MODELOS DE MACHINE LEARNING DE VARIABLE EXPLICADA BINARIA

ÍNDICE:

[**INTRODUCCIÓN** 2](#_Toc82970555)

[**1.- PLANTEAMIENTO DEL CASO GENERAL** 3](#_Toc82970556)

[**2.- CONCEPTOS BÁSICOS DE ANÁLISIS DISCRIMINANTE** 5](#_Toc82970557)

[**3.- MODELOS LINEALES GENERALIZADOS: EL CASO DE LA REGRESIÓN LOGÍSTICA** 7](#_Toc82970558)

[3.1.- Medición de la fiabilidad de un modelo de regresión logística: R-Cuadrado de MacFadden versus Índice de Gini 11](#_Toc82970559)

[3.2.- Regularización en regresión logística 12](#_Toc82970560)

[**4.- MEDIDAS DE FIABILIDAD Y COMPARACIÓN DE MODELOS DE VARIABLE BINARIA** 14](#_Toc82970561)

[4.1.- Índice de Gini 14](#_Toc82970562)

[4.2.- Curvas Lift y Lift del primer decil 15](#_Toc82970563)

[4.3.- Curva ROC 16](#_Toc82970564)

[4.4.- Otras medidas en modelos de variable binaria 18](#_Toc82970565)

[**5.- DECISION TREES** 19](#_Toc82970566)

[**6.- MODELOS CON NOCIÓN DE DISTANCIAS: EL KNN** 21](#_Toc82970567)

[**7.- TEORÍA Y PRÁCTICA DE LOS SVM** 23](#_Toc82970568)

[7.1.- Supuestos de los SVMs 24](#_Toc82970569)

[7.2.- Tipos de SVMs 25](#_Toc82970570)

[7.3.- Planteamiento y resolución con SVMs de separaciones tipo XOR 33](#_Toc82970571)

[7.4.- Extensiones y generalizaciones de los SVMs 35](#_Toc82970572)

[**CONCLUSIONES FINALES** 37](#_Toc82970573)

[**BIBLIOGRAFÍA** 37](#_Toc82970574)

# **INTRODUCCIÓN**

En esta sesión se hace una introducción general a las técnicas de análisis discriminante donde la variable respuesta, por lo general, va a ser una variable discreta (en general binaria) y de lo que se trata es de asociar probabilidades a cada uno de sus valores. La atención se centra en gran parte, en los algoritmos de clasificación en general sin embargo, se cree necesario traer las principales técnicas de discriminación poblacional que ya hace casi 100 años comenzó Fisher.

Antes de comenzar con la temática del tema se hace una breve referencia a una conocida superficie topológica conocida como Botella de Klein, cuya construcción geométrica y el hacer uso de espacios multidimensionales, sin duda va a ayudar a entender mejor el concepto de Kernel que se aplica en los SVM, por tanto se describe la Botella de Klein como una superficie geométrica, no orientable, que sólo puede representarse adecuadamente en , mientras que en se tiene una aproximación que parece que en cierto punto la figura se corta y se atraviesa así misma

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Diagrama

Descripción generada automáticamenteImagen que contiene persona, azul, hombre, camisa

Descripción generada automáticamente

**Figura 0.1 Representación geométrica de una Botella de Klein**

Actualmente hay incluso empresas que “venden” Botellas de Klein (aunque siempre sean imperfectas)

Mano sosteniendo una botella de plástico

Descripción generada automáticamente con confianza media

**Figura 0.2 Botella de Klein en**

En una dimensión superior, dicha botella que tiene unas ecuaciones tal como siguen:



No “se atraviesa así misma”. En topología se pone como ejemplo de figuras en el espacio que no pueden representarse correctamente en y para ello se recurre a una dimensión superior, a para su adecuada representación, aunque dicha dimensión escape de la intuición humana. Esta superficie en cierto modo fue importante porque fue la primera superficie de dimensión mayor que 2 no orientable y que aunque se puede embeder en sin embargo se necesita una dimensión superior para su adecuada representación.

En cierto modo se observa que para explicar relaciones en sencillas, se va a recurrir a dimensiones superiores y a espacios “extraños” donde resolver el problema y devolverlo resuelto a nuestro “sencillo mundo” de 3 dimensiones.

# **1.- PLANTEAMIENTO DEL CASO GENERAL**

A continuación, se introduce una nueva familia de modelos donde cambia un poco el objetivo a predecir. En vez de tratar de ajustar los datos al comportamiento de una variable de carácter continuo y por tanto, medir el “parecido medio” resultado de un modelo, con el que existe en la realidad, en los modelos que se plantean aquí lo que se va a tratar de medir es la ocurrencia o no de un evento.

Sin embargo, tratar de crear modelos que directamente midan la ocurrencia vs no ocurrencia haciendo uso de funciones de minimización de energía como las que se utilizan en los modelos de variable continua, se observa que no ofrecen resultados adecuados y pueden generar interpretaciones confusas, por lo que se cambia aquí el objetivo a predecir, donde lo que se predice no es la ocurrencia propiamente dicha sino la probabilidad de que tenga lugar un hecho frente a otro. Así pues, si *Y* es la variable explicada, de lo que se trata de predecir va a ser la siguiente probabilidad condicional:

Bajo esta formulación, algunos elementos, vistos en la sesión anterior se mantienen y por supuesto otros cambian. Entre los que se mantiene serán las propiamente estructurales respecto a datos, por lo que se mantiene la existencia de variables *variables explicativas* y de *variable explicada* o *target* y los datos también van a venir dispuestos en una tabla análoga a la vista en la sesión anterior:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ido\_id | var1 | … | varp | Target |
| individuo1 | v11 | … | v1p | t1 |
| … | … | … | … | … |
| individuom | vm1 | … | vmp | tm |

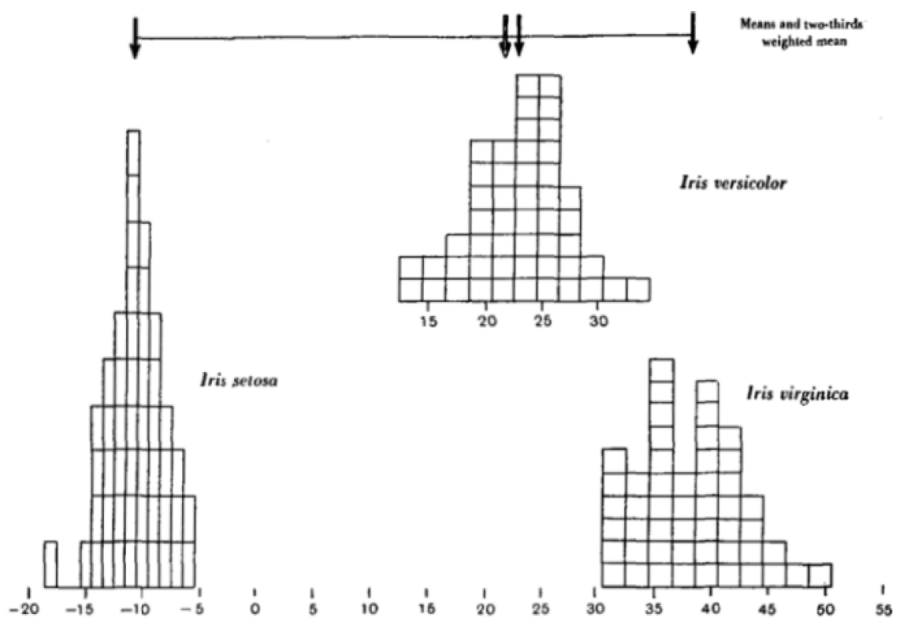
Donde sólo cambiarían los valores t1, … tm que sólo pueden ser iguales a 0 o a 1. De nuevo la variable denominada aquí *ido\_id* obviamente no intervendrá en el modelo y solamente está añadida por motivos de claridad y por lo general va a tener un valor distinto por individuo de la muestra, de modo que cuando llegue un nuevo individuo, se le podría dar el valor de identificación individuom+1 del cual se va a conocer a priori los valores de las variables explicativas y por tanto lo que realmente se quiere es medir:

En lo que sigue se analizarán distintas variantes, algunas incluso bastante complejas de posibles funciones *F* a emplear sin embargo para estimar la fiabilidad de un modelo que trate de medir probabilidades de ocurrencias, a pesar de que muchos de los modelos que se van a tratar aquí tienen su análogo con un modelo de la sesión anterior, va a suceder que internamente cambian muchos elementos que no se observan a simple vista, ya que en apariencia, a nivel de “mera preparación de datos”, ambos tipos de modelos resultan totalmente análogos.

