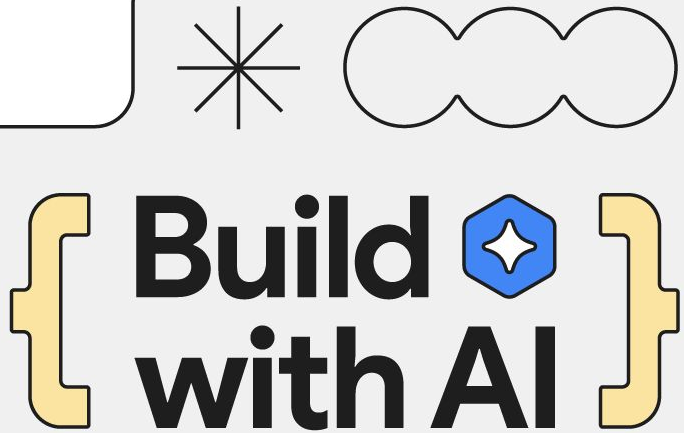
Google Developer Group
A Coruña

Inteligencia Artificial en la Detección del Cáncer

Aprende a entrenar modelos de Machine Learning y predecir el cáncer de mama



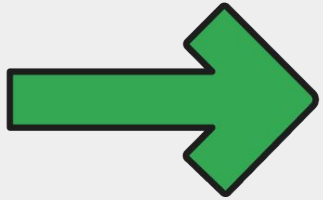
Soy **Cris Correa**

Estudiante de Computer Systems Engineering en la Universidad de Sunderland

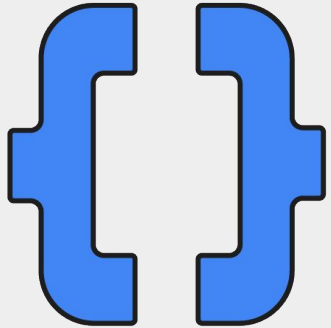
Capítulo 1

¿Qué son los modelos de **ML**?

¿Cómo se entrenan y se testean?

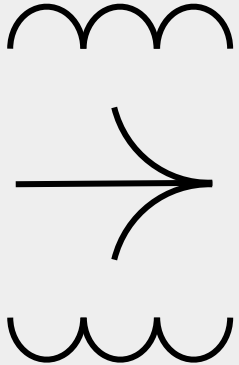


Modelos de ML



Los modelos de Machine Learning son programas informáticos que aprenden a hacer predicciones o tomar decisiones a partir de datos, sin necesidad de ser programados explícitamente para cada tarea.

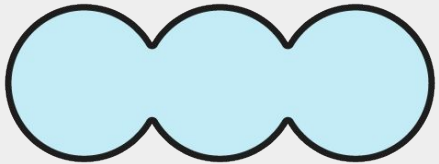
¿Cómo describirías lo que es un girasol?

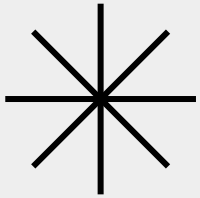


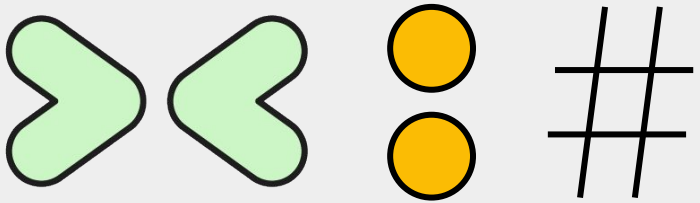
¿Qué pasa con los datos?

Tipos de datos

En la mayoría de los casos, los datos utilizados para entrenar un modelo de Machine Learning se pueden clasificar en tres tipos principales: **tabulares**, **imágenes** y **multimodales**.







¡Pero hay más!

Cualquier cosa puede ser un dato: un audio, un vídeo, tu chat de WhatsApp... todo es información con la que se puede entrenar un modelo.

Nuestra base de datos

```
1000025,5,1,1,1,2,1,3,1,1,2
1002945,5,4,4,5,7,10,3,2,1,2
1015425,3,1,1,1,2,2,3,1,1,2
1016277,6,8,8,1,3,4,3,7,1,2
1017023,4,1,1,3,2,1,3,1,1,2
1017122,8,10,10,8,7,10,9,7,1,4
1018099,1,1,1,1,2,10,3,1,1,2
1018561,2,1,2,1,2,1,3,1,1,2
1033078,2,1,1,1,2,1,1,1,5,2
```

