Instituto Tecnológico de Costa Rica
Estructura de Datos
Proyecto 1
Autores: Andrés Chavarría Elizondo y Joxan Andrey

I. Resumen

Se desean investigar y probar algoritmos de machine learning que pertenezcan a los dos grupos: Supervisado y No supervisado; Además se busca observar el comportamiento de este algoritmo frente a entradas de datos que varíen en tamaño para valorar su desempeño. Ante el problema de comparar el rédito y eficacia de cada algoritmo existen se realizaron 3 procesos de análisis.

En la medición empírica se tomaron los resultados de asignaciones, comparaciones y líneas ejecutadas para diversas entradas de datos, además se realizó un cuadro de factores de talla a partir de estos resultados. La medición analítica busca dar un valor de "1" a cada instrucción ejecutada, por lo que se obtiene un polinomio f(n) donde n es la cantidad de datos a procesar. Por último, la medición gráfica busca exponer el crecimiento de la asignaciones y comparaciones respecto a la entrada de datos, esto por medio de gráficos. Lo anterior con miras a obtener una descripción más reveladora del comportamiento de cada algoritmo.

Finalmente se provee una evaluación de los resultados obtenidos a partir del análisis de complejidad. Además se busca indicar los impactos de la propuesta del proyecto, marcar el territorio de para investigaciones futuras y aspectos a mejorar.

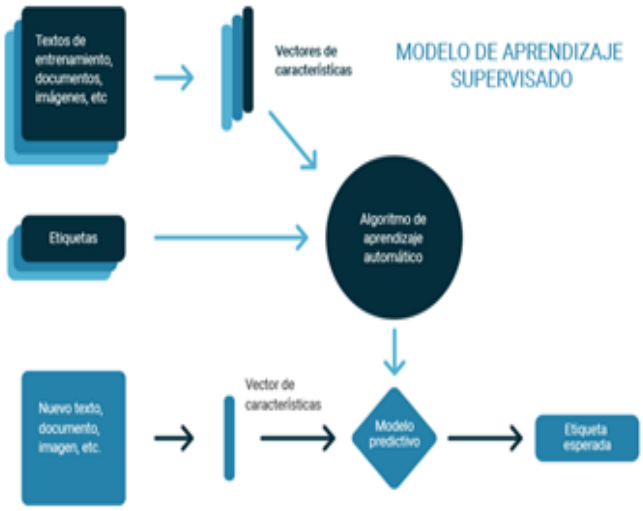
II. Introducción

Los algoritmos de Machine Learning (ML) pretenden que las computadoras aprendan a tomar decisiones sin la necesidad de ser programadas explícitamente. Es por ello que hoy en día podemos escuchar acerca de autos de conducción autónoma, agentes virtuales de atención al cliente (chatbots), sistemas de recomendación y recolección de datos (Netflix, Google, Facebook).

Dependiendo de las necesidades del problema, el ambiente en el que se van a desenvolver y los factores que afectarán la toma de decisiones, podemos encontrar distintos tipos de algoritmos de aprendizaje, entre los cuales vamos a hablar de 2 de ellos: supervisado, no supervisado.

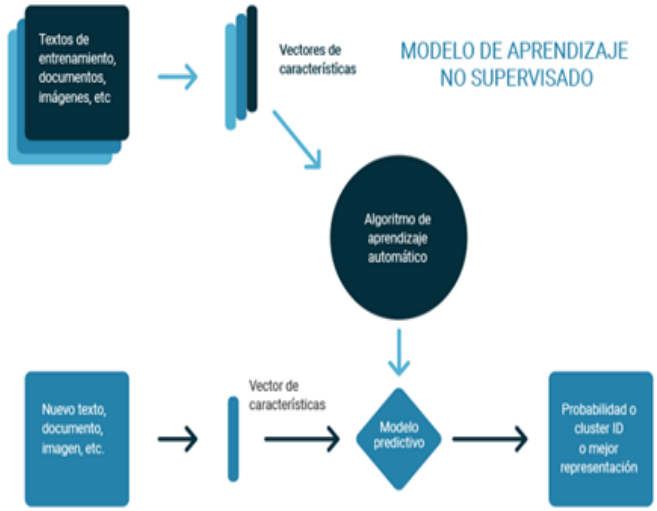
III. Trabajos Relacionados

Los algoritmos supervisados permiten, a partir de los valores de ciertos atributos y su resultado respecto a los mismos (parte supervisada), predecir el resultado de registros nuevos según los atributos que posea. En el caso de la presente investigación se utiliza una tabla de datos con 4 atributos y un resultado para verificar si un billete de banco es autentico.



