UNIVERSIDAD
NACIONAL
DE COLOMBIA

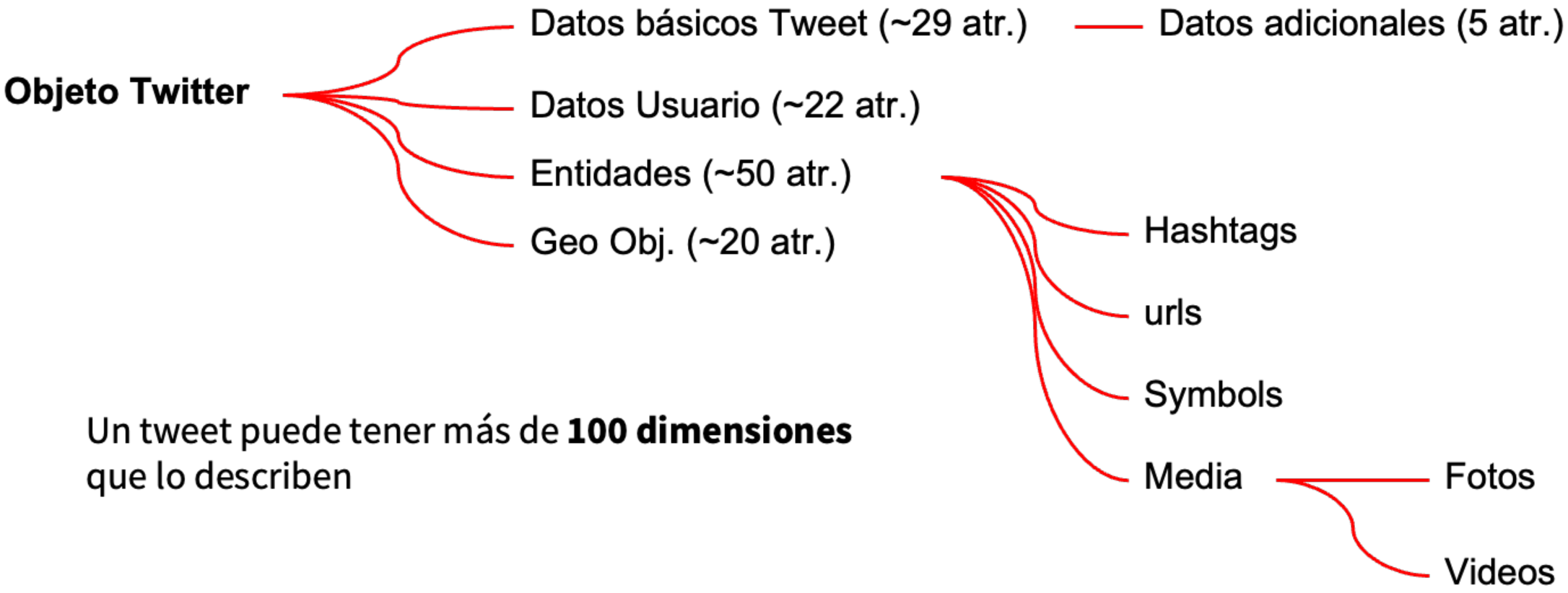
Introducción a la Minería de Datos

Verónica Guarín Escudero
Escuela de Estadística

Correo: jvguarine@unal.edu.co

Reducción de Dimensionalidad

Reducción de dimensionalidad



Un tweet puede tener más de **100 dimensiones** que lo describen

Tweet object - developer.twitter.com

Reducción de dimensionalidad

Estrategias de reducción de datos:

- **Reducción de dimensionalidad**: Remover atributos que no son importantes.
  - Componentes Principales (PCA), Pares correlacionados, VarImp, Escalado multidimensional, etc.
- **Reducción de datos**
  - Representar los datos a partir de modelos.
  - Histograms, clustering, sampling, Data cube aggregation, discretization, etc

Reducción de dimensionalidad

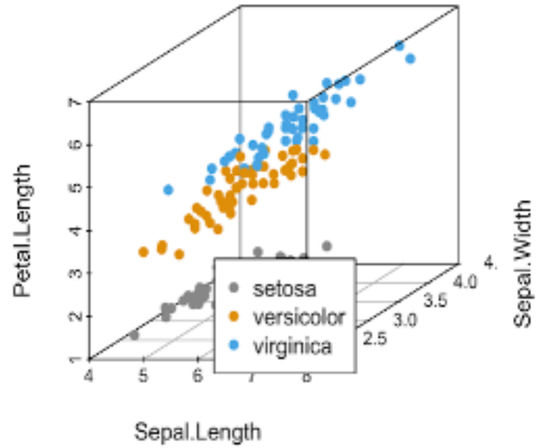
El problema de la dimensionalidad, también conocido como "maldición de la dimensionalidad", es un fenómeno que ocurre cuando trabajamos con conjuntos de datos de alta dimensionalidad. Se refiere a una serie de desafíos y complicaciones que surgen cuando el número de dimensiones (o características) en un conjunto de datos es muy grande en comparación con el número de muestras.

- **Espacio de características disperso:** A medida que aumenta el número de dimensiones, el espacio de características se vuelve cada vez más disperso. Esto significa que las muestras individuales pueden estar muy separadas entre sí en el espacio de características, lo que dificulta la identificación de patrones significativos.
- **Dificultad en la visualización:** A medida que aumenta la dimensionalidad, se vuelve más difícil visualizar los datos. Nuestro mundo es tridimensional, y la mayoría de las técnicas de visualización están limitadas a dos o tres dimensiones.
- **Sobreajuste y generalización pobre:** A medida que aumenta el número de dimensiones, aumenta la complejidad del modelo necesario para describir los datos. Esto puede conducir al sobreajuste, donde un modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos.

Reducción de dimensionalidad

- **Costo computacional:** El procesamiento y almacenamiento de datos de alta dimensionalidad puede ser computacionalmente costoso. Muchos algoritmos de aprendizaje automático experimentan un aumento significativo en el tiempo de ejecución a medida que aumenta el número de dimensiones.
- **Reducción de la eficiencia de los algoritmos:** Muchos algoritmos de aprendizaje automático sufren de una reducción en su eficiencia y rendimiento a medida que aumenta la dimensionalidad. Algunos algoritmos pueden volverse computacionalmente inviables o producir resultados subóptimos en conjuntos de datos de alta dimensionalidad.

Reducción de dimensionalidad



- **Reducción de dimensionalidad**
  - Evita la maldición de la dimensionalidad
  - Ayuda a eliminar características irrelevantes y reduce el ruido
  - Reduce el tiempo y el espacio necesarios en la extracción de datos
  - Facilita la interpretación visual. Permitir una visualización más simple de los datos.

Reducción de dimensionalidad

Eliminar columnas con datos faltantes:

- Si bien podemos trabajar en imputación de datos faltantes, a veces no es posible rellenar
- **Criterio de eliminación**: Predominio de datos faltantes.
  - Por ejemplo, Atributos con menos del 5% o 10% de valores.
- Este método aplica tanto a variables numéricas como categóricas.

Reducción de dimensionalidad

Low Variance Filter

- Una forma de medir cuánta información tiene una columna de datos es medir su varianza.
- En el caso límite donde las celdas de la columna asumen un valor constante, la **varianza es 0** y la columna no sería de ayuda en la discriminación de diferentes grupos de datos.
- Con Low Variance Filter calcula la varianza para cada uno de los atributos y remueve aquellos que están por debajo de un umbral.

- Consideraciones del método:
  - Los rangos de columna de datos deben **normalizarse** para que los valores de varianza sean independientes del rango del dominio de la columna.
  - Para variables booleanas podemos usar Bernoulli. **Var[x] = p(1 - p)**

Reducción de dimensionalidad

Reducción utilizando chi-cuadrado χ^2

Podemos seleccionar *variables* con los valores más altos para el estadístico de la prueba χ^2 entre la clase y cada *variable.*
-Aplica a variables categóricas y no-negativas como booleanos o frecuencias (conteos de términos en la clasificación de documentos).

La prueba χ^2 mide la dependencia, en este sentido, el propósito es eliminar aquellas variables que son independientes de la variable respuesta en la clasificación y en consecuencia son irrelevantes para el análisis.

Reducción de dimensionalidad

Reducción utilizando chi-cuadrado χ^2

```
library(mlbench)
library(FSelector)
data(HouseVotes84)

summary(HouseVotes84)
str(HouseVotes84)

# Calculamos los valores del estadístico de Chi2
chi2_scores = chi.squared(Class~., HouseVotes84)

print(chi2_scores)

# Seleccionamos los Top-K
subset = cutoff.k(chi2_scores, 5)

# Las features que aportan mas varianza son:
formula = as.simple.formula(subset, "Class")
print(formula)
```

chi.squared: utiliza Cramer's *V* coefficient

$$v = \sqrt{\frac{\chi^2}{n * m}}$$

n: cantidad de instancias

m: *mínimo*(filas - 1, columnas -1)

v es un valor entre 0 y 1

donde 1 indica una relación más fuerte entre X e Y

Reducción de dimensionalidad

Reducción utilizando chi-cuadrado χ^2

```
library(mlbench)
library(FSelector)
data(HouseVotes84)

summary(HouseVotes84)
str(HouseVotes84)

# Calculamos los valores del estadístico de Chi2
chi2_scores = chi.squared(Class~., HouseVotes84)

print(chi2_scores)

# Seleccionamos los Top-K
subset = cutoff.k(chi2_scores, 5)

# Las features que aportan mas varianza son:
formula = as.simple.formula(subset, "Class")
print(formula)
```