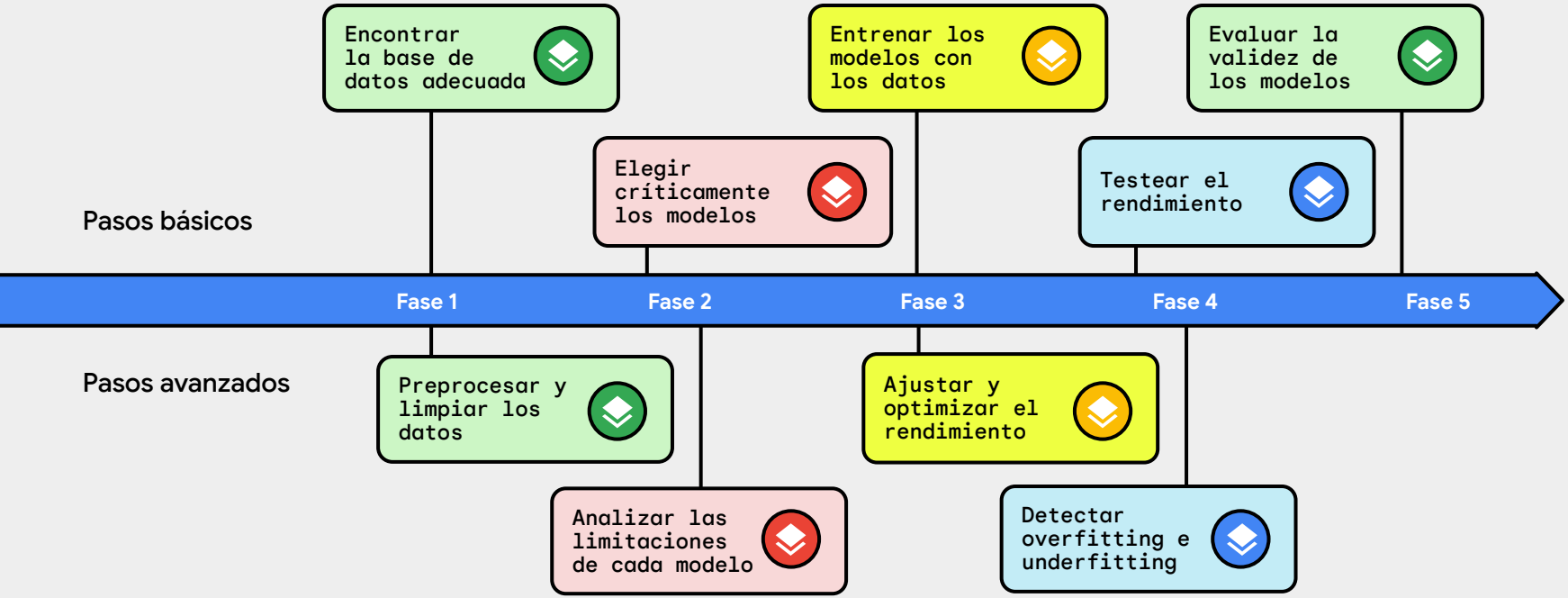
1. Tabular
2. UC Irvine ML Repository
3. Breast Cancer Wisconsin (Original)
4. Necesita preprocesamiento
5. Datos reales

Nuestra base de datos

ID, grosor, tamaño de la célula, forma de la célula, adhesion, tamaño epitelial, nucleos desnudos, cromatina blanda, nucleolos normales, mitosis y clase

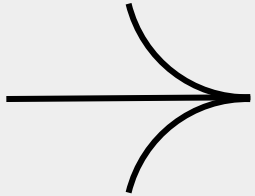
ID	grosor	tamaño_celula	forma_celula	adhesion	tamaño_epitelial	nucleos_desnudos	cromatina_blanda	nucleolos_normales	mitosis	clase
1000025	5	1	1	1	2	0	1	3	1	2
1002945	5	4	4	5	7	1	10	3	2	2
1015425	3	1	1	1	2	2	2	3	1	2
1016277	6	8	8	1	3	3	4	3	7	2
1017023	4	1	1	3	2	4	1	3	1	2
1017122	8	10	10	8	7	5	10	9	7	2
1018099	1	1	1	1	2	6	10	3	1	2
1018561	2	1	2	1	2	7	1	3	1	2
1033078	2	1	1	1	2	8	1	1	1	2
1033078	4	2	1	1	2	9	1	2	1	2

Línea del tiempo del ML



Data Splitting o División del conjunto de datos

Es el proceso de separar los datos en diferentes subconjuntos para entrenar y evaluar un modelo de Machine Learning



67%

Datos para entrenamiento

33%

Datos para testing

Nuestros 3 Modelos

General Linear Model regression

Es un modelo estadístico que relaciona una variable “x” dependiente (respuesta) con una o más variables “y” independientes (predictoras) mediante una combinación lineal.

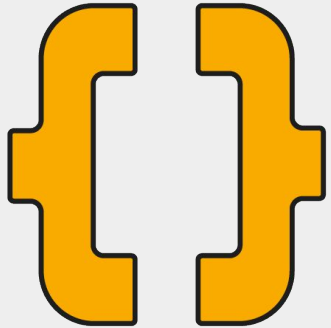
Support Vector Machines

Es un modelo de clasificación que encuentra la mejor línea (o hiperplano en varias dimensiones) que separa los datos en diferentes categorías.