Para entender mejor el funcionamiento de un algoritmo supervisado, específicamente el caso de árbol de decisión se utilizó como referencia la investigación de Rubio, G y Taboada, M sobre la capacidad productiva de suelos en la región pampeana. En esta investigación, la toma de decisión sobre la aptitud de uso agrícola de un suelo determinado, se lleva a cabo recurriendo a un cursograma, que va proponiendo opciones a ser seleccionadas en base a propiedades fácilmente detectables del paisaje y del perfil de los suelos. En un primer nivel, el cursograma recurre a atributos del paisaje y del relieve, es decir externos a los suelos, y que como tales pueden ser detectados a campo, o haciendo uso de imágenes y fotografías aéreas. En un segundo nivel, el cursograma recurre a características morfológicas del perfil de los suelos, que surgen de la inspección visual de los mismos. Siguiendo un esquema binario (si-no), se termina confeccionando un árbol de decisión, que concluye con un tercer nivel de tres posibles categorías de aptitud agrícola (apto, parcialmente apto, no apto)[1].

Los algoritmos de aprendizaje no supervisado trabajan de forma muy similar a los supervisados, con la diferencia de que éstos solo ajustan su modelo predictivo tomando en cuenta los datos de entrada, sin importar los de salida. Es decir, a diferencia del supervisado, los datos de entrada no están clasificados ni etiquetados, y no son necesarias estas características para entrenar el modelo. Dentro de este tipo de algoritmos, el agrupamiento o clustering en inglés, es el más utilizado, ya que particiona los datos en grupos que posean características similares entre sí.



Para nuestro caso de estudio y analisis elegimos el algoritmo K-means, este tipo de algoritmos de aprendizaje no supervisado busca patrones en los datos sin tener una predicción específica como objetivo (no hay variable dependiente). En lugar de tener una salida, los datos solo tienen una entrada que serían las múltiples variables que describen los datos.

K-means necesita como dato de entrada el número de grupos en los que vamos a segmentar la población. A partir de este número k de clusters, el algoritmo coloca primero k puntos aleatorios (centroides). Luego asigna a cualquiera de esos puntos todas las muestras con las distancias más pequeñas. A continuación, el punto se desplaza a la media de las muestras más cercanas. Esto generará una nueva asignación de muestras, ya que algunas muestras están ahora más cerca de otro centroide. Este proceso se repite de forma iterativa y los grupos se van ajustando hasta que la asignación no cambia más moviendo los puntos. Este resultado final representa el ajuste que maximiza la distancia entre los distintos grupos y minimiza la distancia intragrupo.

IV. Solución propuesta

Ante el problema de comparar el rédito y eficacia de cada algoritmo existen se realizaron 3 procesos de análisis. La medición empírica, medición analítica y la medición gráfica, juegan un papel importante a la hora de examinar la complejidad y eficacia de un algoritmo frente a una entrada n de datos.

El algoritmo de árbol de decisión utiliza un conjunto de datos de billetes. Consiste en predecir si un billete determinado es auténtico dada una serie de medidas tomadas de una fotografía. Posee 4 variables y una quinta variable que determina la autenticidad [2]. Este algoritmo posee una extensión que permite tomar los 1372 registros originales y producir un .csv con n cantidad de registros con datos verosímiles basados en los originales.

El algoritmo k-means puede controlar la cantidad de datos de entrada y agrupa sin supervisión en 3 grupos (cantidad arbitraria de grupos elegida por el programador).

En definitiva, para ambos algoritmos se realizan los tres métodos de medición. Esto con el objetivo de encontrar su complejidad y posteriormente realizar una evaluación de los mismos.

V. Metodología de investigación

Arboles de Decisión

Medición Empírica

Operaciones	Tamaños de los arreglos							
	10	50	100	200	500	1000	5000	10000
Asignaciones	1598	25764	86446	312510	1792966	6929066	169501122	674130978
Comparaciones	258	7873	32468	135119	843136	3358757	842212222	3360071006
Cantidad de líneas ejecutadas	1806	33196	118011	445932	2631927	10279681	253682178	1010057936
Tiempo de ejecución (ms)	1	7	28	85.442	545.490	2297.13	59518.692	240090.656
Cantidad de líneas del código	67							

