Mondragon Unibertsitatea

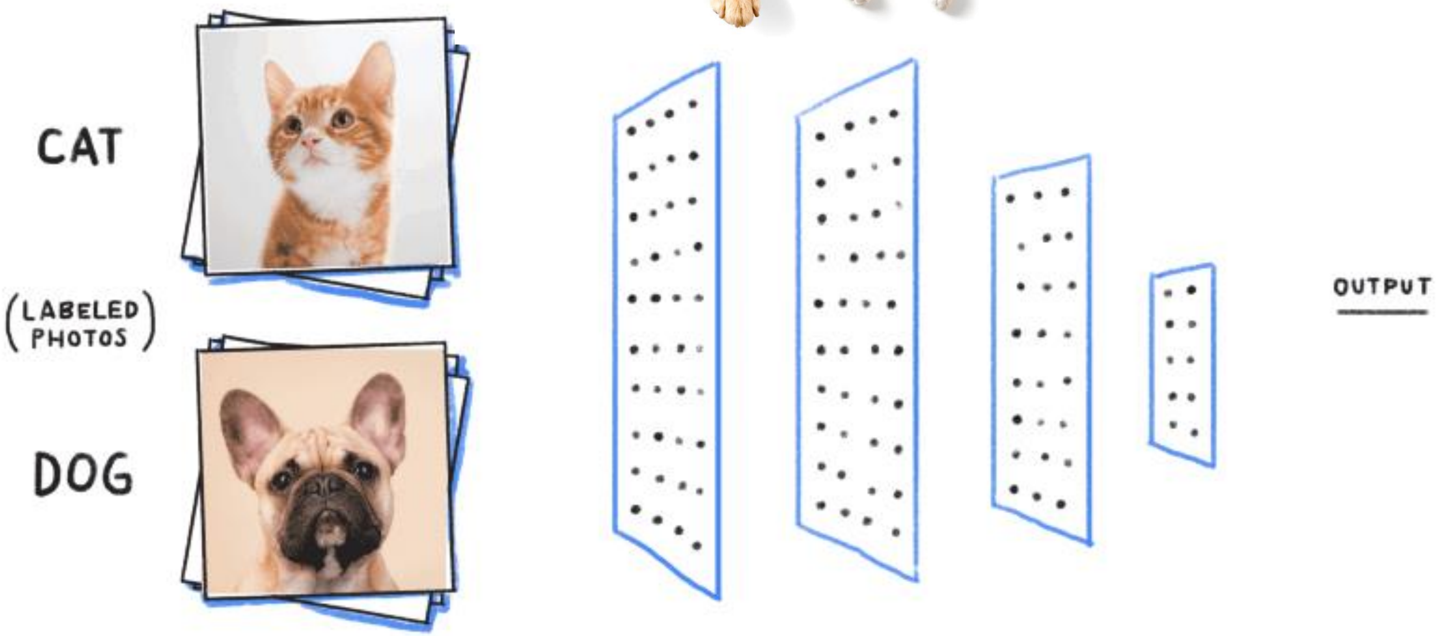
Escuela Politécnica Superior

Classification

Classification

Source: Towards Data Science

Classification



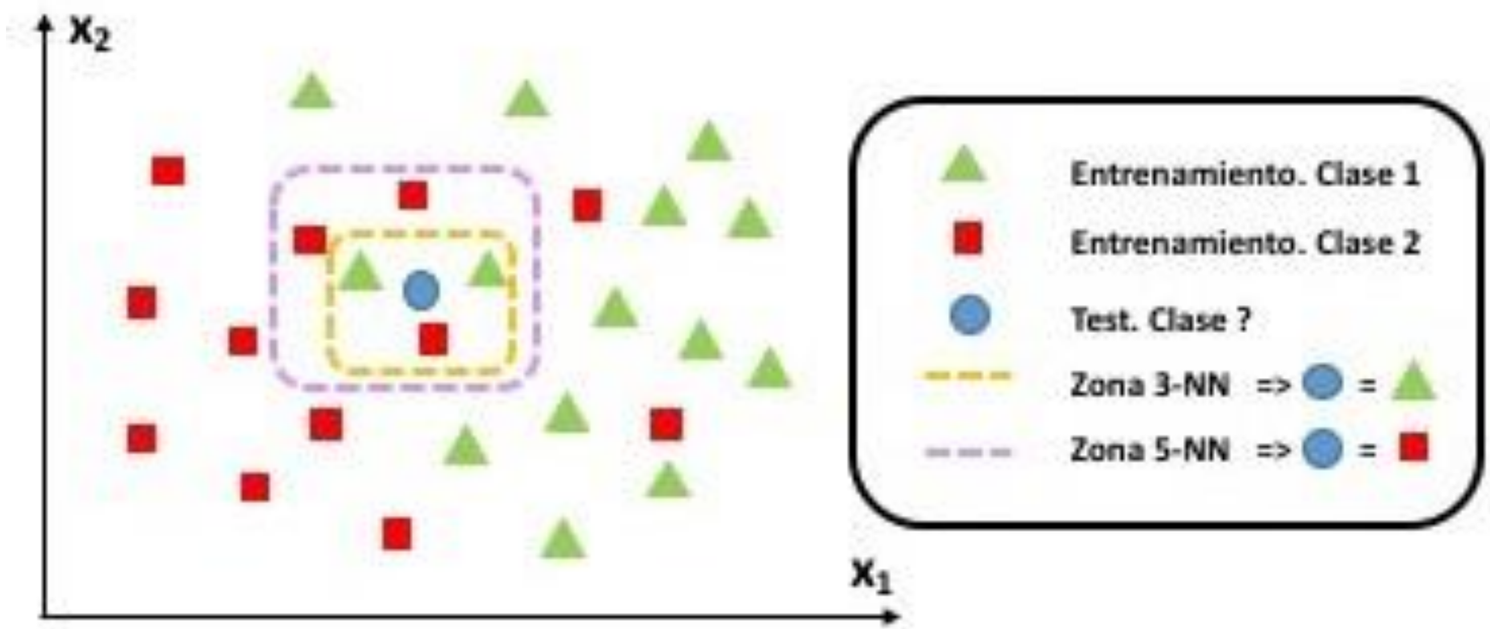
Input Data

Mathematical Model

Class Prediction

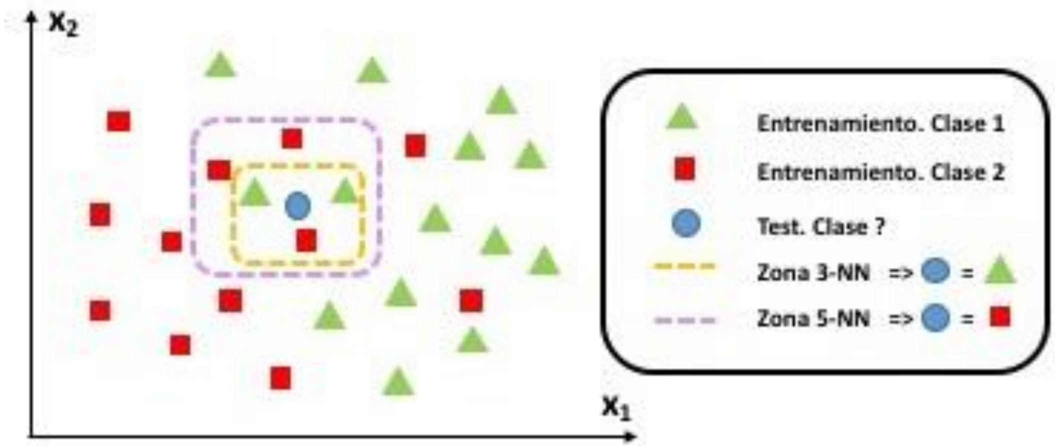
Source: Towards Data Science

K-Nearest Neighbors



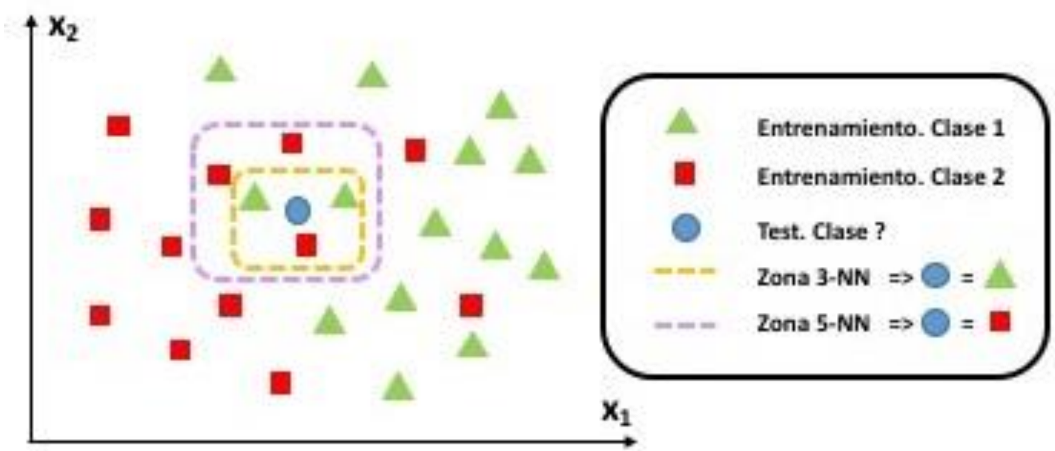
K-Nearest Neighbors

- Given a training set and a sample, for which we want to obtain a prediction



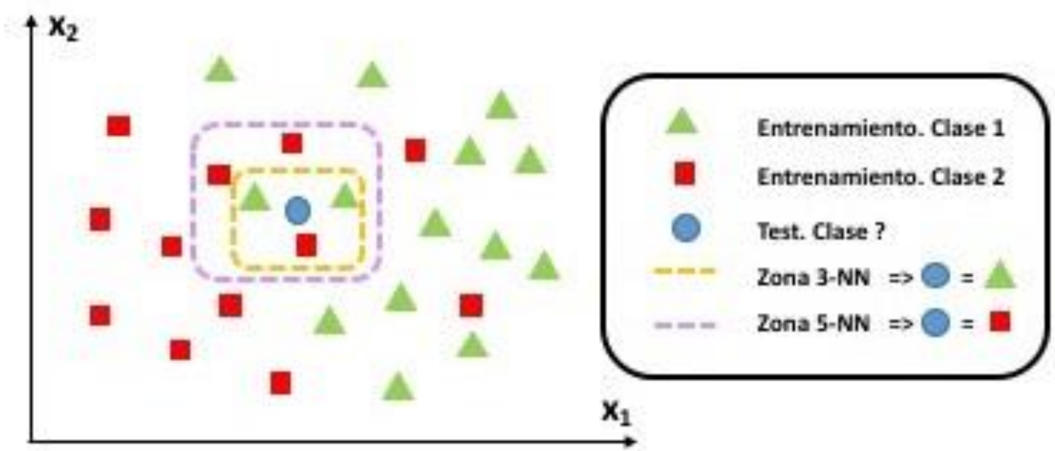
K-Nearest Neighbors

- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples



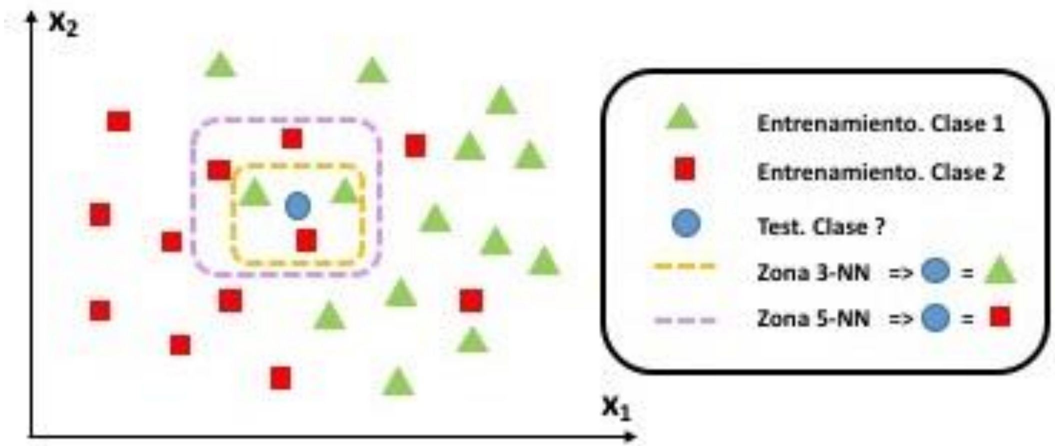
K-Nearest Neighbors

- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples
  - Select the K smaller distances



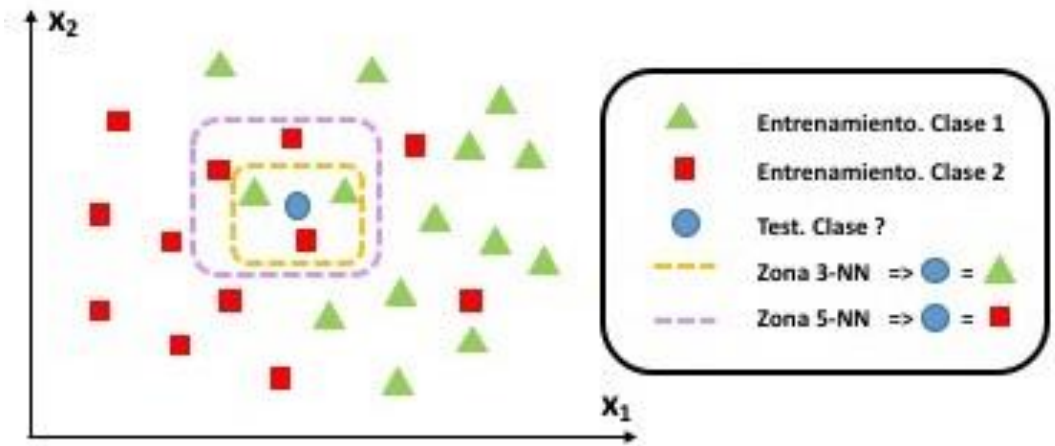
K-Nearest Neighbors

- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples
  - Select the K smaller distances
  - Predict the class for the test sample by means of majority voting among its K nearest neighbors



K-Nearest Neighbors

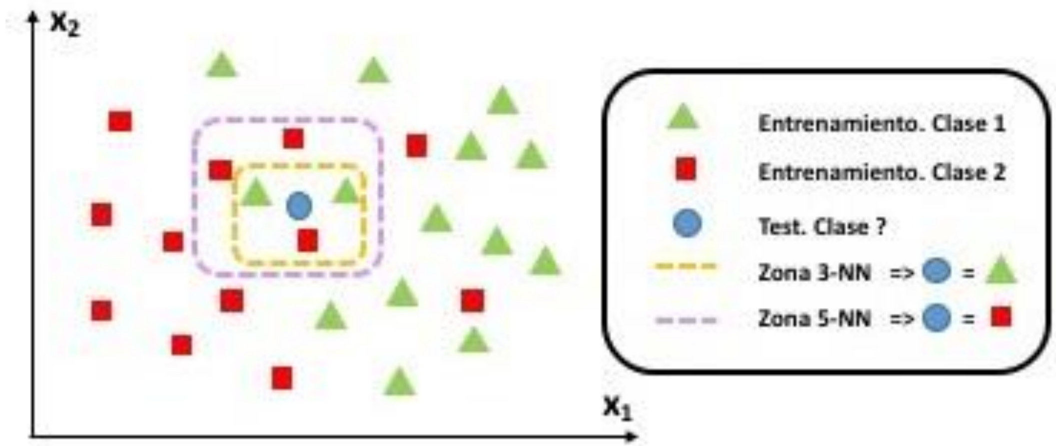
- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples
  - Select the K smaller distances
  - Predict the class for the test sample by means of majority voting among its K nearest neighbors