# **2.- CONCEPTOS BÁSICOS DE ANÁLISIS DISCRIMINANTE**

El origen del análisis discriminante cabe situarlo en el estudio de taxonomía vegetal que en 1936 emprendió uno de los estadísticos más importantes del siglo XX, R. A. Fisher en el trabajo *Fisher R. A. (1936)*.

En este estudio hace uso de los conocidos datos de *Flores de Iris* que fueron ya recopilados por el Dr. E. Andersonun año antes. De lo que se trata en el citado trabajo de Fisher no era más que en base a 4 medidas realizadas sobre unapoblación de 50 plantas de cada una de 3 clases (de las 4 clases) de *Flores de Iris*: *setosa*, *versicolor* y*virgínica* mostrar en primer lugar encontrar cómo se podían separar las 2 primeras especies en base a lasmencionadas medidas de y en segundo lugar y teniendo en cuenta que *virgínica* poseía una composicióncromosómica intermedia se quería analizar si el efecto genético (conjunto de dicha especie respecto a lasanteriores) daba lugar a características intermedias acorde a la Teoría de los Alopoliploides separables delas dos especies iniciales. Tal y como se muestra en la siguiente imagen tomada del artículo, la conclusiónrespecto al segundo hecho resulta clara:



**Figura 1 Estudio preliminar de Fischer del análisis discriminante en las Flores de Iris**

Fuente:

En el citado trabajo de Fisher, la deducción de una fórmula general para el análisis discriminante no resulta muy directa por lo que su desarrollo resulta más instructivo si se sigue la línea que se desarrolla en *James G.; WittenD.; Hastie T. Tibshirani R. (2013).* Aquí se presenta directamente la fórmula utiliza en la actualidad los algoritmos que hacen uso de esta técnica:

Es decir, dado un vector, se quiere saber si está en la clase 1 o en la clase 2. Para ello:

* Se requiere conocer las proporciones de las poblaciones de vectores a priori, junto con las medias y las varianzas de cada una de las poblaciones
* Se hacen las operaciones “matriciales” anteriores y al final se obtiene un número
* En función de si ese número es positivo o negativo, se dirá que el vector a clasificar pertenecerá a una población o a otra

Ejemplo 01 LDA con python

En el programa01.py se hace una comparación muy sencilla entre los resultados que se obtienen de un *lda* frente a un *pca* Hay que entender que la mejor actuación del *lda* frente al *pca* en este caso no está por una superioridad o no del algoritmo, sino porque en el *lda* se ha introducido el carácter supervisado de la técnica cuando se construyen las funciones discriminantes

|  |  |
| --- | --- |
|  |  |

A diferencia de la mayoría de los modelos que se van a ver a continuación, con el análisis discriminante no se están dando unas probabilidades de ocurrencia de un hecho frente a otro, sino que simplemente se definen zonas mediante hiper-planos donde pueden o no caer las observaciones.

Para entender qué era lo que realmente hizo Fisher se va a partir de unos datos simulados y se va a aplicar la anterior fórmula a éstos, la ventaja de partir con datos simulados a diferencia de los anteriores es que se va a conocer al máximo detalle el proceso generador de éstos y por tanto se puede analizar, al menos teóricamente, si lo planteado tiene o no sentido ya que se cumplen las hipótesis subyacentes como son la normalidad de las observaciones, hipótesis clave si se quiere tener las garantías teóricas de funcionamiento del *análisis discriminante* sobre el que hay 2 tipos, el lineal (que era el que recomendaba Fisher por aquella época por no existir ordenadores y que simplifica un poco más la anterior fórmula que es la que corresponde al *análisis discriminante cuadrático*). Actualmente los paquetes tipo *R* como *Python* ofrecen soluciones muy optimizadas para ambos análisis.

Ejemplo 02 Revisión paso a paso de las técnicas de *análisis discriminante*

En el programa02.R se aplica y se comparan tanto las funciones de R, como las realizadas por fuera en base a las anteriores fórmulas sobre una base de datos simulados, distinguiéndose el momento de generación y el momento

# **3.- MODELOS LINEALES GENERALIZADOS: EL CASO DE LA REGRESIÓN LOGÍSTICA**

Para poder utilizar la regresión lineal, es necesario que la variable respuesta sea continua y cumpla las hipótesis estándar del modelo lineal, pero si por ejemplo la variable es binaria, el modelo a ajustar cambia y se cabe por ejemplo ajustar un modelo de *regresión logística*. Así pues, la *regresión logística* forma parte de los llamados modelos lineales generalizados y trata de estimar la probabilidad de ocurrencia de un evento binario (éxito/fracaso, cara/cruz) en función de una serie de variables predictoras. La *Regresión Logística* es una de las técnicas estadístico‐inferenciales más empleadas en la producción científica contemporánea y a día de hoy, indiscutiblemente sigue siendo el más utilizado tanto en banca como en seguros (aunque se aplica en ámbitos muy diversos también). Surge en la década del 60, su generalización dependía de la solución que se diera al problema de la estimación de los coeficientes. El *algoritmo de Walker‐Duncan* para la obtención de los estimadores de máxima verosimilitud vino a solucionar en parte este problema, pero era de naturaleza tal que el uso de computadoras era imprescindible.

Un elemento clave es que en los casos de variable binaria, si se hace uso del método de mínimos cuadrados para ajustar la siguiente expresión:

Se van a cometer los siguientes errores:

* Los valores predichos pueden estar fuera del intervalo (0; 1)
* La teoría de la *regresión lineal* en lo referente a intervalos de confianza y test están referidos a datos que provienen de una distribución *normal*, lo que no es cierto con datos de tipo *binario*

Así pues, la *regresión logística* va a ser idear para responder a preguntas tales como: ¿Se puede predecir con antelación si un cliente que solicita un préstamo a un banco va a ser un cliente moroso? ¿Se puede predecir si una empresa va a entrar en bancarrota? ¿Se puede predecir de antemano que un paciente corra riesgo de un infarto? Por tanto, la formulación de este modelo va a responder a:

Fórmula que matemáticamente tiene unas propiedades bastante agradables, de hecho, es factible considerar lo que se denomina *odds ratios* que responden a la siguiente formulación:

Es decir, el *odds ratio* responde a la tasa de ocurrencia de la probabilidad de evento sobre no evento y que responde a la expresión “la probabilidad de que tenga lugar un hecho es de *n* contra *m*” dando como resultado el siguiente valor:

Es aquí de donde deriva el nombre del modelo de regresión logística, ya que al aplicar la función *logit* (o logaritmo) al *or* se obtiene una expresión lineal:

Claramente, la razón por la que la regresión logística pertenece a la familia de los modelos es porque hace uso del denominado *link logístico* que es aquel adaptado para cuando los datos de la variable respuesta siguen una distribución de carácter *binomial* o en un caso más particular, de tipo *bernoulli*. Así pues, dicho link es una función no lineal definida en el intervalo (0; 1) y que responde a la siguiente expresión:

Claramente el anterior link proyecta el intervalo (0; 1) sobre toda la recta real y la gráfica de la anterior función (con *g*(1/2) = 0) responde a:



Source: <https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica>

Sin embargo, “el mundo de los *modelos lineales generalizados*” no acaba en la regresión logística, sino que en función del supuesto de distribución que se realice sobre la variable explicada, se hace uso de una u otra función link, siendo las principales las siguientes expresiones:

Texto, Carta

Descripción generada automáticamente

Por tanto, desde este punto de modelos tipo *GLM* la *regresión logística* queda como:

En definitiva, lo que se tiene es que la probabilidad en este caso es una función no lineal de las variables a diferencia de lo que ocurre con la *regresión lineal*, esto hace entre otras cosas, que la estimación de sus parámetros ya no se podrá hacer de un modo manual, sino que se va a hacer uso de un algoritmo de tipo iterativo como el de *Newton-Raphson* (descrito en 1671 por Sir Isaac Newton) y que responde a la siguiente formulación general (en una variable):

Convergiendo por tanto hacia aquel punto donde se hace 0 la función *f* y que permite generalización hacia el caso multivariante a través del concepto de Jacobiano:

Ejemplo 03 Encontrar la solución del siguiente sistema de ecuaciones:

Para ello se crea la función 2 variables siguiente:

El Jacobiano resulta ser igual a:

Y posteriormente se seguiría el cálculo (a partir claro está de vectores columnas fácilmente programables en R o Python, lo cual no se hará)

En regresión logística, la función de coste a minimizar viene de aplicar el método de máxima verosimilitud donde al existir un supuesto de *probabilidad de bernoulli* sobre las observaciones va a resultar ser la siguiente función de *p-variables* (no se confunda esta *p* con el parámetro de probabilidad *p,* que se están denotando igual, pero significan distintas cosas):

donde aquí se tiene que:

variable binaria que puede valer 1 o 0 para cada observación

La maximización de la anterior función genera un sistema de ecuaciones que es tratable por el método *Newton-Raphson*:

Diagrama, Esquemático

Descripción generada automáticamente

En este sistema se tiene que *N* = *m* y *n* = *p* y por tanto estamos ante *p* + 1 ecuaciones con *p* + 1 incógnitas y donde cabe demostrar que existe bajo condiciones muy generales una única solución

Bajo estas premisas la interpretación de los coeficientes se realiza bajo el siguiente resultado:

Teorema

En un modelo de *regresión logística*, los coeficientes *a*i representan el cambio en el *logit* resultante de aumentar una unidad en la i-ésima variable *x*i

\_\_\_\_\_

Por tanto, la interpretación de los coeficientes debe realizarse en términos de *odds ratio* de hecho cuando se compara el riesgo entre dos vectores de datos:

e

Cabe definir el *risk odds ratio* de frente a como:

Por tanto, el papel de los coeficientes de la regresión lineal, lo harán la exponencial de éstos que en cierto modo cuando dichos coeficientes sean positivos, indicarán que el término del producto anterior será mayor que 1 y por tanto incidirá en que la probabilidad de evento sea mayor, mientras que cuando el coeficiente sea menor que 0 sucederá todo lo contrario

## 3.1.- Medición de la fiabilidad de un modelo de regresión logística: R-Cuadrado de MacFadden versus Índice de Gini

Una fuente continua de confusión entre los iniciados en DS (y no tan iniciados), es que la medición de la bondad de un modelo de *regresión logística* no responde al parámetro habitual de *r-cuadrado* tal y como se conoce en la *regresión lineal*. La razón fundamental radica a que dado que lo que se mide con la variable target es una variable de respuesta binaria tipo 0-1 y por otro el output del modelo va a ser un valor de probabilidad (ajustada a la media muestral de cómo se hayan entrenado los datos), la medida ya no va a ser acertada. En este sentido, una medida habitual en los paquetes de cálculo estadístico es el conocido como *pseudo-r-cuadrado de McFadden* que surgió a mediados de los 70 según se puede ver en McFadden, D. (1974). Este indicador se denomina *pseudo-r-cuadrado* porque realmente no es un *r-cuadrado* en el sentido de la *regresión lineal*, mientras en este último, lo que se compara es la variabilidad entre lo explicado por el modelo y la realidad, la expresión del indicador de McFadden responde a la siguiente expresión:

donde:

Valor del logaritmo de la verosimilitud del modelo formado por únicamente la constante

Valor del logaritmo de la verosimilitud del modelo con todas las variables consideradas

En este sentido lo que sucederá será:

1. Si el modelo no mejora la previsión básica se tendrá que será similar a y por tanto el cociente bajo el cuadrado será próximo a 1 y por tanto sería próximo a 0
2. En caso de tenerse un modelo perfecto lo que va a sucede es que será próximo a 0 ya que por construcción se tiene que:

Y se tendría un juego de parámetros tal que en los casos en que los datos tengan un valor se tendría que y por tanto el primer sumando se anularía, y en caso contrario se llegaría a que y por tanto y el logaritmo del segundo sumando también se anularía, teniéndose por tanto que

Este coeficiente, aunque fue muy popular su cálculo en el pasado, ya que se aprovechaba el cálculo previo de la función de máxima verosimilitud cuando se estimaba el modelo, pero el problema que tiene es que no es generalizable para otros tipos de modelos como los de Machine Learning en general, donde no interviene función de máxima verosimilitud asociada, además ofrece interpretaciones que cuanto menos resultan dudosas ¿Cuándo es bueno un *MacFadden*? ¿Cuándo vale más o menos de 0.5?

Ejemplo 04 Análisis del *pseudo r-cuadrado*

En el programa04.py se describen 3 situaciones. En las que se está ante un modelo con claros buenos y malos valores *MacFadden* y cuando se está con un valor qué podría considerarse “regular”, además se aprovecha para analizar cómo es posible sacar este valor a partir de la salida que genera *statsmodels*

|  |  |  |
| --- | --- | --- |
| **MacFadden = 0.8336** | **MacFadden = 0.3767** | **MacFadden = 0.0037** |

## 3.2.- Regularización en regresión logística

Mientras el modelo de regresión logística pertenece al Machine Learning tradicional y se podría desplegar toda la teoría al respecto, sobre todo en lo que concierne a la aplicación de test y medida de bondad de ajuste, sin embargo lo que se pretende en este curso, es tener una visión más amplia del resto de modelos relacionados o no con éste, así pues, una familia de modelos que nacen, de modo análogo a como sucedió con la *regresión lineal* son los modelos de *regresión logística regularizados* y de modo análogo lo que se toma es la función de costes que proviene del método de máxima verosimilitud (multiplicada por (-1)) y se hace una alteración de tipo *l1* o *l2* obteniéndose por tanto análogas conclusiones en cuanto al tratamiento de la multicolinealidad y a la posibilidad de tratar casos donde hay más variables que ecuaciones.

Así pues, las funciones de coste de las que se partirán siguen la siguiente formulación:

con y y

A la que se le aplicará la siguiente modificación según sea lasso o ridge respectivamente:

De nuevo cabe aplicar un algoritmo tipo *Newton-Raphson* para minimizar, pero lo usual actualmente es hacer uso del gradiente descendente que resulta más generalizable y aplicable a las distintas situaciones.

