**Regresiones**

Las tres hipótesis que hay que cumplir:

* Media 0 para el error
* Varianza constante para el error
* Que se cumpla la homocedasticidad, distribuidos en una normal

Para las regresiones es mejor que se entrene para obtener un modelo sencillo, ya que si le pasamos datos y hacemos que se los ‘aprenda’ no va a dar predicciones buenas ya que va a estar sobreajustado. Por eso es muy importante ir validándolo según lo vamos entrenando.

Cuando se trata de una distribución normal (una distribución simétrica) lo normal es utilizar las regresiones lineales.

Es muy importante tener en cuenta la colinealidad entre las variables. Si hay dos variables PERFECTAMENTE COLINEALES habrá que eliminar una de ellas, ya que los coeficientes se calculan a través de la inversa de una matriz, y si hay elementos colineales no se podrá hacer esta inversa o los coeficientes se disparan.

NO ES LO MISMO COLINEALIDAD QUE CORRELACIÓN.

**DescTools**: se trata de una herramienta de descripción muy útil. Tiene un montón de gráficos buenos.

Lo importante de los factores de R es que directamente te lo codifica a binario cuando tu realizas modelos de predicción. En Python hay que emplear el ‘One hot encoding’, que consiste en codificar las variables en 0 y 1. Actualmente se utiliza bastante el ‘Mean Target encoding’. Aprovecha los histogramas de las variables categóricas y le asigna un valor en función al sesgo que posea dicha variable. P.E.: en una categoría en la que haya 70/30, le aplicará 70 y 30 a cada coeficiente. Por último existe el ‘Ordinal Encoding’.

**Remuestreo**

Consta de tres fases:

* Train
* Test
* Validación

Todo esto se hace a través de la base de datos de la que disponemos. Habrá que generar submuestras. Las particiones que se hagan tienen que ser INMUTABLES, lo que esta en Train se queda en Train.

El subgrupo de validación se emplea para tomar decisiones durante el proceso de Train.

El grupo de test SOLO SE PUEDE USAR UNA VEZ. Si lo dejas y lo vuelves a utilizar, estas sesgando el modelo para que se ajuste a ese test.

Un ejemplo de validació es: dividies un conjunto en train, validación y test. El test lo apartas y el train y el de validación los usas. Entrenas el train, sacas el error y lo pruebas en validación. Después cambias los sets, y el de train pasas a ser validación y viceversa. Vuelves a hacer lo mismo y sacas el error de validación. Una vez que veas cual tiene menor error de validación, coges las características de este modelo y haces un train DE TODO EL CONJUNTO (menos el test) pero con las características de ese modelo. Por último lo empleas en el test.

**Modelos lineales generalizados**

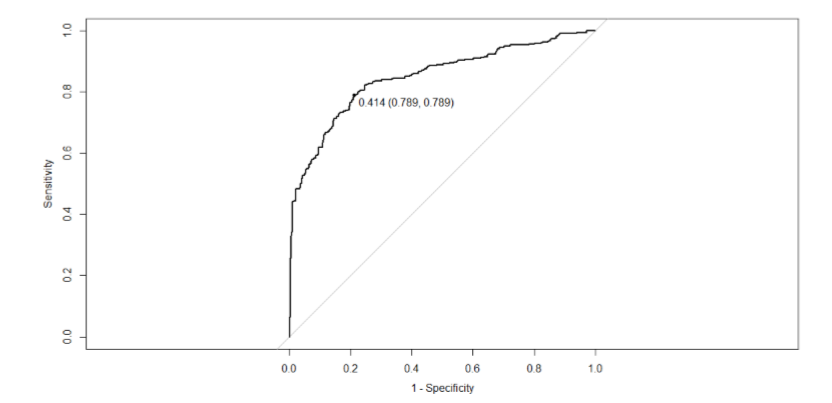
Consiste en, a través de la indicación de la distribución que sigue la variable objetivo, indicarle una función ‘link’ que permita transformar esta distribución y unirla a un modelo lineal de regresión.

Es decir:

Siendo f(Y) la función link que transforma la función de distribución de Y en una nueva distribución. Normalmente esto se hace mediante el ajuste de verosimilitud, que trata de buscar la distribución que mejor se ajusta a Y. Una vez encontrada esta función de distribución se le aplica la función link.

Es importante tener en cuenta que, a la hora de explicar el valor de beta, no podemos decir ‘y’ aumenta una unidad cuando X aumenta en 1. Habría que decir, f(y) aumenta en una unidad cuando x aumenta en 1.

La curva ROC es una ayuda para decidir qué clasificador es mejor:



En el eje X, se le llama 1-specificity. Este eje muestra LOS FALSOS POSITIVOS DIVIDIDO ENTRE LOS FALLOS, es decir, son FALLOS. Por ello buscamos que se ajuste a 0 lo máximo posible. La ‘specificity’ serían los VERDADEROS NEGATIVOS.

En el eje Y tenemos Sensivity, que son los VERDADEROS POSITIVOS. Por ello buscamos que se ajuste lo más que pueda a 1.

**Si mides el área, buscas que se acerque lo más que pueda a 1.** Esto es debido a que el área máximo es 1 (1x1) por lo que cuanto más se acerque mejor.