- Variants

K-Nearest Neighbors

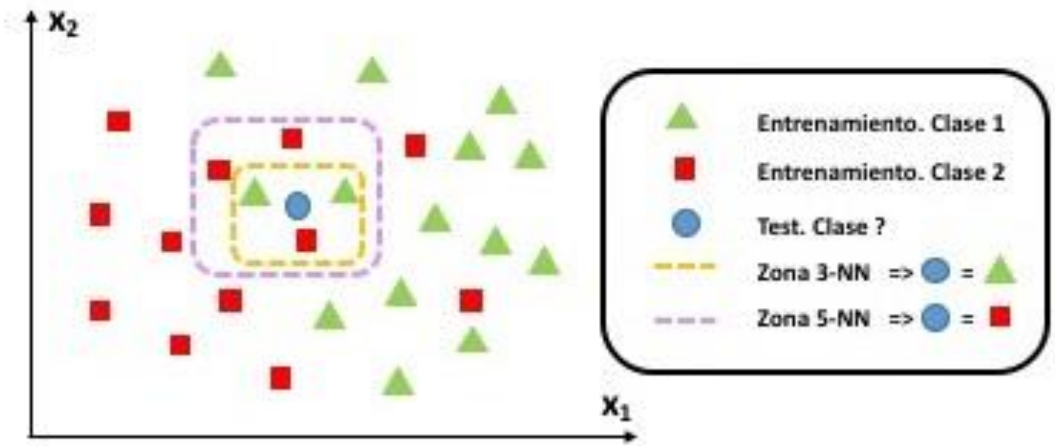
- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples
  - Select the K smaller distances
  - Predict the class for the test sample by means of majority voting among its K nearest neighbors



- Variants:
  - Use a weighted mean, which weights are inversely proportional to the distance

K-Nearest Neighbors

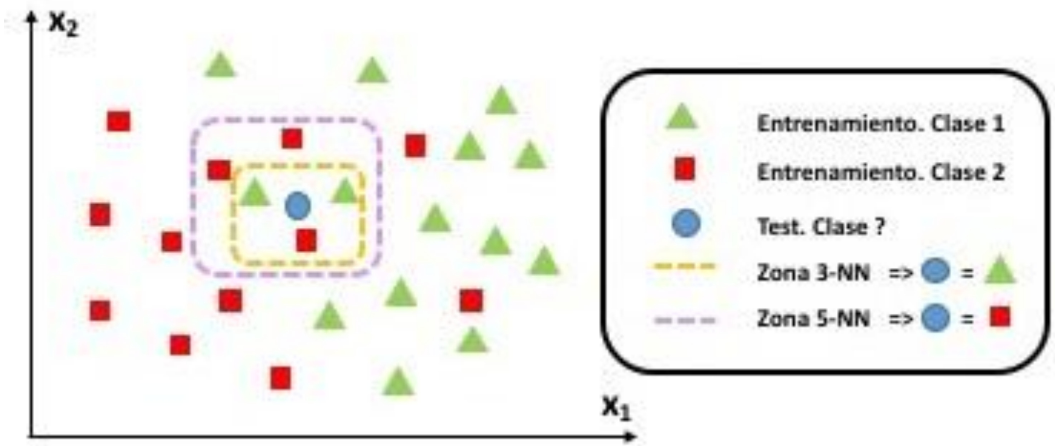
- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples
  - Select the K smaller distances
  - Predict the class for the test sample by means of majority voting among its K nearest neighbors



- Variants:
  - Use a weighted mean, which weights are inversely proportional to the distance
  - Without prefixing K, choose a threshold for defining the neighborhood (radius) and consider as neighbors all training samples that are not further than that radius

K-Nearest Neighbors

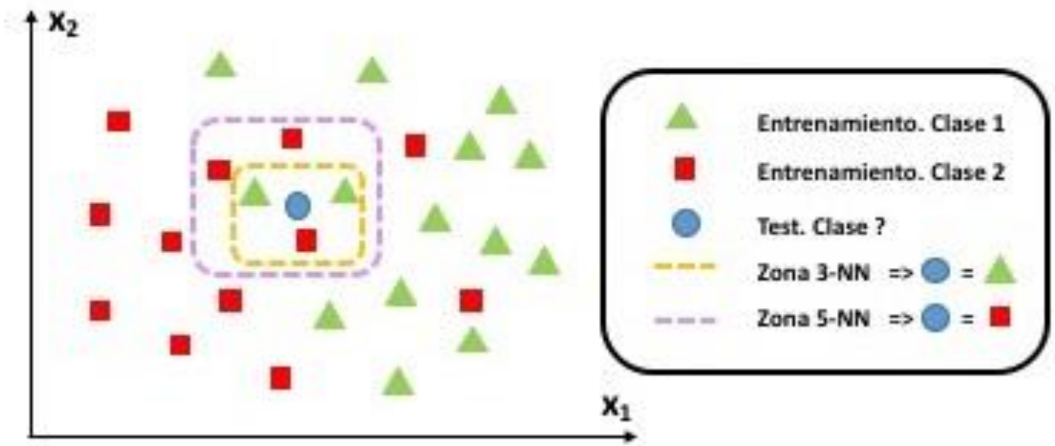
- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise “distances” between the test samples and the train samples
  - Select the K smaller distances
  - Predict the class for the test sample by means of majority voting among its K nearest neighbors



- Variants:
  - Use a weighted mean, which weights are inversely proportional to the distance
  - Without prefixing K, choose a threshold for defining the neighborhood (radius) and consider as neighbors all training samples that are not further than that radius
- **Prefixing K** is a tricky task to solve

K-Nearest Neighbors

- Given a training set and a sample, for which we want to obtain a prediction:
  - Calculate all pairwise "distances" between the test samples and the train samples
  - Select the K smaller distances
  - Predict the class for the test sample by means of majority voting among its K nearest neighbors



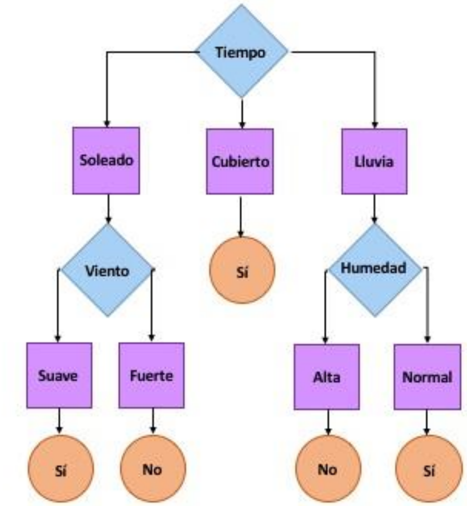
- Variants:
  - Use a weighted mean, which weights are inversely proportional to the distance
  - Without prefixing K, choose a threshold for defining the neighborhood (radius) and consider as neighbors all training samples that are not further than that radius
- **Prefixing K** is a tricky task to solve
- We must design an strategy for **breaking ties**

Decision trees

Mondragon Unibertsitatea
Escuela Politécnica Superior

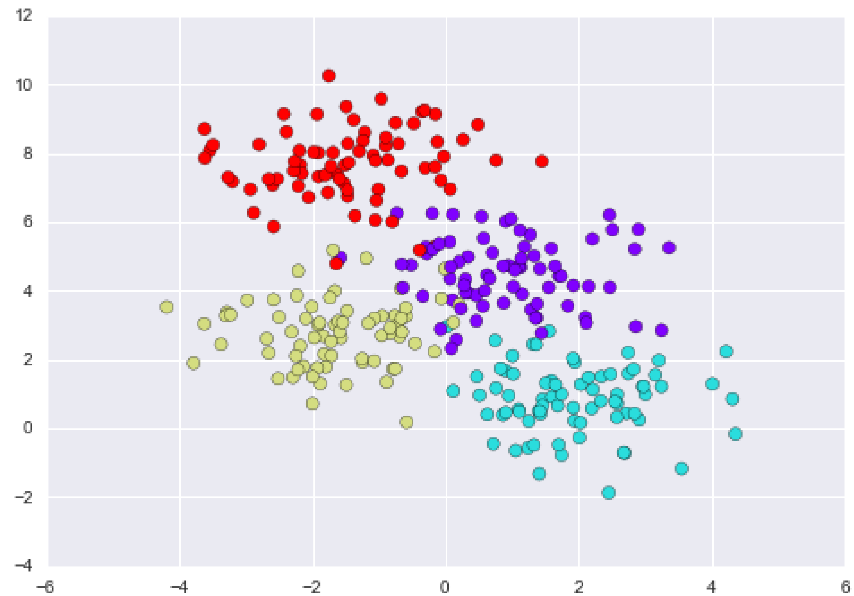
- Decision Trees (C4.5)
  - They can handle both continuous and descrete variables
  - They are highly interpretable if the number of variables is not much high
  - They can suffer from *overfitting* (adapt too much to the training data loosing extrapolation power to unseen data)

Tiempo	Temp.	Humedad	Viento	Jugar golf
Lluvia	Calor	Alta	Suave	No
Lluvia	Calor	Alta	Fuerte	No
Cubierto	Calor	Alta	Suave	Sí
Soleado	Templado	Alta	Suave	Sí
Soleado	Frío	Normal	Suave	Sí
Soleado	Frío	Normal	Fuerte	No
Cubierto	Frío	Normal	Fuerte	Sí
Lluvia	Templado	Alta	Suave	No
Lluvia	Frío	Normal	Suave	Sí
Soleado	Templado	Normal	Suave	Sí
Lluvia	Templado	Normal	Fuerte	Sí
Cubierto	Templado	Alta	Fuerte	Sí
Cubierto	Calor	Normal	Suave	Sí
Soleado	Templado	Alta	Fuerte	No



Decision trees

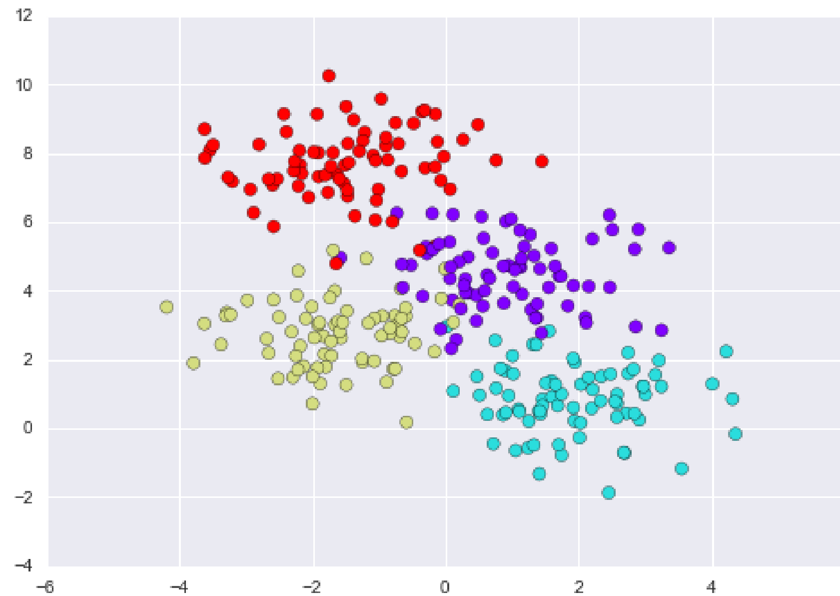
- Decision Trees (C4.5)
  - They can handle both continuous and descrete variables
  - They are highly interpretable if the number of variables is not much high
  - They can suffer from *overfitting* (adapt too much to the training data loosing extrapolation power to unseen data)

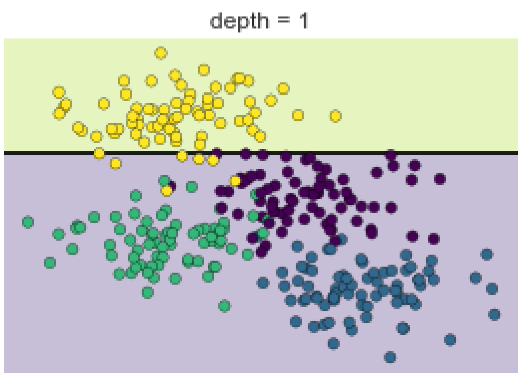


Decision trees

Mondragon Unibertsitatea
Escuela Politécnica Superior

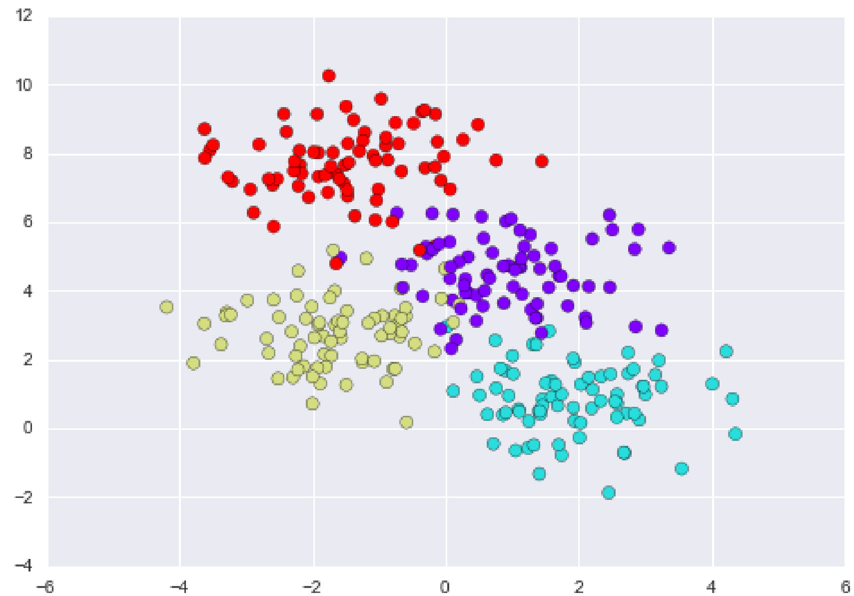
- Decision Trees (C4.5)
  - They can handle both continuous and descrete variables
  - They are highly interpretable if the number of variables is not much high
  - They can suffer from *overfitting* (adapt too much to the training data loosing extrapolation power to unseen data)

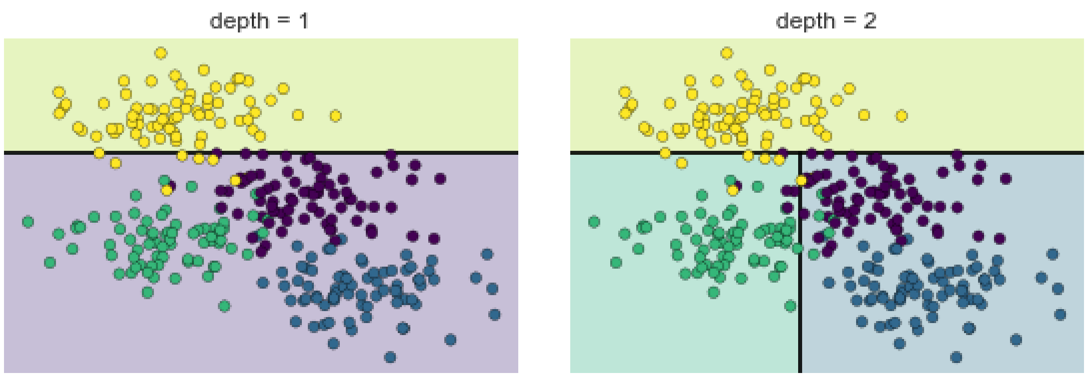




Decision trees

- Decision Trees (C4.5)
  - They can handle both continuous and descrete variables
  - They are highly interpretable if the number of variables is not much high
  - They can suffer from *overfitting* (adapt too much to the training data loosing extrapolation power to unseen data)

