Análisis de Datos Masivos para el Negocio

Tema 6. Técnicas de aprendizaje supervisado

Índice

Esquema

TÉCNICAS DE APRENDIZAJE SUPERVISADO

Aprendizaje supervisado

- Técnicas de aprendizaje automático para crear modelo basados en ejemplos. Los ejemplo son datos que han sido previamente etiquetados.
- El AS se utiliza principalmente para crear modelo de predicción y clasificación

TÉCNICAS DE PREDICCIÓN

Un modelo predictor toma una nueva observación de las **variables independientes** y proporciona una estimación del valor de las **variables dependientes**.

Medidas de la calidad de los modelos de predicción

- **MAE**, error absoluto medio.
- **RMSE**, raíz cuadrada del error cuadrático medio.
- **MBE, MSE, MAPE, nRMSE**.

Técnicas de predicción:

- Regresión lineal.
- Regresión basada en máquinas de soporte vectorial.
- Procesos Gaussianos.
- Redes Neuronales.

TÉCNICAS DE CLASIFICACIÓN

Un modelo clasificador toma una nueva observación de las **variables independientes** y proporciona una estimación de la clase a la que pertenece dicha observación.

Medidas de la calidad de los modelos de clasificación

- **True Positive (TP).** Número de aciertos en la categoría Positive.
- **True Negative (TN).** Número de aciertos en la categoría Positive.
- **False Positive (FP).** Número de observaciones clasificas Positive de forma errónea.
- **False Negative (FN).** Número de observaciones clasificas Negative de forma errónea.
- **Sensitividad, especificidad, accuracy.**

Técnicas de clasificación:

- Análisis discriminante lineal de Fisher.
- Regresión logística.
- K-vecinos más cercanos.
- Máquinas de soporte vectorial.
- Árboles de decisión

6.1. Introducción y objetivos

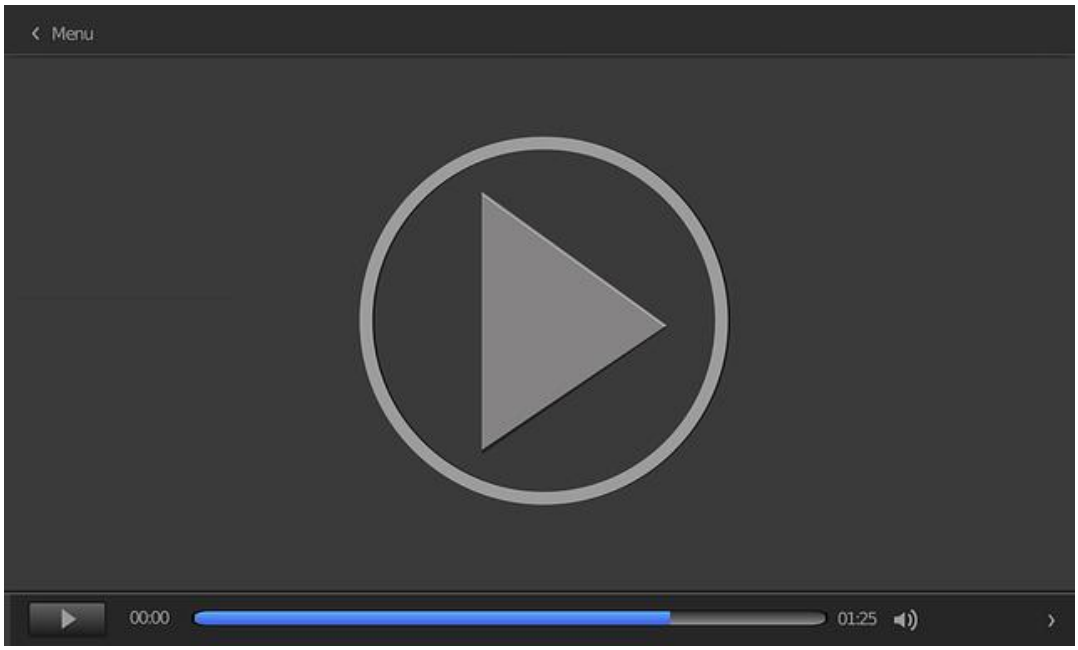
En el tema «Modelo de proceso de un proyecto orientado a datos» de la presente asignatura se presentó las fases de un proyecto orientado a datos. La fase 5 estaba dedicada a la creación de un modelo en base a datos. La construcción de un modelo es un proceso iterativo que incluye la selección de variables y la ejecución y valoración del modelo. El tema «Proceso del modelado de datos» constituyó una presentación general del modelado en base a datos. El aprendizaje automático (*machine learning*) proporciona la herramientas necesarias para la creación de modelos de forma automática. El aprendizaje automático se puede abordar como un aprendizaje supervisado o no supervisado. En aprendizaje supervisado, los modelos se ajustan en base a un conjunto de ejemplos proporcionados. Estos ejemplos son datos etiquetados, variables dependientes e independientes con valores concretos. El aprendizaje supervisado de aplica principalmente a problemas de predicción y clasificación. Este tema presenta las principales técnicas utilizadas de predicción y clasificación.

Objetivos que se pretenden conseguir:

- Entender qué es un modelo de predicción.
- Conocer un conjunto de técnicas de predicción muy usadas.
- Entender qué es un modelo de clasificación.
- Conocer un conjunto de técnicas de clasificación muy usadas.

Ideas clave

Vídeo *Técnicas de aprendizaje supervisado*.



Accede al vídeo:

https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=8f3d484a-df19-4613-a38c-b15d00aa2657

6.2. Introducción a las técnicas de predicción

Una de las principales funciones del aprendizaje automático supervisado es la creación de modelos de predicción. La **predicción** es una de las herramientas más importantes en la toma de decisiones. Una empresa puede tener interés en predecir las ventas que realizará con el objeto de ajustar sus existencias de productos. Se puede intentar predecir la rentabilidad de una inversión en publicidad para ajustar los gastos. Se puede intentar predecir las ventas de un producto para ajustar la producción del mismo. En el contexto empresarial existen infinidad de ejemplos de la utilidad de un modelo de predicción.

Esta sección presenta las principales técnicas de aprendizaje automático usadas para obtener un modelo de predicción. El primer paso que vamos a dar en esta sección es explicar de forma sencilla qué entendemos por predicción. Para ello vamos a utilizar un pequeño ejemplo artificial.

	Variable dependiente	Variable independiente
Observación 1	1	1
Observación 2	2	4
Observación 3	3	6
Observación 4	4	8
Observación 5	5	10

Tabla 1. Conjunto de datos disponibles.

Qué es la predicción

Supongamos que disponemos del conjunto de datos incluidos en la tabla 1. La primera columna representa la variable independiente y la segunda la variable

dependiente. Cada una de las filas son las distintas observaciones. La figura 1 resume las relaciones conocidas entre los datos.

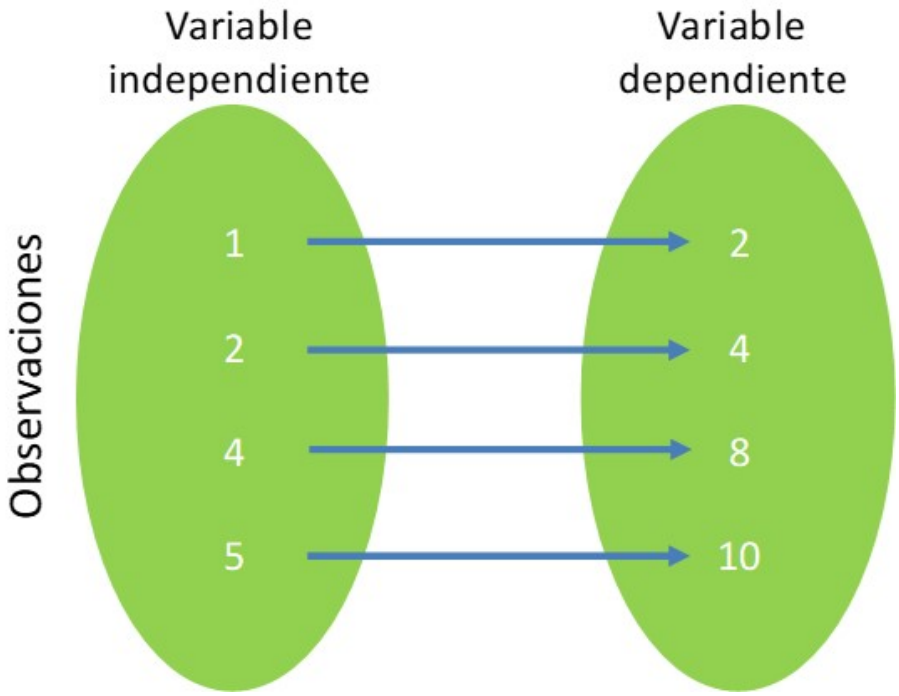


Figura 1. Observaciones disponibles.

El objetivo en un problema de predicción es dado una nueva observación de la variable independiente, estimar qué valor tiene la variable dependiente. Este proceso se puede observar gráficamente en la figura 2.

