PRÁCTICA 9. APRENDIZAJE AUTOMÁTICO

Raquel Blanco Morago

Celia Calvo González

GRUPO 16

# Descripción del conjunto de datos

## Clustering: Seeds

Atributos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Valores válidos** |
| Área | REAL |  |
| Perímetro | REAL |  |
| Compacidad | REAL |  |
| Longitud de la semilla | REAL |  |
| Anchura de la semilla | REAL |  |
| Coeficiente de asimetría | REAL |  |
| Longitud del surco de la semilla | REAL |  |
| Tipo de semilla (salida) | NOMINAL | 1 -> Kama, 2 -> Rosa, 3 -> Canadian |

Describe las peculiaridades de semillas para resolver el problema de clustering, con el que podemos agrupar y sacar conclusiones.

URL: <https://archive.ics.uci.edu/ml/datasets/seeds>

## Clasificación: Car Evaluation

Atributos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Valores válidos** |
| Coste | NOMINAL | alto, bajo, mediano, muy alto |
| Tamaño | NOMINAL | alto, bajo, mediano, muy alto |
| Puertas | INTEGER |  |
| Capacidad | NOMINAL | 1,2,3,4, más |
| Tamaño maletero | NOMINAL | grande, mediano, pequeño |
| Seguridad | NOMINAL | alto, bajo, mediano |
| Evaluación (salida) | NOMINAL | aceptable, bueno, inaceptable, muy bueno |

Describe las características de vehículos para resolver el problema de clasificación, con el que se obtiene como salida una evaluación del coche, que indica cómo de bueno o de malo es el vehículo.

URL: <https://archive.ics.uci.edu/ml/datasets/car+evaluation>

## Regresión: AUTO MPG

URL: <https://archive.ics.uci.edu/ml/datasets/Auto+MPG>

Atributos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo** |
| MPG (salida) |  | REAL |
| Cilindradas |  | REAL |
| Desplazamiento |  | REAL |
| Caballos |  | REAL |
| Peso |  | REAL |
| Aceleración |  | REAL |
| Década fabricación |  | REAL |
| Origen |  | REAL |
| Nombre |  | NOMINAL |

Describe la especificación de automóviles en cuanto a su consumo y sus características. Es un buen data set para tartar el problema de regresión.

# APARTADO 1. CLUSTERING

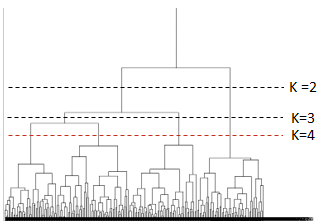
Sí, se deben normalizar las variables porque si no puede tener más peso una variable por tener un rango de valores más amplio y porque tenemos variables en varias unidades diferentes y vamos a establecer relaciones entre estas variables. También se debe quitar, para hacer el análisis, la variable de salida clase de semilla.

## Hiterarchical

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NUM.CLUSTERS | 2 | 3 | 4 | 5 |
| CLUSTERED  INSTANCES | 0 156 (74 %)  1 54(26%) | 0 101(48 %)  1 55 (26 %)  2 54 (26 %) | 0 43(20 %)  1 58 (28 %)  2 55 (26 %)  3 54 (26 %) | 0 43 (20 %)  1 58 (28 %)  2 45 (21 %)  3 54 (26 %)  4 10 ( 5 %) |

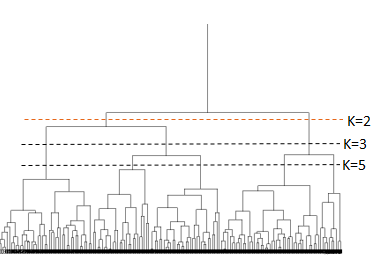
Utilizando la distancia euclídea y la estrategia de enlace completo porque se distinguen mejor los distintos clusters, en cuanto al número de clusters los apropiaos son dos o tres porque los individuos se reparten más uniformemente como vemos en la siguiente tabla:

