Bloque 2: Cardiología-Electrocardiograma

Machine Learning

Luis Bote Curiel
Francisco Manuel Melgarejo Meseguer

DTSC

Curso 24-25

Índice

1. Regresión Lineal para Clasificación
2. Análisis Discriminante Lineal
3. Regresión Logísitca
4. Árboles de decisión
5. Bibliografía

Regresión Lineal para Clasificación

Modelo Lineal

Sea $\mathbf{X}^T = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ un conjunto de vectores de entrada y queramos obtener la recta que mejor se ajusta a una serie de valores $\mathbf{Y}$, utilizaremos un modelo linea de la forma

$$f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j = \hat{y}_i$$

En este tipo de modelos asumiremos que la función de regresión $E(\mathbf{Y}|\mathbf{X})$ es lineal o aproximadamente lineal.

Mínimos Cuadrados

Partiendo de un conjunto de entrenamiento $\mathbf{X}_{tr} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ para estimar los parámetros $\boldsymbol{\beta}$. Mediante el método de mínimos cuadrados.

$$RSS(\boldsymbol{\beta}) = \sum_{i}^{N} (y_i - f(\mathbf{x}_i))^2 = \sum_{i}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

Introducción II

Mínimos Cuadrados

Si llamamos $\bar{\mathbf{X}}$ a una matriz formada añadiendo una columna de 1's a $\mathbf{X}$ podemos representar la ecuación anterior como

$$RSS(\boldsymbol{\beta}) = (\mathbf{y} - \bar{\mathbf{X}}\boldsymbol{\beta})^T (\mathbf{y} - \bar{\mathbf{X}}\boldsymbol{\beta})$$

Solución

Minimizando el valor de $RSS(\boldsymbol{\beta})$ con respecto a $\boldsymbol{\beta}$ obtenemos que la solución a este problema viene dada por

$$\frac{\partial RSS(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\bar{X}^T (\mathbf{y} - \bar{\mathbf{X}}\boldsymbol{\beta}) = 0$$

$$\boldsymbol{\beta} = \left(\bar{\mathbf{X}}^T \bar{\mathbf{X}}\right)^{-1} \bar{\mathbf{X}}^T \mathbf{y}$$

Clasificación

Matriz Indicadora

En el caso de la clasificación, es necesario definir una matriz indicadora $\mathbf{Y}$, que contenga la información de los G grupos codificada en vectores $\mathbf{y}$ de la forma $y_k = 1$ si $\mathbf{x} \in G_k$ y cero en el resto.

Mínimos Cuadrados

Teniendo nuestra matriz $\mathbf{Y}$, podremos aplicar la estimación mediante mínimos cuadrados de la forma

$$\hat{\mathbf{y}} = \bar{\mathbf{X}}\left(\bar{\mathbf{X}}^T\bar{\mathbf{X}}\right)^{-1}\bar{\mathbf{X}}^T\mathbf{y} = \bar{\mathbf{X}}\boldsymbol{\beta}$$

Asignaremos la muestra $\mathbf{x}$ a la clase cuya componente en $\hat{\mathbf{y}}$ sea más alta.

¿Por qué funciona?

Este método funciona para la clasificación ya que como sabréis, la regresión lineal estima la esperanza condicionada de grupo dado un conjunto de datos $E(\mathbf{Y}_k|\mathbf{X} = \mathbf{x}) = Pr(G = k|\mathbf{X} = \mathbf{x})$

Análisis Discriminante Lineal

Introducción

Conocimientos Previos

Como ya sabemos, para una clasificación óptima es necesario conocer las probabilidad a posterior de la clase $Pr(G|\mathbf{X})$.

