UNIVERSIDAD DE LOS LAGOS

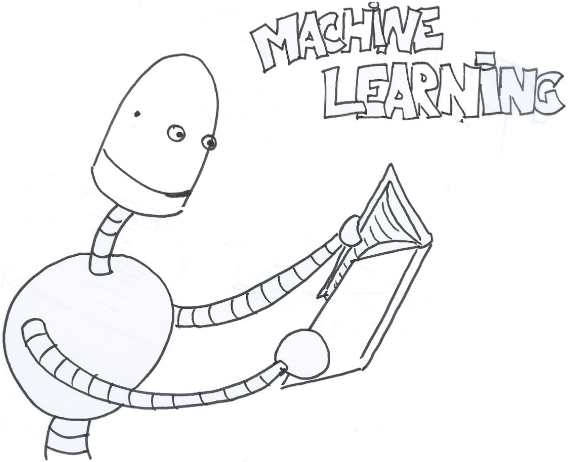
Electivo III: Machine Learning

Clasificación

Joel S. Torres

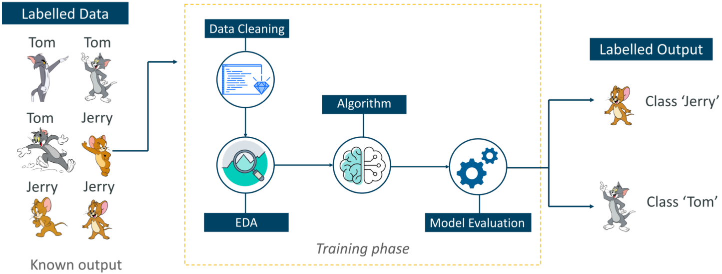
Departamento de Ciencias de la Ingeniería
Ingeniería Civil Informática

MACHINE LEARNING



Clasificación

¿Qué es la Clasificación?

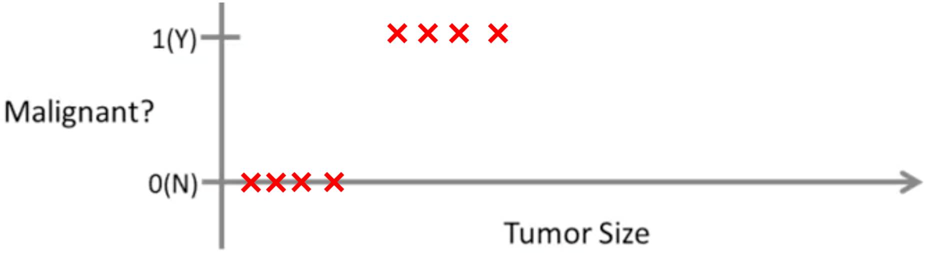


$$y \in \{0, 1\}$$

- 0 Clase Negativa (Tom)
- 1 Clase Positiva (Jerry)

Clasificación

Predicción de tipo de Cancer (M,B)



¿Cómo representar el problema?

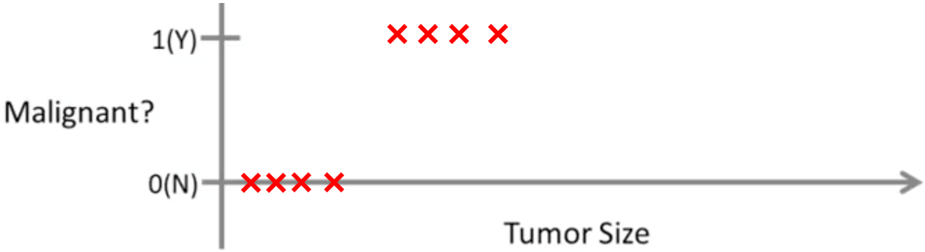
- 0 Tumor Benigno
- 1 Tumor Maligno

¿Qué método utilizar?

Intentemos una regresión lineal...

