PRÁCTICA 4: MACHINE LEARNING - REGRESIÓN

Estudiante 1:	**Grupo:**
Estudiante 2:	

4.1 MÉTODO.

— Mediante el empleo de la regresión en Matlab, se procederá al estudio del comportamiento de variables de salida en función de las pertinentes variables de entrada, con el fin de predecir el estado de las primeras ante la presencia de nuevas variables de entrada desconocidas hasta el momento, aplicable al campo del diseño y mantenimiento de sistemas mecatrónicos.

4.2 OBJETIVOS.

— Conocimientos básicos sobre machine learning.
— Aplicación de la regresión, en sus diferentes tipologías.
— Realización de representaciones gráfica.
— Interpretación de datos para la predicción de valores de variables.

4.3 INTRODUCCIÓN AL MACHINE LEARNING (INTELIGENCIA ARTIFICIAL).

El machine learning es conocido o denominado en español, derivado de su traducción, como aprendizaje de máquinas o aprendizaje automático, siendo esta última terminología la que va a ser usada a partir de ahora.

La idea es que hay un área de la técnica dedicada a hacer que las computadoras aprendan sin tener que ser programadas para realizar una tarea.

¿Cómo se puede conseguir hacer eso? Por ejemplo, lo que se podría hacer es dejar que la máquina te observe realizando una tarea y que aprenda de ello.

Esto se encuentra ligado al concepto de inteligencia artificial, de tal manera que, al construirse máquinas realmente inteligentes, se podría hacer que ellas hicieran cualquier trabajo por el ser humano.

Muchos científicos e ingenieros piensan que la mejor manera de progresar en este campo es por medio de algoritmos de aprendizaje conocidos como redes neurales, las cuales imitan el funcionamiento del cerebro humano. No obstante, muchas veces se resuelven los problemas cotidianos empleando técnicas menos complejas que dan óptimos resultados.

Es posible que hayas usado algoritmos de aprendizaje muchas veces en un día sin darte cuenta. Cada vez que se usa un buscador web como Google o Bing para hacer una búsqueda en internet, una de las razones por las que funciona tan bien es porque un algoritmo de aprendizaje, implementado por

Google o Microsoft, aprendió a clasificar las páginas web. Cada vez que usas Facebook e Instagram reconoce las fotos de tus amigos, eso es aprendizaje automático. Cada vez que lees tu correo electrónico o que tu filtro de spam te salva de tener toneladas de correos basura, eso también es un algoritmo de aprendizaje. Pero claro, esto desde nuestra perspectiva tendrá que trasladarse al ambiente industrial.

Una de las razones por las que los científicos y los técnicos se emocionan cada vez más, es el sueño de la inteligencia artificial, es decir, construir máquinas tan inteligentes como los seres humanos. Aunque todavía estamos lejos de alcanzarlo con un grado de perfección alto, muchos investigadores que trabajan en inteligencia artificial creen que el mejor camino para lograrlo es mediante algoritmos de aprendizaje que traten de imitar el proceso de aprendizaje del cerebro humano.

Entonces, ¿por qué está tan presente hoy en día el aprendizaje automático en cualquier área tecnológica? Resulta que el aprendizaje automático es un campo que nació, como ya se ha hecho referencia, a partir de la inteligencia artificial. Se querían construir máquinas inteligentes, pero realmente las tareas que realizan eran muy básicas, ya que, en gran medida, no se sabía cómo hacer tareas más complejas. De tal manera, se llego a la conclusión de que la única forma de hacer estas tareas más complejas era que una máquina aprendiera a hacerlas por sí misma. Así que el aprendizaje automático se desarrolló como una nueva capacidad para las computadoras y que hoy en día está en muchos sectores de la industria y en otros, debido a que la gama de problemas que trata el aprendizaje automático es muy extensa.

A continuación, se va a trata de dar definiciones más formales de que es aprendizaje automático y de dar una idea de cuando está justificado su empleo.

Hay que decir que incluso entre los practicantes del aprendizaje automático no existe una definición bien aceptada sobre lo que es y lo que no es aprendizaje automático. Pero se van a presentar una serie de ejemplos sobre las maneras en que la gente ha intentado definirlo.

Arthur Samuel definió el aprendizaje automático como “Field of study that gives computers the ability to learn without being explicitly programmed", que traducido sería el campo de estudio que le da a los ordenadores la habilidad de aprender algo sobre lo que no han sido explícitamente programados. La fama de Arthur Samuel se remonta a la década de 1950, cuando desarrollo un programa para jugar a las damas, siendo el un nefasto jugador de damas.

Otra definición más reciente es la dada por Tom Mitchell, que manifiesta que un problema de aprendizaje automático bien planteado es definido de la siguiente forma: un programa de ordenador se dice que aprende de la experiencia E, con respecto a T, y alguna medida de rendimiento P, si esta actuación en T, medida por P mejora la experiencia E.

Por ejemplo, para el juego de damas, la experiencia E, será la experiencia de tener el programa jugando decenas de miles de partidas reiteradas contra él mismo. La tarea T, será la tarea de jugar partidas y la medida de mejora P, será la probabilidad que lo haga ganar la siguiente partida de damas contra un nuevo oponente.

4.4 PRINCIPALES ENFOQUES DEL APRENDIZAJE AUTOMÁTICO.

Dentro de lo que se puede considerar aprendizaje automático nos podemos encontrar con las siguientes disciplinas:

- Aprendizaje memorístico: se establece una correspondencia uno a uno entre las entradas y la representación almacenada.
- Aprendizaje basado en analogías: determinar la correspondencia entre dos representaciones diferentes.
- Aprendizaje inductivo: se utilizan ejemplos específicos para alcanzar conclusiones (hipótesis, modelos, etc.) generales. Pudiendo darse dos vertientes:
  - Aprendizaje supervisado.
  - Aprendizaje no supervisado
- Aprendizaje por refuerzo: Realimentación proporcionada al final de una secuencia de pasos, mediante un refuerzo positivo o negativo.

De entre todas las posibilidades anteriores que existen para el aprendizaje automático, la más empleada es la denominada aprendizaje inductivo, en sus dos versiones supervisada y no supervisada.

A grandes rasgos, ya que posteriormente serán tratados en más profundidad, resulta que, en el aprendizaje supervisado, la idea es que se va a enseñar al equipo a como hacer algo, mientras que en el aprendizaje no supervisado se va a permitir al equipo a aprender por sí mismo.

Quizás viendo esas breves definiciones se crea que el no supervisado sea el más relevante, pero no es así, debido a las dificultades que entraña no disponer del comportamiento en sí del sistema a estudiar, cuestión esta a entender más adelante. Por lo tanto, el más común a emplear debido a la problemática suscitada, será el denominado aprendizaje automático supervisado, aunque no hay que dejar para nada de lado el no supervisado.

4.5 APRENDIZAJE SUPERVISADO Y NO SUPERVISADO.

El término aprendizaje supervisado se refiere al hecho de dar al algoritmo un conjunto de datos donde se muestran las “respuestas correctas”, por ejemplo, se dispone de horno de secado al que van entrando piezas con distinto grado de humedad y tienen que ser secadas por completo, por lo que para su secado cada una de ellas deberá estar en su interior un tiempo determinado, obviamente en función del grado de humedad tomado a la entrada.

Para definir un poco más la terminología, a esto también se denomina un problema de regresión, que significa que el objetivo es predecir un resultado de valor continuo.

Pero aún hay algo más, se puede hablar de problemas de clasificación, donde el objetivo es predecir un resultado de valor discreto. Por ejemplo, tras un proceso de test de velocidad realizado a un microprocesador, se determina si es óptimo o no para su comercialización, es decir, el estado “1” corresponde a óptimo para su comercialización y el estado “0” es de rechazo.

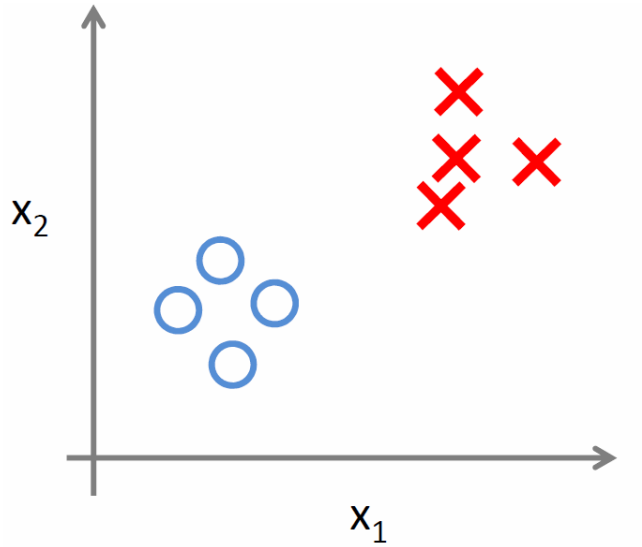
Estos ejemplos son muy básicos, porque en el problema de regresión hay una única variable de entrada, humedad, y una variable de salida, tiempo de la pieza dentro del horno para el secado, pero en la realidad, el número de entrada de variables puede ser elevado.

Por otro lado, en los problemas de clasificación se puede dar también un número elevado de variables de entrada y estados de salidas, clasificados en este caso de uno hasta infinito.

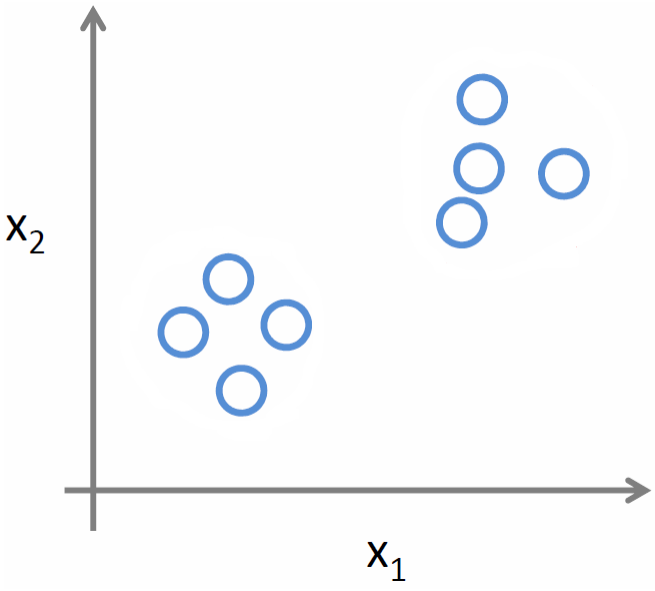
Los algoritmos que se emplean en el aprendizaje supervisado, debido a su relevancia y uso, serán los siguientes:

- Para problemas de regresión.
  - Regresión lineal.
  - Regresión lineal multivariable.
  - Redes neuronales artificiales (RNA)
- Para problemas de clasificación:
  - Regresión logística.
  - Redes neuronales artificiales (RNA).
  - Máquinas de soporte vectorial (MSV).

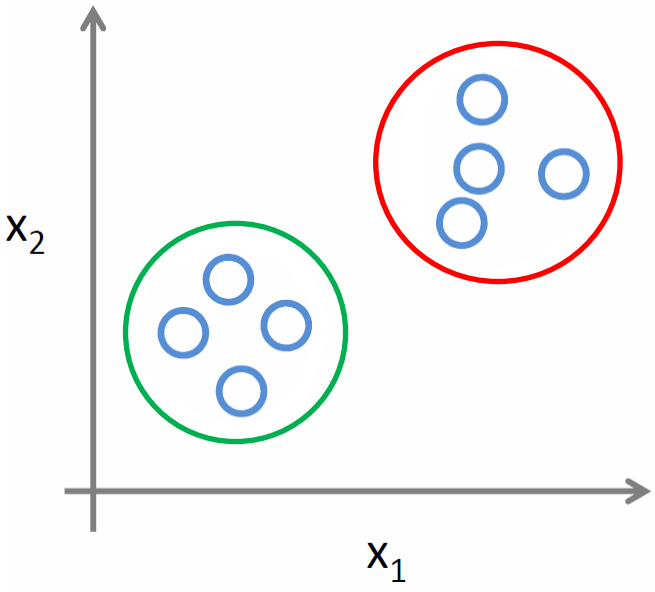
Hasta ahora se ha hablado sobre el aprendizaje supervisado. Entonces, por ejemplo, para un caso de clasificación, se tenía un conjunto de datos, que se suelen ver como aparecen reflejados en la figura, donde cada ejemplo estaba marcado como un ejemplo positivo o negativo, si era un chip óptimo para la comercialización o no.



Así que, en cada ejemplo de aprendizaje supervisado, nos decían explícitamente cuál era la respuesta correcta, si era óptimo o no para la comercialización. En el aprendizaje no supervisado, nos dan datos que se ven diferentes a los datos anteriores, es decir, que no tienen ningún valor asignado o que todos tienen el mismo valor asignado o realmente ninguno, tal como se puede apreciar en la figura siguiente.



Entonces la pregunta típica que se podría plantear sería algo como ¿se puede encontrar alguna estructura en estos datos? De tal manera que, con estos datos, un algoritmo de aprendizaje no supervisado podría decidir que lo datos están en dos grupos diferentes, tal como muestra la figura siguiente.



