Modelación Supervisada

Las organizaciones se encuentran en una etapa de transformación con respecto a la cantidad de datos que está generando un mundo que es cada vez más digital. La revolución del BigData brinda oportunidades a futuros profesionales además de importantes ventajas competitivas a las organizaciones. La inteligencia analítica (analytics en inglés) se eleva más allá de la tradicional inteligencia de negocio. En este campo no solo es conocer al cliente a través de la integración de su información, es ir más allá; encontrar patrones no triviales, modelación matemática sofisticada, predicción. Todo lo anterior basado en la nutrición que los datos proveen.

Metodología de modelado

Para poder realizar un modelo matemático, se requiere una estructura especial de datos. Antes de adentrarnos en el tema, es necesario saber que existirán dos clasificaciones fundamentales para entrenamiento de modelos:

- Aprendizaje supervisado: El modelo aprende con base en ejemplos "etiquetados", es decir, cuenta con variable objetivo a predecir/clasificar.
- Aprendizaje no supervisado: El modelo aprende de observaciones "sin etiqueta", carece de variable objetivo. Su función principal es clasificar.

Una vez entendido lo anterior, expliquemos la metodología fundamental de armado de información para modelado.

El supuesto principal de los modelos predictivos en ciencia de datos es que el futuro se comportará como el pasado reciente. "Reciente" dependerá del contexto en el que nos encontremos y el negocio a analizar. La información histórica del pasado reciente se utilizará para entrenar un modelo para poder predecir futuros estados del sistema. Por ejemplo, para predecir si en los próximos 6 meses un cliente incumplirá el pago de un crédito, podemos determinar con base en un modelo logístico alimentado con la información de los últimos 12 meses la probabilidad de que dicho cliente incumpla. Esto se ejemplifica en el siguiente diagrama:

Observación												Desempeño					
t-11	t-10	t-9	t-8	t-7	t-6	t-5	t-4	t-3	t-2	t-1	t	t+1	t+2	t+3	t+4	t+5	t+6

La tabla de datos que será presentada al modelo deberá contener la siguiente estructura: sean x_i un conjunto de variables explicativas y sea Y la variable objetivo, se busca encontrar un operador matemático f tal que $Y = f(x_i)$. Dichos operadores serán discutidos en la siguiente sección. Cada renglón de la tabla representa un vector de entrada en un hiperespacio, donde tras la presentación sucesiva de dichos vectores al algoritmo de aprendizaje versus la variable objetivo completarán el entrenamiento del modelo.

Adicionalmente, debemos considerar la tarea que deseamos realizar, dichas tareas se conjuntan en dos grandes grupos:

- Clasificación: Variable de respuesta discreta, su intención es determinar a qué clase determinada por la variable objetivo pertenece cada observación.
- Regresión: Variable de respuesta continua, busca estimar un valor numérico en los reales para cada observación en los datos.

Medidas de precisión de los modelos.

Un modelo necesita ser sujeto a evaluación, para ello, consultaremos 4 estadígrafos que nos ayudarán en la elección del modelo que mejor ajuste a nuestras necesidades, cabe señalar que no siempre el modelo con los estadígrafos más potentes será el elegido, nuestra decisión puede cambiar con base en criterios de parsimonia o negocio. Los estadígrafos son: Missclassification rate, ROC Index, GINI Index, Kolmogorom-Smirnov.

ROC Index

Acrónimo de receiver operating characteristic (característica operativa del receptor) es una herramienta que grafica la sensibilidad versus 1-especificidad para un clasificador dicotómico en el cual varía el punto de corte de la clasificación para construir cada punto de la curva. La sensibilidad se refiere a la probabilidad de clasificar correctamente un caso cuyo estado original es "positivo" (1). Se calcula como la proporción de "True Positives" en una matriz de confusión:

		Predicted	
		0	**1**
Observed	**0**	True Negatives	False Positives
	1	False Negatives	True Positives

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

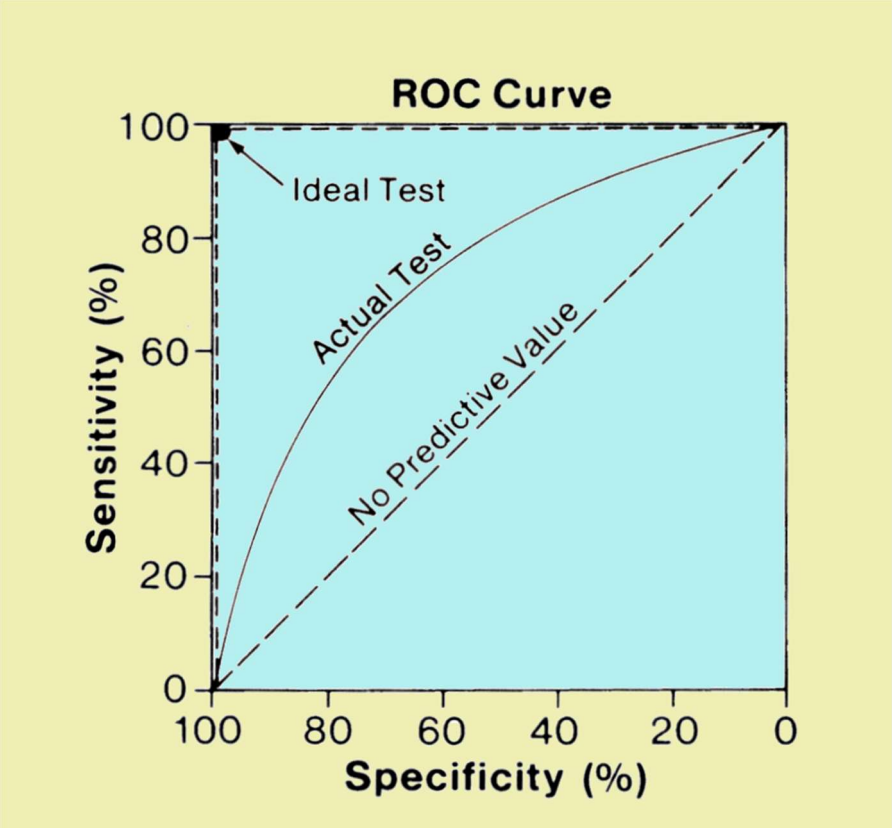
Por otro lado, la especificidad consiste en la probabilidad de clasificar correctamente un caso cuyo estado original es "negativo" (0). Se calcula como la proporción de "True Negatives":

$$Especificidad \quad \frac{TN}{TN + FP}$$

Las medidas anteriores son calculadas para cada punto de corte de probabilidad. Para ejemplificar esto, necesitamos tener calculada la probabilidad de ocurrencia del evento de cada una de las

observaciones dentro de nuestro conjunto de datos, así, si quisiéramos dividir en 10 nuestro diagrama ROC tomaríamos como "unos predichos" a todos aquellos que superaran la probabilidad de 0.1 de nuestro evento de interés, después a los que superaran 0.2 y así sucesivamente, en cada paso calcularemos la sensibilidad y la especificidad formando nuestros pares ordenados para la curva ROC.

El índice ROC es conocido a menudo como AUC (Area Under de ROC Curve) teniendo un valor de 1 para ajuste perfecto, 0.5 indica que el modelo es idéntico al efecto del azar y un valor menor a 0.5 implica que nuestro modelo es peor que el azar.



Sin ser mandatorio, los valores de ROC index se aceptan en el intervalo 0.6-0.8, 0.85 nos indicaría un excelente modelo mientras que un valor mayor a 0.85 indica sospecha. Entre mayor es el valor de ROC index, mejor es el poder discriminatorio del modelo.

