

Machine Learning

Hackathon EURUS 2021

José Carlos Rangel
jose.rangel@utp.ac.pa

FISC-UTP

13 de noviembre de 2021



Machine Learning

1

- Machine Learning
 - Definición
 - Enfoques

Machine Learning

Section 1.1 | Definición

Theorem

Machine Learning es la ciencia (y arte) de programar computadores de manera que estos puedan aprender a partir de los datos.

Aurélien Géron



Definición General

Theorem

Machine Learning es el campo de estudio que da a los computadores la habilidad de aprender sin ser explícitamente programados.

Arthur Samuel, 1959



Definición orientada a Ingeniería

Theorem

Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P , si su desempeño en T , al ser medido por P , mejora la experiencia E .

Tom Mitchell, 1997.



Machine Learning

- Un ejemplo clásico de ML es el filtro de *spam* del correo electrónico.
- Su misión es detectar el correo basura en nuestro buzón.
- Según los ejemplos dados(etiquetados) por el usuario los cuales pueden ser: correos *spam*, o correo regular (*ham*).



Machine Learning

- Estos ejemplos son llamados conjunto de entrenamiento (*training set*).
- Cada ejemplo es llamado una instancia de entrenamiento (*training instance*) o muestra.
- Basándonos de la definición de Mitchell
 - T es el filtrado de correos.
 - E es el conjunto de entrenamiento
 - P sería la tasa de correos correctamente clasificados.
- La medida de desempeño P usualmente elegida para los sistemas de clasificación es la precisión.



Ejemplo SPAM I

Posibles reglas para el sistema de filtrado de SPAM

- Si aparecen los símbolos “¡!”, entonces SPAM. Si no, no SPAM.
- Si aparece la palabra “oferta”, entonces SPAM. Si no, no SPAM.
- Si aparece la palabra “oferta” 4 veces y los símbolos “¡!” 5 veces, entonces spam. Si no, no spam.
- Si el número de veces que aparece “oferta” más el número de apariciones de “¡!” es mayor que 5, entonces SPAM. Si no, no SPAM.



Ejemplo SPAM II

Hey Carlos. ¿Te apetece sushi mañana?
Avísame.



No Spam



Es spam si aparecen "¡!" y la palabra "oferta"

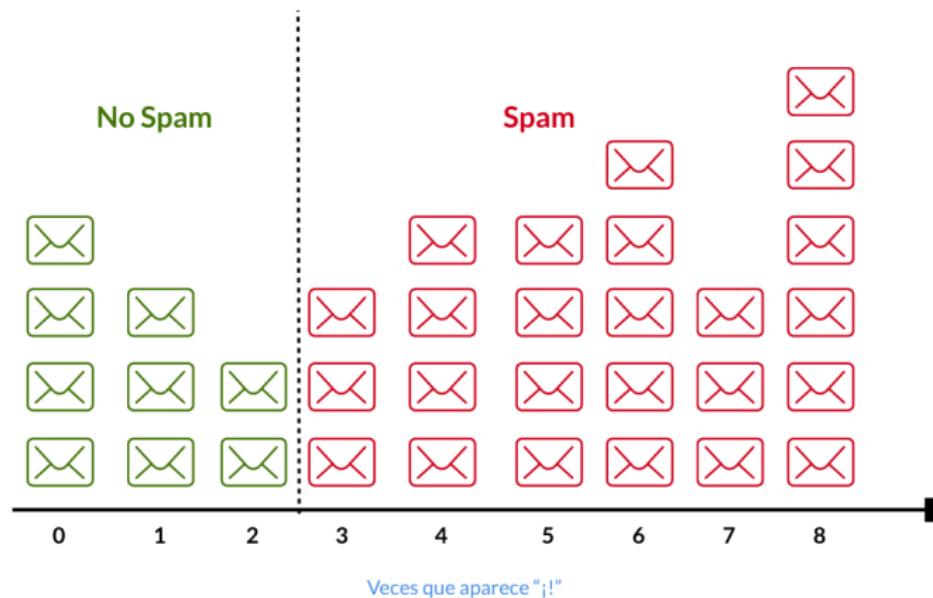
¡Hola! No te pierdas esta oferta única.
¡¡Pocas unidades!!



Spam

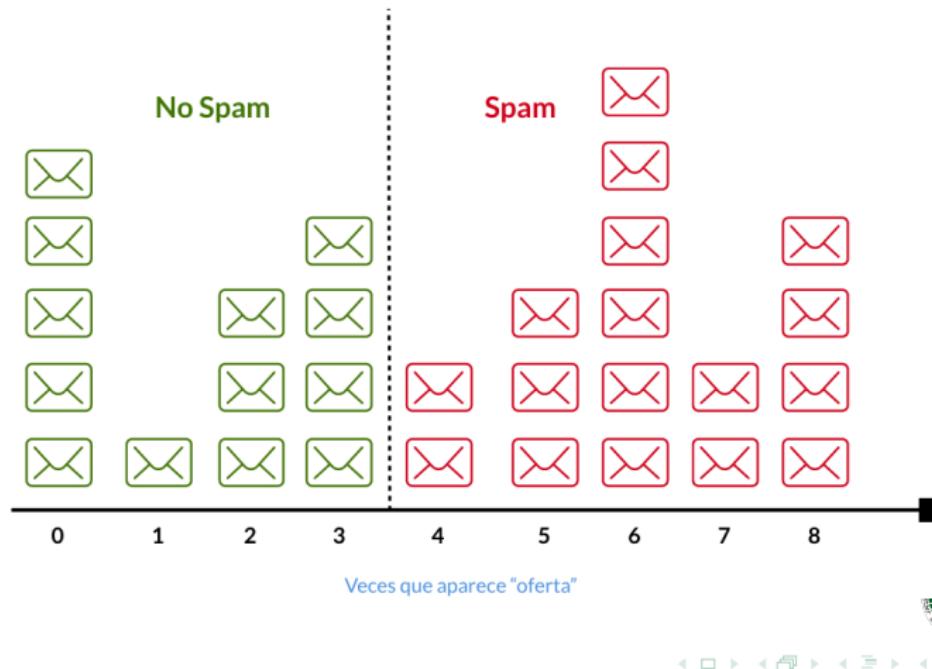
Ejemplo SPAM III

Si aparecen los símbolos “¡!” tres o más veces, entonces spam. Si no, no spam.



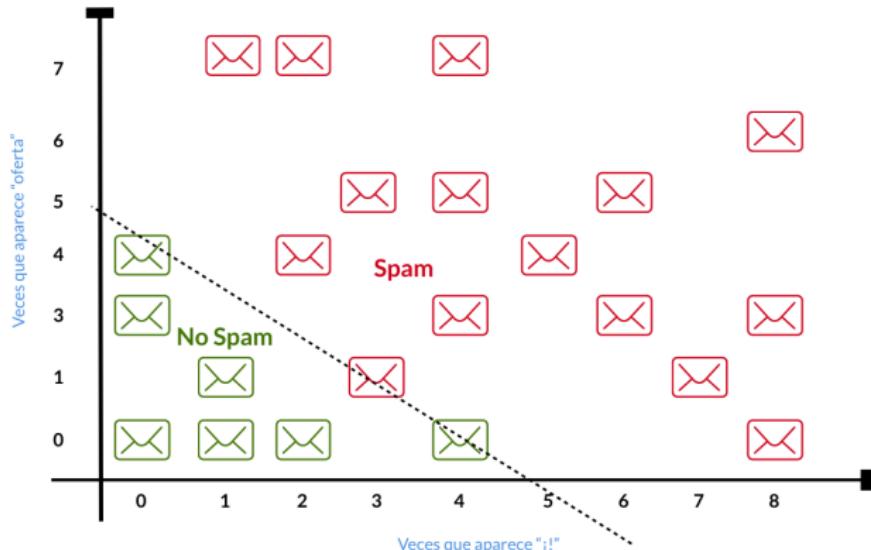
Ejemplo SPAM IV

Si aparece la palabra “oferta” aparece cuatro o más veces, entonces spam. Si no, no spam.



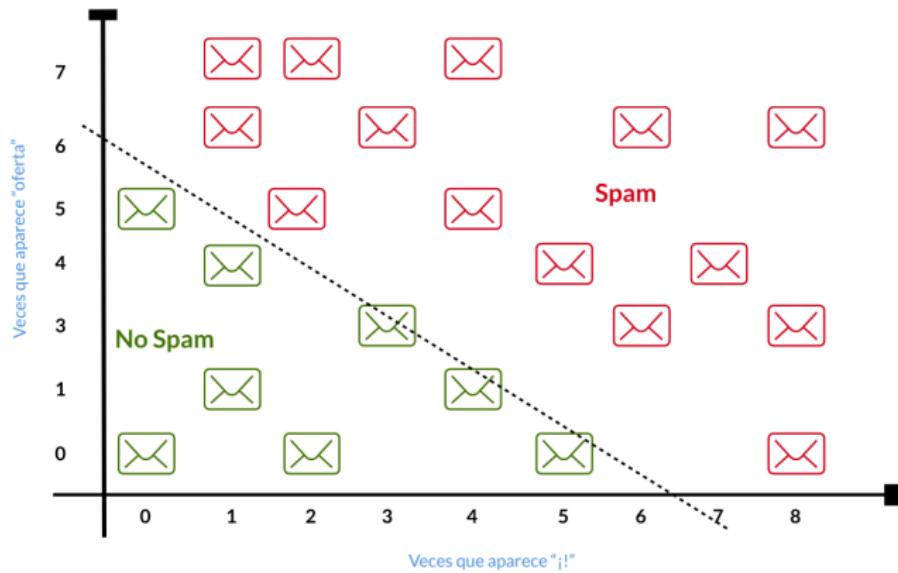
Ejemplo SPAM V

Si aparece la palabra "oferta" 4 veces y los símbolos "¡!" 5 veces, entonces spam. Si no, no spam.