Trainig set



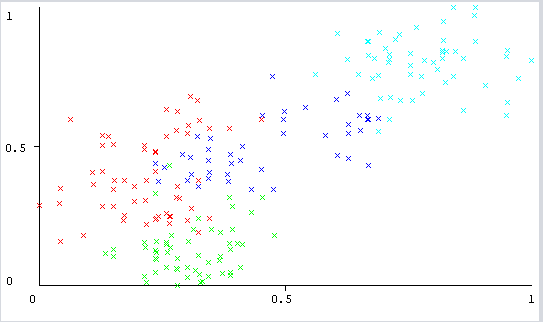
Percentage Split

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NUM.CLUSTERS | 2 | 3 | 4 | 5 |
| CLUSTERED  INSTANCES | 0 43 (60 %)  1 29 (40 %) | 0 22 ( 31 %)  1 21 ( 29 %)  2 29 ( 40 %) | 0 22 (31 %)  1 21( 29 %)  2 19 (26 %)  3 10 (14 %) | 0 22 (31 %)  1 16 (22 %)  2 5 (7%)  3 19 (26 %)  4 10(14 %) |



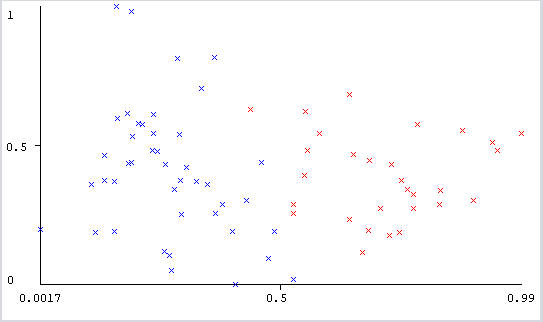
Comparando las tablas podemos ver claramente que se obtiene un mejor resultado utilizando Percentage Split.

En particular vamos a comentar la configuración para cuatro clusters con Training Set y de dos para Percentage Split:

X: longitud\_surco, Y: anchura\_semilla

El clúster que más individuos tuene es el rojo.

Se pueden distinguir bastante bien los cuatro clusters, el primero ( ), el menos homogéneo corresponde con semillas que tienen una anchura media y no les hace falta un surco muy grande, la mayoría son del tipo 1 pero alguna que necesita un surco un poco más grande es del tipo 2. El segundo ( ) corresponde con las semillas más estrechas que necesitan un surco mediano del tipo 1 o 3. El tercer conjunto( ) está compuesto por todas las semillas cuya anchura es pequeña y también necesitan un surco pequeño son en su mayoría del grupo 3 pero hay alguna del tipo 1.Nos damos cuenta de que las semillas del grupo 1 y del 3 tienen aspectos en común por el que podían cultivar en el mismo terreno con modificaciones mínimas. Y el tercer grupo ( ), el más homogéneo son semillas muy anchas que por consiguiente necesitan un surco muy profundo y son del tipo 2.

X: longitud\_semilla

Y: coef\_asimetria

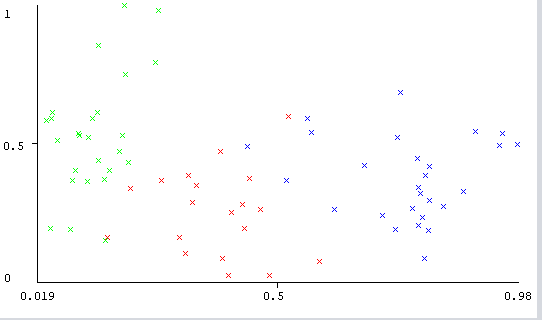
El clúster azul es el más numeroso, en cambio el rojo es el más compacto.

Podemos clasificar de este modo las semillas en dos grupos, distinguibles por la longitud de la semilla, podemos decir que en el segundo ( ) el coeficiente de asimetría es más o menos entable en comparación con el primero( ) en el que puede darse el caso de que haya calores en los extremos, el tipo de semillas del segundo es solo del tipo 2, en contraste con el primer grupo que que tiene de tipo 1 y 3.

De ambas graficas podemos deducir que hay una mayor similitud entre las semillas del tipo 1 y 3, y ambas son perfectamente distinguibles con las del grupo 2

## K-means

Aplicando el algoritmo de k-mean y configurando el método de inicialización con K-means++, los resultados en general son mejores utilizando percentage Split 66 % y en particular el número idóneo de clusters es tres:

X:área, Y:coef\_asimetria

El clúster mas numeroso es el azul el menos el rojo.

