TRANSFORMACIÓN Y NORMALIZACIÓN

Carlos Abad
Universidad Técnica Estatal de Quevedo
Facultad de Ciencias de la Ingeniería
Quevedo, Ecuador
martin.abad2016@uteq.edu.ec

Resumen: El trabajo consiste en la utilización de los métodos de transformación, normalización, discretización y análisis correspondencia. Detallando una breve descripción de cada una de las antes mencionadas. Se utilizó el conjunto de datos Basketball, para aplicar cada uno de los conceptos prácticos de estos métodos de minería de datos, realizados en distintas herramientas para el análisis de datos como Weka, R, Knime.Mostrando los resultados de las aplicaciones de cada uno de ellos en tablas e imágenes.

Palabras Claves: Discretización, transformación, normalización y análisis correspondía principal.

I. Introducción

La eficacia de los algoritmos de extracción de conocimiento depende en gran medida de la calidad de los datos, la cual puede ser garantizada por los algoritmos de reprocesamiento [1]. Sin embargo, en esta era de Big Data, los algoritmos depreprocesamiento tienen dificultades para trabajar con tal cantidad de datos, siendo necesario nuevos modelos que mejoren su capacidad de escalado.

La necesidad de procesar y extraer conocimiento valioso de tal inmensidad de datos se ha convertido en un desafío considerable para científicos de datos y expertos en la materia [2]. El valor del conocimiento extraído es uno de los aspectos esenciales de Big Data, como la discretización y normalización que son encargados de transformar atributos continuos usando intervalos discretos, mientras que la normalización realiza un ajuste en la distribución, y el análisis de correspondencia principal (PCA) con análisis de correspondía (CA) reducen la dimensión del problema [3].

II. Revisión literaria

A. Minería de datos

La DM integra técnicas de análisis de datos y extracción de modelos. Se basa en varias disciplinas, algunas de ellas más tradicionales como la estadística y el aprendizaje automático, se diferencia de ellas en la orientación más hacia el fin que hacia los medios [4]. La minería de datos tiene como objetivo analizar los datos para extraer conocimiento.

Este conocimiento puede ser en forma de relaciones, patrones o reglas inferidos de los datos y (previamente) desconocidos, o bien en forma de una descripción más concisa (es decir, un resumen de estos) [4]. Estas relaciones o resúmenes constituyen el modelode los datos analizados.

B. Transformación

Puede que el algoritmo a usar no pueda manejar los datos, que la forma de los datos no sea regular o que los datos no sean lo suficientemente específicos [5].

1. Normalización
2. Suavizado
3. Agregación
4. Generalización
5. Construcción de atributos

C. Discretización

Es el proceso que transforma datos cuantitativos en datos cualitativos. Algunos algoritmos de clasificación aceptan solo atributos categóricos. El proceso de aprendizaje frecuentemente es menos eficiente y efectivo cuando los datos tienen solamente variables cuantitativas [6].