Gini Index

El valor GINI es equivalente a la expresión $2\,ROC - 1$, a menudo se utiliza debido a que es igual al estadígrafo de Mann-Whitney-Wilcoxon para comparación de dos muestras independientes.

Kolmogorov-Smirnov

El estadígrafo de Kolmogorov-Smirnov (en lo sucesivo KS) es un estimador de la máxima diferencia entre dos distribuciones acumuladas. Si bien es la medida más usada en la industria, posee la desventaja de solo verificar un punto (la máxima separación), es por ello que debemos contrastar todos los estadígrafos para poder tomar una decisión.

KS se construye a partir de la expresión siguiente:

$$KS = \max|B(s) - G(s)|$$

Donde $B(s)$ y $G(s)$ son las distribuciones acumuladas de "Buenos" (usualmente ceros) y "Malos" (usualmente unos) utilizada comúnmente en la terminología de riesgos. Entre mayor sea el valor de KS, más separadas se encuentran las distribuciones y por tanto, el modelo clasifica de mejor manera.

Missclasification Rate

Partiendo de la matriz de confusión, es la proporción que representa la suma de los False Positives y False Negatives con respecto al total. A menudo se utiliza su complemento denominado Accuracy.

Aprendizaje máquina

Conocido en inglés como Machine Learning, es un método de análisis de datos que automatiza la construcción de modelos analíticos a través del uso de algoritmos que iterativamente aprenden de los datos. El aprendizaje máquina permite a las computadoras encontrar patrones careciendo de programación explicita que direccione la búsqueda.

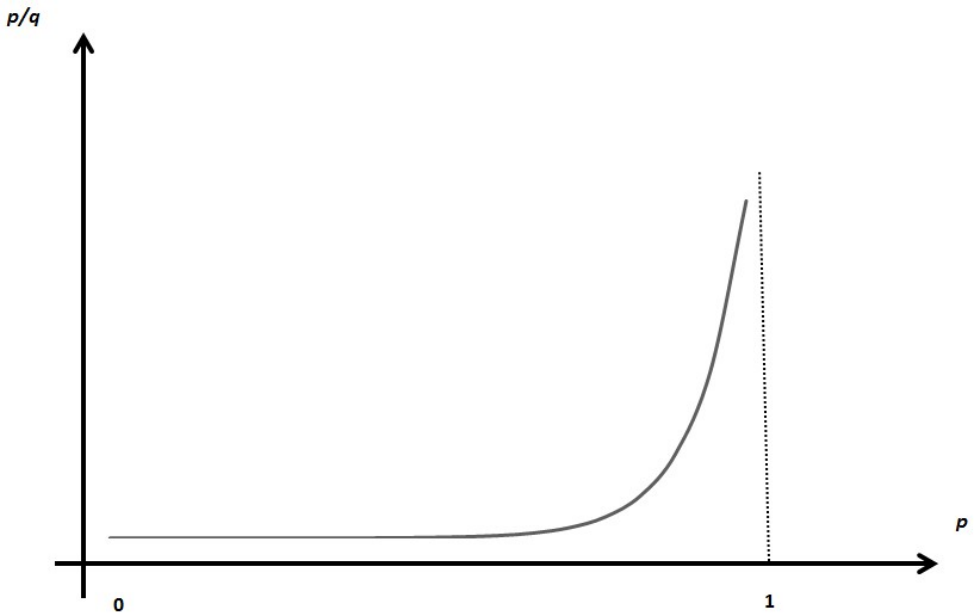
El aprendizaje máquina se ha ido transformando con el paso de los años, si bien muchos de los algoritmos llevan un largo tiempo vigentes, la habilidad para aplicar automáticamente cálculos matemáticos complejos a grandes volúmenes de datos una y otra vez cada vez con mayor velocidad es un desarrollo reciente y es considerado como tecnología de punta. Algunas aplicaciones recientes del aprendizaje máquina se enlistan a continuación:

- Los sistemas de conducción automática de autos como los de Tesla y Google
- Recomendación en línea como Amazon, Netflix, etc.
- Análisis de redes sociales. ¿qué dicen los clientes de mi compañía?
- Prevención de fraudes.
- Reconocimiento de imágenes
- Aplicaciones militares

A continuación analizaremos cada una de las técnicas más utilizadas en el aprendizaje máquina.

Regresión logística

La regresión logística nos permite modelar un evento dicotómico en forma probabilística donde p representa la probabilidad de éxito. Dependiendo del contexto, "éxito" puede interpretarse de manera relativa, en nuestro estudio nos referiremos a "éxito" por aquellos casos de representen una característica de interés (tomar un producto, incumplir un préstamo, enfermar, etc.). El complemento de p se denota por $q = 1 - p$, en ocasiones, es útil el cociente $\frac{p}{q} = \frac{p}{1-p}$ denominado momio (odd's) que representa cuanto más probable es el éxito que el fracaso. Lo anterior es la base de modelación matemática en una cantidad importante de industrias, ¿Tomará el cliente la oferta? ¿Se presentará un fraude? ¿El producto es rentable? ¿El cliente devolverá el crédito? Son ejemplos de preguntas a las cuales podemos responder mediante regresión logística. Observemos la siguiente gráfica:



Estamos interesados en modelar linealmente la probabilidad de cierto evento en función de un conjunto de variables idealmente ortogonales, sin embargo, la curva anterior presenta un comportamiento exponencial. Supongamos que cada variable x_i contribuye proporcionalmente a nuestra variable objetivo (p/q) , al linealizar los momios podemos proponer:

$$Ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{i=1}^{n} \beta_i \, x_i$$

Si despejamos p, tenemos:

$$\frac{p}{1-p} = exp\left(\beta_0 + \sum_{i=1}^{n} \beta_i\, x_i\right) \Rightarrow$$

$$\frac{1-p}{p} = exp\left(\quad_0 - \sum_{i=1}^{n} \beta_i\, x_i\right) \Rightarrow$$

$$\frac{1}{p} = 1 + exp\left(-\beta_0 - \sum_{i=1}^{n} \beta_i\, x_i\right) \Rightarrow$$

$$p(E) = \frac{1}{1 + ex\ \left(-\beta_0 - \sum_{i=1}^{n} \beta_i\, x_i\right)}$$

Que es conocida como función logística (sigmoidal), la cual nos otorga valores entre 0 y 1 los cuales interpretaremos como la probabilidad de ocurrencia del evento para cada vector $\vec{X}$ presentado. Los parámetros β_i pueden estimarse a través del método de máxima verosimilitud.

