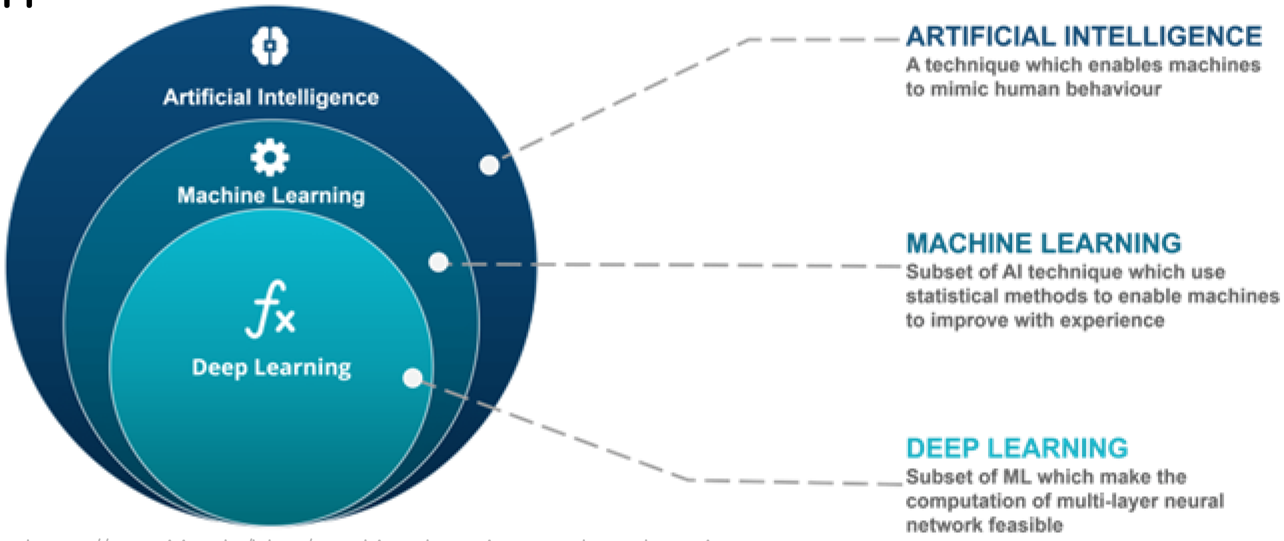
Some interesting books



https://www.statlearning.com/



Introduction



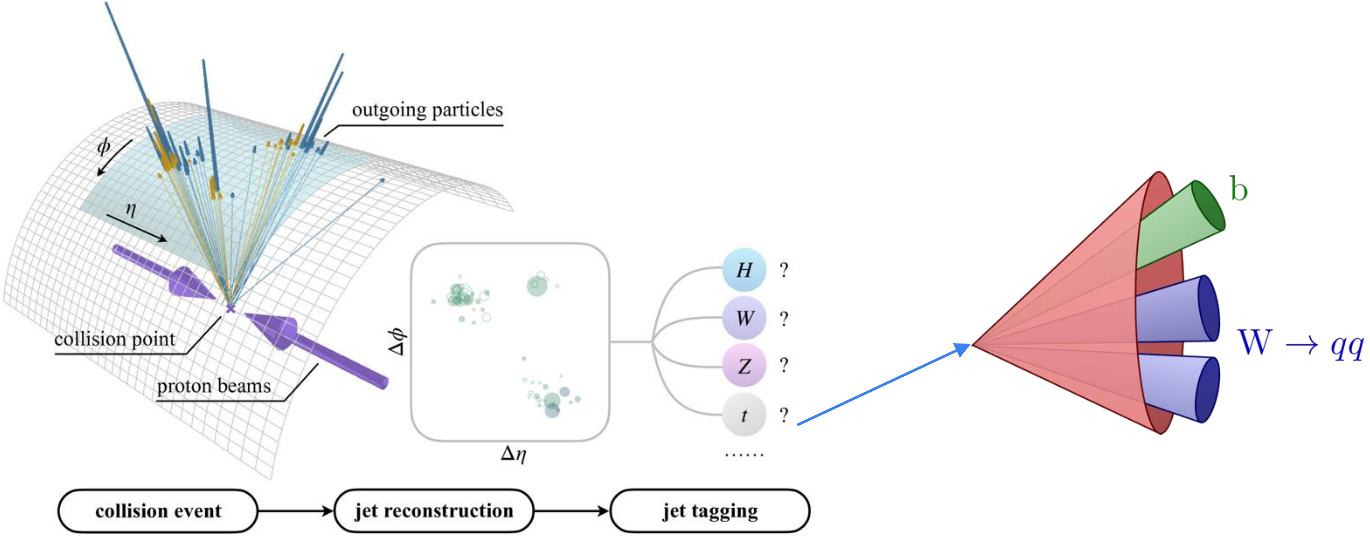
https://cognitive.la/blog/machine-learning-vs-deep-learning

Machine learning (ML) → development of computational algorithms that learn and improve automatically through experience.

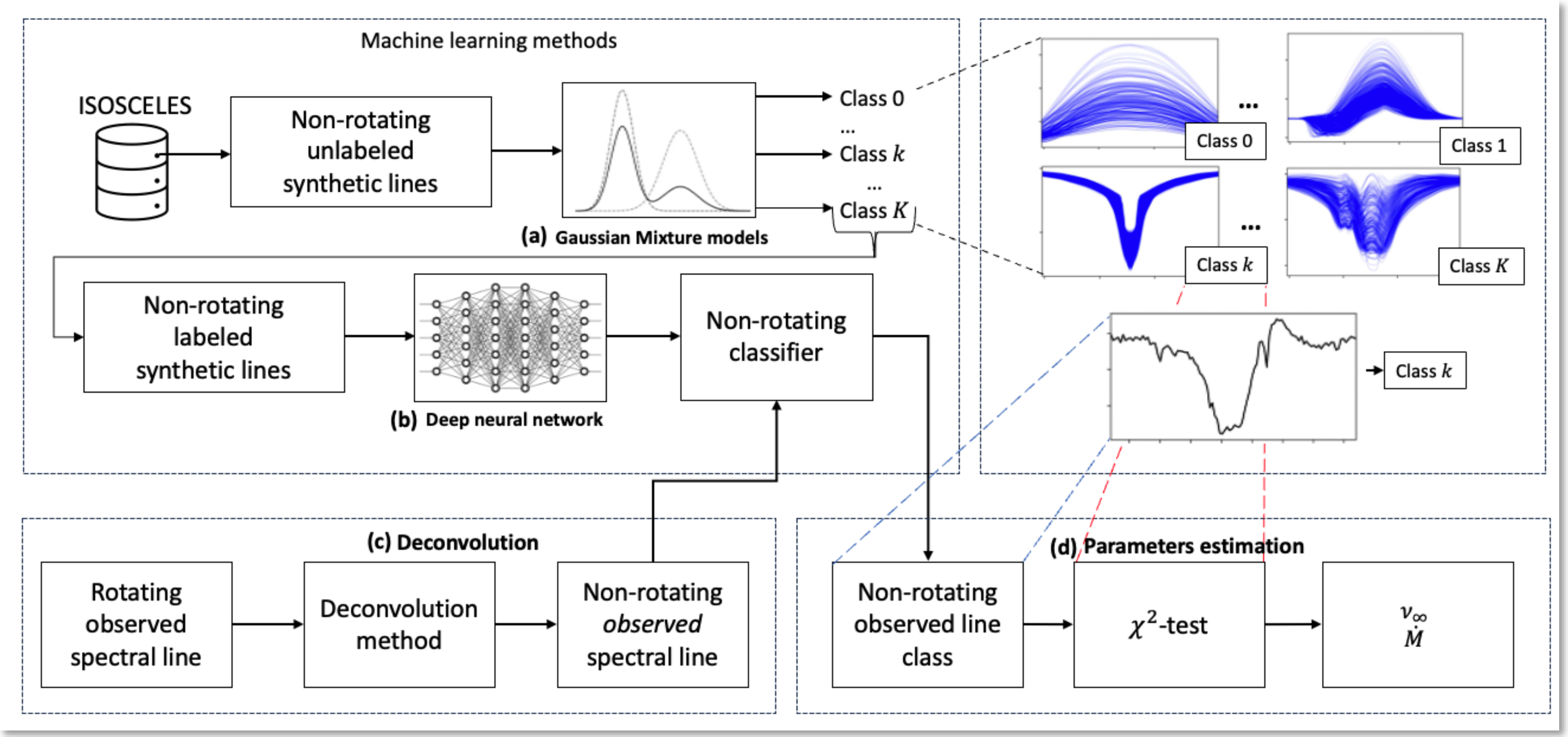
Introduction

- Machine learning is widely used in diverse scientific fields

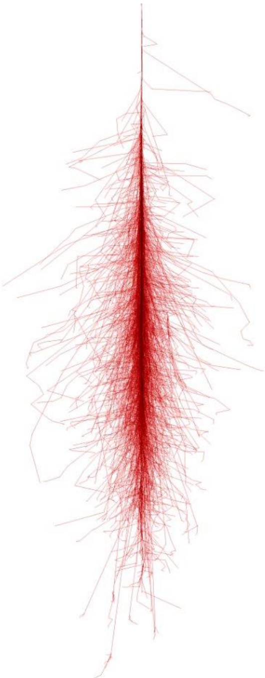
Particle physics: jet tagging



Astrophysics

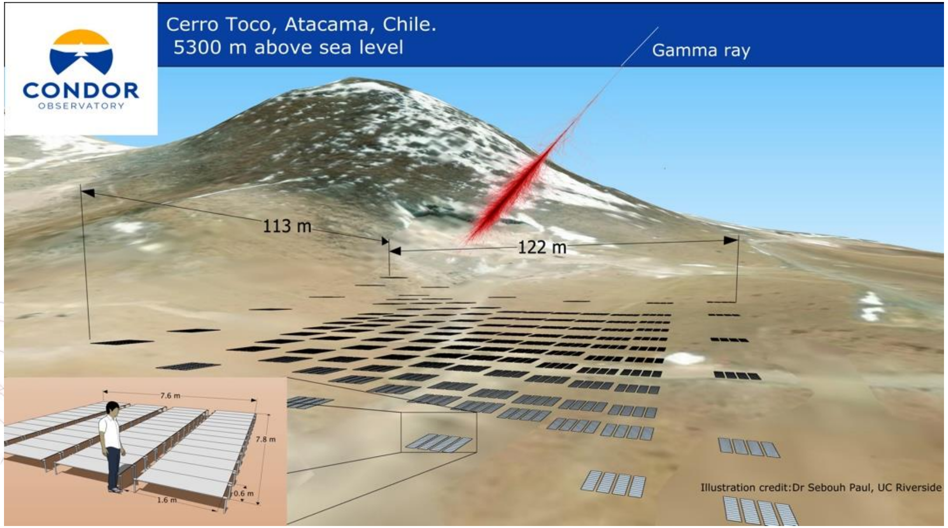


Astroparticle physics



Gamma Ray
(50 GeV Photon)

