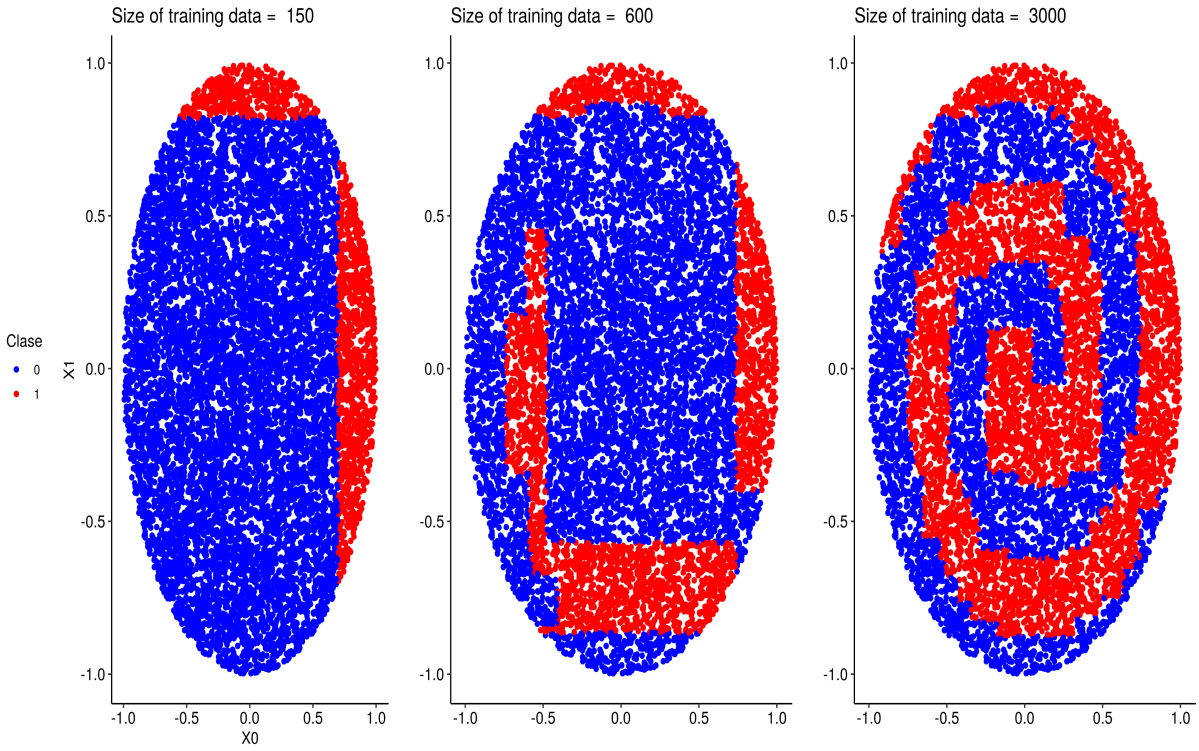
Trabajo práctico 1: Árboles de decisión

Alumno:Pablo Alonso

Ej 4)

Al clasificar el conjunto de test generado con cada árbol, se obtuvieron los siguientes resultados:

