

Tarea 7 y 8. Metodología y métricas de error

Salinas, Iván

29 de junio de 2024

1. Introducción

El Aprendizaje Automatizado (Machine Learning; ML) es una rama de la inteligencia artificial, en gran parte inspirada en el razonamiento humano, que comprende el aprendizaje a partir de experiencia. El aprendizaje automático aborda, a su vez, una serie de problemáticas que tributan a problemas específicos, entre ellos: los problemas de clasificación, asociación, agrupamiento, y selección de rasgos. En el agrupamiento se parte de un conjunto de ejemplos el cual se desea organizar en grupos usualmente de acuerdo a una noción de similitud que generalmente es determinada por una función o métrica de distancia. La proximidad entre ejemplos determina la pertenencia o no a un grupo; por tanto, se estima que un elemento será más similar o tendrá mayores propiedades en común con los elementos de su grupo, que con respecto a los elementos de un grupo diferente [1].

Muchos de los métodos de aprendizaje automático dependen del cálculo de distancias para estimar la similitud entre dos ejemplos teniendo en cuenta la estructura de los datos. Este es el caso, por ejemplo, del algoritmo de los k vecinos más cercanos (k-Nearest Neighbor; kNN) para la comparación de las instancias entrantes con los datos conocidos (ejemplos de entrenamiento) o el k -medias (k-means) para calcular la distancia entre los objetos y su centro más cercano [1].

Muchos algoritmos no supervisados para la reducción de dimensionalidad realizan un aprendizaje no supervisado de métricas de distancias utilizando información de los propios datos o de la dimensión donde se encuentran representados. Este grupo de métodos se pueden clasificar en métodos no lineales y lineales. Los algoritmos de reducción no lineales consideran que cada uno de los datos de alta dimensionalidad puede ser descrito a través de una función compuesta por los parámetros más relevantes y los datos son vistos como extractos de una dimensión subyacente embebida en la dimensión original del espacio. El objetivo es embeber datos que originalmente se encuentran en una dimensión en otra dimensión reducida, al mismo tiempo que se preservan las características principales de los datos. Para cada espacio dimensional debe existir intrínsecamente un espacio reducido; y por tanto, es posible acceder a los datos reducidos a través de algoritmos que interpreten o preserven la naturaleza de los datos embebidos. Entre los métodos más utilizados de este tipo se encuentran ISOMAP,



GIA	Scan. D.N	CIBJO- IDC/HRD
 Colorless (Incoloro)	D	Blanco Excepcional+
	E	Blanco Excepcional
	F	Blanco Extra +
	G	Blanco Extra
 Near colorless (casi Incoloro)	H	Blanco
	I	Blanco Ligero
	J	Color

Figura 1: Clasificación de color.

el cual busca un sub-espacio que preserve mejor las distancias geodésicas entre dos puntos de datos y los métodos LLE y LE, que se enfocan en la preservación de las estructuras de las vecindades locales [1].

1.1. Objetivos

En este reporte se desea usar aprendizaje no supervisado spectral clustering partiendo desde lo visto en clase con k-medias para clasificar los datos que se han estado trabajando a lo largo del curso, además se añadirán los temas previamente trabajados como el procesamiento de los datos, su origen, la descripción de estos mismos y se llegara a una conclusión una vez obtenidos los resultados del análisis.

2. Descripción de los datos

2.1. Origen de los datos

Justificación y explicación de datos El conjunto de datos con el que se trabajara durante el curso sera un dataset con datos que se han juntado de diamantes y sus diferentes características, las cuales son:

Price: En el caso del precio de los diamantes, puede variar de forma continua, desde valores muy bajos hasta valores muy altos, sin límites específicos y con una infinidad de posibilidades entre ellos.

Carat: Son los quilates, el cual representa a 1 quilate = 0.2 gramos

Cut: Es la calidad de corte en el diamante que son Fair, Good, Very Good, Premium e Ideal. Se piensa que es importante por la importancia que puede tener un buen trabajo realizado por el artesano que corto el diamante.

Color: El color se clasifica desde la J a la D, lo cual nos indica que tan incoloro es el diamante, conforme más bajo sea su escala tiene un color cercano al amarillo. Esto se considera importante para el precio ya que nos indica la pureza del diamante 1 .

Clarity: Es una medida de qué tan claro es el diamante (I1 (peor), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (mejor)). Se piensa que es importante ya que los diamantes al ser creados bajo altas temperaturas y presiones pueden llegar a

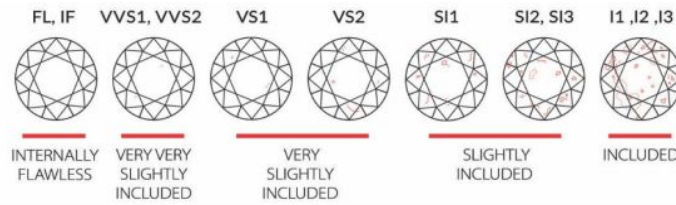


Figura 2: Clasificación de pureza.

