Desempeño de un Modelo de Machine Learning con framework

Eduardo Gonzalez Luna - A01658281

Resumen

En este documento se explicará el proceso de la realización de un modelo de clasificación de machine learning con regresión logística, usaremos el framework de sklearn. Usaremos un dataset sobre el tipo de arroz, esté lo sacamos de UCI Machine Learning Repository.

Introducción

En el contexto de nuestra clase de Inteligencia Artificial Avanzada para la Ciencia de Datos, estamos explorando una serie de temas relacionados con el aprendizaje automático. En esta ocasión, estamos embarcándonos en la implementación de un modelo de regresión logística, un algoritmo ampliamente utilizado en el ámbito del aprendizaje supervisado, y más específicamente, en la resolución de problemas de clasificación.

Nuestro enfoque de clasificación se centra en un conjunto de datos que gira en torno a granos de arroz, que se caracterizan mediante diversos atributos:

***Superficie*:** Esto cuantifica la cantidad de píxeles contenidos dentro de los límites de cada grano de arroz.

Contorno: Se determina calculando la circunferencia a partir de las distancias entre píxeles alrededor de los bordes de los granos de arroz.

Longitud del eje principal: Esta medida representa la longitud máxima que se puede trazar dentro de cada grano de arroz.

Longitud del eje menor: Mide la longitud mínima que se puede trazar en el grano de arroz.

Excentricidad: Evalúa la forma de la elipse que mejor se ajusta a las características del grano de arroz.

Área convexa: Cuantifica el número de píxeles que conforman la capa convexa más pequeña que engloba al grano de arroz.

Extensión: Proporciona la relación entre la región del grano de arroz y los píxeles del cuadro delimitador que lo encierra.

Categoría: Comprende las clases "Commeo" y "Osmancik". Nuestra tarea es utilizar los datos de este conjunto para predecir la categoría a la que pertenece cada grano de arroz que tengamos entre manos.

Metodología

Inicialmente, llevamos a cabo la selección de datos y los transformamos en un DataFrame de pandas. A continuación, realizamos la predicción y eliminamos cualquier información que no sea pertinente para nuestra tarea de predicción. Al usar el framework de sklearn tenemos 3 hiperparametros: Solver, Penalty y Max_iter.

Con la primera configuración obtuvimos una precisión en el test de 81%, podemos ver el comportamiento en la siguiente matriz.

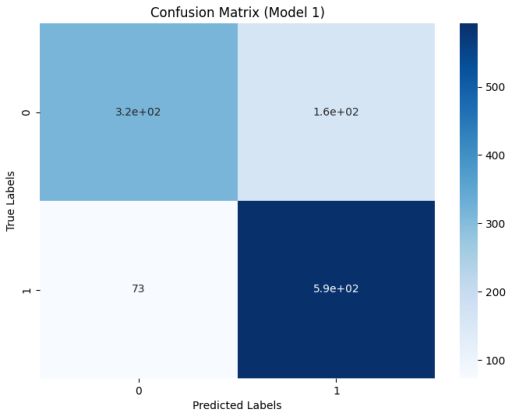

Figura 1. Matriz del modelo 1

Para mejorar la precisión cambiaremos los hiperparametros. Ponemos Solver: ‘lbfgs’ y Max_inter: 5000.

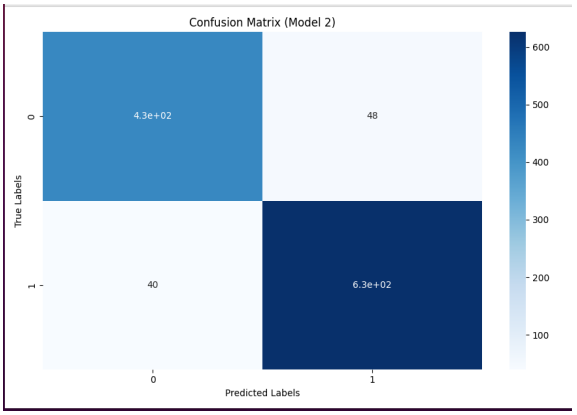

Figura 2. Matriz del modelo 2

Pudimos observar que mejoramos la precisión hasta un 93%.

Análisis y Resultados

En las representaciones gráficas de aprendizaje que se muestran a continuación, podemos examinar tanto el sesgo como la varianza del modelo, lo que nos permite identificar si el modelo está experimentando una situación de underfitting, que se presenta cuando el sesgo es notablemente elevado, o una situación de overfitting, que se manifiesta cuando el sesgo es bajo pero la varianza es alta. También podemos determinar si el modelo está bien ajustado, lo que significa que el sesgo es bajo y la varianza es también baja.

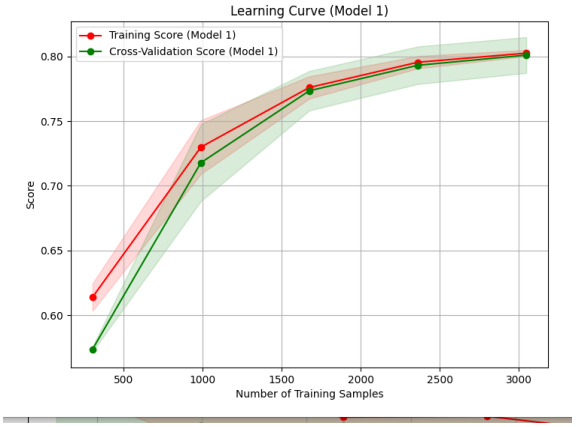

Figura 3. Learning curve modelo 1

En la figura superior, podemos observar cómo el modelo mejora sus predicciones a medida que avanza a través de las iteraciones. En ningún momento llega a experimentar una situación de overfitting, ya que el sesgo es reducido desde el principio y, a medida que avanzan las iteraciones, el modelo converge hacia un ajuste óptimo.

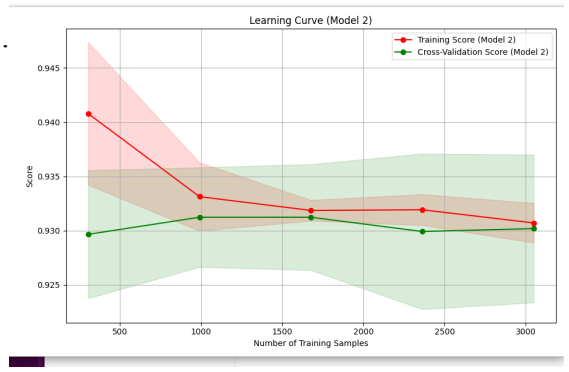

Figura 4. Learning curve modelo 2

En este caso el modelo logra alcanzar un ajuste adecuado. Inicialmente, presenta una situación de underfitting, pero a medida que avanza a través de las iteraciones, logra reducir el sesgo hasta converger y finalmente obtenemos un modelo más adecuadamente ajustado que el anterior.

Conclusiones

Al evaluar los errores tanto en el conjunto de entrenamiento como en el conjunto de prueba. En el primer modelo, los errores son bajos, pero al realizar ajustes en algunos de los hiperparámetros mencionados anteriormente en el marco de trabajo, logramos reducir aún más los errores tanto en el conjunto de entrenamiento como en el conjunto de prueba.

Bibliografía

Rice (Cammeo and Osmancik). (2019). UCI Machine Learning Repository. https://doi.org/10.24432/C5MW4Z.