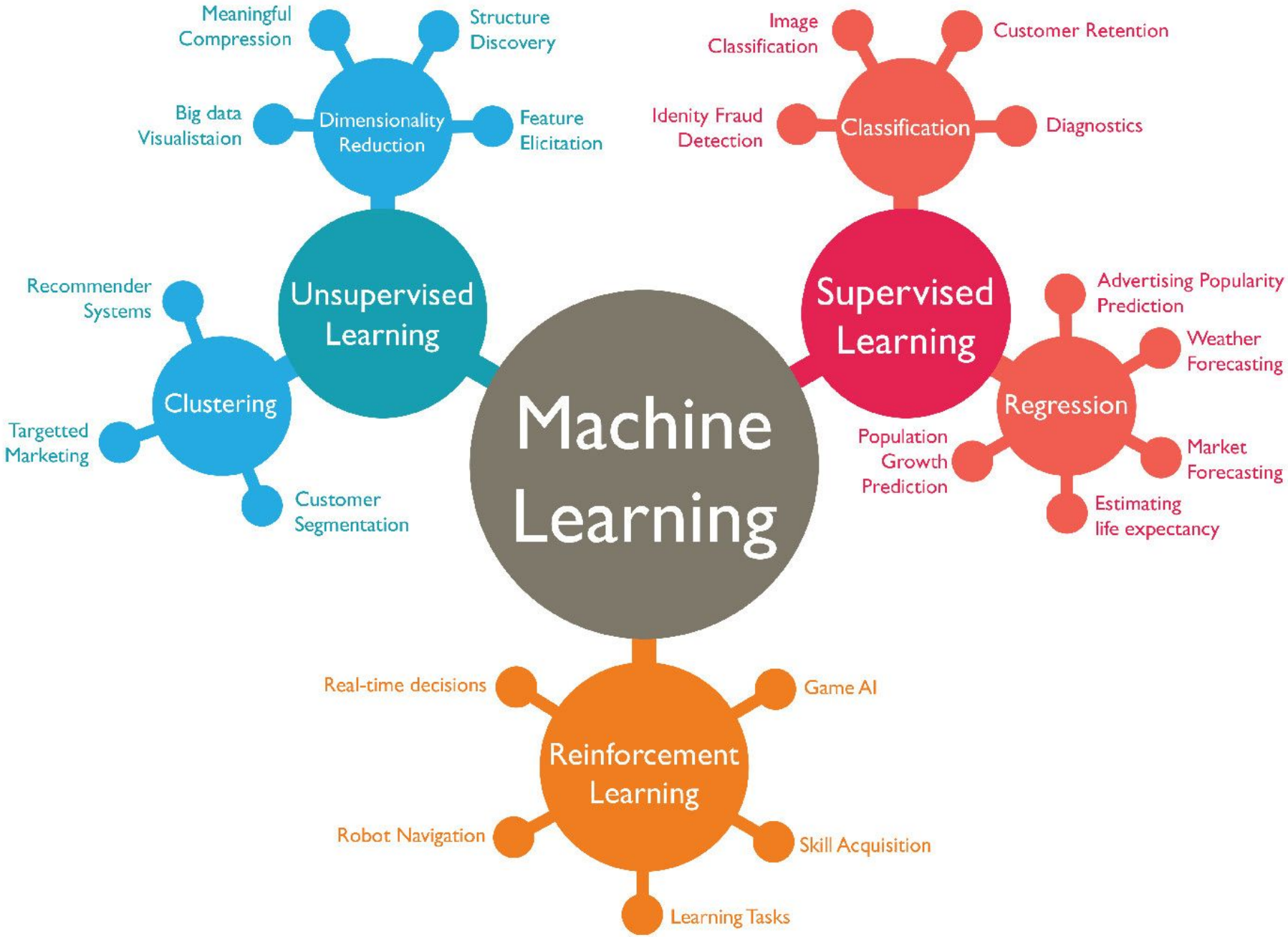
LOYOLA Másteres