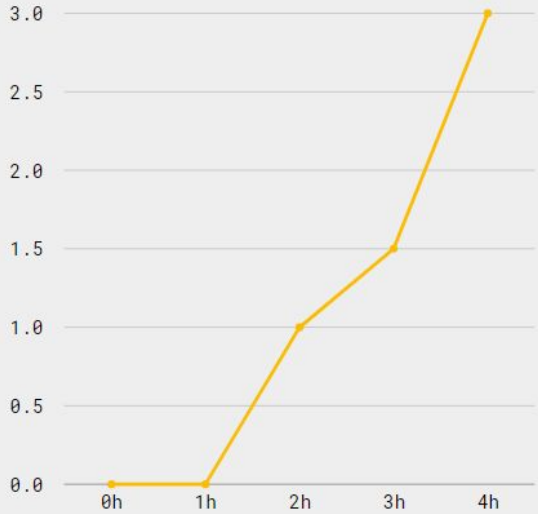
Artificial Neural Networks

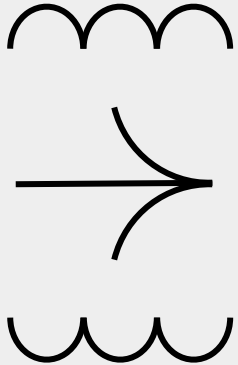
Son modelos inspirados en el cerebro humano, formados por "neuronas" conectadas en capas que procesan la información detectando patrones en los datos

General Linear Model regression (GLM)



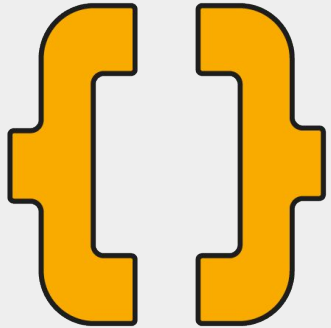
Imagina que quieres saber cuánto influye el número de horas de estudio en la calificación de un examen. Si trazamos un gráfico con estos datos y vemos que las calificaciones aumentan a medida que se estudia más, podemos dibujar una **línea recta que represente esta relación**. Esa línea es lo que el GLM trata de encontrar



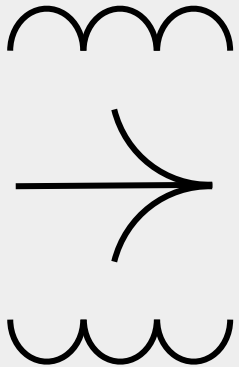


Usaremos LASSO

Least Absolute Shrinkage and Selection Operator

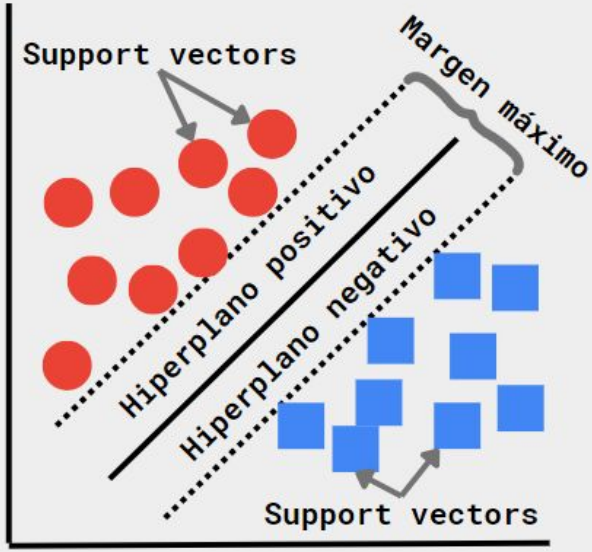


Imagina que quieres predecir la calificación de un examen, pero esta vez usando **varios factores** que podrían influir en ella: horas de estudio, horas de sueño, número de clases asistidas, si desayunaste o no... Tenemos muchos factores, pero **unos son más importantes que otros.**



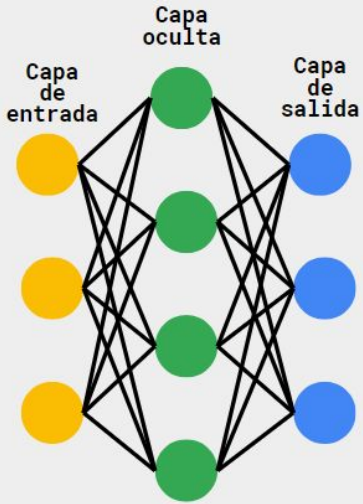
Aquí es donde elegimos usar LASSO porque nos ayuda a encontrar los factores que realmente importan (horas de estudio o número de clases asistidas), elimina o reduce el peso de los factores que no influyen en nuestra nota (si desayunaste o no) y hará nuestro modelo más simple.