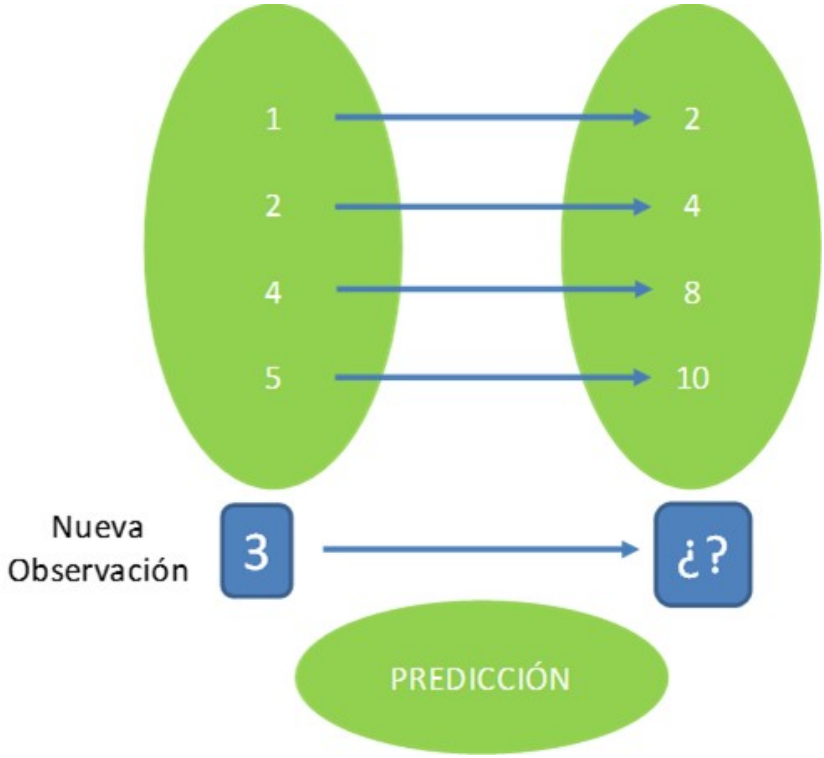


Figura 2. Predecir una nueva observación.

Ante una nueva observación de la variable independiente con valor 3 un predictor debe obtener el valor correspondiente de la variable dependiente. En este caso, una predicción razonable es 6. El proceso mental que nos ha llevado a esa conclusión ha sido observar que la variable dependiente toma valores dobles que la variable independiente. Por tanto, si a la variable dependiente la denominamos y, y a la variable independiente x, la expresión matemática que describe nuestro predictor es $y = 2x$.

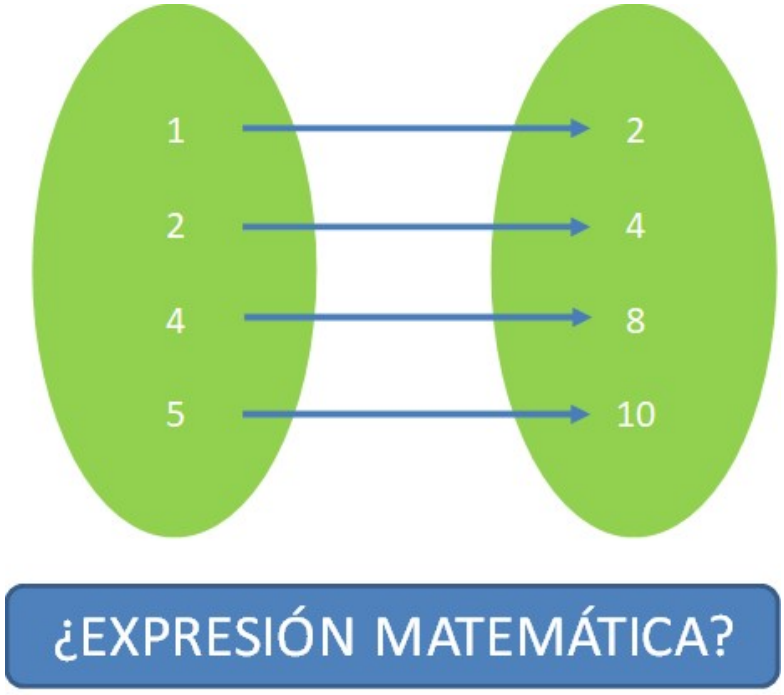


Figura 3. Modelado de un predictor.

Bien, básicamente ese es el proceso de modelado de un predictor, queremos que el computador de forma automática nos proporcione la expresión matemática del predictor basándose en el conjunto de datos de entrenamiento proporcionado.

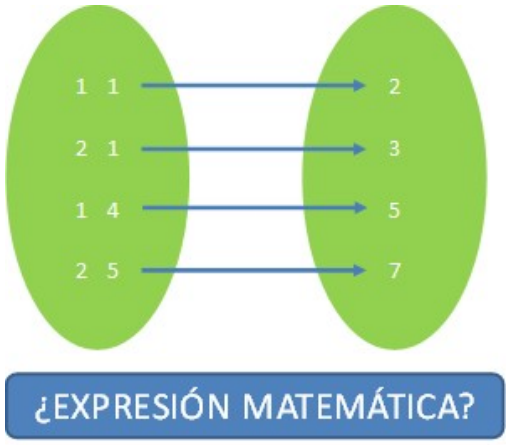


Figura 4. Modelado de un predictor.

La expresión que relaciona las variables dependientes con las independientes no siempre es sencilla de deducir. Por ejemplo, en la figura 4 tenemos dos variables independientes y otra dependiente. En este caso la posible expresión matemática del predictor, aunque sencilla, ya no es tan obvia. La variable dependiente es la suma de las variables dependientes. Queremos que nuestro sistema de modelado sea capaz automáticamente de obtener las expresiones matemáticas del predictor de forma automática, aunque las relaciones sean complejas.

Resumiendo el ejemplo anterior, dado un conjunto de variables independientes (que podemos denominar x) y un conjunto de variables dependientes (que podemos denominar *y*) el objetivo del modelado de un predictor es obtener la expresión matemática *f* que relaciona x con *y*, es decir *y=f(x).* A este proceso se le suele llamar regresión (ver figura 5).

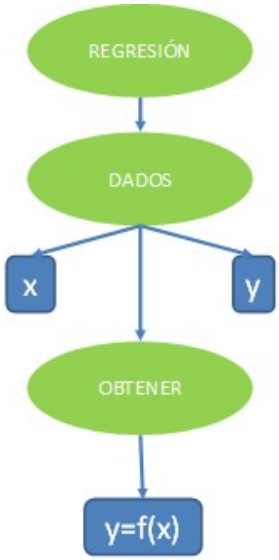


Figura 5. Regresión para obtener un modelo predictor.

Medidas de calidad de un modelo de predicción

A la hora de decidir qué modelo de predicción es el más adecuado para cierto proyecto en necesario disponer de métricas que midan de forma objetiva la precisión

de los distintos modelos. Esta sección presenta varias de dichas medidas. Se asume que disponemos de un conjunto de *n* observaciones {(x1,y1),(x2,y2),…,(xn,yn)} y un modelo predictor

$\tilde{f}\ (x)$. El modelo predictor

$\tilde{f}\ (x)$, toma como entrada una nueva observación de la/s variable/s independiente/s *x* y devuelve una predicción de la variable dependiente

$\tilde{y} = \tilde{f}\ (x)$.La precisión de la predicción se debe basar en la diferencia

$y_i - \tilde{f}\ (x_i)$, siendo la precisión máxima que esa diferencia sea cero. A continuación, se definen las dos medidas más usadas MAE y RMSE.

- Error absoluto medio (Mean Absolute Error, MAE). Es la media de los errores en valor absoluto, esto es, sin considerar el signo.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \tilde{f}\ (x_i)\right|$$

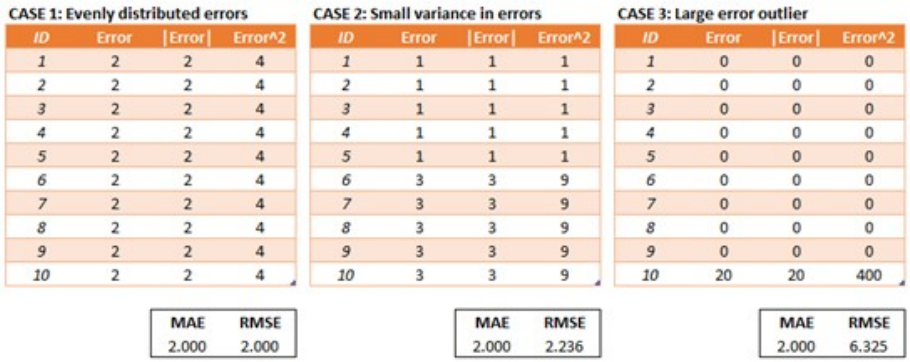
- Raíz cuadrada del Error cuadrático medio (Root Mean Squared Error, RMSE). Es la raíz cuadrada de la media de los errores cuadráticos.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \tilde{f}\ (x_i)\right)^2}$$

Una pregunta pertinente es qué similitudes y diferencia hay entre ambas medidas. Ambas expresan el error de predicción medio en unidades de la variable dependiente *y*. Cuanto menor sea su valor, mejor poder de predicción tiene el modelo asociado. En cuanto a las diferencias, el RMSE da más importancia a los errores grandes (pues los eleva al cuadrado). La figura 6 muestra este punto. Se tienen diez observaciones con un error total de 20. La métrica MAE es insensible a cómo se reparte ese error, sin embargo, la métrica RMSE asigna un mejor valor si el error está muy repartido (caso 1) respecto a errores concentrados en pocas medidas (casos 2 y 3). También se tiene que MAE <= RMSE <=

$\sqrt{n}$MAE. Finalmente, indicar que al utilizar MAE el valor absoluto, se puede complicar la realización de ciertos algoritmos. Por eso muchas veces los algoritmos

prefieren usar el MSE, que es el RMSE al cuadrado.



