# Trabajo Práctico N°1: Árboles de Decisión

## Introducción al Aprendizaje Automatizado


Alumno: Navall, Nicolás Uriel. N-1159/2.

4) En la figura 1.1 se puede observar como la cantidad de datos dada en el set de entrenamiento (150) no es suficiente para modelar el comportamiento deseado del espiral. Lo mismo en la figura 1.2 con el modelo entrenado por los 600 datos.

El modelo generado por el conjunto de 3000 datos que puede verse en la figura 1.3, al ser una cifra más cercana al conjunto universo de nuestro problema, tiene los datos suficientes para generar un árbol capaz de aproximarse al espiral deseado mucho más que los dos anteriores, pero no obtiene un resultado exactamente igual al buscado. Esto se debe a que el árbol generado utiliza las coordenadas cartesianas para discernir la clase, mientras que la clasificación realizada cuando generamos los puntos utiliza coordenadas polares, por lo que no importa el tamaño del set de entrenamiento, nunca va a ser un espiral perfecto por que este tiene que ser generado por acotaciones en el eje x e y.

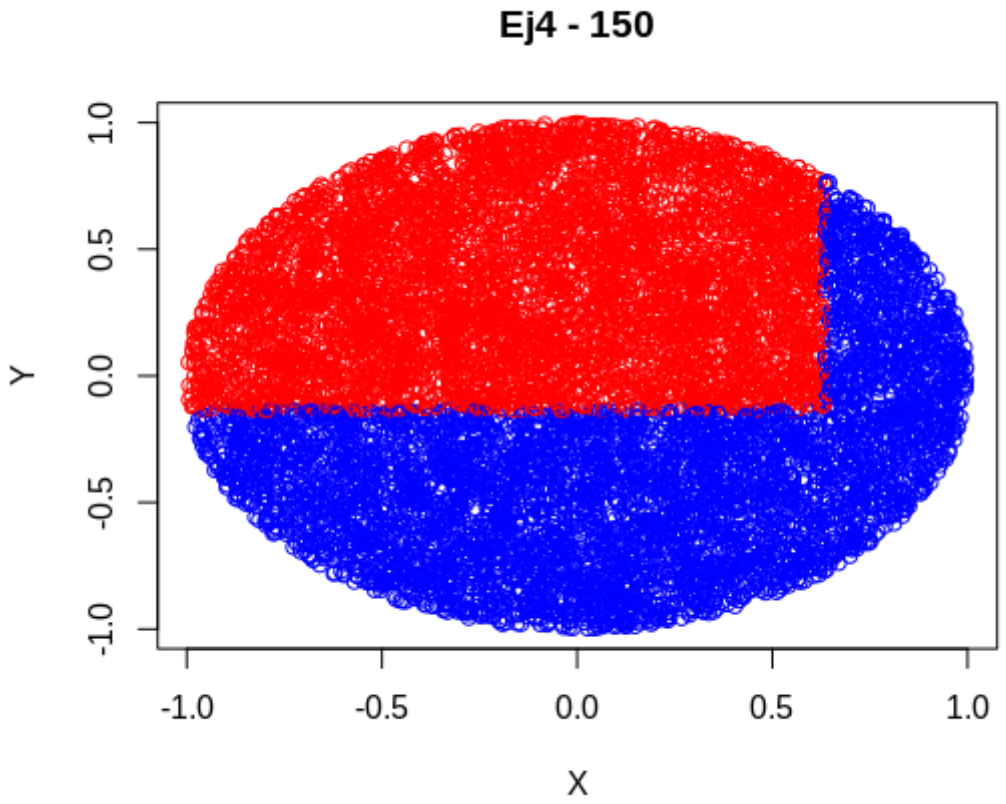



*Figura 1.1*

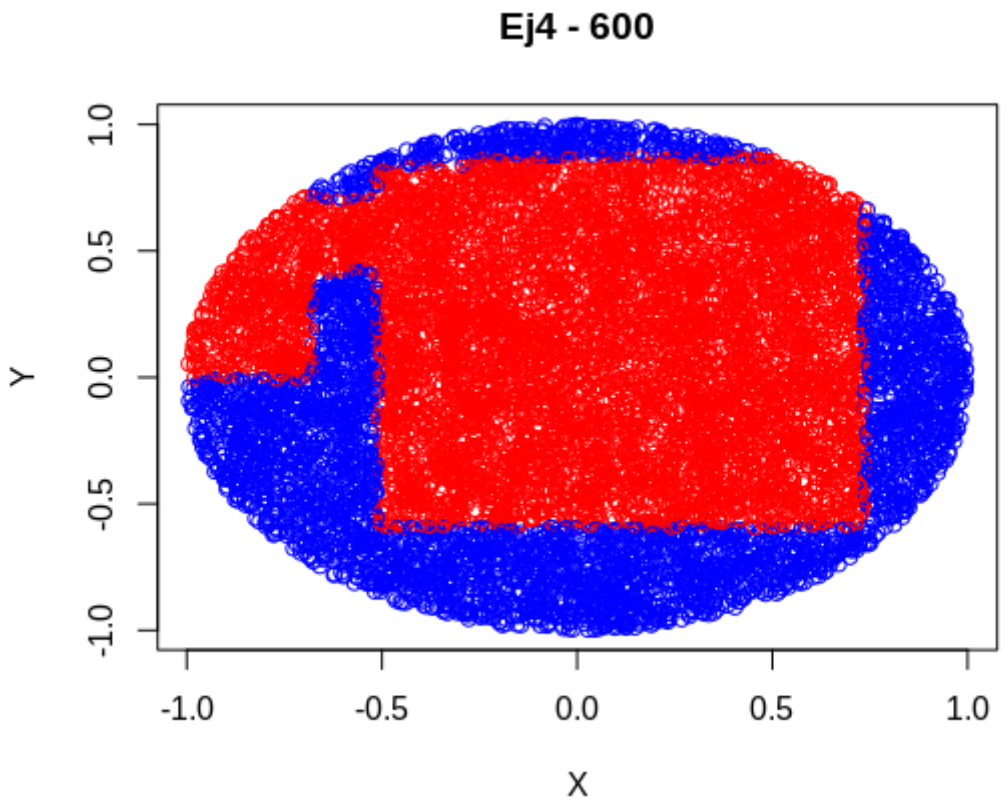



*Figura 1.2*

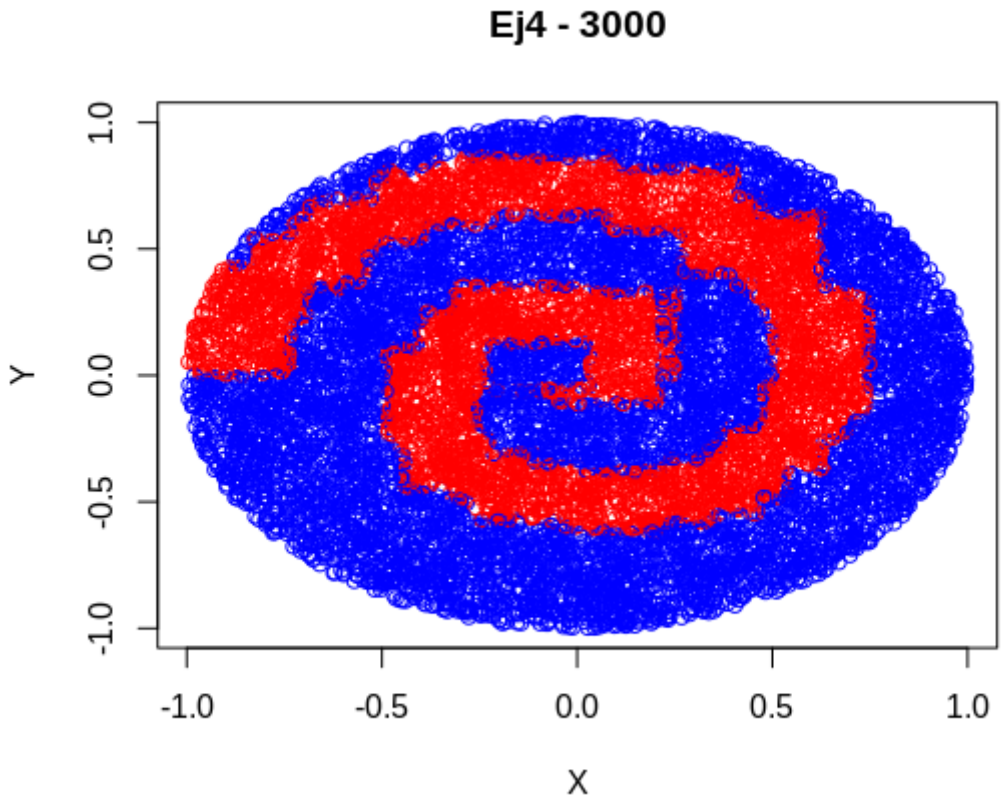



*Figura 1.3*

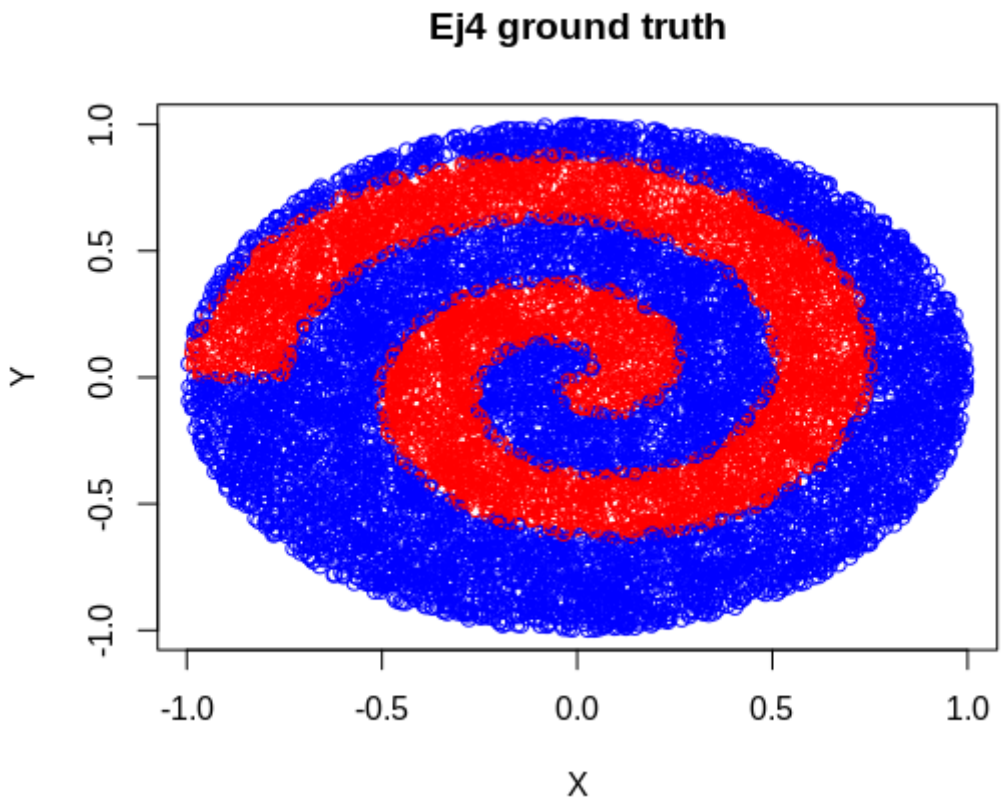



*Figura 1.4*

5) Podemos apreciar en la figura 2.1 como la línea “On diagonal testing data” baja al aumentar el tamaño del conjunto de entrenamiento, lo que refleja cómo aumentar el conjunto de entrenamiento ayuda a definir de mejor manera el comportamiento deseado. Además, al observar la línea “On diagonal training data” se puede observar como, mientras el error en el conjunto de testing desciende mientras aumentamos el tamaño del conjunto de entrenamiento, el error sobre el conjunto de training aumenta, lo que implica que el árbol obtenido está mejor adaptado al set de datos de testing que el de entrenamiento, lo cual es lo deseado ya que el ruido en el conjunto de entrenamiento puede resultar en un clasificador con un error muy grande en el conjunto de testing (que en nuestro caso es el universo).

