TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:
Febrero - Julio 2021

CARRERA:
Ing. en sistemas computacionales

MATERIA:

Datos masivos

TÍTULO:
Proyecto Final

UNIDAD A EVALUAR:
Unidad 4

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Jacuinde Solis Ricardo Javier 16212025
Jimenez Ramirez Julio Fabian 17212147

NOMBRE DEL DOCENTE :

Jose Christian Romero Hernandez

Índice

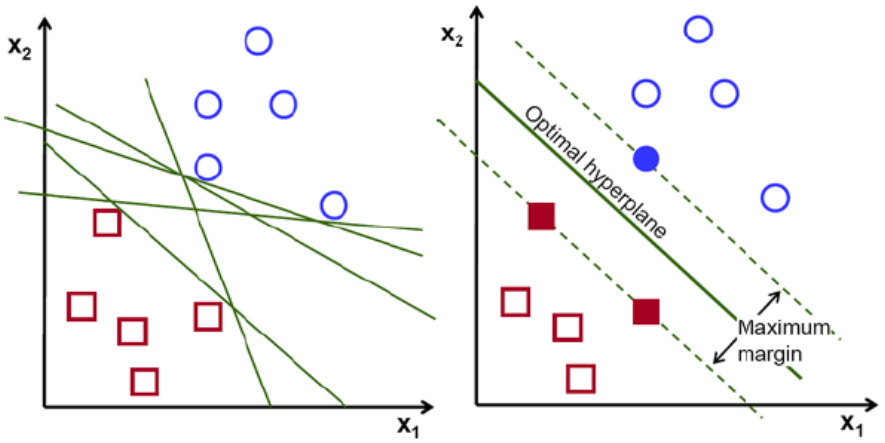
Introducción

Este documento tiene la finalidad de dar una breve explicación acerca de diferentes algoritmos de machine learning, SVM (Support Vector Machine), decision tree, logistic regression y multilayer perceptron. A su vez, su objetivo es comparar la eficiencia y el rendimiento de estos cuatro algoritmos al probarlos con un conjuntos de datos.

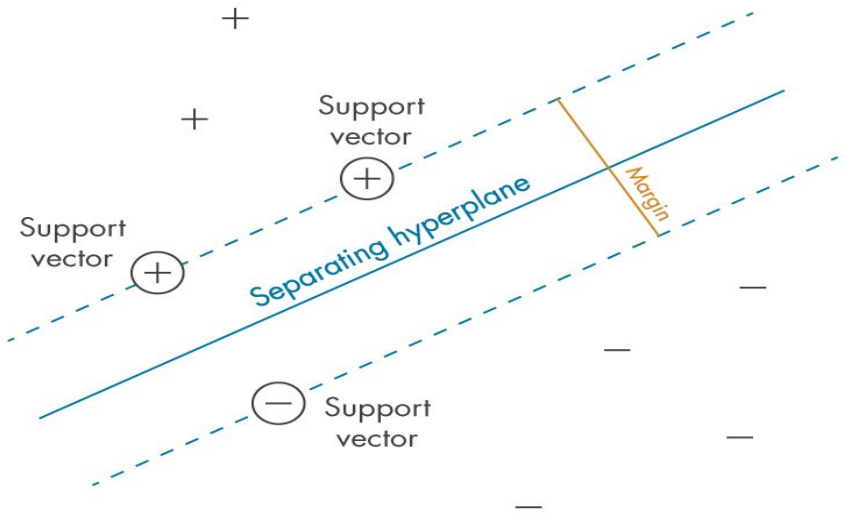
SupportVectorMachine

Una máquina de vectores de soporte (SVM) es un algoritmo de aprendizaje supervisado que se utiliza para muchos problemas de clasificación y regresión, incluidas aplicaciones médicas de procesamiento de señales, procesamiento de lenguaje natural y reconocimiento de voz e imágenes.

