Understanding and building a data set

Genoveva Vargas-Solar
French Council of Scientific Research, LIG
genoveva.vargas@imag.fr

http://vargas-solar.com/data-ml-studios/

UNI, BI AND MULTIVARIATE ANALYSIS

Univariate analysis: descriptive statistical analysis differentiated based on the number of variables involved at a given point in time (e.g pie chart)

Bivariate analysis: attempts to understand the difference of two variables at a time (e.g. scatter plot)

Multivariate analysis: study more than two variables to test and **understand the effect** of variables on the responses

SAMPLING

Random sampling

Cluster sampling:

- used when it becomes difficult to study the target population
- Population spread across a wide area
- Simple random sampling cannot be applied

Systematic sampling:

- statistical technique where elements are selected from an **ordered sample frame**
- List progressed in a circular manner
- **Equal probability method** when an individual is selected from an available units of the population at the time of selecting the unit the probability of selection is equal

EIGEN VECTORS AND VALUES

Eigenvector used for understanding linear transformations

- Directions along which a particular linear transformation acts by flipping, compressing or stretching

Eigenvalue how much variance is in the data in that direction

- The scalar applied to the eigenvector

MISSING VALUES

Look for patterns that can give insight

Perform imputation by substituting missing values with **mean** or **median**

Ignore missing values

For a categorical variable → default

If normal distribution → mean

If 80% are missing → drop variable

EXPECTED VALUE , MEAN VALUE

Expected value

Population mean

Mean of all the means

Random variable context

For distributions

- Irrespectively of the distributions iff the distribution is in the same population

Mean value

Probability distribution

Sample population

The only value that comes from the sampling data

POWER ANALYSIS, INTER-EXTRAPOLATION

Experimental design technique for determining the effect of a given sample size

Interpolation: estimating a value from 2 known variables from a list of variables

Extrapolation: approximating a value by extending a known set of values/facts

DATA TRANSFORMATION

Box Cox: statistical technique to transform non normal dependent variables into a normal shape λ varies from [-5, 5]

$$y(\lambda) = \begin{cases} y(\lambda - 1 / \lambda) & \lambda <> 0 \\ \log y & \lambda = 0 \end{cases}$$

- the dependent variable for a regression analysis might not satisfy one or more assumptions of an ordinary **least square regression**
- **Residuals curve or follow skewed distribution**

Vectorization is a powerful method

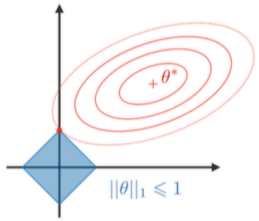
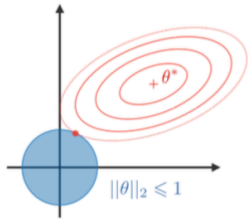
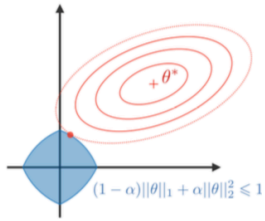
- Remodeling data
- Converting an algorithm from operating on a single value at a time to operation on a set of values at a time

REGULARISATION

The process of adding **tuning parameter** to a model to **induce smoothness** to **prevent overfitting**

Add a **constant multiple** to an existing **weight vector**

The **model predictions** should then **minimize** the **loss function** on the **regularized training set**

LASSO	Ridge	Elastic Net
• Shrinks coefficients to 0 • Good for variable selection	Makes coefficients smaller	Tradeoff between variable selection and small coefficients
		
$\ldots + \lambda \|\|\theta\|\|_1$ $\lambda \in \mathbb{R}$	$\ldots + \lambda \|\|\theta\|\|_2^2$ $\lambda \in \mathbb{R}$	$\ldots + \lambda \left[(1-\alpha)\|\|\theta\|\|_1 + \alpha \|\|\theta\|\|_2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0,1]$

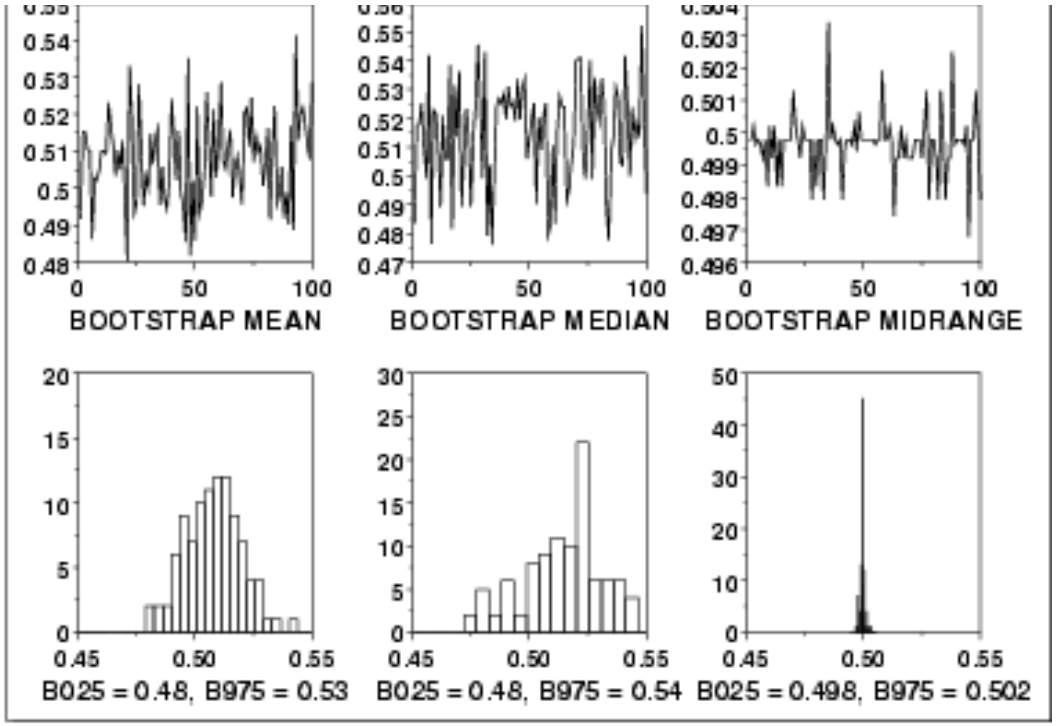
L2 minimizes squared error

L1 penalizes more variables as L2 which results in sparcity

BOOTSTRAP PLOT

Resampling technique used to estimate statistics on a population by sampling a dataset with replacement.

- Allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error) to sample estimates
- used in applied **machine learning** to **estimate the skill** of **machine learning** models when making predictions on data not included in the training data



GRID - SEARCHING

The process of scanning the data to configure optimal parameters for a given model

Calculate the best parameters to use for any given model

T-TEST *VS.* Z-TEST

Require data with normal distribution

t-test statistical method used to see if two sets of data are significantly different

$t = X - \mu / SE$ known population parameters

Z-test statistical method to determine the probability that a new data will be near the point for which a score was calculated

- Variance known, large sample
- Used to calculate populations means to a sample's
- How far in standard deviations a data point is from the mean of a data set's
- **Useful for hypothesis testing**

Standard error $SE = \sigma / (n)^{1/2}$ σ population standard deviation

$z = M - \mu / SE$ μ Sample mean, M population mean

COLLINEARITY *VS.* CORRELATION

Collinearity refers to two or more independent variables acting in concert to explain the variation in a dependent variable

Correlation describes the relationship between two variables i.e. value of one increases/decrease with increase/decrease of another

Correlation means - two variables vary together, if one changes so does the other but it does not imply collinearity or that one can explain the other

FEATURES ENGINEERING (TIME SERIES)

Aggregation by time

- Avg amount : per week
- Per day during a period of time
- Avg amount (merchant) category in some time window

By category

- Two close transactions in distant locations
- Time period within a time window

Location and time

- Two close transactions in distant locations
- Locations per day within a time window
- Operation method per day within a time window

Extracting models and forecasting events

Supervised / Unsupervised learning

Genoveva Vargas-Solar
French Council of Scientific Research, LIG
genoveva.vargas@imag.fr

http://vargas-solar.com/data-centric-smart-everything/

GENERAL MACHINE LEARNING PIPELINE

Define the problem

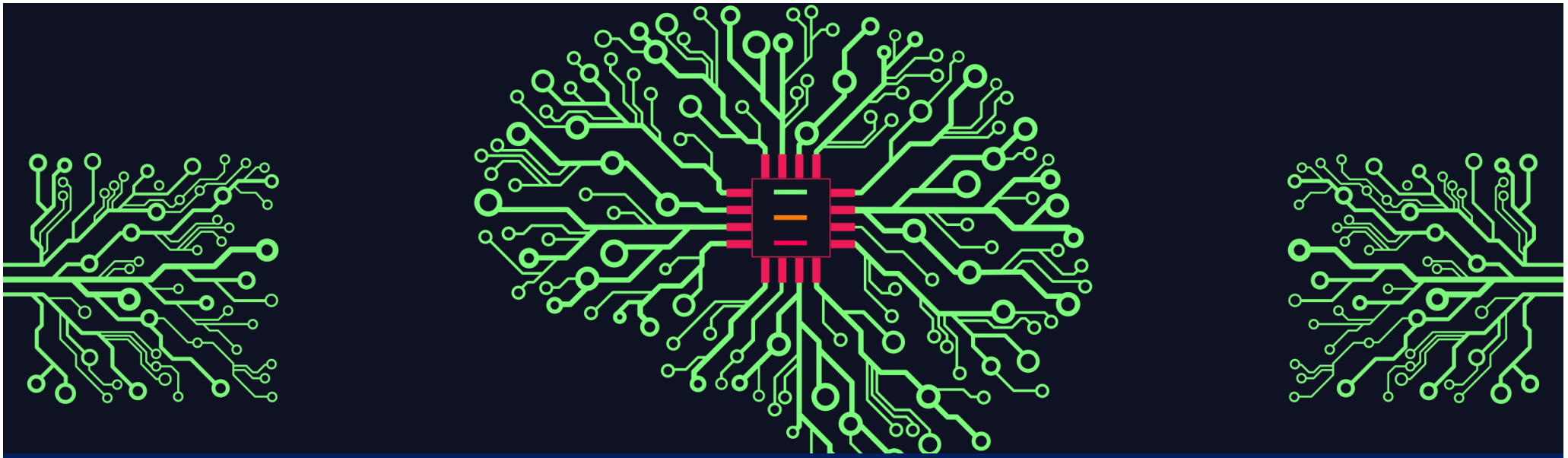
Gather the data

Clean & Explore the data

Model the data

Evaluate the model

Answer the problem



Supervised learning: notations & concepts

SUPERVISED LEARNING

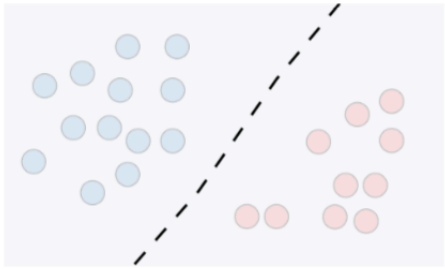
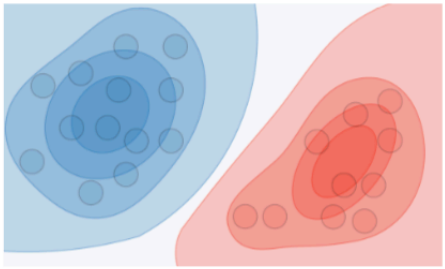
Given $\{x(_1),...,x(_m)\}$ associated to a set of outcomes $\{y(_1),...,y(_m)\}$,

- build a **classifier** that learns how to predict ***y*** from ***x***

Type of prediction

	Regression	Classifier
Outcome	Continuous	Class
Examples	Linear regression	Logistic regression, SVM, Naive Bayes

