Utilizar Machine Learning y no morir en el intento

Codemotion 2017

Acerca de Mi

PhD en Ciencia y Tecnología Informática

Carlos III de Madrid (UC3M)

King's College of London with ESA

- Planificación de Tareas
- Aprendizaje Automático No supervisado
- Aprendizaje Automático por Refuerzo
- Toma de decisiones en espacio

Data Scientist

beBee Clícars KIO AI

T3chFest (Organizador)

Moisés Martínez

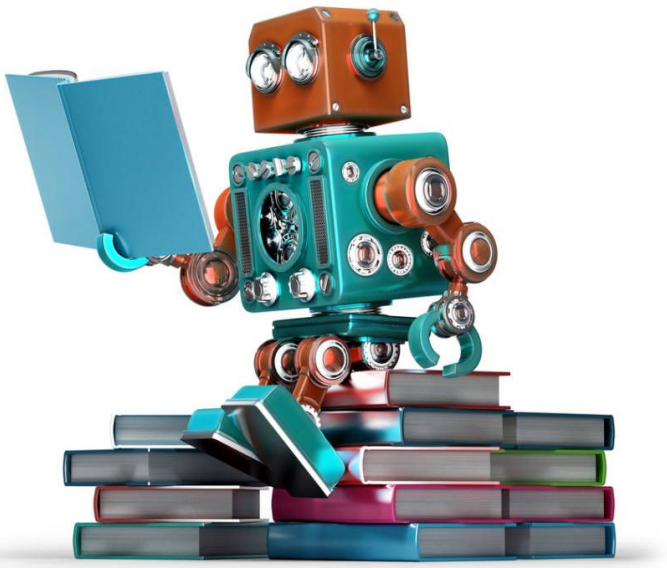
momartinm

moisipm

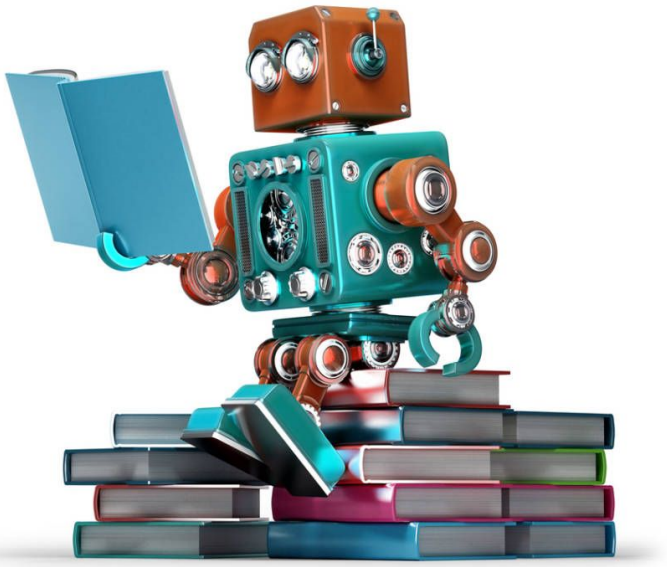
1.
Machine Learning

Qué es esto?

¿Máquinas aprendiendo conceptos del mundo real?



¿Máquinas aprendiendo conceptos del mundo real?



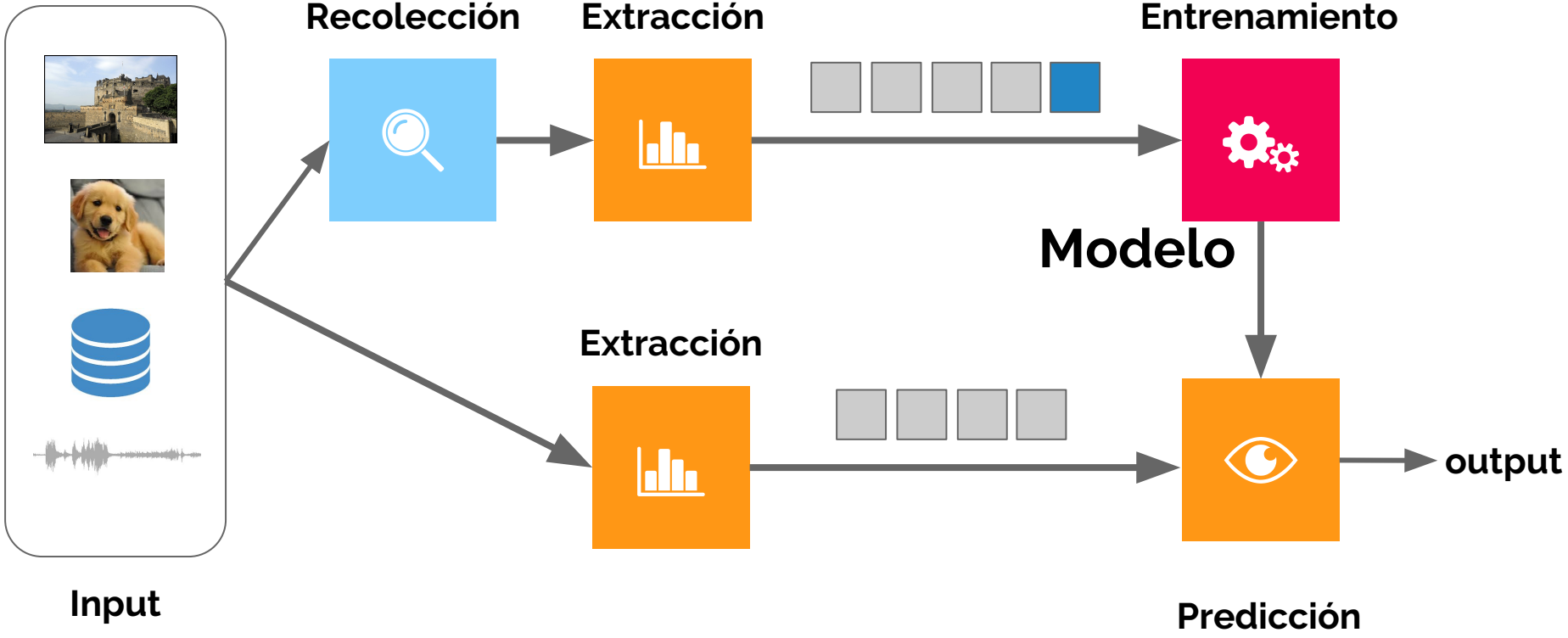
*Sólo son capaces de detecter **patrones** o **similitudes** entre la información (ejemplos)*

¿Para qué necesito ML?

