Clasificación de medicamentos con Machine Learning

Inteligencia Artificial Avanzada para la Ciencia de Datos

Presentado por:

Samuel Pelaez Aleman

Profesores:

Jesús Adrián Rodríguez Rocha

Septiembre 2024

I. INTRODUCCIÓN

En este reporte se presenta una solución para la clasificación de cinco tipos diferentes de medicamentos, basada en diversas características del paciente. El conjunto de datos utilizado, titulado Drug Classification, fue obtenido de la plataforma Kaggle [1]. Este dataset contiene información clave sobre las características de los pacientes, como edad, sexo, presión arterial, nivel de colesterol, y la relación sodio-potasio en el cuerpo, entre otros. A partir de estas características, se construyó un modelo predictivo que permite clasificar de manera automática el tipo de medicamento más adecuado para un paciente dado.

II. EXPLORACIÓN DE DATOS

Antes de proceder con el desarrollo de los modelos predictivos, es fundamental realizar una exploración preliminar del conjunto de datos para comprender su estructura y comportamiento. Este análisis inicial permite identificar aspectos clave como la presencia de valores nulos, la distribución de las variables, y el balance entre las clases de la variable objetivo. A través de esta exploración, se garantiza que el dataset sea adecuado para el modelado y se detectan posibles problemas que podrían afectar el rendimiento de los modelos, como desbalances significativos en las clases o características irrelevantes.

Cada instancia en el conjunto de datos representan características (*features*) clave de los pacientes y el medicamento (*target*): Age, que corresponde a la edad; Sex, que indica el sexo del paciente; BP, que hace referencia a la presión arterial; Na to K, que describe la relación entre los niveles de sodio y potasio en la sangre; Cholesterol, que es el nivel de colesterol del paciente; y Drug, que es el medicamente asignado a ese paciente. El tipo de cada característica se detallan en la Tabla I, mientras que se pueden observar ejemplos representativos en la Tabla II.

Para comprender el comportamiento de cada característica, se realizaron visualizaciones mediante histogramas y gráficos de barras, dependiendo del tipo de dato. Estas visualizaciones permiten observar la distribución y frecuencia de los valores presentes en el conjunto de datos. En particular, la distribución de la edad se ilustra en el histograma de la Fig. 1, mientras que el ratio de sodio y potasio se visualiza en la Fig. 2. Además, se utilizaron gráficos de barras para representar la frecuencia de los valores de sexo en Fig. 3, presión arterial en Fig. 4, y colesterol en Fig. 5. Cabe aclarar que dentro del dataset no hay ningún valor nulo dentro del mismo.

Feature	**Tipo de dato**
Age	numérico
Sex	categórico
BP	categórico
Cholesterol	categórico
Na_to_K	numérico
Drug	categórico

Table I. Tipos de datos del dataset

Feature	**Ejemplo**
Age	23
Sex	F
BP	HIGH
Cholesterol	HIGH
Na_to_K	25.355
Drug	DrugY

Table II. Ejemplos de datos del dataset

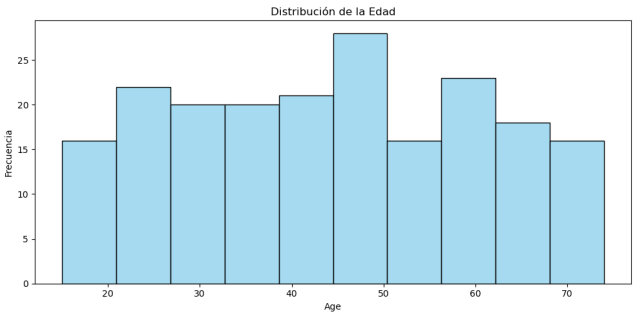

Figure 1. Distribución de la edad

Una vez que se ha comprendido el comportamiento de los datos, es necesario realizar un tratamiento adecuado para asegurar que los modelos puedan generar predicciones más precisas. Los features numéricos, como la edad y el ratio de sodio/potasio, fueron normalizados