Type of model

	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Logistic regression, SVM, NN;
K-nearest neighbours,
Conditional Random Fields

Gaussians, Mixture of multinomials,
Mixture of Gaussians,
mixture of experts Hidden Markov Models,
Sigmoid Belief Networks,
Bayesian Markov Random Fields

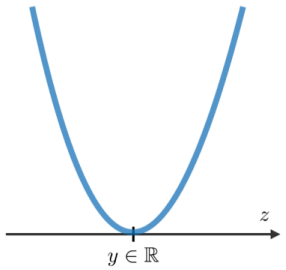
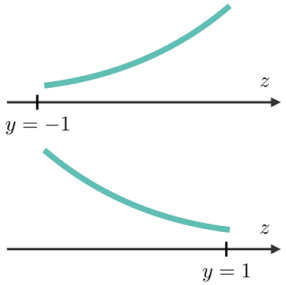
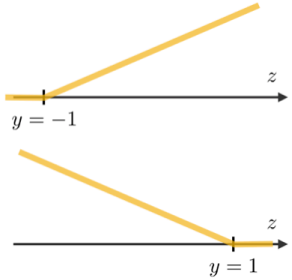
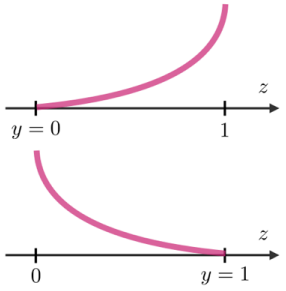
SUPERVISED LEARNING: NOTATIONS & CONCEPTS (1)

Hypothesis: h_θ represents the chosen model

- For a given input data $x^{(i)}$ the model prediction output is $h_\vartheta(x^{(i)})$

Loss function: $L:(z,y)\in\mathbb{R}\times Y \longmapsto L(z,y)\in \mathbb{R}$

- inputs the predicted value z corresponding to the real data value y
- outputs how different they are

Least squared error	Logistic loss	Hinge loss	Cross-entropy
$\frac{1}{2}(y-z)^2$	$\log(1+\exp(-yz))$	$\max(0,1-yz)$	$-\left[y\log(z)+(1-y)\log(1-z)\right]$
			
Linear regression	Logistic regression	SVM	Neural Network

HYPOTHESIS

Scientific: provisional explanation for observations that is falsifiable

Statistical: explanation about the relationship between data populations that is interpreted probabilistically

Machine learning: candidate model that approximates a target function for mapping inputs into outputs

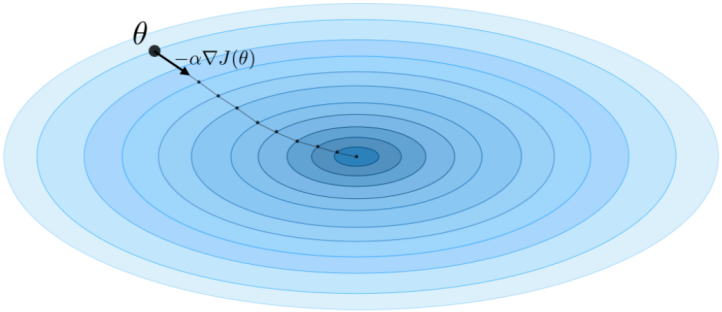
SUPERVISED LEARNING: NOTATIONS & CONCEPTS (2)

Cost function: *J* is used to assess the **performance of a model**, and is defined with the loss function *L* as follows:

$$J(\theta) = \sum_{i=1}^{m} L(h_\theta(x^{(i)}), y^{(i)})$$

- **Gradient descent:** By noting $\alpha \in \mathbb{R}$ the **learning rate**, the **update rule** for gradient descent is expressed with the learning rate and the cost function J as follows:

$$\theta \longleftarrow \theta - \alpha \nabla J(\theta)$$



Remark: Stochastic gradient descent (SGD) is updating the parameter based on each training example, and batch gradient descent is on a batch of training examples.

SUPERVISED LEARNING: NOTATIONS & CONCEPTS (3)

***Likelihood* of *a* a model:** given parameters ϑ, $L(\vartheta)$ is used to find the optimal parameters ϑ through maximizing the likelihood

- In practice, we use the log-likelihood $l(\vartheta) = \log(L(\vartheta))$ which is easier to optimize

$$\boxed{\theta^{\text{opt}} = \arg\max_{\theta} L(\theta)}$$

- ***Newton's algorithm***: is a numerical method that finds ϑ such that $l'(\vartheta) = 0$.

$$\boxed{\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}}$$

- The *multidimensional generalization*, *the Newton-Raphson method*

$$\theta \leftarrow \theta - \left(\nabla^2_\theta \ell(\theta)\right)^{-1} \nabla_\theta \ell(\theta)$$

SUPERVISED LEARNING: NOTATIONS & CONCEPTS (4)

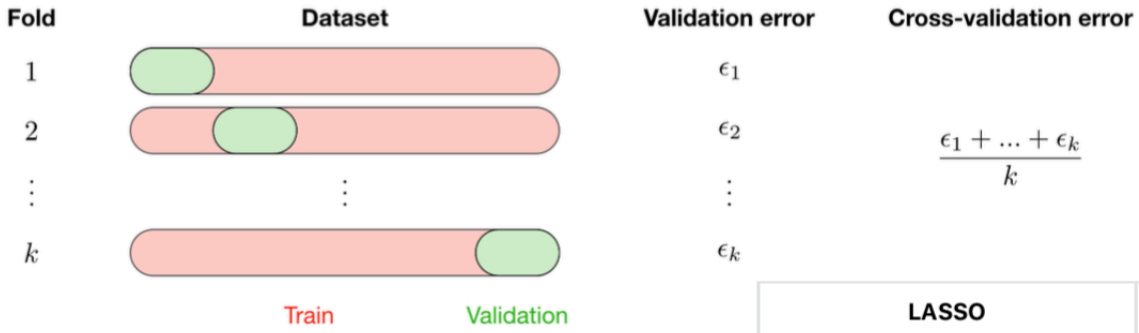
Data set — When selecting a model, we distinguish 3 different parts of the data that we have as follows:

Training set	Validation set	Testing set
• Model is trained • Usually 80% of the dataset	• Model is assessed • Usually 20% of the dataset • Also called hold-out or development set	• Model gives predictions • Unseen data

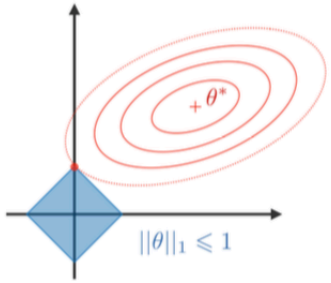
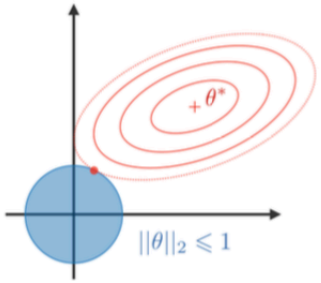
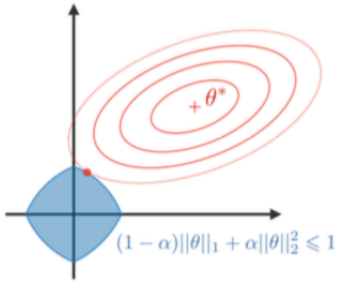
- **Cross-validation** CV, is a method used to select a model that does not rely too much on the initial training set.

k-fold	Leave-p-out
• Training on $k-1$ folds and assessment on the remaining one • Generally $k=5$ or 10	• Training on $n-p$ observations and assessment on the p remaining ones • Case $p=1$ is called leave-one-out

SUPERVISED LEARNING: NOTATIONS & CONCEPTS (5)



- **Regularization —**
  - aims at **avoiding** the model to **overfit** the data
  - deals with **high variance issues**

LASSO	Ridge	Elastic Net
• Shrinks coefficients to 0 • Good for variable selection	Makes coefficients smaller	Tradeoff between variable selection and small coefficients
		
$\ldots + \lambda \lvert\lvert\theta\rvert\rvert_1$ $\lambda \in \mathbb{R}$	$\ldots + \lambda \lvert\lvert\theta\rvert\rvert_2^2$ $\lambda \in \mathbb{R}$	$\ldots + \lambda \left[(1-\alpha) \lvert\lvert\theta\rvert\rvert_1 + \alpha \lvert\lvert\theta\rvert\rvert_2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0,1]$

SUPERVISED LEARNING: NOTATIONS AND CONCEPTS (6)

Bias of a model is the difference between the **expected prediction** and the **correct model** that we try to predict for given data points

Variance — of a model is the **variability** of the **model prediction** for given data points

- **dispersion** (variability, scatter, or spread) is the extent to which a distribution is stretched or squeezed.
- measures of statistical dispersion are the **variance, standard deviation, and interquartile range**

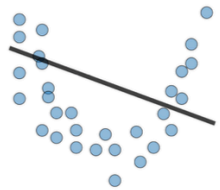
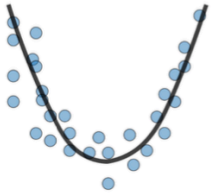
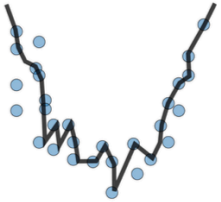
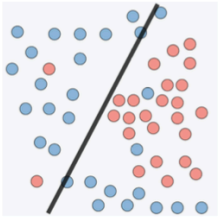
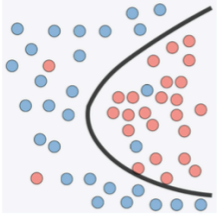
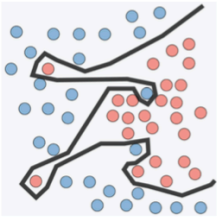
Bias Variance Tradeoff is the tension between the error introduced by the bias and the variance

Error analysis — analyzing the root cause of the difference in performance between the current and the **perfect models**

Ablative analysis analyzing the root cause of the difference in performance between the current and the **baseline models**

SUPERVISED LEARNING: NOTATIONS & CONCEPTS (7)

Bias/variance tradeoff — The simpler the model, the higher the bias, and the more complex the model, the higher the variance

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

SUPERVISED LEARNING: NOTATIONS & CONCEPTS (8)

- **Bayessian estimate** knowledge about the problem/data (prior) look for parameters that explain data
  - Multiple models for making multiple predictions
  - **Minimizes** the posterior expected value of a **loss function**
- **Maximum Likelihood —**
  - method that finds the values of the mean and the std that result in the curve that best fits the data
  - Find parameter values that give the distribution that **maximise the probability of observing the data**

SUPERVISED LEARNING: NOTATIONS & CONCEPTS (9)

Methods for estimating some variable in the setting of probability distributions or graphical models

Maximum likelihood estimation (**MLE**)

- Used when fitting Gaussian: sample mean / sample variance parameters of Gaussian

$P(x / \theta)$ likelihood function $\theta_{MLE} = \text{argmax}_\theta (P(x/\theta) = \text{argmax}_\theta \prod_i P(x_i / \theta)$