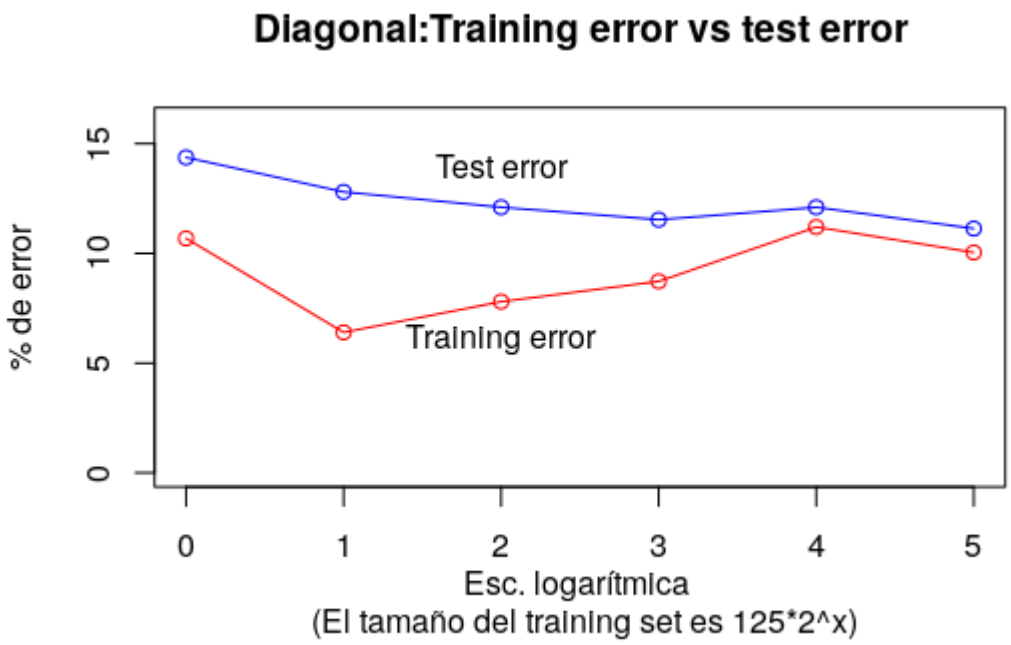


Claramente se observa que al aumentar el tamaño del training set, obtenemos una mejor clasificación de los datos en nuestro universo. Esto se debe a que training sets de mayor tamaño reflejan mejor el concepto que buscamos que nuestro clasificador aprenda sobre el universo en el cual estamos trabajando. ¿Por qué? Sabemos que el algoritmo de construcción del árbol no es más que una búsqueda dentro del espacio de posibles hipótesis que expliquen el concepto que observamos en nuestro universo (osea la separación de las clases). Dado que

en cada paso de la búsqueda determinamos el correspondiente nodo del árbol (el atributo que más ganancia de información brinda) usando medidas estádisticas sobre el conjunto total de los datos, lo que ocurre es que conjuntos de mayor tamaño brindan una información más precisa sobre lo que realmente ocurre en el universo, es decir que tenemos mayor información para decidir cuál es el mejor nodo (o sea cuál es el mejor atributo a testear) en cada paso. Parece natural entonces que a medida que corramos el algoritmo con conjuntos de entrenamieno de mayor tamaño, la búsqueda sea más precisa y que la hipótesis obtenida por el algoritmo sea más próxima al concepto que se quiere aprender.

Ej 5)

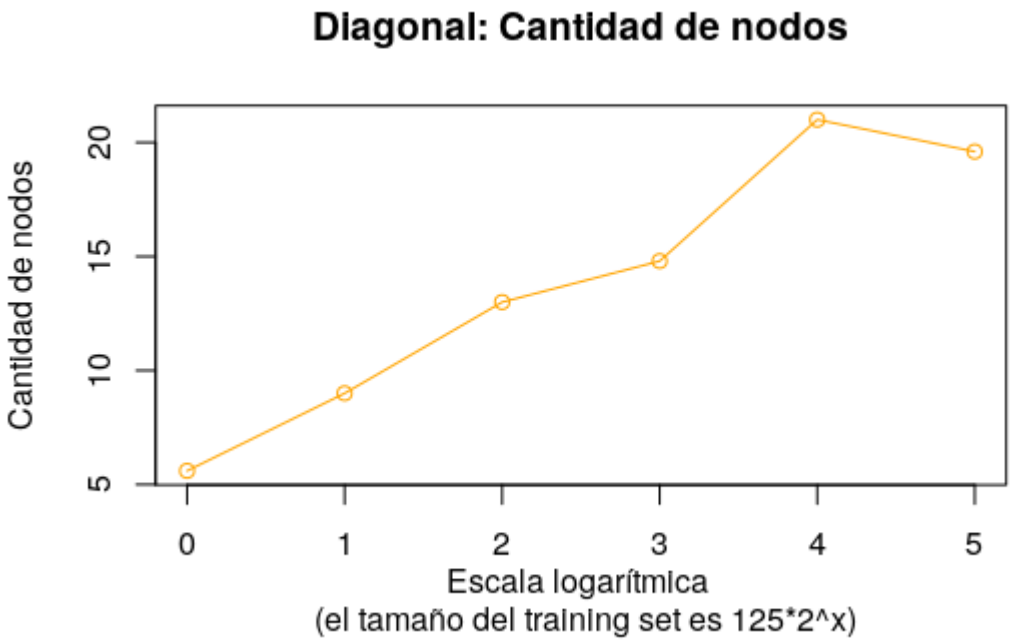
Primero hacemos el experimento sobre los datos diagonales:



Por un lado se observa que las variables no dan la suficiente ganancia de información como para poder clasificar correctamente todos los datos de entrenamiento, por lo que se opta por devolver el valor esperado por los mismos. A medida que el tamaño del training set aumenta, el porcentaje de datos mal clasificados se mantiene entre ciertos valores.

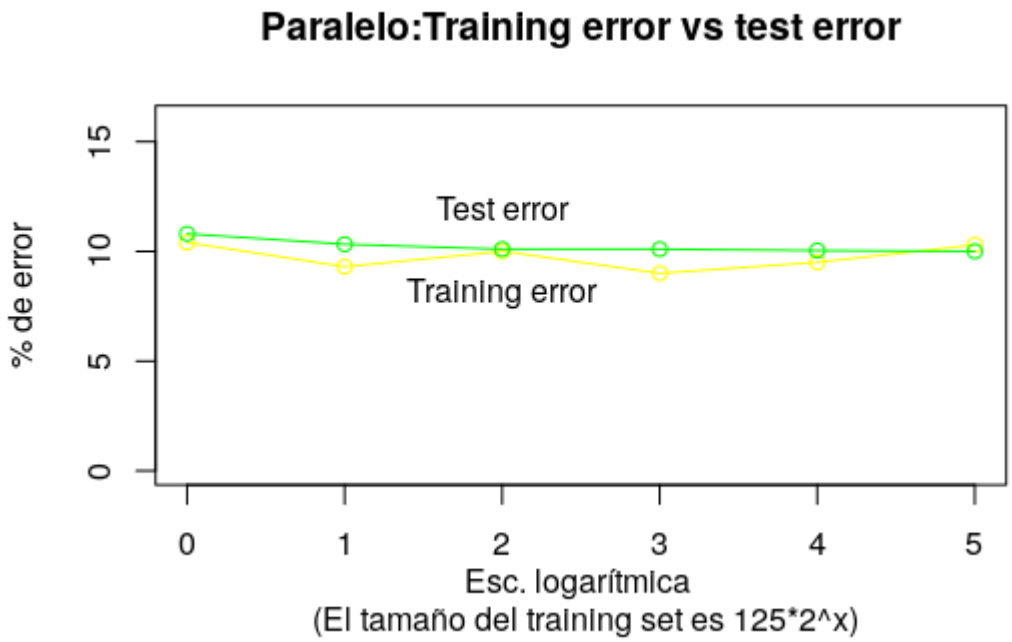
Por otro lado el test error va mejorando porque training sets de mayor tamaño reflejan mejor lo que pasa en nuestro universo y permiten encontrar una mejor hipótesis.

El test error permanece mayor que el training error para los training sets más chicos por la dispersión que hay en los datos pero convergen a un valor parecido a medida que el tamaño del training set se acerca a el tamaño del test set.