Definiciones

- Llamaremos $f_k(\mathbf{x})$ a la densidad de probabilidad clase-condicionada de que $\mathbf{x}$ pertenezca a la clase $G = k$.
- Llamaremos π_k a la probabilidad a priori de la clase k que cumple $\sum_i^K \pi_i = 1$

Teorema de Bayes

Aplicamos el teorema de Bayes obteniendo:

$$Pr(G = k|\mathbf{X} = \mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_l^K f_l(\mathbf{x})\pi_l}$$

Densidades

Tipos de densidades

En este punto, solo tendremos que saber que tipo de densidad de probabilidad tienen nuestros datos, dependiendo de la suposición que hagamos el método divergirá en alguno de los siguientes:

- Densidades Gaussianas: Discrimiantes lineales o cuadráticos.
- Densidades mezclas de Gaussianas: modelos de mezclas.
- Modelos no paramétricos, (e.g. Naive Bayes).

Densidades Gaussiana

Nuestras densidades vendrán dadas por la siguiente ecuación

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{\frac{-1}{2}(\mathbf{x}-\mu_x)^T \Sigma^{-1}(\mathbf{x}-\mu_x)}$$

donde p es la dimensionalidad de nuestros datos y Σ_k la matriz de covarianza de la clase k-ésima.

Linear Discrimitan Analysis (LDA)

LDA

- En el caso de LDA tendremos que $\Sigma_k = \Sigma \forall k$.
- Nos gustaría comprar 2 clases cualquiera (k y l) mediante el logaritmo del cociente

$$\log \frac{Pr(G=k|\mathbf{X}=\mathbf{x})}{Pr(G=l|\mathbf{X}=\mathbf{x})} = \log \frac{f_k(\mathbf{x})}{f_l(\mathbf{x})} + \log \frac{\pi_k}{\pi_l} =$$
$$= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1}(\mu_k - \mu_l) + \mathbf{x}^T \Sigma^{-1}(\mu_k - \mu_l)$$

Hiperplanos

Las matrices de covarianza iguales hacen que los factores de normalización se cancelen, así como la parte cuadrática en los exponentes. Esta función implica que la frontera de decisión entre las clases k y ℓ (el conjunto donde $\Pr(G = k \mid X = x) = \Pr(G = \ell \mid X = x)$) es lineal en x; en p dimensiones, un hiperplano. Esto es cierto para cualquier par de clases, por lo que todas las fronteras de decisión son lineales. Si dividimos $\mathbb{R}^p$ en regiones clasificadas como clase 1, clase 2, etc., estas regiones estarán separadas por hiperplanos.

LDA en la práctica

Estimación de Parámetros

En la práctica, no conocemos los parámetros de las distribuciones gaussianas y necesitamos estimarlos usando nuestros datos de entrenamiento (para cada una de las clases):

- $\hat{\pi}_k = \frac{N_k}{N}$, donde N_k es el número de observaciones de la clase k;
- $\hat{\mu}_k = \frac{1}{N_k} \sum_{g_i = k} \mathbf{x}_i$;
- $\hat{\Sigma} = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{g_i = k} (\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T$.
- Calcular el resultado de las Funciones discriminantes lineales:$\delta_k(\mathbf{x}) = \mathbf{x}^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$
- Asignar el dato a la clase cuya función de discriminante sea mayor.

Regresión Logísitca

Introducción

Supongamos ahora que modelamos las probabilidades a posteriori de las K clases mediante funciones lineales en x, mientras aseguramos que la suma de ellas valga 1 y que todas pertenecen al intervalo $[0,1]$.

$$\log \frac{Pr(G=1|X=x)}{Pr(G=K|X=\mathbf{x})} = \beta_{10} + \boldsymbol{\beta}_1^T \mathbf{x}$$
$$\log \frac{Pr(G=2|X=x)}{Pr(G=K|X=\mathbf{x})} = \beta_{20} + \boldsymbol{\beta}_2^T \mathbf{x}$$
$$\vdots$$
$$\log \frac{Pr(G=K-1|X=x)}{Pr(G=K|X=\mathbf{x})} = \beta_{(K-1)0} + \boldsymbol{\beta}_{K-1}^T \mathbf{x}$$

- Este modelo está expresado en función de $K-1$ funciones logarítmicas.