Y se ve claramente que el más homogéneo es el rojo y el que menos el azul.

El clúster verde corresponde con las semillas que tienen una menor área y parece que por consiguiente son más asimétricas, estas semillas son del tipo 3. En el clúster rojo vemos que cuando el área está cerca de la media la asimetría va disminuyendo, hasta nieves casi nulos en algunos casos semillas del tipo 1. Y, por último, el clúster azul es de semillas que tienen un área mayor las cuales se mantienen en un coeficiente de asimetría estándar, un muy bueno ni muy malo semillas, del tipo 2, esto se puede deber a como hemos visto antes, que estas semillas son las mas anchas.

# APARTADO 2. CLASIFICACIÓN

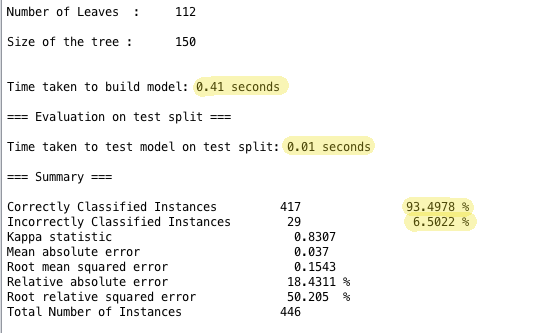
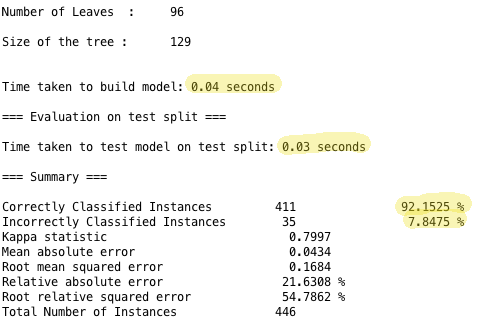
### 1

No hay que normalizar las variables porque todas son nominales, tampoco estandarizar, por la misma razón.

### 2

Ejecutamos J48 con el Percentage Split del 66%, con poda 0.35 y minNumObj = 3.

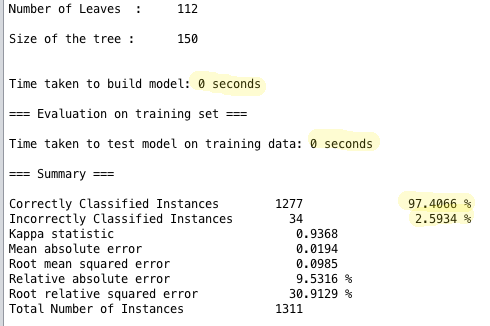
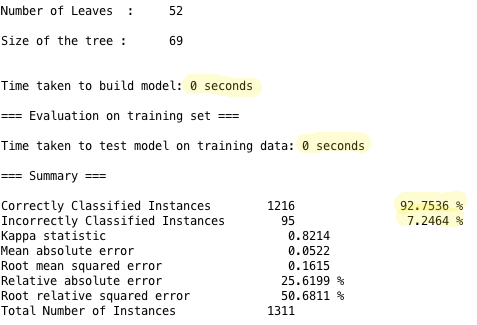
Después de probar varias configuraciones, hemos llegado a la conclusión de que no vale la pena ramificar en una nueva hoja cuando van a clasificarse menos de 3 instancias.



**Percentage Split con Poda=0.35 minNumObj =3.**  **Percentage Split con Poda=0.8 minNumObj =3**

### 3

Ejecutamos J48 con el Training Set, si poda. Tras probar varias configuraciones, hemos visto que es mejor utilizando minNumObj = 2 o 3. Hemos probado también con minNumObj = 5 y el error aumenta considerablemente.



**Training Set sin Poda minNumObj = 2. Training Set sin Poda minNumObj = 5.**

### 4

## Matriz de confusión Percentage Split

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **Recall (Verdaderos positivos)** | **Precisión** |  |
| 323 | 19 | 0 | 0 | 0,944 | 0,988 | **a** |
| 4 | 83 | 0 | 0 | 0,954 | 0,755 | **b** |
| 0 | 5 | 3 | 0 | 0,375 | 0,429 | **c** |
| 0 | 3 | 4 | 2 | 0,222 | 1 | **d** |

Analizando la precisión de las 4 clases, la probabilidad de clasificar coches en las clases a y d es casi del 100% por lo que podríamos confiar en caso de que nos clasifique un coche en alguna de estas clases. En cambio, en b y c, no podemos asegurarnos de que se vaya a hacer una buena clasificación, por lo que tendríamos que consultar a un experto o añadir más instancias.