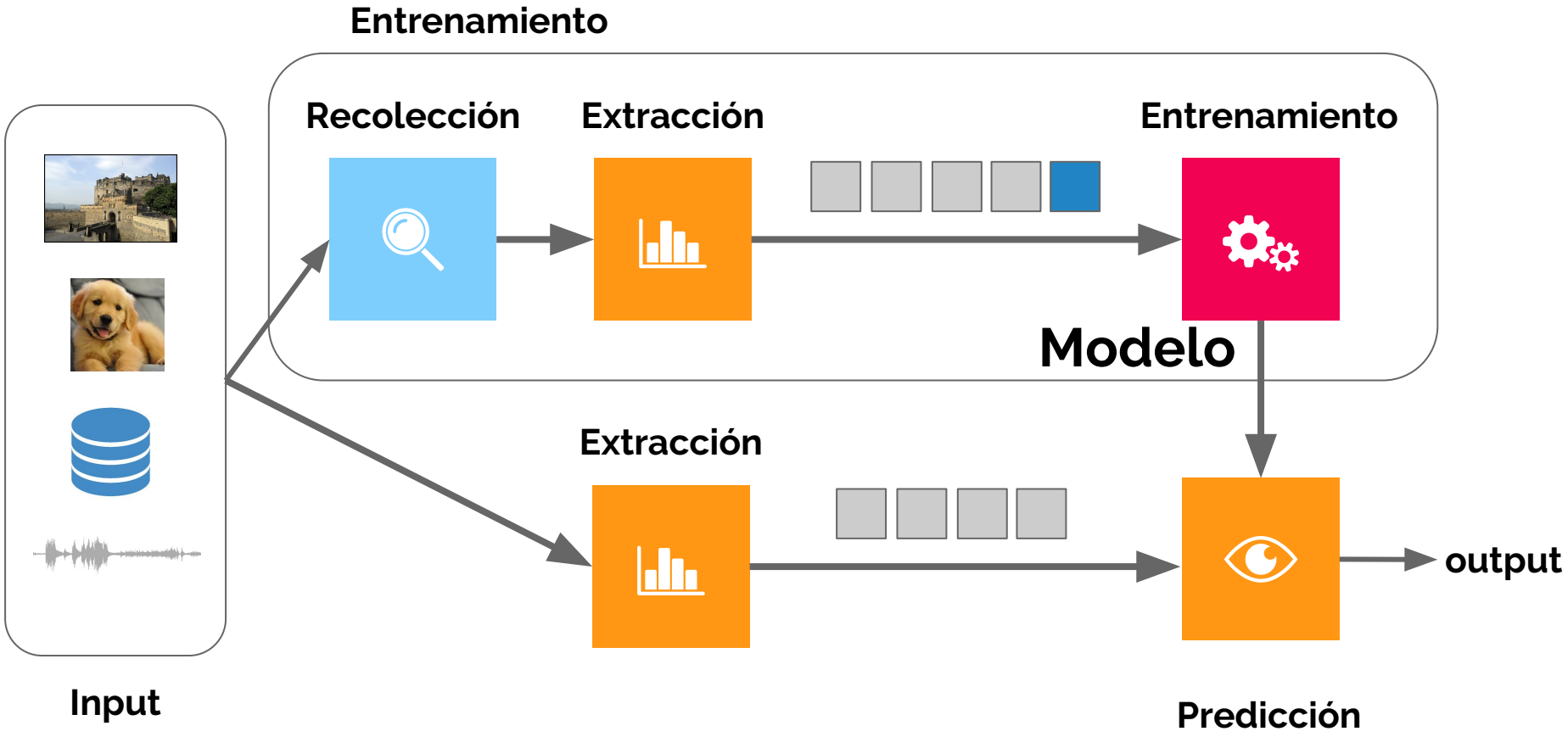
“

- Streaming ETL (Extract, Transform and Load)
- Detección de anomalías en datos de servidores
- Enriquecimiento de datos (redes sociales)
- Detección de anomalías en sensores
- Detección de fraude en transacciones bancarias
- Detección de comportamientos (usuarios)
- Detección de imágenes (animales, personas, productos)
- Detección de sonidos
- Procesamiento del lenguaje (NLP)
- Traducción Simultánea
- Gamificación mediante recompensas

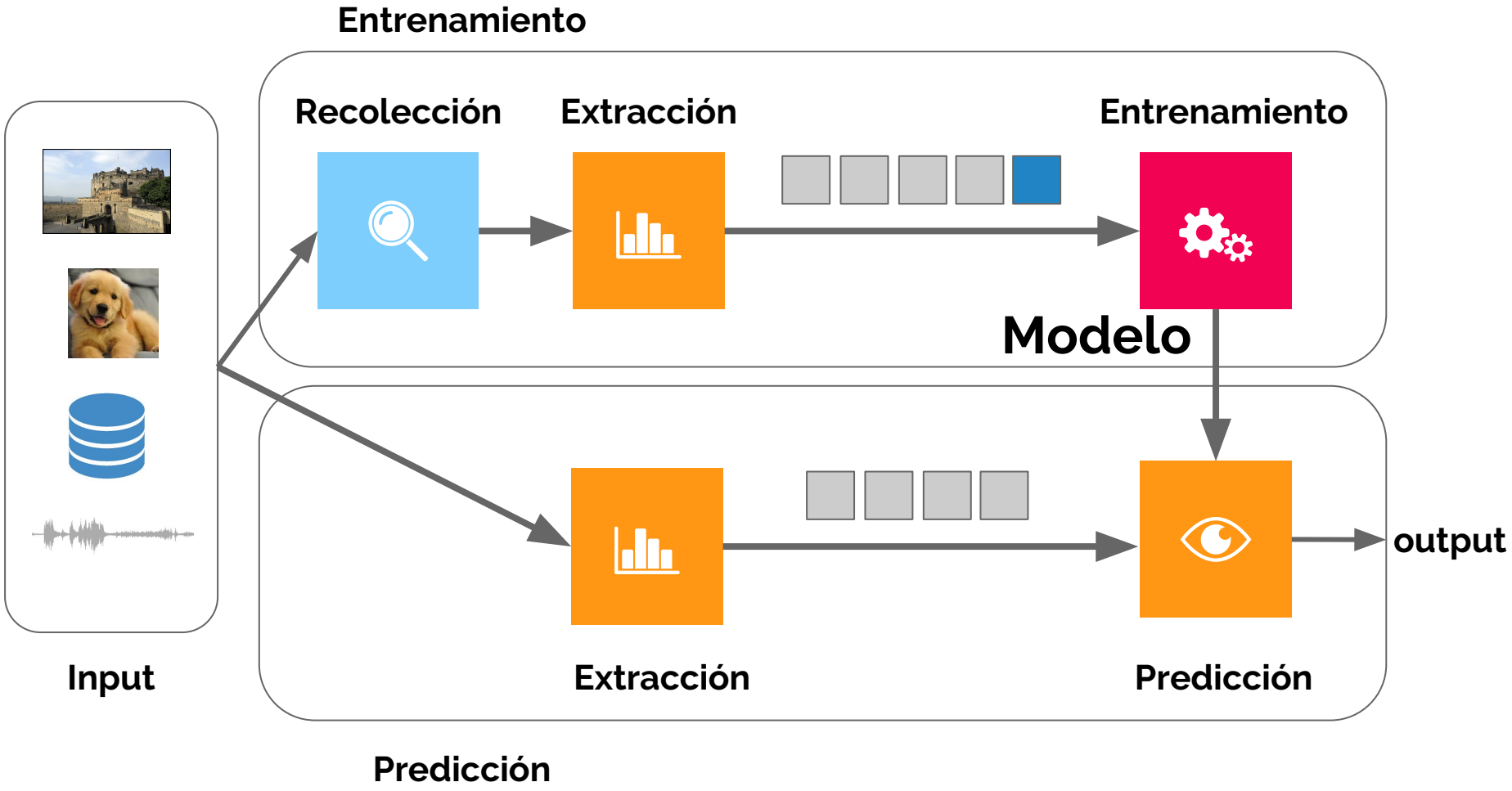
Machine Learning (ML) pipeline



Machine Learning (ML) pipeline



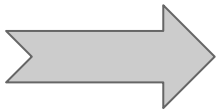
Machine Learning (ML) pipeline



Tipos de aprendizaje

Aprendizaje Supervisado

- Se conocen las clases (Etiquetado)
- Características con misma clase



Tipos de aprendizaje

Aprendizaje Supervisado

- Se conocen las clases (Etiquetado)
- Características con misma clase

Aprendizaje No Supervisado

- No se conocen la clase (Etiquetado)
- Ejemplos con similares características

Tipos de aprendizaje

Aprendizaje Supervisado

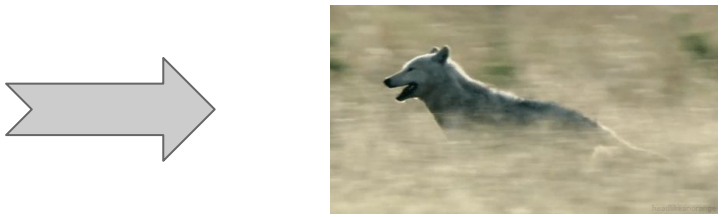
- Se conocen las clases (Etiquetado)
- Características con misma clase

Aprendizaje No Supervisado

- No se conocen la clase (Etiquetado)
- Ejemplos con similares características

Aprendizaje por refuerzo

No se utilizan clases (No hay etiquetado)
Refuerzo
Maximiza o minimiza la recompensa



Tipos de aprendizaje

Aprendizaje Supervisado

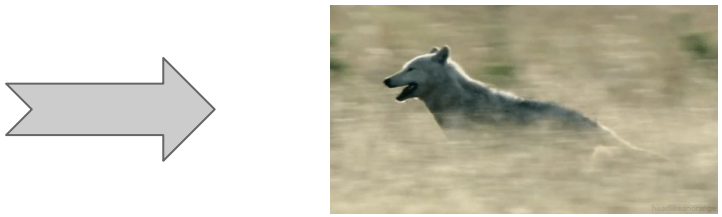
- Se conocen las clases (Etiquetado)
- Características con misma clase

Aprendizaje No Supervisado

- No se conocen la clase (Etiquetado)
- Ejemplos con similares características

Aprendizaje por refuerzo

No se utilizan clases (No hay etiquetado)
Refuerzo
Maximiza o minimiza la recompensa



¿Que me puede aportar Machine Learning?