1. Ordenamiento
2. Evaluación
3. Separación/Unión
4. Parada

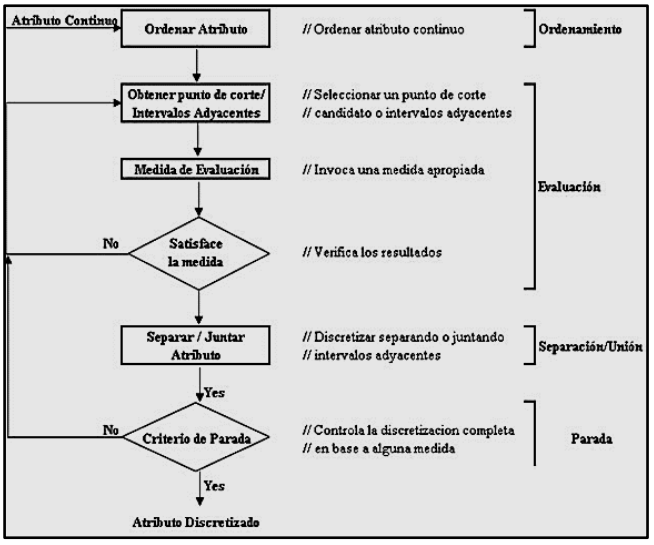


***Figura 1**: Algoritmo del proceso de discretización*

D. Normalización

Normalizar datos es una técnica que se aplica a un conjunto de datos para reducir su redundancia. El objetivo principal de esta técnica es asociar formas similares a los mismos datos en una única forma de datos. Esto es, en cierto modo, cogiendo datos específicos como “Número”, “Num.”, “Nro.”, “Nº” o “#” y normalizando a “Número” en todos los casos [7]. Algunas técnicas de normalización son las siguientes:

1. Normalización min-max
2. Normalización zscore
3. Normalización por escala decimal
4. Suavizado
5. Agregación

E. Reducción de la dimensionalidad

Los métodos de reducción de dimensionalidad son técnicas estadísticas que mapean el conjunto de los datos a subespacios derivados del espacio original, de menor dimensión, que permiten hacer una descripción de los datos a un menor costo. Estas técnicas cobran importancia ya que muchos algoritmos de diversos campos tales como análisis numérico, aprendizaje automático o minería de datos suelen degradar su rendimiento cuando se usan con datos de alta dimensionalidad [8].

1. Step-wise forward selection
2. Step-wise backward elimination
3. Combinación de las 2 anteriores
4. Inducción de árbol de decisión

F. Análisis de correspondencias principales(PCA)

El análisis de correspondencia principal es una técnica estadísticade síntesis de la información, basada en la reducción de la dimensión (número de variables) del problema. Es decir, ante un banco de datos con muchas variables relacionadas, el objetivo será reducirlas a un menor número perdiendo la menor cantidad deinformación posible. Para ello se debe intentar eliminar las posiblesredundancias entre ellas, dando lugar a nuevos componentes principales o factores, que serán una combinación lineal de lasvariables originales, además serán independientes entre sí [9].

1. Análisis de la matriz de correlaciones
2. Selección de los factores
3. Análisis de la matriz factorial
4. Interpretación de los factores
5. Cálculo de las puntuaciones factoriales.
6. Resultado en gráfica

G. Análisis de correspondencia (CA)

El análisis de correspondencia es una técnica descriptiva conceptualmente similar al PCA, con la diferencia de que en el análisis de correspondencias los datos se escalan de modo que filas y columnas se tratan de modo equivalente, y las variables con las que se trabaja son cualitativas o nominales [9].

III. Metodología/Procedimiento

El trabajo fue realizado de manera conjunta entre todos los autores mediante el uso de la herramienta digital de Google Meet, la cual semuestra en el Anexo 1. A cada uno de los autores se les fue asignada tareas específicas para elaborar el trabajo, así mismo como tareas que se deben realizar en conjunto como la revisión bibliográfica útil para todo este proceso lo cual, todo está detallado en el Anexo 2.

Se comenzó con la búsqueda y análisis de elementos bibliográficos para conceptualizar los temas frente a este trabajo, en este caso se utilizaron herramientas útiles que sirvieron para realizar la práctica que fueron: Weka, Rstudio, Knime y Rapidminer, mismas que se ocuparon para analizar el conjunto de datos Basketball con cada uno de los métodos vistos como son: transformación, normalización, discretización, reducción de la dimensionalidad, análisis de correspondencias principales y análisis de correspondencia.

Se elaboró un cuadro comparativo y de características en la tabla 1, tabla 2, tabla 3 y tabla 4 en el apartado de Resultados. Así mismo, se detalla en el literal B la comparación de la calidad de predicción con la normalización y en el literal C se muestra la representación los datos en baja dimensión con el método CA.