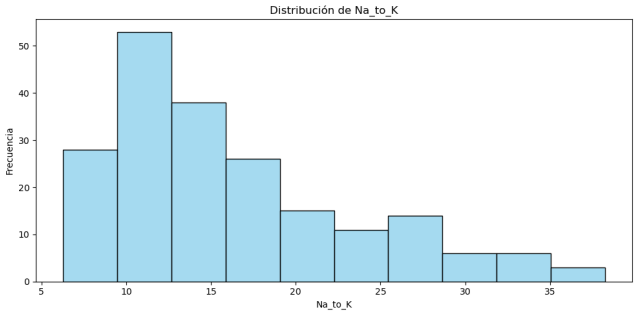

Figure 2. Distribución de Na to K

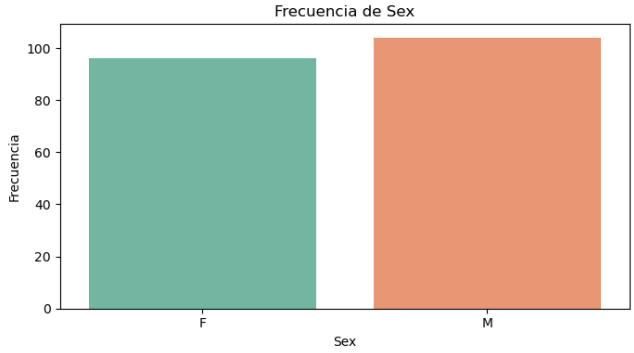

Figure 3. Frecuencia del sexo

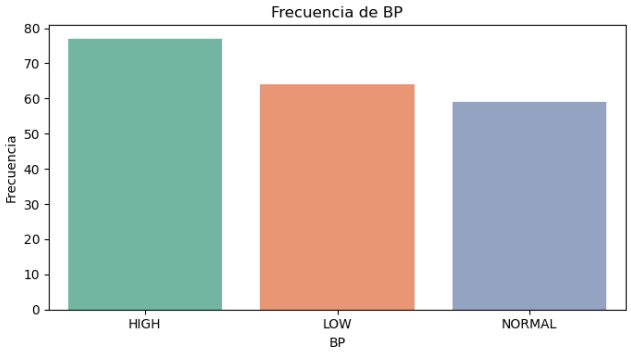

Figure 4. Frecuencia de la presión arterial

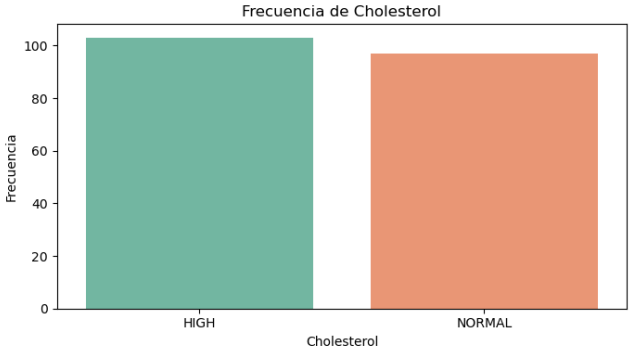

Figure 5. Frecuencia del nivel de colesterol

de manera individual. La normalización es una técnica que ajusta los valores de una variable para que se encuentren en un rango común, normalmente entre 0 y 1. Esto se hace para evitar que los modelos den más importancia a los valores con magnitudes más grandes, como la edad o el ratio de sodio/potasio, que podrían afectar el rendimiento si no se ajustan.

Por otro lado, los features categóricos (como el sexo, la presión arterial y el nivel de colesterol) se transformaron utilizando One Hot Encoding, una técnica que convierte cada categoría en una columna binaria que toma los valores 0 o 1, según corresponda, permitiendo que los modelos trabajen con estas variables de manera eficiente.

El conjunto de datos fue dividido en tres subconjuntos: train, validation y test. Inicialmente, se separó el 20% de los datos para el conjunto test, con el objetivo de reservar una parte del dataset para la evaluación final. El 80

El subconjunto train se utilizó para entrenar los modelos, mientras que el conjunto validation sirvió para evaluar el rendimiento de los modelos durante el proceso de ajuste. Finalmente, el conjunto test se reservó como la última etapa de evaluación, donde se mide la capacidad de generalización del modelo seleccionado.

III. SELECCIÓN, CONFIGURACIÓN Y ENTRENAMIENTO DEL MODELO

A. Selección

Para realizar la predicción del target, es fundamental escoger el modelo más adecuado que sea capaz de generalizar correctamente. En este caso, se consideraron dos modelos: Regresión Logística y Random Forest, debido a su eficacia en tareas de clasificación.