Cosmic Ray
(100 GeV Proton)

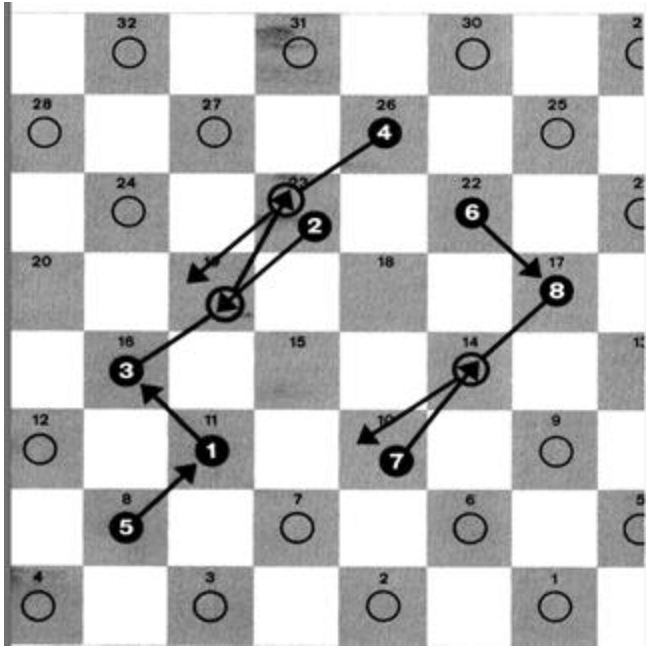


https://condorobservatory.ucr.edu

Some definitions

- Arthur Samuel (1959): ML is the field that *"gives computers the ability to learn without being explicitly programmed"* [1] → Arthur Samuel coined the term "machine learning".

In ML we don't manually write rules or instructions for the system to follow step by step.



[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," in *IBM Journal of Research and Development*, vol. 44, no. 1.2, pp. 206-226, Jan. 2000, doi: 10.1147/rd.441.0206.https://ieeexplore.ieee.org/document/5391906 .

Some definitions

- Tom Mitchell (1998): A computer program is said to learn from experience **E** with respect to some class of tasks **T** and peformance measure **P**, if its performance at tasks in T, as measured by P, improves with experience E [2].

[2] Machine Learning, Tom Mitchell, McGraw Hill, 1997. http://www.cs.cmu.edu/~tom/mlbook.html

Task T

- The task is described in terms of how an ML-based system processes the data

- Some tasks that can be addressed using ML:
    - Classification
    - Regression
    - Anomaly detection
    - Data imputation
    - Clustering
    - Dimensionality reduction
    - Recommendation systems
    - Time series forecasting
    - Generative tasks (e.g., image generation, text generation)

Experience E

- The experience is related to the training process → dataset.

Iris Data Set

Download: Data Folder, Data Set Description

Abstract: Famous database; from Fisher, 1936

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	4142715

Source:

Creator:

R.A. Fisher

Donor:

Michael Marshall (MARSHALL%PLU '@' io.arc.nasa.gov)

Repository: https://archive.ics.uci.edu/ml/datasets/iris

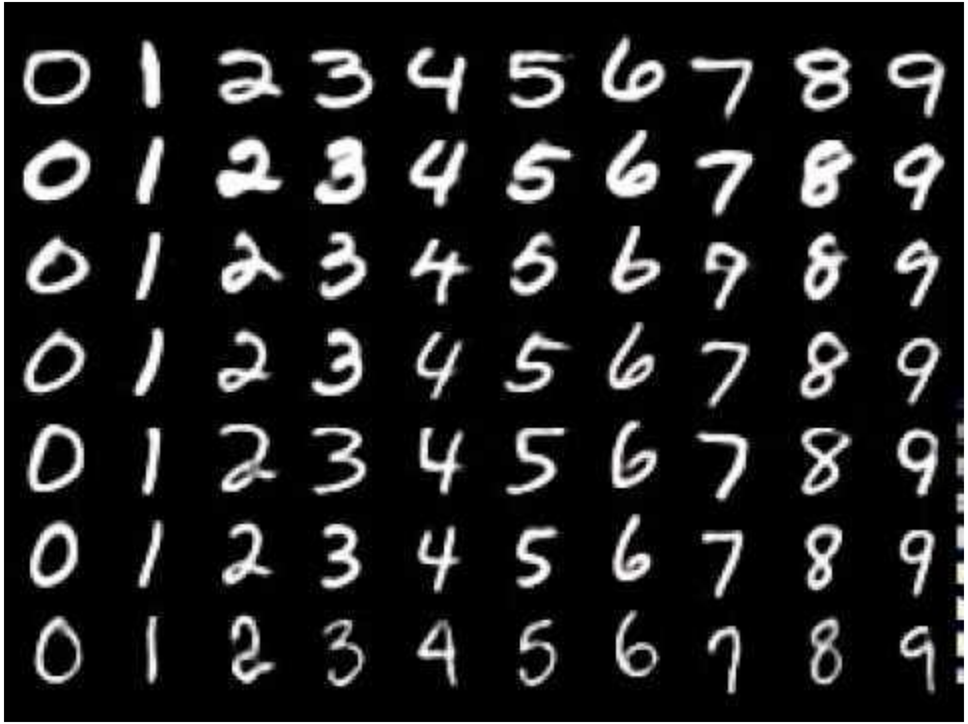
Using Python: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html

Experience E

MNIST

Modified **N**ational **I**nstitute of **S**tandards and **T**echnology database

http://yann.lecun.com/exdb/mnist/



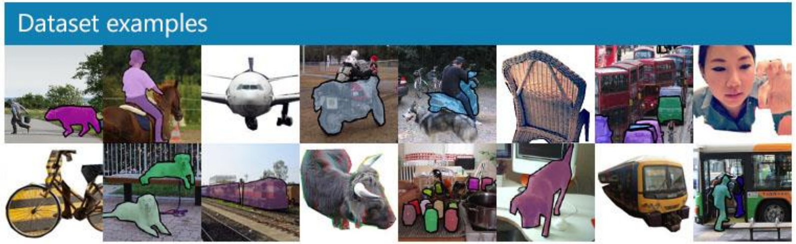
- 60,000 training images and 10,000 testing images labeled with correct answer
- 28 pixel x 28 pixels

Experience E

https://www.kaggle.com/datasets

- 14 million images

https://www.image-net.org/



https://cocodataset.org/

Around **330,000 images**, each annotated with 80 object categories and 5 captions describing the scene.

https://opendata.cern.ch/

Some defintions

- “*Machine Learning is the science (and art) of programming computers so they can learn from data*”. [3]

- “*Machine learning (ML) is a sub-branch of AI that focuses on teaching computers how to learn without the need to be programmed for specific tasks* ”.[4]

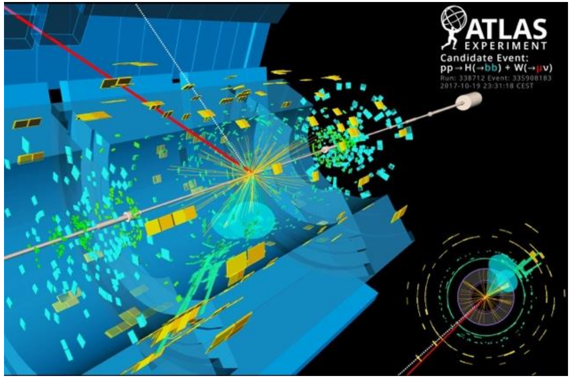
[3] Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. Aurélien Géron. 2019.

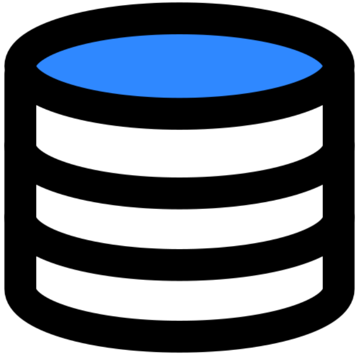
[4] Deep Learning with Keras. Antonio Gulli, Sujit Pal. 2017.

Basic concepts

Main Idea

We have a problem/task





We have data

We need to find the relationships of input and output variables $f(x) = y$ (with good performance!)

