



Universidad **Ricardo Palma**

RECTORADO
PROGRAMA DE ESPECIALIZACIÓN EN CIENCIA DE DATOS

Formamos seres humanos para una cultura de paz

BIG DATA APLICADO


SESIÓN 05 - 01

Expositores:

David Narváez

Eder Pineda

bigdataplicado@gmail.com

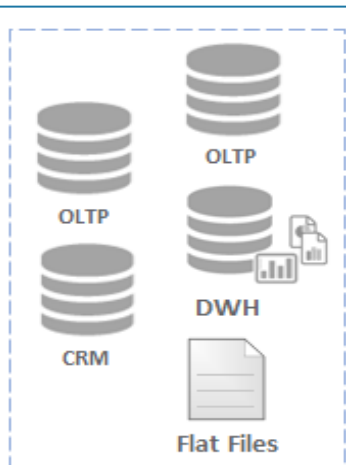


***Big data** is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation (**Gartner**).*



Arquitectura Lambda

DATA SOURCES

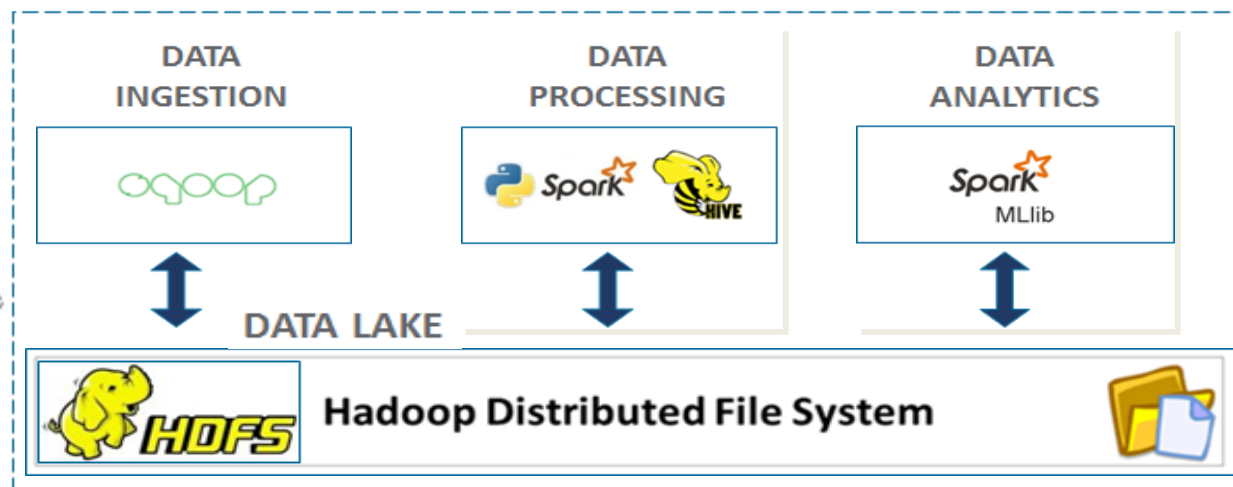


Batch

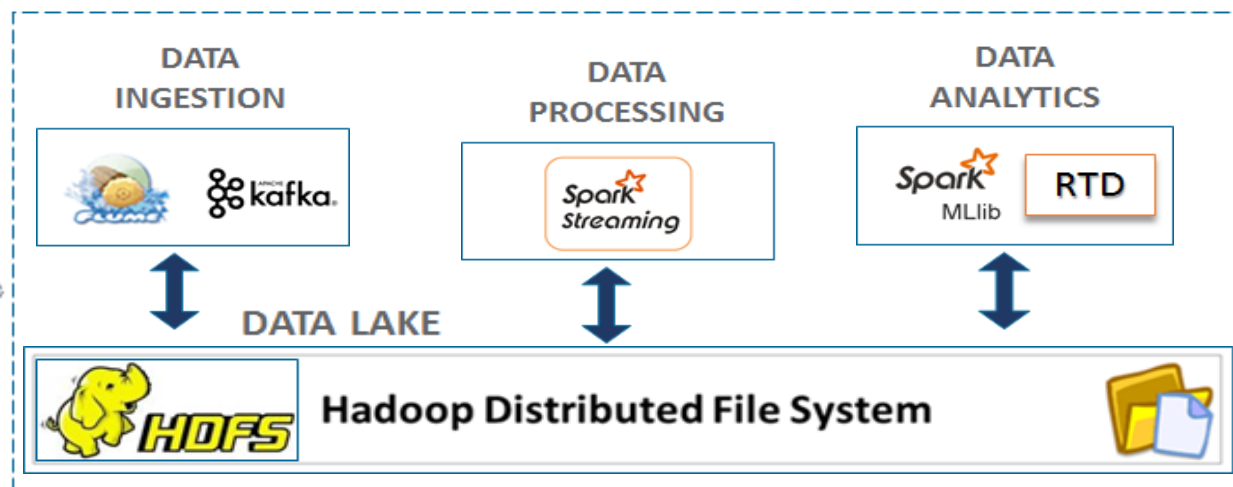
Real Time



BATCH LAYER



SPEED LAYER



SERVING LAYER

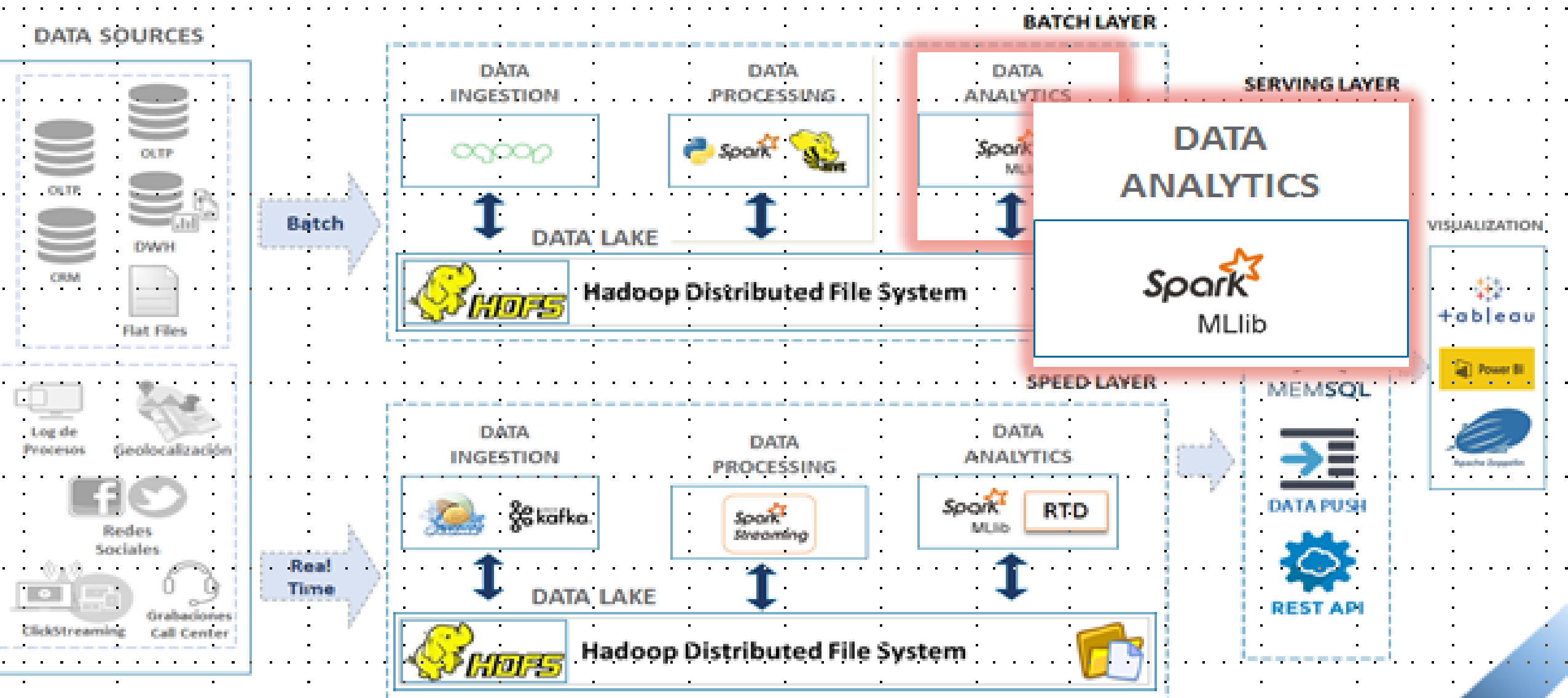


VISUALIZATION



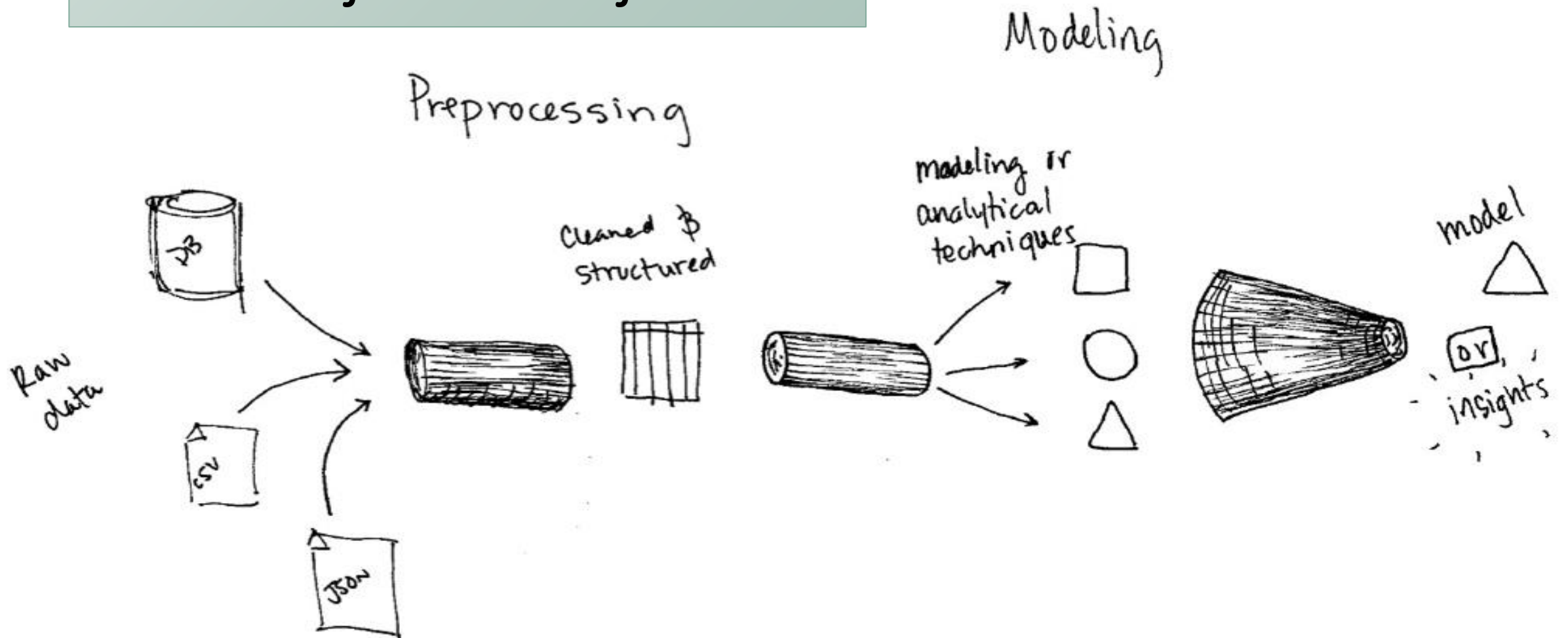


Arquitectura Lambda



1. Introducción a Machine Learning

Flujo de Trabajo



AGENDA

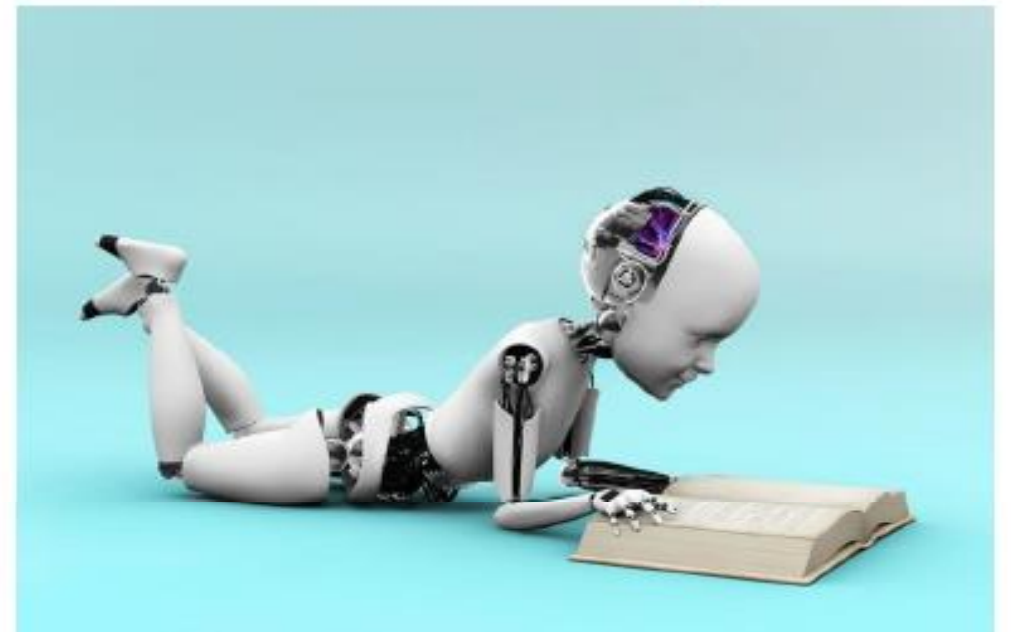
1. Introducción a Machine Learning
2. Tipos de Aprendizaje
 - A. *Aprendizaje Supervisado*
 - B. *Aprendizaje No Supervisado*
 - C. *Paquete MLIB*
 - D. *Indicadores principales de Aprendizaje Automático*
3. Taller de Modelo de Aprendizaje
 - A. *Aprendizaje No Supervisado (K-Means)*
 - B. *Aprendizaje Supervisado (Regresión Lineal / Regresión Logística / Arbol de Decisión / Random Forest / RNA)*

1. Introducción a Machine Learning



1. Introducción a Machine Learning

- El aprendizaje automático (Machine Learning) es el **conjunto de técnicas** que tiene como objetivo emular los **proceso de aprendizaje** en el contexto de una computadora, con el fin de que esta pueda **resolver problemas** basados en **información a priori**.
- Se lo conoce como analítica avanzada



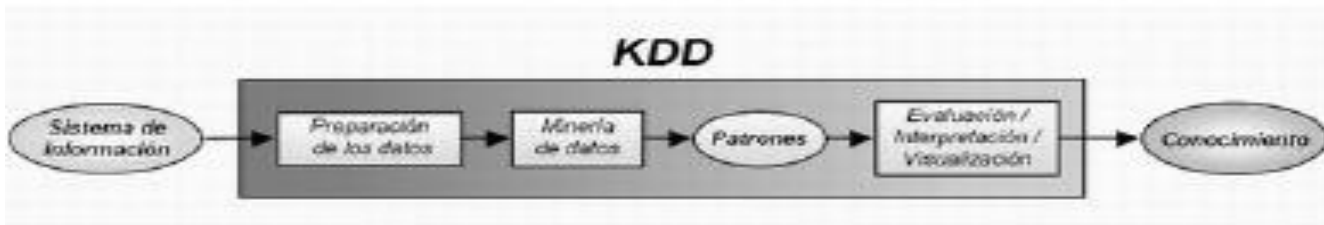
1. Introducción a Machine Learning

¿Dónde se usa Machine Learning?

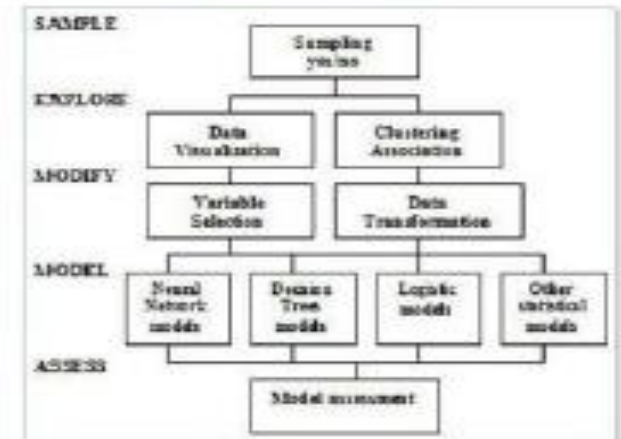
Retail	Manufactura	Servicios Financieros	Telecomunicaciones	Energía	Marketing
Customer Relationship Management	Investigación de producto	Análisis de riesgo	Scoring de crédito	Predicción de consumo	Campañas dirigidas
Optimización de cadena de suministro	Mantenimiento predictivo	Detección y prevención de fraude	Prevención de churn	Modelación de operaciones	Segmentación de clientes
Asignación dinámica de precio	Distribución	Prevención de Churn	Optimización de Red	Análisis histórico	Adquisición de clientes
Detección y prevención de fraude	Optimización	Detección de lavado de dinero	Análisis de consumo		
		Scoring de crédito			

2. Tipos de Aprendizaje

Metodologías



SEMMA



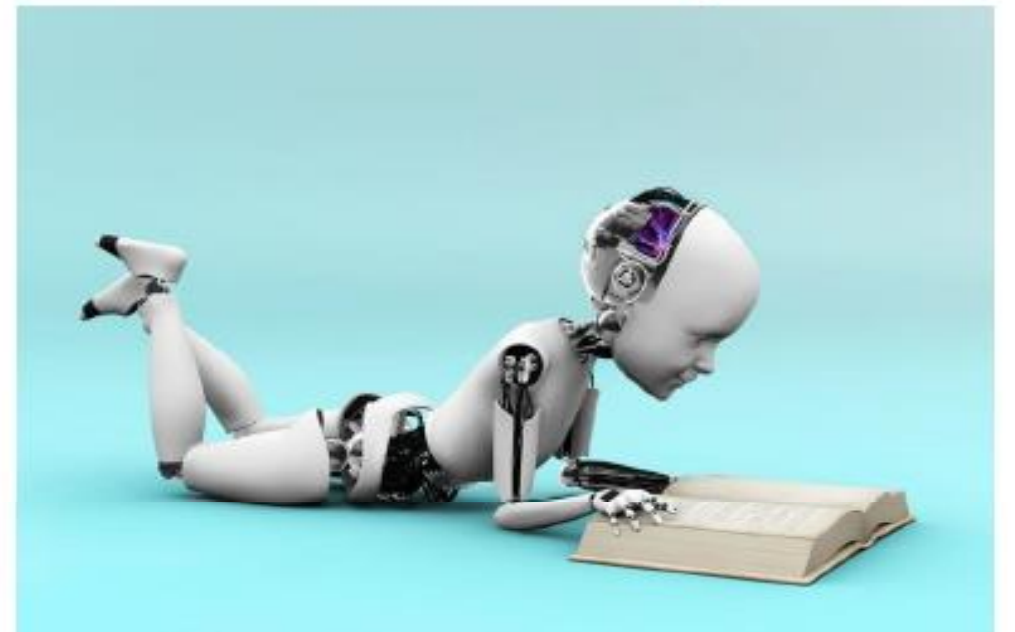
CRISP - DM

Cross Industry Standard Process for Data Mining

Desarrollo, implementación y mantenimiento de CRISP - DM. Como un proceso de Outsourcing que simplifica la obtención de conocimiento sin la adquisición de software o personal experto

1. Introducción a Machine Learning

- Veamos algunos casos de aprendizaje de maquina:



1. Introducción a Machine Learning

1



Problema: Determinar aquellos clientes para los cuales es viable prestarles dinero

1. Introducción a Machine Learning

1



Problema: Determinar aquellos clientes para los cuales es viable prestarles dinero

Conocimiento a priori: Conjunto de datos (atributos) asociados a cada uno de los clientes.

¿Cómo representar (codificar) un cliente?

$$\vec{x} = [x_1, x_2, \dots, x_n] \in \mathfrak{R}^n$$

1. Introducción a Machine Learning

1



$$\vec{x} = [x_1, x_2, \dots, x_n] \in \mathcal{R}^n$$

Ingresos

Gastos

Calificación en el buró de
crédito

1. Introducción a Machine Learning

1

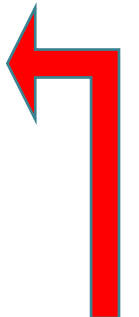


Problema: Determinar aquellos clientes para los cuales es viable prestarles dinero

Conocimiento a priori: Conjunto de datos (atributos) asociados a cada uno de los clientes.

4.9	3.0	1.4
4.7	3.2	1.3
4.6	3.1	1.5
5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.9	1.4
4.9	3.1	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4
4.3	3.0	1.1
5.8	4.0	1.2
5.7	4.4	1.5
5.4	3.9	1.3
5.1	3.5	1.4

4.9	3.0	1.4	1
4.7	3.2	1.3	0
4.6	3.1	1.5	0
5.0	3.6	1.4	1
5.4	3.9	1.7	1
4.6	3.4	1.4	1
5.0	3.4	1.5	0
4.4	2.9	1.4	1



Un experto puede incluir información basada en su experiencia

1. Introducción a Machine Learning

1



Aprendizaje: Encontrar un **modelo** que represente el comportamiento de los datos.

Problema: Determinar aquellos clientes para los cuales es viable prestarles dinero

Conocimiento a priori: Conjunto de datos (atributos) asociados a cada uno de los clientes.

4.9	3.0	1.4
4.7	3.2	1.3
4.6	3.1	1.5
5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.9	1.4
4.9	3.1	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4
4.3	3.0	1.1
5.8	4.0	1.2
5.7	4.4	1.5
5.4	3.9	1.3
5.1	3.5	1.4

4.9	3.0	1.4	1
4.7	3.2	1.3	0
4.6	3.1	1.5	0
5.0	3.6	1.4	1
5.4	3.9	1.7	1
4.6	3.4	1.4	1
5.0	3.4	1.5	0
4.4	2.9	1.4	1

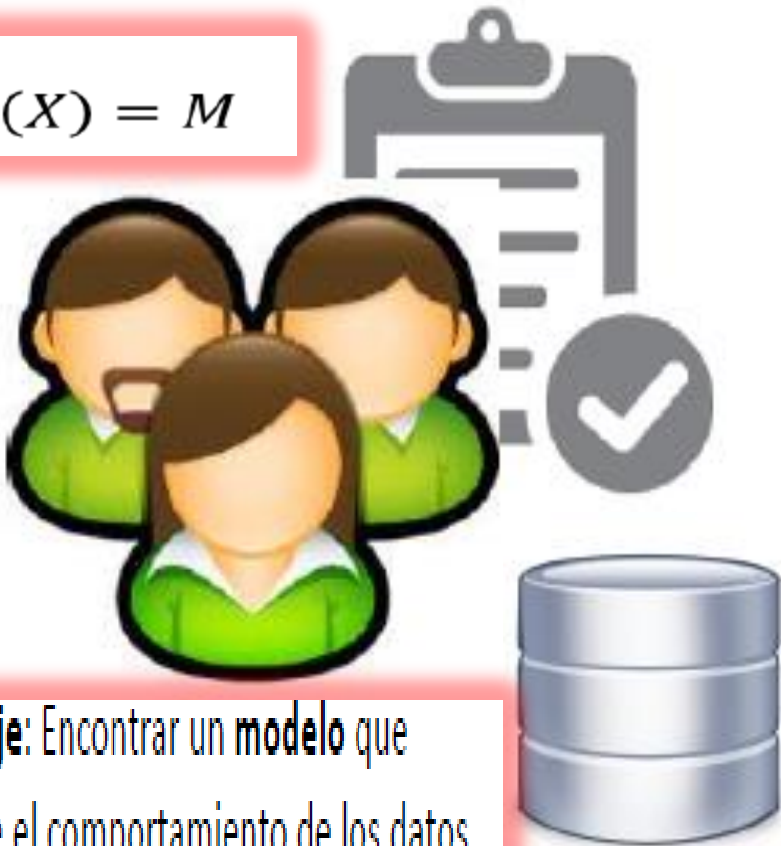


Un experto puede incluir información basada en su experiencia

1. Introducción a Machine Learning

1

$$y = f(X) = M$$



Aprendizaje: Encontrar un **modelo** que represente el comportamiento de los datos.

Problema: Determinar aquellos clientes para los cuales es viable prestarles dinero

Conocimiento a priori: Conjunto de datos (atributos) asociados a cada uno de los clientes.

