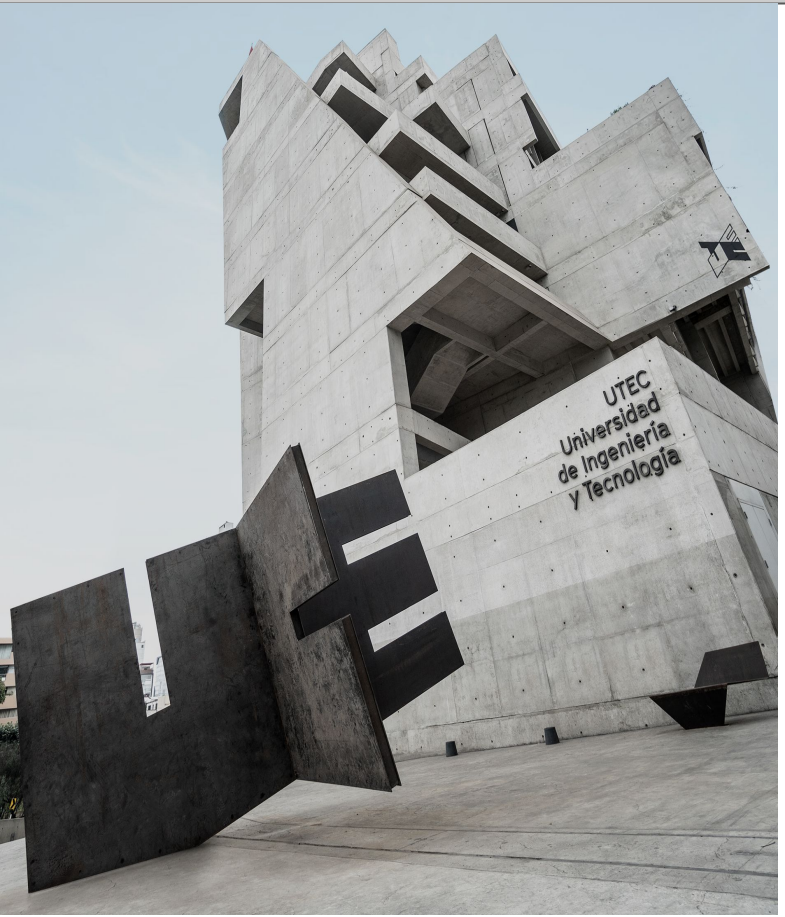
Ciencia de la Computación

KNN

Cristian López Del Alamo
clopezd@utec.edu.pe
IPRODAM3D - Research group

2022

Programa



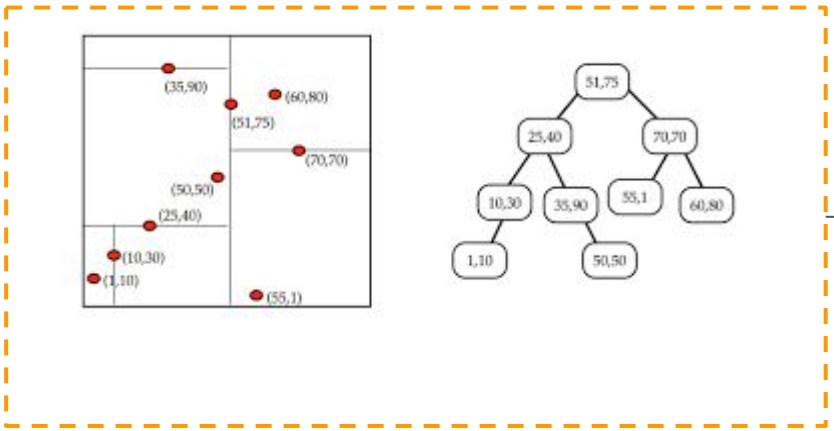
1. KNN
2. Cross Validation
3. Bootstrap

1 KNN

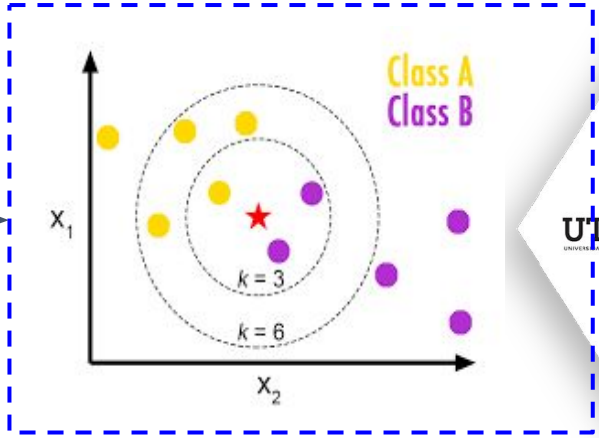
Conceptos clave

- KNN es un algoritmo de aprendizaje supervisado.
- Almacena los ejemplos de entrenamientos etiquetados durante la fase de entrenamiento.
- Lazy Learning Algorithm.

