Nombre: Luis Alejandro Baena
Código: 1023628877

Nombre: Jesús David Cantellon
Código: 1013457170

Estadistica II
Taller III

Resumen

En este paper se presenta una revisión de los métodos de clusterización y análisis de regresión lineal, dos técnicas fundamentales para el análisis de datos exploratorio y la construcción de modelos predictivos. Se describen los diferentes tipos de métodos de clusterización jerárquicos y no jerárquicos, con ejemplos de aplicación en diversos campos. Además, se profundiza en el método de mínimos cuadrados ordinarios (OLS) para la regresión lineal, incluyendo sus supuestos, resultados y pruebas de significancia individual y conjunta. Finalmente, se discuten las consideraciones prácticas para la selección y aplicación de estas técnicas en el contexto de un análisis de datos específico.

Índice

1. Clasificación no supervisada

1.1. Definición del concepto

La clasificación no supervisada, también conocida como agrupamiento o analisis de cluster, es una técnica de aprendizaje automático que se utiliza para descubrir grupos de datos sin necesidad de etiquetas predefinidas. A diferencia de la clasificación supervisada, donde los datos se etiquetan previamente con la clase a la que pertenecen, la clasificación no supervisada busca patrones y estructuras en los datos de forma automática.

Los algoritmos de clasificación no supervisada se basan en la idea de que los datos que pertenecen al mismo grupo comparten características similares entre sí. Estos algoritmos utilizan diversas medidas de similitud o distancia para evaluar la cercanía entre los puntos de datos. En función de estas medidas, los algoritmos agrupan los puntos de datos en clusters de forma iterativa.

La clasificación no supervisada tiene una amplia gama de aplicaciones en diversos campos, como:

- **Análisis de clientes**: Agrupar clientes en función de sus hábitos de compra o comportamiento para identificar segmentos de mercado con características similares.
- **Análisis de redes sociales**: Identificar comunidades en redes sociales con intereses o características compartidas.
- **Detección de fraudes**: Identificar patrones inusuales en las transacciones financieras que podrían indicar actividades fraudulentas.

Si bien la clasificación no supervisada ofrece una herramienta poderosa para descubrir patrones ocultos en datos sin etiquetas, presenta algunas limitaciones que deben considerarse. La interpretación de los clusters puede ser subjetiva y requerir conocimiento experto del dominio para comprender su significado. La elección del algoritmo y los parámetros adecuados es crucial para obtener resultados satisfactorios y evitar sesgos en los clusters. Además, los valores atípicos pueden afectar significativamente los resultados de algunos algoritmos, por lo que es importante identificarlos y manejarlos adecuadamente.

Existen dos grandes categorías de algoritmos de clasificación no supervisada, **Algoritmos jerárquicos** y **Algoritmos no jerárquicos**. En las subsecciones siguientes, se mostrará cada uno a detalle.

1.2. Agrupamiento Jerarquico

Este tipo de algoritmos construyen una jerarquía de clusters, empezando por agrupar los puntos de datos más cercanos y luego fusionando grupos sucesivamente hasta alcanzar un nivel de granularidad deseado.

Los métodos jerárquicos, se dividen en dos grandes categorías: aglomerativos y disociativos. Cada una de estas categorías presenta una amplia gama de variantes, lo que las convierte en herramientas versátiles para la agrupación de datos.

Métodos aglomerativos (ascendentes): Estos comienzan el análisis con tantos grupos como individuos haya en el conjunto de datos. A partir de estas unidades iniciales, se van formando grupos de forma ascendente, fusionando iterativamente los clusters más similares según un criterio de distancia o similitud predefinido. Este proceso continúa hasta que al final, todos los casos tratados se encuentran agrupados en un único conglomerado.

Métodos disociativos (descendentes): Los métodos disociativos, también conocidos como descendentes, siguen un enfoque inverso al de los métodos aglomerativos. En este caso, se comienza con un único conglomerado que engloba a todos los casos del conjunto de datos. A partir de este grupo inicial, se

realizan divisiones sucesivas, separando los clusters más disímiles según el criterio de distancia o similitud elegido. Este proceso continúa hasta que al final, se obtienen tantas agrupaciones como individuos haya en el conjunto de datos.

La elección del método jerárquico más adecuado depende de diversos factores, como las características del conjunto de datos, los objetivos del análisis y las preferencias del investigador. En general, los métodos aglomerativos son más intuitivos y fáciles de interpretar, mientras que los métodos disociativos pueden ser más eficientes para conjuntos de datos de gran tamaño.

Los métodos jerárquicos permiten visualizar las relaciones entre los clusters mediante diagramas de dendrogramas, los cuales representan la jerarquía de las fusiones o divisiones realizadas durante el proceso de agrupamiento. Además, estos métodos pueden combinarse con otras técnicas de análisis de datos para obtener una comprensión más profunda de los patrones presentes en los datos.

Se mostrarán 2 ejemplos de agrupamiento jerarquico usando enlace único y completo con el siguiente data set:

Iris: El conjunto de datos de iris proporciona las medidas en centímetros de las variables longitud y ancho del sépalo y longitud y ancho de los pétalos, respectivamente, para 50 flores de cada una de las 3 especies de iris. Las especies son Iris setosa, versicolor y virginica.

1.2.1. Enlace único y enlace completo

Enlace unico

La estrategia conocida como "single linkage" en la literatura anglosajona considera la distancia o similitud entre dos clústeres como la mínima distancia (o máxima similitud) entre sus componentes. Por ejemplo, si después de la etapa K-ésima ya se han formado $n-K$ clústeres, la distancia entre los clústeres C_i (con n_i elementos) y C_j (con n_j elementos) sería:

$$d(C_i, C_j) = \min_{x_l \in C_i, x_m \in C_j} \{d(x_l, x_m)\}, \quad l = 1, \ldots, n_i, \quad m = 1, \ldots, n_j$$

y la similitud, si se emplea una medida de este tipo, sería:

$$s(C_i, C_j) = \max_{x_l \in C_i, x_m \in C_j} \{s(x_l, x_m)\}, \quad l = 1, \ldots, n_i, \quad m = 1, \ldots, n_j$$

Entonces, la estrategia seguida en el nivel $K+1$ sería:

1. En el caso de emplear distancias, se unirían los clústeres C_i y C_j si

$$d(C_i, C_j) = \min_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \{d(C_{i_1}, C_{j_1})\} = \min_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \left\{ \min_{x_l \in C_{i_1}, x_m \in C_{j_1}} \{d(x_l, x_m)\} \right\}$$

2. En el caso de emplear similitudes, se unirían los clústeres C_i y C_j si

$$s(C_i, C_j) = \max_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \{s(C_{i_1}, C_{j_1})\} = \max_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \left\{ \max_{x_l \in C_{i_1}, x_m \in C_{j_1}} \{s(x_l, x_m)\} \right\}$$

Se sigue la norma general de maximizar las similitudes o minimizar las distancias.

Enlace completo

En este método, también conocido como el procedimiento de amalgamamiento completo (complete linkage), se considera que la distancia o similitud entre dos clústeres se debe medir teniendo en cuenta sus elementos más dispares. Esto significa que la distancia o similitud entre clústeres se define respectivamente como la máxima distancia (o mínima similitud) entre sus componentes.