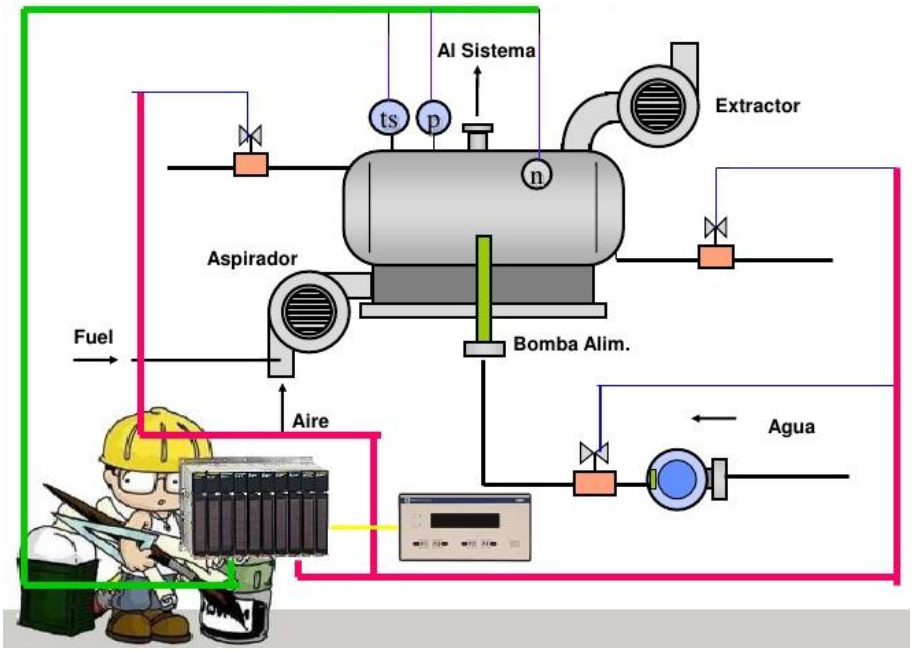
Los algoritmos que se van a emplear en el aprendizaje no supervisado, debido a su relevancia y uso, prácticamente centrados en problemas de clasificación, serán:

- Algoritmo K-means.
- Algoritmo gaussiano de detección de anomalias.

4.6 REGRESIÓN LINEAL.

Para entender mejor el algoritmo de regresión lineal se va a poner un ejemplo, de los muchos que se pueden dar a nivel industrial. Se dispone de un depósito en la que para su correcto funcionamiento hay que ajustar su presión teniéndose en cuenta la temperatura a la que está trabajando para que de esta

manera no se produzca ningún problema. Para lo cual se dispone de un sensor de medida de temperatura analógico de 0-10 V, cuya medición es introducida en un autómata programable que, su vez a través de un software desarrollado al efecto, actúa sobre una electroválvula proporcional a través de una salida analógica de 4-20 mA, figura.



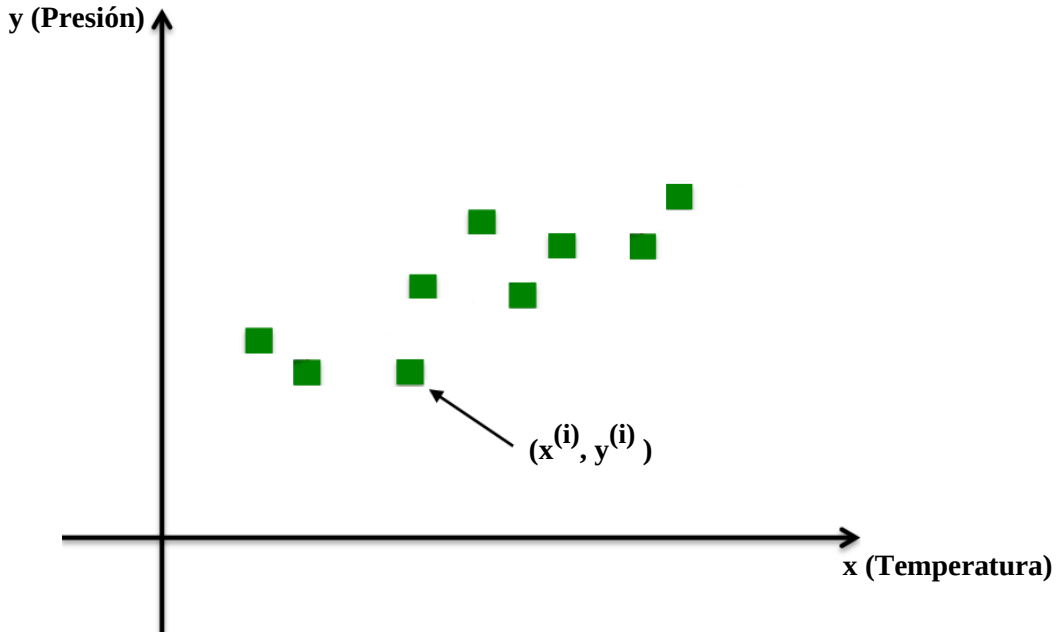
La pregunta que nos debemos hacer sería ¿Para cualquier valor de temperatura que apertura tiene que tener la electroválvula para conseguirse la presión adecuada en el interior del depósito?, y no solo ese si no otra más ¿Cómo desarrollar el software de este control desde el punto de vista del algoritmo de regresión lineal?

Se va a intentar responder a las dos preguntas a la vez, para ello lo primero que se tiene que disponer es de una base de datos en las cuales se reflejen una cantidad de valores de temperatura en un numero relevante (esto de un número relevante ya se aclarara posteriormente) y asociados a cada uno de ellos su correspondiente valor de presión que hace que el sistema funcione a la perfección

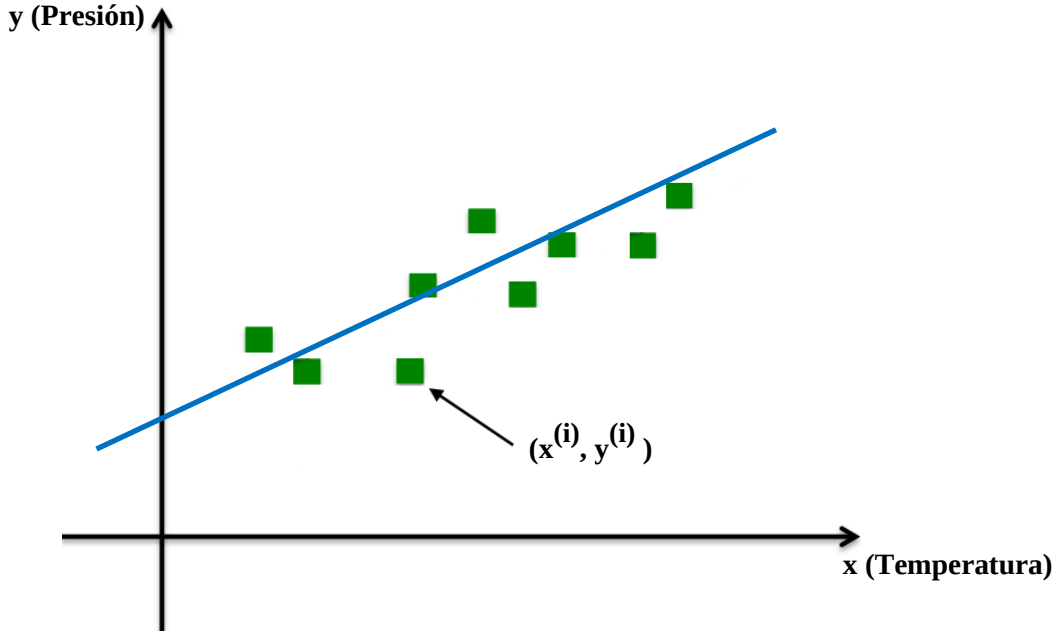
A la temperatura se le denomina variable o característica de entrada, denotándose como “x” y a la variable presión se le denomina variable de salida, denotada como “y”, por lo que se dispondrá de un determinado número de pares de valores (x, y) y al conjunto de todos estos datos se les denomina conjunto de entrenamiento.

Llegado a este punto, tiene que estar claro que lo que se quiere es determinar el valor de la presión en función del valor que toma la temperatura en el proceso, es decir, predecir el valor de la presión para cada valor de temperatura posible que se puede dar y que no ha sido contemplado en el conjunto de entrenamiento, que será el que se emplee para determinar el modelo. Por lo tanto, se tendrá que proceder a ajustar dicho modelo.

Lo primero que se va a hacer es trazar el conjunto de datos de entrenamiento para cada par de datos (x, y), lo que viene en denominarse diagrama de dispersión, figura siguiente.



Tal vez ajustar una línea recta con estos datos sería posible, pudiéndose ver como aparece reflejado en la figura siguiente.



Así que este es un ejemplo de un algoritmo de aprendizaje supervisado, se define como tal porque se da la "respuesta correcta" para cada uno de los ejemplos. Es decir, se dice cuál fue la temperatura real y cuál fue la presión real de cada lectura que se dio para el conjunto de datos y, además, se trata de un ejemplo que refleja un problema de regresión, el término regresión se refiere al hecho de que se predice el resultado del valor real, es decir, la presión.

De forma más formal, debido a que se tiene un conjunto de datos, que como ya se ha dicho, se denomina conjunto de entrenamiento, se está ante un algoritmo de aprendizaje supervisado, de tal manera que el cometido será aprender, a partir de estos datos, cómo predecir las nuevas presiones de trabajar a tener en el control a diseñar.

Se va a proceder a indicar una serie de notaciones que van a tenerse en cuenta en este algoritmo, siendo extrapolables algunas de ellas a otros algoritmos, es decir:

m: Número de datos del conjunto de entrenamiento.

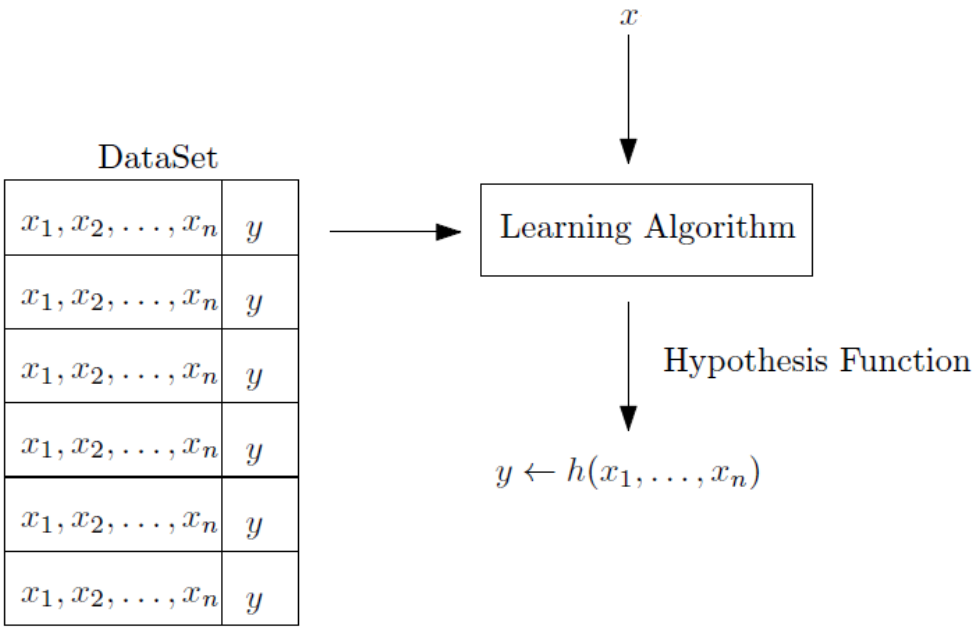
x: Variable de entrada.

y: Variable de salida.

(x, y): Dato de entrenamiento genérico.

$(x^{(i)}, y^{(i)})$: Dato de entrenamiento para el valor i específico.

Entonces, el conjunto de entrenamiento de temperaturas y presiones se presenta al algoritmo de aprendizaje, siendo su trabajo el de generar una función, que mediante convención generalmente está denotada por la letra "h" minúscula, refiriéndose a la denominación de hipótesis, tal como se refleja en el diagrama de la figura.



El trabajo de la hipótesis es una función que toma como entrada el valor de “x” y trata de generar un valor estimado de “y”, siendo entonces “h” una función que se traza de las “x” a las “y”.

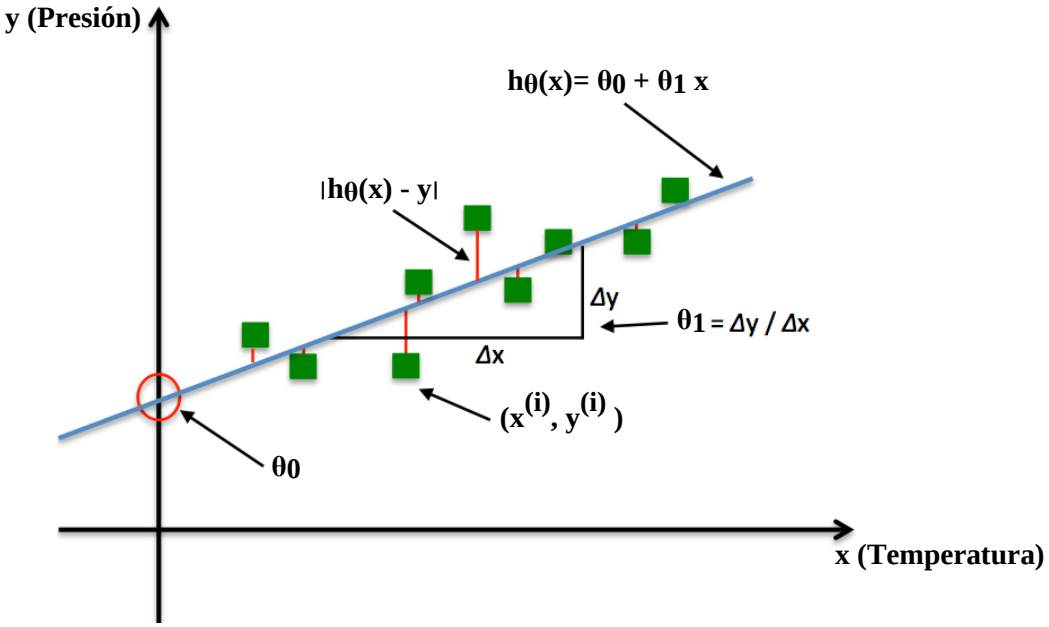
Lo siguiente será decidir cómo representar esta hipótesis “h”, aunque existen varias formas de expresarla, una de las más común que se suele emplear en aprendizaje automático, es la siguiente.