Clasificación

Predicción de tipo de Cancer (M,B)

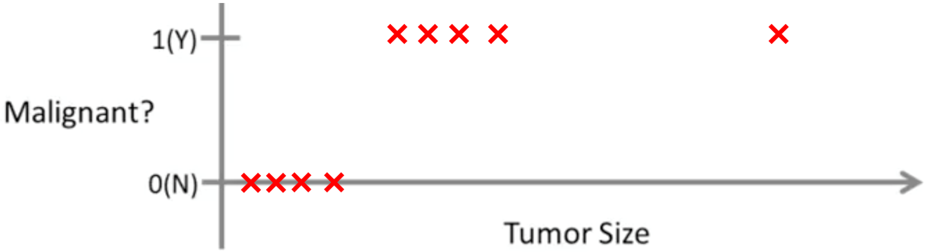


Una pista: Utilizar un limitador (threshold)
Entonces para un threshold $T = 0,5$:

- $h_\theta(x) \geq T$, predice salida $y = 1$
- $h_\theta(x) < T$, predice salida $y = 0$

Clasificación

Predicción de tipo de Cancer (M,B)



Una pista: Utilizar un limitador (threshold)
Entonces para un limitador $L = 0,5$:

- $h_\theta(x) \geq L$, predice salida $y = 1$
- $h_\theta(x) < L$, predice salida $y = 0$

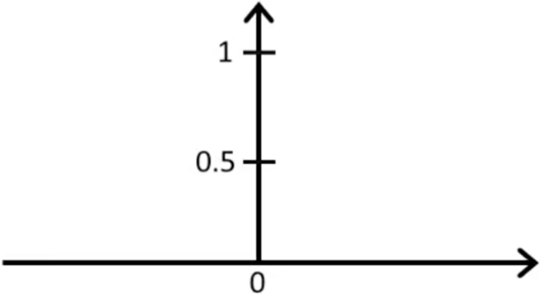
Logistic Regression

Objetivo del modelo
Se necesita restringir el modelo a $0 \leq h\theta(x) \leq 1$
Nuevo Modelo

$$h_\theta(x) = g(\theta^T x)$$

¿Qué es $g()$?

$$g(z) = \frac{1}{1 + e^{-z}}$$



Esta función se llama sigmoidea (sigmoid function) o función logística.

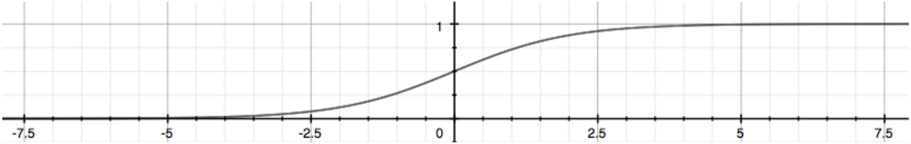
Logistic Regression

Entonces, con $z = \theta^T x$:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Objetivo del modelo

Ajustar los parámetros θ con el menor costo posible.



Interpretación del resultado de la hipótesis

El modelo se basa en probabilidad

$h_\theta(x) =$ Estima la probabilidad de que $y = 1$ para la entrada x

Ejemplo

Si $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ MedidaTumor \end{bmatrix}$; y, $h_\theta(x) = 0,7$

Significa que hay una chance del 70 % de que el tumor sea **maligno**

Entonces,

$$h_\theta(x) = P(y = 1|x; \theta)$$

"la probabilidad de que $y = 1$, dado un x parametrizado por θ"

Algunas Propiedades:

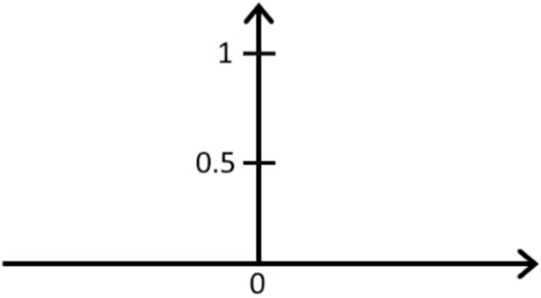
- $P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$
- $P(y = 0|x; \theta) = 1 - P(y = 1|x; \theta)$

