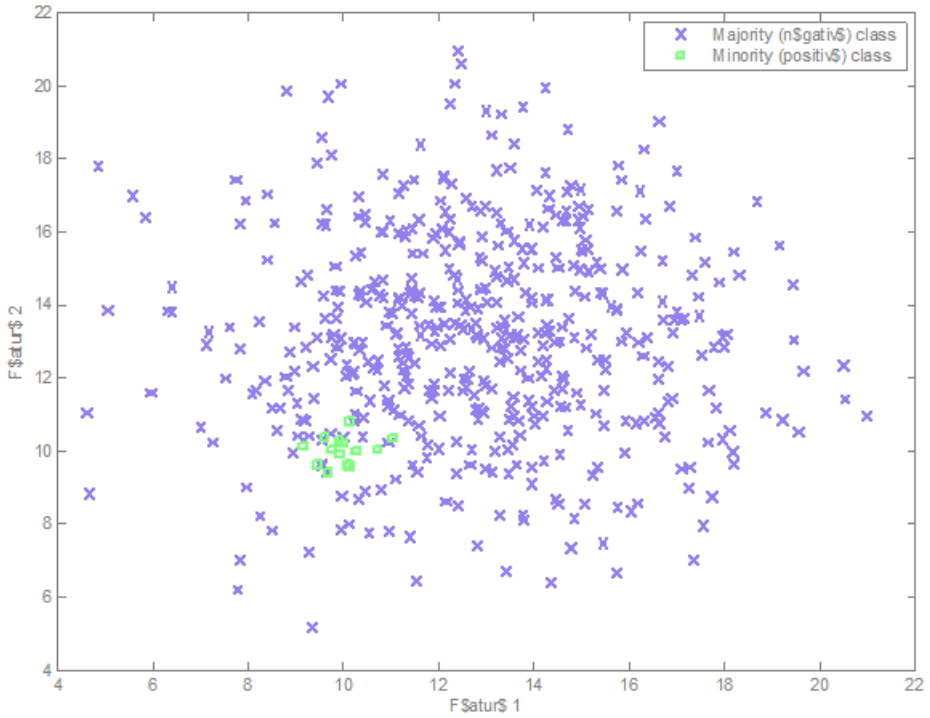
# TEMA 4: PROBLEMAS NO BALANCEADOS TÉCNICAS DE MUESTREO

# Problemas de clasificación no balanceados




Problemas cuya distribución de clases no es homogénea

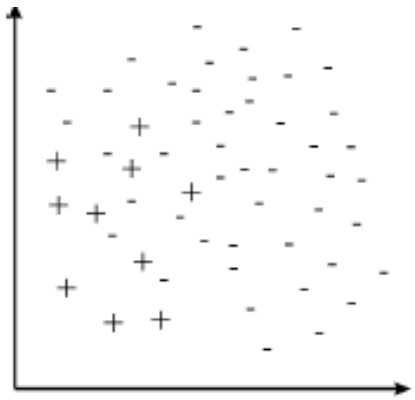
- Clase positiva: la de menor número de ejemplos
- Clase negativa: la de mayor número de ejemplos

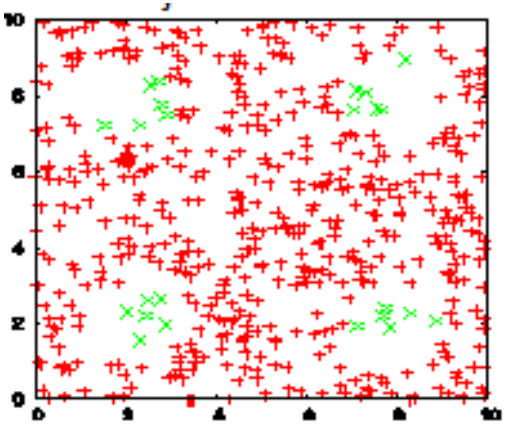
# Problemas de clasificación no balanceados

- Imbalanced Ratio (IR)

$$IR = \frac{nº\ ejemplos\ clase\ negativa}{nº\ ejemplos\ clase\ positiva}$$

- Factores que influyen en la dificultad del problema
  - IR
  - Grado de overlap entre las clases
  - Grupos pequeños de ejemplos (small disjunts)





# Problemas de clasificación no balanceados

- Medidas de rendimiento

| | | Clasificación como | |
|---|---|---|---|
| | | Si | No |
| Clase real | SI | Verdadero positivo (VP) | Falso negativo (FN) |
| | NO | Falso Positivo (FP) | Verdadero Negativo (VN) |

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

$$Error = 1 - Accuracy$$

# Problemas de clasificación no balanceados

- Clasificación de cáncer
  - Si solo el 0.5% de los pacientes tienen cáncer…
  - Prediciendo siempre “no cáncer”
    - 99.5% de acierto
- En este marco
  - El ratio de clasificación no refleja la calidad del clasificador
  - Se hacen necesarias otras medidas de evaluación

# Problemas de clasificación no balanceados

- Recall: ejemplos de la clase positiva clasificados correctamente
  - También llamada "True positive rate" o sensitividad
- Precision: proporción de ejemplos clasificados en la clase positiva que son realmente de la clase positiva

$$TPR = Recall = \frac{VP}{VP + FN}$$

$$Precision = \frac{VP}{VP+FP}$$

- Ejemplo: filtro spam
  - Recall: Proporción de spam identificado por el sistema
  - Precision: Proporción de spam dentro de la carpeta spam

| | | Clasificación como | |
|---|---|---|---|
| | | Si | No |
| Clase real | SI | Verdadero positivo (VP) | Falso negativo (FN) |
| | NO | Falso Positivo (FP) | Verdadero Negativo (VN) |

# Problemas de clasificación no balanceados

- Especificidad: ejemplos de la clase negativa clasificados correctamente
  - También llamada "True negative rate"

$$TNR = Especificidad = \frac{VN}{VN + FP}$$

| | | Clasificación como | |
|---|---|---|---|
| | | Si | No |
| Clase real | SI | Verdadero positivo (VP) | Falso negativo (FN) |
| | NO | Falso Positivo (FP) | Verdadero Negativo (VN) |