$h_\theta(x) = \theta_0 + \theta_1 x$

θ_i: Parámetro de predicción.

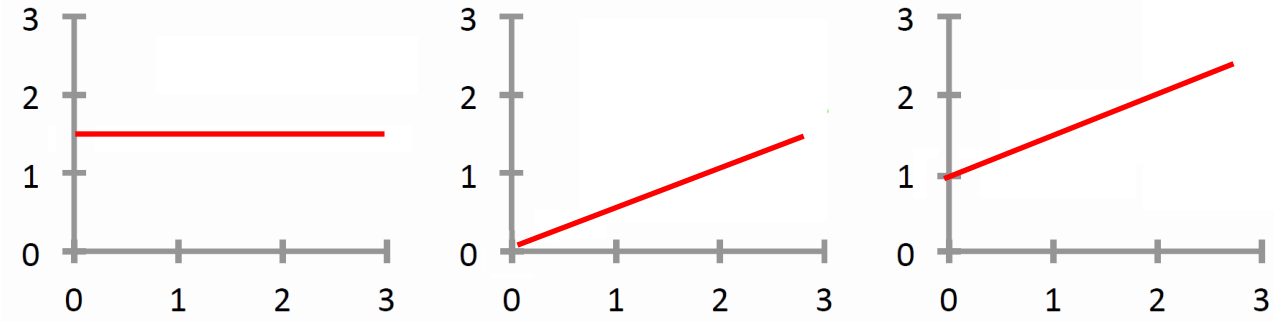
A este algoritmo genéricamente se le denomina regresión lineal. No obstante, el ejemplo en concreto que se ha tratado sería una regresión lineal real con una variable, una variable que es “x” y predice todos los valores de “y”, como funciones de la variable “x”. Otro nombre que se le puede dar es regresión lineal univariante o simple. Univariante es sólo una forma elegante de decir una variable.

En la figura siguiente se puede ver un resumen gráfico de todo lo comentado hasta el momento.



Ahora llegado este punto la siguiente pregunta a hacerse será ¿cómo ajustar la mejor línea recta posible para los datos del conjunto de entrenamiento? Está claro que lo que hay que determinar son los parámetros θ's, ya que en función de ellos la recta variara, figura siguiente, para ello se emplearan las herramientas disponibles en Matlab.

$$h_\theta(x) = \theta_0 + \theta_1 x$$



4.7 REGRESIÓN LINEAL MULTIVARIABLE.

En este apartado se va a trabajar sobre una versión nueva y más poderosa de la regresión lineal, aquella que trabaja con múltiples variables o con múltiples características.

En la versión inicial de la regresión lineal que se ha desarrollado en el apartado anterior, se tenía una sola característica "x", el tamaño de la casa, y se quería usar para predecir "y", el precio de la casa, de tal manera que a partir de todo esto se formulaba la correspondiente hipótesis.

Pero ahora imagina, qué pasa si no solo se tuviese el tamaño de la casa como característica o como variable entrada para tratar de predecir el precio de la casa, sino que también se supiese el número de habitaciones, el número de plantas y la edad de la casa. Parece que esto daría mucha más información para predecir su precio.

Para introducir un poco de notación, que en cierto modo ya ha sido mencionada antes, se van a utilizar las variables "x_1", "x_2", "x_3" y así sucesivamente hasta denotar las variables de entrada o características y se va a seguir utilizando "y" para denotar la variable de salida, que se está tratando de predecir. No obstante, se va a introducir un poco más de notación como la siguiente:

n: Número de variables de entrada o características (número de columnas de una tabla).

m: Número de ejemplos de entrenamiento (número de filas de una tabla).

$x^{(i)}$: Variable de entrada o característica del ejemplo de entrenamiento "i-ésimo".

$x_j^{(i)}$: Variable de entrada o característica "j" en el ejemplo de entrenamiento "i-ésimo".

Todo esto necesita una aclaración con un ejemplo en concreto, para la cual vamos a seguir con el ejemplo de las viviendas, teniéndose ahora la tabla siguiente, que como se puede observar refleja más información que en el ejemplo anterior.

Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

Entonces, para este ejemplo en concreto se podrá denotar como:

x_1 = Size.

x_2 = Number of bedroms.

x_3 = Number of floors.

x_4 = Age.

y = Price.

n = 4.

$$x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix} \in \Re^4$$

$$x_3^{(2)} = 2$$

Ahora que se tienen múltiples características, nos tendríamos que preguntar ¿cómo debería ser la formulación de la hipótesis? Anteriormente, esta era la formulación de la hipótesis $h_\theta(x) = \theta_0 + \theta_1 x$,

donde "x" era la única característica, pero ahora que se tienen múltiples características, ya no se puede emplear dicha expresión, si no que la formulación de la hipótesis pasará a ser la siguiente.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \cdots + \theta_n x_n$$

Para conveniencia de la notación, se va a definir una nueva variable como $x_0 = 1$ o mejor notación $x_0^{(i)} = 1$, con lo que la expresión de la hipótesis quedará de la manera siguiente.

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \cdots + \theta_n x_n$$

Esto se puede pensar como disponer de una característica “cero” adicional de valor constante la unidad.

Mientras que anteriormente se tuvo "n" características debido a "x_1" hasta "x_n", como ahora se está definiendo una nueva adicional, "x_0", ahora el vector de características "x" se convierte en uno de dimensiones n+1, al igual que el de θ’s, es decir.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \in \Re^{n+1} \text{ y } \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \Re^{n+1}$$

A partir de todo esto se tendrá los siguiente.

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x$$

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \cdots & \theta_n \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Entonces, esta es la formulación de una hipótesis cuando se tienen características múltiples. Y, solo para darle otro nombre diferenciado de la regresión lineal vista con anterioridad, se le denomina regresión lineal multivariable.

El término multivariable es solamente una forma elegante de decir que se tienen múltiples características, o varias variables de entrada para tratar de predecir el valor de "y".

4.8 REGRESIÓN POLINOMIAL.

Una cuestión importante a tener en cuenta, se encuentra relacionada con la elección de las variables de entrada que se tienen disponibles y de cómo se puede obtener diferentes algoritmos de aprendizaje, a veces muy poderosos eligiendo las variables apropiadas.

De manera particular, se puede decir que la regresión polinomial permite utilizar la maquinaria de la regresión lineal para ajustar funciones muy complicadas, incluso no lineales.

Tómese de nuevo el ejemplo que nos viene acompañando hasta hora, el de predecir el precio de una casa. Supóngase que se tienen dos variables, la fachada de la casa y la profundidad de la casa. Encontrándose aquí la imagen de la casa que se está tratando de vender.

Con estas variables es posible construir un modelo de regresión lineal como el reflejado en la expresión, en donde la fachada es la primera variable x_1 y la profundidad es la segunda variable x_2.

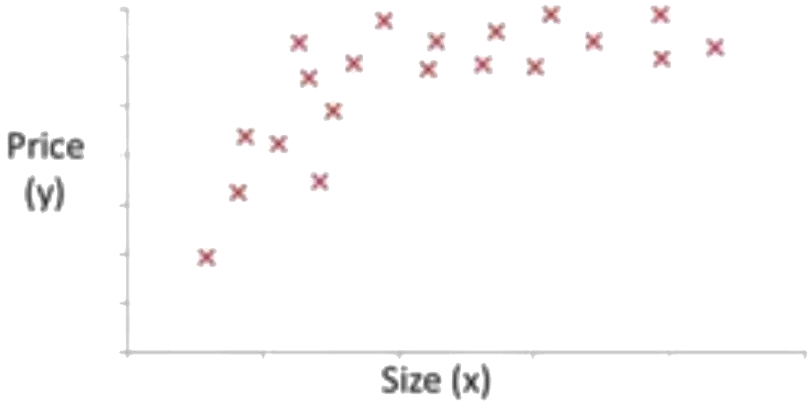
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = \theta_0 + \theta_1 \cdot \text{fachada} + \theta_2 \cdot \text{profundidad}$$

Pero cuando se está aplicando la regresión lineal, no necesariamente se pueden usar solamente las variables x_1 y x_2 dadas. Lo que se puede hacer es crear nuevas variables, Así que, si se quiere predecir el precio de una casa, lo que se podría hacer en su lugar es decidir que lo que realmente determina el tamaño de la casa es su superficie. Así, podría crear una nueva variable como la siguiente y una nueva expresión para la hipótesis.

$$\text{superficie} = \text{fachada} \cdot \text{profundidad} = x \Rightarrow h_\theta(x) = \theta_0 + \theta_1 x$$

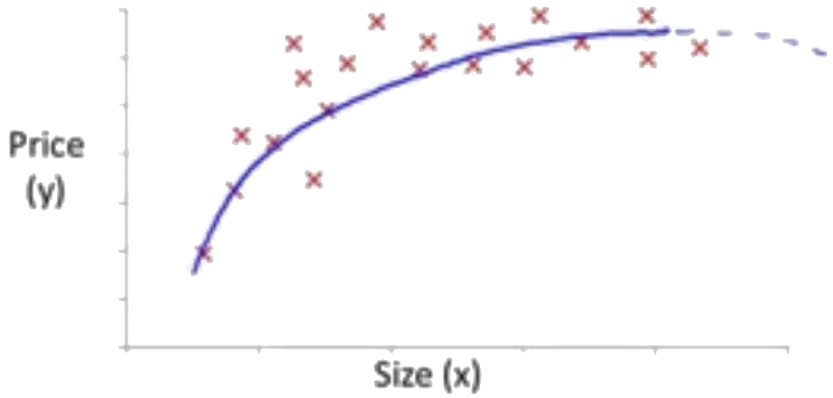
Así es que, dependiendo del entendimiento que se puede tener sobre un problema particular, en lugar de simplemente tomar las variables fachada y profundidad que son las proporcionadas en un principio, a veces mediante la definición de nuevas variables se podría conseguir un mejor modelo.

Estrechamente relacionada con la idea de elegir las variables está la idea denominada regresión polinomial. Digamos que se tiene un conjunto de datos de precios de vivienda que tienen el aspecto reflejado en la siguiente figura.



Se pueden dar algunos modelos diferentes con los que se podría ajustar a este conjunto de datos. Lo que se podría hacer es ajustar un modelo cuadrático como se muestra en la figura, ya que no parece que una línea recta se ajuste muy bien a los datos.

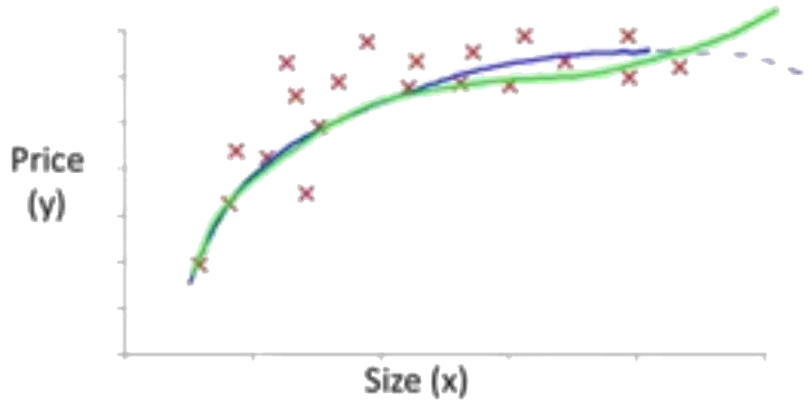
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$



Así que tal vez se quiera ajustar un modelo cuadrático como éste en donde se piensa que la superficie de la vivienda contribuye su precio de esta forma. Pero entonces puede ocurrir, que se decida que el modelo cuadrático no tiene sentido, porque eventualmente esta función empieza a descender a partir de un punto y no es posible que los precios de las viviendas deban bajar cuando su superficie sea muy elevada.

Tal vez, se pueda elegir un nuevo modelo polinomial diferente y optar por utilizar una función cúbica, en donde se tiene un término de tercer orden. Tal vez, este nuevo modelo se ajusta un poco mejor a los datos, ya que se observa en la nueva figura que los precios no bajan eventualmente.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$



Entonces, para el ejemplo en concreto que se está tratando se tendrá lo siguiente.

