Aplicación de Aprendizaje Automático en la Resolución de Problemas de Clasificación de Alubias Secas

Pablo Chantada Saborido Aldana Smyna Medina Lostaunau
Pablo Verdes Sánchez Claudia Vidal Otero

May 6, 2024

Contents

1 Introducción

Las alubias secas son las leguminosas más importantes utilizadas para el consumo humano directo, comprenden casi la mitad de los granos leguminosos consumidos en la mayoría de países en desarrollo. Por lo tanto, el uso de técnicas Aprendizaje Automático y Deep Learning para aplicar métodos de clasificación es importante en ámbitos como la agricultura.

Categorizar las diferentes variedades de semillas y evaluar su calidad es de suma importancia, ya que tiene un impacto significativo en la producción de cultivos o en las industrias alimenticias para una fácil y rápida selección de las variedades de alubias.

Este proyecto se basa en el registro de siete tipos de alubias mediante fotos de alta resolución. Sus características se obtuvieron a partir de estas imágenes, dividiéndose en varias categorías, tales como forma, textura, aspecto, redondez... Estas características serán identificadas y evaluadas por el sistema desarrollado.

A continuación, se describirá el problema a resolver de manera exhaustiva y detallada, seguido de un Análisis Bibliográfico de las fuentes que se utilizaron. También se describe la base de datos del problema y se explicarán los resultados que se obtienen de cuatro modelos distintos, redes de neuronas artificiales (RR.NN.AA), máquinas de soporte vectorial (SVM), árboles de decisión y kNN. Finalmente, discutiremos los resultados obtenidos y una solución al problema propuesto.

2 Descripción del Problema

Nuestro problema a resolver se basa en la clasificación múltiple utilizando un conjunto de datos de alubias(Link) recopilado del repositorio de datos de Machine Learning de la Universidad de California, Irvine (UCI). El conjunto de datos inicial consta de aproximadamente 14.000 instancias de alubias con proporciones dispares[Figura 1], por ello decidimos balancearlo, quedándonos con 522 instancias de cada tipo de alubia, en total 3654 instancias [Figura 2]. El objetivo es clasificar estas alubias según su variedad de origen.

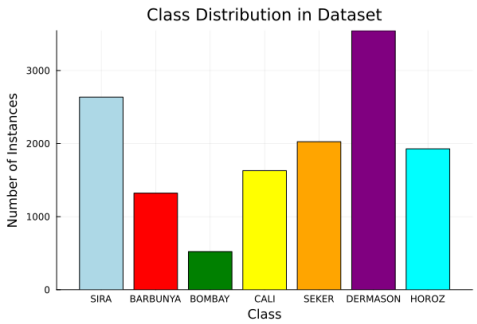

Figure 1: Representación de las clases en el dataset original.

Nuestra base de datos comprende un conjunto de 16 medidas de 7 tipos de alubias, doce dimensiones y cuatro formas distintas, aunque nosotros sólo utilizaremos 6 medidas por motivos que explicaremos más adelante. Estas variables son representadas por números reales, habiendo unas pocas excepciones de datos representados como enteros.

Con el fin de obtener imágenes de los granos secos, se diseñó e implementó un sistema de visión por computadora [1]. El sistema consiste en una lente de cámara, una cámara de captura de imágenes y una caja de iluminación especial para evitar la formación de sombras en el fondo. Para proporcionar un entorno de iluminación homogéneo, la caja se iluminó con luces en la parte superior. Las señales de las muestras de alubia seca fueron capturadas por la cámara y transferidas a la computadora. En total, se obtuvieron 13.611 muestras de alubias secas de 236 imágenes.

Esencialmente, para evaluar la precisión de nuestro modelo, se utilizan métricas como la precisión, la F1-score y la matriz de confusión, que evalúan cómo de bien el modelo clasifica las muestras de datos en las clases correctas. Para simplificar el análisis utilizaremos como clasificador únicamente la precisión.