Etapa de Entrenamiento

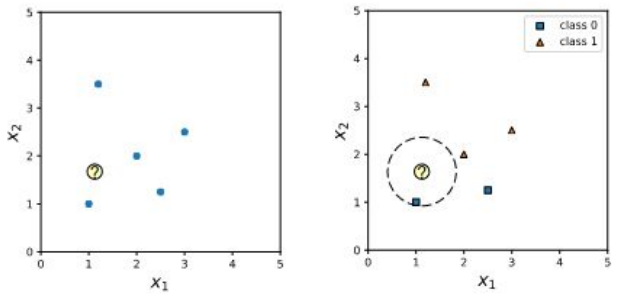


Fuente: Click



Fuente: Click

Etapa de Predicción



Fuente: Click

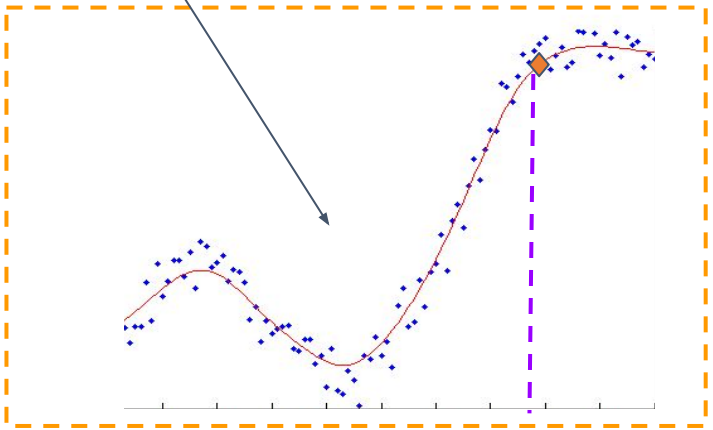
- Búsqueda de los k vecinos más cercanos.
- Obtenemos las etiquetas de los vecinos.
- Realizamos un ***voting.***
- Puede ser utilizado para clasificación o regresión.

Key idea: En lugar de aproximar una función f(x) = y de manera global, ***knn***, se aproxima a la función objetivo de manera local.

Regresión KNN

- KNN no tiene un paso de entrenamiento explícito.
- Pospone los cálculos hasta el momento de la predicción.
- Lazy algorithms

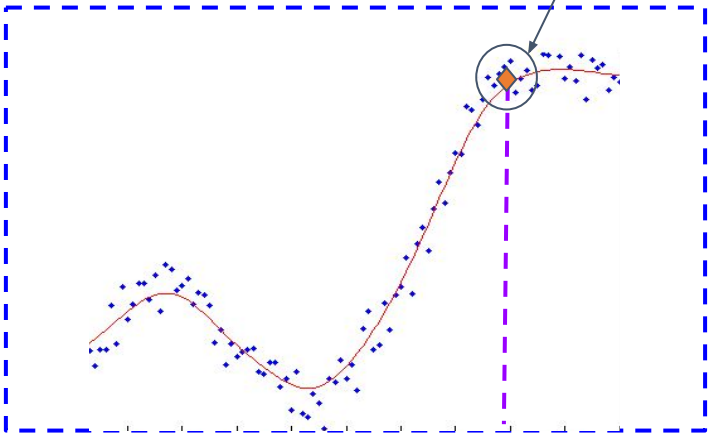
Aproximación Loca f(x) = y



Fuente: Click

Aproximación Loca

$$h(x_q) = \frac{1}{k}\sum_{i=0}^{n} f(x_i)$$

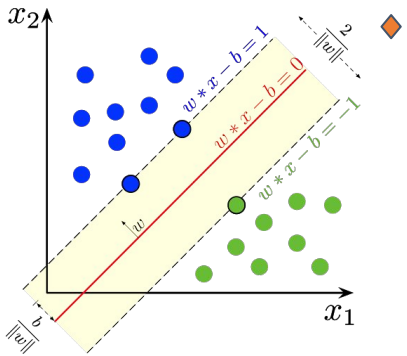


Fuente: Click

Clasificación KNN

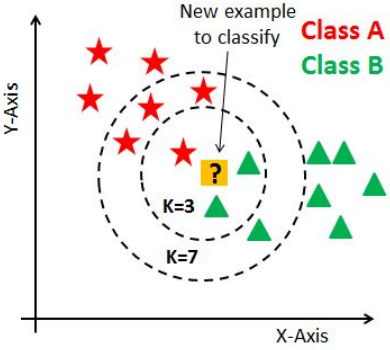
- Compara un elemento de consulta con un subconjunto de elementos de entrenamiento
- KNN está categorizado como basado en instancia o basado en memoria
- Se le considera como un modelo no paramétrico.