Aprendizaje Supervisado

Predecir el precio de productos
Predecir demanda de productos
Detectar información en imágenes

¿Que me puede aportar Machine Learning?

Aprendizaje Supervisado

Predecir el precio de productos
Predecir demanda de productos
Detectar información en imágenes

Aprendizaje No Supervisado

Etiquetar a mis clientes
Agruparlos por comportamientos
Detectar grupos de clientes

¿Que me puede aportar Machine Learning?

Aprendizaje Supervisado

Predecir el precio de productos
Predecir demanda de productos
Detectar información en imágenes

Aprendizaje No Supervisado

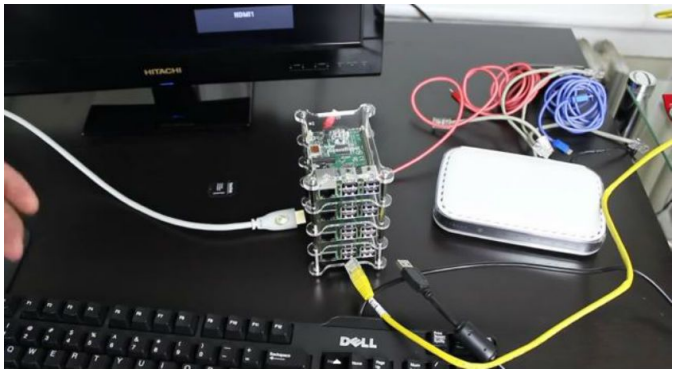
Etiquetar a mis clientes
Agruparlos por comportamientos
Detectar grupos de clientes

Aprendizaje por refuerzo

Testear nuevas funcionalidades
Gamificar a mis usuarios
Fomentar el uso de ciertas partes de mi APP o mi aplicación

¿Qué infraestructura necesito?

Depende de lo que necesites



Dependará de tus datos

2.

Recogiendo Datos

Recogiendo todos los datos del mundo?

Necesito tus datos

Los datos son lo más importante

“

Los datos son lo más importante

Flink

APACHE kafka

Scrapy

RabbitMQ

S3

APACHE Spark

Google Cloud Platform

Sino tienes consiguelos

Prepara los datos

1. **Comprueba el formato de los datos**
2. **Elimina los valores nulos**
3. **Elimina información redundante**
4. **Crea valores calculados**

Confía en los datos

Confía en los datos

Sólo si son de buena calidad

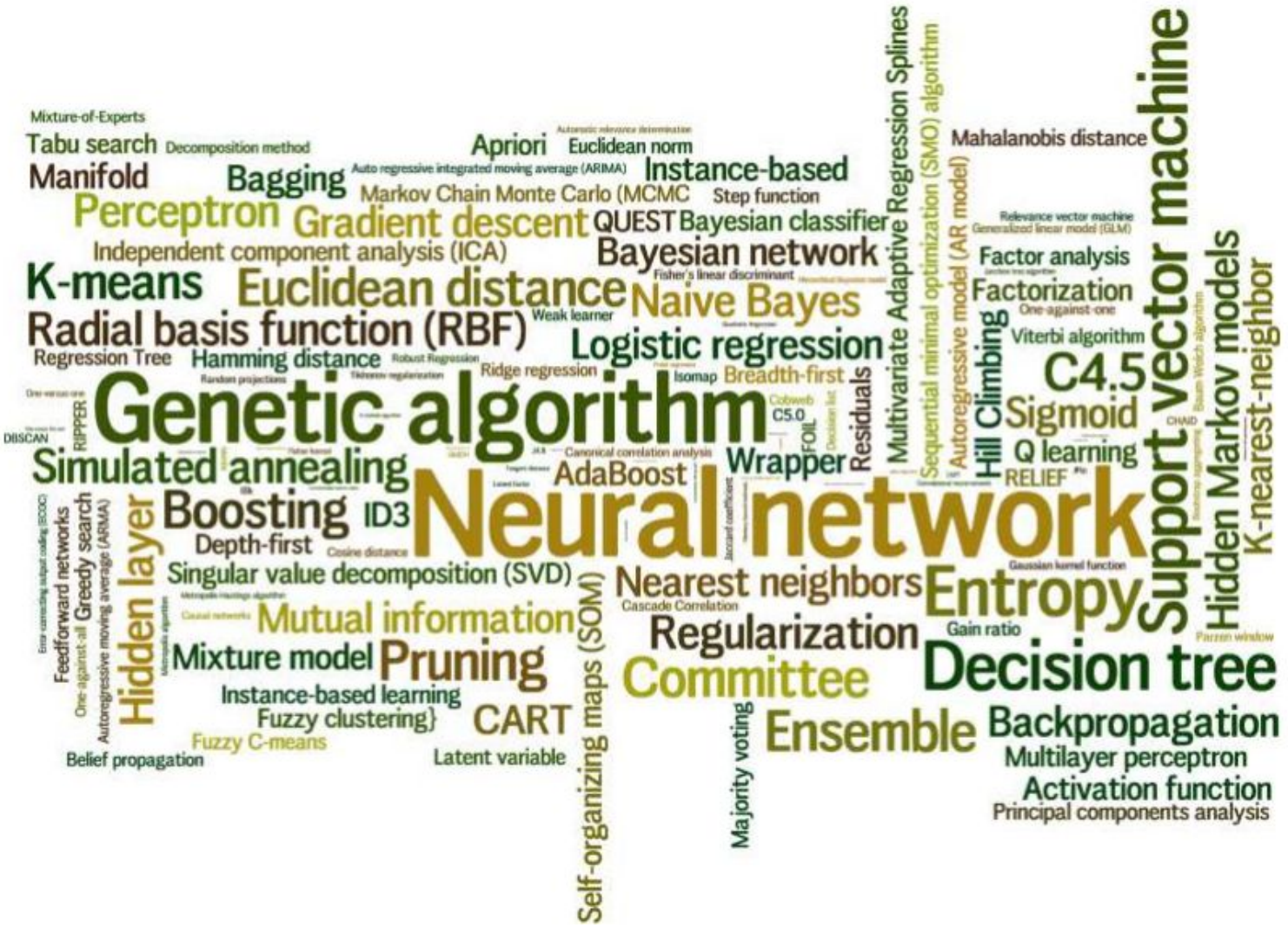
3.
Construyendo mi modelo

Aprendiendo a contar ovejas

¿Cómo aprendo?

¿Que algoritmo utilizo?

Algoritmos de aprendizaje



¿Los ejecuto todos?

Algoritmos de aprendizaje

Input

Etiquetado

- Cada ejemplo tiene una clase definida

Formato de las características

- Valores Discretos
- Valores Contínuos

Algoritmos de aprendizaje

Input

Etiquetado

- Cada ejemplo tiene una clase definida

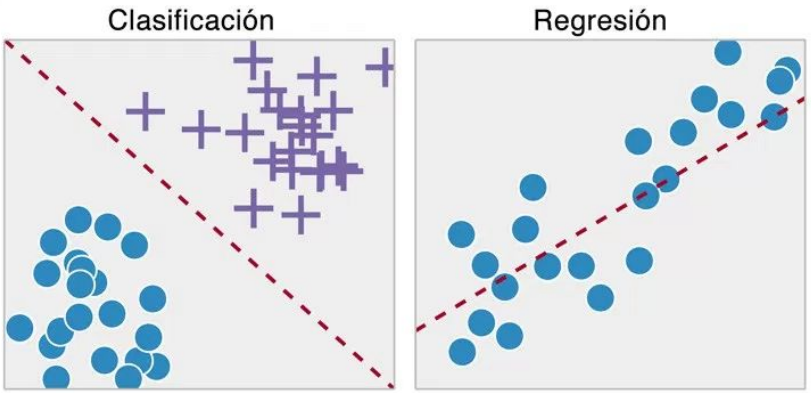
Formato de las características

- Valores Discretos
- Valores Contínuos

Output

Formato de la salida

- Clase (Clasificación)
- Valor numérico (Regresión)
- Clustering (Agrupamiento)



Algoritmos de aprendizaje

Etiquetado	Características	Output	Algoritmo
Si	Continuos discretos	Clase/valor numérico	Árboles de decisión
Sí	Continuos discretos	Clase/valor numérico	Random Forest
Si	Continuos	Clase	KNN
Si	Continuos	Valor Numérico	Regresión Lineal
No	Continuos	Clase	K–means
Si	Continuos	Valor Númerico	Gradient Descent