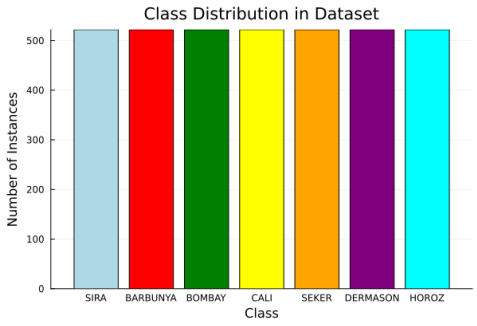

Figure 2: Representación de las diferentes clases en el dataset balanceado

Un problema de clasificación múltiple en Machine Learning implica asignar una etiqueta o categoría a cada instancia de datos de un conjunto, donde hay más de dos posibles clases a las que puede pertenecer, en el caso de nuestro problema existen 7 posibles clases (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira). Para abordar este tipo de problema, se utilizan modelos de Machine Learning que aprenden a reconocer patrones y características distintivas en los datos de entrenamiento. Estos modelos pueden variar desde máquinas de vectores de soporte (SVM) y regresión logística multinomial hasta enfoques más avanzados como redes neuronales convolucionales (CNN) o árboles de decisión. En el caso de nuestra aplicación, usaremos redes de neuronas artificiales (RR.NN.AA), máquinas de soporte vectorial (SVM), árboles de decisión(DT) y k-Vecinos Más Cercanos (kNN).

Una vez entrenados, estos modelos pueden realizar predicciones sobre nuevas muestras de datos, asignándoles una clase basada en los patrones que ha aprendido durante el entrenamiento.

3 Análisis Bibliográfico

Los estudios seleccionados abordan diversas técnicas y metodologías para la clasificación de datos de alubias secas, utilizando herramientas como visión por computadora, Machine Learning y Deep Learning [1]. Mendigoria, C. H. y Concepcion, R. [2] proponen un enfoque innovador que utiliza segmentación de semillas y modelos de regresión para clasificar variedades de frijol seco según características como calidad y tamaño. Por otro lado, Taspinar, Y. S. y Dogan, M. [3] destacan la eficacia de las máquinas de vectores de soporte (SVM) en la detección de enfermedades en cultivos, empleando técnicas de transferencia de aprendizaje para optimizar los resultados. Además, Sahu, P. y Singh, A. [5] demuestran la superioridad de los modelos de deep Learning, como GoogleNet y VGG16, en la detección automática de enfermedades en hojas de frijol.

Hasan, M. [7] muestra cómo las redes neuronales superan a los enfoques tradicionales. Naik, N. [4] enseña la optimización de datos en el aprendizaje automático, y F. Ferreira Lima dos Santos [8] utiliza técnicas de procesamiento de imágenes en la clasificación de granos de café. Además, el estudio de Md. Salauddin KhaN. [6] presenta el uso del clasificador Naïve Bayes (NB), un clasificador probabilístico que aplica el teorema de Bayes que no es sensible al ruido, adecuado para aplicaciones en tiempo real en la clasificación y predicción de enfermedades. Estos estudios resaltan el papel crucial de la tecnología en la agricultura moderna.

4 Desarrollo

4.1 Descripción

Para minimizar el ruido hemos decidido eliminar las columnas que presentan una desviación típica inferior a 1 (AspectRatio, Eccentricity,Exent,Solidity, Roundness, Compactness y Shapefactor 1,2,3 y 4.)[Tabla 1].

Esta base de datos contiene 13611 filas de instancias. Nos quedamos con 6 columnas de features 3, siendo estas las que describen las diferentes medidas de las alubias secas que estamos evaluando, y una última columna de objetivo, de tipo categórico, que nos indica las 7 posibles salidas de nuestro problema, es decir, los posibles tipos de alubias (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz y Sira).

La base de datos no presenta valores nulos, siendo cada uno de los features específico para cada alubia. Usamos en todas las features la normalización MinMax, ya que nuestros valores se encuentran en un intervalo controlado, no presentan valores fuera del rango, por lo que no existe la necesidad de utilizar la normalización de media 0.

Nuestra base de datos posee las siguientes columnas de datos después de la limpieza:

- Area (Integer): Área de la alubia, número de píxeles que la componen.
- Perimeter (Continuous): Circunferencia de la alubia, el largo de su borde.
- MajorAxisLenght (Continuous): Distancia entre los dos puntos más alejados de la alubia.
- MinorAxisLenght (Continuous): Distancia de la linea más larga que se puede dibujar manteniéndose perpendicular al eje central.
- ConvexArea (Integer): Número de píxeles que ocupa el polígono convexo más pequeño que puede contener el área de una alubia.
- EquivDiameter (Continuous): Diámetro de un círculo con el mismo área que la alubia.

