Proyecto. GridSearch para el diagnóstico de tumores malignos de cáncer de mama

Introducción

El cáncer de mama es uno de los cánceres más comunes y que provoca más muertes entre las mujeres. Si es diagnosticado a tiempo, este cáncer puede tratarse y curarse, previniendo así la muerte de la paciente diagnosticada.

En este trabajo, se utilizó una base de datos de diagnóstico de cáncer de mama obtenida de Kaggle (https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset). En dicha base de datos, se tiene el registro de 569 muestras de tumores de mama y los valores promedio, error estándar y máximo de las siguientes 10 características de dichos tumores:

- Radio de los lóbulos
- Textura superficial
- Perímetro externo de los lóbulos
- Área de los lóbulos
- Niveles de suavidad
- Compacidad
- Concavidad
- Puntos cóncavos
- Simetría
- Dimensión Fractal

A la par de dichas características, se tiene el diagnóstico del tumor (maligno o benigno).

Desarrollo

Por lo mencionado anteriormente, estamos ante un problema de clasificación binaria cuya solución se determinó realizando una búsqueda de rejilla o *grid search* de 6 diferentes clasificadores (KNN, Random Forest, Perceptron, SVM, Gaussian Naive Bayes y Regresión logística).

Para esto, primero fue necesario manipular los datos de la base de datos, de modo que los valores categóricos fueran numéricos, y asegurarse de que no existieran valores de tipo NaN. Luego, se crearon los conjuntos de entrenamiento y validación para así normalizarlos.

Posteriormente, se realizó un análisis de componentes principales (PCA) para determinar el mínimo de características necesarias para lograr una suma de varianza explicada mayor a 99.99% sobre todo el conjunto de datos. Este análisis determinó que los promedios del radio, textura, perímetro, área, suavidad y compacidad eran las características cuya suma de varianza explicada es mayor al 99.99% sobre todo el conjunto de datos. El PCA para estos mismos componentes

en el conjunto de entrenamiento dio una suma de varianza explicada de 88.88% y de 89.82% sobre el conjunto de validación.

Finalmente, se pudo realizar la búsqueda de rejilla en cada uno de los clasificadores previamente mencionados. Se evaluó el desempeño de cada clasificador con 3 diferentes métricas (MCC, F1 y AUC-ROC) en las etapas de entrenamiento y validación. No obstante, se determinó el valor del MCC como métrica para determinar al mejor conjunto de hiperparámetros durante el entrenamiento.

Se tomó en cuenta que la búsqueda de rejilla puede ser un proceso que demora a medida que se consideran más valores de hiperparámetros a evaluar, por lo que no se tomaron en cuenta muchos hiperparámetros en el caso de los clasificadores Random Forest, Perceptron y SVM. A pesar de esto, se obtuvieron resultados satisfactorios de acuerdo a las métricas.

Resultados

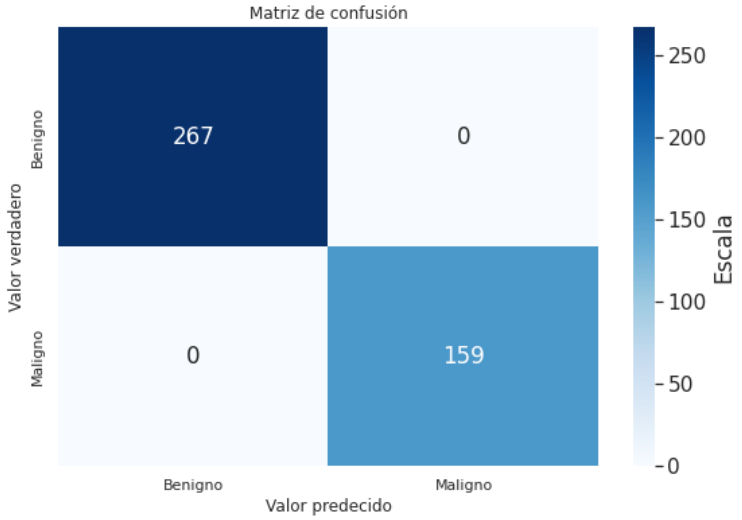

Figura 1. Matriz de confusión en la etapa de entrenamiento de KNN. MCC=0.92, F1=1, AUC-ROC=1. Mejores hiperparámetros: 'algorithm': 'ball_tree', 'n_neighbors': 10, 'p': 1, 'weights': 'distance'.