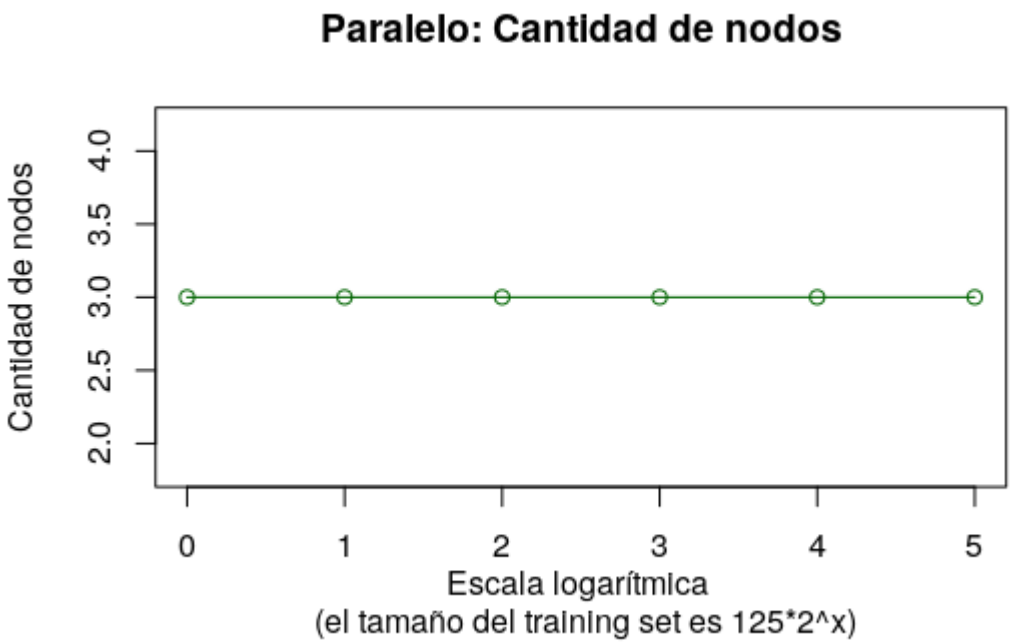


En la gráfica se observa que para algunos valores de x, la hipótesis que nos devuelve el algoritmo es más general (menos nodos = reglas más chicas) y para otros más específica (más nodos = reglas más grandes). A medida que el conjunto de entrenamiento se vuelve más grande, el algoritmo necesita encontrar una hipótesis cada vez más específica y compleja para aprender de los datos de entrenamiento, lo que se refleja en una cantidad mayor de nodos.

Ahora repetimos el experimento para los datos paralelos:



Los errores son muy parecidos. Esto se debe a que al haber poca dispersión en los datos, bastan training sets relativamente chicos para obtener hipótesis que se parezcan al concepto buscado.

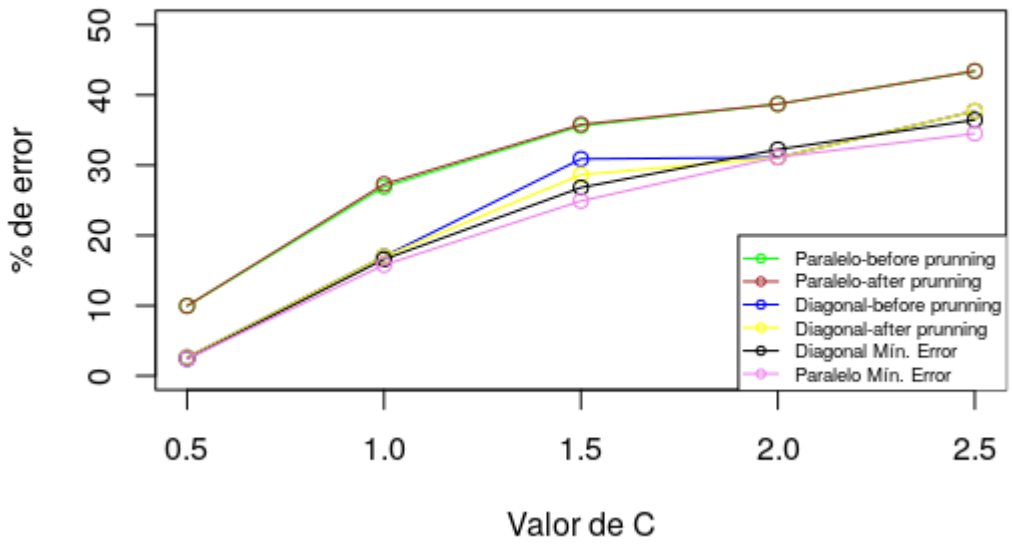


Para los datos paralelos, los árboles se mantienen muy pequeños. Si tenemos en cuenta que el test error y el training error son prácticamente iguales en todos los casos, se podría concluir que se encuentran hipótesis muy sencillas que justifiquen estos datos.