El objetivo del algoritmo SVM es encontrar un hiperplano que en la mejor medida posible separe los puntos de datos de una clase de los de la otra, para separar dos clases de datos hay muchos hiperplanos posibles. El objetivo del algoritmo es encontrar el que esté a mayor distancia entre los datos de ambas clases.



Para obtener el plano más alejado de los dos conjuntos de clases se tiene que calcular la distancia perpendicular de cada observación a un determinado hiperplano. La menor de estas distancias (conocida como margen) determina como de alejado está el hiperplano de las observaciones de entrenamiento.



Los puntos que se encuentran el las fronteras de los margenes son conocidos como vectores de soporte, SVM busca maximizar el margen entre los puntos y el hiperplano.

Tipos

La función matemática utilizada para la transformación se conoce como función kernel. Por lo regular admiten:

- Lineal
- Polinómico
- Función de base radial (RBF)
- Sigmoide

Ventajas

Los clasificadores de Máquinas de Vectores de Soporte ofrecen una buena precisión y realizan predicciones más rápidas en comparación con el algoritmo de Naive Bayes. También utilizan menos memoria porque utilizan un subconjunto de puntos de entrenamiento en la fase de decisión. Este algoritmo funciona bien con un claro margen de separación y con un espacio dimensional elevado.

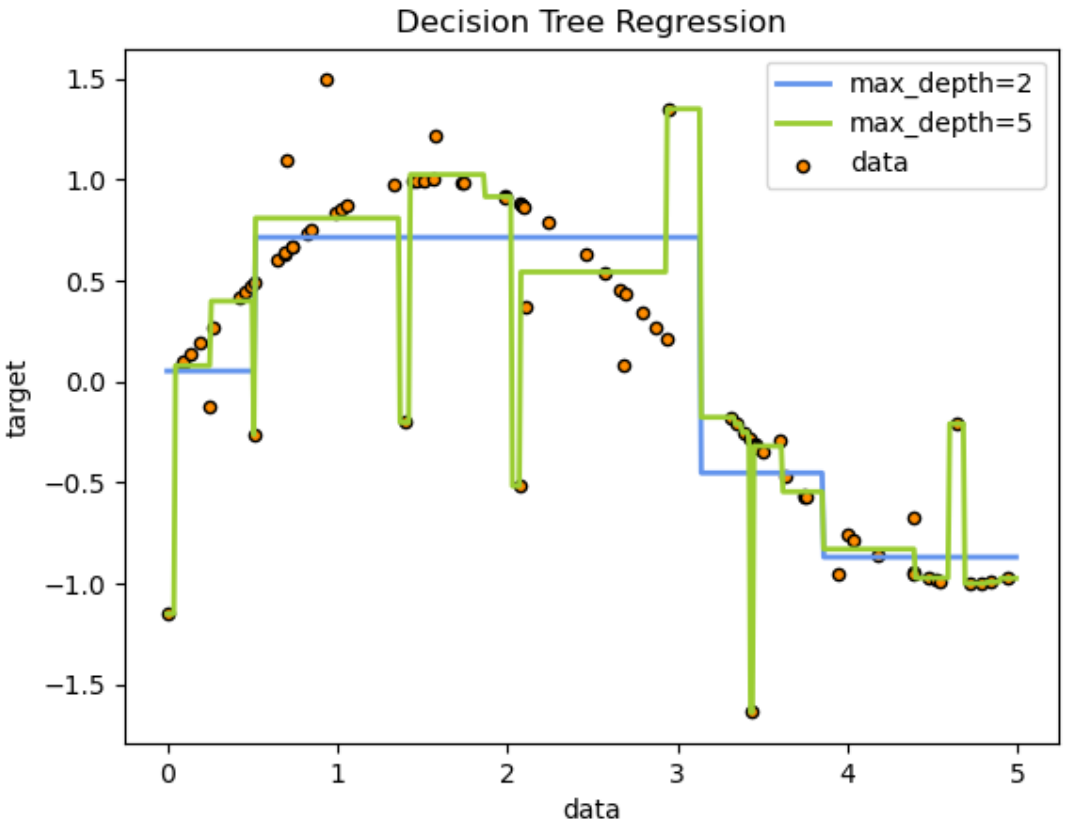
Desventajas

Las Máquinas de Vectores de Soporte no son adecuadas para grandes conjuntos de datos debido a su alto tiempo de formación y también requiere más tiempo de formación en comparación con Naive Bayes. Funciona mal con clases superpuestas y también es sensible al tipo de núcleo utilizado.