Main Idea

- **To predict** an output value from input data
- And we have data:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

Feature, input variable

Label, output variables, target

- A (true) function $Y = f(X) + \epsilon$ and an estimate $\hat{Y} \approx \hat{f}(X)$
- We use a **loss function** to measure the goodness of the approximation → **optimization problem**

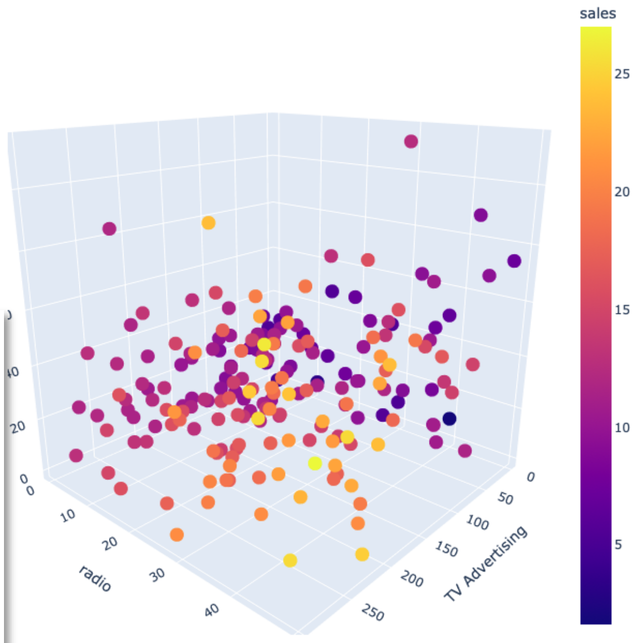
Learning from data

- **Motivation**: Let's assume we are hired to provide advice on how to improve sales of a particular product.

- We have an `advertising` dataset
    - Typical tabular dataset
    - Columns: TV, radio and newspaper are the (advertisement) features
    - Column "sales" is the output

	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
...	...	...	...	...
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

It indicates the number of units sold (thousands of units)



Plots here: https://github.com/rpezoa/ML-HEP-School/blob/main/notebooks/Advertisement_Dataset.ipynb

Learning from data

- Goal: To develop an **accurate** model that can be used to **predict** some value.

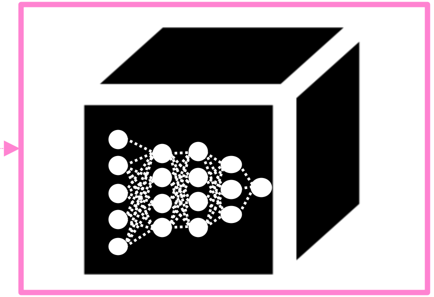
- We have **input variables** or features $(X_1, X_2, \dots, X_p = X)$, and
- **output variable** or label Y. We assume X and Y are related and can be written in the very general form:

$$Y = f(X) + \boxed{\epsilon}$$

It does not depend on features $X_1, X_2, \dots, X_p$

- f is some fixed unknown function of $X_1, X_2, \dots, X_p$, and ϵ is a random **error term** that is independent of X and has a zero mean.

Estimating f



- We can predict using:

$$\hat{Y} = \hat{f}(X),$$

here $\hat{f}$ represents the **estimate of f**, and $\hat{Y}$ is the resulting prediction for Y.

- $\hat{f}$ can be seen as a **black-box** → we care more about accurate predictions for $\hat{Y}$ than the exact form of $\hat{f}$.
- $\hat{Y}$ accuracy depends on:
  - **Reducible error:**
    - $\hat{f}$ is not an exact estimate for f
    - **Can be reduced** using a proper statistical learning technique
  - **Irreducible error**
    - Due to ϵ and its variability
    - Recall that ϵ is independent from X, so no matter how well we estimate f, **we cannot reduce this error**.

Estimating f

- It is very difficult to obtain **the exact relationship** between X and Y.
- ϵ could have unmeasured variables useful to predict Y or may contain unmeasured variation → **no prediction model will be perfect**.
- Let us consider a given estimate $\hat{f}$ and a set of inputs X → $\hat{Y} = \hat{f}(X)$.

$$E\left[(Y-\hat{Y})^2\right] = E\left[\left(f(X)+\epsilon-\hat{f}(X)\right)^2\right] = \underbrace{E\left[\left(f(X)-\hat{f}(X)\right)^2\right]}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}}$$

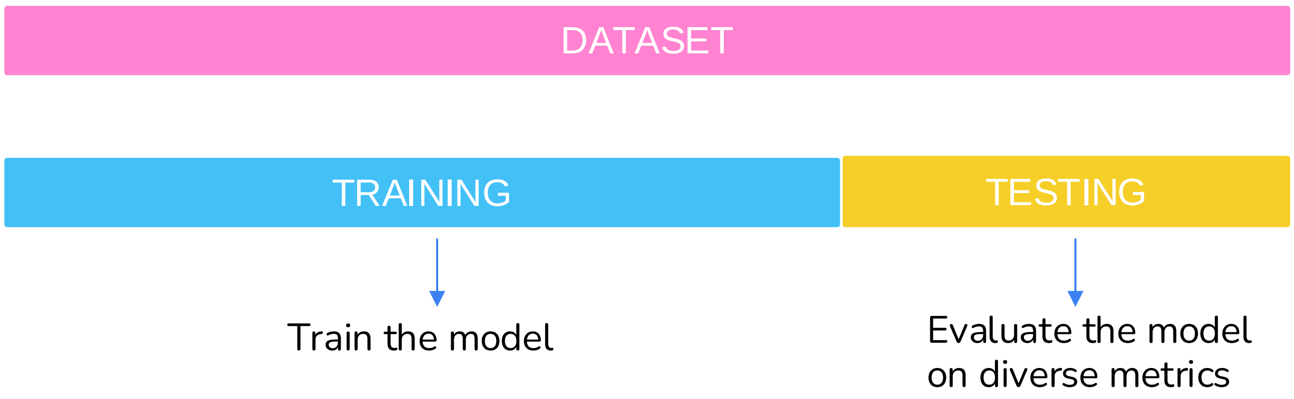
- Here, $E\left[(Y-\hat{Y})^2\right]$ is the average or **expected value**, of the square difference between the predicted and actual value of Y.
- $\mathrm{Var}(\epsilon)$ is the variance associated with the term ϵ.

Estimating f

- First, some notations:
  - n: Number of observations
  - x_{ij}: Value of the jth feature, for ith observation
  - y_i: output (label) of the ith observation
- **Training data**:
  - Set of observations used to estimate f
  - $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, and $x_i = \left(x_{i1}, x_{i1}, \ldots, x_{ip}\right)^T$
- Goal: Find a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y)