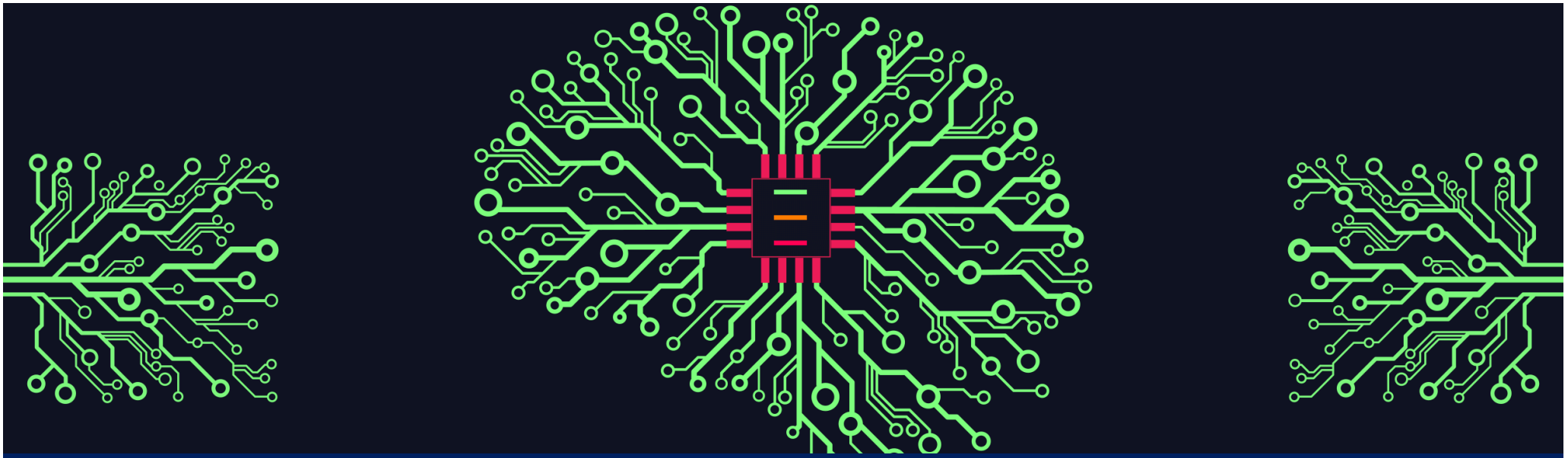
Maximum a posteriori estimation (**MAP**)

- Bayesian setting

$$\theta_{MLE} = \text{argmax}_\theta \sum_i \log P(x_i / \theta)$$

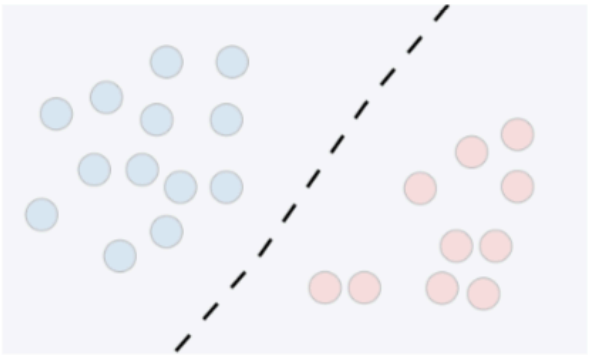
$$P(\theta / x) = \frac{P(x/\theta)\, P(\theta)}{P(x)}$$

$$\theta_{MLE} = \text{argmax}_\theta \sum_i \log P(x_i / \theta) + \log P(\theta)$$

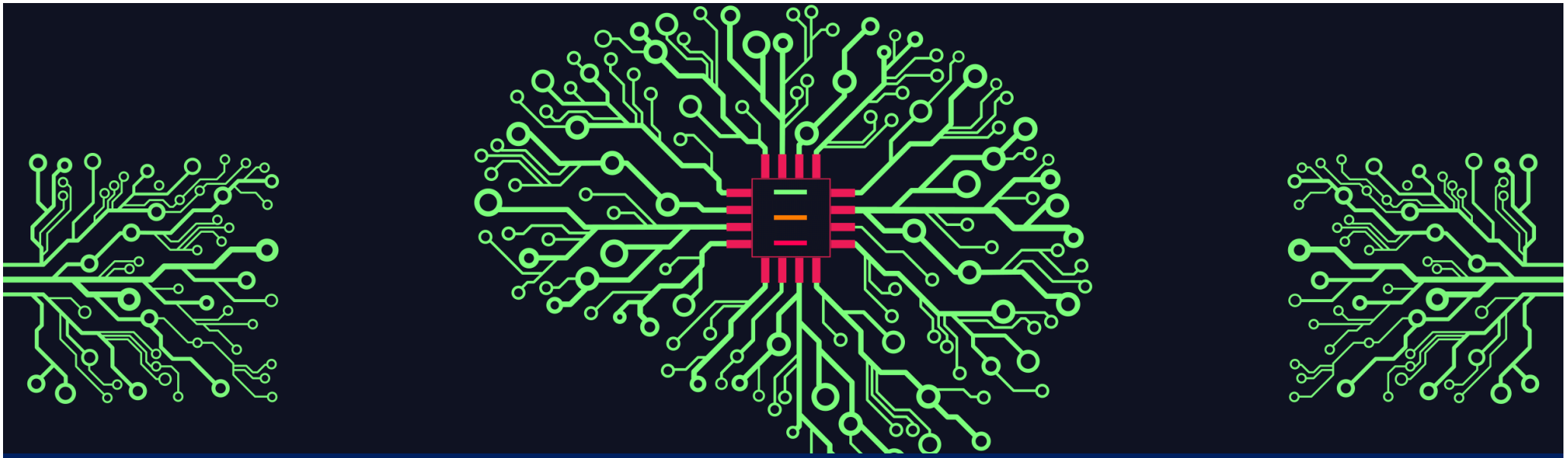


Discriminative models

DISCRIMINATIVE MODEL

	Discriminative model
Goal	Directly estimate $P(y\|x)$
What's learned	Decision boundary
Illustration	
Examples	Regressions, SVMs

Logistic regression, SVM, NN;
K-nearest neighbours,
Conditional Random Fields



Linear regression

LINEAR REGRESSION ASSUMPTIONS

- Linearity of residuals
- Independence of residuals
- Normal distribution of residuals
- Equal variance of residuals

LINEAR REGRESSION
LINEAR MODELS (DISCRIMINATIVE)

Assumptions:

- linearity, independence, normal distribution & equal variance of residuals
- $y \mid x; \vartheta \sim \mathrm{N}(\mu, \sigma^2)$

Normal equations: X the matrix design, the value of ϑ that **minimizes the cost function** is a closed-form solution such that

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

- Design matrix is a matrix of values of explanatory variables of a set of objects.
  - Each row represents an individual object
  - with the successive columns corresponding to the variables and their specific values for that object

LINEAR REGRESSION
LINEAR MODELS (DISCRIMINATIVE)

A way of calculating the relationship between two variables

- y dependent, x independent,
- A and B coefficients determining the slope and intercept of the equation
- A and B calculated to minimize the error between the models prediction and actual data

$$y = Bx + A \text{ , error} = (\text{Actual} - \text{Prediction})^2$$

$$A = \text{mean}(y) - B\ \text{mean}(x)$$

$$B = \text{correlation}(x,y) \times \text{std}(y)/\text{std}(x)$$

For ZX and ZY standardised versions of x and y, means = z, std = 1

$$ZX_i = [X_i - \text{mean}(X)]/\text{std}(X)\ ;\ ZY_i = [Y_i - \text{mean}(Y)]/\text{std}(Y)$$

$$\mathbf{r(X,Y) = \text{sum}[ZX_i \times ZY_i] / n - 1\ ,\ n - \text{sample size}}$$

REGRESSION METRICS (1)

Basic metrics — Given a regression model *f*, the following metrics are commonly used to assess the performance of the model:

Total sum of squares	Explained sum of squares	Residual sum of squares
$\mathrm{SS}_{\mathrm{tot}} = \sum_{i=1}^{m} (y_i - \bar{y})^2$	$\mathrm{SS}_{\mathrm{reg}} = \sum_{i=1}^{m} (f(x_i) - \bar{y})^2$	$\mathrm{SS}_{\mathrm{res}} = \sum_{i=1}^{m} (y_i - f(x_i))^2$

- **Coefficient of determination**— r^2, measure of how well the observed outcomes are replicated by the model [0,1]

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}$$

- **Division ratio**— Sum of squared errors SS_{res} / *Total sum of squares* SS_{tot}

REGRESSION METRICS (2)

Main metrics — commonly used to **assess the performance** of regression models, by taking into account the number of variables *n* that they take into consideration:

Mallow's Cp	AIC	BIC	Adjusted R^2
$\frac{\mathrm{SS}_{\mathrm{res}} + 2(n+1)\widehat{\sigma}^2}{m}$	$2\Big[(n+2) - \log(L)\Big]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1}{m-n-1}$

where L is the likelihood and $\widehat{\sigma}^2$ is an estimate of the variance associated with each response.

Root Mean Square Error (RMSE) : STD of the residuals, how far are data points from the regression line?

$$RMSE_{fo} = [\Sigma_{i=1}^{N} (Z_{fi} - Z_{oi})2 \ / \ N)]^{1/2}$$

$[(f - o)^2]^{1/2}$ f forecasts expected values or unknown results, o observed values

REGRESSION METRICS (3)

P-value or calculated probability: when performing a hypothesis in statistics **determines the strength of the result**s

- Probability of finding the observed or more extreme results when the null hypothesis $\mathbf{H_0}$ of a study question is true
- **Alternative hypothesis $\mathbf{H_1}$** is there a significant (not due to change) different in blood pressures between groups A and B if A receives the drug and B sugar

p-value [0, 1]	*The claim in trial is called the null hypothesis no difference in blood pressure A,B*
p-value <= 0,05	Strength against the null hypothesis: we can reject the null hypothesis
p-value > 0,05	Accept

REGRESSION METRICS WRAP UP

Mallow's Cp	AIC	BIC	Adjusted R^2
$\frac{\mathrm{SS}_{\text{res}} + 2(n+1)\hat{\sigma}^2}{m}$	$2\left[(n+2) - \log(L)\right]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1}{m-n-1}$

where L is the likelihood and $\hat{\sigma}^2$ is an estimate of the variance associated with each response.

Root Mean Square Error (RMSE) : STD of the residuals, how far are data points from the regression line?

$$RMSE_{fo} = [\Sigma_{i=1}^{N} (Z_{fi} - Z_{oi})2 \ / \ N)]^{1/2}$$

f forecasts expected values or unknown results,
o observed values

$[(f - o)^2]^{1/2}$

P-value calculated probability: evaluates the strength of a result

SS_{TOT} total sum squares

SS_{REG} explained sum squares

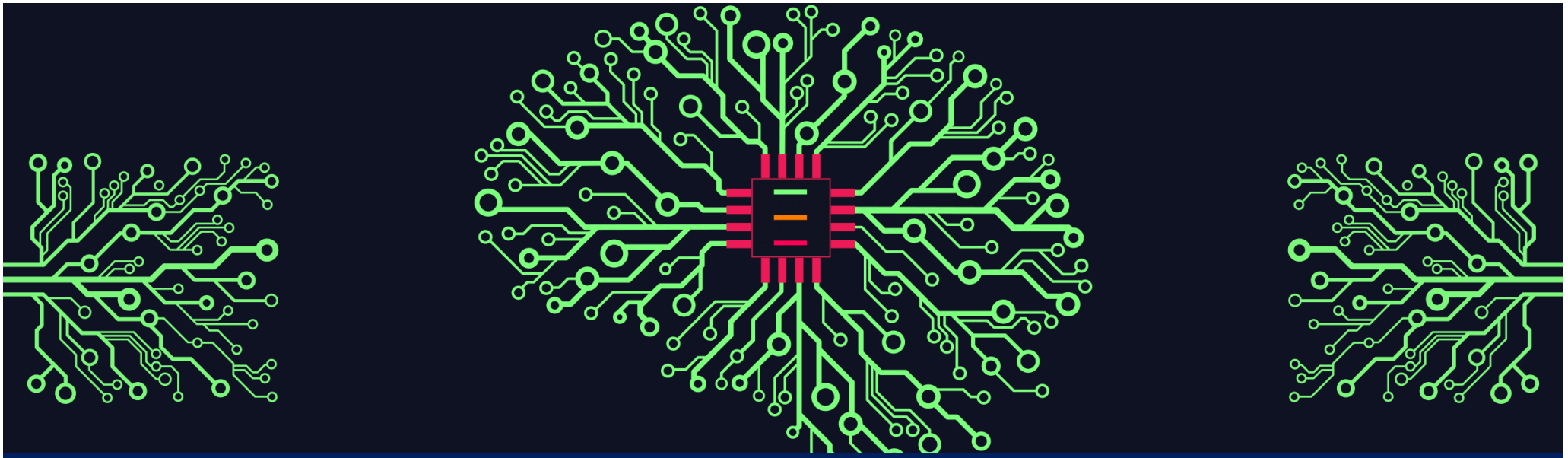
Ss_{res} residual sum squares

Division ratio SS_{res}/SS_{TOT}

$R^2 = 1 - SS_{res}/SS_{TOT})^2$ coefficient determination

Total sum of squares	Explained sum of squares	Residual sum of squares
$\mathrm{SS}_{\text{tot}} = \sum_{i=1}^{m}(y_i - \bar{y})^2$	$\mathrm{SS}_{\text{reg}} = \sum_{i=1}^{m}(f(x_i) - \bar{y})^2$	$\mathrm{SS}_{\text{res}} = \sum_{i=1}^{m}(y_i - f(x_i))^2$