Límites de Decisión

El impacto de los límites

Para predecir $y = 1$ debemos definir un límitador $h_\theta(x) \geq 0,5$

Por lo tanto, $y = 0$ debe tener un límitador $h_\theta(x) < 0,5$



Si, $g(z) \geq 0,5$ cuando $z \geq 0$

Entonces,

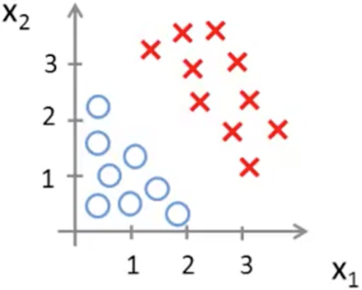
$$h_\theta(x) = g(\theta^T x) \geq 0,5$$

cuando,

$$\theta^T x \geq 0$$

Límites de Decisión

Ejemplo



Asimilando lo anterior,

Si, $h_\theta(x) = g(\theta_0 + \theta_1 * x_1 + \theta_2 * x_2)$

Considerando un vector $\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

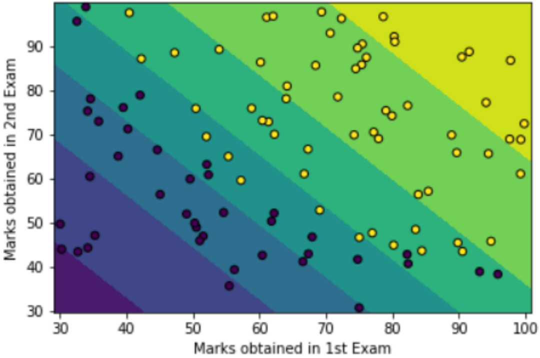
Para predecir $y = 1$ tendremos que comprobar que $-3 + x_1 + x_2 \geq 0$

Lo que implica que, $x_1 + x_2 \geq 3$

Límites de Decisión

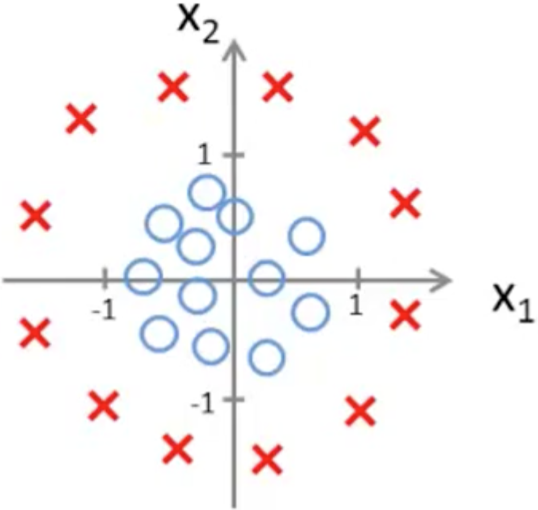
Definición

Un límite de desición en una línea que representa la separación entre la probabilidad de pertenecer a alguna de las categorías propuestas

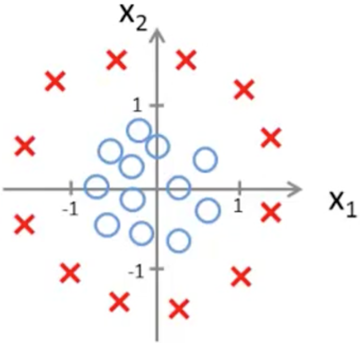


Límites de Decisión No Lineales

¿Cómo solucionar este problema?