Support Vector Machines (SVM)



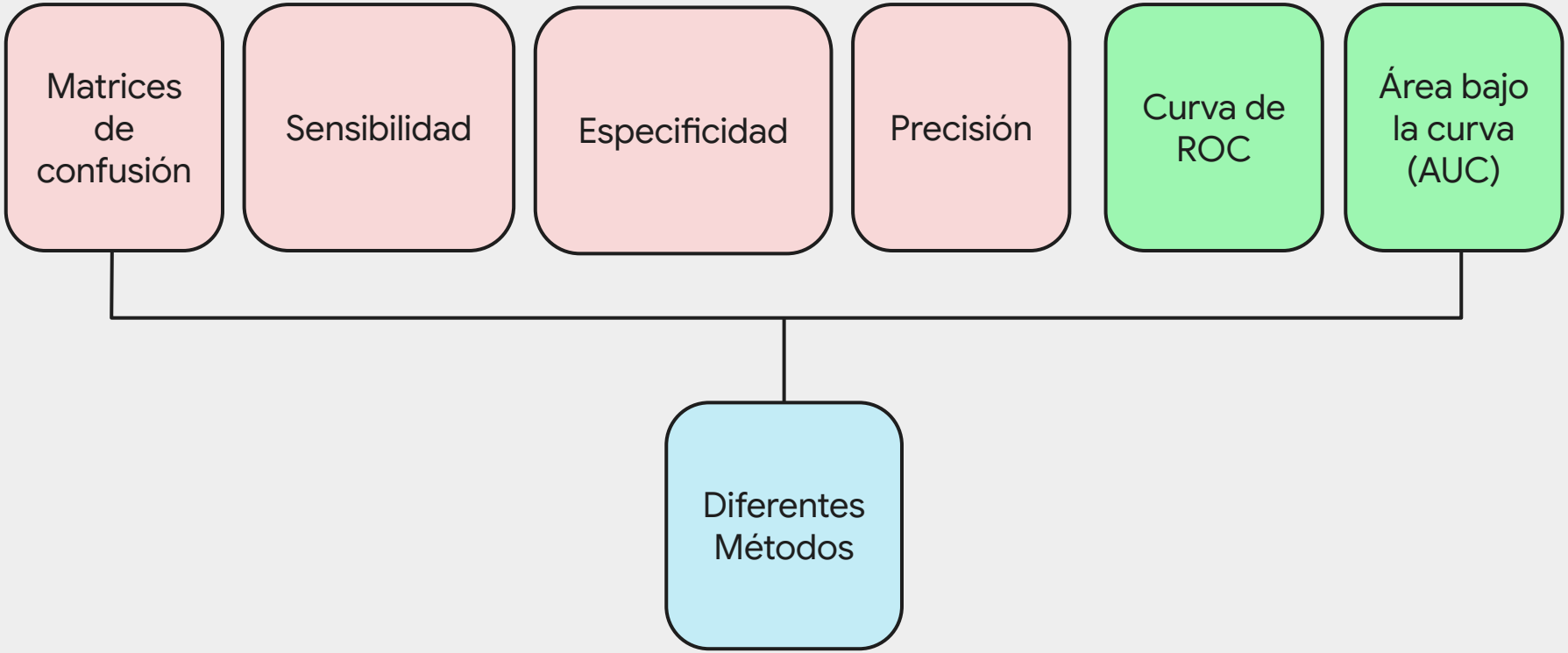
Imagina que tenemos dos grupos de puntos (support vectors) en un gráfico y queremos separarlos con una línea. SVM busca la **mejor línea posible para dividirlos**, asegurándose de que la separación sea lo más clara posible, es decir, que el **margen** máximo sea lo más grande posible.

Artificial Neural Networks (ANN)