HANDS ON



Logistic regression

LOGISTIC REGRESSION
LINEAR MODELS (DISCRIMINATIVE)

Dependent variable is binary
Observations are independent of each other
Little or no multicollinearity among the independent variables
Linearity of independent variables and log odds

Sigmoid function or logistic function:

$$\forall z \in \mathbb{R}, \quad \boxed{g(z) = \frac{1}{1+e^{-z}} \in]0,1[}$$

- **Logistic regression:** assume that $y \mid x; \vartheta \sim \text{Bernoulli}(\varphi)$

$$\boxed{\phi = p(y=1|x;\theta) = \frac{1}{1+\exp(-\theta^T x)} = g(\theta^T x)}$$

Remark: there is no closed form solution for the case of logistic regressions.

- **Odds ratio** represents the constant effect of predictor X on the likelihood that on output will occur

LOGISTIC REGRESSION PIPELINE (1)
LINEAR MODELS (DISCRIMINATIVE)

Reading data

Basic explanatory data analysis (EDA)

- Find non-numerical values / missing / null values
- Descriptive Analysis: skewness, outliers, mean & median, correlation using pair plot
- Pair plot many distributions each for every variable

Model: select independent attributes, class variables, test size, seed repeatability of the code

Train and test data splitting

Accuracy report

LOGISTIC REGRESSION PIPELINE
LINEAR MODELS (DISCRIMINATIVE)

Step 1: Classifying inputs to be in class 0/1

- Compute the probability that an observation belongs to class 1 using a **logistic response function**
  - **Logit function**
  - **Log odds**

$$P(y=1) = 1 \,/\, 1 + e^{-(\beta 0+\beta 1 xi \ldots +\beta nxn)}$$

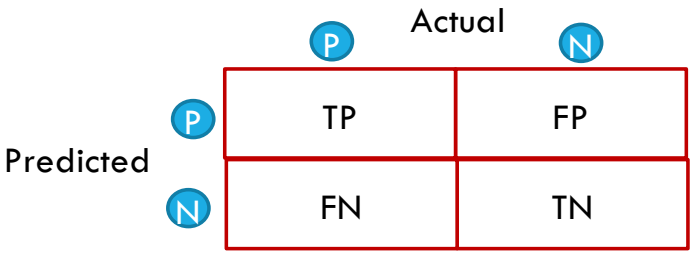
β_i selected to maximize the likelihood of predicting
A high probability to the observations belonging to class 1

Step 2: Defining the boundary for the odds (> 0,5)

- p determines the FN FP to allow

Logit(P) = a+ bx

Odds= P(y=1) / P(y=0) = the odds > 1 with high probability of y=1
the odds < 1 with high probability of y=0



Accuracy: how often is it correct
Precision when P how often is it correct
Recall when actually positive how often is it correctly predicted
F1 harmonic mean
AUC (receiver operating characteristic) TP rate sensitivity, FP rate specificity TN/TN+FP
FPR 1- Specificity

CLASSIFICATION METRICS (1)

In a context of a binary classification

Confusion matrix — used to have a more complete picture when assessing the performance of a model

		Predicted class	
		+	**-**
Actual class	**+**	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

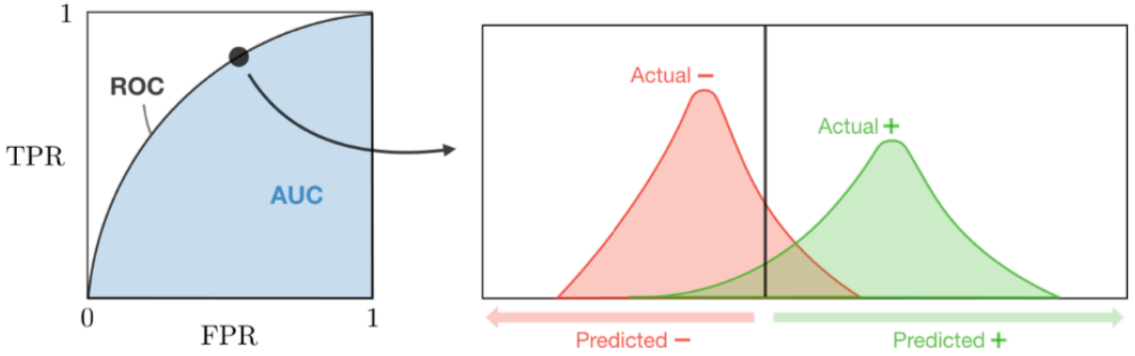
Metric	Formula	Interpretation
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$	Overall performance of model
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$	Coverage of actual positive sample
Specificity	$\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$	Coverage of actual negative sample
F1 score	$\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$	Hybrid metric useful for unbalanced classes

CLASSIFICATION METRICS (2)

Receiver operating curve (ROC), is the plot of TPR versus FPR by varying the threshold

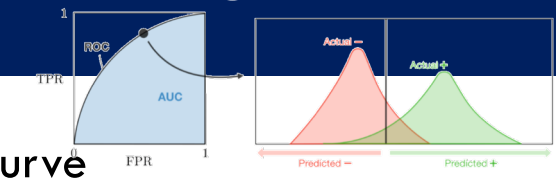
Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$	Recall, sensitivity
False Positive Rate FPR	$\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$	1-specificity

- **AUC/AUROC** — area under the receiving operating curve is the area below the ROC




HANDS ON

LOGISTIC REGRESSION METRICS WRAP UP



Confusion matrix

		Predicted class: +	**Predicted** class: -
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

Metric	Formula	Interpretation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

ROC receiver operating curve

Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Recall, sensitivity
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity

Maximum Likelihood Estimation (MLE)

$P(x / \theta)$ likelihood function

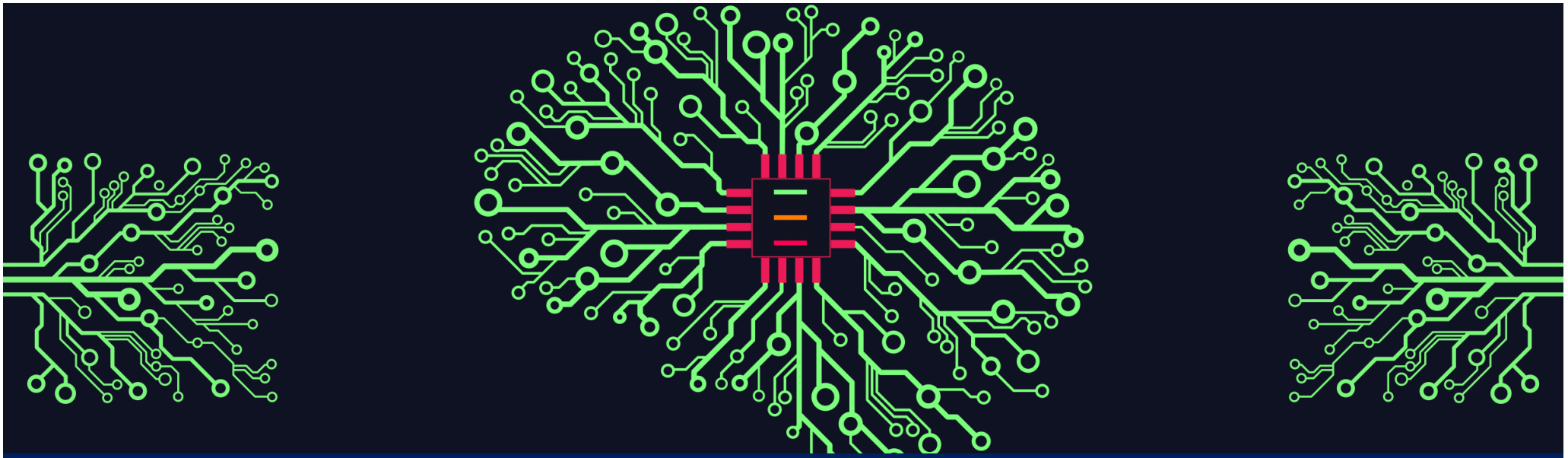
$$\theta_{MLE} = \text{argmax}_\theta (P(x/\theta) = \text{argmax}_\theta \prod_i P(x_i / \theta)$$

$$\theta_{MLE} = \text{argmax}_\theta \sum_i \log P(x_i / \theta)$$

Maximum a priori estimation (MAPE)

$$P(\theta / x) = \frac{P(x/\theta)\ P(\theta)}{P(x)}$$

$$\theta_{MAPE} = \text{argmax}_\theta \sum_i \log P(x_i / \theta) + \log P(\theta)$$



Support Vector Machine

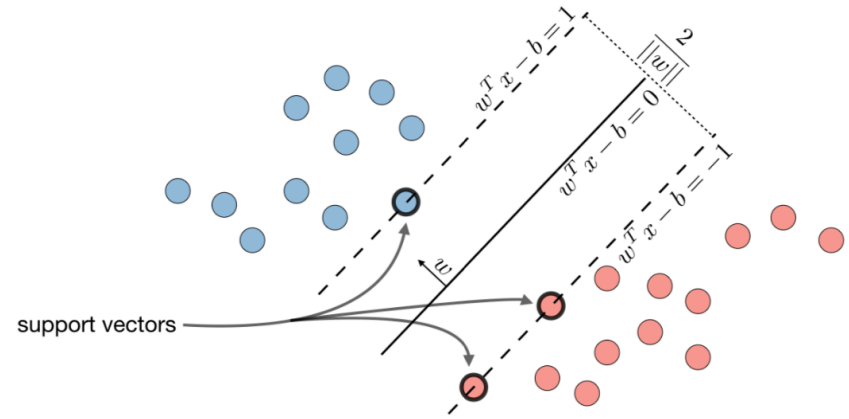
SUPPORT VECTOR MACHINE (1) (DISCRIMINATIVE)

Objective: find the line that maximizes the minimum distance to the line

Optimal margin classifier h

$$\boxed{h(x) = \text{sign}(w^T x - b)}$$

where $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ is the solution of the following optimization problem



Remark: the line is defined as $\boxed{w^T x - b = 0}$.

$$\boxed{\min \frac{1}{2}||w||^2} \quad \text{such that} \quad \boxed{y^{(i)}(w^T x^{(i)} - b) \geqslant 1}$$

SUPPORT VECTOR MACHINE (2) (DISCRIMINATIVE)

Hinge loss: used in the setting of SVMs and is defined as follows:

$$\boxed{L(z,y) = [1 - yz]_+ = \max(0, 1 - yz)}$$

- **Kernel** — Given a feature mapping φ, we define the kernel K to be defined as:

$$\boxed{K(x,z) = \phi(x)^T \phi(z)}$$

In practice, the kernel K defined by $K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$ is called the Gaussian kernel and is commonly used.