Referencia

Dado el modelo presentado, podemos escribir la probabilidad de la clases como

$$Pr(G=1|X=\mathbf{x}) = Pr(G=K|X=\mathbf{x})e^{\beta_{10}+\boldsymbol{\beta}_1^T\mathbf{x}}$$

También sabemos que $\sum_i^K Pr(G=g_i|X=\mathbf{x}) = 1$ que puede reescribirse como

$$\sum_i^K Pr(G=g_i|X=\mathbf{x}) = Pr(G=K|X=\mathbf{x}) + \sum_i^{K-1} Pr(G=K|X=\mathbf{x})e^{\beta_{i0}+\boldsymbol{\beta}_i^T\mathbf{x}}$$

Sacando factor común

$$Pr(G=K|X=\mathbf{x})\left[1+\sum_i^{K-1} e^{\beta_{i0}+\boldsymbol{\beta}_i^T\mathbf{x}}\right] = 1$$

Obteniendo que

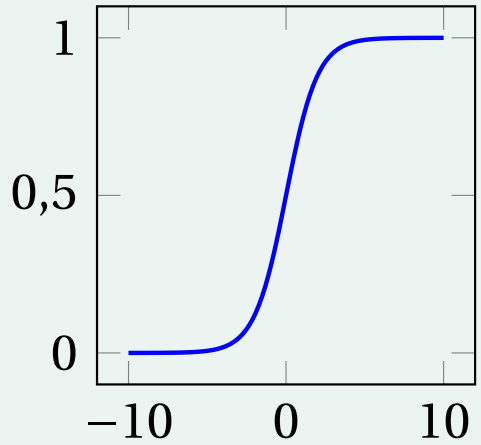
$$Pr(G=K|X=\mathbf{x}) = \frac{1}{\left[1+\sum_i^{K-1} e^{\beta_{i0}+\boldsymbol{\beta}_i^T\mathbf{x}}\right]}$$

De manera que podemos expresar todas las probabilidades a posteriori de clase en función de lo anterior.

Ajustando el modelo

Forma probabilidad

Tenga en cuenta, que dado el modelo anterior, las probabilidades a priori tendrán forma de *softmax*, es decir



Ajustes del modelo

Para ajustar el modelo completo deberemos trabajar con un conjunto de parámetros $\theta = \{\beta_1^T.\beta_2^T,\ldots\beta_{K-1}^T\}$. Para reducir la complejidad de la notación, llamaremos $p_k(\mathbf{x};\theta)$ a $Pr(G = K|X = \mathbf{x})$.

Ajustando el modelo II

Criterio de Ajuste

- Utilizaremos como criterio de ajuste del modelo la **máxima verosimilitud** condicionada G dada X.
- Dado que las probabilidades a posteriori especifican completamente la distribución condicional y que un sujeto solo puede pertenecer a una clase, utilizaremos la distribución multinomial.
- Podemos expresar la función de verosimilitud logarítmica como

$$l(\theta) = \sum_{i}^{N} \log p_{g_i}(x_i;\theta)$$

- Dado que trabajamos con dos clases $p_1(\mathbf{x};\theta) = p(\mathbf{x};\theta)$ y $p_2(\mathbf{x};\theta) = 1 - p(\mathbf{x};\theta)$
- De esta manera, podemos expresar $l(\theta)$ como $l(\boldsymbol{\beta})$

$$l(\boldsymbol{\beta}) = \sum_{i}^{N} \left\{ y_i \boldsymbol{\beta}^T \mathbf{x}_i - \log(1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i} \right\}$$

- Asumimos que $\boldsymbol{\beta} = \{\beta_{01}, \boldsymbol{\beta}_1\}$ y que $\mathbf{x}_i = \{1, \mathbf{x}_i\}$