No	Features	Min	Max	Mean	Std
1	Area	20420.00	254616.00	53048.28	29324.10
2	Perimeter	524.74	1985.37	855.28	214.29
3	MajorAxisLength	183.60	738.86	320.14	85.69
4	MinorAxisLength	122.51	460.20	202.27	44.97
5	AspectRation	1.02	2.43	1.58	0.25
6	Eccentricity	0.22	0.91	0.75	0.09
7	ConvexArea	20684.00	263261.00	53768.20	29774.92
8	EquivDiameter	161.24	569.37	253.06	59.18
9	Extent	0.56	0.87	0.75	0.05
10	Solidity	0.92	0.99	0.99	0.00
11	Roundness	0.49	0.99	0.87	0.06
12	Compactness	0.64	0.99	0.80	0.06
13	ShapeFactor1	0.00	0.01	0.01	0.00
14	ShapeFactor2	0.00	0.00	0.00	0.00
15	ShapeFactor3	0.41	0.97	0.64	0.10
16	ShapeFactor4	0.95	1.00	1.00	0.00

Table 1: Distribución estadística de la base de datos original.

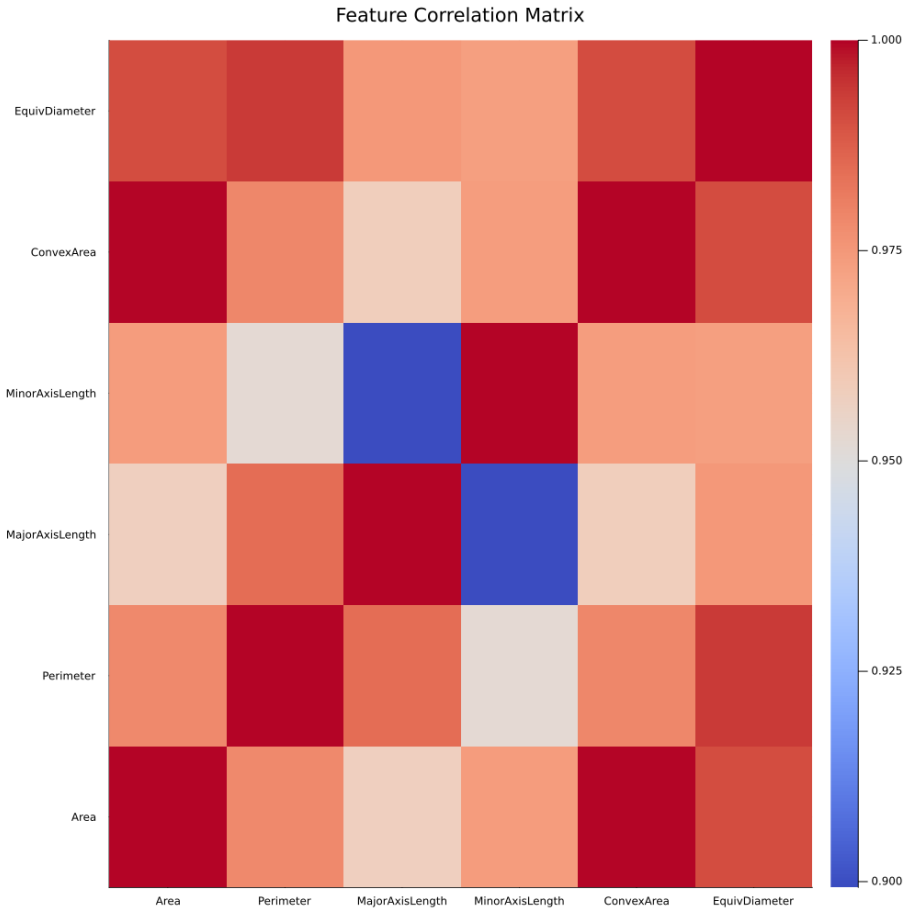

Figure 3: Gráfica representativa de la relación entre las variables.

En la creación de nuestro modelo destacamos las siguientes características:

- Número de Folds: pensamos en utilizar un rango de entre 5 y 10 Folds, pero finalmente decidimos utilizar 5 Folds, ya que, consideramos que la mejora entre 5 y 10 Folds no es suficiente para suplir el coste temporal.
- Índices de validación: para obtener los índices de validación utilizamos el método cross-validation, en este método, se parte el conjunto de datos en k subconjuntos disjuntos y se realizan k experimentos. En el k-ésimo experimento, el subconjunto k se separa para realizar test, y los k-1 restantes se utilizan para entrenar, realizando un k-fold cross-validation. Finalmente, el valor de test correspondiente a la métrica adecuada será el valor promedio de los valores de cada uno de los k experimentos.

- Parámetros de los modelos:

  RR.NN.AA: Topología (número de capas y número de neuronas en cada una): [20,10], [10, 10], [10, 15], [30, 20], [15, 15],[50, 25], [40, 20], [25, 15]

  SVM:

  kernel: [”linear”, ”rbf”, ”poly” y ”sigmoid”]

  C: para cada kernel usamos 2 y 3.

