Comparación del rendimiento de los algoritmos de Machine Learning: SVM, Decisión Tree, Logistic Regresión y Multilayer Perceptron

Aguirre Ibarra Jesus Armando, García Hernández Víctor David

Instituto Tecnológico de Tijuana
jesus.aguirre@tectijuana.edu.mx
victor.garcia17@tectijuana.edu.mx

Resumen - Machine Learning es un área de la inteligencia artificial que engloba un conjunto de técnicas que hacen posible el aprendizaje automático a través del entrenamiento con grandes volúmenes de datos. Hoy en día existen diferentes modelos que utilizan esta técnica y consiguen una precisión incluso superior a la de los humanos en las mismas tareas, por ejemplo, en el reconocimiento de objetos en una imagen.

La construcción de modelos de Machine Learning requiere adaptaciones propias debido a la naturaleza de los datos o a la problemática a la que se aplica. Así, surge la necesidad de investigar las diferentes técnicas que permitan obtener resultados precisos y confiables en un tiempo razonable.

Índice de Términos –SVM, decisión three, logistic regresión, multilayer preceptron

I. INTRODUCCION

La primera herramienta para atacar problemas relacionados con patrones de datos es Machine Learning, una disciplina informática con el diseño de algoritmos que permiten a las computadoras desarrollar comportamientos basados en datos empíricos. Estos algoritmos se pueden organizar en la siguiente jerarquía: aprendizaje supervisado, no supervisado y semi-supervisado. Por lo tanto, Machine Learning es más que nada un subcampo multidisciplinario que se ocupa del descubrimiento de patrones en grandes conjuntos de datos que involucran métodos de inteligencia artificial, aprendizaje automático, estadísticas y sistemas de bases de datos.

El aprendizaje automático es esa rama de la informática que otorga a la IA la capacidad de aprender tareas. Para lograrlo, los programadores se basan en los **algoritmos del machine learning**.[1]

II. MARCO TEÓRICO DE LOS ALGORITMOS

A. *Support Vector Machine (SVM)*

Support Vector Machine (SVM) es un tipo de máquina de aprendizaje basada en la teoría de aprendizaje estadístico. La idea básica de aplicar SVM a la clasificación de patrones puede presentarse brevemente de la siguiente manera. Primero, mapee los vectores de entrada en un espacio de características (posible con una dimensión más alta), ya sea lineal o no lineal, lo cual es relevante con esta selección de la función del núcleo. Luego, dentro del espacio de características del primer paso, busque una división lineal optimizada, es decir, construya un hiperplano que separe dos clases (esto puede extenderse a varias clases). El entrenamiento de SVM siempre busca una solución global optimizada y evita el ajuste excesivo, por lo que tiene la capacidad de manejar una gran cantidad de características. Una descripción completa de la teoría de SVM para el reconocimiento de patrones se encuentra en el libro de Vapnik.[2]

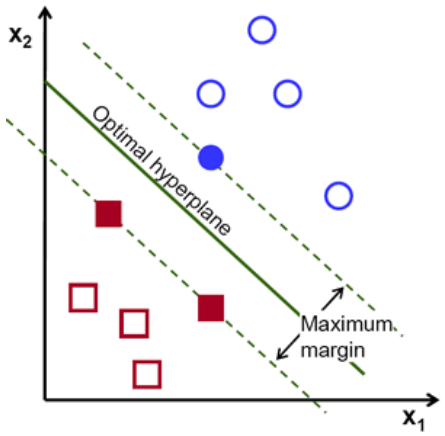

Fig. 1. Diagrama del modelo SVM. Para separar las dos clases de puntos de datos, hay muchos hiperplanos posibles que podrían elegirse. Nuestro objetivo es encontrar un plano que tenga el margen máximo, es decir, la distancia máxima entre los puntos de datos de ambas clases. Maximizar

la distancia de margen proporciona cierto refuerzo para que los puntos de datos futuros se puedan clasificar con más confianza.[3]

Las SVM se han utilizado en una amplia gama de problemas, incluido el diseño de fármacos[4], el reconocimiento de imágenes y la clasificación de textos[5], el análisis de datos de expresión de genes de microarrays[6] y el reconocimiento de pliegues de proteínas.[7]

B. Decision Tree (Arboles de decisión)

Un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento. Nos ayudan a tomar la decisión más "acertada", desde un punto de vista probabilístico, ante un abanico de posibles decisiones. Estos árboles permiten examinar los resultados y determinar visualmente cómo fluye el modelo. Los resultados visuales ayudan a buscar subgrupos específicos y relaciones que tal vez no encontraríamos con estadísticos más tradicionales.