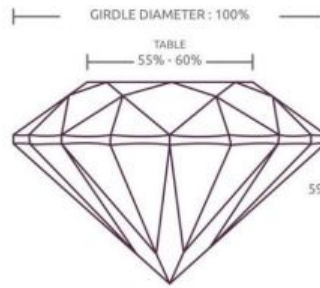


Figura 3: Proporción de table.

tener burbujas o imperfecciones dentro y eso se piensa que afecta el precio de este mismo (2).

Table: Ancho de la parte superior del diamante en relación con el punto más ancho (43-95). Se piensa que es importante en relación con el precio ya que es lo que le da la forma característica de diamante una buena proporción de ancho de mesa (3).

Luego tenemos las variables x , y y z las cuales representan el tamaño de las dimensiones del diamante.

x longitud en mm (0-10.74)

y ancho en mm (0-58.9)

z profundidad en mm (0-31.8)

Estas variables son importantes porque nos describen las dimensiones del diamante mismo, por lo que un diamante con mayores dimensiones podría ser el que tenga mayor precio.

2.2. Preprocesamiento

Para tener una vista general de los datos tomamos una muestra de 5 observaciones aleatorias como se muestra en la tabla 1.

Ahora visualizamos las principales medidas descriptivas de nuestros datos con valores numéricos, con esto podemos notar que tenemos una muestra de 53,940 filas/observaciones, por otro lado, las variables x y z tienen valores de 0 lo cual nos da una idea de que hay datos los cuales hay que limpiar, ya que no

Cuadro 1: Muestra aleatoria de los datos.

	carat	cut	color	clarity	depth	table	price	x	y	z
4339	1.50	Premium	H	I1	61.1	59.0	3599	7.37	7.26	4.47
15117	1.01	Premium	D	SI1	61.8	58.0	6075	6.42	6.37	3.95
25101	1.50	Very Good	D	VS2	63.8	55.0	13629	7.24	7.28	4.63
38478	0.42	Ideal	G	VVS2	62.1	57.0	1031	4.77	4.80	2.97
20351	1.35	Premium	H	VS1	60.5	60.0	8747	7.19	7.16	4.34

existen diamantes bidimensionales o de una dimensión (5).

Cuadro 2: Resumen estadístico de los datos

	carat	depth	table	price	x	y	z
total	53940	53940	53940	53940	53940	53940	53940
media	0.7979	61.7494	57.457184	3932.799722	5.731157	5.7345	3.5387
std	0.4740	1.4326	2.234491	3989.439738	1.121761	1.1421	0.7056
min	0.2000	43	43	326	0	0	0
25 %	0.4000	61	56	950	4.7100	4.720000	2.9100
50 %	0.7000	61.8	57	2401	5.7000	5.710000	3.5300
75 %	1.0400	62.5	59	5324.25	6.5400	6.540000	4.0400
max	5.0100	79	95	18823	10.74	58.9	31.8

Se buscan cuantos datos no son nulos, de esta forma podemos notar que no hay datos nulos en los datos, por lo que ahora procederemos con buscar la cantidad de datos con valor 0 y los borramos. Como resultado de estas acciones terminamos borrando 35 filas que contenían 7 ceros en la variable y, 8 ceros en la variable x y 20 ceros en la variable z.

Cuadro 3: Cantidad de 0 por variable

Variable	Cantidad de 0
carat	0
cut	0
color	0
clarity	0
depth	0
table	0
price	0
x	8
y	7
z	20

Vamos a quitar los valores anormales que excedan nuestros máximos y mínimos, para hacer esto primero hacemos una copia de nuestro dataframe.

Ahora hacemos una lista con nuestras variables categóricas para poder cambiar sus valores a números enteros, de esta forma sera mas fácil trabajar con

Cuadro 4: Recuento de datos no nulos

Index	# Column	Non-Null Count	Dtype
0	53940	non-null carat	float64
1	53940	non-null cut	object
2	53940	non-null color	object
3	53940	non-null clarity	object
4	53940	non-null depth %	float64
5	53940	non-null table %	float64
6	53940	non-null price	int64
7	53940	non-null x (length)	float64
8	53940	non-null y (width)	float64
9	53940	non-null z (depth)	float64

los datos en un futuro, todos los valores serán desde la categoría peor calificada con valor 0 hasta la mejor que tendrá el valor numérico mas alto. Con esto terminamos la limpieza de datos y tenemos un dataframe de dimensiones (53907, 10).

Cuadro 5: Datos de diamantes con valores enteros en categorías.

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	2	1	3	61.5	55.0	326	3.95	3.98	2.43
2	0.21	3	1	2	59.8	61.0	326	3.89	3.84	2.31
3	0.23	1	1	4	56.9	65.0	327	4.05	4.07	2.31
4	0.29	3	5	5	62.4	58.0	334	4.20	4.23	2.63
5	0.31	1	6	3	63.3	58.0	335	4.34	4.35	2.75

2.3. Estadística descriptiva

Comenzaremos vizualisando las medidas descriptivas de nuestros datos que ya han sido limpiados en la seccion anterior.

