


INSTITUTO TECNOLOGICO DE TIJUANA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA EN SISTEMAS COMPUTACIONALES

PROYECTO FINAL

MATERIA:
DATOS MASIVOS

ALUMNOS:
IBARRA REYES CRUZ MANUEL 16210973
cruz.ibarra@tectijuana.edu.mx
JIMÉNEZ DÍAZ DE SANDI RENÉ 15211900
rene.jimenez17@tectijuana.edu.mx

PROFESOR:
M.C. JOSE CHRISTIAN ROMERO HERNANDEZ

TIJUANA B.C. 11 DE ENERO DEL 2021

Índice

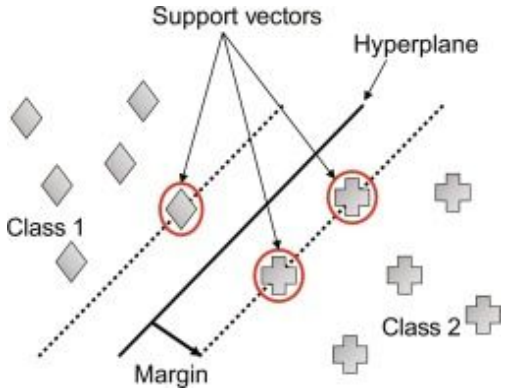
Introducción

La evolución del machine learning es gracias al desarrollo de nuevas tecnologías y algoritmos de programación que facilitan y optimizan cada uno de los procesos. Estos algoritmos tienen funciones específicas o algunas veces en común, en los que cada algoritmo tiene un rendimiento diferente de acuerdo a la manera en que este está lógicamente programado. Existe una gran cantidad de algoritmos de los que en este documento hablaremos y probaremos de manera teórica y práctica para así poder realizar una comparación de acuerdo a los resultados obtenidos. Hablaremos y nos enfocaremos en los algoritmos Decision tree, Logistic Regression, multiplayer perception,SVM. El Machine Learning puede ser una herramienta increíblemente beneficiosa para descubrir información y predecir tendencias futuras

Marco teórico

SVM

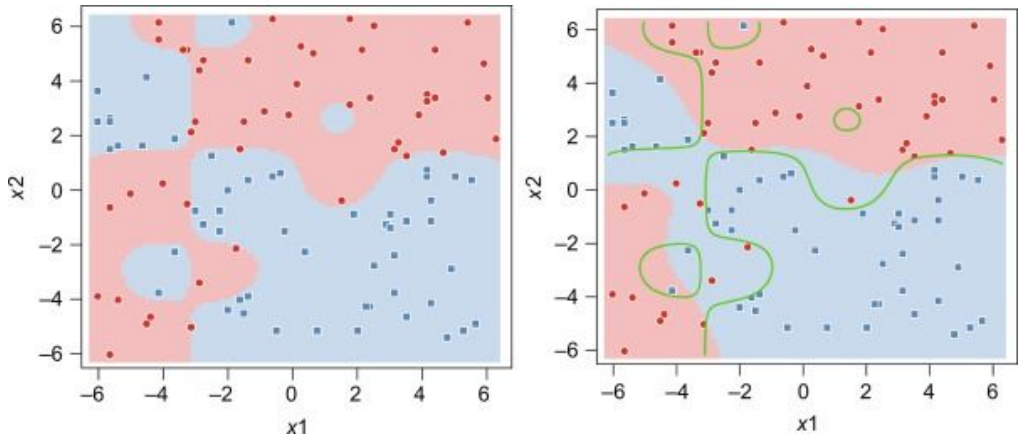
Las máquinas de vectores de apoyo (SVM) son poderosas herramientas de aprendizaje de máquinas para la clasificación y predicción de datos (Vapnik, 1995). El problema de la separación de dos clases se resuelve utilizando un hiperplano que maximiza el margen entre las clases.



Los puntos de datos que se encuentran en los márgenes se llaman vectores de apoyo. El algoritmo SVM busca encontrar el hiperplano que crea el mayor margen entre los puntos de formación de las dos clases. También penaliza la distancia total de los puntos que se encuentran en el lado equivocado de su margen siempre que haya superposición entre las dos clases de datos. Esto permite que se tolere un número limitado de clasificaciones erróneas cerca del margen. La otra característica clave en el SVM es el uso de las funciones del núcleo y el parámetro de penalización para convertir los límites no lineales en el espacio de los parámetros de las entradas en límites lineales en algún espacio transformado de mayor dimensión. En la siguiente figura se ilustra la representación de un problema de dos clases en un espacio bidimensional utilizando SVM.