- **Lagrangian** — We define the Lagrangian L(*w*, *b*) as follows:

$$\boxed{\mathcal{L}(w,b) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)}$$

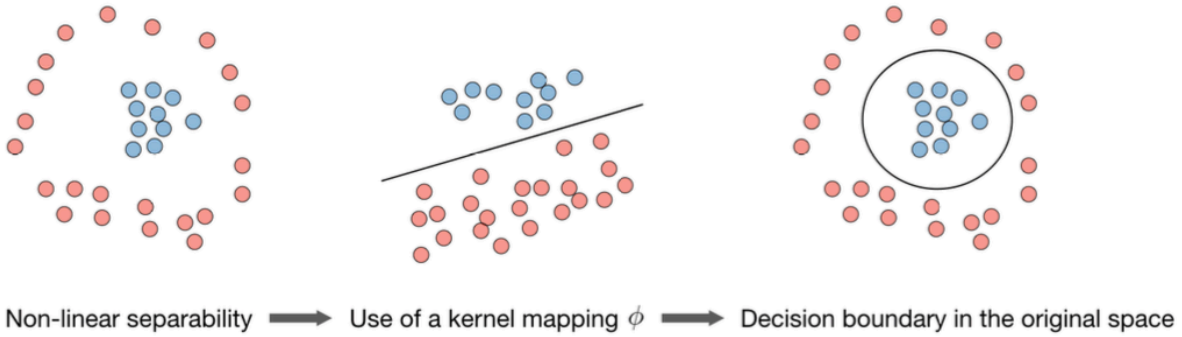
Remark: the coefficients β_i are called the Lagrange multipliers.

SUPPORT VECTOR MACHINE (3) (DISCRIMINATIVE)

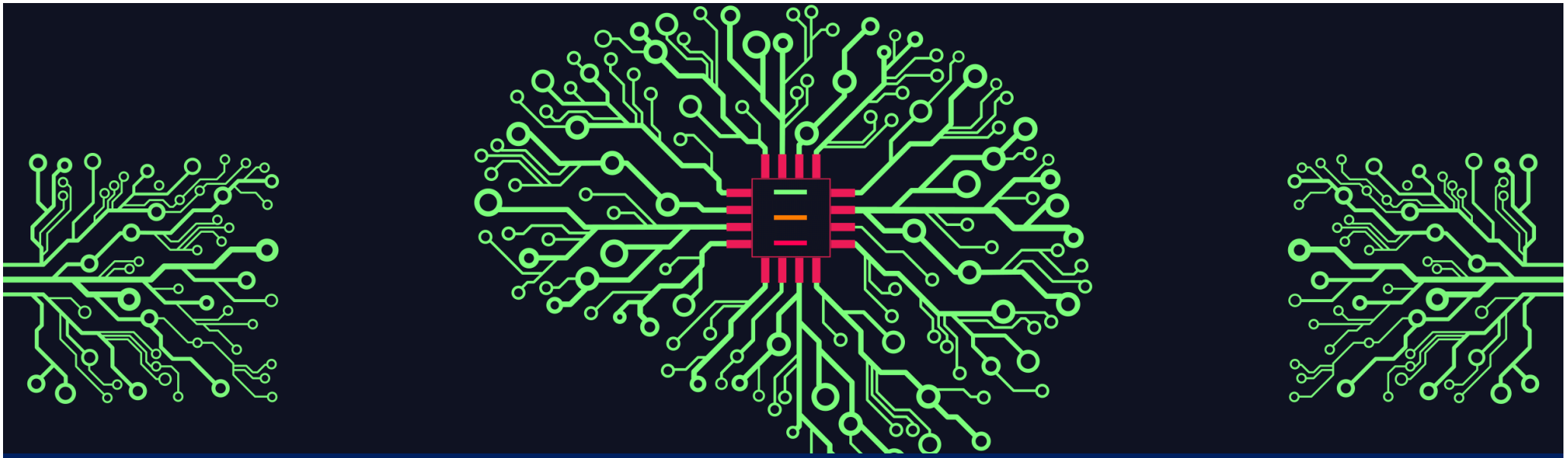
- **Lagrangian** — We define the Lagrangian L(*w*, *b*) as follows:

$$\mathcal{L}(w, b) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

Remark: the coefficients β_i are called the Lagrange multipliers.



Remark: we say that we use the "kernel trick" to compute the cost function using the kernel because we actually don't need to know the explicit mapping ϕ, which is often very complicated. Instead, only the values $K(x, z)$ are needed.



ARIMA

ARIMA PIPELINE

1. Identify if the model **is multiplicative** or **additive**
2. Identify **time series components**: trend, cycle, seasonality, residuals
3. Transform data to make it **linear**
4. Make data **stationary** if it is not
5. Based on 2 choose ARIMA or SARIMA
6. Define order parameter for each model variable/feature
7. Do **grid search** and choose an **optimal model** based on **AIC, BIC, HQIC**
8. Check if model residuals comply with **ordinary least squares (OLS)**
9. Forecast and calculate forecasting **error**: MAPE, MAE, etc

CHANGE POINT DETECTION (CPD)

Detects abrupt shifts in time series trends

- Shifts in a time series velocity hard to pinpoint using traditional statistical approaches

Detect **anomalous sequences / states** in time series

Detect the **average velocity of unique states** in a time series

Detect **sudden change** in a time series state in real time

Offline: require the complete time series for statistical analysis

Online: “on the fly” processing

Search methods: binary segmentation, PELT (Pruned Extract Linear Time), window based change detection, dynamic programming

ARIMA (TIME SERIES -1)

Time series

- f(t) = y sequence of data points measured over time intervals
- Data points can be measured hourly, daily, weekly, monthly …

Elements: trend, seasonality, cycles, residuals

- **Trend**: **additive or multiplicative** times series determined by the way components are combined
- **Seasonality**: true effect of events on a data set which may overlap with a season
- **Residuals** R ~N(0,1)

Stationary data point d_i is independent of previous one d_{i-1}

- **Statistical properties** of data such as mean, variance and standard deviation remain **constant** over time
- Augmented Dickey Fuller test (ADF) p >= 0,05 data non stationary

Autocorrelation (**ACF**) and partial autocorrelation plots (**PACF**): of time series data with its lagged values

- Autoregressive process: present value of the time series can be obtained using previous values
- Stock prices, global temperature
- Optimum features or order of the process obtained PAC:F **MA- ACP**

ARIMA (TIME SERIES CROSS VALIDATION)

Time series ordered by chronological order

Cross validation statistical technique involves partitioning data into subsets: training data; evaluate performance

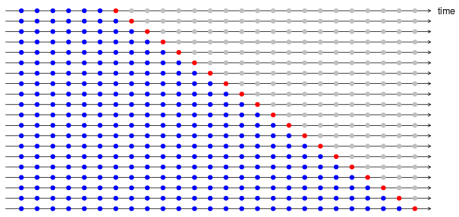
Reduce variability

- amount of spread in a set of data: range interquartile range standard deviation

$$\sigma = [(1/N\text{-}1) \sum_{i\text{-}1}^{N} (x_i - \bar{x})^2]^{1/2}$$

- perform multiple rounds of cross validation and combine results computing models predictive performance
- Techniques: leave one out LOO CV, k-fold, stratified, time series, mean squared error

Forward chaining (rolling origin)



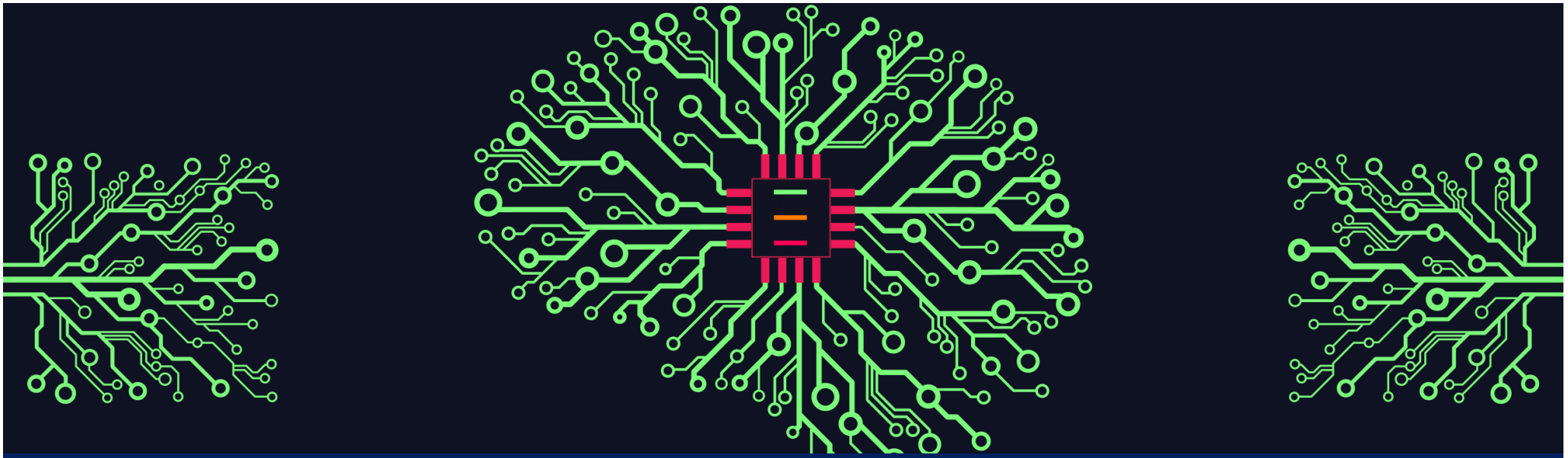
ARIMA: MODEL SELECTION

Independent lagged values, time series dependent

Model selection

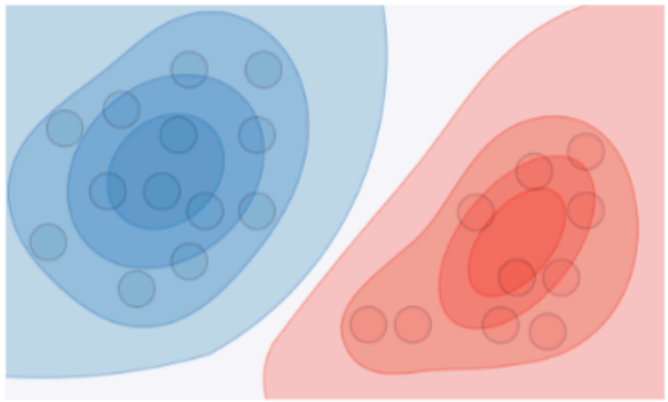
- AIC Akike information Criterion
- BIC Bayesian Information Criterion
- OLS ordinary least criterion
- HQIC Hannon Quin information criterion = $2L_{max} + 2k \ln(\ln(n))$
- MAPE: low values


