Instituto Tecnológico y de Estudios Superiores de Monterrey

Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del modelo.

Inteligencia artificial avanzada para la ciencia de datos I (Gpo 103)
TC3006C.103

Agustin Tapia - A01367639

10 de Septiembre 2023

Abstract

En el presente trabajo, se busca hacer un análisis de rendimientos de métodos de machine learning de clasificación, en especial Regresión Logística y Decision Tree Classifier, además se utilizó gridsearch para lograr optimizar el modelo.

1 Introducción

La clasificación es un problema fundamental en el campo del machine learning supervisado, donde el objetivo es asignar un objeto o dato a una de varias categorías o clases. Los modelos de clasificación supervisados son una herramienta esencial en una amplia gama de aplicaciones, desde diagnósticos médicos hasta detección de spam en correos electrónicos y reconocimiento de patrones en imágenes. Estos modelos se entrenan utilizando datos de entrada etiquetados, lo que significa que se les proporciona información sobre las clases a las que pertenecen los ejemplos de entrenamiento. En esta introducción, exploraremos los conceptos clave detrás de los modelos de machine learning de clasificación supervisada y cómo desempeñan un papel esencial en la toma de decisiones automatizada y la resolución de problemas complejos en la actualidad

2 Marco Teórico

2.1 Modelos de Machine Learning de clasificación

Los modelos de aprendizaje automático de clasificación están diseñados para asignar etiquetas o categorías a puntos de datos en función de sus características. Analizan patrones en los datos de entrada y aprenden a hacer predicciones sobre la clase a la que pertenece un nuevo punto de datos, estos modelos funcionan entrenándose con datos etiquetados y ajustando sus parámetros para minimizar los errores de clasificación. Entre los algoritmos más conocidos están la regresión logística, los árboles de decisión, las máquinas de vectores soporte y las redes neuronales. Su rendimiento se evalúa utilizando métricas como la exactitud, la precisión, la recuperación y la puntuación F1, que miden la capacidad del modelo para clasificar correctamente los casos.

2.1.1 Regresión Logística

La regresión logística es un método estadístico utilizado para tareas de clasificación binaria. Modela la relación entre una variable dependiente binaria y una o más variables independientes estimando la probabilidad del resultado de la variable dependiente. . La regresión logística es útil en diversos campos, como la medicina y las finanzas, para predecir resultados como la presencia de enfermedades o el impago de préstamos basándose en características de entrada. Se utiliza para modelar la probabilidad de que se produzca un evento.

2.1.2 Decision Tree Classifier

Un clasificador de árbol de decisión es un algoritmo de aprendizaje automático supervisado que se utiliza tanto para tareas de clasificación como de regresión. Funciona dividiendo recursivamente el conjunto de datos en subconjuntos basados en el atributo más significativo, creando en última instancia una estructura similar a un árbol en la que cada nodo de hoja representa una predicción de

clase o valor, los árboles de decisión son fáciles de entender e interpretar, lo que los hace útiles para visualizar los procesos de toma de decisiones. Se utilizan mucho en tareas de clasificación para predecir resultados basados en características de entrada. Los árboles de decisión dividen los datos en función de los valores de las características para tomar decisiones secuenciales, lo que permite establecer límites de decisión complejos. Se emplean en diversos campos, como las finanzas, la sanidad y el marketing, para tareas como la detección de fraudes, el diagnóstico médico y la segmentación de clientes.

3 Resultados y análisis

3.1 Función y parámetros

Para nuestro modelo se utilizo la función Logistic_regresion de Sci kit Learn, en concreto la función queda así:

$$log\_reg = LogisticRegression()$$

Esta función tiene valores predeterminados como C: Inverso de la fuerza de regularización (valores más pequeños significan una regularización más fuerte), max_iter: Número máximo de iteraciones para que el solucionador converja, solucionador: Algoritmo a utilizar para la optimización (por ejemplo, 'liblinear', 'lbfgs', 'newton-cg', etc.), penalización: Tipo de regularización ('l1', 'l2', 'elasticnet', o 'none'), multi_class: Enfoque para tratar problemas multiclase ('ovr' para uno contra resto o 'multinomial' para pérdida multinomial) y random_state anteriormente mencionado.