Ejemplo SPAM VI

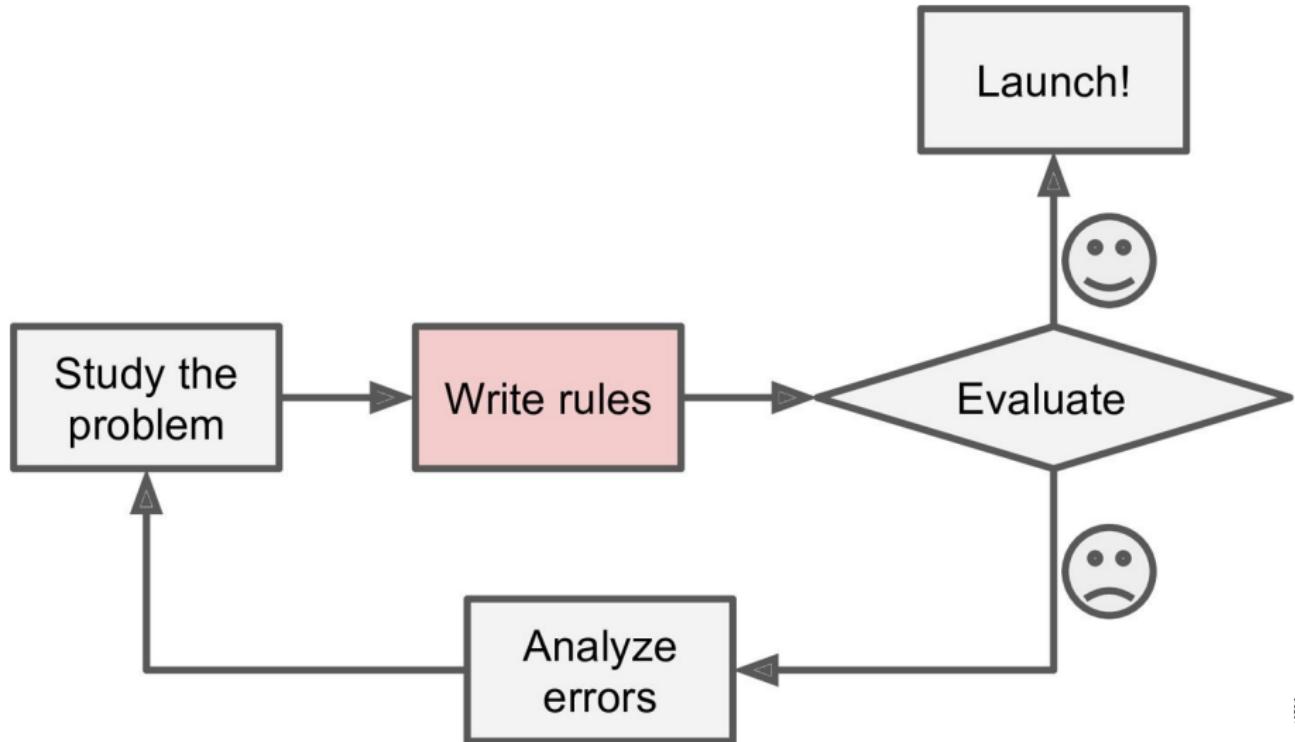
Si el número de veces que aparece "oferta" más el número de apariciones de "¡!" es mayor que 5, entonces spam. Si no, no spam.



Machine Learning

Section 1.2 | Enfoques

Enfoque Tradicional

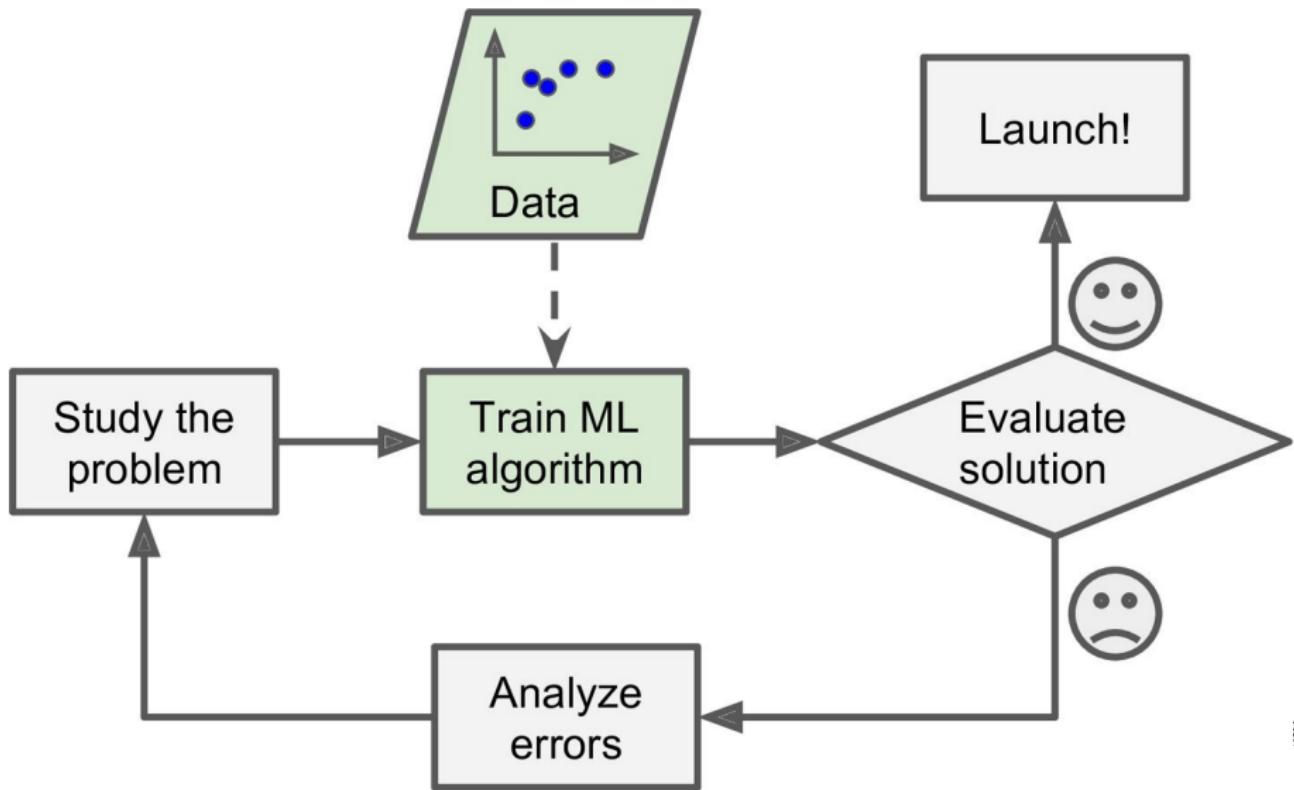


Enfoque Tradicional

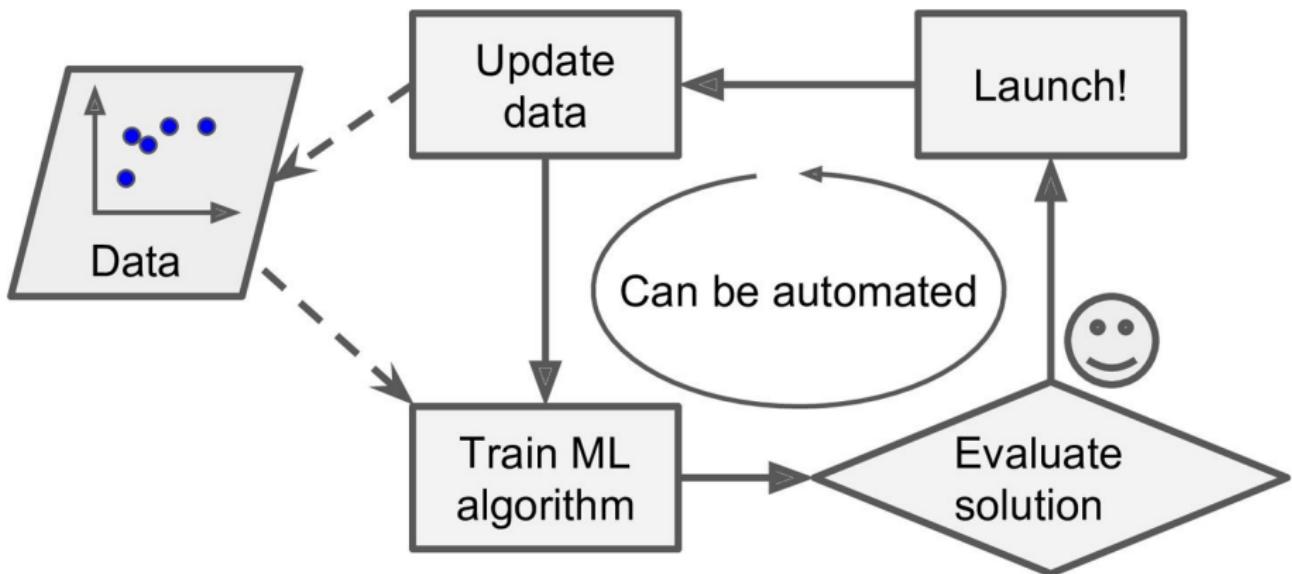
- Este problema no es trivial.
- El programa resultará en una larga y compleja lista de reglas.
- Esta lista será difícil de mantener.