# Problemas de clasificación no balanceados

- Métricas balanceadas
  - F-score (Recall=TPR)

$$F_{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

  - Media geométrica

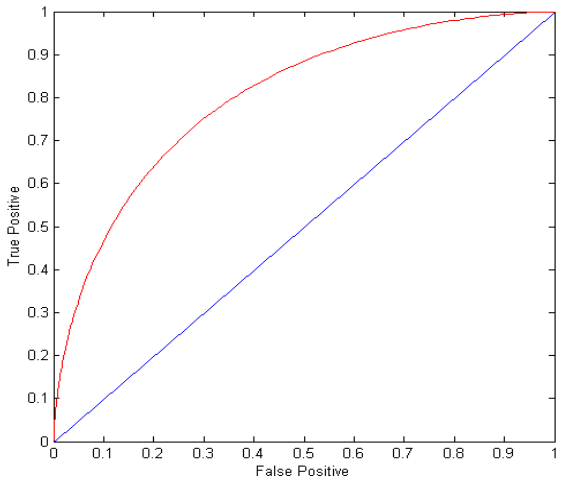
$$GM = \sqrt{\text{Re}\,call \times TNR}$$

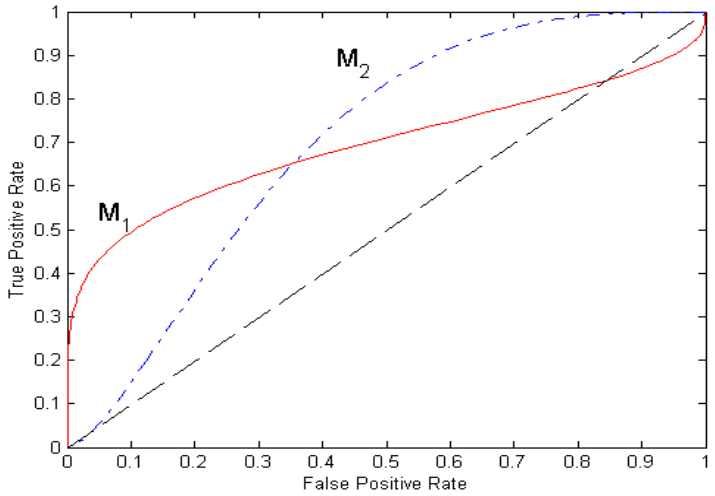
- Basadas en las curvas ROC
  - Área bajo la curva (AUC)

$$AUC_{1\,punto} = \frac{\text{Re}\,call + TNR}{2}$$

# Problemas de clasificación no balanceados

- Métricas basadas en las curvas ROC
  - Área bajo la curva (AUC)
- La curva ROC muestra el TPR (en el eje y) contra FPR (en el eje x)
  - Utilizando diferentes umbrales de probabilidad para realizar la clasificación
- Caracteriza el balance entre aciertos de la clase positiva y falsos positivos (falsas alarmas)

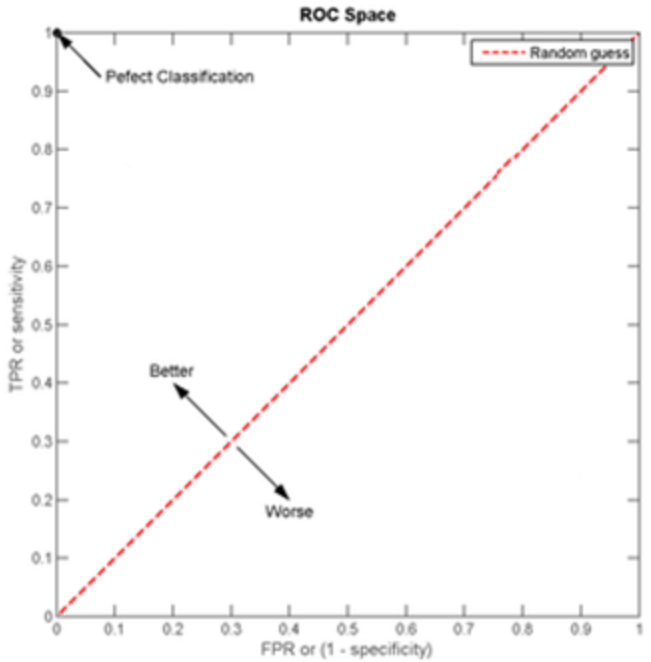



$$FPR = \frac{FP}{FP + VN} = 1 - TNR$$

# Problemas de clasificación no balanceados

- (FPR,TPR):
  - Diagonal: Clasificador aleatorio
  - (0,0): todos los ejemplos clasificados en la clase negativa
  - (1,1): todos los ejemplos clasificados en la clase positiva
  - (0,1): clasificador ideal



# Problemas de clasificación no balanceados

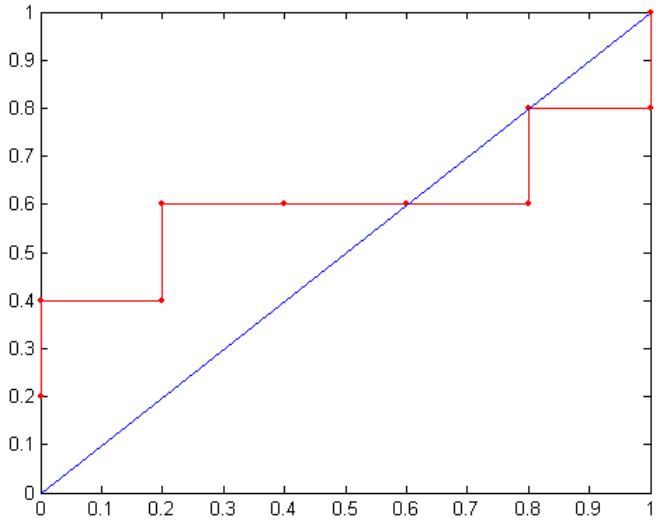
- Construcción de la curva ROC
  - Clasificar todos los ejemplos y almacenar para cada ejemplo la probabilidad de ser clasificado en la clase positiva
  - Ordenar las probabilidades de mayor a menor
  - Por cada valor de probabilidad
    - El valor se utiliza como umbral
    - Todos los ejemplos con probabilidades mayores o iguales que el umbral se clasifican como positivos y el resto como negativos
    - Calcular el TPR y FPR del umbral