  Gamma: automático.

  coef0: 0.5

  DecisionTreeClassifier(max_depth): para la profundidad empezamos en un valor de 4 hasta llegar a un valor de 11.

  KNNeighbors(n_neighbour): para el número de vecinos empezamos en 4 hasta llegar a una cantidad de 11.

4.2 Resultados

A continuación, se mostrarán los resultados de las cuatro técnicas con las que realizamos el experimento: RR.NN.AA, SVM, árboles de decisión y kNN. Probando con distintos hiperparámetros.

A continuación, se muestran las iteraciones para cada modelo y la precisión media así como su desviación estándar:

- RRNNAA : [Tabla 2]
- DecisionTree : [Tabla 3]
- KNNeighbors : [Tabla 4]
- SVN : [Tabla 5]

No	Topology	Mean_accuracy	Std_accuracy
1	[20, 10]	90.94%	0.00677
2	[10, 10]	90.36%	0.00545
3	[10, 15]	90.80%	0.00727
4	[30, 20]	91.56%	0.00606
5	[15, 15]	91.13%	0.00706
6	[50, 25]	91.71%	0.00479
7	[40, 20]	91.62%	0.00536
8	[25, 15]	91.40%	0.00698

Table 2: Resultados de precisión de RR.NN.AA con distintos parámetros.

No	max_depth	Mean_accuracy	Std_accuracy
1	4	81.80%	0.01657
2	5	84.27%	0.00970
3	6	88.34%	0.01040
4	7	89.60%	0.01375
5	8	89.51%	0.01140
6	9	89.71%	0.01096
7	10	89.63%	0.00840
8	11	89.19%	0.01003

Table 3: Resultados de precisión de DecisionTree con distintos parámetros.

No	n_neighbors	Mean_accuracy	Std_accuracy
1	4	90.4%	0.00977
2	5	90.85%	0.01002
3	6	90.90%	0.01064
4	7	91.13%	0.00751
5	8	91.34%	0.00756
6	9	91.21%	0.00936
7	10	91.2%	0.00713
8	11	91.21%	0.01007

Table 4: Resultados de precisión de KNNeighbors con distintos parámetros.

No	C	kernel	Mean_accuracy	Std_accuracy
1	3	"linear"	89.95%	0.00763
2	2	"linear"	89.24%	0.00363
3	3	"rbf"	87.90%	0.00646
4	2	"rbf"	86.18%	0.00729
5	3	"poly"	84.67%	0.00911
6	2	"poly"	82.32%	0.00962
7	3	"sigmoid"	83.71%	0.00996
8	2	"sigmoid"	82.18%	0.01061

Table 5: Resultados de precisión de los SVM con distintos parámetros.

Tenemos que mencionar que estos resultados no son únicos y pueden variar con otro tipo de parámetros. Por ejemplo, en las máquinas de soporte, valores de C superiores a 10 pueden generar resultados pésimos. Asimismo, en el caso de los KNN o DT, el uso de hiperparámetros muy bajos nos da precisiones que no llegan ni al 50%. Para estos dos últimos, es importante destacar que con valores más altos la precisión comienza a disminuir, al igual que pasaba con valores muy bajos.

4.3 Discusión

En todas las iteraciones, los modelos han superado el 80% de precisión. Sin embargo, hay una diferencia sustancial si queremos obtener el mejor resultado (4). Los dos únicos modelos que superan el umbral del 90% son los KNN y las RR.NN.AA.

Al tener KNN y RR.NN.AA resultados muy similares tenemos que optar por elegir precisión o tiempo. Las Redes nos dan una mejor precisión con la desventaja de un tiempo de entrenamiento elevado (a comparación del resto de modelos) con un tiempo medio de 25 minutos. Los KNN nos dan un resultado muy similar al de las Redes y con un tiempo mucho menor, apenas unos segundos; con una pérdida de precisión casi inexistente.

Por estas razones, hemos elegido el modelo el KNN con 8 vecinos como el mejor. Este nos da una precisión media de 91.34% con una desviación de 0.00756, y un tiempo mínimo de apenas unos segundos de generación.