Transformación WOE

La regresión logística y muchas otras técnicas de Machine Learning exigen la introducción de variables estrictamente continuas. Un interesante auxiliar que nos permite resolver esta cuestión es la transformación WOE. WOE representa al acrónimo inglés Weight of Evidence, es una transformación (discretización) de las variables de entrada tal que presenten un riesgo creciente (o decreciente) en cada uno de sus segmentos. Para calcular el WOE, utilizamos la siguiente fórmula:

$$WOE = \ln \frac{P(No\ evento)}{P(Evento)}$$

Una vez transformada, mediremos la potencia de cada variable con respecto a la variable objetivo (poder de discriminación) por medio de una medida conocida como Information Value (IV), calculado de la siguiente manera:

$$IV = \sum\nolimits_{atributo} [(P\ No\ evento - P\ Evento)WOE]$$

Los valores del IV indican el nivel de importancia de la característica, a saber:

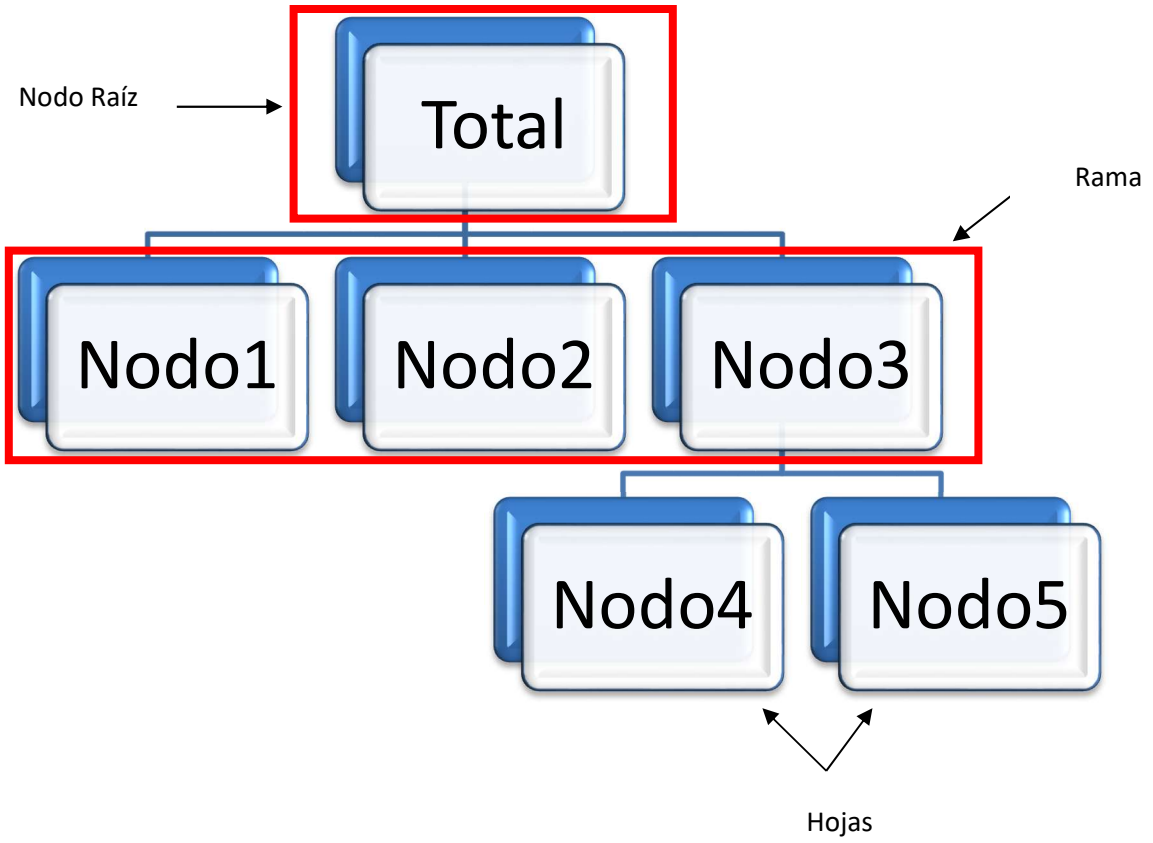
<0.1	Débil
<0.3	Medio
<0.5	Fuerte

>0.5	Sobrepredictivo

Para que una característica ingrese al modelo, se recomienda que tenga un IV mínimo de 0.02

Árboles de decisión

Un árbol empírico representa una segmentación de los datos creada aplicando una serie de reglas simples donde cada regla asigna una observación a un grupo basado en los valores de su vector de variables de entrada. Dichas reglas son aplicadas en cascada, en consecuencia, se formará una jerarquía de segmentos dentro de los segmentos. Tal jerarquía es llamada árbol y cada segmento se denomina nodo. El segmento inicial contendrá la totalidad de los datos y es conocido como nodo raíz. Los sucesores de un nodo formarán una rama, donde los nodos terminales son llamados hojas. El tipo de decisión involucrada dependerá expresamente del contexto.



Entre las principales ventajas de esta técnica se encuentran:

- Producción de un conjunto de reglas fácilmente interpretables
- Inclusión de datos ausentes
- Implementación relativamente sencilla

- Utiliza la mejor técnica matemática de partición de acuerdo a la escala de las variables

El criterio para evaluar la regla de partición se basa en dos fundamentos:

- Test estadístico de significancia (Chi-cuadrado, F)
- La reducción de la varianza (Entropía, medida de impureza de Gini)

Los árboles pueden ser clasificados de acuerdo a la naturaleza de su variable objetivo, a saber:

- Clasificación: Ajusta de acuerdo a la distribución de los datos
- Regresión: Ajusta de acuerdo a la media

Los árboles de decisión utilizan un criterio de disparidad acorde a la escala de las variables para formar las particiones:

- Prueba de diferencia de medias por estadístico F para variables de intervalo
- Pruebas chi-cuadrado para variables discretas
- Entropía o Gini para variables ordinales

Prueba de diferencia de medias

Se prueba la hipótesis de que un conjunto de medias de dos o más grupos son distintas entre sí.

$H_0 = \mu_1 = \mu_2 = \cdots = \mu_k$

$H_a = Al\ menos\ uno\ de\ los\ grupos\ tiene\ una\ media\ distinta\ al\ resto\ de\ los\ grupos$

Se realiza un comparativo entre la razón de dos varianzas lo que deriva en una prueba F

Prueba Chi-cuadrado

La prueba se utiliza principalmente para datos provenientes de escala nominal y prueba la discrepancia entre una distribución observada y una teórica (esperada). El estadígrafo de la prueba se construye como sigue:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Entre mayor sea el valor de χ^2 mayor discrepancia hay entre las distribuciones.

Impureza de Gini

Se mide como la suma de la probabilidad de que un objeto sea elegido multiplicada por la probabilidad de cometer el error de categorizarlo. Supongamos que i toma los valores $\{1,2,\dots,m\}$ y sea f_i la fracción de elementos etiquetados con el valor i en el conjunto.

$$I_G(f) = \sum_{i=1}^{m} f_i\,(1 - f_i\,) = \sum_{i=1}^{m} \left(f_i - {f_i}^2\right) = 1 - \sum_{i=1}^{m} {f_i}^2$$

Ganancia de información

Conocida como la divergencia de Kullback-Leibler o entropía relativa, es una medida no simétrica de la diferencia entre dos distribuciones de probabilidad.

$$I_E(f) = -\sum_{i=1}^{m} f_i\,\log_2 f_i$$

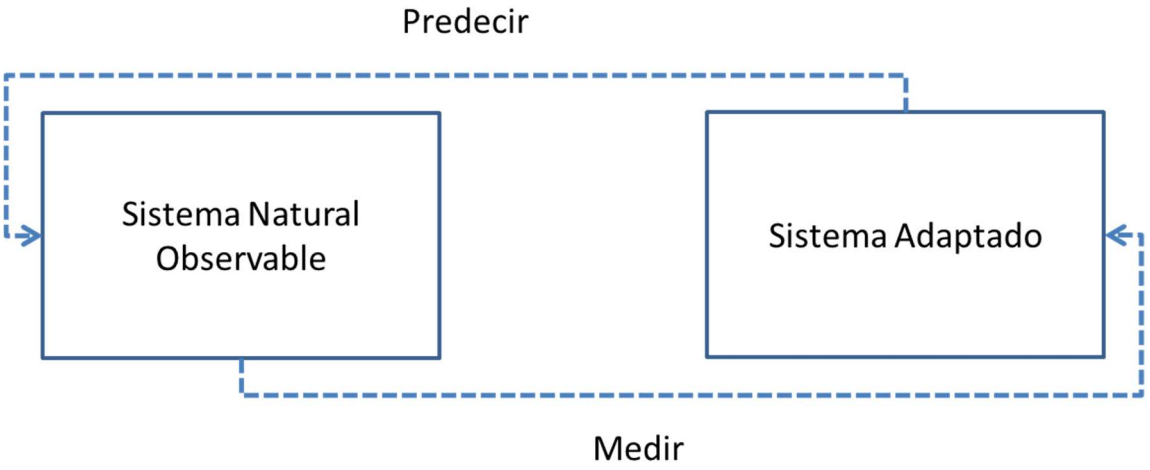
Es equiparable a un promedio ponderado de la diferencia logarítmica entre las probabilidades de dos distribuciones.