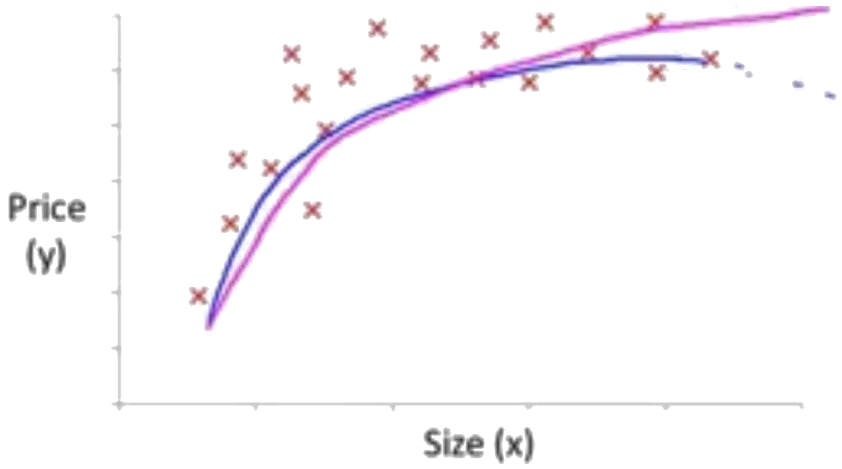
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \begin{cases} x = \text{superficie} \\ x^2 = \text{superficie}^2 \\ x^3 = \text{superficie}^3 \end{cases}$$

$$h_\theta(x) = \theta_0 + \theta_1 \cdot \text{superficie} + \theta_2 \cdot \text{superficie}^2 + \theta_3 \cdot \text{superficie}^3$$

Sólo señalar una cuestión más, y es que, si eligen las variables de esta forma, entonces el escalamiento de variables se hace cada vez más relevante, ya que habrá una gran diferencia entre los rangos de las variables, debido al estar elevada al cuadrado y nada menos que al cubo. Haciéndose imprescindible, si se está usando el gradiente descendiente, poner las variables en rangos de valores comparables.

Pero en realidad esto no se acaba aquí, porque aún podría darse un nuevo modelo que como el relejado a continuación, en el que aparece la particularidad de que la curva se aplana un poco para superficie elevadas. Por lo que, tener un amplio conocimiento sobre este tema, es decir, ser consciente de la forma de la distribución de los datos, saber elegir las posibles variables, etc., normalmente conduce a la elección de mejores modelos.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 \sqrt{x} \Rightarrow h_\theta(x) = \theta_0 + \theta_1 \cdot \text{superficie} + \theta_2 \cdot \sqrt{\text{superficie}}$$



Como se ha podido observar, todo esto parece un poco desconcertante, con todas esas opciones de variables y modelos diferentes. Entonces, ¿cómo se decide qué variables y modelos utilizar? En temas más avanzados, se tratarán algunos algoritmos que automáticamente eligen qué variables utilizar y también seleccionan de manera automática si se desea ajustar una función cuadrática, una función cúbica, o el tipo de función más idónea. Pero, hasta que se estudien esos algoritmos, tiene que haber quedado claro y ser consciente de que se tienes diferentes opciones en cuanto a qué variables utilizar, y mediante el diseño de esas diferentes variables se pueden ajustar funciones más o menos complejas a los datos, solamente ajustando una línea recta, aplicando funciones polinomiales, etc. Y a veces el conocimiento apropiado de la variable permite obtener un modelo mucho mejor que represente el comportamiento de los datos.

4.9 REGRESIÓN LOGÍSTICA.

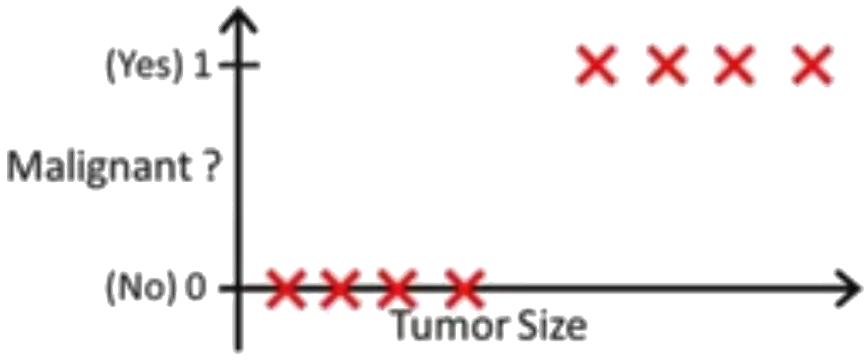
Este tipo de regresión se encuentra ligada a los denominados problemas de clasificación, donde la variable y lo que se quiere predecir tienen valores discretos, en vez de continuos como se ha tratado hasta el momento, en la regresión lineal.

El algoritmo llamado regresión logística, es uno de los algoritmos de aprendizaje más populares y más ampliamente utilizados actualmente. Algunos ejemplos de problemas de clasificación pueden ser detectar correos electrónicos ¿spam o no?, transacciones online ¿fraudulentas o no?, detección de tumores ¿malignos o benignos?, funcionamiento de una máquina ¿anómalo o no?, etc.

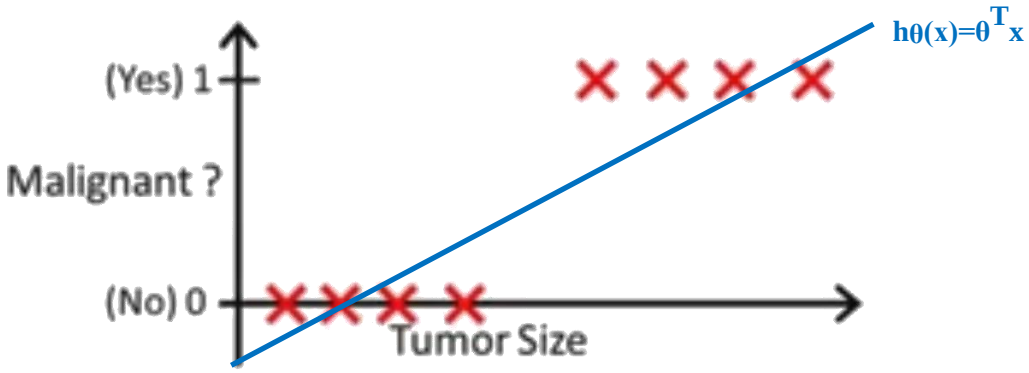
En todos estos problemas, la variable que se trata de predecir es una variable "y" de la que se sabe que toma dos valores, ya sea cero o uno, ya sea correo deseado o no deseado, fraudulento o no fraudulento, maligno o benigno, anómalo o correcto. Otro nombre para la clase que se denota con “0” es la clase negativa, y otro nombre para la clase que se denota con “1” es la clase positiva. Así que “0” puede indicar un tumor benigno y “1”, la clase positiva, puede indicar un tumor maligno. La asignación de las dos clases a positivo y negativo, es decir, a “0” y “1” es un tanto arbitraria y en realidad no importa. Pero a menudo existe esta intuición de que la clase negativa está transmitiendo la ausencia de algo, como la ausencia de un tumor maligno, mientras que uno, la clase positiva, está transmitiendo la presencia de algo que se podría estar buscando.

Por ahora, se va a empezar con problemas de clasificación con sólo dos clases o clasificación binaria; cero y uno. Más adelante, se tratarán problemas multiclase, en donde la variable "y" puede tomar, por ejemplo, el valor de cero, uno, dos y tres. Esto se denomina problema de clasificación multiclase.

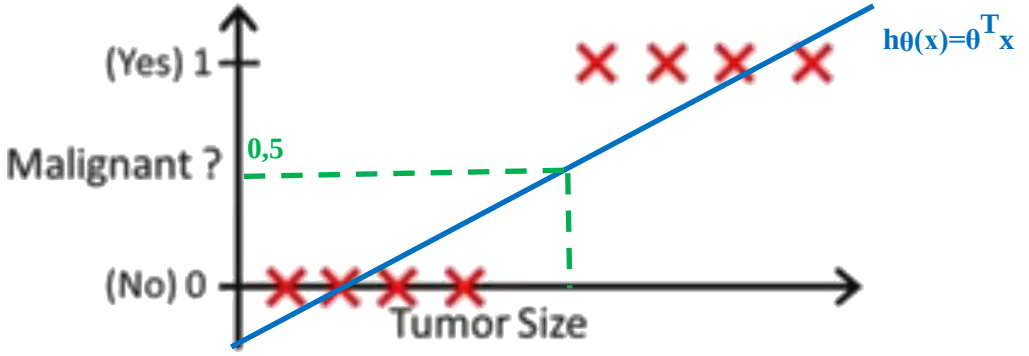
Entonces, ¿cómo se desarrolla un algoritmo de clasificación? Aquí hay un ejemplo de un conjunto de entrenamiento para una tarea de clasificación, para clasificar un tumor como maligno o benigno, observándose que la malignidad adquiere únicamente dos valores, de “0” o No o de “1” o Sí.



Una cosa que se podría hacer dado este conjunto de entrenamiento es aplicar el algoritmo, ya conocido, de la regresión lineal para este conjunto de datos, tratando de ajustar una línea recta a los datos. Así que, si se toma este conjunto de entrenamiento y se ajusta una línea recta al mismo, tal vez se obtenga una hipótesis que se parezca a lo reflejado en la figura siguiente.

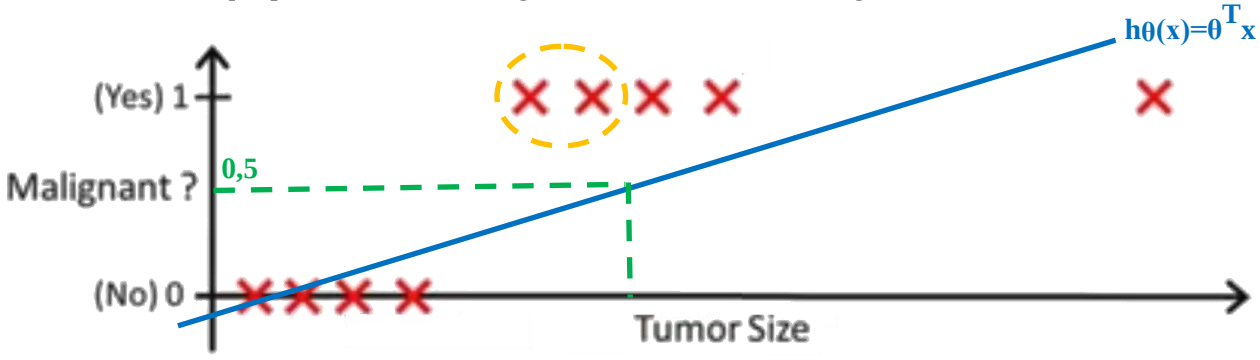


Entonces, si se quieren hacer predicciones, una cuestión que se puede tratar de hacer es fijar el umbral de los resultados del clasificador en 0,5. Y si la hipótesis da como resultado un valor que es mayor o igual que 0,5 se tendrá la predicción de "y = 1", mientras que si es menor que 0,5; la predicción será de "y = 0", tal como se puede observar en la figura siguiente.



Por lo tanto, al usar la regresión lineal de esta manera, todo lo que está a la derecha de ese punto, terminará prediciendo la clase positiva, debido a que los valores resultantes son mayores que 0,5 en el eje vertical y todo lo que está a la izquierda de ese punto, terminará prediciendo un valor negativo. En este ejemplo particular, parece que la regresión lineal está en realidad realizando algo razonable a pesar de que se trate de un problema de clasificación.

Pero ahora se va tratar de cambiar un poco el problema. Se va a ampliar el eje horizontal un poco y se va a tener otro ejemplo de entrenamiento más allá a la derecha, obteniéndose una nueva recta de regresión, que, siguiendo las indicaciones anteriores, se podrá apreciar que no predice bien, ya que hay situaciones en las que predice tumores benignos siendo en realidad malignos.



Entonces, se puede llegar a la conclusión, de que para que no suceda esto, en vez de trabajar con la regresión lineal habrá que trabajar con otro tipo de regresión denominada logística, en la que se cumplirá la expresión siguiente.

$$0 \le h_\theta(x) \le 1$$

Como interesaría que el clasificador genere valores que se encuentren entre cero y uno. Por lo tanto, hay que proponer una hipótesis que satisface esta propiedad, para que las predicciones se encuentren entre dichos valores.

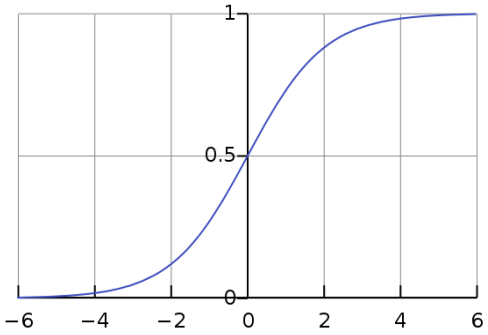
Cuando se usaba la regresión lineal, la forma de la hipótesis era $h_\theta(x) = \theta^T x$. Para la regresión logística, se va a modificar dicha expresión, es decir, $h_\theta(x) = g\left(\theta^T x\right)$. A "g" se le denomina función sigmoide o logística, de ahí el nombre de regresión logística. La expresión de "g" será la siguiente.

$$g(z) = \frac{1}{1 - e^{-z}}$$

Y si se toman estas dos expresiones y, se ponen juntas, entonces aquí se tiene simplemente una manera alternativa de escribir la formulación de la hipótesis.

$$h_\theta(x) = \frac{1}{1 - e^{-\theta^T x}}$$

Se va a mostrar cómo se ve la función sigmoide, trazándola en la figura siguiente. Como se puede observar la función sigmoide, g(z), comienza cerca de cero y luego se eleva hasta que alcanza el valor de 0,5 en el origen, aplanándose de nuevo cuando llega cerca del uno.



Como se puede observar si "z" tiende a infinito la función se aproxima a uno y si "z" tiende a menos infinito se aproxima a cero, por lo tanto, la función g(z) ofrece valores que están entre cero y uno, tal como se quería que ocurriese para la hipótesis $0 \le h_\theta(x) \le 1$.