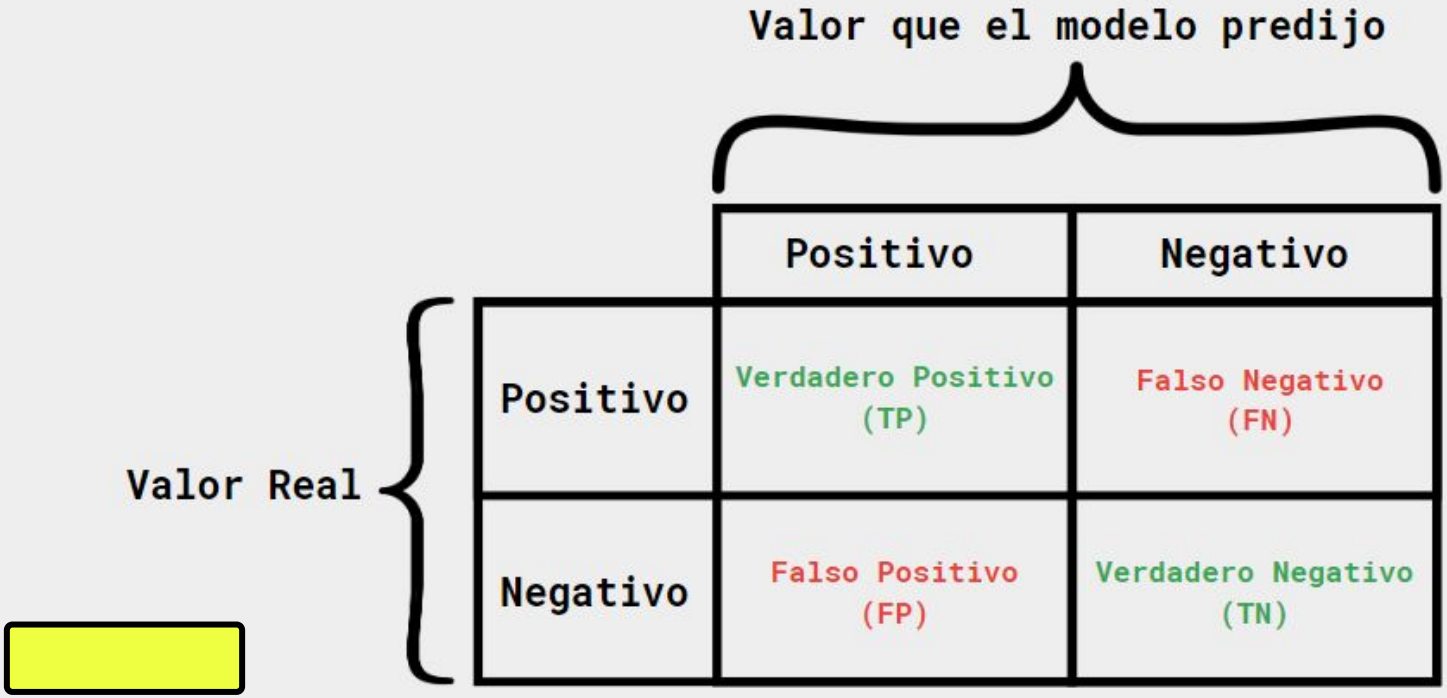


Imagina que tu cerebro aprende a reconocer girasoles viendo muchas fotos, una Red Neuronal Artificial hace algo similar. Utiliza pequeñas unidades matemáticas (**neuronas**) encargadas de procesar la información. Estas están **organizadas en capas** que detectan diferentes características, como colores o formas. A medida que el modelo recibe más datos, **ajusta las conexiones** entre estas neuronas hasta lograr respuestas correctas.