Los árboles de decisión son una herramienta clave para la toma de decisiones en la industria, sin embargo, no están exentos de desventajas, a continuación listamos un cuadro comparativo de ventajas y desventajas de la técnica:

Ventajas	**Desventajas**
• Fácil de entender e interpretar • Requiere muy poca preparación de la información • Es un módulo de "caja blanca" • Trabaja con datos numéricos y categóricos • Evaluable por test estadísticos • Robusto • Puede aplicarse a grandes conjuntos de datos	• Puede sobre ajustar (se soluciona con poda) • Están sesgados a favor de los atributos con más niveles • Aproximan las superficies por medio de "Escalones" lineales

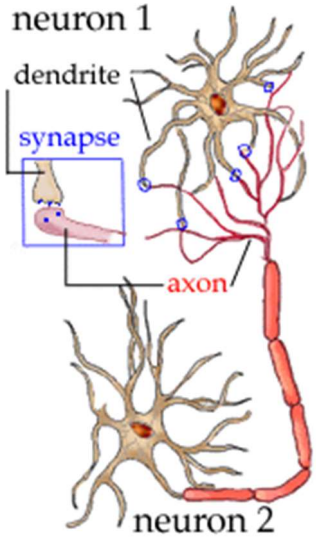
Redes neuronales artificiales

Las redes neuronales imitan a los sistemas naturales que aprenden a responder apropiadamente a los cambios del ambiente.



Neuronas

Las neuronas son la unidad fundamental de la cognición (capacidad de un ser vivo de procesar información a través de la percepción)



Las entradas que ingresan a través de las dendritas son ponderadas por sinapsis adaptables antes de ser sumadas, si la suma es mayor que cierto umbral adaptable, la neurona envía una señal mediante su axón a otras neuronas.

Las redes neuronales artificiales (ANN) fueron desarrolladas originalmente por investigadores que buscaban imitar la neurofisiología del cerebro humano. Al combinar muchos elementos simples de cómputo (neuronas) en un sistema altamente interconectado intentaron reproducir fenómenos complejos como la inteligencia. En años recientes se han incorporado métodos estadísticos y análisis numérico a las redes. Si bien todavía se debate si las ANN son verdaderamente inteligentes, es un hecho que son un modelo matemático muy útil.

Las ANN son una clase de regresión no lineal muy flexible. Al detectar relaciones complejas no lineales en los datos, las redes pueden ayudar a predecir en problemas del mundo real.

Las ANN son de particular utilidad en problemas donde:

- No se conoce fórmula matemática que relacione las entradas con las salidas (principio de caja negra)
- La predicción es más importante que la explicación
- Existen grandes cantidades de datos para entrenar

Arquitectura

La neurona de McCulloch-Pitts se define mediante la ecuación:

$$\hat{y} = f\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$$

La función de paso f convierte convierte cada neurona en un clasificador lineal.

La fuerza de conexión entre las neuronas i y j se ajusta de acuerdo con la ecuación:

$$\Delta w_{ij} = \eta\, \hat{y}_i x_j$$

Que es llamada regla de Hebb donde $0 \leq \eta \leq 1$, η es conocida como tasa de aprendizaje que escala la cantidad de ajuste del peso, x_j es la activación de la neurona entrante de otra neurona y $\hat{y}_i$ es la salida recibida de la neurona.

La regla de Hebb es inestable, Widrow y Hoff propusieron una variante estable dada por:

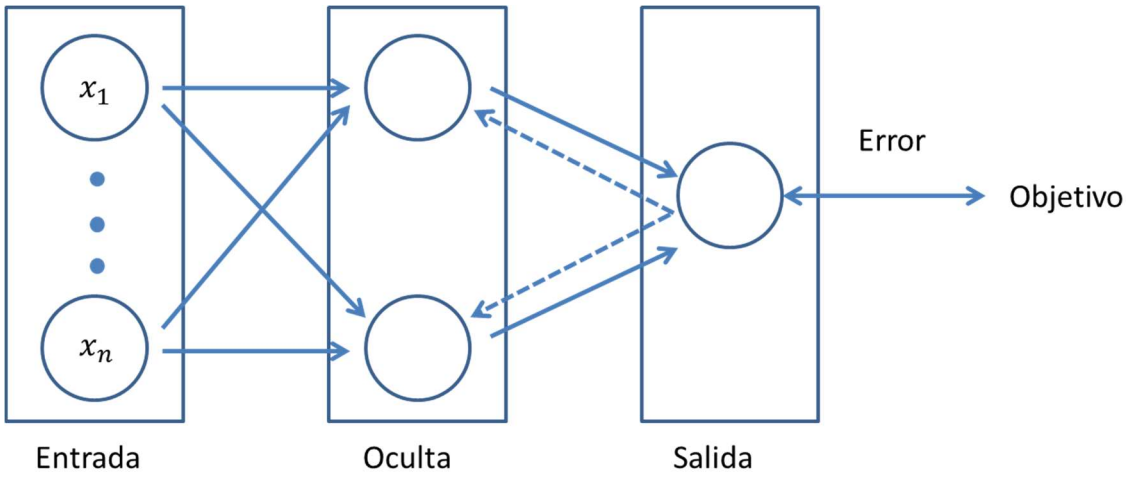
$$\Delta w_{ij} = \eta\, (y_i - \hat{y}_i) x_j$$

Conocida como regla delta.

Uno de los algoritmos más utilizados en el entrenamiento de redes neuronales es el algoritmo Backpropagation, que consiste en los siguientes pasos:

1. Propagar la activación de entradas hacia adelante a través de la red y calcular el error de salida

2. Propagar el error hacia atrás ajustando los pesos, si no converge repetir 1



Tipos de redes neuronales

Si bien no estudiaremos todas, proporcionaremos un listado de las técnicas más importantes que involucran Redes Neurales:

1. Red neuronal hacia adelante
    I. Perceptrón capa única
    II. Perceptrón multicapa
    III. ADALINE
    IV. Radial Basis Function
    V. Mapa auto organizado (Kohonen)