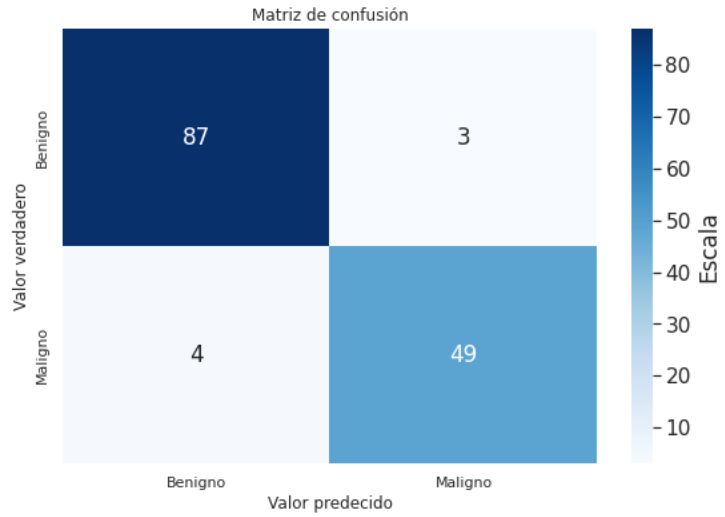

Figura 2. Matriz de confusión en la etapa de validación de KNN. MCC=0.92, F1=0.93, AUC-ROC=0.94. Mejores hiperparámetros: 'algorithm': 'ball_tree', 'n_neighbors': 10, 'p': 1, 'weights': 'distance'.

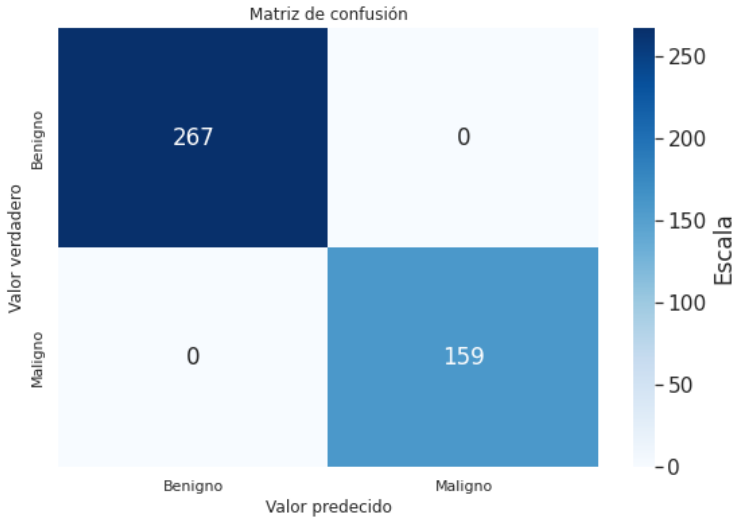

Figura 3. Matriz de confusión en la etapa de entrenamiento de Random Forest. MCC=0.90, F1=1, AUC-ROC=1. Mejores hiperparámetros: 'criterion': 'entropy', 'max_depth': None, 'n_estimators': 200.

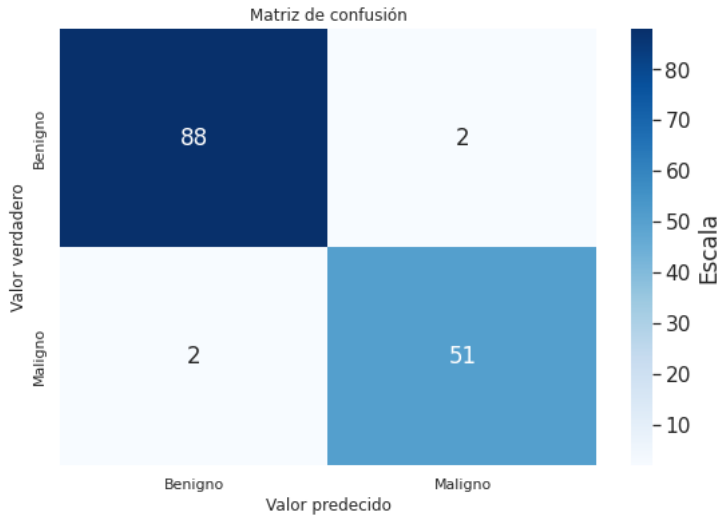

Figura 4. Matriz de confusión en la etapa de validación de Random Forest. MCC=0.94, F1=0.96, AUC-ROC=0.97. Mejores hiperparámetros: 'criterion': 'entropy', 'max_depth': None, 'n_estimators': 200.

