Sistemas de Aprendizaje Automático

Conforme a contenidos del «Curso de Especialización en Inteligencia Artificial y Big Data»

Universidad de Castilla-La Mancha

Escuela Superior de Informática
Ciudad Real

Índice general

Listado de acrónimos

AMI	Adjusted Mutual Information
AUC	Area under the ROC Curve
CVI	Clustering Validity Index
DBSCAN	Density-based spatial clustering of applications with noise
MAE	Mean Absolute Error
MSE	Mean Squared Error
MSLE	Mean Squared Logaritmic Error
MAPE	Mean Absolute Percentage Error
MAPD	Mean Absolute Percentage Deviation
NMI	Normalized Mutual Information
ROC	Receiver Operating Characteristic

Capítulo 9
Validación

9.1. Validación en Problemas de Clasificación

9.1.1. Clasificación Binaria

En una tarea de clasificación binaria, los términos ”positivo” y ”negativo” se refieren a la predicción del clasificador, y los términos ”verdadero” y ”falso” se refieren a si esa predicción corresponde al juicio externo (a veces conocido como ”observación”). Dadas estas definiciones, se puede formular lo siguiente;

Figura 9.1: Resultados Clasificación Binaria.

En este contexto, podemos definir las nociones de precisión, exhaustividad (recall) y métrica F de la siguiente forma:

$$\text{precision} = \frac{tp}{tp + fp}$$

$$\text{recall} = \frac{tp}{tp + fn}$$

$$F_\beta = (1 + \beta^2)\frac{\text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}}.$$

9.1.2. Curva ROC

Una curva ROC (Receiver Operating Characteristic) es una representación gráfica que ilustra la relación entre la sensibilidad y la especificidad de un sistema clasificador para diferentes puntos de corte. Fueron desarrolladas en los años 50 para analizar señales con ruido con el fin de caracterizar el compromiso entre aciertos y falsas alarmas. Son utilizadas para comparar visualmente distintos modelos de clasificación [KH09].

La curva ROC significa el ajuste entre la FPR (especificidad) y la TRP (sensibilidad). Siendo definidas estas a partir de los conceptos de Verdaderos positivos (TP), Falsos Positivos (FP), Verdaderos Negativos (TN) y Falsos Negativos (FN).

$$TPR = \frac{TP}{(TP + FN)}$$

$$FPR = \frac{FP}{(FP + TN)}$$

El clasificador en la esquina superior izquierda especifica que el rendimiento es mejor. Como norma, recibirá puntos de un clasificador aleatorio entre la diagonal. Se puede decir que la prueba es menos precisa si la curva está más cerca de los 45 grados del espacio ROC.

Su construcción sigue el siguiente algoritmo:

1. Se usa un clasificador que prediga la probabilidad de que un ejemplo E pertenezca a la clase positiva P(+|E)
2. Se ordenan los ejemplos en orden decreciente del valor estimado P(+|E)
3. Se aplica un umbral para cada valor distinto de P(+|E), para el que se cuenta el número de TP, FP, TN y FN.

El área bajo la curva AUC es una medida de la precisión del clasificador, cuanto más cerca se encuentre de la diagonal (área cercana a 0.5), menos preciso será el modelo. Un modelo “perfecto” tendrá área 1.

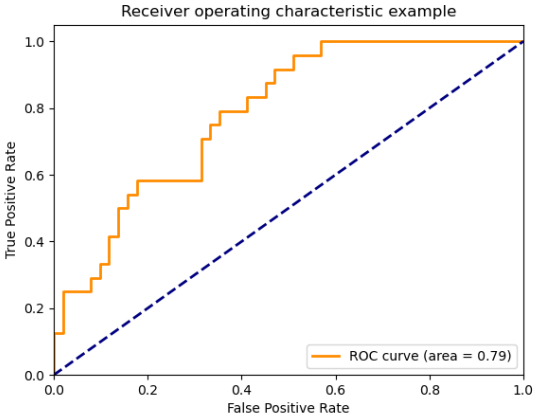

Figura 9.2: Curva ROC.