En conclusión, las diferencias encontradas en las hipótesis que obtuvimos a trávez de los datos paralelos y diagonales se debe a la dispersión que hay en las distribuciones. Los árboles encontrados en los datos diagonales tienen mayor desviación estándar, son mucho más complejos y se necesita training sets de mayor tamaño para lograr test errors tan bajos como en los datos paralelos. Teniendo en cuenta que las curvas obtenidas son after pruning, los resultados obtenidos están libres de sobreajuste, lo que se ve reflejado en la constante mejora de test error para ambos casos.

Ej 6)

Se grafican los resultados obtenidos:

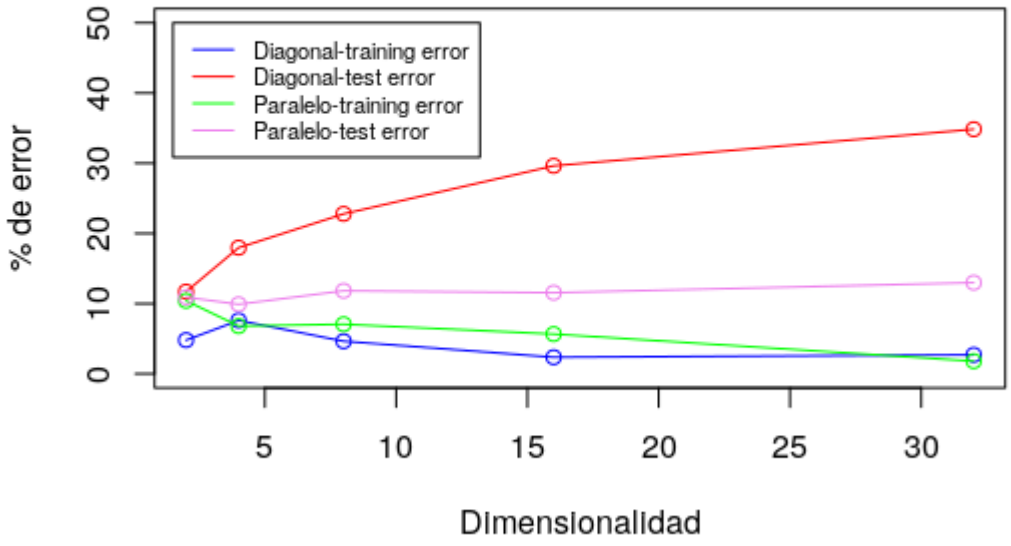


Para obtener los errores mínimos usé un clasificador bayesiano y determiné para cada punto en el test si es más probable clasificar ese punto como de clase 0 o 1, luego el error es el número de puntos mal clasificados sobre el tamaño del test. Cuando el árbol reciba como entrada un punto que es determinado como mal clasificado con el método anterior, su salida tambíen será errónea puesto que los valores de las coordenadas del punto se ajustan más a las condiciones impuestas por un camino en el árbol que conduzca a la clase equivocada. De esta manera nos aseguramos de que el error que obtuvimos al aplicar el proceso a el test completo es mínimo pues al menos esos puntos siempre serán mal clasificados.

Comparando los resultados obtenidos en la gráfica, primero se observa que los errores sobre los datos diagonales son mayores que los datos paralelos, esto debido a que la desviación estándar es mayor por un factor de $\sqrt{5}$. Esta diferencia también se observa en las curvas de error mínimo aunque menos pronunciada. Luego las curvas before-after pruning en ambos casos son prácticamente iguales, esto nos indica que el aumento de dispersión no causa sobreajuste porque la misma dispersión está presente en los training set. Por último, también se observa que la diferencia entre los errores mínimos y los reales también están causados por la dispersión de los datos pues en el caso de los datos paralelos los errores reales que se cometen son prácticamente el mínimo error que se puede cometer, mientras que en los datos diagonales hay una diferencia que se mantiene constante.