Ajustando el modelo III

Optimización

- Para optimizar el modelo, calcularemos la primera derivada de la función de verosimilitud y la igualaremos a cero.

$$\dot{l}(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\sum_{i}^{N} \mathbf{x}_i (y_i) - p(\mathbf{x}_i; \boldsymbol{\beta})) = 0$$

- Las cuales son $p+1$ ecuaciones no lineales en $\boldsymbol{\beta}$.
- Dado que el primer componente de $\boldsymbol{x}_i$ es 1, la primera ecuación de puntuación especifica que $\sum_i^N y_i = \sum_i^N p(\mathbf{x}_i; \boldsymbol{\beta})$
- Por lo que el número de elementos esperados de la clase 1 coincide con el número observado, ídem para clase 2.
- La ecuación de puntuación hay que resolverla mediante métodos numéricos (Newton-Raphson), este requiere el cálculo de la segunda derivada o Hessiano

$$\ddot{l}(\boldsymbol{\beta}) = \frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\sum_{i}^{N} \mathbf{x}_i \mathbf{x}_i^T p(\mathbf{x}_i; \boldsymbol{\beta})(1 - p(\mathbf{x}_i; \boldsymbol{\beta}))$$

Ajustando el modelo IV

Optimización

- De manera que deberemos realizar un calculo iterativo de $\boldsymbol{\beta}$ utilizando la siguiente ecuación

$$\boldsymbol{\beta}^n = \boldsymbol{\beta}^o - \ddot{l}(\boldsymbol{\beta})^{-1}\dot{l}(\boldsymbol{\beta})$$

- Nótese que las derivadas hay que evaluarlas en $\boldsymbol{\beta}^o$
- En notación matricial, podemos expresar esta ecuación como

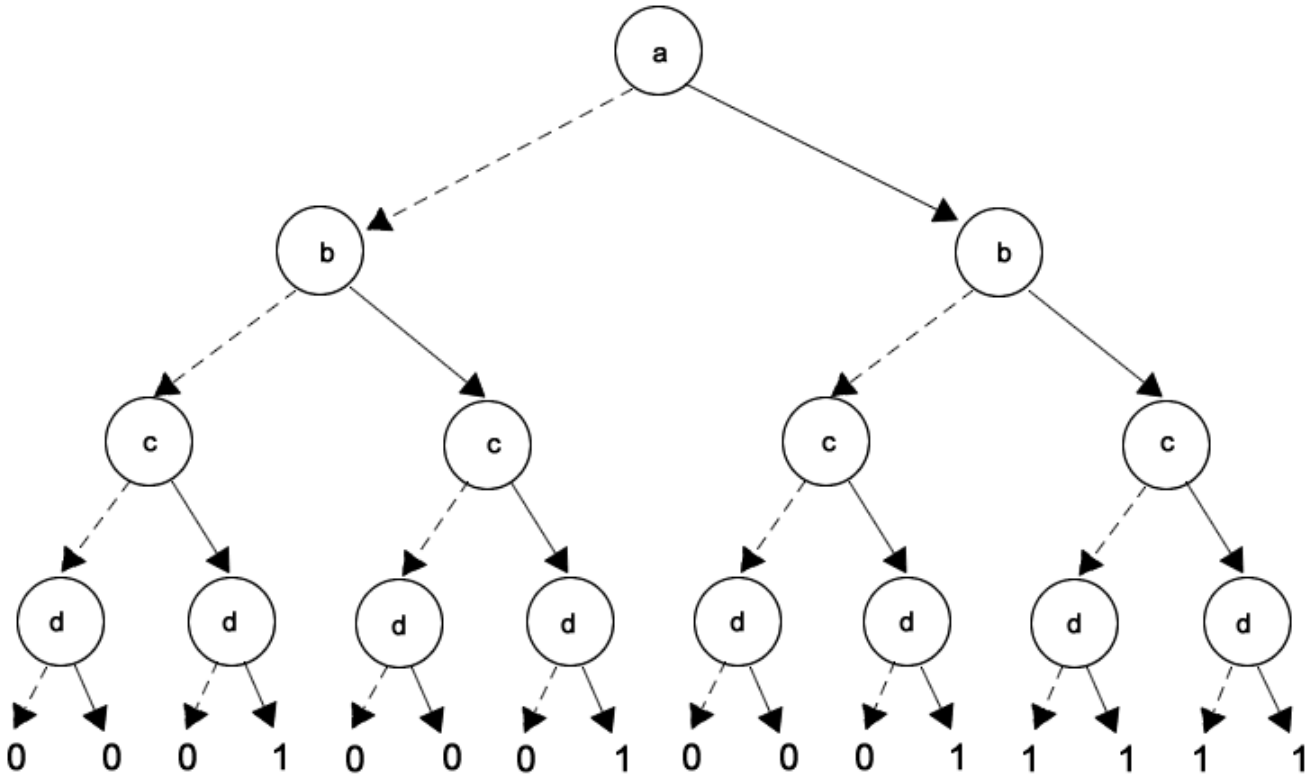
$$\boldsymbol{\beta}^n = (\mathbf{X}^T\boldsymbol{W}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{W}\boldsymbol{z}$$