- Se puede seleccionar un umbral a partir de ella

# Problemas de clasificación no balanceados

| Ej. | P+ | Clase real |
|---|---|---|
| 1 | 0.95 | + |
| 2 | 0.93 | + |
| 3 | 0.87 | - |
| 4 | 0.85 | - |
| 5 | 0.85 | - |
| 6 | 0.85 | + |
| 7 | 0.76 | - |
| 8 | 0.53 | + |
| 9 | 0.43 | - |
| 10 | 0.25 | + |

| Class | + | - | + | - | - | - | + | - | + | + | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.25 | 0.43 | 0.53 | 0.76 | 0.85 | 0.85 | 0.85 | 0.87 | 0.93 | 0.95 | 1.00 |
| TP | 5 | 4 | 4 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 0 |
| FP | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 1 | 0 | 0 | 0 |
| TN | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 5 |
| FN | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 5 |
| TPR | 1 | 0.8 | 0.8 | 0.6 | 0.6 | 0.6 | 0.6 | 0.4 | 0.4 | 0.2 | 0 |
| FPR | 1 | 1 | 0.8 | 0.8 | 0.6 | 0.4 | 0.2 | 0.2 | 0 | 0 | 0 |



# Problemas de clasificación no balanceados

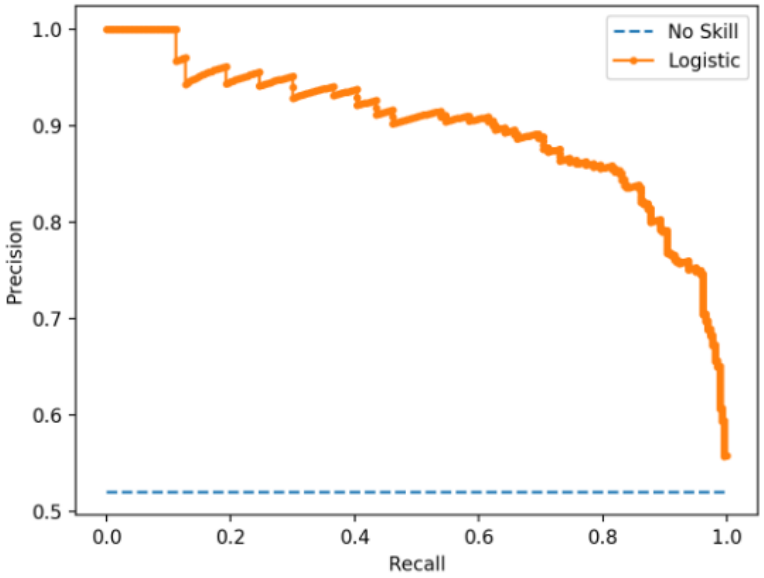
- ¿Por qué el clasificador aleatorio es una diagonal?
- Suposiciones
  - Porcentaje de ejemplos de la clase positiva: $p$
    - De la clase negativa: $1 - p$
  - Clasificador asigna aleatoriamente la clase positiva con probabilidad $u$
    - De la negativa con probabilidad: $1 - u$
  - La matriz de confusión tendrá las siguientes proporciones
    - $TP$: $u * p$
    - $FP$: $u * (1 - p)$
    - $TN$: $(1 - u) * (1 - p)$
    - $FN$: $(1 - u) * p$
  - Por tanto
    - $TPR = \frac{TP}{TP+FN} = \frac{u*p}{u*p+(1-u)p} = u$
    - $FPR = \frac{FP}{TN+FP} = \frac{u*(1-p)}{(1-u)*(1-p)+u*(1-p)} = u$

# Problemas de clasificación no balanceados

- Basadas en la curva Precision-Recall (PR)
  - Área bajo la curva PR
  - La curva PR similar a una curva ROC pero mostrando el balance entre precision (eje y) y recall (eje x)

$$Recall = TPR = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP}$$

  - No utiliza TN
  - Para construirlo, mismo procedimiento que para la curva ROC pero calculando pares precision-recall




- Se puede seleccionar un umbral

# Problemas de clasificación no balanceados

- ¿Por qué el clasificador aleatorio es un horizontal?
  - Igual al porcentaje de ejemplos de la clase positiva (la de menos ejemplos)
- Suposiciones
  - Porcentaje de ejemplos de la clase positiva: $p$
    - De la clase negativa: $1 - p$
  - Clasificador asigna aleatoriamente la clase positiva con probabilidad $u$
    - De la negativa con probabilidad: $1 - u$
  - La matriz de confusión tendrá las siguientes proporciones
    - $TP$: $u * p$
    - $FP$: $u * (1 - p)$
    - $TN$: $(1 - u) * (1 - p)$
    - $FN$: $(1 - u) * p$
  - Por tanto
    - $Recall = \frac{TP}{TP+FN} = \frac{u*p}{u*p+(1-u)p} = u$
    - $Precision = \frac{TP}{TP+FP} = \frac{u*p}{u*p+u*(1-p)} = p$

# Problemas de clasificación no balanceados

- Medidas que tienen en cuenta el coste de los fallos
- Matriz de costes
  - En cada celda se establece el coste de cada situación (fallo o acierto)
    - C(i|j): Coste de clasificar un ejemplo de la clase j como clase i

| | | Clase Predicha | |
|---|---|---|---|
| | C(i\|j) | **Clase=Si** | **Clase=No** |
| Clase Real | **Clase=Si** | C(Si\|Si) | C(No\|Si) |
| | **Clase=No** | C(Si\|No) | C(No\|No) |

  - Coste del modelo: Se multiplica cada celda de la matriz de costes por la respectiva de la matriz de confusión y se suman todas
  - El coste de clasificación será proporcional a la precisión del clasificador sólo si

$$\forall i,j:\ i \neq j \quad C(i|j) = C(j|i)$$
$$C(i|i) = C(j|j)$$

# Problemas de clasificación no balanceados

- Ejemplo

| Matriz coste | Clase Predicha | | |
|---|---|---|---|
| Clase real | C(i\|j) | + | - |
| | + | 0 | 100 |
| | - | 1 | 0 |

| Matriz confusion Modelo $M_1$ | Clase Predicha | | |
|---|---|---|---|
| Clase real | | + | - |
| | + | 150 | 40 |
| | - | 60 | 250 |