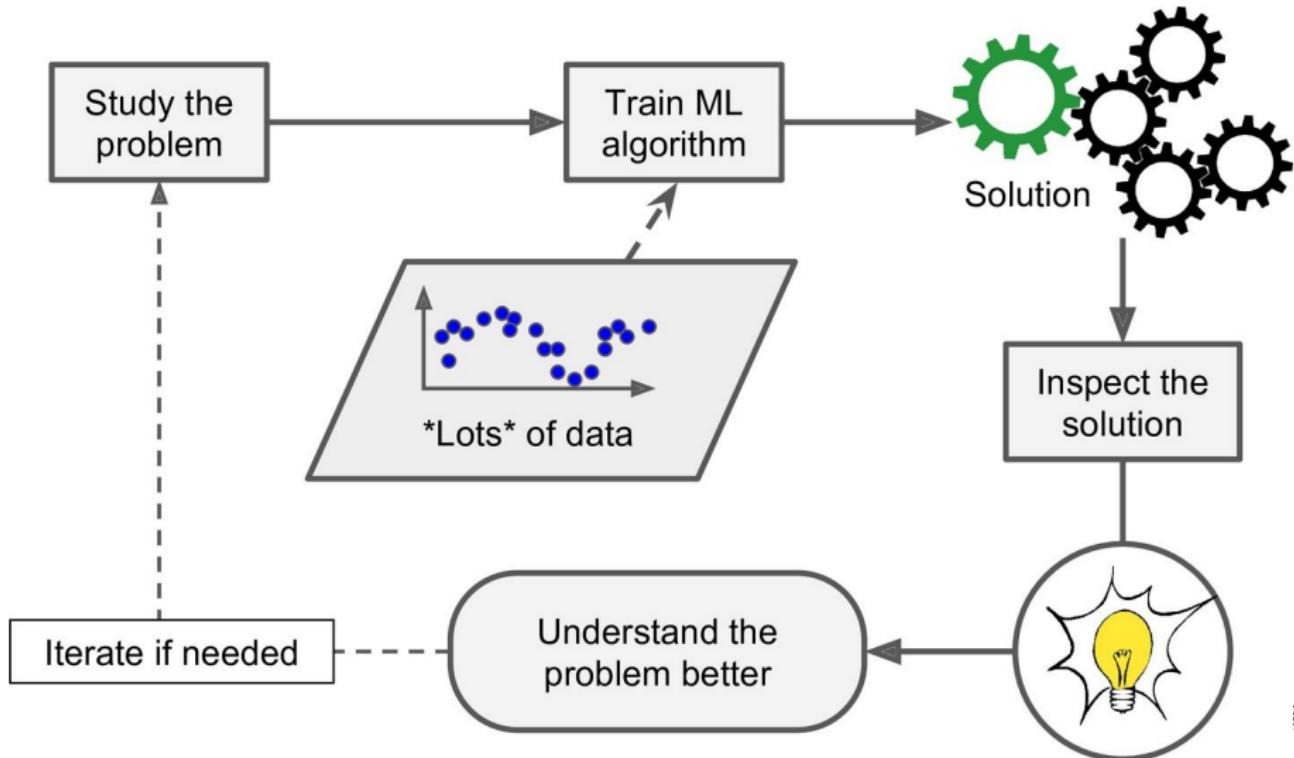
Enfoque ML



Enfoque ML



Enfoque *Data Mining*



En Resumen I

ML es bueno para:

- Problemas para los cuales las soluciones existentes requieren una gran cantidad de procesado manual o una larga lista de reglas: un algoritmo de ML, puede simplificar el código y desempeñarse mejor.
- Problemas complejos para los cuales no hay una buena solución con el enfoque tradicional: la mejor técnica de ML puede encontrar una solución.



En Resumen II

ML es bueno para:

- Ambientes fluctuantes: Los sistemas de ML se pueden adaptar a nuevos datos.
- Obtener información sobre problemas complejos y grandes volúmenes de datos.
- La utilización de ML obliga a los procesos de Ingeniería del Características para lograr una mejor efectividad en los modelos generados.



Machine Learning



2 Conceptos Fundamentales

- Instancias y Descriptores
- Datasets
- Representación de Instancias y Datasets
- Entrenamiento
- Evaluación de Modelos

Conceptos Fundamentales

Section 2.1 | Instancias y Descriptores

Instancias I

Instancias y Descriptores

- Una **instancia** se corresponde con una entidad individual con características diferenciables de otras.
- En un estudio una instancia es básicamente una **muestra** que está bajo análisis



Instancias II

Instancias y Descriptores

- Según el área de estudio una instancia puede ser una imagen, una pista de audio, un texto, una medición de un valor concreto, un conjunto de mediciones.
- Estas se utilizan para que los algoritmos de ML aprendan y puedan generalizar su aprendizaje para identificar otras instancias.



Descriptores I

Instancias y Descriptores

- El ML se construye a base de procesos de **entrenamiento, experimentación y validación** de los modelos entrenados.
- Para la construcción de dichos modelos utilizando los **datasets** seleccionados, la información de estos debe ser procesada en una manera general y uniforme para todo el conjunto.



Descriptores II

Instancias y Descriptores

- Generalmente un algoritmo de ML toma como entrada un conjunto de instancias que ha sido representado a través de un **descriptor**.
- Este busca resaltar de manera general los **atributos característicos** que posee cada instancia de entrenamiento.



Descriptores III

Instancias y Descriptores

- Los descriptores consisten en un arreglo multi-dimensional en la cual cada dimensión describe el mismo atributo para todas las instancias del conjunto de datos.
- Si la dimensión 1 de un descriptor representa la cantidad de pixeles rojos de una imagen, entonces para cada instancia la dimensión 1 representara este valor.
- De esta manera el algoritmo de ML tendrá como entrada el mismo tipo de información para cada instancia en el conjunto de entrenamiento.



Descriptores IV

Instancias y Descriptores

- En el caso de imágenes se utilizan algoritmos descriptores, estos son un tipo de algoritmo que construye o calcula un descriptor para una imagen, crean un vector $n - dimensional$ para cada imagen.
- Para ML se utilizan algoritmos descriptores globales que producen un solo vector para toda la imagen.



Descriptores V

Instancias y Descriptores

Un descriptor puede ser:

Arreglo Textual Conformados por palabras en cada dimensión

Arreglo Numérico Conformados por datos numéricos en cada dimensión

Arreglo Binario Conformados por 1 o 0 en cada dimensión, representando ausencia o presencia de un atributo

Arreglo Semántico Cada dimensión representa un concepto y hay un valor de probabilidad asociada al concepto.

Se debe tener presente que los algoritmos de aprendizaje trabajan solo con información numérica, por lo cual los tipos expuestos anteriormente almacenan la información en formato numérico.



Construcción de un Descriptor I

Instancias y Descriptores

Descripción



Estructura de acero Asiento/respaldo de polietileno Resistente a las manchas Para uso en interiores/exteriores Capacidad máxima de 350lb 58x49,8x82,2cm



Silla THAMES BARNIZ estilo Ercol/Windsor, un clásico del diseño inglés. Acabado natural barnizado. Resiste hasta 270 lb. Medidas: Largo: 42 cm, Ancho: 43 cm, Alto: 84,5 cm. Altura del asiento: 45 cm.



Cuero ecológico Inclinación sincronizada de 2 a 1 Tensión de inclinación ajustable Asiento neumático Brazo rotativo ajustable Capacidad máxima de 250 libras, 107 – 116,5CM



Construcción de un Descriptor II

Instancias y Descriptores

	<i>Altura</i>	<i>PesoSoportado</i>	<i>Material</i>
	0,82	350	Plástico
	0,84	270	Madera
	1,16	250	Cuero



Construcción de un Descriptor III

Instancias y Descriptores

En este caso, este es un descriptor con información mixta y nuestros algoritmos necesitan información numérica para trabajar, por lo cual la ultima columna se puede expresar de manera numérica asignando un numero a cada material, de la siguiente manera.

Material	Numero
Plástico	0
Madera	1
Cuero	2

Por lo cual nuestro descriptor al final tendría la siguiente estructura.



Construcción de un Descriptor IV

Instancias y Descriptores

	<i>Altura</i>	<i>PesoSoportado</i>	<i>Material</i>
	0,82	350	0
	0,84	270	1
	1,16	250	2



Construcción de un Descriptor V

Instancias y Descriptores

Al final la información que entenderá un algoritmo será de la siguiente manera.