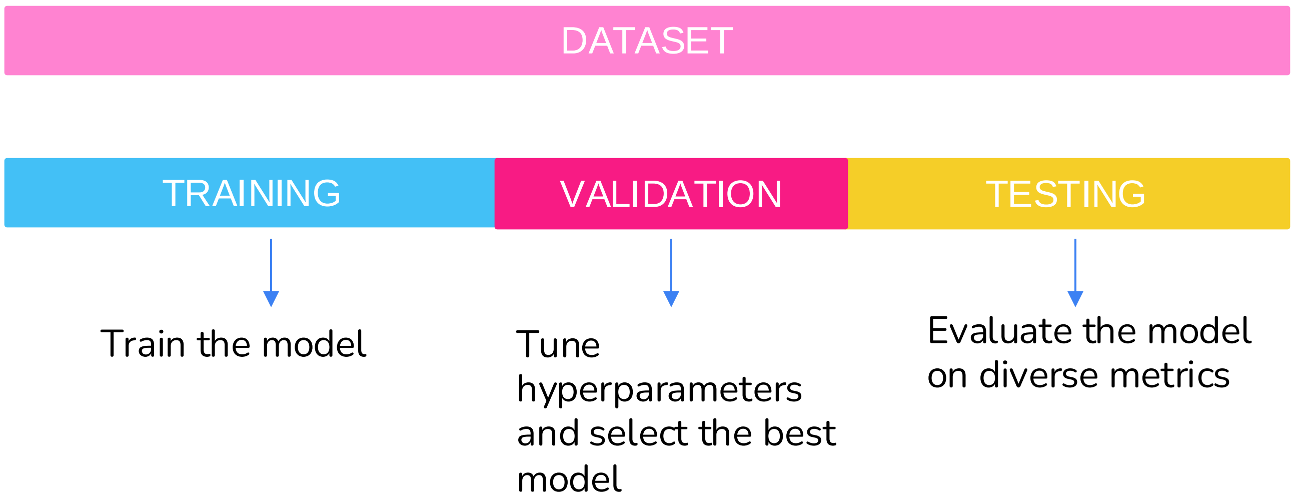
About the training dataset

- More precisely, we use the training (and validation) and testing datasets

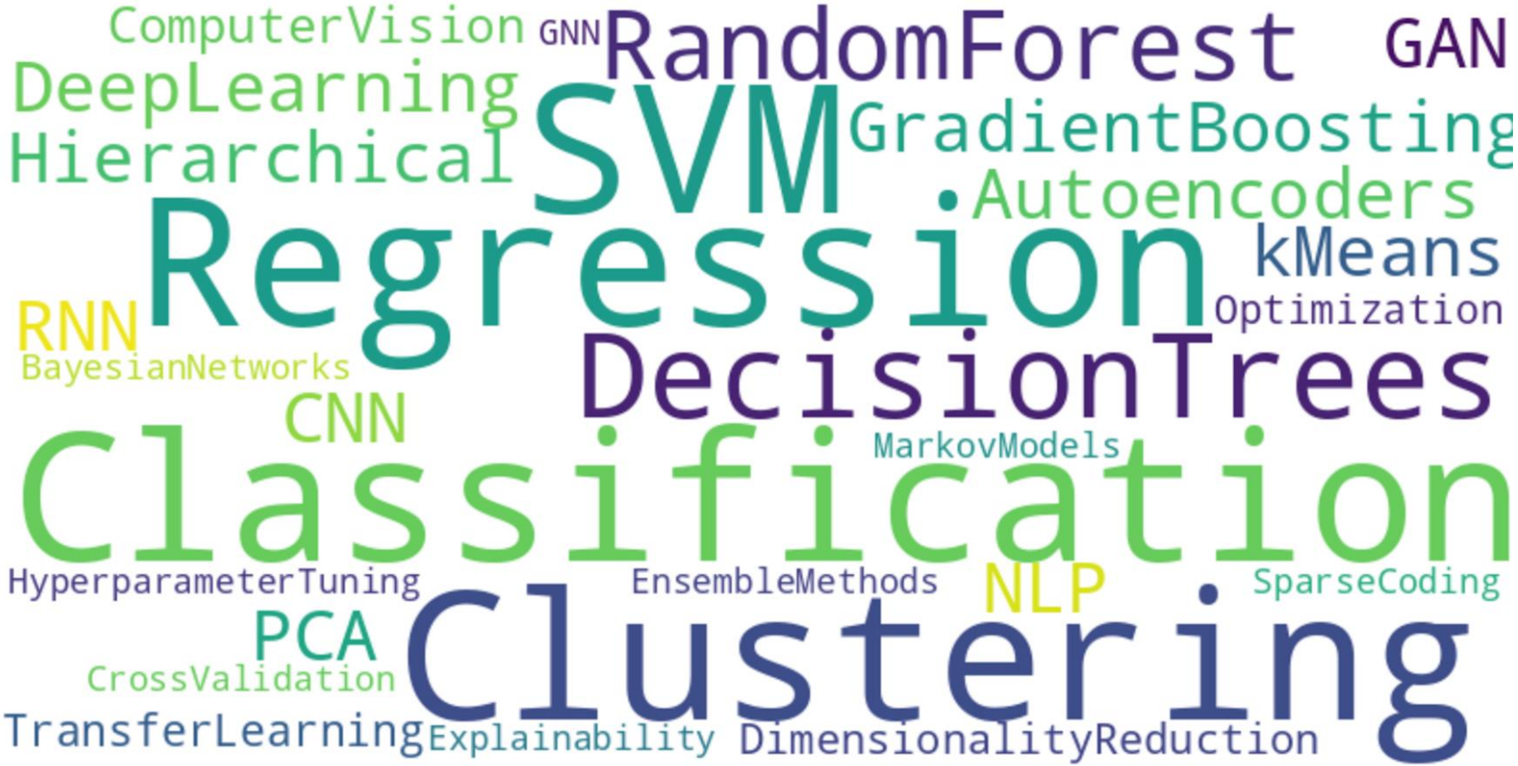


About the training dataset

- More precisely, we use the training (and validation) and testing datasets



Estimating f



ML Concepts

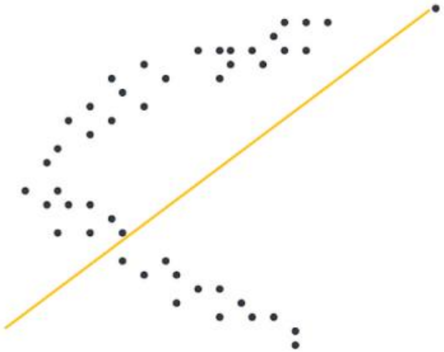
- A main challenge in ML → the algorithm must perform well for new data (data not seen during training) → ***generalization***.

- Generalization refers to the ability of a ML model **to perform well** on unseen or new data **that was not used during training**.
    - It reflects how effectively the model has learned the underlying patterns in the data, **rather than memorizing** the training set.

ML Concepts

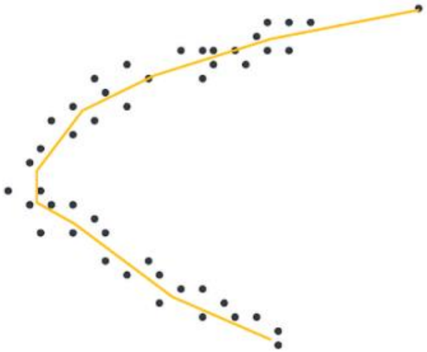
- Good Performance: What is good?
  - <u>Different</u> **peformance metrics** to measure **performance**
- Unseen data:
  - New examples from the same distribution as the training data → testing data
- Good generalization → two concepts:
  - **Overfitting**
  - **Underfitting**
- **Generalization error**: The generalization error is obtained measuring the performance of the model in the testing set.

Overfitting - Underfitting



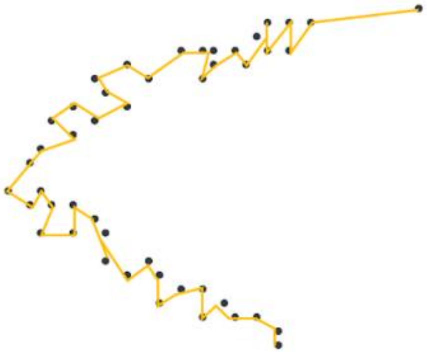
Underfitting

The ML **model is too simple** to capture the underlying relationship in the traning dataset.



Good Fit

Reduce the training error → small generalization error



Overfitting

The model **fits the training data too precisely**, leading to poor performance on new, unseen data

Image source: https://h2o.ai/wiki/overfitting/

Bias-Variance Trade-off

- For the estimate model $\hat{f}$, the goal is **to minimize the expected squared error** between the actual value Y and the predicted value $\hat{Y}$:

$$E\left[Y - \hat{f}(X)^2\right]$$

which can be decomposed as:

$$E\left[Y - \hat{f}(X)^2\right] = (\text{Bias}^2 + \text{Variance}) + \text{Irreducible error}$$

and here,

$$\text{Bias}^2 = \left[f(X) - E\left[\hat{f}(X)\right]\right]^2$$

$$Variance = E\left[\left(\hat{f}(X) - E\left[\hat{f}(X)\right]\right)^2\right]$$

$$\text{Irreducible error} = \text{Var}(\epsilon)$$

Full Error Decomposition Expression

$$E[Y - \hat{f}(X)^2] = \underbrace{\left[f(X) - E[\hat{f}(X)]\right]^2}_{\text{Bias}^2} + \underbrace{E\left[\left(\hat{f}(X) - E[\hat{f}(X)]\right)^2\right]}_{\text{Variance}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible error}}$$

- Therefore, the reducible error is composed of the bias and variance term.

Bias-Variance Trade-off

- **Bias**: $\left[f(X) - E[\hat{f}(X)]\right]$
  - Error due to overlay simplistic assumptions in the model.
  - High bias leads to underfitting
- **Variance**: $E\left[\left(\hat{f}(X) - E[\hat{f}(X)]\right)^2\right]$
  - Error due to model sensitivity to small fluctuations in the training data.
  - High variance leads to overfitting

Goal: Find a balance between variance and bias.

Bias-Variance Tradeoff

- ↑ high bias → ↓ variance (⚠ underfitting)
- ↓ bias → ↑ variance (⚠ overfitting)
- **Model complexity** is a main factor → a model that is based on memorization is not able to predict correctly on unseen data ,

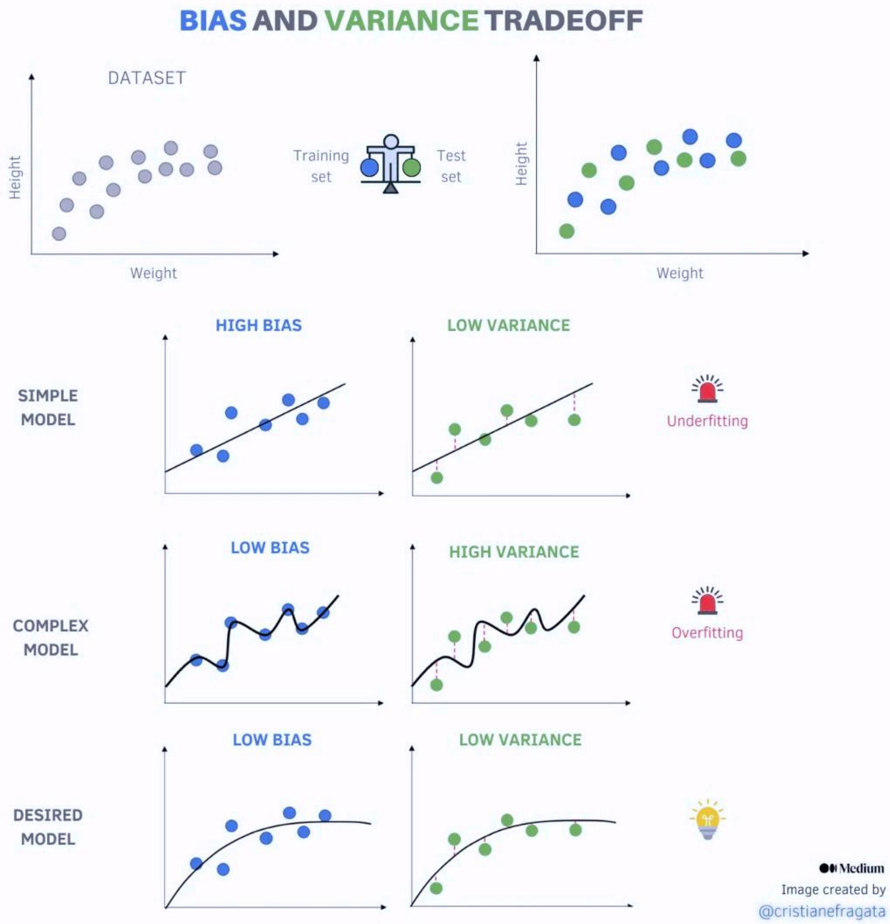


Image source: https://cristianefragata.medium.com/machine-learning-bias-and-variance-26b6ee572af

Bias-Variance Tradeoff

- The model's complexity:
  - ↓ complexity → ↑ bias
  - ↑ complexity→ ↑ variance
- We want to find the