HANDS ON

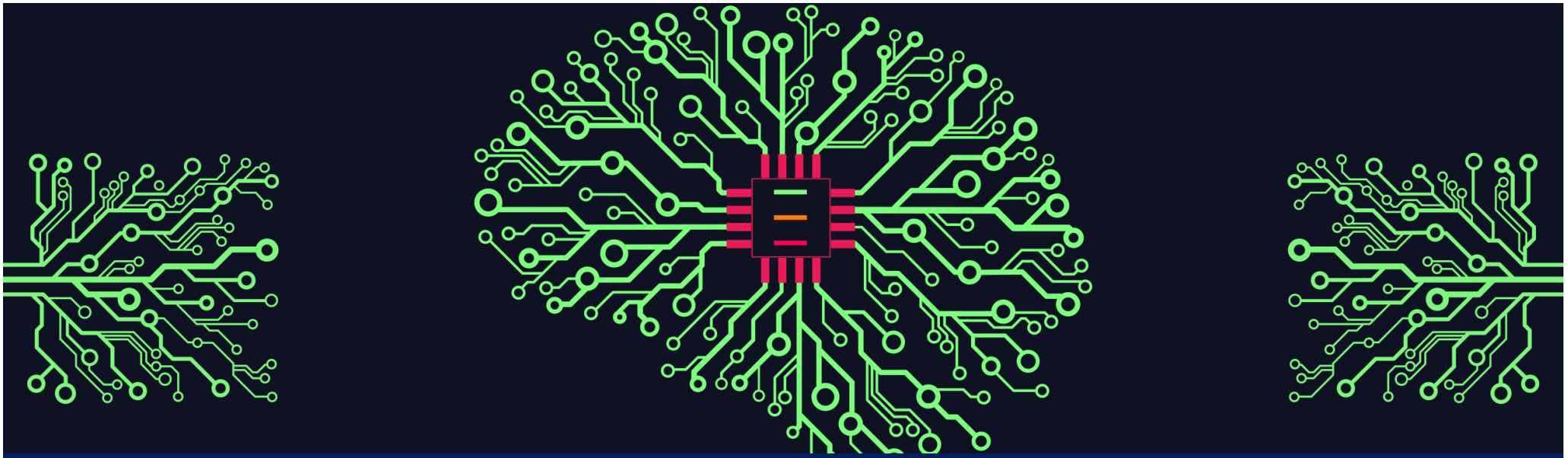


Supervised learning: generative models

GENERATIVE MODELS

Generative model
Estimate $P(x\|y)$ to then deduce $P(y\|x)$
Probability distributions of the data

GDA, Naive Bayes

Gaussians, Mixture of multinomials,
Mixture of Gaussians,
mixture of experts Hidden Markov Models,
Sigmoid Belief Networks,
Bayesian Markov Random Fields



Naïve Bayes

NAIVE BAYES
(GENERATIVE LEARNING)

Posterior probability | target | Prior | Likelihood

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

predictor | Evidence

- **Assumption** — The Naive Bayes model supposes that the features of each data point are all independent:

$$P(x|y) = P(x_1, x_2, \ldots |y) = P(x_1|y)P(x_2|y)\ldots = \prod_{i=1}^{n} P(x_i|y)$$

- **Solutions** — Maximizing the log-likelihood gives the following solutions, with $k \in \{0\ ,\ 1\}$, $l \in [1\ ,\ L]$

$$P(y = k) = \frac{1}{m} \times \#\{j|y^{(j)} = k\} \quad \text{and} \quad P(x_i = l|y = k) = \frac{\#\{j|y^{(j)} = k \text{ and } x_i^{(j)} = l\}}{\#\{j|y^{(j)} = k\}}$$

Remark: Naive Bayes is widely used for text classification and spam detection.

- **Conjugate distributions:**
  - if Prior and posterior are in the same probability distribution family
  - The prior is called conjugate prior for the likelihood function

NAÏVE BAYES PIPELINE

Frequency Table		
Weather	NO	YES
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Posterior probability

target

Prior

Likelihood

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

predictor

Evidence

NAÏVE BAYES PIPELINE

Convert the dataset into a **frequency table**

Create a **likelihood table** finding the probabilities

Apply Naïve Bayes equation

The class with the **highest posterior probab**ility is the **prediction outcome**

Frequency Table		
Weather	NO	YES
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Posterior probability **target** **Prior** **Likelihood**

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

predictor **Evidence**

NAÏVE BAYES PIPELINE

Convert the dataset into a **frequency table**

Create a **likelihood table** finding the probabilities

Apply Naïve Bayes equation

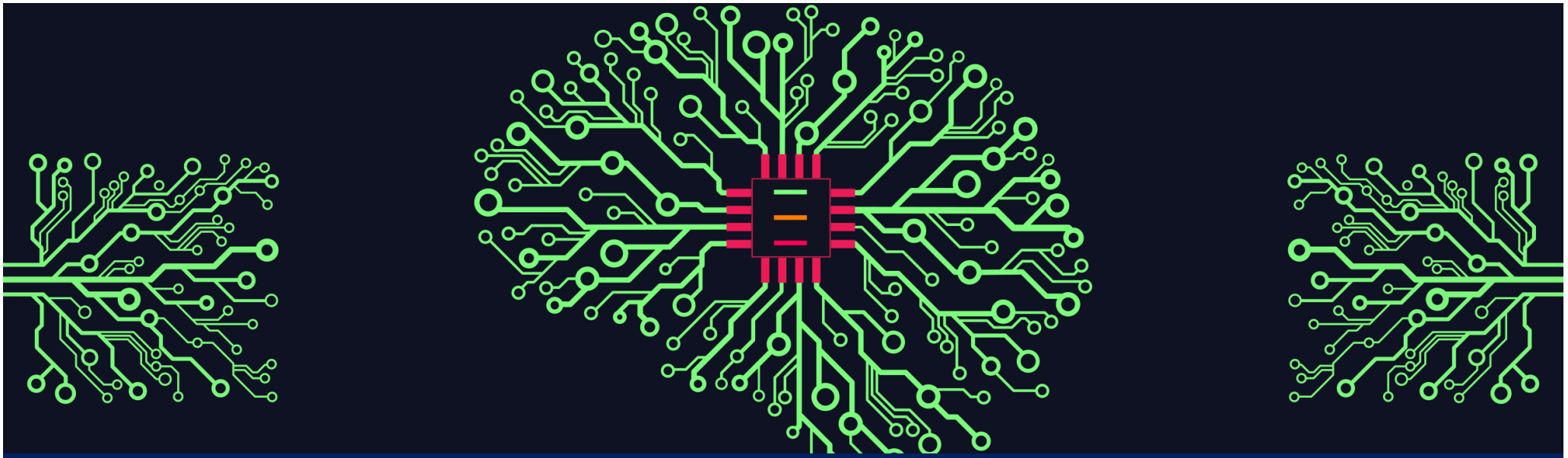
The class with the **highest posterior probab**ility is the **prediction outcome**

Frequency Table

Weather	NO	YES
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood Table

Weather	NO	YES		
Overcast		4	**=4/14**	**0,29**
Rainy	3	2	=5/14	0,36
Sunny	2	3	=5/14	0,36
Grand Total	5	9		
	5/14	9/14		
	0,36	0,64		



Tree based ensemble methods

TREE BASED ENSEMBLE METHODS (GENERATIVE LEARNING)

- These methods can be used for **both regression and classification** problems
- **Classification and Regression Trees (CART),** known as **decision trees,** can be represented as binary trees
- **Random forest** — tree-based technique that uses a high number of decision trees built out of randomly selected sets of features
- **Boosting** — The idea of boosting methods is to combine several weak learners to form a stronger one

Adaptive boosting	Gradient boosting
• Known as Adaboost • High weights are put on errors to improve at the next boosting step	• Weak learners trained on remaining errors

TREES: SPLITTING NODES DECISION MAKING ALGORITHMS

Gini index if two items are selected from a population at random they must be of the same class and probability

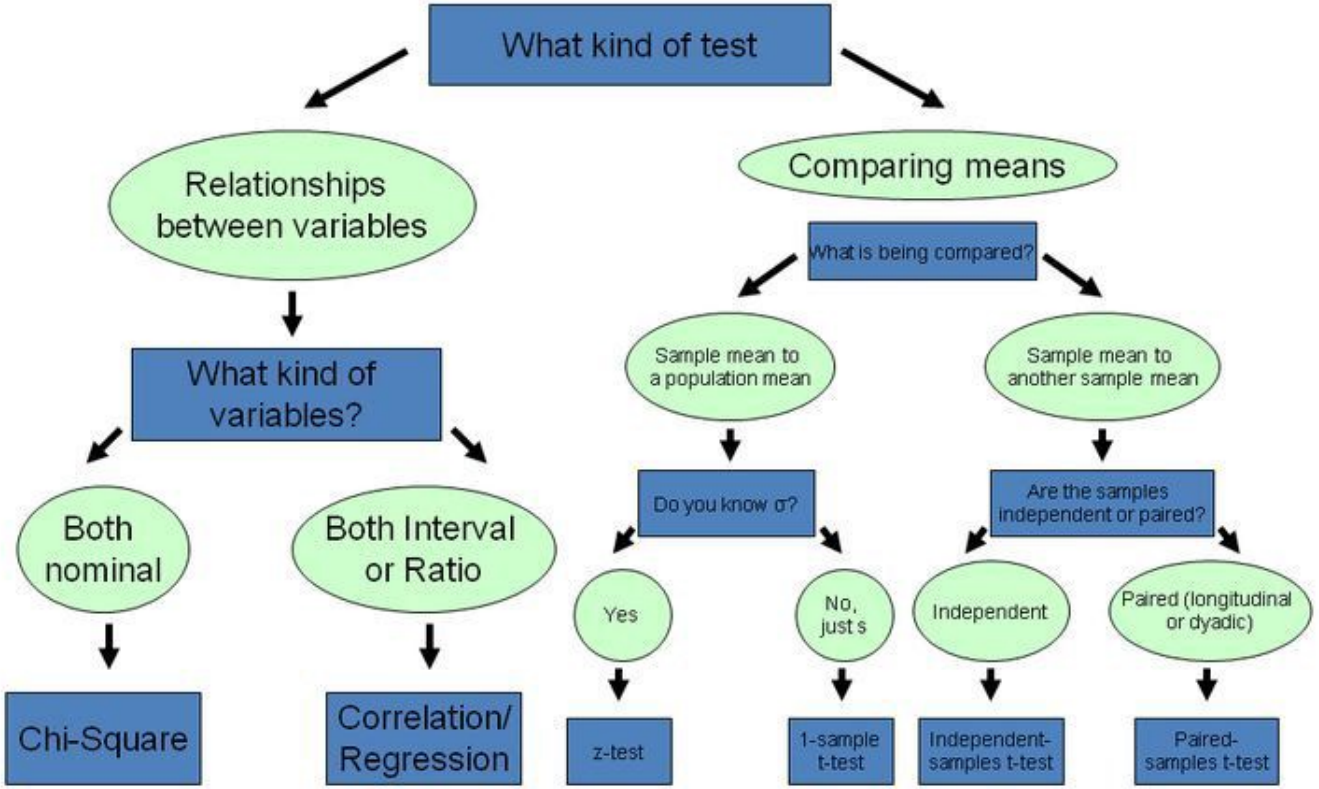
- Gini coefficient measures the inequality among values of a frequency distribution 0 perfect equality
- Binary splits
  - p = 1 --> pure population ; (p^2+q^2) sum of square of probability; calculate gini using a weighting score for each node at split