9.1.3. Matriz de Confusión

La Matriz de Confusión sirve para evaluar la precisión de la clasificación en la que cada fila corresponde a la clase esperada. Por definición, la entrada en una matriz de confusión i, j es el número de observaciones que realmente están en el grupo i , pero que se predice que están en el grupo j [BVC09]. He aquí un ejemplo:

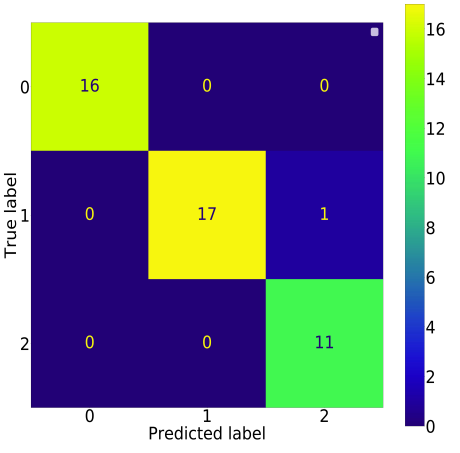

Figura 9.3: Ejemplo de Matriz de Confusión.

9.1.4. Accuracy

El *accuracy* o exactitud representa la fracción o el recuento de predicciones correctas. En la clasificación multi-clase, la función devuelve la precisión del subconjunto. Si todo el conjunto de etiquetas predichas para una muestra coincide estrictamente con el conjunto verdadero de etiquetas, entonces la precisión del subconjunto es $1, 0$; en caso contrario, es $0, 0$.

En la clasificación multietiqueta, la función devuelve la precisión del subconjunto. Si todo el conjunto de etiquetas predichas para una muestra coincide estrictamente con el conjunto verdadero de etiquetas, entonces la precisión del subconjunto es 1,0; en caso contrario, es 0,0.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado y Var la varianza entendida como el cuadrado de la desviación típica, entonces la varianza explicada se estima como sigue:

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} If(\hat{y}_i = y_i)$$

9.2. Validación en Problemas de Regresión

Dado que nuestro modelo producirá una salida dada cualquier entrada o conjunto de entradas, podemos cotejar estas salidas estimadas con los valores reales que intentamos predecir. Llamamos residuo o error a la diferencia entre el valor real y la estimación del modelo. Podemos calcular el residuo para cada punto de nuestro conjunto de datos, y cada uno de estos residuos será útil en la evaluación. Estos residuos desempeñarán un papel importante a la hora de juzgar la utilidad de un modelo.

Si el valor agregado de nuestro error es pequeño, implica que el modelo es bueno, por el contrario, si estos residuos son generalmente grandes, implica que el modelo es un mal estimador.

En resumen, la calidad de un modelo de regresión consiste en lo bien que coinciden sus predicciones con los valores reales. Con el fin de evaluar esta calidad se usarán medidas de error lo cual nos permitirá comparar regresiones con otras regresiones con diferentes parámetros. Estas métricas son resúmenes breves y útiles de la calidad de nuestros modelos.

Varianza Explicada

La varianza explicada mide la proporción en que un modelo matemático explica la variación (dispersión) de un conjunto de datos determinado.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado y Var la varianza entendida como el cuadrado de la desviación típica, entonces la varianza explicada se estima como sigue:

$$explained\_variance(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

El mejor valor posible es 1,0, valores pequeños representan malos resultados.

Error Máximo

Representa el error residual máximo, una métrica que captura el peor caso de error entre el valor predicho y el valor real. En un modelo de regresión de salida única perfectamente ajustado, el valor de esta métrica sería 0 en el conjunto de entrenamiento y, aunque esto sería muy poco probable en el mundo real, esta métrica muestra el grado de error que tenía el modelo cuando se ajustó.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado, entonces el error máximo se formularía como sigue:

$$\text{Max Error}(y, \hat{y}) = max(|y_i - \hat{y}_i|)$$

Error Absoluto Medio

El error absoluto medio es un promedio de los errores absolutos.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado, el error absoluto medio sobre $n_samples$ elementos estaría definido como:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i| .$$

Y su utilización en Python sería la siguiente:

```
from sklearn.metrics import max_error
mean_absolute_error(y_true, y_pred, multioutput='raw_values')
```

El MAE es la métrica más fácil de interpretar, ya que sólo se observa la diferencia absoluta entre los datos y las predicciones del modelo. Dado que utilizamos el valor absoluto del residuo, el MAE no indica un rendimiento inferior o superior del modelo (si el modelo se ajusta o no a los datos reales). Cada residuo contribuye proporcionalmente a la cantidad total de error, lo que significa que los errores más grandes contribuirán linealmente al error global. De esta forma, un MAE pequeño sugiere que el modelo es excelente en la predicción, mientras que un MAE grande sugiere que el modelo puede tener problemas. Un MAE de 0 significa que el modelo es un predictor perfecto de los resultados.