CASE 1: Evenly distributed errors

ID	Error	|Error|	Error^2
1	2	2	4
2	2	2	4
3	2	2	4
4	2	2	4
5	2	2	4
6	2	2	4
7	2	2	4
8	2	2	4
9	2	2	4
10	2	2	4

MAE	RMSE
2.000	2.000

CASE 2: Small variance in errors

ID	Error	|Error|	Error^2
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	9
7	3	3	9
8	3	3	9
9	3	3	9
10	3	3	9

MAE	RMSE
2.000	2.236

CASE 3: Large error outlier

ID	Error	|Error|	Error^2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	20	20	400

MAE	RMSE
2.000	6.325

Figura 6. Ejemplos de métrica de error. Fuente: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Otras medias importantes son:

- Error de sesgo medio (Mean Bias Error, MBE). Esta métrica es útil para detectar sesgos en los modelos de predicción. El sesgo se tiene si el MBE tiene un valor significativamente distinto de 0.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} y_i - \tilde{f}\,(x_i)$$

- Error cuadrático medio (Mean Square Error, MSE).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \tilde{f}\,(x_i)\right)^2$$

Las métricas anteriores están definidas en unidades de la variable dependiente. Cuando se quieren comparar predictores con datos de diferente naturaleza es mejor utilizar métricas que vengan definidas en porcentaje o normalizadas, esto es, sin unidades.

A continuación, se dan varios ejemplos de este tipo de métrica:

- Error absoluto medio porcentual (Mean Absolute Porcentaje Error, MAPE). Esta medida tiene como límite inferior el 0 % y su valor crece con la cantidad del error.

Tiene un grave problema, cuando una de las observaciones y_i toma valor 0 entonces MAPE toma valores infinitos. Una solución a ese problema es dividir por el valor medio de las observaciones.

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \tilde{f}(x_i)}{y_i} \right|$$

- Raíz cuadrada del Error cuadrático medio normalizado (normalized Root Mean Squared Error, nRMSE). Se divide el RMSE por el valor medio de las observaciones de la variable dependiente.

$$nRMSE = \frac{RMSE}{\bar{\gamma}}$$

6.3. Técnicas de predicción

Regresión lineal

En regresión lineal el modelo de predicción utilizado es: $\tilde{f}\left(x\right)=\theta^{T}x$

Donde

θ es un vector de parámetros. Ese vector se ha calculado previamente resolviendo por ejemplo un problema de mínimos cuadrados usando las observaciones disponibles como conjunto de datos de entrenamiento. En el tema «Técnicas estadísticas de análisis de datos» de esta asignatura se estudió este tipo de modelos.

Desde el punto de vista del diseño de un predictor basado en regresión lineal la única decisión a tomar es seleccionar las variables independientes a usar.

Son varias las consideraciones que podemos hacer. La primera es que para realizar una nueva predicción lo único que se necesita es el vector

θ donde se ha codificado toda la información de interés incluida en los datos de entrenamiento. Los modelos basados en regresión lineal son muy interpretables, pero tienen poca flexibilidad a la hora de adaptarse a datos. Si los datos tienen comportamientos no lineales, la regresión lineal puede que no proporcione buenos resultados. Para adaptarse a este tipo de situación se puede incrementar la flexibilidad del modelo incluyendo variables independientes transformadas por algún tipo de función no lineal. Por ejemplo, supongamos que solo tenemos una variable dependiente *x*, el modelo se puede ampliar incluyendo transformaciones de la variable dependiente del tipo *x2, x3*, log(*x*), ex, etc. Por supuesto eso implica incrementar el tamaño del vector

θ y por tanto incrementar la complejidad del modelo.

Regresión basada en soporte vectorial (SVR)

La aplicación de las máquinas de soporte vectorial (explicadas más adelante en la

sección «Introducción a las técnicas de clasificación») a la regresión proporciona el modelo de predicción denominado **regresión basada en soporte vectorial.**

La SVR utiliza el siguiente modelo de predicción:

$$\tilde{f}\ (x) = \sum_{i=1}^{n} \alpha_i k(x,x_i)\,,$$

donde

$k(x,x_i)$ es una función especial denominada **función kernel** y

α_i son un conjunto de constantes. Las constantes

α_i no son conocidas de antemano, pero son calculadas en la fase de entrenamiento del modelo resolviendo un problema de optimización parecido al usado en regresión lineal. Las funciones kernel son funciones bien conocidas que cumplen la denominada **condición de Mercer.** Ejemplos de funciones kernel son:

- Lineal:
  $k(x,x_i) = x^T x_i$
- Polinomial:
  $k(x,x_i) = \left(\tau + x^T x_i\right)^d$
- Gaussiano:
  $k(x,x_i) = e^{\frac{-\|x-x_i\|_2^2}{\sigma}}$

Nótese que algunas de las funciones kernel contienen unas constantes denominados **hiperparámetros** (τ y σ, por ejemplo) que no son conocidos de antemano y por tanto también hay que ajustar en entrenamiento.

La **predicción** es una suma o combinación lineal de la salida de las funciones kernel. Valores diferentes de la variable independiente x activa y desactiva diferentes sumandos. Una característica importante del modelo de regresión basado en soporte vectorial es que, para cada nueva predicción, se necesita recuperar todos los datos

xi, pues aparecen en los sumandos. Por tanto, la regresión basada en soporte vectorial utiliza los datos de entrenamiento cada vez que se realiza una nueva predicción.

Resumiendo, desde el punto de vista del diseño de un predictor que usa SVR, hay que seleccionar:

- Las variables independientes a usar.
- La función kernel a usar.
- El valor de los hiperparámetros asociados a la función kernel elegida.

El diseño de un predictor SVR no es un proceso secuencial. Es un proceso iterativo donde se podrá cambiar la función kernel a usar y el valor de los posibles hiperparámetros, hasta encontrar la configuración adecuada que proporcione el poder predictivo deseado. Cuando el número de datos de entrenamiento es muy elevado, puede que sea necesario métodos especiales de entrenamiento para obtener las constantes

α_i.

Procesos Gaussianos (GP)

Un modelo de predicción basado en GP utiliza el siguiente modelo:

$$\tilde{f}\ (x) = \sum_{i=1}^{n} \alpha_i y_i,$$

donde *y*i son las observaciones disponibles de la variable dependiente y

α_i son un conjunto de constantes. Los valores de

α_i se obtienen a partir de la función kernel

$k(x,x_i)$ y una matriz de covarianza. La idea subyacente es que valores similares de las variables independientes deben generar valores similares en la variable dependiente. Esto se traduce en lo siguiente, si *x* es muy parecido a un valor de *x*i, entonces

α_i toma un valor grande, lo que hace que la medida *y*i tome importancia en la predicción. Por otro lado, si *x* es muy diferente a un valor de *x*j, entonces el correspondiente

α_j toma un valor cercano a cero y la medida *y*j no tiene importancia en la predicción.

Resumiendo, desde el punto de vista del diseño de un predictor basado en GP, hay que seleccionar:

- Las variables dependientes a usar.
- La función kernel a usar.
- El valor de los hiperparámetros asociados a la función kernel elegida.

El diseño de un predictor GP no es un proceso secuencial. Es un proceso iterativo donde se podrá cambiar la función kernel a usar y el valor de los posibles hiperparámetros, hasta encontrar la configuración adecuada que proporcione el poder predictivo deseado. Cuando el número de datos de entrenamiento es muy elevado, puede que sea necesario métodos especiales de entrenamiento para obtener las constantes

α_i.

Redes neuronales

Basada en modelos matemáticos que tratan de imitar el modelo y funcionamiento de los sistemas biológicos: cómo las redes de neuronas almacenan y manipulan la información. Es una potente herramienta con aplicaciones en regresión, clasificación y clusterización. Existen varios tipos, según su aplicación (ver http://hagan.okstate.edu/nnd.html).

Las redes neuronales se basan en el concepto de **neurona.** Una neurona no es más que una función matemática, denominada función de activación, que tiene ciertas entradas y una salida. El valor de la salida depende de los valores de las entradas y

de la función de activación seleccionada.

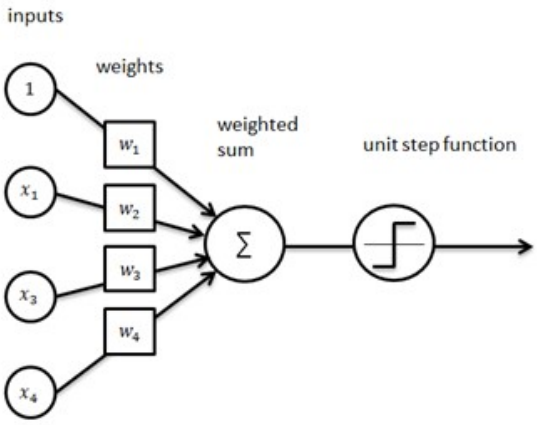


Figura 7. Modelo de neurona. Fuente: https://blog.goodaudience.com/artificial-neural-networks-explained-436fcf36e75

En la figura 7 se puede observar el modelo de una neurona. Se ven las entradas de la neurona (1,*x*1,*x*3,*x*4), unos pesos (*w*1,*w*2,*w*3,*w*4), un operador suma, y una función de activación (en este caso la función escalón unitario). El funcionamiento de la neurona se puede establecer dándole valores concretos a los pesos.