Aproximación Loca f(x) = y



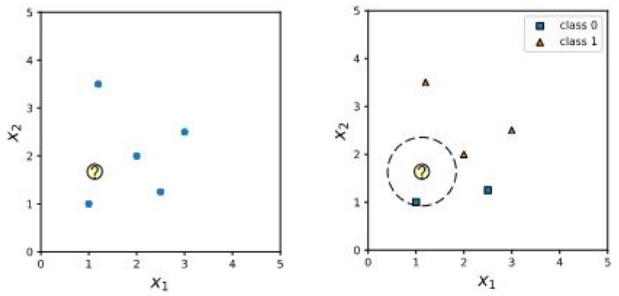
Fuente: Click

Aproximación Local



Fuente: Click

Etapa de Predicción

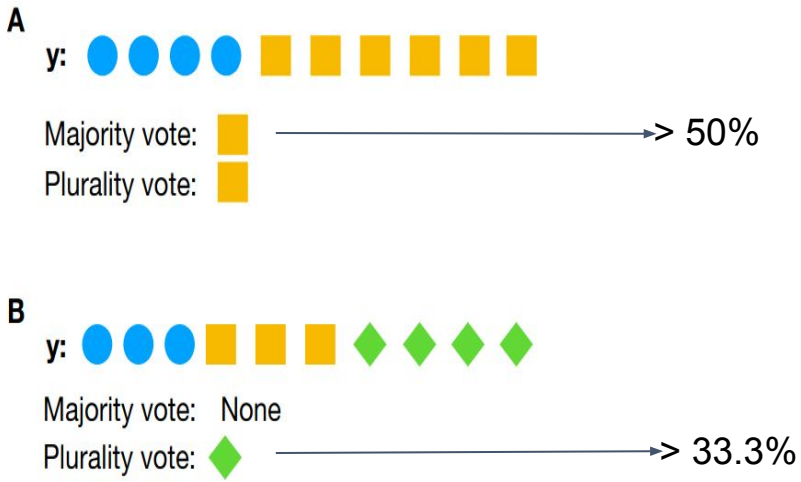


Fuente: Click

- Búsqueda de los k vecinos más cercanos.
- Obtenemos las etiquetas de los vecinos.
- Realizamos un ***voting.***
- Puede ser utilizado para clasificación o regresión.

Key idea: En lugar de aproximar una función f(x) = y de manera global, ***knn***, se aproxima a la función objetivo de manera local.

Voting



- ***Pluralty voting*** entre knn para clasificación
- **Promedio** de los knn en regresión

Formalmente

Se tiene una función $f(x) = y$ que asigna una etiqueta $y \in \{1, 2, ..., t\}$ a un elemento de entrenamiento $f : R^d \rightarrow \{1, ..., t\}$

Identificamos los k vecinos más cercanos $D_k \subset D_n$ para cada punto x_q

$$h(x_q) = \underset{y \in \{1,2,...,t\}}{argmax} \sum_{i=1}^{k} \delta(y, f(x_i))$$

Kronecker Delta function

$$\delta(y, f(x_i)) = \{1 \, si \, a = b; 0 \, si \, a \neq b\}$$

De manera más simple, se trata de encontrar la moda de las etiquetas de los k vecinos más cercanos

$$h(x_q) = mode(f(x_1), f(x_2), ...f(x_k))$$

Algoritmo KNN

Entrenamiento

```
DataStructure D;
for i to len(DataSet)
   D.Insert(<x_i, y_i>)
```

Predicción

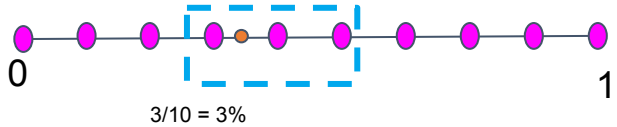
```
h(x_q):
        closest_point = None
        closest_distance = INF

        for i in len(DataSet)
                current_distance = d(x_i, x_q)
                if (current_distance < closest_distance):
                    closest_distance = current_distance
                    closest_point = x_i
return closest_point
```