Utilizar el valor absoluto del error implica el tratar por igual a todos los errores y puede ser necesario el dar más importancia a los valores atípicos o extremos.

Error Cuadrático Medio

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado, el error cuadrático medio MSE sobre $n_samples$ elementos estaría definido como:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Y su utilización en Python sería la siguiente:

```
from sklearn.metrics import max_error
mean_squared_error(y_true, y_pred)
```

Al elevar la diferencia entre las magnitudes al cuadrado, hace que el error crezca cuadráticamente en MSE. Esto significa, en que los valores atípicos en nuestros datos contribuirán a un error total mucho mayor en el MSE que en el MAE. Del mismo modo, nuestro modelo se verá más penalizado por hacer predicciones que difieran mucho del valor real correspondiente.

Error cuadrático Logarítmico Medio

Corresponde al valor esperado del error o pérdida logarítmica (cuadrática) al cuadrado.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado, el error cuadrático logarítmico medio MSLE sobre $n_samples$ elementos estaría definido como:

$$\text{MSLE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (\log_e(1 + y_i) - \log_e(1 + \hat{y}_i))^2.$$

Esta métrica es la mejor para utilizar cuando los objetivos tienen un crecimiento exponencial, como los recuentos de población, las ventas medias de una mercancía durante un periodo de años, etc. Tenga en cuenta que esta métrica penaliza más una estimación subestimada que una sobreestimada.

```
from sklearn.metrics import mean_squared_log_error
mean_squared_log_error(y_true, y_pred)
```

Error Porcentual Medio Absoluto

El error porcentual medio absoluto MAPE, también conocido como desviación porcentual media absoluta MAPD, es una métrica de evaluación para los problemas de regresión. La idea de esta métrica es ser sensible a los errores relativos. Por ejemplo, no cambia por un escalado global de la variable objetivo.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado, el error porcentual medio absoluto MSLE sobre $n_samples$ elementos estaría definido como:

$$\text{MAPE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{|y_i - \hat{y}_i|}{max(\epsilon, |y_i|)}$$

where ϵ es un número arbitrario, pequeño pero estrictamente positivo, para evitar resultados NaN cuando y es cero.

```
from sklearn.metrics import  mean_absolute_percentage_error
 mean_absolute_percentage_error(y_true, y_pred)
```

El MAPE repersenta la distancia que separa las predicciones del modelo de sus resultados correspondientes en promedio. Al igual que el MAE, el MAPE también tiene una interpretación clara. Sin embargo, a pesar de todas sus ventajas, el uso del MAPE es más limitado que el del MAE. El MAPE puede crecer inesperadamente si los valores reales son excepcionalmente pequeños. Por último, el MAPE está sesgado hacia las predicciones que son sistemáticamente menores que los propios valores reales. Es decir, el MAPE será menor cuando la predicción sea menor que la real en comparación con una predicción que sea mayor en la misma medida.

Coeficiente de Determinación R^2

El coeficiente de determinación, normalmente denominado R^2 representa la proporción de la varianza (de y) que ha sido explicada por las variables independientes del modelo. Proporciona una indicación de la bondad del ajuste y, por tanto, una medida de la probabilidad de que las muestras no vistas sean predichas por el modelo, a través de la proporción de varianza explicada.

Como dicha varianza depende del conjunto de datos, R^2 puede no ser significativamente comparable entre diferentes conjuntos de datos. La mejor puntuación posible es $1,0$ y puede ser negativa (porque el modelo puede ser arbitrariamente peor). Un modelo constante que siempre predice el valor esperado de y, sin tener en cuenta las características de entrada, obtendría una puntuación R^2 de $0,0$.

Si $\hat{y}_i$ es el valor predico del elemento i-th, e y_i el valor esperado, el coeficiente de determinación R^2 estaría definido como:

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

donde

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

y

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}\epsilon_i^2$$

Dado que el R^2 puede incrementarse añadiendo un mayor número de variables y puede conducir a un ajuste excesivo del modelo (*overfitting*), se puede recurrir al R^2 ajustado.

Aunque tanto el R^2 como el R^2 ajustado dan una idea de los puntos de datos que caen en la línea de regresión, la única diferencia entre ellos es que el R^2 ajustado encuentra el porcentaje de variación explicado por la variable independiente que realmente afecta a la variable dependiente y el R^2 asume que cada variable explica la variación en la variable dependiente.