La red neuronal está diseñada en capas y cada capa tiene un conjunto de neuronal. Al número de capas, neuronas por capas, conexiones entre neuronas y función de activación seleccionada se le denomina **arquitectura de la red.**

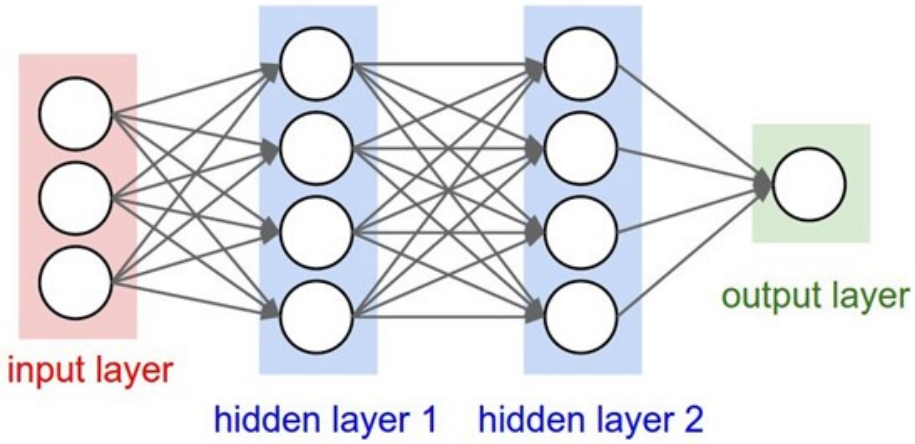


Figura 8. Modelo de red neuronal. Fuente: https://blog.goodaudience.com/artificial-neural-networks-explained-436fcf36e75

En la figura 8 se puede observar un ejemplo de arquitectura de red neuronal. Se pueden observar las distintas capas, las neuronas que forman cada capa y las conexiones entre neuronas. Es importante indicar que cada una de esas conexiones tiene un peso *w*. Para obtener una predicción con la red neuronal hay que colocar el valor de las variables independientes en la entrada, activar la red neuronal y la capa de salida proporciona la predicción correspondiente.

La arquitectura de la red define la complejidad de la misma y por tanto la flexibilidad del modelo. Cuanto mayor es el número de neuronas y de conexiones entre neuronas, mayor es la flexibilidad del modelo y por tanto mayor es el riesgo de caer en sobre-entrenamiento si no se realiza un entrenamiento adecuado de la red neuronal. Por otro lado, la interpretabilidad interna de un modelo basado en red neuronal puede ser compleja.

Un paso importante en el **diseño de una red neuronal** para predicción es el entrenamiento. Básicamente el entrenamiento es el proceso de darle los valores adecuados a los pesos *w* que interconectan las neuronas. En aprendizaje supervisado, esos pesos se asignan utilizando ejemplos de funcionamiento deseado. Existen varios algoritmos iterativos de entrenamiento disponibles. Es importante utilizar técnicas de validación cruzada para evitar el sobre-entrenamiento de la red.

Resumiendo, para diseñar una red neuronal, hay que seleccionar:

- Las variables dependientes a usar.
- La función de activación a utilizar. Existe un conjunto estándar disponible.
- El número de capas de la red. Existen algunos consejos heurísticos en función del tipo de problema.

- El número de neuronas por capa. Normalmente ese número dependerá del tipo de problema que estemos resolviendo.
- El algoritmo de entrenamiento.

El diseño de una red neuronal para predicción no es un proceso secuencial. Es un proceso iterativo donde se podrán cambiar parámetros de la red, como el número de neuronas por capa o la función de activación, hasta encontrar la configuración adecuada que proporcione el poder predictivo deseado.

Es importante indicar que aunque el diseño y entrenamiento de la red neuronal puede ser complejo, su utilización es sencilla, pues simplemente es la evaluación de una función. Una red neuronal bien entrenada ha conseguido codificar toda la información importante de los datos de entrenamiento en su interior y por tanto se puede usar en aplicaciones que requieran muchas predicciones por unidad de tiempo. Recordar que los predictores SVR y GP incluyen en su modelo los propios datos de entrenamiento.

En la tabla 2 se resumen las principales características de los modelos de predicción comentados.

	Propiedades	
	Flexibilidad	Interpretabilidad
Regresión lineal	Baja	Alta
SVR	Kernel lineal: Baja Kernel gaussiano: Alta	Kernel lineal: Sencilla Kernel gaussiano: Baja
GP	Alta	Baja
Redes neuronales	Alta	Baja

Tabla 2. Conjunto de datos disponibles.

Ideas clave

6.4. Introducción a las técnicas de clasificación

El aprendizaje supervisado también se aplica a problemas de clasificación. La clasificación también es una de las herramientas importantes en la toma de decisiones. Básicamente, la **clasificación** es un problema de predicción de categorías. En un problema de predicción el objetivo es estimar una cantidad (de productos, económica, etc.). En un problema de clasificación se intenta estimar a qué categoría pertenece la observación de interés. Por ejemplo, una empresa puede tener interés en estimar si un cliente tiene o no capacidad de pago, si puede o no realizar operaciones fraudulentas, si es o no un buen conductor. Se puede clasificar una empresa como solvente o no, potencial cliente o no, etc. En el contexto empresarial existen infinidad de ejemplos de la utilidad de un modelo de clasificación.

Esta sección presenta las principales técnicas de aprendizaje automático usadas para obtener un modelo de clasificación. El primer paso que vamos a dar en esta sección es explicar de forma sencilla qué entendemos por clasificación binaria. Para ello vamos a utilizar un pequeño ejemplo artificial.

	Variable dependiente	Variable independiente
Observación 1	1	1
Observación 2	3	1
Observación 3	4	1
Observación 4	5	2
Observación 5	6	2
Observación 6	8	2

Tabla 3. Conjunto de datos disponibles.

Ejemplo simplificado para entender qué es la clasificación

Supongamos que disponemos del conjunto de datos incluidos en la tabla 3. La primera columna representa la variable independiente y la segunda la variable dependiente. Cada una de las filas son las distintas observaciones. En un problema de clasificación la variable dependiente toma un conjunto finito de valores discretos también denominados categorías (en este caso solo toma la categoría 1 o 2). La figura 9 resume las relaciones conocidas entre los datos.

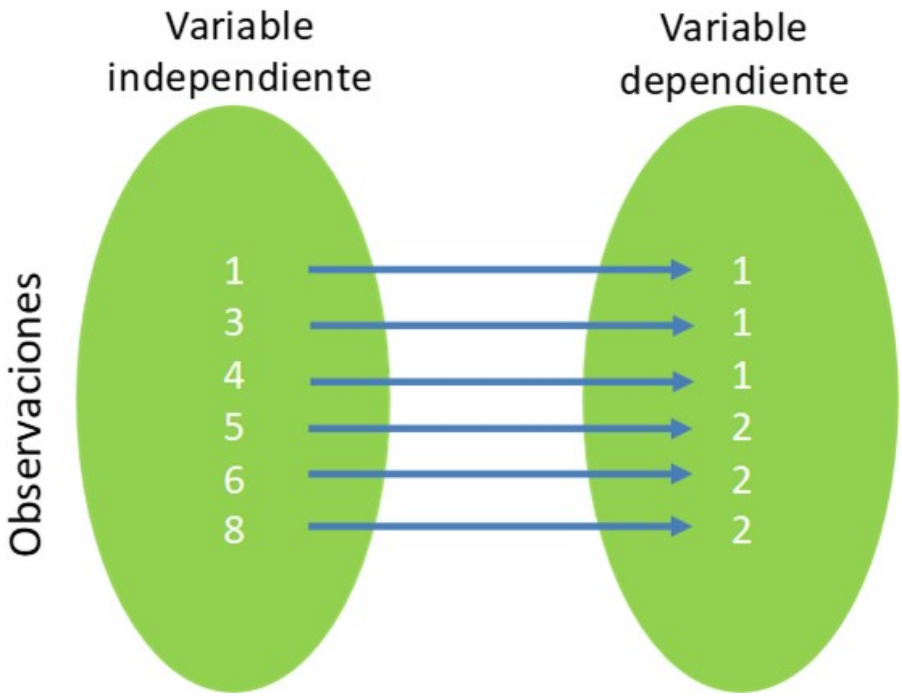


Figura 9. Observaciones disponibles.

El objetivo en un problema de clasificación es dado una nueva observación de la variable independiente, estimar qué valor tiene la variable dependiente esto es, a qué categoría pertenece. Este proceso se puede observar gráficamente en la figura 10.