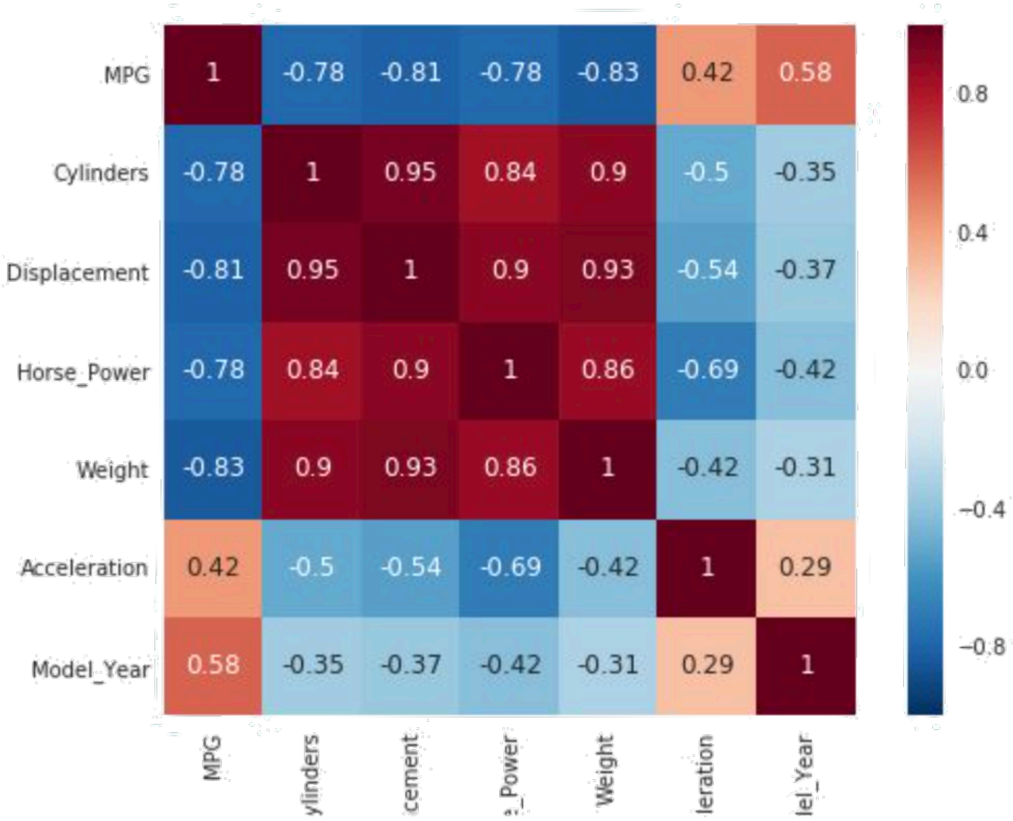
```
print(chi2_scores)
    attr_importance
V1        0.409330348
V2        0.004534049
V3        0.748864321
V4        0.923255954
V5        0.718768923
V6        0.428332508
V7        0.521967369
V8        0.661876085
V9        0.629797943
V10       0.083809300
V11       0.378240781
V12       0.714922593
V13       0.555971176
V14       0.625283342
V15       0.538263037
V16       0.353273580
```

```
print(formula)
Class ~ V4 + V3 + V5 + V12 + V8
```

Reducción de dimensionalidad

Columnas altamente correlacionadas:

- Los atributos correlacionados introducen **redundancia** al dataset.
- Estos atributos redundantes no agregan información y tornan complejo el modelado.
- Se puede eliminar una de las dos columnas sin disminuir drásticamente la cantidad de información disponible.
- El procedimiento consiste en la eliminación de pares correlacionados a partir de la matriz de correlaciones.
- El método puede ser utilizado con variables continuas o discretas con **Coeficiente de correlación de Pearson** y **Prueba x^2 de Pearson**.



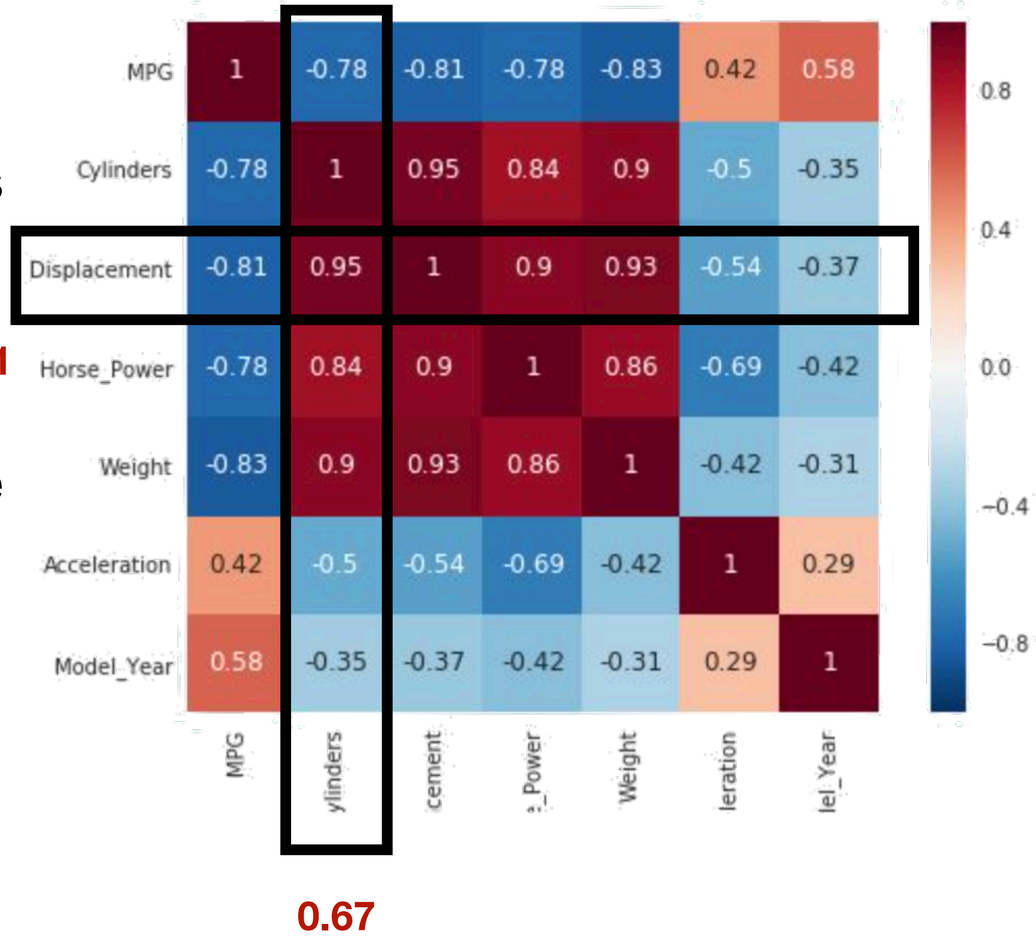
Reducción de dimensionalidad

Columnas altamente correlacionadas:

El procedimiento consiste en la eliminación de pares correlacionados a partir de la matriz de correlaciones.

1. Se define un umbral de correlación.
2. Seleccionamos pares de valores mayores al umbral.
3. El criterio es conservar la variable que en el resto de las correlaciones sea en promedio menor.

Ejemplo:
Umbral=0.7
Displacement vs Cylinders



0.71

0.67

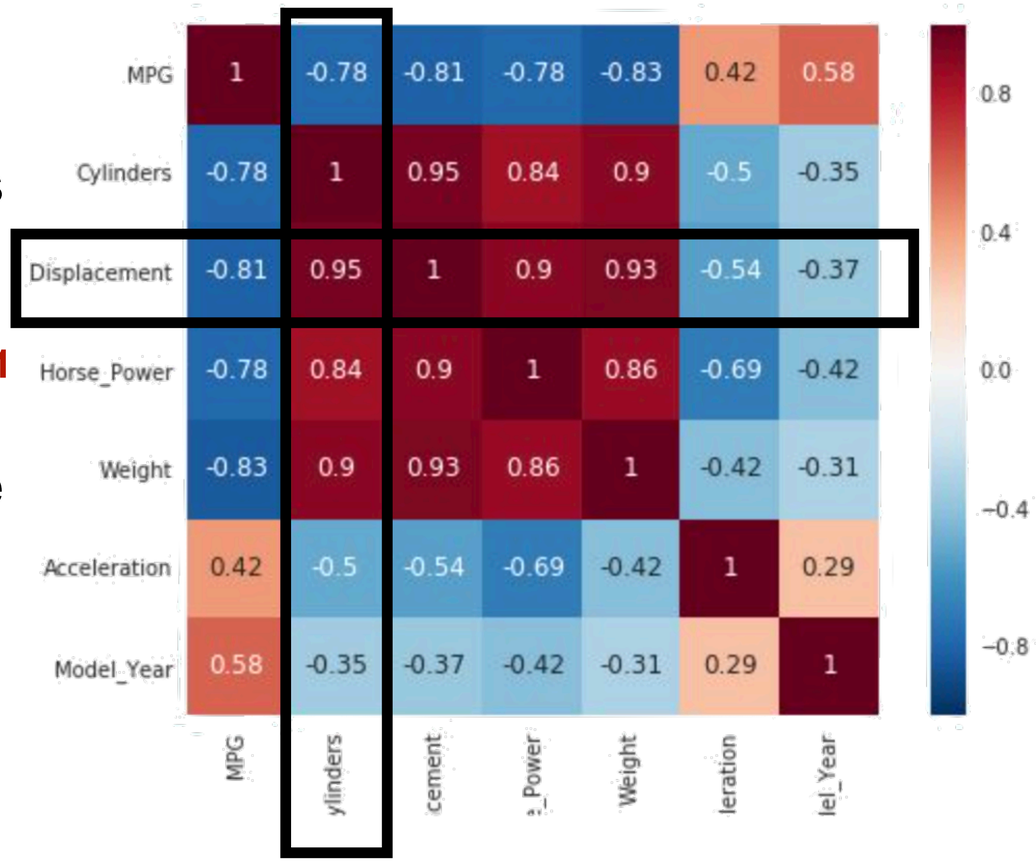
Reducción de dimensionalidad

Columnas altamente correlacionadas:

El procedimiento consiste en la eliminación de pares correlacionados a partir de la matriz de correlaciones.