3.2 Separación del modelo train/test/validation

3.2.1 Train test split

La separación del modelo se realiza a partir de la función de Sci kit Learn $Train\_test\_split$ consiste en dividir matrices o arreglos en subconjuntos aleatorios para el entrenamiento y la prueba, regularmente se utiliza una proporción alrededor de 80% para el dataset de entrenamiento y 20% para el dataset de testing, este valor se ajusta con el argumento $test\_size$, además para asegurar la reproducibilidad del modelo se utiliza el argumento $random\_state$ el cual controla la aleatoriedad o la semilla de aleatoriedad durante determinadas operaciones.

3.2.2 Validation

Se utilizó cross-validation al modelo de Regresión logística,

15-fold cross validation average accuracy	0.758

Table 1: Cross validation

3.3 Scale and transform data

Es importante escalar y transformar los datos porque podría ser útil para maximizar la eficiencia del modelo que se utiliza, cuando las variables tienen diferentes unidades o escalas puede ser difícil

compararlas directamente, además escalar y transformar los datos puede mejorar el rendimiento de algunos modelos que son sensibles a la escala de las variables. Para este proceso se utilizaron los siguientes comandos:

3.4 Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto

3.5 Diagnóstico y explicación el grado de varianza: bajo medio alto

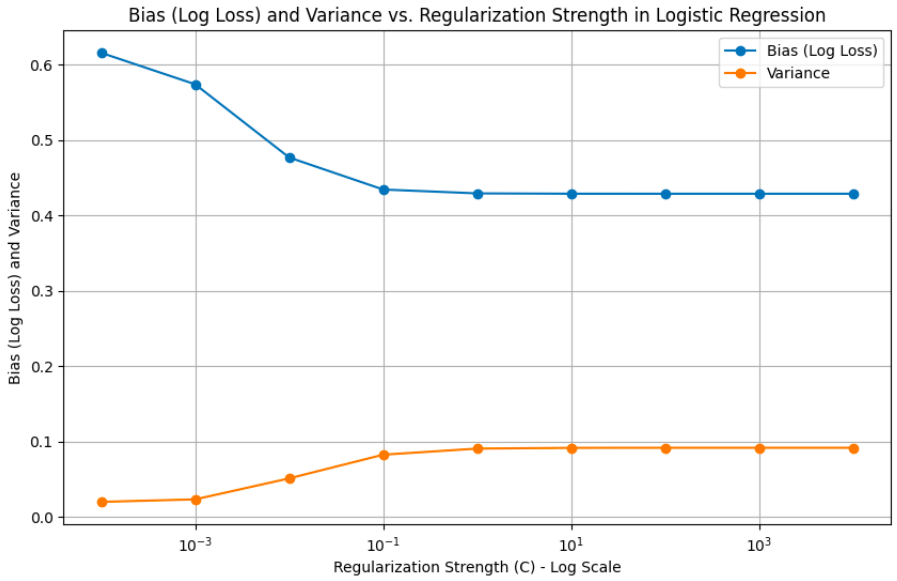


Figure 1: Bias and Variance

El modelo se encuentra en la zon a de underfitting, nunca logra pasar a una zona de overffiting ni realiza alguna interdección del bias y la varianza, por lo tanto se puede inferiri que independientemente del valor c, el modelo tiene muestra poca varianza pero un alto valor de bias.

3.6 Diagnóstico y explicación el nivel de ajuste del modelo: underfitt fitt overfitt

Se obtuvieron los siguientes valores de Training set score y Test set score, los resultados indican que en el modelo el test set score está muy por encima, algo que es inusual, la causa de esto podría ser principalmente en la variabilidad de los datos porque algunos datos pueden ser más favorables en el conjunto de testing.