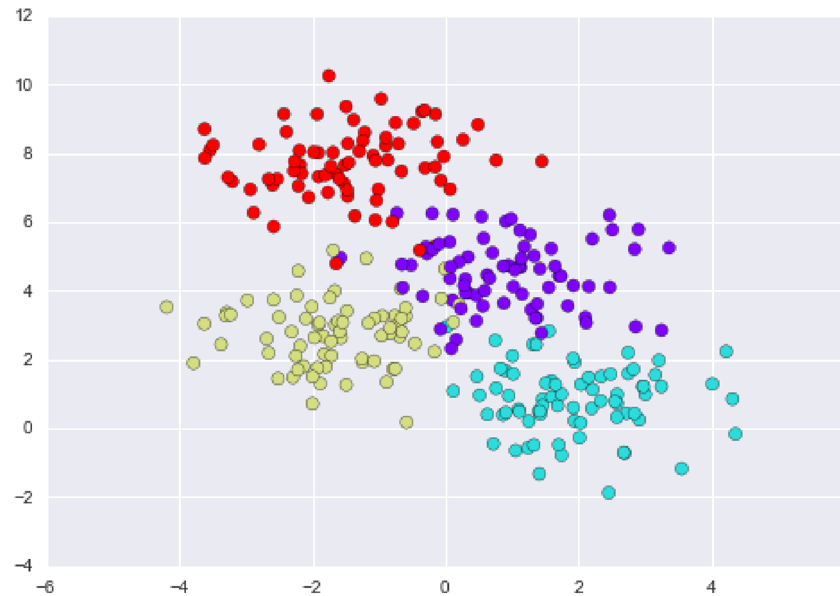


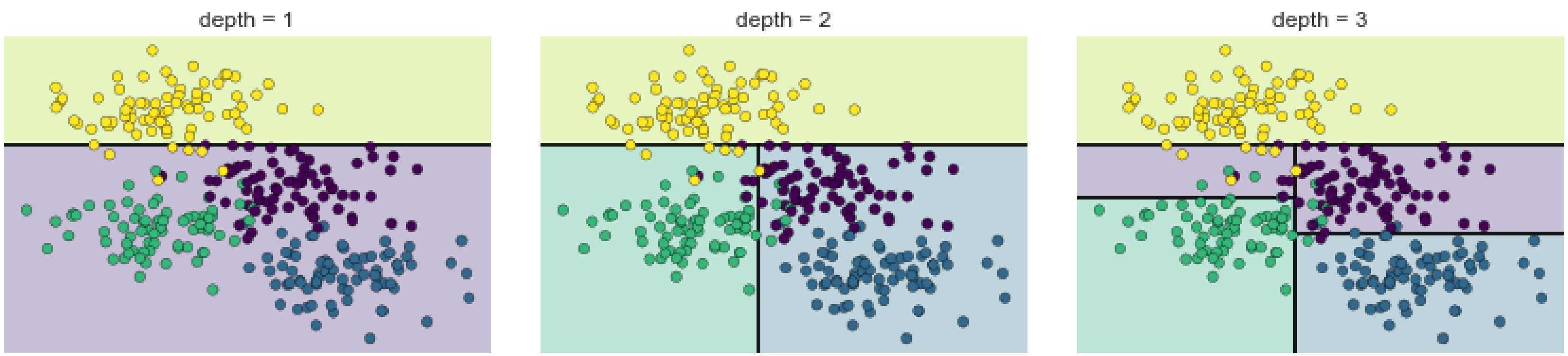


Decision trees

Mondragon Unibertsitatea
Escuela Politécnica Superior

- Decision Trees (C4.5)
  - They can handle both continuous and descrete variables
  - They are highly interpretable if the number of variables is not much high
  - They can suffer from *overfitting* (adapt too much to the training data loosing extrapolation power to unseen data)

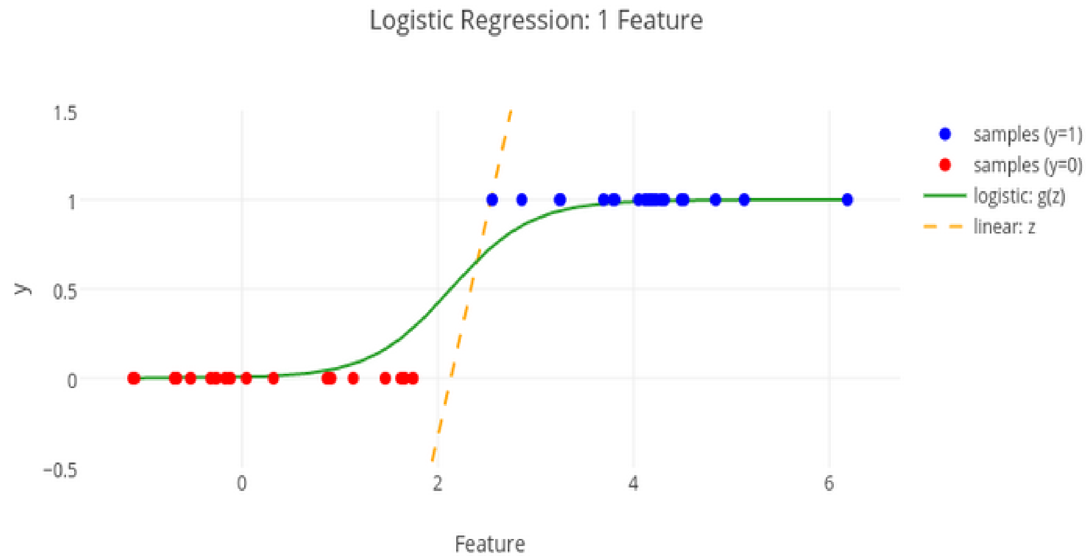




Logistic regression

- Logit function

$$p(X) = \frac{e^{(\beta_0+\beta_1 X)}}{1+e^{(\beta_0+\beta_1 X)}}$$



Logistic regression

- Logit function

$$p(X) = \frac{e^{(\beta_0+\beta_1 X)}}{1 + e^{(\beta_0+\beta_1 X)}}$$

- Horizontal asymptotes
  - y = 0, towards the left
  - y = 1, towards the right

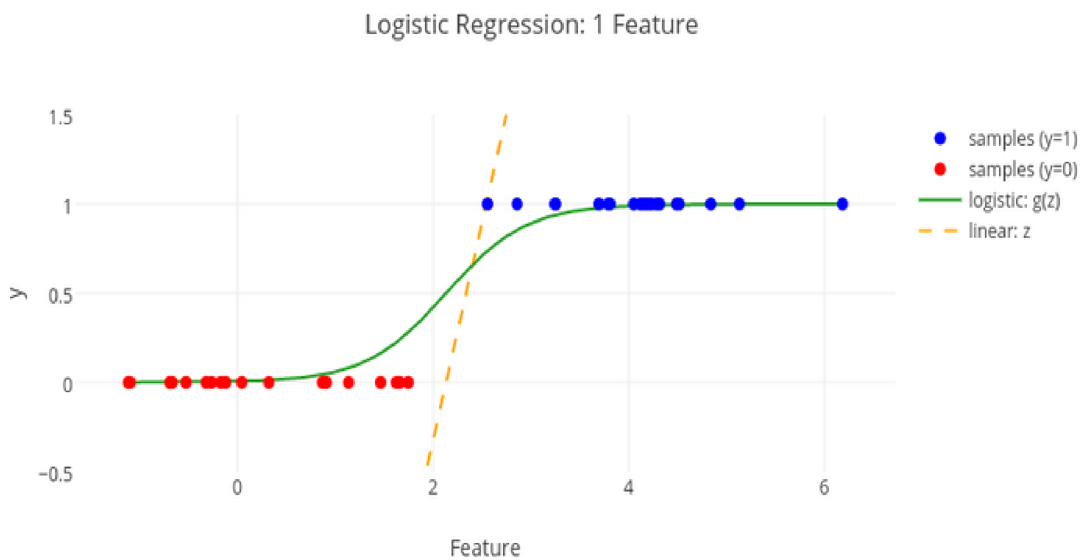
Logistic regression

- Logit function

$$p(X) = \frac{e^{(\beta_0+\beta_1 X)}}{1+e^{(\beta_0+\beta_1 X)}}$$

- Horizontal asymptotes
  - y = 0, towards the left
  - y = 1, towards the right

- It estimates classes probabilities



Logistic regression

- Logit function

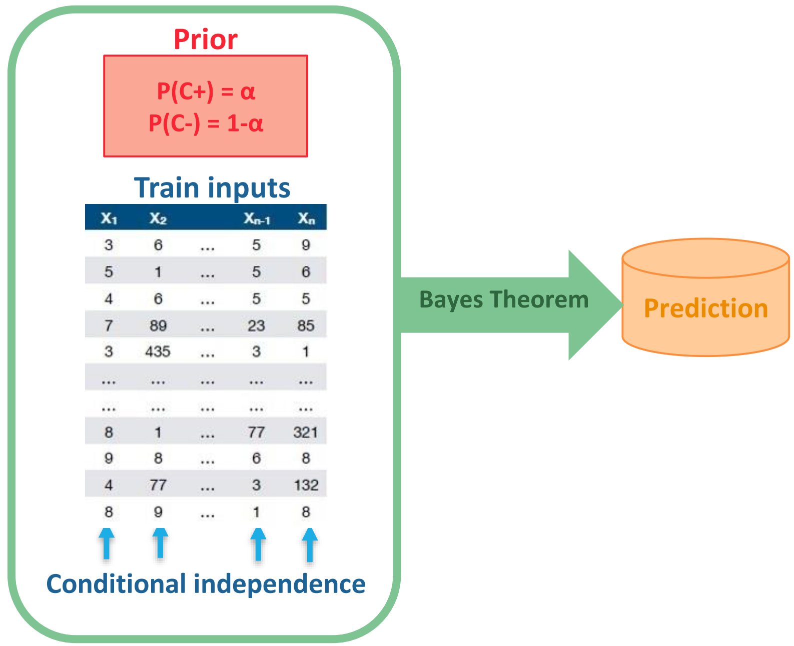
$$p(X) = \frac{e^{(\beta_0+\beta_1 X)}}{1+e^{(\beta_0+\beta_1 X)}}$$

- Horizontal asymptotes
  - y = 0, towards the left
  - y = 1, towards the right

- It estimates classes probabilities

- It is not recommendable for multiclass classification, i.e. not binary

Naïve Bayes

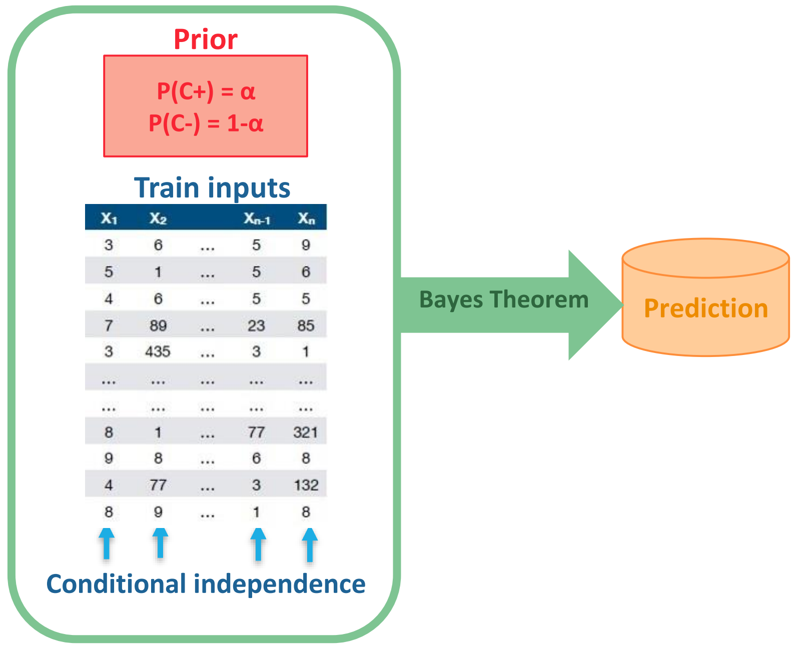


X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

We need to calculate

$$p(C|X_1,\ldots,X_n)$$



Naïve Bayes

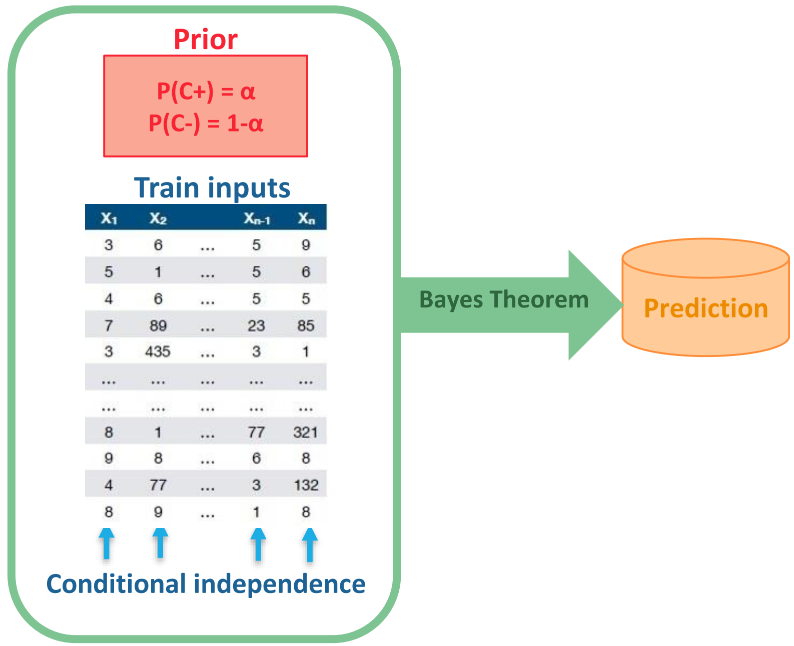
Mondragon Unibertsitatea
Escuela Politécnica Superior

We need to calculate

$$p(C|X_1,\ldots,X_n)$$

Bayes Theorem,

$$p(C|X_1,\ldots,X_n) = \frac{p(C)\cdot p(X_1,\ldots,X_n|C)}{p(X_1,\ldots,X_n)}$$



Naïve Bayes

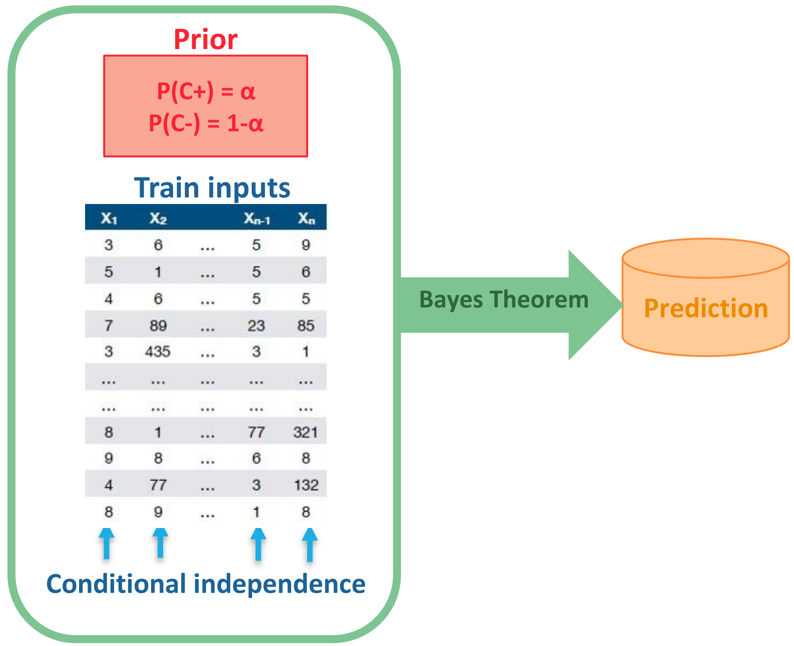
We need to calculate

$$p(C|X_1,\dots,X_n)$$

Joint probability $p(C,X_1,\dots,X_n)$

Bayes Theorem,

$$p(C|X_1,\dots,X_n) = \frac{p(C)\cdot p(X_1,\dots,X_n|C)}{p(X_1,\dots,X_n)}$$



X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

We need to calculate

$$p(C|X_1,\dots,X_n)$$

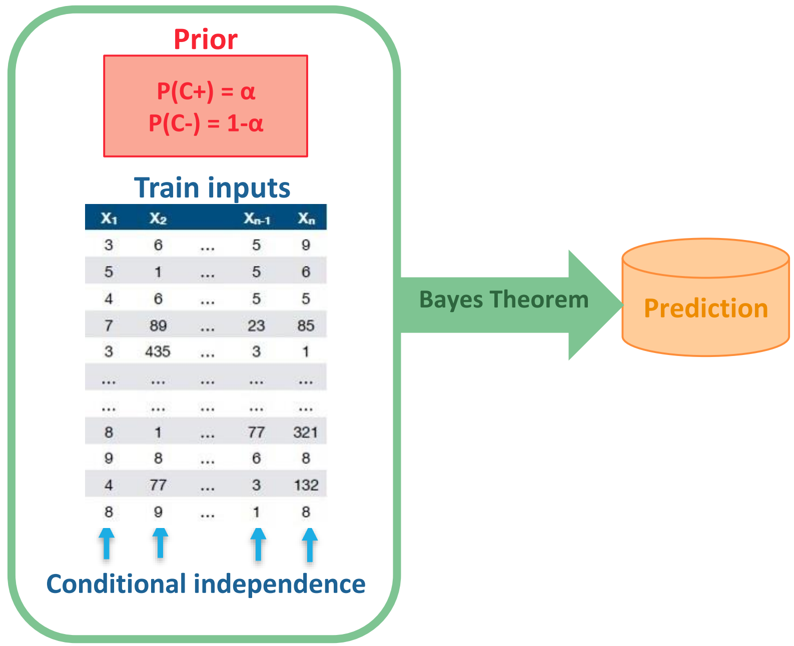
Joint probability $p(C,X_1,\dots,X_n)$

Bayes Theorem,

$$p(C|X_1,\dots,X_n)=\frac{p(C)\cdot p(X_1,\dots,X_n|C)}{p(X_1,\dots,X_n)}$$

Applying the chain rule in conditional probability

$$p(C,X_1,\dots,X_n)=p(C)\cdot p(X_1|C)\cdot p(X_2|C,X_1)\cdot\ldots\cdot p(X_n|C,X_1,\dots,X_{n-1})$$