Dada esta representación de la hipótesis, lo que se necesitaría hacer, siguiendo el mismo proceso visto en prácticas anteriores, es ajustar los parámetros θ al conjunto de datos proporcionado para que la hipótesis obtenida permita hacer predicciones. Con lo cual será necesario hablar de un algoritmo de aprendizaje para el ajuste de los parámetros θ. No obstante, primero se va a hablar un poco acerca de la interpretación de este modelo, es decir, cuando la hipótesis de como resultado algún número, se va a tratar a ese número como la probabilidad estimada de que "y = 1" para un nuevo ejemplo de entrada "x".

Para entenderlo mejor, se va a trabajar con el ejemplo de clasificación de tumores. Se dispone de un vector de variables "x", en el que se tiene $x_0 = 1$, como siempre, y como única variable "x_1 = tamaño del tumor". Supóngase que se tiene un paciente del que se conoce el tamaño del tumor que posee, se introduce dicho tamaño en la hipótesis generada, dando como resultado 0,7; ¿cómo se interpretaría dicho resultado? La hipótesis está diciendo que para un paciente con "x" variables, la probabilidad de que "y" sea igual a 1 es de 0,7. En otras palabras, se le va a decir al paciente que el tumor, por desgracia, tiene un 70 % de probabilidad de ser maligno.

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{Tamaño del tumor} \end{bmatrix} \Rightarrow h_\theta(x) = 0{,}7 \Rightarrow \text{probabilidad del 70 \% de ser maligno } (y = 1)$$

Todo lo anterior tiene la notación matemática siguiente.

$h_\theta(x) = P(y = 1 \mid x; \theta)$, es decir, probabilidad de que "y = 1", dado x y parametrizada por θ.

También, se podría calcular la probabilidad de "y = 0", de la misma forma.

$$P(y = 1 \mid x; \theta) + P(y = 0 \mid x; \theta) = 1 \Rightarrow P(y = 0 \mid x; \theta) = 1 - P(y = 1 \mid x; \theta)$$

En resumen, se tendrá los siguiente:

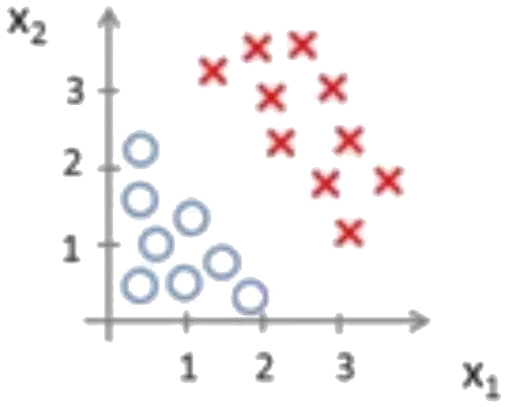
— Se podrá suponer que se predice "y = 1" si $h_\theta(x) \geq 0{,}5 \Rightarrow g(z) \geq 0{,}5$ cuando $z \geq 0$

$h_\theta(x) = g(\theta^T x) \geq 0{,}5$ cuando $\theta^T x \geq 0$

— Se podrá suponer que se predice "y = 0" si $h_\theta(x) < 0 \Rightarrow g(z) < 0$ cuando $z < 0$

$h_\theta(x) = g(\theta^T x) < 0{,}5$ cuando $\theta^T x < 0$

Supóngase que se tiene un conjunto de entrenamiento como el que se muestra en la figura siguiente, además de que la hipótesis tiene la expresión $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$.



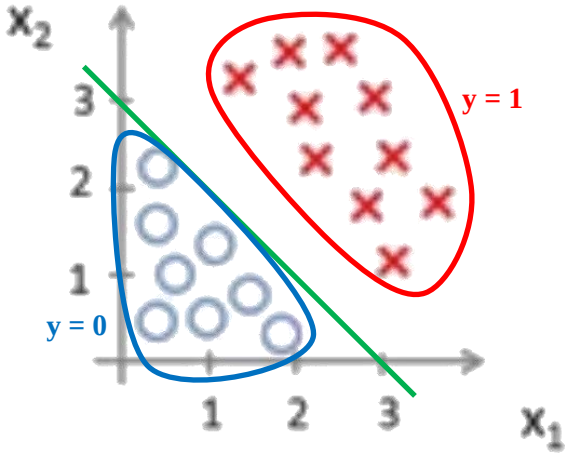
Todavía no se ha hablado sobre cómo ajustar los parámetros de este modelo, se tratará en el siguiente apartado. Pero se supone que, mediante un procedimiento especificado, se terminan eligiendo los valores siguientes para los parámetros θ.

$$\left.\begin{matrix} \theta_0 = -3 \\ \theta_1 = 1 \\ \theta_2 = 1 \end{matrix}\right\} \Rightarrow \theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

Cuando se les da esta opción a los parámetros de la hipótesis, se va a tratar de averiguar dónde la hipótesis terminaría la predicción de "y = 1" e "y = 0". Para lo cual se va a emplear lo visto en el apartado anterior, de tal manera que se tendrá lo siguiente.

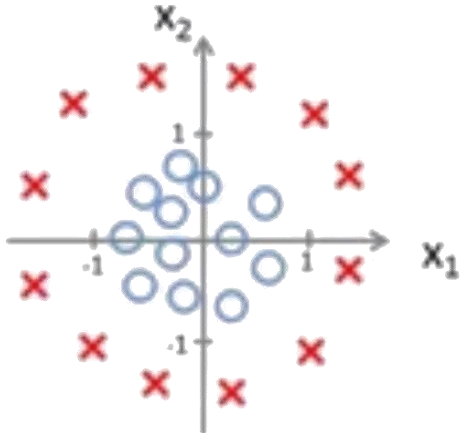
$$\text{Se predice "}y=1\text{" si } \theta^T x \geq 0 \Rightarrow \theta_0 + \theta_1 x_1 + \theta_2 x_2 \geq 0 \Rightarrow -3 + x_1 + x_2 \geq 0 \Rightarrow x_1 + x_2 \geq 3$$

Dicha expresión define la ecuación de una recta y si se dibuja, da la siguiente línea que pasa por el punto 3 tanto en el eje de x_1 como en el de x_2. Así la parte del plano que corresponde a $x_1 + x_2 \geq 3$ va a ser la parte superior derecha a partir de la línea dibujada, siendo la región donde la hipótesis predice "y = 1". En contraste, la región donde $x_1 + x_2 < 3$, es donde la hipótesis predecirá "y = 0".



A esta línea azul que se ha dibujado, se le puede dar un nombre, denominándose límite de decisión. Como se puede observar dicha línea separa la región donde la hipótesis predice "y = 1" de la región donde la hipótesis predice "y = 0". Hay que tener en cuenta que el límite de decisión es una propiedad de la hipótesis y no del conjunto de datos.

A continuación, se presenta un ejemplo más complejo donde como de costumbre, se disponen de cruces para denotar los ejemplos positivos y de círculos para denotar los ejemplos negativos. Dado un conjunto de entrenamiento como este, ¿cómo se puede obtener la regresión logística para ajustarse al tipo de datos? Antes, cuando se hablaba de regresión polinómica o cuando se hablaba de regresión lineal, se comentó cómo se podrían añadir términos polinómicos de orden superior a las características, pudiéndose hacer lo mismo para la regresión logística.



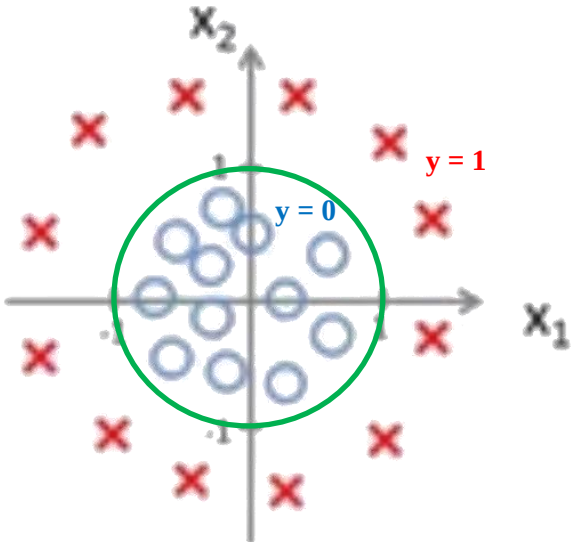
Concretamente, se puede ver la hipótesis como se muestra, donde se ha añadido dos características adicionales, x_1 al cuadrado y x_2 al cuadrado, a las características iniciales. Así que ahora se tienen cinco parámetros para θ, de θ_0 a θ_4. Sin entrar en el procedimiento de la elección de dichos parámetros, ya que se verá a posteriori, seguidamente se refleja el valor de estos parámetros.

$$h_\theta(x) = g\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2\right)$$

$$\left.\begin{array}{l}\theta_0 = -1\\ \theta_1 = 0\\ \theta_2 = 0\\ \theta_2 = 1\\ \theta_2 = 1\end{array}\right\} \Rightarrow \theta = \begin{bmatrix}-1\\0\\0\\1\\1\end{bmatrix}$$

$$\text{Se predice } y = 1 \text{ si } \theta^T x \geq 0 \Rightarrow -1 + x_1^2 + x_2^2 \geq 0 \Rightarrow x_1^2 + x_2^2 \geq 1$$

Entonces, ¿qué aspecto tiene esta decisión? Se puede observar que dicha expresión es la ecuación de un círculo de radio uno, centrado en el origen, tal como se muestra en la figura siguiente.



Así que este es el límite de decisión para este ejemplo. Todo fuera del círculo, se va a predecir cómo "y = 1" y lo que se encuentre dentro del círculo se predecirá como "y = 0". Así que, agregando términos polinómicos, se pueden obtener límites de decisión más complejos, que no sólo traten de separar los ejemplos positivos y negativos en una línea recta.

4.10 CLASIFICACIÓN MULTICLASE (REGRESIÓN LOGÍSTICA MULTINOMIAL).

Uno de los problemas, que posiblemente sea el más común que se dé, vendrá relacionado con cómo lograr que la regresión logística funcione para problemas de clasificación multiclase, para lo cual se trabajará con un algoritmo denominado clasificación "uno contra todos".

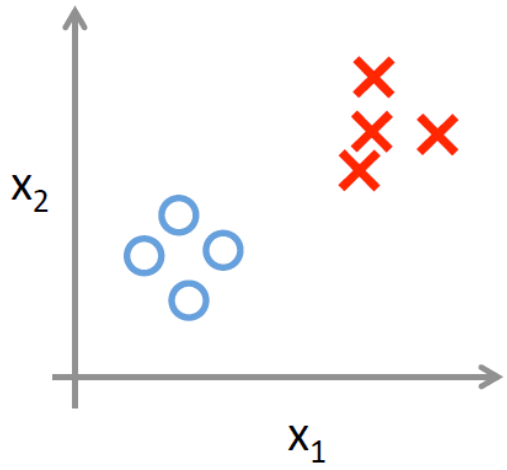
La pregunta que se plantea es ¿Qué es un problema de clasificación multiclase? He aquí algunos ejemplos. Supóngase que se quiere que un algoritmo de aprendizaje coloque automáticamente el correo

electrónico en carpetas diferentes o etiquete automáticamente los correos electrónicos. Entonces, posiblemente se tendrá diferentes carpetas o diferentes etiquetas para el correo electrónico del trabajo, correo electrónico de los amigos, correo electrónico de la familia y correos sobre el tiempo libre. Así pues, aquí se tiene un problema de clasificación con cuatro clases, a las que se les podría asignar los números, es decir, las clases "y = 1", "y = 2", "y = 3" e "y = 4".

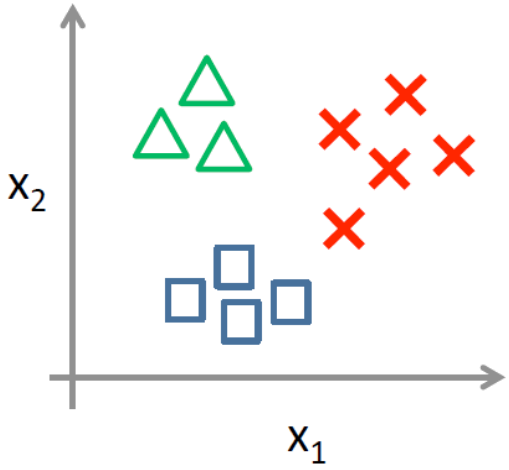
Otro ejemplo, podría ser el de un diagnóstico médico, si un paciente llega a la consulta con tal vez la nariz congestionada, los posibles diagnósticos podrían ser que no está enfermo, tal vez eso es "y = 1"; o que tiene un resfriado, "y = 2"; o que tiene gripe, "y = 3".

Por cierto, en realidad no importa si se indexa ya sea como "0, 1, 2, y 3" o como "1, 2, 3 y 4", aunque lo más normal es indexar las clases a partir del "1" en lugar de empezar desde "0". Pero, de cualquier manera, no suele importa mucho.

Mientras que anteriormente, para un problema de clasificación binaria, el conjunto de datos se veía como se muestra en la figura siguiente.



Para un problema de clasificación multiclase, el conjunto de datos puede verse como se muestra en la figura siguiente, donde se está usando tres símbolos diferentes para representar cada una de las tres clases.



Entonces, la pregunta que hay que hacerse es dado un conjunto de datos con varias clases, ¿cómo se obtiene un algoritmo de aprendizaje que trabaje para este escenario? Ya se sabe cómo hacer la clasificación binaria, utilizando la regresión logística, de tal manera que, por medio de una línea recta, se pueden separar los valores positivas y negativas. Usando la denominada clasificación "uno contra todos", se pueden resolver problemas de clasificación multiclase. A veces también se le suele denominar "uno contra el resto."