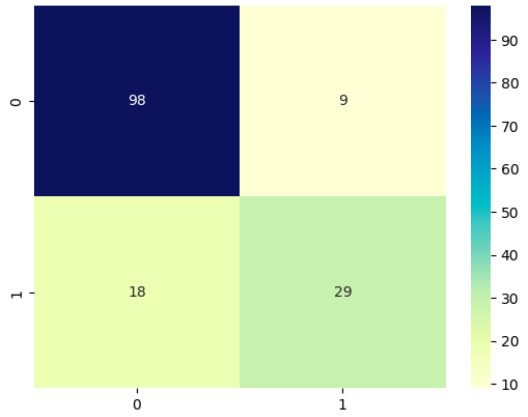
Subset	Score
Model accuracy score	0.8247
Training set score	0.7622
Test set score	0.8247

Table 2: Training set score and test score del modelo de Regresión Logística

3.6.1 Matriz de confusión y reporte de clasificación

	precision	recall	f1-score	support
0	0.84	0.92	0.88	107
1	0.76	0.62	0.68	47
accuracy			0.82	154
macro avg	0.80	0.77	0.78	154
weighted avg	0.82	0.82	0.82	154

(a) Reporte de clasificación



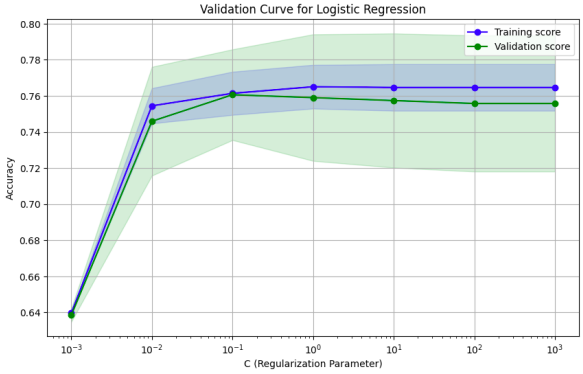
(b) Matriz de confusión

El análisis de la matriz de confusión proporciona información coherente con las expectativas del modelo en cuestión. Se observa que la precisión del modelo es aproximadamente del 80%. Cabe destacar que para este análisis, se estableció una relación de prueba (test score) del 20%, lo que resulta en un tamaño de muestra total de 769 observaciones.

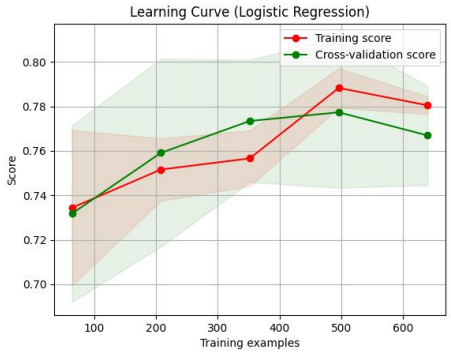
De las 769 observaciones en el conjunto de datos, el modelo identificó correctamente 154 casos como falsos negativos, lo que representa un 63.63% de los casos no diabéticos. Este resultado es coherente con la proporción de individuos sin diabetes en el conjunto de datos, que asciende al 69%.

Sin embargo, es importante señalar que el modelo enfrenta dificultades al identificar correctamente los casos de verdaderos positivos (true true). Solo logra una precisión del 18% en esta categoría, lo que está significativamente por debajo de la proporción general de casos de diabetes en el conjunto de datos, que se encuentra en un 31%. Esto sugiere que el modelo presenta dificultades en la identificación y clasificación precisa de los individuos con diabetes.

3.6.2 Curva de validación y curva de aprendizaje



(c) Validation curve



(d) Learning curve

Lo ideal es que la curva de validación y la curva de entrenamiento sean lo más parecidas posible. Si ambas puntuaciones son bajas, es probable que el modelo no se ajuste lo suficiente. [1]

Si la curva de entrenamiento alcanza una puntuación alta con relativa rapidez y la curva de validación se queda atrás, el modelo está sobreajustado. Esto significa que el modelo es muy complejo y que hay muy pocos datos, o podría significar simplemente que hay muy pocos datos. [1]

Un valor alto de C (por ejemplo, C=1.0 o mayor) da como resultado un modelo menos regularizado y, por lo tanto, más complejo. Un valor bajo de C (por ejemplo, C=0.01 o menor) da como resultado un modelo más regularizado y, por lo tanto, más simple. Un valor de C menor aumentará la regularización, lo que puede dar lugar a coeficientes menores (incluido el término de sesgo).

