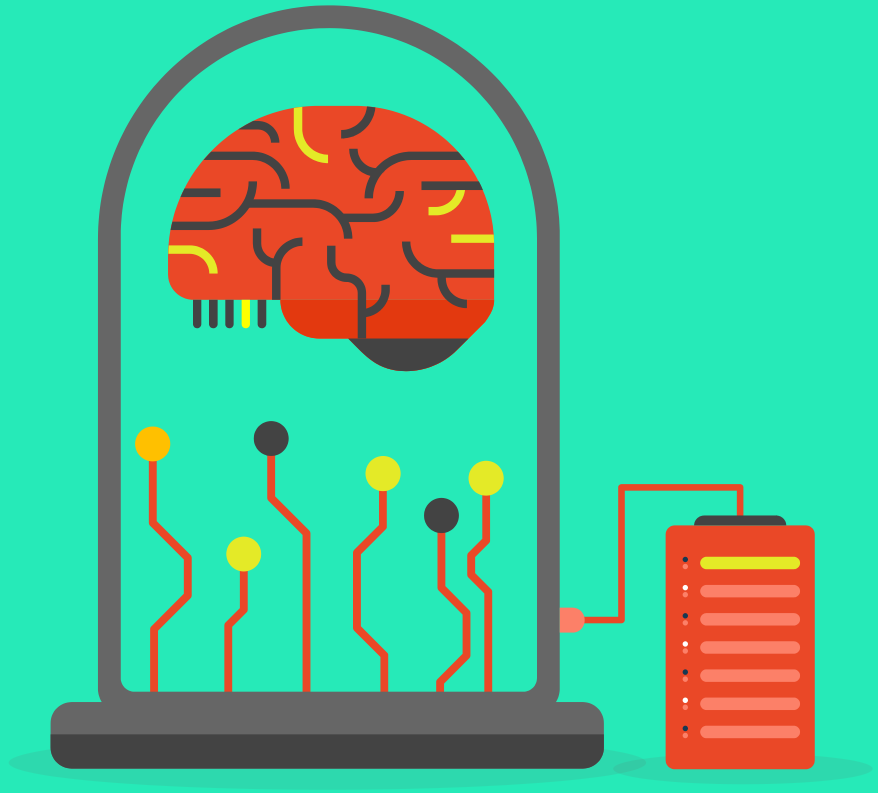
10. Clasificación



Planteamiento del Problema

En la sección pasada conocimos los problemas de Regresión y el cómo estos algoritmos intentan predecir un valor numérico asociado a una variable dependiente. Cuando hablamos ahora de clasificación, nos referimos a que la variable objetivo a predecir se tratará de una *categórica*, es decir, que pertenece a un conjunto de **Clases**. En este sentido, la clasificación puede referirse tanto a datos estructurados como a no estructurados.

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
2	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
5	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
7	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
8	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N

Planteamiento del Problema

Cuando se trata de datos estructurados, todo lo que ya hemos aprendido sobre imputación de variables, codificación de variables categóricas e ingeniería de variables se sigue aplicando de la misma manera, con la diferencia de que la variable a predecir, en este caso, debe conservar su etiqueta categórica que represente aquella clase a la que pertenece el dato.

Por otro lado, un problema de clasificación podría contar con clases binarias (por ejemplo, un dato pertenece o no pertenece a un conjunto) o bien con múltiples clases, e incluso múltiples clases y múltiples etiquetas.

Planteamiento del Problema

Por ejemplo, a partir del dataset a continuación en donde se señalan las clases categóricas:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
2	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
3	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
4	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
5	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
6	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
7	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
8	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
9	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N

Podríamos considerar implementar un modelo de Machine Learning para predecir el valor de la columna binaria “Loan Status”, a partir del resto. O bien, de la columna multiclase “Property Area”.

Planteamiento del Problema

En el caso de problemas de clasificación con datos de entrada no estructurados (como imágenes, texto, audio, video, entre otros), se requiere (si se trata de modelos de aprendizaje supervisado como los que veremos en esta sección) que dichos datos tengan asociados algún tipo de etiqueta que identifique dichas clases.

Perros

Gatos

	spamorham	text
0	ham	Ok lar... Joking wif u oni...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	U dun say so early hor... U c already then say...
3	ham	Nah I don't think he goes to usf, he lives aro...
4	spam	FreeMsg Hey there darling it's been 3 week's n...
5	ham	Even my brother is not like to speak with me. ...
6	ham	As per your request 'Melle Melle (Oru Minnamin...
7	spam	WINNER!! As a valued network customer you have...
8	spam	Had your mobile 11 months or more? U R entitle...
9	ham	I'm gonna be home soon and i don't want to tal...