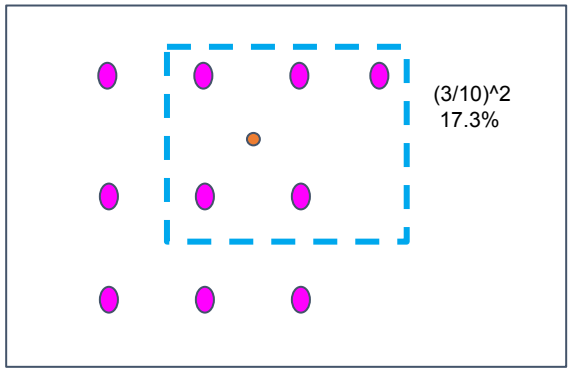
h(x_q) = f(x_q)

Maldición de la dimensionalidad

KNN es susceptible a altas dimensiones (curse of dimensionality)
Curse of Dimensionality se refiere a un número fijo de ejemplos de entrenamiento pero un número creciente de la dimensión o en rango de los valores de las características en cada dimensión en un espacio característico dimensionalmente alto.

10 ejemplos de entrenamiento distribuidos uniformemente.
Se espera una distancia de 0.1
Si k es 3





Lectura: Click Aquí

Validación y entrenamiento en problemas de clasificación

Todas las técnicas de reconocimiento de patrones tienen uno o más hiperparámetros y esto genera dos problemas:

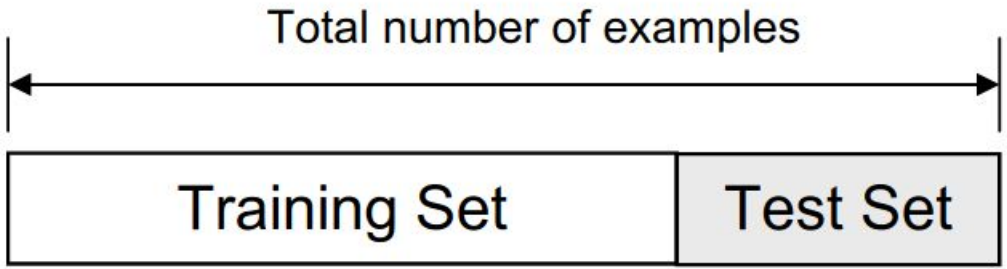
- Selección del Modelo
  - ¿Cómo seleccionamos los hiperparámetros óptimos para un problema de clasificación?
- Validación del Modelo
  - Una vez que tenemos el modelo, como estimamos el ratio de error correcto.
  - El ratio de error correcto es el ratio de error que se tendrían en la población completa de los datos. (Algo que no tenemos)

Validación entrenamiento en Clasificación

Posible solución: Utilizar el conjunto de datos completo para estimar el ratio de error.

¿Qué Problemas le encuentran a esta solución?

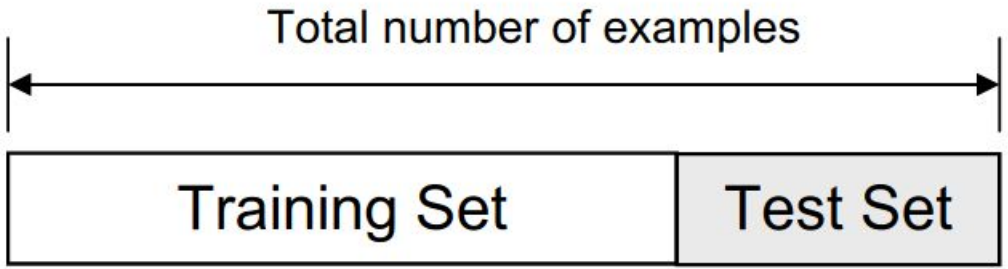
El método holdout



Problemas

- ¿Qué ocurre si tenemos una base de datos poco densa, o muy pequeña?
- Dado que sólo es un experimento de entrenamiento y prueba, la estimación de la tasa de error será engañosa en caso de que tengamos una división defectuosa.

El método holdout



Problemas

- ¿Qué ocurre si tenemos una base de datos poco densa, o muy pequeña?
- Dado que sólo es un experimento de entrenamiento y prueba, la estimación de la tasa de error será engañosa en caso de que tengamos una división defectuosa.