4.9	3.0	1.4
4.7	3.2	1.3
4.6	3.1	1.5
5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.9	1.4
4.9	3.1	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4
4.3	3.0	1.1
5.8	4.0	1.2
5.7	4.4	1.5
5.4	3.9	1.3
5.1	3.5	1.4

4.9	3.0	1.4	1
4.7	3.2	1.3	0
4.6	3.1	1.5	0
5.0	3.6	1.4	1
5.4	3.9	1.7	1
4.6	3.4	1.4	1
5.0	3.4	1.5	0
4.4	2.9	1.4	1



Un experto puede incluir información basada en su experiencia

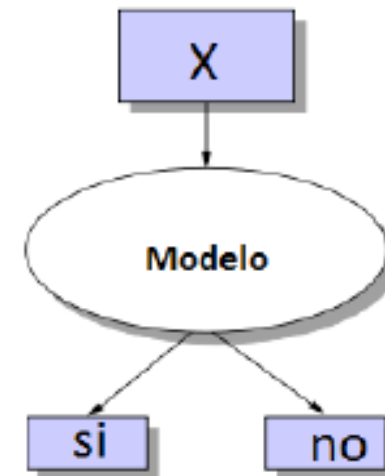
1. Introducción a Machine Learning

1

¡Esto es el famoso aprendizaje de máquina!



$$f(X) = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n$$



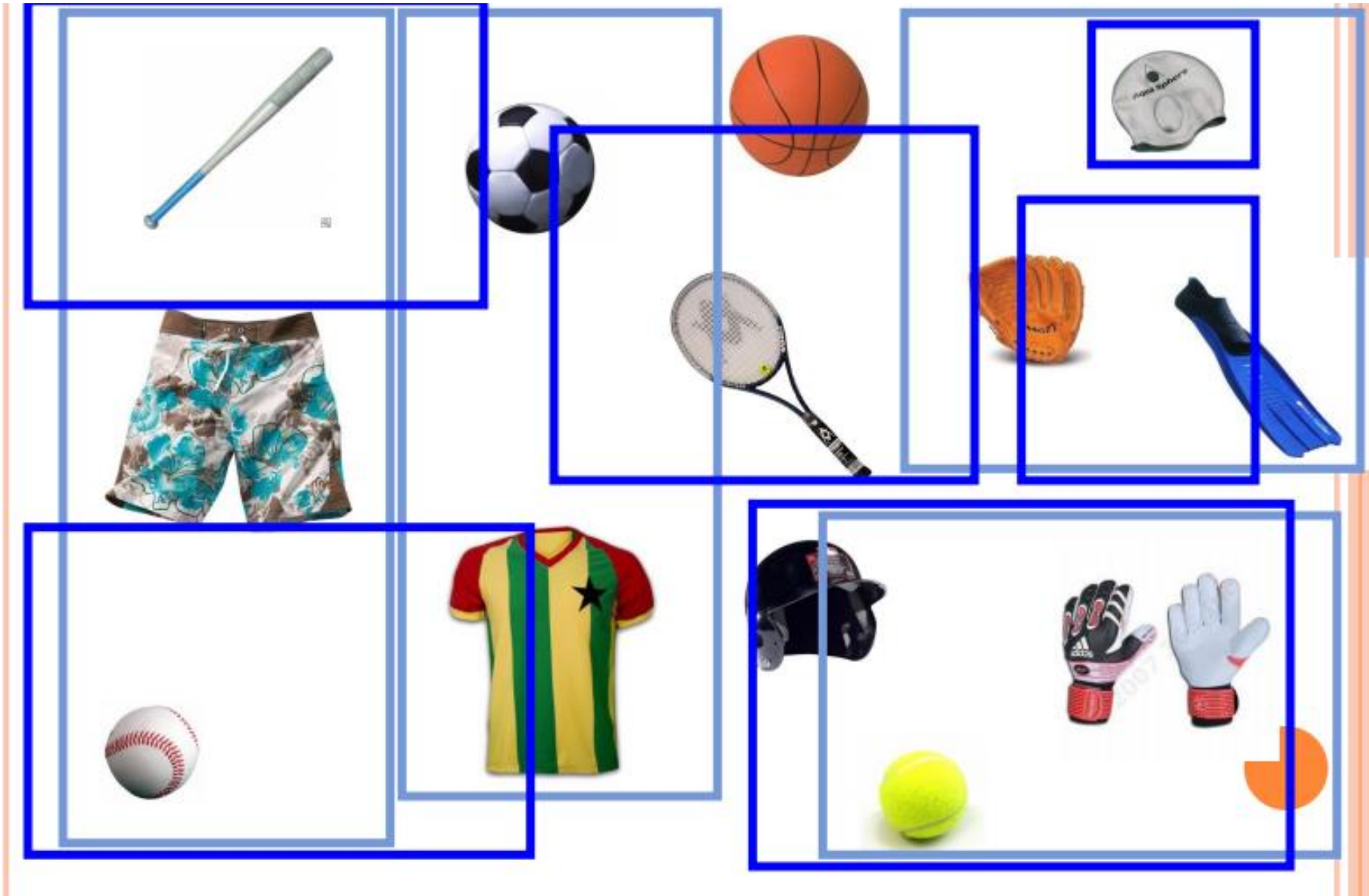
1. Introducción a Machine Learning

2

- En el siguiente caso se nos presenta ciertos objetos no se tiene información a priori sobre los objetos, y se nos pide que busquemos un patrón que nos ayude a diferenciar los artículos.

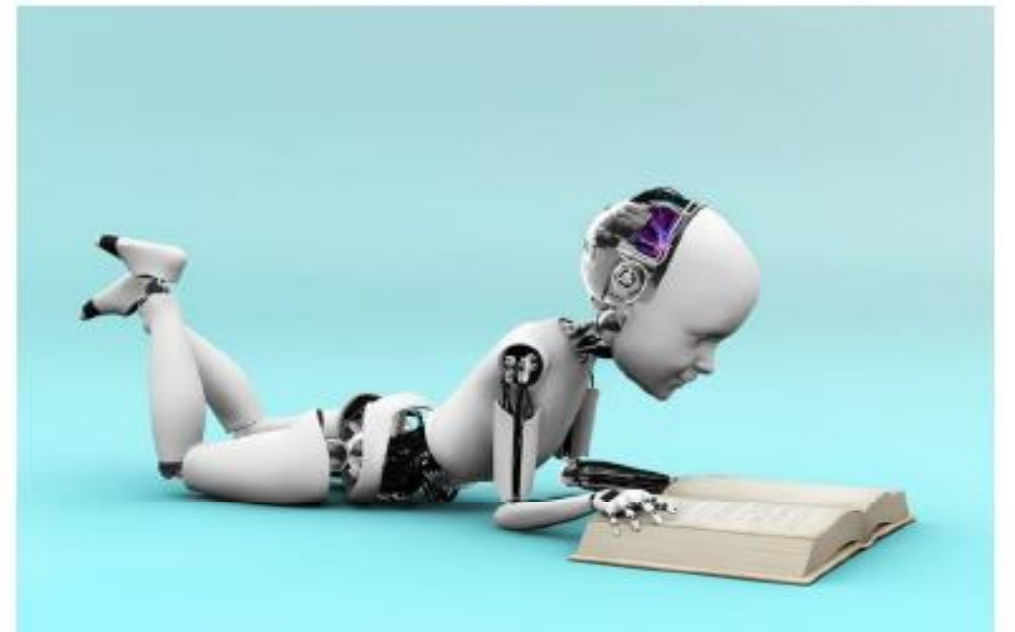
1. Introducción a Machine Learning

2



1. Introducción a Machine Learning

- Como se puede observar dependiendo del **problema** se tiene **diferentes objetivos**; Sin embargo, estas tareas se enmarcan en los siguientes **tipos de aprendizaje**:



2. Tipos de Aprendizaje

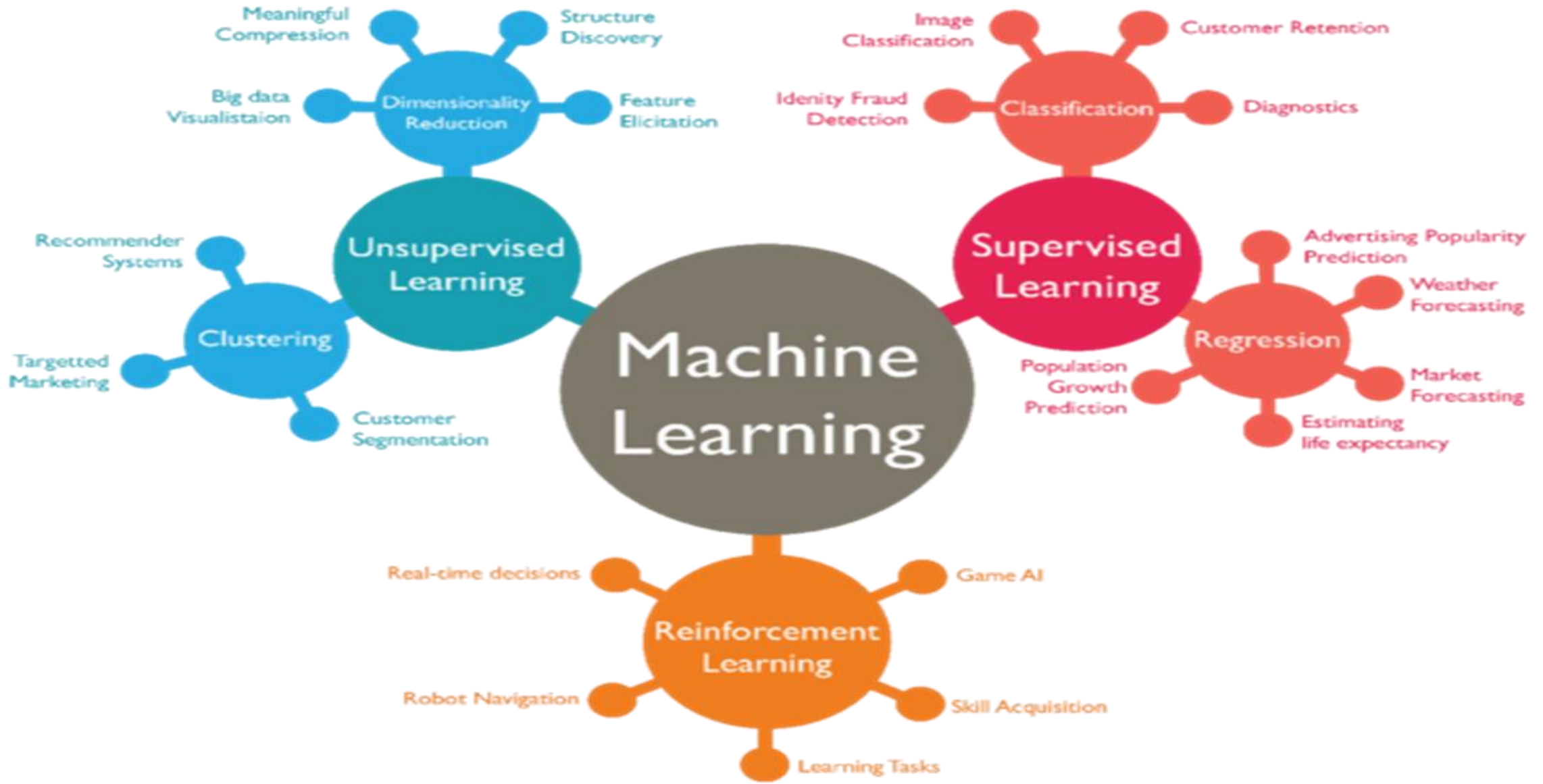
Supervised Learning



Unsupervised Learning



2. Tipos de Aprendizaje



2. Tipos de Aprendizaje

A. Aprendizaje Supervisado

- Se tiene una variable objetivo (Variable de Salida).
- Existe una dependencia de las variables de entrada con las variables de salida

2. Tipos de Aprendizaje

A. Aprendizaje Supervisado

- Regresión.
 - Lineal.
 - No Lineal
 - Logística Binaria
 - Logística Multinomial
 - Logístico con enlaces asimétricos
 - Modelos Lineales Generalizados
- Árboles.
 - CHAID
 - CRT
 - QUEST
- ▶ Discriminante.
- ▶ Series de Tiempo.
 - ▶ Suavizamiento Exponencial.
 - ▶ SARIMAS.
- ▶ Análisis de Sobrevida.
 - ▶ Tablas de Mortalidad.
 - ▶ Kaplan – Meier.
 - ▶ Regresión de Cox.

2. Tipos de Aprendizaje

B. Aprendizaje No Supervisado

- No hay una variable objetivo (Variable de Salida).
- Todas las variables tienen la misma importancia.
- Se busca la interdependencia de las variables.de salida

2. Tipos de Aprendizaje

B. Aprendizaje No Supervisado

- Clúster.
 - K – Medias.
 - Jerárquico.
 - Dos Fases.
 - ...
 - Análisis Factorial.
 - Análisis de Correspondencias.
- ▶ Componentes principales categóricos.
 - ▶ Correlación Canónica.
 - ▶ Análisis de fiabilidad.

2. Tipos de Aprendizaje

C. MLIB Package

MLIB es un paquete de SPARK que incluye:

- Limpieza de datos.
- Generación y selección de características.
- Entrenamiento y prueba de una gran cantidad de modelos supervisados y no supervisados (Machine Learning).

2. Tipos de Aprendizaje

D. Indicadores Principales

Cuando se quiere evaluar o compararlo un modelo se puede realizar siguiente los siguientes criterios.

2. Tipos de Aprendizaje

D. Indicadores Principales

DENOMINACIÓN DE LAS VARIABLES	
X	Y
Predictora, regresor	Criterio
Explicativa	Explicada
Predeterminada	Respuesta
Independiente	Dependiente
Exógena	Endógena
(Explica la variabilidad de otra variable)	(Su variabilidad es explicada por otra variable)

2. Tipos de Aprendizaje

D. Indicadores Principales

Variable X	Variable Y	Coeficiente de correlación
Cualitativa	Cualitativa	Chi-cuadrado Contingencia Phi
Ordinal	Ordinal	Spearman
Cualitativa	Cuantitativa	Biserial-puntual
Cuantitativa	Cuantitativa	Pearson

2. Tipos de Aprendizaje

D. Indicadores Principales

MATRIZ DE CONFUSIÓN

		Condition (as determined by "Gold standard")		
		Condition positive	Condition negative	
Test outcome	Test outcome positive	True positive	False positive (Type I error)	Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$
	Test outcome negative	False negative (Type II error)	True negative	Negative predictive value = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
		Sensitivity = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	Specificity = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Accuracy

2. Tipos de Aprendizaje

D. Indicadores Principales

MATRIZ DE CONFUSIÓN

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- La Precisión P de un modelo de predicción es la proporción del número total de predicciones que son correctas respecto al total. Se determina utilizando la ecuación: $P = (a+d)/(a+b+c+d)$.
- La Precisión Positiva (PP) es la proporción de casos positivos que fueron identificados correctamente, tal como se calcula usando la ecuación: $PP = d/(c+d)$.
- La Precisión Negativa (PN) es la proporción de casos negativos que fueron identificados correctamente, tal como se calcula usando la ecuación: $PN = a/(a+b)$.

2. Tipos de Aprendizaje

D. Indicadores Principales

MATRIZ DE CONFUSIÓN

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- Falsos Positivos (FP) es la proporción de casos negativos que fueron clasificados incorrectamente como positivos, tal como se calcula utilizando la ecuación: $FP = b/(a+b)$.
- Falsos Negativos (FN) es la proporción de casos positivos que fueron clasificados incorrectamente como negativos, tal como se calcula utilizando la ecuación: $FN = c/(c+d)$.

2. Tipos de Aprendizaje

D. Indicadores Principales

MATRIZ DE CONFUSIÓN

- Ejemplo:

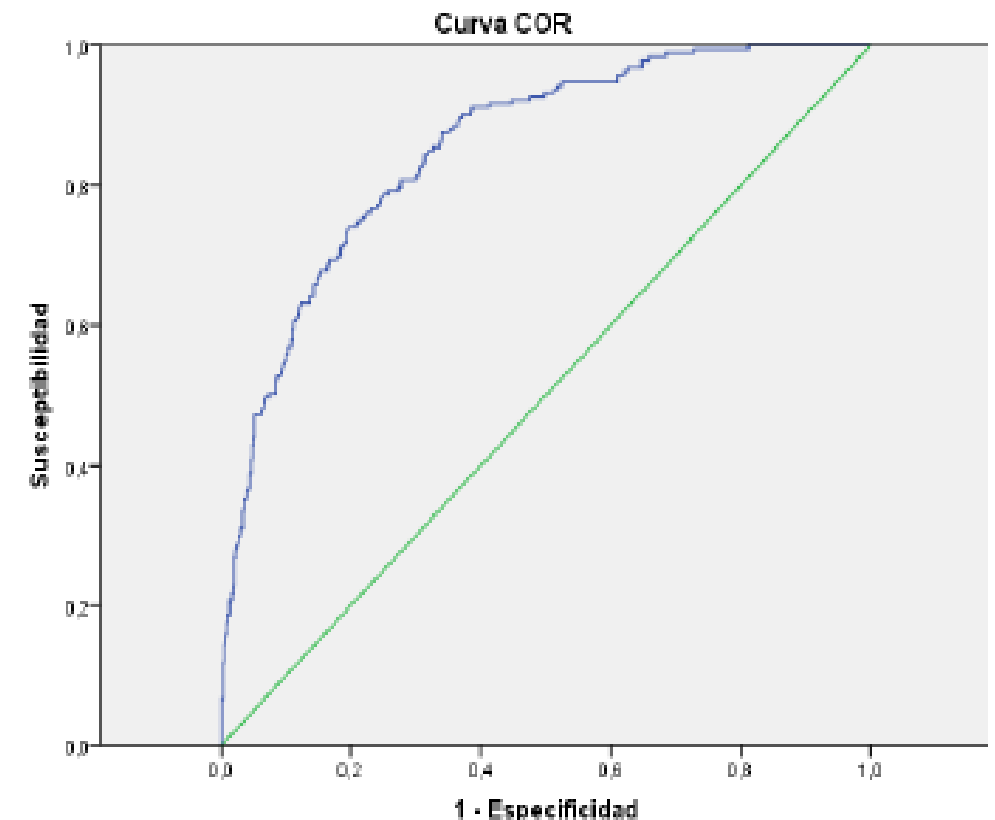
		Predicción		Total
		Mal Pagador	Buen Pagador	
Valor Real	Mal Pagador	800	200	1000
	Buen Pagador	500	1500	2000
Total		1300	1700	3000

2. Tipos de Aprendizaje

D. Indicadores Principales

ROC (Receiver Operating Characteristic)

- Es la representación de la relación entre la sensibilidad y especificidad.
- El indicador de la curva ROC es el **AUC**, sus valores van desde 0 a 1.



3. Taller de Modelo de Aprendizaje

Tipos de Aprendizaje - Modelos

Veamos algunos de los modelos mas usados que resuelven cada tipo de aprendizaje:

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means:

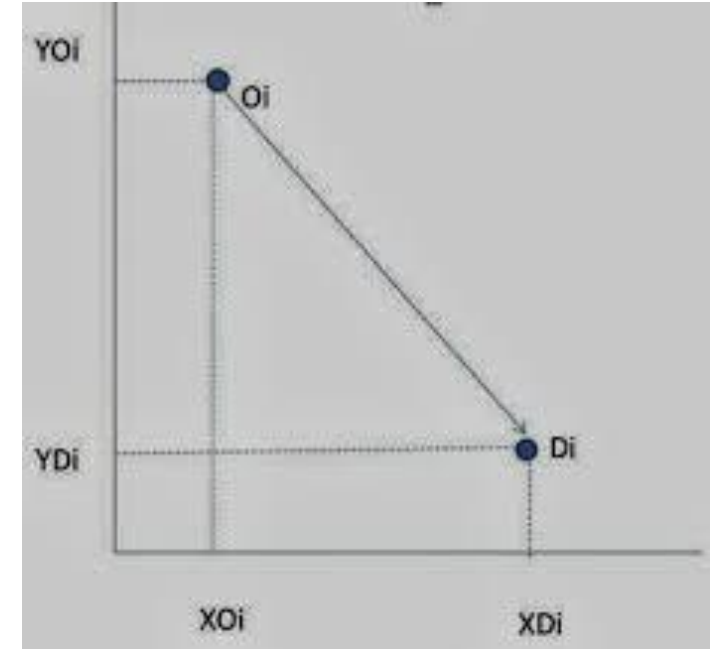
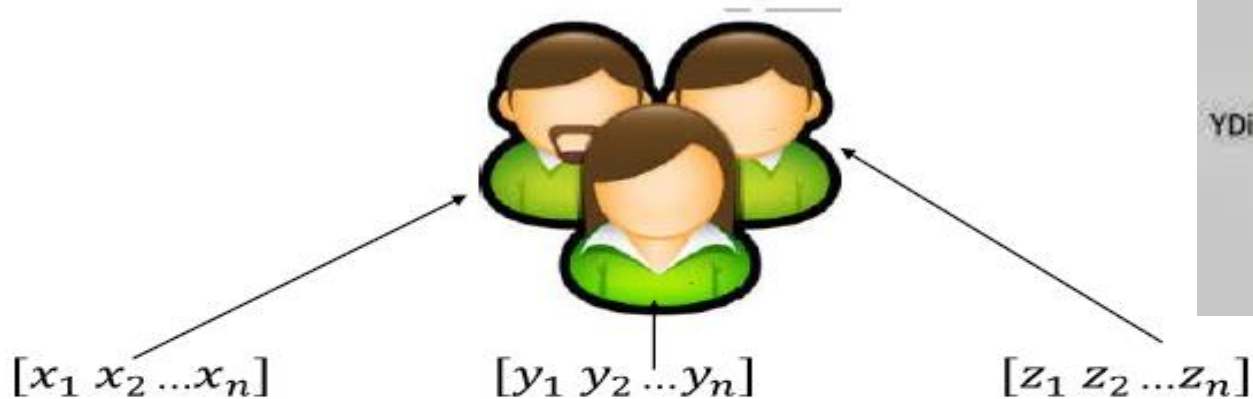
Es un algoritmo de agrupamiento iterativo que tiene como objetivo la **partición** de un conjunto de n observaciones en **k grupos** en el que cada observación pertenece al grupo cuyo valor medio es más **cercano**.

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means

Noción de similitud



Dada una representación vectorial de dos clientes x y y , podemos determinar el grado de similitud entre ellos a través del uso de una **métrica**.

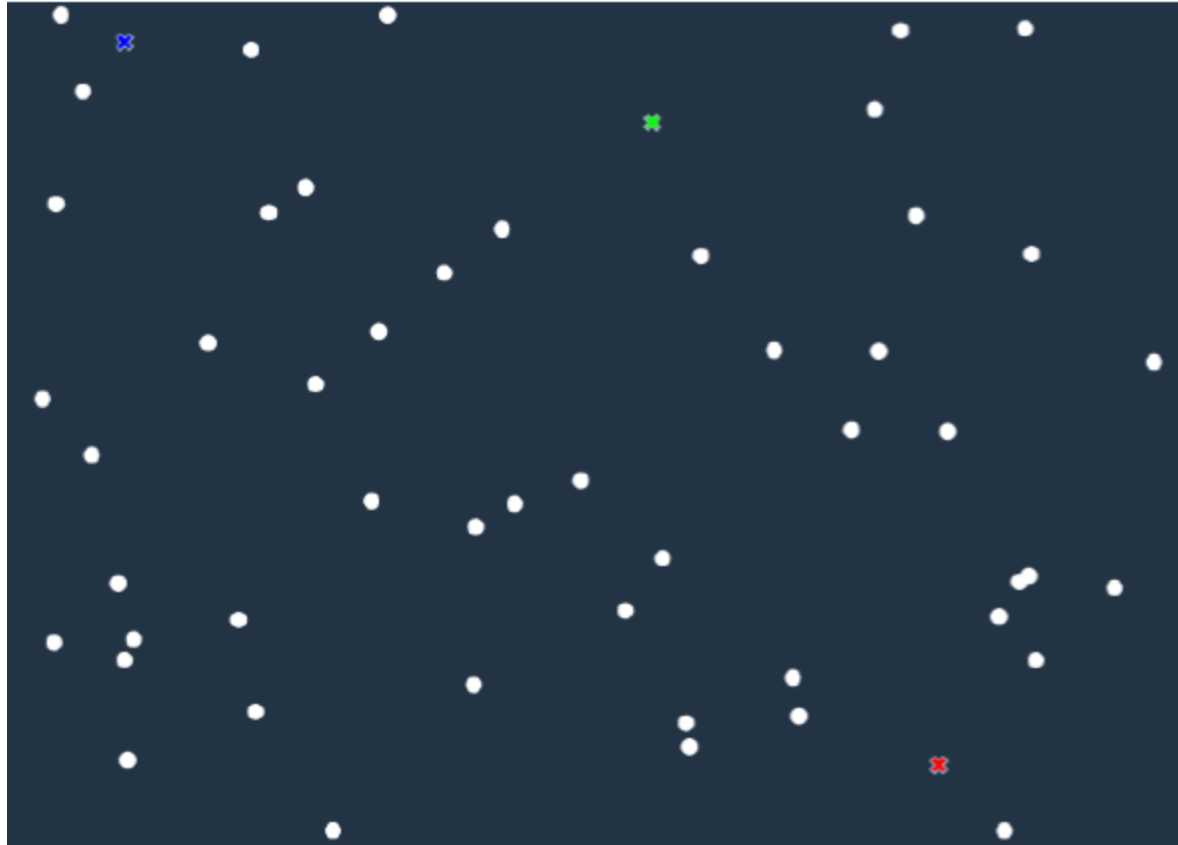
$$d(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}$$

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means:

Se define aleatoriamente
centros de clusters

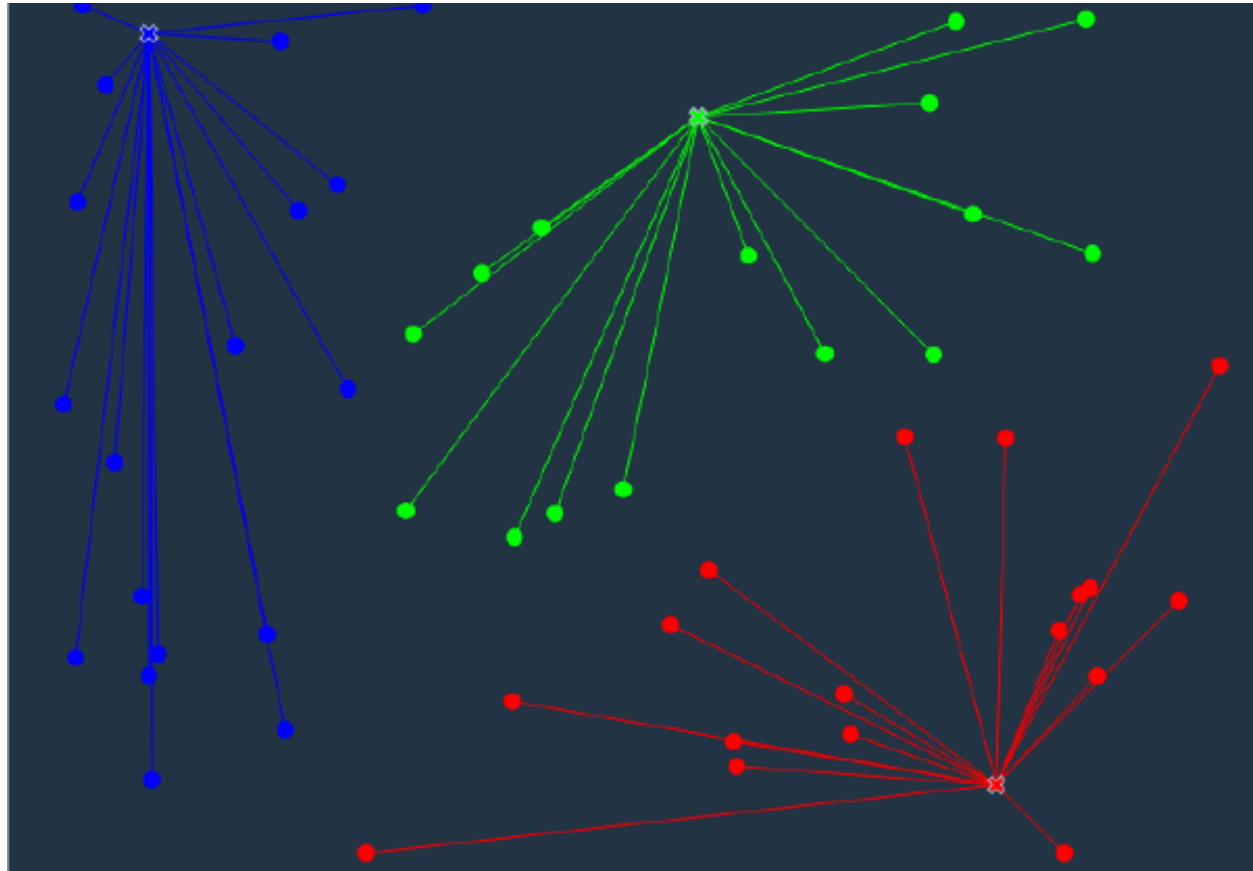


3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means:

Se asocia cada elemento en X a el **centroide** mas cercano

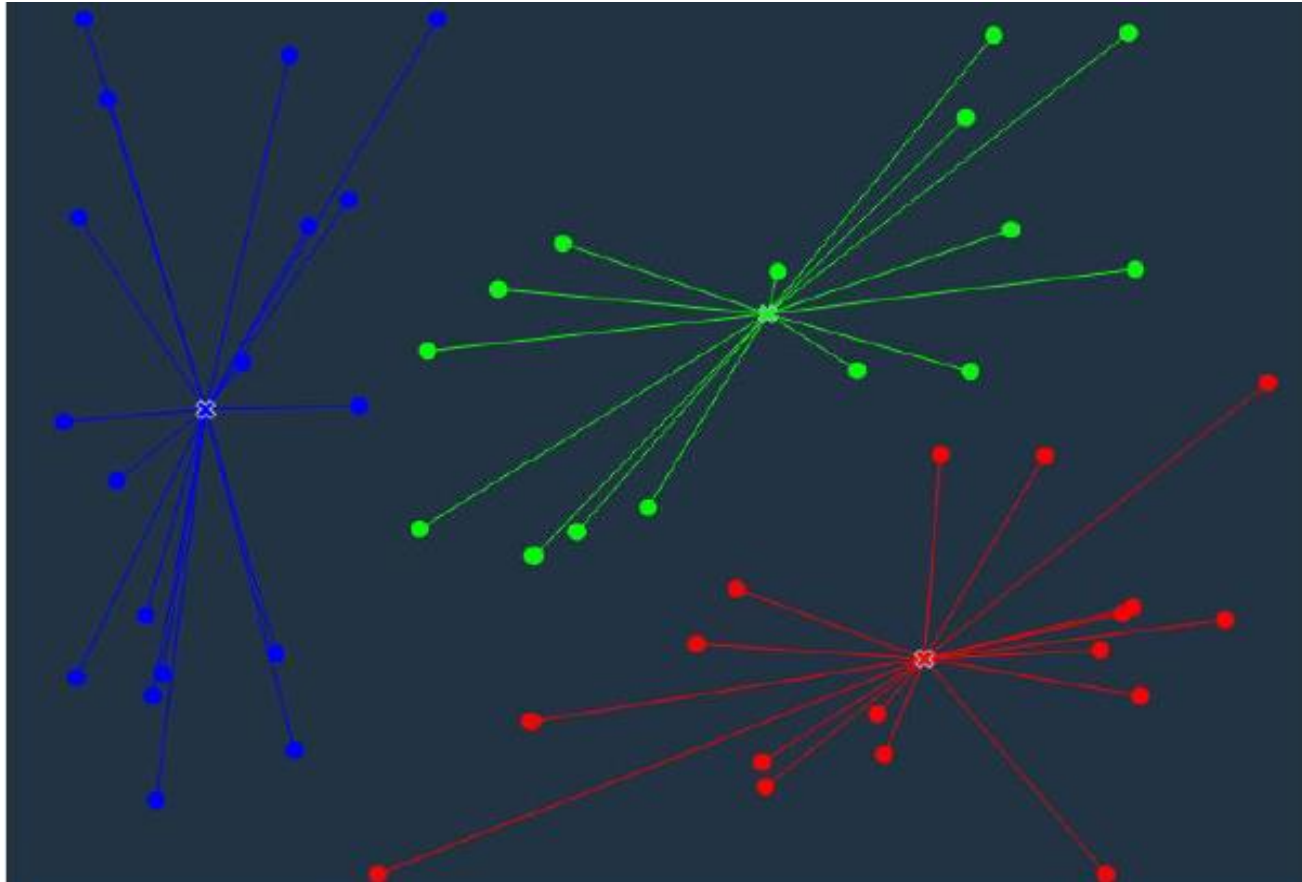


3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means:

Se recalcula los
centroides

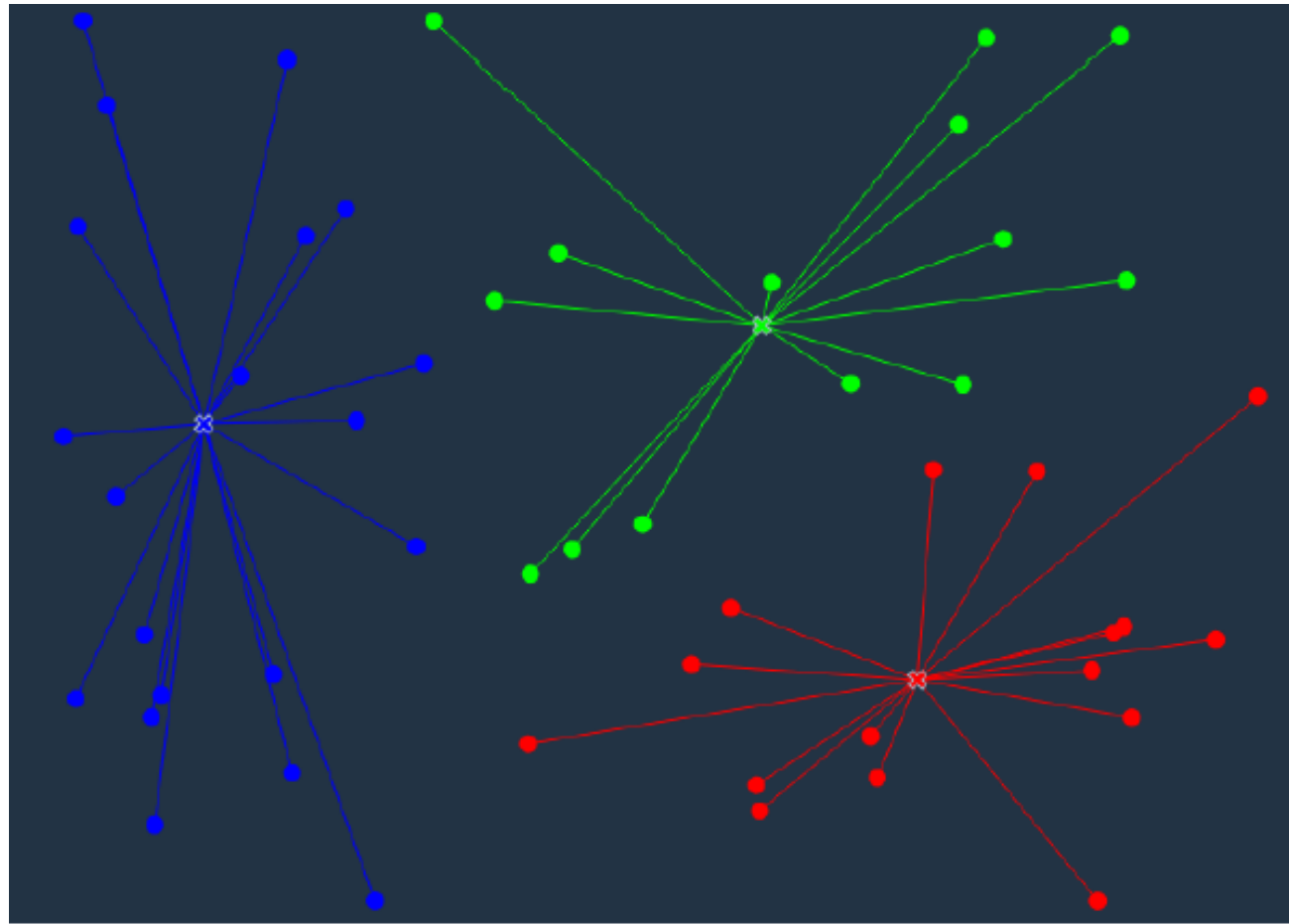


3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means:

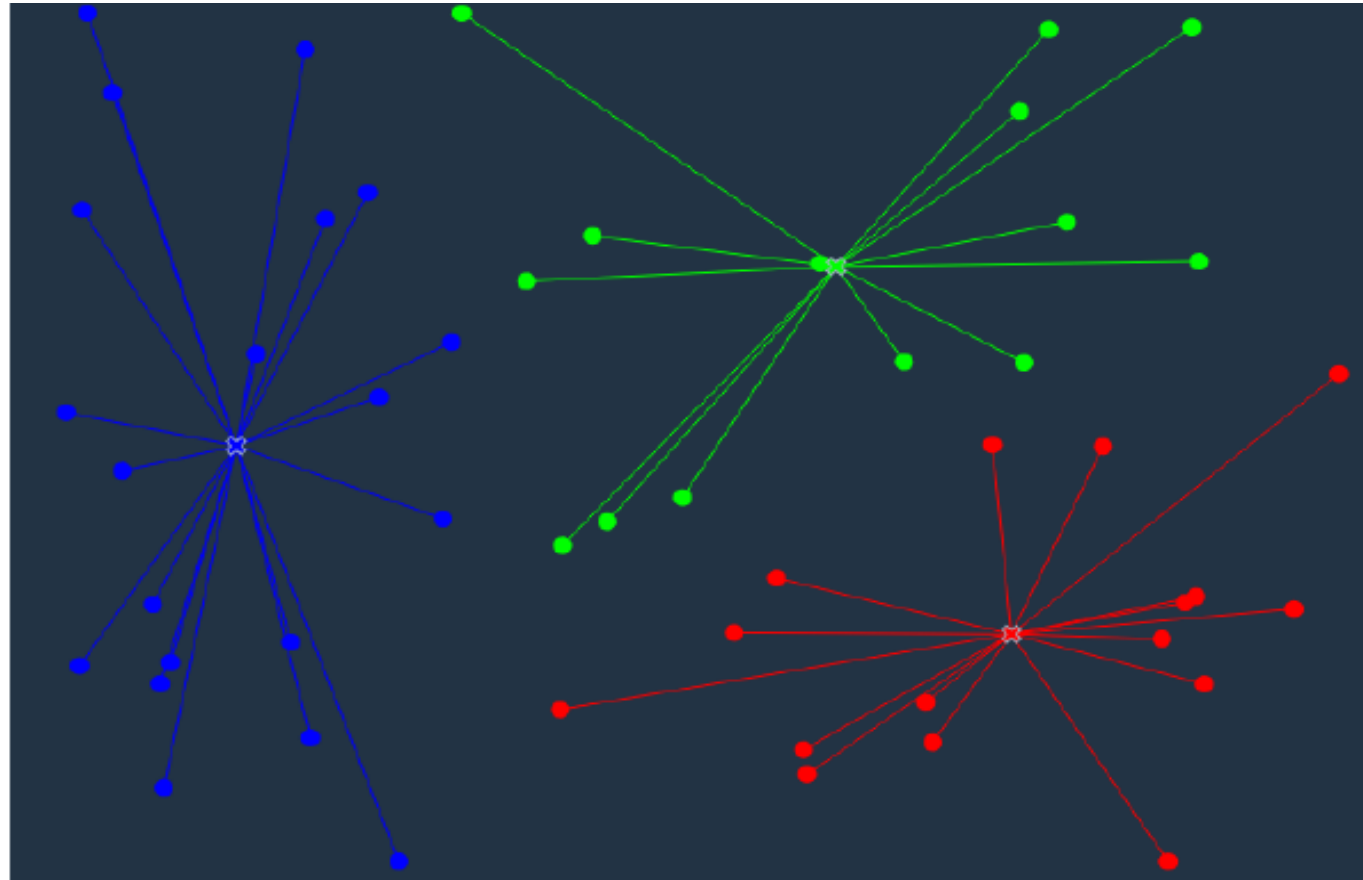
Se recalcula los centroides hasta que no exista variación entre las observaciones con los centroides



3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

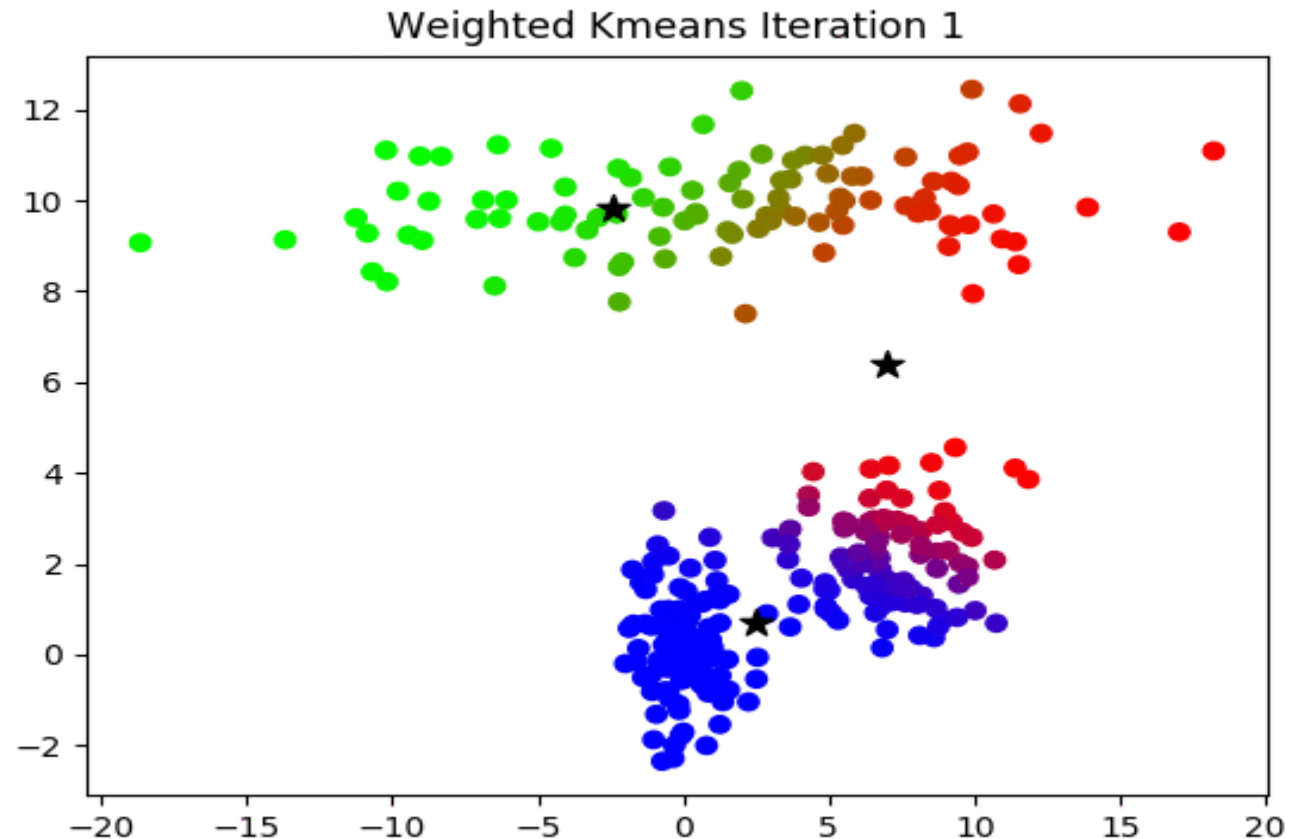
K Means:



3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado

K Means:



3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

1	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
2	Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
3	Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
4	Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
5	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
6	Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
7	Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
8	Duster 360	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
9	Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
10	Merc 230	22.8	4	140.8	95	3.69	3.15	22.8	1	0	4	2

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

- [. 1] mpg Miles/(US) gallon
- [. 2] cyl Number of cylinders
- [. 3] disp Displacement (cu.in.)
- [. 4] hp Gross horsepower
- [. 5] drat Rear axle ratio
- [. 6] wt Weight (1000 lbs)
- [. 7] qsec 1/4 mile time
- [. 8] vs Engine (0 = V-shaped, 1 = straight)
- [. 9] am Transmission (0 = automatic, 1 = manual)
- [.10] gear Number of forward gears
- [.11] carb Number of carburetors

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

```
In [1]: import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('kmeans').getOrCreate()
```

```
In [2]: from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt, numpy as np
```

```
In [3]: ds = spark.read.csv("/tmp/clasespark/mtcars.csv", header=True, nullValue="?", inferSchema=True)
ds.printSchema()
ds.count()
ds.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

In [12]:

```
cluster_pd_df = ds.toPandas()
import seaborn as sns

spearman_corr = cluster_pd_df.corr(method='spearman')
sns.heatmap(spearman_corr, annot=True, fmt=".2f")
```

Out[12]: <matplotlib.axes.AxesSubplot at 0x56cf590>



3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

```
In [9]: assembler = VectorAssembler(inputCols=["mpg", "cyl", "disp", "drat", "wt"], outputCol="features")
assem_data = assembler.transform(ds)

from pyspark.ml.feature import StandardScaler

scaler = StandardScaler(inputCol="features", outputCol="scaled_features", withStd=True, withMean=True)
scaler_model = scaler.fit(assem_data)
scaled_data = scaler_model.transform(assem_data)
scaled_data.show()
```

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	features	scaled_features
	Mazda RX4	21.0	6	160.0	110	3.9	2.62	16.46	0	1	4	4	[21.0,6.0,160.0,3...	[0.15088482464765...
	Mazda RX4 Wag	21.0	6	160.0	110	3.9	2.875	17.02	0	1	4	4	[21.0,6.0,160.0,3...	[0.15088482464765...
	Datsun 710	22.8	4	108.0	93	3.85	2.32	18.61	1	1	4	1	[22.8,4.0,108.0,3...	[0.44954344663064...
	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	[21.4,6.0,258.0,3...	[0.21725340731054...
	Hornet Sportabout	18.7	8	360.0	175	3.15	3.44	17.02	0	0	3	2	[18.7,8.0,360.0,3...	[-0.2307345256639...
	Valiant	18.1	6	225.0	105	2.76	3.46	20.22	1	0	3	1	[18.1,6.0,225.0,2...	[-0.3302873996582...
	Duster 360	14.3	8	360.0	245	3.21	3.57	15.84	0	0	3	4	[14.3,8.0,360.0,3...	[-0.9607889349556...

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

```
In [10]: clusters = 10

from pyspark.ml.clustering import KMeans

## Puedes utilizar también métodos para pasar los hiper-parámetros
kmeans = KMeans()\
    .setK(clusters)\
    .setMaxIter(1000)\
    .setFeaturesCol("scaled_features")\
    .setPredictionCol("prediction")

model = kmeans.fit(scaled_data)
```


3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

In [15]:

```
## Imprimimos los centros de los clusters  
for center in model.clusterCenters():  
    print(center)
```

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

```
In [16]: predict = model.transform(scaled_data)
         predict.show(1000)
         print("="*15)
```

3. Taller de Modelo de Aprendizaje

Aprendizaje No Supervisado – k-means

In [20]:

```
from pyspark.sql.functions import col

for i in range(clusters):
    predictionPerCol = predict.filter(col("prediction") == i)
    print("cluster {}".format(i))
    for c in predictionPerCol.select(col("features"), col("prediction")).collect():
        print(c)
    print("="*75)
```

cluster 0

Row(features=DenseVector([22.8, 4.0, 108.0, 3.85, 2.32]), prediction=0)

Row(features=DenseVector([24.4, 4.0, 146.7, 3.69, 3.19]), prediction=0)

Row(features=DenseVector([22.8, 4.0, 140.8, 3.92, 3.15]), prediction=0)

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

Este tipo de regresión se utiliza cuando existe solo una variable independiente X para una variable dependiente Y . Está definida por la siguiente ecuación lineal en su forma general:

$$Y = b_0 + b_1X + e$$

Donde:

Y Es la variable respuesta o la predicción de la variable Y dado un valor X .

b_0 Es el valor de Y cuando $X = 0$, es decir, es el valor de Y cuando la línea de regresión cruza el eje de las Y .

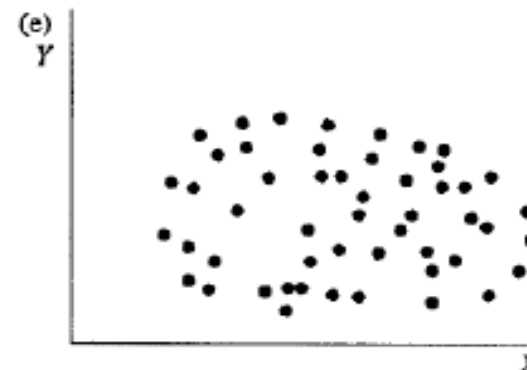
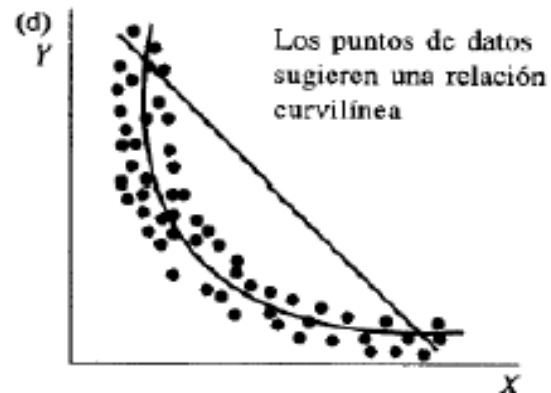
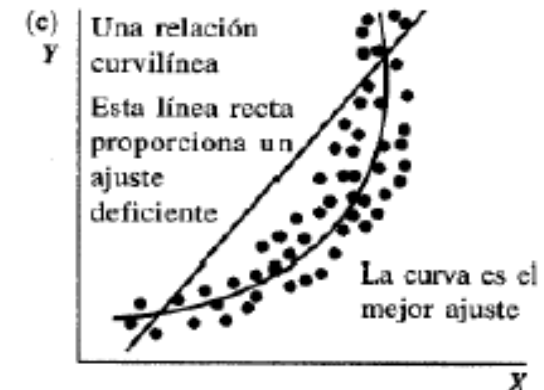
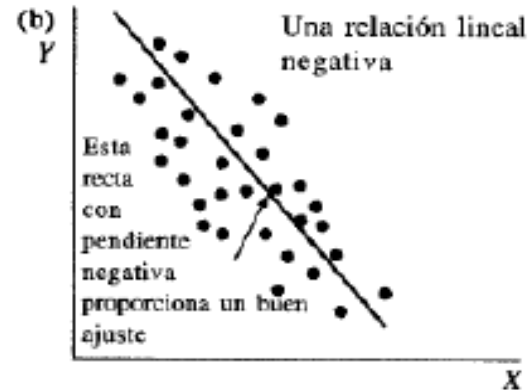
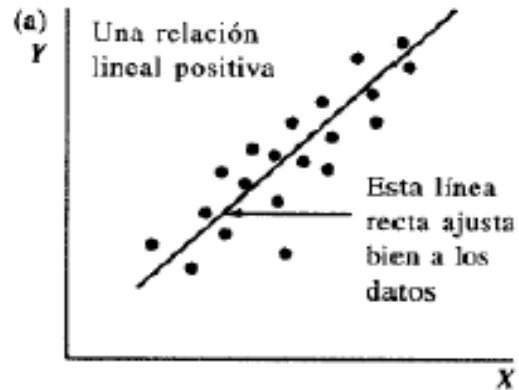
b_1 Es la pendiente de la línea, o la variación promedio en Y por cada variación de una unidad en X .

X Es cualquier valor seleccionado de la variable independiente X .

e Es el error de predicción

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

Horas de estudio (X)	Calificación obtenida (Y)
6	11
13	19
8	14
9	14
11	16
10	15
7	13
10	17



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

```
In [2]: import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('regresion_lineal_simple_1').getOrCreate()
import pandas
```

```
In [3]: from pyspark.ml.regression import LinearRegression
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
from pandas.tools.plotting import scatter_matrix
%matplotlib inline |
import matplotlib.pyplot as plt, numpy as np
```


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

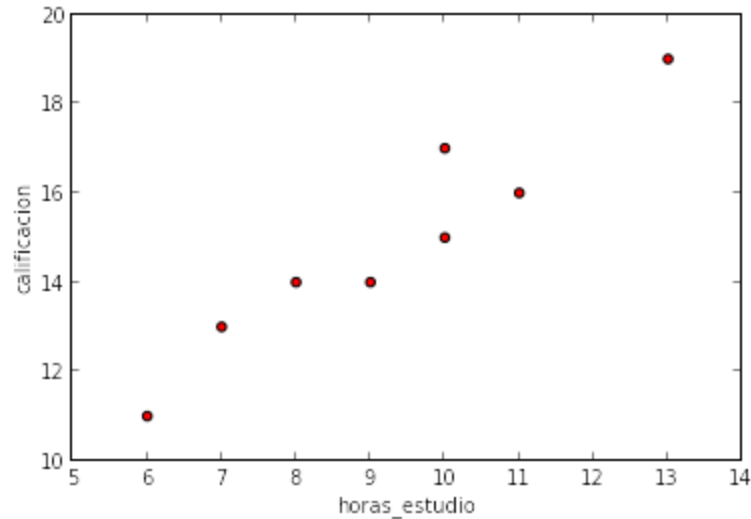
```
In [4]: trainning = spark.read.csv("/tmp/clasespark/regresion_lineal_simple.csv", header=True, nullValue="?", inferSchema=True)
        trainning.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

```
In [52]: # Interpretamos en base a un gráfico de dispersión
ufos_df = training.toPandas()
ufos_df.plot.scatter(x='horas_estudio', y='calificacion', color='Red')
|
```

Out[52]: <matplotlib.axes.AxesSubplot at 0x600d310>



3. Taller de Modelo de Aprendizaje

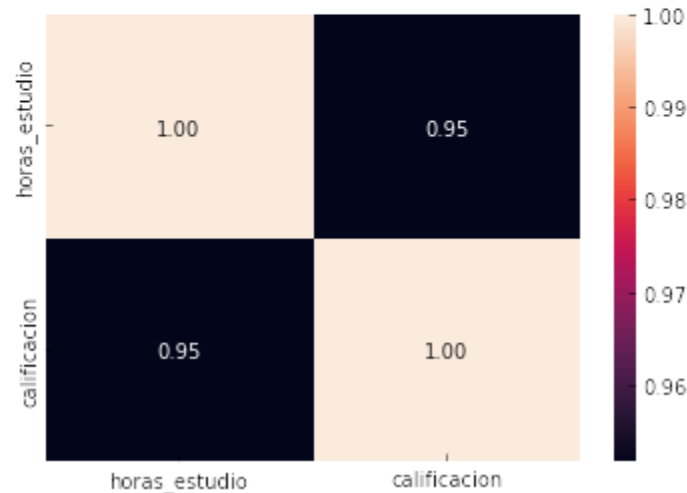
Aprendizaje Supervisado – Regresión Lineal

```
In [56]: # Revisamos la correlación que existen entre las variables
```

```
import seaborn as sns

spearman_corr = ufos_df.corr(method='spearman')
sns.heatmap(spearman_corr, annot=True, fmt=".2f")
```

```
Out[56]: <matplotlib.axes.AxesSubplot at 0x7ead210>
```



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

```
In [18]: assembler = VectorAssembler(inputCols=["horas_estudio"], outputCol="features")
assem_data = assembler.transform(trainning)
assem_data.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

```
In [16]: lr = LinearRegression(featuresCol="features",labelCol="calificacion",predictionCol='prediction')
```

```
In [19]: lrModel=lr.fit(assem_data)
```

```
In [20]: # obtenemos los coeficientes  
lrModel.coefficients|
```

```
Out[20]: DenseVector([1.0493])
```

```
In [21]: lrModel.intercept
```

```
Out[21]: 5.169014084507031
```

```
In [22]: training_summary=lrModel.summary  
training_summary.r2
```

```
Out[22]: 0.9116330636882519
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Lineal

Unas veces nuestro interés está en conocer si las dos variables están asociadas y medir hasta qué punto los cambios en una pueden explicarse por los cambios que ocurren en la otra. En tal caso tenemos un problema de **Correlación**.