Decision Tree

Los árboles de decisión son un método de aprendizaje supervisado no paramétrico que se utiliza para la clasificación y regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas de decisión simples inferidas de las características de los datos. Un árbol puede verse como una aproximación constante a trozos.

en el ejemplo siguiente, los árboles de decisión aprenden de los datos para aproximar una curva sinusoidal con un conjunto de reglas de decisión si-entonces-si no. Cuanto más profundo es el árbol, más complejas son las reglas de decisión y más ajustado es el modelo.



Los tipos de árboles de decisión se basan en el tipo de variable objetivo que tenemos. Puede ser de dos tipos:

1. **Árbol de decisión de variable categórica: árbol de** decisión que tiene una variable objetivo categórica y luego se llama **árbol de decisión de variable categórica.**
2. **Árbol de decisión de variable continua: El árbol de** decisión tiene una variable de destino continua, por lo que se denomina **Árbol de decisión de variable continua.**

Algunas ventajas de los árboles de decisión son:

- Sencillo de entender e interpretar. Los árboles se pueden visualizar.
- Requiere poca preparación de datos. Otras técnicas a menudo requieren la normalización de datos, es necesario crear variables ficticias y eliminar valores en blanco. Sin embargo, tenga en cuenta que este módulo no admite valores perdidos.
- El costo de usar el árbol (es decir, predecir datos) es logarítmico en el número de puntos de datos usados para entrenar el árbol.

Las desventajas de los árboles de decisión incluyen:

- Los aprendices de árboles de decisión pueden crear árboles demasiado complejos que no generalizan bien los datos. A esto se le llama sobreajuste. Mecanismos como la poda, el establecimiento del número mínimo de muestras necesarias en un nodo de la hoja o el establecimiento de la profundidad máxima del árbol son necesarios para evitar este problema.
- Los árboles de decisión pueden ser inestables porque pequeñas variaciones en los datos pueden resultar en la generación de un árbol completamente diferente. Este problema se mitiga mediante el uso de árboles de decisión dentro de un conjunto.
- Las predicciones de los árboles de decisión no son uniformes ni continuos, sino aproximaciones constantes por partes, como se ve en la figura anterior. Por lo tanto, no son buenos para la extrapolación.