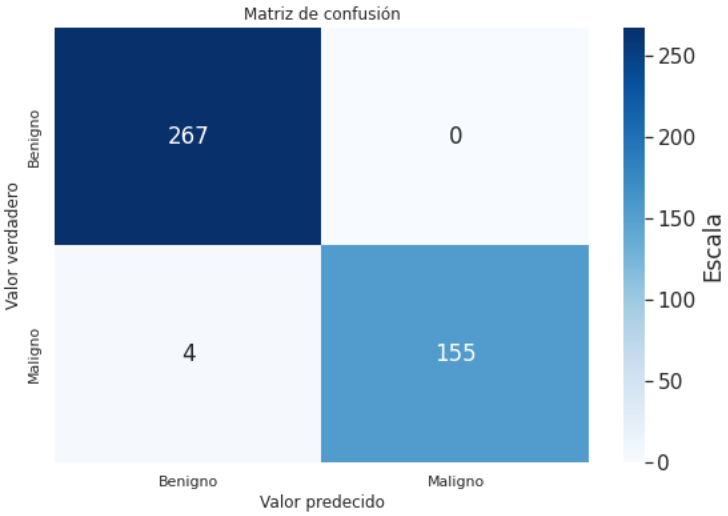

Figura 5. Matriz de confusión en la etapa de entrenamiento de Perceptron. MCC=0.96, F1=0.98, AUC-ROC=0.98. Mejores hiperparámetros: 'activation': 'logistic', 'hidden_layer_sizes': (100, 5), 'learning_rate': 'invscaling', 'max_iter': 500.

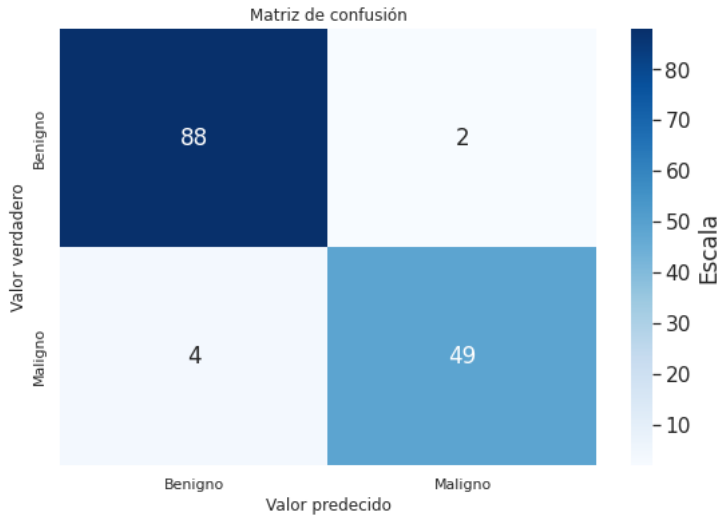

Figura 6. Matriz de confusión en la etapa de validación de Perceptron. MCC=0.90, F1=0.94, AUC-ROC=0.95. Mejores hiperparámetros: 'activation': 'logistic', 'hidden_layer_sizes': (100, 5), 'learning_rate': 'invscaling', 'max_iter': 500.

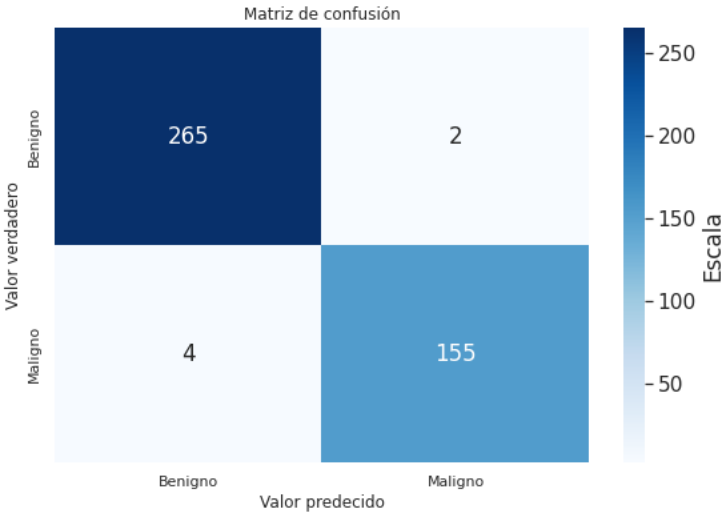

Figura 7. Matriz de confusión en la etapa de entrenamiento de SVM. MCC=0.96, F1=0.98, AUC-ROC=0.98. Mejores hiperparámetros: 'algorithm': 'C': 1, 'coef0': 0, 'degree': 3, 'gamma': 'scale', 'kernel': 'linear'.