Los árboles de decisión son una técnica estadística para la segmentación, la estratificación, la predicción, la reducción de datos y el filtrado de variables, la identificación de interacciones, la fusión de categorías y la discretización de variables continuas. [8]

Un árbol de decisión es un modelo de predicción cuyo objetivo principal es el aprendizaje inductivo a partir de observaciones y construcciones lógicas. Son muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva para la solución de un problema. Constituyen probablemente el modelo de clasificación más utilizado y popular. El conocimiento obtenido durante el proceso de aprendizaje inductivo se representa mediante un árbol. Un árbol gráficamente se representa por un conjunto de nodos, hojas y ramas. El nodo principal o raíz es el atributo a partir del cual se inicia el proceso de clasificación; los nodos internos corresponden a cada una de las preguntas acerca del atributo en particular del problema. Cada posible respuesta a los cuestionamientos se representa mediante un nodo hijo. Las ramas que salen de cada uno de estos nodos se encuentran etiquetadas con los posibles valores del atributo[9]. Los nodos finales o nodos hoja corresponden a una decisión, la cual coincide con una de las variables clase del problema a resolver (Ver Figura 2).

Este modelo se construye a partir de la descripción narrativa de un problema, ya que provee una visión gráfica de la toma de decisión, especificando las variables que son evaluadas, las acciones que deben ser tomadas y el orden en el que la toma de decisión será efectuada. Cada vez que se ejecuta este tipo de modelo, sólo un camino será seguido dependiendo del valor actual de la variable evaluada. Los valores que pueden tomar las variables para este tipo de modelos pueden ser discretos o continuos.[10]

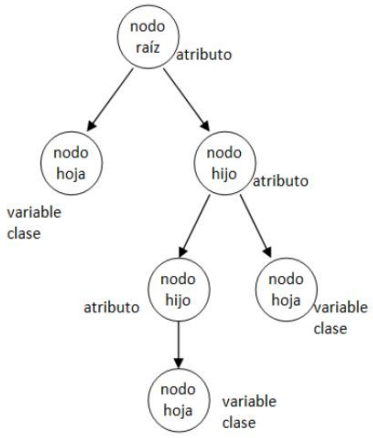

Fig. 2. Estructura de un árbol de decisión.

Un algoritmo de generación de árboles de decisión consta de 2 etapas: la primera corresponde a la inducción del árbol y la segunda a la clasificación. En la primera etapa se construye el árbol de decisión a partir del conjunto de entrenamiento; comúnmente cada nodo interno del árbol se compone de un atributo de prueba y la porción del conjunto de entrenamiento presente en el nodo es dividida de acuerdo con los valores que pueda tomar ese atributo.

La construcción del árbol inicia generando su nodo raíz, eligiendo un atributo de prueba y dividiendo el conjunto de entrenamiento en dos o más subconjuntos; para cada partición se genera un nuevo nodo y así sucesivamente. Cuando en un nodo se tienen objetos de más de una clase se genera un nodo interno; cuando contiene objetos de una clase solamente, se forma una hoja a la que se le asigna la etiqueta de la clase. En la segunda etapa del algoritmo cada objeto nuevo es clasificado por el árbol construido; después se recorre el árbol desde el nodo raíz hasta una hoja, a partir de la que se determina la membresía del objeto a alguna clase. El camino a seguir en el árbol lo determinan las decisiones tomadas en cada nodo interno, de acuerdo con el atributo de prueba presente en él.