IV. Resultados

A. Diferencias y similitudes entre los métodos.

Técnicas/ Métodos	Diferencias	Similitudes
Fayyad and Irani	Este método se diferencia de los otros métodos porque se encarga de buscar la mejor partición de cada uno de los atributos.	Este Fayyad and Irani y el MDL son métodos de discretización de entropía.
ChiMerge	-Este método utiliza un enfoque de juntas intervalos. -Ordena los valores y después usa la prueba X 2 para determinar cuando los intervalos deben ser combinados.	Es un método de discretización supervisado.
Discretización 1R (One rule)	-Este método ordena los datos y los divide en intervalos disjuntos. - Cada intervalo contiene un mínimo de seis instancias a excepción del intervalo final que puede contener las instancias restantes que no hayan sido agrupadas en ningún intervalo.	Es un método de discretización supervisado que usa intervalos de valores.
Discretizador de frecuencia	-El dominio de cada atributo se divide en n partes de tamaño igual.	Este método tiene como similitud con el método tamaño uniforme que dividen las variables en k intervalos.
Transformación	Manipula datos no regulares	Modifica los datos acordes al método de discretización
Discretización	Maneja variables cuantitativas	Es un método de la transformación
Reducción de la dimensionalidad	Reduce el número de variables en una colección de datos	Este método tiene como similitud con el metodo de frecuencia que dividen las variables en k intervalos
Análisis de correspondencias principales (PCA)	Es una técnica descriptiva	Relaciona subyacentes de los datos
Análisis de correspondencia (CA)	Trabaja con variables cualitativas escalandode modo en filas y columnas	Similitud con el funcionamiento de PCA

Tabla 1: *Comparación entre diferencia y similitudes de las técnicas de transformación y normalización*

Software	Características	Lenguaje
Weka	Muchos métodos de clasificación	Java
R studio	No tiene limitadas sus funciones, es una herramienta estadística	S

Knime	Es de acceso a los análisis predictivos	Java

Tabla 2: *Características y lenguaje de los softwares utilizados.*

Algoritmo	Tipo	Características
Intervalos de igual longitud	No supervisado, global y estático.	Divide el rango de cada variable en k intervalos.
Intervalos de igual frecuencia	No supervisado, global y estático.	Cada intervalo tendrá el mismo número de instancias o similar.
Metodo 1R	Supervisado	Cada intervalo tiene un número mínimo de instancias (6).
Basada en Entropia	Supervisado, global y estático.	Utilizan la información existente de la clase en los datos.
ChiMerge	Supervisado	Utiliza un enfoque de juntar intervalos.
Binning	No supervisado, global y estático	Crea bins de tal manera que el rango de todos los bins es (casi) igual
Tamaño	No supervisado, global y estático	Crea bins de tal manera que cada bin tiene un tamaño especificado por el usuario

Tabla 3: *Descripción de algoritmo, tipo y características*

Características	RSTUDIO	WEKA	RAPIDMINER	KNIME
Archivos permitidos	xlsx, txt, csv, DB	csv, databa	xlsx, txt, csv, DB	xlsx, txt, csv, DB
Igual longitud	SI	SI	SI	
Igual frecuencia	SI	SI	SI	
Metodo 1R	SI			
Entropía	SI	SI	SI	
Tamaño			SI	
CAIM Binner				SI
ChiMerge	SI			
Librerías requeridas	dprep o discretization	discretize (defecto)	(defecto)	(defecto)

Tabla 4:*Características de los datos permitidos en Rstudio, Weka,RapidMiner y Knime*

B. Normalización Z-SCORE

Los valores V son normalizados en base a la media y desviación estándar.
V' = (V - mean) / std
Este método trabaja bien en los casos en que no se conoce el máximo y mínimo de los datos de entrada o cuando existen outliers que tienen un gran efecto en el rango de los datos.

```
basketball <- read.csv(file.choose())
basketball01=basketball[1:10,1:3]
basketball01

##    assists_per_minuteReal heightInteger time_playedReal
## 1                  0.0888           201           36.02
## 2                  0.1399           198           39.32
## 3                  0.0747           198           38.80
## 4                  0.0983           191           40.71
## 5                  0.1276           196           38.40
## 6                  0.1671           201           34.10
## 7                  0.1906           193           36.20
## 8                  0.1061           191           36.75
## 9                  0.2446           185           38.43
## 10                 0.1670           203           33.54
```