$$\boldsymbol{z} = \mathbf{X}\boldsymbol{\beta}^o + \boldsymbol{W}^{-1}(\mathbf{y} - \boldsymbol{p})$$

- donde $\boldsymbol{p}$ es el vector de probabilidades del elemento i-ésimo $p(\mathbf{x}_i;\boldsymbol{\beta}^o)$ y $\boldsymbol{W}$ es una matriz $N \times N$ cuya diagonal vale $p(\mathbf{x}_i;\boldsymbol{\beta}^o)(1 - p(\mathbf{x}_i;\boldsymbol{\beta}^o))$

Árboles de decisión

Árbol



Introducción

Árboles

- Los modelos basados en árboles, dividen el espacio en regiones rectangulares y ajustan un modelo sencillo (cte) en su interior.
- Consideraremos un problema de regresión en el que querremos obtener $\mathbf{y}$ a partir de una variable $\mathbf{x}$ de dos dimensiones.
- Para simplificar todo el procedimiento, trabajaremos con árboles de decisión binario, esto es, en cada decisión, el espacio será divido en 2 regiones de acuerdo con la media del atributo $\mathbf{y}$ de cada región.

Regresión

- De esta manera, tendremos N observaciones de la forma $(\mathbf{x}_i, y_i)$, donde $\mathbf{x}_i$ es un vector de p elementos.
- Si nuestro espacio se encuentra divido en M regiones de la forma R_m, la función de regresión vendrá dada por

$$f(\mathbf{x}) = \sum_{m}^{M} c_m I(\mathbf{x} \in R_m)$$

División

Criterio

- Si aplicamos un criterio de división basado en la minimización de la suma errror cuadrático $e = \sum_(y_i - f(\mathbf{x}_i))^2$, se deriva que el valor de óptimo de c_m es

$$\hat{c}_m = \langle (y_i | \mathbf{x}_i \in R_m) \rangle$$

- Este no es el mejor criterio existente para los árboles, pero dada su sencillez es el que vamos a ver en teoría, no así en prácticas.