La Regresión Logística es un modelo de clasificación lineal que estima la probabilidad de que una observación pertenezca a una clase particular. Funciona al ajustar una función sigmoide (o logística) que transforma las predicciones lineales en probabilidades. Es ideal para

problemas de clasificación binaria y se puede extender a casos multiclase, como el presente, mediante técnicas como la regresión logística multinomial. Es un modelo simple pero eficaz, especialmente cuando las relaciones entre las características y el target son aproximadamente lineales.

Random Forest es un modelo de ensamble basado en la creación de múltiples árboles de decisión. En lugar de construir un único árbol, Random Forest construye varios árboles durante el entrenamiento y combina sus resultados para mejorar la precisión y evitar el sobreajuste. Cada árbol es entrenado con una muestra aleatoria del dataset, lo que permite al modelo captar múltiples patrones de los datos. Este modelo es particularmente efectivo en problemas de clasificación multiclase y puede manejar tanto características numéricas como categóricas con facilidad.

B. Configuración y entrenamiento

Para la Regresión Logística, se utilizaron los hiperparámetros por defecto de la librería scikit-learn, evaluando así el modelo en su configuración más básica. De igual manera, el modelo de Random Forest se configuró con los hiperparámetros predeterminados de la misma librería.

Una vez entrenados ambos modelos utilizando el conjunto de train, se realizaron predicciones en el conjunto de validation y se evaluaron los resultados utilizando las métricas de accuracy, precision y F1-score, que se describen brevemente a continuación:

- Accuracy: Es el porcentaje de predicciones correctas sobre el total de predicciones realizadas, y refleja el rendimiento general del modelo.
- Precision: Mide la proporción de predicciones positivas correctas sobre el total de predicciones positivas realizadas por el modelo, enfocándose en la calidad de las predicciones positivas.
- F1-score: Es el promedio armónico entre precisión y recall, y ofrece un equilibrio entre ambas métricas, siendo útil cuando las clases están desbalanceadas o se desean evitar falsos negativos o positivos.

Las métricas resultantes de esta evaluación se presentan en la Tabla III, mientras que las matrices de confusión correspondientes a cada modelo se visualizan en las Fig. 6 y Fig. 7.

Modelo	Accuracy	Precision	F1-Score
Random Forest	1.0	1.0	1.0
Logistic Regression	0.8542	0.8776	0.8487

Table III. Métricas de rendimiento de los modelos en el conjunto de validación

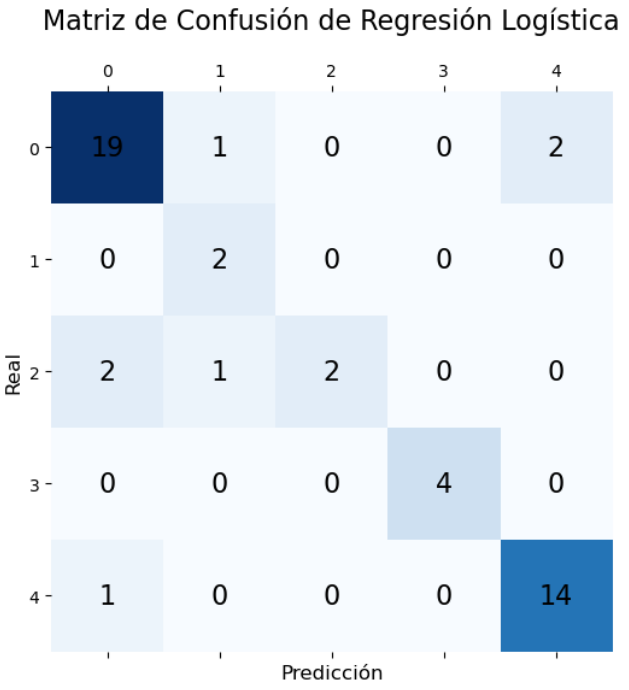

Figure 6. Matriz de confusión regresión logística

El modelo de Random Forest muestra métricas perfectas en el conjunto de validación, con accuracy, precision y F1-Score de 1.0. Aunque estos resultados pueden parecer ideales, en realidad indican un posible sobreajuste. Esto ocurre cuando el modelo se ajusta demasiado bien a los datos de entrenamiento y validación, capturando tanto patrones relevantes como ruido, lo que reduce su capacidad para generalizar a nuevos datos.