Cuadro 6: Estadísticas descriptivas de los datos limpios.

	carat	cut	color	clarity	depth	table	price	x	y	z
total	53907	53907	53907	53907	53907	53907	53907	53907	53907	53907
media	0.8	2.6	2.6	3.8	61.7	57.5	3930.6	5.7	5.7	3.5
std	0.5	1.0	1.7	1.7	1.4	2.2	3987.2	1.1	1.1	0.7
min	0.2	0.0	0.0	0.0	50.8	43.0	326.0	3.7	3.7	2.1
25 %	0.4	2.0	1.0	2.0	61.0	56.0	949.0	4.7	4.7	2.9
50 %	0.7	2.0	3.0	4.0	61.8	57.0	2401.0	5.7	5.7	3.5
75 %	1.0	3.0	4.0	5.0	62.5	59.0	5322.0	6.5	6.5	4.0
max	5.0	4.0	6.0	7.0	73.6	79.0	18823.0	10.7	10.5	7.0

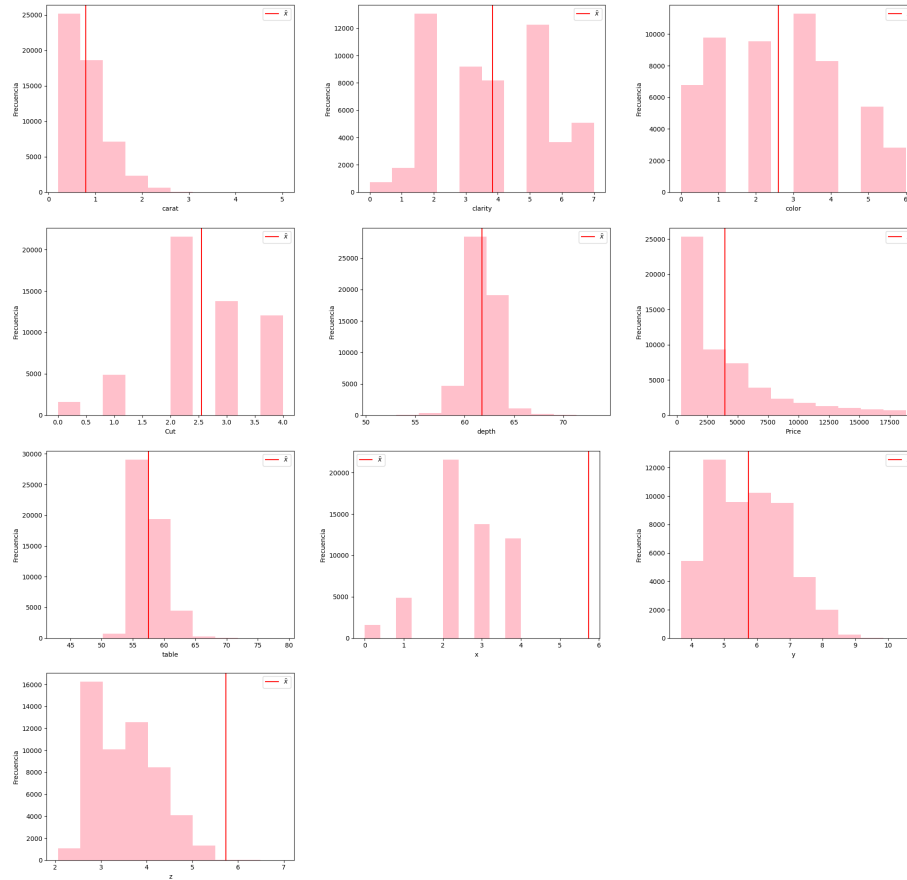


Figura 4: Histogramas

Para una correcta visualización de los datos graficamos los histogramas de nuestras 10 variables y hacemos sus diagramas de dispersión y de violín.

3. Antecedentes

El clustering o agrupamiento es el proceso de particionar un conjunto de datos (u objetos) en un conjunto de subclases significativas llamadas grupos (clusters). Un grupo es una colección de objetos de datos que son similares a otros y así pueden ser tratados colectivamente como un grupo. El agrupamiento es una forma de clasificación no supervisada en la que, a diferencia de la supervisada, no se conocen las etiquetas de las clases (no hay clases predefinidas) y puede que tampoco se conozca el número de grupos. Un buen método de agrupamiento produce grupos de alta calidad en los cuales la similitud dentro del