Analizando los verdaderos positivos, vemos que en las clases a, b tiene un alto porcentaje de acertar positivos, pero en las clases c y d se aciertan muy pocos verdaderos positivos, la mayoría de los positivos no los acierta.

Todos los que ha predicho que iban a ser de la clase d los ha acertado (1), pero ha dado por positivos muchos que no pertenecían a la clase (0,22). Predice que pertenecen a la clase d más de los que deberían, para evitar esto se necesitaría especificar más las características para ser de la clase d.

Puede que una solución a esta mala predicción a la clasificación en las clases c y d sea la introducción de nuevos atributos que describan a estas clases, sin que puedan llegar a afectar a las clases a y b, que sí que realizan una buena predicción. También la introducción de nuevas instancias de las clases c y d, ya que se tienen muy pocas en esta data set.

Esta sería la manera en la que se comportaría con la llegada de nuevos individuos.

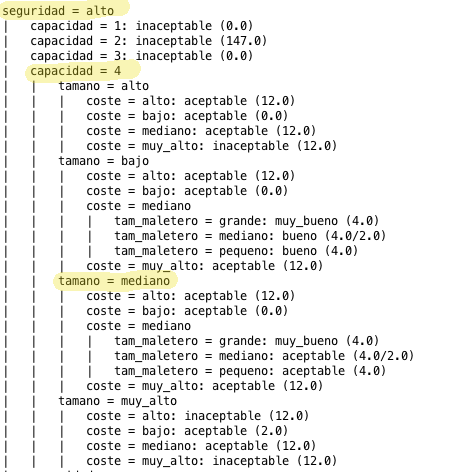
## Matriz de confusión Training set

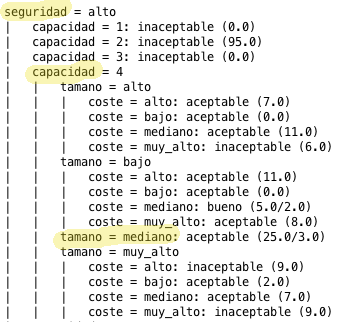
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **Recall (Verdaderos positivos)** | **Precisión** |  |
| 947 | 17 | 1 | 0 | 0,981 | 0,993 | **a** |
| 7 | 288 | 1 | 1 | 0,970 | 0,932 | **b** |
| 0 | 2 | 20 | 1 | 0,870 | 0,833 | **c** |
| 0 | 2 | 2 | 22 | 0,846 | 0.975 | **d** |

En este caso, la matriz de confusión nos indica que el entrenamiento del árbol funciona muy bien ya que el número de verdaderos positivos es casi del 100% y el de precisión también.

La mayoría de los individuos los clasifica en la clase que le corresponde.

### 5

Sobre aprende más el training set porque no valida los datos, solo hace el árbol para los individuos que tiene.



**Rama del árbol con percentage split** **Rama del árbol con training set**

Observando el árbol que se produce, podemos ver como con el training set predice mejor la clase muy bueno, en cambio, con el percentage Split se predice bastante mal la clase muy bueno, tiene más fallos.

Podemos ver que, en la clase muy bueno, muchos de los individuos no se clasifican bien en esta clase, por lo que debemos buscar en el árbol de percentage Split hojas en las cuales la clasificación en alguna clase tenga un mayor número de fallos, correspondiendo estos fallos a los de la clase muy bueno. Estos fallos se deben a que utilizando training set, aplica mayor número de filtros, por lo tanto, hay menor número de errores.

Viendo los resultados generales, es más fiable utilizar training set por el hecho de que tiene mayor número de individuos con los que hace el árbol, pero si en el percentage Split utilizáramos validación cruzada éste sería mejor ya que nos ofrecería una visión más acertada de lo que pasaría en una situación real. Esto es debido a que el percentage Split realiza varios árboles de training, los cuales son distintos unos de otros, que se asemeja al número de individuos que se utilizan en un training set solo, pero añade la mejora de que valida esos árboles como si de un caso real se tratara.