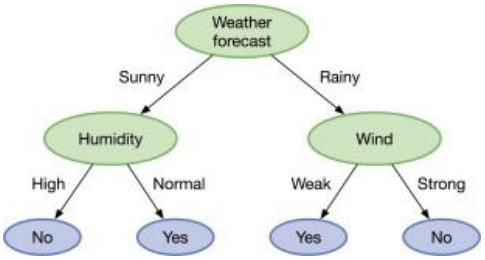
Aquí, la demarcación de los límites entre las clases roja y azul (panel izquierdo) muestra un espacio predominantemente continuo para la clase roja con bolsas azules incrustadas.

El modelo SVM encajado (panel derecho) también crea un patrón diagonalmente dominante, aunque uno donde la clase azul es continua. La fracción relativa del espacio azul frente al rojo es muy similar en ambos casos.



Árbol de decisión

El árbol de decisiones es una técnica de aprendizaje de máquina supervisada para inducir un árbol de decisiones a partir de los datos del entrenamiento. Un árbol de decisión (también denominado árbol de clasificación o árbol de reducción) es un modelo predictivo que es un mapeo desde las observaciones sobre un elemento hasta las conclusiones sobre su valor objetivo. En las estructuras de los árboles, las hojas representan clasificaciones (también denominadas etiquetas), los nodos sin hojas son características y las ramas representan conjunciones de características que conducen a las clasificaciones.

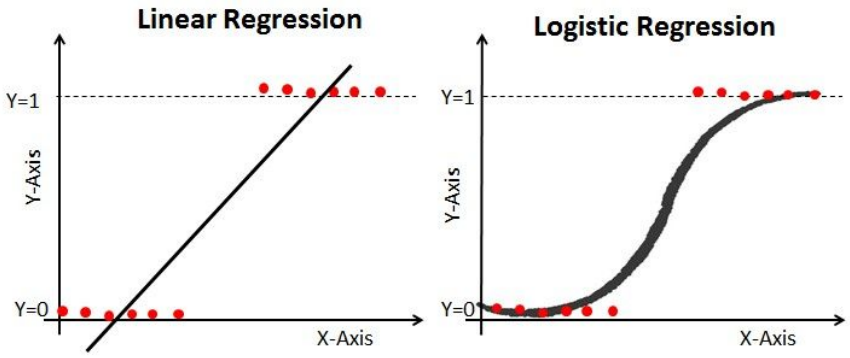


La construcción de un árbol de decisión que sea coherente con un conjunto de datos determinado es fácil. El reto consiste en construir buenos árboles de decisión, lo que normalmente significa los árboles de decisión más pequeños [2].

Regresión logística

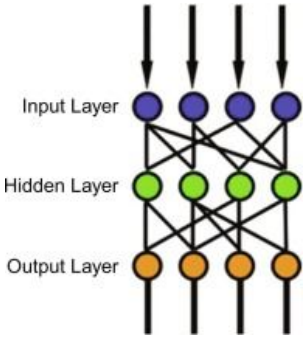
El análisis de regresión logística se utiliza para examinar la asociación de variables independientes con una variable dependiente dicotómica. Esto contrasta con el análisis de regresión lineal en el que la variable dependiente es una variable continua. Consideren un ejemplo en el que la regresión logística podría utilizarse para examinar la pregunta de investigación: "¿Están los antecedentes de intentos de suicidio asociados con el riesgo de un intento posterior (es decir, observado prospectivamente)? El modelo de regresión logística compara las probabilidades de un intento prospectivo en aquellos con y sin intentos anteriores. La proporción de esas probabilidades se denomina proporción de probabilidades. Una regresión logística no analiza las probabilidades, sino una transformación logarítmica natural de las probabilidades, las probabilidades logísticas. Aunque los cálculos son más complicados cuando hay múltiples variables independientes, se pueden utilizar programas informáticos para realizar los análisis. Sin embargo, debido a la transformación logarítmica de la proporción de probabilidades, la interpretación de los resultados de la salida de la computadora no es necesariamente sencilla. La interpretación requiere una transformación de vuelta a la escala original tomando el inverso del logaritmo natural del coeficiente de regresión, lo que se llama exponenciación. El coeficiente de regresión exponencial representa la fuerza de la asociación de la variable independiente con el resultado. Más específicamente, representa el aumento (o disminución) del riesgo del resultado que se asocia con la variable independiente. El coeficiente de regresión exponencial representa la diferencia de riesgo del resultado (por ejemplo, intento de suicidio) para dos sujetos que difieren en un punto de la variable independiente. En este caso, es la diferencia entre los que tienen y los que no tienen historial de intentos (es decir, cuando el historial de intentos está codificado: 0 = no y 1 = sí). El modelo de regresión logística puede ampliarse para incluir varias variables independientes (es decir, factores de riesgo hipotéticos). Por ejemplo, ¿son los antecedentes de intentos, la gravedad de la depresión y la situación laboral factores de riesgo para la conducta suicida, el control del diagnóstico, la edad

y el sexo? Cada odds ratio de un modelo de este tipo representa el cambio en el riesgo del resultado (es decir, un intento de suicidio) que se asocia con la variable independiente, controlando por las otras variables independientes [3].