Ilustración 1. Conjunto de datos Basketball.

```
basketball01.zcs=scale(basketball01, center = TRUE, scale=TRUE)
basketball01.zcs

##    assists_per_minuteReal heightInteger time_playedReal
## 1             -0.98253089    0.94002238      -0.5230791
## 2             -0.01083884    0.40793424       0.9070460
## 3             -1.25064944    0.40793424       0.6816930
## 4             -0.80188364   -0.83360476       1.5094320
## 5             -0.24472949    0.05320881       0.5083445
## 6              0.50638277    0.94002238      -1.3551519
## 7              0.95324702   -0.47887933      -0.4450723
## 8             -0.65356274   -0.83360476      -0.2067181
## 9              1.98008402   -1.89778104       0.5213456
## 10             0.50448122    1.29474781      -1.5978398
## attr(,"scaled:center")
## assists_per_minuteReal          heightInteger        time_playedReal
##                0.14047              195.70000              37.22700
## attr(,"scaled:scale")
## assists_per_minuteReal          heightInteger        time_playedReal
##             0.05258868               5.63816361              2.30749046
```

Ilustración 2. scale función de Z-SCORE

C. Normalización Min-Max

Este método realiza una transformación lineal de los datos originales V en el intervalo especificado [newmin,newmax].

V' = (V - min) * (newmax - newmin) / (max - min) + newmin

La ventaja de este método es que preserva exactamente todas las relaciones entre los datos. No introduce ningún potencial sesgo en los datos. La desventaja es que se encontrará un error “fuera del límite” ("out of bounds“) si un futuro ingreso de datos cae fuera del rango original.

```
basketball01

##    assists_per_minuteReal heightInteger time_playedReal
## 1                  0.0888           201           36.02
## 2                  0.1399           198           39.32
## 3                  0.0747           198           38.80
## 4                  0.0983           191           40.71
## 5                  0.1276           196           38.40
## 6                  0.1671           201           34.10
## 7                  0.1906           193           36.20
## 8                  0.1061           191           36.75
## 9                  0.2446           185           38.43
## 10                 0.1670           203           33.54
```

Ilustración 3. Preparación del conjunto de datos con el que se va a trabajar.

```
normalize <- function(x) {
  return ((x - min(x))*(1-0) / (max(x) - min(x)))
}
normalize(basketball01)
```

```
##    assists_per_minuteReal heightInteger time_playedReal
## 1              0.0000694837     0.9901442       0.1771356
## 2              0.0003213005     0.9753604       0.1933978
## 3              0.0000000000     0.9753604       0.1908352
## 4              0.0001162990     0.9408649       0.2002476
## 5              0.0002606871     0.9655045       0.1888641
## 6              0.0004553400     0.9901442       0.1676740
## 7              0.0005711461     0.9507208       0.1780227
## 8              0.0001547367     0.9408649       0.1807330
## 9              0.0008372539     0.9112974       0.1890119
## 10             0.0004548472     1.0000000       0.1649144
```

Ilustración 4. Normalización a traves de una función de Min-Max.

D. Normalización por escalamiento decimal

Este método realiza la normalización moviendo el punto decimal de los valores. El número de puntos "n" decimales movidos depende del máximo valor absoluto.

V' = V / 10j

donde j es el entero más pequeño tal que Max(|V'|) < 1. Sólo es útil cuando los valores de los atributos son mayores que 1 en valor absoluto.

```
ed=basketball[1:10,2]
ed
```

```
##  [1] 201 198 198 191 196 201 193 191 185 203
```

Ilustración 5. Conjunto de datos para el escalamiento decimal.

```
jmax=max(ed)
j=nchar(jmax)
V=ed/10^j
V
```

```
##  [1] 0.201 0.198 0.198 0.191 0.196 0.201 0.193 0.191 0.185 0.203
```

Ilustración 6. normalización por escalamiento decimal.

E. Normalización Sigmoidal

Este método realiza una transformación no lineal de los datos de entrada en el rango -1 a 1, usando una función sigmoidal.