La primera pregunta que se realiza en el nodo raíz es el tipo de seguridad del vehículo, por la cual se lleva a cabo una clasificación bastante importante, separando en tres hojas, de las cuales una es terminal y asigna 437 individuos a la clase inaceptable. Las otras dos ramas continúan la clasificación.

Estos dos otros nodos hijos de la raíz (seguridad alta y media) se pregunta por la capacidad y en ambos, clasifica a los individuos de capacidad =2 como inaceptable, siendo 147 individuos en cada rama.

Con esas pocas preguntas, ya se llevaría más de la mitad de la muestra (473+147+147=731), ya que la muestra tiene 1311 instancias.

La clase que mejor se clasifica es la de inaceptable, que es la que primero se clasifica y de una forma más rápida. A partir de ahí se ramifica más y los nodos terminales como mucho clasifican 12 individuos, a excepción de los de la clase inaceptable, que pueden llegar hasta 16.

Seguridad es la variable más relevante porque genera la clasificación con mayor número de individuos, al igual que capacidad, que también clasifica un gran número de individuos.

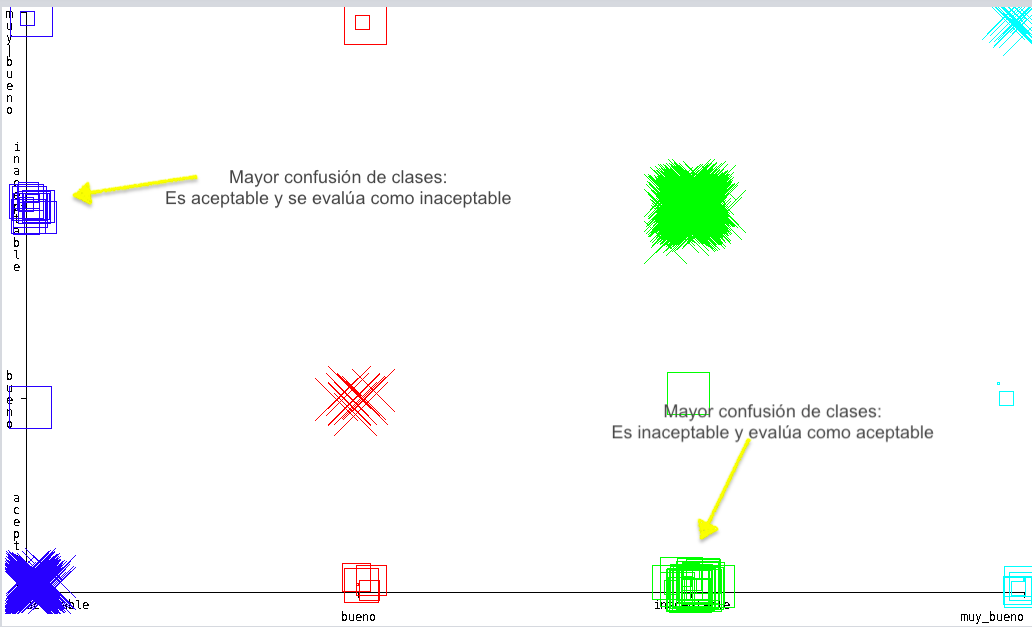
La entropía de todo el árbol es muy baja, ya que las clases de individuos están muy diferenciadas.

El árbol siempre elige variables diferentes a lo largo de las distintas ramas.

## Interpretacion del árbol

* Un coche con seguridad baja es inaceptable, eso es obvio.
* Los coches con seguridad media y capacidad menor o igual de dos, también son inaceptables.
* El tamaño del maletero es un poco irrelevante.
* Los mejores coches son los pequeños.
* Un coche grande no es una buena opción, a no ser que sea de coste mediano o alto, bajo o muy alto no.
* El tamaño va relacionado con el coste. Según este data set, es mejor un coche pequeño y caro, que uno grande y caro.

## Clases mayormente confundibles



En las clases que se produce mayor confusión entre sí a la hora de predecir son en aceptable e inaceptable, donde ambas se confunden en algunas ocasiones.

