**Diseño de experimentos - Decision Tree Machine Learning**

**Planeación y realización**

Delimitación del objeto de estudio:

El fenómeno de estudio de este experimento es el comportamiento de dos implementaciones diferentes de Decision Tree, la primera creada manualmente y la segunda obtenida de la librería de Accord.Net. Esto es importante debido a que es necesario tener claridad en cuál es el más eficaz en un entorno real, en donde es común hallar una gran variedad de casos y es necesario clasificar diferentes cantidades de datos con el menor tiempo y la máxima correctitud posible.

Variables de respuesta

Las variables de respuesta que serán tomadas en cuenta en este experimento son el tiempo y la correctitud, ya que son la principal métrica para evaluar el desempeño de una implementación de Decision Tree. Las unidades utilizadas para la medición del tiempo serán los milisegundos, ya que proveen la suficiente precisión buscada para el desarrollo de las comparaciones.

Nuestra herramienta de evaluación para el tiempo será el método Stopwatch de C# y para la correctitud, el porcentaje de aciertos respecto al total.

Debido a que la medición varía con esta herramienta, se probará cada caso 4 veces y se utilizará el tiempo promedio entre cada caso, logrando así una medida más precisa del desempeño.

Factores a estudiar

Factores controlables:

* Implementación del decision Tree utilizada
* porcentaje de los datos del dataset utilizado para entrenamiento
* Lenguaje de programación
* Cantidad de procesos ejecutados en segundo plano
* Cantidad de ram del pc (haciendo uso de máquinas virtuales)
* Cantidad de variables utilizadas para la clasificación

Factores no controlables:

* Procesos fundamentales del sistema operativo
* Nivel de fragmentación del disco duro del computador donde se ejecuta el algoritmo

Factores estudiados:

* Implementación del Decision Tree utilizada
* Porcentaje de los datos del dataset utilizado para entrenamiento
* Cantidad de ram del pc (haciendo uso de máquinas virtuales)

Niveles de cada factor

|  |  |
| --- | --- |
| Variables | Niveles |
| Implementación del Decision Tree utilizada | 2 niveles |
| Porcentaje de los datos del dataset utilizado para entrenamiento | 4 niveles |
| Cantidad de ram del pc | 3 niveles |

Implementación del Decision Tree utilizada.

|  |  |
| --- | --- |
| Nivel | Valor |
| 1 | Implementación manual |
| 2 | Implementación con librería Accord |

Porcentaje de los datos del dataset utilizado para entrenamiento

|  |  |
| --- | --- |
| Nivel | Valor |
| 1 | 20% |
| 2 | 40% |
| 3 | 60% |
| 4 | 80% |