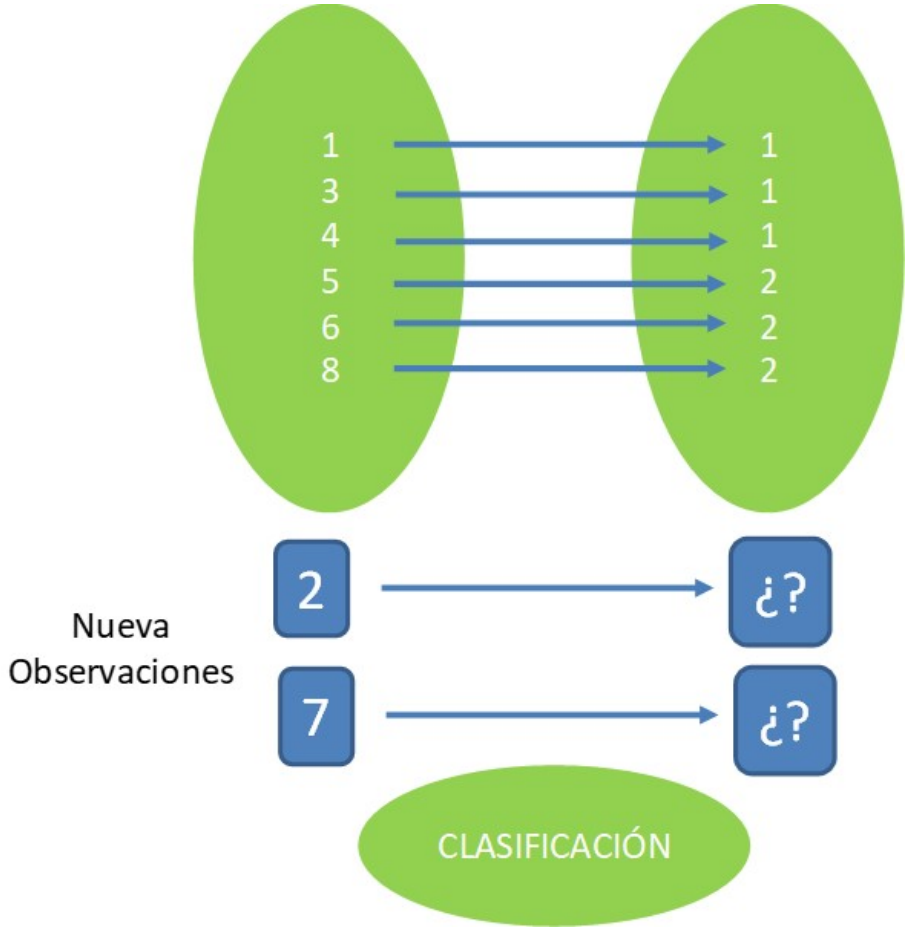


Figura 10. Clasificar nuevas observaciones.

En este caso para la observación x=2, una clasificación razonable es que pertenece a la categoría 1. Por otro lado, para la observación x=7, una clasificación razonable es que pertenece a la categoría 2. Es importante aclarar que a las categorías se le asigna un número pues los computadores solo saben procesar números. Pero una categoría es un concepto que se puede referir a una idea abstracta. Esta idea es la que se ilustra en la figura 11, donde tenemos dos categorías, las estrellas y los triángulos. A la hora de procesar la información en un computador las estrellas se pueden sustituir por un valor numérico 1 y los triángulos por un valor numérico 2.

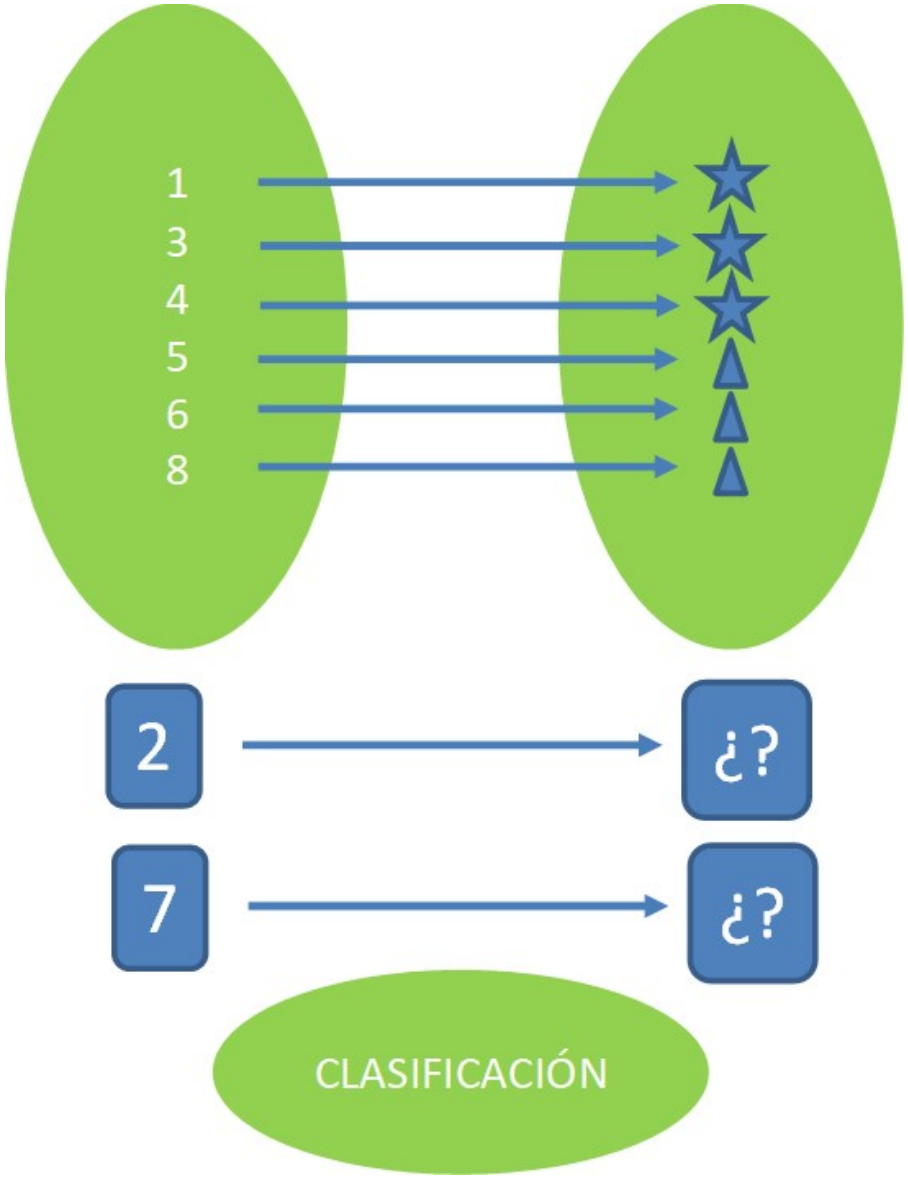
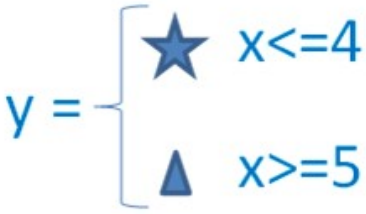


Figura 11. Clasificar nuevas observaciones.

El proceso mental que nos ha llevado a clasificar las nuevas observaciones ha sido observar que la variable dependiente toma valor estrella (uno) si $x<=4$ y toma valor triángulo (dos) si $x>=5$. Por tanto, si a la variable dependiente la denominamos y, y a la variable independiente x, la expresión matemática que describe nuestro clasificador es:



Básicamente ese es el proceso de modelado de un clasificador, queremos que el computador de forma automática nos proporcione la expresión matemática del clasificador basándose en el conjunto de datos de entrenamiento proporcionado.

La expresión que relaciona las variables dependientes con las independientes no siempre es sencilla de deducir. Por ejemplo, en la figura 12 tenemos dos variables independientes (el espacio de las variables dependientes es de dimensión dos) y otra dependiente. En este caso la posible expresión matemática del clasificador, aunque sencilla, ya no es tan obvia.

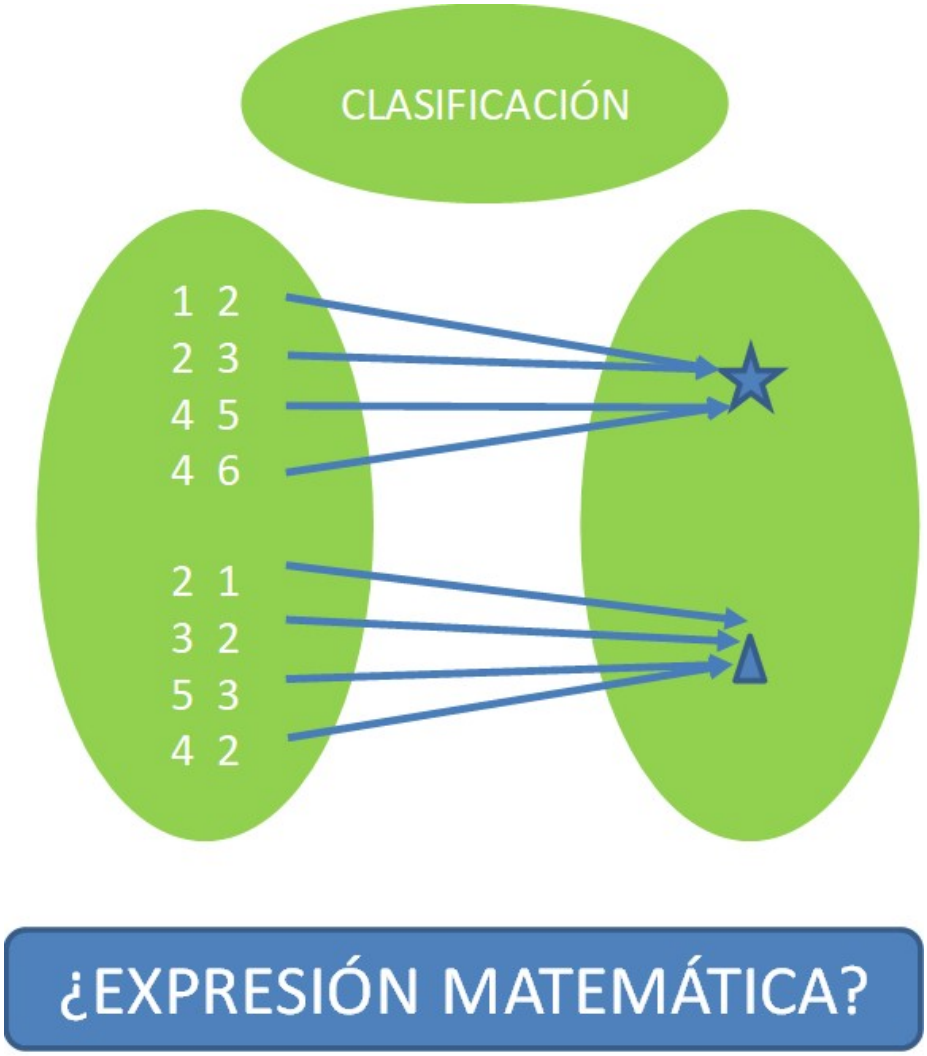


Figura 12. Clasificar nuevas observaciones.