En las líneas “On parallel testing data” y “On parallel training data” podemos observar un comportamiento similar, y como estas tienden a converger a un error del 10%.
Si observamos el tamaño del árbol generado por el set de datos paralelos en la figura 2.2 podemos ver como el árbol se mantiene relativamente chico con respecto al tamaño del árbol en el set de datos diagonales. Esto se debe a que el problema a clasificar es más simple en el set de datos paralelos que diagonales, lo que resulta en un árbol más chico para describir la clasificación que el problema necesita.

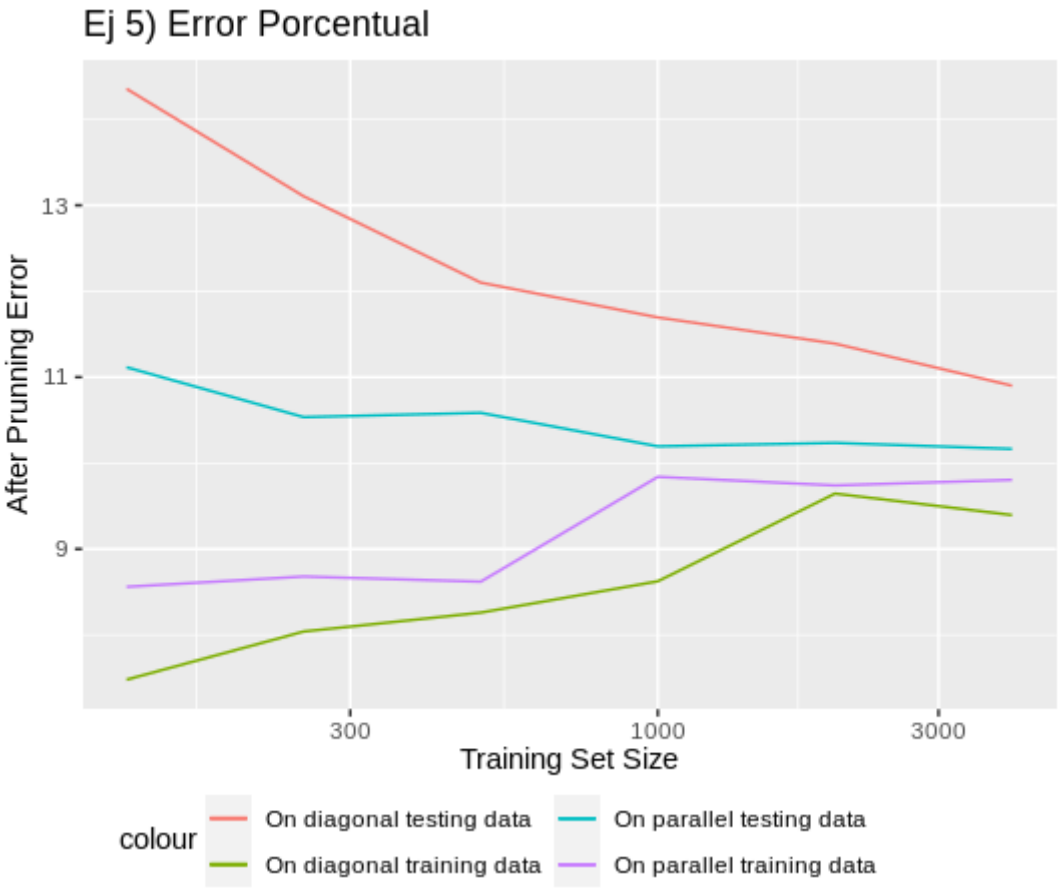



*Figura 2.1*

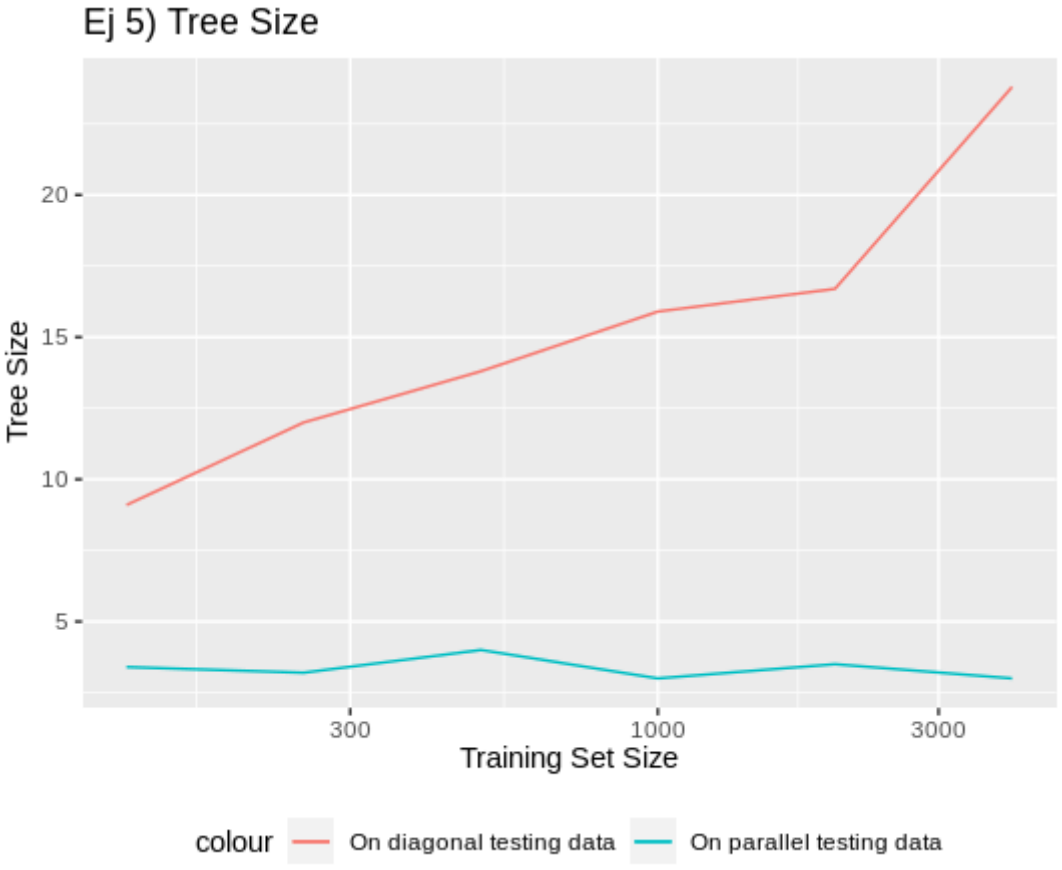



*Figura 2.2*

6) En la gráfica puede observarse como el error en el conjunto de datos parallelo obtenido con el arbol “after prunning” es levemente menor al obtenido con el arbol “before prunning”, lo cual muestra que el podado está funcionando como debe, ayudando a reducir el error producido por el árbol generado.

Al fusionar las dos gaussianas aumentando el valor de C estamos inyectando más ruido en los datos, por lo cual es de esperarse que el error aumente conforme aumenta el valor de C.

Para obtener los valores de error mínimos defini un clasificador ideal que clasifica los puntos dependiendo la distancia de estos con respecto a los orígenes de las dos distribuciones gaussianas de cada problema ((1,1,1,1,1) y (-1,-1,-1,-1,-1) para el problema diagonal, y (1,0,0,0,0) y (-1,0,0,0,0) para el problema paralelo) y devuelve la clase asociada al origen más cercano, ya que el punto tiene más probabilidades de pertenecer a ese origen.