V' = (1 - e^(-a)) / (1 + e^(-a)) donde a = (V - mean) / std

Los datos dentro de una desviación estándar de la media son mapeados a la región casi linear del sigmoide. Los puntos anómalos son comprimidos lo largo de las colas de la función sigmoidal.

La normalización sigmoidal es especialmente apropiada cuando se tienen datos anómalos que se desean incluir en el conjunto de datos.

```
a=scale(basketball01, center = TRUE, scale=TRUE)
a
```

```
##    assists_per_minuteReal heightInteger time_playedReal
## 1              -0.98253089    0.94002238      -0.5230791
## 2              -0.01083884    0.40793424       0.9070460
## 3              -1.25064944    0.40793424       0.6816930
## 4              -0.80188364   -0.83360476       1.5094320
## 5              -0.24472949    0.05320881       0.5083445
## 6               0.50638277    0.94002238      -1.3551519
## 7               0.95324702   -0.47887933      -0.4450723
## 8              -0.65356274   -0.83360476      -0.2067181
## 9               1.98008402   -1.89778104       0.5213456
## 10              0.50448122    1.29474781      -1.5978398
## attr(,"scaled:center")
## assists_per_minuteReal          heightInteger        time_playedReal
##                0.14047              195.70000              37.22700
## attr(,"scaled:scale")
## assists_per_minuteReal          heightInteger        time_playedReal
##             0.05258868               5.63816361               2.30749046
```

Ilustración 7. Variable "a", obtiene el Z-SCORE.

```
sig=(1-(exp(-a)))/(1+exp(-a))
sig
```

```
##    assists_per_minuteReal heightInteger time_playedReal
## 1              -0.455220222    0.43820836      -0.2557350
## 2              -0.005419364    0.20118489       0.4247906
## 3              -0.554824524    0.20118489       0.3282329
## 4              -0.380754530   -0.39423319       0.6379540
## 5              -0.121757651    0.02659813       0.2488366
## 6               0.247916257    0.43820836      -0.5899412
## 7               0.443535421   -0.23496642      -0.2189340
## 8              -0.315625735   -0.39423319      -0.1029925
## 9               0.757380237   -0.73928035       0.2549247
## 10              0.247023709    0.56989944      -0.6634325
## attr(,"scaled:center")
## assists_per_minuteReal          heightInteger        time_playedReal
##                0.14047              195.70000              37.22700
## attr(,"scaled:scale")
## assists_per_minuteReal          heightInteger        time_playedReal
##             0.05258868               5.63816361               2.30749046
```

Ilustración 8. Se aplica la fórmula de la normalización Sigmoidal.

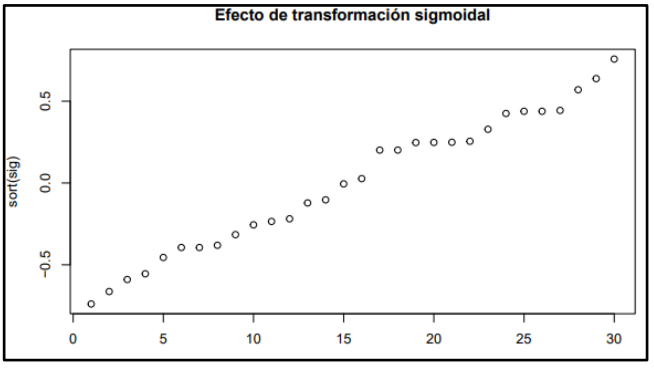


Ilustración 9. plot(sort(sig), main="Efecto de transformación sigmoidal")

F. Normalización SoftMax

Se llama así porque llega suavemente hacia su valor máximo y mínimo. La transformación es mas o menos lineal en el rango medio, y tiene una ligera no linealidad a ambos extremos. El rango total cubierto es 0 a 1 y la transformación asegura que no ocurran valores futuros que caigan fuera del rango

V' = 1 / (1 + e^(-a))