Algoritmos de aprendizaje

Etiquetado	Características	Output	Algoritmo
Si	Continuos discretos	Clase/valor numérico	Árboles de decisión
Sí	Continuos discretos	Clase/valor numérico	Random Forest
Si	Continuos	Clase	KNN
Si	Continuos	Valor Numérico	Regresión Lineal
No	Continuos	Clase	K-means
Si	Continuos	Valor Númerico	Gradient Descent

3.1 Regresión lineal

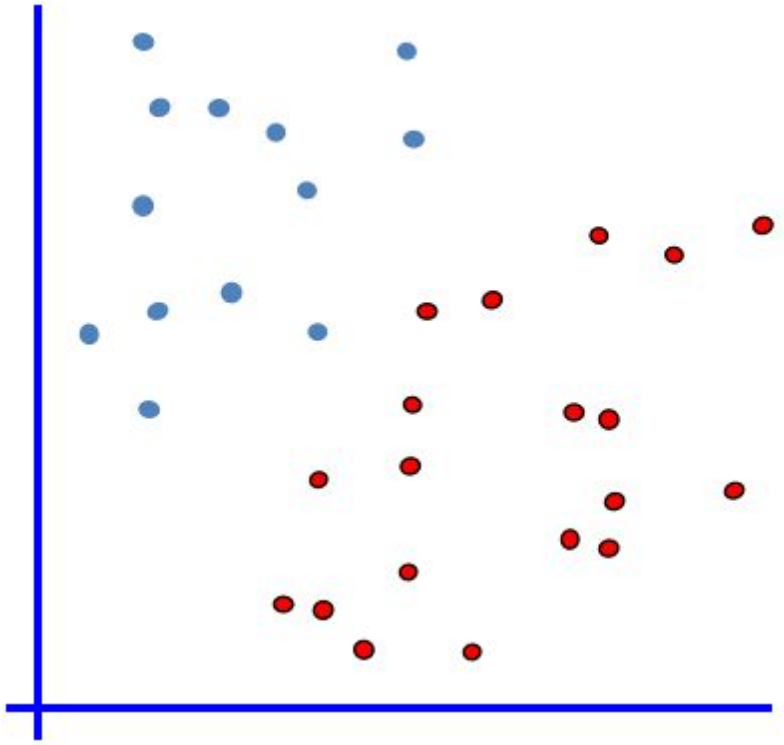
Algoritmos: Regresión Linear

Clasificación y regresión

Relación entre variables

- **Variable dependiente**
- **Conjunto de variables independientes**
  - **Continuas**
  - **Discretas**

$$f(x)=ax+b$$



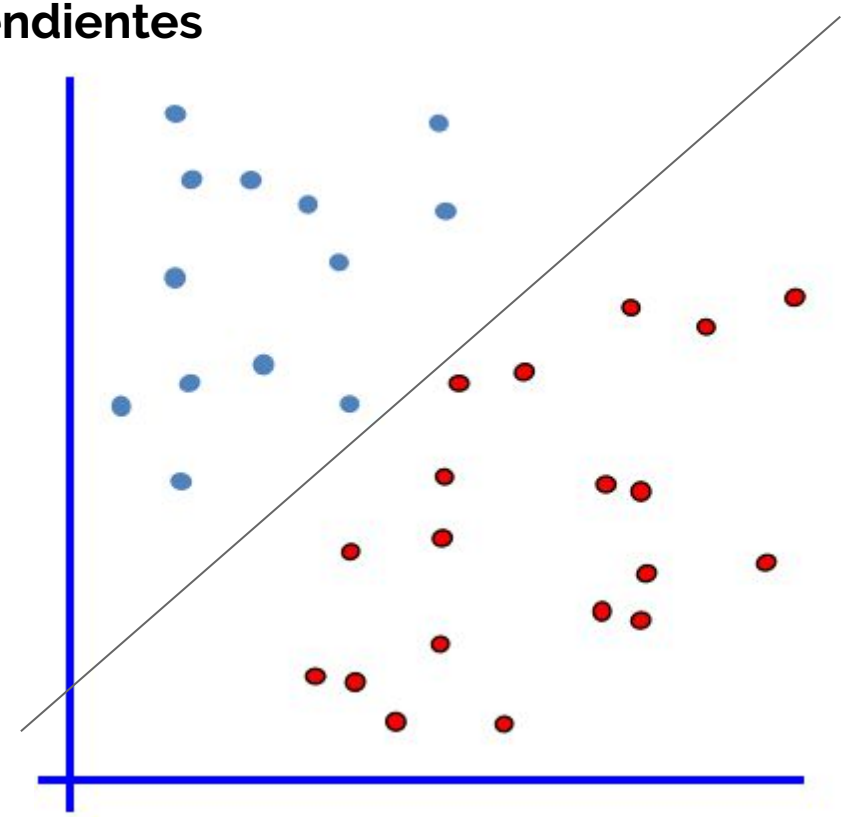
Algoritmos: Regresión Linear

Clasificación y regresión

Relación entre variables

- **Variable dependiente**
- **Conjunto de variables independientes**
  - **Continuas**
  - **Discretas**

$$f(x)=ax+b$$



Construimos una recta de regresión

Algoritmos: Regresión Linear

Clasificación y regresión

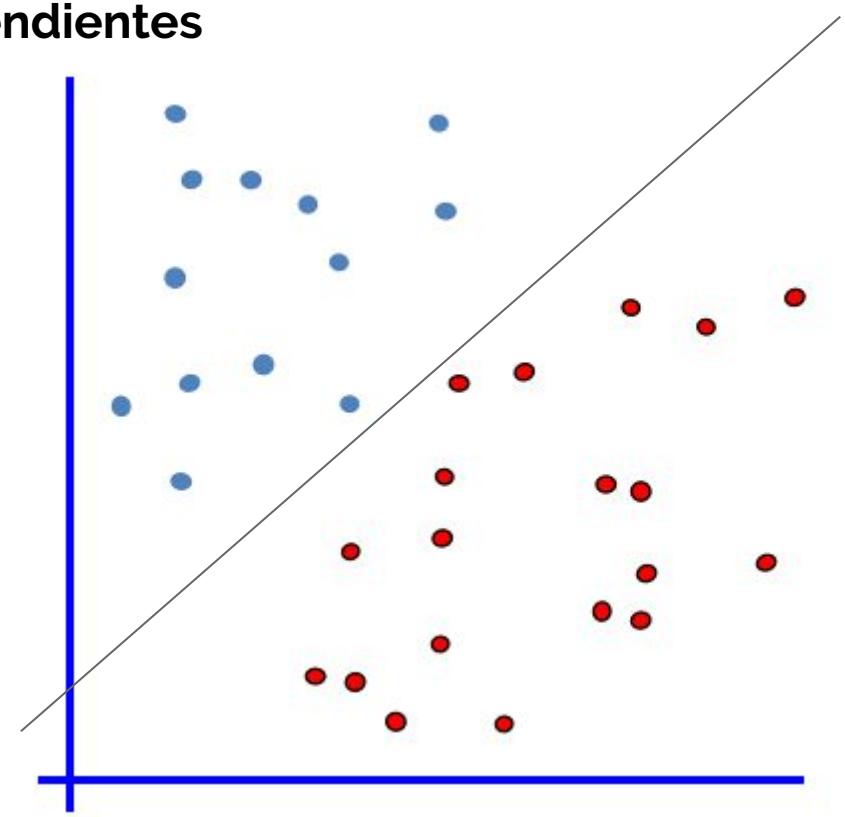
Relación entre variables

- **Variable dependiente**
- **Conjunto de variables independientes**
  - **Continuas**
  - **Discretas**