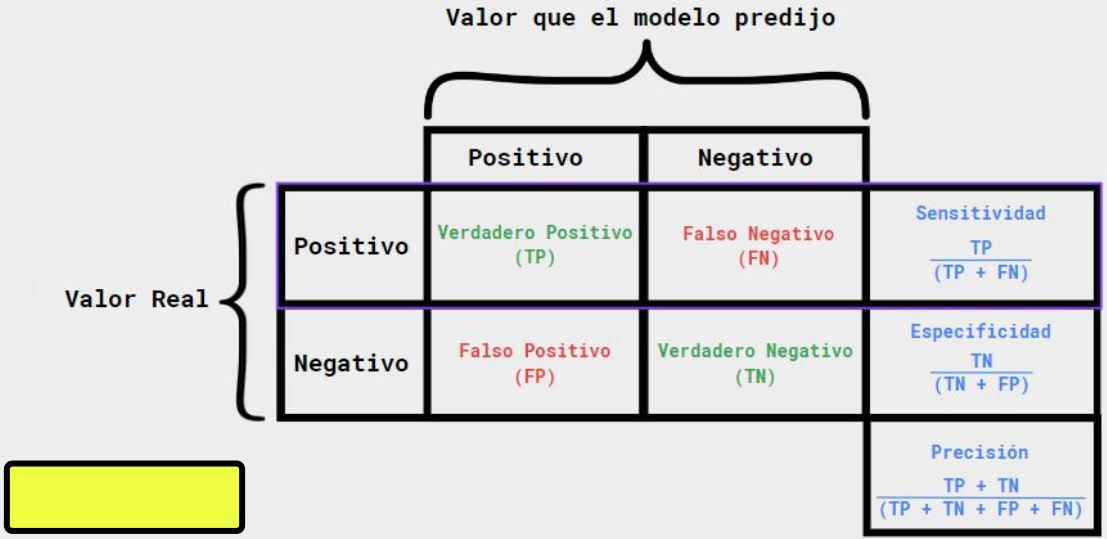
Evaluación de Modelos



Matriz de confusión



Sensibilidad, Especificidad y Precisión

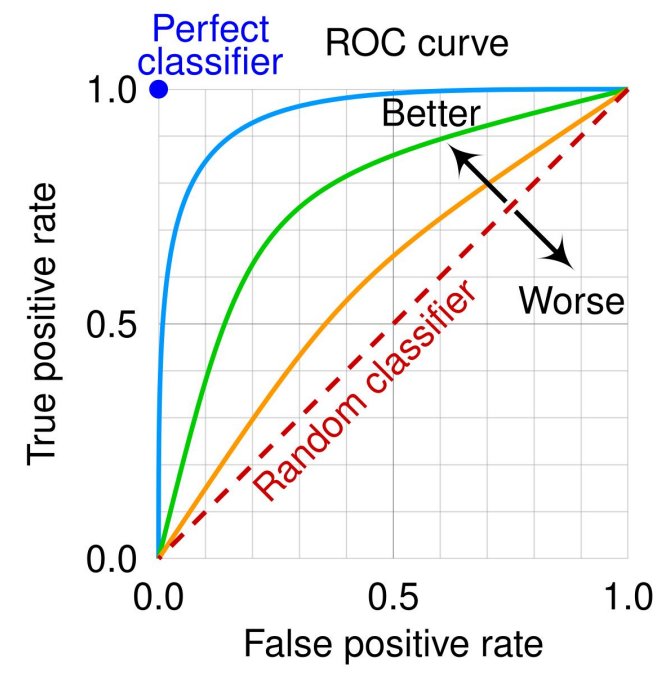


$$\text{Sensitividad} = \frac{TP}{(TP + FN)}$$

$$\text{Especificidad} = \frac{TN}{(TN + FP)}$$

$$\text{Precisión} = \frac{TP + TN}{(TP + TN + FP + FN)}$$

Curva ROC y AUC



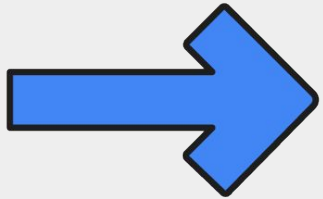
La **curva ROC** (Receiver Operating Characteristic) es una gráfica que evalúa el **rendimiento de un modelo** de clasificación al representar la sensibilidad (tasa de verdaderos positivos) frente a la tasa de falsos positivos (1 - especificidad) en diferentes umbrales.

El **AUC** (Área Bajo la Curva) mide el área bajo la curva ROC y cuantifica la **capacidad del modelo** para distinguir entre clases.

Capítulo 2

Detección de cancer con **R**

Desarrollaremos código con R y RStudio



Paso 1: Importación y Preprocesamiento

1. Entender y analizar la base de datos
2. Importar los datos y asignar nombres a las columnas
3. Preprocesamiento de datos
4. Data Splitting o División del conjunto de datos
5. Preparar matrices de datos

Nuestra base de datos

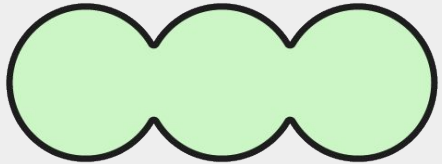
ID, grosor, tamaño de la célula, forma de la célula, adhesión, tamaño epitelial, núcleos desnudos, cromatina blanda, nucleolos normales, mitosis y clase

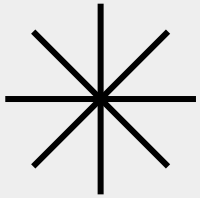
ID	grosor	tamaño_celula	forma_celula	adhesion	tamaño_epitelial	nucleos_desnudos	cromatina_blanda	nucleolos_normales	mitosis	clase
1000025	5	1	1	1	2	0	1	3	1	2
1002945	5	4	4	5	7	1	10	3	2	2
1015425	3	1	1	1	2	2	2	3	1	2
1016277	6	8	8	1	3	3	4	3	7	2
1017023	4	1	1	3	2	4	1	3	1	2
1017122	8	10	10	8	7	5	10	9	7	2
1018099	1	1	1	1	2	6	10	3	1	2
1018561	2	1	2	1	2	7	1	3	1	2
1033078	2	1	1	1	2	8	1	1	1	2
1033078	4	2	1	1	2	9	1	2	1	2

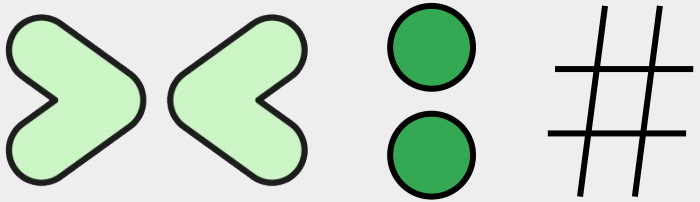
Preprocesamiento

Valores faltantes

Algunas filas tienen “?” como valor. Lo que haremos será eliminar todas las filas que contengan interrogaciones para garantizar que los datos restantes estén completos y listos para su análisis.







Transformación de la columna “clase”

Los valores de “clase” son 2 y 4 según si un paciente tiene cancer o no. Nosotros tendremos que convertir esos valores en binarios.

Paso 2: Entrenar los modelos de ML

1. Cargar el paquete glmnet y entrenar el modelo LASSO
2. Crear las gráficas de LASSO
3. Cargar el paquete e1071 y entrenar el modelo SVM
4. Cargar el paquete nnet y entrenar el modelo ANN

Los paquetes

“glmnet” para LASSO

Es un paquete de R que permite entrenar modelos de regresión lineal y logística con regularización LASSO y Ridge.