El R^2 ajustado es la versión modificada de la R-cuadrada, cuyo valor aumenta sólo cuando la variable en el modelo le añade valor, por lo que cuanto menos útil sea la variable presente en el modelo, menor será el valor del R^2 ajustado y mayor será el valor de la R^2.

9.3. Validación en Problemas de Análisis Clúster

Los índices de validez de los clústeres se utilizan para validar los resultados de la agrupación y para encontrar un conjunto de clústeres que se ajuste mejores particiones naturales para un conjunto de datos determinado. La mayoría de los índices de validez existentes dependen considerablemente del número de objetos de datos en los clusters, de los centroides de los clusters y de los valores medios. De esta forma, tienen una tendencia a ignorar los clusters pequeños y los clusters con baja densidad [TSK16].

La presencia de una gran variabilidad en las formas geométricas de los clusters densidades y tamaños y el número de grupos que no siempre puede conocer a priori, son dificultades importantes a la hora de agrupar. Muchos algoritmos de clustering (como k-means y fuzzy c-means) requieren que el usuario defina previamente el número de grupos antes del proceso de clustering. Sin embargo, a veces es imposible conocer el número de clusters de antemano. Los resultados de la agrupación dependen de la elección del número de grupos. La determinación del número adecuado de grupos y la validez de la partición obtenida son dos problemas fundamentales en el clustering. Encontrar el número de grupos que mejor se ajuste a la partición natural para un conjunto de datos dado es difícil, ya que para un mismo conjunto de datos existen varias particiones dependiendo del nivel de detalle.

Los índices de validez suelen utilizarse para obtener el número óptimo de grupos. Esto requiere que un algoritmo de clustering se ejecute varias veces, con un número diferente de grupos como objetivo en cada ejecución. Otra alternativa para identificar el número correcto de de clusters es mejorar la función de optimización y descubrir el número de clusters dinámicamente durante la ejecución del algoritmo de clustering que satisface la nueva función de optimización.

La mayoría de los índices de validación de validación propuestos durante las últimas décadas se han centrado en la compacidad y la separación. La separación es una medida del aislamiento de los clusters entre sí entre sí y la compacidad es una medida de la proximidad de los objetos de datos dentro de un clúster. Un valor bajo de varianza es un indicador de cercanía. Los miembros de cada clúster deben estar lo más cerca posible unos de otros posible y los clústeres deben estar muy separados. La mayoría de las medidas de medidas de validez tienen la tendencia a ignorar los clusters con baja densidad y no son eficientes en la validación de particiones que tienen diferentes tamaños y densidades.

9.3.1. Validación Interna

Un índice de validez interna de los clusters CVI tiene como objetivo medir lo bien que una partición determinada de un conjunto de datos refleja la estructura subyacente del dominio modelado.

Coeficiente Silhouette

El Coeficiente Silhouette representa una de las métricas de validez interna más utilizadas dentro del aprendizaje automático. Una mayor puntuación de este coeficiente se relaciona con un modelo con clusters mejor definidos. El Coeficiente Silhouette se define para cada elemento y se compone de dos puntuaciones [Kog07].

- **a**: La distancia media entre un elemento y todos los demás elementos de la misma clase.
- **b**: La distancia media entre una elementos y todos los demás elementos del siguiente clúster más cercano.

El Coeficiente de Silhouette s para una sola muestra se da entonces como:

$$s = \frac{b - a}{max(a, b)}$$

El Coeficiente Silhouette para un conjunto de muestras se da como la media del Coeficiente Silhouette para cada muestra.

La puntuación está limitada entre -1 para una agrupación incorrecta y +1 para una agrupación muy densa. Las puntuaciones en torno a cero indican que los clusters se solapan. La puntuación es más alta cuando los clusters son densos y están bien separados, lo que se relaciona con un concepto estándar de clúster.

El principal inconveniente que presenta este coeficiente es su sesgo a favor de las agrupaciones convexas lo cual perjudica a los grupos obtenidos mediante criterios de densidad como los de DBSCAN.