El análisis de la gráfica aprendizaje revela la presencia de underfitting, , dado que tanto la puntuación de validación como la de entrenamiento son notoriamente bajas y están cercanas entre sí, lo que revela un bias alto en todos los puntos conforme se incrementa muestras de entrenamiento.

Asimismo, en el análisis de la gráfica de validación, se observa una tendencia similar entre las puntuaciones de entrenamiento y validación en todos los valores de C evaluados, además que el valor de accuracy permance constante lo que significa que no logra aprender conforme el modelo se hace más complejo. Este hallazgo no proporciona evidencia concluyente de que el modelo tiene un sesgo alto en todos los valores de C ya que el valor de puntuación no logra incrementarse significativamente, por lo que se podría inferir que el problema recae en los datos. El modelo logra predecir acertadamente los datos en data que no ha visto, sin embargo, estos valores de accuracy no son altos.

3.7 Optimización

3.8 Gridsearch CV

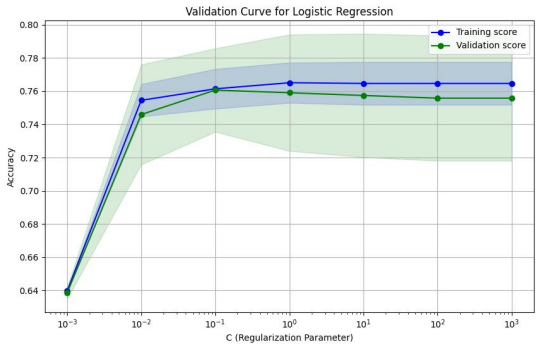
La función *Gridsearch* consiste en buscar sistemáticamente a través de una parrilla de parámetros especificada para encontrar los valores óptimos de los hiperparámetros para el modelo. Posterior a esto, estos valores se utilizan en el cross-validation para encontrar la mejor combinación de hiperparámetros para un modelo de aprendizaje automático. En la librería sci kit learn existe una función

que realiza estas dos partes y tiene el nombre de *GridsearchCV*. En cuánto a nuestro modelo de regresión logística, los hiperparámetros en donde se podría analizar para lograr un mejor aprendizaje es el valor "*C*", que afecta principalmente en la regularización del modelo y tiene que ver con el bias, esta relación es inversa, a medida que el valor de C aumenta, la fuerza de la regularización disminuye y, por lo tanto, el modelo se vuelve menos sesgado, y como antes fue mencionado, más complejo.

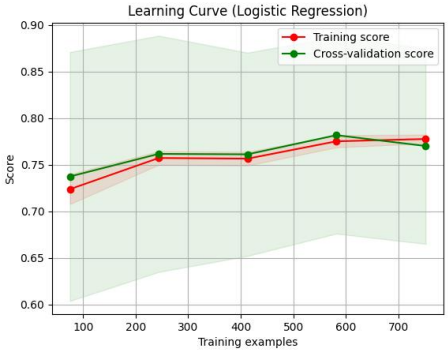
Subset	Score
Model accuracy score	0.7606
Training set score	0.7573
Test set score	0.8247

Table 3: Valores obtenidos con la optimización del Grid Search CV

Al analizar los valores, observamos una disminución en training de 0.7622 a 0.7573, sin embargo el test score se mantiene de 0.8247, estos resultados parecen un poco inusuales, lo que refleja algún error en la utilización del gridsearch cv, por lo que se podría optar por usar otro método como random search. De igual manera las gráficas respaldan el underfitting del modelo, que no logra capturar patrones en los datos.