En estas extensiones de los modelos de *regresión logística*, también cabe hablar de la combinación *elastic-net* y de composiciones bastante imaginativas que ofrecen más versatilidad y flexibilidad a los modelos de regresión logística.

Al igual que ocurría con la *regresión lineal*, el efecto de las regularizaciones en los parámetros será bastante variado

Ejemplo 05 Construcción de un modelo de regresión logística al completo con *regularización l1*

En el programa05.py se genera un modelo de regresión logística bajo *regularización* *l1* y sin ésta y se observa cómo podría dar solución a problemas relacionados con “multicolinealidad perfecta”

Texto

Descripción generada automáticamente

# **4.- MEDIDAS DE FIABILIDAD Y COMPARACIÓN DE MODELOS DE VARIABLE BINARIA**

Antes de seguir adelante con el resto de los modelos es conveniente analizar las medidas de fiabilidad que se usan en unos y en otros de hecho, ya se ha observado con el caso de la *regresión lineal* y la *logística* que se cambian tanto los conceptos como en el modo de interpretarlos, es lo que se ha visto en 3.1 con el caso del *r-cuadrado* “habitual” frente al concepto que proponía *MacFadden*.

Las medidas habituales que se utilizan para poder hacer comparaciones entre este tipo de modelos tienen una fuerte componente geométrica y suelen estar basadas en distintas curvas que tienen que ver con cómo se distribuyen los *eventos* frente a los *no eventos*.

A continuación, se repasan algunas de las más habituales.

## 4.1.- Índice de Gini

Es quizás la medida más sencilla de explicar y en muchos casos la que se cree que se entiende cuando se habla de la *curva ROC* que se detalla más adelante.

Este índice trata de detectar si el modelo es capaz de colocar en las posiciones de mayor probabilidad los valores de tipo 1. Si esto es así cabe plantear el siguiente gráfico en el plano, cuya área puede ser medida y es lo que se denominaría *Índice de Gini* muy utilizado como indicador para medir la desigualdad en economía:

* Primero se ordena a la población de mayor a menor probabilidad asignada por el modelo
* Se considera unos puntos de corte que pueden ser *percentiles*, *deciles,* etc
* Se calcula la concentración de evento acumulada a nivel de cada punto de corte
* En el último punto de corte, la proporción de evento acumulada será del 100% y se tendrá al 100% de la población

Bajo esta construcción se tiene que si por ejemplo la tasa de evento de una población es del 4% y se hace una subdivisión en deciles y en el primer decil ya se tiene el 100% de ese 4%, entonces bajo esta subdivisión, el *índice de Gini* tendrá un área igual a: , mientras que en una división de percentiles sería igual a: , prácticamente la unidad.

La interpretación de este indicador es bastante directa como se observa y las áreas de negocio suelen entenderlo bastante bien. Además, es un indicador utilizable tanto para las regresiones logísticas como para todo el resto de modelos, que se desarrollan a partir de ahora, tanto en esta sesión, como en la relativa de modelos de ensemble.

## 4.2.- Curvas Lift y Lift del primer decil

Son curvas donde conceptualmente se manejan los mismos conceptos que en la *curva Gini* de hecho, las curvas *lift* pueden ser acumuladas y no acumuladas siendo las primeras una imagen especular de la citada *curva Gini*.

Su construcción parte al igual que en la *curva Gini* de que se haya previamente ordenado a la población y que se haya hecho una segmentación en partes iguales bien mediante *deciles*, *quintiles*, *percentiles*, etc.

Una vez hechos los grupos de modo equi-poblacionales, el siguiente paso consistiría en establecer el número de *eventos* que habría en cada división ante una distribución uniforme de éstos, ante el caso de distribución uniforme si existen *g* grupos y hay un número igual de *e* eventos, lo que sucedería, si el modelo no tuviera ningún efecto sobre la distribución de los eventos es que en cada grupo habría un total de *e/g* eventos (este número puede ser decimal pero a efectos prácticos no supone un problema en el cálculo de la *lift*).

Así pues, lo que debe suceder en aquellos grupos donde la tasa esperada de evento teórico sea mayor es que la cantidad de eventos que cabe denominarse sea muy grande y supere al factor *e/g* y lo contrario debería pasar en aquellos tramos de baja probabilidad. Por tanto, si en el eje horizontal se disponen los grupos creados, el dato del eje vertical es la *lift no acumulada* cuando se cumple que:

con y

En función por tanto del número de grupos *g* que se forme el valor de la *lift no acumulada* podrá ir de un número positivo de a lo sumo igual a *g* hasta un número que podrá llegar a ser 0.

En modelos bien construidos es de esperar que esta curva sea decreciente y sin picos y si se ordenan los elementos de una población de mayor a menor probabilidad teórica asignada por el modelo, lo que debería ocurrir es que fuese:

* Una curva siempre decreciente, siendo ideal que el decrecimiento sea estricto:
* Que es decir, a mayor “elevación” real de la probabilidad en los grupos de probabilidad teórica alta, mucho más predictivo será el modelo a plantear
* Que es decir, a menor “elevación” real de la probabilidad en los grupos de probabilidad teórica baja, mucho más predictivo será el modelo a plantear

Sin embargo, se observará que por lo general esta curva no va a tener un comportamiento como el anterior y es de esperar la existencia de “picos” que en función de su número e intensidad ayudarán a elegir o no un determinado modelo.

Otro conceto relacionado con el anterior es la *curva* *lift acumulada*. Bajo esta curva lo que se trata es captar la elevación que se va acumulando conforme se avanza en los grupos equi-poblacionales establecidos, así pues, utilizando la misma notación y de nuevo ordenando a la población de mayor a menor probabilidad de evento, las alturas que se muestran en el eje Y serían ahora:

Nótese que ahora la curva si va a ser siempre decreciente, de modo que su menor valor va a ser igual a 1 y el máximo a alcanzar será igual a *g*.

En este sentido cobra especial interés el valor relativo en el primer grupo donde se verifica:

Cuando las agrupaciones equi-poblacionales se dividen en deciles, al valor anterior se le denomina *lift en el primer decil* y en general, cuando las tasas de evento son relativamente pequeñas, para considerar que en las pruebas de validación un modelo sea aceptable se va a requerir un valor superior a 2.5 o que en el primer decil se tenga 2.5 veces más oportunidad de encontrar un evento que ante una distribución uniforme sin existencia de modelo.

Ejemplo 06 *Índice de Gini* y *curva lift* en modelos de regresión logística

En el siguiente ejemplo se hace uso de los datos del ejemplo anterior, donde se observa algo que pocas veces tiene lugar en la realidad como es un ajuste tan elevado en términos de *Curva de Gini* y de *Lift Acumulada*, además se observa la concordancia de resultados cuando se calcula la *lift del primer decil*

|  |  |
| --- | --- |
|  |  |
|  | |

## 4.3.- Curva ROC

Esta curva recibe su nombre del acrónimo *Receiver Operating Characteristic* y lo que trata de medir es la variación del concepto de *sensibilidad* frente al de *especificidad* bajo un sistema de clasificación binaria según varía el umbral de clasificación propiamente dicho.