El Coeficiente Silohuette se utiliza habitualmente para elegir el valor óptimo del número de clusters para un algoritmo k-means

Listado 9.1: Cálculo del coeficiente silhouette

```
1 import numpy as np
2 from sklearn.cluster import KMeans
3 kmeans_model = KMeans(n_clusters=5)
4 kmeans_model.fit(X)
5 labels = kmeans_model.labels_
6 metrics.silhouette_score(X, labels, metric='euclidean')
```

Calinski-Harabasz

El índice Calinski-Harabasz, también conocido como Criterio de Relación de Varianza- puede utilizarse para evaluar un resultado de clustering donde una puntuación Calinski-Harabasz más alta se relaciona con un modelo con grupos mejor definidos [CJ19].

El índice es el cociente de la suma de la dispersión entre clusters y de la dispersión dentro de los clusters para todos los clusters (donde la dispersión se define como la suma de las distancias al cuadrado):

Para un conjunto de datos E de tamaño n_e que se ha agrupado en k clusters entonces:

$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1}$$

donde $tr(B_k)$ es la traza (suma de sus elementos diagonales) de la matriz de dispersión del grupo y $tr(W_k)$ es la traza de la matriz de dispersión del cluster, siendo: siendo

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

con C_q el conjunto de elementos del cluster q, c_q el centroide del cluster q, c_E el centro del conjunto de datos E y n_q el número de elementos en el cluster q

La puntuación, qué es rápida de calcular, es mayor cuando los clusters son densos y están bien separados, lo que se relaciona con un concepto estándar de cluster. Esto hace que sea generalmente más alto para los clusters convexos que para otros conceptos de clusters, como los clusters basados en la densidad tal y como sucedía con silhouette.

Listado 9.2: Cálculo del coeficiente silhouette

```
import numpy as np
from sklearn.cluster import KMeans
kmeans_model = KMeans(n_clusters=5)
kmeans_model.fit(X)
labels = kmeans_model.labels_
metrics.calinski_harabasz_score(X, labels)
```

9.3.2. Validación Externa

Estas medidas se calculan haciendo coincidir la estructura de los clusters con alguna clasificación predefinida de instancias en los datos (*ground truth*).

Rand Index

Compara los dos clusters y trata de encontrar la proporción de observaciones coincidentes y no coincidentes entre dos estructuras de clustering (C y K). Su valor se sitúa entre 0 y 1.

Si C es una asignación a grupos que funciona como *ground truth* y K es el clustering entonces se pueden definir:

- a: Número de pares de elementos que están en la mismo grupo en C y en el mismo conjunto en K.
- b: Número de pares de elementos que están en diferentes grupos en C y en diferentes grupos en K

El Rand Index no ajustado vendría dado por:

$$\text{RI} = \frac{a+b}{C_2^{n_{samples}}}$$

where $C_2^{n_{samples}}$ es el número total de pares en el conjunto de datos.

Lo cual se calcularía con el código:

```
from sklearn import metrics
metrics.rand_score(labels_true, labels_pred)
```

Dado que el Rand Index no garantiza que una distribución aleatoria tenga un valor cercano a 0 entonces el RI se ajusta en el ARI descontando el RI esperado en una distribución aleatoria:

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\text{máx}(\text{RI}) - E[\text{RI}]}$$

A diferencia de otros índices, el Rand Index (ajustado o no ajustado) requiere el conocimiento del *ground truth* que casi nunca está disponible en la práctica o requiere la asignación manual por parte de anotadores humanos (como en el aprendizaje supervisado). Sin embargo, puede ser útil en un entorno puramente no supervisado como bloque de construcción para un índice de consenso que puede utilizarse para la selección de modelos de agrupación.

```
from sklearn import metrics
metrics.adjusted_rand_score(labels_true, labels_pred)
```

Homogeneidad, Exhaustividad y V-Score

Siempre contando con la disponibilidad de la referencia de los clusters reales (*ground trouth*) es posible definir métricas utilizando el análisis de entropía condicional. En [RH07] se definen los siguientes dos objetivos deseables para cualquier asignación de clusters;

- **homogeneidad**: cada clúster contiene sólo miembros de una única clase
- **exhaustividad**: todos los miembros de una clase determinada se asignan al mismo clúster.

Estos dos conceptos pueden convertirse en medidas de calidad de los clusters, ambas estarían limitadas por debajo de 0 y por encima de 1,0. Cuanto mayor fuera el valor obtenido mejor. Una de las grandes ventajas de estas métricas es que no hacen ninguna suposición sobre la estructura de los clusters.