C. Logistic Regretion (Regresión Logística)

Los métodos de regresión de variable dependiente cualitativa abarcan diferentes modelos que tratan de explicar y predecir una característica cualitativa a partir de los datos de otras variables conocidas, bien cuantitativas o cualitativas que actúan como variables explicativas.

La característica que se quiere explicar puede ser: a) una cualidad que puede únicamente tomar dos modalidades (modelos binomiales), son las más frecuentemente utilizadas, b) una cualidad que puede tomar más de dos modalidades diferentes, exhaustivas y mutuamente excluyentes (modelos multinomiales), c) una característica con varias modalidades que presentan entre ellas un orden natural (modelos ordenados) y d) la característica a explicar corresponde a una decisión que puede suponer decisiones encadenadas (modelos anidados).

Como es conocido, el concepto de regresión hace referencia a la ley experimental o fórmula matemática que traduce la relación entre variables correlacionadas. Generalmente cuando se quiere poner una variable en función de otra (o de otras), se acude al bien conocido recurso de la regresión lineal (simple o múltiple). Esta función utiliza normalmente el método de mínimos cuadrados y funciona fluidamente desde el punto de vista aritmético.

Pero cuando la variable a explicar sólo puede tomar dos valores, es decir, la ocurrencia o no de un cierto proceso, al evaluar la función para valores específicos de las variables independientes se ob-tendrá un número que será diferente de 1 y de O (los valores posibles de la variable dependiente), lo cual carece de todo sentido. En este caso, la regresión lineal debe ser descartada, en cambio la RL se ajusta adecuadamente a esta situación.

Mediante la RL se pretende es la probabilidad de que ocurra el hecho en cuestión como función de ciertas variables que se presumen relevantes o influyentes. Por lo tanto, la RL consiste en obtener una función logística de las variables independientes que permita clasificar a los individuos en una de las dos subpoblaciones o grupos establecidos por los dos va-lores de la variable dependiente.

La función logística es aquella que halla, para cada individuo según los valores de una serie de variables (Xi), la probabilidad (p) de que presente el efecto estudiado. Una transformación logarítmica de dicha ecuación, a la que se le llama logit, consiste en convertir la probabilidad (p) en odds. De aquí surge la ecuación de la regresión logística, que es parecida a la ecuación de la regresión lineal múltiple.

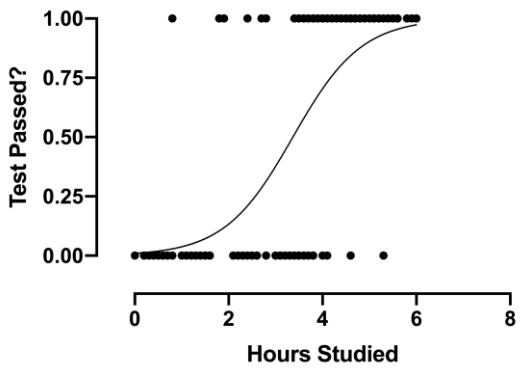

Fig. 3. Ilustración gráfica regresión logística. En este gráfico, todos nuestros puntos de datos toman el valor 0 (falla) o 1 (pase). El ajuste logístico es la curva S que modela la probabilidad de éxito en función de las horas de estudio. En este ejemplo, los instructores estarán contentos.[12]

D. Multilayer perceptrón (Perceptrón Multicapa)

El perceptrón es muy útil para clasificar conjuntos de datos que son linealmente separables. Encuentran serias limitaciones con los conjuntos de datos que no se ajustan a este patrón como se descubrió con el problema XOR. El problema XOR muestra que para cualquier clasificación de cuatro puntos existe un conjunto que no es linealmente separable.

Fig. 4. Representación gráfica de perceptrón multicapa.

El perceptrón multicapa (MLP) rompe esta restricción y clasifica los conjuntos de datos que no son linealmente separables. Lo hacen mediante el uso de una arquitectura más robusta y compleja para aprender modelos de regresión y clasificación para conjuntos de datos difíciles.[13]

III. Implementación

Herramientas

Implementamos Apache Spark que es un sistema de computación distribuido de código abierto basado en Hadoop, pensado para el análisis y procesamiento de datos en los campos del Big Data y el Machine Learning.