Definición: Sensibilidad de un clasificador binario

Es la razón de Verdaderos Positivos o *VPR* y se define como:

\_\_\_\_\_

Definición: Especificidad de un clasificador binario

Es la razón de Verdaderos Negativos *NPR* y se define como:

\_\_\_\_\_

La anterior notación procede de la siguiente representación en tabla de contingencia 2x2:

|  |  |
| --- | --- |
| VP = Verdaderos Positivos | FP = Falsos Positivos |
| FN = Falsos Negativos | VN = Verdaderos Negativos |

Generalmente lo que se hace es que se establecen distintos puntos de corte a distintos niveles de probabilidad (entre 0 y 1) y se representa en el eje Y la *sensibilidad* y en el eje X, 1 – *especificidad* o lo que es lo mismo, la *razón de falsos positivos* o *FPR* tratando por tanto esta curva de expresar los intercambios entre los aciertos o *verdaderos positivos* y los errores o *falsos positivos* (que claramente tienen un coste)

Ejemplo 07 Reconstrucción en excel de escenarios de la curva ROC

En el fichero programa07\_curva\_roc.xlsx se explica con un ejemplo sencillo cómo es el proceso de reconstrucción paso a paso de la *curva ROC* para distintas ordenaciones (bajo distintos modelos) de 50 elementos donde hay 10 eventos

Imagen que contiene interior, llenado, entero, computadora

Descripción generada automáticamente

## 4.4.- Otras medidas en modelos de variable binaria

En general, tener un buen valor en alguna de las medidas anteriores, va a implicar que dicho “buen valor” va a tenerse en el resto de las medidas de hecho, existe una relación donde se cumple que:

Por lo que, en cierto modo, aunque ambos índices provienen de distintas interpretaciones y áreas analíticas, en el fondo, gracias a la fórmula anterior resultan ser lo mismo, así pues, un modelo que sobre un *test* tenga una aceptable área *ROC* o *AUC* igual a 0.70, tendría por consiguiente un *Índice de Gini* igual 0.40 y esta relación es 1 a 1.

Claramente no debe confundirse ninguno de los 2 anteriores términos con el concepto de *accuracity*, que se haya obtenido un *AUC* de 0.70 no quiere decir que la tasa *VP* + *VN* será igual a 0.70; también basadas en tablas 2x2 y no comparables con la *AUC* estarían las siguientes medidas que cubren determinados matices cuando se mide la *bondad* de un determinado modelo como resultan ser las siguientes:

* *Precision:* Resulta ser la tasa de acierto entre los que el modelo indica que son datos con evento

* *Recall:* Resulta ser la tasa de acierto entre los que en la realidad resultan ser datos con evento (sin que lo diga el modelo)
* *F1-score:* resulta ser una media armónica entre los conceptos de *Precision* y *Recall*

Para finalizar esta sesión se destaca una última medida que suele utilizarse bastante en los modelos de riesgo de crédito fundamentalmente como es el *estadístico de Kolmogorov-Smirnov*, este estadístico resulta ser una prueba no paramétrica y lo que se quiere es probar si 2 muestras independientes provienen de la misma distribución, lo que se calcula por tanto es la máxima diferencia absoluta entre 2 distribuciones empíricas que en el caso de modelos de variable binaria serán la *distribución del evento* frente a la *distribución del no evento*, por lo que su expresión matemática se construye del siguiente modo:

* Se consideran las funciones empíricas de cada una de las distribuciones de 1 y 0 que se expresaría como:

donde: Tipología 0 – 1 del elemento muestral

Elemento i perteneciente a los elementos tipo *j*

* Posteriormente se calcula la “*distancia supremo*“ entre ambas distribuciones dada por la fórmula siguiente formulación

Como se podría ver si se amplia el ejemplo anterior este estadístico también viene implementado en librería *scikitplot* de Python mediante a función *plot\_ks\_statistic()* que dibuja las 2 distribuciones acumuladas y calcula el estadístico en sí:

Imagen que contiene Gráfico

Descripción generada automáticamente

# **5.- DECISION TREES**

Los modelos de árboles son modelos de tipo no lineal que lo que hacen es particionar el espacio de entrada en regiones tipo hiper-cubos. Mientras que en los denominados *Decision Trees*, la variable a predecir es de tipo binario, en los *Regression Trees* la variable es continua e interesará por tanto en cada una de las regiones *J* constituidas cuál es el valor medio de la variable, de modo que, es de esperar que las diferencias sean lo mayores posibles. A diferencia de los *Regression Trees* en los *Decision Trees* va a regir la siguiente función que va a tratar de minimizarse como por ejemplo la siguientes:

-Ganancia de información o reducción de la entropía acorde a la fórmula de entropía de Shannon siguiente:

-Uso del índice de Impureza de Gini que sigue la siguiente expresión:

Este índice tiene que ver con la reducción de la varianza de la cantidad de información ya que cuando hay *J* clases a dividir, la expresión anterior se convertiría en:

Como sucedía en los *regression trees*, al ser modelos sencillos, el analista puede tener todo el control sobre crecimiento de estos *decision tree*, de modo que tratará por ejemplo de que los *nodos* y/o *hojas* finales resultantes tengan suficiente representatividad y significatividad. Siendo una práctica habitual dejar crecer el modelo de *árbol* suficientemente y después aplicar la técnica conocida como *poda*. La necesidad pues de las técnicas de *poda* surgen porque en principio es posible construir muchísimas regiones sin embargo, en estos casos extremos, se está bajo lo que se conoce como *overfitting* y lo que sucederá es que el poder de generalización será muy reducido ante nuevos datos, por otro lado, el número de regiones a constituir y de posibilidades es inmenso y por tanto hay que considerar alguna estrategia de construcción de dichas regiones, las cuáles suelen partir de ciertos criterios heurísticos tal y como se hacía con los métodos de selección de variables en las regresiones.

Una técnica habitual, que viene implementada en las librerías de construcción de *decision trees* (y también en las de *regression trees*) es la conocida como *minimal cost-compleity pruning*, según esta técnica, lo que se trata es de minimizar la siguiente medida:

donde:

Número de hojas en el modelo *T*

Ratio de *misclassification* de las hojas (o la ratio de observaciones clasificadas erróneamente en dichas hojas del árbol)

Cantidad mayor o igual a 0 conocido como parámetro de complejidad

Esta técnica (que también tiene su análogo en *regression trees*), está debidamente implementada *sklearn* y estima el parámetro *a* (denominado habitualmente como *alpha*) a partir de un proceso de *crossvalidation* que en términos de uso ya está debidamente implementado

Ejemplo 08 Decision tree con poda mediante la técnica *Cost Complexity Pruning*

En el ejemplo programa08.py se observa cómo mediante una adecuada poda, se pasa de un árbol complejo, a otro más simple sin apenas haber perdido *accuracity* en la *matriz de confusión* del dataset *test* y evitando en gran medida el habitual problema del *overfitting*

|  |  |
| --- | --- |
| Árbol Inicial | Árbol Final |

# **6.- MODELOS CON NOCIÓN DE DISTANCIAS: EL KNN**

La versión binaria del *knn* va a resultar en todos los sentidos totalmente análoga a la de tipo *continuo*, en el momento de asociar una clase a un punto en el espacio, se tiene también que calcular las distancias a los puntos más cercanos para asociar, en función de la mayoría que puntos que se encuentre en la zona, una u otra clase.

Como ocurría en su versión continua, la elección del número *k* de vecinos próximos resulta esencial en este algoritmo dando lugar a distintas fronteras que pueden ir desde aquellas muy ajustadas que generarían un claro overfitting cuando *k* es bajo, hasta aquellas que generarían fronteras parecidas a hiperplanos y por tanto este modelo convergería a uno de tipo de *análisis discriminante cuadrático* de los comentados en la sesión 2.