X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

Mondragon Unibertsitatea
Escuela Politécnica Superior

We need to calculate

$$p(C|X_1,\ldots,X_n)$$

Joint probability $p(C,X_1,\ldots,X_n)$

Bayes Theorem,

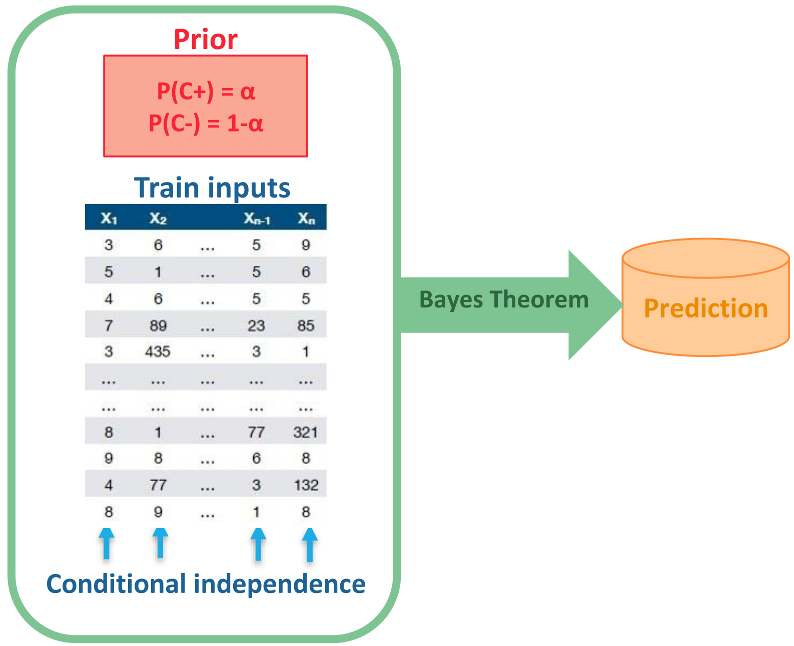
$$p(C|X_1,\ldots,X_n)=\frac{p(C)\cdot p(X_1,\ldots,X_n|C)}{p(X_1,\ldots,X_n)}$$

Applying the chain rule in conditional probability

$$p(C,X_1,\ldots,X_n)=p(C)\cdot p(X_1|C)\cdot p(X_2|C,X_1)\cdot\ldots\cdot p(X_n|C,X_1,\ldots,X_{n-1})$$

Assuming conditional independence between X_i, then if $i \neq j$

$$p(X_i|C,X_j)=p(X_i|C)$$



X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

We need to calculate

$$p(C|X_1,\ldots,X_n)$$

Bayes Theorem,

Joint probability $p(C,X_1,\ldots,X_n)$

$$p(C|X_1,\ldots,X_n) = \frac{p(C)\cdot p(X_1,\ldots,X_n|C)}{p(X_1,\ldots,X_n)}$$

Applying the chain rule in conditional probability

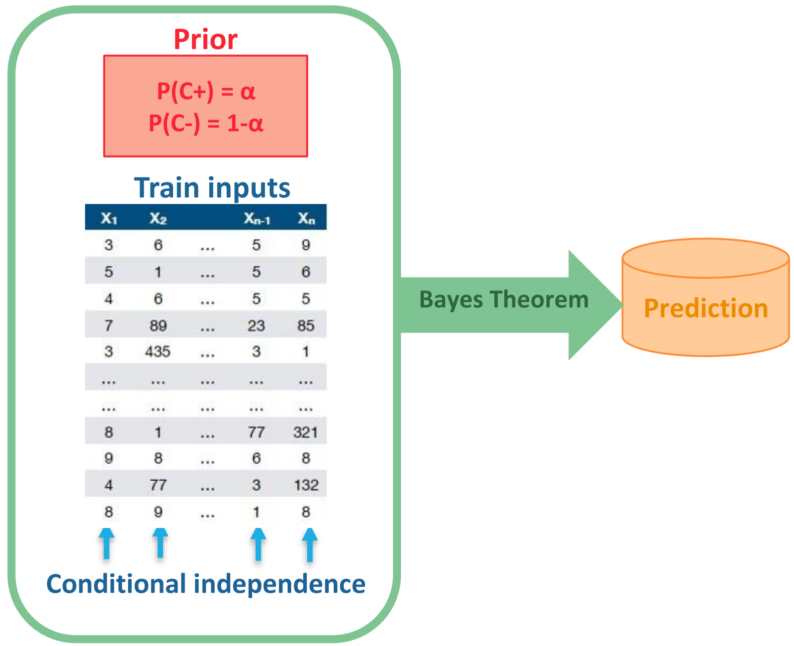
$$p(C,X_1,\ldots,X_n) = p(C)\cdot p(X_1|C)\cdot p(X_2|C,X_1)\cdot\ldots\cdot p(X_n|C,X_1,\ldots,X_{n-1})$$

Assuming conditional independence between X_i, then if $i \neq j$

$$p(X_i|C,X_j) = p(X_i|C)$$

Therefore,

$$p(C,X_1,\ldots,X_n) = p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$



X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

Mondragon Unibertsitatea
Escuela Politécnica Superior

We need to calculate

$$p(C|X_1,\ldots,X_n) \qquad \text{Joint probability} \qquad p(C,X_1,\ldots,X_n)$$

Bayes Theorem,

$$p(C|X_1,\ldots,X_n) = \frac{p(C)\cdot p(X_1,\ldots,X_n|C)}{p(X_1,\ldots,X_n)}$$

Applying the chain rule in conditional probability

$$p(C,X_1,\ldots,X_n) = p(C)\cdot p(X_1|C)\cdot p(X_2|C,X_1)\cdot\ldots\cdot p(X_n|C,X_1,\ldots,X_{n-1})$$

Assuming conditional independence between X_i, then if $i \neq j$

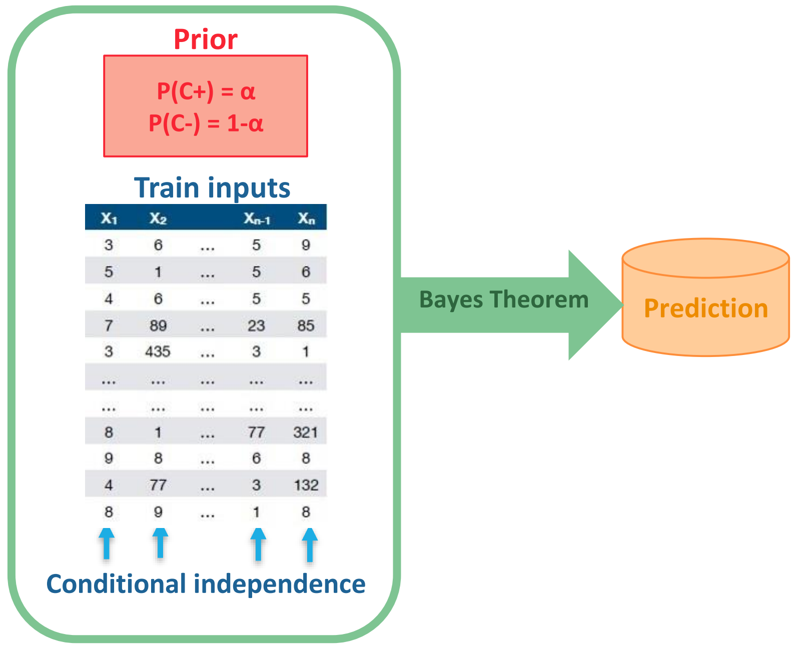
$$p(X_i|C,X_j) = p(X_i|C)$$

Therefore,

$$p(C,X_1,\ldots,X_n) = p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$

Then,

$$p(C|X_1,\ldots,X_n) \propto p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$



X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

We need to calculate

$$p(C|X_1,...,X_n) \qquad p(C,X_1,...,X_n)$$

Joint probability

Bayes Theorem,

$$p(C|X_1,...,X_n) = \frac{p(C)\cdot p(X_1,...,X_n|C)}{p(X_1,...,X_n)}$$

Applying the chain rule in conditional probability

$$p(C,X_1,...,X_n) = p(C)\cdot p(X_1|C)\cdot p(X_2|C,X_1)\cdot ...\cdot p(X_n|C,X_1,...,X_{n-1})$$

Assuming conditional independence between X_i, then if $i \neq j$

$$p(X_i|C,X_j) = p(X_i|C)$$

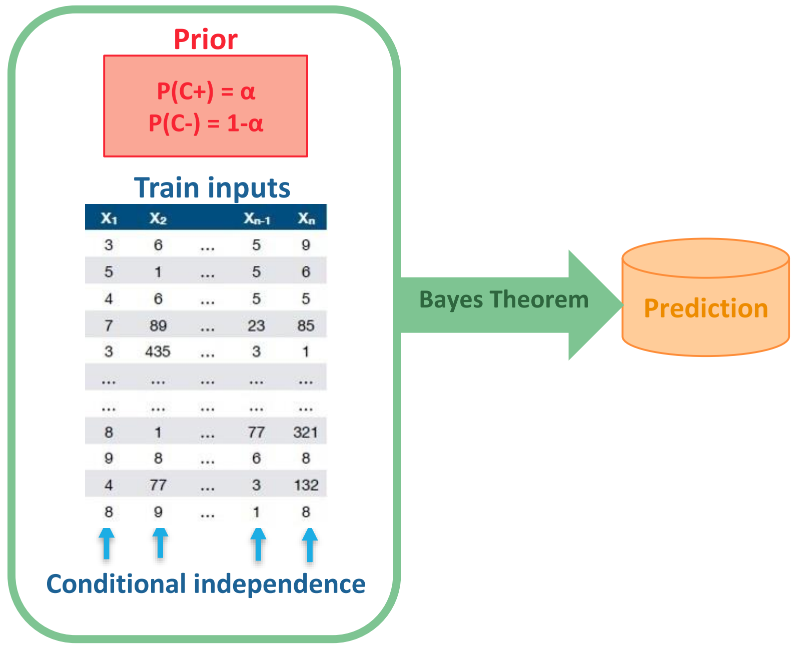
Therefore,

$$p(C,X_1,...,X_n) = p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$

Then,

$$p(C|X_1,...,X_n) \propto p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$

where the proportionality constant is $\dfrac{1}{p(X_1,...,X_n)}$



X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Naïve Bayes

We need to calculate

$$p(C|X_1,\ldots,X_n) \qquad\qquad p(C,X_1,\ldots,X_n)$$

Joint probability

Bayes Theorem,

$$p(C|X_1,\ldots,X_n) = \frac{p(C)\cdot p(X_1,\ldots,X_n|C)}{p(X_1,\ldots,X_n)}$$

Applying the chain rule in conditional probability

$$p(C,X_1,\ldots,X_n) = p(C)\cdot p(X_1|C)\cdot p(X_2|C,X_1)\cdot\ldots\cdot p(X_n|C,X_1,\ldots,X_{n-1})$$