Si se tiene un conjunto de entrenamiento, como se muestra a la izquierda de la figura siguiente, donde se tienen tres clases. Así que, si "y = 1", se denota con un triángulo, si "y = 2" con un cuadrado y si "y = 3" con una cruz. Lo que se va a hacer es, tomar un conjunto de entrenamiento, y convertir el problema inicial en tres problemas de clasificación binaria separados.

Se va a empezar con la clase 1, que es un triángulo. Se crea esencialmente un nuevo conjunto de entrenamiento "falso", en donde las clases 2 y 3 quedan asignadas a la clase negativa y la clase 1 queda asignada a la clase positiva. Cuando se crea un nuevo conjunto de entrenamiento, como el que se muestra a la derecha de la figura siguiente, se va a ajustar un clasificador, que va a ser denominado como $h_\theta^{(1)}(x)$, en donde, los triángulos son los ejemplos positivos y los círculos son los ejemplos negativos. Por lo tanto, a los triángulos se les ha asignado el valor "1" y a los círculos, suma de cuadrados y cruces, el valor de "0". A continuación, se hace lo mismo para la clase 2 y 3, ajustándose los clasificadores $h_\theta^{(2)}(x)$ y $h_\theta^{(3)}(x)$.

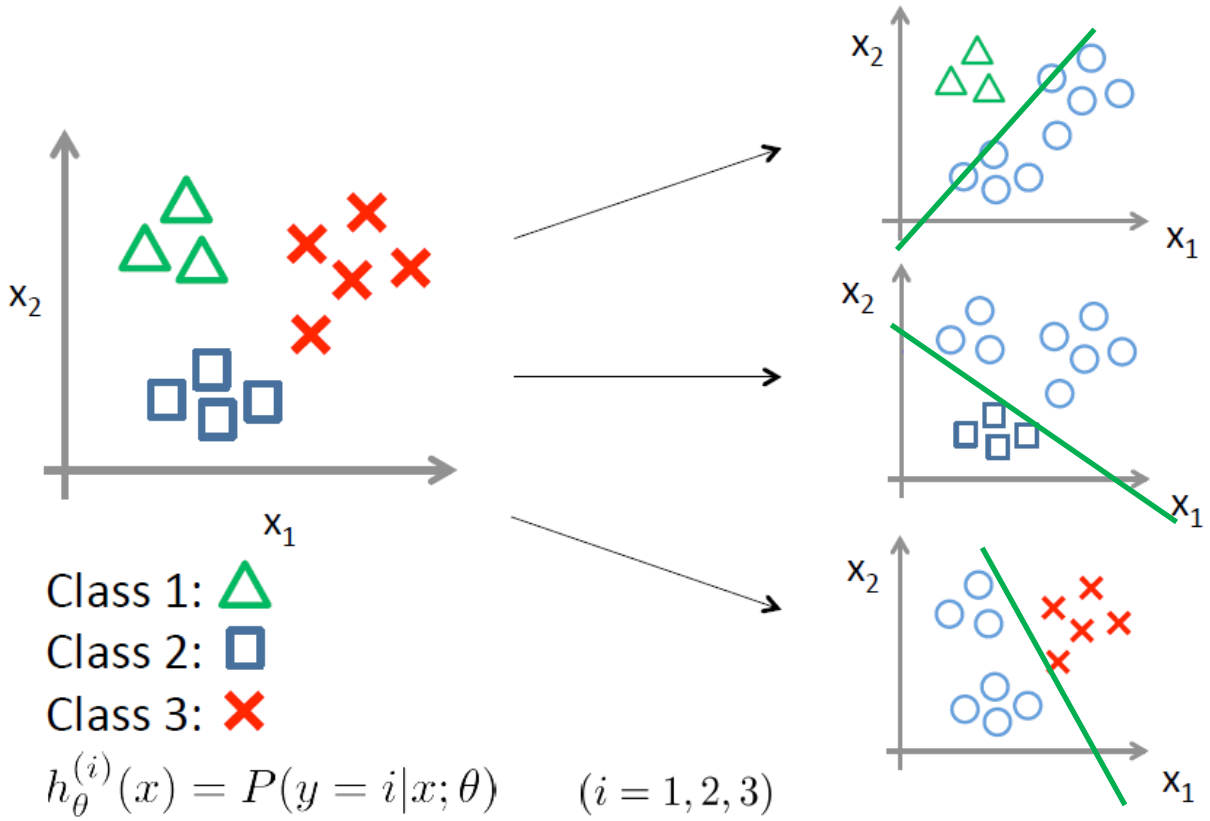
Así, en la primera instancia, el clasificador va aprendiendo a identificar los triángulos, es decir, piensa en los triángulos como una clase positiva, por lo que $h_\theta^{(1)}(x)$ está esencialmente tratando de calcular cuál es la probabilidad de que $P\left(y=1 \middle| x;\theta\right)$. Y lo mismo está sucediendo para los cuadrados y las cruces. Por lo que ahora se tienen tres clasificadores, cada uno de los cuales fue entrenado para identificar una de las tres clases.

Todo esto se puede expresar de forma generalizada a través de la expresión siguiente.

$h_\theta^{(i)}(x) = P\left(y=1 \middle| x;\theta\right)$; siendo $i = 1, 2, \cdots, n$

En resumen, lo que se hace es entrenar un clasificador de regresión logística $h_\theta^{(i)}(x)$ para cada clase de "i" para predecir la probabilidad de que "y = i".

Por último, para hacer una predicción cuando se ha agregado una nueva entrada "x", lo que se hace es simplemente ejecutar los tres clasificadores para la entrada "x" y después tomamos la clase "i" que maximiza a los tres, es decir, $\max_i h_\theta^{(i)}(x)$. Así que, básicamente se elige el clasificador que sea más confiable, o que de forma más entusiasta diga que piensa que tiene una clase correcta. En definitiva, cualquier valor de "i" que, de la mayor probabilidad, entonces se predice que "y" tendrá ese valor.



4.11 CUESTIONES.

4.11.1 CUESTIÓN 1.

4.11.1.1 INTRODUCCIÓN.

Esta cuestión se basa en la resolución de un ejercicio, en el que implementará una regresión lineal con una variable, empleándose Matlab, para predecir los beneficios por ventas de una empresa.

Supóngase que somos es el CEO de una empresa que fábrica tarjetas electrónicas para el control de motores de c.c. y está considerando en modificar, una vez más su producción debido a las demandas fluctuantes del mercado. La empresa dispone de históricos y bases de datos que relacionan los beneficios obtenidos en función del número de unidades de tarjetas vendidas.

Se deberán utilizar estos datos para ayudar al CEO a determinar el beneficio que se obtendrá, a partir del número estimado de ventas de tarjetas, para que, de esta forma, el departamento financiero de la empresa desarrolle el plan futuro de viabilidad de la empresa.

Para ello se proporciona un archivo denominado cuestion41_data.txt que contiene el conjunto de datos de entrenamiento para dicho problema de regresión lineal, como ya se ha comentado, se trata de la base de datos que relacionan los beneficios obtenidos, en tiempos pasados, en función del número de ventas del producto. La primera columna es el número de tarjetas vendidas y la segunda columna es el beneficio obtenido debido a esas ventas. Un valor negativo de dicho beneficio indicaría la entrada en pérdidas para la empresa.

4.11.1.2 GRÁFICO DE DISPERSIÓN PARA LOS DATOS.

Antes de comenzar cualquier tarea, a menudo es útil comprender los datos, visualizándolos. Para este conjunto de datos, se puede utilizar un diagrama de dispersión, ya que sólo tiene dos variables, el beneficio y las ventas, hay que comentar que otros problemas que se dan en la vida real son multidimensionales y no será posible trazar un gráfico 2D.

Se indicará en el guion a entregar de la práctica el código generado y su resultado al ser ejecutado, es decir, en este caso será el correspondiente gráfico de dispersión y = f(x).

4.11.1.3 OBTENCIÓN DE LA RECTA DE REGRESIÓN LINEAL.

Se obtendrá la recta de regresión lineal mediante el empleo en Matlab, de los comandos siguientes:

- "Polyfit" y "polyval".
- "Fitlm".
- App denominada "curve fitting tool".

Se indicará en el guion a entregar de la práctica, el código generado, la expresión matemática de la recta de regresión lineal obtenida para cada caso, que tiene que ser coincidente, y el gráfico resultante de la regresión lineal, dibujando dicha recta sobre el gráfico de dispersión de los datos.

4.11.1.4 PREDICCIÓN.

Una vez determinada la expresión de la regresión lineal, se tendrá que determinar la predicción del beneficio que aportará una venta de 35000 y 70000 unidades de tarjetas electrónicas.

Se indicará en el guion a entregar de la práctica, el código generado y la predicción del beneficio obtenido para las dos cantidades indicadas.

4.11.2 CUESTIÓN 2.

4.11.2.1 INTRODUCCIÓN.

Esta cuestión se basa en la resolución de un ejercicio, en el que se implementará una regresión lineal con múltiples variables, empleándose Matlab, para predecir las instrucciones por segundo que podrá ejecutar un microprocesador de bajo coste.

El equipo técnico de una compañía que fabrica microprocesadores de bajo coste necesita conocer las instrucciones que podrá ejecutar un microprocesador en un segundo antes de ser fabricado, para analizar si es necesario seguir con la misma tecnología de fabricación o cambiar a otra más novedosa. Una forma de hacerlo sería disponer de una base de datos con información recopilada de todos los modelos que ha fabricado hasta la fecha y hacer un modelo para predecir el número de instrucciones por segundo que puede ejecutar un microprocesador.

El fichero cuestion42_data.txt contiene la información necesaria para el estudio a realizar. La primera columna es la frecuencia del microprocesador en kHz, la segunda columna es el número de núcleos que tiene y la tercera columna es el número de instrucciones que puede ejecutar en un segundo.

4.11.2.2 GRÁFICO DE DISPERSIÓN PARA LOS DATOS.

Antes de comenzar cualquier tarea, a menudo es útil comprender los datos, visualizándolos. Para este conjunto de datos, se puede utilizar un diagrama de dispersión en 3D, para lo cual se empleará el comando "scatter3".

Se indicará en el guion a entregar de la práctica el código generado y su resultado al ser ejecutado, es decir, en este caso será el correspondiente gráfico de dispersión 3D.

4.11.2.3 OBTENCIÓN DE LA RECTA DE REGRESIÓN LINEAL MULTIVARIABLE.

Se obtendrá la recta de regresión lineal mediante el empleo en Matlab, de los comandos siguientes:

- "Readtable" y "fitlm".
- Ecuación normal: Se aplicará la metodología basada en la ecuación normal para determinar los parámetros de θ (coeficientes de la regresión), para lo cual se tendrá que implementar el código que defina la expresión $\theta = \left(X^T X \right)^{-1} X^T \bar{y}$, donde X será una matriz en cuya primera columna estará el intercepto o termino independiente, que tendrá que ser todo unos, la segunda columna será los valores de la frecuencia y la tercera el número de núcleos. Además, para generar la matriz traspuesta se emplea el comando "X'" y para la inversa "pinv"
- "Regress".

Dibujar el plano de regresión sobre el gráfico de dispersión de los datos, para lo cual se podrá empelar el código siguiente.

```
scatter3(x1,x2,y,'filled')
title('Gráfico de dispersión');
xlabel('Frecuencias (kHZ)');
ylabel('Número de núcleos');
zlabel('Instrucciones/s');
hold on
[X1FIT,X2FIT]=meshgrid(x1,x2);
coeficientes=table2array(mdl.Coefficients);
YFIT=coeficientes(1,1)+coeficientes(2,1)*X1FIT+coeficientes(3,1)*X2FIT;
mesh(X1FIT,X2FIT,YFIT);
hold off
```

Se indicará en el guion a entregar de la práctica, el código generado, la expresión matemática de la recta de regresión lineal multivariable obtenida para cada caso, que tiene que ser coincidente, y el plano de regresión.

4.11.2.4 PREDICCIÓN.

Una vez determinada la expresión de la regresión lineal multivariable, se tendrá que determinar la predicción del número de instrucciones por segundo que podría ejecutar un microprocesador de una frecuencia de 1650 kHz y 3 núcleos.

Se indicará en el guion a entregar de la práctica, el código generado y la predicción del número de instrucciones por segundo obtenido.

4.11.2.5 APP DE MATLAB.

Determinar si existe una app de Matlab, que pueda resolver el problema planteado, si es así indicar cual es, como se procede para trabajar con ella y comparar los resultados con los anteriormente obtenidos.

4.11.3 CUESTIÓN 3.

4.11.3.1 INTRODUCCIÓN.

Se implementará una regresión polinomial para predecir la cantidad de agua que sale por la presa de un embalse en función del cambio de nivel de agua que se produce en dicho embalse.

Al cargar el fichero cuestion43_data.mat en el entorno de trabajo de Matlab, se dispondrá de los conjuntos de datos siguientes, generados de forma aleatoria del total de datos disponibles:

- Un conjunto de entrenamiento como base para la obtención de varios modelos: “xtrain” e “ytrain”.
- Un conjunto de validación cruzada para evaluar los modelos anteriores y determinar el mejor, “x_{val}”, e “y_{val}”.
- Un conjunto de prueba o test para evaluar el rendimiento del modelo obtenido. Estos son ejemplos "invisibles" que el modelo no vio durante el entrenamiento: “x_{test}” e “y_{test}”.