Mapa, Gráfico de dispersión

Descripción generada automáticamente

**Figura 1: k-NN con k = 1 y k = 100 respectivamente**

**Fuente: Introduction to Statistical Learning**

Entre las ventajas de usar un *knn* cabe citar su simplicidad, efectividad y fácil implementación, pero tiene algunos inconvenientes como:

* Que en realidad no se crea un modelo sino que lo que se genera es un conjunto de instancias sobre las que se van a tomar distancias, por lo que tendrá un rápido proceso de entrenamiento, pero un muy lento proceso de predicción, que generarían problemas en los pasos a producción de este tipo de modelos
* El modelo adolece del conocido problema *curse of dimensionality*

Finalmente, el número de parámetros va a ser tratado en general como un hiperparámetro que habrá de ser determinado con alguna técnica aceptable como puede ser la *cross validation*.

Gráfico, Gráfico radial

Descripción generada automáticamente

**Figura 2: Elección de la clase de un punto ante distintos entornos**

**Fuente:** [**https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn**](https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn)

Ejemplo 9 Ajuste de un *knn* binario con datos sencillos pero con distintas categorías

En el código programa09.py se aplica el método *knn* para un dataset con 3 clases. Dado que tiene pocas observaciones y columnas, se pretende determinar el *k* óptimo. Cuando se aplica un bucle, evaluando sobre el conjunto test, se observa la siguiente salida:

Texto

Descripción generada automáticamente

El resultado es muy inestable, aquí el óptimo aparece en *k* = 3, pero dependiendo de la semilla de aleatorización, todo lo anterior puede cambiar. A diferencia de lo que se pone en el ejemplo procedente de *datacamp* que han actuado sin hacer uso de *random\_state*, el resultado, por falta de datos resulta ser muy inestable (recuérdese que este modelo está afectado por “the curse of dimensionality”), aunque dos ideas básicas caben ser aplicadas para corregir lo anterior:

* Normalizar las observaciones de las variables
* Realizar un proceso de reducción de la dimensionalidad mediante un *pca* por ejemplo

# **7.- TEORÍA Y PRÁCTICA DE LOS SVM**

Es una técnica desarrollada por Vladimir Vapnik (1936 - ) en los laboratorios AT&T durante los años 90 para el reconocimiento de escritura manual y que fue ya recogida en (Cortes, C.; Vapnik, V. 1995). El objetivo de la técnica es la clasificación de datos, por tanto, la *Variable Explicada* será obligatoriamente discreta (desde una variable binaria, hasta una con más valores discretos y finitos de tipo nominal) Grosso modo, esta técnica lo que consigue es separar en regiones linealmente disjuntas, bien en el espacio de partida, bien en un espacio de mayor dimensión, un conjunto de observaciones determinado, por esta razón se clasifican dentro de la categoría de los *Clasificadores Lineales*

Es decir, bajo un enfoque general, si se tiene un conjunto de observaciones :

Con esta técnica, lo que se hace es usar una función para “embeder” (cuando es necesario) dicho espacio en uno de dimensión superior donde en este caso, sí se podrían separar los puntos mediante un hiperplano de dicho espacio de elevada dimensión

Es pues en el espacio donde se lleva a cabo la separación en sí por hiperplanos (piénsese de nuevo en la botella de Klein, hasta dimensión 3, las cosas parecen “muy normales” pero a partir de la cuarta dimensión se puede conseguir separar cosas que en principio en dimensión 1, 2 y 3 no se conseguiría)

|  |  |
| --- | --- |
| **Figura 3 Puntos separable y no separable**  **Fuente: Elaboración propia** |  |

## 7.1.- Supuestos de los SVMs

La variable dependiente es de tipo nominal finito (lo ideal es que esta variable sea de tipo binario, existen extensiones no binarias, pero no son “directas”).

Se hacen uso de supuestos de optimización convexa (se describen un poco más abajo) poco restrictivos y que obedecen más a cómo se presenta matemáticamente el algoritmo a implementar y al tipo de separación o clasificación que se tiene finalmente con este tipo de modelos.

Los “kernel” a elegir deben cumplir las hipótesis básicas del Teorema de Merecer. Es decir, se tiene un “juego reducido” de posibilidades de elección de este elemento (más adelante se describe brevemente que significa en sí dicho elemento).

Así pues, una de las ventajas de esta técnica es que estos supuestos son muy poco restrictivos y lo mejor de todo es que no se imponen condiciones específicas y directas sobre el comportamiento en sí de los datos, por lo que para aplicar el modelo sin “problemas” teóricos, es más importante atenerse por tanto a una serie de condiciones meramente formales de éstos y de los elementos y/o parámetros que se va a ir a considerar en la variante SVM que se desee aplicar (variante que se elegirá a veces en función del problema a analizar, a veces en función de la experiencia del investigador, a veces mediante prueba – error, etc).

Claramente lo anterior tiene algunas desventajas como se verá más adelante, pero si se está ante un problema de clasificación binario, y se tiene un conjunto de características a asociar a la salida de la variable dependiente, la aplicación de la técnica en sí directamente deriva una salida que, de tener sentido, en términos teóricos no daría lugar a problemas posteriores.

A las anteriores propiedades debe añadirse el supuesto de Espacio de Hilbert tanto en el Espacio de las observaciones, como en el denominado Espacio de las Características que es donde se embede lo anterior, por tanto, existe una función de distancias que hará que el algoritmo sea lento si el número de observaciones es muy elevado.

Por lo que, en resumen, los SVMs:

- Tienen un riguroso fundamento teórico.

- Alto poder discriminante en problemas de elevada dimensión.

- Admite su aplicabilidad en problemas no lineales.

Respecto a las principales desventajas de esta técnica cabría citar las siguientes:

- Son muy efectivos en clasificaciones binarias, siendo un poco más complicado cuando el objetivo de la clasificación es no binaria o continua.

- Al utilizarse "embedimientos" e hiperplanos de dimensión muy elevada, la interpretabilidad del modelo puede resultar complicada.

## 7.2.- Tipos de SVMs

De los modelos SVM existen varias variantes, aquí se van a analizar las 3 principales, cuyo mayor detalle pueden encontrarse en (Gareth, J.; Witten, D.; Hastie, T. y Tibshirani R. 2013) y que son las siguientes:

-Maximal Margin Classifiers

-Support Vector Classifiers

-Support Vector Machines

* MAXIMAL MARGIN CLASSIFIER

Este es el caso más sencillo de este tipo de clasificadores y por lo general no será muy habitual encontrarse en la realidad este tipo de ejemplos. Se tiene un conjunto de ejemplos linealmente separables como el que se muestra en la siguiente figura:

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

**Figura 4 Conjunto de Ejemplos de un problema linealmente separable**

**Fuente: Elaboración propia**

Notacionalmente dicho conjunto de ejemplos se denota por y se compone por los siguientes elementos:

donde

En este caso, se tiene para cada observación una medida de dos características en concreto y en este caso es representable todo el conjunto en el plano, sin embargo, y aunque no lo parezca, se está utilizando una representación propia de ya que está el elemento "color" que puede verse como +1 ó -1, o bien como +1 o 0, ... en todo caso una representación que se haría en un tercer eje z

¿Cuál es el objetivo del problema mostrado en la figura 4? El objetivo, es obtener una recta que separe en 2 regiones disjuntas las observaciones, es decir, conseguir separar los colores, así pues, se plantea la siguiente cuestión ¿Cuántas posibles rectas, a la vista de la figura 4 existirán?