Una opción de expresión para el nuevo clasificador sería:

$$y = \begin{cases} \bigstar & x_1 < x_2 \\ \blacktriangle & x_1 > x_2 \end{cases}$$

Como se puede observar un clasificado es básicamente una división del espacio de

las variables independientes. A cada división se le asigna una clase o categoría.

Es importante indicar que la clasificación binaria se refiere a casos donde solo hay dos categorías o clases. Los métodos de clasificación se pueden generalizar a un mayor número de categorías.

Medidas de calidad de un modelo de clasificación

A la hora de decidir qué modelo de clasificación es el más adecuado para cierto proyecto en necesario disponer de métricas que midan de forma objetiva la precisión de los distintos modelos. Esta sección presenta varias de dichas medidas. Se comienza presentando qué es la matriz de confusión.

Una **matriz de confusión** es una herramienta para medir la calidad de un modelo de clasificación. Dado un conjunto de observaciones, las cuales tienen etiquetadas la categoría a la que pertenecen, la matriz de confusión se construye de la siguiente forma. Las columnas representan las categorías reales de las observaciones. Las filas las categorías predichas por el modelo para cada observación. Por tanto, en la diagonal se tiene el número de observaciones con predicción acertada. El resto de posiciones de la matriz indica cuántas observaciones de cada categoría se han predicho erróneamente y cuál ha sido la predicción realizada.

Matriz de confusión		
	▲	★
▲	5 (TP)	3 (FP)
★	2 (FN)	6 (TN)

Tabla 4. Ejemplo de matriz de confusión.

La tabla 4 representa un ejemplo de matriz de confusión. De ella se deducen los siguientes datos:

- Se tienen un total de 16 observaciones. 7 observaciones pertenecen a la categoría triángulo y 9 a la estrella.
- Se ha utilizado un modelo de clasificación para estimar la categoría de cada observación.
- Se han acertado 5 observaciones pertenecientes a la categoría triángulo y 6 observaciones pertenecientes a la categoría estrella.
- 2 observaciones de la categoría triángulo se clasificaron erróneamente como estrella. Y 3 de la categoría estrella se clasificaron erróneamente como triángulo.

Si a la categoría triángulo se la denomina Positive, y a la categoría estrella se la denomina Negative, se definen las siguientes métricas:

- **True Positive (TP).** Número de aciertos en la categoría Positive. En este caso TP = 5.
- **True Negative (TN).** Número de aciertos en la categoría Positive. En este caso TP = 6.
- **False Positive (FP).** Número de observaciones clasificas Positive de forma errónea. FP = 3.

- **False Negative (FN).** Número de observaciones clasificas Negative de forma errónea. FN = 2.

Ahora es posible presentar cuatro métricas que indican la calidad de un modelo de clasificación. Todas las métricas toman valor entre 0 (mal resultado) y 1 (mejor resultado).

- Sensitividad o ratio de True Positive. Es el cociente entre TP y P (número de observaciones de la categoría Positive). En el ejemplo es 5/7.
- Especificidad o ratio de True Negativa. Es el cociente entre TN y N (número de observaciones de la categoría Negative). En el ejemplo es 6/9.
- Accuracy. Es el cociente entre observaciones bien clasificadas y el conjunto total de observaciones. En el ejemplo es (5+6)/(7+9).
- Balanced Accuracy. Es la sensitividad más la especificidad dividido todo ello entre 2. Adecuado para clases con número de componentes muy diferentes.

Es importante entender que un modelo de clasificación debe mejorar los resultados de los clasificadores obvios, esto es, clasificar todo como Positive o bien todo como Negative. En ambos casos se obtienen los accuracy P/(P+N) y N/(P+N). Por tanto, el accuracy de un modelo de clasificación debiera mejorar como mínimo esos resultados.

6.5. Técnicas de clasificación

Análisis discriminante lineal de Fisher

El **análisis discriminante de Fisher** es uno de los primeros métodos de clasificación que se diseñaron. La idea básica es encontrar una recta sobre la que proyectar las observaciones, de forma que se facilite la clasificación.

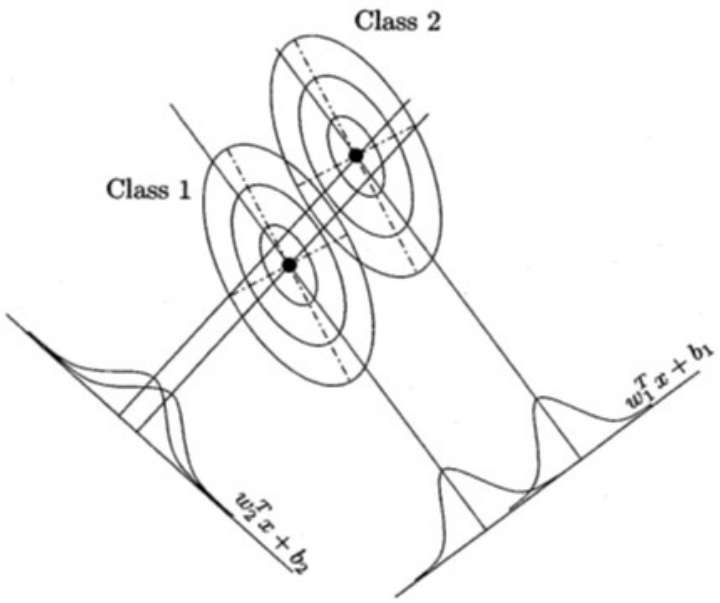


Figura 13. Clasificación basada en ADF. Fuente: Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D. y Vandewalle, J. (2002)

La figura 13 ilustra el método. Se asume un problema de clasificación con dos variables independientes (espacio de dimensión dos) y observaciones de dos clases o categorías diferentes. En la figura aparecen esas observaciones representadas mediante dos conjuntos de elipses y un punto central que representa el centroide de cada clase. El análisis discriminante de Fisher proyecta las observaciones sobre una recta y selecciona la recta que mejor separa las clases. En la figura 13 se pueden observar dos ejemplos de rectas de proyección, la de la izquierda genera alta confusión entre clases, la de la derecha permite una separación bastante clara de las

clases. Por tanto, esta segunda recta de proyección sería la seleccionada. Cuando queremos clasificar una nueva observación, se proyecta sobre la recta de clasificación y según la posición que ocupe se clasifica como clase uno o dos.

Este método de clasificación es adecuado cuando las observaciones de cada clase siguen una distribución cercana a la Gaussiana y por tanto admiten una separación por proyección.

Resumiendo, para crear un modelo clasificador basado en análisis discriminante de Fisher, hay que seleccionar las variables independientes a usar.

Regresión logística

Otro método de clasificación adecuado para problemas de dos clases es la denominada regresión logística. Un modelo de este estilo se basa en la denominada **función logística** que tiene forma de S, con valor inferior 0 y superior 1. Por tanto, dicha función permite diferenciar entre observaciones de la clase 0 y observaciones de la clase 1. La figura 14 ilustra la idea. Se pueden ver observaciones de dos clases (rojos asociados a la clase 0 y azul asociados a la clase 1). La función en forma de S intenta ajustarse para diferenciar adecuadamente las clases. Ante una nueva observación, se evalúa la función S y el valor obtenido (entre 0 y 1) indica la clase de la observación. Hay que definir un umbral de separación entre las dos clases, por ejemplo, se puede tomar el valor de la función S igual a 0.5. Todas las observaciones con un valor menor o igual a 0.5 se toman como clase roja y todas las observaciones con valor mayor a 0.5 en la función se toman como clase azul.

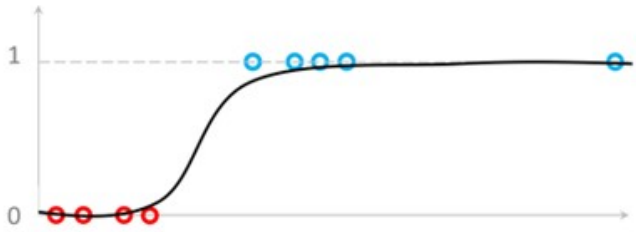


Figura 14. Clasificación basada en KNN. Fuente: https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102

Resumiendo, para crear un modelo clasificador basado en regresión logística, hay que seleccionar las variables independientes a usar.