Límites de Decisión No Lineales



Nueva hipótesis,

$h_\theta(x) = g(\theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_1^2 + \theta_4 * x_2^2)$

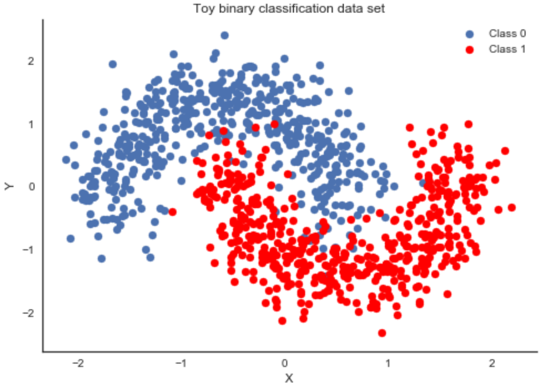
Considerando un vector $\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

Para predecir $y = 1$ tendremos que comprobar que $-1 + x_1^2 + x_2^2 \geq 0$

Lo que implica que, $x_1^2 + x_2^2 \geq 1$

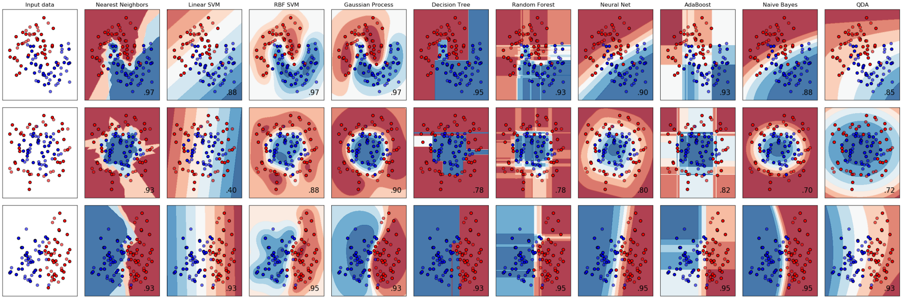
Límites de Decisión No Lineales

¿Cómo implementar otras figuras?



Hipótesis General

$$h_\theta(x) = g(\theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_1^2 + \theta_4 * x_1^2 x2 + \theta_5 * x_1^2 x_2^2 + \theta_6 * x_1^3 x_2 + \ldots)$$



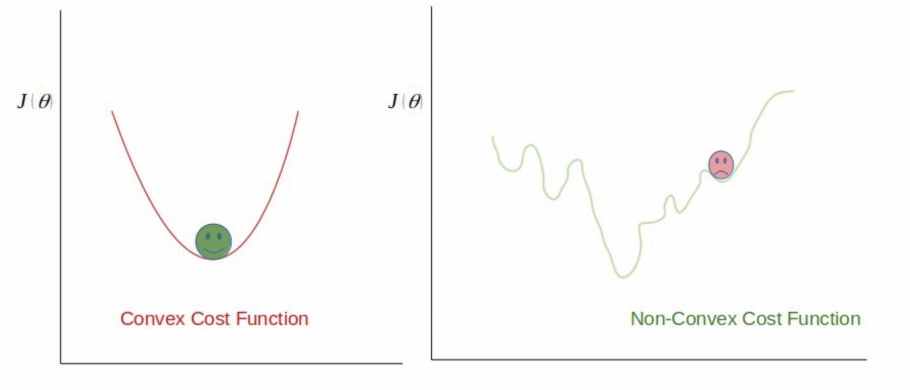
Taller de Clasificación

Próxima Clase:

- Revisaremos la Función de costo y sus consecuencias
- Implementación en código
- Google Colab : https://colab.research.google.com/
- R Studio Cloud : https://rstudio.cloud/

Función de Costo

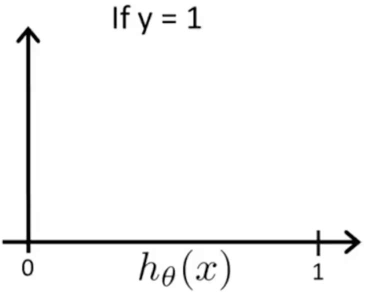
Probando con Función de costo anterior:
$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$, con $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ (Sigmoid)

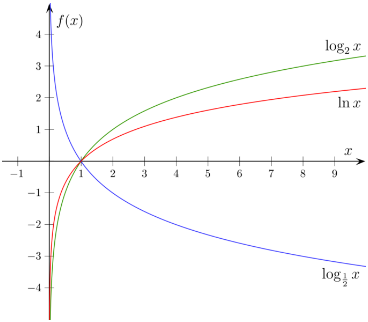


Función de Costo

Nueva Función de costo: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(Cost(h_\theta(x), y))$