"zone of solutions"

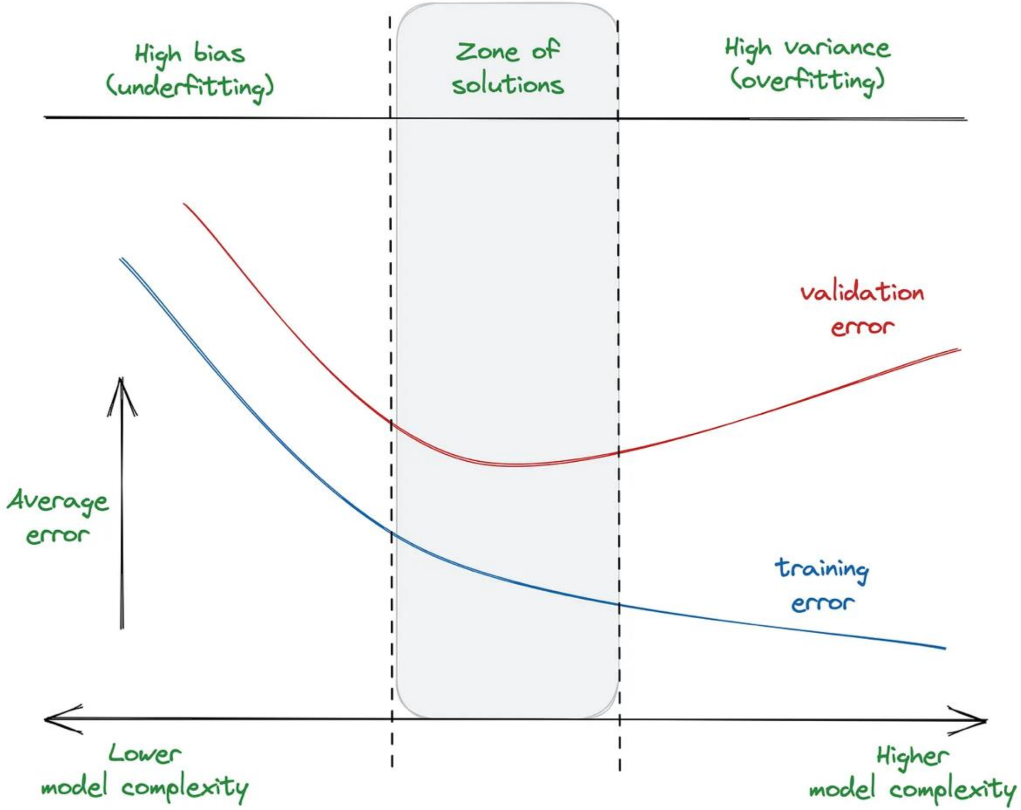
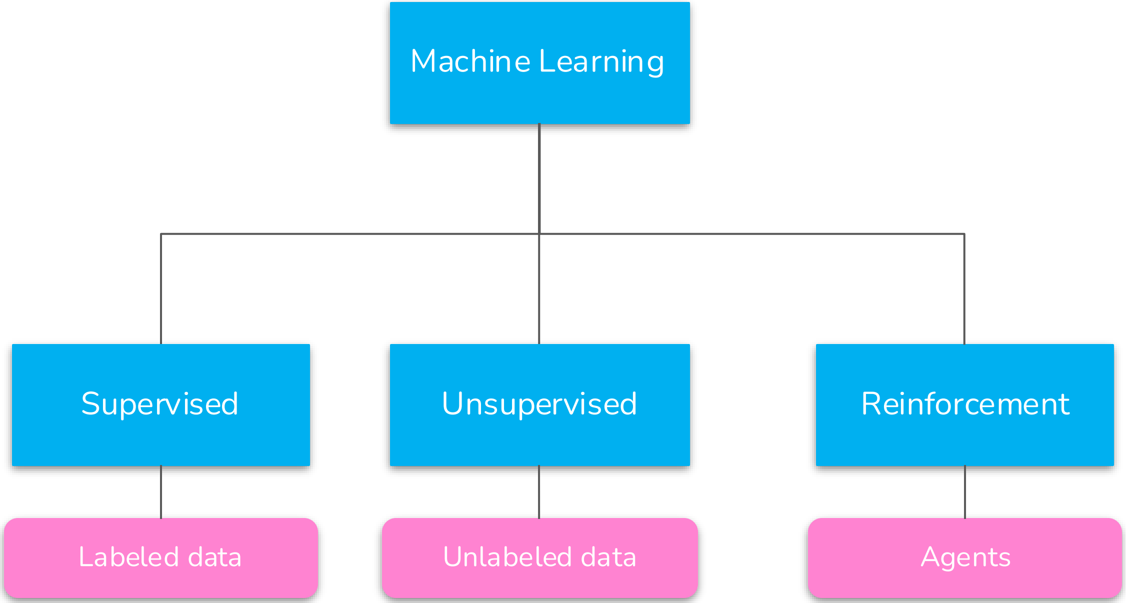


Image source: https://medium.com/@francesco.disalvo/the-bias-variance-tradeoff-an-illustrated-guide-6c79214b0c2b

Supervised and Unsupervised ML Methods

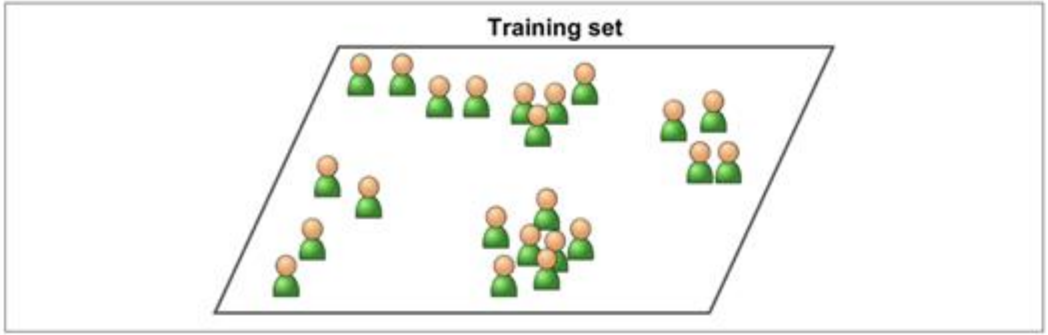
Estimating f

- Three main categories of methods:



ML Methods – Unsupervised

- Main property: unlabeled data



- Clustering: K-Means, DBSCAN, Gaussian Mixture Models
- Anomaly detection: One-class SVM, Isolation Forest, Autoencoders
- Visualization and dimensionality reduction — Principal Component Analysis (PCA), Kernel PCA
- ...

ML Methods – Unsupervised

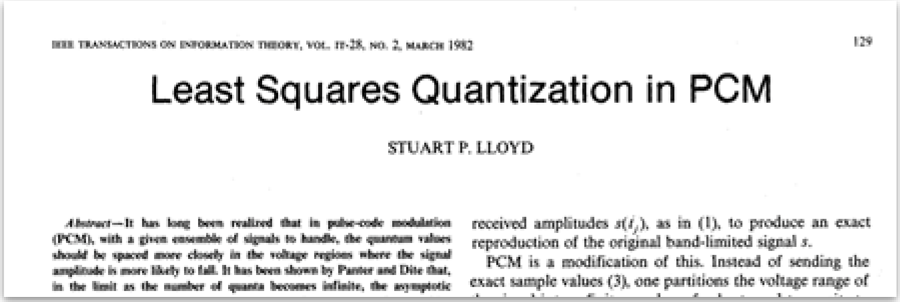
- All of training examples are **unlabeled**, in this type of learning.
- Because unlabeled examples are learned depending on their **similarities**, it is important to define the similarity metric among them.
- The data **clustering** is the typical task to which the unsupervised learning algorithms are applied.

Clustering

- It is the **process of segmenting** a group of sample data into subgroups each of which contains similar ones.
- It is required to define a **similarity metric** between items for executing data clustering.

K-Means

- It was proposed by Stuart Lloyd at the Bell Labs in 1957 as a technique for pulse-code modulation.



IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-28, NO. 2, MARCH 1982 129

Least Squares Quantization in PCM

STUART P. LLOYD

Abstract—It has long been realized that in pulse-code modulation (PCM), with a given ensemble of signals to handle, the quantum values should be spaced more closely in the voltage regions where the signal amplitude is more likely to fall. It has been shown by Panter and Dite that, in the limit as the number of quanta becomes infinite, the asymptotic

received amplitudes $s(t_j)$, as in (1), to produce an exact reproduction of the original band-limited signal s.

PCM is a modification of this. Instead of sending the exact sample values (3), one partitions the voltage range of

https://ieeexplore.ieee.org/document/1056489

- But it was only published outside of the company in 1982, in a paper titled "Least square quantization in PCM".

K-Means

- A simple approach for partitioning a dataset into K **different**, **non-overlapping** clusters.

- First, we must specify the number of clusters K.

- K-means will assign each observation to exactly one of the K clusters.

K-Means

- Let's define some notation. Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster, which satisfy:
  - $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$ → Each observation belongs to at least one of the clusters
  - $C_K \cap C_{K'} = \emptyset$ for all $k \neq k'$ → The clusters are non-overlapping
- Main idea: a good clustering is the one that with **within-cluster variation** is as small as possible :

In other words, we want the data points within each cluster to be very similar to one another.

$$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

$W(C_k)$: within-cluster variation of cluster C_k → An amount indicating how the observations within a cluster differ from each other.

K-Means

- How do we define the within-cluster variation?
  - There are many ways, the most common is the squared Euclidean distance:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{P} \left(x_{ij} - x_{i'j}\right)^2$$

Sum of all the pairwise squared Euclidean distance between the observations in the cluster, divided by the number of observations in the cluster.