Por lo tanto, si ya estamos en la etapa K-ésima, y por lo tanto ya hay formados $n-K$ clústeres, la distancia y similitud entre los clústeres C_i y C_j (con n_i y n_j elementos respectivamente) serían:

$$d(C_i, C_j) = \max_{x_l \in C_i, x_m \in C_j} \{d(x_l, x_m)\}, \quad l = 1, \ldots, n_i, \quad m = 1, \ldots, n_j$$

$$s(C_i, C_j) = \min_{x_l \in C_i, x_m \in C_j} \{s(x_l, x_m)\}, \quad l = 1, \ldots, n_i, \quad m = 1, \ldots, n_j$$

Entonces, la estrategia seguida en el siguiente nivel, $K+1$, sería:

1. En el caso de emplear distancias, se unirían los clústeres C_i y C_j si

$$d(C_i, C_j) = \min_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \{d(C_{i_1}, C_{j_1})\} = \min_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \left\{ \max_{x_l \in C_{i_1}, x_m \in C_{j_1}} \{d(x_l, x_m)\} \right\}$$

2. En el caso de emplear similitudes, se unirían los clústeres C_i y C_j si

$$s(C_i, C_j) = \max_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \{s(C_{i_1}, C_{j_1})\} = \max_{\substack{i_1, j_1 = 1, \ldots, n-K \\ i_1 \neq j_1}} \left\{ \min_{x_l \in C_{i_1}, x_m \in C_{j_1}} \{s(x_l, x_m)\} \right\}$$

Se sigue la norma general de minimizar las distancias o maximizar las similitudes.

Para explorar visualmente las relaciones entre las variables del conjunto de datos Iris, emplearemos una gráfica **SPLOM** (Scatter Plot Matrix). Una gráfica SPLOM es una matriz que permite visualizar el diagrama de dispersión de cada variable del conjunto contra todas las demás variables. En cada celda (i, j) de la matriz se muestra el diagrama de dispersión de la variable X_i en el eje horizontal frente a la variable X_j en el eje vertical. De esta manera, podemos visualizar las relaciones entre las variables Sepal Length, Sepal Width, Petal Length y Petal Width del dataset Iris.

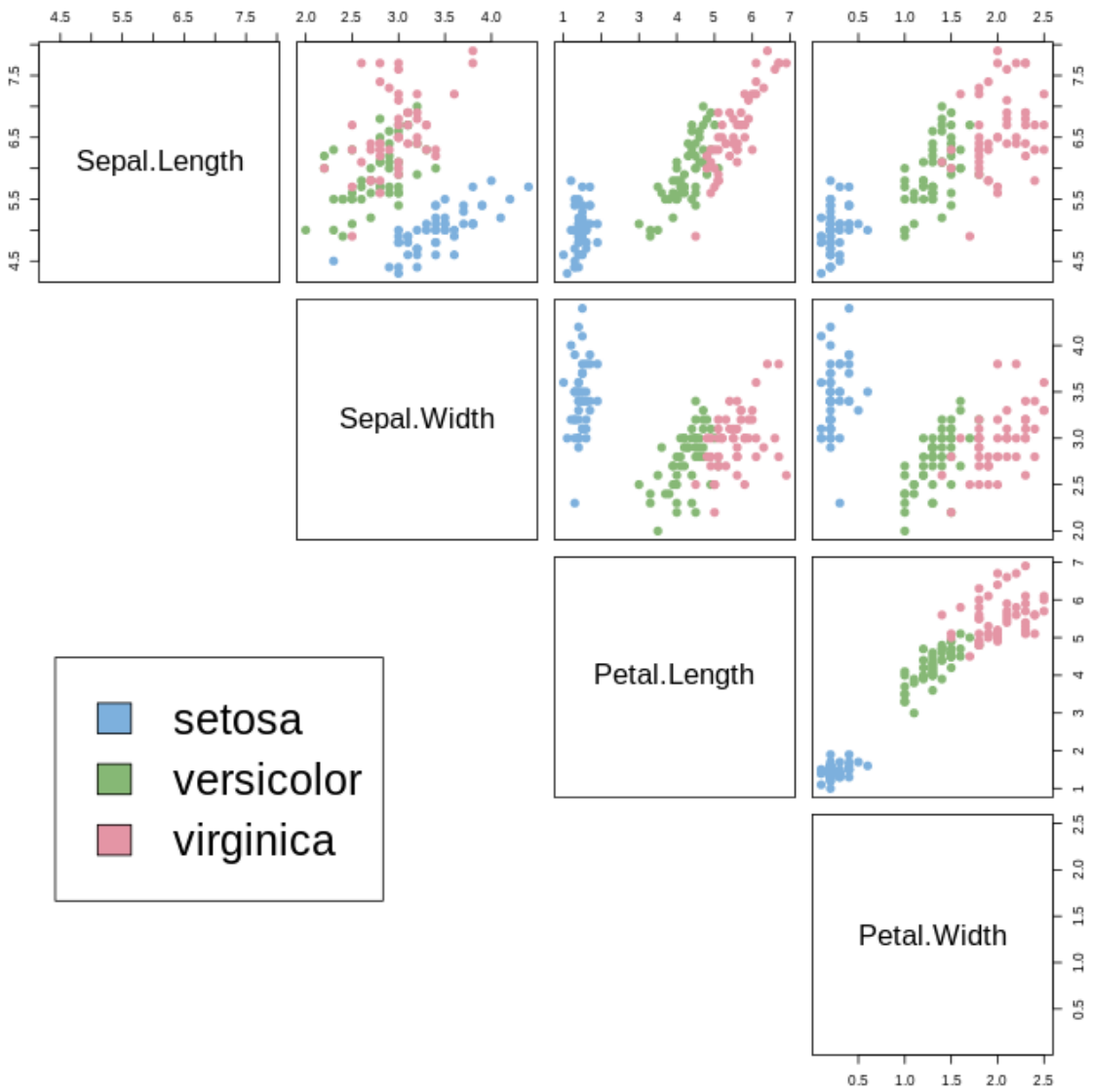

Figura 1: Gráfico SPLOM del dataset **iris**

Podemos ver que las especies Setosa son claramente diferentes de Versicolor y Virginica (tienen pétalos más largos y anchos). Pero Versicolor y Virginica no se pueden separar fácilmente basándose en las medidas del ancho/largo de sus sépalos y pétalos.

Ahora, construiremos dendrogramas que representen la estructura jerárquica de los grupos obtenidos con cada método.

Los dendrogramas nos permitirán visualizar las relaciones entre las flores Iris y comprender mejor la agrupación resultante. Observaremos cómo las diferentes especies de Iris (Setosa, Versicolor y Virginica) se distribuyen en los clusters formados por cada método de enlace.

Adicionalmente, compararemos los dendrogramas obtenidos con enlace único y enlace completo para identificar posibles diferencias en la agrupación de las flores. Esta comparación nos ayudará a comprender mejor las características distintivas de cada método de enlace y su impacto en la estructura final de los clusters.