Donde a= (V - mean) / std

```
datitos=scale(basketball01, center = TRUE, scale=TRUE)
datitos

##    assists_per_minuteReal heightInteger time_playedReal
## 1              -0.98253089    0.94002238      -0.5230791
## 2              -0.01083884    0.40793424       0.9070460
## 3              -1.25064944    0.40793424       0.6816930
## 4              -0.80188364   -0.83360476       1.5094320
## 5              -0.24472949    0.05320881       0.5083445
## 6               0.50638277    0.94002238      -1.3551519
## 7               0.95324702   -0.47887933      -0.4450723
## 8              -0.65356274   -0.83360476      -0.2067181
## 9               1.98008402   -1.89778104       0.5213456
## 10              0.50448122    1.29474781      -1.5978398
## attr(,"scaled:center")
## assists_per_minuteReal          heightInteger          time_playedReal
##                0.14047              195.70000               37.22700
## attr(,"scaled:scale")
## assists_per_minuteReal          heightInteger          time_playedReal
##             0.05258868              5.63816361               2.30749046
```

Ilustración 10. La variable "datitos" obtiene el Z-Score

```
#aplicando el método SoftMax al conjunto de datos
sm1=1/(1+exp(-datitos))
sm1

##    assists_per_minuteReal heightInteger time_playedReal
## 1               0.2723899     0.7191042       0.3721325
## 2               0.4972903     0.6005924       0.7123953
## 3               0.2225877     0.6005924       0.6641164
## 4               0.3096227     0.3028834       0.8189770
## 5               0.4391212     0.5132991       0.6244183
## 6               0.6239581     0.7191042       0.2050294
## 7               0.7217677     0.3825168       0.3905330
## 8               0.3421871     0.3028834       0.4485037
## 9               0.8786901     0.1303598       0.6274624
## 10              0.6235119     0.7849497       0.1682838
## attr(,"scaled:center")
## assists_per_minuteReal          heightInteger          time_playedReal
##                0.14047              195.70000               37.22700
## attr(,"scaled:scale")
## assists_per_minuteReal          heightInteger          time_playedReal
##             0.05258868              5.63816361               2.30749046
```

Ilustración 11. Se aplica la fórmula del SoftMax.

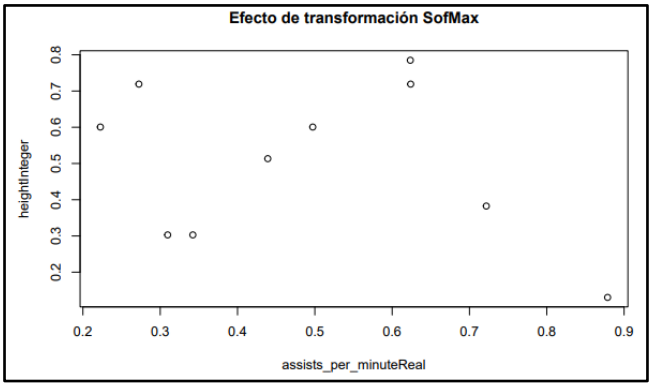


Ilustración 12. plot(sm1, main="Efecto de transformación SofMax")

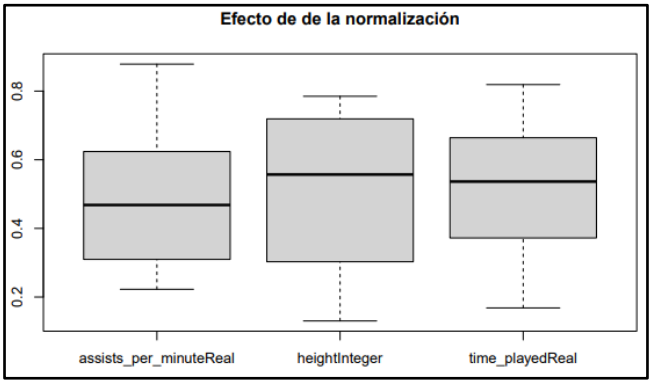


Ilustración 13. boxplot(sm1, main="Efecto de de la normalización")