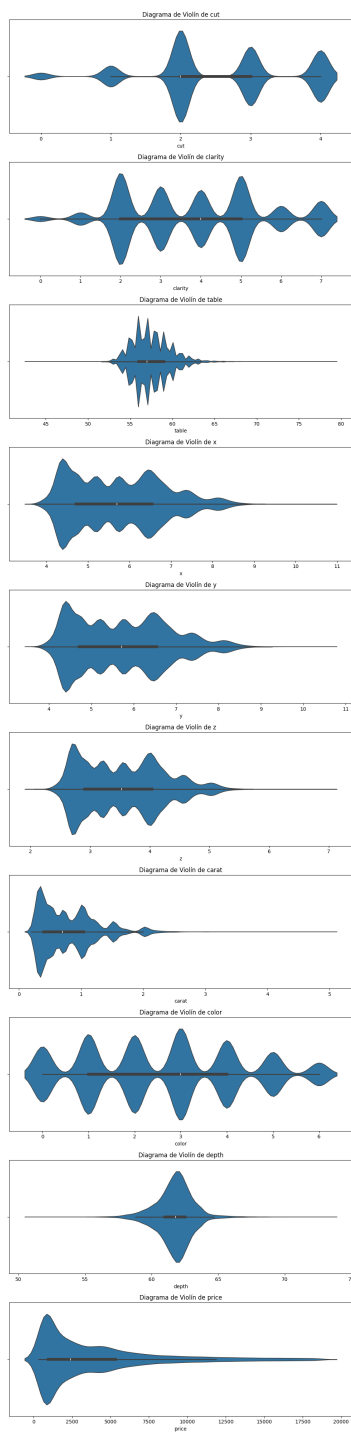


Figura 5: Diagramas de violín.

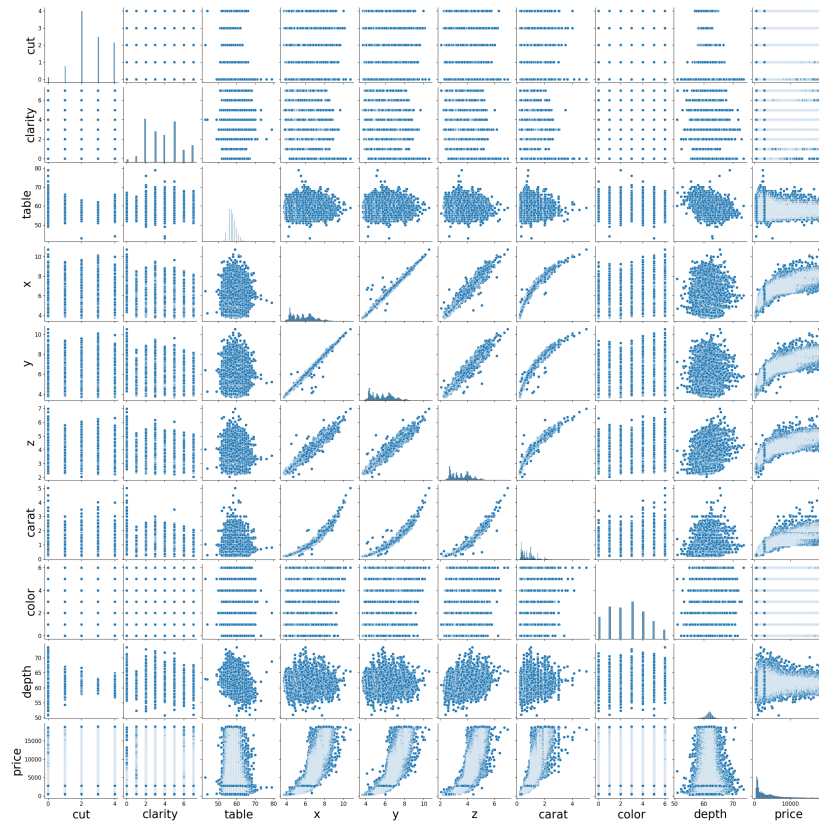


Figura 6: Diagramas de dispersión.

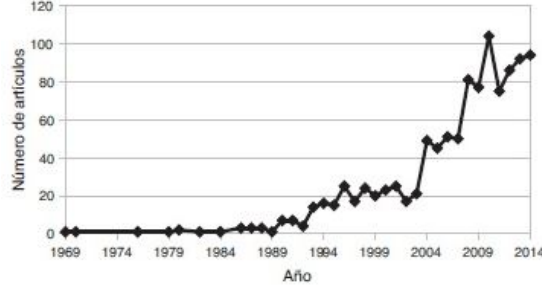


Figura 7: Aumento de artículos publicados con aprendizaje automático y diagnóstico médico [3]

grupo es alta y la similitud entre las clases es baja. La medida de similitud se define usualmente por proximidad en un espacio multidimensional [2].

Como ejemplo de la gran influencia de métodos de aprendizaje automático nos podemos ir a campos como la medicina en donde los modelos de aprendizaje computacional se han aplicado con éxito tanto a problemas motivados por la práctica clínica, como el diagnóstico asistido por ordenador, como a problemas de análisis de datos de investigación médica básica. En los últimos años ha habido un gran auge en la investigación y desarrollo de modelos de aprendizaje computacional aplicados al diagnóstico médico de diversas enfermedades y condiciones médicas [3].

La figura 7 muestra el número de artículos publicados entre 1969 y 2014 sobre aplicaciones de técnicas de aprendizaje computacional al diagnóstico médico. Como se puede observar, hay una tendencia creciente, la cual se ha acelerado durante los últimos 10 años, alcanzando un volumen de cerca de 100 artículos sobre el tema por año. Los tipos de problemas de diagnóstico médico abordados con estas técnicas cubren prácticamente todas las especialidades de la medicina, algunos ejemplos incluyen: diagnóstico de glaucoma, identificación de enfermedades cardiovasculares, detección de la enfermedad de Alzheimer y detección del cáncer de próstata [3].