En términos de sesgo, Random Forest tiene un sesgo muy bajo, lo que significa que ha aprendido todos los patrones disponibles, pero a costa de una alta varianza. La alta varianza sugiere que el modelo es muy sensible a los datos específicos de entrenamiento y podría

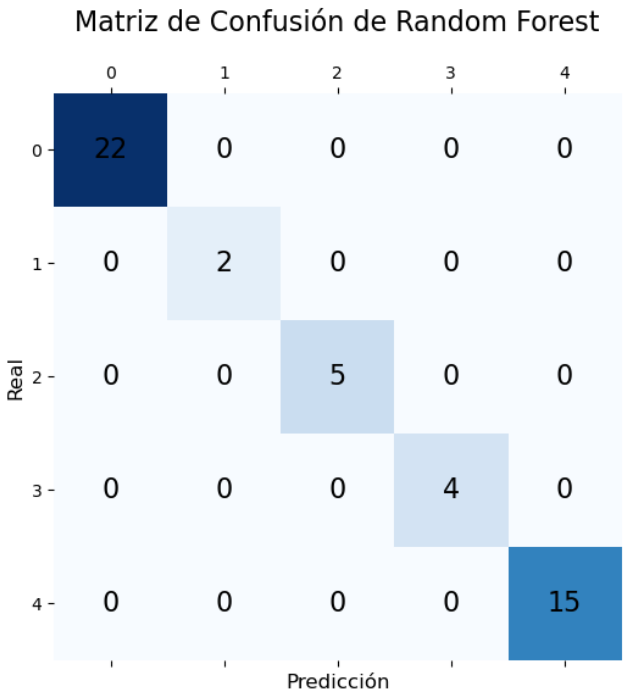

Figure 7. Matriz de confusión random forest

no funcionar bien en datos nuevos. Debido a esto, no conviene hacer el modelo más complejo con técnicas de regularización o refinamiento, ya que podría sobreajustarse aún más.

El modelo de Regresión Logística muestra métricas de accuracy, precision y F1-Score más equilibradas, lo que sugiere que tiene una mejor capacidad de generalización en comparación con Random Forest. El modelo tiene un sesgo moderado, lo que significa que no está capturando toda la complejidad de los datos, pero tampoco está sobreajustado.

La varianza del modelo es baja, lo que indica que no es tan sensible a los cambios en los datos de entrenamiento y generaliza mejor. Debido a que hay espacio para mejorar su rendimiento, se aplicarán técnicas de refinamiento y regularización para reducir el sesgo y obtener predicciones más precisas y cercanas a la realidad.

IV. REFINAMIENTO DEL MODELO

Para mejorar el rendimiento del modelo de Regresión Logística, se utilizó Grid Search, una técnica que explora exhaustivamente diversas combinaciones de hiperparámetros para encontrar la configuración óptima. En este caso, se busca mejorar la precisión del modelo mediante la optimización de ciertos parámetros.

El proceso de regularización es clave en modelos de clasificación como la regresión logística. La regularización impone una penalización a los coeficientes del modelo para evitar que se ajusten demasiado a los datos de entrenamiento, reduciendo así el riesgo de sobreajuste. Los hiperparámetros que se optimizan incluyen el parámetro C, que controla la intensidad de la regularización. En este caso, los valores evaluados para C son: [0.01, 0.1, 1, 10, 100]. Valores más bajos de C implican una regularización más fuerte, mientras que valores más altos permiten mayor flexibilidad al modelo. Las penalizaciones posibles que se exploraron son L1 (Lasso) y L2 (Ridge). La penalización L1 tiende a forzar algunos coeficientes a cero, eliminando características irrelevantes, mientras que L2 reduce el peso de los coeficientes sin eliminarlos por completo.

El solver utilizado es liblinear, un algoritmo eficiente para resolver problemas de regresión logística en datasets pequeños. Este solver es compatible tanto con la penalización L1 como con L2, lo que lo hace adecuado para este tipo de problemas.

En el Grid Search, se utiliza cv=5 para realizar una validación cruzada de 5 pliegues, lo que asegura que el modelo se evalúe de manera robusta en diferentes subconjuntos de datos. La métrica de evaluación seleccionada es precision macro, que optimiza la precisión promedio entre todas las clases, garantizando que el modelo prediga con precisión todas las clases de manera equilibrada.