1. Se define un umbral de correlación.
2. Seleccionamos pares de valores mayores al umbral.
3. El criterio es conservar la variable que en el resto de las correlaciones sea en promedio menor.

Ejemplo:
Umbral=0.7
Displacement vs Cylinders



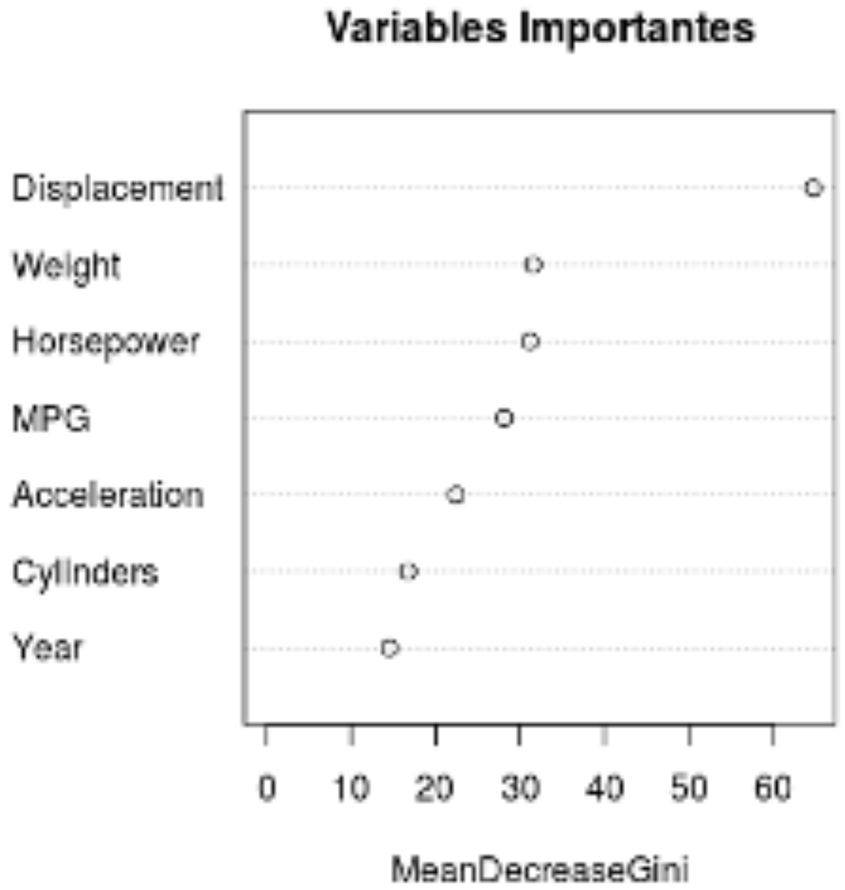
0.71

0.67

Reducción de dimensionalidad

Variables Importantes (RF)

- Son productos derivados de la salida de un modelo de ensamble Random Forest (RF).
- La inducción de árboles de decisión involucra la utilización de **medidas internas de importancia**.
- RF realiza un muestreo de variables para cada árbol y mide la importancia de cada variable para esa muestras. Al finalizar calcula la importancia promedio de cada variable a partir de todas las muestras en las que salió seleccionada.



Reducción de dimensionalidad

Variables Importantes (RF)

Un **Random Forest** (bosque aleatorio) es un algoritmo de aprendizaje automático utilizado tanto para tareas de clasificación como de regresión. Se basa en la idea de construir múltiples árboles de decisión durante el entrenamiento y combinar sus resultados para obtener una predicción más precisa y robusta.

Creación de Árboles de Decisión: Un Random Forest construye muchos árboles de decisión. Cada árbol se construye utilizando una muestra aleatoria de los datos de entrenamiento y una selección aleatoria de características en cada nodo del árbol. Esta aleatorización ayuda a que los árboles sean menos correlacionados entre sí.

Muestreo Bootstrap: Para cada árbol en el bosque, se usa un conjunto de datos de entrenamiento diferente, obtenido mediante el método de muestreo bootstrap. Esto significa que para cada árbol se selecciona aleatoriamente una muestra con reemplazo del conjunto de datos original.

Selección Aleatoria de Características: En cada nodo de un árbol, solo un subconjunto aleatorio de las características se considera para la división. Esto introduce más diversidad entre los árboles y ayuda a evitar el sobreajuste.

Predicción:

- **Para Clasificación**: Cada árbol emite una "votación" sobre la clase a la que debería pertenecer una muestra. La clase final se determina por mayoría de votos: la clase que recibe más votos de los árboles.
- **Para Regresión**: La predicción final se calcula promediando las predicciones de todos los árboles.

Reducción de dimensionalidad

Variables Importantes (RF)

Un **árbol de decisión** es una herramienta utilizada en el aprendizaje automático y la estadística para tomar decisiones basadas en una serie de preguntas. Un árbol de decisión una serie de decisiones que se ramifican a medida que se responden preguntas sobre los datos.

Estructura de un Árbol de Decisión

1. **Raíz**: La raíz del árbol es el nodo inicial, que contiene toda la información del conjunto de datos. En este nodo se realiza la primera pregunta (o división) sobre una de las características de los datos.
2. **Nodos Internos**: Estos nodos representan decisiones adicionales. Cada nodo interno corresponde a una pregunta sobre una característica específica del conjunto de datos. Basado en la respuesta (por ejemplo, si una característica es mayor o menor que un valor umbral), los datos se dividen en ramas.
3. **Ramas**: Las ramas conectan los nodos entre sí. Cada rama representa el resultado de una pregunta (por ejemplo, "Sí" o "No") y lleva a otro nodo o a una hoja.
4. **Hojas**: Las hojas son los nodos terminales del árbol, donde se hace la clasificación final o predicción. En un problema de clasificación, una hoja puede representar una clase específica. En un problema de regresión, una hoja puede representar un valor numérico.

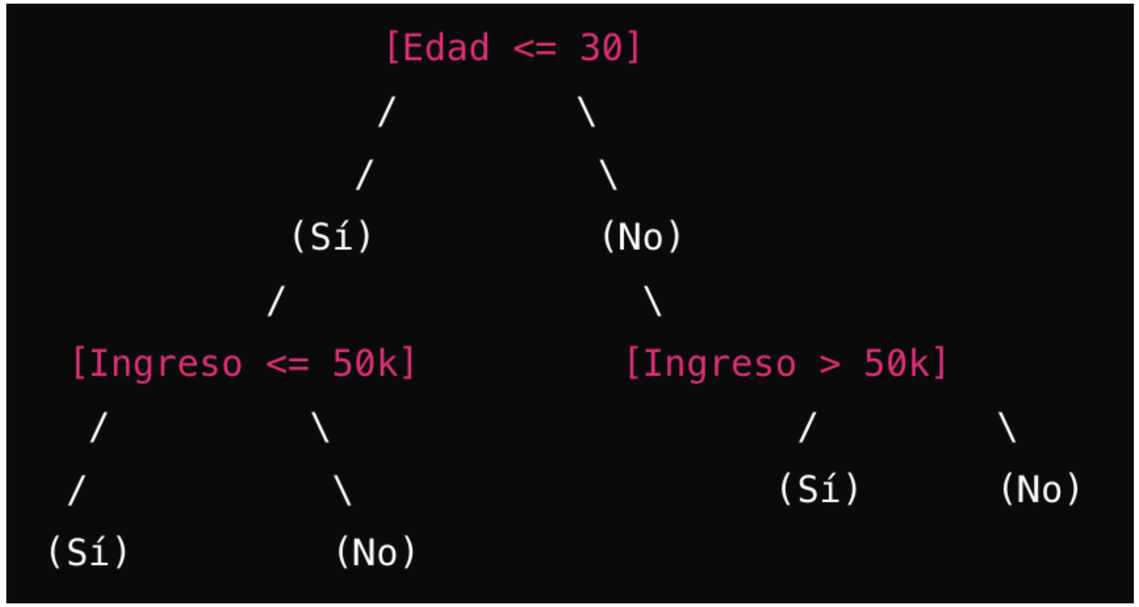
Reducción de dimensionalidad

Variables Importantes (RF)

Ejemplo de Árbol de Decisión para Clasificación

Problema: Clasificar si un cliente comprará un producto basado en su edad y el ingreso.

1. **Características**: Edad, Ingreso
2. **Objetivo**: Decidir si el cliente comprará (Sí) o no (No) el producto.
3.



Interpretación:

- **Raíz**: Si la edad del cliente es 30 años o menos:
  - **Sí**: Pasar al siguiente nodo.
  - **No**: El cliente no comprará (ya que se encuentra en la rama de 'No').
- **Nodo Interno**: Si el ingreso es 50k o menos (para clientes de 30 años o menos):
  - **Sí**: El cliente comprará el producto.
  - **No**: El cliente no comprará el producto.
- **Nodo Terminal**:
  - **Sí**: El cliente comprará el producto.
  - **No**: El cliente no comprará el producto.

Reducción de dimensionalidad

Variables Importantes (RF)

La **importancia de las características** en un modelo de Random Forest se refiere a qué tan valiosa es cada característica para hacer predicciones. Un Random Forest calcula la importancia de las características como parte del proceso de entrenamiento, y esto se hace utilizando métodos basados en cómo las características afectan la calidad de las divisiones (o nodos) en los árboles del bosque.

Importancia Basada en la Reducción de Impureza (Mean Decrease in Impurity)

Este método es el más común y se basa en la cantidad de reducción en la impureza (o variabilidad) que una característica proporciona en los árboles de decisión del Random Forest. La impureza puede ser medida por índices como la **entropía** (en clasificación) o la **varianza** (en regresión) o el índice **gini** (también usada en clasificación y es una medida de impureza que varía de 0 (perfectamente puro) a 1 (totalmente impuro)).