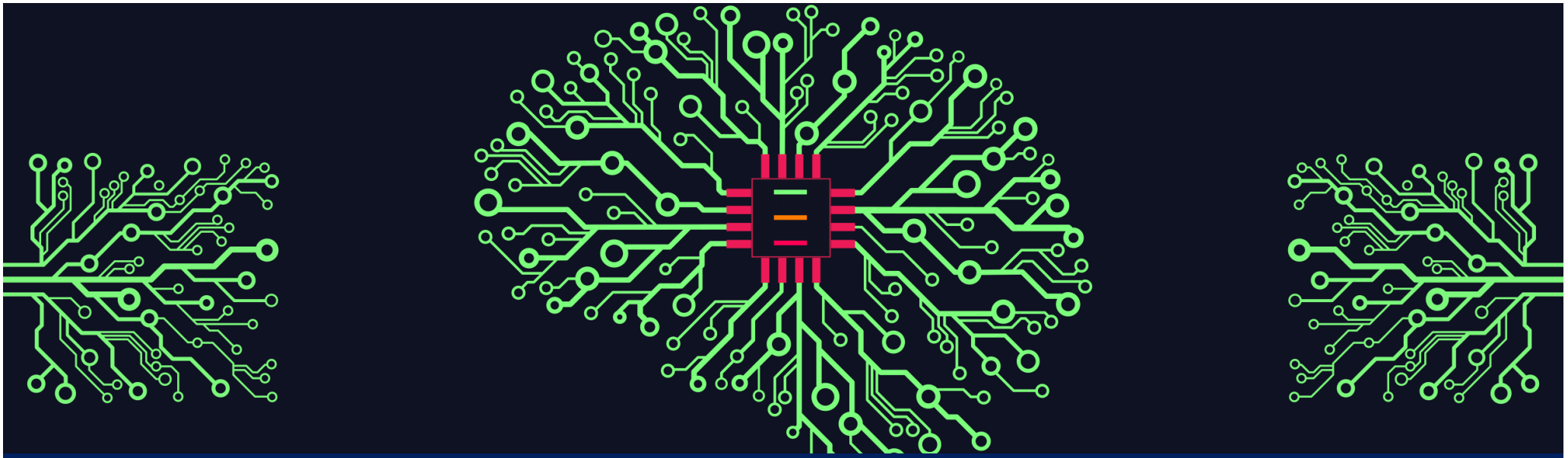
Chi-square statistical significance among the differences between the subnodes and parent node: $chi^2 = ((actual - expected)^2/expected)^{1/2}$

Information gain = 1 - Entropy

- **Entropy** = $- p \log_2 p - q \log_2 q$ (p, q resp success and failure)

Decision Tree



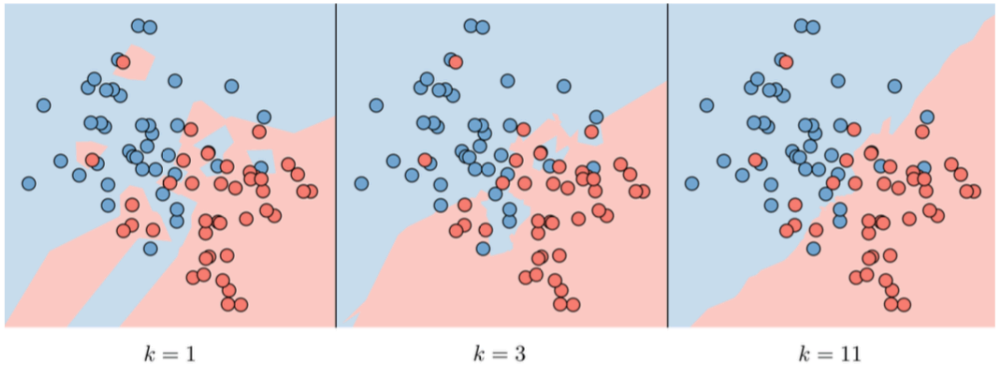


K-nearest neighbours

K-NEAREST NEIGHBOURS (NON PARAMETRIC APPROACHES)

- **Algorithm *k*-NN**, where the response of a data point is determined by the nature of its *k* neighbors from the training set
- It can be used in both classification and regression settings

Remark: The higher the parameter k, the higher the bias, and the lower the parameter k, the higher the variance.



- **Manhattan distance**, between a point P and a line L is defined as the smallest distance between any point M in L on the line and P

$$d(P, L) = \min_{M \in L} (d(M,P))$$
$$d(M, P) = | M_x - P_x | + | M_y - P_y |$$

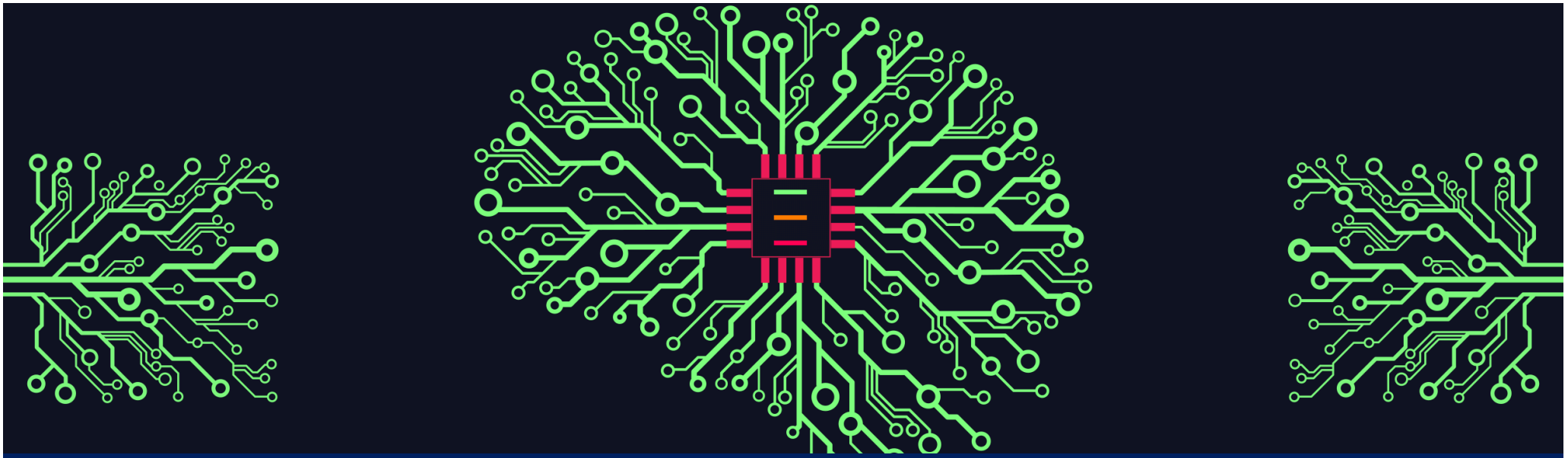
DETERMINE K

Elbow curve plotting (bending point)

- plot the percentage of variance explained by the clusters against
- the number of clusters
- the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph

Hierarchical clustering

Hyperparameters



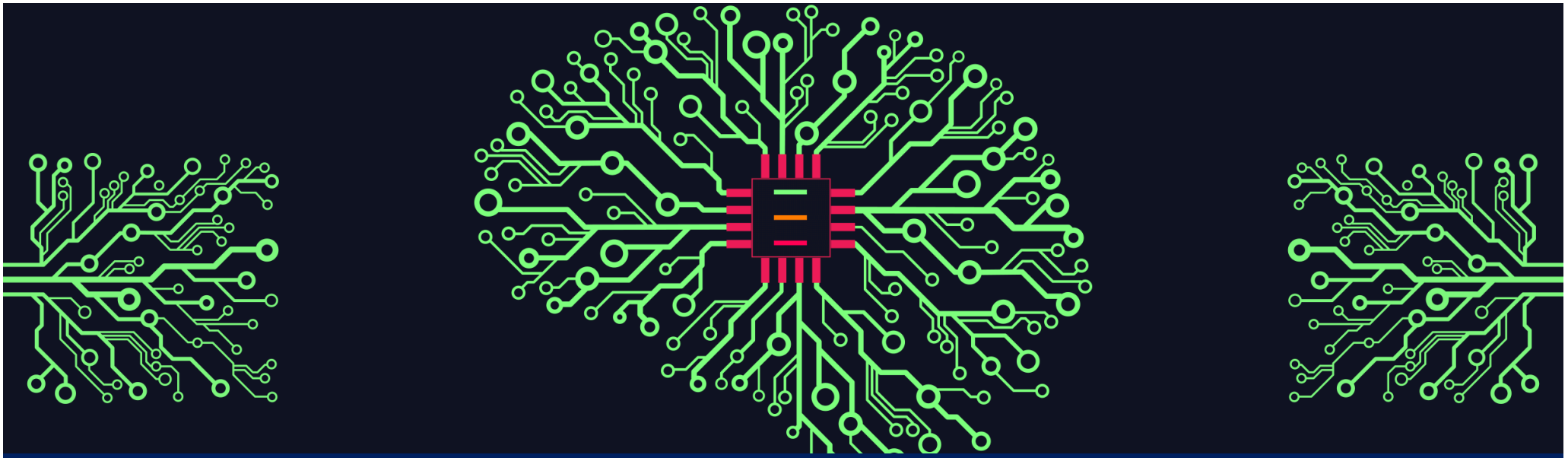
Unsupervised learning

UNSUPERVISED LEARNING

Motivation — The goal of unsupervised learning is to **find hidden patterns** in **unlabeled data** $\{ x(1),...,x(m)\}$.

Jensen's inequality — Let f be a convex function and X a random variable

$$\boxed{E[f(X)] \geqslant f(E[X])}$$



K-means

ASPECTS TO CONSIDER

- What is a natural grouping among the objects?

  → Need to define the "**groupness**" and the "**similarity/distance**" among data

- How can we group samples?
  - What are the best procedures?
  - Are they efficient? Are they fast? Are they deterministic?

How many clusters should we look for in the data?
- Shall we state this number a priori?
- Should the process be completely data driven or can the user guide the grouping process?
- How can we avoid "trivial" clusters?
- Should we allow final clustering results to have very large or very small clusters?
- Which methods work when the number of samples is large? Which methods work when the number of classes is large?

What constitutes a good grouping?
- What objective measures can be defined to evaluate the quality of the clusters?

DISTANCES

The most widespread distance metric is the *Minkowski distance*:

$$d(a,b) = (\sum_{i=1}^{d} |a_i - b_i|^p)^{1/p}$$

- `d(a, b)` stands for the distance between two elements $a, b \in \mathbb{R}^d$
- `d` is the dimensionality of the data
- `p` is a parameter

The best-known instantiations of this metric are as follows:

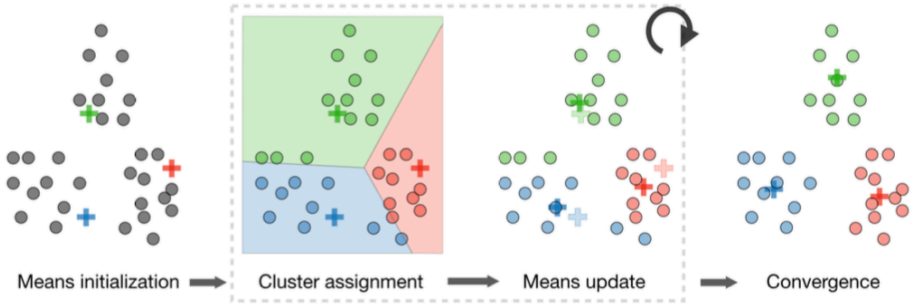
- `p = 2`: *Euclidean distance*
- `p = 1`: *Manhattan distance*, and
- `p = inf`: *max-distance* corresponding to the component $|a_i - b_i|$ with the highest value

K-MEANS (CLUSTERING)

We note c(*i*) the cluster of data point *i* and μj the center of cluster *j*.

Algorithm — After randomly initializing the cluster centroids $\mu 1 , \mu 2 , \ldots , \mu k \in \mathbb{R}n$, repeats the following until convergence:

$$c^{(i)} = \arg\min_j ||x^{(i)} - \mu_j||^2 \quad \text{and} \quad \mu_j = \frac{\sum_{i=1}^{m} 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^{m} 1_{\{c^{(i)}=j\}}}$$



Distortion function — determine if the algorithm converges re-estimate each cluster model as follows:

$$J(c, \mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

HIERARCHICAL CLUSTERING (CLUSTERING)

It is a clustering algorithm with an agglomerative hierarchical approach that build nested clusters in a successive manner.