Accuracy = 80%

Coste = 4060

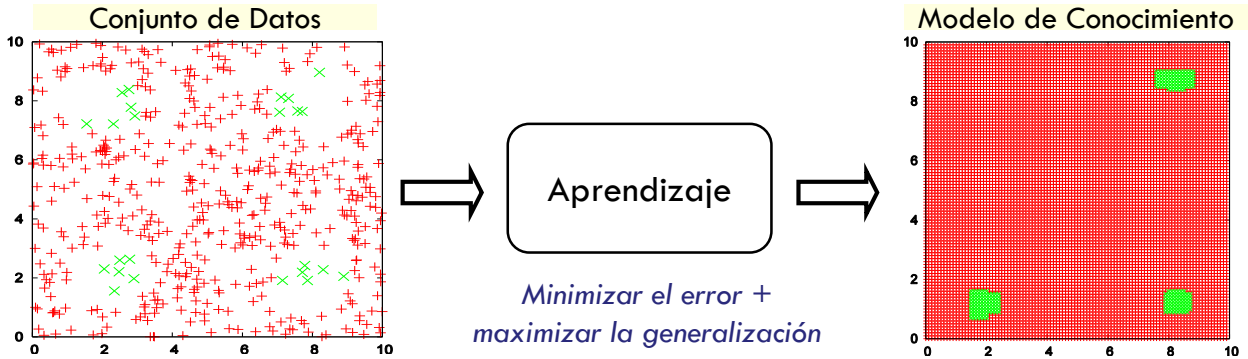
| Matriz confusion Modelo $M_2$ | Clase Predicha | | |
|---|---|---|---|
| Clase real | | + | - |
| | + | 250 | 45 |
| | - | 5 | 200 |

Accuracy = 90%

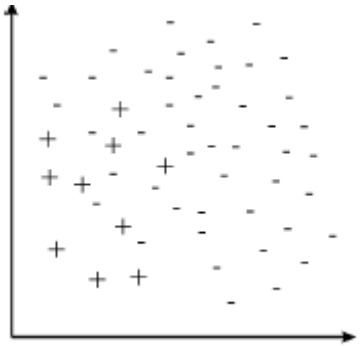
Coste = 4505

# Problemas para el aprendizaje con datos no balanceados

1. Proceso de búsqueda guiado por la tasa de acierto estándar
2. Las reglas de clasificación sobre la clase positiva altamente especializadas



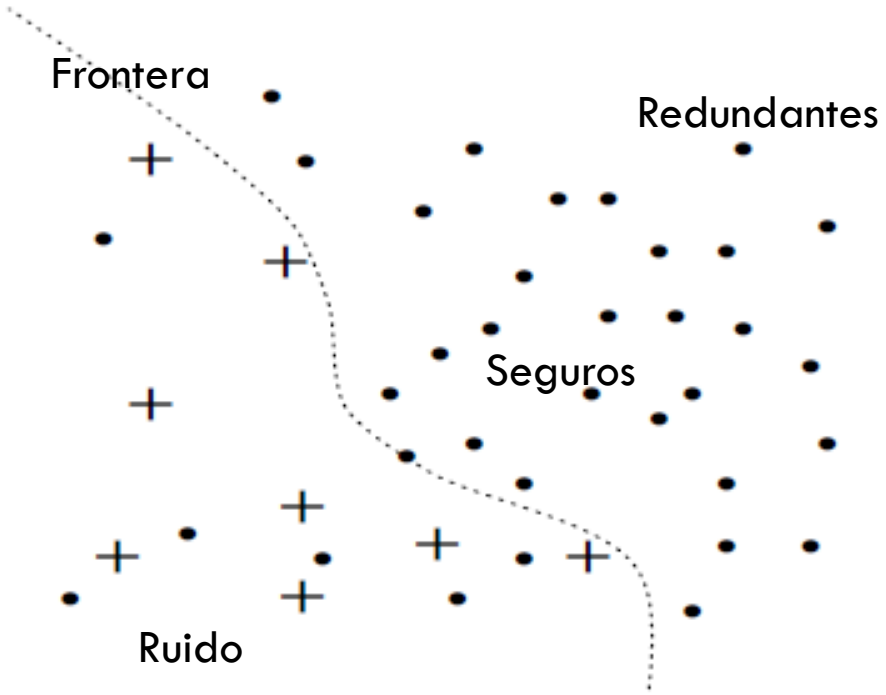

3. Distinción entre ejemplos ruidosos y ejemplos de la clase positiva
4. Solapamiento entre los ejemplos de distintas clases



# Tipos de ejemplos en problemas no balanceados

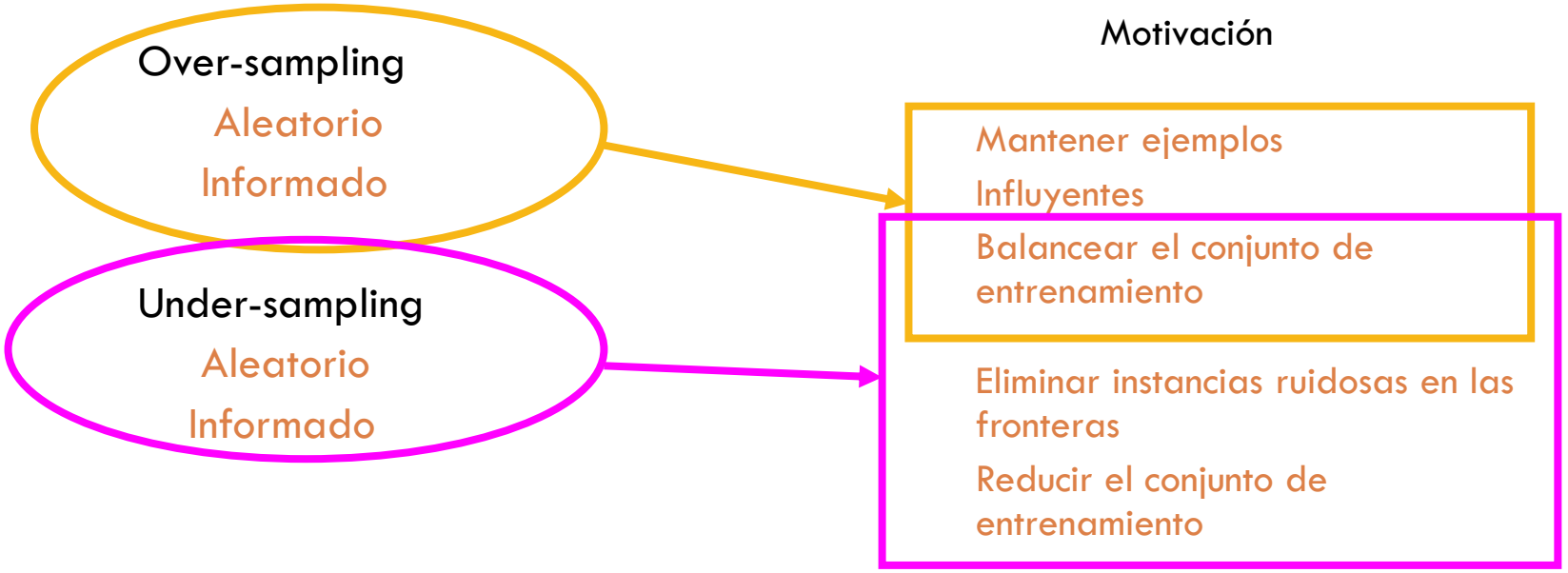
Tipos de ejemplos negativos:

1. Ruido
2. Frontera
3. Redundante
4. Seguros



# Problemas no balanceados: Soluciones

1. A nivel de datos: pre-procesamiento de ejemplos (muestreo)



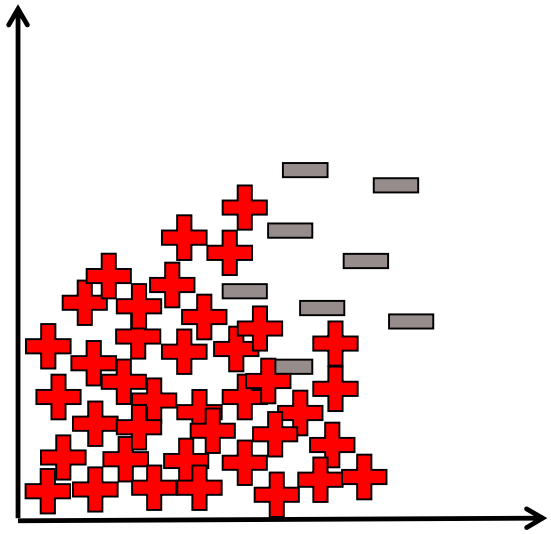

2. A nivel algorítmico: modificar características del algoritmo base (Sensible al Coste)
3. Ensembles

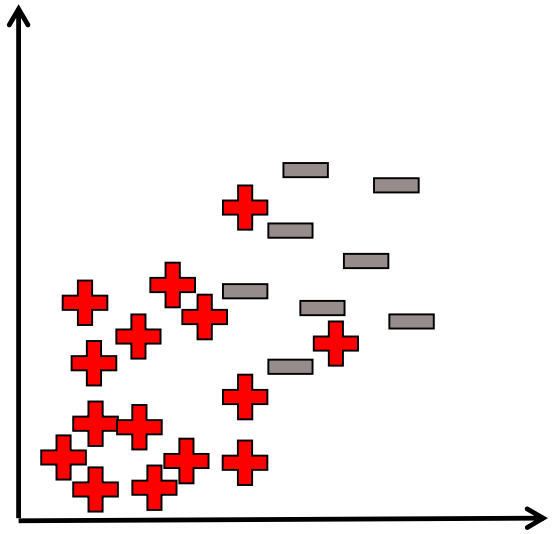
# Comentarios generales

- El muestreo se ejecuta sobre el conjunto de entrenamiento para facilitar el aprendizaje
  - El algoritmo de aprendizaje se ejecuta sobre el conjunto de datos muestreado
- El conjunto de test NO se muestrea
- Para obtener el rendimiento en entrenamiento se utiliza el conjunto de entrenamiento original, no el muestreado

# Técnicas de under-sampling

- Random under-sampling
  - Método no heurístico
  - Elimina aleatoriamente ejemplos de la clase negativa hasta balancear el conjunto de ejemplos
  - Problema: puede eliminar ejemplos potencialmente útiles para el aprendizaje





# Técnicas de under-sampling

- Condensed nearest neighbour rule (CNN)
  - Un subconjunto E' es consistente con E si usando 1NN, todos los ejemplos de E se clasifican correctamente utilizando E' como conjunto de entrenamiento
- Objetivo: Obtener el subconjunto consistente E'
- Idea
  - Eliminar los ejemplos de la clase negativa que estén lejos de la frontera de decisión
- Etapas de CNN
  - Crear E' inicial:
    - Incluir todos los ejemplos de la clase positiva y uno de la negativa (elegido aleatoriamente)
    - Eliminar los ejemplos de E' de E
  - Aplicar 1NN para clasificar E usando E', mover todos los ejemplos fallados de E a E'
  - Repetir hasta que no haya ejemplos fallados
- Problemas
  - Puede que no se obtenga el conjunto consistente E' mínimo
  - Sensible al ruido: los ejemplos ruido serán fallados y por tanto considerados en E', lo que puede afectar al rendimiento posterior

# Técnicas de under-sampling

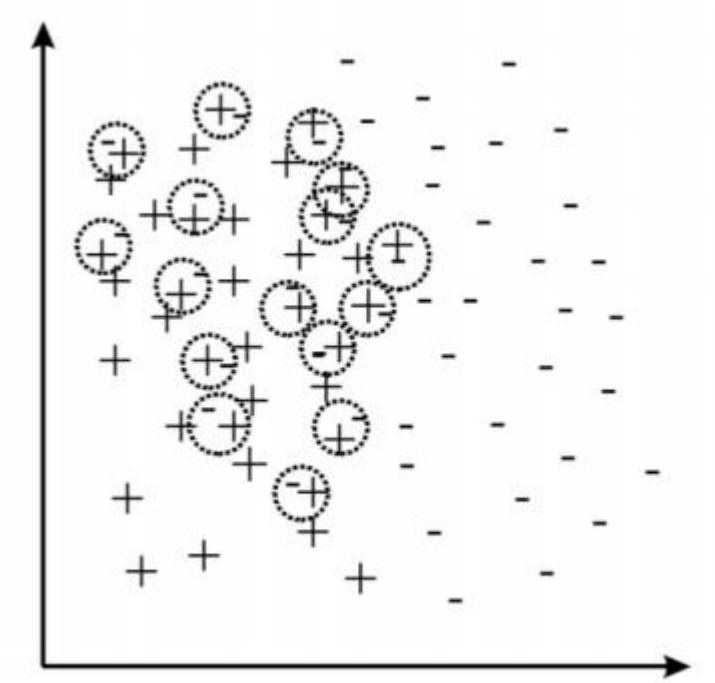
- Tomek links
  - Sean $e_i$ y $e_j$ dos ejemplos de clases diferentes
  - Sea $d(e_i, e_j)$ la distancia entre los dos ejemplos
  - El par $(e_i, e_j)$ es llamado Tomek link si no existe ningún ejemplo $e_k$ que esté "entre" ellos. Por tanto, un par $(e_i, e_j)$ NO es un Tomek link si