G. Análisis de correspondencia para hombres.

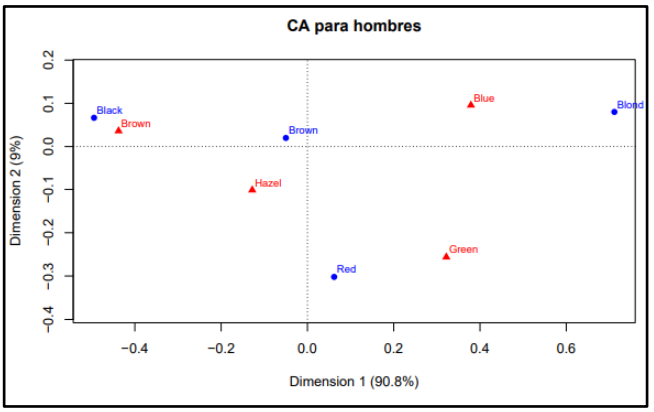


Ilustración 14. plot(ca(tabla.hombres), main="CA para hombres")

H. Análisis de correspondencia para mujeres.

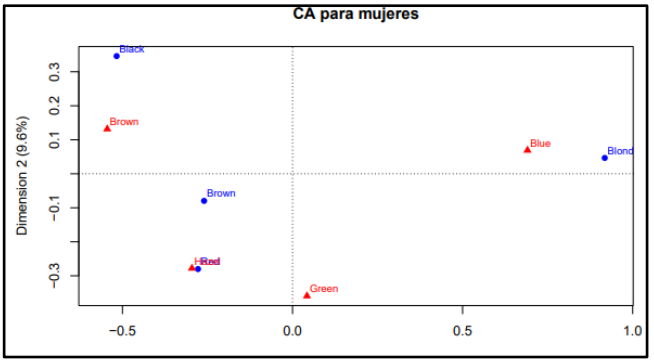


Ilustración 15. plot(ca(tabla.mujeres), main="CA para mujeres")

I. Análisis de correspondencia para datos conjuntos.

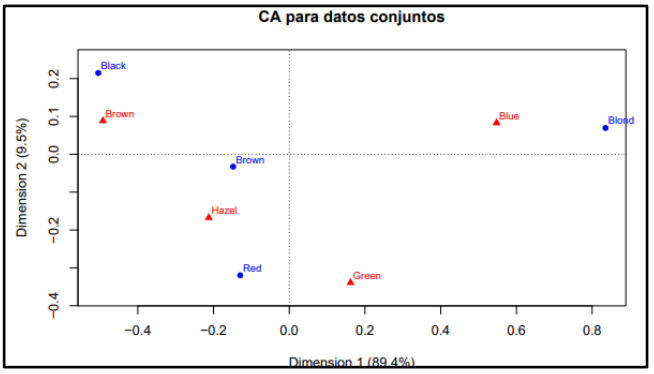


Ilustración 16. plot(ca(tabla.global), main="CA para datos conjuntos")

V. Conclusiones

Las técnicas de discretización, normalización y análisis de correspondencia permiten tener una visión más clara de los conjuntos de datos tratados, y en muchos casos mostrar de forma resumida y clara la diferencia entre los datos dentro del mismo conjunto, reduciendo la dificultad de procesamiento y análisis de estos datos.

Existen muchas herramientas que facilitan lo antesmencionado y que son usadas con frecuencia por analistas y expertos estadísticos, pero cada herramienta aplica técnicas y métodos distintos lo cual puede hacer una herramienta más rápida o de mejor aprovechamiento de memoria que otra, pero con resultadossemejantes.

Para cada una de estas técnicas cabe señalar la importancia de los conceptos matemáticos y estadísticos que estas presentan, ya quesin estos no podrían llevarse a cabo algunos procesos por no decirla

mayoría, dentro del análisis de datos, tal es el ejemplo de la normalización que usa en conceptos estadísticos para re-escalar valores dentro de un conjunto de datos y así que entre estos no exista mucha diferencia entre sus valores, facilitando la comprensión de estos al estar de manera más resumida.