Factor de Crecimiento

Talla	Factor talla	Factor Asig	Factor Comp	Factor Cantidad de líneas ejecutadas	Factor Tiempo de ejecución
De 10 a 50	5	16.12	30.52	18.38	0
De 50 a 100	2	3.36	4.12	3.55	4
De 100 a 200	2	3.61	4.16	3.78	3.052
De 500 a 1000	2	3.86	3.98	3.91	4.211
De 1000 a 10000	10	97.29	100.04	98.26	104.518
De 5000 a 10000	5	24.46	25.08	24.67	25.91

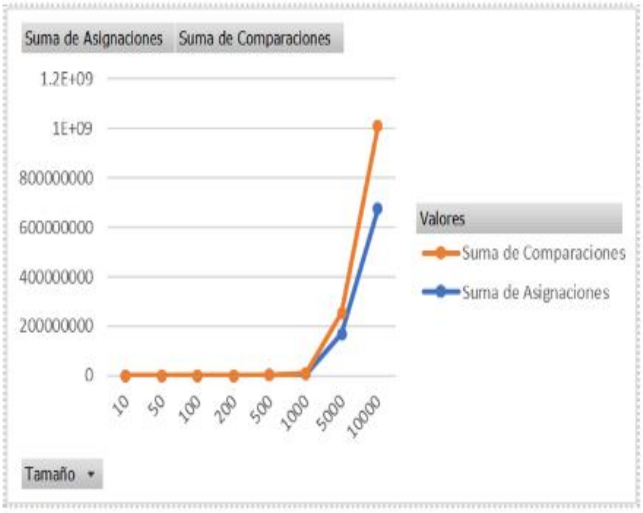
Clasificación del comportamiento de las asignaciones	$O(n^2)$
Clasificación del comportamiento de las comparaciones	$O(n^2)$

Clasificación según su entrada de los datos use la notación O Grande según corresponda	
Entrada de los datos	Aleatoria
Clasificación	$O(n^2)$

Medición analítica

Código Fuente (Sup)	Medición de líneas ejecutadas
Total (La suma de los pasos)	$15n^4 + 30n^3 + 30n^2 + 7n + 59$
Notación en O grande	$O(n^4)$

Medición Gráfica



Algoritmo Kmeans

Medición Empírica

Operaciones	Tamaños de los arreglos							
	10	50	100	200	500	1000	5000	10000
Asignaciones	145	705	1405	2805	7005	14005	70005	140005
Comparaciones	0	0	0	0	0	0	0	0
Cantidad de líneas ejecutadas	145	705	1405	2805	7005	14005	70005	140005
Tiempo de ejecución (ms)	0.00	0.08	0.1	0.3	2.48	8.5	222.9	1506.45
Cantidad de líneas del código	86							

Factor de Crecimiento

Talla	Factor talla	Factor Asig	Factor Comp	Factor Cantidad de líneas ejecutadas	Factor Tiempo de ejecución
De 10 a 50	5	705/145= 4.86	0/0=0	705/145= 4.86	0.08/0.00=0
De 50 a 100	2	1405/705=1.99	0/0=0	1405/705=1.99	0.1/0.08=1.25
De 100 a 200	2	2805/1405=1.99	0/0=0	2805/1405=1.99	0.3/0.1=3
De 500 a 1000	2	14005/7005=1.99	0/0=0	14005/7005=1.99	8.5/2.48=3.42
De 1000 a 10000	10	140005/14005=9.99	0/0=0	140005/14005=9.99	1506.45/8.5=177.22
De 5000 a 10000	5	140005/70005=1.99	0/0=0	140005/70005=1.99	1506.45/222.9=6.75

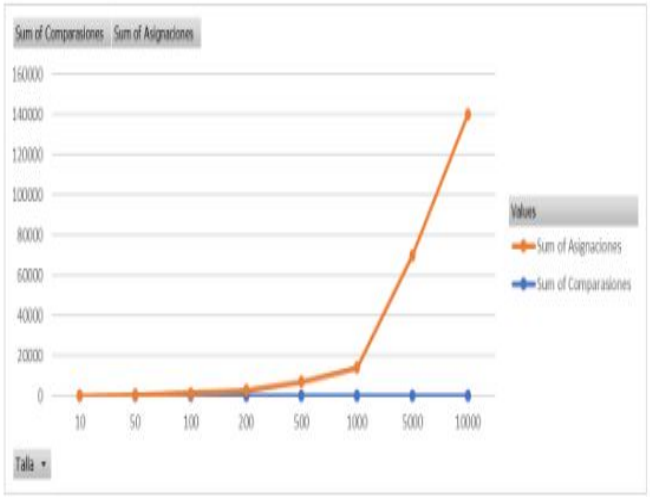
Clasificación del comportamiento de las asignaciones	$O(n^2)$
Clasificación del comportamiento de las comparaciones	NA

