Instituto Tecnológico y de Estudios Superiores de Monterrey

Modelación del aprendizaje con inteligencia artificial

TC2034, Grupo 101

Docente:

Dra. María Valentina Narváez Terán

Situación problema:

Proyecto de aprendizaje supervisado y no supervisado

Presentado por:

Daniela Márquez Campos A00833345

13 de marzo de 2023

1. Introducción

Los modelos de Machine Learning (ML) permiten el aprendizaje automático a partir de datos, estos modelos se utilizan en distintas tareas, como es minería de datos, procesamiento de imágenes, análisis predictivo, entre otros. Como parte de este trabajo, se buscó implementar en dos etapas distintas, modelos de aprendizaje supervisado, en este caso, *Random Forest* y Regresión Logística, los cuales fueron de utilidad para realizar clasificaciones y detectar en el contexto del problema, si un tumor es maligno o benigno; así como modelos de aprendizaje no supervisado, siendo los seleccionados *K-means clustering* y *DBSCAN clustering*, los cuales permitirán agrupar los datos de la manera más óptima posible. A continuación, se presentan una breve explicación de la manera en que cada modelo trabaja:

El modelo Random Forest es considerado como un método muy efectivo de clasificación y regresión, que consiste en combinar una variedad aleatoria de árboles de decisión para así, evaluar distintas clasificaciones y realizar una predicción.

El modelo de regresión logística estima la probabilidad de que ocurra un evento en función de un conjunto de datos de variables independientes, propiciando el ajuste a una función logística, como es la función sigmoide, de la relación entre dichas variables independientes para predecir la dependiente.

El modelo *K-means* es un algoritmo que permite el agrupamiento de las observaciones en un número predefinido de *clusters* mediante la asignación de centroides aleatorios que van siendo refinados con base en el cálculo de la distancia media entre el mismo y las observaciones pertenecientes a cada cluster.

El modelo *DBSCAN clustering* consiste en agrupar las observaciones conforme a la densidad espacial de los mismos, es decir, asigna un registro a cierto *cluster* si éste se encuentra cercano a un número específico de registros pertenecientes a dicho *cluster*.

Como parte final, se evaluarán tanto los resultados como el desempeño de la implementación de estos modelos y se expondrán los hallazgos más relevantes.

2. Descripción de la problemática

Según un artículo de la Organización Mundial de la Salud, “cerca de una de cada 12 mujeres enfermarán de cáncer de mama a lo largo de su vida” (2021). Dicho cáncer es

considerado como la primera causa de mortalidad en las mujeres y el que más prevalece en el mundo. Las únicas mejoras que han habido en cuestión a evitar que el índice de mortalidad aumente es la temprana detección del tumor, es por esto que el reto principal es la clasificación de los tumores como malignos (cancerígenos) o benignos (no cancerígenos), por esto se eligió la dataset nombrada "Breast Cancer Dataset" para lograr encontrar un modelo de aprendizaje computacional que permita predecir si el tumor es benigno o maligno y de esta manera poder aportar a la detección temprana.

3. Descripción del dataset

El Breast Cancer Wisconsin Dataset cuenta con 32 variables, de las cuales hay 30 variables numéricas, 1 categórica y una llave primaria *"id"*. La variable categórica *"diagnosis"* fue seleccionada como la clase, esta indica si el diagnóstico del paciente es un tumor benigno (B) o maligno (M). Por su parte, las variables numéricas brindan información sobre las características que posee el tumor del paciente y aportan a la clasificación del tumor. Cada una de las variables se pueden ver en la siguiente imagen, dicha salida se obtuvo con *df.columns*.

```
['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
 'fractal_dimension_se', 'radius_worst', 'texture_worst',
 'perimeter_worst', 'area_worst', 'smoothness_worst',
 'compactness_worst', 'concavity_worst', 'concave points_worst',
 'symmetry_worst', 'fractal_dimension_worst'],
```

Para ambas etapas del proyecto se utilizó la misma base de datos.

3.1 Limpieza y normalización

De forma general, fue necesaria la ejecución de una exploración y limpieza del *dataset*, con el fin de poder tener una base de datos manejable, sin datos nulos, y corregir errores en las variables de entrada. Se comenzó visualizando las variables con *df.head()* y *df.tail()* posteriormente a esto se utilizó el método *.isnull().sum()* de la librería pandas para contar los valores nulos del *dataframe*, y se observó que no contaba con ninguno, lo cual facilitó el manejo de esta misma.

Por último, se aplicó un proceso de normalización para el preparamiento de los datos utilizando la función *prep = preprocessing.normalize(x, axis=0)*, siendo que de esta manera los algoritmos elegidos logran modelar los datos con mayor efectividad, sin subestimar o sobreestimar variables debido a la magnitud de sus valores, siendo que están en la misma escala.