$$d(e_i, e_k) < d(e_i, e_j) \text{ o } d(e_j, e_k) < d(e_i, e_j)$$

  - Si $(e_i, e_j)$ es unTomek link tenemos dos posibles situaciones
    - Los ejemplos forman parte de la frontera
    - Uno de ellos es ruido
  - Objetivo del método
    - Eliminar el ejemplo del Tomek link de la clase negativa (under-sampling)
    - Eliminar ambos ejemplos del Tomek link (limpieza de datos)

# Técnicas de under-sampling

- Ejemplo de Tomek links



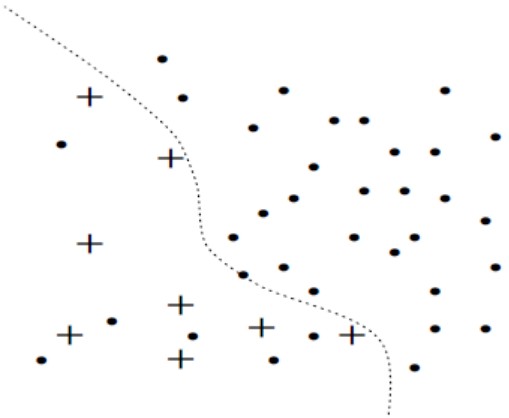
# Técnicas de under-sampling

- One-Sided Selection (OSS)
  - Aplica secuencialmente CNN y Tomek links
    - CNN elimina ejemplos de la clase negativa alejados de la frontera de decisión
    - Tomek links elimina ejemplos de la clase negativa considerados como ruido o ejemplos en la frontera

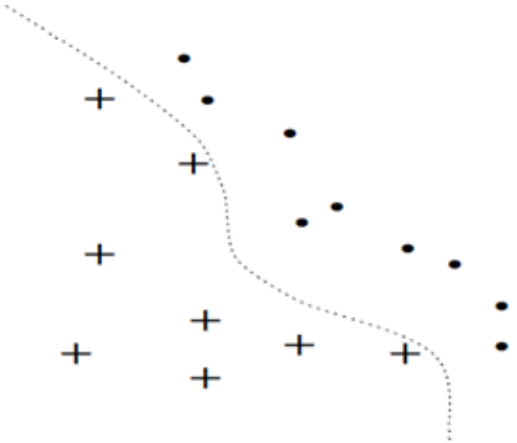
# Técnicas de under-sampling

- Ejemplo de One-sided selection (OSS)

Dataset original



Dataset muestreado (OSS)



# Técnicas de limpieza de datos

- Edited nearest neighbour rule (ENN)
  - Método de Wilson
- Elimina ejemplos de ambas clases
- Funcionamiento
  - Para cada ejemplo $e_i$ se aplica 3NN
  - Si el ejemplo $e_i$ se falla, $e_i$ es eliminado

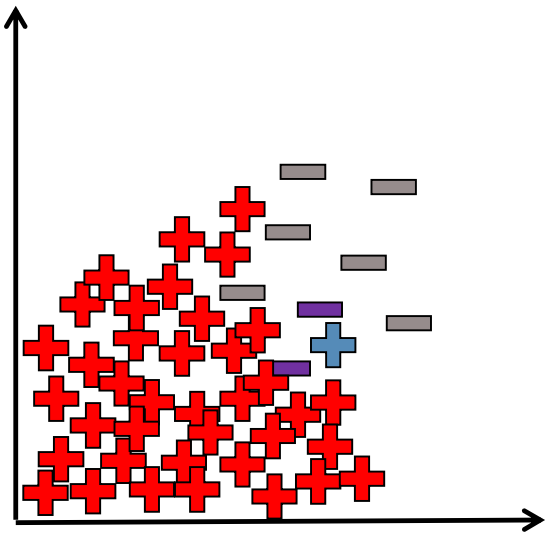
# Técnicas de under-sampling

## Neighbourhood cleaning rule (NCL)

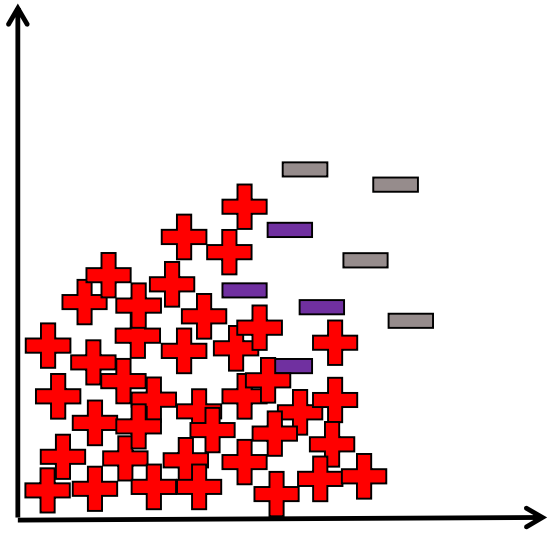
- Modificación de ENN para incrementar la eliminación de ejemplos
  - Trata de eliminar ejemplos ruido más que balancear el conjunto de ejemplos
- Algoritmo
  - Se dividen la BD inicial en dos: P y N que contienen todos los ejemplos de la clase positiva y negativa, respectivamente
  - Se obtiene el subconjunto A1: aplicar ENN sobre los ejemplos en N (utilizando BD) para encontrar los ejemplos ruido (los fallados), que son incluidos en A1
  - Se obtiene el subconjunto A2: limpiar la vecindad de los ejemplos en P
    - Para cada ejemplo $e_i$ Є P se obtienen sus 3 vecinos más cercanos (3NN) de toda la BD inicial (P y N)
    - Si se falla $e_i$, se insertan en A2 los vecinos de $e_i$ que estén en N
  - El nuevo conjunto es CE' = CE – (A1 U A2) → Los ejemplos de P se mantienen