$|C_k|$ represents the number of observations in the kth cluster.

- Therefore, the minimization of the within-cluster variation:

$$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{P} \left(x_{ij} - x_{i'j}\right)^2 \right\}$$

K-Means

- How can we solve the minimization problem?

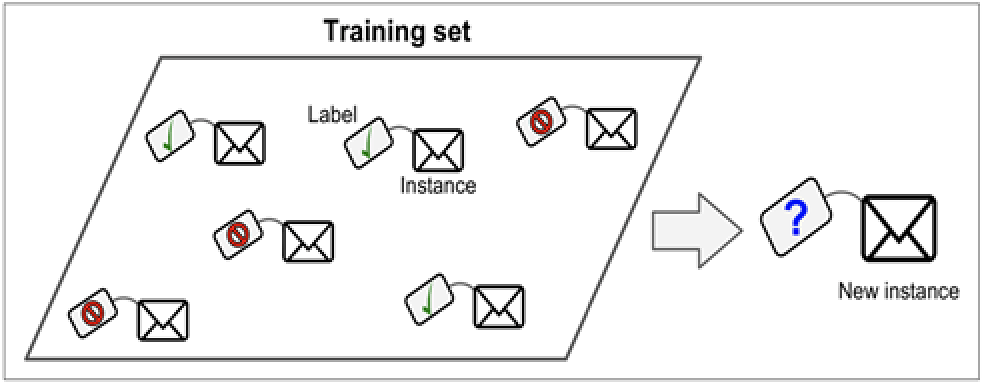
Algorithm K-Means Clustering

1. Initialization: Randomly assign a number, from 1 to K to each observation
2. Iterate until the cluster assignment does not change:
    1. For each cluster, compute the cluster **centroid**.
    2. Assign each observation to the cluster whose centroid is **closest**.

The vector of the p feature means for the observations in the k cluster

Euclidean distance

ML Methods – Supervised



- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees, Random Forests
- Artificial neural networks
- …

Supervised ML Techniques

- The two main tasks in supervised ML are:

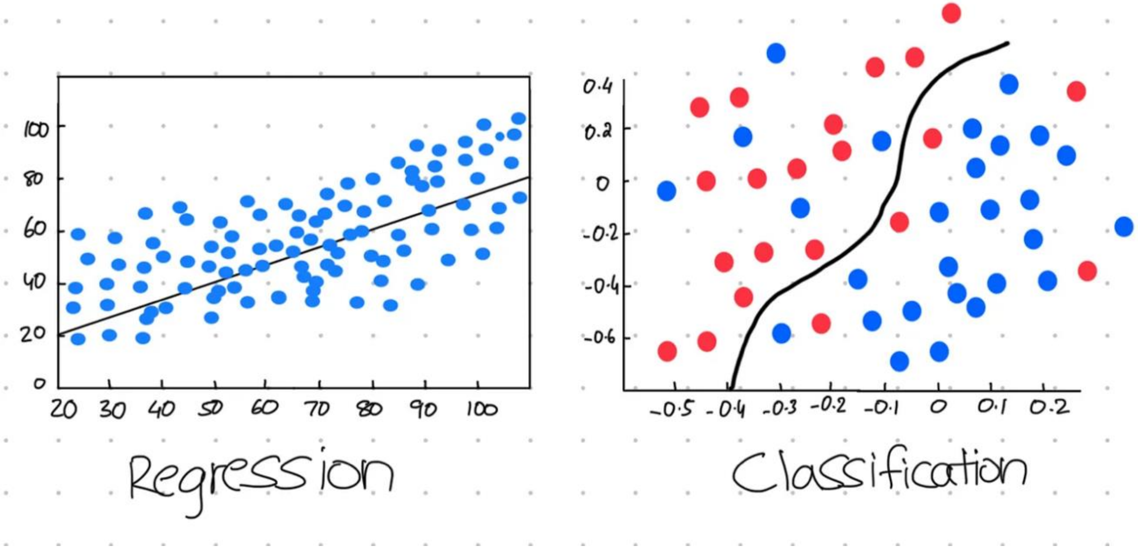


Image source: https://pub.towardsai.net/knns-k-means-the-superior-alternative-to-clustering-classification-310526c73484

Common Regression Techniques

Linear Regression

Polynomial Regression

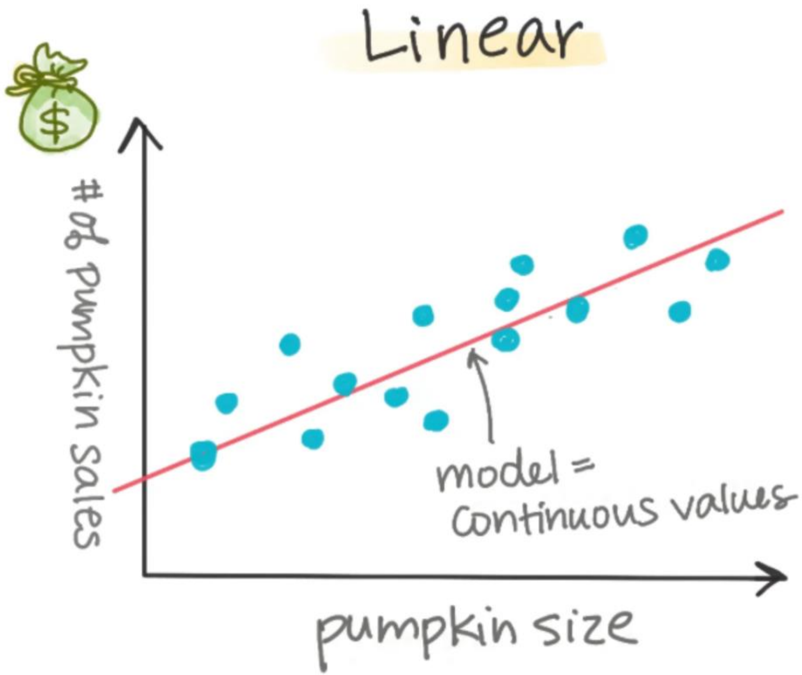
Ridge and Lasso Regression

Decision Tree Regression

Random Forest Regression

Support Vector Regression (SVR)

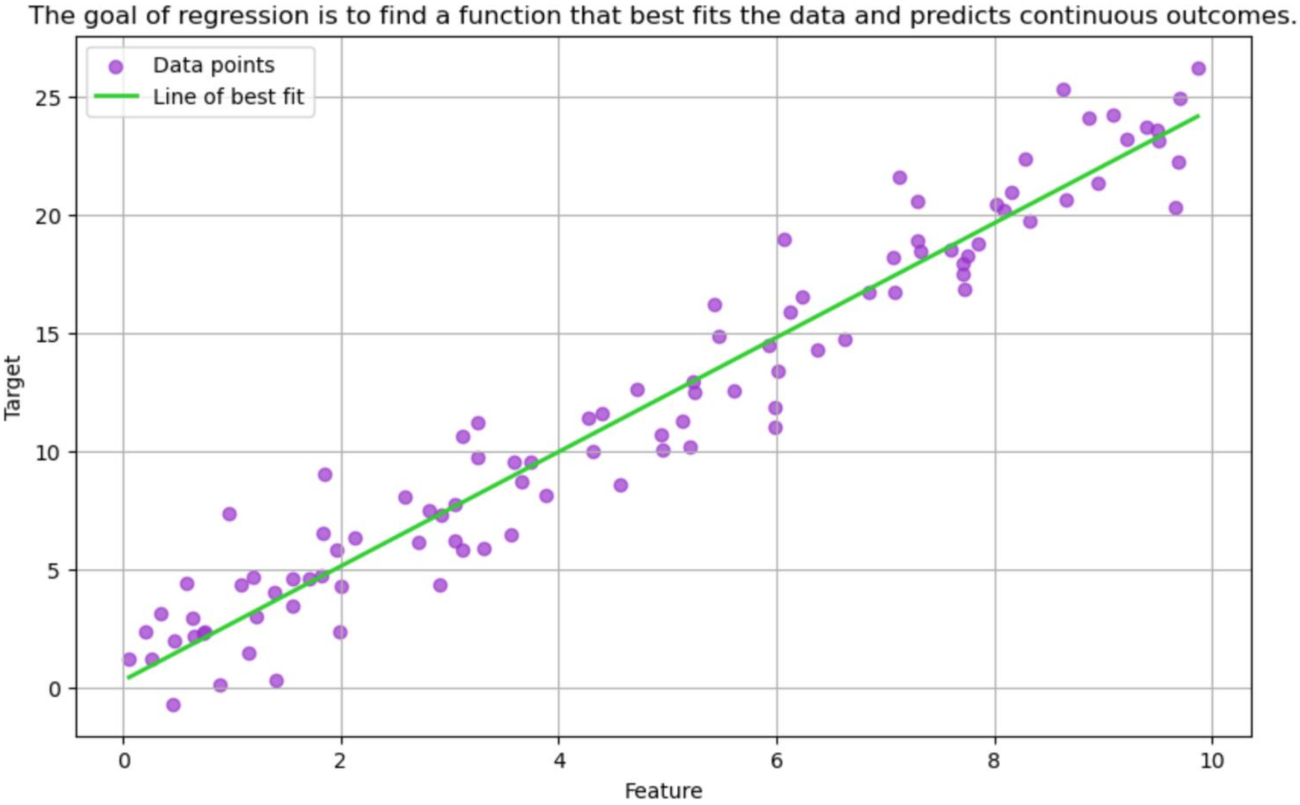
Artificial Neural Networks



Regression

- Regression is a supervised learning task where the goal is **to predict a continuous numerical value** based on input features.
- Main Steps:
  - **Data Collection**: Gather labeled data with input features and corresponding continuous targets.
  - **Training**: Fit the model to minimize the error between predicted and actual values.
  - **Prediction**: Use the model to estimate values for new data.
  - **Evaluation**: Assess how well the model predicts unseen data.