(e) Validation curve



(f) Learning curve

3.9 Usar otro modelo de clasificación

Para intentar observar otro comportamiento y mejores resultados se utilizo Decision Tree Classifier para el mismo conjunto de datos, los resultados son los siguientes:

Subset	Score
Model score using Decision Tree	0.7662
Training set score	0.7622
Test set score	0.8247

Table 4: Resultados obtenidos utilizando Decision Tree

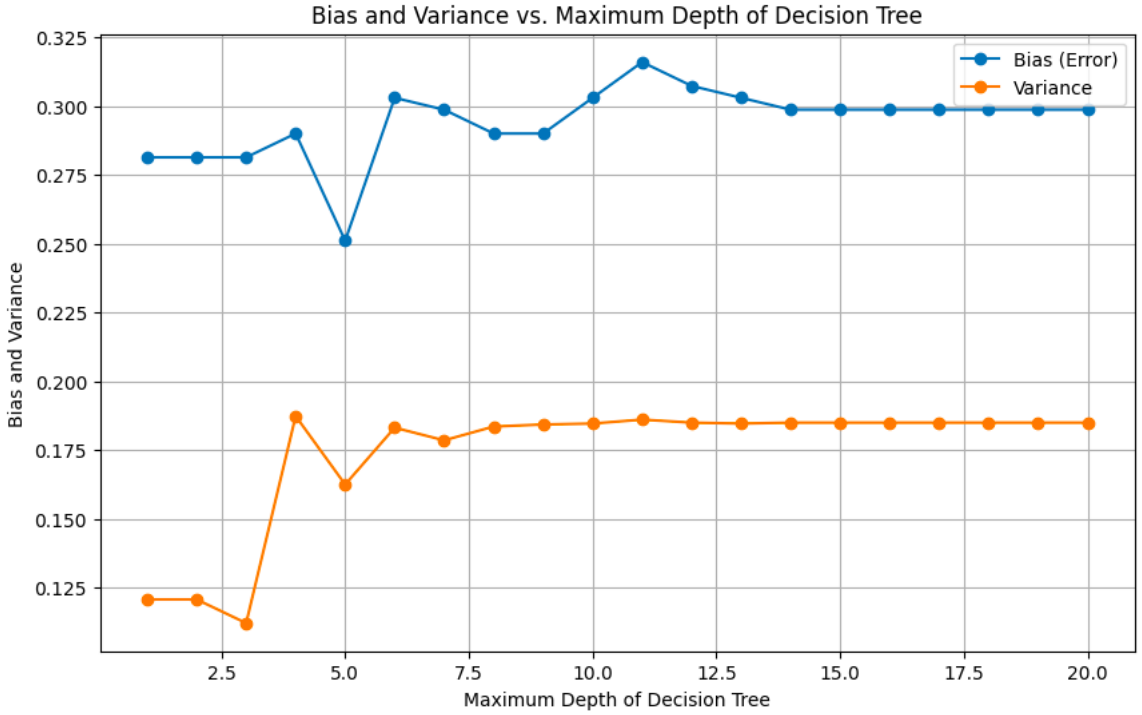
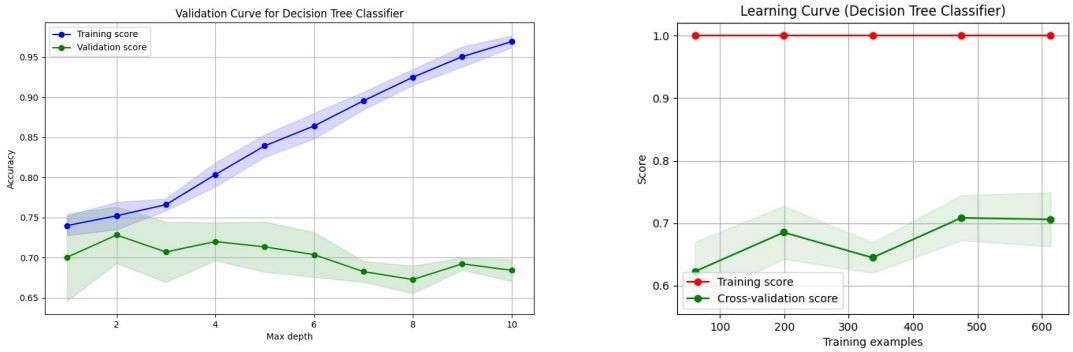


Figure 2: Caption

Al analizar la gráfica utilizando decision Tree, nos damos cuenta que el modelo está sobre ajustado en casi todo los valores de profundidad del modelo, ya que se tiene una varianza alta y un bias bajo.