4.11.3.2 VISUALIZACIÓN DE LOS DATOS.

Lo primero será realizar un gráfico de dispersión, que incluya los tres conjuntos de datos a la vez, pero que se pueda distinguir unos de otros, de alguna manera, para de esta manera comprobar su distribución a lo largo de todo el espectro de los datos tomados, observándose que no ha habido ningún sesgo al generar los tres conjuntos de datos, si no que ha sido de forma aleatoria.

Se indicará en el guion a entregar de la práctica el código generado y su resultado al ser ejecutado, es decir, en este caso será el correspondiente gráfico de dispersión.

4.11.3.3 REGRESIÓN LINEAL.

Trabajando únicamente con los datos del conjunto de entrenamiento, determinar la recta de regresión lineal, como primer paso a la obtención de un modelo explicativo del comportamiento de los datos, empleándose los comandos "polyfitn" y "polyvaln". Son dos funciones hechas a medida por John D'Errico que tienen como base "polyfit" y "polyval" de Matlab pero que dan mucha mayor información del modelo generado, que estos últimos.

Seguidamente se trazará la línea de ajuste de la regresión, sobre el diagrama de dispersión de los datos de entrenamiento, observándose que el modelo obtenido no es un buen ajuste a los datos, debido, a como se podrá observar, los datos siguen un patrón no lineal.

Se indicará en el guion a entregar de la práctica, el código generado, la expresión matemática de la recta de regresión lineal obtenida y el gráfico resultante solicitado.

4.11.3.4 MODELOS DE REGRESIÓN POLINOMIAL.

El modelo obtenido mediante la regresión lineal resulto ser demasiado simple para el conjunto de datos con el que se realizó el estudio, obteniéndose un sesgo alto y por lo tanto un subajuste. Por lo tanto, para mejorar el modelo lo que habrá que hacer es agregar más funciones, lo que dará lugar a la denominada regresión polinomial.

Para poderse utilizar la regresión polinomial, la hipótesis tendrá la forma siguiente.

$$h\theta(x) = \theta_0 + \theta_1 \cdot (\text{nivel de agua}) + \theta_2 \cdot (\text{nivel de agua})^2 + \cdots + \theta_p \cdot (\text{nivel de agua})^p$$

No obstante, se definen las siguientes características,

$$x_1 = (\text{nivel de agua})$$

$$x_2 = (\text{nivel de agua})^2$$

.....

$$x_p = (\text{nivel de agua})^p$$

Se obtendrá un modelo de regresión lineal donde las características son las potencias del valor original "nivel de agua", como el siguiente.

$$h\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \cdots + \theta_p \cdot x_p$$

Se tendrá que definir nueve modelos más, es decir hasta llegar a x^{10}, empleándose los comandos ya indicados de "polyfitn" y "polyvaln".

Seguidamente se trazará la línea de ajuste de la regresión, sobre el diagrama de dispersión de los datos de entrenamiento para cada modelo, observándose que el modelo obtenido es más o menos un buen ajuste a los datos, debido, a como se podrá observar, los datos siguen un patrón no lineal.

Se indicará en el guion a entregar de la práctica, el código generado, la expresión matemática de las regresiones polinomiales obtenidas y el gráfico resultante solicitado.

4.11.3.5 ELECCIÓN DEL MODELO DE LA REGRESIÓN POLINOMINAL.

Una vez obtenido los diez modelos de regresión polinomial, el de grado 1 se considerará también, aunque se sepa que regresión lineal no funciona bien, se tendrá que evaluar cuál de ellos es el mejor ante la presencia de una serie de datos, que no han sido visto anteriormente por dichos modelos generados, para lo cual se trabajara con el conjunto de datos de validación cruzada, “xval” e “yval”.

Se trazará la línea de ajuste de la regresión, sobre el diagrama de dispersión de los datos de validación cruzada para cada modelo y se calculará el error cometido a partir del coeficiente de determinación, R^2, pudiéndose emplear la función “calculateR2” realizada por Shoaibur Rahman.

Se indicará en el guion a entregar de la práctica, el código generado, los gráficos resultantes y se indicará cuál de los diez modelos polinomiales es el mejor.

4.11.3.6 DESEMPEÑO DE LA REGRESIÓN POLINOMIAL.

Una vez obtenido el modelo de regresión polinomial que mejor comportamiento tiene ante el conjunto de datos de validación, para poder conocer su desempeño definitivo se tendrá que presentar a un nuevo conjunto de datos, el conjunto de datos de test, “xtest” e “ytest”.

El procedimiento a seguir será el mismo que se habrá realizado en el apartado anterior pero únicamente para el mejor modelo obtenido y los datos del conjunto de test.

Se indicará en el guion a entregar de la práctica, las conclusiones obtenidas.

4.11.3.7 APP DE MATLAB.

Determinar si existe una app de Matlab, que pueda resolver el problema planteado, si es así indicar cual es, como se procede para trabajar con ella y comparar los resultados con los anteriormente obtenidos.

4.11.4 CUESTIÓN 4.

4.11.4.1 INTRODUCCIÓN.

Se implementará una regresión logística para predecir si una máquina fallará en un tiempo inmediato, en base al estado de dos variables que se han controlado durante un largo tiempo y de las cuales depende su correcto funcionamiento, que han dado lugar a una base de datos recogida en el fichero cuestion44_data.xlsx.

Por lo tanto, las variables independientes serán la humedad relativa y la presión atmosférica, y la dependiente de las anteriores, será el fallo o no de la máquina en cuestión.

4.11.4.2 VISUALIZACIÓN DE LOS DATOS.

Lo primero que habrá que hacer es cargar los datos de la hoja de excel y generar la matriz de contendrá las variables de entrada, "x", formada por la humedad relativa, "x1", y la presión atmosférica, "x2", y el vector de la variable de salida, indicativo del fallo o no de la máquina, "y".

La variable "y", tendrá valores de 0 y 1, pero resulta que habrá que convertirlos en 1 y 2, porque el comando de Matlab, con el que se va a trabajar a posteriori, no admite ceros, bastará con hacer la operación "y = y + 1".

Se tendrá que realizar los histogramas correspondientes a las tres variables, para ver su distribución. Así como, el grafico de dispersión que relacione las variables de entrada.

Se indicará en el guion a entregar de la práctica, el código y los gráficos generados.

4.11.4.3 VISUALIZACIÓN DE LAS CLASE.

Con el comando "gscatter" se generará un gráfico de dispersión, en el que aparecerán clasificados los sucesos, en diferentes colores, según se trata de fallo o no fallo, de esta forma se podrá apreciar su distribución.

Se indicará en el guion a entregar de la práctica, el código y el gráfico generado.

4.11.4.4 ENTRENAMIENTO DE LA REGRESIÓN LOGÍSTICA.

Para llevarse a cabo el entrenamiento, será necesario dividir el conjunto de datos en otros dos, el de entramiento y el de prueba o test, el primero de ellos dedicado a la obtención del modelo y el segundo para valorar el desempeño, en este caso se va a prescindir del conjunto de datos de validación cruzada, debido a los buenos resultados que se obtendrá, tal como se podrá observar. Algo muy extendido es que el conjunto de entrenamiento este formado por el 70 % de los datos totales y el resto para el de prueba, de igual modo si existiese conjunto de validación la proporción sería 60, 20 y 20 %, siempre el porcentaje del conjunto de entrenamiento, como es lógico, tendrá que ser muy superior a los otros dos. Otra condicionante a tener en cuenta es que estos conjuntos se tendrán que formas de manera aleatoria, para lo cual el código a emplear podrá ser el siguiente.

```
P=0.7; % 70 % del conjunto de datos para entrenamiento
idx=randperm(m);
xtrain=x(idx(1:round(P*m)),:);
ytrain=y(idx(1:round(P*m)),:);
xtest=x(idx(round(P*m)+1:end),:);
ytest=y(idx(round(P*m)+1:end),:);
```

Una vez que se tengan los dos conjuntos, en primera instancia se trabajará con el de entrenamiento para obtener los valores óptimos de θ, a este proceso se le conoce propiamente como entrenamiento, que formen parte de la expresión siguiente, empleándose el comando "mnrfit".

$$h_\theta(x) = \frac{1}{1 - e^{-\theta^T x}}$$

Se indicará en el guion a entregar de la práctica, el código generado y el resultado del vector theta, que contiene los valores óptimos de θ.

4.11.4.5 GENERACIÓN DE LA HIPOTESIS PARA EL CONJUNTO DE DATOS DE ENTRENAMIENTO Y PRUEBA.

A partir de la expresión de la hipótesis, reflejada en el apartado anterior, tendrá que ser calculada para los dos conjuntos de datos, teniéndose en cuanta que a los conjuntos de datos les falta la variable que acompañara al termino independiente, que obviamente será "1", para que entre en juego el coeficiente theta0, para ello a las matrices "xtrain" y "xtest", habrá que añadirles una columna entera de unos, que deberá ser la primera, para lo cual se puede emplear el código "xtrain2=[ones(mtrain,1) xtrain];", se la ha denominada "xtrain2" por el simple hecho de diferenciarla de la original.

Una vez generadas las respetivas hipótesis, para ver su distribución, será interesante generar sus correspondientes histogramas.

Se indicará en el guion a entregar de la práctica, el código y los gráficos generados.

4.11.4.5 VISUALIZACIÓN DE LA SALIDA DEL MODELO DE CLASIFICACIÓN.

De manera gráfica se tendrá que comprobar el comportamiento y distribución de la hipótesis del modelo frente a los datos del conjunto de prueba, datos que el modelo los ve por primera vez, para lo cual se generará el correspondiente gráfico de dispersión, para lo cual se empleará el código siguiente.

```
scatter(xtest(:,1),xtest(:,2),50,htest);
cb=colorbar();
```

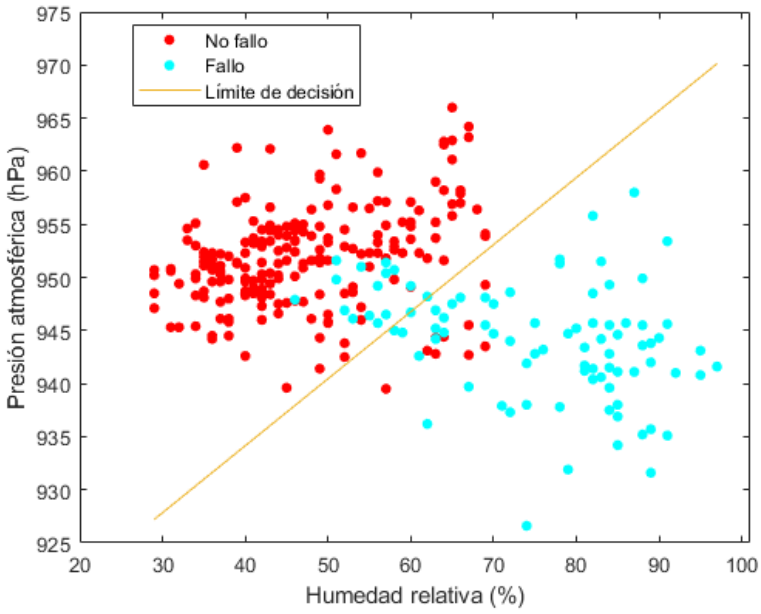
Se indicará en el guion a entregar de la práctica, el gráfico generado.

4.11.4.6 VISUALIZACIÓN DEL LÍMITE DE DECISIÓN.

El límite de decisión será una línea recta que divide al conjunto total de datos en dos zonas, intentando separar el mayor número de datos que forman las dos clases existentes, fallo y no fallo, para lo cual se tendrá que trabajar con la totalidad de los datos, empleándose el comando "gscatter" para generar el grafico de dispersión donde aparecen reflejadas las clases y para dibujar el límite de decisión se tendrá que dibujar la función siguiente.

```
theta0*1 + theta1*x1 + theta2*x2 = 0
theta(1)*1 + theta(2)*x1 + theta(3)*x2 = 0
x2 = -(theta(1)*1 + theta(2)*x1)/theta(3)
```

A modo de comprobación, si todo el proceso se ha realizado correctamente, el grafico resultante será como el representado en la figura siguiente.



Se indicará en el guion a entregar de la práctica, el código y el gráfico generado.

4.11.4.7 EVALUACIÓN DEL MODELO.

Es muy necesario, desde el punto de vista del análisis de errores tener una métrica, es decir, disponer de una métrica de evaluación con un número real, para saber cómo es el desempeño de un algoritmo de aprendizaje.

En el contexto de la evaluación y de la métrica de errores, hay un caso importante en el que es especialmente difícil obtener una métrica de errores o de evaluación apropiada para el algoritmo de aprendizaje, denominado clases sesgadas.

En un problema de calificación se tendrían clases sesgadas, cuando la proporción de ejemplos positivos y negativos está muy cerca a alguno de los extremos, por ejemplo, que el número de ejemplos positivos es mucho más pequeño que el número de ejemplos negativos o viceversa.

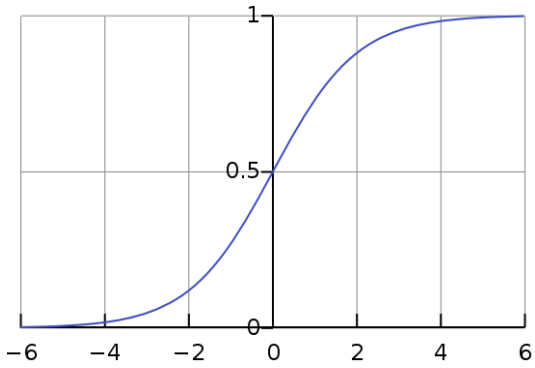
Si este no es el caso, una primera métrica de evaluación, consistiría en determinar el porcentaje de aciertos que tiene el modelo, para lo cual hay que tener en consideración cuando la hipótesis se considera "0" o ª1", lo estandarizado es lo reflejado en la expresión siguiente, en base al desarrollo de la función sigmoide, figura siguiente.