# Técnicas de under-sampling

- Ejemplo NCL



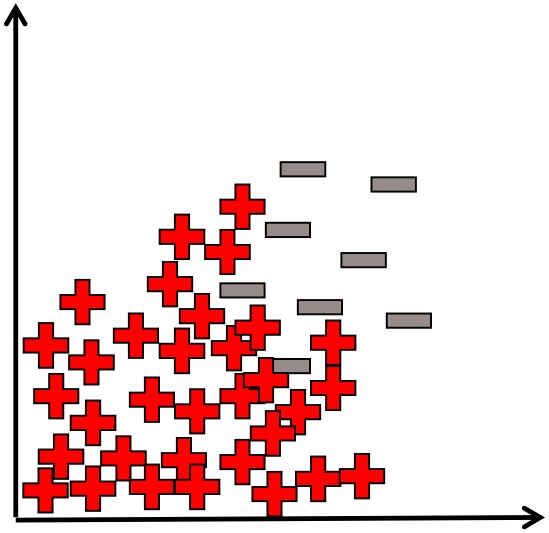
$E_i \in$ 

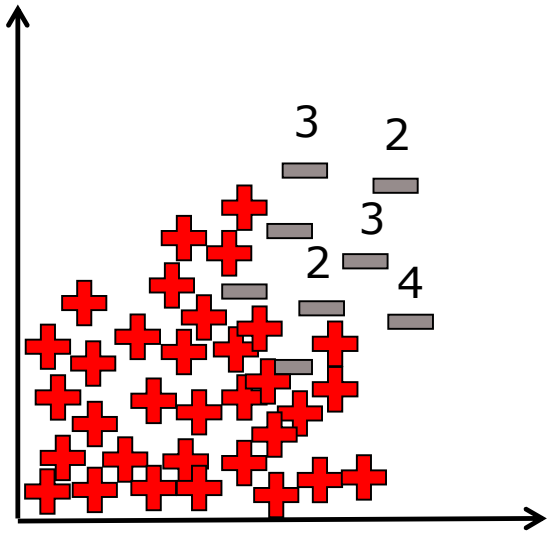


$E_i \in$ 

# Técnicas de over-sampling

- Random over-sampling (ROS)
  - Método no heurístico
  - Crea copias de ejemplos de la clase positiva que son elegidos aleatoriamente hasta balancear el conjunto de ejemplos
  - Problema: posible incremento del sobre-aprendizaje puesto que crea copias exactas de ejemplos de la clase positiva





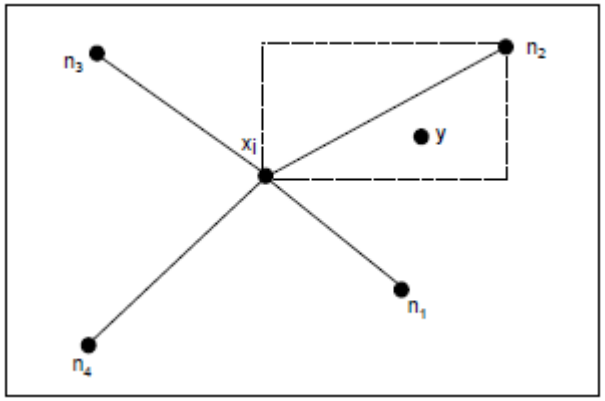
# Técnicas de over-sampling

- Synthetic minority over-sampling technique (SMOTE)
  - Crea ejemplos de la clase positiva interpolando ejemplos de la clase positiva que estén cerca
  - Soluciona el problema del sobre-aprendizaje
  - Algoritmo:
    - Iniciar el conjunto de ejemplos sintéticos a vacío
    - Para cada ejemplo $e_i$ de la clase positiva
      - Calcular los 5 vecinos más cercanos de $e_i$ ($n_1$, $n_2$, $n_3$, $n_4$ y $n_5$)
      - Número de ejemplos a crear → $\text{numEjCrear} = floor(IR) - 1$
      - Desde 1 hasta $\text{numEjCrear}$ hacer
        - Seleccionar aleatoriamente uno de la 5 vecinos más cercanos ($n_a$)
        - Para cada atributo j ∈ N
          - Calcular la distancia entre $e_{ij}$ y $n_{aj}$ → $dist_j = n_{aj} - e_{ij}$
          - Crear ejemplo sintético $eS_j = e_{ij} + rand([0,1]) * dist_j$
        - Añadir el nuevo ejemplo $eS$ al conjunto de ejemplos sintéticos

# Técnicas de over-sampling

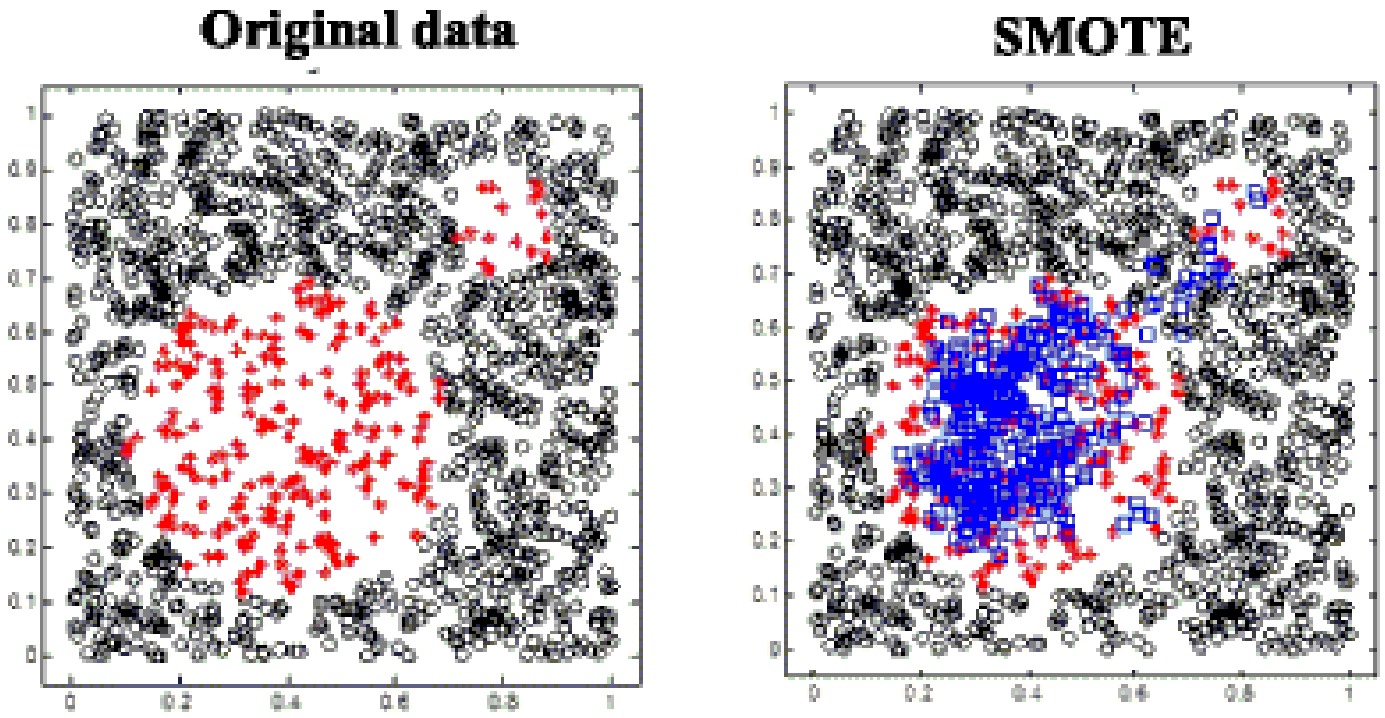
- Ejemplo de creación de un ejemplo sintético con SMOTE