- **Reducción de Impureza en un Nodo**: En cada división de un árbol de decisión, el algoritmo evalúa qué tan buena es la división en función de la reducción de impureza que produce. La importancia de una característica se calcula como la suma de las reducciones de impureza ponderadas que esa característica contribuye a través de todas las divisiones en todos los árboles del bosque.
- **Cálculo**: Para cada característica, se suman las reducciones en la impureza que la característica contribuye a lo largo de todos los nodos y árboles en el bosque. Luego, esta suma se promedia y se normaliza para obtener la importancia relativa.

Reducción de dimensionalidad

Variables Importantes (RF)

- from sklearn.ensemble import RandomForestClassifier
- from sklearn.datasets import load_iris
- import pandas as pd

- # Cargar datos
- data = load_iris()
- X = data.data
- y = data.target
- feature_names = data.feature_names

- # Crear y entrenar el modelo
- model = RandomForestClassifier(n_estimators=100)
- model.fit(X, y)

- # Obtener importancias de características
- importances = model.feature_importances_

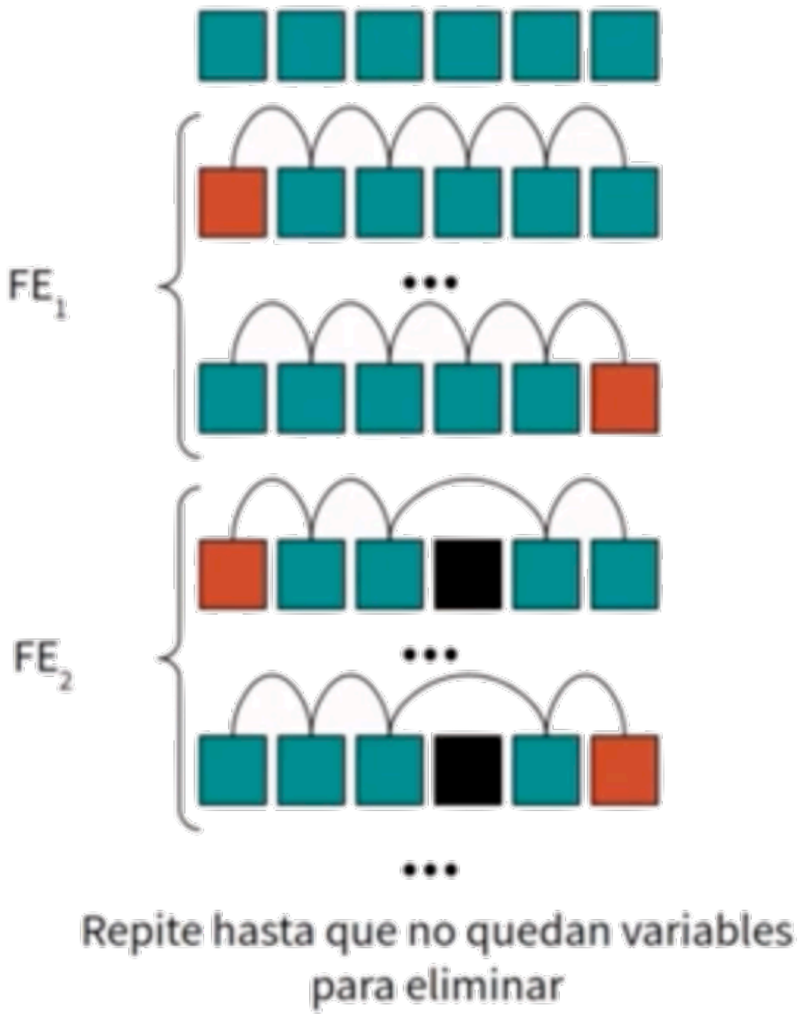
- # Crear un DataFrame para visualizar las importancias
- importance_df = pd.DataFrame({
- 'Feature': feature_names,
- 'Importance': importances
- }).sort_values(by='Importance', ascending=False)

- print(importance_df)

Reducción de dimensionalidad

Eliminación Backward

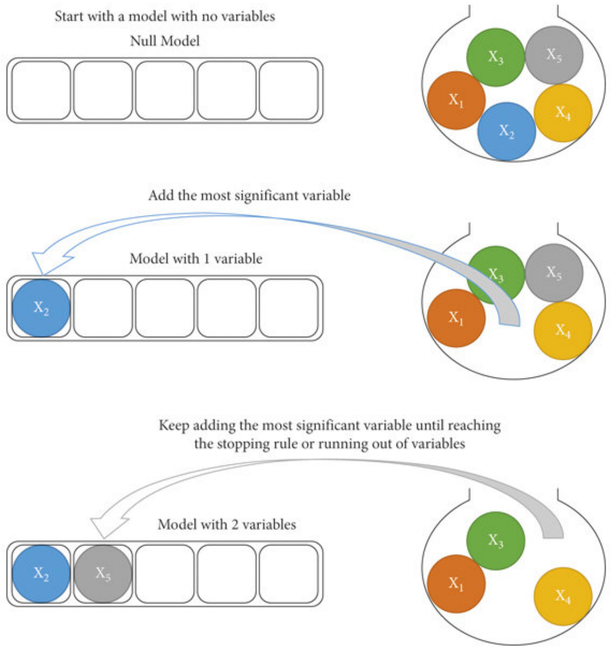
- Eliminación Backward realiza un *loop* y utiliza un algoritmo de aprendizaje automático para medir cómo disminuye el error al quitar algún atributo.
- El procedimiento comienza con el conjunto completo de atributos.
- En cada paso, elimina el peor atributo que queda en el conjunto
- La principal desventaja de esta técnica es el **alto número de iteraciones** para datasets con gran dimensionalidad, generalmente esto conduce a tiempos de cómputo muy elevados.



Reducción de dimensionalidad

Selección Forward

- El procedimiento comienza con un conjunto vacío de atributos como **conjunto de reducción**.
- El mejor de los atributos originales se determina y agrega al conjunto de reducción.
- En cada iteración o paso posterior, el mejor de los atributos originales restantes se agrega al conjunto.

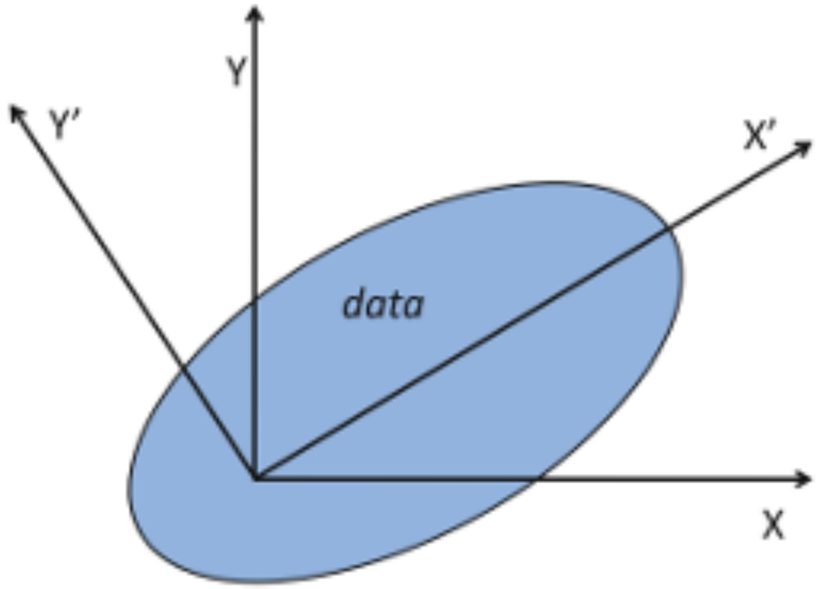


Reducción de dimensionalidad

Análisis de Componentes Principales- PCA

- Encuentra una proyección que captura la mayor cantidad de variación en los datos.

- Los **datos originales se proyectan en un espacio mucho más pequeño**, lo que resulta en la reducción de dimensionalidad.
- Buscamos los autovectores de la matriz de covarianza, y estos autovectores definen el nuevo espacio



Reducción de dimensionalidad

Análisis de Componentes Principales- PCA

Es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos, manteniendo la mayor cantidad posible de variabilidad o información. PCA transforma un conjunto de variables originales en un nuevo conjunto de variables llamadas componentes principales, que son combinaciones lineales de las variables originales.
Pasos en la ejecución de un PCA:

1. **Estandarización de los datos**. Se estandarizan los datos para que cada variable tenga media cero y desviación estándar uno, especialmente si las variables originales tienen diferentes escalas. Esto es importante porque PCA es sensible a la escala de las variables.
2. **Se calcula la matriz de covarianza de los datos estandarizados**. La matriz de covarianza mide cómo varían juntas las diferentes variables. Una covarianza alta entre dos variables indica que tienden a aumentar o disminuir juntas.
3. **Cálculo de los Valores Propios y Vectores Propios:** Se calculan los valores propios (eigenvalues) y vectores propios (eigenvectors) de la matriz de covarianza. Los vectores propios determinan las direcciones principales (componentes principales) de la variabilidad en los datos, mientras que los valores propios indican la cantidad de varianza en cada dirección.
4. **Ordenación y Selección de Componentes Principales:** Los vectores propios se ordenan en función de sus valores propios, de mayor a menor. Los primeros vectores propios (componentes principales) corresponden a las direcciones con la mayor varianza en los datos.Se seleccionan los primeros componentes principales (generalmente menos que el número original de variables) que capturan la mayor parte de la variabilidad en los datos.
5. **Transformación de los Datos:** Finalmente, los datos originales se proyectan sobre los componentes principales seleccionados para obtener un conjunto de datos de menor dimensión.

Reducción de dimensionalidad

Análisis de Componentes Principales- PCA

Ejemplo: Imaginemos que tenemos un conjunto de datos con dos variables, X_1 y X_2. Supongamos que tenemos los siguientes datos para dos variables.