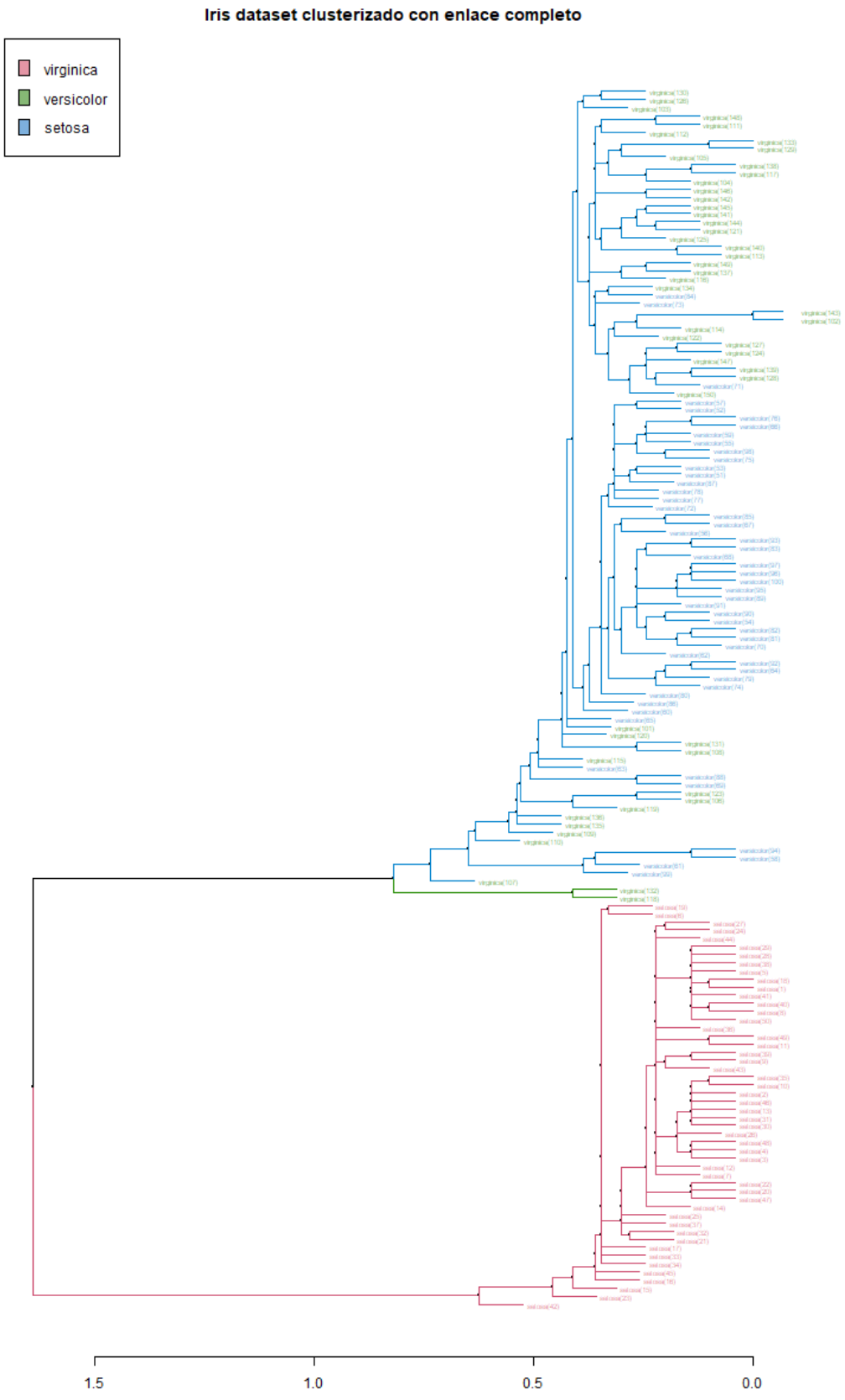

Figura 2: Gráfico Dendrograma del dataset **iris** con enlace único

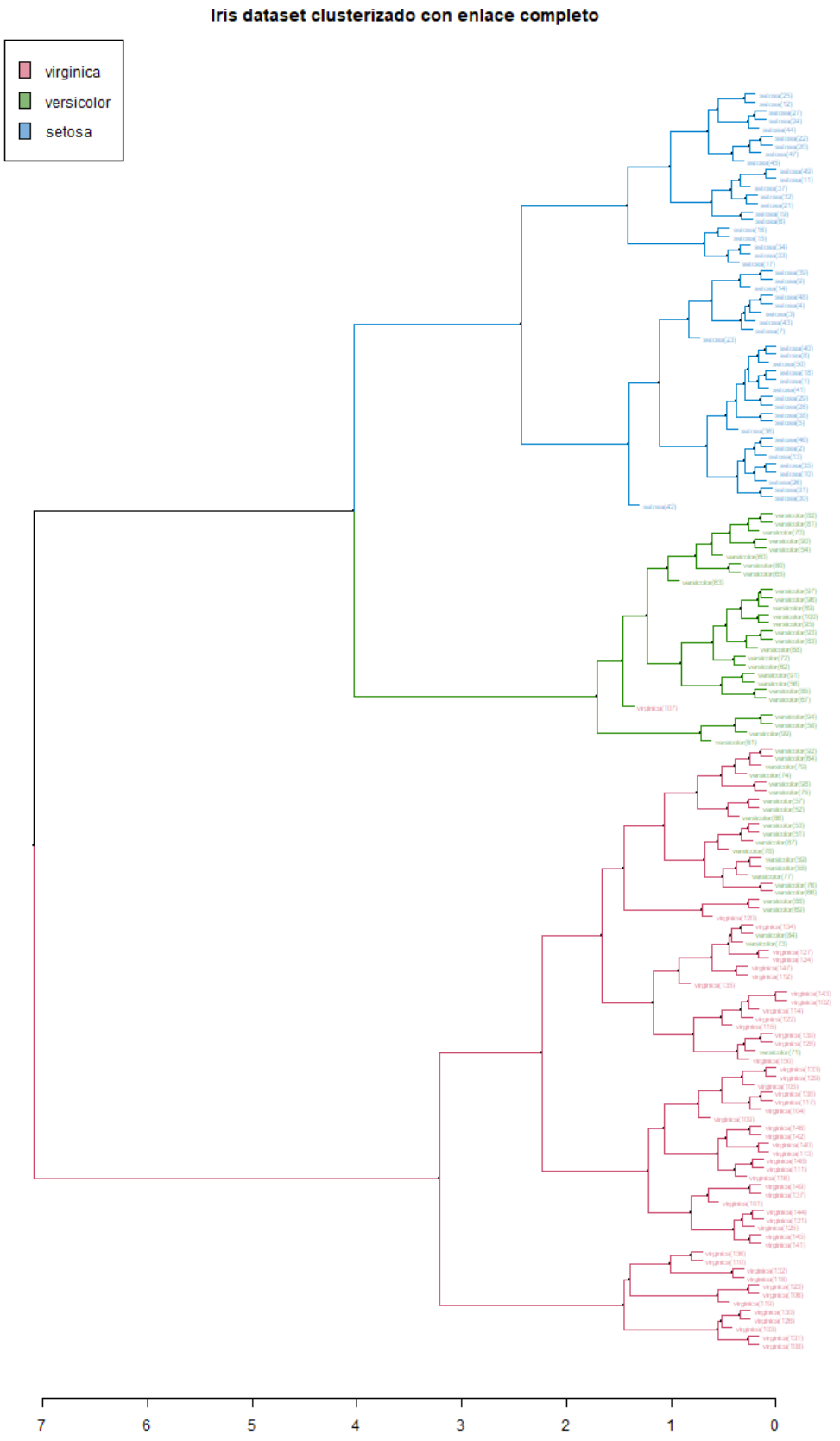

Figura 3: Gráfico Dendrograma del dataset **iris** con enlace completo

Estas visualizaciones demuestran claramente que la separación de la especie "Setosa" lograda por el clustering jerárquico es muy buena. Sin embargo, el método comete errores al etiquetar muchas especies "Versicolor" como "Virginica".