Observando las características de las predicciones de evaluación que se confunden entre sí, son las que están sobre todo la rama de seguridad mediano, capacidad 4 o más y tamaño de maletero mediano. Esto ocurre porque el tamaño del maletero es irrelevante cuando la seguridad es mediana y la capacidad de personas es grande, no aporta nueva información, sólo ramifica.

Con lo cual, el nodo-pregunta seguridad es el más destacable ya que clasifica exactamente un tercio de la muestra. También cabe destacar el nodo-pregunta capacidad, que clasifica gran parte de la clase inaceptable.

El nodo-pregunta menos relevante es el ya dicho tamaño del maletero, que no clasifica bien e incluso sobraría, ya que confunde la gran parte de sus predicciones.

# APARTADO 3. RegResión

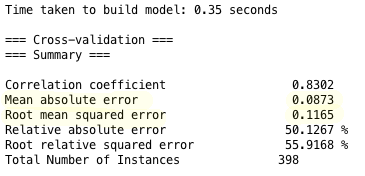
### 1

Sí, debemos normalizar las variables reales, ya que algunas variables pueden tener un rango de valores más amplio que otras con lo cual unas tener más que otras. Además, debemos eliminar las variables nombre, origen y década, ya que son irrelevantes si queremos llevar a cabo el método de regresión.

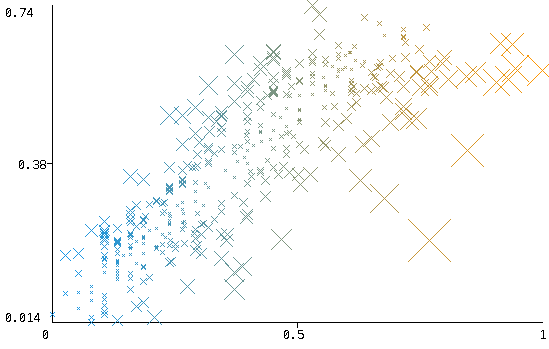
### 2

El valor de k adecuado para la validación cruzada en k partes es 10, ya que es el óptimo para reducir los errores en nuestro data set. Este es el que utilizaremos tanto para el perceptrón multicapa como para el kNN.

### 3

Con la configuración automática del perceptrón multicapa con validación cruzada en 10 partes obtenemos el siguiente error:

Como podemos observar, los errores MAE y RMSE son bastante altos, con lo que seguramente convendría otra configuración más apropiada, si es que queremos utilizar este método.



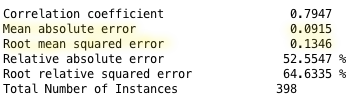
Esta es la gráfica de errores de clasificación, donde X=mpg real e Y=mpg predecido.

Podemos ver como los puntos generan una recta tal que Y = X + error, donde, como puede observarse, el error es de gran tamaño, ya que no se asemeja mucho a Y = X (error = 0). Los puntos están bastante alejados y dispersos.

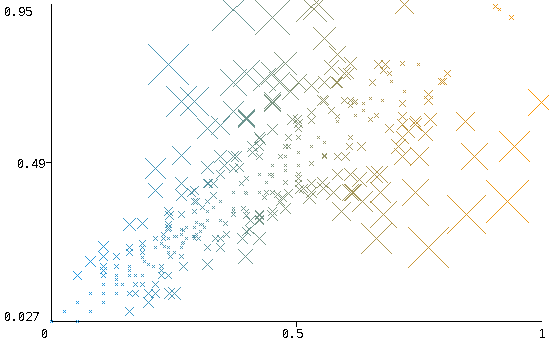
### 4

Intentamos obtener unos errores iguales o menores a los del perceptrón multicapa obtenidos en el apartado anterior.

## Configuración automática del clasificador IBk (**kNN=1**)



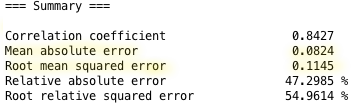
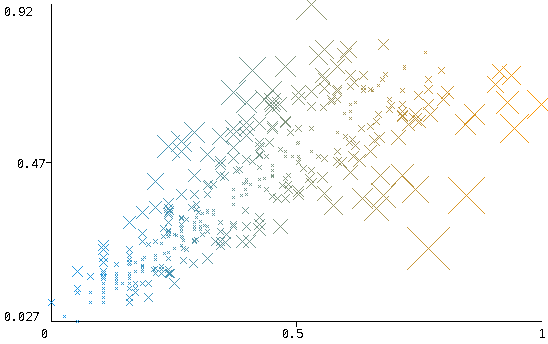
Que no mejora al error cometido en el perceptrón multicapa.