Visualization



Linear Regression

- It finds the best-fitting line (plane/hyperplane) that describes the relationship between the variables.

- How do we train the model? By minimizing the loss function **Mean Squared Error (MSE)**

- The linear regression model predicts $\hat{Y}$:

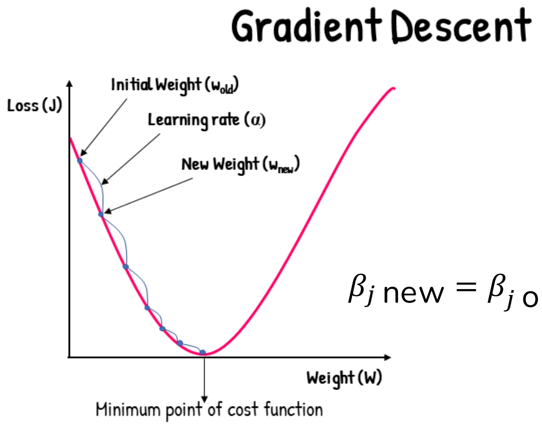
$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

And we must find the values β that mimize the error between the target Y and the predicted value $\hat{Y}$:

$$\min_{\beta_0, \beta_1, \ldots, \beta_p} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

Optimization problem

Gradient Descent



https://editor.analyticsvidhya.com/uploads/28566Slide7.PNG

$$\beta_j \,\text{new} = \beta_j \,\text{old} - \eta \frac{\partial MSE}{\partial \beta_j}$$

It iteratively updates the model parameters β to minimize the loss function.

1. Initialize parameters:

   Start with random or zero values for $\beta_0,\ \beta_1, \beta_2, \dots, \beta_p$.

2. Compute predictions:

   For each data point i compute the predicted value: $\hat{Y} = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}$

3. Compute the gradient:

   Partial derivates of MSE with respect to each β_j:

$$\frac{\partial MSE}{\partial \beta_j} = -\frac{2}{n} \sum_{i=1}^{n} X_{ij}(Y_i - \hat{Y}_i)$$

4. Update the parameters:

$$\beta_j \leftarrow \beta_j - \boxed{\eta} \cdot \frac{\partial MSE}{\partial \beta_j}$$

η is the learning rate, a small positive number controlling the step size.

Evaluation Metrics

1. **Mean Squared Error (MSE)**: – Measures average squared difference between actual and predicted values.

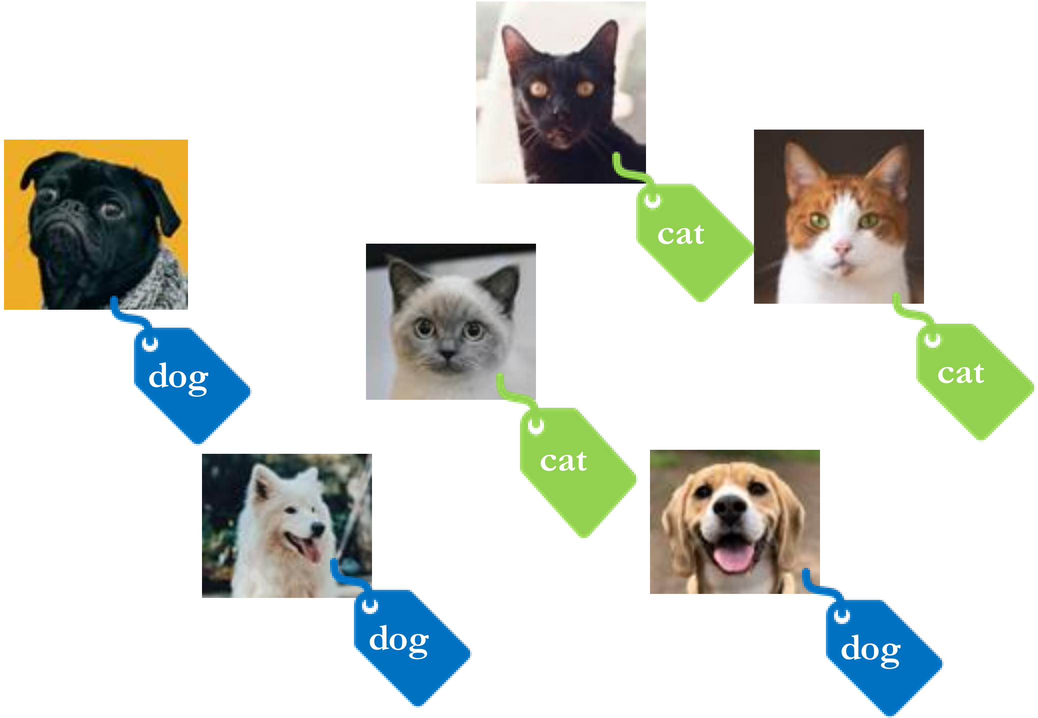
2. **Root Mean Squared Error (RMSE)**: – Square root of MSE for interpretable units.

3. **Mean Absolute Error (MAE)**: – Average of absolute errors, less sensitive to outliers.

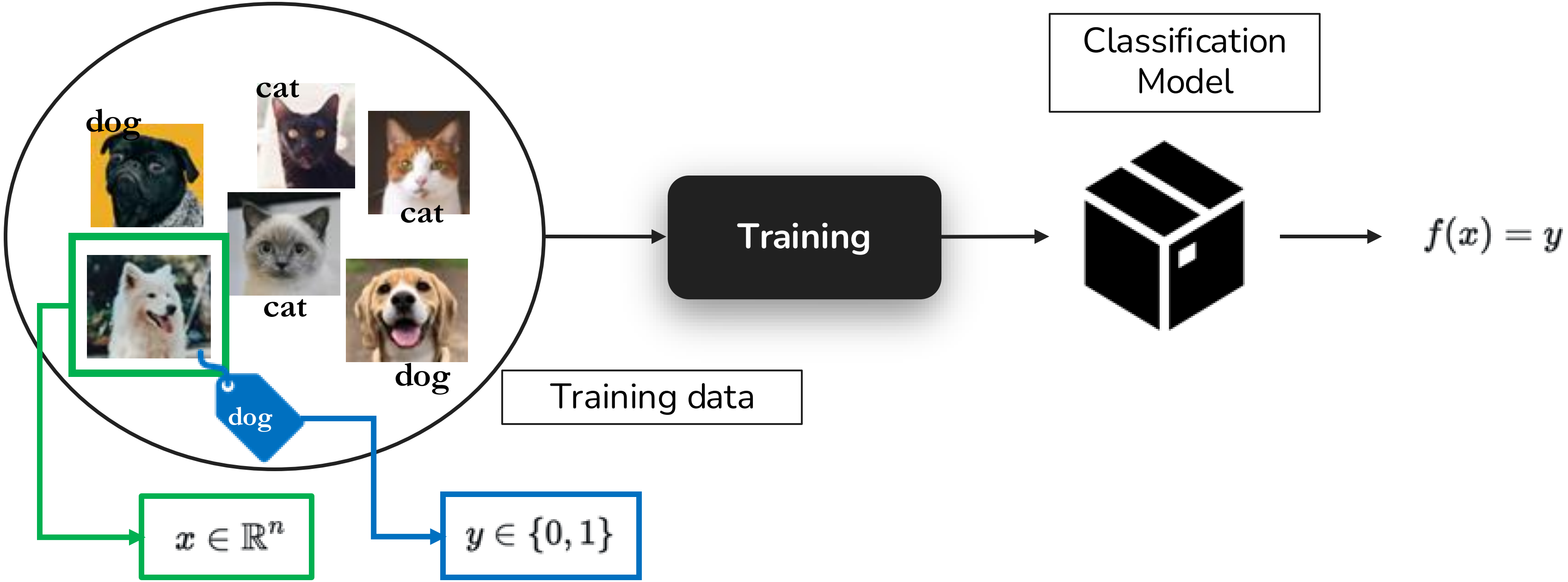
Challenges

- **Underfitting**:The model is too simple and fails to capture patterns in the data.
- **Overfitting**: The model is too complex and captures noise, leading to poor generalization.
- **Multicollinearity**: Strong correlations between features can distort the model.
- **Heteroscedasticity**: Non-constant variance in the errors can affect accuracy.
- **Tip**: Use techniques like regularization, feature selection, and cross-validation to address these issues.