Por qué Apache Spark

Utilizamos Apache Spark, debido a su:

-***Velocidad:*** su diseño se ha enfocado en optimizar el rendimiento en el procesamiento de datos a gran escala, aprovechando conceptos como el procesamiento en

memoria y otras optimizaciones. Es una 100 veces más rápido que Hadoop y además consiguió el récord mundial de clasificación de datos a gran escala almacenados en disco.

-***Facilidad de uso****:* dispone de APIs sencillas de utilizar para trabajar con grandes conjuntos de datos. Tiene más de 100 operadores para transformarlos y manipular datos semiestructurados.

-***Motor unificado:*** viene empaquetado con bibliotecas de nivel superior, que incluyen soporte para consultas SQL, transmisión de datos, aprendizaje automático y procesamiento de gráficos. Estas bibliotecas estándar aumentan la productividad del desarrollador y se pueden combinar sin problemas para crear flujos de trabajo complejos.[14]

IV. RESULTADOS

Tabuladores de predicciones

Tabla 1. Precisión de regresión lineal.

LINEAR REGRESSION		
Prediction = 0.0 (20 ROWS)		
label	features	prediction
0	[18.0,1.0,156.0]	0,0
1	[18.0,1.0,3.0]	0,0
1	[18.0,1.0,108.0]	0,0
1	[18.0,1.0,108.0]	0,0
0	[19.0,1.0,103.0]	0,0
0	[19.0,1.0,103.0]	0,0
1	[19.0,1.0,134.0]	0,0
0	[19.0,1.0,1247.0]	0,0
0	[19.0,1.0,88.0]	0,0
0	[19.0,1.0,626.0]	0,0
0	[19.0,1.0,60.0]	0,0
1	[19.0,1.0,108.0]	0,0
1	[20.0,1.0,423.0]	0,0
0	[20.0,2.0,-172.0]	0,0
0	[20.0,1.0,134.0]	0,0
1	[20.0,1.0,153.0]	0,0
0	[20.0,1.0,210.0]	0,0
1	[20.0,1.0,215.0]	0,0
0	[20.0,1.0,291.0]	0,0
0	[20.0,1.0,602.0]	0,0
Prediction = 1.0 (ALL)		
label	features	prediction
0	[51.0,1.0,102127.0]	1,0
0	[52.0,2.0,66653.0]	1,0
0	[56.0,2.0,58932.0]	1,0
0	[60.0,2.0,71188.0]	1,0
Linear Regression Accuracy -> 0.8802791306244094		

Podemos observar aquí que nos dio como precisión 0.8802, eso significa que, para este modelo, la regresión lineal es efectiva en un 88% de acuerdo a este conjunto de datos establecido en el ejercicio.

Tabla 2. Precisión de perceptrón multicapa.

MULTILAYER PERCEPTRON		
Prediction = 0.0 (20 ROWS)		
label	features	prediction
1	[18.0,1.0,608.0]	0,0
1	[18.0,1.0,608.0]	0,0
0	[18.0,1.0,1944.0]	0,0
0	[19.0,1.0,56.0]	0,0
1	[19.0,1.0,302.0]	0,0
1	[19.0,1.0,329.0]	0,0
0	[19.0,1.0,424.0]	0,0
0	[19.0,1.0,527.0]	0,0
0	[19.0,1.0,0.0]	0,0
1	[19.0,1.0,108.0]	0,0
1	[19.0,1.0,779.0]	0,0
0	[19.0,1.0,1169.0]	0,0
0	[20.0,1.0,66.0]	0,0
0	[20.0,2.0,-172.0]	0,0
0	[20.0,1.0,67.0]	0,0
0	[20.0,1.0,130.0]	0,0
0	[20.0,1.0,602.0]	0,0
1	[20.0,1.0,1819.0]	0,0
0	[20.0,1.0,79.0]	0,0
0	[20.0,1.0,2764.0]	0,0
Prediction = 1.0 (ALL)		
label	features	prediction
Multilayer Perceptron Accuracy -> 0.882741899933877		

Podemos observar aquí que nos dio como precisión 0.8827, eso significa que, para este modelo, perceptrón multicapa es efectiva en un 88.27% de acuerdo a este conjunto de datos establecido en el ejercicio.