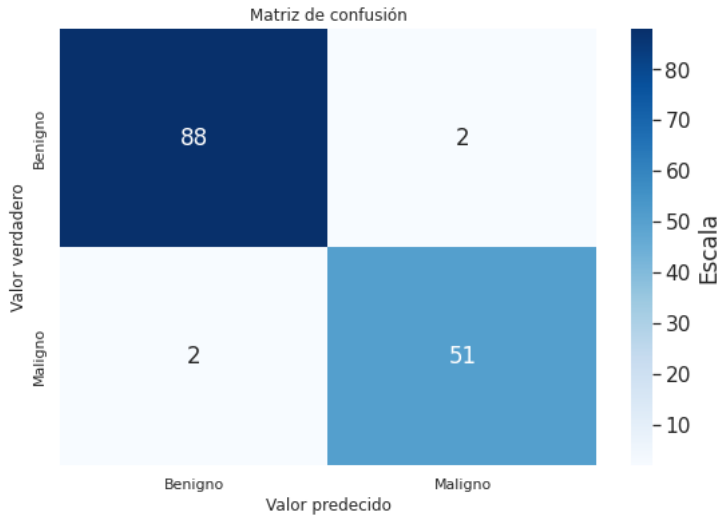

Figura 8. Matriz de confusión en la etapa de validación de SVM. MCC=0.94, F1=0.96, AUC-ROC=0.97. Mejores hiperparámetros: 'C': 1, 'coef0': 0, 'degree': 3, 'gamma': 'scale', 'kernel': 'linear'.

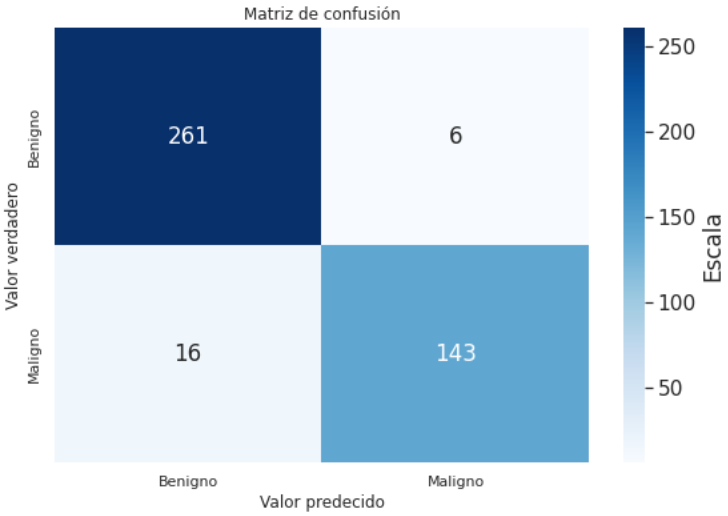

Figura 9. Matriz de confusión en la etapa de entrenamiento de Gaussian Naive Bayes. MCC=0.88, F1=0.92, AUC-ROC=0.93. Gaussian Naïve Bayes no tiene hiperparámetros a iterar.

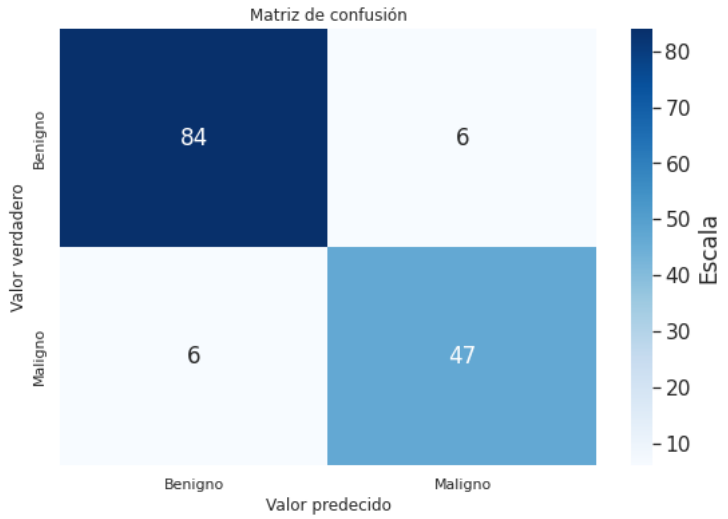

Figura 10. Matriz de confusión en la etapa de validación de Gaussian Naive Bayes. MCC=0.0.82, F1=0.88, AUC-ROC=0.91. Gaussian Naïve Bayes no tiene hiperparámetros a iterar.

