

Data Mining



Reducción de Dimensionalidad



Outline

- ❑ Reducción de dimensionalidad.

Reducción de datos

Estrategias de reducción

Reducción de datos: Obtener una representación reducida del conjunto de datos que es más pequeña en volumen pero aún produce los mismos (o casi los mismos) resultados analíticos.

¿Por qué reducir los datos? Una base de datos puede almacenar terabytes de datos. El análisis de datos complejo puede tomar mucho tiempo para ejecutarse en el conjunto completo de datos.

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**
 - ❑ Regression and Log-Linear Models
 - ❑ Histograms, clustering, sampling, Data cube aggregation
- ❑ **Compresión de datos**

Reducción de dimensionalidad

❑ Maldición de dimensionalidad

- ❑ Cuando aumenta la dimensionalidad, **los datos se vuelven cada vez más malos.**
- ❑ La densidad y la distancia entre los puntos, que es crítica para la agrupación, el análisis de valores atípicos, se vuelve menos significativa.
- ❑ Las posibles combinaciones de subespacios crecerán exponencialmente

❑ 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
- ❑ Permitir una visualización más fácil

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.
- ❑ Se calcula la proporción de faltantes y se busca un valor de corte que sea óptimo.
 - ❑ En un problema de clasificación podemos buscar el % de eliminación que maximiza la precisión.
- ❑ Este método aplica tanto a variables numéricas como categóricas.

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. **$\text{Var}[x] = p(1 - p)$**

Reducción utilizando χ^2

Podemos seleccionar *features* con los valores más altos para el estadístico de la prueba χ^2 entre la clase y cada *feature*.

- Aplica a Features categóricas y no-negativas como booleanos o frecuencias (conteos de términos en la clasificación de documentos).



La prueba de chi-cuadrado mide la dependencia, quitamos las variables con mayor probabilidad de ser independientes de la clase y, por lo tanto, irrelevantes para la clasificación.

Ejemplo x²

```
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 es lo más relacionado con el entre el X e Y

Ejemplo x²

```
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)
```

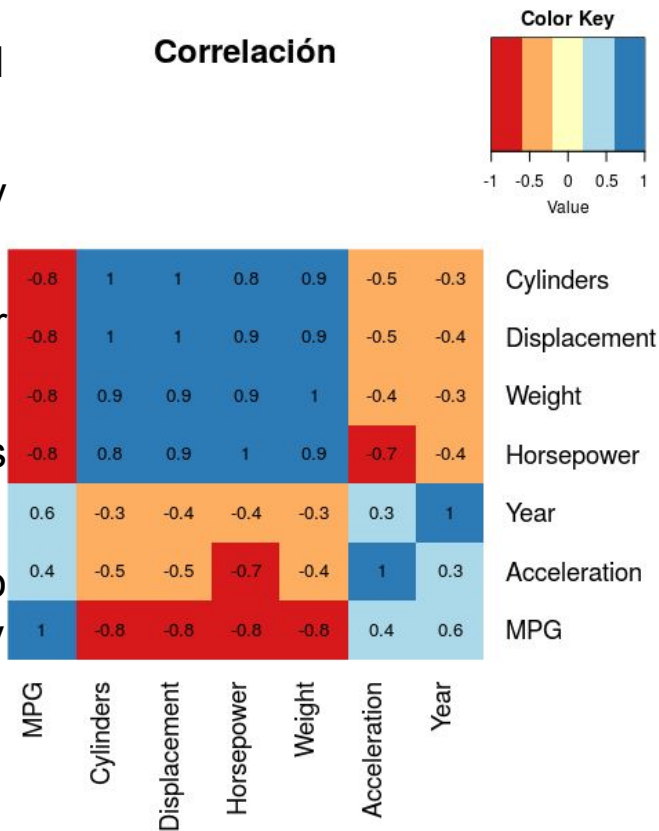
	attr_importance
V1	0.409330348
V2	0.004534049
V3	0.748864321
V4	0.923255954
V5	0.718768923
V6	0.428332508
V7	0.521967369
V8	0.661876085
V9	0.629797943
V10	0.083809300
V11	0.378240781
V12	0.714922593
V13	0.555971176
V14	0.625283342
V15	0.538263037
V16	0.353273580