4. Métodos de aprendizaje supervisado

4.1 Preprocesamiento

Para la correcta aplicación de los modelos supervisados elegidos y con el fin de tener una variable *dummy* que solo tome dos valores para la variable *target*, se modificó el contenido de la variable "*diagnosis*", ya que esta variable contiene M en caso de que el tumor sea maligno o B en caso de ser benigno, sin embargo, se cambió a valores binarios: 0, representando que el tumor es benigno y 1 que es maligno.

También se realizó una segmentación de los datos, dividiéndolos en la variable dependiente o *target*, almacenado en la variable de nombre *y,* y las independientes que son todas menos *diagnosis* y *id*, contenidos en la variable llamada *x*.

4.2 Random Forest

Para poder obtener el mejor modelo de clasificación, con el fin de cumplir con el objetivo principal del proyecto, se eligieron dos métodos. El primer método de machine learning utilizado fue el Random Forest, este algoritmo combina múltiples árboles de decisión para así unificarlos en un solo resultado; cada uno de estos árboles se encarga de analizar diversas partes de los datos de entrenamiento, además los Random Forest, al estar compuestos de muchos árboles de decisión, cuentan con diversas preguntas en los nodos de las hojas para poder verificar si se debería continuar o no dependiendo lo que se esté buscando; en conjunto, cada una de estas pequeñas preguntas generará una decisión final, que en este caso será clasificar si la persona tiene un tumor maligno o en otro caso, benigno.

Es importante mencionar que este algoritmo tiene una agrupación de características que genera un subconjunto aleatorio de las características, lo cual hace que se tenga una correlación baja entre los árboles de decisión y de esta manera se puede reducir el riesgo al sesgo, obteniendo predicciones más precisas. Por la misma razón de que los Random Forest obtienen predicciones y modelos de clasificación o

regresión muy precisos (overfitting), no tienen una muy buena generalización, sin embargo, al utilizar varios árboles de decisión, se sabe que los modelos de clasificación elegirán los mejores resultados de los árboles y de los que se esté más seguro de ser el correcto.

4.3 Regresión logística

El otro modelo que se utilizó fue el de regresión logística; estos modelos tienen gran aplicación para predecir la presencia o ausencia de una característica en específico, como lo es en este caso, un tumor maligno o benigno, por esto mismo, los modelos de regresión logística se enfocan en las variables dependientes dicotómicas, es decir, que solo pueden tomar dos valores diferentes.

Con este algoritmo se intenta encontrar la relación entre dos factores de datos y con esa relación se intenta predecir el valor de un factor en función del otro. Además, estos modelos son considerados como simples, veloces, flexibles y con fácil visibilidad. En el caso del proyecto, se tienen diversas variables independientes que afectarán el resultado de la variable clase (si se tiene o no cáncer), es por esto que el modelo generado se asume una relación lineal entre las variables independientes, teniendo que:

$$y = f(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_n x_n)$$

en donde β es el coeficiente de regresión y x son las variables independientes.

Esta ecuación es ajustada a una función logística para que finalmente, al ser evaluada, se obtenga un resultado igualmente dicotómico, tal como lo es la variable dependiente.

4.4 Resultados obtenidos

A partir de la aplicación de ambos métodos, Random Forest y Regresión Logística, a la base de datos, con el fin de encontrar un modelo óptimo de clasificación entre los tumores malignos y benignos, así como para poder identificar de manera correcta si el paciente tiene o no cáncer de mama, se obtuvieron los siguientes resultados.

Con el modelo de Random Forest se hizo una división de datos, de los cuales 70 % de ellos se destinaron para entrenamiento y 30% se destinaron para prueba. A partir de esta división se hizo un ensamblaje de árboles de decisión, es decir un bosque aleatorio con la métrica Gini, utilizando 10 árboles; posteriormente, fue posible entrenar al modelo. Una vez realizado este proceso, se calculó el *"accuracy"* promedio, donde se obtuvo un resultado de 0.959.

```
# Accuracy promedio
BA_model.score(X_test, y_test)

0.9590643274853801
```

Esto indica que el modelo realiza una adecuada clasificación de los datos y es eficaz para predecir si el tumor de la persona es maligno o benigno. Posteriormente, se realiza la predicción del modelo, lo cual permite generar una matriz de confusión.