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{, si } y = 1 \\ -log(1 - h_\theta(x)) & \text{, si } y = 0 \end{cases}$$

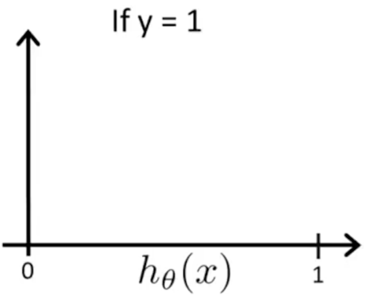




Función de Costo

Nueva Función de costo: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(Cost(h_\theta(x), y))$

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{, si } y = 1 \\ -log(1 - h_\theta(x)) & \text{, si } y = 0 \end{cases}$$



Funcionamiento del Costo

- Si $h_\theta(x) = 1$, entonces $Cost(h_\theta(x), y) = 0$
- Si $h_\theta(x) = 0$, entonces $Cost(h_\theta(x), y) = \infty$

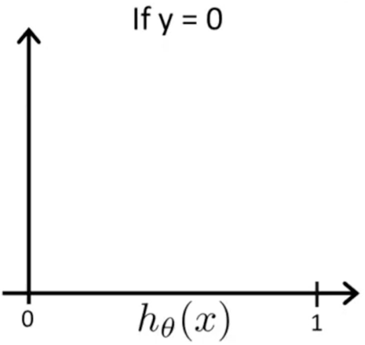
Intuición

Si $h_\theta(x) = 0$ predice que $P(y = 1|x;\theta) = 0$, cuando debería ser $y = 1$, entonces se PENALIZA con un alto costo.

Función de Costo

Nueva Función de costo: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(Cost(h_\theta(x), y))$

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{, si } y = 1 \\ -log(1 - h_\theta(x)) & \text{, si } y = 0 \end{cases}$$



Funcionamiento del Costo

- Si $h_\theta(x) = 0$, entonces $Cost(h_\theta(x), y) = 0$
- Si $h_\theta(x) = 1$, entonces $Cost(h_\theta(x), y) = \infty$

Intuición

Si $h_\theta(x) = 1$ predice que $P(y = 0|x;\theta) = 1$, cuando debería ser $y = 0$, entonces se PENALIZA con un alto costo.

Función de Costo

Nueva Función de costo considerando todos los atributos entrada:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(Cost(h_\theta(x^{(i)}), y^{(i)}))$$

Se puede reescribir como:

$$Cost(h_\theta(x^{(i)}), y^{(i)}) = -y^{(i)}log(h_\theta(x^{(i)})) - (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))$$

Entonces la Función de Costo para Regresión Lineal es:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}log(h_\theta(x^{(i)}) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)})))$$

El objetivo es Minimizar el costo:

$$\min_\theta J(\theta)$$

Algoritmo Descenso del Gradiente

Algorithm 1: Descenso de Gradiente

Result: Encuentra $\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$

repeat
    for $j \leftarrow 0$ **to** m **do**
        $\theta_j := \theta_j - \alpha \frac{d}{d\theta_j} J(\theta)$;
    end
until *Converger*;

Algoritmo Descenso del Gradiente

Algorithm 2: Descenso de Gradiente

Result: Encuentra $\min\limits_{\theta_0,\theta_1} J(\theta_0,\theta_1)$

repeat

  for $j \leftarrow 0$ **to** n **do**

    $\theta_j := \theta_j - \alpha \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) * x_j^{(i)}$;

  end

until *Converger*;

Otros Algoritmos de Aprendizaje

- Gradiente Conjugado
- BFGS
- L-BFGS

Ventajas:

- No es necesario fijar el α.
- Más rápidos que el Descenso de Gradiente.