Además, la estructura del dendograma (árbol) del clustering jerárquico nos ayuda a identificar observaciones extremas. Por ejemplo, podemos observar que hay una observación Virginica que está donde hay claramente especies Versicolor. De igual forma, hay algunas Versicolor que se encuentran localizadas dentro del grupo de flores Virginica. En nuestro caso, el método por link completo tuvo un mejor desempeño que el método por link simple

1.2.2. Código

```
# Gráfico 1 (SPLOM)

# Cargamos la biblioteca colorspace para obtener colores agradables
library(colorspace)
library(dendextend)

# Cargamos el conjunto de datos de iris desde el paquete datasets
iris <- datasets::iris
# Creamos un nuevo dataframe excluyendo la quinta columna que
# contiene las etiquetas de especies
iris2 <- iris[, -5]
# Extraemos las etiquetas de especies del dataframe original
species_labels <- iris[, 5]

# Creamos un vector de colores basado en las etiquetas de especies
species_col <- rev(rainbow_hcl(3))[as.numeric(species_labels)]

# Dibujamos un SPLOM (scatterplot matrix):
pairs(iris2, # Usamos el dataframe iris2
    col = species_col, # Asignamos colores según las etiquetas de especies
    lower.panel = NULL, # No mostramos paneles inferiores
    # Ajustamos el tamaño de las etiquetas y puntos en el gráfico
    cex.labels = 2, pch = 19, cex = 2
)

# Agregamos una leyenda
legend(
    x = 0.05, y = 0.4, cex = 2, # Posición y tamaño de la leyenda
    # Etiquetas de especies como leyenda
    legend = as.character(levels(species_labels)),
    # Colores únicos asociados a las etiquetas de especies
    fill = unique(species_col)
)


## Grafico 2 (Dendrograma con enlace único)

# Calculamos la distancia euclidiana entre las observaciones en el
# conjunto de datos iris2
```

```r
d_iris <- dist(iris2)
# Realizamos un agrupamiento jerárquico completo utilizando
# la distancia calculada
hc_iris <- hclust(d_iris, method = "single")
# Definimos el orden del dendrograma lo más cercano posible al
# orden de las observaciones
iris_species <- rev(levels(iris[, 5]))
dend <- as.dendrogram(hc_iris)
dend <- rotate(dend, 1:150)

# Coloreamos las ramas basadas en los clusters
dend <- color_branches(dend, k = 3)

# Asignamos manualmente los colores de las etiquetas, en la medida de
# lo posible, a la clasificación real de las flores
labels_colors(dend) <-
    rainbow_hcl(3)[sort_levels_values(
        as.numeric(iris[, 5])[order.dendrogram(dend)]
    )]

# Agregamos el tipo de flor a las etiquetas
labels(dend) <- paste(as.character(iris[, 5])[order.dendrogram(dend)],
    "(", labels(dend), ")",
    sep = ""
)
# Ajustamos la altura del dendrograma
dend <- hang.dendrogram(dend, hang_height = 0.1)
# Reducimos el tamaño de las etiquetas
dend <- set(dend, "labels_cex", 0.5)
# Y dibujamos el dendrograma
par(mar = c(3, 3, 3, 7))
plot(dend,
    main = "Iris dataset clusterizado con enlace único",
    horiz = TRUE, nodePar = list(cex = .007)
)
# Agregamos una leyenda
legend("topleft", legend = iris_species, fill = rainbow_hcl(3))


## Grafico 3 (Dendrograma con enlace completo)

# Calculamos la distancia euclidiana entre las observaciones
# en el conjunto de datos iris2
d_iris <- dist(iris2)
# Realizamos un agrupamiento jerárquico completo utilizando la
# distancia calculada
hc_iris <- hclust(d_iris, method = "complete")
# Definimos el orden del dendrograma lo más cercano posible al
# orden de las observaciones
iris_species <- rev(levels(iris[, 5]))
```

```r
dend <- as.dendrogram(hc_iris)
dend <- rotate(dend, 1:150)

# Coloreamos las ramas basadas en los clusters
dend <- color_branches(dend, k = 3)

# Asignamos manualmente los colores de las etiquetas, en la medida
# de lo posible, a la clasificación real de las flores
labels_colors(dend) <-
    rainbow_hcl(3)[sort_levels_values(
        as.numeric(iris[, 5])[order.dendrogram(dend)]
    )]

# Agregamos el tipo de flor a las etiquetas
labels(dend) <- paste(as.character(iris[, 5])[order.dendrogram(dend)],
    "(", labels(dend), ")",
    sep = ""
)
# Ajustamos la altura del dendrograma
dend <- hang.dendrogram(dend, hang_height = 0.1)
# Reducimos el tamaño de las etiquetas
dend <- set(dend, "labels_cex", 0.5)
# Y dibujamos el dendrograma
par(mar = c(3, 3, 3, 7))
plot(dend,
    main = "Iris dataset clusterizado con enlace completo",
    horiz = TRUE, nodePar = list(cex = .007)
)
# Agregamos una leyenda
legend("topleft", legend = iris_species, fill = rainbow_hcl(3))
```

1.3. Agrupamiento No Jerarquico

Estos algoritmos no construyen una jerarquía de clusters, sino que asignan directamente cada punto de datos a un cluster. El algoritmo **K-means** es uno de los ejemplos más populares de este tipo de algoritmo.

A continuación se describe:

1.3.1. K-means

El algoritmo de K-means es una técnica de agrupamiento ampliamente utilizada para dividir un conjunto de datos en un número predeterminado de grupos (K). El algoritmo se compone de tres fases principales:

1. Inicialización:
En esta fase, se seleccionan K puntos aleatorios como centros iniciales de los grupos. Existen diferentes estrategias para elegir estos puntos:

- **Aleatoria**: Se asignan aleatoriamente los elementos del conjunto de datos a los grupos y se toman como centros los promedios de cada grupo.

- **Puntos más alejados**: Se seleccionan los K puntos del conjunto de datos que se encuentran a mayor distancia entre sí.
- **Selección informada**: Se utilizan técnicas como el análisis de componentes principales (PCA) para identificar puntos representativos del conjunto de datos y utilizarlos como centros iniciales.

2. Asignación de elementos:

Una vez establecidos los centros iniciales, se calcula la distancia de cada elemento del conjunto de datos a cada uno de los centros. Posteriormente, cada elemento se asigna al grupo cuyo centro le resulte más cercano. Al introducir un nuevo elemento al grupo, se recalcula la media del mismo para reflejar la nueva información.