“e1071” para SVM

Es un paquete de R que proporciona herramientas para aprendizaje automático y estadísticas. Es muy conocido por ser uno de los estándares de SVM y Naive Bayes.

“nnet” para ANN

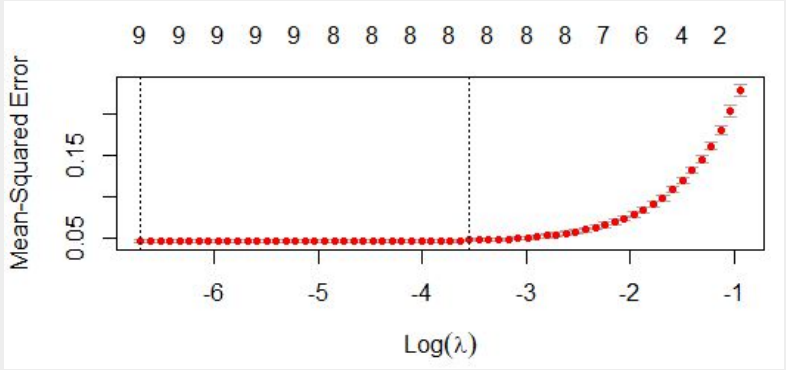
Es un paquete de R que se usa para entrenar redes neuronales artificiales de una sola capa oculta.

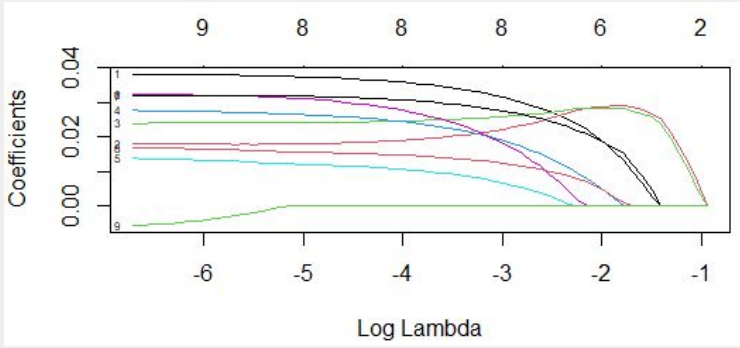
Si queremos entrenar redes neuronales de varias capas (DL), lo ideal sería usar Keras.

Gráficas de LASSO

Validación cruzada

Es un gráfico de validación cruzada para elegir el mejor valor del **parámetro de regularización** (lambda). El eje X muestra la penalización en LASSO. El eje Y muestra el error de validación cruzada





Trayectorias de coeficientes

Es un gráfico de trayectorias de los coeficientes **en función del parámetro de regularización** (lambda). El eje X controla cuánto se penalizan los coeficientes. El eje Y muestra los valores de los coeficientes.

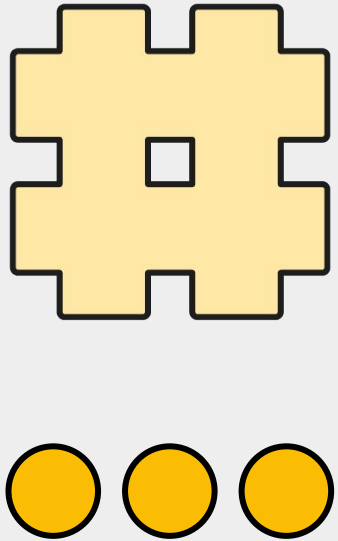
Paso 3: Testear los modelos de ML

1. Predicciones de LASSO
2. Predicciones de SVM
3. Predicciones de ANN
4. Opcional: Crear un modelo ensemble

¿Qué es el modelo ensenble?

Es un modelo de ML que **combina las predicciones** de múltiples modelos para mejorar la precisión y la generalización. En nuestro caso, el modelo ensemble utilizará tanto LASSO como SMV y ANN, fusionando sus resultados mediante una **votación**.

El ensemble sumará las predicciones de los tres modelos fila por fila, calculará la media y redondeará a “0” o “1”, haciendo que sea una votación mayoritaria.



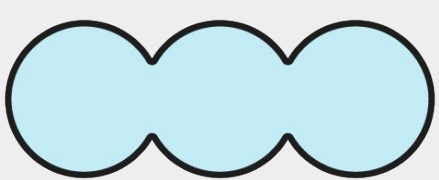
Paso 4: Evaluación de los modelos

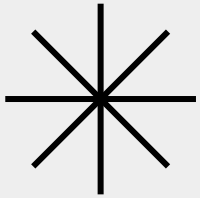
1. Cargar paquete “caret” y crear la matriz de confusión
2. Comprensión de los resultados
3. Cargar paquete “pROC” y análisis de la curva ROC
4. Creación de las curvas ROC
5. Cálculo los valores de AUC

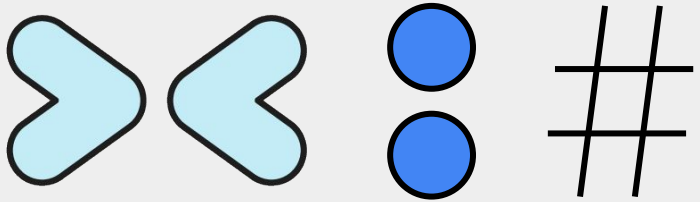
Los paquetes

“caret” para Matrices de Confusión

Es un paquete de R que facilita el entrenamiento, evaluación y selección de modelos de ML.