<i>Instancia</i>	<i>Altura</i>	<i>PesoSoportado</i>	<i>Material</i>
Silla 1	0,82	350	0
Silla 2	0,84	270	1
Silla 3	1,16	250	2



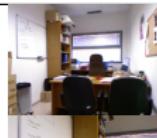
Estructura de un Descriptor Numérico

Instancias y Descriptores

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	...	$l_{ \mathcal{D} }$
Instancia 1	0.97	0.95	0.93	0.91	0	0	0	...	0
Instancia 2	0	0.94	0	0.92	0.93	0.94	0	...	0
Instancia 3	0.91	0	0	0	0.94	0.97	0.93	...	0
...
Instancia n	0	0	0	0	0	0	0	...	0.91



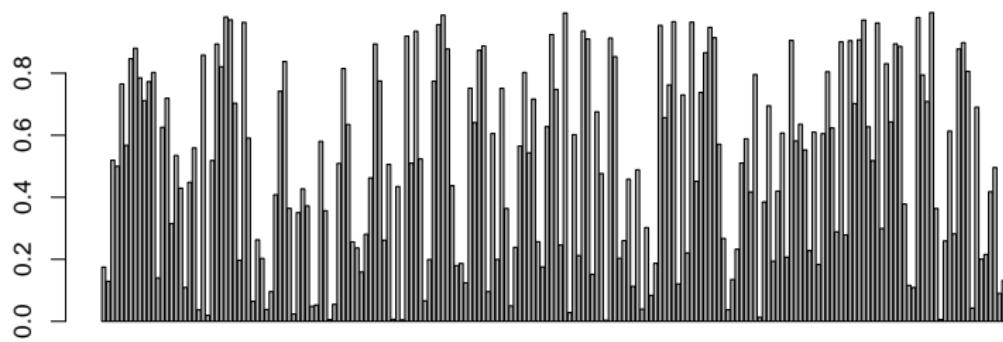
Estructura de un descriptor para imágenes

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	...	$l_{ \mathcal{D} }$
	0.97	0.95	0.93	0.91	0	0	0	...	0
	0	0.94	0	0.92	0.93	0.94	0	...	0
	0.91	0	0	0	0.94	0.97	0.93	...	0
...
	0	0	0	0	0	0	0	...	0.91



Descriptores

- Visualmente un descriptor se puede representar como un histograma donde cada columna representa una dimensión y la altura de la misma, el valor otorgado para ese atributo por el algoritmo.



Conceptos Fundamentales

Section 2.2 | Datasets

Datasets

- Cuando una gran cantidad de datos se reúne, agrupa y organiza en categorías o secuencias, recibe el nombre de *Dataset*.
- Son precisamente estos los que han permitido aumentar el auge del ML en los últimos años.
- En la actualidad existen *Dataset* para casi todo tipo de datos.
- Generalmente se crean por parte de algún equipo de investigación, el cual se propone resolver un problema concreto y para ello requiere una gran cantidad de datos.



Datasets



WordNet

A lexical database for English



Conceptos Fundamentales

Section 2.3 | Representación de Instancias y Datasets

Representación de Instancias y Datasets I

- Una instancia se representa en nuestros algoritmos mediante un descriptor
- Estos descriptores engloban información de cada instancia en sus diferentes dimensiones.
- Un *dataset* se compone generalmente de un conjunto de instancias las cuales pueden estar representados con un conjunto de descriptores.
- Los algoritmos de ML representan estos descriptores en espacios hiper-dimensionales



Representación de Instancias y Datasets II

- Usando estos espacios se trata de generar un modelo que represente de forma fiable la información que contiene el *dataset*
- Por lo cual un algoritmo de ML se enfoca en aprender la relación que existe entre los atributos que tiene cada instancia en su descriptor.



Representación de Instancias y Datasets III

- Cuanto se tratan de descriptores de 2 y 3 dimensiones su representación gráfica es fácil de visualizar, pero para más dimensiones el escenario se complica.
- Los algoritmos de ML se encargan entonces de representar la información internamente y analizarla, sin necesidad de que un usuario deba ver dicha representación.



Visualización de Descriptores I

Representación de Instancias y Datasets

Graficando información del conjunto de sillas

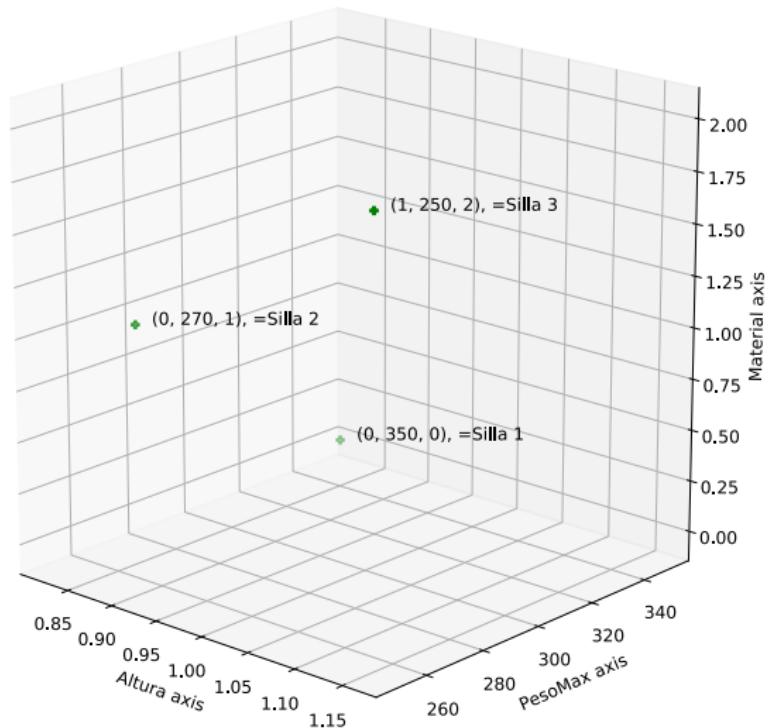
Nuestros ejemplos de descriptores de sillas se pueden representar gráficamente utilizando la información de sus dimensiones, por lo cual un *dataset* como este, se representa utilizando la siguiente gráfica.

<i>Instancia</i>	<i>Altura</i>	<i>PesoSoportado</i>	<i>Material</i>
Silla 1	0,82	350	0
Silla 2	0,84	270	1
Silla 3	1,16	250	2



Visualización de Descriptores II

Representación de Instancias y Datasets



Dataset Iris I

Representación de Instancias y Datasets

Un ejemplo común de *dataset* usado para aprender ML es el *Dataset Iris* este contiene 4 atributos de 3 tipos de flores, con 50 instancias para cada tipo de flor.



iris setosa



iris versicolor



iris virginica



Dataset Iris II

Representación de Instancias y Datasets



iris setosa

Atributos

[Dimensiones]

- sepal length
- sepal width
- petal length
- petal width

Especies [Categorías]

- iris setosa
- iris versicolor
- iris virginica

- Los 4 atributos del *dataset* Iris son tomados como dimensiones del descriptor.
- Las especies son tomadas como clases o categorías de salida.

Muestras de Instancias del Dataset Iris

Largo de sépalo	Ancho de sépalo	Largo de pétalo	Ancho de pétalo	Especies
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
.
5.0	3.3	1.4	0.2	I. setosa
7.0	3.2	4.7	1.4	I. versicolor
.
5.1	2.5	3.0	1.1	I. versicolor
5.7	2.8	4.1	1.3	I. versicolor
.
7.1	3.0	5.9	2.1	I. virginica
6.3	2.9	5.6	1.8	I. virginica



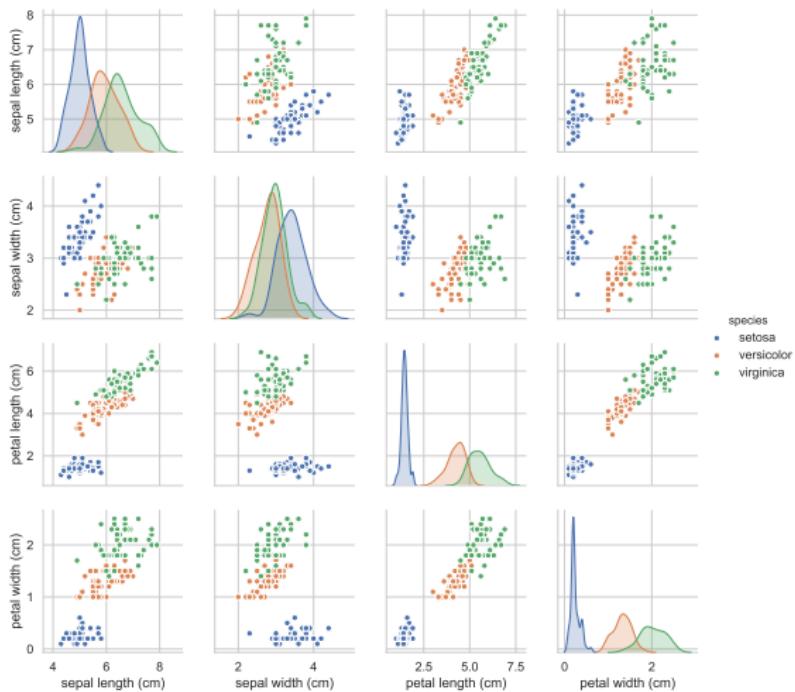
Gráficas del *Dataset Iris* I

Representación de Instancias y Datasets

Al tener 4 dimensiones su graficación visual es compleja, por lo cual para estudiar el *dataset* generalmente se hacen procesos de reducción de dimensiones y también la graficación en pares de datos.