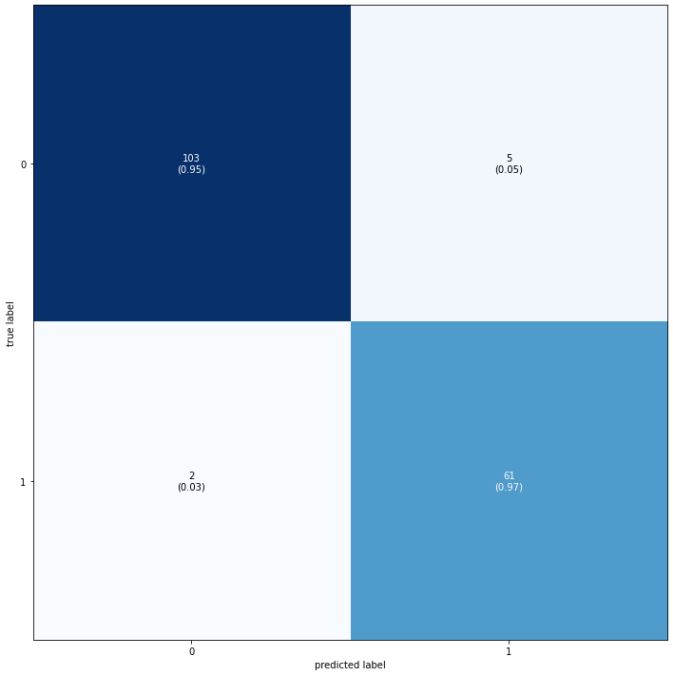

Figura 1: Matriz de confusión Random Forest

En dicha matriz se observa el comportamiento del modelo y de tal forma evaluar su precisión de forma porcentual. En el primer cuadrante se observa que el modelo acertó un 95% de las veces para clasificar los verdaderos negativos; es decir la mayor parte de las veces acertó en clasificar a las personas que presentaban tumores benignos; de la misma manera se puede observar en el cuarto cuadrante que el 97% de

las veces clasificó correctamente los verdaderos positivos; es decir, la clasificación fue adecuada para determinar aquellas personas que poseían tumores malignos. Por su parte, los cuadrantes restantes (falso positivo y falso negativo) tienen porcentajes muy bajos, lo cual indica que fueron pocas veces en las que el modelo tuvo clasificaciones erróneas

Finalmente, se crearon y visualizaron los árboles del bosque, mediante un ciclo for; donde se decidió crear 10 árboles, de tal forma que se pudiera observar el comportamiento del modelo en distintos escenarios; haciendo uso de clasificación y regresión para su uso en el modelado predictivo de atributos discretos y continuos.

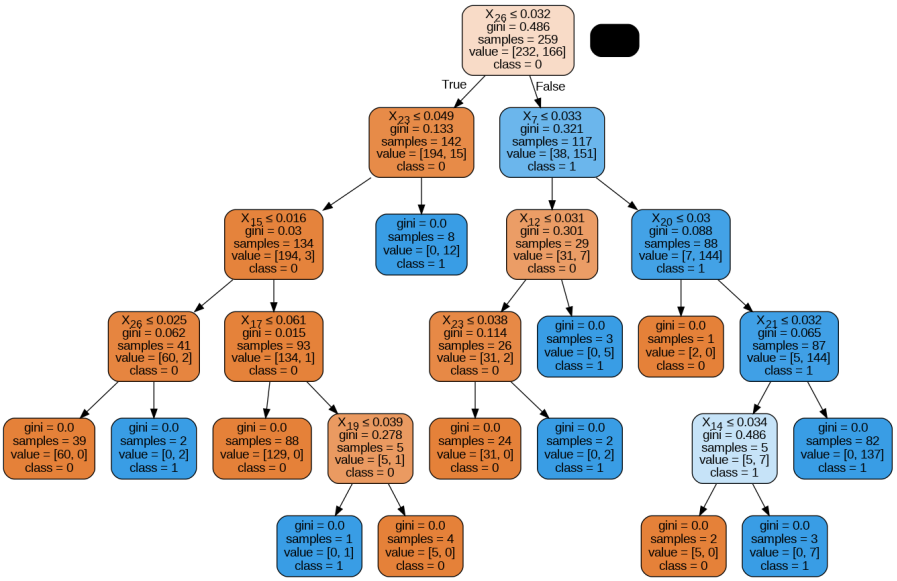

Figura 2: Visualización del árbol de decisión

Dicho árbol es uno de los que fueron creados, el cual permite ver la jerarquización de las variables mediante el cálculo del gini y posteriormente las va clasificando; de la misma manera sucede con cada árbol.

Con el segundo algoritmo de Regresión Logística se utilizó el Recursive Feature Elimination, para el cual, el modelo se entrenó con todas las variables del data frame, pero con el objetivo de identificar las variables independientes más representativas para el modelo de predicción, únicamente se le pidió que utilizara 10 variables. Con esto se identificaron las variables más significativas sin contar a la variable target ni a la variable *id.* Las cuales fueron las siguientes:

```
Selected: area_mean Rank: 1
Selected: concavity_mean Rank: 1
Selected: concave points_mean Rank: 1
Selected: radius_se Rank: 1
Selected: perimeter_se Rank: 1
Selected: area_se Rank: 1
Selected: area_worst Rank: 1
Selected: compactness_worst Rank: 1
Selected: concavity_worst Rank: 1
Selected: concave points_worst Rank: 1
```