- Se podrá suponer que se predice "y = 1" si $h_\theta(x) \geq 0{,}5 \Rightarrow g(z) \geq 0{,}5$ cuando $z \geq 0$

  $h_\theta(x) = g\left(\theta^T x\right) \geq 0{,}5$ cuando $\theta^T x \geq 0$

- Se podrá suponer que se predice "y = 0" si $h_\theta(x) < 0 \Rightarrow g(z) < 0$ cuando $z < 0$

  $h_\theta(x) = g\left(\theta^T x\right) < 0{,}5$ cuando $\theta^T x < 0$



No obstante, si se quiere estar más en la certeza de que, es este caso exista fallo, el parámetro del 0,5 podría cambiarse por 0,7, por ejemplo, esto tiene sus ventajas y sus inconvenientes, porque igual nos estamos dejando por el camino fallos.

Este proceso se realizará tanto para el conjunto de entrenamiento como para el conjunto de prueba, pudiéndose observar que el valor obtenido para el porcentaje de aciertos, en el conjunto de entrenamiento es mayor, que en el conjunto de prueba, como es lógico. El código a emplear para el conjunto de entrenamiento será el siguiente.

```
ytrainpred=htrain<0.5; % para asegurar más se podía comparar con 0,7 pero no se va a hacer
ytrainpred=ytrainpred+1;
gscatter(xtrain(:,1),xtrain(:,2),ytrainpred);
exactitud_training=mean(double(ytrainpred==ytrain))*100
```

Otra de esas métricas de evaluación es la denominada matriz de confusión. Para explicarla, se partirá de la evaluación de un clasificador binario, “0” o “1”, a través del conjunto de prueba. Se dibujará una tabla de dos por dos, como la que aparece en la figura siguiente, dependiendo de la clase real y la clase predicha.

		Actual Class: p	Actual Class: n
Predicted Class	Y	True Positives	False Positives
	N	False Negatives	True Negatives

Si se tiene un ejemplo en el que la clase real es uno y la clase predicha es uno, entonces se le denomina un ejemplo positivo verdadero. Esto quiere decir que el algoritmo predijo que es positivo y de hecho el ejemplo es positivo.

Si el algoritmo predijo que algo es negativo, clase cero, y la clase real es cero, entonces a esto se le denomina negativo verdadero. Se predijo que sería cero y es realmente cero.

En las otras dos celdas, si el algoritmo predice que la clase es uno, pero la clase real es cero, entonces se le denomina un falso positivo. Esto quiere decir que el algoritmo pensó que el paciente tiene cáncer cuando en realidad no lo padece.

Finalmente, la última casilla es cero-uno. A esto se le llama falso negativo porque nuestro algoritmo predijo un cero, pero la clase real fue de uno.

A partir de la matriz de confusión se podrán calcular dos índices denominados precisión y recuperación, con los que se podrá evaluar el desempeño de un algoritmo.

Partiendo de un ejemplo bastante empleado, en la docencia de estos temas, se trata de los pacientes con cáncer, para que se entienda mejor, la precisión indica de todos los pacientes a los que se les dijo “creemos que usted puede tener cáncer”, ¿qué fracción tienen cáncer realmente?, viniendo dada por la expresión siguiente.

$$\text{Precisión} = \frac{\text{Verdaderos positivos}}{\text{Predicciones positivas}} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos positivos}}$$

Como puede verse, una precisión alta es deseable. Lo quiere decir que, de los pacientes a quienes se les dijo “creemos que tiene cáncer, lo sentimos mucho” la mayoría realmente tienen cáncer; por lo tanto, las predicciones realizadas fueron precisas.

La recuperación indica, de todos los pacientes del conjunto de prueba que realmente tienen cáncer ¿en qué fracción de ellos se detectó el cáncer correctamente?, viviendo dada por la expresión siguiente.

$$\text{Recuperación} = \frac{\text{Verdaderos positivos}}{\text{Reales positivos}} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos negativos}}$$

Una vez más, tener una recuperación alta será bueno.

Llegado este punto, lo que interesa es compensar, de alguna manera, la precisión y la recuperación. Si se continua con el ejemplo de clasificación de los enfermos de cáncer, en el que “y = 1” si el paciente tiene cáncer e “y = 0” si el paciente está sano, se partirá del estudio de una regresión logística dada por la expresión siguiente.

$$0 \le h\theta(x) \le 1 \begin{cases} \text{Se predice "1" si } h\theta(x) \ge 0{,}5 \\ \text{Se predice "0" si } h\theta(x) < 0{,}5 \end{cases}$$

Supóngase que se quiere predecir que un paciente tiene cáncer, sólo si se está seguro de que lo padezca, una manera de hacer esto, es modificar el algoritmo para que, en vez de fijar el umbral en 0,5, es ponerlo en 0,7, entonces sólo se dirá a un enfermo que padece cáncer, si hay una probabilidad igual o mayor al 70 % de que lo padezca. Esto se refleja en la expresión siguiente.

$$0 \le h\theta(x) \le 1 \begin{cases} \text{Se predice "1" si } h\theta(x) \ge 0{,}7 \\ \text{Se predice "0" si } h\theta(x) < 0{,}7 \end{cases}$$

Si se hace esto, se puede predecir el cáncer sólo cuando se esté seguro de ello, y, por lo tanto, se tendrá un clasificador con una precisión muy alta, pero al realizar las predicciones se deja un margen de seguridad muy cerrado y el clasificador tendrá una recuperación muy baja, porque ahora se hará una predicción de "y = 1" en un número menor de pacientes.

Esto se puede llevar más lejos, en vez de fijar el umbral de 0,7; se puede subir a 0,9 y predecir qué "y = 1" sólo si se tiene más del 90 % de seguridad de que el paciente posee cáncer. Una fracción mayor de pacientes tendrá cáncer, por lo tanto, será un clasificador con predicción más alta y recuperación más baja. En este caso, se detecta correctamente sólo a los pacientes que tengan cáncer. Esto se refleja en la expresión siguiente.

$$0 \le h\theta(x) \le 1 \begin{cases} \text{Se predice "1" si } h\theta(x) \ge 0{,}9 \\ \text{Se predice "0" si } h\theta(x) < 0{,}9 \end{cases}$$

Ahora, se considera un ejemplo distinto. Supóngase que se quiere evitar obviar muchos casos de cáncer, se quieren evitar los falsos negativos. Ya que, si un paciente tiene cáncer, pero no se le dice que lo padece, puede tener consecuencias graves. Así que, cuando se tenga dudas, se quiere predecir que un paciente tiene cáncer para que, por lo menos, puedan investigar más y tratarse en caso de que realmente se padezca.

En este nuevo caso, en vez de poner un umbral de probabilidad alto, se bajará a 0,3; dando lugar a la expresión siguiente.

$$0 \le h\theta(x) \le 1 \begin{cases} \text{Se predice "1" si } h\theta(x) \ge 0{,}3 \\ \text{Se predice "0" si } h\theta(x) < 0{,}3 \end{cases}$$

Al hacerse esto, se está diciendo que, si hay más del 30 % de probabilidad de que el paciente tenga cáncer, se será conservador y se le dirá al paciente que puede tener cáncer para que busque el tratamiento adecuado. En este caso, se dispondrá de un clasificador con una recuperación más alta para marcar una fracción mayor de pacientes con cáncer, pero se seguirá teniendo una precisión menor, porque entre más grande sea la fracción de pacientes a los que se les diagnostique cáncer, mayor será la fracción de ellos que, finalmente, no lo padecen.

En resumen, es que dependiendo de si se quiere una precisión alta y una recuperación baja o, al contrario, se puede terminar prediciendo que "y = 1" cunado $h\theta(x)$ es mayor que un umbral.

Todo esto desemboca en unas preguntas interesantes ¿hay alguna manera de elegir el umbral automáticamente? O, si se tienen algoritmos diferentes ¿cómo comparar los diferentes valores de precisión y recuperación obtenidos?

Supóngase que se disponen de tres algoritmos de aprendizaje distintos o tal vez un único algoritmo con umbrales de tres valores distintos ¿cómo se decide cuál de estos algoritmos es el mejor?

	Precisión (P)	Recuperación (R)
Algoritmo 1	0,5	0,4
Algoritmo 2	0,7	0,1
Algoritmo 3	0,02	1

Se necesita obtener una métrica de evaluación de un número real, ya que hacer comparativas con los dos valores a la vez es muy complicado.

Lo primero que se puede intentar es calcular la media entre la precisión y la recuperación, viendo que clasificador tiene el promedio más alto, determinado a partir de la expresión siguiente.

$$\text{Media} = \frac{P + R}{2}$$

	Precisión (P)	Recuperación (R)	Media
Algoritmo 1	0,5	0,4	0,45
Algoritmo 2	0,7	0,1	0,4
Algoritmo 3	0,02	1	0,51

Como se puede observar esta solución no es buena, ya que si se tiene un clasificador que predice "y = 1" todo el tiempo (R = 1 en el algoritmo 3), se tendrá una recuperación muy alta y una precisión muy baja, que dará lugar a una media alta, como es el caso. Si se da el otro extremo, recuperación muy baja y precisión muy baja, sucederá lo mismo. Por lo que los dos extremos, según la media, darán un falsificador particularmente bueno, y eso no es cierto. Entonces trabajar con la media no es una buena solución.

No obstante, hay una manera distinta y efectiva de combinar la precisión y la recuperación, se trata del denominado valor F o también llamado F_1, que vendrá dado por la expresión siguiente.

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

	Precisión (P)	Recuperación (R)	Valor F
Algoritmo 1	0,5	0,4	0,444
Algoritmo 2	0,7	0,1	0,175
Algoritmo 3	0,02	1	0,039

Tendrá valores comprendidos entre 0 y 1 y se elegirán aquellos valores de precisión y recuperación cuyo valor F calculado sea mayor, en este ejemplo en cuestión, será el algoritmo 1.

Como se puede observar se cumplirá lo siguiente.

$P = 0 \text{ y } R = 0 \Rightarrow F = 0$

$P = 1 \text{ y } R = 1 \Rightarrow F = 1$

Hecha esta breve introducción se tendrá que determinar la precisión y la recuperación, para lo cual se empelará el código siguiente.

```
verdad_fallo=sum(double(ytest==2))
predice_fallo=sum(double(ytestpred==2))
verdadero_positivo=sum(double(ytest==2).*double(ytestpred==2))
precision=verdadero_positivo/predice_fallo
recuperacion=verdadero_positivo/verdad_fallo
```

Por otro lado, a partir de dichos indicadores se tendrá que calcular el valor F.

Se indicará en el guion a entregar de la práctica, el código y los gráficos generados, así como el valor de los indicadores obtenidos, sacando las pertinentes conclusiones.

4.11.4.8 EVALUACIÓN DE VARIOS MODELOS.

Comprobar como la elección aleatoria de los datos que componen el conjunto de entrenamiento afecta al modelo.

Para lo cual bastará con ejecutar el script, varias veces, con tres será suficiente, para de esta manera obtener tres modelos diferentes y en función del valor F, determinar cuál es el mejor.

Lo único, que habrá que tener en cuenta es que el conjunto de prueba tendrá que ser siempre el mismo, por lo tanto, lo que se podrá hacer es reservar un 15 % de los datos para dicho conjunto y del resto de los datos, 85 %, obtener el 70 %, realmente, es como si se tuviesen los tres conjuntos de datos, entrenamiento (70 %), validación cruzada (15 %) y prueba (15 %).

4.11.4.9 APP DE MATLAB.

Determinar si existe una app de Matlab, que pueda resolver el problema planteado, si es así indicar cual es, como se procede para trabajar con ella y comparar los resultados obtenidos en ambas metodologías.

4.11.5 CUESTIÓN 5.

Realmente la cuestión 4, era un ejemplo sencillo de aplicación de la regresión logística, para poder afrontar esta nueva cuestión, es decir, se va a predecir si una máquina falla en un tiempo inmediato, en base al estado de cuatro variables que se han controlado durante un largo tiempo y de las cuales depende su correcto funcionamiento, que han dado lugar a una base de datos recogida en el fichero cuestion45_data.xlsx.

Se pide, en base a lo realizado en la cuestión 4, llevar a cabo esta nueva cuestión, teniéndose en cuenta que algunas representaciones graficas no se podrán llevar a cabo, debido a que no estamos en 2D, es decir, se tienen más de dos variables.

4.11.6 CUESTIÓN 6.

En el fichero cuestión46_data.xlsx, se encuentran tres tipos de posibles fallos, que se pueden dar en una máquina, agrupados en tres clases (1, 2 y 3), en función del estado de cuatro variables (x1, x2, x3 y x4), es decir, se trata de un problema de clasificación multiclase o regresión logística multinomial.

En base al algoritmo denominado clasificación "uno contra todos", y teniendo como apoyo lo estudiado en las cuestiones 4 y 5, desarrollar el procedimiento para obtener el algoritmo que sea capaz de clasificar futuros fallos en dichas tres clases.

Se indicará en el guion a entregar de la práctica, el código y los gráficos generados, así como los resultados y conclusiones obtenidas.

Determinar si existe una app de Matlab, que pueda resolver el problema planteado, si es así indicar cual es, como se procede para trabajar con ella y comparar los resultados obtenidos en ambas metodologías.