2. Red recurrente
    I. Red recurrente simple
    II. Red de hopfield

3. Redes neuronales estocásticas
    I. Máquina de Boltzmann

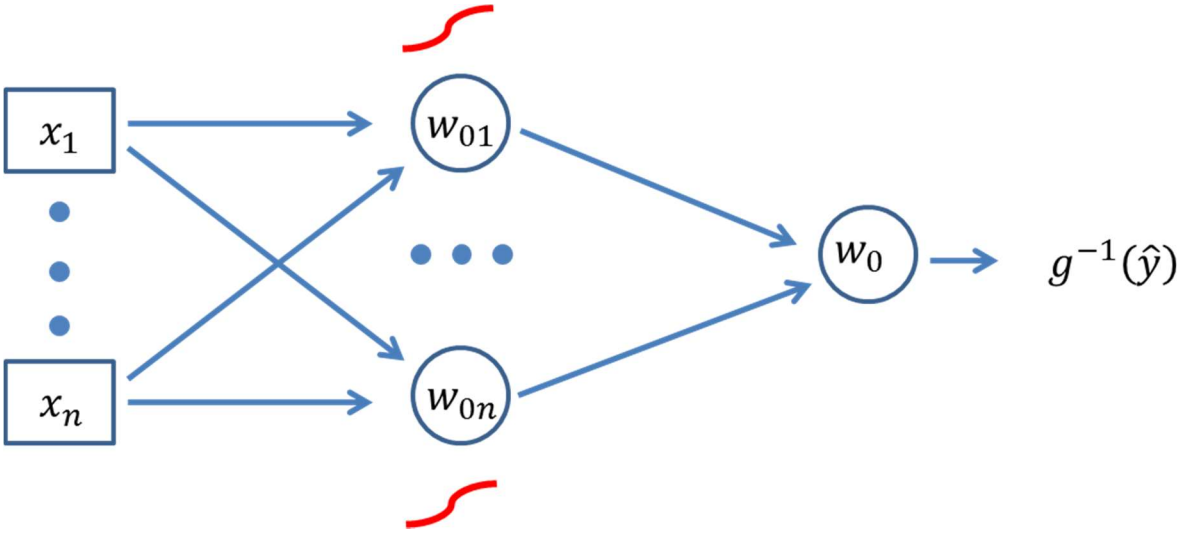
4. Redes neuronales modulares

I. Comité de máquinas
II. Red neuronal asociada

5. Otros
   I. Redes entrenadas al instante
   II. Red neuronal de picos
   III. Red neuronal dinámica
   IV. Red neural en cascada
   V. Redes neuro difusas

La más utilizada de las redes neuronales en la práctica es el perceptrón multicapa (MLP). MLP utiliza funciones de activación sigmoidales (tangente hiperbólica, función logística). El número de parámetros en un MLP con una capa oculta de h neuronas y k entradas está dada por

$$h(k+1)+h+1=h(k+2)+1$$



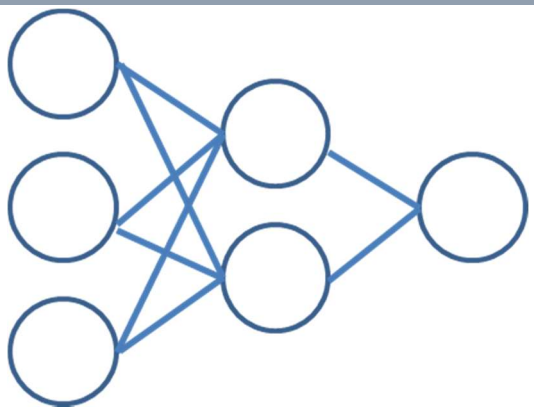
Donde

$$g^{-1}(\hat{y})=w_0+\sum_{i=1}^{h}w_i y_i\left(w_{0i}+\sum_{j=1}^{d}w_{ij}x_j\right)$$

El MLP más simple consiste de tres capas:

- Entrada: contiene una neurona por cada variable de entrada y no cuenta con pesos
- Oculta: con neuronas intermedias que (por defecto) realizan una transformación sigmoidal de las activaciones sumadas y ponderadas que llegan a cada neurona
- Salida: forma y combina los valores no lineales de las activaciones de la capa oculta

Por ejemplo, veamos el siguiente perceptrón:



$$g_0^{-1}\big(E(y)\big) = w_0 + w_1 H_1 + w_2 H_2$$

$$H_1 = \tanh(w_{01} + w_{11}x_1 + w_{21}x_2 + w_{31}x_3)$$

$$H_2 = \tanh(w_{02} + w_{12}x_1 + w_{22}x_2 + w_{32}x_3)$$

Si bien la función de activación más utilizada es la logística, existen alternativas:

- Softmax: $\frac{e^{\eta_i}}{1+\sum_{j=1}^{r-1} e^{\eta_i}}$; $\eta_i = \ln\left[\frac{P(y=i|x)}{P(y=niv \quad de\ referencia|x)}\right]$
- Tangente hiperbólica: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Elliot: $\frac{x}{1+|x|}$
- Arcotangente: $\frac{2}{\pi}\arctan(x)$
- Exponencial: e^x
- Identidad: x

Análisis Discriminante

El análisis discriminante (en lo sucesivo AD) es una técnica estadística que permite asignar o clasificar nuevos individuos dentro de grupos previamente definidos. AD intenta realizar la misma tarea que la regresión lineal múltiple al predecir una salida, sin embargo, la regresión está limitada a los casos donde la variable dependiente es tal que la combinación de sus predictoras, a través de la ecuación de regresión, producirá los valores numéricos estimados de la media mediante combinaciones ponderadas de los valores de las variables independientes. EL AD nos permitirá adentrarnos dentro de variables de interés no continuas como podrían ser: Intención de voto, estatus de empleo, intención de compra de un producto, si un cliente es sujeto de crédito o no, marca de preferencia, y en general, cualquier otra variable categórica que sea de utilidad para el investigador.

El AD entra dentro de la minería de datos en la categoría de modelación supervisada, debido a que es necesaria una variable objetivo que nos permita clasificar. Su objetivo fundamental es producir una regla o esquema de clasificación tal que nos permita predecir la población a la que es más probable pertenecer una nueva observación (donde se conocen previamente dichas poblaciones).

Para poder llevar a cabo esta labor, se requiere contar con una tabla de n observaciones en las que se han medido p variables de intervalo (explicativas) además de una variable dependiente categórica de cardinalidad mayor o igual que 2 que represente el grupo al que pertenece cada observación.

El AD se presenta desde dos enfoques:

- Por obtención de funciones discriminantes: Se construyen ecuaciones similares a las de regresión lineal múltiple
- Canónico: Emplea técnicas de correlación canónica y componentes principales

Clasificación para dos grupos.

Estudiaremos en primera instancia el AD para clasificar en dos categorías, este problema fue resuelto por Ronald Fisher (1924).

Sean k variables explicativas para clasificar individuos separados en 2 y sólo 2 categorías. Se propone una función discriminatoria dada por:

$$D = \sum_{i=1}^{k} u_i X_i$$

Conocida como ***función discriminante de Fisher.***

La intención es obtener los coeficientes de ponderación u_i por medio de n observaciones, de esta manera, podemos expresar una función discriminante para cada una de ellas como sigue:

$$D_j = \sum_{i=1}^{k} u_i X_{ij} \quad j = 1,2,\dots,n$$

D_j representa la puntuación discriminante correspondiente a la $j-ésima$ observación, si estandarizamos las variables con respecto a la desviación estándar, podemos expresar la relación anterior de forma matricial:

$$\begin{bmatrix} D_1 \\ \dots \\ D_n \end{bmatrix} = \begin{bmatrix} X_{11} & \dots & X_{k1} \\ \dots & \dots & \dots \\ X_{1n} & \dots & X_{kn} \end{bmatrix} \begin{bmatrix} u_1 \\ \dots \\ u_k \end{bmatrix}$$