4. Marco teórico

4.1. Método de k medias (clustering)

Esta técnica no supervisada consiste en generar $K \in N$ grupos para n elementos que incluyan a los n_k más cercanos (con base en cierta medida distancia, usualmente euclidiana) respecto a un centroide $c_k = (\bar{x}, \bar{y})$ tal que

$$\bar{x} = \frac{1}{n_k} \sum_{x_i \in S_k} x_i, \quad \bar{y} = \frac{1}{n_k} \sum_{y_i \in S_k} y_i. \quad (1)$$

Formalmente, la distancia más cercana respecto a los centroides c_k se define como la minimización del error cuadrado para cada grupo:

$$SS_k = \sum_{i \in k} (x_i - \bar{x}_k)^2 + (y_i - \bar{y}_k)^2. \quad (2)$$

Este algoritmo es iterativo y tiene como función objetivo

$$\min \sum_{i \in K} SS_k. \quad (3)$$

4.2. Spectral Clustering

Spectral Clustering es un algoritmo de agrupamiento (clustering) que utiliza las propiedades del espectro (valores propios) de una matriz derivada de los datos para realizar la agrupación. A diferencia de otros métodos de clustering como k-means, que se basan en la minimización de una función objetivo específica (como la distancia dentro de los grupos), Spectral Clustering se basa en la teoría de grafos y la álgebra lineal [4].

4.2.1. Conceptos clave

- **Grafo de Similitud:** Los datos son representados como un grafo donde cada nodo representa un punto de datos y los bordes (con pesos) representan la similitud entre los puntos. La matriz de adyacencia A se construye a partir de este grafo, donde A_{ij} es la similitud entre los puntos de datos i y j .
- **Matriz Laplaciana:** Se define la matriz de grado D , una matriz diagonal donde D_{ii} es la suma de las similitudes del nodo i con todos los otros nodos. La matriz Laplaciana L se puede definir de varias formas, siendo una común $L = D - A$.
- **Descomposición en Valores Propios:** Se calcula la descomposición en valores propios de la matriz Laplaciana. Los vectores propios (eigenvectors) asociados con los menores valores propios (eigenvalues) son utilizados para reducir la dimensionalidad de los datos.
- **Agrupamiento:** Los datos son proyectados en el espacio de menor dimensión usando los vectores propios. Se aplica un algoritmo de clustering, como k-means, en este nuevo espacio de características para encontrar los grupos.

4.2.2. Ventajas de Spectral Clustering

- **Capacidad de Manejar Estructuras No Lineales:** Es eficaz en situaciones donde los clústeres no tienen una forma esférica, a diferencia de k-means.

- **Flexibilidad:** Puede ser adaptado para diferentes definiciones de similitud entre puntos de datos.
- **Relación con la Teoría de Grafos:** Proporciona una base teórica sólida para muchos problemas de agrupamiento.

4.2.3. Pasos en Spectral Clustering

1. Construcción del Grafo de Similitud:

- Seleccionar una medida de similitud adecuada (por ejemplo, distancia euclidiana, RBF kernel, k-nearest neighbors).
- Construir la matriz de similitud A .

2. Cálculo de la Matriz Laplaciana:

- Construir la matriz de grado D y la matriz Laplaciana L .

3. Descomposición en Valores Propios:

- Realizar la descomposición en valores propios de L y seleccionar los vectores propios correspondientes a los menores valores propios.

4. Agrupamiento en el Espacio Reducido:

- Utilizar los vectores propios seleccionados para proyectar los datos en un espacio de menor dimensión.
- Aplicar un algoritmo de clustering (como k-means) en este espacio para encontrar los clústeres.

4.3. Algoritmo XGBoost

XGBoost (Extreme Gradient Boosting) es una implementación eficiente y escalable del algoritmo de boosting de árboles de decisión. A continuación se explica su funcionamiento y se proporciona un ejemplo de aplicación [5].

4.3.1. Funcionamiento de XGBoost

- **Boosting:** Boosting es una técnica de ensemble que combina varios modelos débiles para crear un modelo fuerte. En XGBoost, los modelos débiles son árboles de decisión.
- **Algoritmo de Gradient Boosting:** XGBoost utiliza el algoritmo de gradient boosting, que construye modelos secuencialmente. Cada nuevo modelo intenta corregir los errores de los modelos anteriores. El proceso se basa en minimizar una función de pérdida mediante gradientes.

- **Árboles de Decisión:** En cada iteración, se construye un nuevo árbol de decisión que intenta reducir el error residual de la iteración anterior. Los árboles de decisión son modelos no paramétricos que dividen el espacio de características en regiones basadas en las características más importantes.
- **Regularización:** XGBoost incluye términos de regularización en su función objetivo para controlar la complejidad del modelo y prevenir el sobreajuste. Esto se logra mediante la inclusión de términos de penalización en la función de pérdida.
- **Paralelización:** XGBoost es altamente eficiente debido a su capacidad de paralelizar la construcción de árboles de decisión. Esto acelera el proceso de entrenamiento y permite manejar grandes volúmenes de datos.
- **Optimización:** La función objetivo en XGBoost incluye la función de pérdida y los términos de regularización. La optimización se realiza mediante la técnica de boosting de gradientes, que ajusta los pesos de las instancias para minimizar la pérdida.