Otras veces, cuando estamos seguros que existe un alto grado de asociación entre las dos variables, el análisis se encamina a cuantificar la relación existente con el fin de predecir cuáles serán los valores de la variable respuesta, en este caso tenemos un problema de **Regresión**.

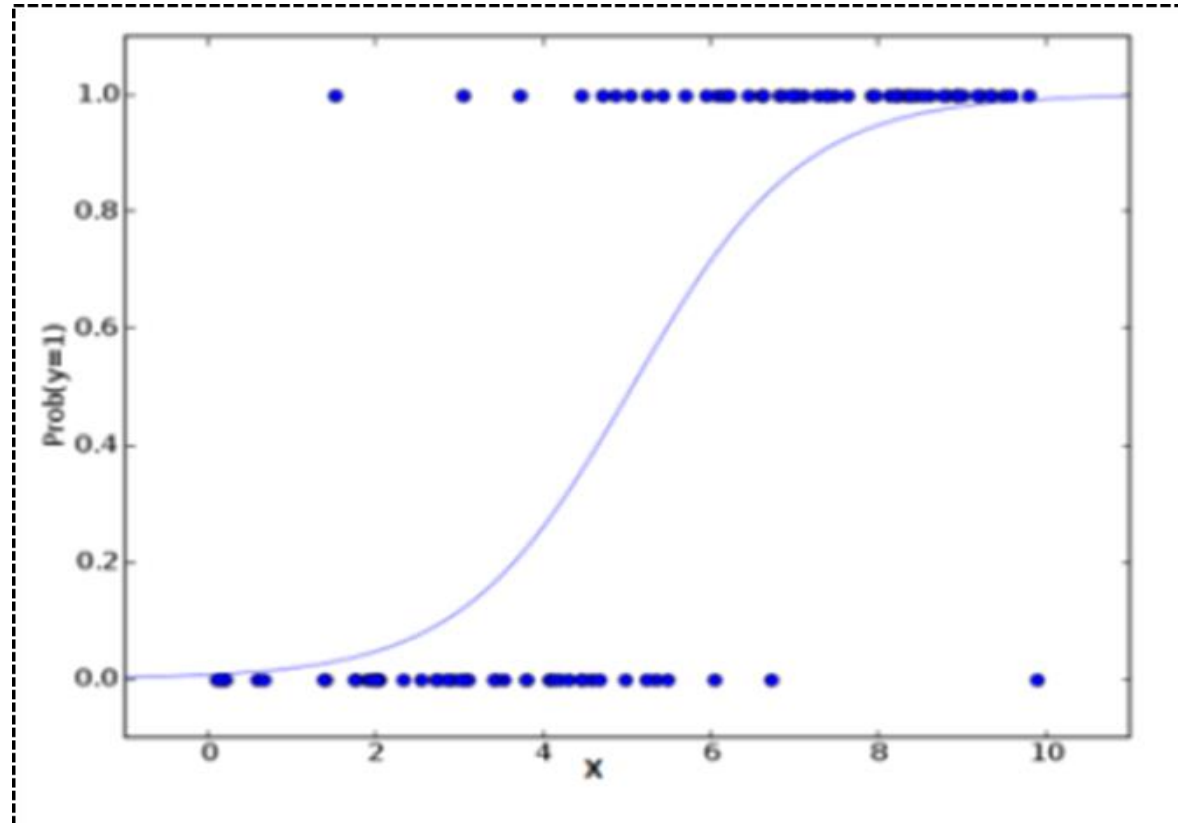
3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

- Regresión logística es un algoritmo de clasificación. Se usa para predecir un resultado binario (1/0, Sí / No, Verdadero / Falso) dado un conjunto de variables independientes.
- También puede pensar en la regresión logística como un caso especial de regresión lineal cuando la variable de resultado es categórica.

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

La representación matemática del modelo es la siguiente:

$$z_i = \log \frac{P_i}{1-P_i} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$$

z_i : Variable dependiente del modelo: “Moroso” y “ No Moroso”

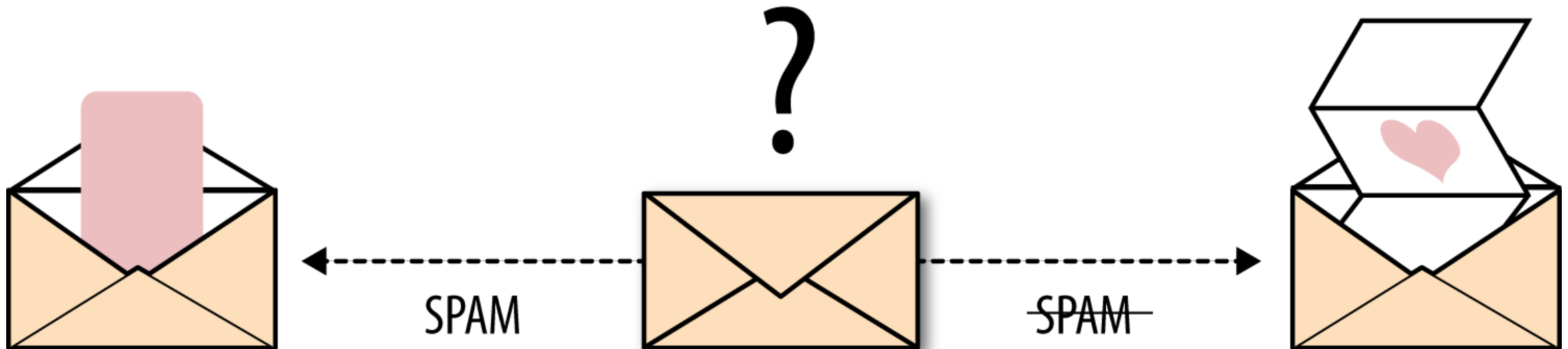
p_i : Probabilidad de que el cliente sea “Moroso”

β_i : Coeficientes del modelo (parámetros a estimar)

x_i : Variables explicativas del modelo

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística



Detección de SPAM es un típico ejemplo de un problema de clasificación binaria

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

0 10 20 30 40 50 60 70 80 90 100 110 120 130

1 ham Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...

2 ham Ok lar... Joking wif u oni...

3 spam Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's app

4 ham U dun say so early hor... U c already then say...

5 ham Nah I don't think he goes to usf, he lives around here though

6 spam FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send

7 ham Even my brother is not like to speak with me. They treat me like aids patent.

8 ham As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to cop

9 spam WINNER!! As a valued network customer you have been selected to receive a £900 prize reward! To claim call 09061701461. Claim code KL3

10 spam Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update C

11 ham I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.

12 spam SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 i

13 spam URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD

14 ham I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my

15 ham I HAVE A DATE ON SUNDAY WITH WILL!!

16 spam XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> <http://wap.xxxmobilemovieclub.com>

17 ham Oh k...i'm watching here:)

18 ham Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.

19 ham Fine if thats the way u feel. Thats the way its gota b

20 spam England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/ú1.20

21 ham Is that seriously how you spell his name?

22 ham I'm going to try for 2 months ha ha only joking

23 ham So ü pay first lar... Then when is da stock comin...

24 ham Aft i finish my lunch then i go str down lor. And 3 smth lor. U finish ur lunch already?

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

```
In [1]: import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('regresion_logistica').getOrCreate()
```

```
In [4]: from pyspark.sql.types import *    # Importamos los tipos de datos para definir el esquema

## El dataset sólo tiene dos columnas, el mensaje SMS (texto)
## y una etiqueta que indica si fué spam o no
spam_schema = StructType([
    StructField("spam", StringType(), True),
    StructField("message", StringType(), True)
])

ds = spark.read.csv("/tmp/clasespark/SMSSpamCollection", sep="\t", schema=spam_schema)
ds.show()
ds.show(truncate=False)
ds.printSchema()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

```
In [5]: from pyspark.ml.feature import StringIndexer
indexer = StringIndexer(inputCol="spam", outputCol="label")
indexed = indexer.fit(ds).transform(ds)
indexed.show()

from pyspark.ml.feature import Tokenizer
tokenizer = Tokenizer(inputCol="message", outputCol="tokens")
tokenized = tokenizer.transform(indexed)
tokenized.show()

from pyspark.ml.feature import HashingTF, IDF, VectorAssembler
hashingTF = HashingTF(inputCol="tokens", outputCol="tf")
tf_data = hashingTF.transform(tokenized)
tf_data.show()

idf = IDF(inputCol="tf", outputCol="idf")
idfModel = idf.fit(tf_data)
idf_data = idfModel.transform(tf_data)
idf_data.show()

assembler = VectorAssembler(inputCols=["idf"], outputCol="features")
assembled_data = assembler.transform(idf_data)
assembled_data.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

```
In [7]: ## Esto no habría que hacerlo, deberíamos usar una especie de magic loop, cross-validation, etc
training_data, test_data = assembled_data.randomSplit(weights=[0.7, 0.3], seed=12345)

from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(labelCol="label", featuresCol="features")
lrModel = lr.fit(assembled_data)

predict = lrModel.transform(test_data)
predict.select("spam", "probability", "prediction", "label").show(truncate=False)

from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator = BinaryClassificationEvaluator().setRawPredictionCol("prediction")
accuracy = evaluator.evaluate(predict)

"Test error: {}".format(1.0 - accuracy)
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Regresión Logística

```
In [7]: summary = lrModel.summary|  
summary
```

```
Out[7]: <pyspark.ml.classification.BinaryLogisticRegressionTrainingSummary at 0x222ce50>
```

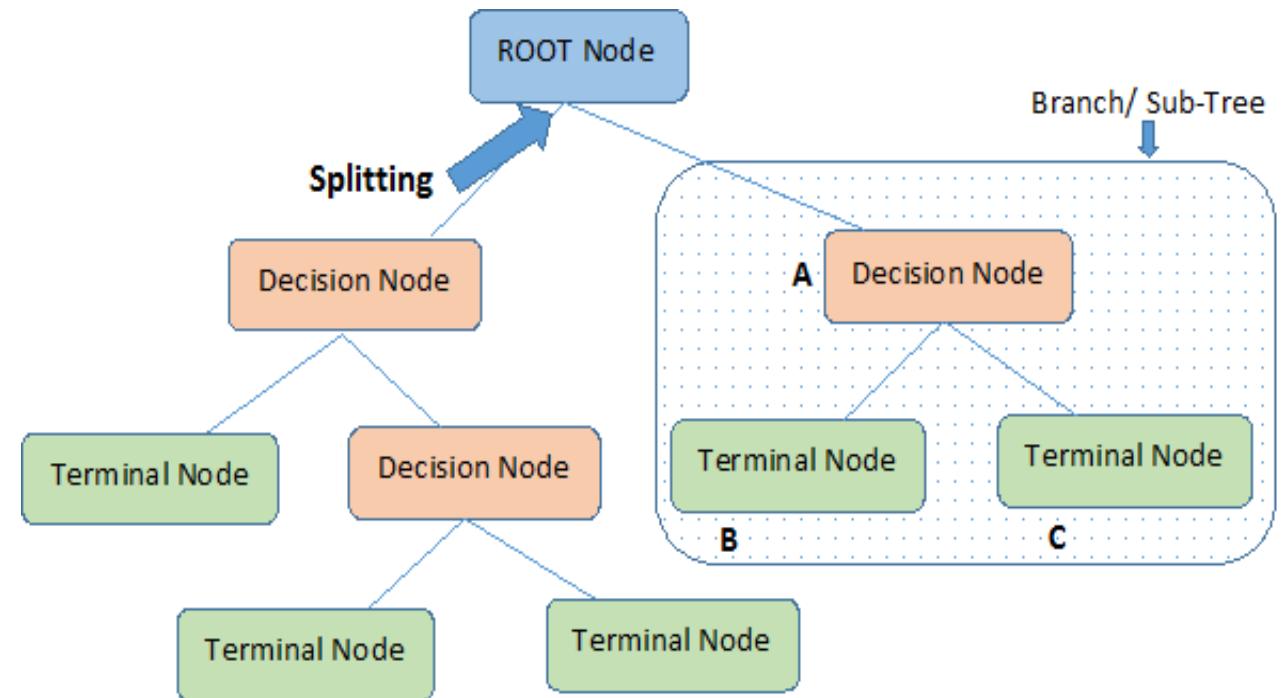
```
In [8]: print("ROC",summary.areaUnderROC)  
print("vars:",summary.roc)  
  
( 'ROC', 0.9999126399944089)  
( 'vars:', DataFrame[FPR: double, TPR: double])
```


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

Un Árbol de Decisión es simplemente **un proceso que nos ayuda a discriminar nuestras variables en base a una variable objetivo**. Consta de las siguientes partes:

- Root Node: Representa toda la población o muestra, y se divide en dos o más conjuntos homogéneos.
- Decision Node: Cuando un sub-nodo se divide en otros subnodos, se llama nodo de decisión.
- Terminal Node: Cuando los nodos no pueden dividirse.



Note:- A is parent node of B and C.

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

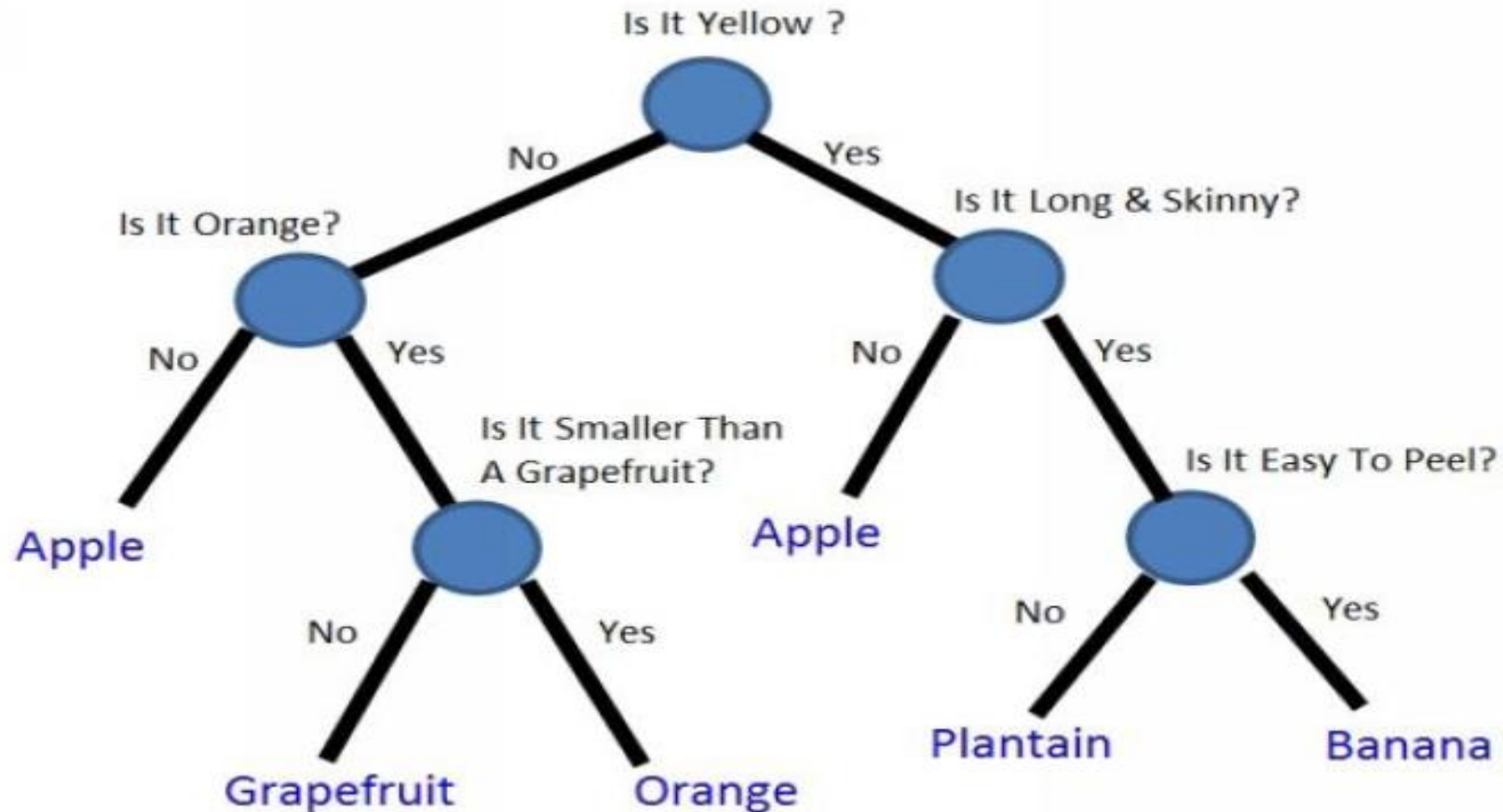
Por ejemplo, digamos que tiene una canasta de fruta frente a ti, e intentas enseñar a alguien que nunca había visto estos tipos de fruta antes, el cómo distinguirlos. ¿Como puedes hacerlo?

- Una forma sería mostrarle a la persona un montón de la fruta y **nombrarla**. Levante un plátano y diga "Esto es un plátano" o "Esto es una manzana". Ese método funcionaría con suficiente tiempo y suficientes ejemplos, y es básicamente cómo por lo general, enseñe a los niños qué cosas diferentes son.
- Una forma diferente sería **hacer un diagrama de flujo de preguntas** para pasar a determinar qué tipo de fruta es, su diagrama de flujo podría tener preguntas tales como:
 - ¿Es amarillo?
 - Si es así, ¿es largo y flaco?
 - Si es así, ¿es fácil de pelar?
 - Entonces es un plátano.
- Esto es efectivamente lo que es un árbol de decisión. Un gráfico para este árbol de decisión simple se muestra a continuación



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

- Como se puede observar el algoritmo permite desplegar visualmente un problema y organizar el trabajo de cálculos que deben realizarse.
- Busca encontrar las mejores particiones las cuales ayuden a discriminar la variable objetivo.
- **Pero** como hace para decidir/dividir cual categoría es la mas importante para realizar la partición:
 - Normalmente se usan los siguientes indicadores, error de clasificación, índice GINI y la entropía.

3. Taller de Modelo de Aprendizaje

D. Indicadores Principales

GINI

- Se usa para medir el nivel de desigualdad (varianza) existente entre la clase objetivo.
- Cuanto mayor es el valor de Gini, mayor es la homogeneidad de cada grupo.
- CART (Árbol de Clasificación y Regresión) usa el método Gini para crear divisiones binarias.
- Revisar: Entropia and Gain



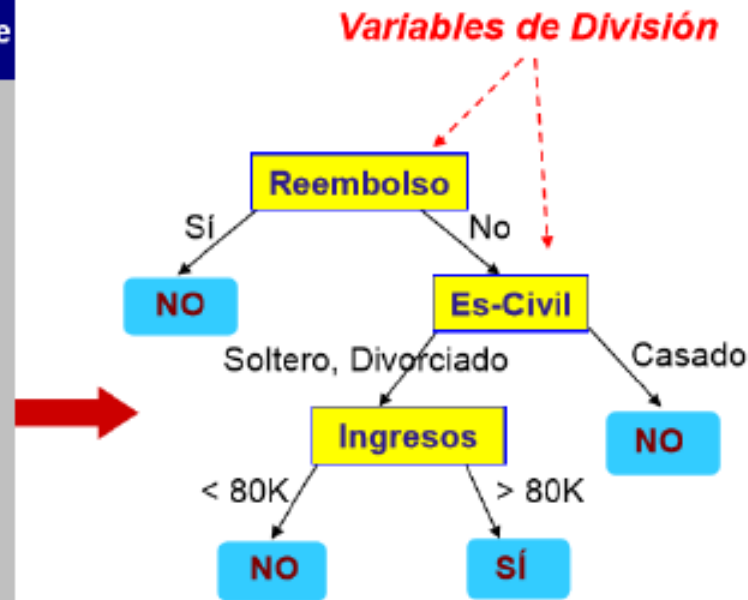
$$Gini = 1 - \sum_{i=1}^c (p_i)^2$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

	<i>categorica</i>	<i>categorica</i>	<i>continua</i>	<i>clase</i>
<i>Id</i>	Reembolso	Estado Civil	Ingresos Anuales	Fraude
1	Sí	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Sí	Casado	120K	No
5	No	Divorciado	95K	Sí
6	No	Casado	60K	No
7	Sí	Divorciado	220K	No
8	No	Soltero	85K	Sí
9	No	Casado	75K	No
10	No	Soltero	90K	Sí

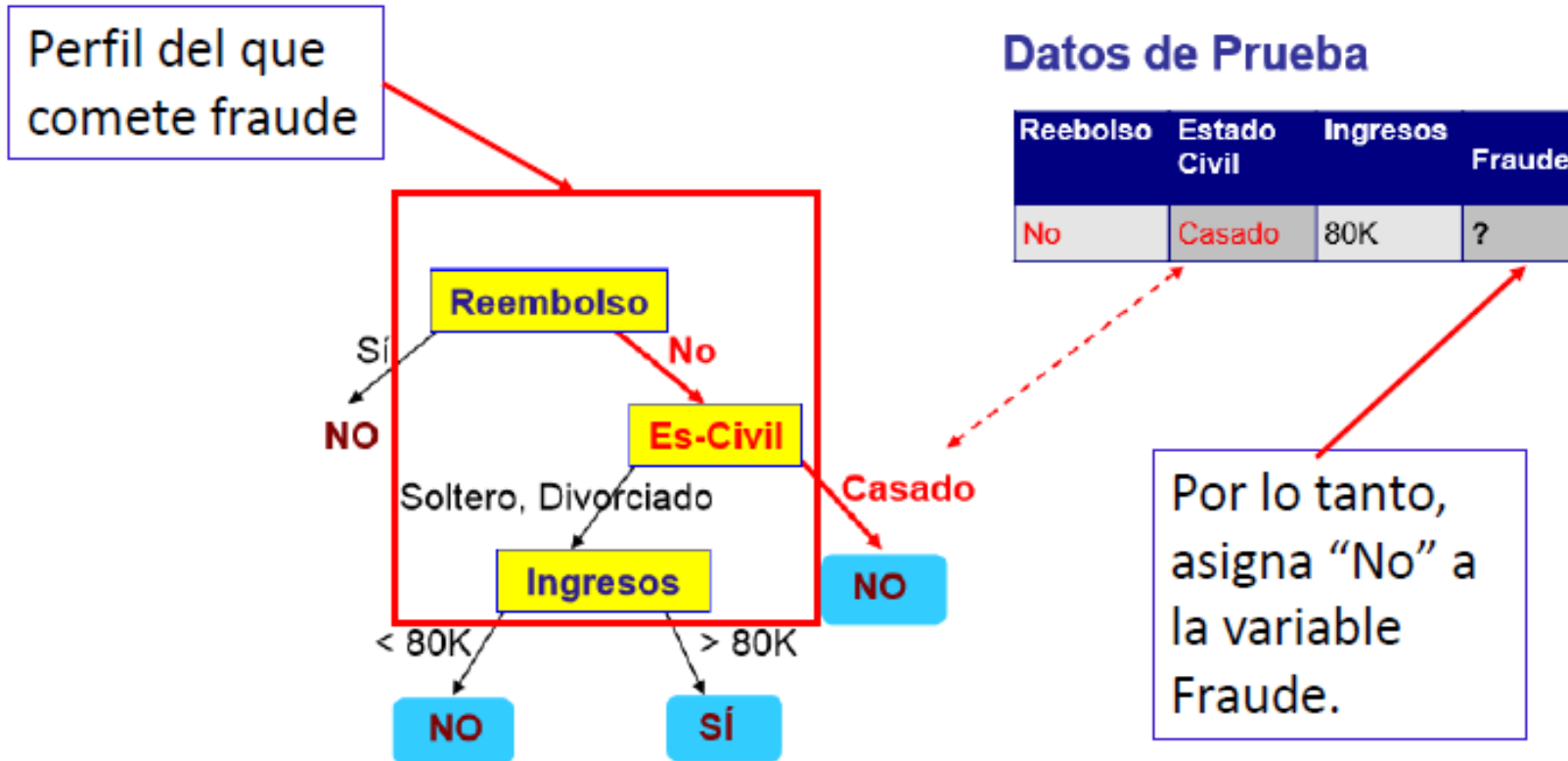
Tabla de Aprendizaje



Modelo: Árbol de Decisión

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

- Ejemplo:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