Una vez entrenado el modelo de Regresión Logística utilizando Grid Search, los mejores hiperparámetros encontrados fueron C = 10 y penalty = L1 (Lasso). Estos hiperparámetros permiten un buen equilibrio entre regularización y flexibilidad, haciendo que el modelo elimine características irrelevantes mientras conserva las más importantes.

En el conjunto de validación, el modelo refinado alcanzó una precisión de 0.91, lo que representa una mejora significativa en comparación con la configuración inicial. Este resultado sugiere que el modelo ha mejorado su capacidad de generalización, prediciendo

de manera más precisa las clases en el dataset.

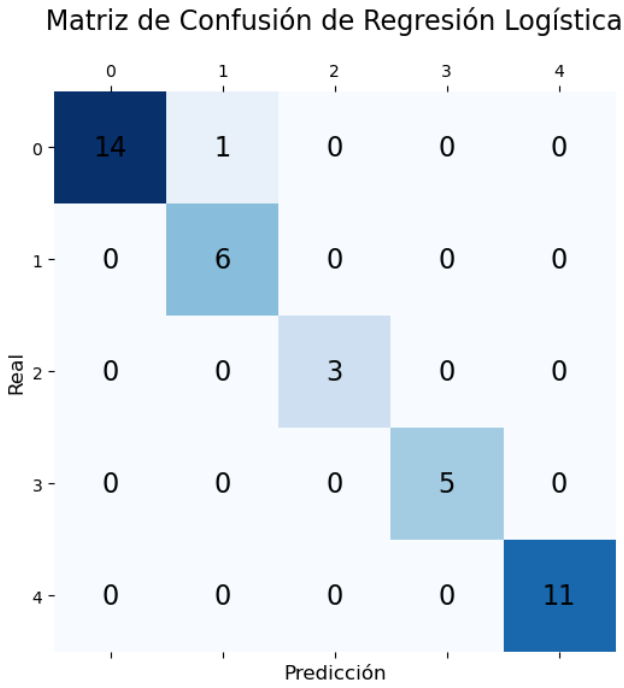

Figure 8. Matriz de confusión regresión logística con grid search

V. RESULTADOS Y CONCLUSIÓN

Finalmente, se evaluó el modelo de Regresión Logística con Regularización en el conjunto de test para comprobar su capacidad de generalización con datos "reales".

Las métricas obtenidas en el conjunto de prueba son las siguientes: Accuracy = 0.975, Precision = 0.971, y F1-Score = 0.978. Estas métricas indican que el modelo está generalizando bien a nuevos datos. En la Fig. 8 se presenta la matriz de confusión, la cual permite identificar en qué clases el modelo tiende a cometer errores y dónde puede mejorar.

El modelo muestra un sesgo bajo, lo que significa que ha capturado adecuadamente los patrones de los datos, ajustándose bien a las relaciones presentes. Sin embargo, el sesgo no es tan bajo como para causar sobreajuste, lo cual es positivo. En cuanto a la varianza, el rendimiento alto en el conjunto de prueba sugiere que la varianza es moderada, lo que indica que el modelo es robusto y no es excesivamente sensible a los datos de entrenamiento. Esto permite que el modelo generalice bien sin caer en un comportamiento errático frente a nuevos datos.

En este estudio, se desarrollaron y compararon dos modelos de clasificación, Random Forest y Regresión Logística con Regularización, para predecir el tipo de medicamento adecuado basado en características de los pacientes. Aunque Random Forest presentó métricas perfectas en el conjunto de validación, estas indicaron un claro sobreajuste, lo que limita su capacidad de generalización. Por otro lado, el modelo de Regresión Logística mostró un mejor equilibrio entre sesgo y varianza, lo que lo convirtió en una opción más robusta y confiable para predecir con precisión en nuevos datos. Después de aplicar un proceso de refinamiento mediante Grid Search, el modelo de regresión logística mejoró significativamente, alcanzando una precisión del 0.91 en el conjunto de validación y 0.971 en el conjunto de prueba, demostrando su capacidad de generalización y rendimiento estable en datos reales.

[1] Pratham Tripathi, *Drug Classification* (Kaggle, 2020)