Logistic Regression

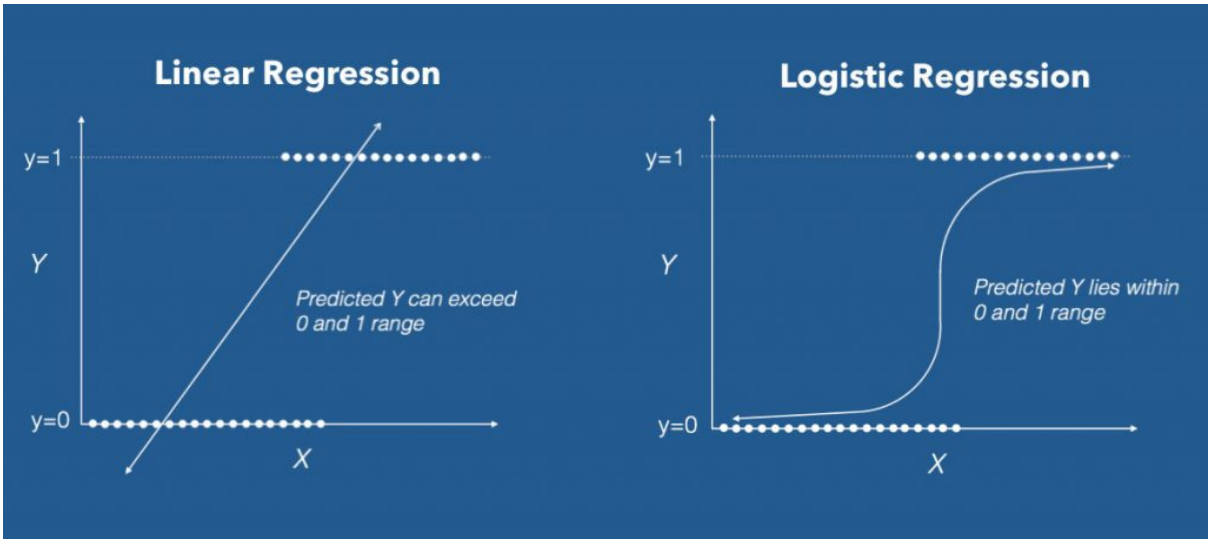
La regresión logística es un algoritmo de modelado predictivo que se utiliza cuando la variable Y es categórica binaria. Es decir, solo puede tomar dos valores como 1 o 0. El objetivo es determinar una ecuación matemática que se puede usar para predecir la probabilidad del evento 1. Una vez que se establece la ecuación, se puede usar para predecir la Y cuando solo las X son conocidas.

En la regresión lineal, la variable Y es siempre una variable continua. Si suponga que la variable Y es categórica, no puede usarla como modelo de regresión lineal. Cuando Y es una variable categórica con 2 clases, la regresión logística se puede utilizar para modelar y resolver estos problemas, también llamados problemas de clasificación binaria.

Un punto clave a tener en cuenta aquí es que Y solo puede tener 2 clases y no más que eso. Si Y tiene más de 2 clases, se convertiría en una clasificación de clases múltiples y ya no puede usar la regresión logística de vainilla para eso.

Sin embargo, la regresión logística es una técnica clásica de modelado predictivo y sigue siendo una opción popular para modelar variables categóricas binarias.

Otra ventaja de la regresión logística es que calcula una puntuación de probabilidad de predicción de un evento. Más sobre eso cuando empieces a construir los modelos



Comparando con la regresión lineal, cuando la variable de respuesta tiene solo 2 valores posibles, es deseable tener un modelo que prediga el valor como 0 o 1 o como una puntuación de probabilidad que oscile entre 0 y 1.

La regresión lineal no tiene esta capacidad. Porque, si utiliza la regresión lineal para modelar una variable de respuesta binaria, es posible que el modelo resultante no restrinja los valores de Y predichos entre 0 y 1

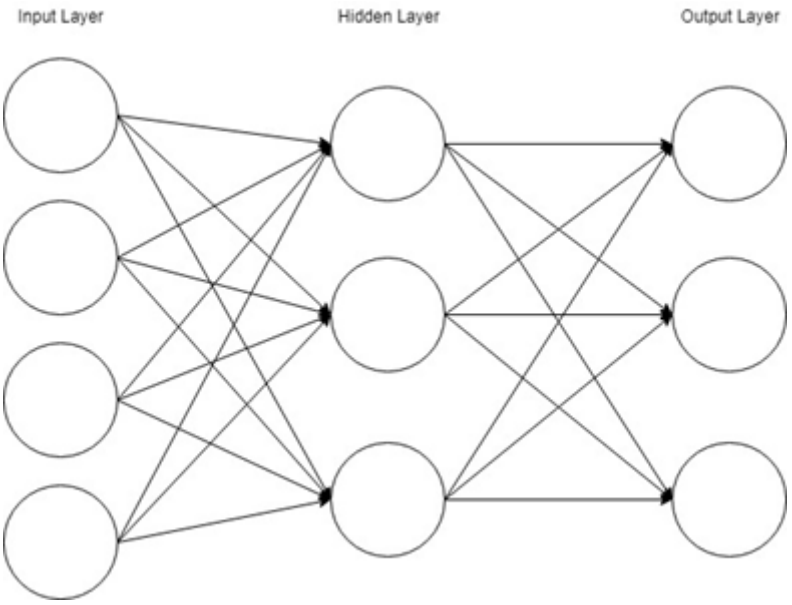
Multilayer perceptron

Un perceptrón multicapa (MLP) es una clase de red neuronal artificial de retroalimentación