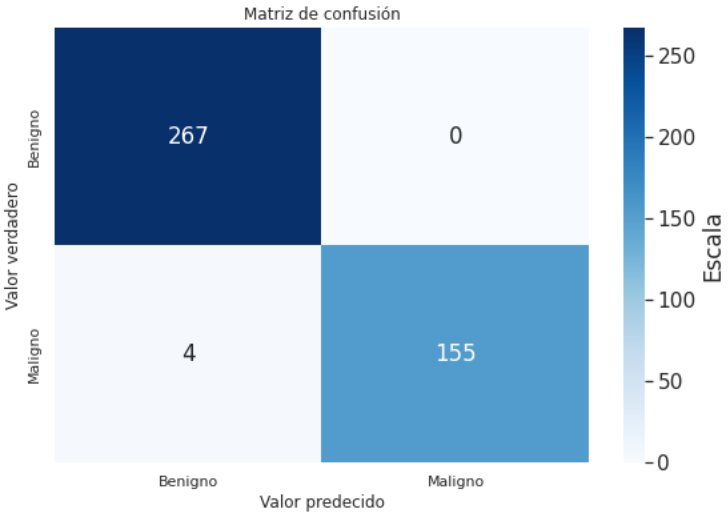

Figura 11. Matriz de confusión en la etapa de entrenamiento de Regresión Logística. MCC=0.96, F1=0.98, AUC-ROC=0.98. Mejores hiperparámetros: 'C': 0.1, 'class_weight': None, 'dual': False, 'penalty': 'none', 'solver': 'sag'.

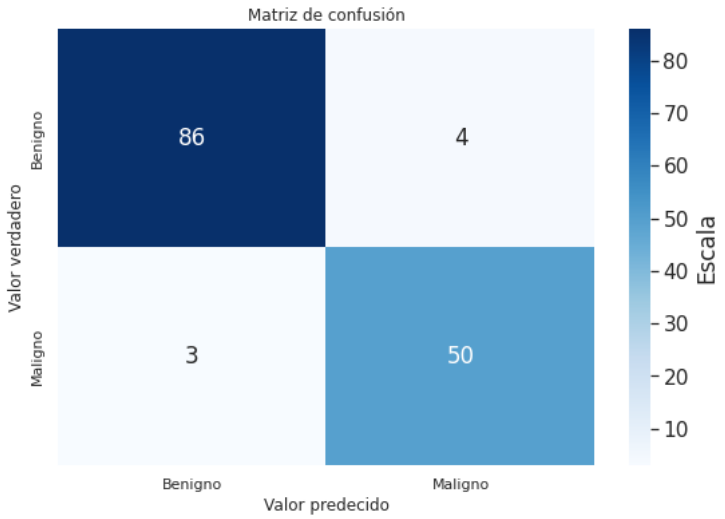

Figura 12. Matriz de confusión en la etapa de validación de Regresión Logística. MCC=0.89, F1=0.93, AUC-ROC=0.94. Mejores hiperparámetros: 'C': 0.1, 'class_weight': None, 'dual': False, 'penalty': 'none', 'solver': 'sag'.

Con base en los valores de MCC de cada uno de los clasificadores, notamos que el clasificador con mejor desempeño fue SVM con valores de MCC 0.96 y 0.94 en las etapas de entrenamiento y validación respectivamente, mientras que el peor clasificador de Gaussian Naive Bayes con valores de MCC de 0.88 y 0.82 en etapas de entrenamiento y validación respectivamente.

Conclusiones

El método de búsqueda por rejilla compromete recursos computacionales y tiempo a cambio de resultados de alta confiabilidad que ayudan a tomar decisiones sobre cuál es el método de machine learning que resuelve nuestro problema y cuáles son los hiperparámetros que dan el desempeño óptimo de acuerdo a una métrica dada. De requerirse un mayor desempeño, podría haberse realizado una segunda búsqueda de rejilla sobre los demás parámetros de SVM.