Assuming conditional independence between X_i, then if $i \neq j$

$$p(X_i|C,X_j) = p(X_i|C)$$

Therefore,

$$p(C,X_1,\ldots,X_n) = p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$

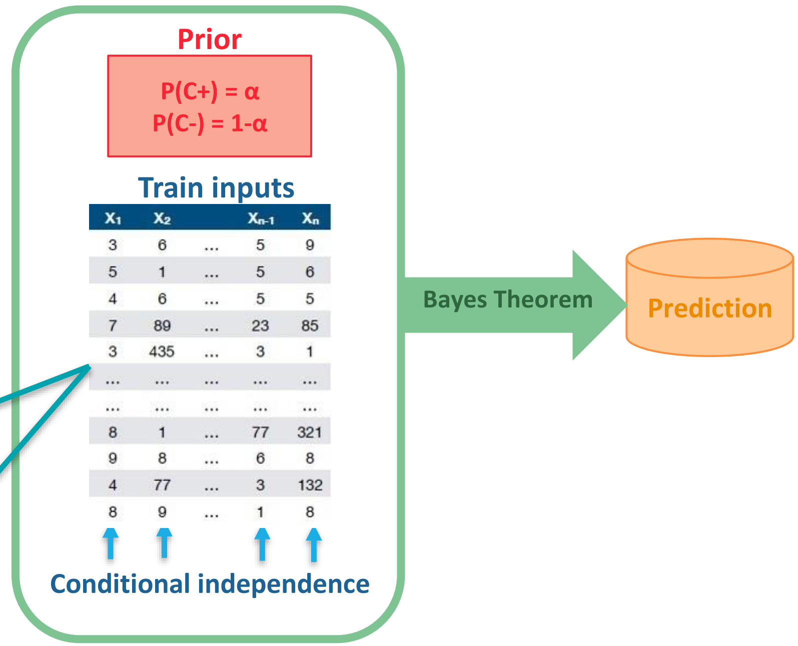
Then,

$$p(C|X_1,\ldots,X_n) \propto p(C)\cdot\prod_{i=1}^{n} p(X_i|C)$$

Sample data

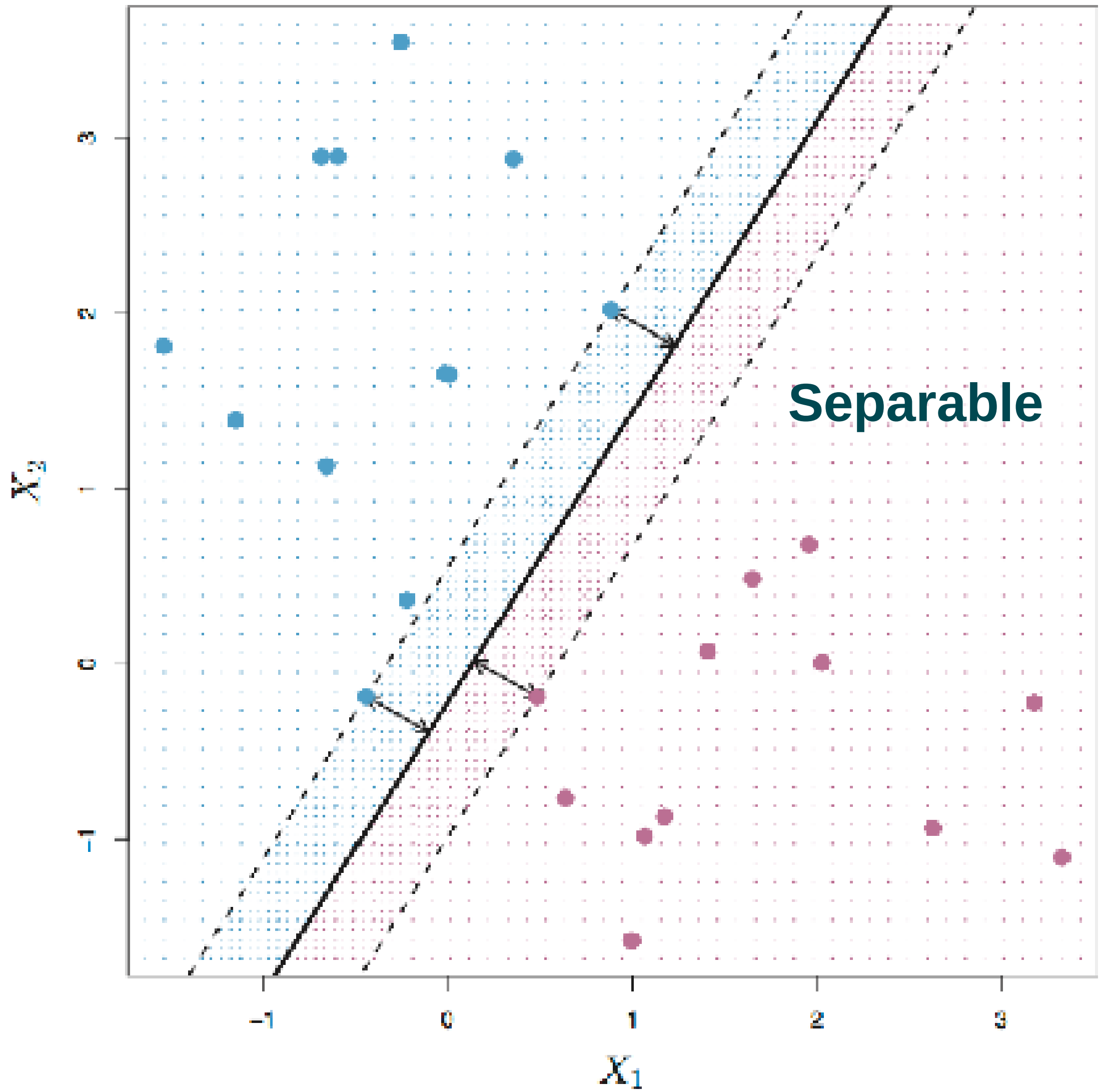
Sample data

where the proportionality constant is $\dfrac{1}{p(X_1,\ldots,X_n)}$

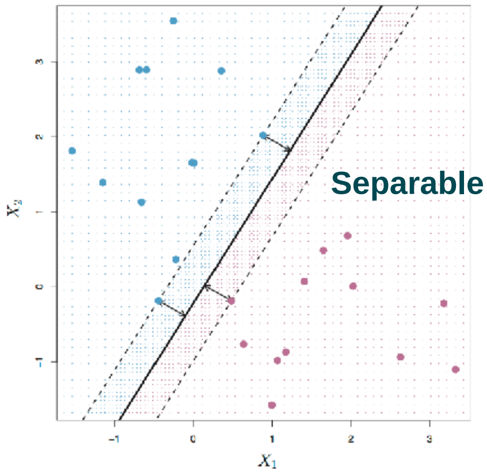


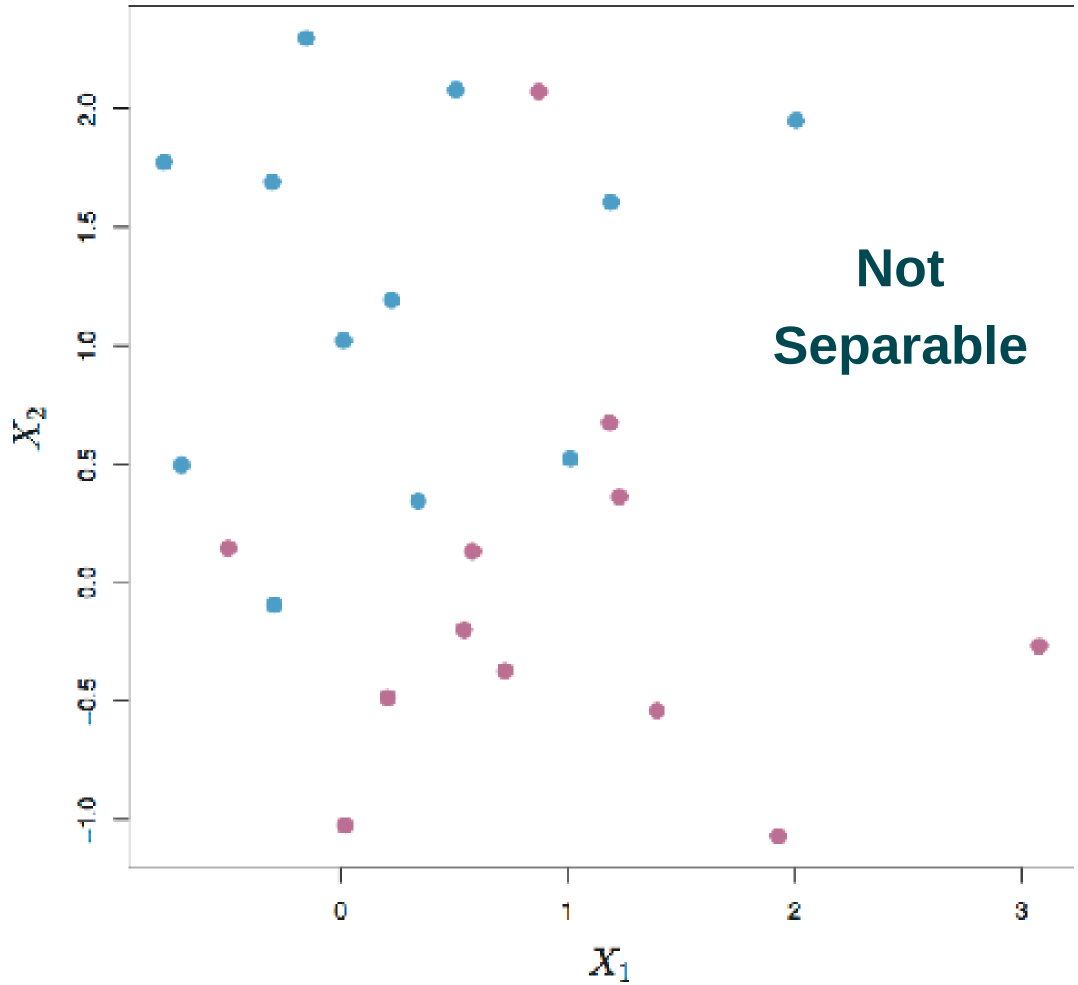
X_1	X_2		X_{n-1}	X_n
3	6	...	5	9
5	1	...	5	6
4	6	...	5	5
7	89	...	23	85
3	435	...	3	1
...	...	...	...	...
...	...	...	...	...
8	1	...	77	321
9	8	...	6	8
4	77	...	3	132
8	9	...	1	8

Support Vector Machines



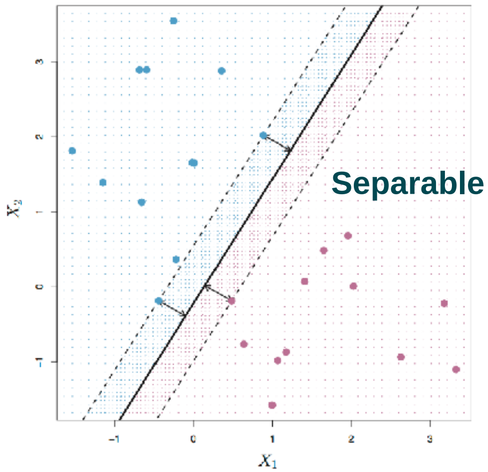
Support Vector Machines

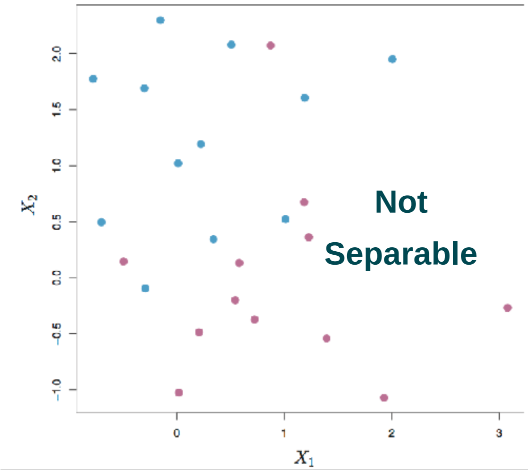


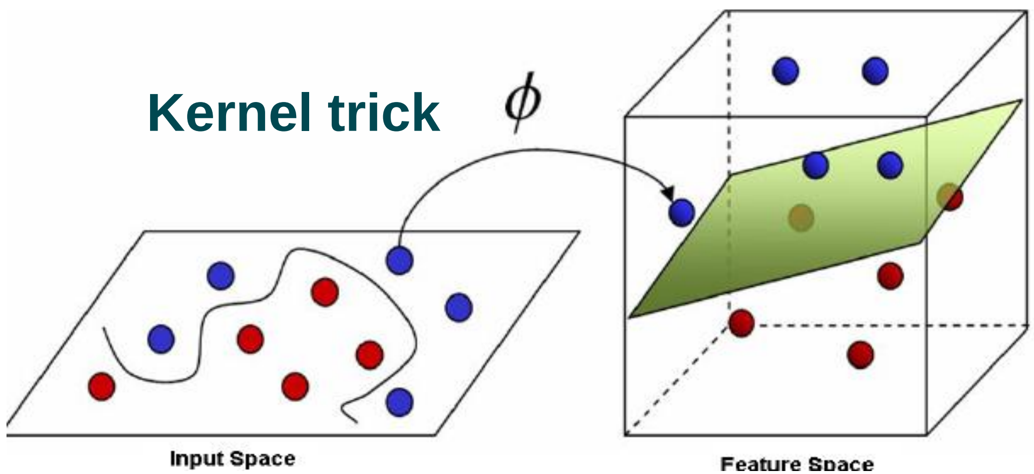


Support Vector Machines

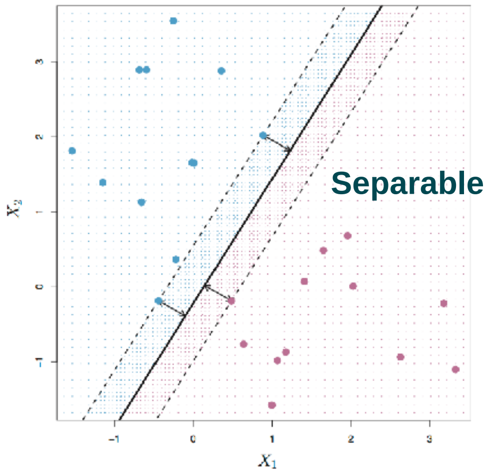
Mondragon Unibertsitatea
Escuela Politécnica Superior