```
In [1]: import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('arbol_decision').getOrCreate()
import pandas as pd
```

```
In [21]: from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StandardScaler
```

```
In [3]: training = spark.read.csv("/tmp/clasespark/iris_ds.csv", header=True, nullValue="?", inferSchema=True)
training.show()
```

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species	label
5.1	3.5	1.4	0.2	setosa	1
4.9	3.0	1.4	0.2	setosa	1
4.7	3.2	1.3	0.2	setosa	1
4.6	3.1	1.5	0.2	setosa	1
5.0	3.6	1.4	0.2	setosa	1
5.4	3.9	1.7	0.4	setosa	1
4.6	3.4	1.4	0.3	setosa	1
5.0	3.4	1.5	0.2	setosa	1
4.4	2.9	1.4	0.2	setosa	1
4.9	3.1	1.5	0.1	setosa	1

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

```
In [39]: training = spark.read.csv("/tmp/clasespark/iris_ds.csv", header=True, nullValue="?", inferSchema=True)
training.show()
```

```
In [40]: assembler = VectorAssembler(inputCols=["Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width"], outputCol="features")
assem_data = assembler.transform(training)
assem_data.show()
```

```
In [8]: train_scaler = StandardScaler(inputCol="features", outputCol="scaled_features", withStd=True, withMean=True)
train_scaler_model = train_scaler.fit(assem_data)
scaled_data = train_scaler_model.transform(assem_data)
scaled_data.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

```
In [12]: data_train, data_test = scaled_data.randomSplit(weights=[0.7, 0.3], seed=1234)
```

```
In [13]: dt = DecisionTreeClassifier(labelCol="label", featuresCol="scaled_features")  
dtModel = dt.fit(data_train)
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Árboles de Decisión

```
In [15]: predictions = dtModel.transform(data_test)
         predictions.show()
```

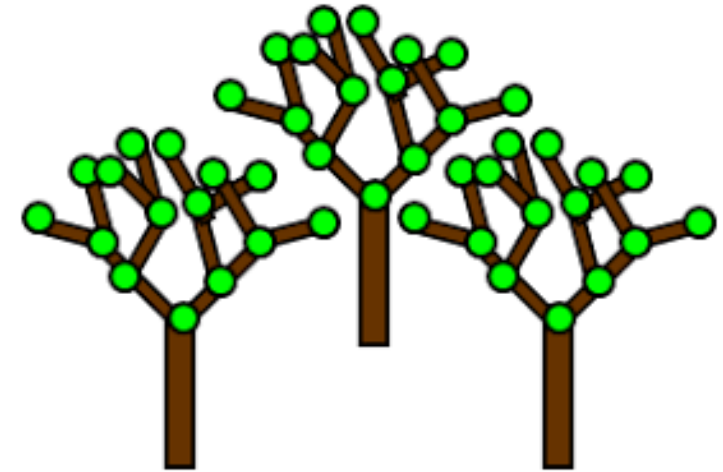
```
In [18]: predictions.createOrReplaceTempView("prediction_rf")
```

```
In [20]: spark.sql("""
           select label, prediction, count(1) as cantidad
           from prediction_rf
           group by label, prediction|
           """).show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Random Forest

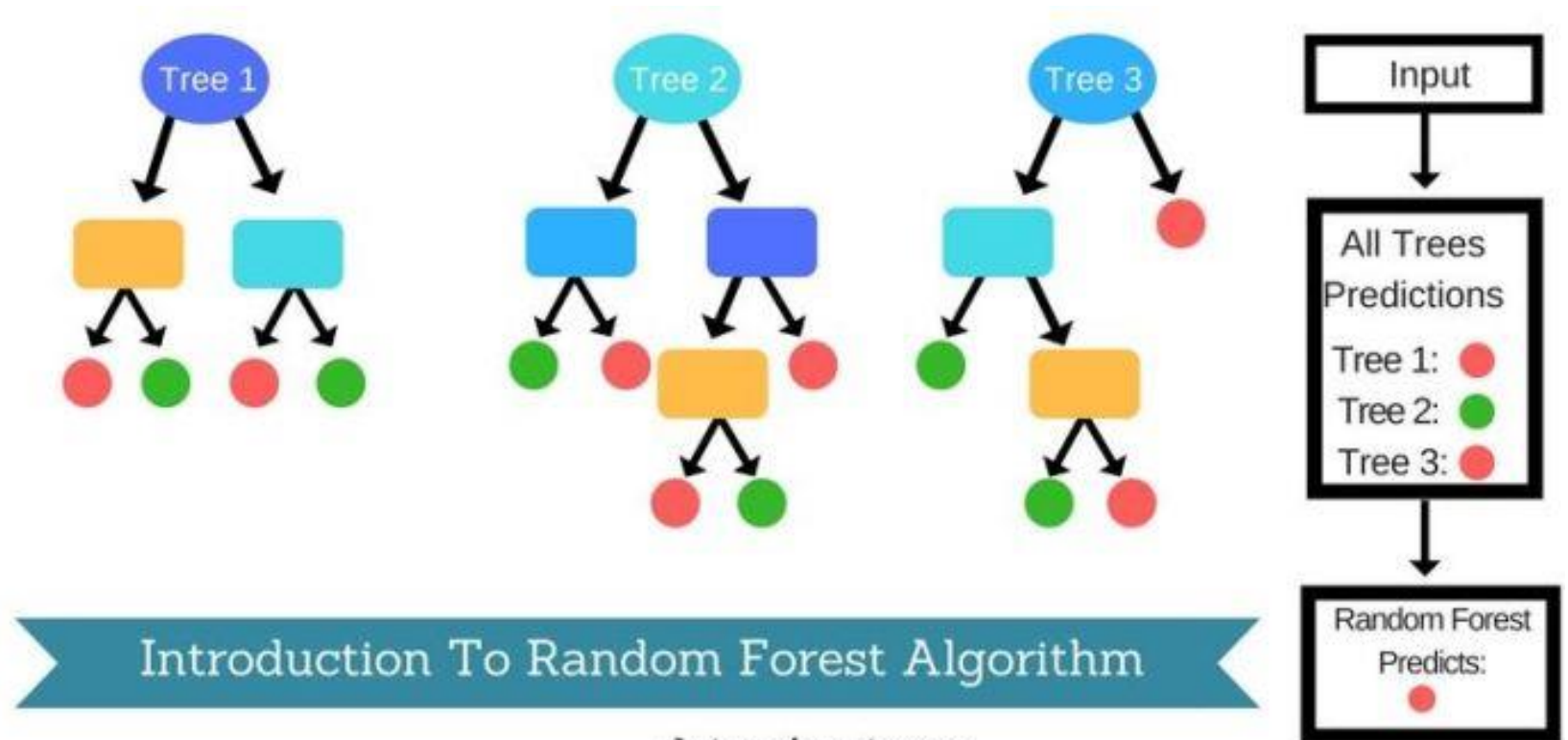
- Random forest es un método que combina una gran cantidad de arboles de decisión independientes sobre conjuntos de datos aleatorios con igual distribución.
- Cada árbol es formado por $n-1$ variables y una muestra de datos.
- Luego de obtener el resultado de cada árbol se procede a una votación, donde la clase mas votada es la que será la clase



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Random Forest

- Gráficamente se representa de la siguiente manera.



Introduction To Random Forest Algorithm

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Random Forest

```
In [1]: import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('random_forest').getOrCreate()
import pandas as pd
```

```
In [12]: from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StandardScaler
```

```
In [13]: training = spark.read.csv("/tmp/clasespark/iris_ds.csv", header=True, nullValue="?", inferSchema=True)
training.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Random Forest

```
In [4]: assembler = VectorAssembler(inputCols=["Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width"], outputCol="features")
assem_data = assembler.transform(trainning)
assem_data.show()
```

```
In [8]: train_scaler = StandardScaler(inputCol="features", outputCol="scaled_features", withStd=True, withMean=True)
train_scaler_model = train_scaler.fit(assem_data)
scaled_data = train_scaler_model.transform(assem_data)
scaled_data.show()
```


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Random Forest

```
In [17]: data_train, data_test = scaled_data.randomSplit(weights=[0.7, 0.3], seed=4321)
```

```
In [19]: rf = RandomForestClassifier(labelCol="label", featuresCol="features")  
rfModel = rf.fit(data_train)
```

```
In [21]: predictions = rfModel.transform(data_test)  
predictions.show()
```

3. Taller de Modelo de Aprendizaje

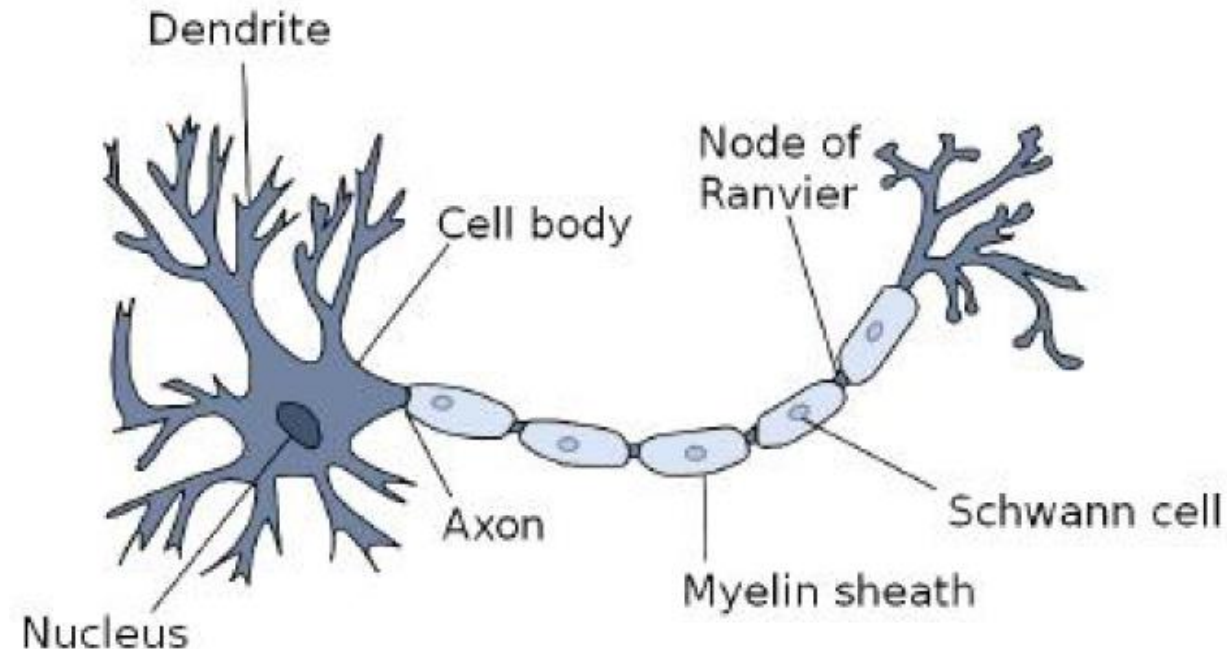
Aprendizaje Supervisado – Random Forest