(a) Validation curve

(b) Learning curve

Observamos el mismo comportamiento donde el test set score es las grande que el training set score, algo completamente inusual, se podrías inferir que el problema podría estar relacionado con el dataset utilizado. Al utilizar este modelo observamos un overfitting en la gráfica de la curva de aprendizaje, el modelo se ajusta demasiado al modelo de entrenamiento, pero es incapaz de hacer el

datos que no ha visto, como lo observamos en la línea de cross validation que nunca logra ser similar a la línea de entrenamiento.

De igual manera, el análisis de la gráfica de validación respalda la idea de que existe overfitting para valores mayores de profundidad el modelo es incapaz de hacer predicciones de nuevos datos, sin embargo, el modelo cuadno incluyes valores de profundiad bajos hace buenas predicciones.

3.10 Usar otra base de datos

Se utilizó otra base de datos para el conjunto de entrenamiento de regresión logistica, el data set consiste la identificación de tumores en beningno y maligno de Kaggle, los resultados fueron los siguientes:

Subset	Score
Model accuracy score CV	0.9649
Training set score	0.9890
Test set score	0.9649

Table 5: Regresión Logística utlizando otra base de datos

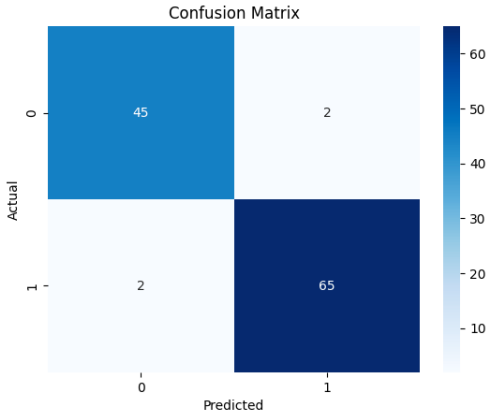
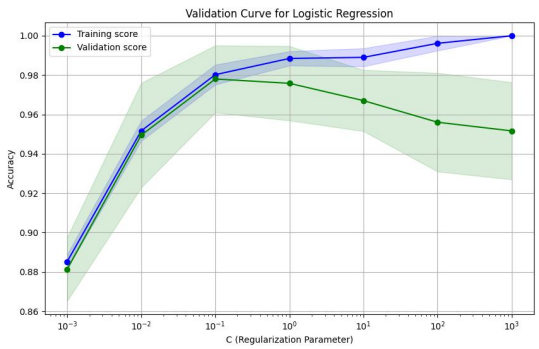
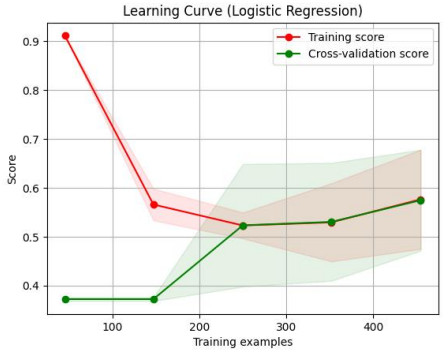


Figure 3: Matriz de confusión



(a) Validation curve



(b) Learning curve

Al emplear un conjunto de datos diferente, se evidencia una notable mejora en múltiples aspectos. Analizando la gráfica de curva de aprendizaje nos damos cuenta en algún punto de muestra de entrenamiento los modelos logran tener las mismas predicciones, sacrificando el valor de accuracy o score, esto demuestra que el modelo después de determinadas muestras de entrenamiento pasa a ser underfitting.