Medidas del Error en Clasificación

Ahora bien, casi todos los algoritmos de Machine Learning que vimos para el caso de regresión pueden aplicarse a problemas de clasificación tal y como se presentaron en la Sección pasada. Sin embargo, algo fundamental que debe considerarse ahora es la manera de “evaluar” el desempeño de los modelos.

Es decir, ya que no se trata de variables numéricas a predecir, la función de costo cambia y las métricas del error son distintas.

En este caso, estudiaremos las siguientes métricas del error:

- Matriz de Confusión.
- Exactitud.
- Precisión.
- Recall.
- F1-Score.
- AUC.

Matriz de Confusión

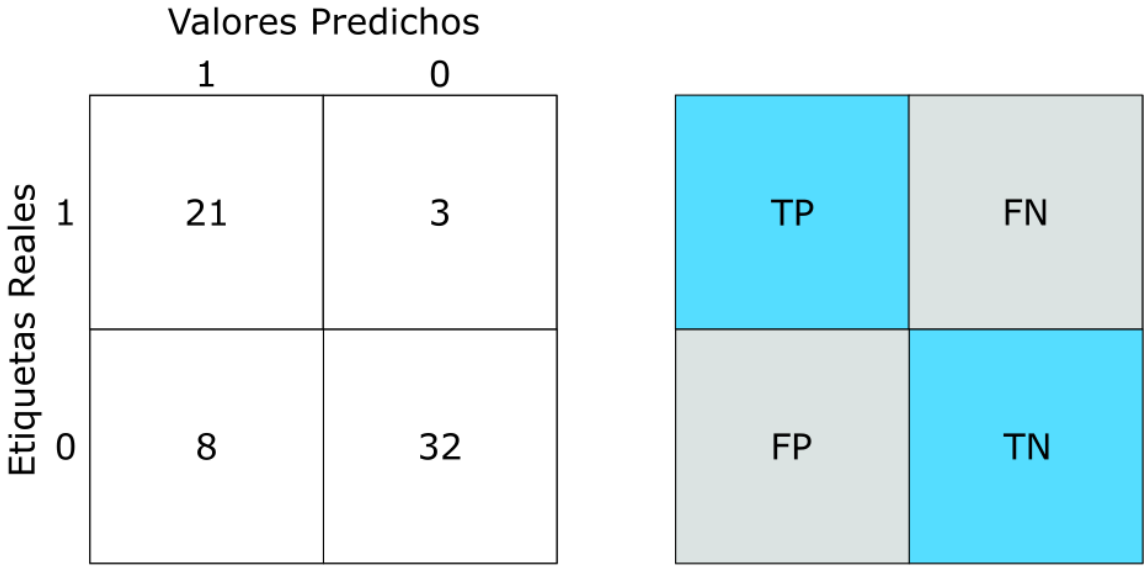
Supongamos que tenemos un conjunto de datos con el cual implementamos un modelo capaz de clasificar datos que pertenecen (1) o no pertenecen (0) a una clase (por ejemplo, un clasificador de imágenes de perros y no-perros). Esto es un clasificador binario. Tras realizar el proceso de entrenamiento/validación cruzada, podemos realizar entonces predicciones sobre el conjunto de entrenamiento (luego, sobre el conjunto de prueba) y contar la cantidad de veces que el modelo se equivoca.

Los diferentes casos a contar serán los siguientes:

- Cuando el dato pertenece a la clase 1, y el modelo predice 1 (Verdadero Positivo).
- Cuando el dato pertenece a la clase 0, el modelo predice 0 (Verdadero Negativo).
- Cuando el modelo pertenece a la clase 1, y el modelo predice 0 (Falso Negativo).
- Cuando el modelo pertenece a la clase 0, y el modelo predice 1 (Falso Positivo).

Matriz de Confusión

Al contar cada una de las ocurrencias de todos estos casos, podemos construir una matriz en donde vamos a colocar, en las filas, los datos que corresponden a los valores reales de las etiquetas, y en las columnas los valores predichos. A esta matriz se le llama **Matriz de Confusión**:



Exactitud

Es importante notar que, al sumar todos los elementos de la matriz de confusión se debe obtener el total de datos clasificados para construir la misma (por ejemplo, el total de datos del conjunto de entrenamiento o prueba). Una primera métrica importante a extraer de esta matriz es la **Exactitud** (accuracy, en inglés). Esta se define como:

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

Es decir, la suma de todos los datos clasificados de manera correcta, entre el total de datos del conjunto. Nótese que si se tiene un clasificador perfecto, la matriz de confusión solo tendrá valores en la diagonal de los TP y TN, y ceros en el resto.