Que puede escribirse en notación compacta como:

$$\vec{d} = X\,\vec{u}$$

Para expresar la variabilidad de la función discriminante recurrimos a la suma de cuadrados de las variables discriminantes estandarizadas:

$$\vec{d}^T\vec{d} = \vec{u}^T\,X^T X\,\vec{u}$$

Donde $X^T X$ es una matriz simétrica que contiene la suma de cuadrados total de las variables explicativas, en consecuencia, podrá descomponerse en la suma de las matrices entre grupos F e intragrupos W:

$$X^T X = T = F + W$$

Y por tanto:

$$\vec{d}^T\vec{d} = \vec{u}^T\,X^T X\,\vec{u} = \vec{u}^T\,T\,\vec{u} = \vec{u}^T\,F\,\vec{u} + \vec{u}^T\,W\,\vec{u}$$

De la expresión anterior podemos calcular las matrices T, F y W a partir de los datos muestrales, los coeficientes u_i son las incógnitas a determinar. Para poder determinar dichas incógnitas, Fisher planteó maximizar la razón de la variabilidad entre grupos respecto de la variabilidad intragrupos. Si nos detenemos a pensar, dicha razón provocará que las distribuciones estén lo más separadas entre sí (queremos que cada categoría de la variable nominal este lo más lejos posible de su contraparte) y al mismo tiempo dentro de cada una exista la menos dispersión posible (grupos compactos). Matemáticamente, expresamos esto de la siguiente manera:

$$\lambda = \frac{\vec{u}^T\,F\,\vec{u}}{\vec{u}^T\,W\,\vec{u}}$$

Como lo mencionamos con anterioridad, debemos maximizar el valor de λ , por tanto:

$$\frac{\partial \lambda}{\partial \vec{u}} = \frac{\vec{u}^T W \vec{u} \frac{\partial}{\partial \vec{u}}(\vec{u}^T F \vec{u}) - \vec{u}^T F \vec{u} \frac{\partial}{\partial \vec{u}}(\vec{u}^T W \vec{u})}{(\vec{u}^T W \vec{u})^2}$$

$$= \frac{\vec{u}^T W \vec{u} \ [2 F \vec{u}] - \vec{u}^T F \vec{u}[2 W \vec{u}]}{(\vec{u}^T W \vec{u})^2} = 0 \Rightarrow 2 F \vec{u}(\vec{u}^T W \vec{u}) - 2 W \vec{u}(\vec{u}^T F \vec{u}) = 0$$

$$\Rightarrow \frac{2 F \vec{u}}{2 W \vec{u}} = \frac{\vec{u}^T F \vec{u}}{\vec{u}^T W \vec{u}} = \lambda \Rightarrow F \vec{u} = \lambda W \vec{u} \Rightarrow W^{-1} F \vec{u} = \lambda \vec{u} \Rightarrow (W^{-1} F - \lambda I)\vec{u} = 0$$

Esto nos indica que para obtener los ejes discriminantes habremos de calcular los valores propios de la matriz $W^{-1} F$ y al ser λ el ratio a maximizar, escogeremos siempre los valores propios en orden descendente así como su vector propio asociado $\vec{u}$.

Definiremos ahora las puntuaciones discriminantes como:

$$D = \sum_{i=1}^{k} u_i X_i$$

Lo anterior equivale a proyectar cada vector del espacio de variables de k dimensiones sobre el eje discriminante.

Por otro lado, debemos definir los centroides de cada categoría dentro de la variable nominal que corresponden a los estadígrafos que resumirán la información de cada una de ellas, a saber:

$$\bar{x}_I = \begin{bmatrix} \bar{X}_{1,I} \\ \bar{X}_{2,I} \\ \dots \\ \bar{X}_{k,I} \end{bmatrix} \qquad \bar{x}_{II} = \begin{bmatrix} \bar{X}_{1,II} \\ \bar{X}_{2,II} \\ \dots \\ \bar{X}_{k,II} \end{bmatrix}$$

Con estos vectores procederemos a calificar el punto de corte discriminante C dado por:

$$C = (\bar{D}_I + \bar{D}_{II})/2$$

Dónde:

$$\bar{D}_I = \sum_{i=1}^{k} u_i \bar{X}_{i,I} \quad , \ \bar{D}_{II} = \sum_{i=1}^{k} u_i \bar{X}_{i,II}$$

Así, si $D_i < C$, clasificamos al individuo i en el grupo I y si $D_i > C$, clasificaremos al individuo i en el grupo II

Clasificación canónica.

El segundo enfoque se basa en la obtención de las componentes principales dentro de nuestro conjunto de datos. Sabemos que las componentes principales son una combinación lineal de las

variables originales que explican la mayor variabilidad posible y además son ortogonales entre sí. Si tomamos en cuenta que la primera componente C_1 está asociada al mayor valor propio de la matriz de datos, entonces las componentes sucesivas tendrán asociados los valores propios en orden descendente y, en consecuencia, podríamos utilizar dichas componentes como ejes discriminantes, donde la función discriminante equivaldría a la ecuación del componente principal de acuerdo a la cardinalidad de la variable dependiente.

Máquinas de soporte vectorial.

Una máquina de soporte vectorial, es una técnica de modelación supervisada utilizada en problemas tanto de clasificación como de regresión. Consiste en construir un hiperplano(o un conjunto de hiperplanos) en una mayor dimensión, que incluso pudiese ser infinita, para separar patrones linealmente. Considere la muestra de entrenamiento $\{(\vec{x}_i, d_i)\}_{i=1}^{N}$ donde $\vec{x}_i$ son los patrones de entrada para la i-ésima observación y d_i es la correspondiente salida deseada. Asumimos que el patrón (clase/categoría) representado por el subconjunto $d_i = +1$ y que el patrón representado por el subconjunto $d_i = -1$ son "linealmente separables", así, la ecuación de una superficie de decisión en la forma de un hiperplano que separe ambos patrones será:

$$\vec{w}^T \vec{x} + b = 0$$

Donde $\vec{x}$ es un vector de entrada, $\vec{w}$ es un vector de pesos ajustables y b es un sesgo. Esto nos lleva a escribir que:

$$\vec{w}^T \vec{x} + b \geq 0 \text{ para } d_i = +1$$
$$\vec{w}^T \vec{x} + b < 0 \text{ para } d_i = -1$$

Dado un vector de pesos $\vec{w}$ y un sesgo b, la separación entre el hiperplano propuesto y el punto en los datos más cercano es llamada el margen de separación y se denota por la letra ρ. El objetivo de la máquina de soporte vectorial es encontrar el hiperplano particular tal que el margen de separación ρ se maximice. Bajo esta condición, la superficie de decisión es llamada *hiperplano óptimo*. Sean $\vec{w}^*$ y b^* los valores óptimos del vector de pesos y el sesgo respectivamente, entonces, el hiperplano óptimo, representando una superficie lineal multidimensional de decisión en el espacio de entrada, está definida por:

$$\vec{w}^* \vec{x} + b^* = 0$$

Si definimos $g(\vec{x}) = \vec{w}^* \vec{x} + b^*$ obtenemos una medida algebraica de la distancia desde $\vec{x}$ hasta el hiperplano óptimo, para ello, expresemos $\vec{x}$ como sigue:

$$\vec{x} = \vec{x}_p + r \frac{\vec{w}^*}{\|\vec{w}^*\|}$$

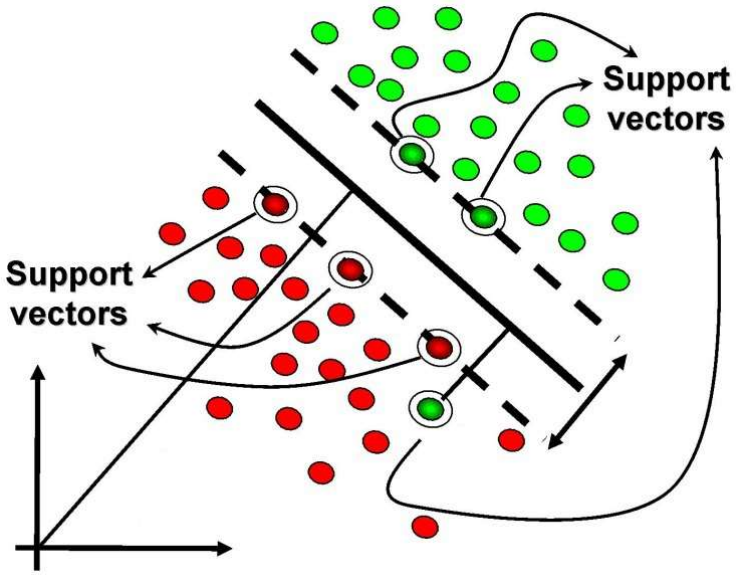
Donde $\vec{x}_p$ es la proyección ortogonal de $\vec{x}$ sobre el hiperplano óptimo y r es la distancia algebraica deseada. Donde r es positiva si $\vec{x}$ se encuentra en el lado positivo del hiperplano y negativa en caso de encontrarse del lado negativo. Dado que $g\left(\vec{x}_p\right)=0$, tenemos:

$$g\left(\vec{x}\right)=\vec{w}^{*T}\vec{x}+b^{*}=r\left\|\vec{w}^{*}\right\|\rightarrow r=\frac{g\left(\vec{x}\right)}{\left\|\vec{w}^{*}\right\|}$$

El trabajo entonces, será encontrar los parámetros $\vec{w}^{*}$ y b^{*} para el hiperplano óptimo dado el conjunto de entrenamiento $\mathrm{T}=\left\{\left(\vec{x}_i,d_i\right)\right\}$ y dado que:

$$\vec{w}^{*}\vec{x}_i+b^{*}\geq 1 \text{ para } d_i=+1$$
$$\vec{w}^{*}\vec{x}_i+b^{*}\leq 1 \text{ para } d_i=-1$$

Los puntos particulares $\left(\vec{x}_i,d_i\right)$ para los cuales las expresiones anteriores se convierten en igualdad son llamados *vectores soporte* y de ahí la derivación del nombre máquina de soporte vectorial. En términos conceptuales, tales vectores son los puntos que se encuentran más cerca de la superficie de decisión y son, en consecuencia, los más difíciles de clasificar. La siguiente figura ilustra este hecho:



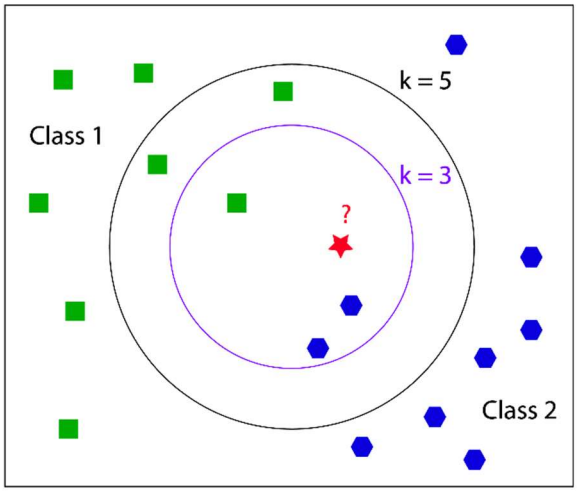
K-Vecinos más cercanos

La idea detrás de este método se basa en la estimación a partir de un número fijo k de observaciones lo más cercanas al punto estudiado. Para problemas de clasificación, la categoría elegida dependerá de la mayoría de voto de los k vecinos involucrados. La cercanía será computada basada en un criterio de distancia, en consecuencia, la estandarización de los datos será mandatoria.

Algunas de sus propiedades más importantes son:

- No paramétrico: las variables no tienen ningún supuesto de distribución
- Algoritmo perezoso: No construye ningún modelo, espera a que se requisite la predicción para el cómputo.
- Es local: Asume que la clase dependerá únicamente de los vecinos más cercanos, no construirá un modelo global.
- Es simple: solamente requiere cómputos de distancias simples.

Un diagrama simple de la tarea se muestra a continuación:



El valor óptimo de k puede ser localizado mediante técnicas de selección de híper parámetros.

Gradiente estocástico descendiente

El gradiente estocástico descendiente implementa un algoritmo de aprendizaje que soporta variase funciones de pérdida y penalización incluyendo máquinas vector soporte y regresión logística. Matemáticamente, se plantea como sigue:

Dado un conjunto de muestras de entrenamiento $(\vec{x}_i, y_i)$ con $\vec{x}_i \in \mathbb{R}^n$ y $y_i = \{-1, 1\}$ se busca una función de clasificación $f(\vec{x}) = \vec{w}^t \vec{x} + b$ con parámetros $\vec{w} \in \mathbb{R}^m$ y sesgo $b \in \mathbb{R}$. Para hacer predicciones, simplemente se revisa el signo de f. La elección más común para ajustar los parámetros del modelo es mediante la minimización del error de entrenamiento regularizado dado por:

$$E(\vec{w}, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\vec{x}_i)) + \alpha \, R(\vec{w})$$

Donde L es una función de error que mide el ajuste del modelo y R es el término de regularización (penalización) que penaliza la complejidad del modelo. $\alpha > 0$ es un híper parámetro.

L se escoge de entre las siguientes opciones:

- Visagra: Máquinas vector soporte
- Logística: Regresión logística
- Mínimos cuadrados: Regresión de cresta
- Insensible a épsilon: Regresión de vector soporte

En cuanto a la elección de R:

- Normalización L2: $R(\vec{w}) = \frac{1}{2} \sum_{i=1}^{n} w_i^2$
- Normalización L1: $R(\vec{w}) = \frac{1}{2} \sum_{i=1}^{n} |w_i|$
- Red elástica: $R(\vec{w}) = \frac{\rho}{2} \sum_{i=1}^{n} w_i^2 + (1 - \rho) \sum_{i=1}^{n} |w_i|$ (combinación convexa de L1 y L2

Para encontrar el óptimo del problema planteado se utiliza el algoritmo del descenso por gradiente estocástico, el cual aproxima el gradiente verdadero de $E(\vec{w}, b)$ considerando un solo ejemplo de entrenamiento a la vez. El algoritmo itera sobre los ejemplos de entrenamiento y por cada uno actualiza los parámetros de acuerdo a la siguiente regla:

$$\vec{w} \leftarrow \vec{w} - \eta \left(\alpha \frac{\partial R(\vec{w})}{\partial \vec{w}} + \frac{\partial L(\vec{w}^T + b, y_i)}{\partial \vec{w}} \right)$$

Donde η es la tasa de aprendizaje que controla el tamaño de paso en el espacio paramétrico. El sesgo b se actualiza similarmente pero sin incluir la regularización. La tasa de aprendizaje puede ser tanto constante como gradualmente decreciente, en problemas de clasificación, la tasa de aprendizaje óptima para el tiempo t estará dada por:

$$\eta^{(t)} = \frac{1}{\alpha\left(t_0 + t\right)}$$

Donde t_0 es un valor heurístico inicial.