Una vez teniendo las variables más significativas filtradas, se aplicó el modelo de Regresión Logística a estas mismas, tomando como variable target *diagnosis*, para lo cual la salida muestra un p-value de 2.1986e-124. A partir de esto se utilizó la regla de decisión en donde se eliminaron las variables que tenían un valor mayor a alfa = 0.05, es decir: si $p - value < 0.05$ entonces la variable es significativa. Con esto únicamente quedaron las variables *'area_mean', 'concave points_mean', 'radius_se', 'area_se', 'area_worst'*. Con esto se seleccionaron los datos de entrenamiento (70%) y los de prueba (30%), con los cuales se entrenó el modelo y se obtuvo como resultado final la siguiente matriz de confusión:

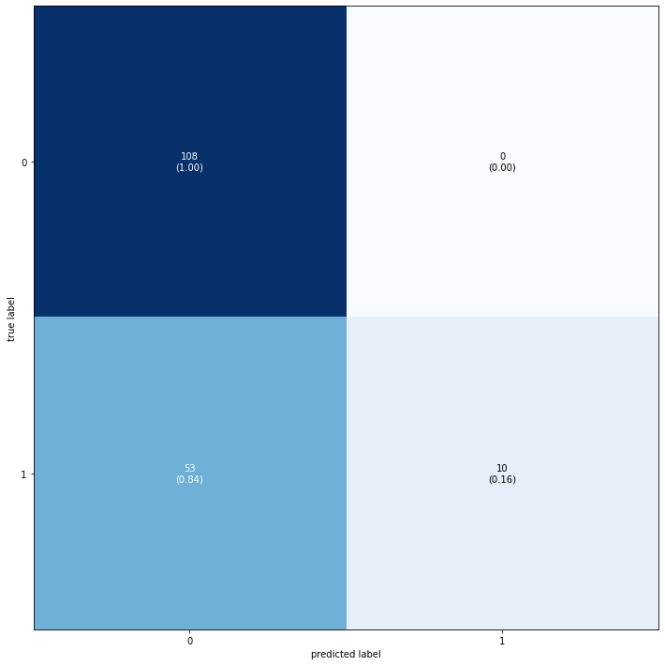

Figura 3: Matriz de confusión Regresión Logística

La matriz de confusión permite identificar el porcentaje de predicción explicado por el modelo, en este caso, de clasificación de tumores. En el cuadrante de color azul fuerte se observa que el modelo acertó el 100% de las veces al clasificar los

verdaderos negativos, es decir logró clasificar bien a los que no tienen cáncer, sin embargo, como se observa en el cuadrante inferior derecho, solamente el 16% de las veces clasificó los datos correctamente como verdaderos positivos, los que si tenían cáncer y aunque el 0% de las veces los datos fueron clasificados como falsos positivos, el 84% se clasificaron incorrectamente como falsos negativos, lo cual es un porcentaje muy alto de incorrectos para el modelo, puesto que quiere decir que se dio un diagnóstico incorrecto de los que no tenían cáncer. Por último, es importante mencionar que el nivel de accuracy promedio que tiene este modelo es de solamente 0.69.

4.5 Hallazgos de los modelos supervisados

Se concluye que el modelo de random forest dió mejores resultados al momento de realizar una clasificación de datos y realizar las debidas predicciones. Esto se puede observar tanto en los resultados de *"accuracy"* como en las matrices de confusión, puesto que random forest arrojaba mejores resultados a diferencia de la regresión logística. Sin embargo, también es importante tomar en cuenta que los modelos de regresión suelen ser afectados cuando existe una fuerte correlación entre las variables (tal es el caso de la base de datos analizada); por su parte el random forest no presenta deficiencias ante estos modelos. Cabe recalcar que el uso de regresión logística para la selección de características se debe utilizar cuando los factores independientes sean de gran cantidad; ya que se observa que cuando la cantidad de factores es mayor, random forest arroja mejores resultados.

Por último es importante recalcar las limitaciones de ambos métodos. Las predicciones que hace random forest no son de naturaleza continua, es decir no puede predecir más allá del rango de los valores del conjunto de entrenamiento, por lo cual este se ajusta a los datos que la persona le da, sin embargo, si esta distribución cambia suele presentar dificultades. Por su parte la regresión logística solo nos permite trabajar con datos linealmente separables, además que si no cuenta con muchos datos, el modelo puede carecer de precisión.

Como área de mejora, se propone el over-sampling, que consiste en balancear y distribuir adecuadamente la proporción de las clases en los datos de entrenamiento. Es decir del 70% de los datos que se toman para entrenamiento; distribuirlos

adecuadamente; en este caso que el 50% sea para tumores malignos y el otro 50% para tumores benignos. Dicho método podría aumentar la precisión de ambos métodos y contribuir a la mejora de resultados.

5. Modelos de aprendizaje no supervisado

5.1 Preprocesamiento

En este caso, fue necesario eliminar la variable target o label, pues no fue requerida para los métodos de clusterización que fueron aplicados y evitar generar un modelo de clasificación.