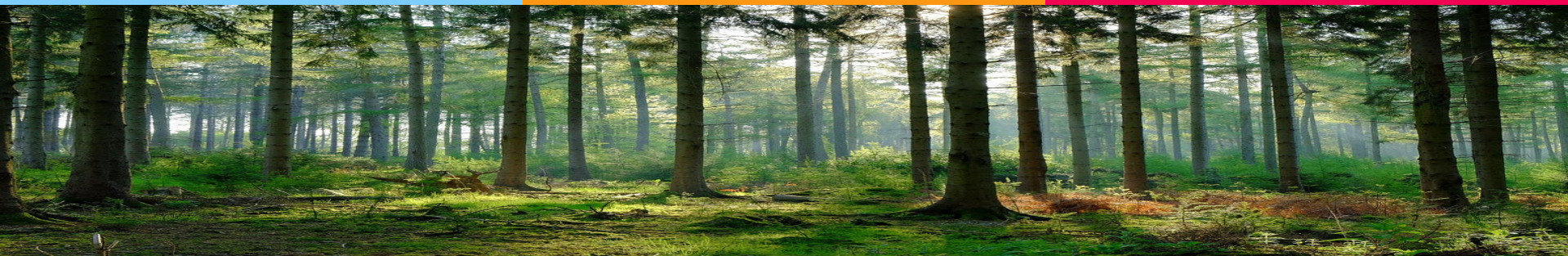
$$f(x)=ax+b$$

calculamos

Construimos una recta de regresión



3.2 Random Forest

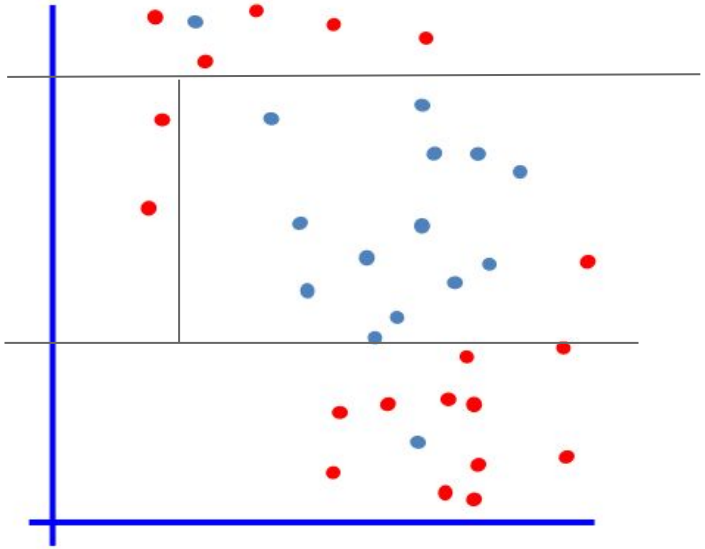


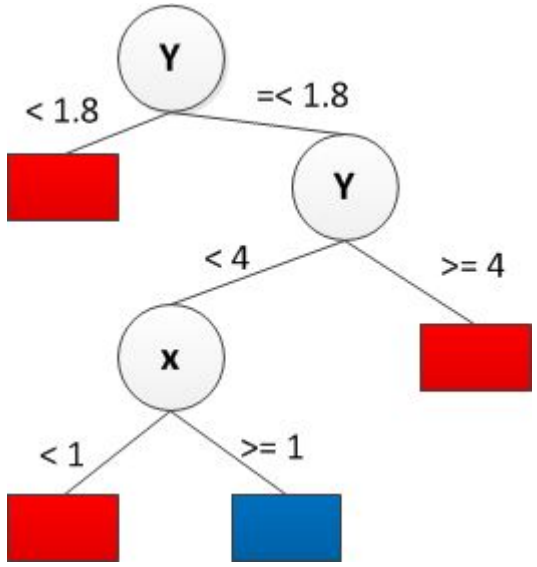
Algoritmos: Random Forest

Clasificación y regresión

Modelo basado en árboles de decisión

- **Nodos ⇒ variables**
- **Hojas ⇒ clases**



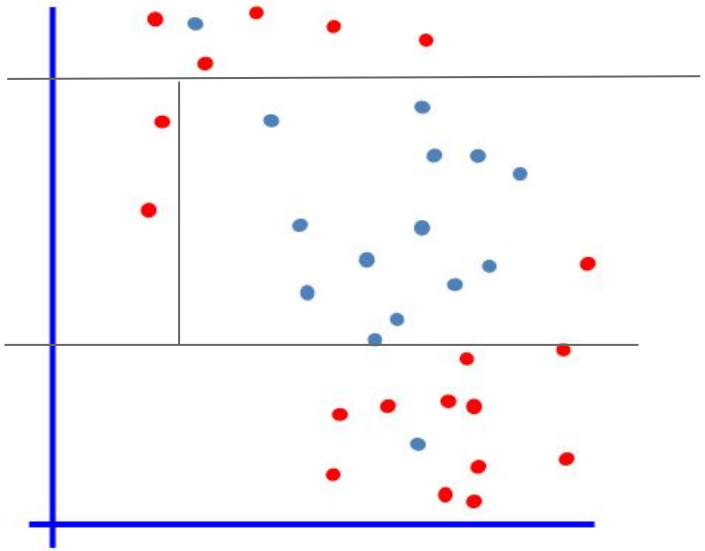


Algoritmos: Random Forest

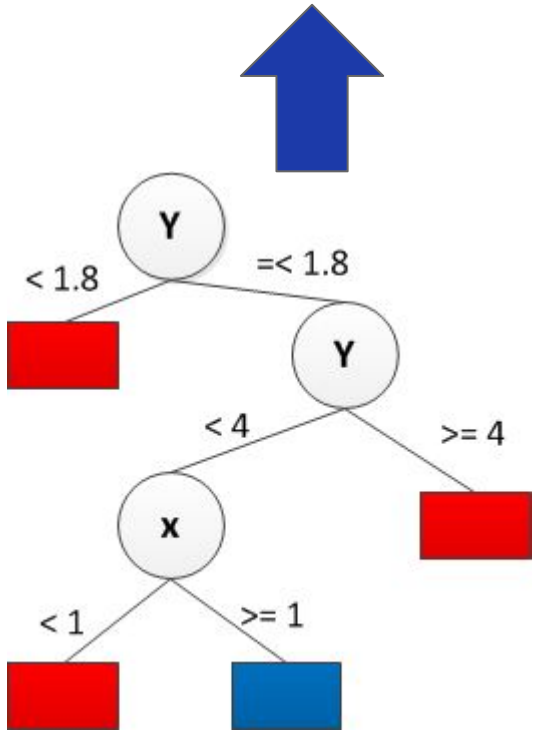
Clasificación y regresión

Modelo basado en árboles de decisión

- **Nodos ⟹ variables**
- **Hojas ⟹ clases**



Hay mas soluciones



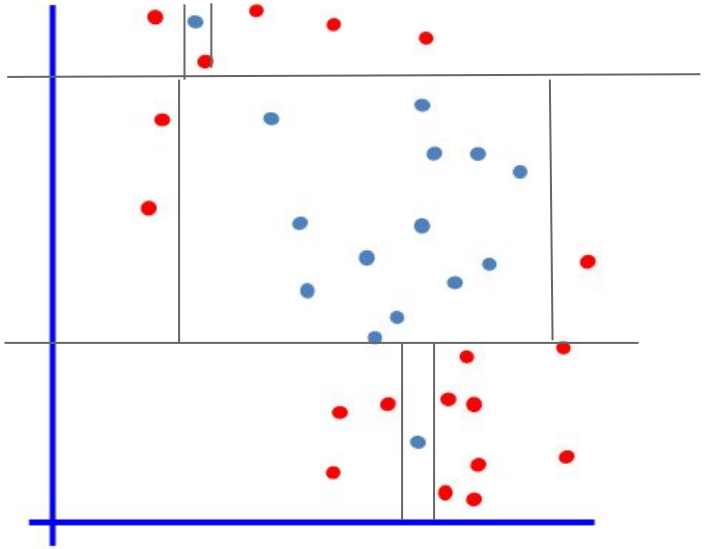
Algoritmos: Random Forest

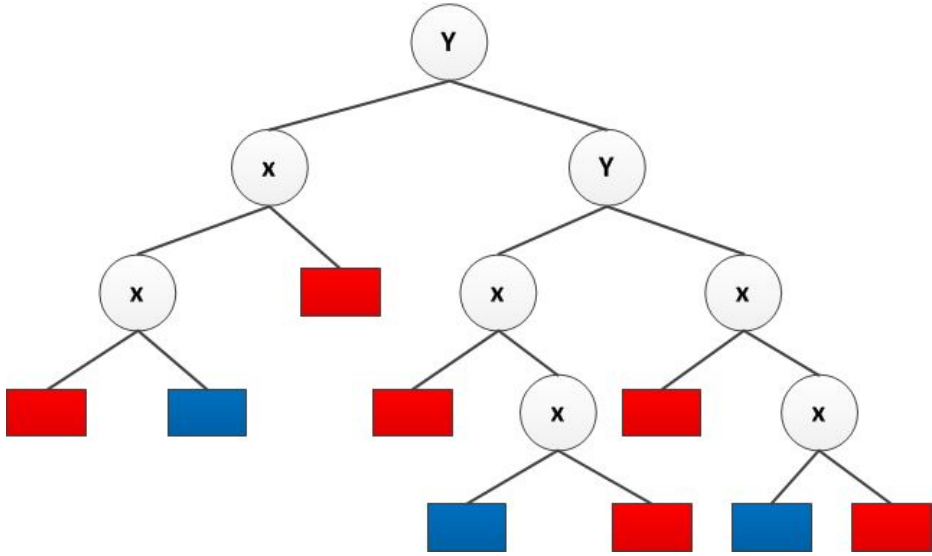
Clasificación y regresión

Modelo basado en árboles de decisión

- **Nodos ⇒ variables**
- **Hojas ⇒ clases**

Hay mas soluciones





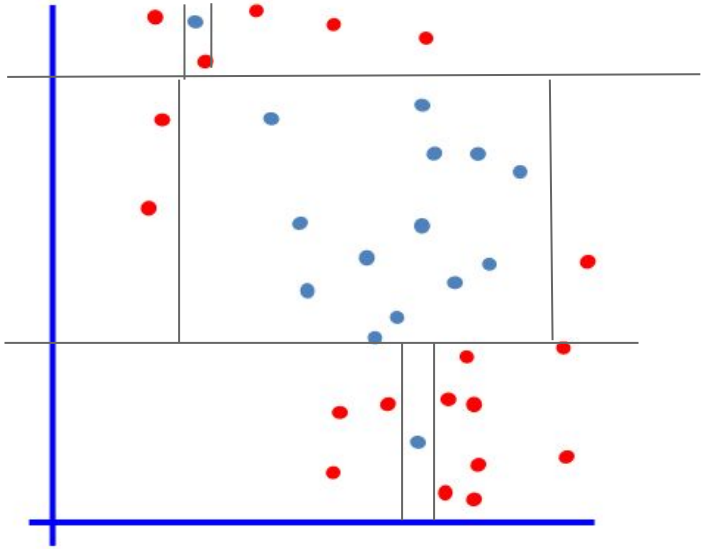
Algoritmos: Random Forest

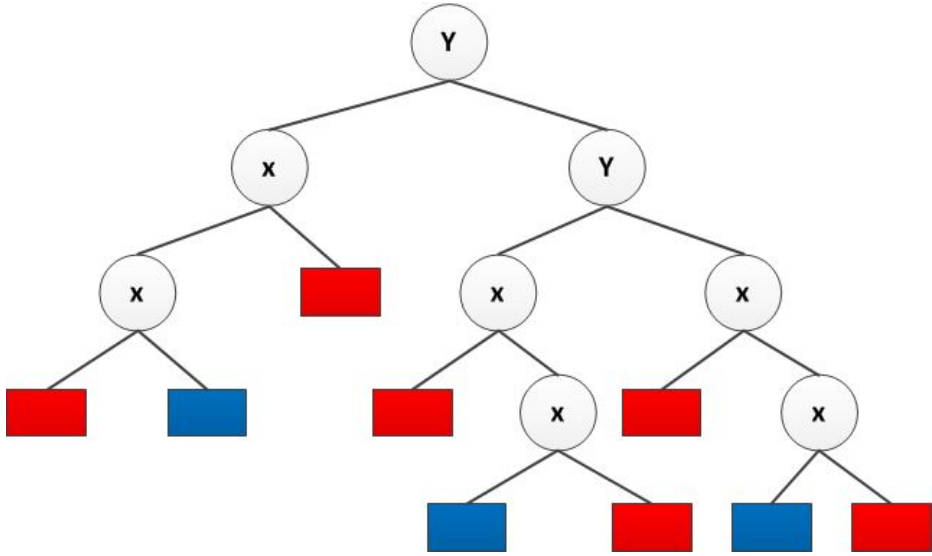
Clasificación y regresión

Modelo basado en árboles de decisión

- **Nodos ⇒ variables**
- **Hojas ⇒ clases**

Hay mas soluciones





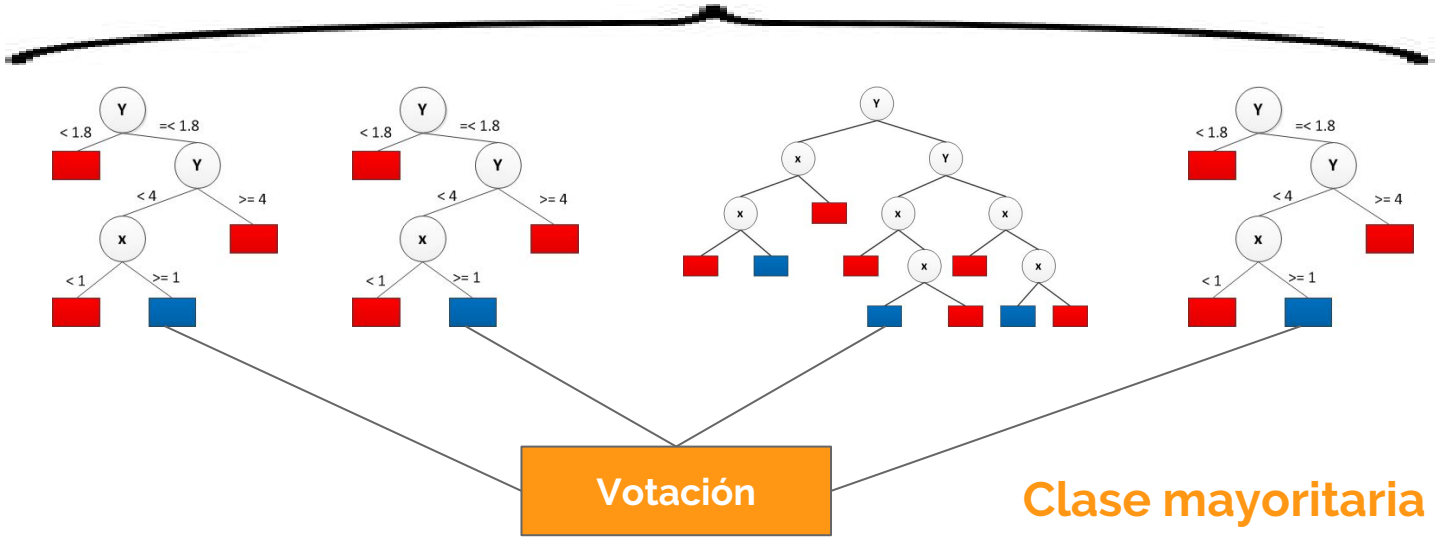
Algoritmos: Random Forest

Clasificación y regresión

Modelo basado en árboles de decisión

- **Nodos ⇒ variables**
- **Hojas ⇒ clases**

Generación aleatoria de conjuntos de entrenamiento



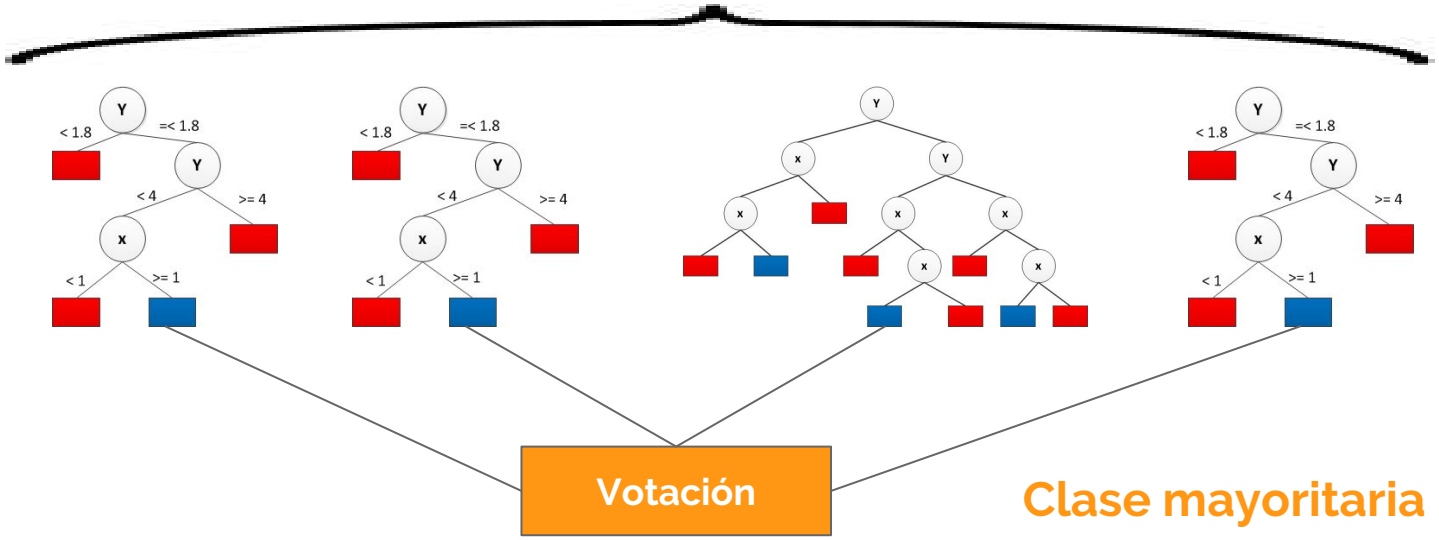
Algoritmos: Random Forest