3. Convergencia:

En esta fase, se evalúa si la asignación actual de los elementos a los grupos es óptima. Para ello, se utiliza un criterio de optimalidad, como la minimización de la suma de las distancias cuadradas de cada elemento a su centroide. Si reasignar un elemento a otro grupo mejora este criterio, se realiza la reasignación y se recalculan los centros de los grupos afectados. El proceso se repite hasta que no se observe ninguna mejora en el criterio de optimalidad, lo que indica que se ha alcanzado la convergencia y se obtienen los grupos finales.

En resumen, estos son los pasos para implementar el algoritmo:

1. Elegir el número de clústeres k
2. Seleccionar k puntos aleatorios de los datos como centroides
3. Asignar todos los puntos al centroide de clúster más cercano
4. Recalcular los centroides de los clústeres recién formados
5. Repetir los pasos 3 y 4

El siguente ejemplo aplicativo ilustra la aplicación del algoritmo k-means para la clasificación de flores Iris en base a sus características físicas. El objetivo principal es determinar el número óptimo de grupos (clústeres) que representen adecuadamente las relaciones entre las flores de las diferentes especies (Iris setosa, Iris versicolour e Iris virginica).

Para ello, se carga y prepara el conjunto de datos Iris, seleccionando las características relevantes. Posteriormente, se emplea la técnica del método del codo "elbow method" para identificar el número óptimo de clústeres, evaluando la suma de cuadrados dentro del clúster (WCSS) para cada valor considerado. Una vez determinado el número óptimo, se ajusta el modelo k-means y se predicen las etiquetas de clúster para cada flor.

Finalmente, se visualizan los resultados obtenidos mediante una gráfica de dispersión, donde se observan claramente tres grupos distintos de flores, cada uno correspondiente a una especie de Iris. La presencia de centroides en cada grupo y la separación entre ellos respaldan la efectividad del algoritmo k-means en la identificación de patrones en el conjunto de datos.