Gráficas del *Dataset Iris II*

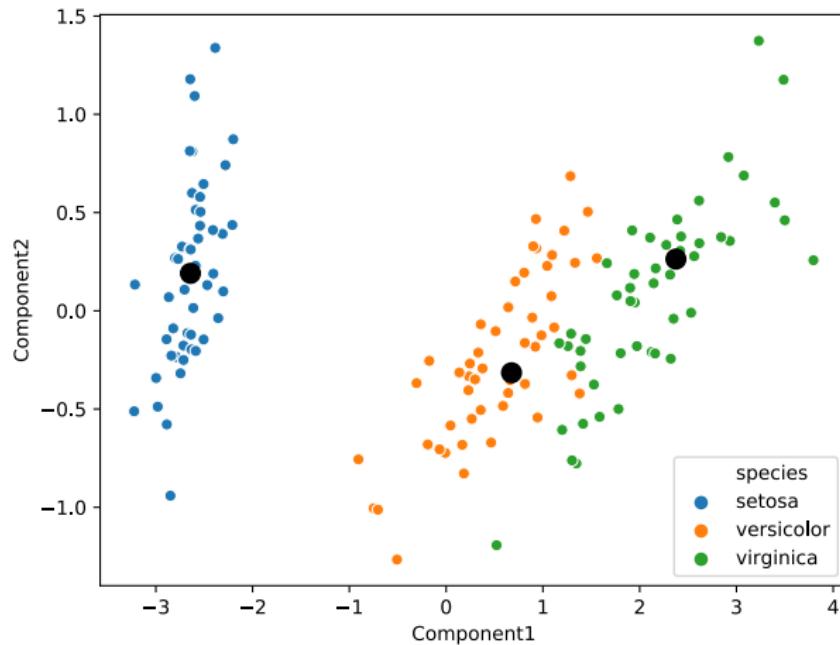
Representación de Instancias y Datasets



Gráficas del *Dataset Iris* III

Representación de Instancias y Datasets

Representación Gráfica utilizando la reducción de dimensiones.



Conceptos Fundamentales

Section 2.4 | Entrenamiento

Entrenamiento I

Conceptos Fundamentales



División en tres Sets

- Trainig Data: entrena el modelo
- Validation Data: Permite ajustar los parámetros
- Test Data: Conjunto de datos de prueba



Entrenamiento II

Conceptos Fundamentales

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

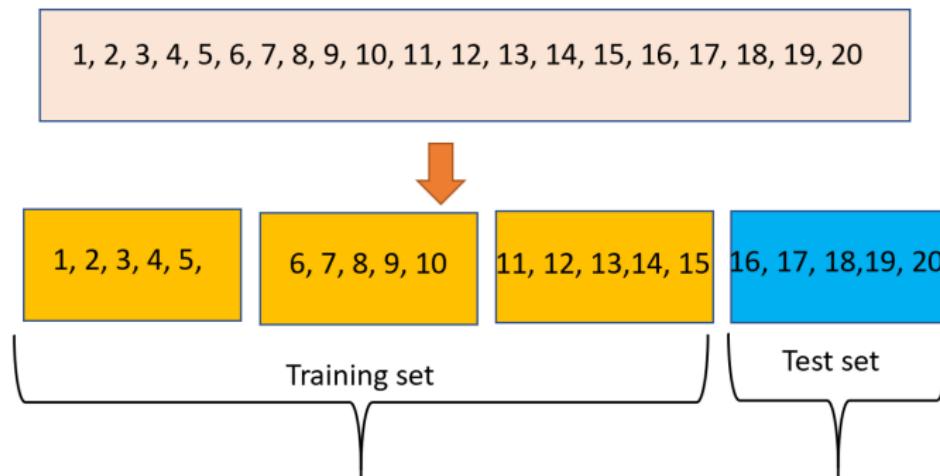


Utilización Completa



Entrenamiento III

Conceptos Fundamentales

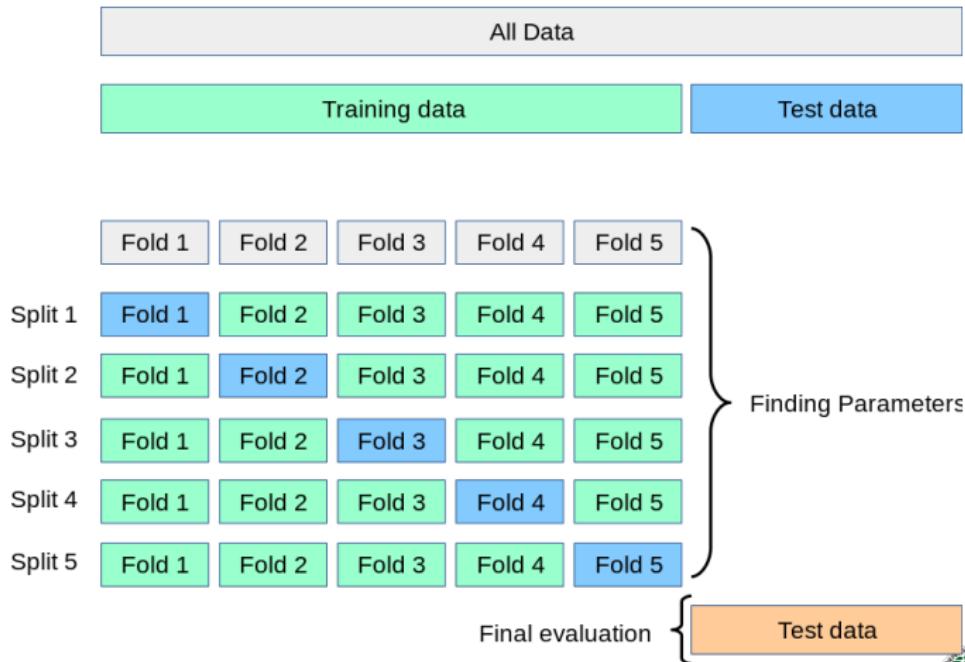


División en 2 Sets (Split)



Entrenamiento IV

Conceptos Fundamentales



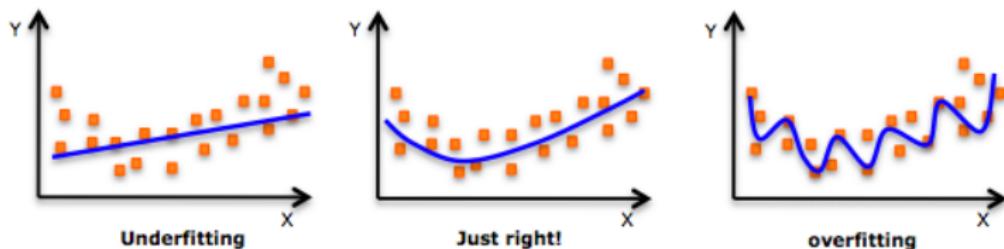
Croosvalidation



Entrenamiento V

Conceptos Fundamentales

Problemas al Entrenar



- Sobre-Aprendizaje (overfitting): Significa que el modelo que entrenamos ha entrenado “demasiado bien” y ahora se ajusta demasiado bien al conjunto de datos de entrenamiento. [▶ Video Overfitting](#)
- Pobre-Aprendizaje(underfitting): Significa que el modelo no se ajusta a los datos de entrenamiento y, por lo tanto, pierde las tendencias  en los datos.



Entrenamiento VI

Conceptos Fundamentales



**THE BEST WAY TO
EXPLAIN OVERTFITTING**

Overfitting



Conceptos Fundamentales

Section 2.5

Evaluación de Modelos

Medida de Precisión

Conceptos Fundamentales

Precisión del Modelo

El cálculo de la precisión del modelo se hace usando la siguiente ecuación:

$$prec = \frac{\text{numero_de_aciertos}}{\text{numero_total_de_instancias_en_el_dataset}}$$

Esta indica cuantas instancias fueron identificadas adecuadamente por el sistema. Esta medida se emplea también para sistemas de clasificación multiclas.



Machine Learning

- 3 Tipos de Aprendizaje
- Aprendizaje Supervisado
 - Aprendizaje No Supervisado
 - Aprendizaje Semi-Supervisado
 - Aprendizaje por Refuerzo

Modelos de Clasificación

Tipos de Aprendizaje

Theorem

Un modelo en ML es un sistema que ha sido entrenado con un conjunto de datos los cuales le permiten aprender las características presentes en dicho conjunto, y a partir de ellas buscarlas en instancias nunca antes vistas, con el objetivo de asignar una categoría a dicha información o calcular un valor relativo a la instancia de entrada.

En otras palabras un modelo aprende a identificar la categoría de una muestra u obtener un valor numérico relacionado, basándose en los atributos identificados y aprendidos de los datos de entrenamiento.