1

Muestra	X1	X2
1	2	3
2	3	4
3	4	5
4	5	6

2

Primero, estandarizamos los datos para que cada variable tenga una media de 0 y una desviación estándar de 1. Esto se hace restando la media y dividiendo por la desviación estándar de cada variable. Supongamos que da lo siguiente:

Muestra	X1	X2
1	1.34	-1.34
2	-0.67	-0.67
3	0	0
4	0.67	0.67

Reducción de dimensionalidad

Análisis de Componentes Principales- PCA

Se calcula la matriz de variantes y covarianzas:

3

$$\text{Matriz de Covarianza} = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) \\ \text{Cov}(X_1, X_2) & \text{Var}(X_2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

4

Cálculo de Valores Propios y Vectores Propios

Para obtener los componentes principales, calculamos los valores propios (eigenvalues) y los vectores propios (eigenvectors) de la matriz de covarianza. Para simplificar, en este ejemplo, los valores propios y vectores propios resultan ser:

Valores propios: $\lambda_1 = 2,\ \lambda_2 = 0$

Vectores propios:

$$\text{Vector propio para } \lambda_1 = \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$$

$$\text{Vector propio para } \lambda_2 = \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}$$

Reducción de dimensionalidad

Análisis de Componentes Principales- PCA

Transformación de datos:
Para transformar los datos en el nuevo espacio definido por los componentes principales, proyectamos los datos originales sobre los vectores propios. Usamos el vector propio asociado con el mayor valor propio (componente principal principal) para la proyección.

Entonces, para cada muestra, calculamos:

CP1=$X_1 \cdot 0.707 + X_2 \cdot 0.707$

CP2=$X_1 \cdot (-0.707) + X_2 \cdot 0.707$

https://www.kdnuggets.com/2015/05/7-methods-data-dimensionality-reduction.html

Reducción de dimensionalidad

Dimensionality Reduction	Reduction Rate	Accuracy on validation set	Best Threshold	AuC	Notes
Baseline	0%	73%	–	81%	Baseline models are using all input features
Missing Values Ratio	71%	76%	0.4	82%	–
Low Variance Filter	73%	82%	0.03	82%	Only for numerical columns
High Correlation Filter	74%	79%	0.2	82%	No correlation available between numerical and nominal columns
PCA	62%	74%	–	72%	Only for numerical columns
Random Forrest / Ensemble Trees	**86%**	**76%**	–	**82%**	–
Backward Feature Elimination + missing values ratio	99%	94%	–	78%	Backward Feature Elimination and Forward Feature Construction are **prohibitively slow** on high dimensional data sets. It becomes practical to use them, only if following other dimensionality reduction techniques, like here the one based on the number of missing values.
Forward Feature Construction + missing values ratio	91%	83%	–	63%	

Tomado de: https://www.kdnuggets.com/2015/05/7-methods-data-dimensionality-reduction.html

Ingeniería de Características

Ingeniería de Características

- Concepto de Feature Engineering.
- Métodos de construcción de variables por:
  - Discretización, normalización y binning.
- Evaluación de las transformaciones.

Ingeniería de Características

- Dentro del Proceso de Descubrimiento de Conocimiento se corresponde con la etapa de **Transformación de datos**.
  - La transformación de datos engloba, en realidad, cualquier proceso que modifique la forma de los datos.
  - Prácticamente todos los procesos de preparación de datos entrañan algún tipo de transformación.

- ***Feature Engineering*** es la tarea de **mejorar el rendimiento** del modelado en un conjunto de datos mediante la **transformación** de su ***feature space***.

Ingeniería de Características

Normalización

La normalización consiste en **escalar los features** (numéricos) de manera que puedan ser mapeados a un rango más pequeño.

Por ejemplo: 0 a 1 ó -1 a 1.

La normalización es particularmente utilizada en:

- Tareas de mining donde las unidades de medidas dificultan la comparación de features.
- Medidas de Distancias. Vecinos más cercanos, Clustering, etc.

Ayuda a evitar que atributos con mayores magnitudes tengan mayor peso que los rangos pequeños.

Los métodos más utilizados para normalizar son:

- Min-Max
- Z-Score
- Decimal Scaling

Ingeniería de Características

Normalización Min-Max

La ***Normalización Min-Max*** funciona al ver cuánto más grande es el valor actual del valor mínimo ***min(X)*** y escala esta diferencia por el rango.

$$X^*_{\text{mm}} = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Los valores de normalización min-max van de 0 a 1.

Ejemplo: Dataset Iris y variable Sepal.Length

```
Sepal.Length

Min.    4.300
1st Qu. 5.100
Median  5.800
Mean    5.843
3rd Qu. 6.400
Max.    7.900
```

$$X_{mm} = \frac{X - 4.3}{7.9 - 4.3}$$

Para los valores extremos es 0 y 1

Ingeniería de Características

Normalización Z-Score

Los valores para un **atributo X**, se normalizan en base a la **media** y **desviación estándar** de **X.**

$$Z-score = \frac{X-mean(X)}{sd(X)}$$

Sepal.Length	
Min.	4.300
Median	5.800
Mean	5.843
Max.	7.900
SD	0.828

Para una Iris con el largo del sépalo más corto:

$$Z-score = \frac{4.3-5.843}{0.828} = -1,863$$

Para una Iris con el largo del sépalo más largo:

$$Z-score = \frac{7.9-5.843}{0.828} = 2,484$$

Ingeniería de Características

Normalización Decimal Scaling

Decimal Scaling asegura que cada valor normalizado se encuentra entre - 1 y 1.

$$X_{decimal} = \frac{X}{10^d}$$

donde **d** representa el número de dígitos en los valores de la variable con el valor absoluto más grande.

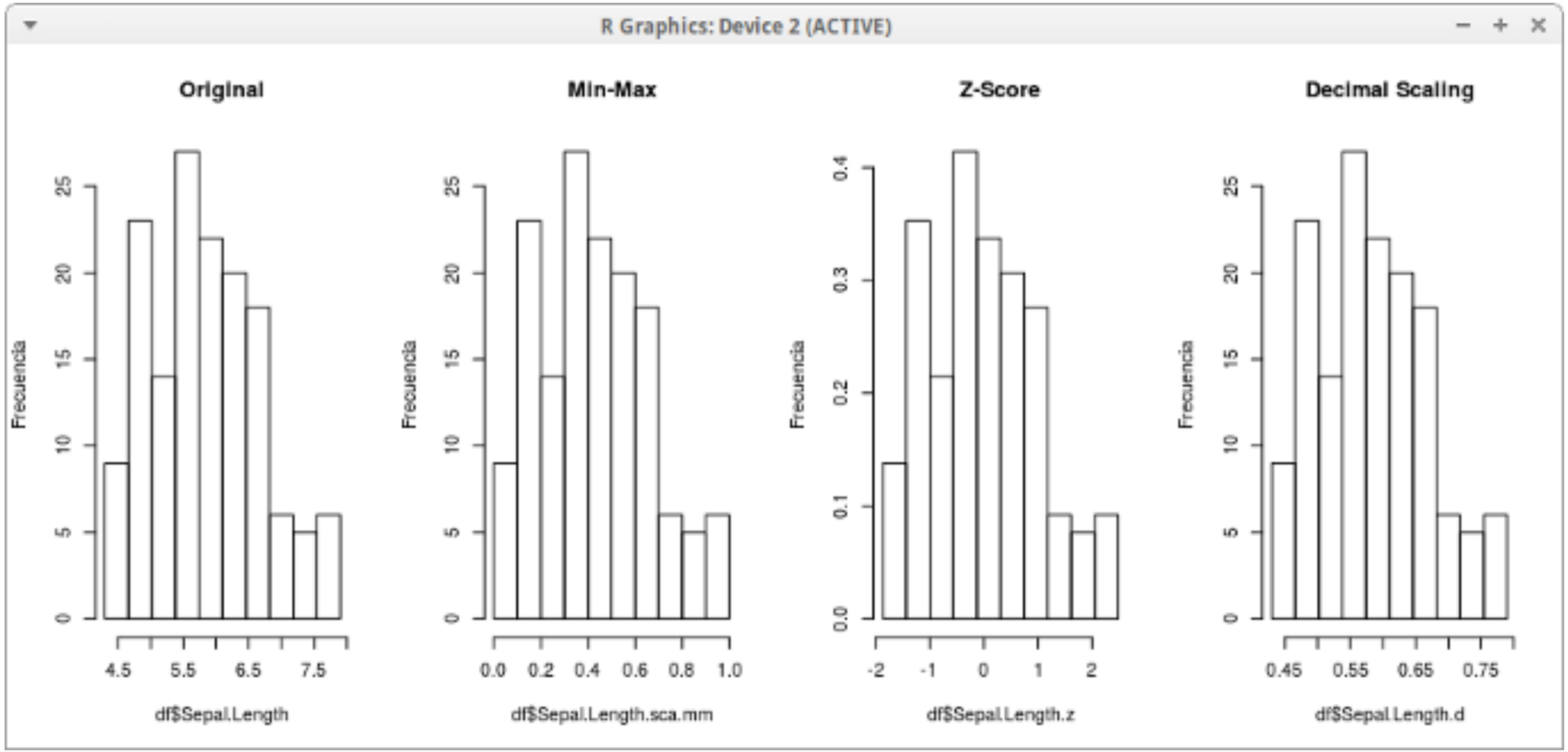
```
Sepal.Length

Min.    4.300
Median  5.800
Mean    5.843
Max.    7.900
SD      0.828
```