K-vecinos más cercanos (KNN)

Uno de los métodos más intuitivos a la hora de crear un modelo de clasificación es el basado en los k-vecinos más cercanos. Dada una nueva observación, el algoritmo básicamente busca observaciones similares (vecinas), comprueba a qué categorías pertenecen esas observaciones similares y le asigna a la nueva observación la categoría mayoritaria.

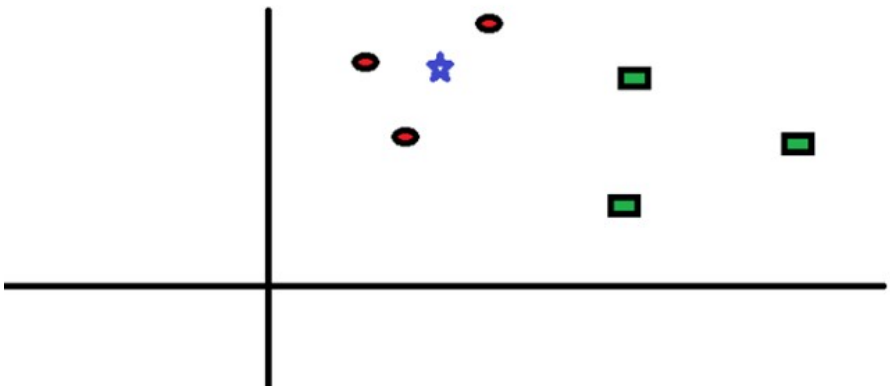


Figura 15. Clasificación basada en KNN. Fuente: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

En la figura 15 se puede observar el proceso para un clasificador con K=3. Se tienen observaciones de dos clases (círculos rojos y cuadrados verdes) y una nueva observación (estrella) que queremos clasificar. Las observaciones están en un espacio de dimensión dos, esto es, se tienen dos variables independientes.

Para realizar la clasificación se buscan las 3 observaciones más parecidas a la nueva. En la figura 16 se muestran las observaciones similares. El segundo paso es realizar una votación, hay que averiguar cuál es la clase mayoritaria en esas observaciones vecinas. En este caso, se tienen tres observaciones vecinas de las cuales tres pertenecen a la clase círculo rojo y cero a la clase cuadrado verde. Por tanto, a la observación estrella se le asigna la clase círculo rojo.

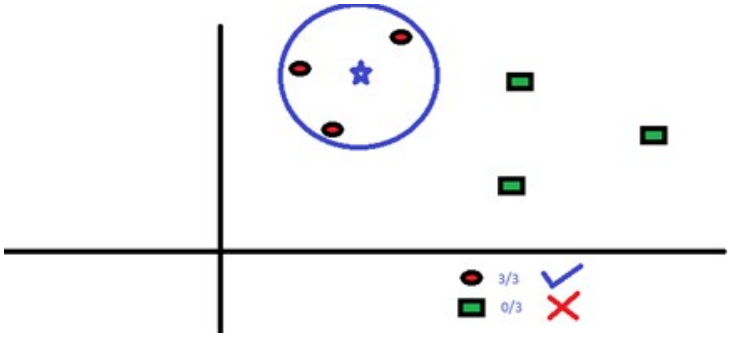


Figura 16. Clasificación basada en 3-NN. Fuente:
https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

El funcionamiento del clasificador cambia en función del número de vecinos considerado. En la figura 17 se puede observar el efecto que tiene en la clasificación el número de vecinos K usado. A medida que K se incrementa la superficie de separación entre las dos clases se hace más suave. Por tanto, K pequeña implica mayor flexibilidad pero mayor riesgo de sobre-entrenamiento y K grande implica menor flexibilidad y por tanto posible pérdida de potencia predictiva pero también menos riesgo de sobre-entrenamiento y por tanto mayor capacidad de generalización.

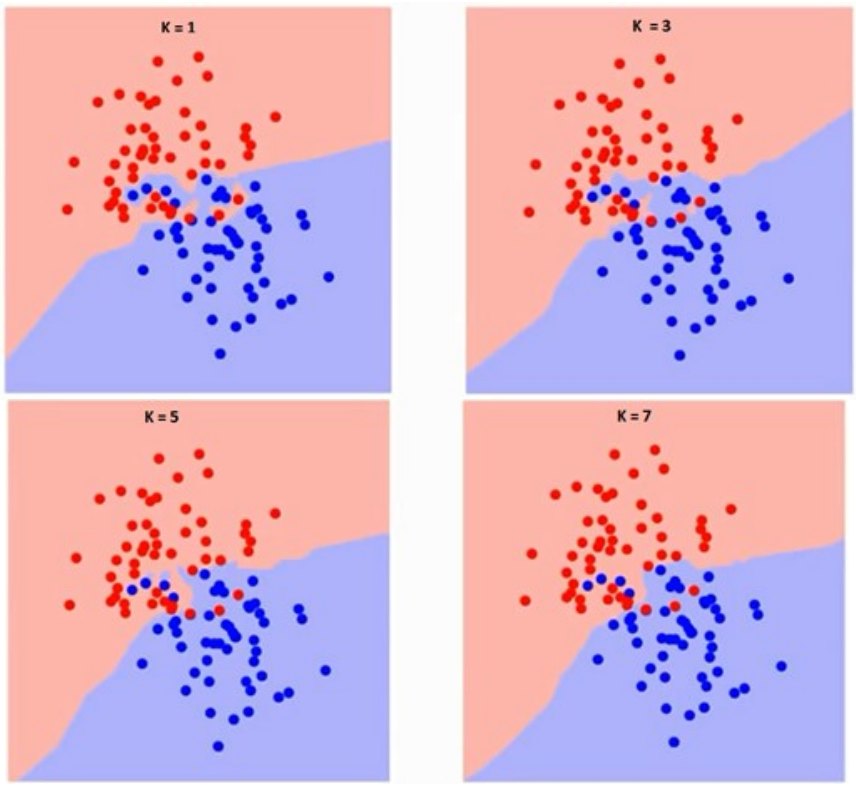


Figura 17. Clasificación basada en (1,3,5,7)-NN. Fuente:
https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Resumiendo, para crear un modelo clasificador basado KNN, hay que seleccionar:

- Las variables independientes a usar.
- El número de vecinos a considerar.

Máquinas de soporte vectorial (SVM)

Las máquinas de soporte vectorial es una técnica muy usada en clasificación. Básicamente la idea de un clasificador SVM es crear un hiperplano en el espacio de las variables dependientes que separe las dos categorías. En la figura xx se representa esta idea. Las dos clases-categorías se representan mediante puntos verdes y azules. El objetivo es crear un hiperplano que deje a un lado una clase y a otro la otra clase. En dimensión dos (las variables dependientes son dos) un

hiperplano es una recta. De todas las opciones posibles (hay infinitos hiperplanos separadores) se elige aquel que maximiza el margen separador, tal y como se puede observar en la figura 18 y 19.

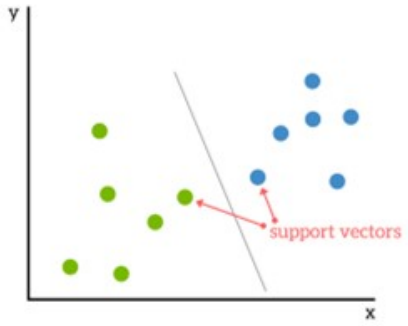


Figura 18. Hiperplano separador. Fuente: http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/

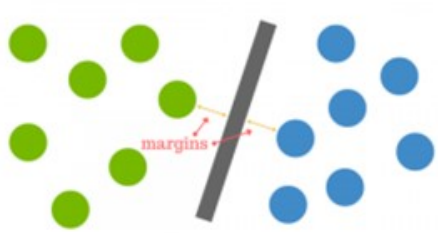


Figura 19. Maximizar margen de separación. Fuente: http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/

En los problemas reales, las clases no son directamente separables mediante un hiperplano. En la figura 20 se puede observar que no es posible trazar una recta que separe la clase azul de la verde. Para este tipo de casos, las SVMs utilizan kernels no lineales que proporcionan hiperplanos separadores, pero en dimensiones superiores. Este «truco» se puede observar en la figura 21. En dimensión 2, las clases verdes y azules no son separables linealmente, pero sí de forma artificial llevamos los valores de esas variables independientes a una dimensión mayor (por ejemplo, dimensión 3 en la figura 20), en esa nueva dimensión sí puede ser posible esa separación lineal. Cuando ese hiperplano separador de dimensión mayor vuelve

a la dimensión original se convierte en una superficie de separación que ya no es lineal.

Figura 20. Clases no separables linealmente. Fuente: http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/

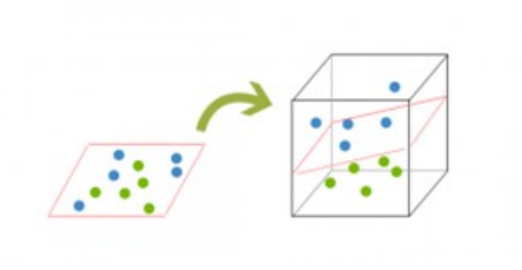


Figura 21. Hiperplano separador en dimensión superior. Fuente: http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/

Cuando tenemos una nueva observación y queremos clasificarla, hay que mirar en qué lado del hiperplano está y asignarle la categoría correspondiente.