Tabla 3. Precisión del árbol de decisión.

DECISION TREE		
Prediction = 0.0 (20 ROWS)		
label	**features**	**prediction**
1	[18.0,1.0,608.0]	0,0
1	[18.0,1.0,608.0]	0,0
0	[18.0,1.0,1944.0]	0,0
0	[19.0,1.0,56.0]	0,0
1	[19.0,1.0,302.0]	0,0
1	[19.0,1.0,329.0]	0,0
0	[19.0,1.0,424.0]	0,0
0	[19.0,1.0,527.0]	0,0
0	[19.0,1.0,0.0]	0,0
1	[19.0,1.0,108.0]	0,0
1	[19.0,1.0,779.0]	0,0
0	[19.0,1.0,1169.0]	0,0
0	[20.0,1.0,66.0]	0,0
0	[20.0,2.0,-172.0]	0,0
0	[20.0,1.0,67.0]	0,0
0	[20.0,1.0,130.0]	0,0
0	[20.0,1.0,602.0]	0,0
1	[20.0,1.0,1819.0]	0,0
0	[20.0,1.0,79.0]	0,0
0	[20.0,1.0,2764.0]	0,0
Prediction = 1.0 (20 ROWS)		
label	**features**	**prediction**
0	[61.0,3.0,369.0]	1,0
1	[61.0,2.0,997.0]	1,0
1	[61.0,2.0,1058.0]	1,0
0	[61.0,3.0,410.0]	1,0
0	[61.0,2.0,50.0]	1,0
0	[62.0,2.0,13.0]	1,0
1	[62.0,3.0,0.0]	1,0
1	[62.0,3.0,6.0]	1,0
1	[62.0,2.0,973.0]	1,0
0	[63.0,2.0,937.0]	1,0
0	[63.0,3.0,859.0]	1,0
1	[64.0,2.0,48.0]	1,0
0	[64.0,3.0,890.0]	1,0
1	[65.0,3.0,90.0]	1,0
0	[66.0,3.0,204.0]	1,0
0	[67.0,2.0,65.0]	1,0
1	[67.0,3.0,443.0]	1,0
1	[68.0,3.0,250.0]	1,0
1	[68.0,3.0,695.0]	1,0
0	[69.0,3.0,344.0]	1,0
Decision Tree Accuracy -> 0.8817867900962457		

Podemos observar aquí que nos dio como precisión 0.8817 eso significa que, para este modelo, arboles de decisión es efectiva en un 88.17% de acuerdo a este conjunto de datos establecido en el ejercicio.

Tabla 4. Precisión de resultados de SVM.

SVM		
Prediction = 0.0 (20 ROWS)		
label	**features**	**prediction**
0	[58.0,2.0,2143.0]	0,0
0	[44.0,1.0,29.0]	0,0
0	[33.0,2.0,2.0]	0,0
0	[47.0,2.0,1506.0]	0,0
0	[33.0,1.0,1.0]	0,0
0	[35.0,2.0,231.0]	0,0
0	[28.0,1.0,447.0]	0,0
0	[42.0,3.0,2.0]	0,0
0	[58.0,2.0,121.0]	0,0
0	[43.0,1.0,593.0]	0,0
0	[41.0,3.0,270.0]	0,0
0	[29.0,1.0,390.0]	0,0
0	[53.0,2.0,6.0]	0,0
0	[58.0,2.0,71.0]	0,0
0	[57.0,2.0,162.0]	0,0
0	[51.0,2.0,229.0]	0,0
0	[45.0,1.0,13.0]	0,0
0	[57.0,2.0,52.0]	0,0
0	[60.0,2.0,60.0]	0,0
0	[33.0,2.0,0.0]	0,0
Prediction = 1.0 (ALL)		
label	**features**	**prediction**
SVM Results Accuracy -> 0.8830151954170445		

Podemos observar aquí que nos dio como precisión 0.8830, eso significa que, para este modelo, support vector machine es efectiva en un 88.30% de acuerdo a este conjunto de datos establecido en el ejercicio.