4.3.2. Ventajas y Desventajas de XGBoost

Las principales ventajas del algoritmo XGBoost son:

- Puede manejar grandes bases de datos con múltiples variables.
- Puede manejar valores perdidos.
- Sus resultados son muy precisos.
- Excelente velocidad de ejecución.

Por otra parte, sus principales desventajas son:

- Puede consumir muchos recursos computacionales en grandes bases de datos, por lo que se recomienda antes de aplicar esta técnica en bases de este tipo, determinar cuáles son las variables que aportarán más información a fin de considerar solo dichas variables en la obtención del modelo.
- Se deben ajustar correctamente los parámetros del algoritmo a fin de minimizar el error de precisión y evitar sobreajuste del modelo (lo que puede darse si se maneja un número muy grande de árboles).
- Solo trabaja con vectores numéricos, por lo que se requieren convertir previamente los tipos de datos no numéricos a numéricos.

4.4. Métricas para Analizar Errores en Modelos Supervisados

Para evaluar el rendimiento de un modelo supervisado, se utilizan varias métricas que permiten cuantificar la precisión de las predicciones realizadas por el modelo. A continuación, se describen algunas de las métricas más comunes [6]:

4.4.1. Error Absoluto Medio (MAE)

El Error Absoluto Medio (Mean Absolute Error, MAE) mide el promedio de los errores absolutos entre los valores predichos y los valores reales. Se define como:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (4)$$

donde y_i es el valor real, \hat{y}_i es el valor predicho y n es el número de observaciones. El MAE es una métrica que proporciona una medida clara y directa de la magnitud del error.

4.4.2. Error Cuadrático Medio (MSE)

El Error Cuadrático Medio (Mean Squared Error, MSE) mide el promedio de los errores al cuadrado entre los valores predichos y los valores reales. Se define como:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (5)$$

El MSE da más peso a los errores grandes debido a la cuadratura, lo que significa que penaliza más los errores grandes que los pequeños.

4.4.3. Raíz del Error Cuadrático Medio (RMSE)

La Raíz del Error Cuadrático Medio (Root Mean Squared Error, RMSE) es la raíz cuadrada del MSE. Se define como:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (6)$$

La RMSE se expresa en las mismas unidades que la variable de salida y es útil para interpretar la magnitud del error en el mismo contexto que los valores originales.

4.4.4. Coeficiente de Determinación (R^2)

El Coeficiente de Determinación (R^2) mide la proporción de la varianza de la variable dependiente que es explicada por el modelo. Se define como:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (7)$$

donde \bar{y} es el valor medio de los valores reales. El R^2 varía entre 0 y 1, donde un valor de 1 indica que el modelo explica perfectamente la variabilidad de los datos y un valor de 0 indica que el modelo no explica nada de la variabilidad de los datos.

5. Metodologia

5.1. Pasos en Spectral Clustering

1. Carga y Preprocesamiento de Datos:

- a) Se carga el archivo CSV `diamante.csv`.
- b) Se extrae la columna `cut` y se eliminan del conjunto de datos las características que se usarán para el clustering.
- c) Los datos se escalan usando `StandardScaler` para normalizar las características.

2. Aplicación de Spectral Clustering:

- a) Se aplica el algoritmo de `Spectral Clustering` con `n_clusters=5` y se obtienen los clústeres.
- b) Se agregan los resultados del clustering al `DataFrame` original.

3. Aplicación de t-SNE:

- a) Se aplica `t-SNE` para reducir la dimensionalidad de los datos escalados a 2 componentes para facilitar la visualización.

5.2. Pasos en XGBoost

1. Carga de datos:

- Se cargaron los datos desde un archivo CSV utilizando la biblioteca `pandas`.

2. Separación de características y variable objetivo:

- Las características (`X`) se obtuvieron eliminando la columna `price` del conjunto de datos.
- La variable objetivo (`y`) se estableció como la columna `price`.

3. División de los datos:

- Los datos se dividieron en conjuntos de entrenamiento y prueba usando `train_test_split` de `sklearn`, con un 80% de los datos para entrenamiento y un 20% para prueba.

4. Escalado de datos:

- Se aplicó el escalado a las características utilizando `StandardScaler` para normalizar las características.

5. Entrenamiento del modelo:

- Se creó un modelo de regresión utilizando `XGBRegressor` con el objetivo `reg:squarederror` y 100 estimadores.
- El modelo se entrenó con los datos escalados de entrenamiento.

6. Predicciones:

- Se realizaron predicciones en el conjunto de prueba utilizando el modelo entrenado.

7. Cálculo de métricas de error:

- Se calcularon las métricas de error MAE (Error Absoluto Medio), MSE (Error Cuadrático Medio), RMSE (Raíz del Error Cuadrático Medio) y R^2 utilizando funciones de `sklearn`.