Tipos de Aprendizaje

Section 3.1 | Aprendizaje Supervisado

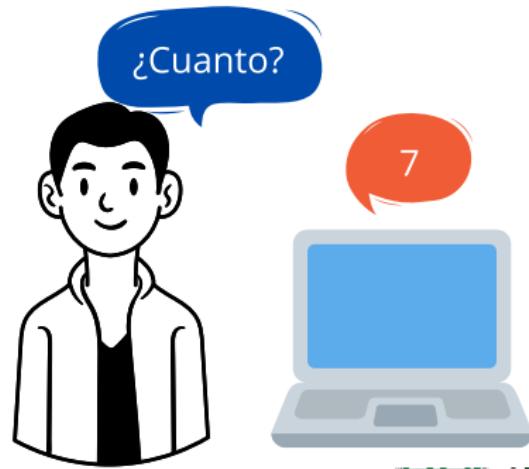
Enfoques de Aprendizaje Supervisado I

Tipos de Aprendizaje

Clasificación

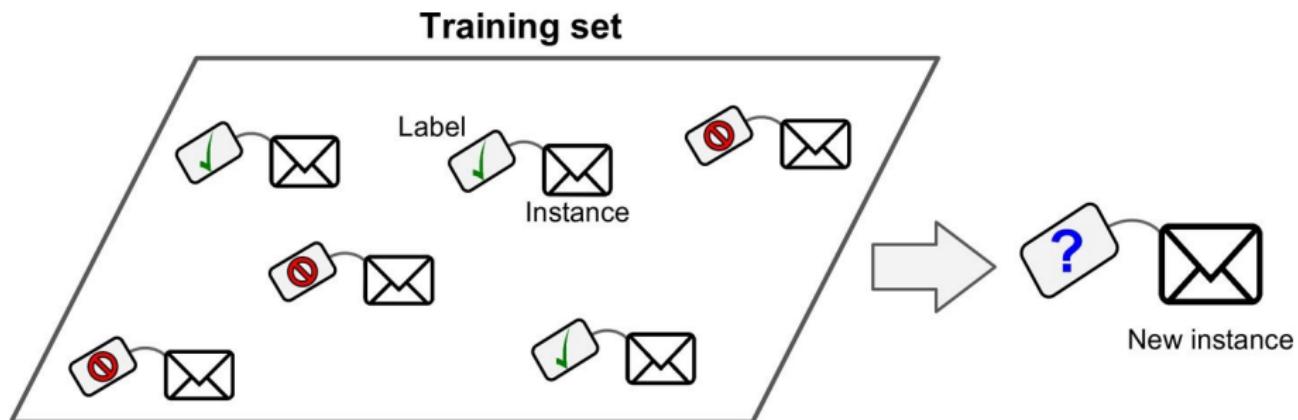


Regresión Lineal



Aprendizaje Supervisado I

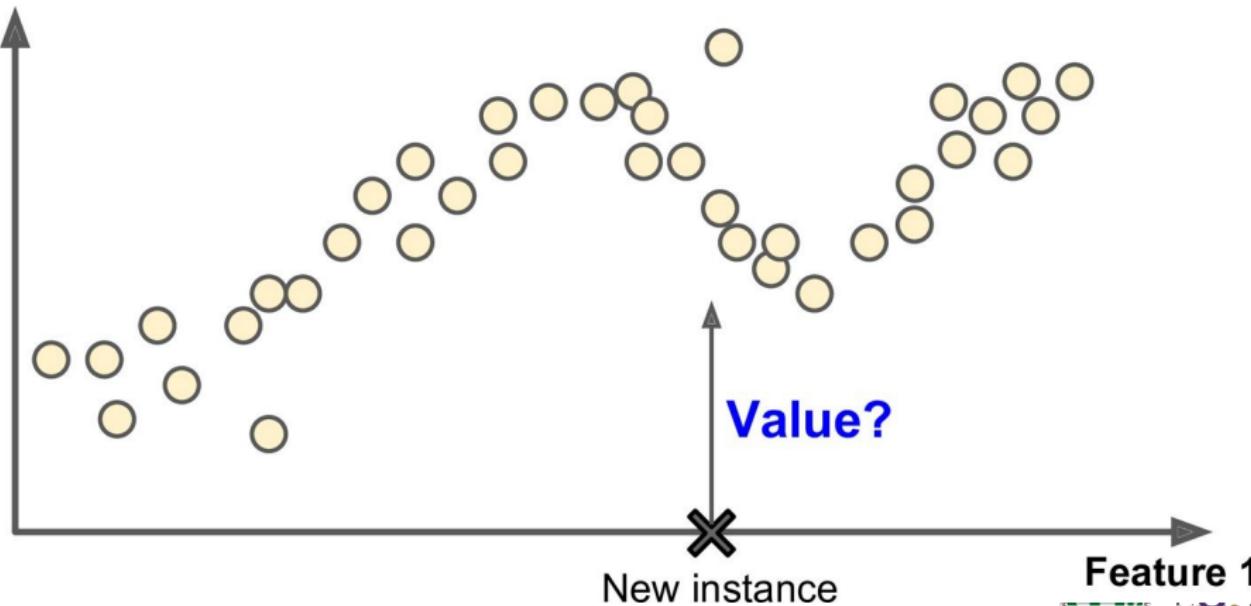
Tipos de Aprendizaje



Regresión I

Tipos de Aprendizaje

Value



New instance

Feature 1

Sistemas Supervisados más importantes

Tipos de Aprendizaje

- Regresión Lineal
- Regresión Logística
- k-*Nearest Neighbors* (k-Vecinos más cercanos) - knn
- Maquinas de Vectores de Soporte (*Support Vectors Machines-SVM*)
- Árboles de Decisión y Bosques Aleatorios (*Decision Trees and Random Forest*)
- Redes Neuronales Artificiales



Aprendizaje Supervisado

Resumen

The infographic is titled "SUPERVISED LEARNING" and defines it as "IN SUPERVISED LEARNING LABELED DATA ARE USED TO TRAIN THE ML ALGORITHM". It highlights two main types: "REGRESSION" and "CLASSIFICATION".
REGRESSION: A regression problem is when the output variable is a real value, such as "PRICE" or "WEIGHT" or "QUANTITY".
CLASSIFICATION: A classification problem is when the output variable is a category, such as "Y" or "N" or "DISEASE" and "NO DISEASE".
Below these, under "TYPES OF ALGORITHMS", are four hexagonal icons: "LINEAR REGRESSION" (orange), "SUPPORT VECTOR REGRESSION" (red), "LOGISTIC REGRESSION" (purple), and "RANDOM FOREST CLASSIFICATION" (pink).
The bottom right corner features the DataScienceLearn logo and a "Save or Regret!" button.

Follow us! Share it!

SUPERVISED LEARNING

IN SUPERVISED LEARNING LABELED DATA ARE USED TO TRAIN THE ML ALGORITHM

TYPES OF SUPERVISED LEARNING

REGRESSION
A REGRESSION PROBLEM IS WHEN THE OUTPUT VARIABLE IS A REAL VALUE, SUCH AS "PRICE" OR "WEIGHT" OR "QUANTITY".

CLASSIFICATION
A CLASSIFICATION PROBLEM IS WHEN THE OUTPUT VARIABLE IS A CATEGORY, SUCH AS "Y" OR "N" OR "DISEASE" AND "NO DISEASE".

TYPES OF ALGORITHMS

LINEAR REGRESSION
SUPPORT VECTOR REGRESSION
LOGISTIC REGRESSION
RANDOM FOREST CLASSIFICATION

Like To Support DataScienceLearn Save or Regret!

Aprendizaje Supervisado

Resumen

Follow us! Share it!

CLASSIFICATION

CLASSIFICATION ANALYSIS IS A PROCESS OF FINDING A FUNCTION WHICH DIVIDES THE DATASET INTO CLASSES BASED ON DIFFERENT PARAMETERS.

TYPES OF ALGORITHMS

- LOGISTIC REGRESSION
- K-NEAREST NEIGHBOURS
- SUPPORT VECTOR MACHINES
- NAÏVE BAYES CLASSIFIER

APPLICATIONS OF CLASSIFICATION

- EMAIL SPAM DETECTION
- HANDWRITING RECOGNITION
- CUSTOMER CHURN PREDICTION
- BREAST CANCER DETECTION

Like To Support DataScienceLearn Save or Regret!

This infographic provides an overview of classification. It starts with a definition: 'Classification analysis is a process of finding a function which divides the dataset into classes based on different parameters.' Below this, four types of algorithms are listed: Logistic Regression, K-Nearest Neighbours, Support Vector Machines, and Naïve Bayes Classifier. The section 'Applications of Classification' shows four examples: Email Spam Detection, Handwriting Recognition, Customer Churn Prediction, and Breast Cancer Detection. The design features a dark background with yellow highlights for sections like 'CLASSIFICATION' and 'APPLICATIONS OF CLASSIFICATION'. Icons for social media sharing and a 'Save or Regret!' button are included at the bottom.

Follow us! Share it!

REGRESSION

REGRESSION ANALYSIS IS A PROCESS OF FINDING THE CORRELATION BETWEEN DEPENDENT & INDEPENDENT VARIABLES IN PREDICTING A CONTINUOUS VALUE

TYPES OF ALGORITHMS

- LINEAR REGRESSION
- POLYNOMIAL REGRESSION
- DECISION TREE REGRESSION
- RANDOM FOREST REGRESSION

APPLICATIONS OF REGRESSION

- CAR PRICE PREDICTION
- HOUSE PRICE PREDICTION
- INSURANCE PREMIUM CHARGES
- WEATHER DATA ANALYSIS

Like To Support DataScienceLearn Save or Regret!