Por otro lado, al tener un gran número de variables características, se aplicó el método de Análisis de Componentes Principales, *PCA* por sus siglas en inglés, ya que permite la reducción de dimensionalidad del *set* de datos, en el cual se tenían treinta columnas y con esta implementación se realizó una combinación lineal de las variables originales para obtener únicamente dos columnas, *PC1* y *PC2* que representan de la mejor forma la variabilidad de los datos.

	PC1	PC2
0	0.199095	-0.016644
1	0.036075	-0.062749
2	0.114292	-0.023569
3	0.120728	0.115834
4	0.093005	-0.027815

Figura 4. Visualización de las variables generadas con el método PCA

Se incorporó esta transformación de las dimensiones del *dataset* con el fin de facilitar la aplicación y visualización de los modelos, pues comúnmente es una tarea más compleja realizar *clustering* y examinarlo gráficamente sobre sets de datos con alto número de dimensiones.

5.2 *K-means*

Previamente, se introdujo que como técnicas de agrupamiento se seleccionaron dos, de los cuales la primera es el método de K-means. Éste consiste en la asignación de *k* centroides aleatorios, los cuales, mediante iteraciones, sufren

cambios hasta encontrar aquellos puntos óptimos con los cuales la suma de los cuadrados de la distancia entre él mismo y las observaciones de su *cluster* sea mínima, ya que es regido por un criterio en el que la similitud es determinada por la distancia y en donde se denota que "la diversidad reducida en los *clusters* conduce a la obtención de datos más similares o idénticos en el mismo *cluster*" (Sharma, 2021).

Se le conoce como un método de los más antiguos y sencillos de aplicar, pues solo necesita un hiperparámetro, es decir el valor de *k*, que se señala la cantidad de centroides y por tanto de *clusters* que el algoritmo computará. Para obtenerlo, se aplican diferentes métodos entre ellos el Método del Codo y el Coeficiente de Siluetas, los cuales generan una curva en la que, de acuerdo a los puntos de inflexión observados, se identifican los valores óptimos para *k*. En el caso del primer método, se compara el valor del *Sum of Square Error (SSE)* contra el número de conglomerados; cabe destacar que simplemente no se escoge el valor que conlleve menor valor de *SSE*, pues éste se va reduciendo conforme el número de *k* aumenta y si se lleva demasiado lejos, un gran número de *clusters* puede igualar *k* al número de muestras, lo cual es ineficiente. En el caso del segundo método, se compara el valor del Coeficiente de Siluetas contra el número de conglomerados y se busca el valor más cercano a 1 del mencionado coeficiente, ya que al ser una métrica de evaluación del desempeño de modelos de *clustering*, un valor de 1 o cercano al mismo, indica que los *clusters* poseen datos similares dentro y que se diferencían de agrupaciones externas.

5.3 *DBSCAN*

El segundo método aplicado es el *Density-based spatial clustering of applications with noise*, o por sus siglas en inglés *DBSCAN*, el cual, como su nombre lo indica es un algoritmo de agrupamiento que identifica aquellas áreas con mayor densidad de observaciones y las separa de aquellas áreas con menor densidad de datos o que se encuentran vacías. Asimismo, aquellos registros más dispersos que no sean asignados a un *cluster* los etiqueta como ruido. Según el algoritmo encuentre y asigne los datos a los conglomerados, éstos van tomando la forma y el tamaño concorde a la densidad.

De los modelos de *clustering* basados en densidad, en este proyecto se utilizó el más rápido de todos, en el cual se define un rango de distancia de búsqueda y se asume que todos los *clusters* potenciales o significativos poseen una misma densidad.

Aunado a esto, el modelo recibe dos hiperparámetros de suma importancia: epsilón y el mínimo de muestras. El primero es aquel rango de distancia de búsqueda que se utilizará como umbral para determinar la cantidad de vecinos más cercanos a la observación analizada; éste se puede identificar a partir de varios métodos, dentro de los cuales destaca el Método de *Nearest Neighbors*, en el cual se calculan el promedio de las distancias de cada registro a su *k* número de vecinos más cercanos, para seguidamente aplicar el Método del Codo comparando dicho promedio de distancias por punto, para finalmente seleccionar el punto de inflexión como distancia representativa. El segundo hiperparámetro mencionado es la cantidad de observaciones vecinas dentro del umbral determinado por epsilón para considerar a un registro como punto central. Se asigna de manera arbitraria y se puede ir ajustando conforme al desempeño del modelo.

5.4 Resultados obtenidos

A partir de la aplicación de ambos métodos, *K-means* y *DBSCAN*, a la base de datos, con el fin de encontrar un modelo óptimo de agrupamiento para las observaciones presentadas en el *set*, se obtuvieron los siguientes resultados.

Para la aplicación del método de *K-means*, primero fue necesario determinar el valor de *k*. Por tanto, se desarrolló el Método del Codo, en el cual se obtuvo el siguiente gráfico.