$$X_{decimal} = \frac{4.3}{10^d}$$

$$X_{decimal} = \frac{7.9}{10^d}$$

Ingeniería de Características

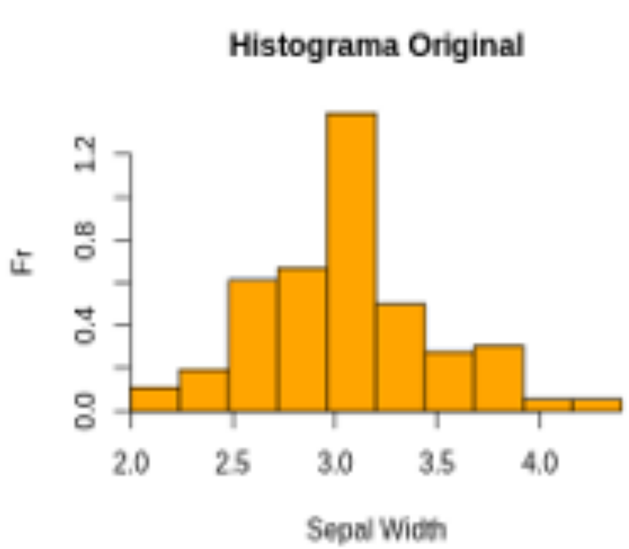
Comparación

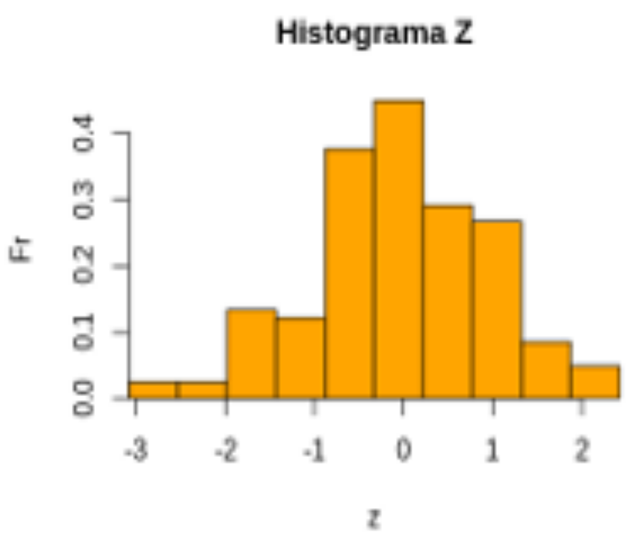


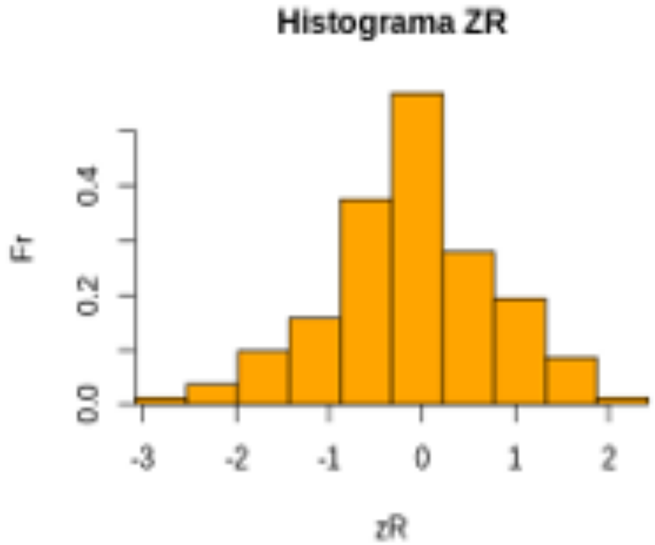
Ingeniería de Características

Escalados Robustos

- Si nuestro dataset contienen muchos valores atípicos, es probable que un escalado utilizando la media y la varianza de los datos no funcione muy bien.
- En estos casos, puede usar un método *robusto* como reemplazo.
- Usan estimaciones más sólidas para el **centro** y el **rango** de sus datos.
- Por ejemplo: Mediana (o algún percentil) e IQR







Ingeniería de Características

Transformación para lograr Normalidad

$$sesgo = \frac{3 * (media - mediana)}{desviacion}$$

- Si la media es mayor que la mediana entonces hay sesgo a derecha (Sesgo+)
- Si la media es menor que la mediana entonces hay sesgo a izquierda (Sesgo-)

Podemos reducir este sesgo a partir de transformaciones:

1. Raíz cuadrada
2. Logaritmos
3. Inversa de la Raíz Cuadrada

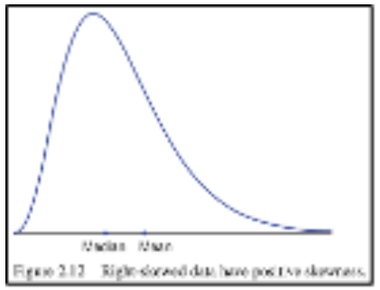


Figure 2.12 Right-skewed data have positive skewness.

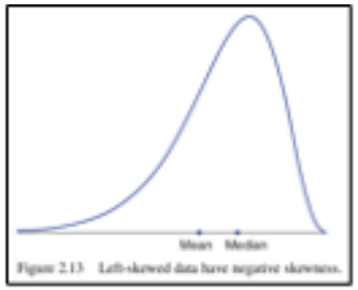
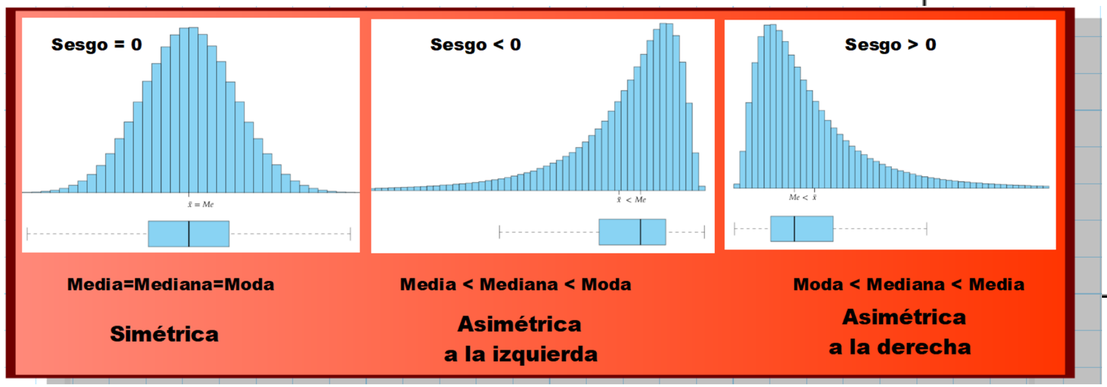


Figure 2.13 Left-skewed data have negative skewness.

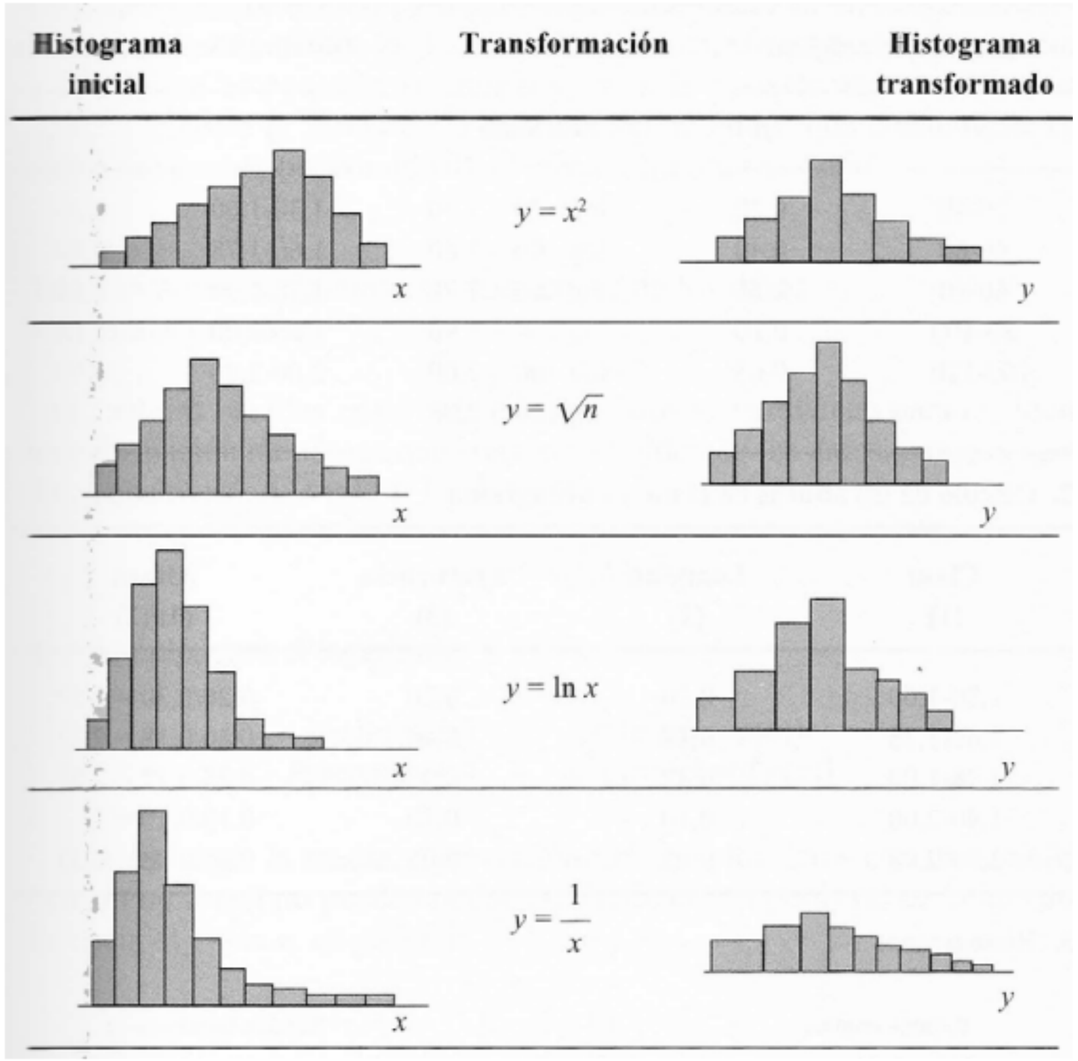
Los histogramas de la diapo anterior, existía un leve sesgo positivo:

```
> print(sesgo.ori)
[1] 0.1569923
> print(sesgo.mm)
[1] 0.1569923
> print(sesgo.z)
[1] 0.1569923
> print(sesgo.d)
[1] 0.1569923
```

```
> print(sesgo.tr.sq)
[1] 0.052761
> print(sesgo.tr.ln)
[1] -0.05237737
```