There are different sorts of hierarchical clustering algorithms that aims at optimizing different objective functions

Ward linkage	Average linkage	Complete linkage
Minimize within cluster distance	Minimize average distance between cluster pairs	Minimize maximum distance of between cluster pairs

ASSESSMENT METRICS (CLUSTERING)

Silhouette coefficient — By noting *a* and *b* the mean distance between a sample and all other points in the same class, and between a sample and all other points in the next nearest cluster, the silhouette coefficient *s* for a single sample is:

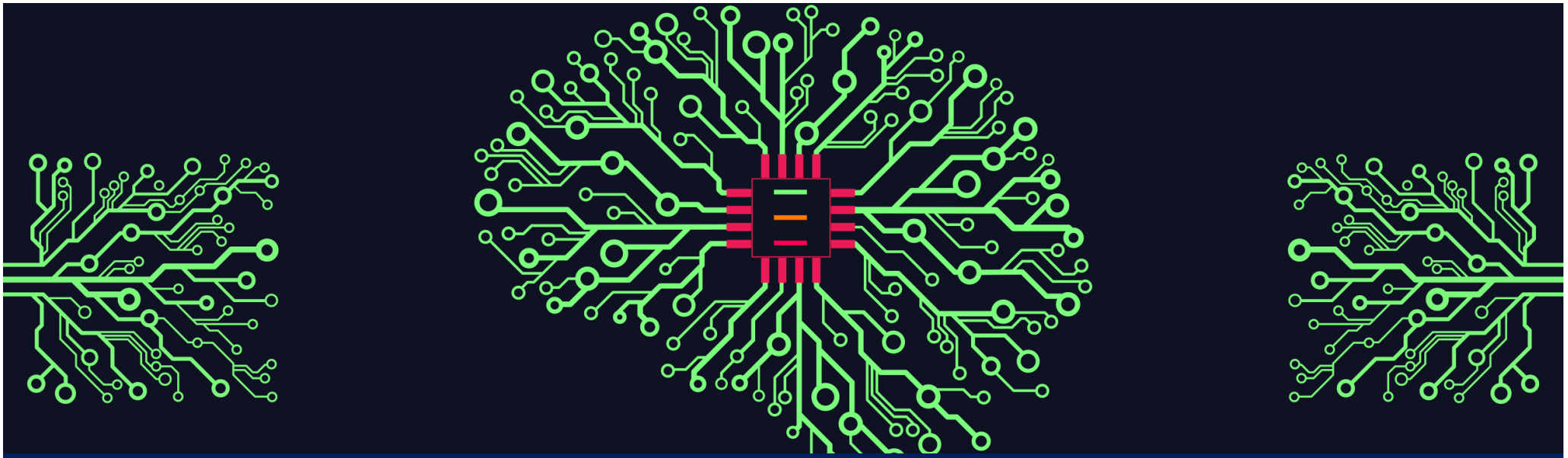
$$\boxed{s = \frac{b - a}{\max(a, b)}}$$

Calinski-Harabaz index — By noting *k* the number of clusters, *Bk* and *Wk* the between and within-clustering dispersion matrices respectively defined as

$$B_k = \sum_{j=1}^{k} n_{c^{(i)}} (\mu_{c^{(i)}} - \mu)(\mu_{c^{(i)}} - \mu)^T, \qquad W_k = \sum_{i=1}^{m} (x^{(i)} - \mu_{c^{(i)}})(x^{(i)} - \mu_{c^{(i)}})^T$$

The Calinski-Harabaz index *s*(*k*) indicates how well a clustering model defines its clusters, such that the higher the score, the more dense and well separated the clusters are

$$\boxed{s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N - k}{k - 1}}$$



Principal component analysis

PRINCIPAL COMPONENT ANALYSIS (PCA, 1)
(DIMENSION REDUCTION)

It is a dimension reduction technique that finds **the variance maximizing directions** onto which to project the data

Eigenvalue, eigenvector — Given a matrix $A \in \mathbb{R}^{n \times n}$, λ is said to be an eigenvalue of A if there exists a vector $z \in \mathbb{R}^n \setminus \{0\}$, called eigenvector :

$$\boxed{Az = \lambda z}$$

Spectral theorem — Let $A \in \mathbb{R}^{n \times n}$. If A is symmetric, then A is diagonalizable by a real orthogonal matrix $U \in \mathbb{R}^{n \times n}$. By noting $\Lambda = \text{diag}(\lambda 1, \ldots, \lambda n)$, we have:

$$\boxed{\exists \Lambda \text{ diagonal}, \quad A = U \Lambda U^T}$$

Remark: the eigenvector associated with the largest eigenvalue is called principal eigenvector of matrix A.

PRINCIPAL COMPONENT ANALYSIS (PCA, 2)
(DIMENSION REDUCTION)

Step 1: Normalize the data to have a mean of 0 and standard deviation of 1

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \text{where} \quad \mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \quad \text{and} \quad \sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$$

Step 2: Compute symmetric with real eigenvalues

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)T} \in \mathbb{R}^{n \times n},$$

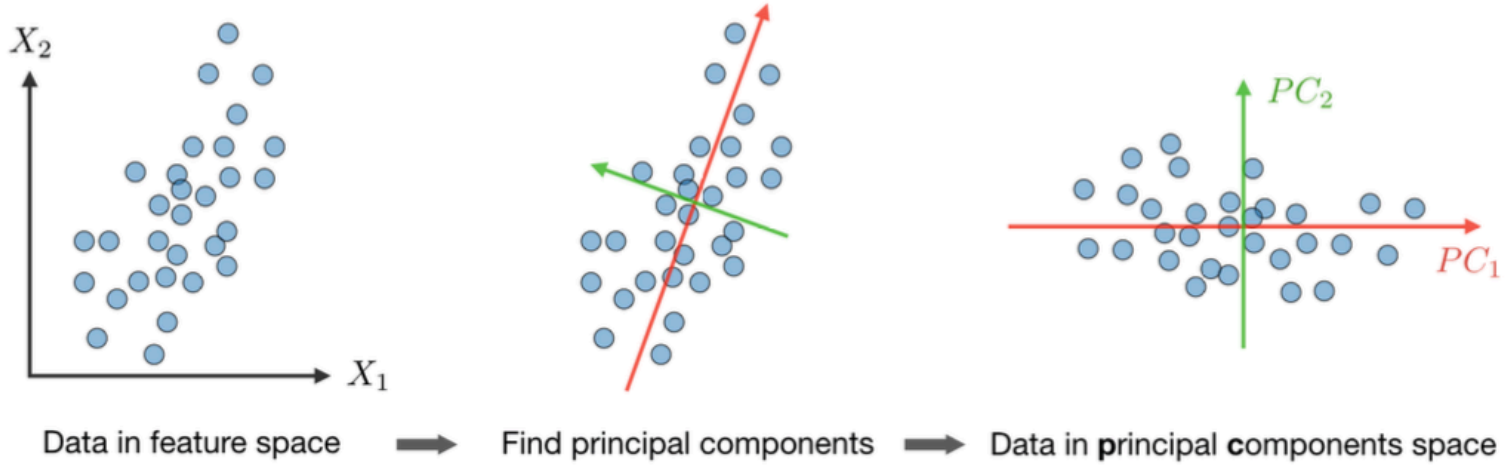
Step 3: Compute $u_1, \ldots, u_k \in \mathbb{R}^n$ igenvectors of Σ, i.e. the orthogonal eigenvectors of the k largest eigenvalues

Step 4: Project the data on $\text{span}_{\mathbb{R}}(u_1, \ldots, u_k)$

This procedure maximizes the variance among all k-dimensional spaces

PRINCIPAL COMPONENT ANALYSIS (PCA, 3) (DIMENSION REDUCTION)

This procedure maximizes the variance among all *k*-dimensional spaces



FEATURE SELECTION / DIMENSION REDUCTION

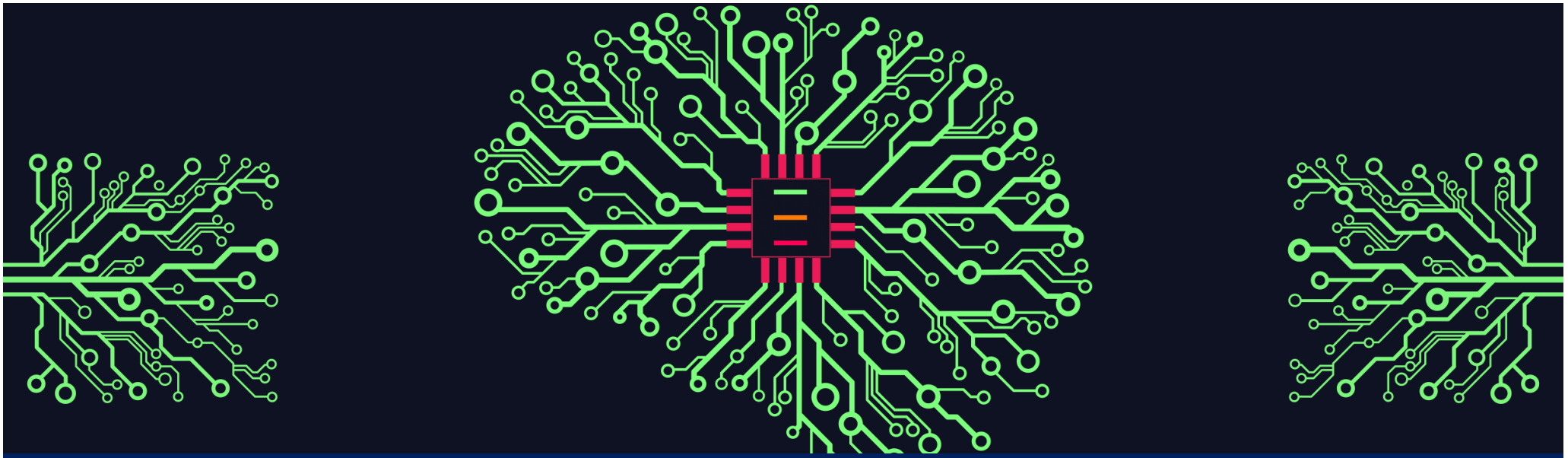
1. **Missing values ratio:** remove columns with too many missing values
2. **Low variance filter:** remove columns with little changes applying normalisation first
3. **High correlation filter:** columns with high correlations reduced to 1
4. **Random Forest / Ensemble trees**
5. **Principal Component Analysis (PCA)** statistical procedure that transforms the original n coordinates of a data set into a new set n coordinates called PC
    1. The first PC has the largest possible variance
    2. Succeeding PCs with the highest possible variance iff orthogonal with preceeding PC's
    3. m < n first reduces data dimensionality retaining the most data information (i.e., variation in data)
6. **Backward feature elimination:** eliminate features sequentially at every step of a classification algorithm training
    1. Remove the feature producing the smallest increase in error rate
7. **Forward feature construction:** add progressively features use features producing the highest increase in performance

DIMENSIONS REDUCTION

Latent Semantic Analysis (LSA) reduce dimensions for classification

Latent Dirichet Analysis (LDA) based on Term Frequency (TF) and IDF inversed data frequency for defining topics

--> Topic modelling



Final remarks

MACHINE LEARNING CHEAT SHEET

Skewed data: log function+1, Normalisation (sensitive to outliers) not for tree models, linear models (0,1)

Hot encoding: convert categorical to binary variables

Imbalanced data (not well distributed)

- Proper evaluation metrics, under sampling (*abundant class* Tomek links KNN AB) – over sampling (insufficient quantity of data), k-fold cross-validation, deal with abundant classes, use penalised models

Drop/not drop outliers

Feature selection

- Missing values ratio: remove columns with too many missing values
- Law variance filter: remove columns with data with little changes
- High correlation filter: columns with high correlation reduced to 1
- Random forest/Ensemble trees
- PCA (principal components analysis): m transform original n coordinates to n PCs use decreasing variance of PC for including them according to highest possible variance teste subsequently
- Backward feature elimination: eliminate features subsequently remove those producing the smallest increase of error rate
- Forward feature elimination: construct 1 by 1 measuring increase of performance


HANDS ON