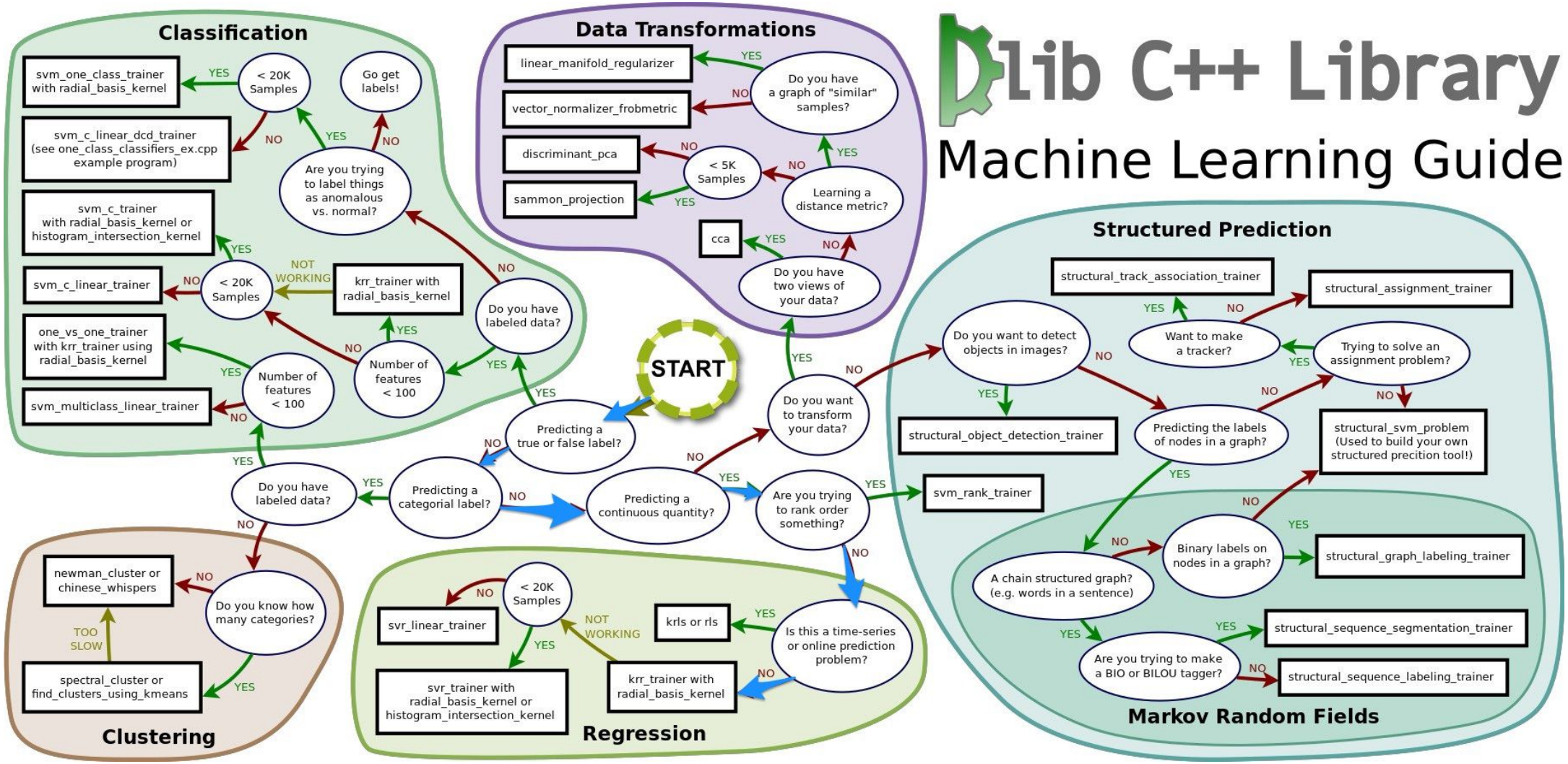
TEMA 3
Modelos de Regresión con Scikit-Learn