Precisión y Recall

Por otro lado, tenemos la medida conocida como **Precisión** (precision, en inglés). Esta cuenta la exactitud de las predicciones positivas entre todas las positivas:

$$Precision = \frac{TP}{TP + FP}$$

Y esta métrica siempre se usa en combinación con el **Recall**, que representa la proporción de datos positivos que son correctamente clasificados por el modelo:

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

En general, las métricas de precisión y recall se suelen combinar en una sola cantidad llamada el F_1-score, el cual está definido como:

$$F_1 = \frac{TP}{TP + \frac{FP + FN}{2}}$$

Y que se suele emplear para escoger entre modelos distintos. Es decir, sirve como una medida balanceada de evaluación de la calidad de las predicciones entre un grupo de modelos distintos a fin de escoger el más apropiado para la tarea de clasificación dada.

La métrica del F_1-score favorece a clasificadores que tengan un balance entre la precisión y el recall.

Balance entre FP y FN

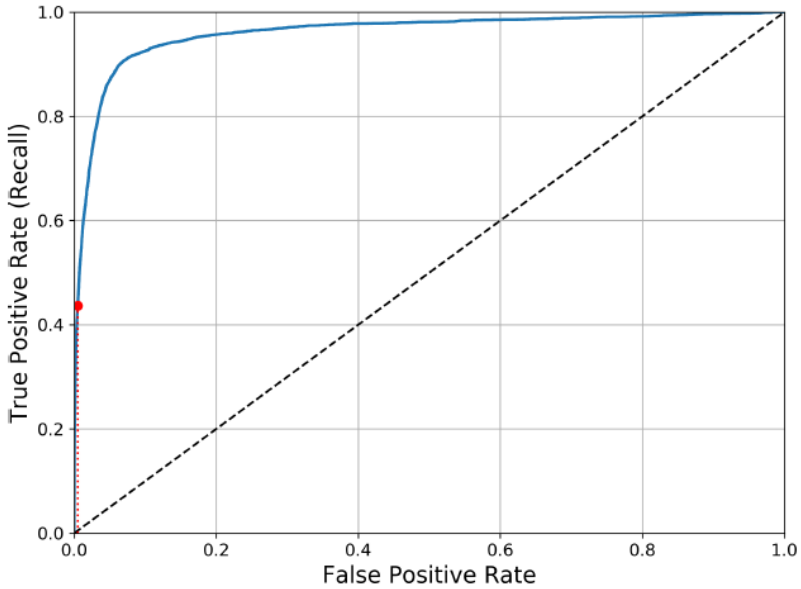
Hablando del balance entre precisión y recall, es importante tener en cuenta lo siguiente: un modelo con 100% de precisión, significa que no existen falsos positivos. Por lo tanto, cada predicción positiva es correcta. Un modelo con 100% de recall significa que no existen falsos negativos, de manera que cada predicción negativa es correcta.

Sin embargo, dependiendo de la aplicación del modelo que deseamos implementar, es importante tener en cuenta este balance y si es necesario o no favorecer una métrica sobre la otra.

Por ejemplo, imaginemos que construimos un clasificador capaz de detectar la presencia de cáncer a partir de imágenes de TAC. Para dicho modelo qué es más deseable: ¿que existan más **Falsos Positivos**, o **Falsos Negativos**?

Área bajo la Curva

Otra medida de desempeño de los clasificadores (binarios) es la conocida como Área bajo la Curva (o **AUC**, por sus siglas en inglés). Esta es una cantidad que se obtiene al calcular el área que se obtiene bajo una curva **ROC** (receiver operating characteristic), que grafica el recall contra la proporción de falsos negativos.



Área bajo la Curva

En este sentido, un clasificador perfecto tendrá un AUC = 1.0, mientras que un clasificador totalmente aleatorio tendrá un AUC = 0.5.

La curva ROC y el valor del AUC se obtienen a partir de los resultados de la matriz de confusión y la librería *Scikit-Learn* de Python ya incorpora funciones que permiten obtener ambas de una manera directa y sencilla.