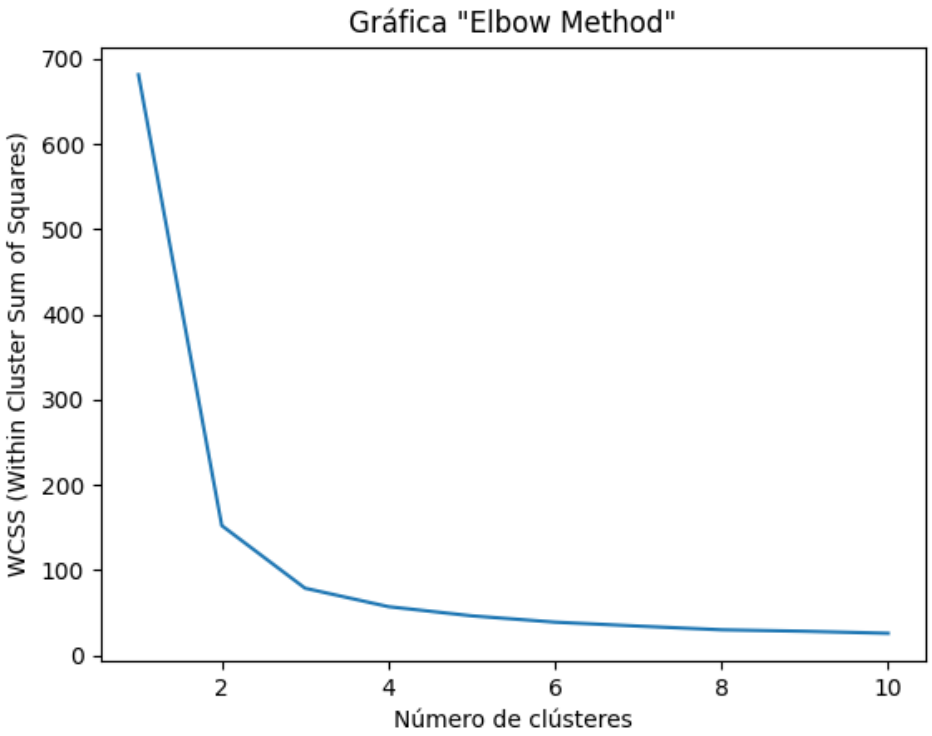

Figura 4: Gráfica de codo para el dataset **iris**

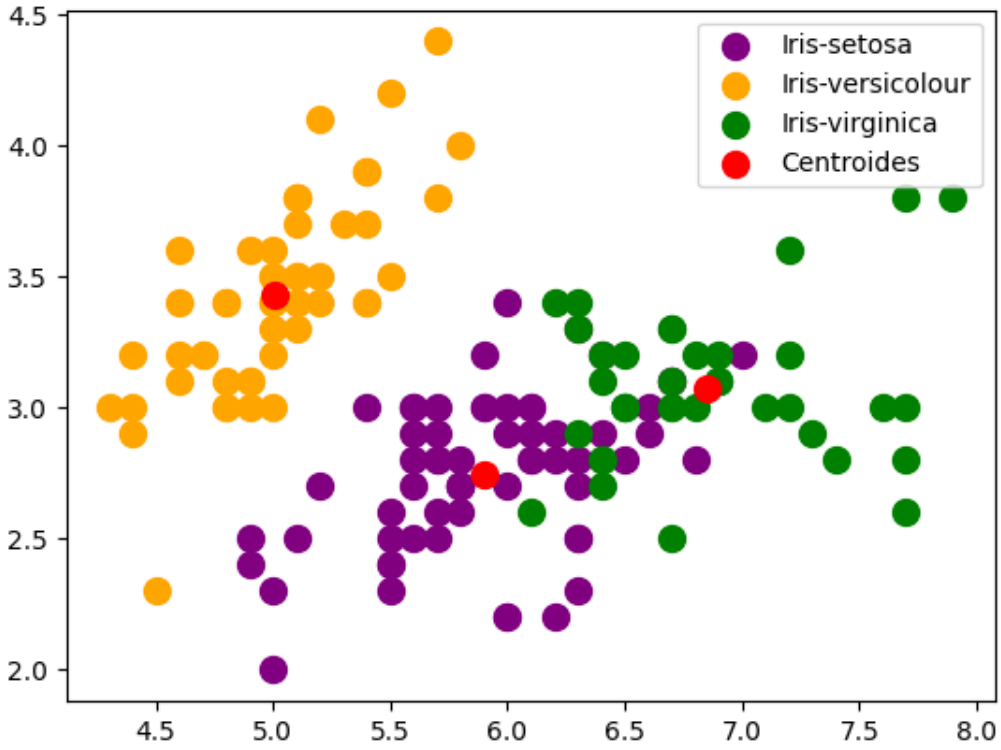

Figura 5: Gráfica de clusters para el dataset **iris**

1.3.2. Código *(python)*

```python
# Encontrar el número óptimo de clústeres para la clasificación con k-means
# Importar la clase KMeans de sklearn para realizar clustering
from sklearn.cluster import KMeans
# Importar pandas para manipulación de datos
import pandas as pd
# Importar matplotlib para visualización
import matplotlib.pyplot as plt

# Cargar el conjunto de datos
iris = pd.read_csv("iris.csv")

# Seleccionar las características (features) del conjunto de datos
x = iris.iloc[:, [0, 1, 2, 3]].values

# (Within Cluster Sum of Squares) Inicializar una lista para almacenar la suma de
# cuadrados dentro del clúster (WCSS)
wcss = []

# Iterar sobre un rango de posibles números de clústeres (de 1 a 10)
for i in range(1, 11):
```

```
    # Inicializar y ajustar el modelo KMeans con el número actual de clústeres
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300,
        n_init=10, random_state=0)
    kmeans.fit(x)
    # Calcular y almacenar la suma de cuadrados dentro del clúster para este
    # número de clústeres
    wcss.append(kmeans.inertia_)

# Visualizar la "curva del codo" para determinar el número óptimo de clústeres
plt.plot(range(1, 11), wcss)
plt.title('Gráfica "Elbow Method"')
plt.xlabel('Número de clústeres')
# Suma de cuadrados dentro del clúster
plt.ylabel('WCSS (Within Cluster Sum of Squares)')
plt.show()

# Inicializar y ajustar el modelo KMeans con el número óptimo de clústeres encontrado (3)
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10, random_state=0)
y_kmeans = kmeans.fit_predict(x)

# Visualizar los clústeres y los centroides
plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s=100,
    c='purple', label='Iris-setosa')
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s=100,
    c='orange', label='Iris-versicolour')
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s=100,
    c='green', label='Iris-virginica')
plt.scatter(kmeans.cluster_centers_[:, 0],
    kmeans.cluster_centers_[:, 1], s=100, c='red', label='Centroides')
plt.legend()
plt.show()
```

2. Clasificación supervisada

El aprendizaje supervisado, también conocido como machine learning supervisado, es una rama del aprendizaje automático y la inteligencia artificial. Se caracteriza por utilizar conjuntos de datos etiquetados para entrenar algoritmos que pueden clasificar datos o hacer predicciones con exactitud.

Durante el entrenamiento del modelo, los datos de entrada se procesan y el modelo ajusta sus ponderaciones hasta que logra una adaptación adecuada, lo cual se verifica mediante el proceso de validación cruzada. El aprendizaje supervisado permite a las organizaciones abordar diversos problemas del mundo real a gran escala, como la clasificación de correos electrónicos de spam en una carpeta separada de la bandeja de entrada. Además, se puede emplear para desarrollar modelos de machine learning con alta precisión.

El aprendizaje supervisado emplea un conjunto de entrenamiento para instruir a los modelos a generar el resultado esperado. Este conjunto de datos contiene tanto las entradas como las salidas correctas, lo que permite al modelo aprender gradualmente. El algoritmo evalúa su precisión mediante una función de pérdida y se ajusta hasta que el error se haya reducido a un nivel aceptable.

2.1. Analisis de dependencia

En el análisis estadístico multidimensional, es fundamental identificar la interdependencia o relación entre dos o más características analizadas.

La dependencia entre dos (o más) variables puede ser una relación funcional exacta, como la que existe entre la velocidad y la distancia recorrida por un móvil; o puede ser estadística. La dependencia estadística es un tipo de relación en la que, aunque no se puede determinar con exactitud el valor de la variable dependiente a partir de los valores de las variables independientes, sí se puede prever un comportamiento general. Por ejemplo, la relación entre el peso y la estatura en una población es una relación estadística.

El análisis de la dependencia estadística se puede abordar desde dos perspectivas (aunque están íntimamente relacionadas):

- El estudio del grado de dependencia entre las variables, lo cual se aborda en la teoría de la correlación.
- La determinación de la estructura de dependencia que mejor describe la relación, lo cual se analiza mediante la regresión.

Una vez determinada la estructura de esta dependencia, la regresión tiene como objetivo final poder asignar un valor a la variable Y en un individuo, conociendo el valor de la(s) variable(s) X $(X_1, X_2, \ldots, X_n)$.

En el caso bidimensional, dadas dos variables X e Y con una distribución conjunta de frecuencias (x_i, y_j, n_{ij}), llamamos regresión de Y sobre X (Y/X) a una función que explica la variable Y para cada valor de X, y regresión de X sobre Y (X/Y) a una función que explica la variable X para cada valor de Y. Cabe destacar que, en general, estas dos funciones no tienen por qué coincidir.

Ahora, considerando esto podemos ubicarnos en el campo semántico de los métodos estadísticos de aprendizaje supervisado, pues, en el ámbito de la minería de datos, el aprendizaje supervisado se desglosa en dos categorías principales: clasificación y regresión.

- La clasificación implica el uso de algoritmos para asignar con precisión los datos de prueba en diferentes categorías predefinidas. Su objetivo es identificar entidades específicas dentro del conjunto de datos y proporcionar conclusiones sobre cómo etiquetar o definir esas entidades. Algunos de los algoritmos de clasificación más comunes incluyen clasificadores lineales, máquinas de vectores de soporte (SVM), árboles de decisión, el método K de los vecinos más cercanos y bosques aleatorios.
- Por otro lado, la regresión se emplea para comprender la relación entre variables dependientes e independientes. Es útil para realizar proyecciones, como predecir los ingresos por ventas de un negocio específico. Algunos algoritmos populares de regresión son la regresión lineal, la regresión logística y la regresión polinomial.

Profundizaremos en la categoría que ha estado presente en las dos clasificaciones presentadas: Regresión. Mas especificamente nos enfocaremos en la regresión lineal y el método de mínimos cuadrados ordinarios.

2.2. Metodo de mínimos cuadrados ordinarios

2.2.1. El modelo de regresion lineal

Los modelos de regresión tienen gran variedad de aplicaciones, tanto en economía como en biología (por ejemplo). Sin importar el campo de aplicación, al momento de usar estos modelos se busca especificar

y estimar un modelo de relación entre las variables relativas a determinada cuestión conceptual. En su forma más general y, por tanto, más abstracta, tal modelo de relación puede representarse como:

$$Y = f(X_1, X_2, X_3, \ldots, X_k; \beta)$$

donde Y es la variable cuyo comportamiento se pretende explicar, y $X_1, X_2, \ldots, X_k$ son las distintas variables que se suponen potencialmente relevantes como factores explicativos de la primera. El vector denota una lista de parámetros que recogen la magnitud con que las variaciones en los valores de las variables X_i se transmiten a variaciones en la variable Y.

Nos enfocaremos en el estudio de modelos de relación o modelos de regresiones lineales, es decir, del tipo:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k$$

en el que resulta evidente que los parámetros transmiten directamente efectos inducidos por los valores de las variables X_i sobre la variable Y, que se pretende explicar.

La estimación de tales relaciones se efectúa a partir de información muestral acerca de los valores tomados por $Y, X_1, X_2, \ldots, X_k$, y trata de cuantificar la magnitud de dependencia entre ellas. Esta estimación es hecha por los mínimos cuadrados ordinarios.

2.2.2. Estimador Mínimos Cuadrados Ordinarios

El modelo de regresión lineal asume una relación lineal (en parámetros) entre una variable dependiente Y_i y un conjunto de variables explicatorias $X_i' = (X_{i0}, X_{i1}, \ldots, X_{ik})$. Consideremos una muestra de n observaciones $i = 1, 2, \ldots, n$. Cada observación i sigue

$$Y_i = \mathbf{X_i'}\beta + u_i$$

donde β es un vector columna de parámetros de dimensión $k+1$, $\mathbf{X_i'}$ es un vector fila de dimensiones $k+1$ y u_i es un escalar llamado el termino error.