Ej 7)

Hacemos el experimento para las dos clases y graficamos los resultados obtenidos:

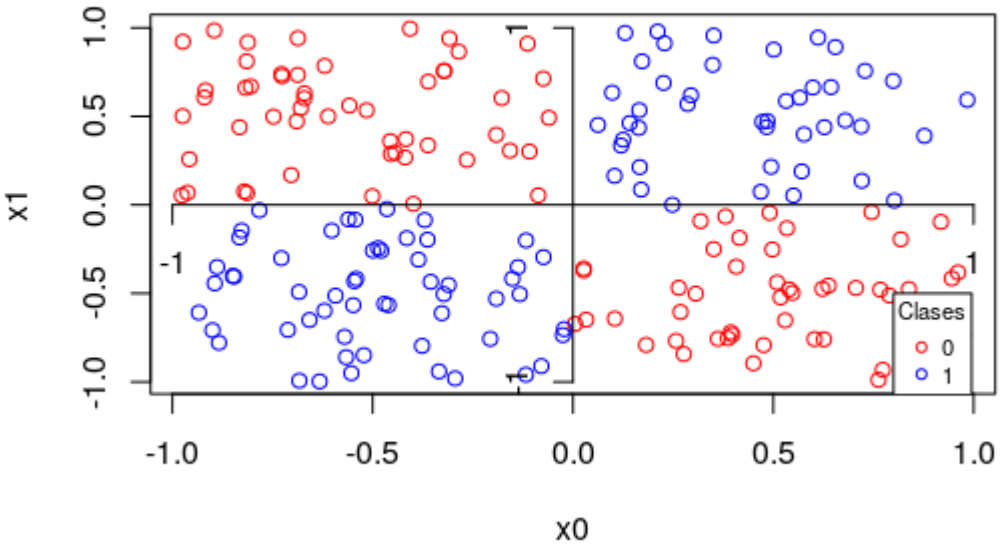


En los datos diagonales se observa un menor training error contra un mayor test error a medidad que se incrementa la dimensionalidad. También hay que tener en cuenta que la desviación estándar en este caso, al dejar la constante c fija, depende de su dimensionalidad, por lo tanto a mayor dimensionalidad también se obtiene una mayor desviación estándar. Se observa que obtenemos hipótesis que se ajustan cada vez mejor al training data pero funcionan cada vez peor sobre los test data.

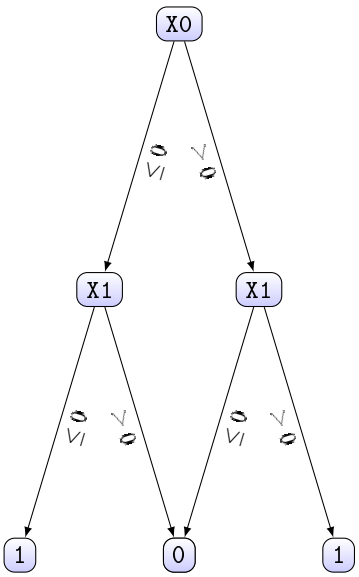
En los datos paralelos, la diferencia es mucho menor. En este caso el aumento de la dimensionalidad baja el training error pero no mejora el test error, ya que este se mantiene prácticamente igual en todas las dimensiones. La conclusión en este caso es que si se deja una desviación estándar fija pero se aumenta la dimensionalidad se obtienen mejoras en el training error pero no en test error.

Ej 8)

Primero graficamos las clases:



El árbol más simple que clasifica correctamente todos los puntos es el siguiente:



En contraste, el árbol que C4.5 devuelve solo tiene el siguiente nodo:

Se observa que el árbol que devuelve C4.5 consiste en un solo nodo, es decir la clase 0. Si corremos C4.5 con la opción -v i (i $\in\{1,2,3\}$), el verbose nos informa de que ninguna de las variables tiene suficiente ganancia de información y no se pueden hacer splits de los datos. Entonces se aplica el caso base que es devolver la clase con mayor frecuencia. Como tenemos la misma cantidad de observaciones de cada clase es igual que el algoritmo determine que la clase esperada es 0 o 1 porque siempre obtendremos el 50% de error.

Mientras que un humano es capaz de encontrar un árbol que no comete errores, el algoritmo solo encuentra uno que resuelve bien la mitad de los casos.