Conjuntos de entrenamiento y prueba

Minería de Datos (CC3074) - 2026

Conjuntos de Entrenamiento y Prueba

Semestre 01, 2026

El problema

- Queremos construir un modelo que generalice
- No basta con que funcione en los datos conocidos
- Necesitamos medir desempeño en datos no vistos

Aprendizaje Supervisado

- Tenemos variables predictoras (X)
- Tenemos una variable objetivo (Y)
- Queremos aprender una función $$(f(X) \rightarrow Y) $$

El modelo aprende a partir de datos etiquetados.

Riesgos

Si evaluamos el modelo con los mismos datos con los que entrenamos:

- Obtendremos resultados optimistas
- No medimos generalización
- Caemos en sobreajuste

División de datos

Dividimos el dataset en:

- Conjunto de entrenamiento
- Conjunto de prueba

Cada uno cumple un rol distinto.

Conjunto de entrenamiento

- Se usa para ajustar el modelo
- Permite estimar parámetros
- Es donde el modelo “aprende”

Conjunto de prueba

- Se usa únicamente para evaluar
- Simula datos nuevos
- No participa en el entrenamiento

Mide la capacidad de generalización.

Esquema general

Datos totales

→ Separación

→ Entrenamiento

→ Evaluación en prueba

Proporciones comunes

- 70% entrenamiento / 30% prueba
- 80% entrenamiento / 20% prueba (no recomendado)
- Depende del tamaño del dataset

Técnicas de muestreo

Técnica 1: Muestreo Aleatorio Simple

- Selección aleatoria de observaciones
- Cada fila tiene la misma probabilidad
- No considera la variable objetivo

Desventaja: Puede generar desbalance en las clases.

Ejemplo

```
numFilasTrain <- sample(nrow(iris), 0.7*nrow(iris))
train_iris <- iris[numFilasTrain,]
```

```
test_iris  <- iris[-numFilasTrain,]
```

Divide 70% entrenamiento 30% prueba

Técnica 2: Muestreo Estratificado

- Se divide el dataset por clases (estratos)
- Se toma el mismo porcentaje de cada clase
- Mantiene la distribución original

Reduce riesgo de desbalance.

Ejemplo

```
setosas <- iris[iris$Species=="setosa",]
versicolors <- iris[iris$Species=="versicolor",]
virginicas <- iris[iris$Species=="virginica",]

numFilasTrainSetosa <- sample(nrow(setosas), 0.7*nrow(setosas))
```

Se repite para cada clase Luego se combinan con `rbind()`

Técnica 3: Reproducibilidad

- Controla la aleatoriedad
- Permite repetir exactamente la misma partición
- Fundamental para experimentos científicos

Ejemplo

```
set.seed(123)
sample(1:10,3)
```

Sin `set.seed()` → resultado distinto cada vez

Con `set.seed()` → mismo resultado siempre

Error común

Evaluar sobre datos de entrenamiento.

Resultado:

- Métricas infladas
- Modelo sobreajustado
- Mala generalización

Data Leakage

- Información del conjunto de prueba se filtra al entrenamiento
- Escalamiento hecho antes de dividir
- Feature engineering mal aplicado

Produce resultados irreales.

Buenas prácticas

- Dividir antes de transformar
- Mantener el test intacto
- Usar validación cruzada cuando sea posible
- Reportar métricas en datos no vistos