La muestra completa de n observaciones puede ser expresada en notación matricial,

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{u}$$

donde $\mathbf{Y}$ es un vector columna de dimensiones n, $\mathbf{X}$ es una matriz de dimension $n \times (k+1)$ y $\mathbf{u}$ es un vector columna de dimensión n de terminos de errores.

Ahora, el estimador de Mínimos Cuadrados Ordinarios minimiza los cuadrados de las distancias entre la variable dependiente observada $\mathbf{Y}$ y su predicción:

$$S(\beta) = \sum_{i=1}^{n} (Y_i - X_i'\beta)^2 = (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) \to \min_{\beta}$$

El resultado del estimador OLS de β es:

$$\hat{\beta} = (\mathbf{X'X})^{-1}\mathbf{X'Y}$$

Varios de los siguientes supuestos se formulan en diferentes alternativas. Diferentes conjuntos de suposiciones conducirán a diferentes propiedades del estimador OLS. Los centrales son

- Supuesto 1: Linealidad

$$Y_i = \mathbf{X_i'}\beta + u_i \text{ y } E[u_i] = 0$$

- Supuesto 2: Independencia

$$\{\mathbf{X_i}, Y_i\}_{i=1}^{n} \text{ i.i.d}$$

- Supuesto 3: Exogenidad

$$E\left[u_i|\mathbf{X_i}\right] = 0$$

- Supuesto 4: Varianza del error

$$V\left[u_i|\mathbf{X_i}\right] = \sigma^2 < \infty \text{ (Homocedasticidad)}$$

$$V\left[u_i|\mathbf{X_i}\right] = \sigma_i^2 = g(\mathbf{X_i}) < \infty \text{ (Heterocedasticidad Condicional)}$$

- Supuesto 5: Identificabilidad

$$E\left[\mathbf{X_i}\mathbf{X_i'}\right] = \mathbf{Q_{XX}} \text{ es definido positivo y finito.}$$

$$rango(\mathbf{X}) = k + 1 < n$$

2.2.3. Analisis de significancia individual

El análisis de significancia individual se refiere a evaluar la significancia estadística de cada coeficiente β_i por separado en el modelo de regresión. Esto se hace mediante pruebas t, donde se evalúa la hipótesis nula de que el coeficiente β_i es igual a cero (es decir, que la variable independiente X_i no tiene un efecto significativo sobre la variable dependiente Y).

Asumiendo los supuestos 1, 2, 3, 4 y 5, la hipótesis nula simple de la forma $H_0 : \beta_k = 0$ es probada con la prueba t. Si la hipótesis nula es verdadera, el estadístico t definido como

$$t = \frac{\hat{\beta}_k}{\hat{ee}\left[\hat{\beta}_k\right]} \sim t_{n-k-1}$$

sigue una distribución t con $n-k-1$ grados de libertad. El error estandar $\hat{ee}\left[\hat{\beta}_k\right]$ es la raiz cuadrada del elemento en la fila $k+1$ y la columna $k+1$ de $\hat{V}\left[\hat{\beta}|\mathbf{X}\right]$.

2.2.4. Analisis de significancia conjunta

El análisis de significancia conjunta se refiere a evaluar si, en conjunto, las variables independientes tienen un efecto significativo sobre la variable dependiente. Esto se hace mediante una prueba F, que evalúa la hipótesis nula de que todos los coeficientes $\beta_1, \beta_2, \ldots, \beta_k$ son iguales a cero simultáneamente.

Una hipótesis nula de la forma $H_0 : \mathbf{R}\beta = 0$ con J restricciones lineales es conjuntamente probada con la prueba F. Si la hipótesis nula es cierta, el estadístico F definido como

$$F = \frac{(\mathbf{R}\hat{\beta})'(\mathbf{R}\hat{\mathbf{V}}(\hat{\beta}|\mathbf{X})\mathbf{R}')^{-1}(\mathbf{R}\hat{\beta})}{J} \sim F_{J,n-k-1}$$

sigue una distribución F con J grados de libertad del numerador y $n-k-1$ grados de libertad del denominador. Solo bajo homocedasticidad (Supuesto 4), el estadístico F puede ser calculado como

$$F = \frac{(SSR_{restringido} - SSR)/J}{SSR/(n-k-1)} = \frac{(R^2 - R^2_{restringido})/J}{(1-R^2/(n-k-1))} \sim F_{J,n-k-1}$$

donde $SSR_{restringido}$ y $R^2_{restringido}$ son, respectivamente, estimados por minimos cuadrados restringidos que minimizan $S(\beta)$ sujeto a $\mathbf{R}\beta = 0$.

2.2.5. Coeficiente de determinación

El coeficiente de determinación, denotado como R^2, mide la proporción de la variabilidad total en la variable dependiente que es explicada por el modelo de regresión. Su valor oscila entre 0 y 1, donde un valor cercano a 1 indica que el modelo explica bien la variabilidad de los datos, mientras que un valor cercano a 0 indica que el modelo explica muy poco de la variabilidad.

La bondad de ajuste de una regresión OLS puede ser medido como

$$R^2 = 1 - \frac{SSR}{SST} = \frac{SSE}{SST}$$

donde $SST = \sum_{i=1}^{n}(Y_i - \bar{Y})^2$ es la suma total de cuadrados de la desviación con respecto a la media de la variable dependiente y $SSR = \sum_{i=1}^{n} \hat{u}_i^2$ es la suma de los cuadrados de los residuales. $SSE = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})$ es llamado la suma de cuadrados explicada si la regresión contiene una constante y, por lo tanto, $\bar{Y} = \bar{\hat{Y}}$. R^2 se encuentra por definición entre 0 y 1 e informa la fracción de la variación de la muestra en Y que se explica por los regresores.

R^2 crece (por construcción) con cada regresor adicional (incluso con los irrelevantes) y por lo tanto en ocasiones no es un buen criterio para la selección de regresores. El R^2 ajustado es una versión modificada que no necesariamente crece con regresores adicionales:

$$R^2_{aj} = 1 - \frac{n-1}{n-k-1}\frac{SSR}{SST}$$

2.3. Ejemplo practico

Vamos a considerar la base de datos llamada "kc_house_data.csv"tomada de https://www.kaggle.com/datasets/harlfoxem/housesalesprediction que tiene la siguiente descripción: Este conjunto de datos contiene los precios de venta de casas en el condado de King, que incluye Seattle. Incluye casas vendidas entre mayo de 2014 y mayo de 2015. Es un excelente conjunto de datos para evaluar modelos de regresión simples.