V. Conclusiones

Es importante que se analice la naturaleza de los datos a procesar de manera que se utilicen las características clave que son ideales para obtener un modelo de mayor precisión, ya que tal como sucede en algoritmos como arboles de decisión, existen características que aportan una mayor ganancia de información que otras,

En el caso particular del presente trabajo los diferentes métodos de clasificación utilizados arrojaron valores de precisión bastante similares. Aunque cabe mencionar que en realidad se notó que de los algoritmos Support Vector Machine y Multilayer Perceptron surgieron modelos que predecían siempre un mismo valor. Mientras que el resto de ellos predecían los diferentes valores dentro del dominio de opciones de la clase objetivo.

RECONOCIMIENTO

Un gran reconcomiendo al profesor Jose Christian Romero Hernández por impartirnos la materia de Datos Masivos, gracias por dejar que aprendiéramos de usted como maestro y gracias por dejar que usted aprendiera de nosotros, es increíble su paciencia y dedicación para enseñarnos a ser grandes personas.

REFERENCES

[1] Amparo Albalate and Wolfgang Minker. Semi-Supervised and Unsupervised Machine Learning: Novel Strate-gies. John Wiley & Sons, 2013.

[2] Vapnik VN: Statistical Learning TheoryWiley-Interscience, NewYork, 199825.

[3] Rohith Gandhi. (2018). Support Vector Machine — Introduction to Machine Learning Algorithms. 14/06/2020, de towards datas cience Sitio web: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca4

[4] Robert B, Matthew T, Sean H, Bernard B: Drug Design by MachineLearning: Support Vector Machine for Pharmaceutical DataAnalysisProceedings of the AISB'00 Symposium on Artificial Intelligencein Bioinformatics. 20001-426.

[5] Joachims T: Text Categorization with Support Vector Ma-chines: Learning with Many Relevant Features"Proceedings ofthe European Conference on Machine Learning, Springer, 199827.

[6] Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet C, Ares JM,Haussler D: Knowledge-based Analysis of Microarray GeneExpression Data by using Support Vector MachinesProc. Natl.Acad. Sci. 2000, 97:262-26728.

[7] Ding CHQ, Dubchak I: Multi-class Protein Fold Recognition Us-ing Support Vector Machines and Neural NetworksBioinfor-matics 2001, 4(17):349-358

[8] Berlanga, V., Rubio Hurtado, M. J., & Vilà Baños, R. (2013). Cómo aplicar árboles de decisión en SPSS. REIRE. Revista d'Innovació i Recerca en Educació, 2013, vol. 6, num. 1, p. 65-79.

[9] Russell, S. and P. Norvig, Artificial Intelligence: A Modern Approach. Second ed. Upper Saddle River (N J): Prentice Hall/ Pearson Education; 2003.3.

[10] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees, Wadsworth (New York); 1994.

[11] Fiuza Pérez, M., & Rodríguez Pérez, J. C. (2000). La regresión logística: una herramienta versátil. Nefrología, 20(6), 495-500.

[12] GraphPad. (2019). How simple logistic regression differs from simple linear regression. 14/06/2020, de GraphPad Sitio web: https://www.graphpad.com/guides/prism/8/curve-fitting/reg_simple_logistic_and_linear_difference.htm

[13] Jeremy Bernstein. (2018). Multilayer Perceptron. 14/06/2020, de DeepAI Sitio web: https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron

[14] UNIR Revista. (2020). Apache Spark en big data: qué es y para que se emplea. 14/06/2020, de UNIR Revista Sitio web: https://www.unir.net/ingenieria/revista/noticias/apache-spark-big-data/549204985652/