Es más, como podemos observar en los datos y en la gráfica de errores, los puntos están aun más dispersos, con lo que la recta Y = X + error tiene un error aun más grande.

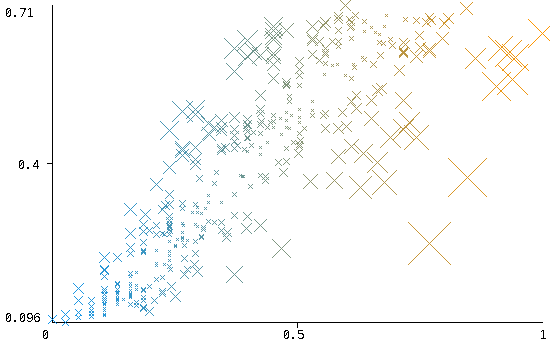
En las siguientes pruebas, la configuración es la de por defecto a excepción de kNN que vamos variando.

## kNN = 2



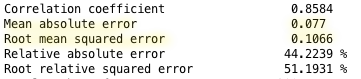
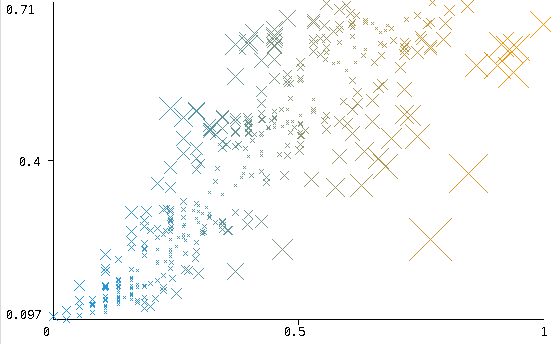
Podemos ver como mejoran los errores con respecto a los del perceptrón multicapa, RMSE algo más que MAE. Aún así, los errores no se han reducido demasiado.

## **kNN = 10**



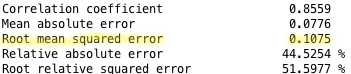
Los errores MAE y RMSE mejoran bastante con respecto a los del perceptrón multicapa.

## **kNN = 11**



Con esta configuración de kNN hemos logrado el menor error posible.

## **kNN = 12**

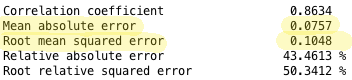


A partir de kNN = 12 hemos podido comprobar que los errores empeoran considerablemente. Sobre todo RMSE, que aumenta más deprisa.

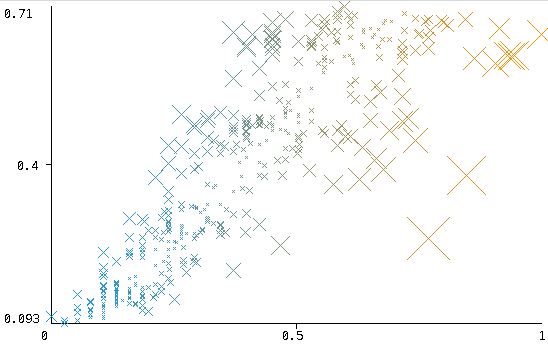
Es por esto por lo que la configuración con kNN = 11 es la mejor.

## **kNN = 11** y **distanceWeighting = 1/distance**

Hemos decidido utilizar un peso que darles a los vecinos relacionado con la distancia, con lo que en la configuración hemos elegido distanceWeighting = 1/distance, de manera que el peso de cada vecino esté relacionado inversamente con la distancia. El resultado ha sido:



Esta ha sido la mejor combinación de variables de configuración que hemos logrado obtener para reducir los errores MAE y RMSE al máximo en el kNN.



Como podemos observar, en la gráfica de errores de clasificación del kNN, la recta Y = X + error que forman los puntos, es más clara que la del perceptón multicapa, los puntos están más concentrados.