En general, cuando se tiene un número elevado de características, y se está por tanto en espacios de dimensión 4 o más, se suele hablar del concepto de *Hiperplano de Separación*

Definición: Hiperplano de separación

Es aquel que cumple lo siguiente:

de modo que se cumple lo siguiente:

Si

Si

Una propiedad inmediata que debería tener una definición coherente de lo que debería ser un hiperplano de separación óptimo es que éste debería equidistar del ejemplo más cercano de cada clase, así pues, si se está ante un conjunto de ejemplos linealmente separable como los anteriormente vistos ¿Cuántos hiperplanos que cumplan con la anterior propiedad existirían? Llegados a este punto se cae en la cuenta de que se ha olvidado un elemento esencial de este relato que se plantea con la siguiente cuestión: ¿Tiene sentido calcular distancias de un punto a un plano? y si es así ¿Cómo se calcularía esto en general? En ambos casos se tiene respuesta intuitiva, pues se observa fácilmente en y en que los pasos a seguir son los siguientes:

(1) Se tiene una recta (o un plano) y un punto no contenido en la recta (o en el plano respectivamente)

(2) Se traza una recta que pase por el punto y que corte perpendicularmente a la recta (o al plano respectivamente)

(3) Se mide la longitud del segmento formado por el punto original y el punto de corte con la recta (o el plano respectivamente)

Matemáticamente y usando la notación anterior, la distancia puede expresarse tal como sigue para un elemento en concreto del conjunto de ejemplos:

En este tipo de problemas lo que se requiere es maximizar el valor de o lo que es lo mismo, se desea maximizar la siguiente expresión:

Es decir, maximizar es por tanto equivalente a disminuir la norma lo máximo posible. Por tanto, cabe plantear el siguiente problema primal de la búsqueda de un hiperplano óptimo, como aquel que:

Permite obtener los parámetros: a partir de encontrar:

s. a.

Sin entrar en detalle en la ecuación, lo que se pretende es conservar el razonamiento anteriormente expresado de búsqueda de un margen máximo, pero sujeto a la restricción de que los datos tienen que ser en todo caso linealmente separable. Así pues, ante la situación de la Figura 3.1, el resultado que daría es un modelo de Máximal Margin Classifier con una frontera de separación de las observaciones tal como sigue:

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

**Figura 5 Conjunto de Ejemplos de un problema linealmente separable, separados por un hiperplano de Margen Máximo**

**Fuente: Elaboración propia**

¿Cómo se plantea pues la búsqueda de un algoritmo general que cumpla la anterior misión en la práctica? Para ello hay que resolver el problema anterior, lo que en la práctica no se realiza “a mano”, sino que lo que se hace es que se demuestra que dado que el problema es de los de tipo “programación cuadrática” se puede deducir un problema equivalente denominado problema dual cuya dificultad (cantidad de cálculos a realizar) depende exclusivamente y de modo lineal del número de datos existente *n* lo que hace que la convergencia a la solución dual es muy rápida:

s. a. con

donde el operador producto escalar se define como:

Es por ello por lo que en los casos donde pueda aplicarse este algoritmo, se tiene el añadido que, aunque el número de observaciones sea muy elevado, una computadora como las que se fabrican actualmente permitirá obtener sin ningún problema la solución final de un modo bastante rápido y por tanto la expresión del hiperplano de separación sería la siguiente:

Ejemplo 10 Construcción de un MMC (Gareth, J.; Witten, D.; Hastie, T. y Tibshirani R. 2013; pp 368-369 Ex 3)

* SUPPORT VECTOR CLASSIFIERS

A continuación, se realiza una extensión que cabe denominarla de “sentido común” a todo lo presentado anteriormente. Una clara exigencia de la técnica MMC es que las observaciones tenían que ser linealmente separables. Pero esta situación, como ya se comentó, va a ser poco realista siendo quizás más realista el que las observaciones no puedan separarse mediante el uso de un hiperplano óptimo. De hecho, incluso en situaciones muy sencillas, se observa fácilmente que la hipótesis de separabilidad no se mantiene.

Así pues, se está en una situación tal como la que se describe con la siguiente figura:

**Gráfico, Gráfico de dispersión

Descripción generada automáticamente**

**Figura 6 Conjunto de Ejemplos de un problema no separable linealmente**

**Fuente: Elaboración propia**

Para resolver este problema hay que variar el problema primal anteriormente planteado, para ello se recurre a la introducción del concepto de “variables de holgura” y que en cierto modo tiene que ver en hasta qué punto se está dispuesto a hacer una “vista gorda” a lo que, en principio, en el espacio de característica aquí planteado, sería imposible de separar. Así pues, se plantea un nuevo problema primal que consistiría en la búsqueda de un hiperplano óptimo, como aquel que:

Permite obtener los parámetros: a partir de encontrar:

s. a. con

Donde aquí son lo que se denominan las variables de holgura y pueden tomar:

-El valor 0 en los casos en que exista separación perfecta

-Un valor mayor que 1 en el caso en que los ejemplos en ningún caso sean separables, es decir, la nube de punto de la Figura 3.3 estarían superpuesta y no tendría sentido en principio aplicar esta técnica para este problema

-Un valor entre 0 y 1 que serán los casos que interesan en este apartado

Como se observa en el problema, existe además una constante que en general suele tomar un valor elevado de modo que:

-Valores muy elevados implicarían pequeños valores de la función de holgura para que el sumatorio que existe en la función objetivo a minimizar tome realmente valores pequeños

-Por el contrario, valores muy pequeños de la constante implicaría que las holguras podrían tomar valores muy elevados y como consecuencia podría ocurrir que en casos extremos los ejemplos se clasificasen sin criterio alguno

Así pues, el hiperplano obtenido se denomina de separación de *margen blando*, mientras que al obtenido en el modelo MMC se suele denominar también *hiperplano de separación de margen duro*.

Como se ha hecho en el caso anterior, una vez planteado el primal del problema, falta resolverlo, en este sentido la técnica es análoga al caso anterior y gracias a esto los softwares estadísticos como R pueden tener implementaciones generales que pueden ser aplicados al mundo real:

s. a. con

Donde existe una expresión totalmente análoga para el hiperplano de separación de margen blando obtenida anteriormente y denotada por

Ejemplo 11: Discusión de fronteras en el caso SVC programa11.R

En este ejemplo se centra atención también la atención en el análisis de sensibilidad del parámetro que se introduce en el modelo a estimar. Se observa aquí que el valor de es bastante robusto, es decir, poner un valor entre 1 y 1000 da lugar a resultados análogos, pero el problema es que hay que tomar una decisión y dado que lo “extremo”, está dentro de lo teóricamente posible se precisaría de algún método que facilitase dicha elección.

* SUPPORT VECTORS MACHINE

El último grado de generalización que se le exige a la técnica, por el momento lineal que se ha ido desarrollando, sería la extensión de ésta a los casos en que una separación lineal no tiene sentido ni en sentido duro ni en sentido blando, pudiendo incluso suceder que la aplicación en sí de una frontera no lineal mejoraría a la anterior en sí. Es decir, se podría estar ante situaciones como la que se representan en la siguiente figura (o incluso otras mucho más diversas):

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

**Figura 7 Conjunto de Ejemplos de un problema no linealizables**

La solución intuitiva dada para este tipo de problemas pasa por la utilización de las *Funciones kernel* que lo que hacen es utilizar Espacios de Hilbert de dimensión superior al Espacio de los ejemplos, en el cuál sí existe posibilidad de una separación puramente lineal con el correspondiente hiperplano (recuérdese el caso de la botella de Klein, no representable sin fisuras en el espacio y se tiene que recurrir a la cuarta dimensión para que se pueda representar sin “autocortes superficiales”)

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

**Figura 8 Acción intuitiva de una función kernel**

Antes por tanto de llegar al concepto en sí de la Función Kernel, es necesario considerar un tipo de Función de Transformación que, del espacio original de los ejemplos, conduzca a estos al ya citado Espacio de las Características de tipo Hilbert, así pues, supóngase que existen dichas funciones tal y como se mostraron anteriormente:

En el nuevo espacio, podría considerarse una función de decisión, lo equivalente a un Hiperplano de Separación de los ya observados tal como sigue:

Nótese que se está ante un caso de Producto Escalar de 2 vectores, ahora de dimensión m. Es por ello por lo que se exige que el Espacio de Características sea de tipo Hilbert, ya que es en estos espacios es donde existe al menos definido un producto vectorial y por tanto tiene sentido el poder medir distancias a través de lo que se conoce como una norma.