Clasificación y regresión

Modelo basado en árboles de decisión

- **Nodos ⇒ variables**
- **Hojas ⇒ clases**

Generación aleatoria de conjuntos de entrenamiento

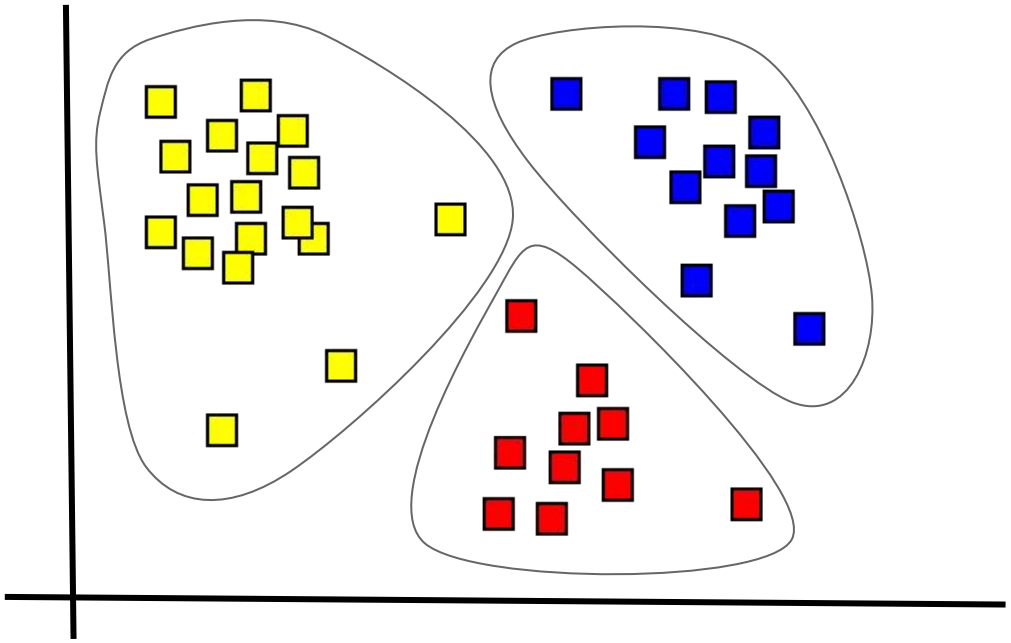


3.3 K-Means

Algoritmos: K-Medias

Clustering

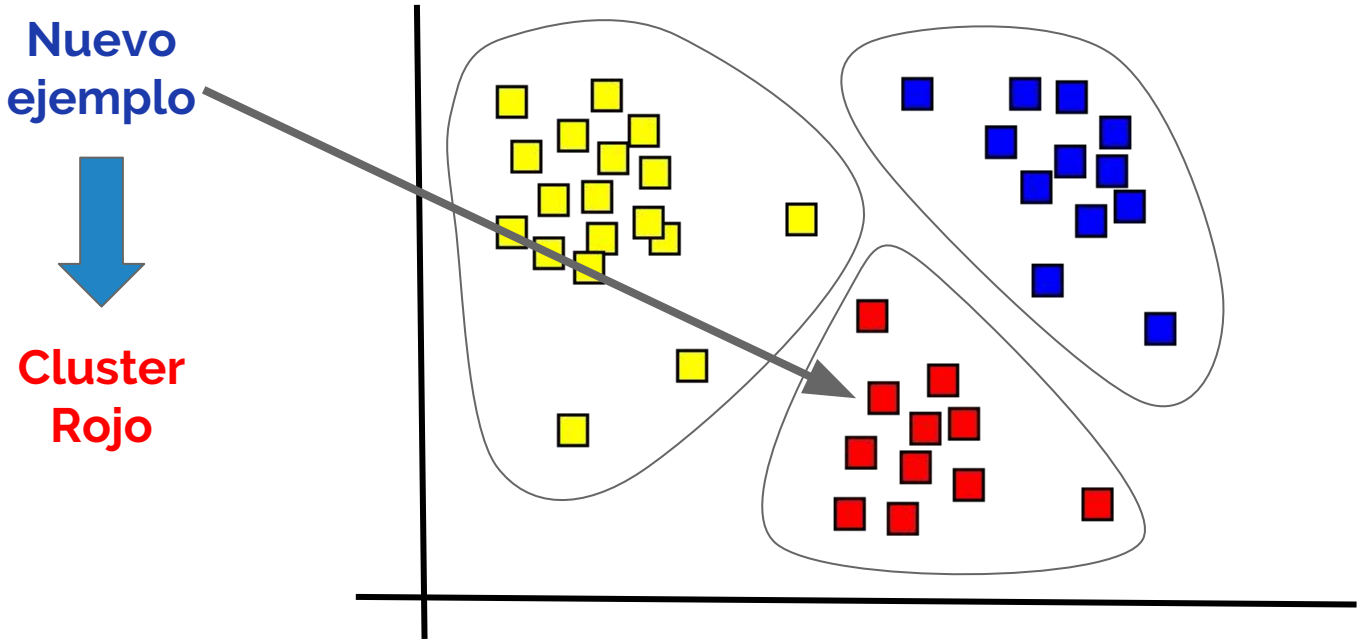
- **Divide el espacio en regiones (clusters)**
- **Cada región posee un centroide**
- **Agrupa los ejemplos de entrenamiento en base a los centroides**



Algoritmos: K-Medias

Clustering

- Divide el espacio en regiones (clusters)
- Cada región posee un centroide
- Agrupa los ejemplos de entrenamiento en base a los centroides



¿Qué pasa con las redes de neuronas?

¿Qué pasa con las redes de neuronas?

“

Tensorflow 101: From theory to practice

Moisés Martínez y Nerea Luis

Sabado 25 - Track 4 - 10:30

5. Prediciendo el precio de una vivienda

Predecir el precio de venta

ID de venta
Precio
Fecha
Localización (ciudad)
Tipo de propiedad
Tiempo de venta
.
.
.

https://data.gov.uk/dataset/land-registry-monthly-price-paid-data

2015 FULL Price Paid Data-Single file 1995-2015 text → 3.6 GB

Entrenamiento → Datos desde 1995 hasta 2014
Test → Datos de 2015

Predecir el precio de venta

ID de venta
Precio
Fecha
Localización (ciudad)
Tipo de propiedad
Tiempo de venta
.
.
.

Etiquetada → **Sí. Tenemos el precio de venta**
Formato → **Mixto. Tenemos que transformar los datos**
Salida → **Valor Real**

Regresión Lineal, Random Forest, Gradient Descent

Predecir el precio de venta

```
with open(path) as infile:
        while True:
                lines = infile.readlines(buffer_size)
                if not lines: break
                for line in lines:
                        data = line.split(',')
                        year = int(data[2][0:4])
                        if (previous_year < 2015):
                                if (previous_year != year):
                                        model.partial_fit(np.array(data_X_train).astype(np.float),
                                        np.array(data_y_train).astype(np.float))
                                        print("Partial fit year %s" % previous_year)
                                        previous_year = year
                                        data_X_train = []
                                        data_Y_train = []
                                data_X_train.append(generateExample(data, cities, 1))
                                data_y_train.append(data[1])
                        elif (previous_year == 2015):
                                data_X_test.append(generateExample(data, cities, mode))
                                data_y_test.append(data[1])
prediction = model.predict(np.array(data_X_test).astype(np.float))
```