Clasificación según su entrada de los datos use la notación O Grande según corresponda	
Entrada de los datos	Aleatoria
Clasificación	$O(n^2)$

Medición analítica

Código Fuente (No Sup)	Medición de líneas ejecutadas
Total (La suma de los pasos)	$5+5n+3n^2$
Notación en O grande	$O(n^2)$

Medicion Grafica



VI. Evaluacion

Árbol de decisión:

La medición empírica deja interpretar, mediante el comportamiento del factor de talla, que conforme va creciendo la entrada de datos el algoritmo sigue una distribución de crecimiento cuadrática O(n2) de forma que con un factor de talla 2 en la entrada de datos las asignaciones y comparaciones poseen un valor cercano a 4 y con un factor de talla 5 poseen un valor cercano a 25. La medición analítica arroja un polinomio de grado 4 O(n4) ya que una función con 2 for anidados llama a otra que posee igualmente 2 fors anidados. Además, la medición gráfica muestra un crecimiento de la forma Xa .

Por lo que podemos concluir que la distribución es exponencial O(na), lo cual en la realidad puede no ser muy práctico ya que un algoritmo lineal O(n), o bien un algoritmo recursivo de complejidad logarítmica O(log(n)) es preferible.

K-Means:

Para el algoritmo kmeans podemos notar basándonos en el factor de talla, que el algoritmo no cuenta con asignaciones por lo cual esta no tiene una medición o se podria decir que se acopla a un analisis en O(n), se dice que se transforma en

un valor lineal ya que a lo largo del incremento en el factor de talla todos los valores fueron cero.

Por otro lado, tenemos la valoración de los datos de comparaciones, las cuales, si muestran un patrón y similitud según incrementa el factor de talla, indicando que tiene un crecimiento exponencial O(na), implicando asi que no es tan practico a diferencia de una complejidad lineal o logarítmica, las cuales son preferibles y tienen un mejor resultado.

VII. Conclusiones

Se logró analizar ambos algoritmos de ML tanto el árbol de decisión como el K-means. Se realizaron exitosamente los cambios necesarios en el código para poder examinar el desempeño del código respecto al tiempo y entrada de datos.

Existe una incongruencia entre los resultados de la medición empírica del árbol de decisión O(n2) y la medición analítica O(n4), ya que, aunque ambos presentan resultados exponenciales el exponente no concuerda. La medición analítica claramente muestra el llamado de una función con 2 for anidados dentro de una función que posee, igualmente, 2 for anidados. Por esta razón se descarta que la notación O(n4) para su caso sea incorrecta. En lo que respecta a la medición empírica, las asignaciones, comparaciones e instrucciones ejecutadas fueron detalladamente sumadas en todas las funciones donde y cuando correspondían.

Cualquier trabajo futuro puede utilizar la información acerca de los desempeños en complejidad respecto a entrada de datos y tiempo encontrados en esta investigación, esto con el objetivo de crear algoritmos análogos con un mejor rendimiento según corresponda.

Link del repositorio Git del Proyecto 1

https://github.com/JoxanF/proyecto-1-algoritmos

Bibliografia

[1] G. Rubio and M.A. Taboada, '"Árbol de decisión para diagnosticar la capacidad productiva de suelos de la región Pampeana," Ciencia del suelo, vol. 31, no. 2, Dec 1, pp. 235-243.

[2] J. Brownlee, '"How To Implement The Decision Tree Algorithm From Scratch In Python,", vol. 2022, no. 12/10/, "Nov 12, ".

[3]https:www.aprendemachinelearning.comauthoruserauthor. (2018, 12 marzo). K-Means con Python paso a paso. Aprende Machine Learning. Recuperado 14 de octubre de 2022, de https://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/

[4]MINERIA DE DATOS - Capitulo 1.pdf. (s. f.). DocDroid. Recuperado 14 de octubre de 2022, de https://www.docdroid.net/kkD37aj/mineria-de-datos-capitulo-1-pdf

[5]Elias David Niño Ruiz. (2020, 13 abril). Minería de Datos - Explicación Simple de K-Means - Implementación en MATLAB. YouTube. Recuperado 14 de octubre de 2022, de https://www.youtube.com/watch?v=ggJW4Hh9PiA

[6]Normalización en desempeño de k-means sobre datos climáticos. (s. f.). Recuperado 14 de octubre de 2022, de https://www.researchgate.net/publication/340477103$_Normalizacion_en_desempeno_de_k-means_sobre_datos_climaticos_Normalization_in_k-means_performance_on_climate_data$