El perceptrón es muy útil para clasificar conjuntos de datos que son linealmente separables. Se encuentran con serias limitaciones con 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 separable linealmente.

MultiLayer Perceptron (MLP) rompe esta restricción y clasifica conjuntos de datos que no son separables linealmente. Lo hacen utilizando una arquitectura más robusta y compleja para aprender modelos de regresión y clasificación para conjuntos de datos difíciles.

El Perceptrón consta de una capa de entrada y una capa de salida que están completamente conectadas. Los MLP tienen las mismas capas de entrada y salida, pero pueden tener múltiples capas ocultas entre las capas mencionadas anteriormente, como se ve a continuación.



- La **capa de entrada** consta de neuronas que aceptan los valores de entrada. La salida de estas neuronas es la misma que la de los predictores de entrada. Los nodos de la capa de entrada representan los datos de entrada. Todos los demás nodos asignan entradas a salidas mediante una combinación lineal de las entradas con los pesos de los nodos w y el sesgo aplicando una función de activación. Esto se puede escribir en forma de matriz para MLPC con capas K + 1 de la siguiente manera: Input_Layer
- Las **capas ocultas** se encuentran entre las capas de entrada y salida. Normalmente, el número de capas ocultas varía de una a muchas. Es la capa de cálculo central que tiene las funciones que mapean la entrada a la salida de un nodo. Los nodos de las capas intermedias utilizan la función sigmoidea (logística), como sigue Hidden_Layer
- La **capa de salida** es la capa final de una red neuronal que devuelve el resultado al entorno del usuario. Basado en el diseño de una red neuronal, también indica a las capas anteriores cómo se han desempeñado en el aprendizaje de la información y, en consecuencia, mejoraron sus funciones. Los nodos de la capa de salida utilizan la función softmax. Output_Layer

El número de nodos N, en la capa de salida, corresponde al número de clases.

Implementación

En este proyecto utilizamos mayormente dos tecnologías en este caso son:

- scala
- spark

spark
Apache Spark es un motor de análisis unificado para el procesamiento de datos a gran escala. Proporciona API de alto nivel en Java, Scala, Python y R, y un motor optimizado que admite gráficos de ejecución general. También es compatible con un amplio conjunto de herramientas de alto nivel que incluyen Spark SQL para SQL y procesamiento de datos estructurados, MLlib para aprendizaje automático, GraphX para procesamiento de gráficos y Structured Streaming para procesamiento incremental y flujo de datos.

scala

Scala combina programación funcional y orientada a objetos en un lenguaje conciso de alto nivel. Los tipos estáticos de Scala ayudan a evitar errores en aplicaciones complejas, y sus tiempos de ejecución de JVM y JavaScript le permiten construir sistemas de alto rendimiento con fácil acceso a enormes ecosistemas de bibliotecas.