Predecir el precio de venta

```
with open(path) as infile:
        while True:
                lines = infile.readlines(buffer_size)
                if not lines: break
                for line in lines:
                        data = line.split(',')
                        year = int(data[2][0:4])
                        if (previous_year < 2015):
                                if (previous_year != year):
                                        model.partial_fit(np.array(data_X_train).astype(np.float),
                                        np.array(data_y_train).astype(np.float))
                                        print("Partial fit year %s" % previous_year)
                                        previous_year = year
                                        data_X_train = []
                                        data_Y_train = []
                                data_X_train.append(generateExample(data, cities, 1))
                                data_y_train.append(data[1])
                        elif (previous_year == 2015):
                                data_X_test.append(generateExample(data, cities, mode))
                                data_y_test.append(data[1])
prediction = model.predict(np.array(data_X_test).astype(np.float))
```

Entrenamiento

Predecir el precio de venta

```
with open(path) as infile:
        while True:
                lines = infile.readlines(buffer_size)
                if not lines: break
                for line in lines:
                        data = line.split(',')
                        year = int(data[2][0:4])
                        if (previous_year < 2015):
                                if (previous_year != year):
                                        model.partial_fit(np.array(data_X_train).astype(np.float),
                                        np.array(data_y_train).astype(np.float))
                                        print("Partial fit year %s" % previous_year)
                                        previous_year = year
                                        data_X_train = []
                                        data_Y_train = []
                                data_X_train.append(generateExample(data, cities, 1))
                                data_y_train.append(data[1])
                        elif (previous_year == 2015):
                                data_X_test.append(generateExample(data, cities, mode))
                                data_y_test.append(data[1])
prediction = model.predict(np.array(data_X_test).astype(np.float))
```

Transformación

Entrenamiento

Predecir el precio de venta

```
with open(path) as infile:
        while True:
                lines = infile.readlines(buffer_size)
                if not lines: break
                for line in lines:
                        data = line.split(',')
                        year = int(data[2][0:4])
                        if (previous_year < 2015):
                                if (previous_year != year):
                                        model.partial_fit(np.array(data_X_train).astype(np.float),
                                        np.array(data_y_train).astype(np.float))
                                        print("Partial fit year %s" % previous_year)
                                        previous_year = year
                                        data_X_train = []
                                        data_Y_train = []
                                data_y_train = []data_X_train.append(generateExample(data, cities, 1))
                                data_y_train.append(data[1])
                        elif (previous_year == 2015):
                                data_X_test.append(generateExample(data, cities, mode))
                                data_y_test.append(data[1])
prediction = model.predict(np.array(data_X_test).astype(np.float))
```

Generación de tests

Predecir el precio de venta

```
with open(path) as infile:
        while True:
                lines = infile.readlines(buffer_size)
                if not lines: break
                for line in lines:
                        data = line.split(',')
                        year = int(data[2][0:4])
                        if (previous_year < 2015):
                                if (previous_year != year):
                                        model.partial_fit(np.array(data_X_train).astype(np.float),
                                        np.array(data_y_train).astype(np.float))
                                        print("Partial fit year %s" % previous_year)
                                        previous_year = year
                                        data_X_train = []
                                        data_Y_train = []
                                data_y_train = []data_X_train.append(generateExample(data, cities, 1))
                                data_y_train.append(data[1])
                        elif (previous_year == 2015):
                                data_X_test.append(generateExample(data, cities, mode))
                                data_y_test.append(data[1])
prediction = model.predict(np.array(data_X_test).astype(np.float))
```

Predicción

Predecir el precio de venta

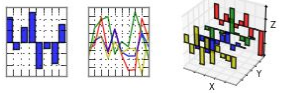
```
with open(path) as infile:
        while True:
                lines = infile.readlines(buffer_size)
                if not lines: break
                for line in lines:
                        data = line.split(',')
                        year = int(data[2][0:4])
                        if (previous_year < 2015):
                                if (previous_year != year):
                                        model.partial_fit(np.array(data_X_train).astype(np.float),
                                        np.array(data_y_train).astype(np.float))
                                        print("Partial fit year %s" % previous_year)
                                        previous_year = year
                                        data_X_train = []
                                        data_Y_train = []
                                data_y_train = []data_X_train.append(generateExample(data, cities, 1))
                                data_y_train.append(data[1])
                        elif (previous_year == 2015):
                                data_X_test.append(generateExample(data, cities, mode))
                                data_y_test.append(data[1])
prediction = model.predict(np.array(data_X_test).astype(np.float))
```

Predicción

¿Qué mas puedo usar?

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



SciPy

Stanford University NLP Natural Language Processing

TensorFlow™

APACHE Spark™

mahout

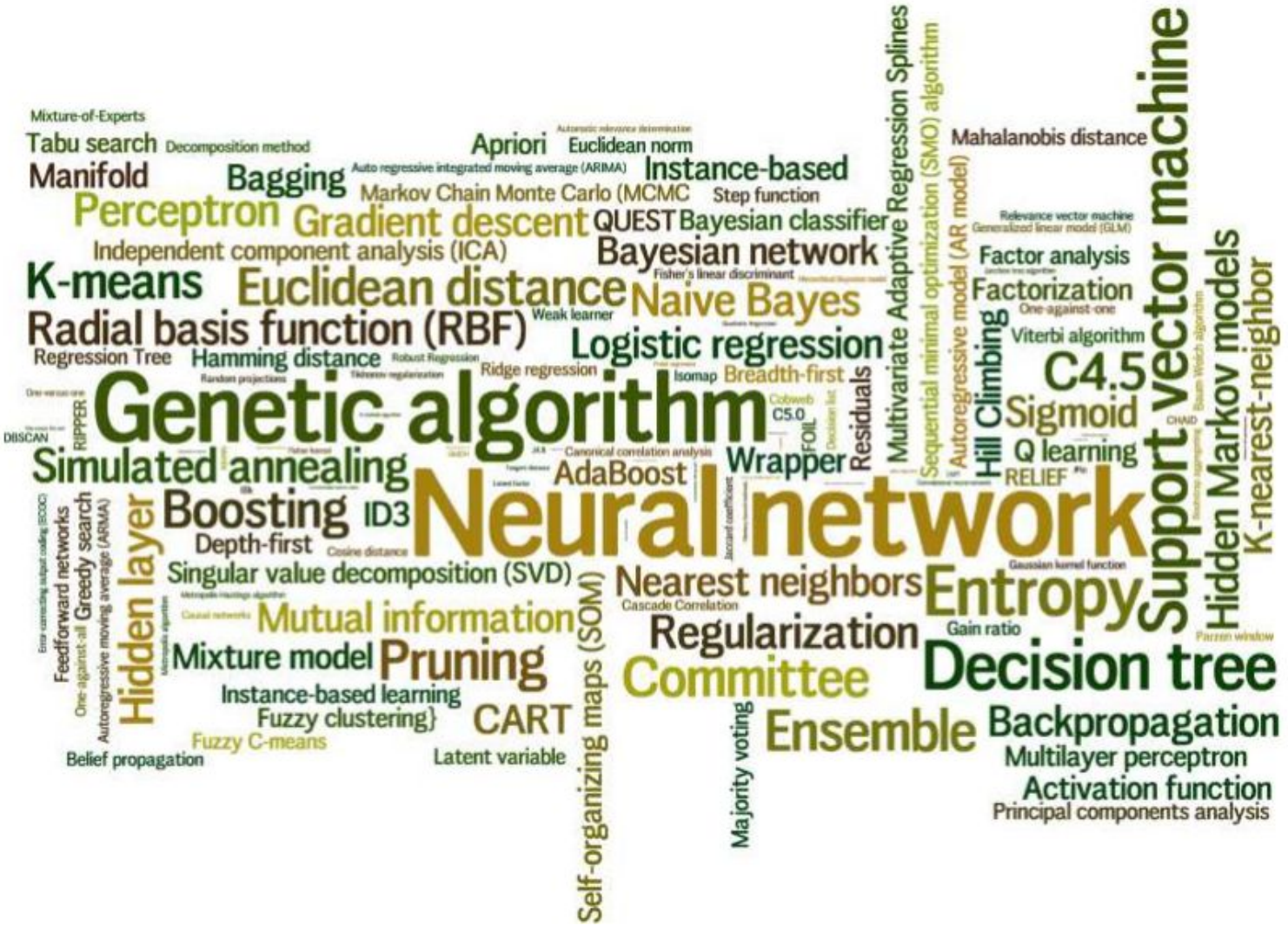
scikit learn

amazon web services™

IBM Watson

Google Cloud Platform

Intentar usarlos todos



T3chFest is coming

http://t3chfest.uc3m.es

Envianos tu propuesta

Muchas Gracias!

Preguntas?

Puedes encontrarme:
momartinm@gmail.com
@moisipm

https://github.com/momartinm/codemotion2017-MachineLearning