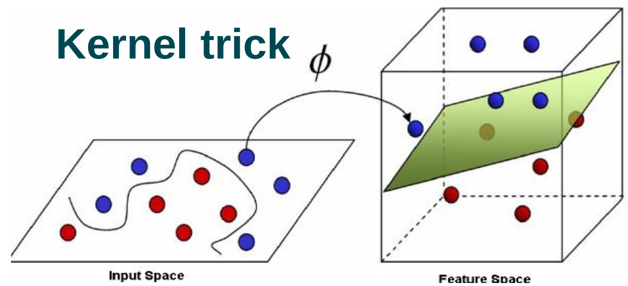


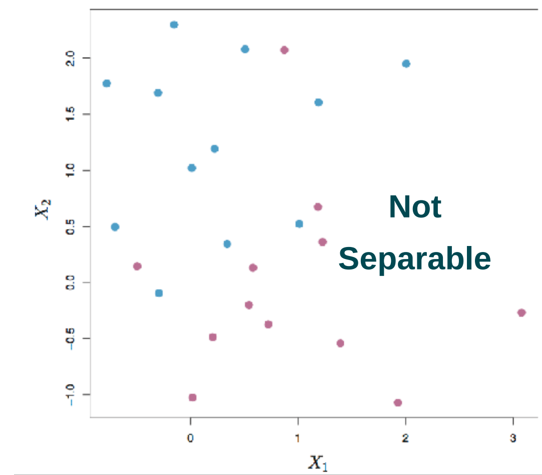


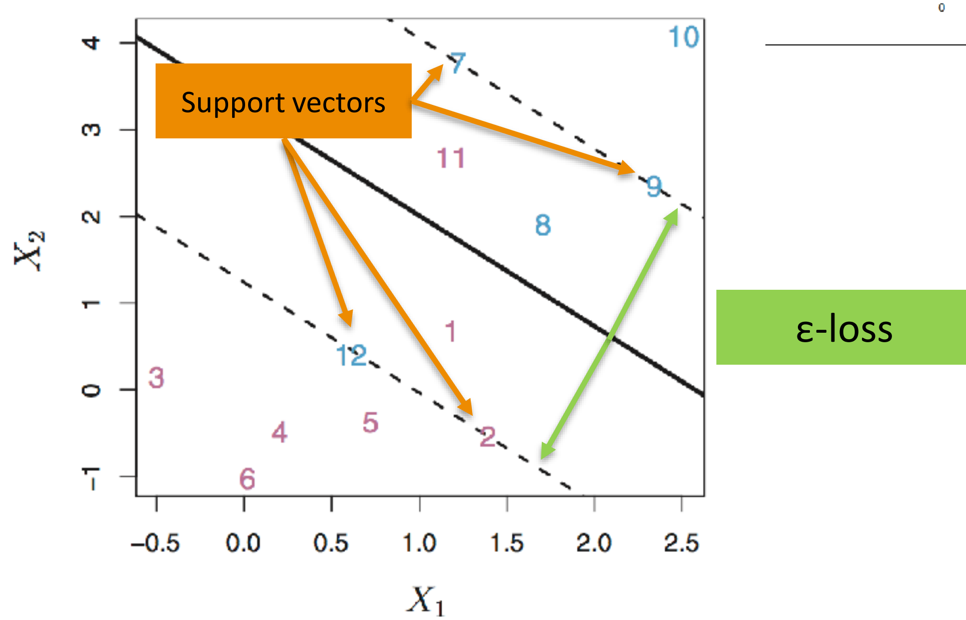


Support Vector Machines



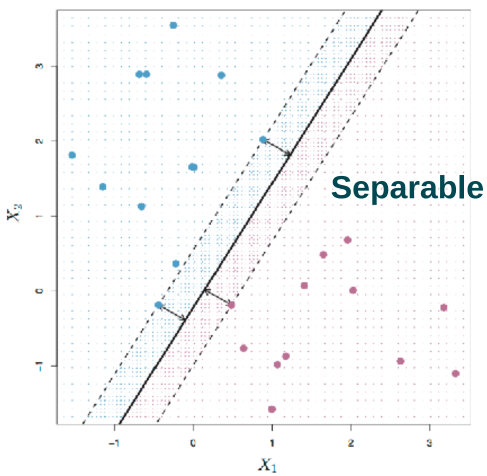


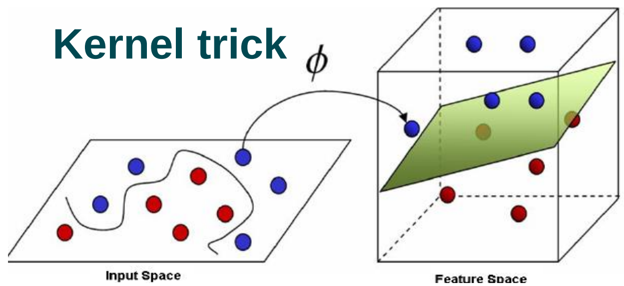


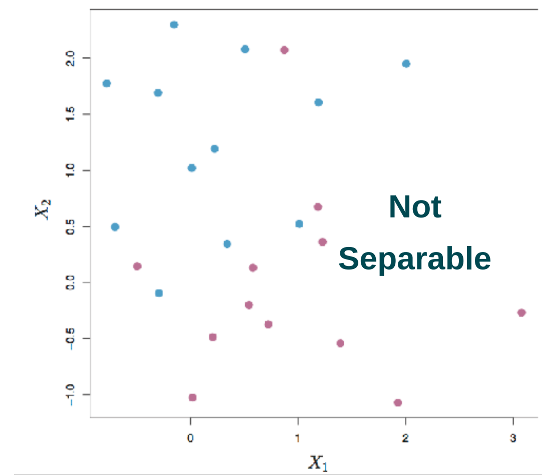


Support Vector Machines

Mondragon Unibertsitatea
Escuela Politécnica Superior





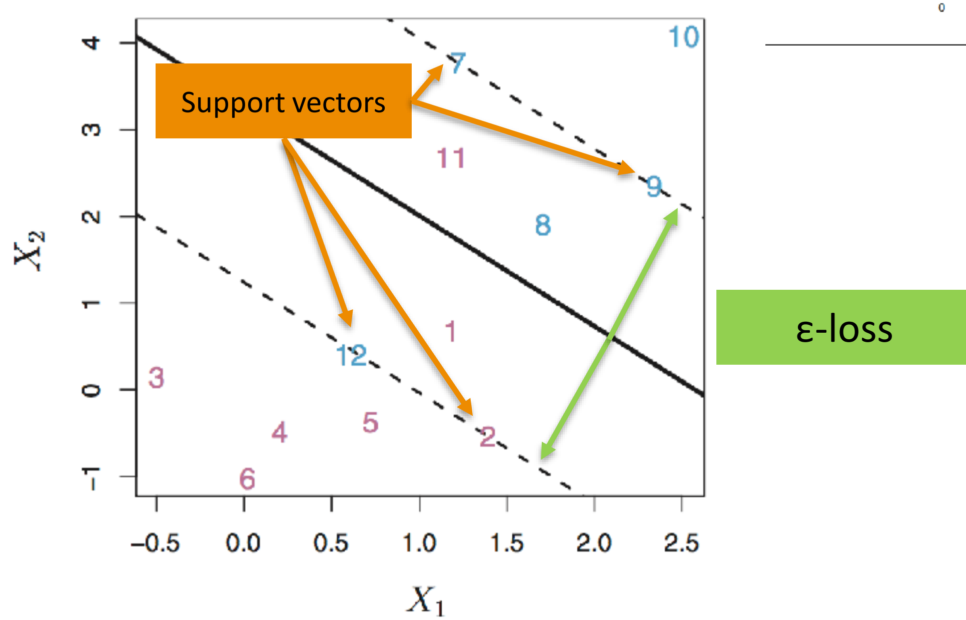


$$\arg \max_{\beta_0,\beta_1,\ldots,\beta_n} M$$

$$\text{subject to} \sum_{j=1}^{n} \beta_j^2 = 1,$$

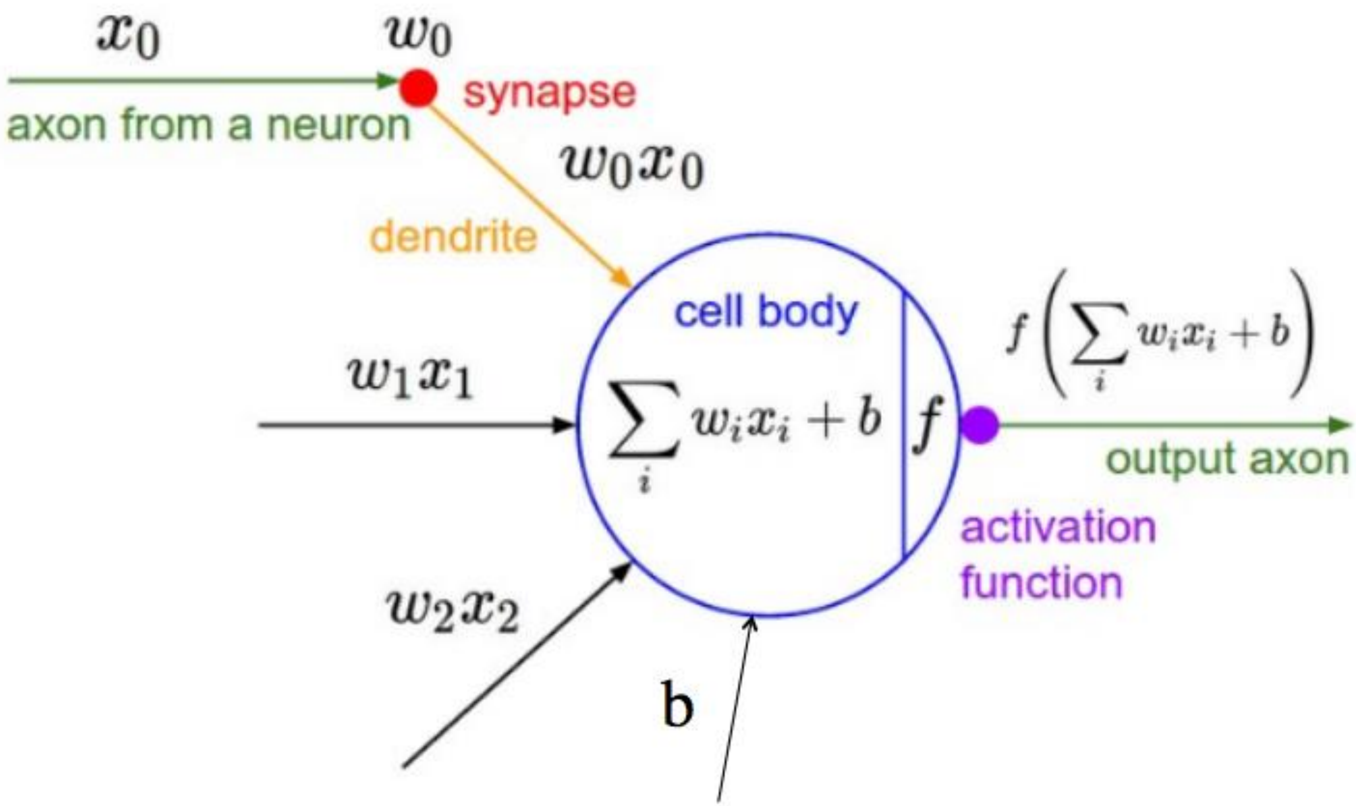
$$y_i * (\beta_0 + \sum_j X_{ij}\beta_{ij}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \sum_i \epsilon_i \leq C$$



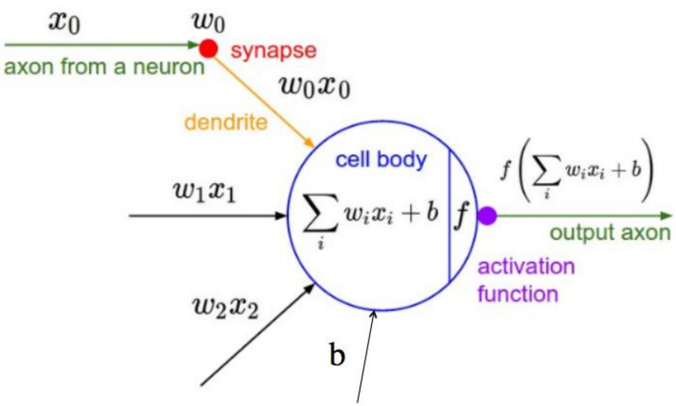
Neural networks

- Neuron

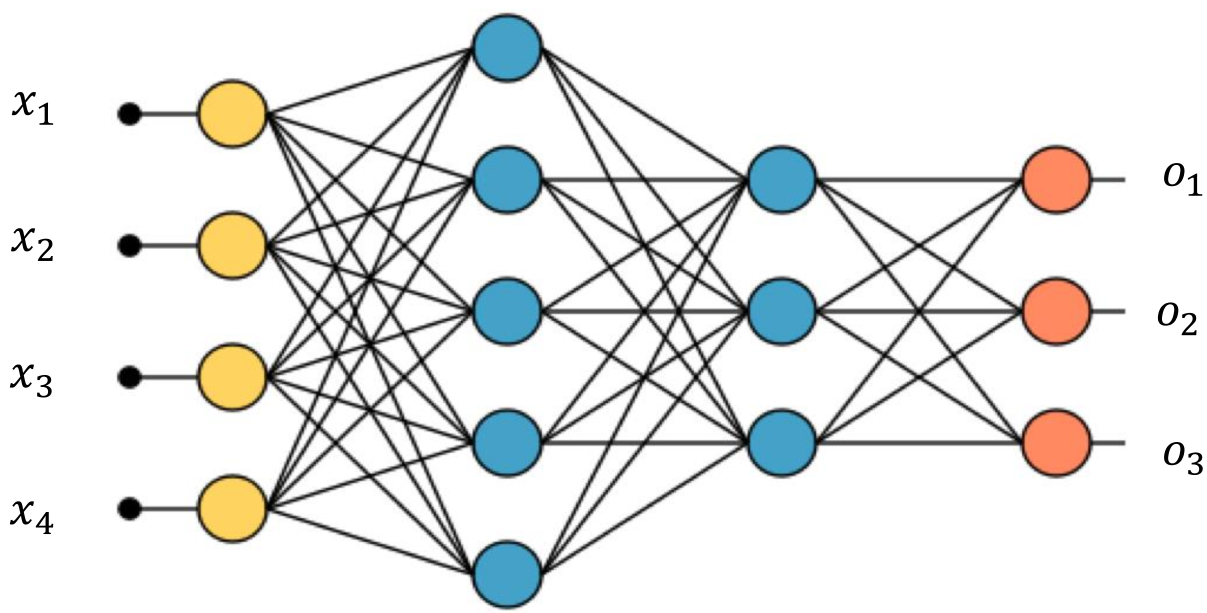


Neural networks

- Neuron

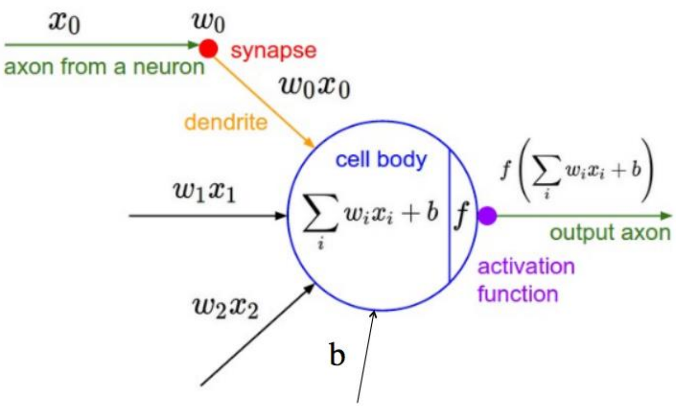


- Neural Network

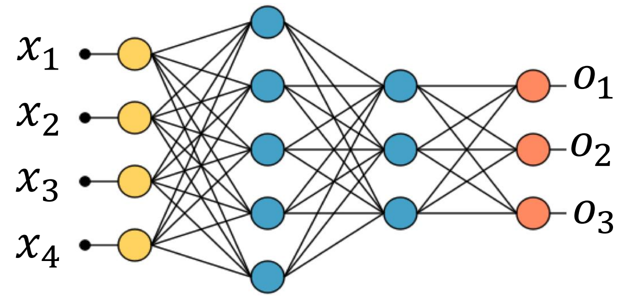


Neural networks

- Neuron



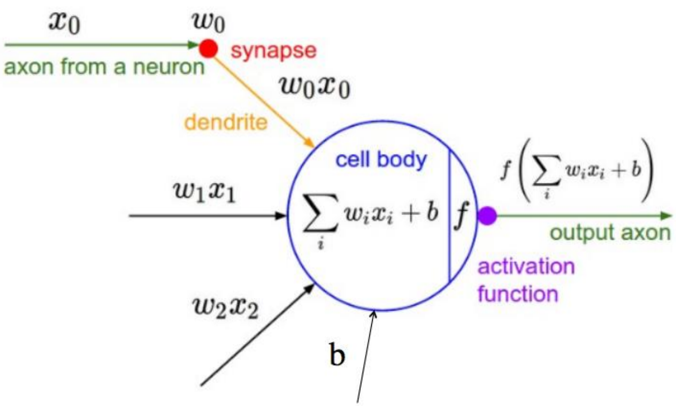
- Neural Network



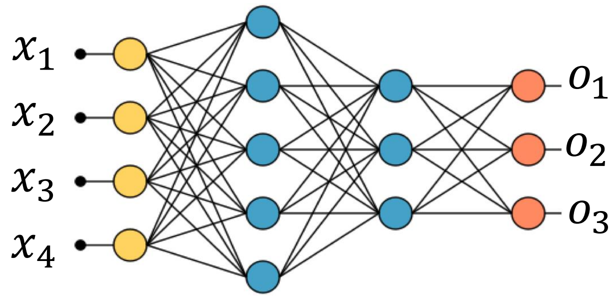
The union, in reticular form, of neurons is **neural network**.

Neural networks

- Neuron



- Neural Network

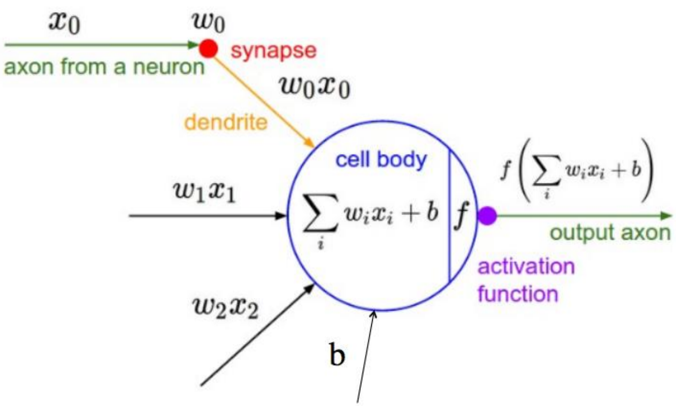


The union, in reticular form, of neurons is **neural network**.

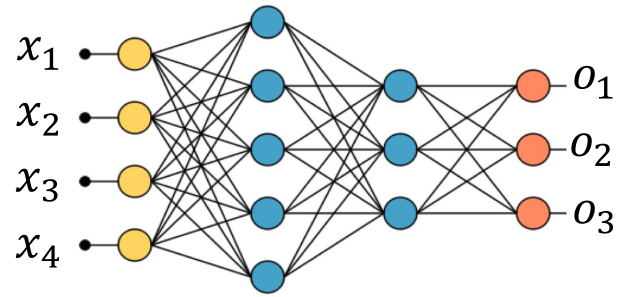
It is divided into **layers**, having an **input layer**, one or more hidden layers and an output layer.

Neural networks

- Neuron



- Neural Network



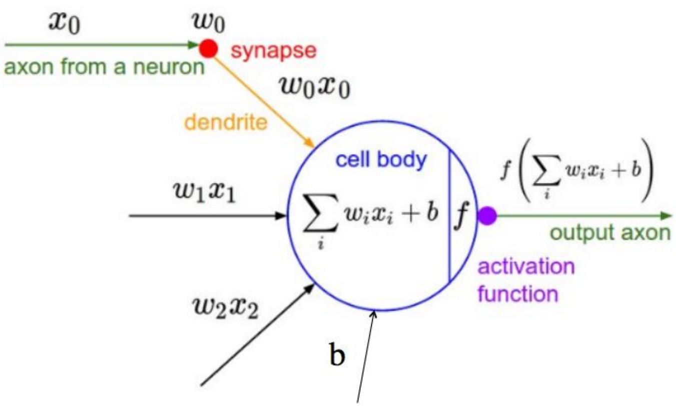
The union, in reticular form, of neurons is **neural network**.

It is divided into **layers**, having an **input layer**, one or more hidden layers and an output layer.

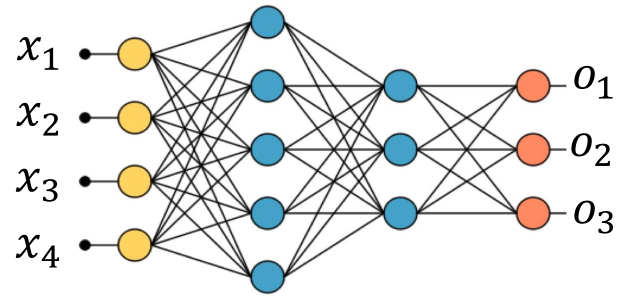
They are used as many neurons as variables in the input and as many as classes in the output.

Neural networks

- Neuron



- Neural Network

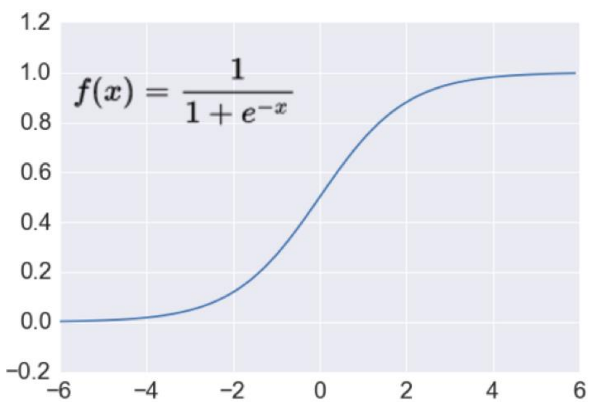


The union, in reticular form, of neurons is **neural network**.

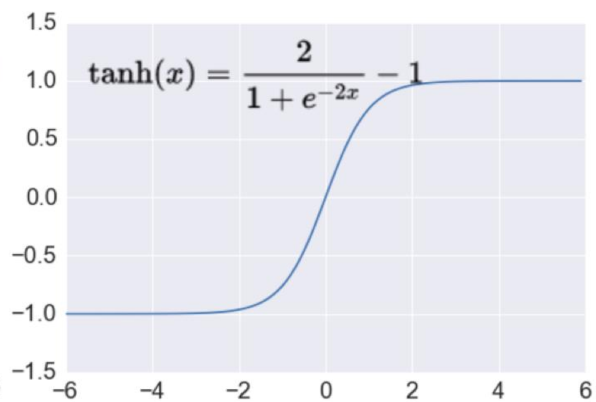
It is divided into **layers**, having an **input layer**, one or more hidden layers and an output layer.

They are used as many neurons as variables in the input and as many as classes in the output.