Resumiendo, para crear un modelo clasificador basado SVM, hay que seleccionar:

- Las variables independientes a usar.
- La función kernel a usar.
- El valor de los hiperparámetros asociados a la función kernel elegida.

Árboles de decisión

Los **árboles de decisión** se basan en subdividir de forma iterativa el espacio de las variables independientes para conseguir que cada subdivisión esté asociada a una única clase.

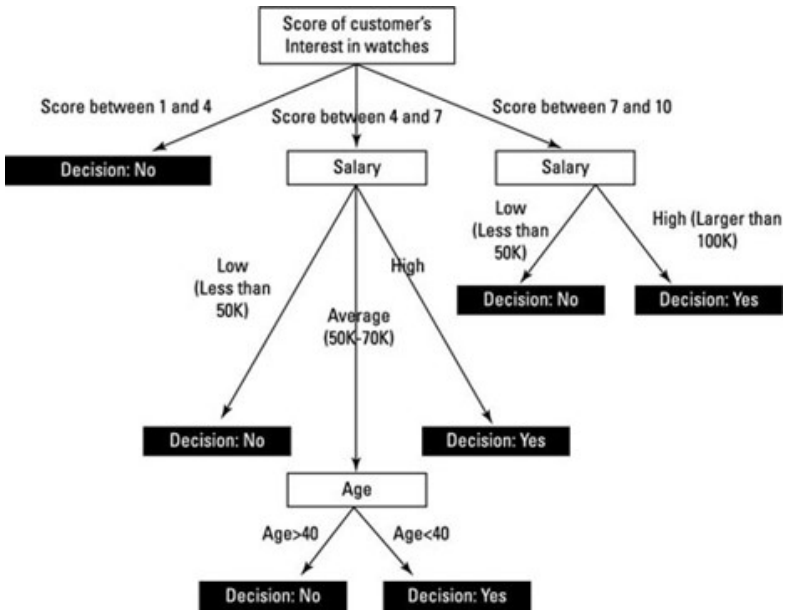


Figura 22. Ejemplo de árbol de decisión. Fuente: https://www.dummies.com/programming/big-data/data-science/how-to-use-predictive-analysis-decision-trees-to-predict-the-future/

Veamos un pequeño ejemplo para entender los árboles de decisión. La figura 22 representa un árbol decisión que clasifica a los clientes en dos categorías, los que podrían estar interesados en comprar un reloj y los que no. Esa es la variable dependiente. Por otro lado, tenemos tres variables independientes, el interés por los relojes, la edad y el salario de los clientes. Por tanto, el espacio de las variables de interés es de dimensión tres. Como se puede observar, ese espacio se divide en subespacios y a cada uno de ellos se le asigna una de las categorías.

Ante la llegada de un nuevo cliente, si se conoce su interés por los relojes, su edad y salario, un recorrido por el árbol de decisión nos puede ayudar a clasificar el cliente, esto es, saber si puede o no estar interesado en comprar el reloj.

Los algoritmos de clasificación basados en árboles de decisión tienen como objetivo crear un árbol de decisión en base a datos de entrenamiento (ejemplos previos). El proceso de subdivisión podría ser infinito, por tanto, al algoritmo hay que indicarle cuál es el nivel máximo de división que se desea. Esto se traduce en limitar el

número de hojas terminales incluidas. En el ejemplo anterior se tienen siete hojas terminales. Se puede construir un árbol más sencillo eliminando algunas hojas o bien complicar el árbol permitiendo un mayor número de hojas. Una de las ventajas de un modelo basado en árbol de decisión es su alto grado de interpretabilidad. Sin embargo, este grado de interpretabilidad se degrada a medida que el número de hojas crece.

Resumiendo, para crear un modelo clasificador basado en árbol de decisión, hay que seleccionar:

- Las variables dependientes a usar.
- La cantidad de niveles de división u hojas terminales a usar.

La tabla 5 resume las principales propiedades de los modelos de clasificación vistos.

	Propiedades		
	Flexibilidad	**Interpretabilidad**	**Velocidad de clasificación**
ADF	Baja	Alta	Rápida
Logística	Baja	Alta	Rápida
KNN	Mayor cuanto menor es K	Baja	Media
SVM	Kernel lineal: Baja Kernel gaussiano: Alta	Kernel lineal: Sencilla Kernel gaussiano: Baja	Media
Árboles de decisión	Pocas hojas: Baja Muchas hojas: Alta	Sencilla	Rápida
Redes neuronales	Alta	Baja	Rápida

Tabla 5. Conjunto de datos disponibles.

Otras técnicas de clasificación

Además de las comentadas en la sección anterior, también podemos encontrar modelos de clasificación basados en:

- **Redes neuronales.** Se entrena una red neuronal para que su predicción sea la categoría o clase que corresponde a cada observación.
- ***Boosting.*** Consiste en combinar varios clasificadores sencillos para obtener uno más robusto. El aprendizaje es secuencial, cada nuevo clasificador sencillo se especializa en aprender las observaciones mal clasificadas por los anteriores.
- ***Bagging.*** A partir del conjunto original de observaciones de entrenamiento se generan m conjunto nuevos de datos con las mismas propiedades. Con cado uno de estos conjuntos se entrena un clasificador y finalmente se combinan los m clasificadores para obtener uno robusto.

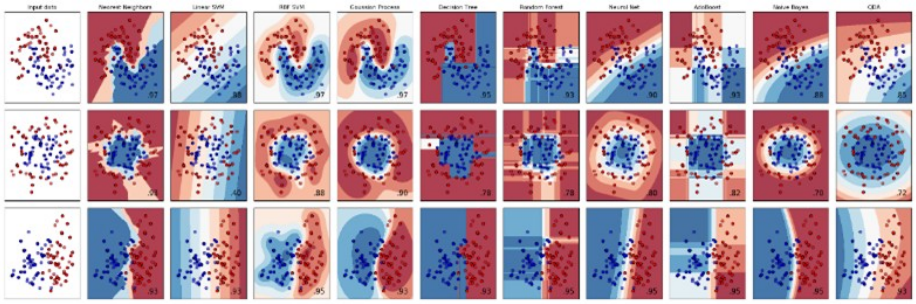


Figura 23. Ejemplo de otras técnicas de clasificación.

6.6. Referencias bibliográficas

Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D. y Vandewalle, J. (2002). *Least squares support vector machines.* Singapore: World Scientific.

Curva AUC-ROC

Narkhede, S. (2018). Understanding AUC-ROC Curve. Towards Data Science.

Uno de los métodos más importantes a la hora de evaluar modelos de clasificación es estudiar la curva AUC-ROC. En este documento se explica de forma sencilla su construcción e interpretación.

Accede al artículo a través del aula virtual o desde la siguiente dirección web: https://towardsdatascience.com/understanding-the-roc-curve-and-auc-dd4f9a192ecb/

Test

1. Si la salida de un predictor es una combinación lineal de las observaciones disponibles de la variable independiente, el predictor está basado en:

A. Procesos Gaussianos.

B. Redes neuronales.

C. Árboles de decisión.

D. Ninguna de las anteriores es correcta.

2. Las funciones kernel son necesarias en:

A. Regresión lineal.

B. Procesos Gaussianos.

C. Árboles de decisión.

D. Redes neuronales.

3. Aumentar el número de neuronas por capas de una red neuronal:

A. Siempre es beneficioso pues se incrementa la potencia predictora de la red.

B. Puede provocar sobreentrenamiento.

C. Nunca es recomendable pues provoca sobreentrenamiento.

D. No es posible pues viene prefijado de antemano.

4. Un modelo clasificador con un Acc = 0.9:

A. Es un buen clasificador, pues tiene un Acc cercano a 1.

B. No es posible.

C. Es un buen clasificador si las categorías están balanceadas.

D. Ninguna de las anteriores es correcta.

5. Si un modelo de clasificación utiliza un hiperplano que maximiza el margen de separación para diferenciar las clases estamos ante un modelo basado en:

A. Los k vecinos más cercanos.

B. Una máquina de soporte vectorial.

C. Una red neuronal.

D. Un árbol de decisión.

6. ¿Qué métrica de error penaliza más los errores grandes en la predicción?

A. MAE.

B. RMSE.

C. MAPE.

D. MBE.

7. En un clasificador KNN, ¿qué efecto tiene usar un valor de K muy pequeño?

A. Aumenta la capacidad de generalización del modelo.

B. Reduce el riesgo de sobreajuste.

C. Aumenta la flexibilidad y riesgo de sobreajuste.

D. El modelo no puede ser entrenado correctamente.

8. ¿Qué modelo no utiliza funciones de activación?

A. Red neuronal.

B. Regresión logística.

C. SVM.

D. Árbol de decisión.

9. ¿Qué característica distingue a las técnicas de *boosting* en clasificación?

A. La aleatoriedad en la selección de datos.

B. La combinación de clasificadores entrenados en paralelo.

C. La combinación secuencial de clasificadores que se especializan en errores anteriores.

D. La normalización previa de los datos.

10. ¿Cuál es una desventaja típica de las redes neuronales profundas frente a otros clasificadores?

A. Su baja capacidad de adaptación.

B. Requieren datos categóricos exclusivamente.

C. Baja capacidad computacional.

D. Dificultad de interpretación del modelo.