This infographic provides an overview of regression. It starts with a definition: 'Regression analysis is a process of finding the correlation between dependent & independent variables in predicting a continuous value.' Below this, four types of algorithms are listed: Linear Regression, Polynomial Regression, Decision Tree Regression, and Random Forest Regression. The section 'Applications of Regression' shows four examples: Car Price Prediction, House Price Prediction, Insurance Premium Charges, and Weather Data Analysis. The design features a dark background with yellow highlights for sections like 'REGRESSION' and 'APPLICATIONS OF REGRESSION'. Icons for social media sharing and a 'Save or Regret!' button are included at the bottom.



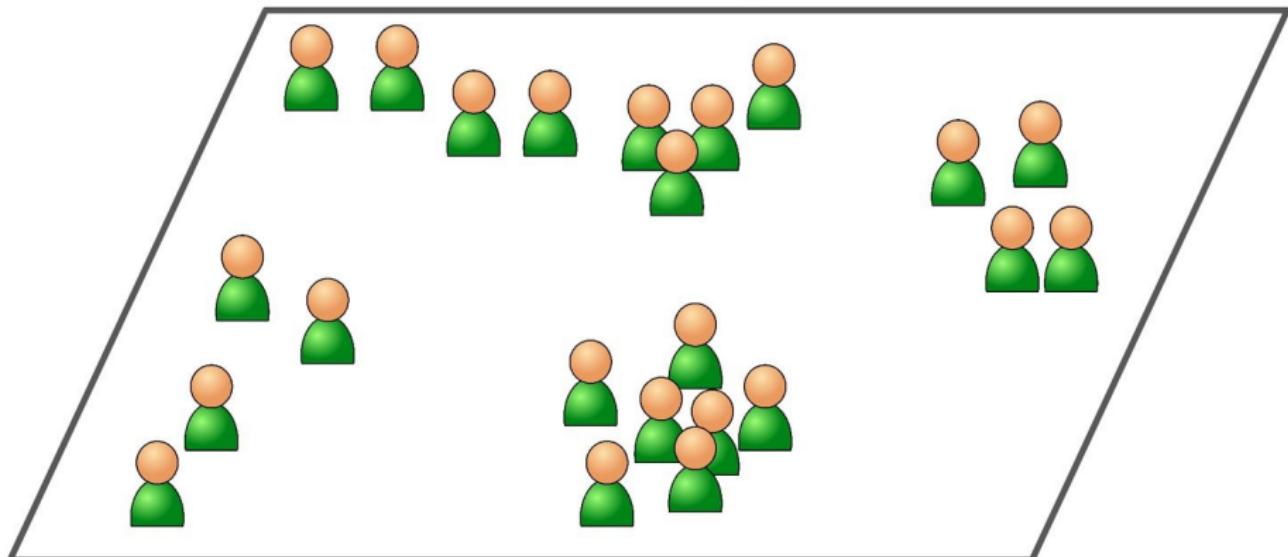
Tipos de Aprendizaje

Section 3.2 | Aprendizaje No Supervisado

Aprendizaje No Supervisado I

Tipos de Aprendizaje

Training set



Sistemas No-Supervisados más importantes

Tipos de Aprendizaje

- Agrupamiento(*Clustering*)
 - k-Means (K-medias)
 - Análisis de Agrupaciones Jerárquico
 - Maximización de la Expectativa
- Reducción y Visualización de la dimensionalidad
 - Análisis de componentes principales (PCA)
 - Kernel PCA
 - *Locally-Linear Embedding* (LLE)
 - *t-distributed Stochastic Neighbor Embedding* (t-SNE)
- Aprendizaje basado en asociación de reglas
 - Apriori
 - Eclat



Detección de Anomalías de funcionamiento

Aprendizaje No Supervisado

Feature 2



Aprendizaje No Supervisado

Resumen

The infographic is titled "UNSUPERVISED LEARNING" and defines it as training with unlabeled data. It then branches into two main types: Clustering and Dimensionality Reduction.

CLUSTERING: Process of grouping similar entities together, and then the grouped data is used to make clusters.

DIMENSIONALITY REDUCTION: Dimensions of the data is reduced to remove the unwanted data from the input.

Below these, there are two sections for "TYPES OF ALGORITHMS" each containing two hexagonal icons:

- K-MEANS CLUSTERING** (Orange hexagon)
- HIERARCHICAL CLUSTERING** (Red hexagon)
- PRINCIPAL COMPONENT ANALYSIS** (Purple hexagon)
- KERNEL PRINCIPAL COMPONENT ANALYSIS** (Pink hexagon)

At the bottom, there are social media links for "Like To Support" and "DataScienceLearn", and a "Save or Regret!" button.

Aprendizaje No Supervisado

Resumen

Follow us! Share it!

CLUSTERING

CLUSTERING IS A PROCESS OF GROUPING SIMILAR DATA OBJECTS IN ONE GROUP THAT IS FAR FROM THE OTHER DISSIMILAR OBJECTS

TYPES OF ALGORITHMS

- K-MEANS CLUSTERING
- HIERARCHICAL CLUSTERING
- DENSITY BASED CLUSTERING
- FUZZY CLUSTERING

APPLICATIONS OF CLUSTERING

- CUSTOMER SEGMENTATION
- ANOMALY DETECTION
- SOCIAL NETWORK ANALYSIS
- IMAGE SEGMENTATION

Like To Support DataScienceLearn Save or Regret!

Follow us! Share it!

DIMENSIONALITY REDUCTION

DIMENSIONALITY REDUCTION IS THE TRANSFORMATION OF DATA FROM HIGH-DIMENSIONAL SPACE TO A LOW-DIMENSIONAL SPACE

TYPES OF ALGORITHMS

- PRINCIPAL COMPONENT ANALYSIS
- LINEAR DISCRIMINANT ANALYSIS
- GENERALIZED DISCRIMINANT ANALYSIS
- SINGULAR VALUE DECOMPOSITION

APPLICATIONS OF DIMENSIONALITY REDUCTION

- IMAGE COMPRESSION
- QUANTITATIVE FINANCE
- COMPUTER NETWORK ATTACKS
- FACIAL RECOGNITION

Like To Support DataScienceLearn Save or Regret!

Tipos de Aprendizaje

Section 3.3 Aprendizaje Semi-Supervisado

Aprendizaje Semi-Supervisado I

Tipos de Aprendizaje

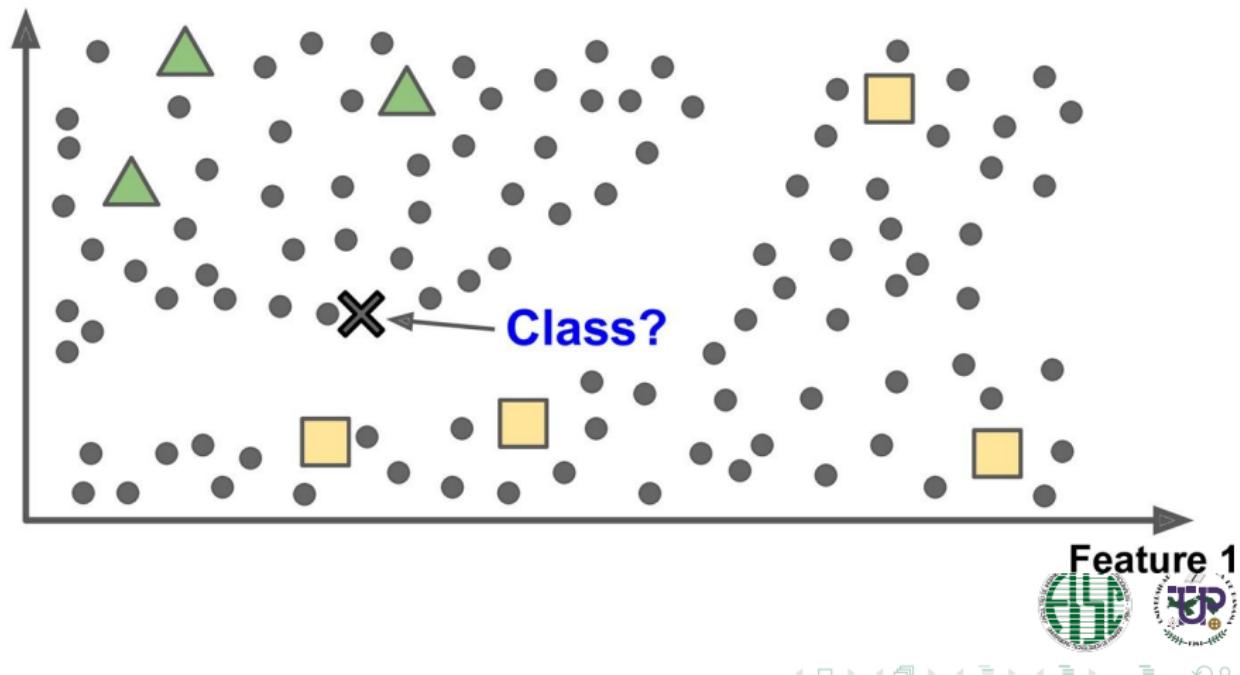
- Estos algoritmos son una combinación de los algoritmos supervisados y no supervisados.
- Son entrenados secuencialmente en una manera no supervisada y el sistema completo es ajustado usando técnicas de aprendizaje supervisado.