```
In [22]: predictions.createOrReplaceTempView("prediction_rf")
spark.sql("""
        select label, prediction, count(1) as cantidad
        from prediction_rf
        group by label, prediction
        """).show()
```

3. Taller de Modelo de Aprendizaje

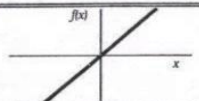
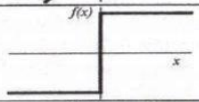
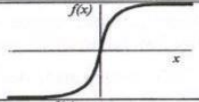
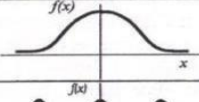
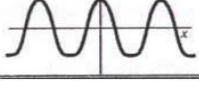
Aprendizaje Supervisado – Redes Neuronales Perceptron

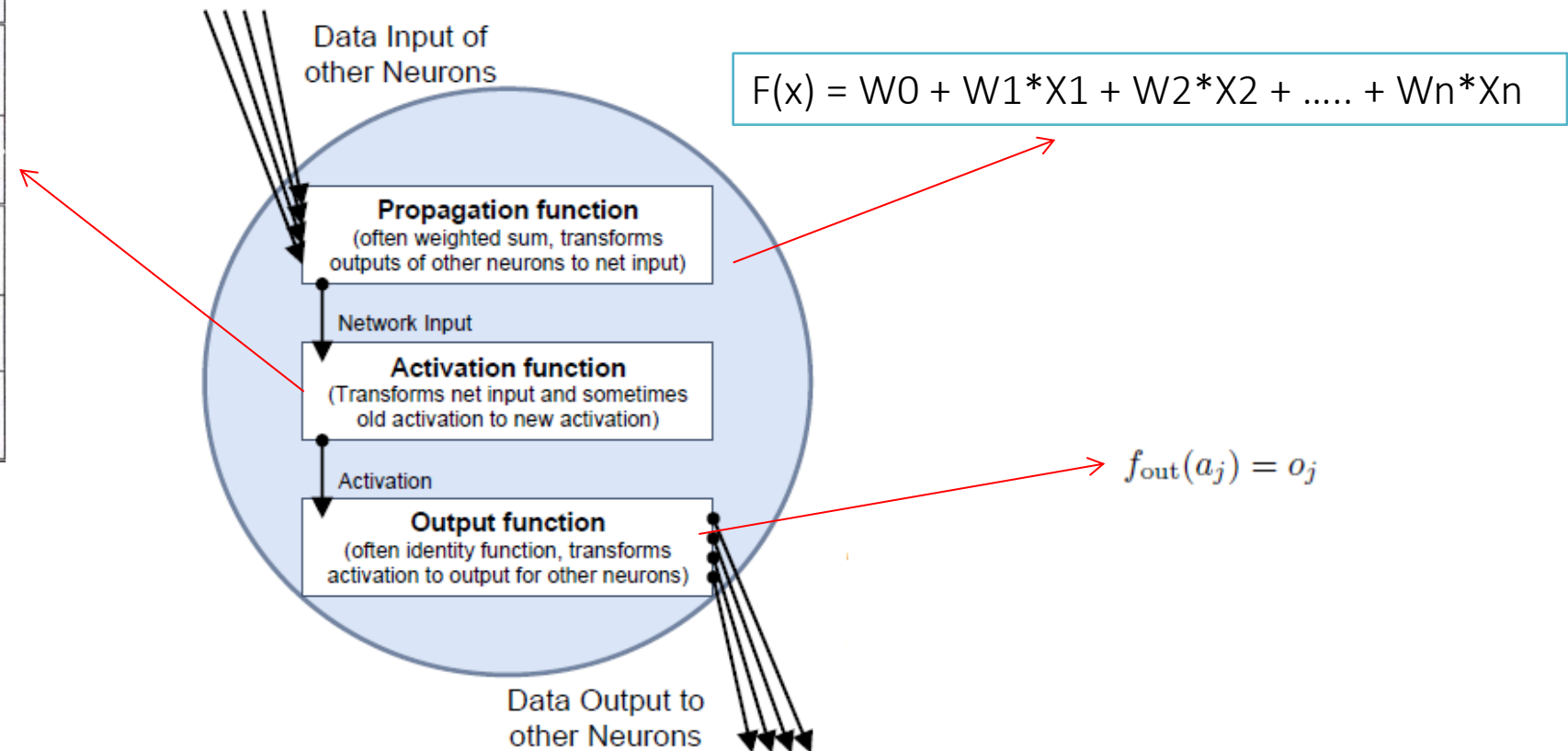
- Es una abstracción del proceso biológico de redes neuronales.
- Típicamente es definida como una unidad de procesamiento de información.



3. Taller de Modelo de Aprendizaje

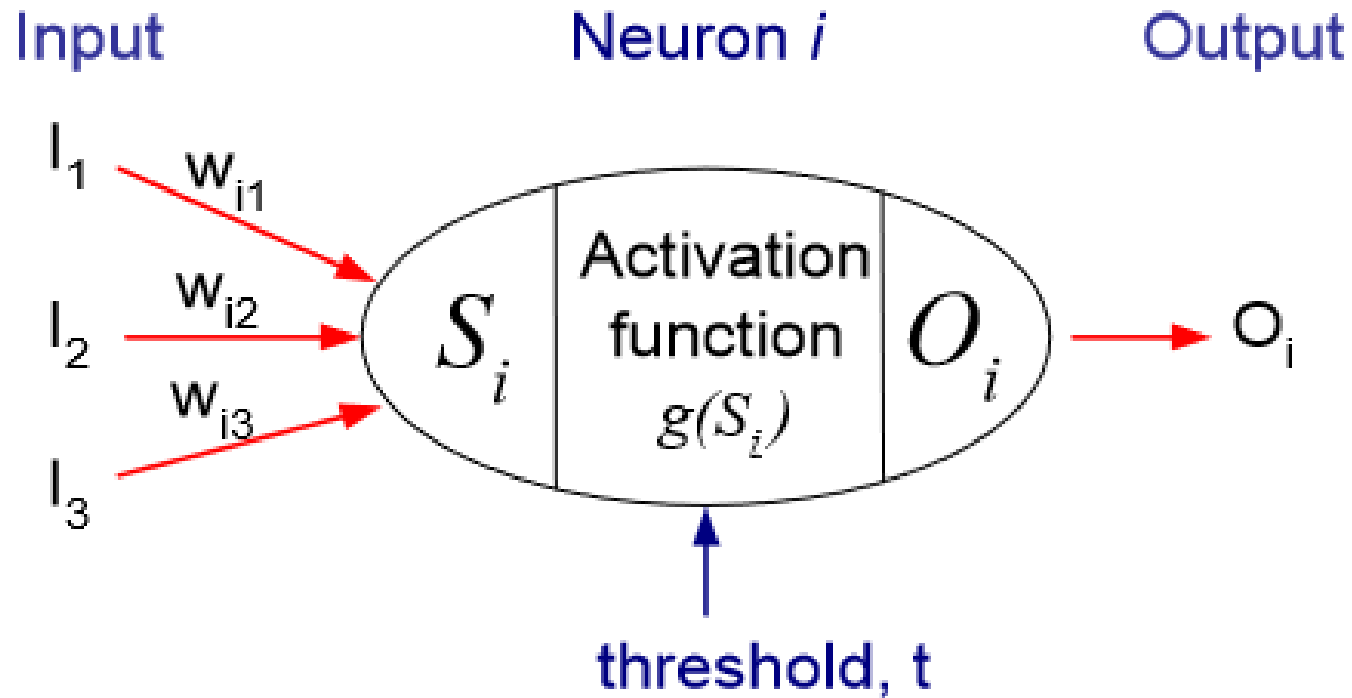
Aprendizaje Supervisado – Redes Neuronales Perceptron

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \sin(\omega x + \varphi)$	$[-1, +1]$	



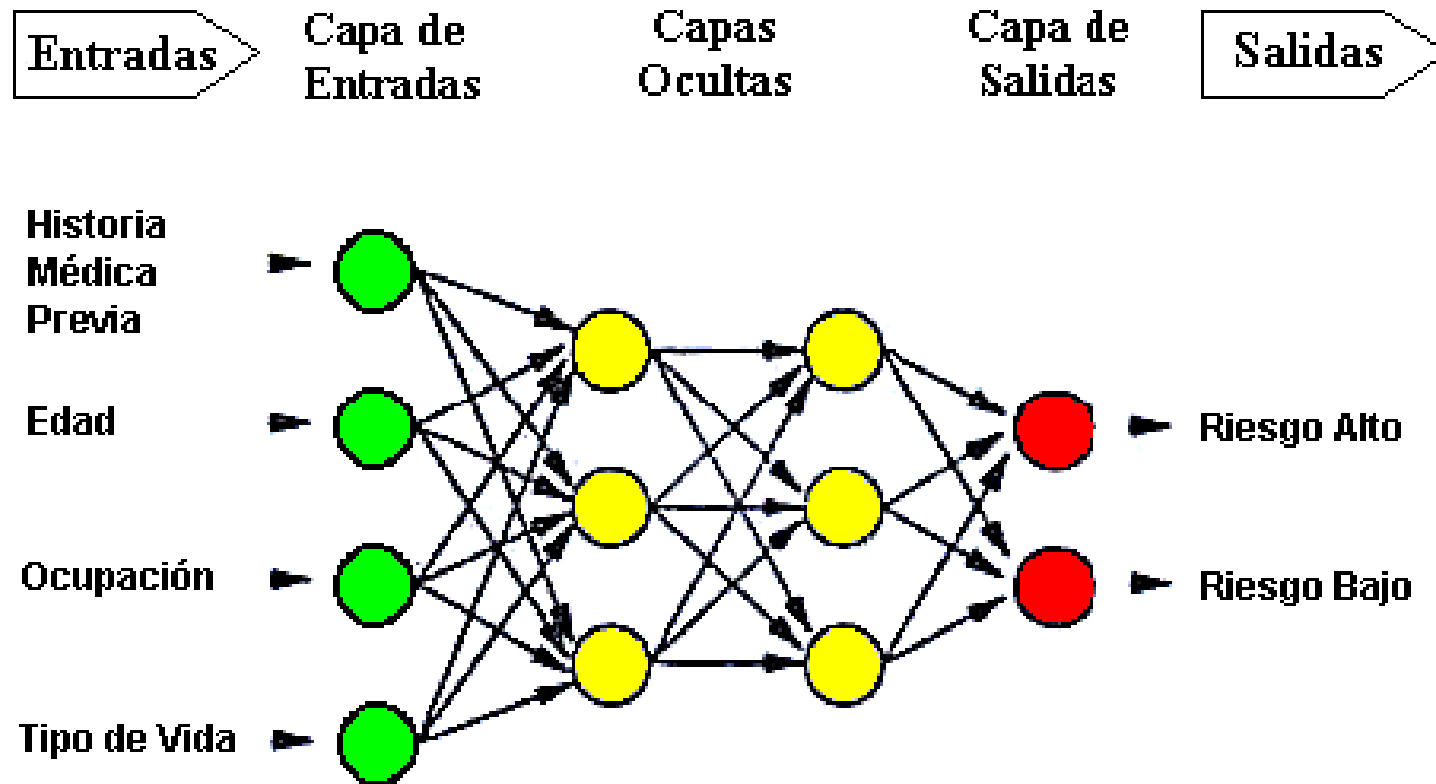
3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

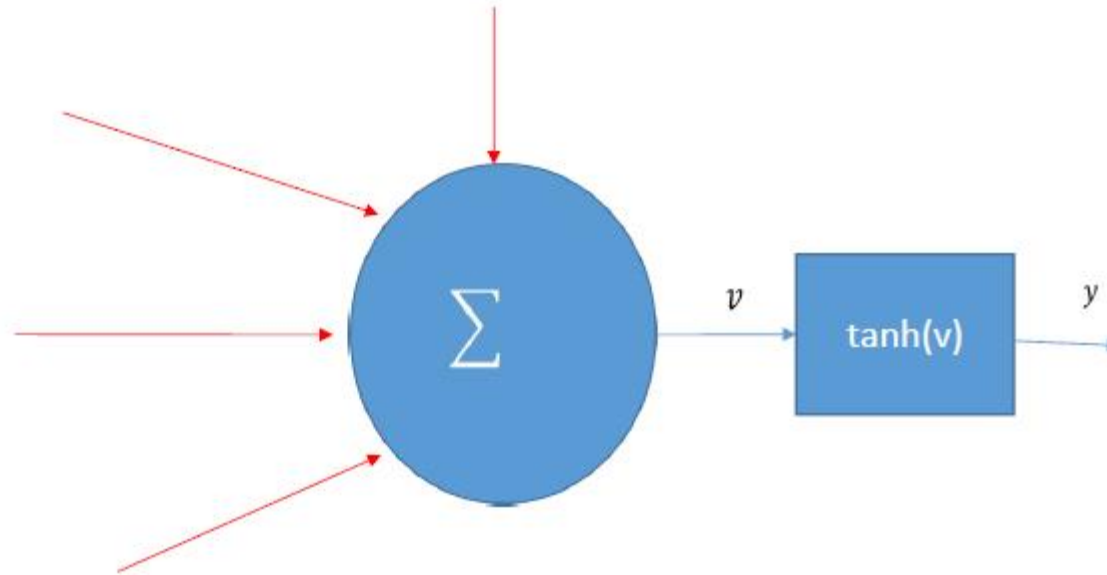


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

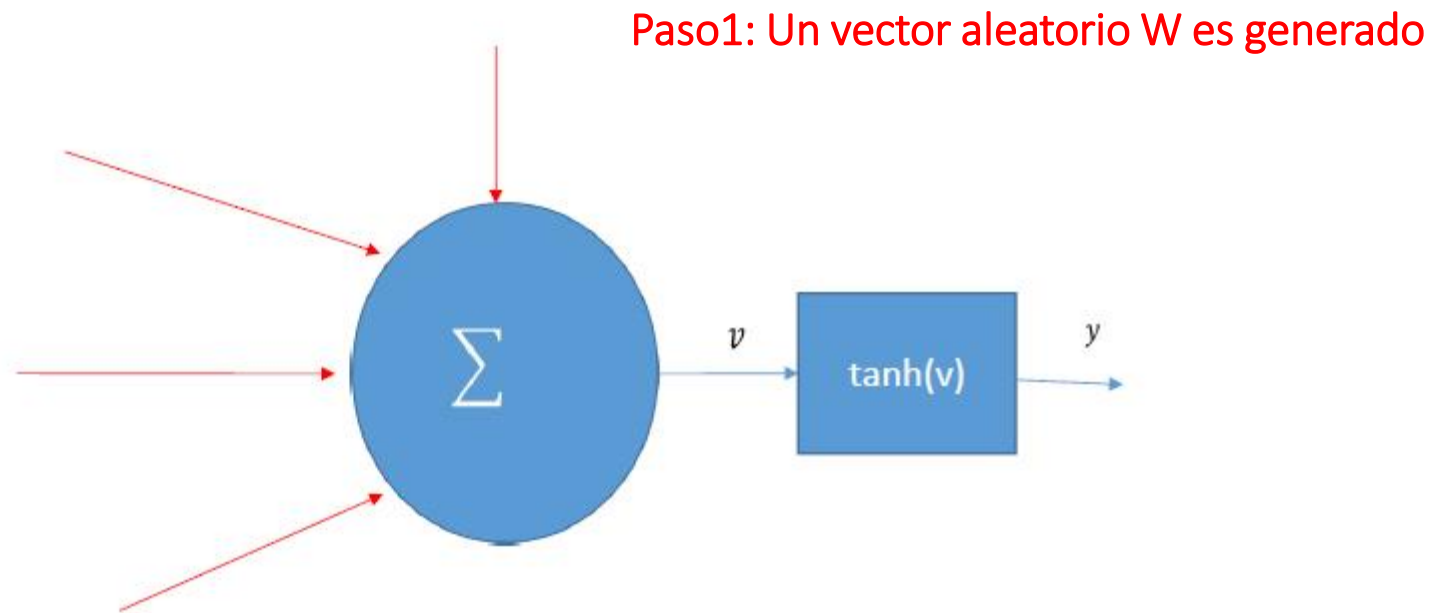


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

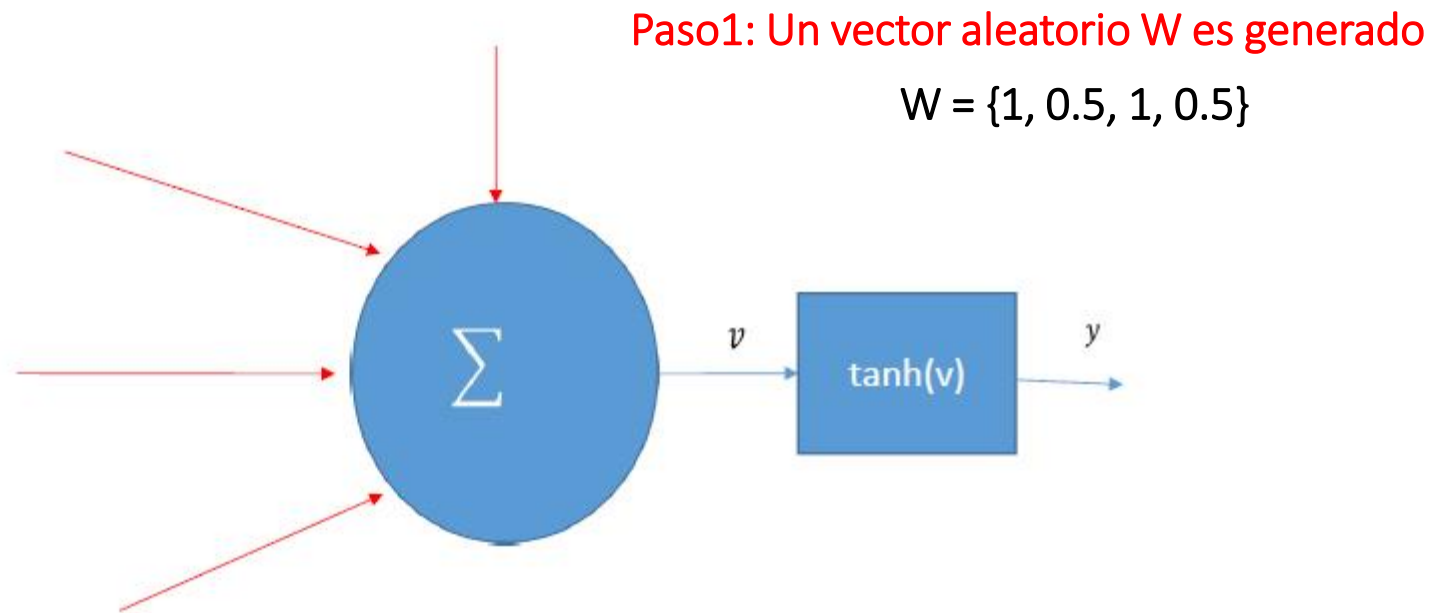


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

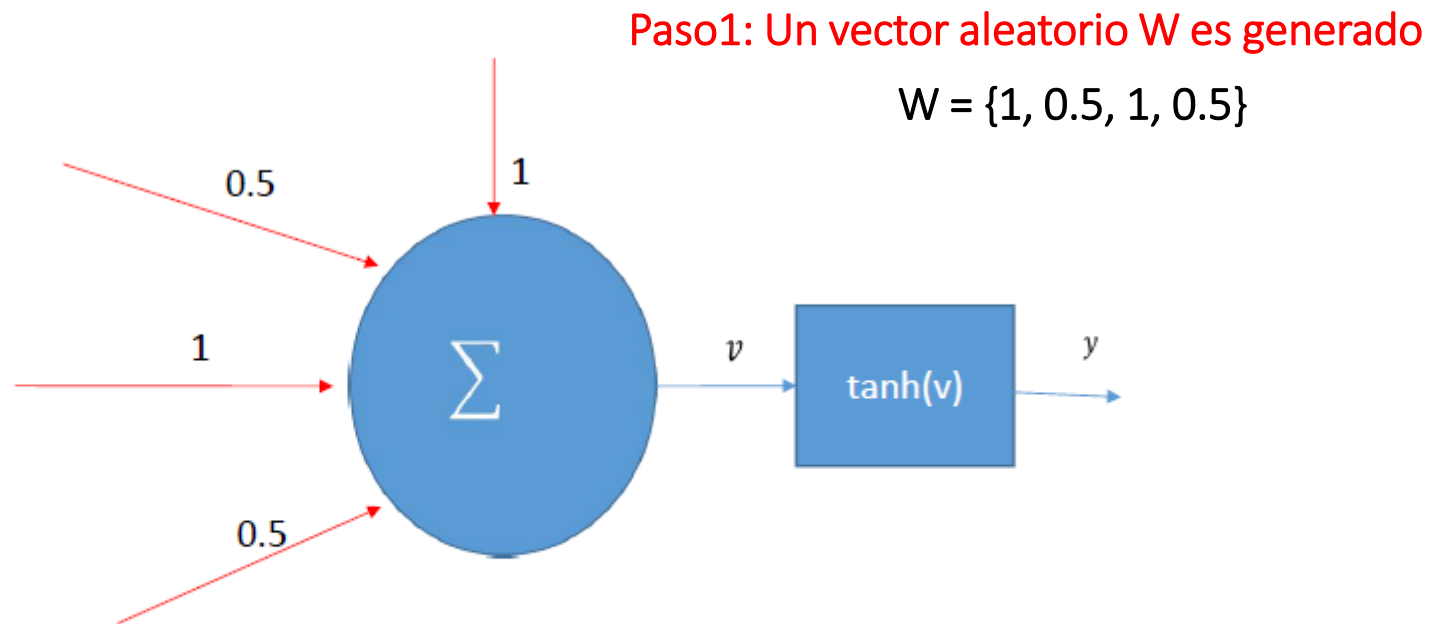


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

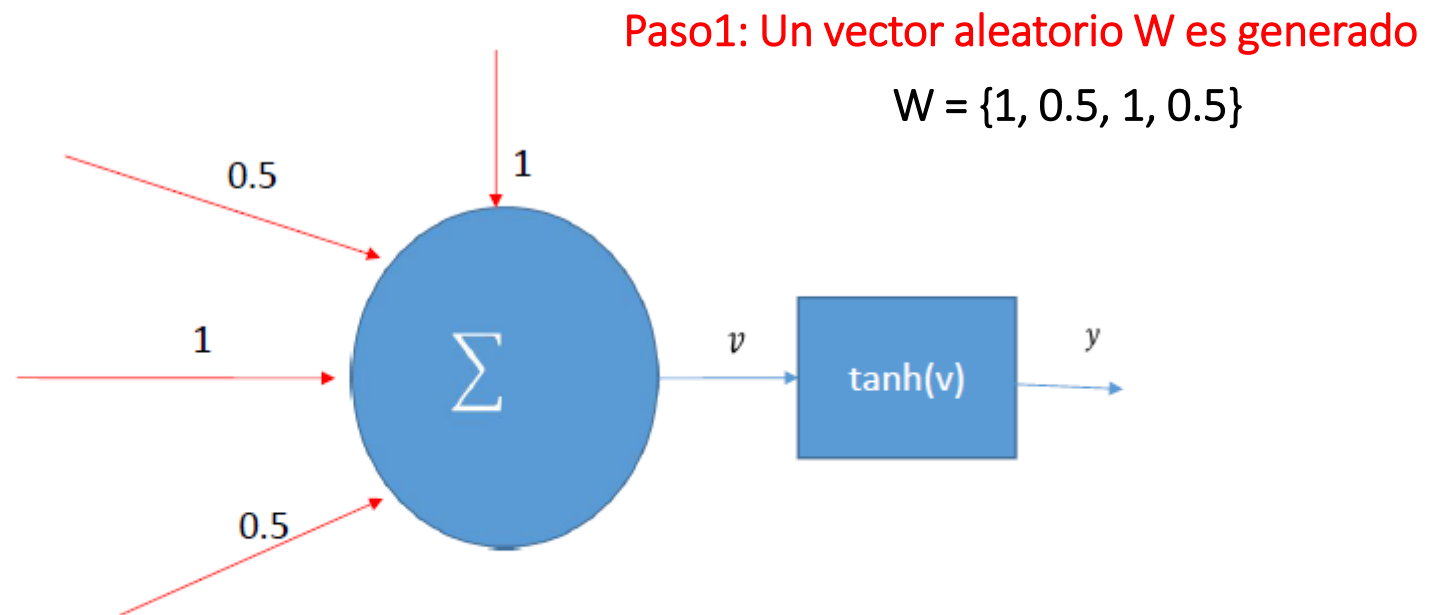


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

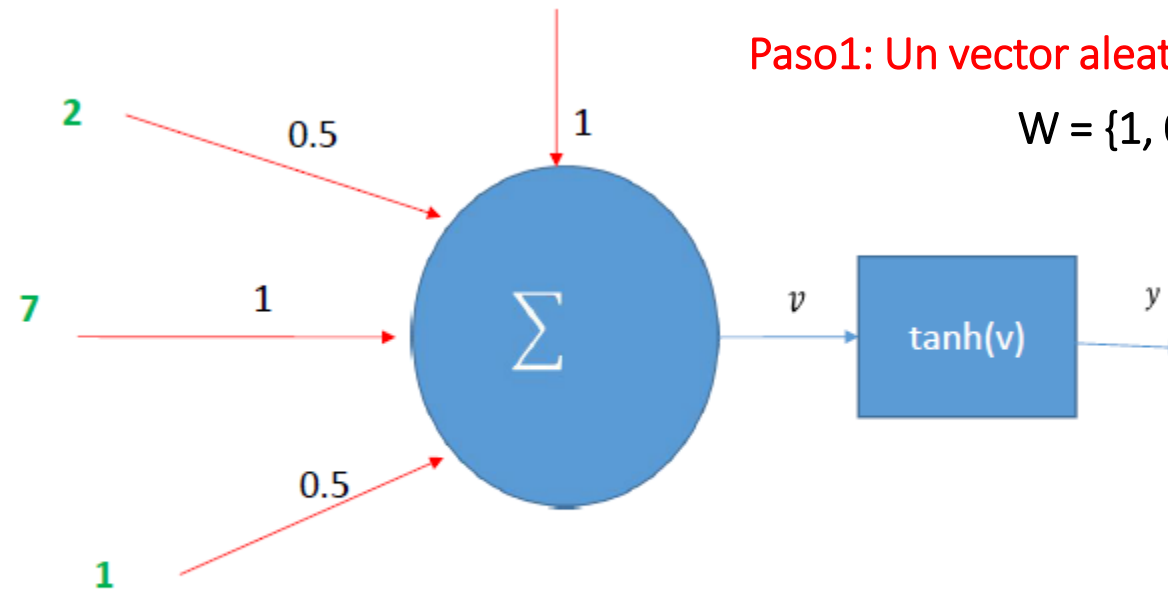


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

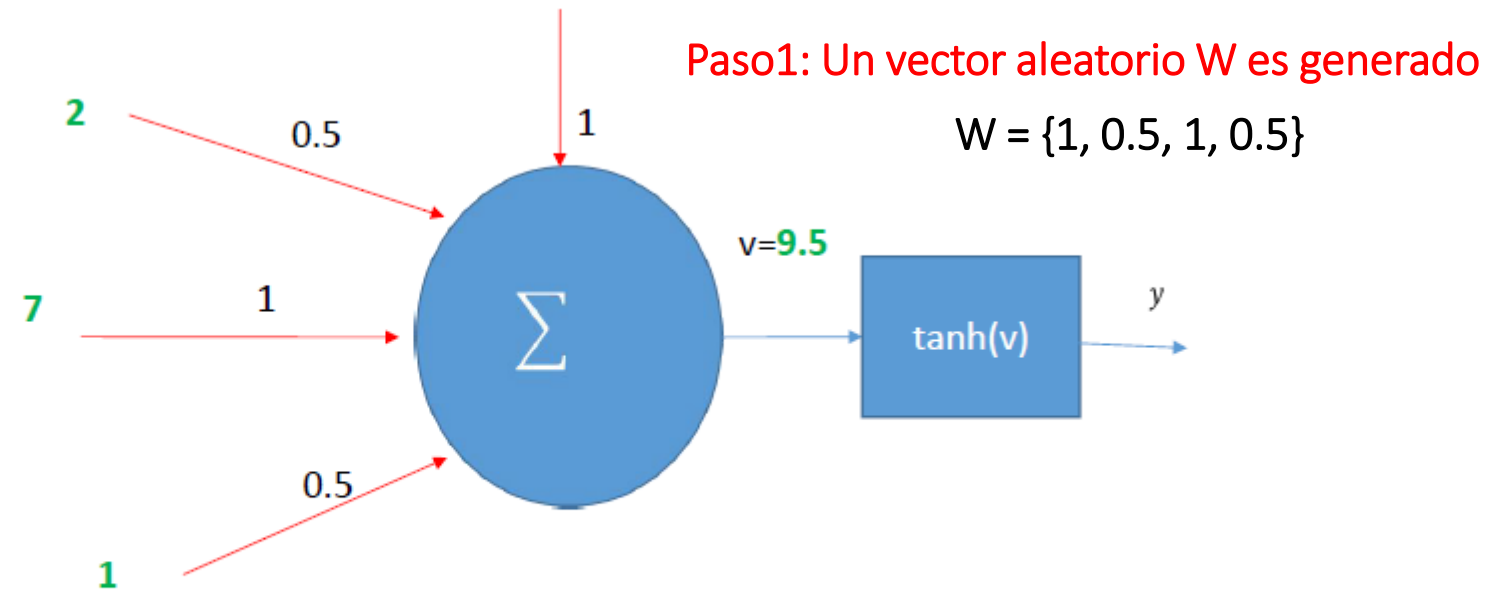


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

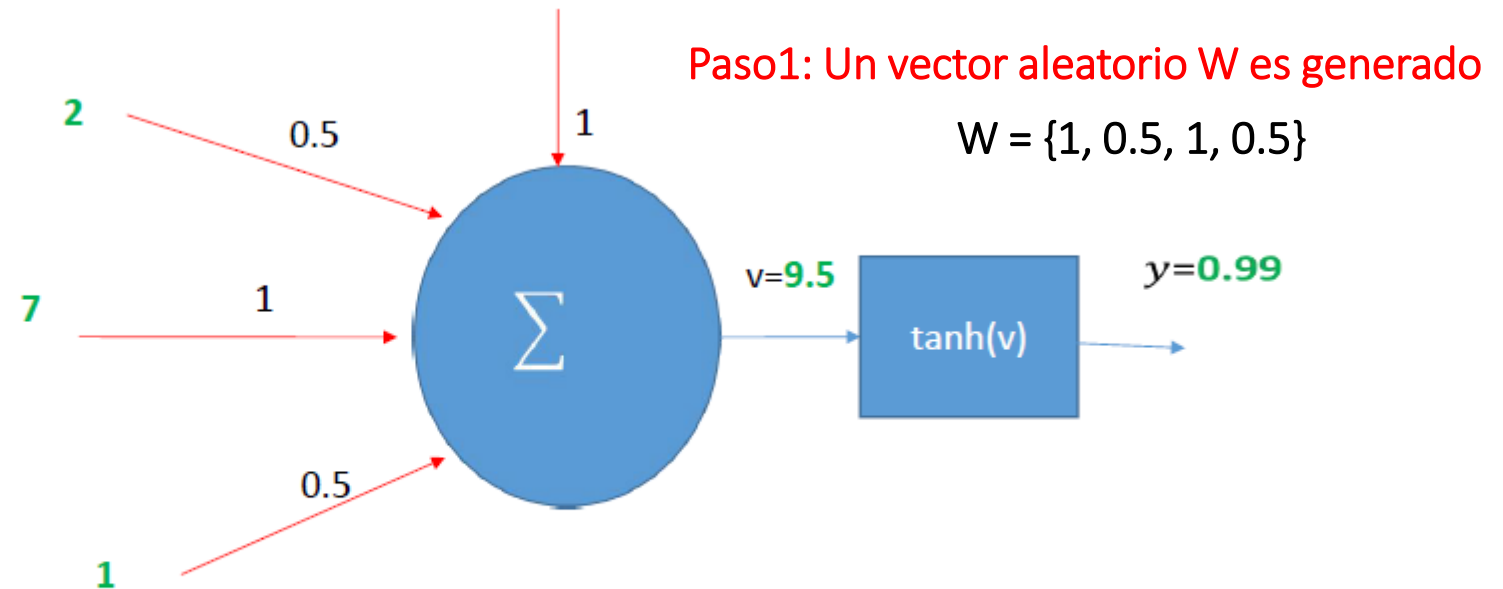


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

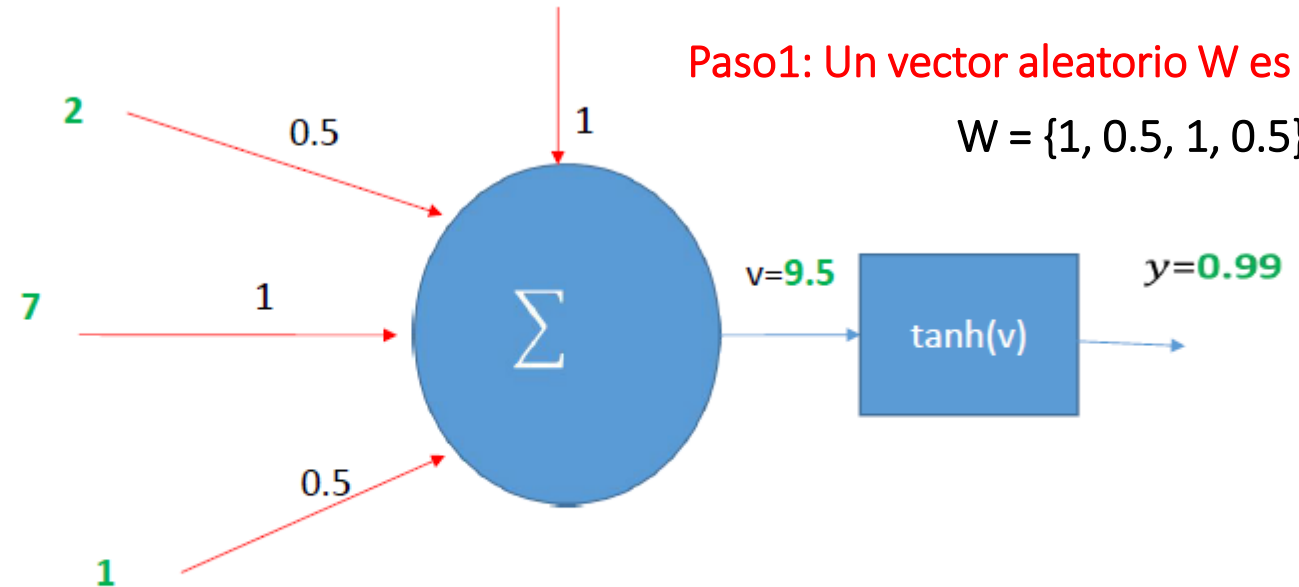


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



Mapear el valor de y a la clase label:

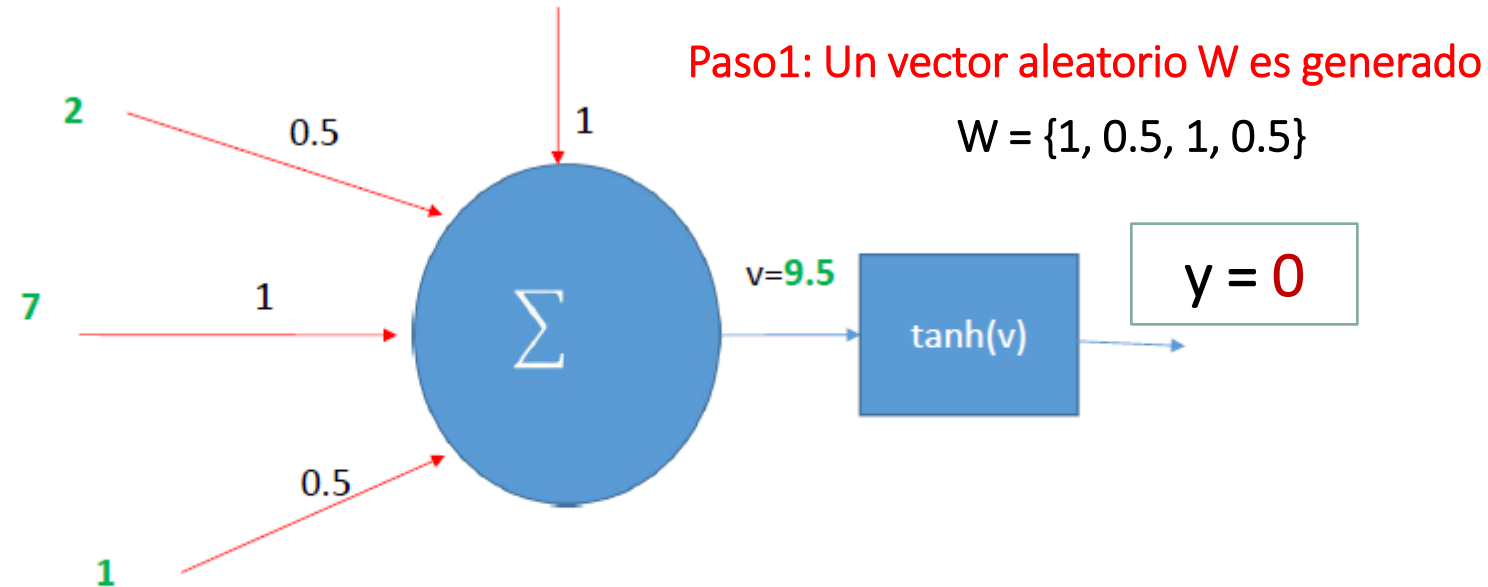
$$y = \begin{cases} 0 & \text{si } y \geq 0 \\ 1 & \text{de otra manera} \end{cases}$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



Mapear el valor de y a la clase label:

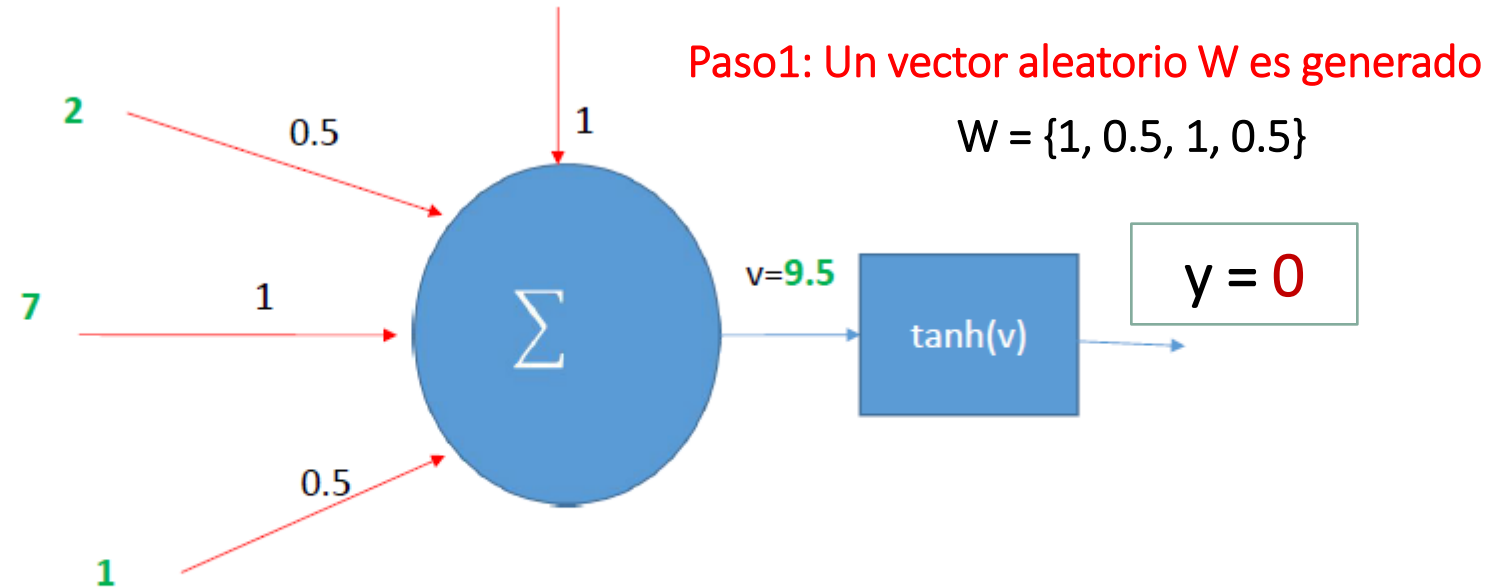
$$y = \begin{cases} 0 & \text{si } y \geq 0 \\ 1 & \text{de otra manera} \end{cases}$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



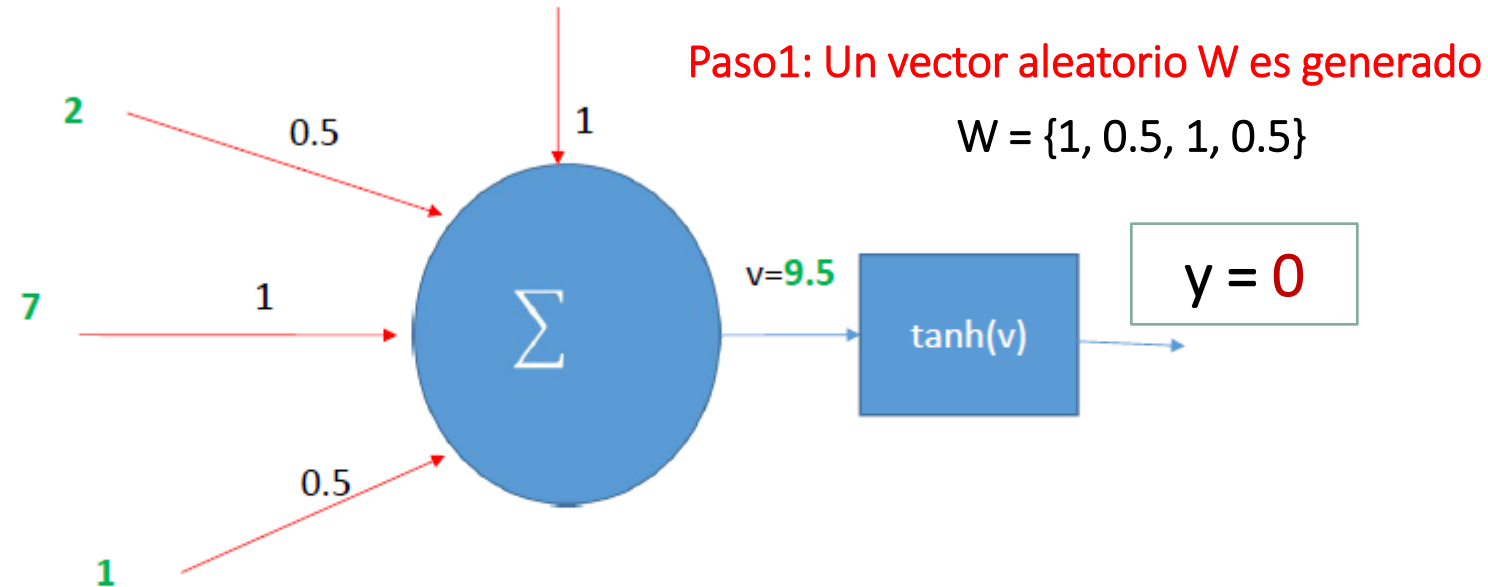
Calcula el valor de error dada la clase esperada c
 $e = y - c$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



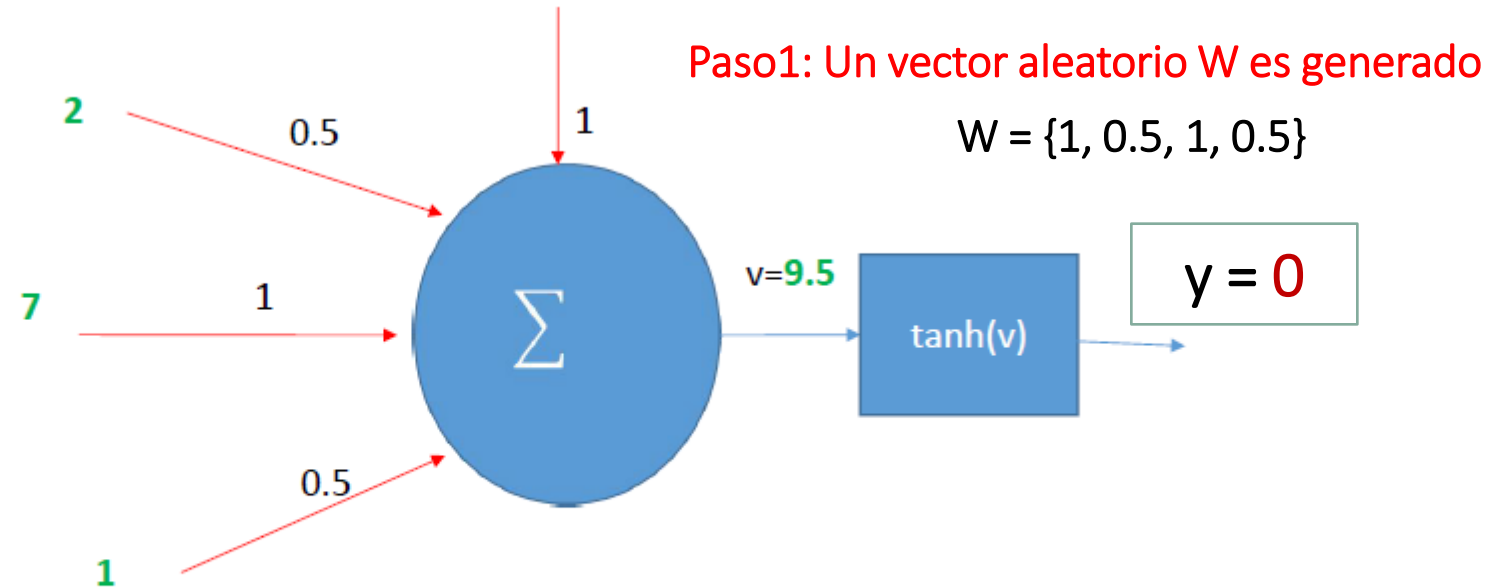
Calcula el valor de error dada la clase esperada c
 $e = 0 - 0 = 0$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



Por cada componente de W, calcula la variación

$$\Delta w_i = n * e * x_i$$

Donde n es un **factor de perturbación**. $n = 0.01$

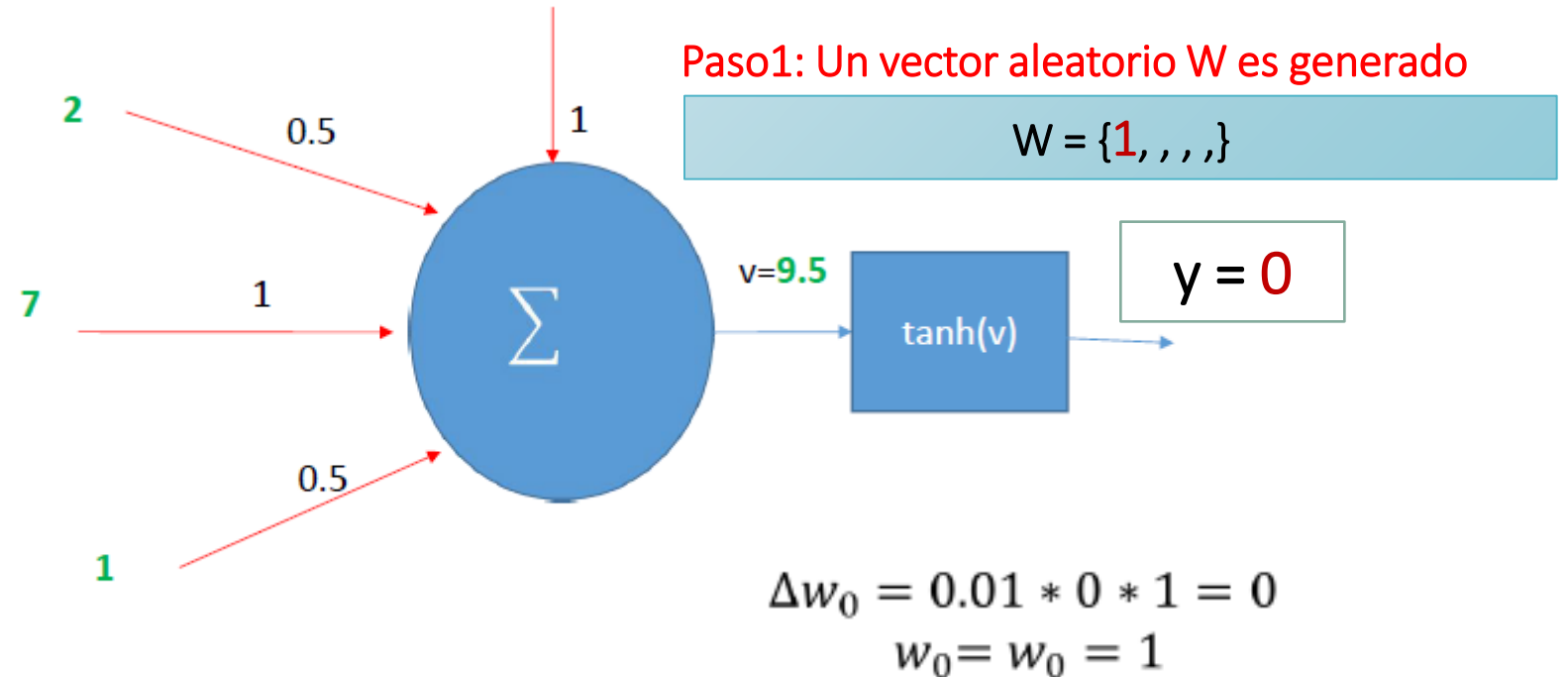