Multilayer Perceptron

El perceptrón multicapa (MLP) es un complemento de la red neuronal de avance. Consiste en tres tipos de capas: la capa de entrada, la capa de salida y la capa oculta. La capa de entrada recibe la señal de entrada para ser procesada. La tarea requerida, como la predicción y la clasificación, la realiza la capa de salida. Un número arbitrario de capas ocultas que se colocan entre la capa de entrada y la de salida son el verdadero motor computacional del MLP. De manera similar a una red de alimentación hacia adelante en un MLP, los datos fluyen en la dirección hacia adelante desde la capa de entrada a la de salida. Las neuronas del MLP se entrenan con el algoritmo de aprendizaje de retropropagación. [4]



Implementación

Para el desarrollo de estas pruebas hemos utilizado el entorno de desarrollo de Visual studio code como editor de código, para poder ejecutar cada una de las líneas de código en un “compilador” hemos utilizado spark-shell que nos permite la ejecución de código de manera en la que pueda ser incluso compilado en diferentes computadores. Para poder ejecutar de manera más eficiente todos los procesos utilizamos linux como sistema operativo, a pesar de que se puede usar cualquier otro sistema operativo, linux nos permite de manera más eficiente ejecutar cada una de las pruebas de los algoritmos. El lenguaje de programación utilizado es Scala, que nos permite de una manera más eficiente internamente la ejecución de los algoritmos, ya que cuenta con una gran cantidad de librerías que permiten correr cada uno de los algoritmos así como poder personalizar la manera en la que se nos es necesario compilar. A pesar de que se tienen estas librerías, también es posible modificar como funcionan y modificar su compatibilidad y comportamiento de acuerdo a la problemática que deseamos resolver.

Resultados

Tiempo de ejecución en segundos

Iteración	Decision Tree	Logistic Regression	Multilayer Perceptron	SVM
1	12	15	17	15
2	11	13	12	13
3	11	14	13	14
4	13	11	12	12
5	13	11	12	13
6	12	12	13	13
7	15	12	13	13
8	11	12	15	12
9	11	11	13	14
10	10	12	14	12
Promedio	**11.9**	**12.3**	**13.4**	**13.1**

Al analizar los resultados de los tiempos de ejecución de cada algoritmo podemos notar que la primera iteración de cada clasificador es usualmente la que más tiempo requiere y que no se observan “picos” de tiempo en cada una de las siguientes iteraciones.

Precisión en porcentaje

Iteración	Decision Tree	Logistic Regression	Multilayer Perceptron	SVM
1	89	89	89	88
2	89	89	89	88
3	89	88	89	88
4	89	89	89	88
5	89	89	89	88
6	89	89	89	88
7	89	89	89	88
8	89	89	89	88
9	89	89	89	88
10	89	89	89	88
Promedio	**89**	**88.9**	**89**	**88**

Al analizar los resultados de la precisión de cada algoritmo podemos notar que cada clasificador mantiene una precisión constante durante cada iteración, variando en ± 1%.

Como resultado final podemos observar que el árbol de decisión (decision tree) es uno de los mejores clasificadores ya que obtiene un 11.9 segundos y 89%.

Tabla comparativa de resultados

Iteración	Decision Tree	Logistic Regression	Multilayer Perceptron	SVM
Tiempo de ejecución	11.9	12.3	13.4	13.1
Precisión	89	88.9	89	88

Conclusiones

Existe una gran cantidad de algoritmos en la computación, en especial en el machine learning, pero la posibilidad de probar diferentes algoritmos nos permite darnos cuenta que son eficientes todos de maneras diferentes, en las que como programadores podemos darnos cuenta que cada algoritmo es tan potente como es necesario para la solución de un problema, y por ende debemos entender que están desarrollados para poder resolver una gran cantidad de problemas, por ello cada uno es potencialmente eficiente de acuerdo a lo que deseamos solucionar.

Referencias

[1] Support Vector Machine - an overview | ScienceDirect Topics

[2] Decision Trees - an overview | ScienceDirect Topics

[3] Logistic Regression Analysis - an overview | ScienceDirect Topics

[4] Multilayer Perceptron - an overview

[5] Decision Tree

[6] ¿Cuáles son los tipos de algoritmos del machine learning?