Resultados

Decision Tree		Support Vector Machine		Logistic Regression		Multilayer Perceptron	
Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
0.84288999332 39374	21562 ms	0.88091389950 04407	16371 ms	0.88483350298 0951	8367 ms	0.83964201877 93427	20347 ms
0.83608368940 62178	22957 ms	0.88091389950 04407	16708 ms	0.88483350298 0951	13895 ms	0.83964201877 93427	18610 ms
0.83968900407 25657	21603 ms	0.88091389950 04407	16335 ms	0.88483350298 0951	9118 ms	0.83964201877 93427	33013 ms
0.84047827354 91397	24642 ms	0.88091389950 04407	17896 ms	0.88483350298 0951	16356 ms	0.83964201877 93427	19514 ms
0.83848594407 14233	24100 ms	0.88091389950 04407	18313 ms	0.88483350298 0951	8061 ms	0.83964201877 93427	17979 ms
0.83933333333 33334	25455 ms	0.88091389950 04407	18219 ms	0.88483350298 0951	8165 ms	0.83964201877 93427	20775 ms
0.84331968738 36993	26843 ms	0.88091389950 04407	19133 ms	0.88483350298 0951	8024 ms	0.83964201877 93427	19600 ms
0.84146250651 57495	25858 ms	0.88091389950 04407	18292 ms	0.88483350298 0951	7946 ms	0.83964201877 93427	21976 ms
0.83891884015 73656	24228 ms	0.88091389950 04407	16390 ms	0.88483350298 0951	8184 ms	0.83964201877 93427	19585 ms
0.83843826112 62824	24648 ms	0.88091389950 04407	16357 ms	0.88483350298 0951	8083 ms	0.83964201877 93427	20185 ms
0.83848	23649	0.88091	16623	0.88483	8128	0.83964	20123

594407 14233	ms	389950 04407	ms	350298 0951	ms	201877 93427	ms
0.82408 368940 62178	25142 ms	0.88091 389950 04407	16751 ms	0.88483 350298 0951	8214 ms	0.83964 201877 93427	19475 ms
0.83733 433553 12334	24846 ms	0.88091 389950 04407	16413 ms	0.88483 350298 0951	8050 ms	0.83964 201877 93427	20236 ms
0.84118 495352 39474	23252 ms	0.88091 389950 04407	17008 ms	0.88483 350298 0951	7856 ms	0.83964 201877 93427	18789 ms
0.82116 495452 824113	26483 ms	0.88091 389950 04407	16732	0.88483 350298 0951	8286 ms	0.83964 201877 93427	19419 ms

Average	Accuracy	Time
Decision Tree	0.8374235607	24351.2 ms
Support Vector Machine	0.8809138995004407	17169.4 ms
Logistic Regression	0.884833502980951	9115.533333333ms
Multilayer Perceptron	0.8396420187793427	20729.07143 ms

Conclusión

Para la realización de esta práctica de tuvimos que correr cada modelo un número determinado de veces lo cual fue una tarea laboriosa y tardada pero al final tuvimos los resultados esperados, al realizar el promedio podemos apreciar que el método de Logistic Regression es el más rápido de los cuatro, al mismo tiempo es el que tiene mayor precisión que los demás, esto es bastante importante al desarrollar cualquier tipo de proyecto ya que queremos saber que los resultados siempre sean consistentes.

Aun así después de comparar los resultados, se aprendió que todos los modelos son distintos, existen ventajas y desventajas que se pueden acomodar a nuestras necesidades.

Referencias

SVM

http://stening.blog/clasificacion-de-dispositivo-medico-utilizando-diferentes-tecnicas-de-machine-learning/

https://programmerclick.com/article/18211343732/

https://idus.us.es/bitstream/handle/11441/43808/Mart%C3%ADn%20Guare%C3%B1o%2C%20Juan%20Jos%C3%A9%20TFG.pdf?sequence=1&isAllowed=y

https://www.mathworks.com/discovery/support-vector-machine.html

https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=models-how-svm-works

Decision tree

https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

https://scikit-learn.org/stable/modules/tree.html

Logistic Regression

https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple

https://www.machinelearningplus.com/machine-learning/logistic-regression-tutorial-examples-r/#h-1-introduction-to-logistic-regression

https://spark.apache.org/docs/latest/ml-classification-regression.html#logistic-regression

https://spark.apache.org/docs/latest/

https://www.scala-lang.org/

Multilayer perceptron
https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1