8. Almacenamiento de resultados:

- Los resultados de las predicciones se guardaron en un archivo CSV.
- El modelo entrenado se guardó en un archivo `.pkl` utilizando `joblib`.

9. Generación de gráficos:

- Se generaron gráficos para el análisis de resultados, incluyendo:
 - **Gráfico de dispersión de valores reales vs. predichos:** Se creó un gráfico de dispersión para comparar los valores reales con los predichos.
 - **Gráfico de los residuos:** Se creó un gráfico para analizar los residuos de las predicciones.

6. Resultados

6.1. Aprendizaje no supervisado - Spectral Clustering

Se tuvieron 2 gráficas como resultado, en la figura 8 se puede ver como se distribuyen las 5 categorías de cut, cada categoría representada con un color que van desde el nivel mas bajo rojo hasta el de mejor calidad que es el morado, podemos notar que se tiene una dispersión uniforme lo cual es lo opuesto a lo que se obtuvo en la figura 9, la cual representa nuestros resultados al emplear el spectral clustering y utilizando el metodo de T-SNE para visualizarlos.

1. Visualización:

- a) Se crea una gráfica t-SNE coloreada por la variable original cut figura 8.
- b) Se crea una gráfica t-SNE coloreada por los resultados del clustering figura 9.

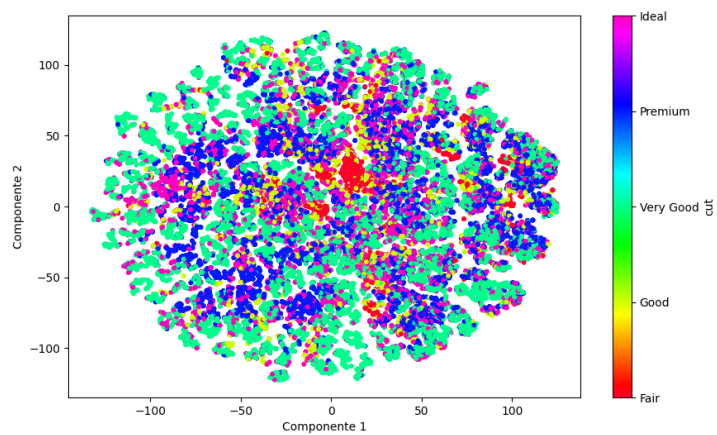


Figura 8: Gráfica t-SNE coloreada por la variable original cut

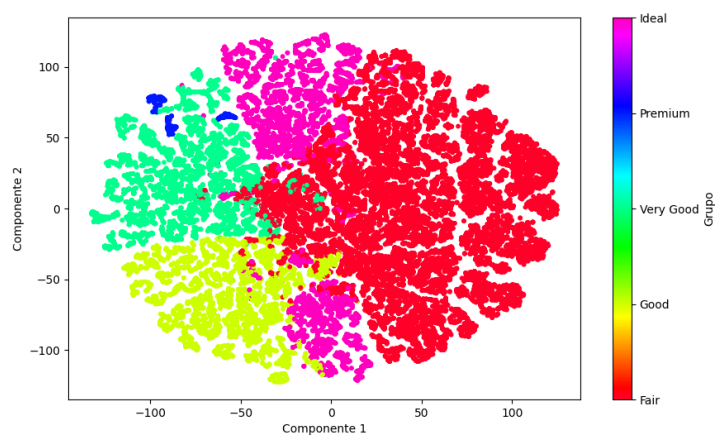


Figura 9: Gráfica t-SNE coloreada por los resultados del clustering.

6.2. Aprendizaje supervisado - XGBoost

6.2.1. Métricas de desempeño

Los resultados de las métricas de desempeño para el modelo son los siguientes:

- **MAE:** 271.97
- **MSE:** 298375.63
- **RMSE:** 546.24
- R^2 : 0.9804

Las métricas de evaluación indican lo siguiente:

- El **Error Absoluto Medio (MAE)** es de 271.97, lo que significa que, en promedio, las predicciones del modelo están desviadas de los valores reales en 271.97 unidades.
- El **Error Cuadrático Medio (MSE)** es de 298375.63. El MSE penaliza los errores grandes más que los pequeños, lo que puede indicar la presencia de algunos errores grandes en las predicciones.
- La **Raíz del Error Cuadrático Medio (RMSE)** es de 546.24, que está en la misma unidad que la variable objetivo. Este valor proporciona una medida más interpretable de la desviación estándar de los errores.
- El **Coefficiente de Determinación (R^2)** es 0.9804, lo que sugiere que el modelo explica aproximadamente el 98.04 % de la varianza en los datos de la variable objetivo, indicando un buen ajuste del modelo.

6.2.2. Análisis de Residuos

La Figura 10 muestra la dispersión de los errores (residuos) en función de los valores predichos. En esta gráfica:

- Los residuos están mayormente distribuidos alrededor de la línea de cero, lo que indica que los errores son en su mayoría pequeños.
- La dispersión de los residuos parece aumentar ligeramente con el aumento de los valores predichos, lo que podría sugerir que el modelo tiene mayor dificultad para predecir valores más altos con la misma precisión.

