

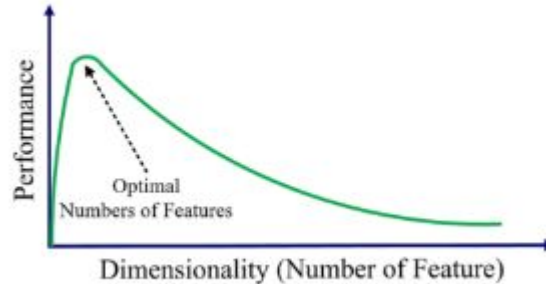
Machine Learning

Reducción de dimensionalidad

Christian Oliva Moya
Luis Fernando Lago Fernández

Maldición de la dimensionalidad

- A medida que aumenta el número de atributos (dimensiones) surgen ciertos problemas:
 - Densidad de datos: Conforme aumenta la dimensionalidad, la cantidad de datos para llenar ese espacio aumenta exponencialmente.
 - Sobreajuste: Con más dimensiones, los modelos pueden tener overfitting con más facilidad



Maldición de la dimensionalidad

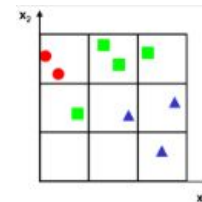
- Veámoslo con un ejemplo:

- Tengo mis datos en una única dimensión con 3 bloques:



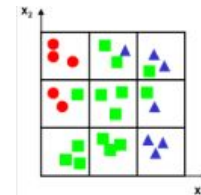
en cada región del espacio tengo 3 puntos: OK

- Aumentamos la dimensión a 2, por tanto tengo $3^2 = 9$ bloques:



ahora cada región se ha quedado vacía. Necesito rellenar con más datos

relleno cada región del espacio con 3 puntos ($3^3 = 27$ en total): OK

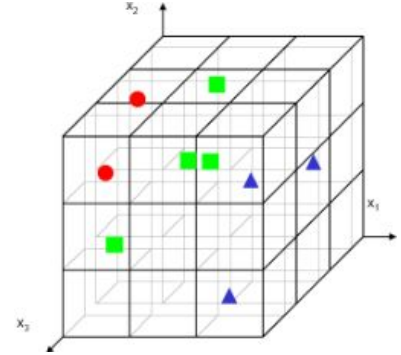


Maldición de la dimensionalidad

- Veámoslo con un ejemplo:
 - Aumentamos la dimensión a 3, por tanto tengo $3^3 = 27$ bloques:

cada región se ha quedado más vacía. Necesito rellenar:

relleno cada región del espacio con 3 puntos ($3^4 = 81$ en total)
- La densidad de datos necesita aumentar **de forma exponencial**



Maldición de la dimensionalidad

- ¿Qué puedo hacer con esta maldición?
 - Aumentar los datos de entrenamiento de forma exponencial
 - Reducir la dimensionalidad de mi dataset