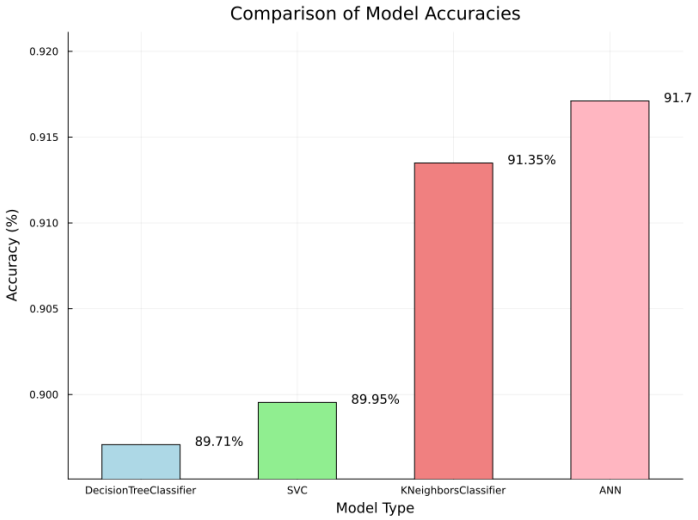

Figure 4: Comparativa de la precisión de los modelos.

5 Conclusión

Con el resurgimiento de la Inteligencia Artificial en los últimos años, las Redes Neuronales están en boca de todos. Herramientas como ChatGPT, Gemini o Copilot han hecho que se prefieran este tipo de sistemas siempre que se piense en utilizar I.A. Sin embargo, como observamos en los resultados obtenidos en este trabajo, modelos más simples pueden obtener resultados iguales o mejores según el problema al que se enfrente.

Los KNN en este caso son una mejor elección frente a las RR.NN.AA, y también al resto de modelos; ya que nos otorgan la mejor precisión en el mejor tiempo. Cabe destacar que este resultado es propio de nuestro trabajo, ya que otros estudios obtienen configuraciones en los que las SVM u otro tipo de modelos obtienen mejores resultados que los KNN.

6 Trabajo Futuro

Para mejorar nuestros modelos podríamos añadir más variedad de alubias a la base de datos. Esto implicaría la recopilación de datos, tomar imágenes de especímenes de más variedades de alubias y sus respectivas medidas. Posteriormente, se llevaría a cabo el reentrenamiento de nuestros modelos con estos datos adicionales.

Otra forma de avanzar el trabajo sería mejorar los hiperparamétros de los modelos. Añadir mejores topologías, optimizar las máquinas de soporte o incluso añadir otros modelos como XgBoost.

También podríamos explorar la posibilidad de agregar funciones más avanzadas, como la detección de enfermedades o anomalías en las alubias, la estimación de la calidad o el tamaño de las alubias. Estas aproximaciones nos permitirían mejorar nuestros modelos para poder usarlos en entornos más complejos, o mejorarlos para casos que ya conocemos.

7 Bibliografía

References

[1] Koklu, M., & Ozkan, I. A. (2020). Multiclass classification of dry beans using computer vision and Machine Learning techniques. Computers and Electronics in Agriculture, 174, 105507.

[2] Mendigoria, C. H., Concepcion, R., Dadios, E., Aquino, H., Alaias, O. J., Sybingco, E., ... & Cuello, J. (2021, September). Seed architectural phenes prediction and variety classification of dry beans (phaseolus vulgaris) using Machine Learning algorithms. In 2021 IEEE 9th region 10 humanitarian technology conference (R10-HTC) (pp. 01-06). IEEE.

[3] Taspinar, Y. S., Dogan, M., Cinar, I., Kursun, R., Ozkan, I. A., & Koklu, M. (2022). Computer vision classification of dry beans (Phaseolus vulgaris L.) based on deep transfer Learning techniques. European Food Research and Technology, 248(11), 2707-2725.

[4] Naik, N. K., Sethy, P. K., Amat, R., Behera, S. K., & Biswas, P. (2023). Evaluation of optimization techniques with support vector Machine for identification of dry beans. Indonesian Journal of Electrical Engineering and Computer Science, 32(2), 704-714.

[5] Sahu, P., Chug, A., Singh, A. P., Singh, D., & Singh, R. P. (2021). Deep Learning models for beans crop diseases: Classification and visualization techniques. International Journal of Modern Agriculture, 10(1), 796-812.

[6] Khan, M. S., Nath, T. D., Hossain, M. M., Mukherjee, A., Hasnath, H. B., Meem, T. M., & Khan, U. (2023). Comparison of multiclass classification techniques using dry bean dataset. International Journal of Cognitive Computing in Engineering, 4, 6-20.

[7] Hasan, M. M., Islam, M. U., & Sadeq, M. J. (2021, December). A deep neural network for multi-class dry beans classification. In 2021 24th international conference on computer and information technology (ICCIT) (pp. 1-5). IEEE.

[8] Santos, F. F. L. D., Rosas, J. T. F., Martins, R. N., Araújo, G. D. M., Viana, L. D. A., & Gonçalves, J. D. P. (2020). Quality assessment of coffee beans through computer vision and Machine Learning algorithms.