6.2.3. Valores Reales vs. Predichos

La Figura 11 compara los valores reales con los valores predichos por el modelo. En esta gráfica:

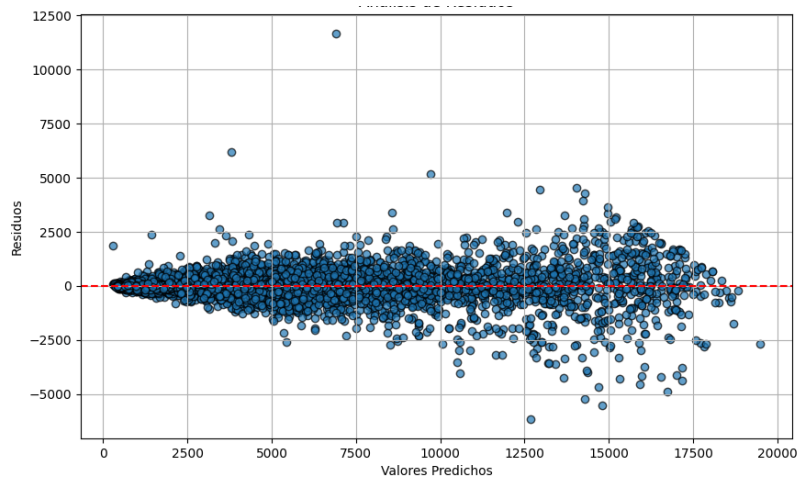


Figura 10: Análisis de residuos.

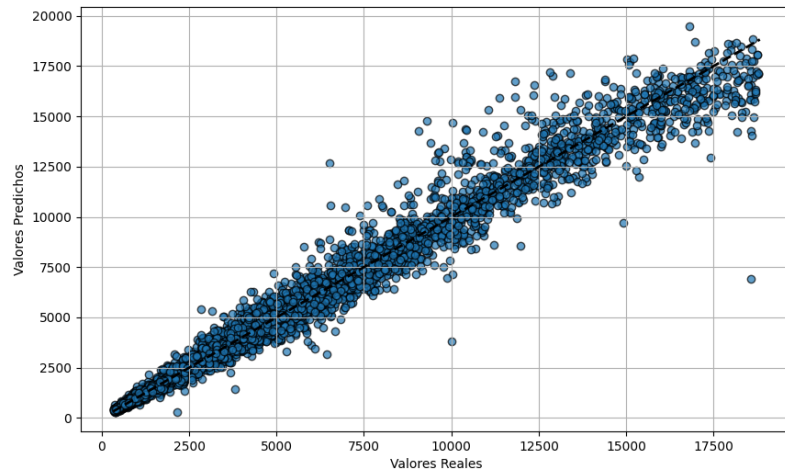


Figura 11: Valores reales vs. valores predichos.

- Los puntos están fuertemente agrupados alrededor de la línea de perfecta predicción (diagonal), lo que indica que las predicciones del modelo están muy cerca de los valores reales.
- Aunque hay algunos puntos dispersos fuera de la diagonal, estos son relativamente pocos y representan casos en los que el modelo no predijo con precisión.

7. Discusión

Viendo ambas representaciones, se puede deducir que las predicciones no han sido muy buenas. La figura de predicción 9 define 5 clusters más o menos separados. Sin embargo, nosotros conocemos las clases reales en la figura 8 y vemos que están muy mezcladas entre sí, sin ver clusters únicos ni definidos. Tal vez la única parte donde se pueden ver semejanzas sería en el centro de la figura donde está un poco revuelto y resalta más el color rojo.

El análisis de las métricas de evaluación y las gráficas de residuos y comparación de valores reales vs. predichos indican que el modelo tiene un buen rendimiento general, con alta precisión y capacidad de explicación de la varianza en los datos. Sin embargo, es importante seguir investigando y afinando el modelo para mejorar la predicción en casos extremos y reducir la dispersión de los residuos en valores altos.

Referencias

- [1] Isabel Cristina Pérez Verona and Leticia Arco García. Una revisión sobre aprendizaje no supervisado de métricas de distancia. *Revista Cubana de Ciencias Informáticas*, 10(4):43–67, 2016.
- [2] Servicios Tello, Jesús Cáceres y Informáticos. Reconocimiento de patrones y el aprendizaje no supervisado. *Universidad de Alcalá, Madrid*, 2007.
- [3] Fabio A González. Modelos de aprendizaje computacional en reumatología. *Revista Colombiana de Reumatología*, 22(2):77–78, 2015.
- [4] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.
- [5] Javier Jesús Espinosa-Zúñiga. Aplicación de algoritmos random forest y xgboost en una base de solicitudes de tarjetas de crédito. *Ingeniería, investigación y tecnología*, 21(3), 2020.
- [6] GrowUp CR. Métricas de precisión en machine learning, Fecha de publicación 28 de Octubre del 2022. URL <https://www.growupcr.com/post/metricas-precision>. Fecha de acceso (Junio del 2024).