Si tienes dos modelos con el mismo rendimiento, elige siempre el menos complejo

menos atributos → más fácil de interpretar

Reducción de la dimensionalidad

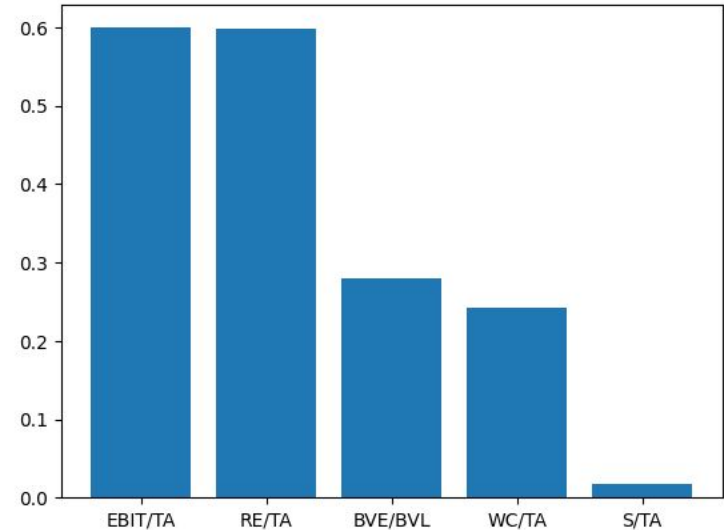
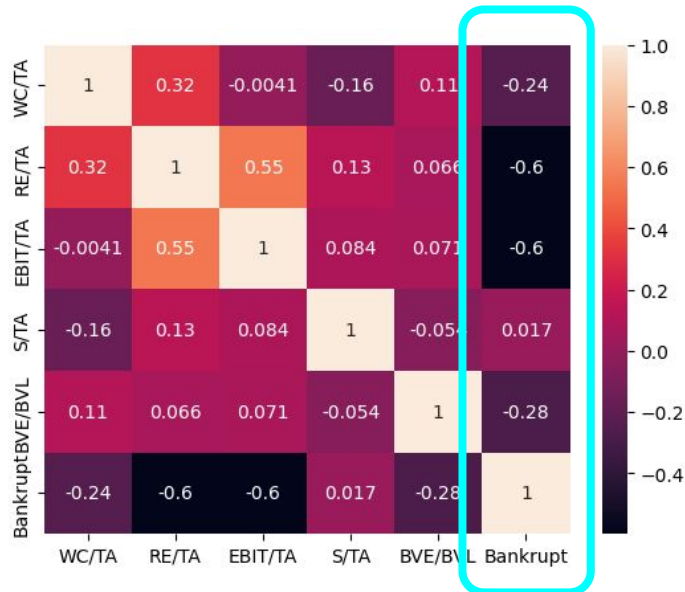
- Proceso por el cual se reduce el número de atributos originales. Hay dos estrategias:
 - Selección de atributos
 - Extracción de atributos

Reducción de la dimensionalidad - Selección de atributos

- En muchas aplicaciones reales, más que la predicción, lo interesante es encontrar una explicación. ¿Qué atributos influyen en mi resultado?
- Queremos encontrar una **métrica de relevancia** $R(x)$ sobre la predicción de mi modelo M que nos indique **cuánto contribuye** el atributo x a la decisión de mi modelo
- ¿Para qué?
 - Entender los datos e interpretar el comportamiento del modelo
 - Seleccionar características para reducir la dimensionalidad
- Objetivo: Hacer un **ranking con las N mejores** características

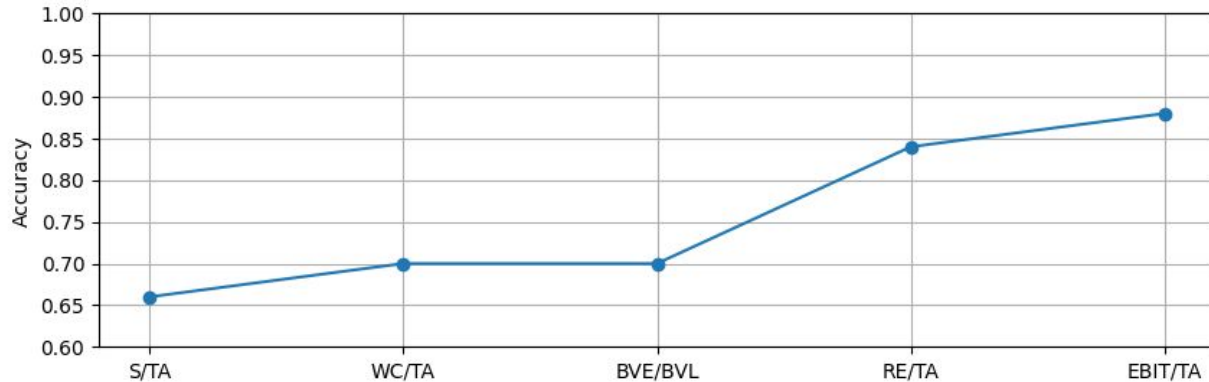
Reducción de la dimensionalidad - Selección de atributos

- Importancia por Correlación:
 - Seleccionar los atributos con **mayor correlación con la clase en valor absoluto**