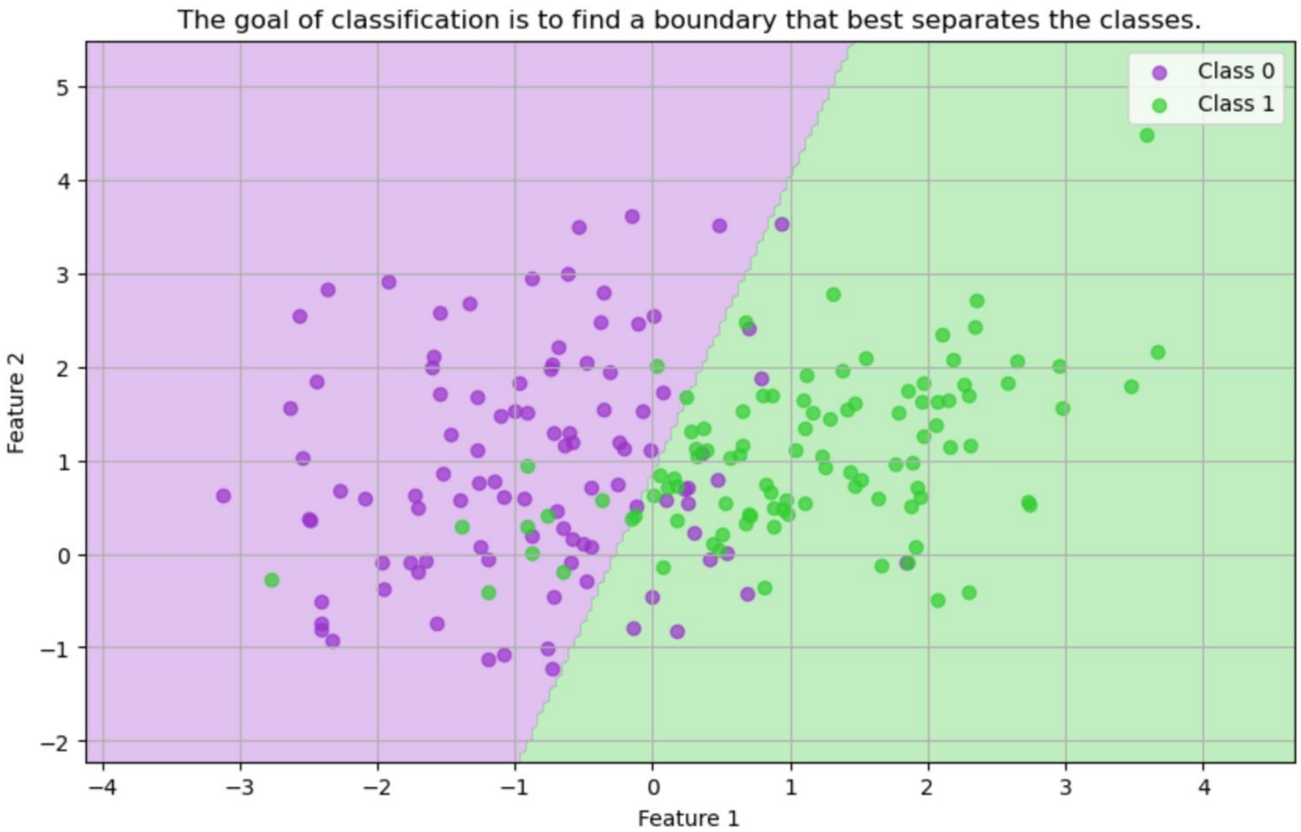
What is Classification?



What is Classification?



Decision Boundaries in Classification



Classification

- The goal is **to predict class labels**, which is a choice from a predefined set of labels or classes.

- There is a huge amount of machine learning methods for approaching the classification task.
- Main steps:
  - **Data Collection**: Gather labeled data points.
  - **Training**: Use the data to teach the model.
  - **Prediction**: Assign labels to new, unseen data points.
  - **Evaluation**: Measure the model's **accuracy** and **reliability**.

Common Classification Algorithms

Logistic Regression

k-Nearest Neighbors (k-NN)

Decision Trees

Support Vector Machines (SVM)

Naive Bayes

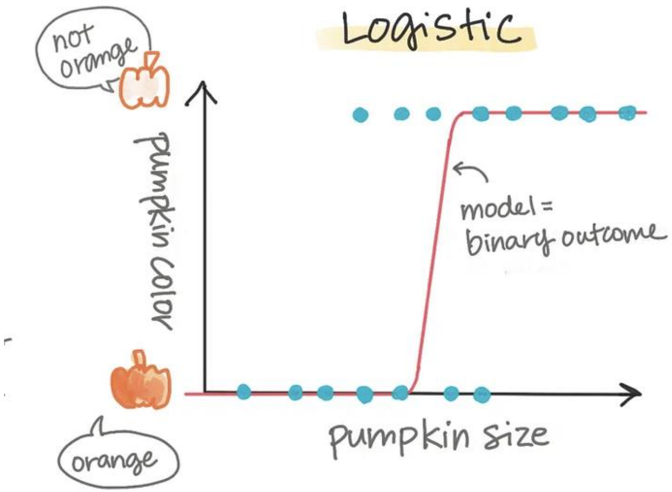


Image source: https://blog.gopenai.com/linear-and-logistic-regression-same-regression-but-different-purpose-f6ff5f93b7ef

Logistic Regression

- It computes a weighted sum of the input features (plus a bias term), but it outputs the logistic of this result.

- It predicts the ***probability*** of a data point belonging to a specific class, usually class 1:

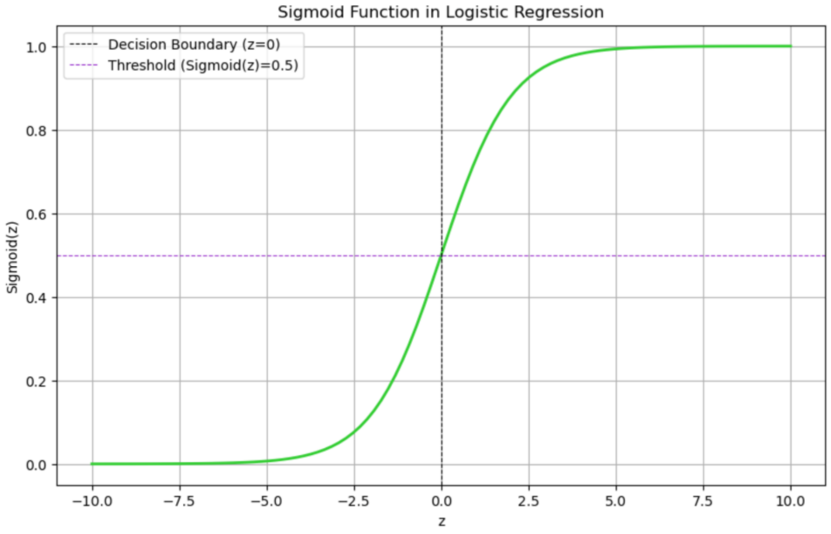
$$P(Y = 1|X) = \frac{1}{1 + e^{-z}}$$

and $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$, β_0 is the intercept and $\beta_1, \dots, \beta_p$ are the coefficients for the features $X_1, \dots, X_p$

Logistic Regression

- $\sigma(z) = \frac{1}{1+e^{z}}$ is the logistic or sigmoid function

- Logistic Regression converts the "probability" into class labels, using a threshold, (usually 0.5).
  - If $P(Y = 1|X) \geq 0.5$ predicts $Y = 1$
  - If $P(Y = 0|X) < 0.5$ predicts $Y = 0$



$$\boxed{\text{As } z \to \infty, \sigma(z) \to 1; \text{as } z \to -\infty, \sigma(z) \to 0}$$

Training process

- To set the parameter vector β so that the model estimates high probabilities for positive instances $(Y = 1)$ and low probabilities for negative instances $(Y = 0)$.
- How do we train the model? Maximizing the likelihood function:

$$L(\beta) = \prod_{i=1}^{n} P(Y_i|X_i)^{Y_i}(1 - P(Y_i|X_i))^{1-Y_i}$$

- But it is better to minimize the negative log-likelihood:

$$NLL = -\frac{1}{n}\sum_{i=1}^{n}[Y_i \log(P(Y_i|X_i)) + (1 - Y_i)\log(1 - P(Y_i|X_i))]$$

Convex function→ so gradient descent (or any other optimization algorithm) is guaranteed to find the global minimum.

Evaluation Metrics in Classification

Correctly classified

Predicted value

Wrong classified

Actual Value		0	1
	0	**TN** True Negatives	**FP** False Positives
	1	**FN** False Negatives	**TP** True Positives

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

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = 2\,\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Values between 0 and 1

Wrong classified

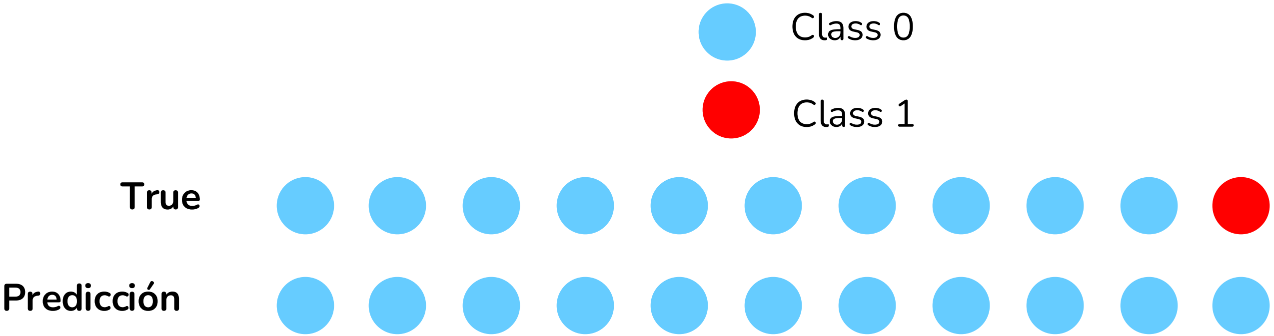
Correctly classified

Class imbalance

Predicted value

Actual Value

TN	FP
FN	TP



9	0
1	0

$$\text{Accuracy} = \frac{\boxed{TP}+\boxed{TN}}{\boxed{TP}+\boxed{TN}+\boxed{FP}+\boxed{FN}}$$

$$\text{Accuracy} = \frac{0+9}{0+9+0+1} = 0.9$$

Accuracy metric is not representative when we have imbalanced classes.

Challenges

- **Overfitting**: Model is too complex and memorizes the training data.
- **Underfitting**: Model is too simple and misses patterns in the data.
- **Class Imbalance**: One class dominates, leading to biased predictions.
- **Tip**: Use techniques like cross-validation, regularization, and resampling to address these challenges.

¡Gracias!

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

CCTVal
CENTRO CIENTÍFICO TECNOLÓGICO DE VALPARAÍSO

Acknowledgements: ANID PIA/APOYO AFB230003

SCAN ME

https://github.com/rpezoa/ML-HEP-School/