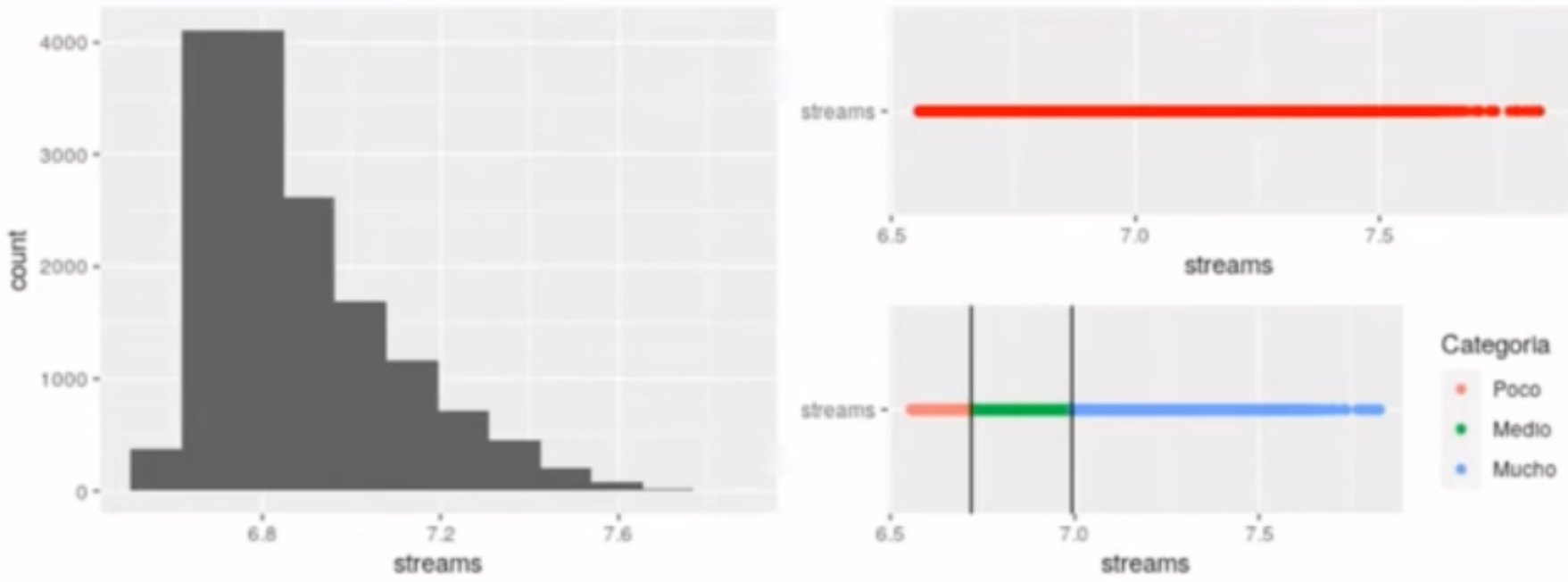
Ingeniería de Características



Ingeniería de Características

Discretización

- Es una técnica que permite dividir el rango de una variable continua en intervalos.
- Vamos de valores continuos a un número reducido de etiquetas.
- Esto conduce a una representación concisa y fácil de utilizar.



Ingeniería de Características

Discretización: Binning

- Es **Top-Down**.
- Se basa en un número específico de **bins**.
- Los criterios de agrupamiento pueden ser por:
- Igual-Frecuencia: La misma cantidad de observaciones en un bin.
- Igual-Ancho: Definimos rangos o intervalos de clases para cada bin.

A su vez para cada uno de los agrupamientos podemos hacer:

- Reemplazo por **media**
- Reemplazo por **mediana**
- O una etiqueta (valor entero)

No se utiliza la información de la clase, por lo tanto es **no supervisado**.

Ingeniería de Características

Discretización: Binning

Discretización Top-Down:

- **Enfoque:** Comienza con el rango completo de los datos y luego divide ese rango en intervalos discretos. Se basa en una visión general de los datos y en la definición de intervalos que abarcan todo el rango de valores posibles.
- **Proceso:**
  - Primero, se decide el número de intervalos o categorías.
  - Luego, se divide el rango total de los datos en esos intervalos. Estos intervalos pueden ser iguales en tamaño o pueden ser definidos en función de criterios específicos.
  - Cada dato se asigna al intervalo correspondiente.
- **Ventajas:**
  - Puede ser más sencillo de implementar si se tiene una idea clara de la cantidad de intervalos necesarios.
  - Es útil cuando se desea un control explícito sobre el número de categorías o intervalos.
- **Desventajas:**
  - Puede no capturar bien la variabilidad intrínseca de los datos si los intervalos no están bien ajustados.
  - Puede ser menos flexible si los datos tienen una distribución no uniforme.

Discretización Bottom-Up:

- **Enfoque:** Comienza con los datos individuales y luego agrupa estos datos en intervalos o categorías. Se basa en la estructura y distribución de los datos en lugar de imponer una división predefinida.
- **Proceso:**
  - Se analiza la distribución de los datos para identificar patrones, grupos o intervalos naturales.
  - A partir de esta información, se definen los intervalos o categorías de manera que reflejen mejor la estructura subyacente de los datos.
  - Cada dato se agrupa en el intervalo que mejor representa su valor.
- **Ventajas:**
  - Puede capturar mejor la distribución real de los datos.
  - Es más flexible y puede adaptarse a distribuciones no uniformes.
- **Desventajas:**
  - Puede ser más complejo de implementar, ya que requiere un análisis detallado de los datos para definir los intervalos.
  - Puede resultar en un número variable de intervalos dependiendo de la distribución de los datos.

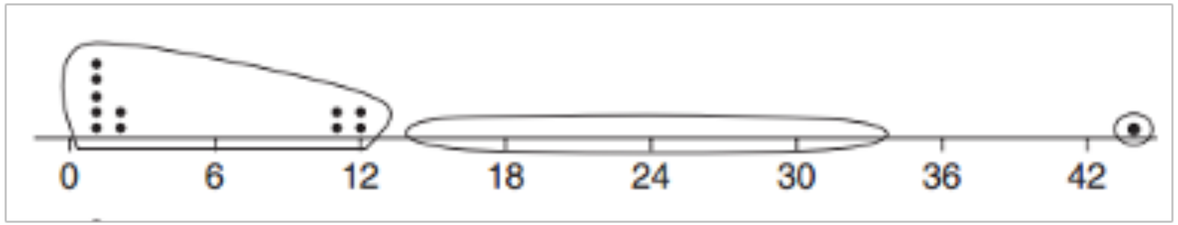
Ingeniería de Características

Discretización: Binning

Supongamos que vamos a discretizar **X** en 3 categorías.

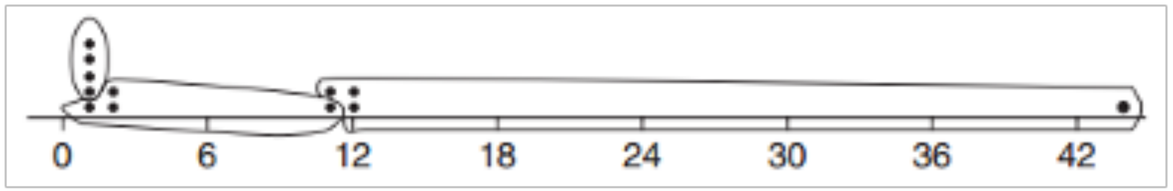
$$X = \{1, 1, 1, 1, 1, 2, 2, 11, 11, 12, 12, 44\}$$

Igual ancho



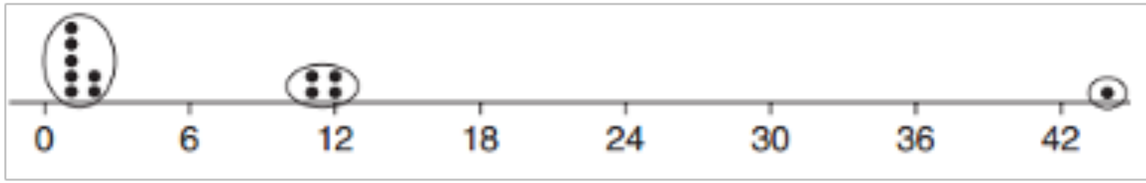

```
Bajo:  0 ≤ X < 15
Medio: 15 ≤ X < 30
Alto:  30 ≤ X < 45
```

Igual frecuencia




```
n = 12
bins = 3
n/bins = 4
```

K-means



Identifica lo que parece ser la partición intuitivamente correcta

Ingeniería de Características

Discretización: Otros no supervisados

- ***Rank***: El ranking de un número **es su tamaño relativo a otros valores** de una variable numérica. Primero, ordenamos la lista de valores, luego asignamos la posición de un valor como su rango.
- **Los mismos valores reciben el mismo rango** pero la presencia de valores duplicados afecta a las filas de valores posteriores (por ejemplo, 1,2,3,3,4).
- Rango es un sólido método de binning con un inconveniente importante, los valores pueden tener rangos diferentes en diferentes listas.