2022/2023

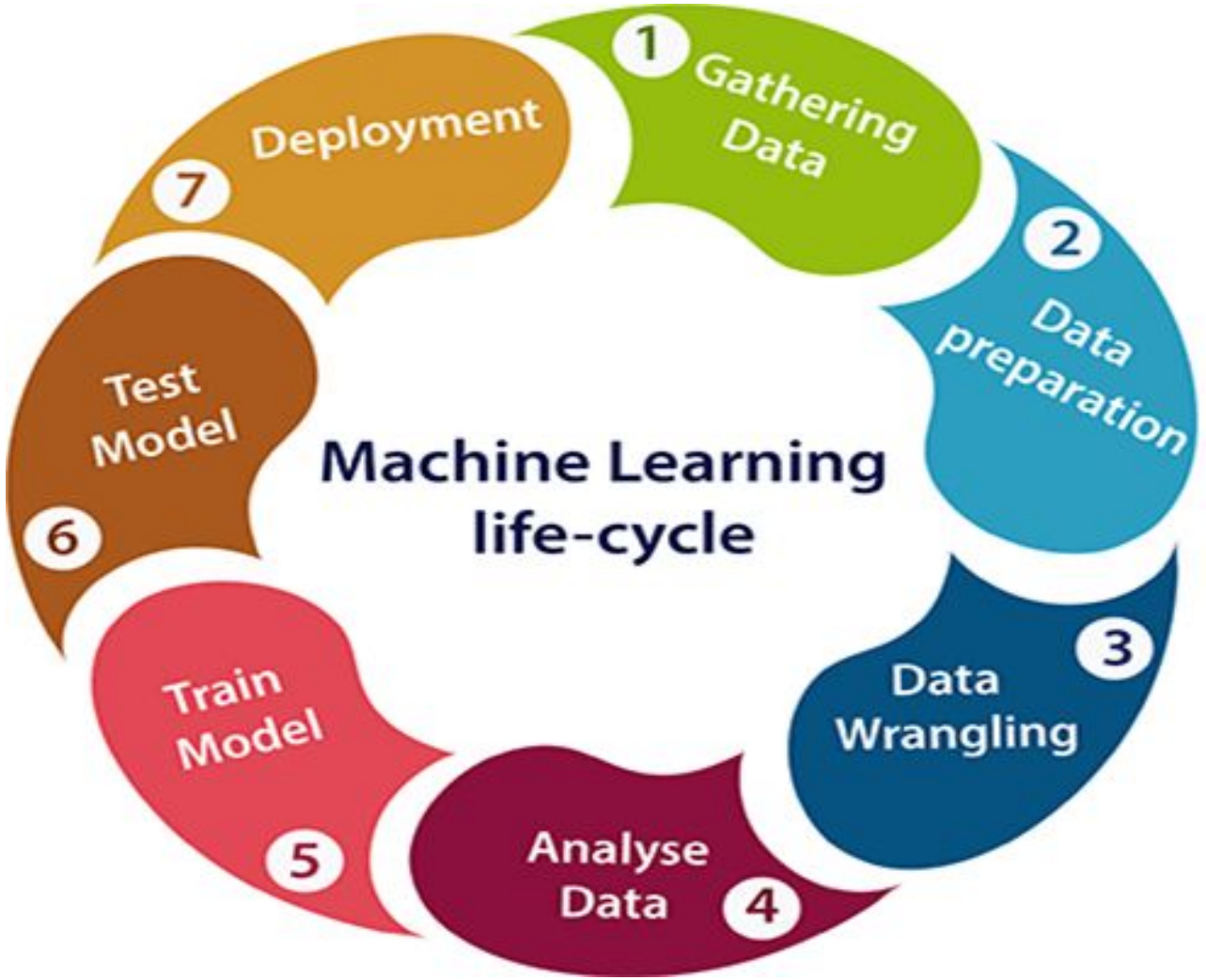
ASIGNATURA Modelos de Regresión


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

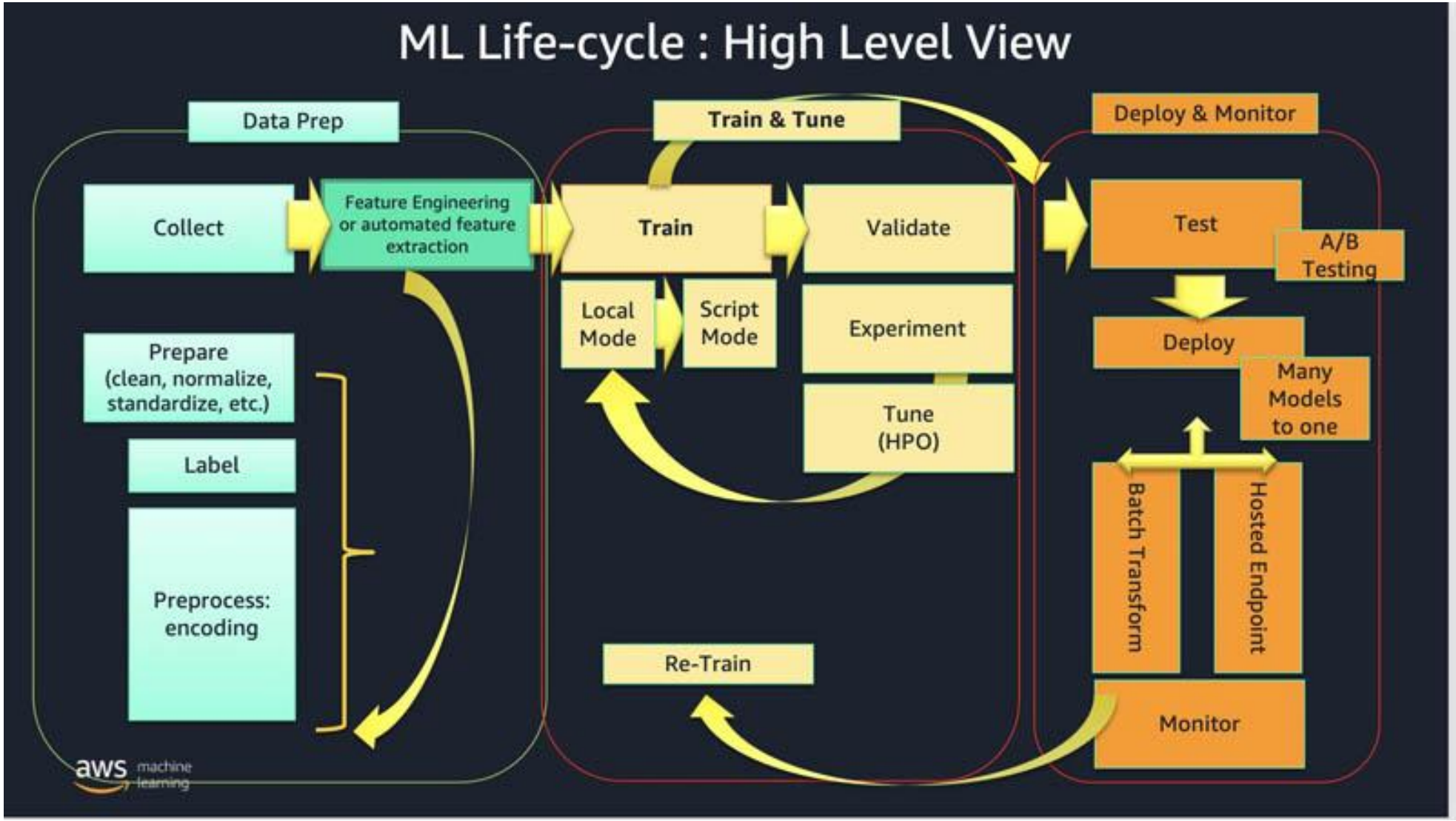
Elección de un Modelo



Ciclo de Vida de un Modelo Predictivo



Implementación en Cloud



HTML5 Summary

Online overview of Scikit Learn

Fundamentos de la Regresión Lineal Múltiple

Each row of matrix X represents a point of the sample data:

$$\mathbf{X_{ij}} = \begin{pmatrix} \mathbf{x_{i1}} & \mathbf{x_{i2}} & \dots & \mathbf{x_{in}} \end{pmatrix}$$

The hypothesis function is multidimensional, i.e. number of columns in X:

$$\mathbf{h_\theta(x_i)} = \theta_0 + \theta_1 \mathbf{x_{i1}} + \theta_2 \mathbf{x_{i2}} + \dots . \theta_j \mathbf{x_{ij}} \dots . . . \theta_n \mathbf{x_{mn}}$$

$\theta_0, \theta_1, \dots . \theta_j \dots . \theta_n$ are parameter of hypothesis, $\mathbf{m}$ Number of training exaples, $\mathbf{n}$ Number of independent variable, $\mathbf{x_{ij}}$ is $\mathbf{i^{th}}$ training exaple of $\mathbf{j^{th}}$ feature.

So we represent hypothesis function in vectorize form

$$\mathbf{y} = \mathbf{h_\theta(x)} = \mathbf{X}\theta$$

Función de costo (planteamiento 1)

$$\mathbf{J}(\theta) = \frac{\mathbf{1}}{\mathbf{m}} \sum_{\mathbf{i=1}}^{\mathbf{m}} (\hat{\mathbf{y}}_\mathbf{i} - \mathbf{y}_\mathbf{i})^\mathbf{2}$$

$$\mathbf{J}(\theta) = \frac{\mathbf{1}}{\mathbf{m}} \sum_{\mathbf{i=1}}^{\mathbf{m}} (\mathbf{h}_\theta(\mathbf{x}_\mathbf{i}) - \mathbf{y}_\mathbf{i})^\mathbf{2}$$

Expresada en forma matricial:

$$\mathbf{J}(\theta) = \frac{\mathbf{1}}{\mathbf{m}} (\mathbf{X}\theta - \mathbf{y})^\mathbf{T} (\mathbf{X}\theta - \mathbf{y})$$

Búsqueda del mínimo

Derivada parcial respecto a cada variable independiente (*feature*):

$$\mathbf{min}_{\theta_0,\theta_1..\theta_n}\mathbf{J}(\theta_0,\theta_1..\theta_n)$$

$$\frac{\partial \mathbf{J}(\theta_j)}{\partial \theta_j} = \mathbf{0}$$

Desarrollando la derivada:

$$\frac{\partial \mathbf{J}(\theta)}{\partial \theta} = \mathbf{X^T X} 2\theta - \mathbf{2X^T y} + \mathbf{0}$$

$$\mathbf{0} = \mathbf{2X^T X}\theta - \mathbf{2X^T y}$$

$$\mathbf{X^T X}\theta = \mathbf{X^T y}$$

Solución matricial

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

pero…

Invertir matrices es computacionalmente muy costoso,
e inviable para un número grande de *features*

Por ello, recurrimos a un algoritmo numérico de convergencia con el número de iteraciones, *Gradient Descent* (ver planteamiento 2 a continuación)

Función de costo (planteamiento 2)

$$\hat{L}(f_w) = \frac{1}{m}\sum_{i=1}^{m}(w^T x^{(i)} - y^{(i)})^2 = \frac{1}{m}||Xw - y||_2^2$$