Nuevamente, como el problema sobre los datos paralelos es más “simple” que el problema sobre los datos diagonales, el árbol generado por C4.5 tiene un error más cercano al error mínimo, siendo este una alternativa aceptable al modelo que genera el mínimo error, en comparación con el árbol generado por los datos diagonales.

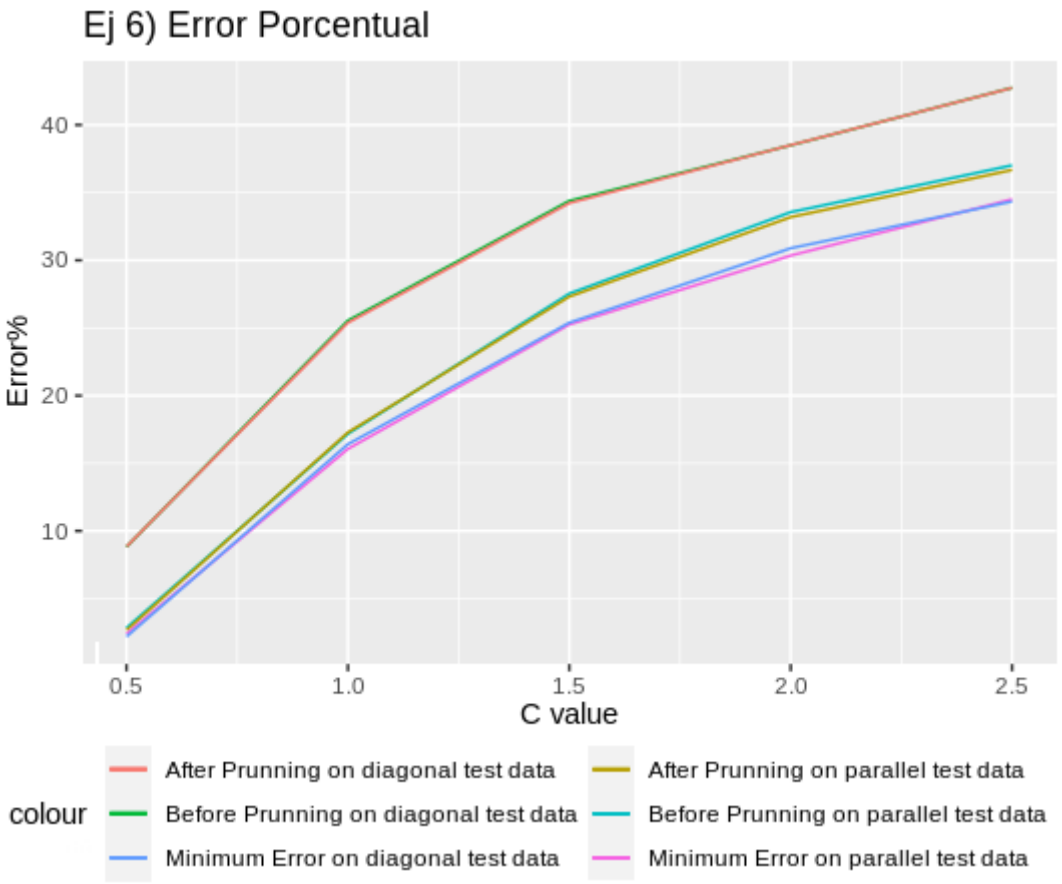



*Figura 3*

7) El incremento en dimensiones en el set de entrenamiento complejiza el problema al agregar más variables, haciendo que la hipótesis necesaria para un clasificador sea más compleja, aumentando así el error obtenido en el árbol generado sobre el set de testeo.