Referencias

[1] A. Fernández *et al.*, "Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks," *WIREs Data Min. Knowl Discov*, vol. 4, pp. 380–409, 2014, doi: 10.1002/widm.1134.

[2] J. M.Criado, "Minería de Datos: Análisis de componentesprincipales | AnálisisDeDatos.net." http://analisisdedatos.net/mineria/tecnicas/PCA.php#dos (accessed Jul. 11, 2020).

[3] J. Han, M. Kamber, and J. Pei, "Data Mining. Conceptsand Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)," 2011.

[4] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no.1, pp. 97–107, Jan. 2014, doi: 10.1109/TKDE.2013.109.

[5] J. H. O. M. J. R. Q. C. F. Ramírez, *Introduccion a la mineria de datos*, Libro impr. Madrid: Pearson;PrenticeHall, 2004.

[6] S. García, J. Luengo, F. Herrera, S. García, J. Luengo, and F. Herrera, "Data sets and proper statistical analysis of data mining techniques," *Intell. Syst. Ref. Libr.*, vol. 72,pp. 19–38, 2015, doi: 10.1007/978-3-319-10247-4_2.

[7] E. Acuña, "Preprocesamiento: Reducción deDatos-Discretización Minería de Datos."

[8] S. García, S. Ra-Mírez-Gallego, J. Luengo, and F. Herrera,"Big Data Preprocesamiento y calidad de datos." Accessed: Jul. 12, 2020. [Online]. Available: www.highlycited.com.

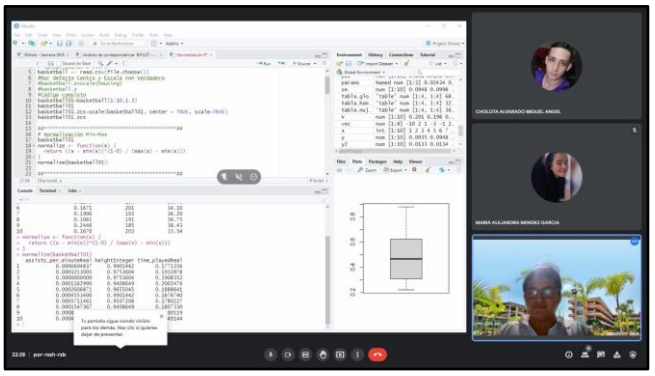
[9] C. L. Hernández and J. E. Rodríguez, "Preprocesamientode datos estructurados Structured Data Preprocessing."

Anexos

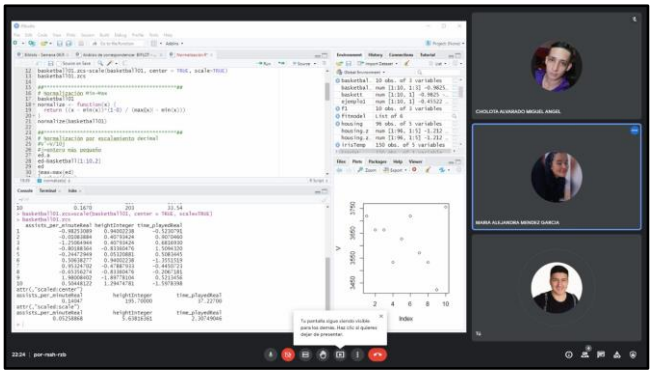
Anexo 1. Tabla de Actividades designadas

Tarea asignada	Delegado/s
Revisión bibliográfica	Carlos Abad
Redacción de la Introducción	Carlos Abad
Redacción de la Revisión Literaria	Carlos Abad
Cuadro de diferencias y similitudes de los métodos los programas Weka, R, Knime	Alejandra Méndez
Elaboración del resumen y conclusión	Todos
Edición del formato y corrección	Carlos Abad
Diferentes métodos de Normalización	Carlos Abad Miguel Cholota

Anexo 2. Diálogo y resultado de Normalización Z-Score y Min-Max.



Anexo 3. Diálogo y resultado de Normalización de escalamiento decimal.



Anexo 4. Diálogo y resultado de Normalización de Sigmoidal (2 versiones).