Crecimiento

- Llamaremos j a la variable divisora y s al valor de dicha variable. De esta manera, definimos los semiplanos

$$R_1(j,s) = \{\mathbf{X} | \mathbf{x}_j \leq s\}, \quad R_2(j,s) = \{\mathbf{X} | \mathbf{x}_j > s\}$$

- Entonces, solo hay que buscar la pareja de variables j, s que minimizan la función de pérdidas

$$\min_{c1} \sum_{\mathbf{X} \in R_1} (y_i - c_1)^2 + \min_{c2} \sum_{\mathbf{X} \in R_2} (y_i - c_2)^2$$

División

Criterio

- Para sorpresa de nadie, el resultado de la minimización anterior es

$$\hat{c}_a = \langle (y_i | \mathbf{x}_i \in R_a(j, s)) \rangle$$

- Este proceso puede extenderse hasta un número de etapas inabarcable, por ello se suele limitar el tamaño máximo del árbol.
- Consideraciones:
  - Si el árbol es muy pequeño, probablemente no podrá captura la esencia de los datos.
  - Si el árbol es muy grande, probablemente sobreajustará los datos.

Poda

- Una forma buena de lidiar con este problema, es dejar crecer el árbol y luego aplicar un criterio de poda, eliminando así aquellas ramas que aporten menos.

Poda

¿Qué es aportar menos?

- Definiremos T como un sub-árbol obtenido tras la poda de T_0, esto es, eliminando alguno de sus nodos y reubicando esas muestras.
- Llamaremos m a los nodos terminales y $|T|$ al número de nodos terminales en T

$$N_m = \#\{\mathbf{x}_i \in R_m\}$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \hat{c}_m)^2$$

- Definiremos el criterio de coste-complejidad como

$$C_\alpha(T) = \sum_{m}^{|T|} N_m Q_m(T) + \alpha |T|$$

Poda II

Objetivo

- Para encontrar el T_α que minimiza $C_\alpha(T)$ usamos la poda del eslabón más débil:
  1. Colapsamos sucesivamente el nodo interno que produce el menor aumento por nodo en $mN_mQ_m(T)$.
  2. Continuamos hasta producir el árbol de un solo nodo (raíz).
  3. Esto da una secuencia (finita) de sub-árboles.
  4. Se puede demostrar que esta secuencia debe contener T_α.

Clasificación

- Para el caso de clasificación deberemos realizar unos pequeños cambios:
  - Cambiar los c_m por

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} I(y_i = k)$$

  - La clasificación se basa en asignar a cada $\mathbf{x}_i$ el valor mayoritario en el sub-espacio R_m que toque.
  - Cambiar el $Q_m(T)$ por una función como el índice Gini o la entropía cruzada.

Índices

Índice Gini

$$Gini = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Mide la probabilidad de que un elemento seleccionado al azar sea clasificado incorrectamente si se asigna una etiqueta aleatoria basada en la distribución de clases en el conjunto de datos. Un valor de Gini de 0 indica una pureza perfecta (todos los elementos pertenecen a una sola clase), mientras que un valor más alto indica mayor impureza.

Entropía

$$E = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

La entropía mide la cantidad de incertidumbre o impureza en un conjunto de datos. Un valor de entropía de 0 indica que todos los elementos pertenecen a una sola clase (pureza máxima), mientras que valores más altos indican mayor impureza.