```
print(formula)
```

```
Class ~ V4 + V3 + V5 + V12 + V8
```

Reducing Highly Correlated Columns

- ❑ 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 **Coefficiente de correlación de Pearson** y **Prueba χ^2 de Pearson**.



Reducing Highly Correlated Columns

Ejemplo: Dataset [cars](#) y el paquete caret


```
library(caret)
ds.cars = read.csv('./cars.csv', sep=';', dec = ',')
head(ds.cars)
ds.cor = cor(ds.cars[2:8], use = "complete.obs")
```

```
print(ds.cor)
summary(ds.cor[upper.tri(ds.cor)])[c(1,6)]
```


Min.	Max.
-0.8322442	0.9508233

```
vars.cor.alta <- findCorrelation(ds.cor, cutoff=0.8)
print(names(ds.cars)[vars.cor.alta])
ds.cars.filtrado <- ds.cars[, -vars.cor.alta]
```

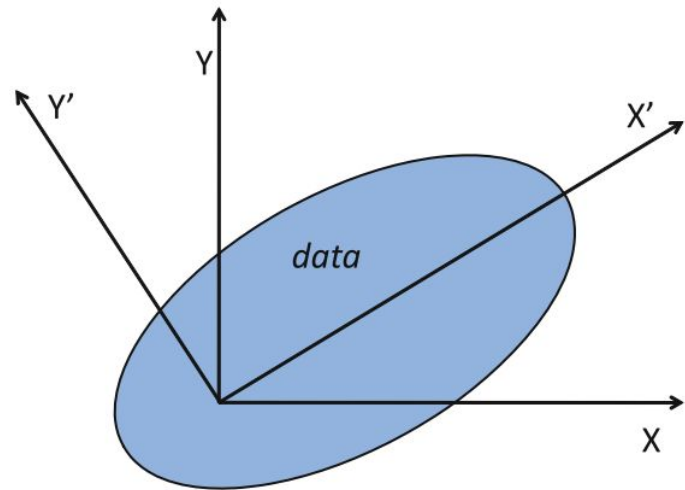
```
ds.cor2 = cor(ds.cars.filtrado[2:4], use = "complete.obs")
summary(ds.cor2[upper.tri(ds.cor2)])[c(1,6)]
```



Min.	Max.
-0.4146753	0.2951990

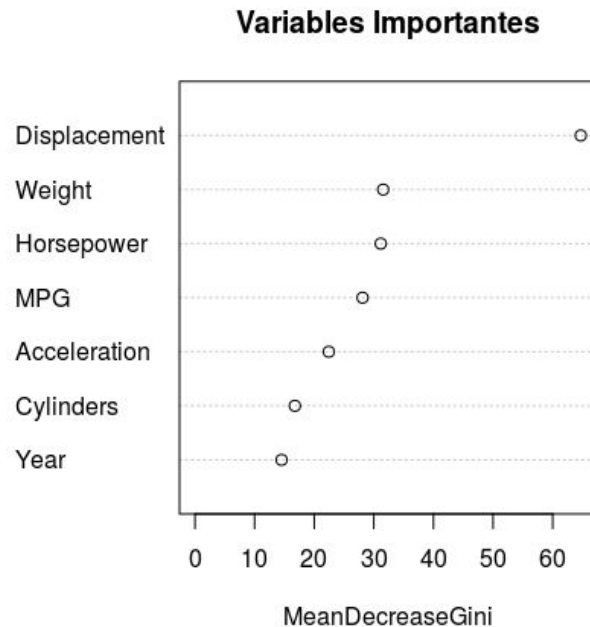
PCA - Componentes Principales

- ❑ Encuentra una proyección que capture 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



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.



Backward Feature Elimination

Recursion feature elimination

- ❑ Backward Feature Elimination 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.

Forward Feature Construction

- ❑ 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.

Bibliografía

- ❑ Jiawei Han, Micheline Kamber, Jian Pei. 2012. Tercera edición. Data Mining: Concepts and Techniques. Cap. 3