Ambas métricas vienen formalizadas por las siguientes fórmulas:

$$h = 1 - \frac{H(C|K)}{H(C)}$$

$$c = 1 - \frac{H(K|C)}{H(K)}$$

donde $H(C|K)$ es la entropía condicional de las clases dadas las asignaciones de los clusters y está dada por:

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n}\cdot\log\left(\frac{n_{c,k}}{n_k}\right)$$

y $H(C)$ es la entropía de las clases y viene dada por:

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log\left(\frac{n_c}{n}\right)$$

con n el número total de elementos, n_c y n_k el número de elementos pertenecientes a la clase c y al cluster k, y finalmente $n_{c,k}$ el número de elementos de la clase c asignados al cluster k.

La entropía condicional de los clusters dada la clase $H(K|C)$ y la entropía de los clusters $H(K)$ se definen de forma simétrica.

Con el fin de equilibrar los dos conceptos se propone la **V-measure** que es la media armónica entre estas dos medidas.

$$v = 2 \cdot \frac{h \cdot c}{h + c}$$

Este indicador ofrece una explicación bastante intuitiva del resultado, un resultado con un baja V-measure analizarse cualitativamente en términos de homogeneidad y exhaustividad para conocer mejor qué "tipo"de errores se cometen en la asignación.

Dependiendo del número de elementos, clusters y clases objetivo, un resultado completamente aleatorio no siempre producirá los mismos valores de homogeneidad, exhaustividad y V-measure. En particular, no resultarán puntuaciones nulas, especialmente cuando el número de grupos sea grande. Este problema puede ignorarse con seguridad cuando el número de muestras es superior a mil y el número de grupos es inferior a 10. Para volúmenes de datos más pequeños o un mayor número de clusters es más seguro utilizar un índice ajustado como el Rand Index ajustado (ARI).

Listado 9.3: Función para el cálculo de métricas de Clustering basados en ground truth

```
1
2  def calc_metrics (data, labels_pred, labels_true):
3
4      print("Homogeneity: %0.3f" % metrics.homogeneity_score(labels_true, labels_pred))
5      print("Completeness: %0.3f" % metrics.completeness_score(labels_true, labels_pred))
6      print("V-measure: %0.3f" % metrics.v_measure_score(labels_true, labels_pred))
7      print("Adjusted Rand Index: %0.3f"
8            % metrics.adjusted_rand_score(labels_true, labels))
9      print("Adjusted Mutual Information: %0.3f"
10         % metrics.adjusted_mutual_info_score(labels_true, labels))
11     print("Silhouette Coefficient: %0.3f"
12           % metrics.silhouette_score(data, labels))
```

Mutual Information Index

La información mutua es una función que mide la concordancia de las dos asignaciones, ignorando las permutaciones. Existen dos versiones normalizadas de esta medida, la información mutua normalizada NMI y la información mutua ajustada AMI. El NMI se utiliza a menudo en la literatura, mientras que el AMI se propuso más recientemente y se normaliza con respecto al azar

Las asignaciones de etiquetas aleatorias (uniformes) tienen una puntuación AMI cercana a 0,0 para cualquier valor de número de grupos y de elementos. Los valores cercanos a cero indican dos asignaciones de etiquetas que son en gran medida independientes, mientras que los valores cercanos a uno indican un acuerdo significativo. Además, un AMI de exactamente 1 indica que las dos asignaciones de etiquetas son iguales (con o sin permutación).

Para la información mutua normalizada y la información mutua ajustada, el valor normalizador suele ser alguna media generalizada de las entropías de cada agrupación. Existen varias medias generalizadas y no hay reglas firmes para preferir una sobre las otras. La decisión depende en gran medida de cada campo.

Bibliografía

[BVC09] Luis Paulo Vieira Braga, Luis Iván Ortiz Valencia, and Santiago Segundo Ramírez Carvajal. *Introducción a la Minería de Datos*. Editora E-papers, 2009.

[CJ19] Gustavo Adolfo Urbina Cortés and Sergio Arturo Bárcena Juárez. *Herramientas de Análisis Multivariado para la Investigación Social. Una guía práctica en STATA*. Tecnológico de Monterrey/Porrúa, 2019.

[KH09] Wojtek J Krzanowski and David J Hand. *ROC curves for continuous data*. Crc Press, 2009.

[Kog07] Jacob Kogan. *Introduction to clustering large and high-dimensional data*. Cambridge University Press, 2007.

[RH07] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[TSK16] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.