Ventajas y desventajas

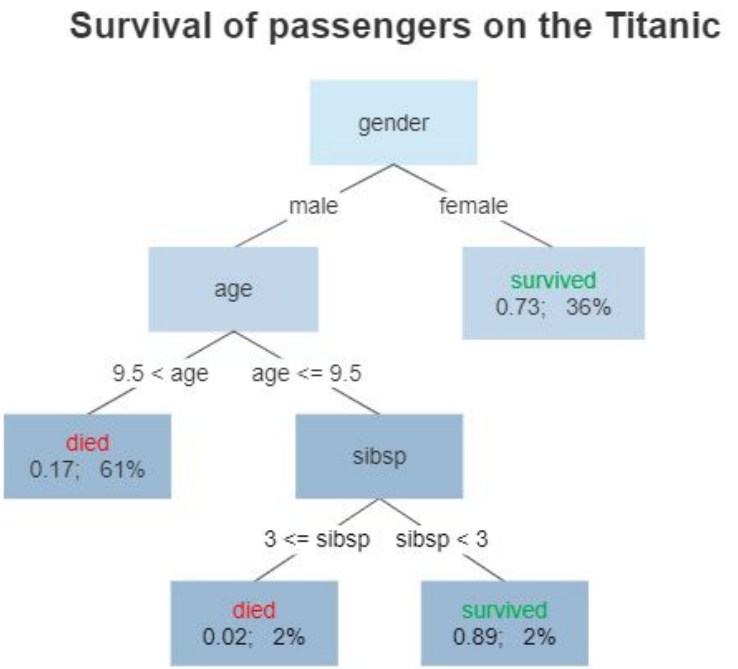
Ventajas

- Los árboles son muy fáciles de explicar a las personas. De hecho, ¡son incluso más fáciles de explicar que la regresión lineal!
- Algunas personas creen que los árboles de decisión reflejan más de cerca la toma de decisiones humana.
- Los árboles pueden mostrarse gráficamente y son fácilmente interpretados incluso por un no experto (especialmente si son pequeños).
- Los árboles pueden manejar fácilmente predictores cualitativos sin la necesidad de crear variables ficticias.

Desventajas

- Los árboles generalmente no tienen el mismo nivel de precisión predictiva que algunos de los otros enfoques de regresión y clasificación.
- Los árboles pueden ser muy no robustos. En otras palabras, un pequeño cambio en los datos puede causar un gran cambio en el árbol estimado final.

Técnicas de reducción de varianza: Bagging



Técnicas de reducción de varianza: Bagging II

Bagging

Esta técnica se utiliza para reducir la varianza de los árboles mediante la generación de B árboles formados a partir de copias del conjunto de entradas realizados mediante selección de muestras con reemplazo.

Varianza

- Recordad que para un conjunto de n observaciones independientes $Z_1, Z_2 \dots Z_n$ su media $\bar{Z}$ tiene varianza σ^2/n.
- Calcularemos los B $\hat{f}_1(\mathbf{x}_i), \hat{f}_2(\mathbf{x}_i), \dots, \hat{f}_B(\mathbf{x}_B)$ y obtendremos la salida $\hat{f}_{avg}(\mathbf{x}) = \frac{1}{B}\sum_i^B \hat{f}_i(\mathbf{x}_i)$

Error Out-Of-Bag (OOB)

OOB

- En la técnica de Bagging tenemos un problema para calcular la validación cruzada ya que en datasets grandes, es muy pesado.
- Se puede observar que cada árbol utiliza $2/3$ del total de las muestras. Entonces calcularemos este error sobre las muestras que se quedan fuera.

Interpretabilidad

- Al dividir un nodo, se busca reducir la impureza.
- Reducción de la impureza = diferencia entre la impureza del nodo padre y la suma ponderada de las impurezas de los nodos hijos.
- La importancia se calcula como la reducción promedio de la impureza que la característica proporciona a lo largo de todos los árboles.

Técnicas de reducción de varianza III

Random Forest

- Es una modificación de Bagging, en la que intentamos forzar la incorrelación entre los árboles, de manera que estos sean más independientes y así que la media de la que hablamos antes se cumpla.
- Para ello, forzaremos que en cada una de las divisiones solo se use un número $m < p$ predictores, típicamente $m = \sqrt{p}$.

Boosting

- Es una técnica que consiste en la mejora sucesiva y **controlada** de los árboles mediante conjuntos de datos bootstrapeados.
- Esto puede llevar al overfitting por ello es importante el control mediante un parámetro de ajuste λ.
- El algoritmo se centra en la mejora de los residuos en vez de en la salida global, de manera que va arreglando poco a poco el árbol general. En cada iteración, se van añadiendo *arbolitos* en los *peores* nodos terminales.