Y su gradiente (vector de derivadas parciales respecto a cada coeficiente):

$$\begin{aligned} & \nabla_w \hat{L}(f_w) \\ = \quad & \nabla_w \tfrac{1}{m}||Xw - y||_2^2 \\ = \quad & \nabla_w [(Xw - y)^T (Xw - y)] \\ = \quad & \nabla_w [w^T X^T X w - 2w^T X^T y + y^T y] \\ = \quad & 2X^T X w - 2X^T y \end{aligned}$$

Igualado a 0 nos permitirá determinar el mínimo respecto a los coeficientes de la regresión lineal **w**

Algoritmo de *Gradient Descent*

- Linear regression: $\hat{L}(f_w) = \frac{1}{m}\sum_{i=1}^{m}(w^T x^{(i)} - y^{(i)})^2 = \frac{1}{m}||Xw - y||_2^2$
- The gradient is $\nabla_w \hat{L}(f_w) = 2X^T Xw - 2X^T y$
- Gradient Descent algorithm is
  - Set step size ϵ, tolerance δ to small, positive numbers.
  - *While* $||X^T Xw - X^T y||_2 > \delta$ do

$$w \longleftarrow w - \epsilon(X^T Xw - X^T y)$$

Evaluation metrics & Model Selection in Linear Regression

Evaluación del modelo

Mean Squared Error (MSE)

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

MSE penalizes large errors

Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

MAE DOES NOT penalize large errors

Evaluación: error de la regresión

When we know the values of the independent variables, we can calculate the regression error.

We know that residual is the difference between actual and predicted value. Thus, RSS (Residual sum of squares) can be calculated as follows.

$$SE_{Line} = RSS = (y_1 - (\theta_0 + \theta_1 x_1))^2 + (y_2 - (\theta_0 + \theta_1 x_2))^2 + \ldots + (y_n - (\theta_0 + \theta_1 x_n))^2$$

Evaluación: varianza de *y* (valores reales)

The total variation in Y can be given as a sum of squared differences of the distance between every point and the arithmetic mean of Y values. This can be termed as **TSS** (Total sum of squares).

$$SE_{\bar{y}} = TSS = (y_1 - \bar{y})^2 + (y_2 - \bar{y})^2 + \ldots + (y_n - \bar{y})^2$$

Evaluación: coeficiente de determinación

$\frac{RSS}{TSS}$ ---> represents the part of the variance of y, not described by the line

Coeficiente de determinación:

$R^2 = 1 - \frac{RSS}{TSS}$ ---> represents the part of the variance of y, described by the independent variables

Evaluación: medida del error, RMSE

Basically, RMSE is just the root of the average of squared residuals. We know that residuals are a measure of how distant the points are from the regression line. Thus, RMSE measures the scatter of these residuals.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Ejemplos de código y ejercicios: Github

https://github.com/loyola-masters/regression-models

Ejemplo: Boston Houses Prices

https://github.com/loyola-masters/regression-models/tree/main/BostonHousePrices

El **dataset** contiene las siguientes columnas:

- **crim**: tasa de criminalidad per cápita por ciudad
- **zn**: proporción de terrenos residenciales para lotes de más de 25.000 pies cuadrados
- **indu**: proporción de acres comerciales no minoristas por ciudad
- **Chas**: variable ficticia del río Charles (= 1 si la zona linda con el río; 0 en caso contrario)
- **Nox**: concentración de óxidos de nitrógeno (partes por 10 millones)
- **rm**: número medio de habitaciones por vivienda
- **Edad**: proporción de unidades ocupadas por sus propietarios construidas antes de 1940
- **dis**: media ponderada de las distancias a cinco centros de trabajo de Boston
- **Rad**: índice de accesibilidad a las autopistas radiales
- **Impuesto**: tasa del impuesto sobre la propiedad por cada 10.000 dólares
- **ptratio**: proporción de alumnos por profesor por ciudad
- **black**: 1000(Bk - 0,63)^2 donde Bk es la proporción de negros por ciudad
- **lstat**: estatus inferior de la población (porcentaje)
- **medv**: valor medio de las viviendas ocupadas por sus propietarios en 1000$