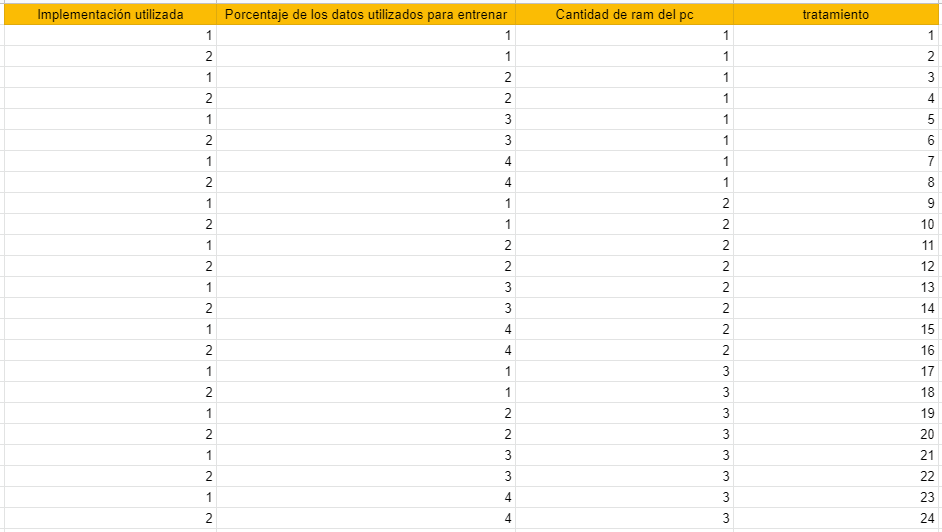
Cantidad de ram del pc

|  |  |
| --- | --- |
| Nivel | Valor |
| 1 | 4 GB RAM |
| 2 | 6 GB RAM |
| 3 | 8 GB RAM |

**Organización del trabajo experimental**

El experimento será realizado en solo una computadora, para evitar cambios en los resultados producto de las diferentes especificaciones de cada uno. En cada experimento se harán 4 repeticiones de cada implementación del Decision Tree según los tratamientos definidos en la siguiente tabla:

https://docs.google.com/spreadsheets/d/1RuDCKnhJvwbzD51M5dYCw4XpOMzoLV0L4g2cbKDAofc/edit?usp=sharing



El tiempo solo será tomado durante el proceso de clasificación, ya que en el código también se exportarán los datos.

Los pasos que se llevarán a cabo en cada uno de los tratamientos serán los siguientes:

1. Entrenar el árbol con un porcentaje de los datos del dataset

2. Testear el árbol con los registros restantes del dataset, tomando el tiempo que se demora y la correctitud que se obtiene.

3. Repetir los pasos 1 y 2, 4 veces.

4. Generar una tabla con los tiempos y correctitud de cada iteración, además de sus promedios.

5. Exportar la tabla en un archivo formato csv.

Los resultados del experimento serán registrados en un archivo con formato .csv, donde irán el tiempo utilizado en milisegundos, el porcentaje de correctitud y el tratamiento al cual pertenecen esos resultados. El documento se dividirá en dos hojas: En una estarán todos los tiempos tomados y la correctitud en cada tratamiento, y en otra estarán los promedios de estos.

**Análisis de resultados**

**ANOVA MÚLTIPLES FACTORES EN R COMMANDER**

**Variable de respuesta:** Correctitud

En primer lugar comparamos nuestros factores estudiados con la variable de respuesta correctitud.

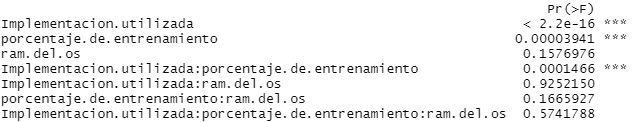
**Planteamiento de hipótesis:**

H0 = Las medias son iguales

H1 = Al menos una media es diferente.

Rechazo H0 si y sólo si valor P < 0.05 (nivel de significancia).

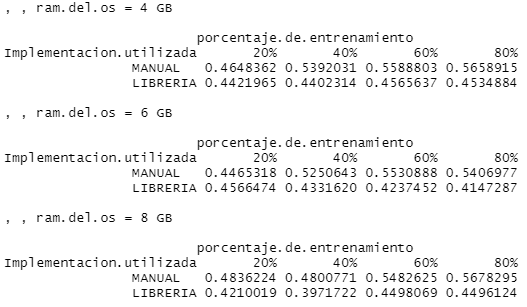
**Resultado en R commander**



**Interpretación:**

La implementación utilizada y el porcentaje de entrenamiento son factores relevantes para la variable de respuesta correctitud ya que, el valor P es menor a 5%. Por lo cual, con un nivel de significancia del 5% podemos concluir que existe suficiente evidencia para rechazar H0, entonces se acepta H1, es decir, la variable de respuesta correctitud se ve muy influenciada por la implementación usada y el porcentaje de entrenamiento porque existe por lo menos una media de correctitud que es estadísticamente diferente en estos factores.