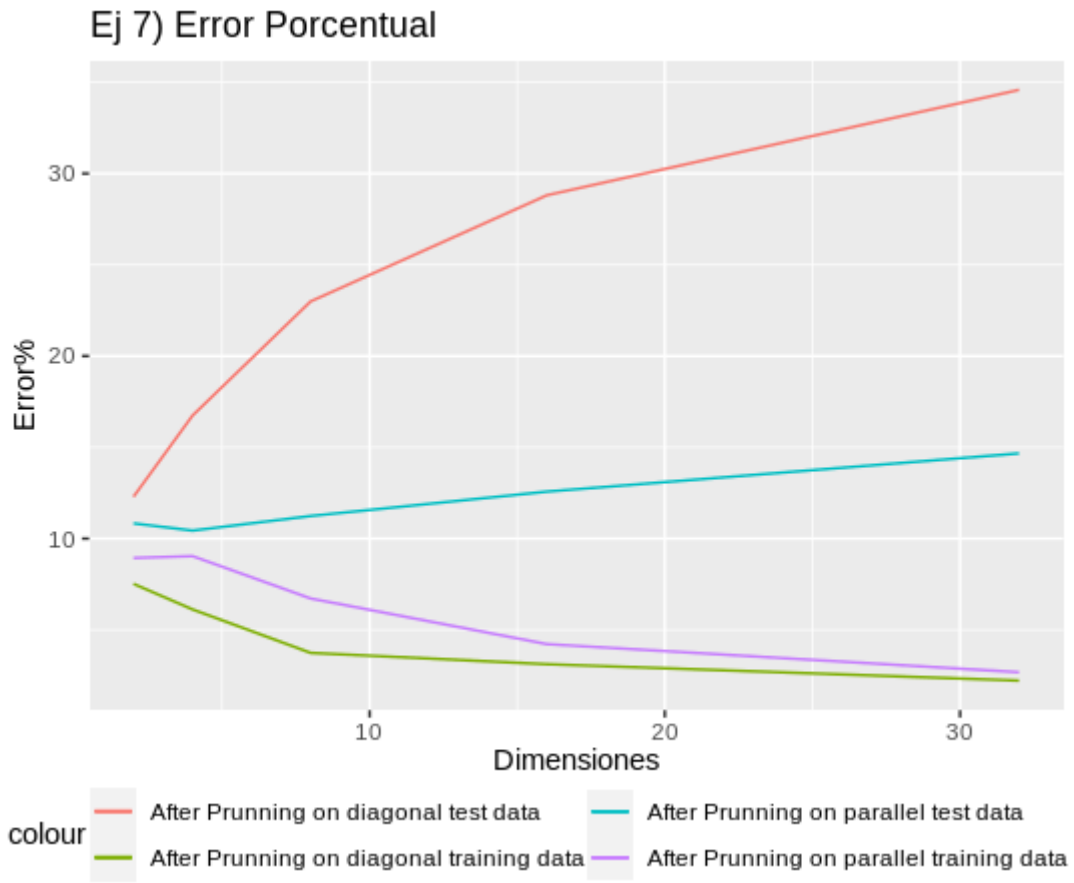
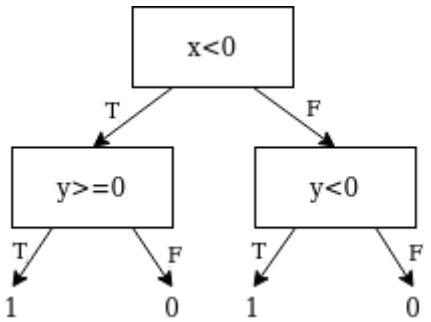



*Figura 4*

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




El árbol generado por C4.5 es un solo nodo que siempre devuelve 0, dando un error porcentual del 50% en el set de training (equivalente a random).

Lo que sucede es que el atributo ‘x’ e ‘y’ no realizan una buena separación de clases por sí mismos, es decir solo viendo el valor ‘x’ de un punto no podemos inferir a qué clase este pertenece, ya que tienen iguales chances de estar clasificado como 0 o 1. Lo mismo sucede con ‘y’. Por lo cual la ganancia de información de los atributos ‘x’ e ‘y’ es muy baja (y la entropía muy alta).