$$w_i = w_i + \Delta w_i$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

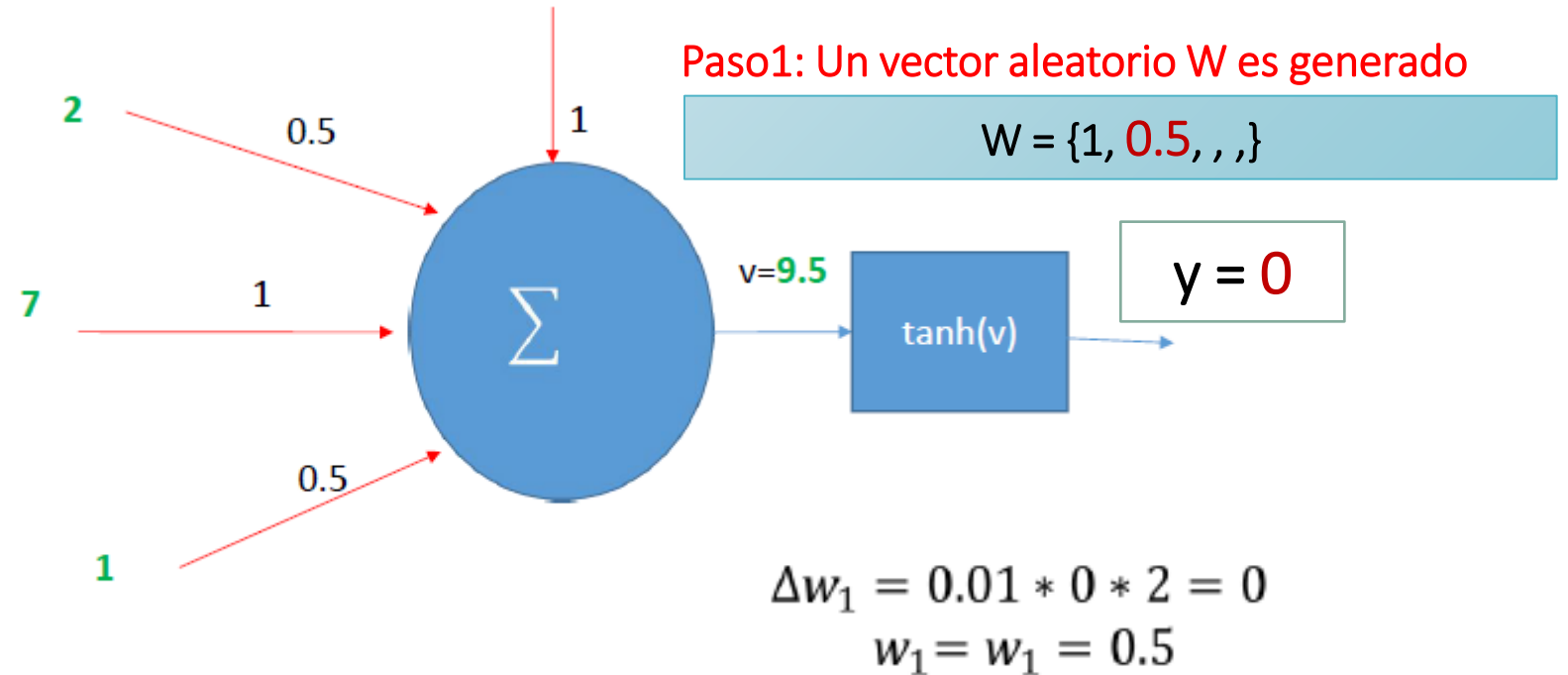


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

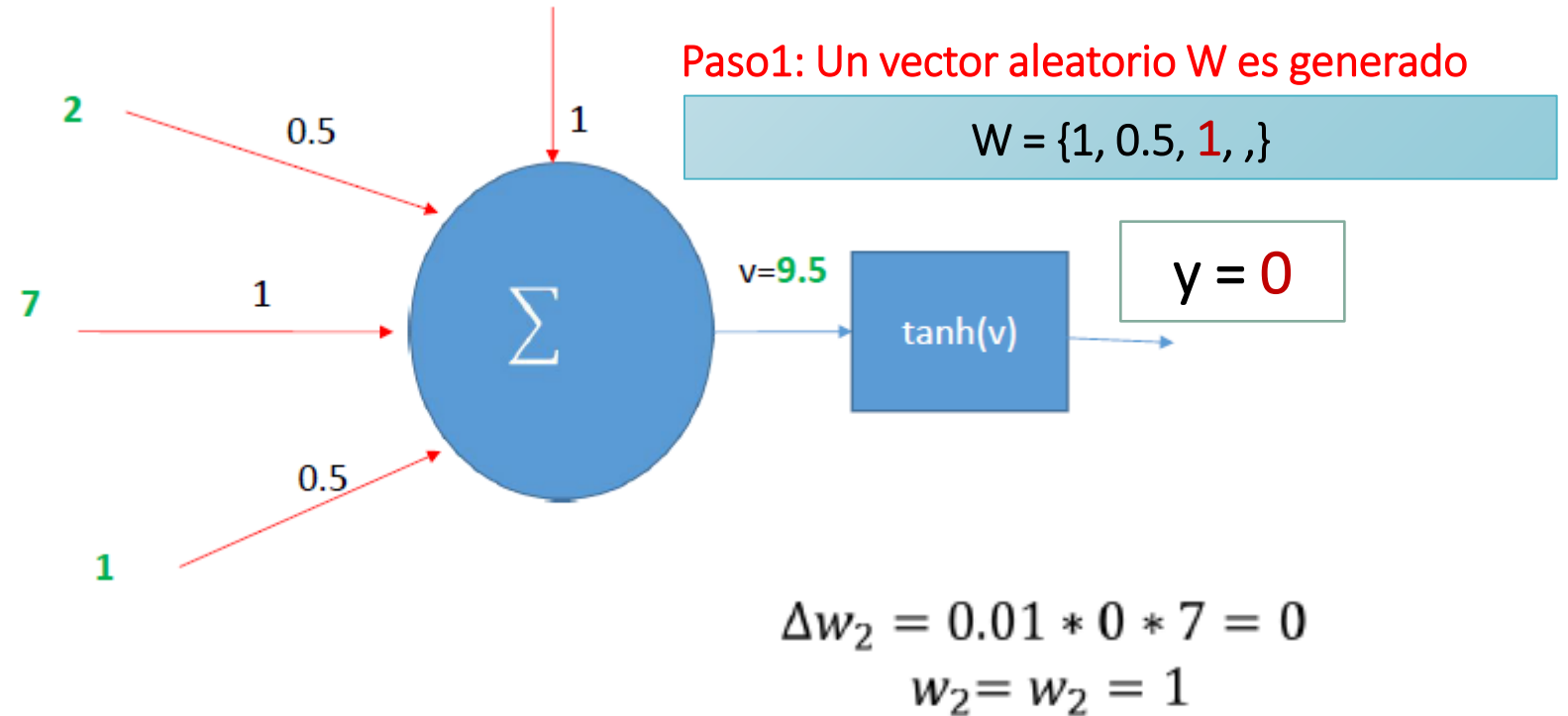


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

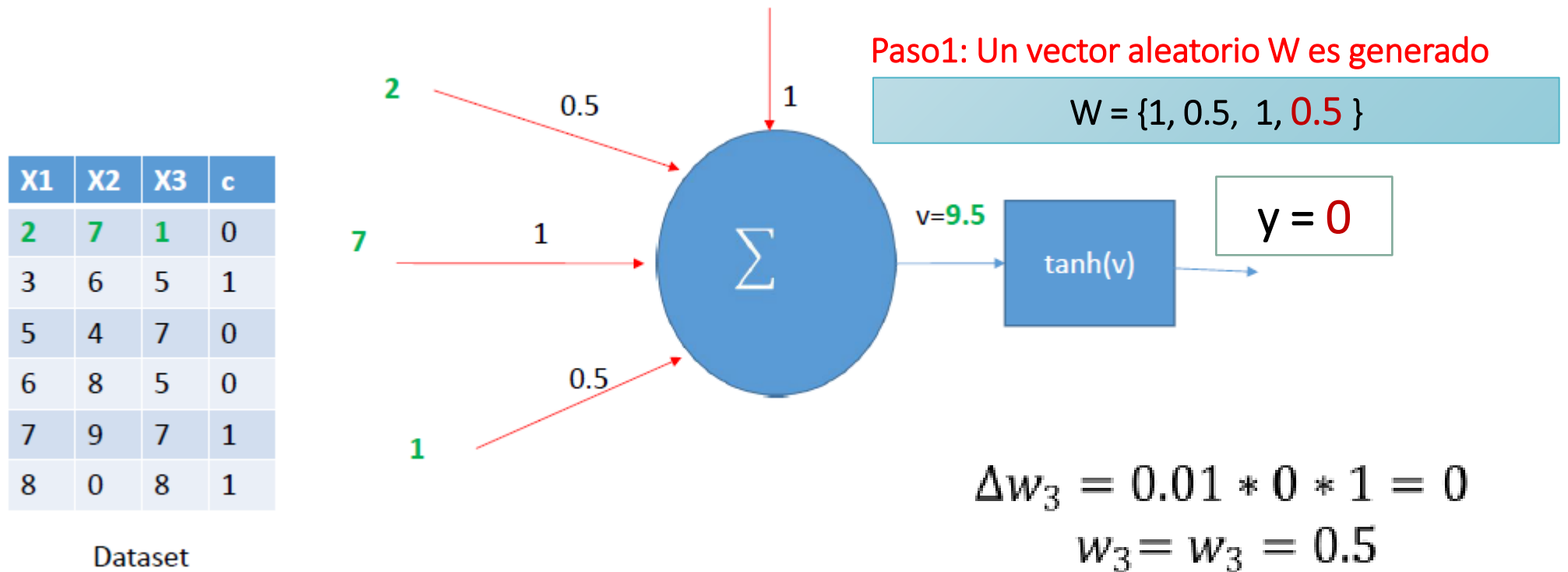
X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron



Como se puede observar no cambio el vector de pesos, probemos ahora con otros patrones

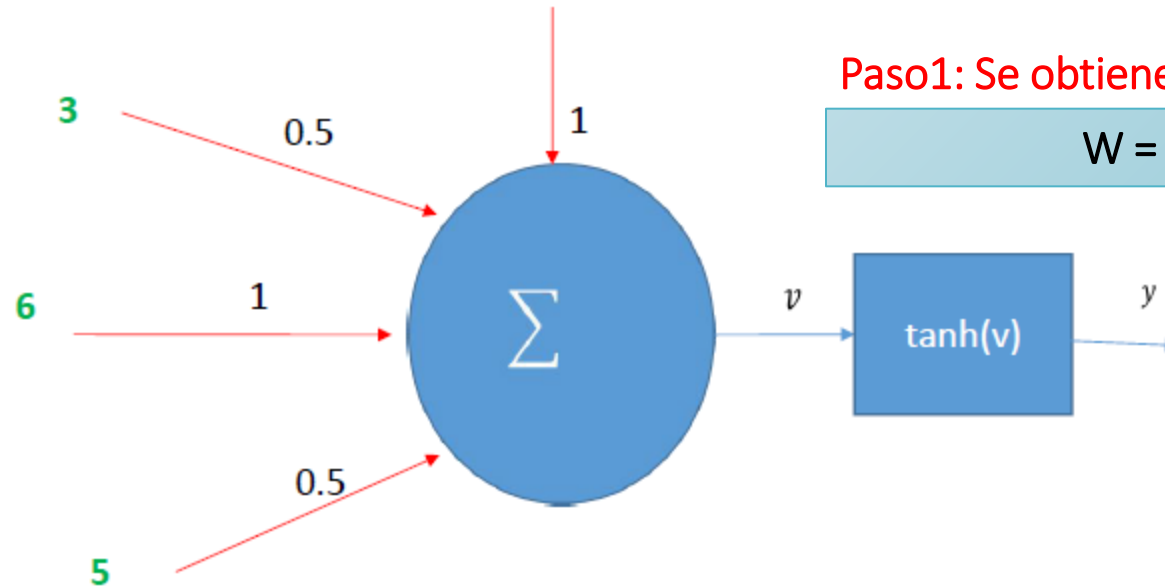
3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

Se procesa el segundo patron:

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



Paso1: Se obtiene el vector de pesos W anterior

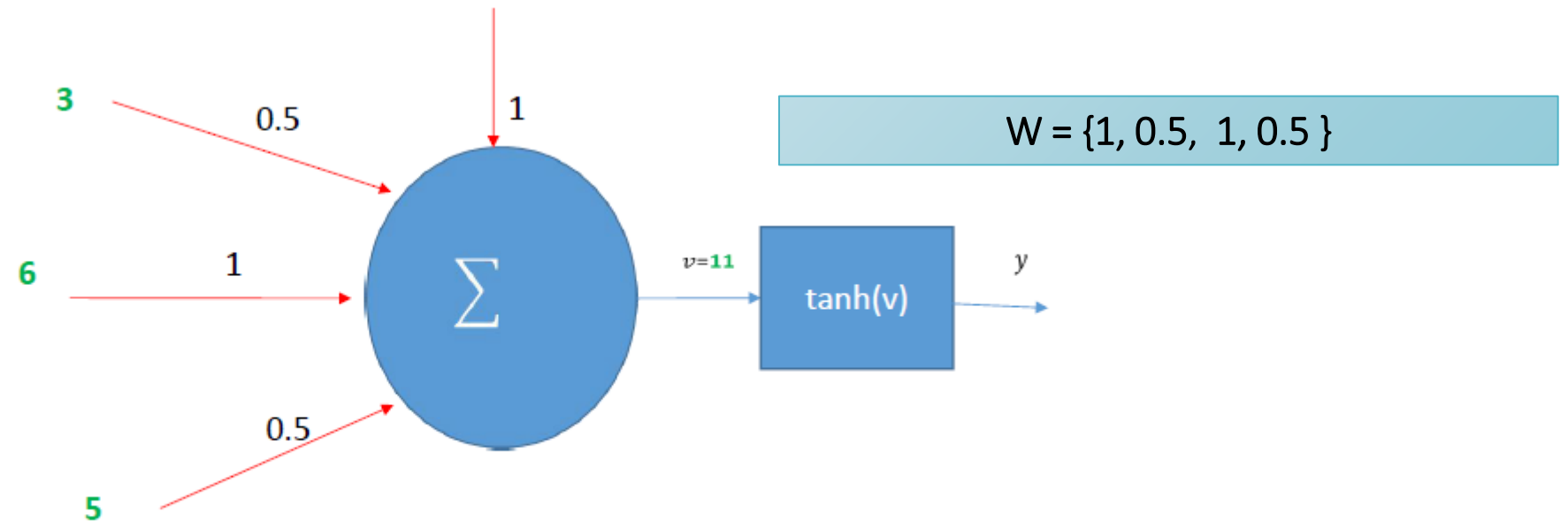
$$W = \{1, 0.5, 1, 0.5\}$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

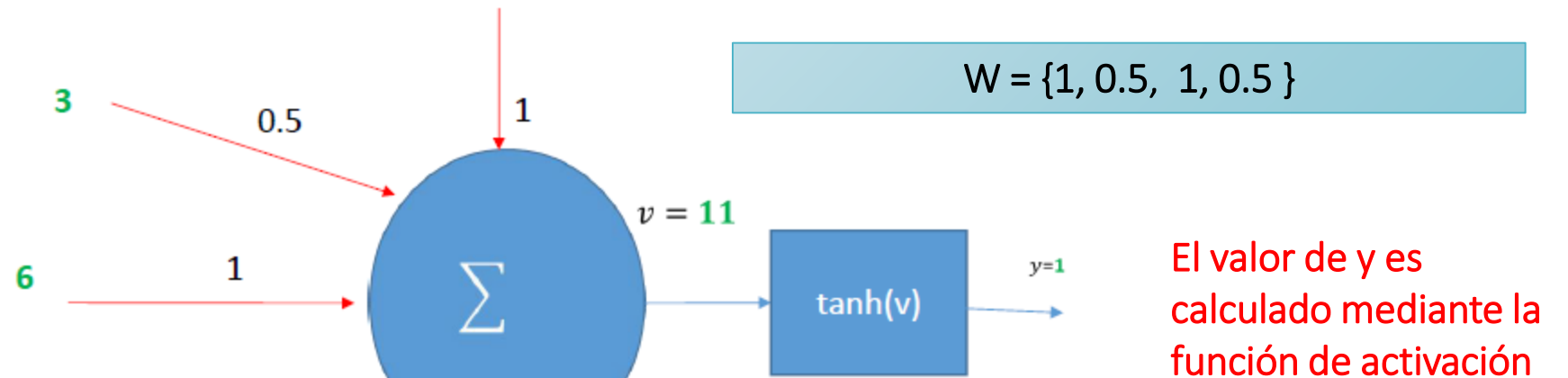


3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



Mapear el valor de y a la clase label:

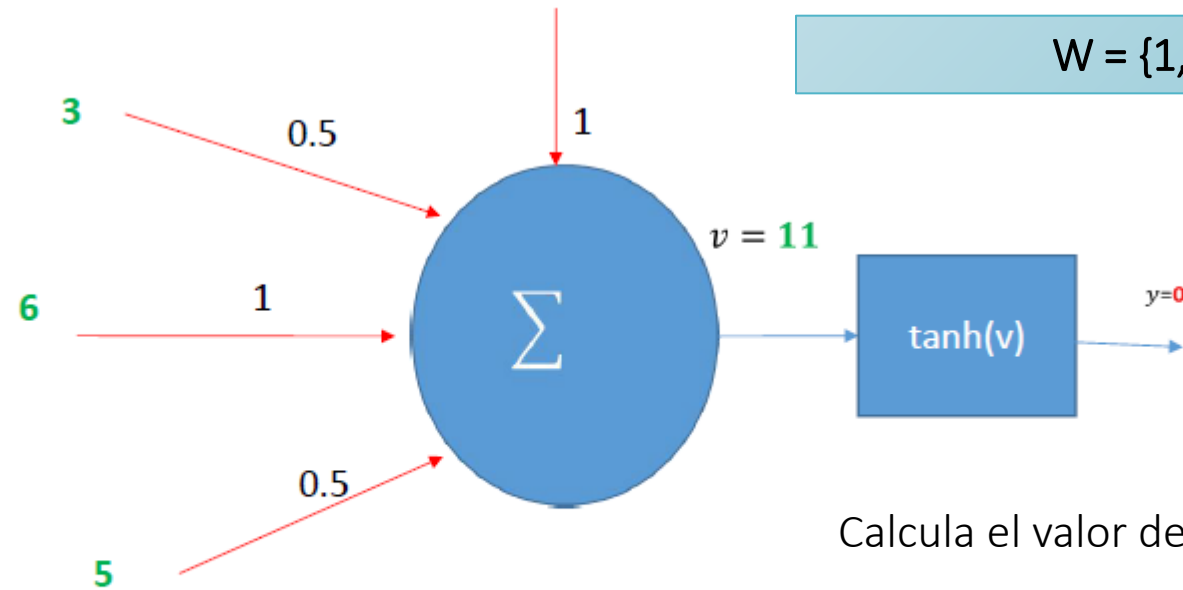
$$y = \begin{cases} 0 & \text{si } y \geq 0 \\ 1 & \text{de otra manera} \end{cases}$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



$$W = \{1, 0.5, 1, 0.5\}$$

Calcula el valor de error dada la clase esperada c

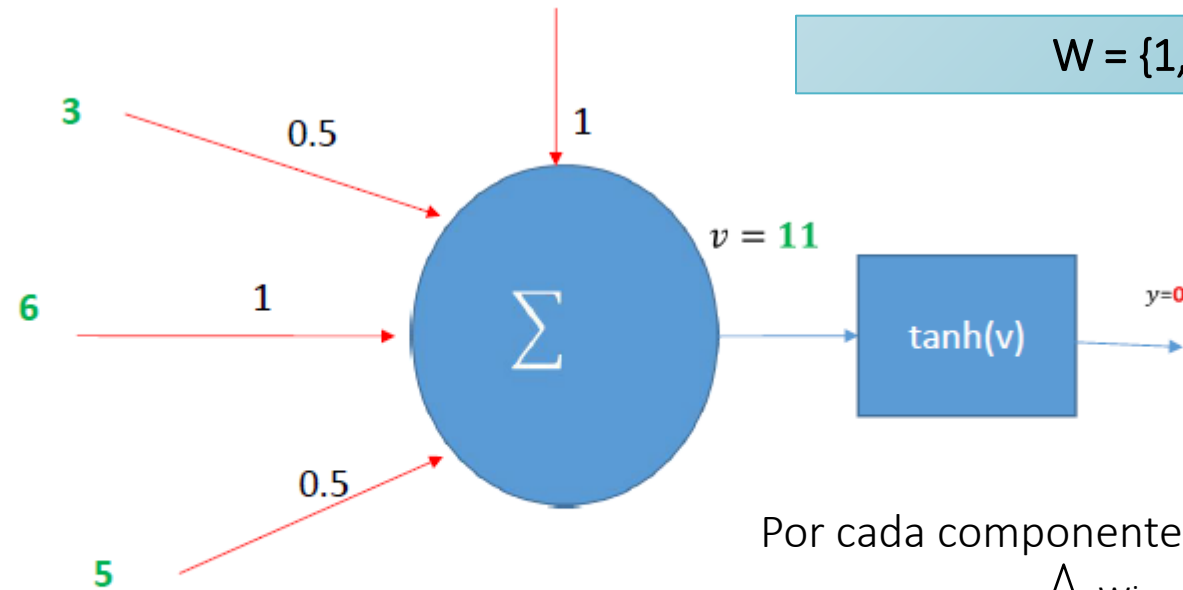
$$e = y - c$$
$$e = 0 - 1 = -1$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



Por cada componente de W , calcula la variación

$$\Delta w_i = n * e * x_i$$

Donde n es un **factor de perturbación**. $n = 0.01$

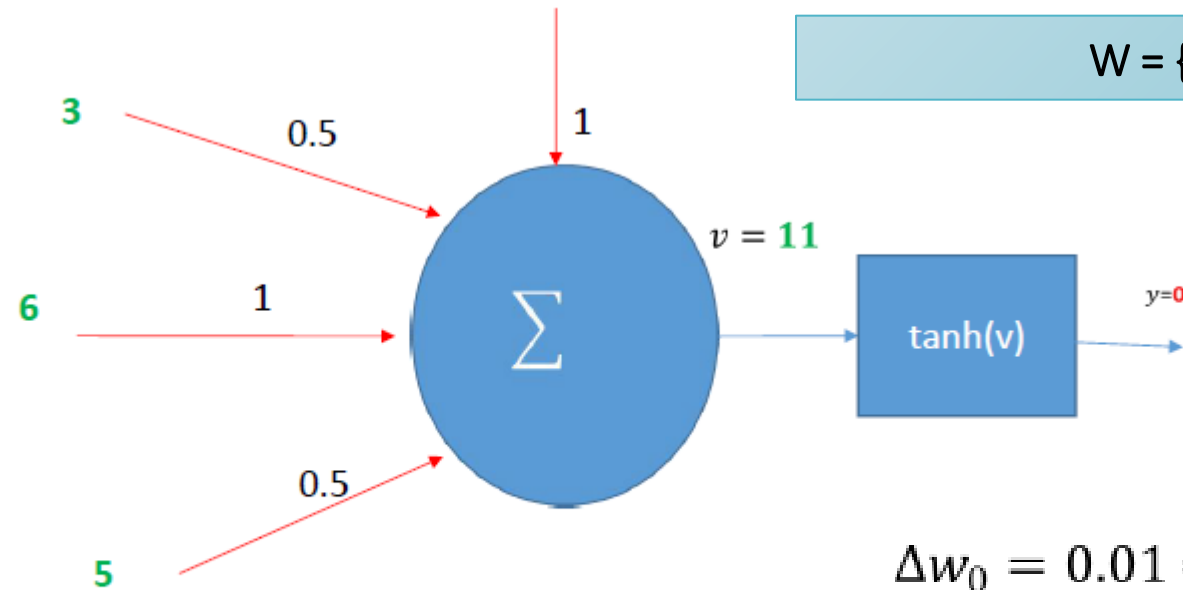
$$w_i = w_i + \Delta w_i$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



$$W = \{ 0.99, \dots \}$$

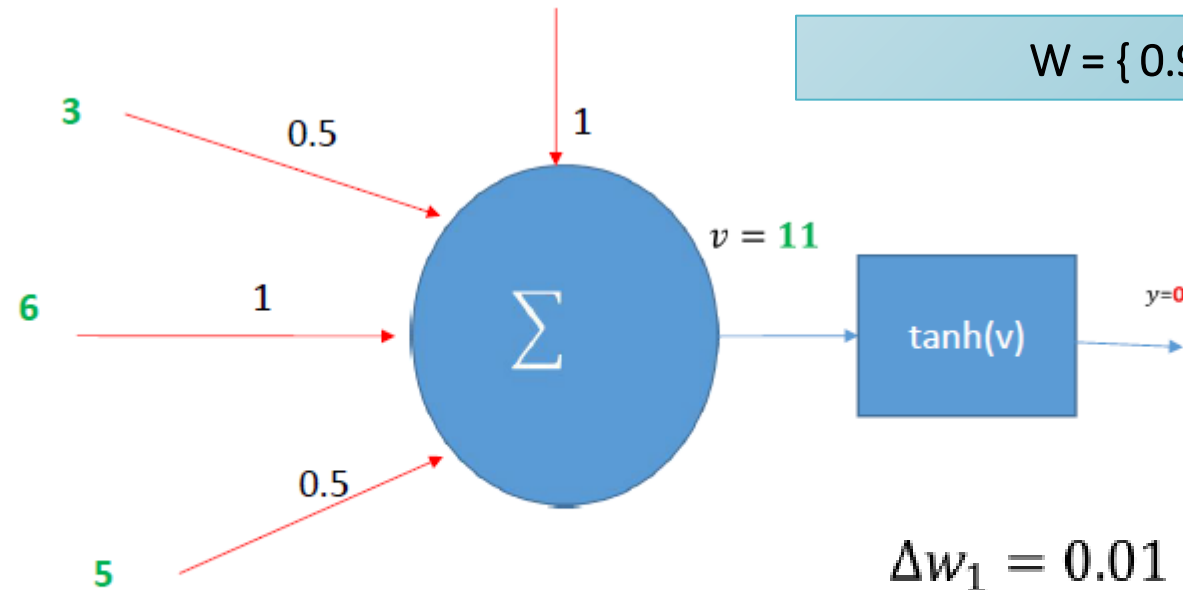
$$\Delta w_0 = 0.01 * -1 * 1 = -0,01$$
$$w_0 = 1 - 0,01 = 0,99$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



$$W = \{ 0.99, 0.47, \dots \}$$

$$\Delta w_1 = 0.01 * -1 * 3 = -0.03$$

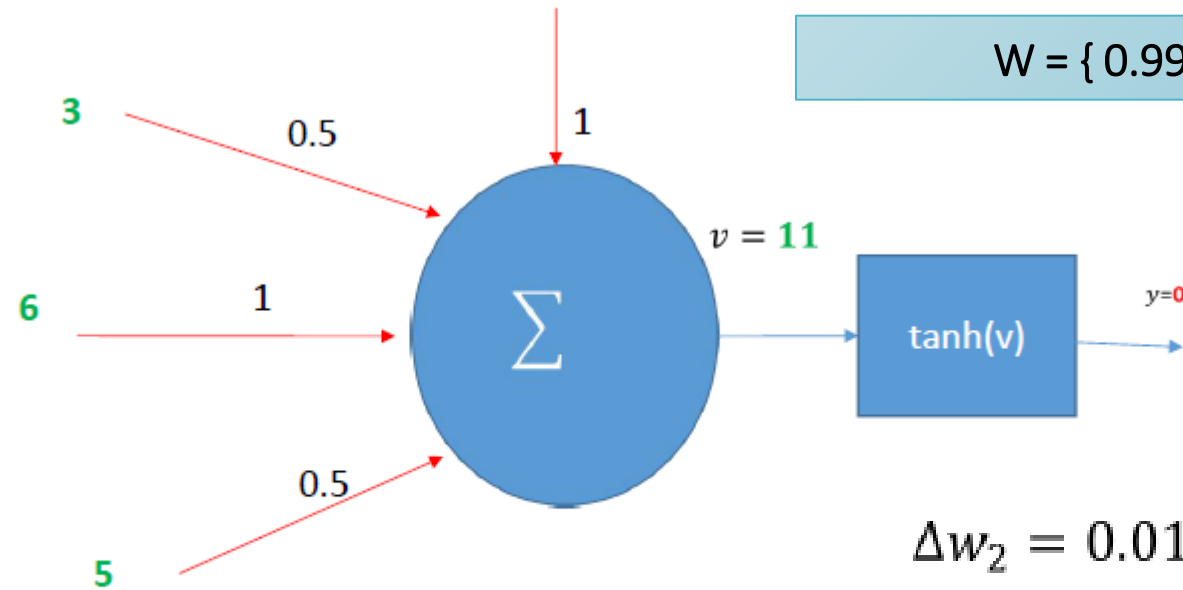
$$w_1 = 0.5 - 0.03 = 0.47$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



$$W = \{ 0.99, 0.47, \textcolor{red}{0.94}, \}$$

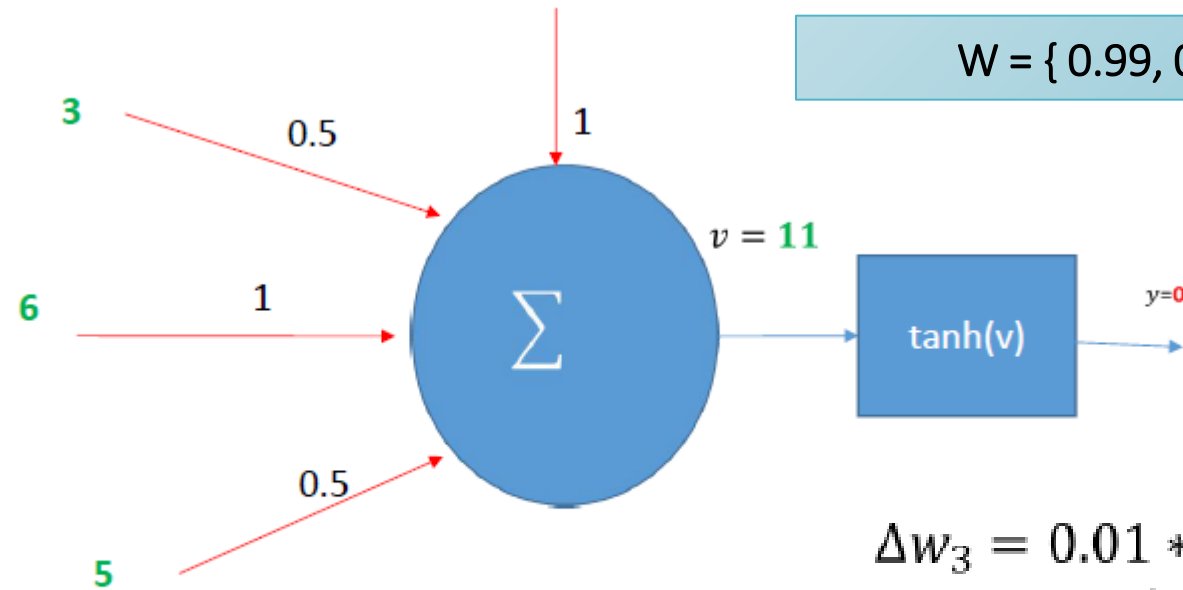
$$\Delta w_2 = 0.01 * -1 * 6 = 0.06$$
$$w_2 = 1 - 0.06 = 0.94$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset



$$W = \{ 0.99, 0.47, 0.94, \textcolor{red}{0.45} \}$$

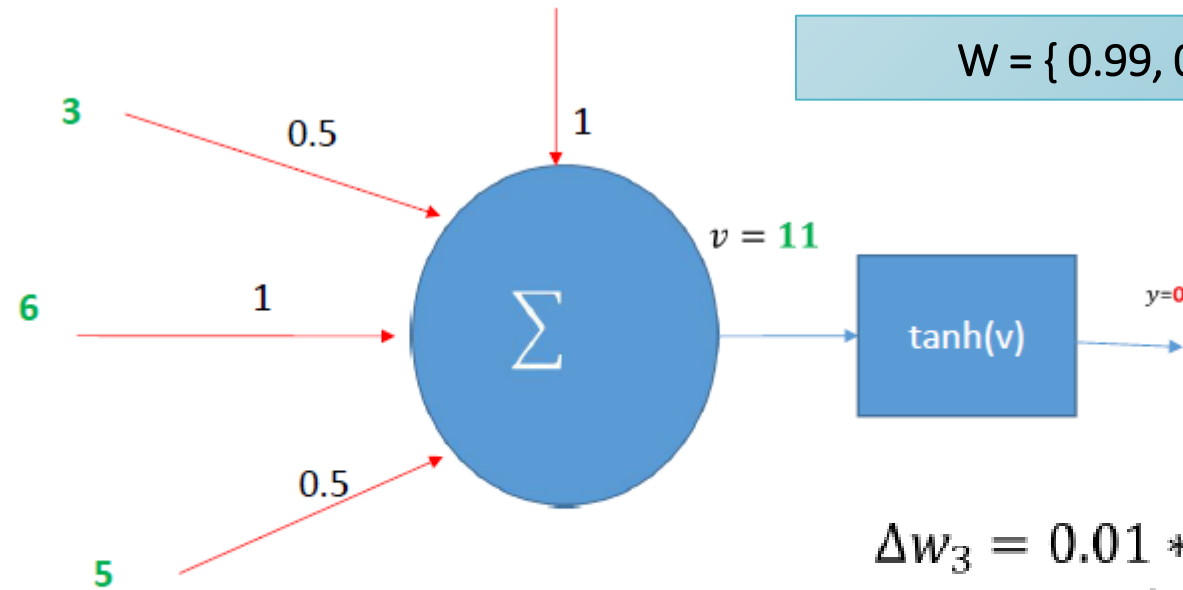
$$\Delta w_3 = 0.01 * -1 * 5 = -0.05$$
$$w_3 = 0.5 - 0.05 = 0.45$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

X1	X2	X3	c
2	7	1	0
3	6	5	1
5	4	7	0
6	8	5	0
7	9	7	1
8	0	8	1

Dataset

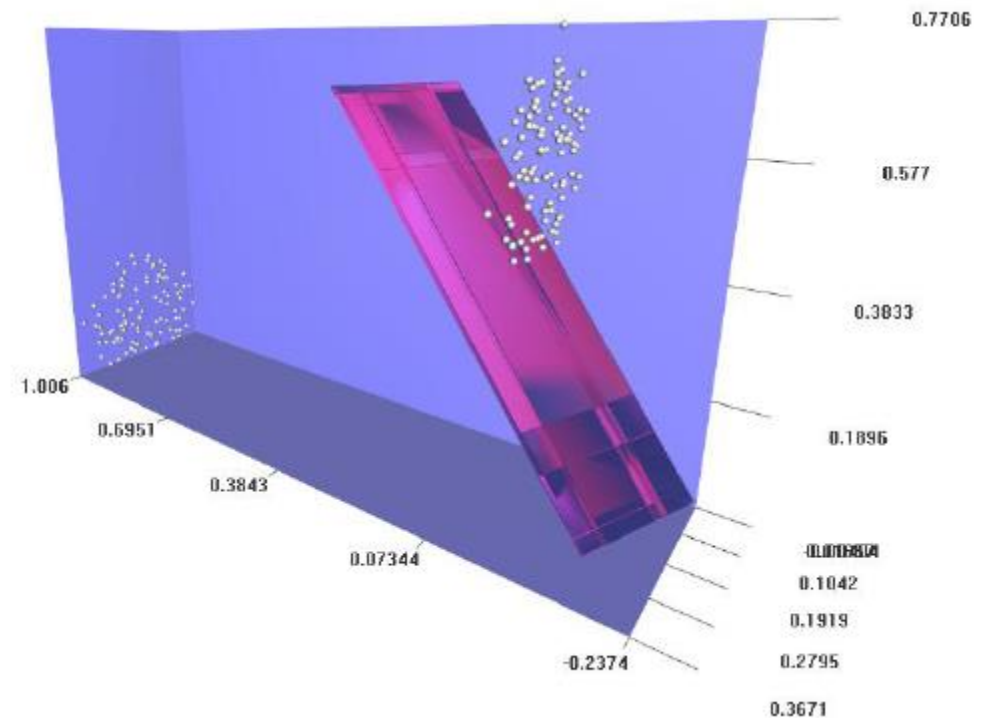
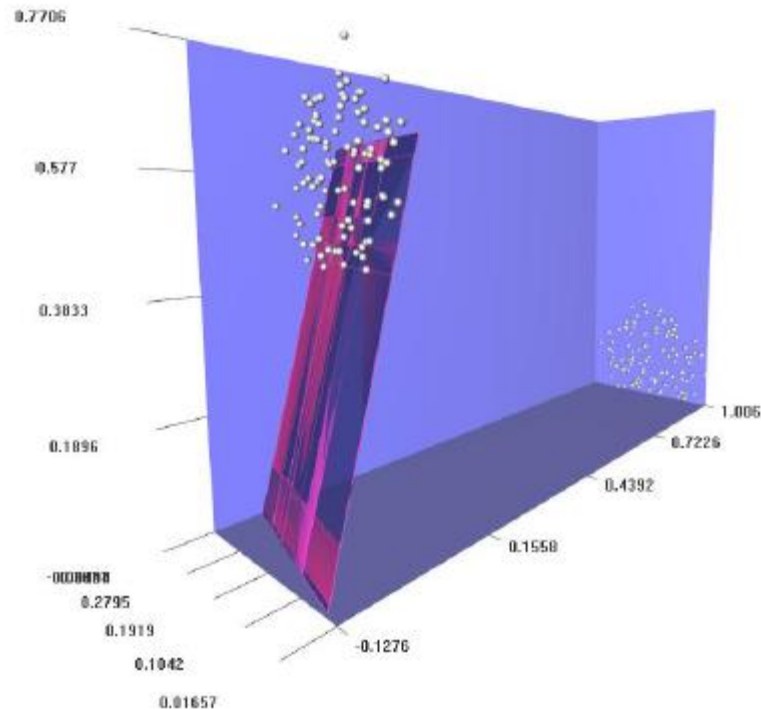


$$W = \{ 0.99, 0.47, 0.94, \textcolor{red}{0.45} \}$$

$$\Delta w_3 = 0.01 * -1 * 5 = -0.05$$
$$w_3 = 0.5 - 0.05 = 0.45$$

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron



3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