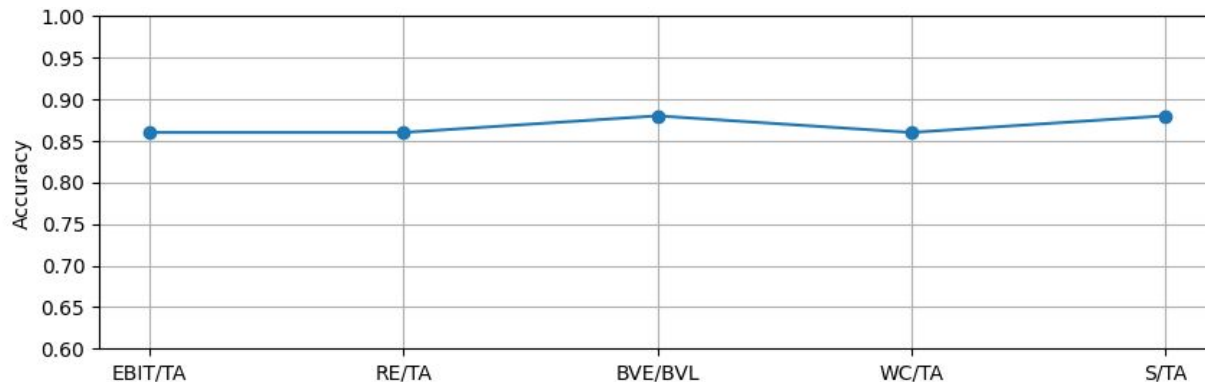
Reducción de la dimensionalidad - Selección de atributos

- Importancia por Correlación:
 - Seleccionar los atributos con **mayor correlación con la clase en valor absoluto**
- Si vamos incluyendo al dataset de **menor a mayor** relevancia:



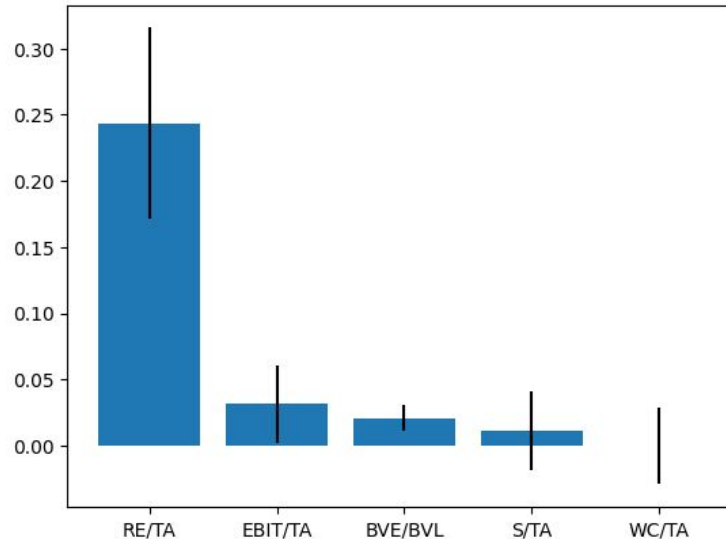
Reducción de la dimensionalidad - Selección de atributos

- Importancia por Correlación:
 - Seleccionar los atributos con **mayor correlación con la clase en valor absoluto**
- Si vamos incluyendo al dataset de **mayor a menor** relevancia:



Reducción de la dimensionalidad - Selección de atributos

- Importancia por Permutación:
 - Calcula la pérdida en la métrica de evaluación cuando cambiamos un atributo x
 - Buscamos cuánto depende el modelo M del atributo x



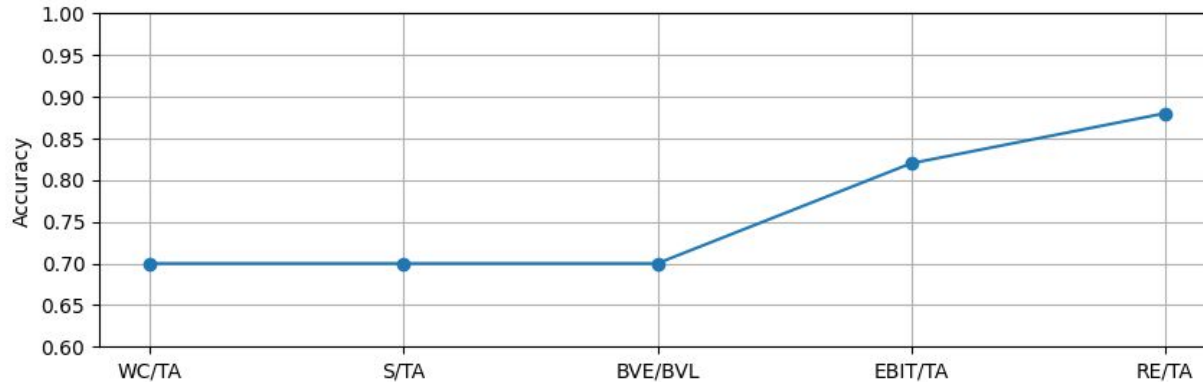
Reducción de la dimensionalidad - Selección de atributos