Clasificador ingenuo de Bayes

Es un clasificador basado en el Teorema de Bayes que nos permite conocer la probabilidad de pertenencia de una observación en el conjunto de entrenamiento dadas sus características asumiendo independencia entre las mismas. El clasificador modela la probabilidad de pertenencia a una clase C basado en las predictoras independientes X_i, como sigue:

$$P\left(C \middle| X_1, X_2, \ldots, X_n\right)$$

Si aplicamos el Teorema de Bayes, tenemos:

$$P\left(C \middle| X_1, X_2, \ldots, X_n\right) = \frac{P(C)P\left(X_1, X_2, \ldots, X_n \middle| C\right)}{P\left(X_1, X_2, \ldots, X_n\right)}$$

De donde sabemos que la probabilidad posterior es proporcional a la verosimilitud presentada en el numerador, en consecuencia:

$$P\left(C \middle| X_1, X_2, \ldots, X_n\right) \propto P(C)\prod_{i=1}^{n} P\left(X_i \middle| C\right)$$

De donde:

$$\hat{C} = \arg\max_C P(C)\prod_{i=1}^{n} P\left(X_i \middle| C\right)$$

Los diferentes clasificadores de Bayes difieren principalmente por la suposiciones con respecto a la distribución de $P\left(X_i \middle| C\right)$.

Los clasificadores ingenuos de Bayes son conocidos por clasificar adecuadamente en la práctica, sin embargo, debe considerarse la desventaja de que son malos estimadores probabilísticos.

Los clasificadores ingenuos de Bayes más comunes se presentan a continuación:

- Gaussiano: Se asume que las variables predictoras son normales:

$$P\left(X_i|C\right)=\frac{1}{\sqrt{2\pi\sigma_C{}^2}}\exp\left(-\frac{\left(X_i-\mu_C\right)^2}{2\sigma_C{}^2}\right)$$

Los parámetros μ_C y σ_C se estimarán mediante máxima verosimilitud.

- Multinomial: Muy utilizado en clasificación de texto, se utiliza para datos multinomiales. La distribución se parametriza por los vectores $\theta_C=\left(\theta_{C_1},\ldots,\theta_{C_N}\right)$ para cada clase C, donde n es el número de predictores y θ_{C_i} es la probabilidad $P\left(X_i|C\right)$ de que el predictor i aparezca en una muestra perteneciente a la clase C. Los parámetros θ_C son estimados mediante una versión suavizada de máxima verosimilitud, es decir, por conteo de frecuencias relativas:

$$\hat{\theta}_{C_i}=\frac{N_{C_i}+\alpha}{N_C+\alpha n}$$

Donde $N_{C_i}=\sum_{X\in T}X_i$ es el número de veces que el predictor i aparece en una muestra de la clase C en el conjunto de entrenamiento T y $N_C=\sum_{i=1}^{|T|}N_{C_i}$ es el conteo de todos los predictores para la clase C. El suavizado $\alpha\geq 0$ cuenta para predictores no presentes en los ejemplos de entrenamiento y previene probabilidades cero. Cuando $\alpha=1$ es llamado suavizamiento laplaciano, mientras que si $\alpha<1$ es llamado suavizamiento de Lidstone.

- Bernoulli: Se utiliza en casos donde los datos están distribuidos de acuerdo a distribuciones Bernoulli multivariantes, es decir, múltiples predictores pero cada uno se asume binario. La regla de decisión para Bayes ingenuo Bernoulli está basada en:

$$P\left(X_i|C\right)=P\left(i|C\right)X_i+\left(1-P\left(i|C\right)\right)\left(1-X_i\right)$$

Que difiere de la regla de Bayes Ingenuo multinomial en el hecho de que penaliza explícitamente la no ocurrencia del predictor i que es un indicador para la clase C.

Ensambles

Basados en el principio: "Dos cabezas piensan mejor que una", o como es nuestro caso, dos modelos clasificarán mejor juntos que por separado, los ensambles de modelos combinan las predicciones de un conjunto de algoritmos con la intención de mejorar la generalización/robustez que tendría un estimador individual. Los ensambles se dividen en dos grupos fundamentalmente:

- Ensambles por promedio (Averaging)
- Ensambles por impulso (Boosting)

Respectivamente, los primeros basan la estimación en el promedio de las estimaciones individuales, mientras que los segundos son construidos de forma secuencial buscando reducir el sesgo del estimador combinado, en resumen, se construyen varios modelos débiles para producir un ensamble poderoso.

Los ensambles más populares son:

- Bosque Aleatorio: Son una combinación de árboles de decisión predictores tal que cada árbol depende de los valores de un vector aleatorio muestreado independientemente y con la misma distribución para todos los árboles en el bosque. El error de generalización de un bosque de árboles clasificadores depende de la potencia de los árboles individuales en el bosque y la correlación entre ellos. Formalmente, un bosque aleatorio es un clasificador consistente en una colección de árboles clasificadores $\left\{h\left(\vec{x},\Theta_k\right),k=1,...\right\}$ donde $\left\{\Theta_k\right\}$ son independientes e idénticamente distribuidos vectores aleatorios y cada árbol emite un voto unitario para la clase más popular para la entrada $\vec{x}$.

- ADABoost: Propuesto en 1997 por Freund y Schapire, la base del algoritmo es la construcción de predictores débiles (ligeramente mejores que el azar) en versiones de los datos repetidamente modificadas. Las predicciones de cada uno de ellos es combinada mediante una mayoría ponderada de voto. Fue el primer algoritmo práctico de impulso. El algoritmo se define de la siguiente manera:

  Dado $\left(\vec{x}_i,y_i\right)$ donde $\vec{x}_i$ son vectores de entrada y $y_i\in\left\{-1,+1\right\}$.

  Inicializar $D_1\left(i\right)=\frac{1}{m}$ para $i=1,\ldots,m$

  Para $t=1,\ldots,T$:

  - Entrenar un aprendiz débil usando la distribución D_t
  - Obtener una hipótesis débil $h_t:X\to\left\{-1,+1\right\}$
  - Apuntar: seleccionar h_t con error bajo ponderado $\varepsilon_t=\Pr_{i\sim D_t}\left[h_t\left(\vec{x}_i\right)\neq y_i\right]$
  - Elegir $\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

- Actualizar, para $i = 1, \ldots, m$: $D_{t+1}(i) = \dfrac{D_t(i)\exp\left(-\alpha_t y_i h_t(\vec{x}_i)\right)}{Z_t}$ donde Z_t es un factor de normalización (elegido tal que D_{t+1} sea una distribución)
- Extraer hipótesis final: $H(\vec{x}) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(\vec{x})\right)$

- XGBoost: Es un sistema de ensambles de impulso escalables y punta a punta, es una abreviatura de Extreme Gradient Boosting. El algoritmo ha demostrado ser muy potente en problemas relacionados a la ciencia de datos. Fue propuesto por Tianqi Chen (2016). Está construido para permitir cómputo en paralelo y computo distribuido. XGBoost se basa en ensambles de impulso de árboles gradiente. Formalmente, sea $\hat{y}_i^{(t)}$ la predicción de la i-ésima instancia en la iteración t, se necesita agregar f_t para minimizar la siguiente función objetivo:

$$L^{(t)} = \sum_{t=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\vec{x}_i)\right) + \Omega(f_t)$$

Esto significa agregar ambiciosamente la f_t que mejora más el modelo. Se puede usar una aproximación de segundo orden para optimizar rápidamente la función objetivo:

$$L^{(t)} \simeq \sum_{t=1}^{n}\left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(\vec{x}_i) + \frac{1}{2} h_i f_t^2(\vec{x}_i)\right] + \Omega(f_t)$$

Donde $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ y $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ son los estadígrafos gradientes de primer y segundo orden de la función de pérdida, si se quitan los términos constantes, se obtiene la siguiente función objetivo simplificada para el paso t :

$$\bar{L}^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(\vec{x}_i) + \frac{1}{2} h_i f_t^2(\vec{x}_i)\right] + \Omega(f_t)$$