```
In [1]: import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('Redes_Neuronales').getOrCreate()
import pandas as pd
```

```
In [2]: from pyspark.ml.classification import MultilayerPerceptronClassifier
from pyspark.ml.linalg import Vectors, SparseVector, DenseVector
from pyspark.ml.feature import VectorAssembler
```

```
In [*]: training = spark.read.csv("/tmp/clasespark/iris_ds.csv", header=True, nullValue="?", inferSchema=True)
training.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

```
In [5]: assembler = VectorAssembler(inputCols=["Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width"], outputCol="features")
assem_data = assembler.transform(training)
assem_data.show()
```

```
In [10]: train_scaler = StandardScaler(inputCol="features", outputCol="scaled_features", withStd=True, withMean=True)
train_scaler_model = train_scaler.fit(assem_data)
scaled_data = train_scaler_model.transform(assem_data)
scaled_data.show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

```
In [*]: splits = scaled_data.randomSplit([0.7, 0.3], 1234)
        train = splits[0]
        test = splits[1]

        # Creando modelo asignando las capas

        layers = [4, 1, 4]

        trainerasem = MultilayerPerceptronClassifier(maxIter=100, layers=layers, blockSize=128, seed=1234)
        trainedModel = trainerasem.fit(train)
```

```
In [12]: trainedModel.layers
        # help(trainedModel)
```

```
Out[12]: [4, 1, 4]
```

```
In [13]: trainedModel.weights
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

```
In [15]: result = trainedModel.transform(test)
result.createOrReplaceTempView("red_neuronal")
```

```
In [17]: spark.sql("""
            select label, prediction, count(1) as cantidad
            from red_neuronal
            group by label, prediction
            """).show()
```

3. Taller de Modelo de Aprendizaje

Aprendizaje Supervisado – Redes Neuronales Perceptron

```
In [ ]: spark.stop()
```

A close-up photograph of a right hand holding a silver ballpoint pen, writing the words "Thank you" in a cursive script on a white surface. The pen is positioned at the end of the word "you", and the ink is a dark grey or black. The background is a plain, light-colored surface.

Thank you