Aprendizaje Semi-Supervisado II

Tipos de Aprendizaje

Feature 2

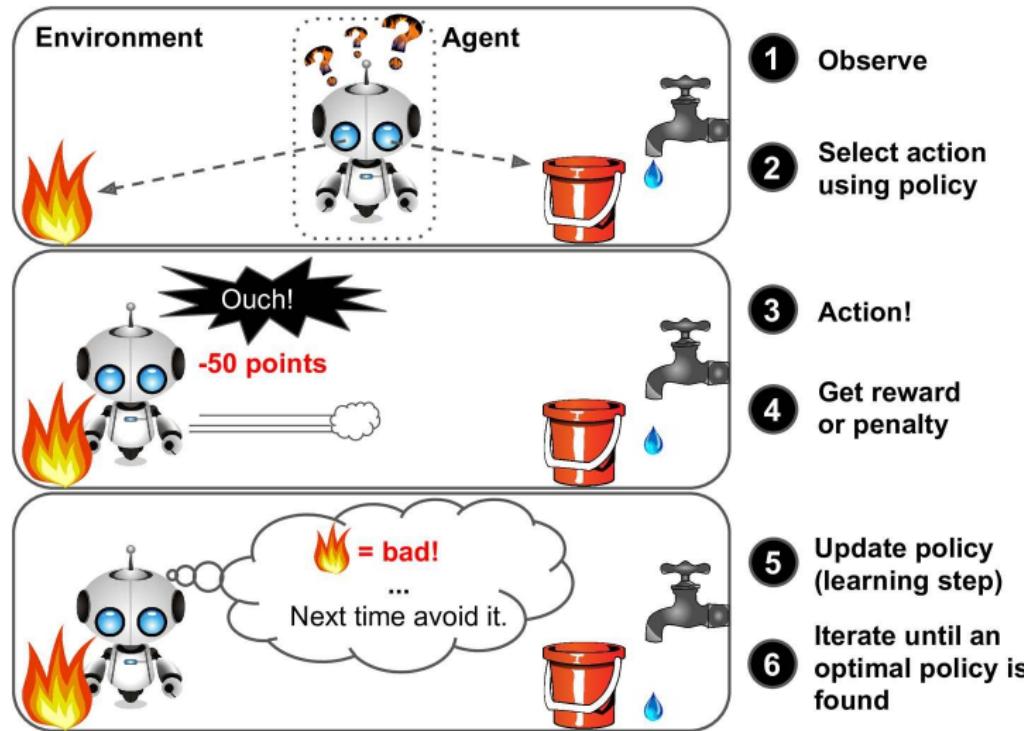


Tipos de Aprendizaje

Section 3.4 Aprendizaje por Refuerzo

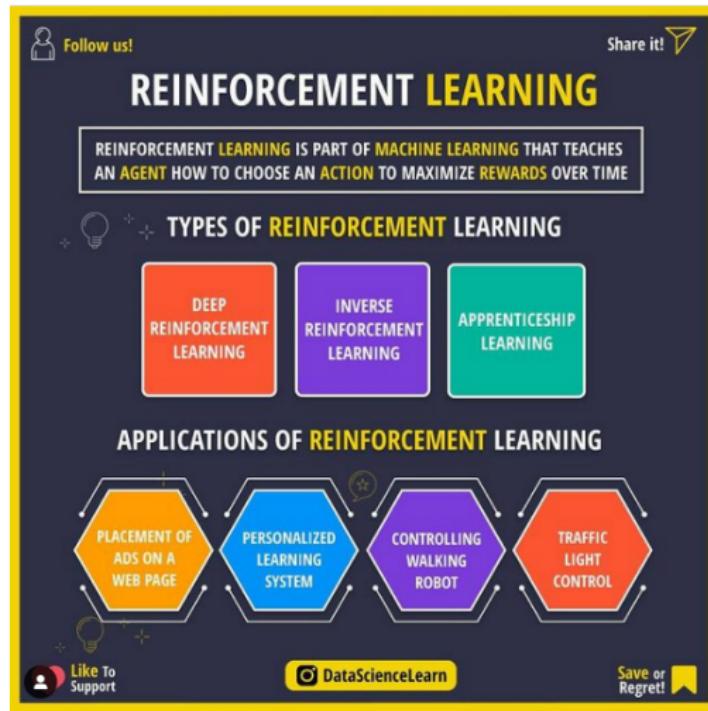
Aprendizaje por Refuerzo

Tipos de Aprendizaje



Aprendizaje por Refuerzo

Resumen



The infographic is titled "REINFORCEMENT LEARNING" in large, bold, yellow capital letters. Below the title, a box contains the text: "REINFORCEMENT LEARNING IS PART OF MACHINE LEARNING THAT TEACHES AN AGENT HOW TO CHOOSE AN ACTION TO MAXIMIZE REWARDS OVER TIME". Underneath this, a section titled "TYPES OF REINFORCEMENT LEARNING" lists three categories: "DEEP REINFORCEMENT LEARNING" (orange), "INVERSE REINFORCEMENT LEARNING" (purple), and "APPRENTICESHIP LEARNING" (teal). Below this, another section titled "APPLICATIONS OF REINFORCEMENT LEARNING" shows four hexagonal icons: "PLACEMENT OF ADS ON A WEB PAGE" (orange), "PERSONALIZED LEARNING SYSTEM" (blue), "CONTROLLING WALKING ROBOT" (purple), and "TRAFFIC LIGHT CONTROL" (red). The bottom of the infographic features social media sharing icons and links to "Like To Support" and "DataScienceLearn".

Follow us!

Share it!

REINFORCEMENT LEARNING

REINFORCEMENT LEARNING IS PART OF MACHINE LEARNING THAT TEACHES AN AGENT HOW TO CHOOSE AN ACTION TO MAXIMIZE REWARDS OVER TIME

TYPES OF REINFORCEMENT LEARNING

- DEEP REINFORCEMENT LEARNING
- INVERSE REINFORCEMENT LEARNING
- APPRENTICESHIP LEARNING

APPLICATIONS OF REINFORCEMENT LEARNING

- PLACEMENT OF ADS ON A WEB PAGE
- PERSONALIZED LEARNING SYSTEM
- CONTROLLING WALKING ROBOT
- TRAFFIC LIGHT CONTROL

Like To Support

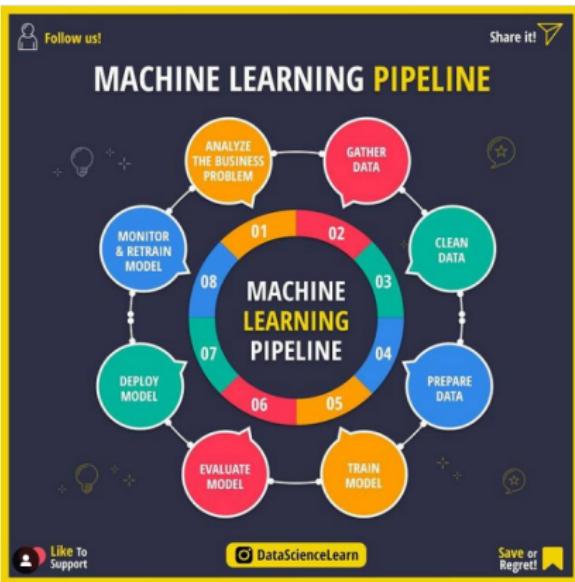
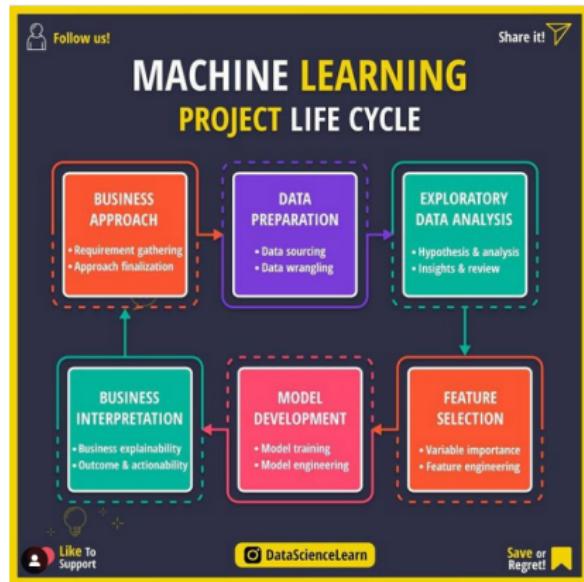
DataScienceLearn

Save or Regret!

Tipos de Aprendizaje



Ciclo de Vida de un Proyecto de ML



Machine Learning



- 4 Desarrollo de Modelos con ML
 - Herramientas de Desarrollo

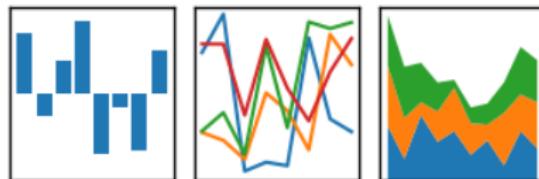
Section 4.1 | Herramientas de Desarrollo

Desarrollo de Modelos con ML I



pandas

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



Desarrollo de Modelos con ML II



Machine Learning

Hackathon EURUS 2021

José Carlos Rangel
jose.rangel@utp.ac.pa

FISC-UTP

13 de noviembre de 2021