En cuánto al análisis de la curva de validación se observa que el modelo, muestra overfitting a valores mayores de C, ya que los ressultados de entrenamiento logran entrenarse perfectamente pero no tan bien cuando se le presenta data que no ha visto, además es importante recalcar que con 0.1 el modelo presenta predicciones altamente exactas y precisas respecto a la data de validación.

Es fundamental esclarecer la influencia del valor 'C' en el sesgo del modelo, cuando 'C' asume valores más bajos, la regularización se torna más rigurosa, lo que se traduce en una mayor inclinación hacia el sesgo en el modelo debido a su restricción para adaptarse plenamente a los datos de entrenamiento. En este contexto, se observa eficazmente la manifestación del subajuste, ya que las representaciones gráficas exhiben una proximidad significativa entre sí.

En contraste, conforme el valor de 'C' aumenta, la brecha entre ambas representaciones gráficas se amplía, respaldando así la influencia del hiperparámetro en el proceso de ajuste del modelo. En otras palabras, un 'C' más elevado resulta en una disminución de la regularización, lo que conlleva a una mayor susceptibilidad al sobreajuste por parte del modelo , que puede traducirse en un sesgo reducido, pero conlleva el riesgo de una varianza más elevada.

De igual manera se utilizó Decision Tree Classifier con el otro conjunto de los datos, los resultados fueron los siguientes:

Subset	Score
Model accuracy score with criterion gini	0.9035
Training set score	0.9890
Test set score	0.9649

Table 6: Resultados utilizando otra base de datos y Decision Tree Classifier

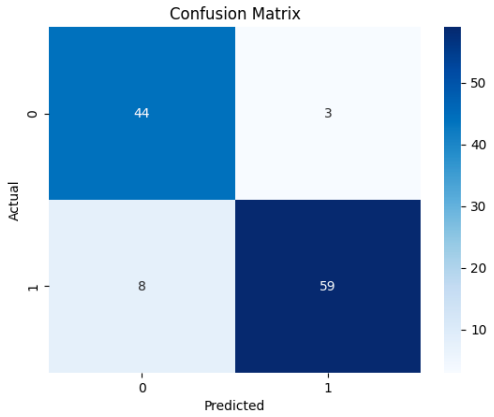
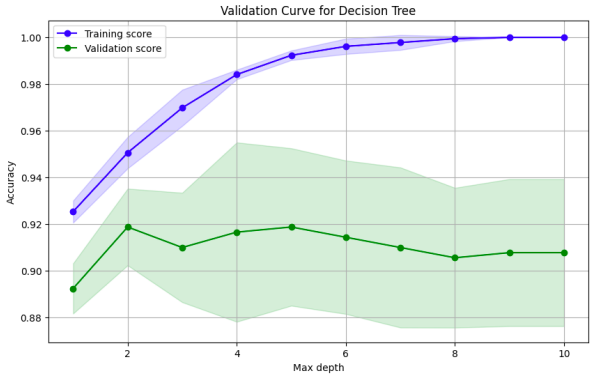
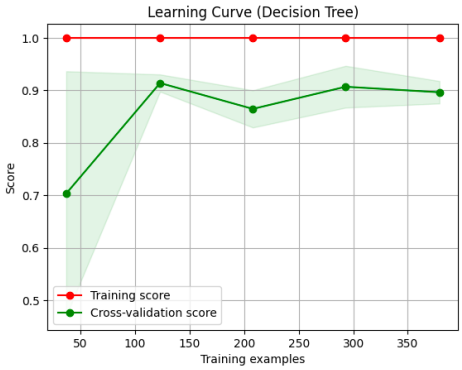