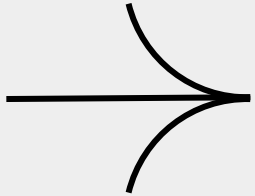
“pROC” para Curvas ROC

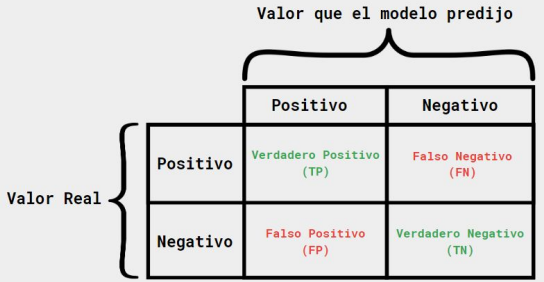
Es un paquete de R que analiza el rendimiento de modelos de clasificación mediante la curva ROC y el AUC.

Matrices de confusión

En RStudio, si las matrices las hacemos con “caret”, nos aparecerán un poco diferentes a lo que vimos anteriormente, pero siguen habiendo los mismos datos:

```
> confusionMatrix(as.factor(prediccion_glm), as.factor(y_prueba))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 145   6
         1   4  72

               Accuracy : 0.9559
                 95% CI : (0.9205, 0.9787)
    No Information Rate : 0.6564
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9017

 Mcnemar's Test P-Value : 0.7518

            Sensitivity : 0.9732
            Specificity : 0.9231
         Pos Pred Value : 0.9603
         Neg Pred Value : 0.9474
             Prevalence : 0.6564
         Detection Rate : 0.6388
   Detection Prevalence : 0.6652
      Balanced Accuracy : 0.9481

       'Positive' Class : 0
```





0.5

Si el valor del AUC es 0.5, el modelo no es efectivo. Lo ideal es tener valores 0.9XX

Aplicar nuevos datos a los modelos entrenados

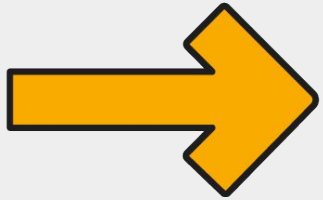
Este es un paso opcional

1. Definir datos hardcodeados
2. Predecir el resultado con LASSO
3. Predecir el resultado con SVM
4. Predecir el resultado con ANN
5. Imprimir los resultados

Capítulo 3

ML con **Python** y **Tensorflow**

Compararemos el código de R con el de Python



pandas

Manipulación y análisis de datos, proporcionando estructuras para manejar datos tabulares de manera eficiente

numpy

Manejo de arrays y operaciones matemáticas avanzadas, incluyendo matrices y estadísticas

matplotlib

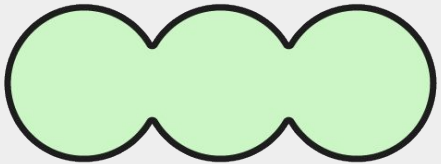
Visualización de datos, gráficos y curvas como la curva ROC o las distribuciones de datos

scikit-learn

Preprocesamiento de datos, data splitting, modelos de ML, métricas y optimización de hiperparámetros entre otros

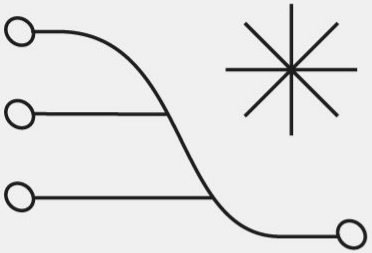
TensorFlow y Keras

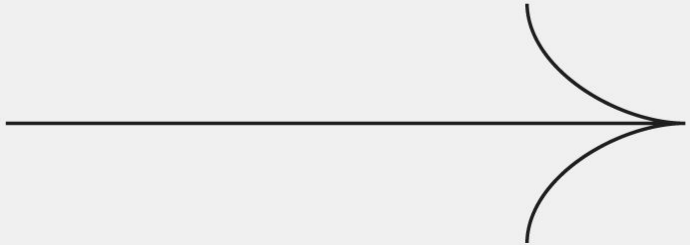
Construcción de modelos de redes neuronales, definición de capas y arquitecturas de redes, etc

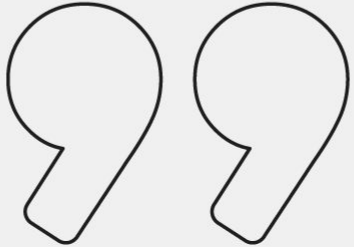


Google Developer Group
A Coruña

Muchas gracias!







Build with AI