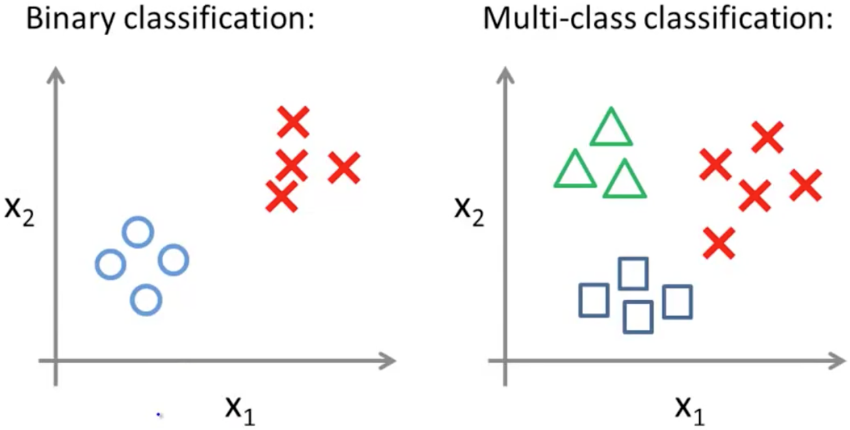
Desventaja:

- Mucho más complicados de implementar.

Clasificación Multiclase

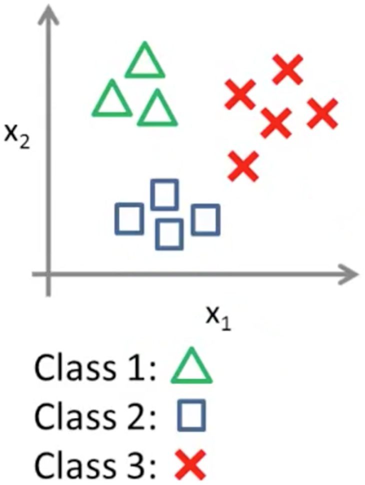
¿Y si ahora queremos identificar más clases?

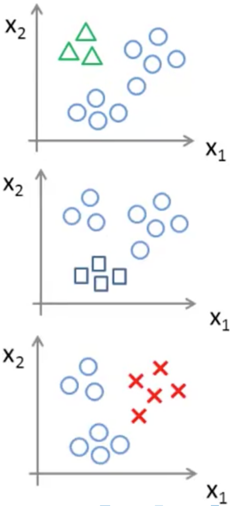
Clasificación Multiclase



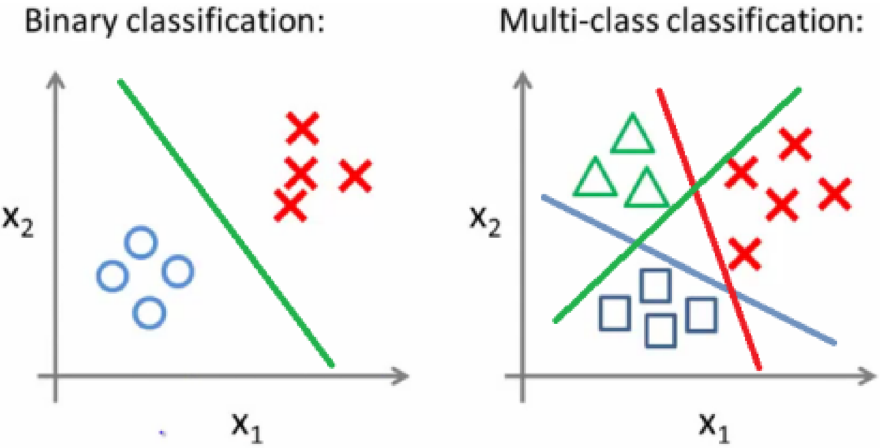
Clasificación Multiclase

Clasificación One vs All con Regresión Logística





Clasificación One vs All con Regresión Logística



Taller de Clasificación

- Google Colab : https://colab.research.google.com/
- R Studio Cloud : https://rstudio.cloud/