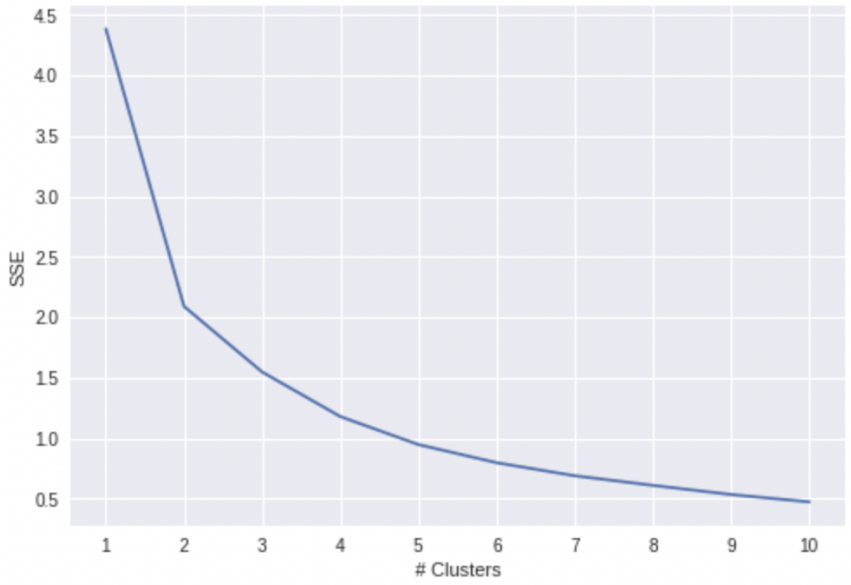

Figura 5. Método del Codo ejecutado para determinar k

Como se podrá observar, hay un claro punto de inflexión en el valor 2, por lo cual, se seleccionó dicha cantidad de *clusters* a realizar para el método. Posteriormente, se aplicó el modelo de clusterización a los datos generados por el PCA, para finalmente obtener una columna nueva con los *clusters* respectivos a cada registro, la cual fue concatenada a los datos originales para así finalmente visualizarlos.

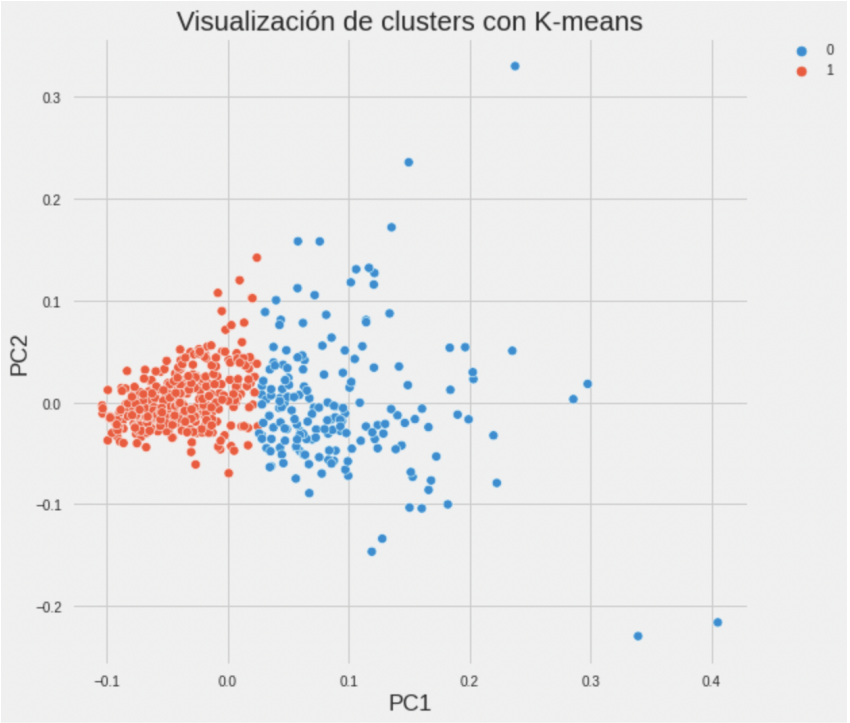

Figura 6. Gráfico de clusters generados con K-means

Finalmente, se utilizó la métrica del Coeficiente de Siluetas para evaluar el desempeño del modelo. Con lo cual se obtuvo un coeficiente promedio de 0.545.

```
[ ]  silhouette_avg

     0.5458569392619401
```

Como la tendencia del valor es hacia 1, indica que los conglomerados generados por el algoritmo son similares por dentro y diferenciables entre ellos.