Si se recuerdan los anteriores modelos, se estaba ante casos donde se presentaba el problema de optimización primal y se comentaba que éstos se resolvían a partir de un problema equivalente denominado dual. En este caso general, también se hace algo análogo y se prefiere utilizar el problema dual directamente, así pues, puede demostrarse que la anterior frontera de decisión es equivalente a usar la siguiente expresión:

Nótese que en esta nueva expresión, equivalente a los duales donde el concepto de producto escalar se ha sustituido por el de la Función Kernel

De hecho, el problema dual que se trata de resolver es el siguiente:

s. a. con

Llegados a este punto hay que precisar más el concepto de Función Kernel y para ello se hace uso del siguiente Teorema matemático:

**Teorema de Aronszajn**

Para cualquier función que sea simétrica y semidefinida positiva, existe un Espacio de Hilbert y una función tal que se cumple que:

\_\_\_\_\_

Este resultado que aquí se enuncia sin demostrar es lo que dota de sentido el uso de las Funciones Kernel aquí introducidas. Es decir, lo que permite el teorema es:

* + Que lo que importa es la forma en sí de la función, es decir, que sea simétrica y semidefinida positiva
  + Que da igual el cómo sea la función que “embede” las observaciones en el Espacio de las Características, por lo que la atención se centra en sí en una Función Kernel, sin tener que detallar la función
  + Además, tampoco es necesario realizar ningún tipo de productos escalares, sólo basta con analizar los resultados de la Función Kernel directamente
  + Incluso en el caso de que la dimensión del Espacio de las Características tipo Hilbert fuera infinito, este resultado y las consecuencias anteriormente comentadas seguirían siendo válidas

Una expresión aún más clara que cumple las anteriores condiciones es el resultado conocido como Teorema de Mercer:

**Teorema (Condición de Mercer)**

Existe una transformación y una expansión en series de la función sí y sólo sí para cualquier función para la que la integral sea finitia, se tiene que:

\_\_\_\_\_

Llegados a este punto, sólo basta encontrar Funciones Kernel que cumplan las 2 condiciones principales del Teorema y observar directamente sus efectos. Algunas de estas posibles funciones serían las que se definen tal como siguen:

* + Kernel Lineal:

* + Kernel Polinómico de grado-p y parámetros-g,h:

* + Kernel gaussiano de parámetro-g:

con

* + Kernel sigmoidal de parámetros-g,h:

Ejemplo 12: Support Vector Machine para la separación de fronteras complejas no linealizables

## 7.3.- Planteamiento y resolución con SVMs de separaciones tipo XOR

Tabla

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

## 7.4.- Extensiones y generalizaciones de los SVMs

Finalmente, el viaje aquí planteado a lo largo de la teoría de los SVM va a finalizar con algunas generalizaciones posibles de los modelos presentados en un apartado anterior. En este sentido, aunque el ámbito “natural” de modelos es la clasificación de observaciones en variables binarias lo cierto es que se ha avanzado hacia la posibilidad de relajar el objetivo un poco más y tratar de permitir:

- La posibilidad de clasificación multi-categórica

- La posibilidad de permitir que la variable explicada sea de tipo continuo

* EXTENSIÓN DE LOS SVM AL CASO MULTI-CATEGORÍA

En este caso se supone que la variable explicada tiene un número de clases por lo que hay dos modalidades de proceder para estos casos:

* + Modalidad 1: Se consideran las posibilidades y se ajustan todos los posibles modelos para separar las posibles clases que se formarían. Posteriormente las comprobaciones de los test o de las nuevas observaciones se llevarían a cabo sabiendo que la asignación final de observaciones se llevaría en función de las clases donde se asocia una observación más veces
  + Modalidad 2: En este caso sólo se ajustan modelos, porque lo que se hace en cada uno de ellos es considerar una de las clases como base y la otra las formadas por todas las demás, esto daría lugar por tanto hiperplanos óptimos del tipo y por tanto la clase final a asociar a los datos test o a los nuevos datos, sería aquella en la que se maximiza la anterior cantidad

Ejemplo 13: Separación en multicategórica en SVM

En el programa13.R Se aplican los SVMs para un caso de 3 categorías separables

* EXTENSIÓN DE LOS SVM AL CASO DE VARIABLES EXPLICADAS CONTINUAS

En este caso se re-escribe el conjunto de ejemplos de entrenamiento tal como sigue:

donde

Ahora el modelo SVM, recibe el nombre de SVR y de lo que se trata es de estimar los parámetros de la siguiente función lineal de :

Y en este caso no se usa un método de mínimos cuadrados, lo cual sería una regresión para la estimación en sí de los parámetros y , sino que lo que se hace es uso de una función de pérdida tal como sigue:

Siguiendo en la línea de razonamiento, es natural encontrar que tanto Python como R permiten la extensión continua al caso continuo de esta técnica que se hace de modo bastante natural a nivel de código.

# **CONCLUSIONES FINALES**

Los modelos de variable discreta resultan ser de los más utilizados en distintas aplicaciones que van desde el riesgo – fraude hasta la predicción del abandono o churn de clientes.

Son unos modelos que pueden ir desde las simples regresiones logísticas, hasta los complejos modelos de ensemble, que aún no han sido tratados. La posibilidad de crear variables binarias para predecir la ocurrencia de un hecho de relevancia en la empresa, resulta básico a nivel de negocio, siendo lo realmente complejo la construcción de la tabla de datos adecuadas para tener un target y unas variables explicativas representativas de la realidad y con posibilidad de hacer uso de estos modelos de un modo eficiente.

De entre los modelos considerados aquí, prácticamente todos menos los SVM resultan hasta cierto punto bastantes intuitivos en su construcción, los SVM en cambio, parecen sencillos de interpretar pero dejan abiertas cuestiones relevantes que negocio sin duda preguntará como ¿Qué significan los vectores soporte en sí? ¿Cuáles son las variables relevantes que influyen en el comportamiento? Además al igual que los KNN, entrenar (sobre todo en este caso) y ejecutar resulta largo o costoso al tenerse que estimar distancias en espacios con dimensiones usualmente altas y además los buenos resultados que se obtuvieron con ellos durante el período 2014-2016 fueron totalmente eclipsados por la familia de los modelos de ensemble, mucho más rápidos de entrenar y más eficientes a nivel predictivo como se estudiará.

# **BIBLIOGRAFÍA**

**Amat Rodrigo J.** *Regularización Ridge, Lasso y Elastic Net con Python, available under a Attribution 4.0 International (CC BY 4.0)* at <https://www.cienciadedatos.net/documentos/py14-ridge-lasso-elastic-net-python.html>

**McFadden, D. (1974)** *Conditional logit analysis of qualitative choice behaviour* pp. 105-142 in P. Zarembka(ed.) Frontiers in Cenometrics. Academic Press

**Gareth, J.; Witten, D.; Hastie, T. y Tibshirani R. (2013)** *An Introduction to Statistical Learning with Applications in R* Springer Science + Business Media New York ISBN 978-1-4614-7137-0