Contiene las columnas:

- id
- date
- price (Numérica)
- bedrooms (Numérica)
- bathrooms (Numérica)
- sqft_living (Numérica)
- sqft_lot (Numérica)
- floors (Numérica)
- waterfront (Numérica)
- view (Numérica)
- condition (Numérica)

- grade (Numérica)
- sqft_above (Numérica)
- sqft_basement (Numérica)
- yr_built (Numérica)
- yr_renovated (Numérica)
- zipcode (Numérica)
- lat
- long (Numérica)
- sqft_living15 (Numérica)
- sqft_lot15 (Numérica)

Haremos una regresión lineal en `R` de la variable `price` sobre las otras variables de la base de datos menos de `date`, `zipcode` e `id` que no tienen sentido numérico. A continuación el código implementado:

```
# Importamos la base de datos
df <- read.csv("C:/Users/David Espitia/Desktop/kc_house_data.csv")

# Tomamos solo las variables de interes
df_interes <- df[, -c(1,2,17)]

# Nos aseguramos que todas las filas sean números
df_interes <- as.data.frame(lapply(df_interes, as.numeric))
# Quitamos los datos vacios
df_interes <- df_interes[, colSums(is.na(df_interes)) < nrow(df_interes)]

# Realizamos la regresión de price respecto a las otras variables
regresion <- lm(price ~ ., df_interes)

# Imprimimos la regresion
summary(regresion)
```

Los resultados son los siguientes:

```
Coefficients: (1 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.686e+07  1.595e+06 -23.105  < 2e-16 ***
bedrooms      -3.415e+04  1.903e+03 -17.945  < 2e-16 ***
bathrooms      4.216e+04  3.276e+03  12.868  < 2e-16 ***
sqft_living    1.467e+02  4.412e+00  33.236  < 2e-16 ***
sqft_lot       1.274e-01  4.827e-02   2.640   0.0083 **
floors         7.607e+02  3.606e+03   0.211   0.8329
waterfront     5.878e+05  1.748e+04  33.625  < 2e-16 ***
view           4.943e+04  2.146e+03  23.028  < 2e-16 ***
condition      3.103e+04  2.353e+03  13.186  < 2e-16 ***
grade          9.722e+04  2.167e+03  44.866  < 2e-16 ***
sqft_above     3.286e+01  4.390e+00   7.484 7.48e-14 ***
sqft_basement         NA         NA      NA       NA
yr_built      -2.456e+03  7.258e+01 -33.842  < 2e-16 ***
yr_renovated   2.153e+01  3.680e+00   5.850 4.97e-09 ***
lat            5.611e+05  1.055e+04  53.197  < 2e-16 ***
long          -1.170e+05  1.200e+04  -9.755  < 2e-16 ***
sqft_living15  2.743e+01  3.457e+00   7.935 2.20e-15 ***
sqft_lot15    -3.933e-01  7.379e-02  -5.330 9.94e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 202700 on 21596 degrees of freedom
Multiple R-squared:  0.6954,    Adjusted R-squared:  0.6952
F-statistic:  3082 on 16 and 21596 DF,  p-value: < 2.2e-16
```

Empecemos con los coeficientes. Cada coeficiente representa el cambio esperado en el precio de la casa por un aumento unitario en la variable predictora, manteniendo todas las demás variables constantes. Por ejemplo, el coeficiente para `bedrooms` es `-3.415e+04`, lo que significa que se espera que el precio de la casa disminuya en aproximadamente $34,150 por cada habitación adicional, manteniendo todas las demás variables constantes.

Ahora hablemos de la significancia individual de los coeficientes. Los valores de `Pr(>|t|)` nos indican la probabilidad de observar el valor del coeficiente que hemos observado, suponiendo que el coeficiente real es cero (es decir, no hay efecto). En este caso, la mayoria (exceptuando a `floors` y a `sqft_basement`) de los predictores tienen valores muy bajos de `Pr(>|t|)`, lo que indica que son estadísticamente significativos para predecir el precio de la casa.

Para evaluar la significancia conjunta del modelo, podemos mirar el estadístico F y su correspondiente valor p. El valor F es una relación entre la varianza explicada por el modelo y la varianza no explicada. En este caso, el valor del estadístico F es 3082 con un valor p muy cercano a cero, lo que indica que el modelo en su conjunto es significativo.

El coeficiente de determinación, R cuadrado (R-squared), es 0.6954, lo que significa que alrededor del 69.54 % de la variabilidad en el precio de la casa puede ser explicada por los predictores incluidos en el modelo. Sin embargo, dado que estamos usando múltiples predictores, es importante tener en cuenta el R cuadrado ajustado. El R cuadrado ajustado tiene en cuenta el número de predictores en el modelo y tiende a penalizar la adición de predictores irrelevantes. En este caso, el R cuadrado ajustado es 0.6952, muy similar al R cuadrado, lo que sugiere que el modelo no está sobreajustado y proporciona un buen ajuste a los datos.

Concluimos que este modelo parece ser significativo tanto individualmente como en su conjunto, y explica una cantidad considerable de variabilidad en el precio de la casa.

2.4. Conclusiones

La clusterización y la regresión lineal son herramientas fundamentales en el análisis de datos exploratorio y la construcción de modelos predictivos. Estas técnicas permiten identificar patrones, relaciones y tendencias en conjuntos de datos complejos, lo que facilita la toma de decisiones informadas en diversos campos. Existen diversos métodos de clusterización, tanto jerárquicos como no jerárquicos, cada uno con sus propias ventajas y aplicaciones específicas. Del mismo modo, el método de mínimos cuadrados ordinarios (OLS) es un enfoque ampliamente utilizado en la regresión lineal, ofreciendo una forma robusta de modelar la relación entre variables predictoras y una variable objetivo. Es crucial validar los resultados de los análisis mediante pruebas de significancia individual y conjunta. Estas pruebas proporcionan una evaluación cuantitativa de la relevancia de los predictores en la regresión lineal, así como la significancia global del modelo. Al aplicar estas técnicas en un contexto específico, es importante considerar cuidadosamente factores como la selección de variables, la interpretación de resultados y la validez de los supuestos subyacentes. La elección adecuada de métodos y la comprensión de sus limitaciones son clave para obtener conclusiones válidas y relevantes.

Referencias

[1] IBM. *Aprendizaje supervisado*. URL: https://www.ibm.com/mx-es/topics/supervised-learning.

[2] Alfonso Novales. *Análisis de Regresión*. @Copyright Alfonso Novales. 20 de Septiembre de 2010: Departamento de Economía Cuantitativa, Universidad Complutense, 2010.

[3] Sergio A. Pernice. «Descomposición en valores singulares y análisis de factores en ciencias humanas y sociales». En: *Revista de Métodos Cuantitativos para la Economía y la Empresa* 37 (2024), págs. 1-29. DOI: 10.46661/revmetodoscuanteconempresa.8004.

[4] ALVIN C. RENCHER. *Methods of Multivariate Analysis*. Second edition. ISBN: 0-471-41889-7. 2002. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118391686.

[5] Kurt Schmidheiny. *Short Guides to Microeconometrics: The Multiple Linear Regression Model*. University of Basel, 2023.

[6] *Serrano.Academy en Español. Descomposicion en valores singulares [Vídeo]. YouTube*. URL: https://www.youtube.com/watch?v=wp3AJ0ZJCjw.

[7] Universidad de Valencia. *Introducción al Análisis de Regresión*. URL: https://www.uv.es/ceaces/base/regresion/intro.htm.

[8] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. URL: https://egrcc.github.io/docs/math/all-of-statistics.pdf.