- Importancia por Permutación (algoritmo):
 - Para cada atributo x repetir K veces:
 - Permutar los valores de x para generar un dataset alterado D'
 - Evaluar sobre D' para obtener un score s'
 - La relevancia de x es:

$$R(x) = s - \frac{1}{k} \sum_{i=1}^n s'_i$$

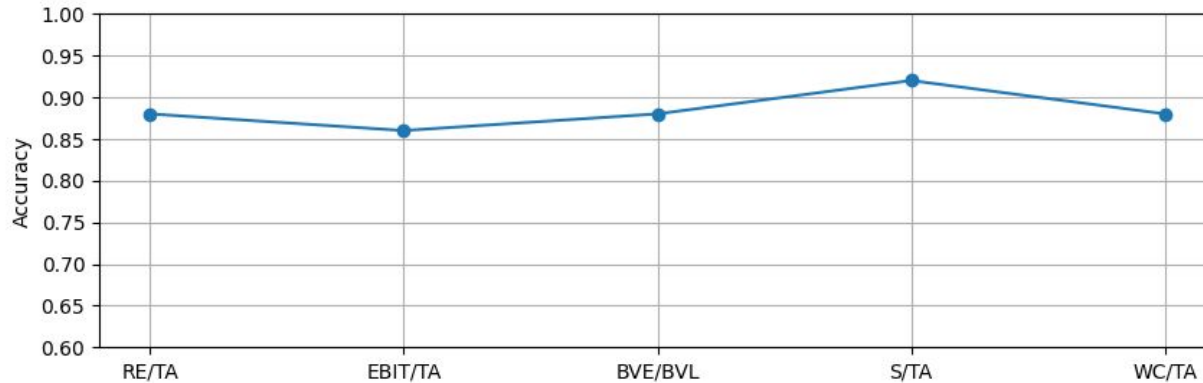
Reducción de la dimensionalidad - Selección de atributos

- Importancia por Permutación:
 - Buscamos cuánto depende el modelo del atributo x
- Si vamos incluyendo al dataset de **menor a mayor** relevancia:



Reducción de la dimensionalidad - Selección de atributos

- Importancia por Permutación:
 - Buscamos cuánto depende el modelo del atributo x
- Si vamos incluyendo al dataset de **mayor a menor** relevancia:

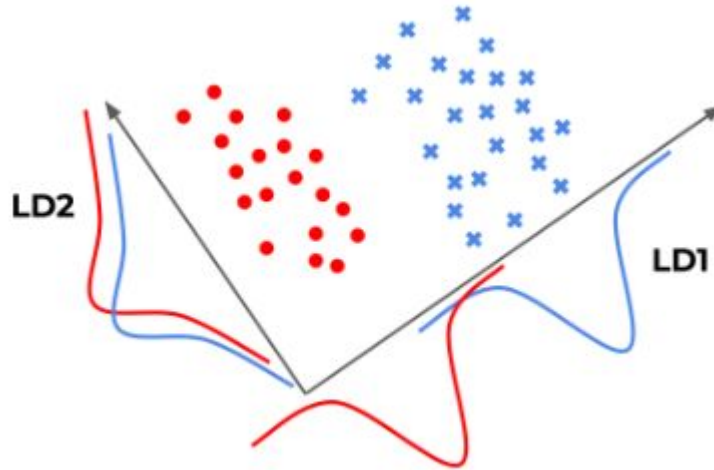


Reducción de la dimensionalidad - Extracción de atributos

- Queremos transformar los atributos a un nuevo conjunto de menor dimensión sin perder la información inicial. Hay dos tipos:
 - Métodos lineales:
 - Linear Discriminant Analysis (LDA) → Supervisado
 - Principal Component Analysis (PCA) → No supervisado
 - Métodos no lineales:
 - Locally Linear Embedding (LLE)
 - Isomap
 - Autoencoders

Reducción de la dimensionalidad - Extracción de atributos

- Linear Discriminant Analysis (LDA) → Supervisado
 - Extrae nuevos atributos que mantengan la información de la clase
 - Maximiza la separabilidad entre clases



Reducción de la dimensionalidad - Extracción de atributos

- Principal Component Analysis (PCA) → No supervisado
 - Extrae nuevos atributos que mantengan la varianza del espacio original
 - Intenta obtener variables no correlacionadas (componentes principales)
 - Las componentes quedan ordenadas en términos de la varianza explicada