Ejemplo:

$X = \{1, 1, 1, 1, 1, 2, 2, 11, 11, 12, 12, 44\}$

$Rank_X = \{1, 1, 1, 1, 1, 2, 2, 3, 3, 4, 4, 5\}$

Ingeniería de Características

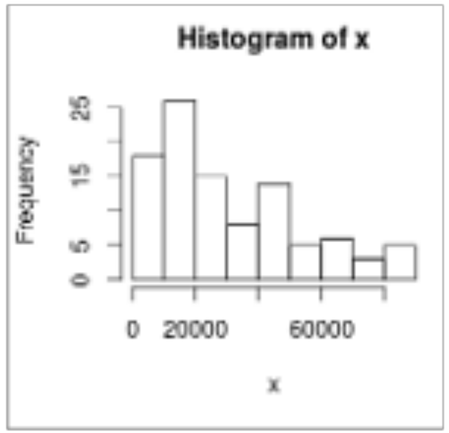
Discretización: Otros no supervisados

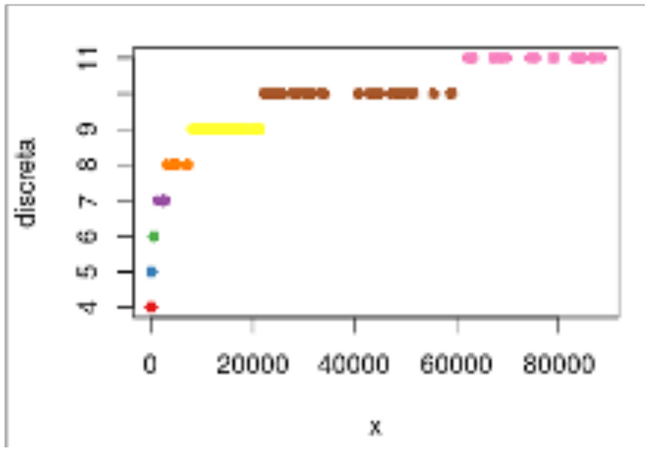
❏***Quantiles*** (median, quartiles, percentiles, ...): ***Quantiles*** también son métodos binning muy útiles pero como Rank, un valor puede tener cuantil diferente si la lista de valores cambia.
❏***Math functions***: Por ejemplo, FLOOR(LOG(X)) es un método binning efectivo para las variables numéricas con distribución altamente sesgada (por ejemplo, ingreso).

```
x = sample(10:100000, size = 100, replace = T, prob = seq(1.0, 0.0001,-0.00001))
```

```
summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     69   12958   23680   30513   44262   88312
```

```
unique(floor(log(x)))
[1] 11 10  8  9  5  7  6  4
```





Ingeniería de Características

Discretización basada en Entropia

La entropía es una medida de la incertidumbre o el desorden en un conjunto de datos. En el contexto de la teoría de la información, se utiliza para cuantificar la cantidad de información contenida en un conjunto de datos. Para una variable discreta Y con clases $C_1, C_2, C_3, \ldots, C_k$. La entropía se define como:

$$H(Y) = -\sum_{i=1}^{k} p(C_i)\log_2 p(C_i)$$

Donde $P(C_i)$ es la probabilidad de la clase i .

Ingeniería de Características

Discretización basada en Entropia

- Para variables continuas, el proceso es un poco más complejo. La idea es discretizar los datos en función de la entropía para encontrar los puntos de corte óptimos. Primero, se deben probar diferentes puntos de corte (o divisiones) en el rango continuo y calcular cómo afecta a la entropía.

- Para cada posible punto de corte t, divide el rango continuo en dos intervalos: $(-\infty, t], (t, \infty)$. Luego calcula la entropía condicional para cada intervalo:
  - Entropia del intervalo 1: $H(Y|X \leq t)$
  - Entropia del intervalo 2: $H(Y|X > t)$

  Donde X es la variable continua que se está discreteando y Y la variable objetivo.

  - Calcula la entropía total después de la discretización utilizando el punto de corte t. Esto se hace sumando las entropías ponderadas de los intervalos resultantes

  $H(Y|X \text{ después de discretización}) = P(X \leq t) * H(Y|X \leq t) + P(X > t) * H(Y|X > t)$.

  Donde $P(X \leq t)$ y $P(X > t)$ son las proporciones de datos que caen en los intervalos.

  - Compara la entropía total calculada para cada punto de corte. El mejor punto de corte es aquel que minimiza la entropía total después de la discretización, lo que indica que la división proporciona la mayor ganancia de información (es decir, la mayor capacidad para distinguir entre diferentes clases).

Ingeniería de Características

Discretización basada en Entropia

Ejemplo: Supongamos que se la variable continua "Edad" y se quiere discretizarla para predecir si un cliente es "Premium" o "No Premium".

Primero, calculamos la entropía de la variable de destino ("Clase de Cliente") sin ninguna discretización:
- Total de datos: 8
- Número de clientes Premium: 4
- Número de clientes No Premium: 4

Las probabilidades en cada clase son:
$$P(premium) = P(NoPremium) = 4/8 = 0.5.$$

La entropia inicial es:
$$H(Y) = -(0.5 * log_2(0.5) + 0.5 * log_2(0.5)) = 1.$$

Puntos de corte propuestos: 30,35 y 40.

Edad	Clase de Cliente
22	No Premium
25	No Premium
30	Premium
35	Premium
40	No Premium
45	Premium
50	Premium
55	No Premium

Ingeniería de Características

Discretización basada en Entropia

Punto de corte: 30 años.

- **Edad** ≤ 30**:**
- Número de clientes: 3
- Número de Premium: 1
- Número de No Premium: 2

-Probabilidades:

$$P(Premium) = \frac{1}{3}$$

$$P(NoPremium) = \frac{2}{3}$$

Entropia en este intervalo:

$$H(Y|X \leq 30) = -(1/3 * log_2(1/3) + 2/3 \log_2(2/3))$$

$$H(Y|X \leq 30) = 0.918$$

Punto de corte: 30 años.

- **Edad** > 30**:**
- Número de clientes: 5
- Número de Premium: 3
- Número de No Premium: 2

-Probabilidades:

$$P(Premium) = \frac{3}{5}$$

$$P(NoPremium) = \frac{2}{5}$$

Entropia en este intervalo:

$$H(Y|X > 30) = -(3/5 * log_2(3/5) + 2/5 \log_2(2/5))$$

$$H(Y|X > 30) = 0.971$$

Edad	Clase de Cliente
22	No Premium
25	No Premium
30	Premium
35	Premium
40	No Premium
45	Premium
50	Premium
55	No Premium

Ingeniería de Características

Discretización basada en Entropia

Edad	Clase de Cliente
22	No Premium
25	No Premium
30	Premium
35	Premium
40	No Premium
45	Premium
50	Premium
55	No Premium

Punto de corte: 30 años.

- **Edad** ≤ 30:

$$H(Y|X \leq 30) = -(1/3 * log_2(1/3) + 2/3 \log_2(2/3))$$

$$H(Y|X \leq 30) = 0.918$$

Punto de corte: 30 años.

- **Edad** > 30:

$$H(Y|X > 30) = -(3/5 * log_2(3/5) + 2/5 \log_2(2/5))$$

$$H(Y|X > 30) = 0.971$$

Entropia total para punto de corte 30 años:

$$H(Y|X \text{ después de discretización}) = P(X \leq 30) * H(Y|X \leq 30) + P(X > 30) * H(Y|X > 30)$$

$$H(Y|X \text{ después de discretización}) = \frac{3}{8} * 0.918 + \frac{5}{8} * 0.971 \approx 0.953$$

Ingeniería de Características

Discretización basada en Entropia

- Es supervisada y Separación Top-Down
- Explora la distribución de información en la clase para el cálculo y determinación del **split-point.**
- Para un dataset D $\rightarrow\{A_1,..,A_N\}$ el método para discretizar A es:

1. Cada **Valor de A** se considera como un posible **split-point** para hacer una discretización **binaria.**
2. Calculo la **Entropía** para la Clase

$$H(S) = \sum -p_i \ln p_i$$

3..Calcula la **Entropía** para la Clase y el **split-point** a evaluar

$$H(S,A) = \sum \frac{|S_v|}{|S|} H(S_v)$$

4. Calculo **Information Gain** para esa partición, como:

$$InformationGain = H(S) - H(S,A)$$

Ingeniería de Características

Discretización basada en Entropia

Discretizar la variable de temperatura usando el algoritmo basado en entropía.

O-Ring Failure	
Y	N
7	17

Paso 1: Calculamos Entropía para la variable objetivo.

$$H(Failure) = H(7,17) = -0.29 * log_2(0.29) - 0.71 * log_2(0.71) = 0.871$$

Paso 2: Calculamos Entropía para la variable objetivo dado un bin.

$$H(Failure, Temperature) = P(<=60) * H(3,0) + P(>60) * H(4,17) = \frac{3}{24} * 0 + \frac{21}{24} * 0.7 = 0.615$$

Paso 3: Calculamos Ganancia de Información} (GI) dado un bin.

$$GI = H(S) - H(S,A)$$

$$GI(Failure, Temperature) = 0.256$$

		O-Ring Failure	
		Y	N
Temperature	<= 60	3	0
	> 60	4	17

Ingeniería de Características

Recodificación de Variables

- Algunos métodos analíticos, como la regresión, requieren que los **predictores sean numéricos**.
- Cuando tenemos descriptores categóricos, podemos recodificar la variable categórica en una o más **variables Dummy o Flags o One-Hot encoding**.

Variables con dos categorias

```
If sex = female = then sex_flag = 0;
if sex = male then sex_flag = 1.
```

Variables con N categorias

```
north_flag:        If region = north then north_flag = 1; otherwise north_flag
= 0.
east_flag:         If region = east then east_flag = 1; otherwise east_flag =
0.
south_flag:        If region = south then south_flag = 1; otherwise south_flag
= 0.
```

Ingeniería de Características

Ñapa: índice GINI

En economía y sociológica, el índice de Gini es una medida de desigualdad en una distribución, como la distribución del ingreso o la riqueza entre una población. Se calcula a partir de la curva de Lorenz, que representa la proporción acumulada del ingreso o riqueza frente a la proporción acumulada de la población.

En el contexto de árboles de decisión, el índice de Gini se usa para medir la "impureza" de un nodo en el árbol. El objetivo es encontrar la división en los datos que minimice el índice de Gini, logrando así nodos más puros y homogéneos.La fórmula del índice de Gini para un nodo es:

$$G = 1 - \sum_{i=1}^{k} p_i^2$$

Donde p_i es la proporción de datos en el nodo que pertenecen a la clase i. k es el número total de clases.

UNIVERSIDAD
NACIONAL
DE COLOMBIA

Gracias