Para el segundo modelo utilizado, también fue necesario determinar primero el valor de epsilón a través del método de *Nearest Neighbors*, su visualización y selección con el Método del Codo. Del proceso mencionado anteriormente se obtuvo la siguiente gráfica:

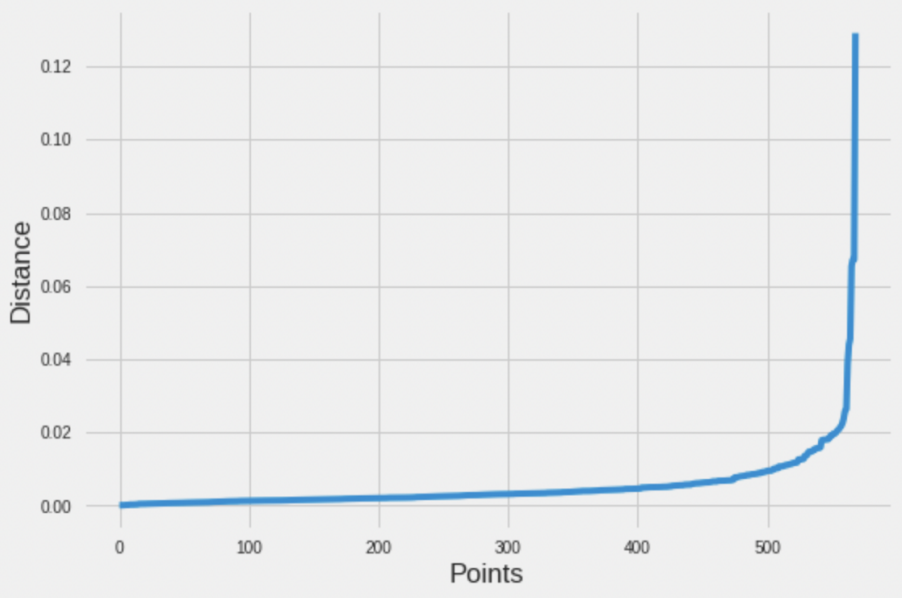

Figura 7. Método del Codo ejecutado para determinar epsilón

Es posible observar que hay un ligero punto de inflexión aproximadamente en el valor 0.02, el cual fue definido como el rango de distancia de búsqueda para el algoritmo. Posteriormente, se aplicó el modelo de clusterización a los datos generados por el PCA, para de la misma manera que en el modelo anterior, obtener una columna nueva con los *clusters* respectivos a cada registro, la cual fue concatenada a los datos originales para así finalmente visualizarlos.

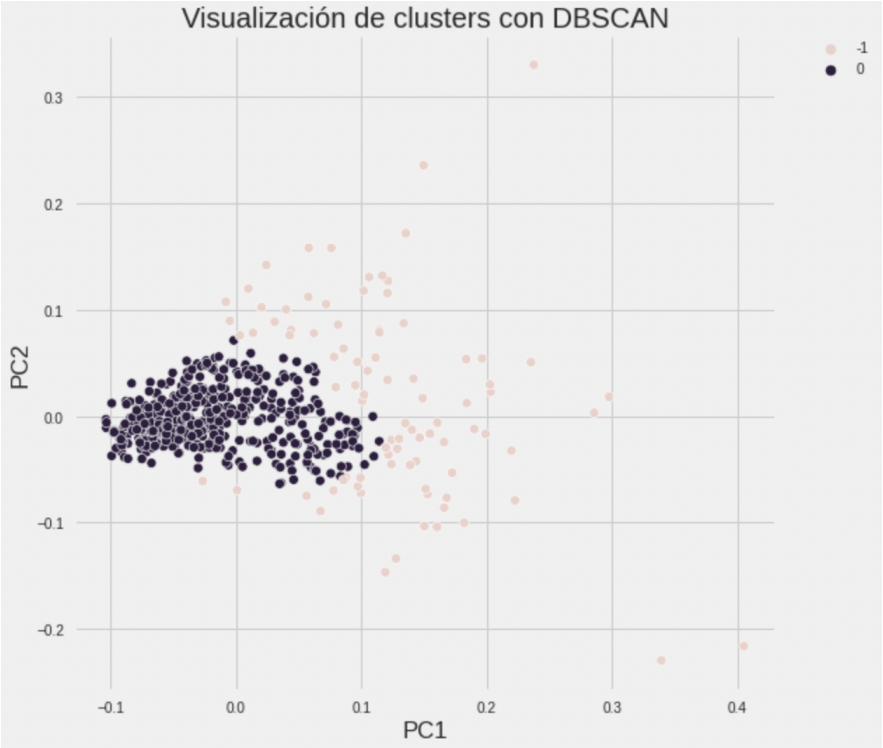

Figura 6. Gráfico de clusters generados con DBSCAN

Finalmente, se usó la métrica del Coeficiente de Siluetas para evaluar el desempeño del modelo. Con lo cual se obtuvo un coeficiente promedio de 0.521.

```
[ ]  print(silhouette_avg2)

0.5209354344463358
```

Como la tendencia del valor también es hacia 1, indica que los *clusters* generados por este segundo algoritmo, igual son similares por dentro y diferenciables entre ellos.