Consider a sample (6,4) and let (4,3) be its nearest neighbor.
(6,4) is the sample for which k-nearest neighbors are being identified.
(4,3) is one of its k-nearest neighbors.
Let:
f1_1 = 6  f2_1 = 4  f2_1 - f1_1 = -2
f1_2 = 4  f2_2 = 3  f2_2 - f1_2 = -1
The new samples will be generated as
(f1',f2') = (6,4) + rand(0-1) * (-2,-1)
rand(0-1) generates a random number between 0 and 1.



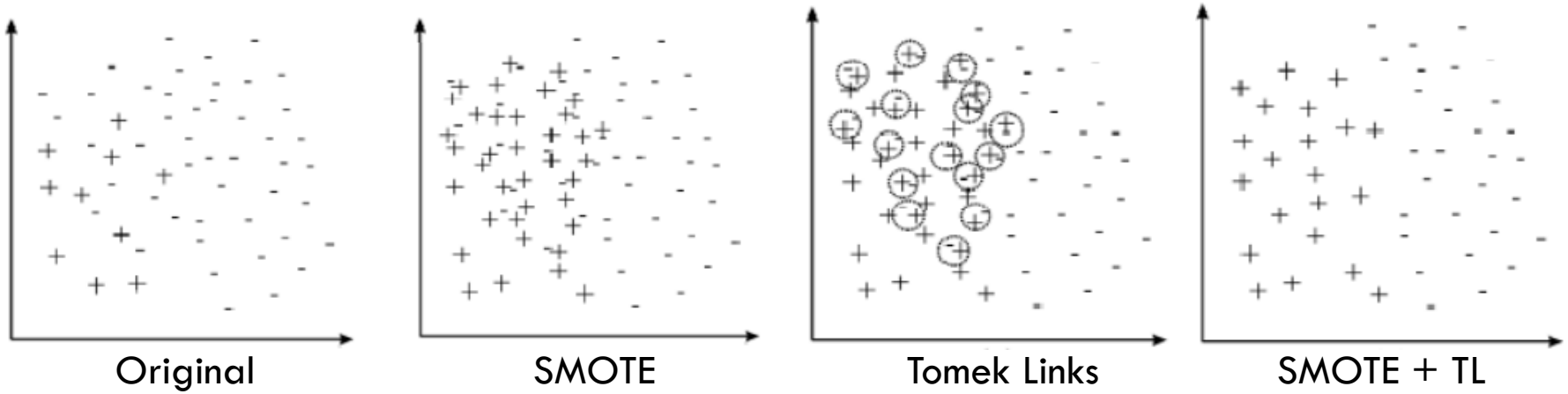
# Técnicas de over-sampling

- Ejemplo del efecto de SMOTE



# Técnicas híbridas

- SMOTE + Tomek links
  - Se aplican secuencialmente SMOTE y Tomek links
    - Aplicar SMOTE para mejorar los *clusters* de la clase positiva
    - Aplicar Tomek links como técnica de limpieza de datos
  - Idea: mejorar los *clusters* de clases debido a
    - Ejemplos de la clase negativa invaden los *clusters* de la clase positiva
    - Al crear ejemplos de la clase positiva se pueden adentrar en el área de la clase negativa



# Técnicas híbridas

- SMOTE + ENN
  - Motivación: mejorar el proceso de limpieza de datos
  - Se aplican secuencialmente SMOTE y ENN
    - ENN suele eliminar más ejemplos que Tomek links

# Soluciones algorítmicas

- CART
  - Cálculo de las probabilidades en base a la proporción de ejemplos de cada clase en el nodo con respecto a la proporción del problema inicial
  - Para cada clase, $C_j \in [C_1, \ldots, C_M]$
    - $prop_{C_j} = \frac{numEjemplosNodo_{C_j}}{numEjemplosRaiz_{C_j}}$
  - Probabilidad de cada clase
    - $p_{C_j} = \frac{prop_{C_j}}{\sum_{i=1}^{M} prop_{C_i}}$
  - El resto del proceso de construcción del árbol se mantiene

# Soluciones algorítmicas

- C4.5 sensible al coste
  - Probabilidad para cada clase, $C_j \in [C_1, \dots, C_M]$
    - $p_{C_j} = \frac{w_{C_j} * N_{C_j}}{\sum_{i=1}^{M} w_{C_i} * N_{C_i}}$
    - $w_{C_j} = C(C_j) * \frac{N}{\sum_{i=1}^{M} C(C_i) * N_{C_i}}$
      - $C(C_j)$ es el coste de fallar un ejemplo de la clase $C_j$
      - $N_{C_i}$ es el número de ejemplo de la clase $C_j$ en el nodo
      - $N = \sum_{i=1}^{M} w_{C_i} * N_{C_i}$
  - El resto del proceso de construcción del árbol se mantiene