Figure 4: Matriz de confusión



(a) Curva de validación Decision Tree Classifier



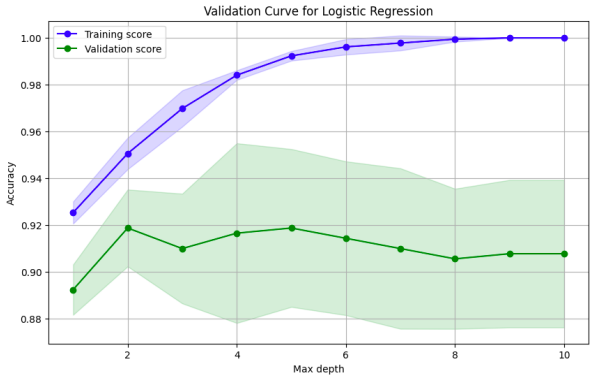
(b) Curva de aprendizaje Decision Tree Classifier

Aquí tenemos un caso de overfitting cuando la profundindad del decisión tree classifier es mayor a 2, esto es porque para valores más grandes el modelo aprendió muy bien de los datos de training, incluyendo el ruido y los patrones específicos que pueden no generalizarse bien a datos que jamás ha visto el modelo, y para datos que no ha visto no puede lograr identificar los patrones.

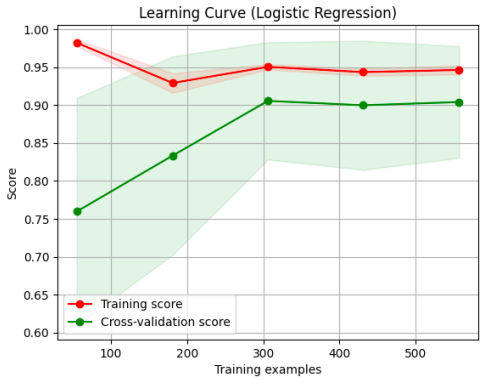
Además, en cuánto al análisis de la curva de aprendizaje, el modelo aprende perfectamente las predicciones de los datos de entrenamiento independientemente de las muestras de entrenamiento, y el modelo logra regularizarse para los datos que no ha visto.

Subset	Score
Model accuracy best score	0.9187
Training set score	0.9516
Test set score	0.9649

Table 7: Diagnóstico uilizando GridSearch CV



(c) Curva de validación Decision Tree Classifier utilizando los hiperparámetros del Gridsearch CV



(d) Curva de aprendizaje Decision Tree Classifier utilizando los hiperparámetros del Gridsearch CV

Cuando se utiliza el gridsearch, la gráfica es la misma porque se vuelve a graficar la profundiad del modelo de Decision Tree Classifier, sin embargo al analizar la curva de aprendizaje, observamos un comportamiento óptimo para realizas predicciones con bastante nivel de exactitud y precisión, por ello se recalca la importancia de utilizar diferentes métodos de machine learning para abordar un mismo problema.

4 Conclusión

En resumen, este trabajo destaca varias conclusiones cruciales en el campo de los modelos de Machine Learning. En primer lugar, se resalta la importancia de los parámetros de optimización, ya que estos desempeñan un papel fundamental en la capacidad de un modelo para aprender de los datos y realizar predicciones precisas. Además, se enfatiza la necesidad de contar con bases de datos limpias y diversas, ya que la calidad y variedad de los datos son factores determinantes en la efectividad de los modelos.

Asimismo, se subraya la relevancia de la capacidad de aplicar diferentes métodos en el proceso de construcción del modelo, ya que la elección adecuada de algoritmos y técnicas puede marcar la diferencia en los resultados obtenidos. Finalmente, se destaca que la complejidad de un modelo no siempre se traduce en un mejor rendimiento, ya que tanto el underfitting como el overfitting pueden obstaculizar la capacidad de generalización del modelo, de igual manera es importante utiizar adecuados modelos de optimización dependiendo de los parámetros que tienen los diferentes modelos.

En conjunto, estas conclusiones subrayan la importancia de un enfoque equilibrado y bien fundamentado en el diseño y entrenamiento de modelos de Machine Learning, teniendo en cuenta tanto los aspectos técnicos como la calidad de los datos y la capacidad de generalización para lograr resultados efectivos en problemas del mundo real.

References

[1] GeeksforGeeks. Validation curve - geeksforgeeks. https://www.geeksforgeeks.org/validation-curve/, 2023. Accedido el 10 de septiembre de 2023.