5.5 Hallazgos de los modelos no supervisados

Se puede concluir que el modelo *K-means* presentó mejores resultados en el proceso de agrupación por conglomerados de los datos generados. Esto es ligeramente evidente al observar los resultados de la métrica de evaluación elegida, en este caso, los coeficientes de silueta, puesto que *K-means* concluyó con ligeramente mejores resultados que *DBSCAN*. Esto lleva a considerar las diferencias entre los modelos como los factores que propiciaron que uno tuviera una mejor *performance* que el otro. En ello destacamos que *K-means* no se basa en la densidad de los datos como *DBSCAN*, sino es la distancia euclidiana; así mismo, tal como se aprecia en la Figura

6, *K-means* presenta una tendencia a agrupar los datos de forma esférica y particional, mientras que *DBSCAN* los clusteriza en tamaño y forma con respecto puramente a la densidad, como se puede ver en la Figura 7. Finalmente, también se realiza énfasis en que para *DBSCAN* no es necesario señalar el número de *clusters* a realizar y asume que misma densidad para ellos, contrastando con *K-means* esto no aplica.

Además, también es importante tomar en cuenta que aplicar el Análisis de Componentes Principales (*PCA*) conlleva cierta pérdida de variabilidad e información en los datos, pues hubo una transformación radical de una dimensión de treinta *features* a tan solo dos, es una posible afección al desempeño de los modelos.

Por último, es importante recalcar las limitaciones de ambos métodos, dentro de las cuales destacan la tendencia en la forma esférica para los clusters en *K-means*, así como la generalización de densidad en *DBSCAN*. También es claro que la determinación de los hiperparámetros para cada modelo complica su aplicación y es de vital importancia su correcta ejecución, ya que ambos modelos poseen un nivel alto de sensibilidad con respecto a ellos.

Como área de mejora, en el caso de *DBSCAN,* se puede considerar variaciones en el modelo, ya que existen otros algoritmos de la misma rama con mayor flexibilidad como el *OPTICS*, o con autoajuste a los datos como el *HDBSCAN*. También, de forma general, se puede considerar la implementación de otros métodos más precisos y eficientes para la reducción de dimensionalidad de la base de datos

6. Conclusiones

Así se concluye que, de forma general, los modelos de aprendizaje supervisado tuvieron mucho mejor desempeño que los modelos de aprendizaje no supervisado. La mayor eficiencia de los primeros mencionados puede deberse a que la presencia de etiquetas de clase en el set de datos propició mucho más la aplicación de los modelos de clasificación.

Asimismo, para ejecutar los modelos de *clustering*, se mencionó que fue necesario aplicar el Análisis de Componentes Principales, provocando la reducción de dimensionalidad

del dataset, lo cual conllevó una ligera pérdida de información y variabilidad, siendo sugerido que ello pudo afectar su *performance*.

El modelo de mejor desempeño fue el *Random Forest*, con un *accuracy* de casi 95 %, asegurando que se debe a la forma natural en que su ensamble de árboles de decisión es inmune a la alta correlación en variables independientes, reduce el sesgo presente en los datos y por tanto, el error, así como maneja de buena manera los *outliers*; características contrastantes en el modelo de regresión logística aplicado, pues es claramente afectado por la presencia de variables altamente correlacionadas y su aplicación sobre pocas observaciones es ineficiente. De la misma forma, se identificó que las desventajas más marcadas en los modelos de *clustering* son la gran sensibilidad con respecto a sus hiperparámetros, así como la dificultad para generar conglomerados irregulares en *K-means* y la generalización de la densidad de los grupos en *DBSCAN*.

Finalmente, este proyecto muestra claramente la importancia de los modelos de inteligencia artificial y su aplicación en áreas como la salud. En primera instancia, la salud humana es un tema delicado, sin embargo, gracias al desarrollo, el buen entrenamiento y la mejora de diversos modelos como los de Machine Learning, utilizando bases de datos eficientes y estandarizadas, se pueden obtener resultados fascinantes, precisos y significativos que pueden ser de vital ayuda para la detección de enfermedades de forma temprana, lo que permite tratarlas lo más pronto posible e, incluso, permite salvar vidas.

Referencias

Cáncer de mama. (2021, marzo 26). https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer

¿Qué es la regresión logística? - Explicación del modelo de regresión logística - AWS. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/logistic-regression/

¿Qué es el Random Forest? | IBM. (s. f.). https://www.ibm.com/mx-es/topics/random-forest

Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762_Machine_Learning_Algorithms_-A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf?eid=5082902844932096

Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25, 197-227. https://0-link-springer-com.biblioteca-ils.tec.mx/content/pdf/10.1007/s11749-016-0481-7.pdf

Sharma, P. (2021, 24 noviembre). Understanding K-means Clustering in Machine Learning(With Examples). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/

González, L. (2021, 8 septiembre). DBSCAN Teoría. Aprende IA. https://aprendeia.com/dbscan-teoria/

ArcGIS Pro. (s. f.). *Cómo funciona el clustering basado en densidad*. esri. https://pro.arcgis.com/es/pro-app/latest/tool-reference/spatial-statistics/how-density-based-clustering-works.htm