Técnicas de reducción de varianza IV

Algoritmo Boosting

1. Establecer $\hat{f}(\mathbf{x}) = 0$ y $r_i = y_i$ para todos los i en el conjunto de entrenamiento.
2. Para $b = 1, 2, \ldots, B$, repetir:
   1. Ajustar un árbol $\hat{f}_b$ con d divisiones ($d+1$ nodos terminales) a los datos de entrenamiento $(\mathbf{X}, r)$.
   2. Actualizar $\hat{f}$ añadiendo una versión reducida del nuevo árbol:

$$\hat{f}(x) \leftarrow \hat{f}(\mathbf{x}) + \lambda \hat{f}_b(\mathbf{x}).$$

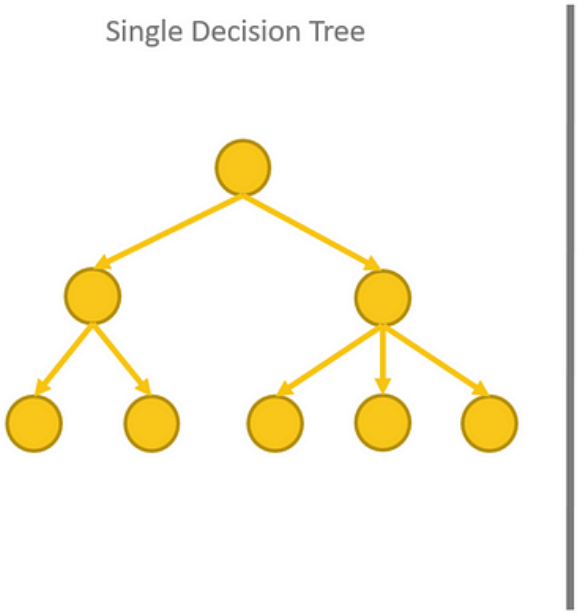
   3. Actualizar los residuos:

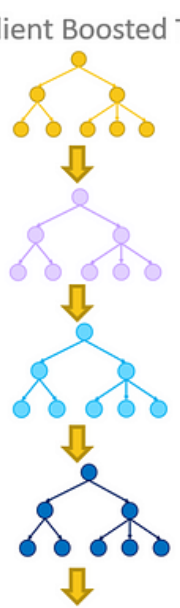
$$r_i \leftarrow r_i - \lambda \hat{f}_b(\mathbf{x}_i).$$

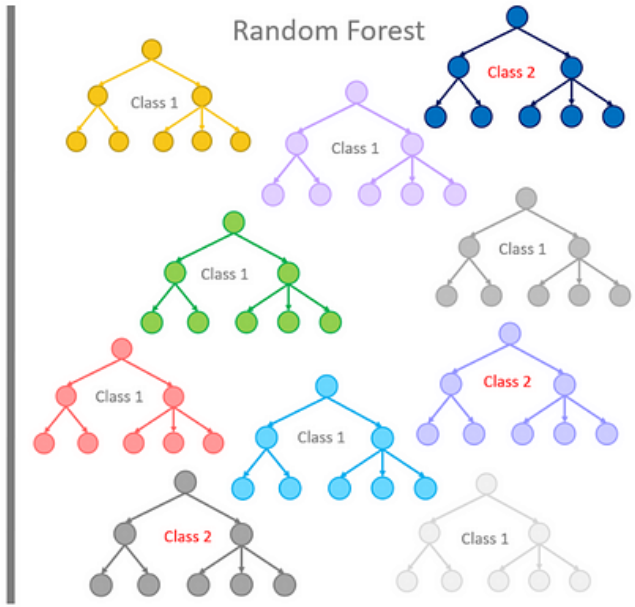
3. Devolver el modelo potenciado:

$$\hat{f}(\mathbf{x}) = \sum_{b=1}^{B} \lambda \hat{f}_b(\mathbf{x}). \quad \hat{\mathbf{y}} = \frac{1}{1 + e^{-\hat{f}(\mathbf{x})}} > 0{,}5$$

Comparación







Bibliografía

Bibliografía

G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*.
Springer International Publishing, 2023.

T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*.
Springer New York, 2009.