Por otro lado, el factor memoria ram es poco relevante para la variable de respuesta correctitud ya que, su valor P es mayor a 5%. Por lo cual, con un nivel de significancia del 5% podemos concluir que no existe suficiente evidencia para rechazar H0, entonces no se puede aceptar H1, es decir, la variable de respuesta correctitud no se ve influenciada por la memoria ram que se utilice porque las medias de correctitud son estadísticamente iguales en este factor.

En la siguiente comparación se puede corroborar la interpretación anterior, ya que independientemente de la memoria ram que se esté utilizando (4,6 o 8 GB), las medias son estadísticamente iguales (no hay mayor diferencia).

**Variable de respuesta:** Tiempo usado

Ahora compararemos nuestros factores estudiados con la variable de respuesta tiempo usado.

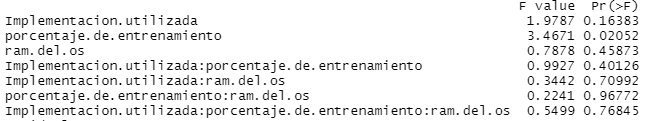
**Planteamiento de hipótesis:**

H0 = Las medias son iguales

H1 = Al menos una media es diferente.

Rechazo H0 si y sólo si valor P < 0.05 (nivel de significancia).

**Resultado en R commander**

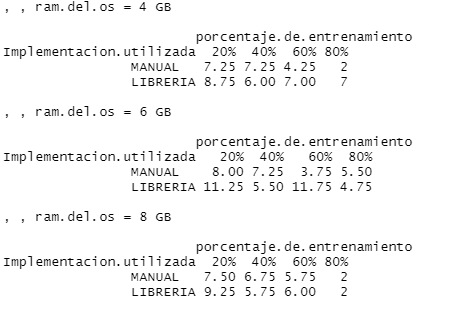
****

**Interpretación:**

La memoria ram y la implementación usada son factores poco relevantes para la variable de respuesta tiempo usado ya que, el valor P es mayor a 5%. Por lo cual, con un nivel de significancia del 5% podemos concluir que no existe suficiente evidencia para rechazar H0, entonces se rechaza H1 y se acepta H0, es decir, la variable de respuesta tiempo usado no se ve influenciada por la implementación usada y el porcentaje de entrenamiento porque las medias de tiempo usado son estadísticamente iguales en estos factores.

Por otro lado, el factor porcentaje de entrenamiento es relevante para la variable de respuesta tiempo usado ya que, su valor P es menor a 5%. Por lo cual, con un nivel de significancia del 5% podemos concluir que existe suficiente evidencia para rechazar H0, entonces se acepta H1, es decir, la variable de respuesta tiempo usado se ve muy influenciada por el porcentaje de entrenamiento que se utilice porque existe por lo menos una media de tiempo usado que es estadísticamente diferente en este factor.

En la siguiente comparación se puede corroborar la interpretación anterior, ya que en cada combinación de factores las medias de tiempo usado varían bastante dependiendo el porcentaje de entrenamiento utilizado en la mayoría una diferencia que parece ser inversamente proporcional entre más porcentaje de entrenamiento menor es la media de tiempo usado.



Conclusiones del experimento:

* El porcentaje de correctitud no se ve afectado por la cantidad de RAM que se utilice.
* El porcentaje de correctitud de la implementación manual del árbol de decisión es mayor a la implementación de librería.
* El porcentaje de correctitud es mayor entre mayor sea el porcentaje de entrenamiento del árbol de decisión.
* El tiempo usado en clasificar el resto del dataset es inversamente proporcional al porcentaje de entrenamiento, entre mayor sea el porcentaje de entrenamiento menor será el tiempo usado en clasificar el resto del dataset.
* El tiempo usado en clasificar el resto del dataset no se ve afectado por la memoria RAM utilizada.
* El tiempo usado en clasificar el resto del dataset no se ve afectado por la implementación del árbol de decisión utilizada.