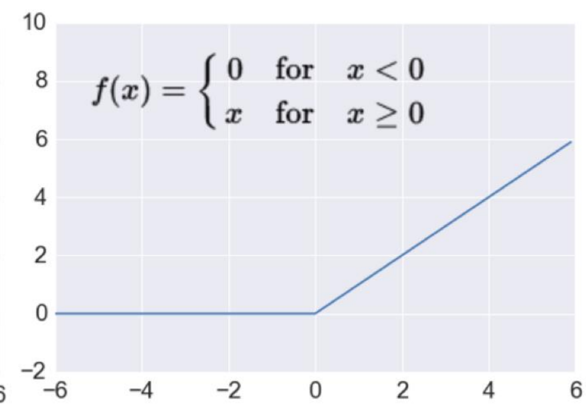
- Usual activation functions



Sigmoid



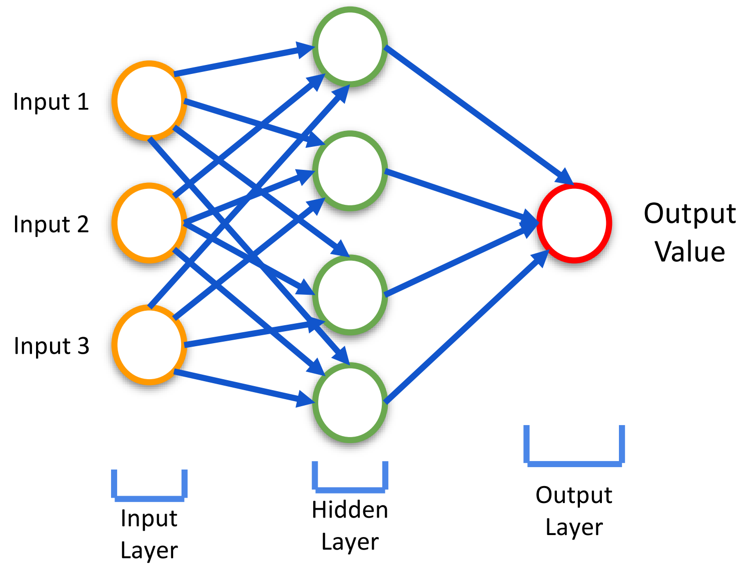
Hyperbolic tangent



ReLu

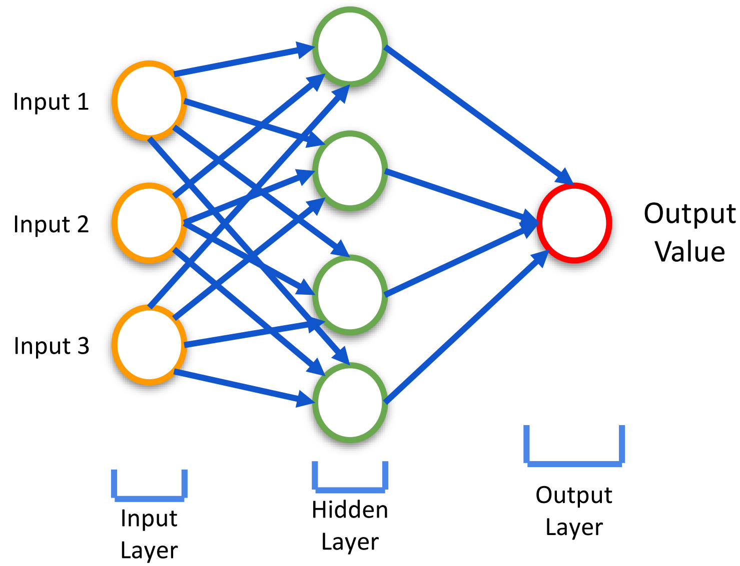
Neural networks

- Shallow Learning

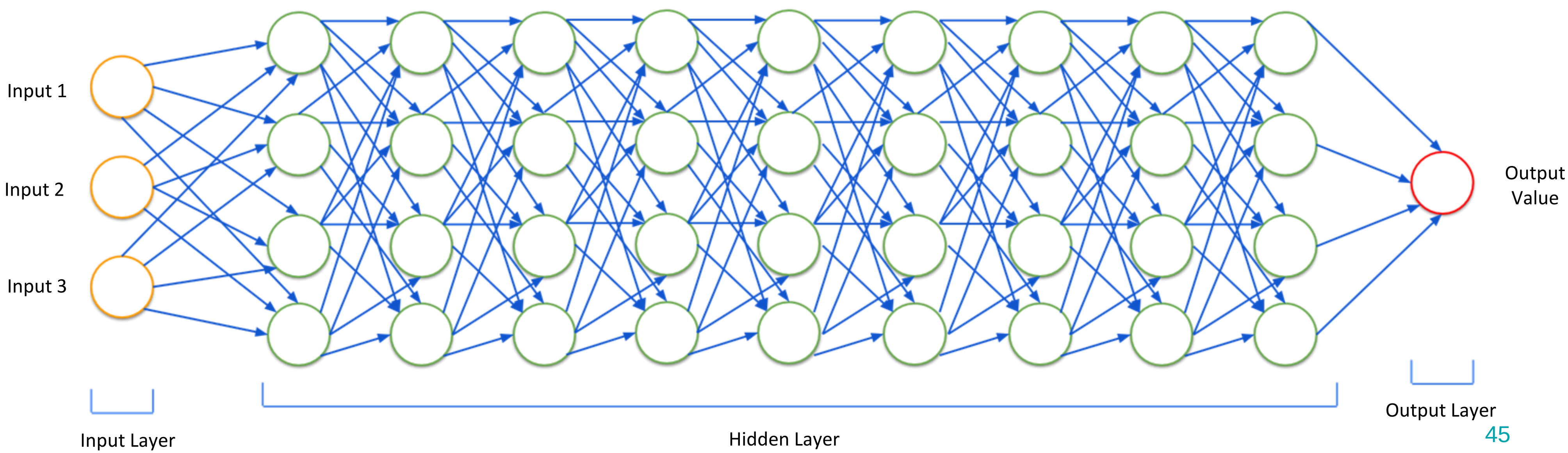


Neural networks

- Shallow Learning



- Deep Learning



Ensemble learning

- Combination of several predictions for a given paradigm

Ensemble learning

Mondragon Unibertsitatea
Escuela Politécnica Superior

- Combination of several predictions for a given paradigm

- Different configurations and/or training data subsets are employed

Ensemble learning

- Combination of several predictions for a given paradigm
- Different configurations and/or training data subsets are employed
- Results aggregation. Several ways depending of the type of problem or data

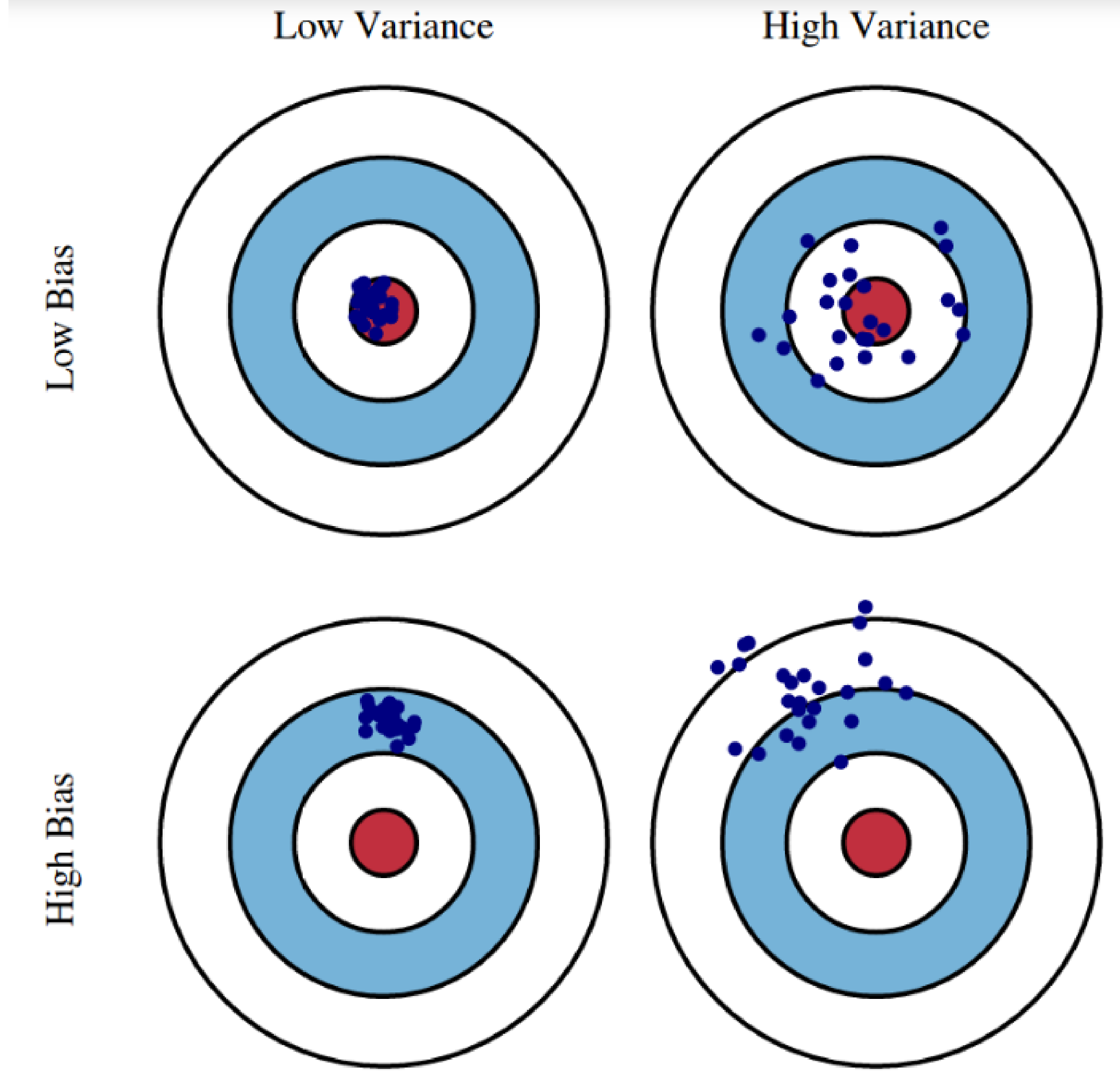
Ensemble learning

Mondragon Unibertsitatea
Escuela Politécnica Superior

- Combination of several predictions for a given paradigm
- Different configurations and/or training data subsets are employed
- Results aggregation. Several ways depending of the type of problem or data
- Variance reduction keeping bias

Ensemble learning

- Combination of several predictions for a given paradigm
- Different configurations and/or training data subsets are employed
- Results aggregation. Several ways depending of the type of problem or data
- Variance reduction keeping bias



Boosting & AdaBoost

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)

Boosting & AdaBoost

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

Boosting & AdaBoost

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps

Boosting & AdaBoost

Mondragon Unibertsitatea
Escuela Politécnica Superior

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps:
  - Train the weak learners iteratively

Boosting & AdaBoost

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps:
  - Train the weak learners iteratively
  - Aggregate their predictions using a weighted voting process, where the weights are adjusted using the weak kearners performance

Boosting & AdaBoost

Mondragon Unibertsitatea
Escuela Politécnica Superior

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps:
  - Train the weak learners iteratively
  - Aggregate their predictions using a weighted voting process, where the weights are adjusted using the weak kearners performance
  - Update the weights every time a new weak learner is trained

Boosting & AdaBoost

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps:
  - Train the weak learners iteratively
  - Aggregate their predictions using a weighted voting process, where the weights are adjusted using the weak kearners performance
  - Update the weights every time a new weak learner is trained

- **Adaboost** stands for **adaptive boosting**

Boosting & AdaBoost

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps:
  - Train the weak learners iteratively
  - Aggregate their predictions using a weighted voting process, where the weights are adjusted using the weak kearners performance
  - Update the weights every time a new weak learner is trained

- **Adaboost** stands for **adaptive boosting**
- It is a boosting algorithm in which the sequential aggregation of weak learners is guided by an **adaptive process**

Boosting & AdaBoost

Mondragon Unibertsitatea
Escuela Politécnica Superior

- Question: Can a set of weak learners create a single strong learner? (Kearns & Valiant)
- Answer: Yes. Think of an analogy with a set of experts with different levels of expertise taking decisions vs a single oracle.

- Boosting steps:
  - Train the weak learners iteratively
  - Aggregate their predictions using a weighted voting process, where the weights are adjusted using the weak kearners performance
  - Update the weights every time a new weak learner is trained

- **Adaboost** stands for **adaptive boosting**
- It is a boosting algorithm in which the sequential aggregation of weak learners is guided by an **adaptive process**
- Such process considers the error of each weak learner on each sample in order to **adapt the weights for the weak learners** to be aggregated, so that **the overall error decreases as much as possible**.

Bagging

Mondragon Unibertsitatea
Escuela Politécnica Superior

- **Bagging** stands for **boostrap aggregating**

Bagging

- **Bagging** stands for **boostrap aggregating**

- Process

Bagging

Mondragon Unibertsitatea
Escuela Politécnica Superior

- **Bagging** stands for **boostrap aggregating**

- Process:
  Given a dataset A with n samples, a number of samples $n' \leq n$, and a number of models m,

Bagging

Mondragon Unibertsitatea
Escuela Politécnica Superior

- **Bagging** stands for **boostrap aggregating**

- Process:
  Given a dataset A with n samples, a number of samples $n' \leq n$, and a number of models m,

  – Sample m datasets, with n' samples each, by sampling with replacement

Bagging

Mondragon Unibertsitatea
Escuela Politécnica Superior

- **Bagging** stands for **boostrap aggregating**

- Process:
  Given a dataset A with n samples, a number of samples $n' \leq n$, and a number of models m,
  - Sample m datasets, with n' samples each, by sampling with replacement
  - For each dataset, create a predictive model

Bagging

- **Bagging** stands for **boostrap aggregating**

- Process:

  Given a dataset A with n samples, a number of samples $n' \leq n$, and a number of models m,

  - Sample m datasets, with n' samples each, by sampling with replacement
  - For each dataset, create a predictive model
  - Aggregate the output of all m models by voting

Bagging

- **Bagging** stands for **boostrap aggregating**

- Process:
  Given a dataset A with n samples, a number of samples $n' \leq n$, and a number of models m,
  - Sample m datasets, with n' samples each, by sampling with replacement
  - For each dataset, create a predictive model
  - Aggregate the output of all m models by voting

- The most famous bagging-type algorithm is **Random Forests**

Bagging

Mondragon Unibertsitatea
Escuela Politécnica Superior

- **Bagging** stands for **boostrap aggregating**

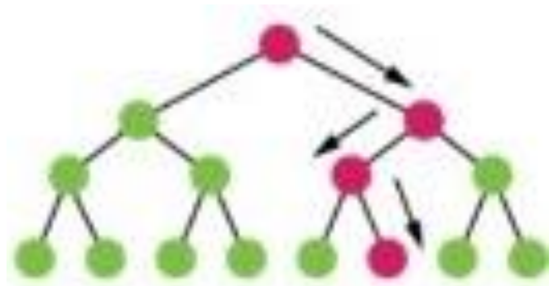
- Process:
  Given a dataset A with n samples, a number of samples $n' \leq n$, and a number of models m,
  - Sample m datasets, with n' samples each, by sampling with replacement
  - For each dataset, create a predictive model
  - Aggregate the output of all m models by voting

- The most famous bagging-type algorithm is **Random Forests**

- More recently, new approaches arised, such as **Gradient Boosting classifiers**, being **Extreme Gradient Boosting** (XGBoost) the new ML rockstar

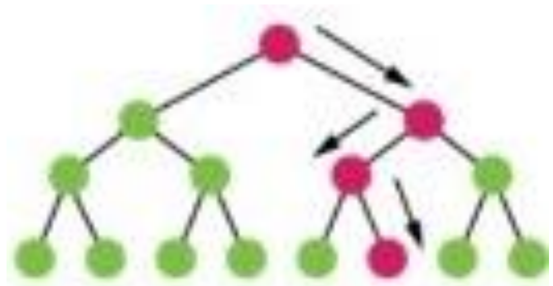
Random Forests

- Ensemble learning for decision trees



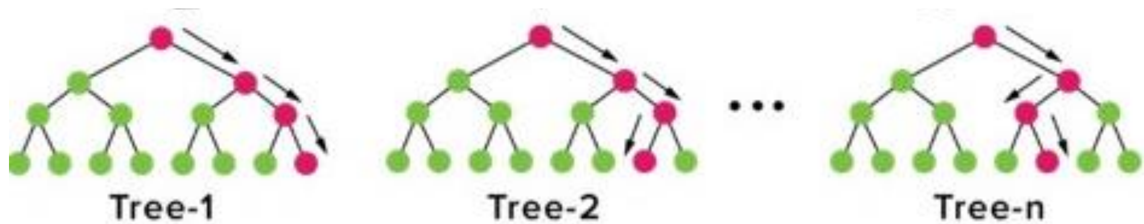
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen



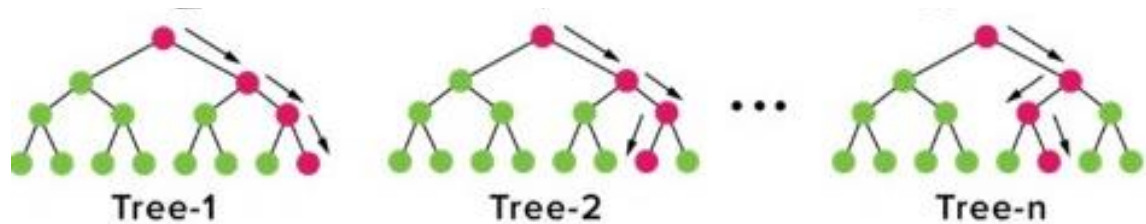
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree



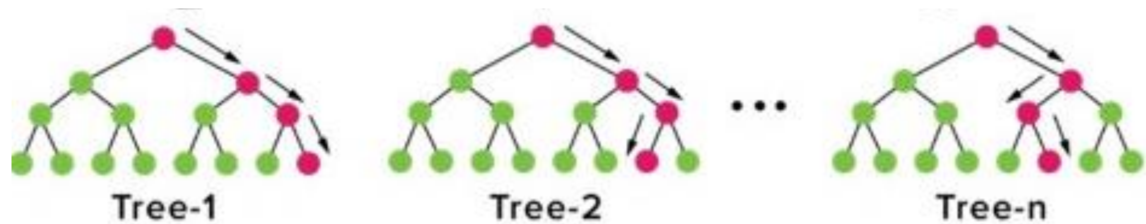
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of $\boldsymbol{L}$ variables** are randomly chosen. By default, $L = \lceil \sqrt{M} \rceil$



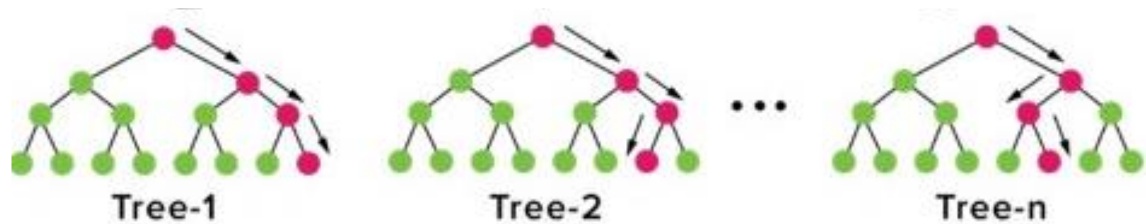
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of L variables** are randomly chosen. By default, $L = \lceil \sqrt{M} \rceil$
  - K samples are chosen by **random sampling with reposition**. By default, $K = N$



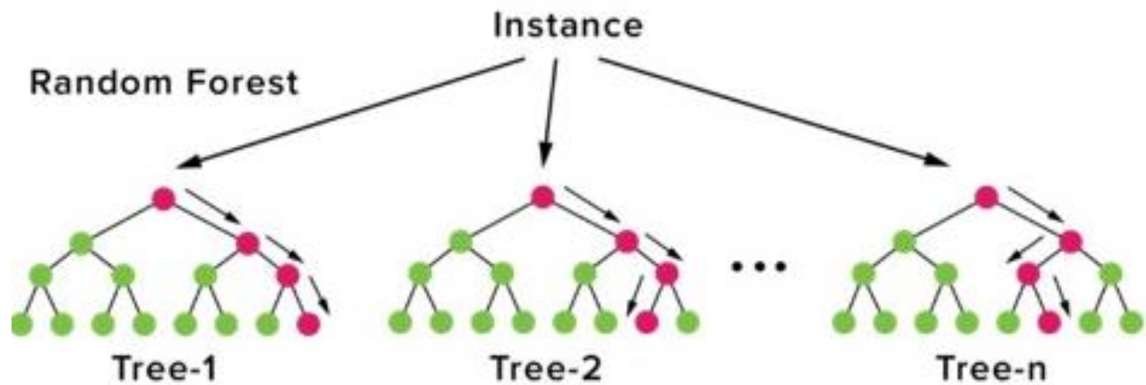
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of L variables** are randomly chosen. By default, $L = \lceil \sqrt{M} \rceil$
  - K samples are chosen by **random sampling with reposition**. By default, $K = N$
- All trees are trained



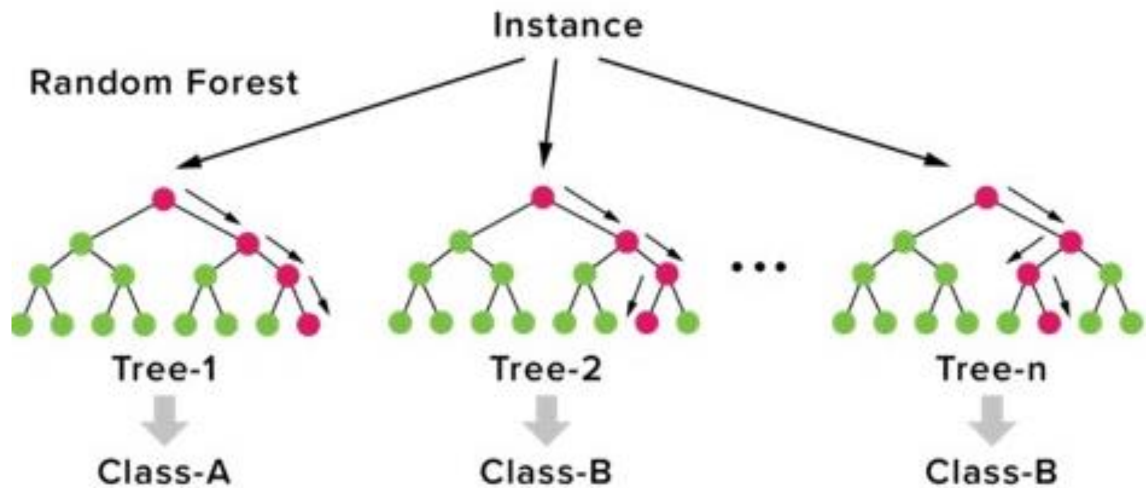
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of $\boldsymbol{L}$ variables** are randomly chosen. By default, $L = \lceil\sqrt{M}\rceil$
  - K samples are chosen by **random sampling with reposition**. By default, $K = N$
- All trees are trained
- For each sample, n predictions are obtained



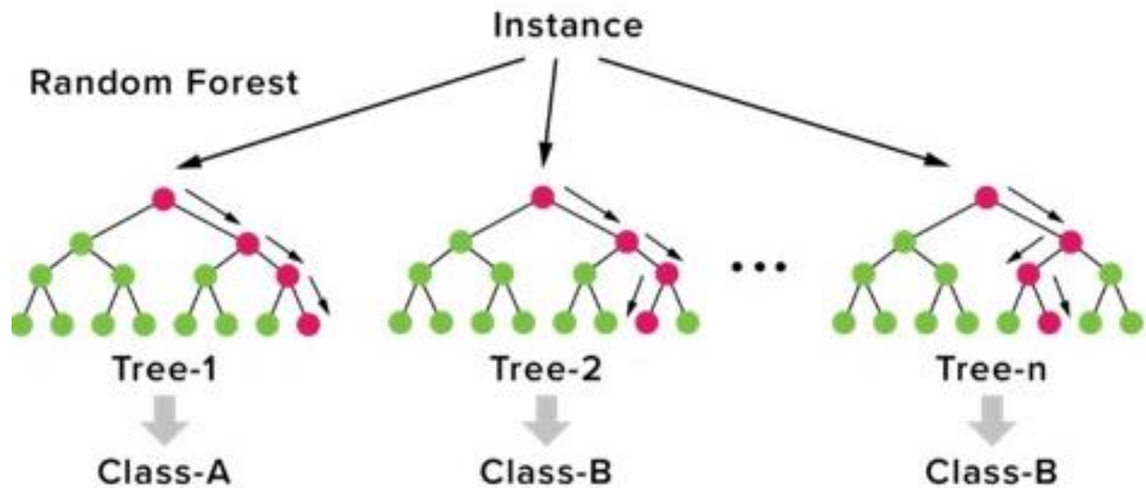
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of L variables** are randomly chosen. By default, $L = \lfloor\sqrt{M}\rfloor$
  - K samples are chosen by **random sampling with reposition**. By default, $K = N$
- All trees are trained
- For each sample, n predictions are obtained



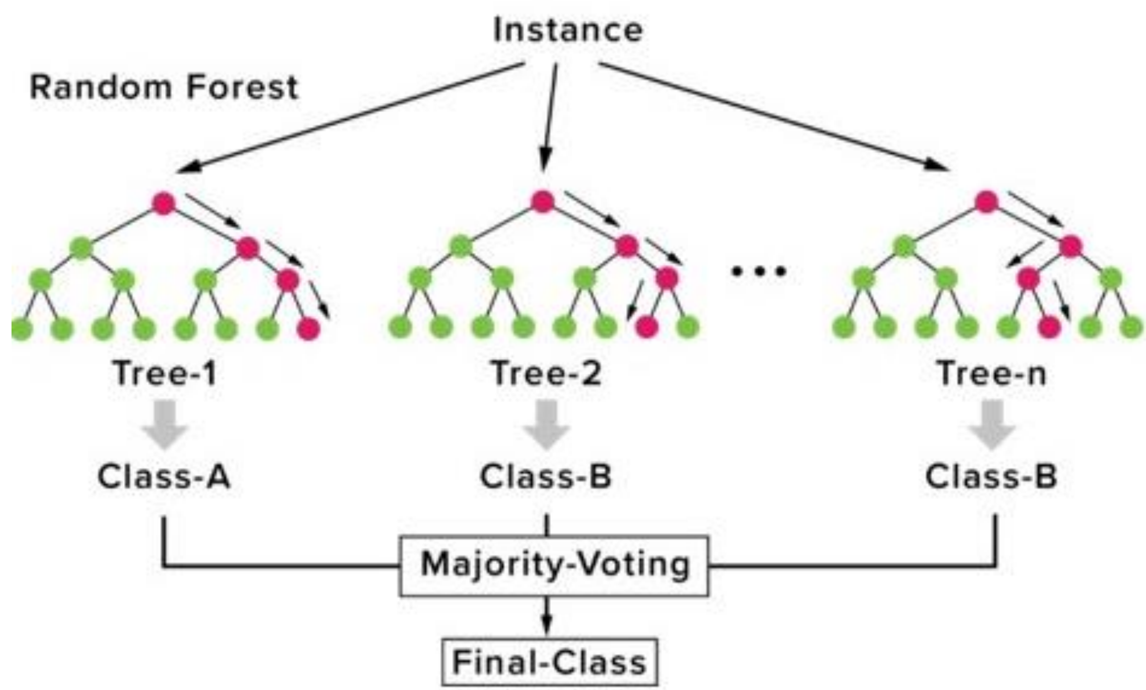
Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of L variables** are randomly chosen. By default, $L = \lceil\sqrt{M}\rceil$
  - K samples are chosen by **random sampling with reposition**. By default, $K = N$
- All trees are trained
- For each sample, n predictions are obtained
- Predictions are **aggregated** by **majority voting**. Different possible options



Random Forests

- Ensemble learning for decision trees
- The number n of trees is chosen
- For each tree:
  - A **subset of L variables** are randomly chosen. By default, $L = \lceil \sqrt{M} \rceil$
  - K samples are chosen by **random sampling with reposition**. By default, $K = N$
- All trees are trained
- For each sample, n predictions are obtained
- Predictions are **aggregated** by **majority voting**. Different possible options



Validation

- We must distinguish binary from multiclass classification

Validation

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case

Validation

Mondragon Unibertsitatea
Escuela Politécnica Superior

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case
- Only a few are directly applicable to multiclass situations

Validation

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case
- Only a few are directly applicable to multiclass situations

Actual \ Predicted	Iris-setosa	Iris-versicolor	Iris-virginica	Σ
Iris-setosa	100.0 %	0.0 %	0.0 %	50
Iris-versicolor	0.0 %	88.7 %	6.4 %	50
Iris-virginica	0.0 %	11.3 %	93.6 %	50
Σ	50	53	47	150

Validation

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case
- Only a few are directly applicable to multiclass situations

Actual \ Predicted	Iris-setosa	Iris-versicolor	Iris-virginica	Σ
Iris-setosa	100.0 %	0.0 %	0.0 %	50
Iris-versicolor	0.0 %	88.7 %	6.4 %	50
Iris-virginica	0.0 %	11.3 %	93.6 %	50
Σ	50	53	47	150

- All scores are usable by means of 2 possible simplifications:

Validation

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case
- Only a few are directly applicable to multiclass situations

		Predicted			
		Iris-setosa	Iris-versicolor	Iris-virginica	Σ
Actual	Iris-setosa	100.0 %	0.0 %	0.0 %	50
	Iris-versicolor	0.0 %	88.7 %	6.4 %	50
	Iris-virginica	0.0 %	11.3 %	93.6 %	50
	Σ	50	53	47	150

- All scores are usable by means of 2 possible simplifications:
  - One-vs-All: If we have k classes, we train k binary model predicting each class against the rest iteratively

Validation

Mondragon Unibertsitatea
Escuela Politécnica Superior

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case
- Only a few are directly applicable to multiclass situations

		Predicted			
		Iris-setosa	Iris-versicolor	Iris-virginica	Σ
Actual	Iris-setosa	100.0 %	0.0 %	0.0 %	50
	Iris-versicolor	0.0 %	88.7 %	6.4 %	50
	Iris-virginica	0.0 %	11.3 %	93.6 %	50
	Σ	50	53	47	150

- All scores are usable by means of 2 possible simplifications:
  - One-vs-All: If we have k classes, we train k binary model predicting each class against the rest iteratively
  - One-vs-One: If we have k classes, we train all pairwise possible binary models. In total, the total number of binary models is k(k-1)/2

Validation

- We must distinguish binary from multiclass classification
- All classification metrics seen in FAA are valid for the binary case
- Only a few are directly applicable to multiclass situations

		Predicted			
		Iris-setosa	Iris-versicolor	Iris-virginica	Σ
Actual	Iris-setosa	100.0 %	0.0 %	0.0 %	50
	Iris-versicolor	0.0 %	88.7 %	6.4 %	50
	Iris-virginica	0.0 %	11.3 %	93.6 %	50
	Σ	50	53	47	150

- All scores are usable by means of 2 possible simplifications:
  - One-vs-All: If we have k classes, we train k binary model predicting each class against the rest iteratively
  - One-vs-One: If we have k classes, we train all pairwise possible binary models. In total, the total number of binary models is k(k-1)/2
- In both cases, numerical scores are **aggregated by averaging**

Python functions

Mondragon Unibertsitatea
Escuela Politécnica Superior

Algoritmo	Entorno	Función
K-Nearest Neighbors	Scikit-learn	KNeighborsClassifier
Decision tree	Scikit-learn	DecisionTreeClassifier
Logistic Regression	Scikit-learn	LogisticRegression
Naïve Bayes	Scikit-learn	GaussianNB MultinomialNB ComplementNB BernoulliNB
Support Vector Machines	Scikit-learn	SVC NuSVC LinearSVC
Neural Networks	Scikit-learn	MLPClassifier
AdaBoost	Scikit-learn	AdaBoostClassifier
Bagging	Scikit-learn	BaggingClassifier
Random Forests	Scikit-learn	RandomForestClassifier
Gradient Boosting	Scikit-learn	GradientBoostingClassifier HistGradientBoostingClassifier

Mondragon Unibertsitatea
Escuela Politécnica Superior

Eskerrik asko

Muchas gracias

Thank you

Carlos Cernuda
ccernuda@mondragon.edu

MGEP
Goiru, 2
20500 Arrasate – Mondragon
Tlf. 662420414