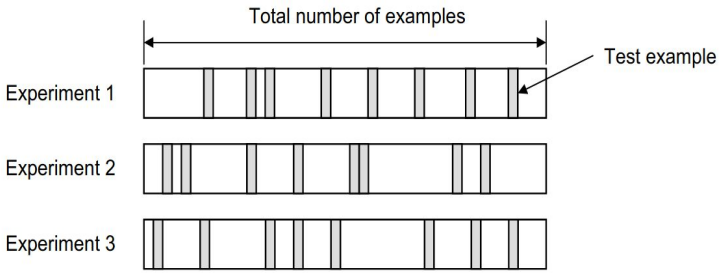
Resampling method

- Cross Validation
  - Random Subsampling
  - K-Fold Cross-Validation
  - Leave-one-out-Validation
- Bootstrap

Problemas

- Alto costo computacional

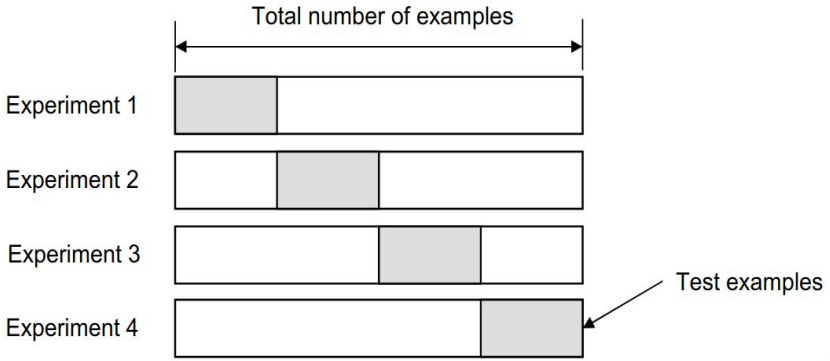
Random Subsampling



La Estimación del error se obtiene:

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

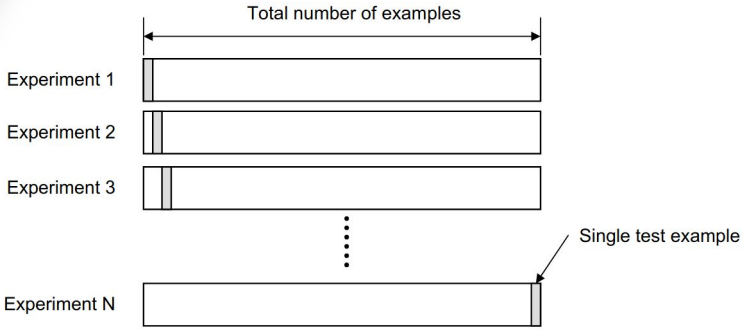
K-Fold Cross-Validation



La Estimación del error se obtiene:

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Leave-one-out-Validation



La Estimación del error se obtiene:

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

- Para un dataset con N ejemplos, se realizan N experimentos.
- Para cada experimento se utiliza N-1 ejemplos para entrenamiento y el restante para testing.

¿Cuentos folds son necesarios?

Un alto número de folds

- El bias del ratio de error estimado será pequeño (El estimador será más preciso)
- La varianza del estimador del error será mayor.
- El costo computacional será mayor pues tendremos más experimentos.

Un bajo número de folds

- El número de experimentos será menor y por lo tanto el costo computacional
- La varianza del estimador será menor
- El bias del estimador será mayor (menos preciso)

En la práctica

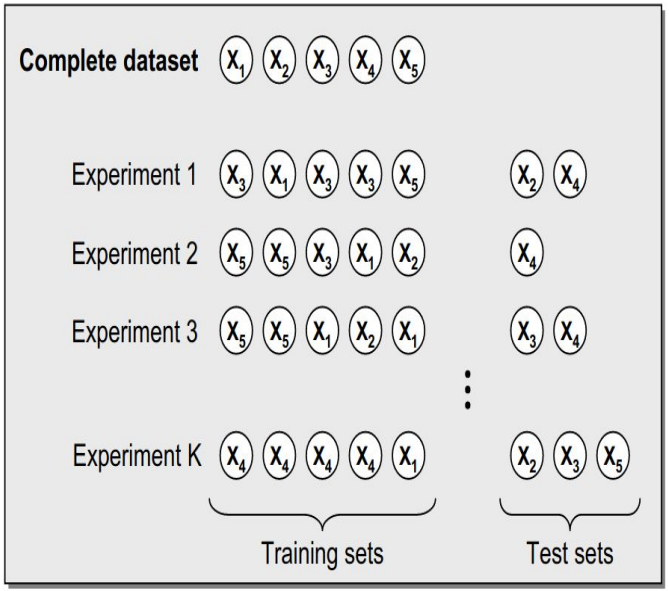
- Para base de datos grandes, incluso 3-fold Cross Validation será suficiente.
- Una opción común en K-fold Cross Validation es K=10
- Para bases de datos pequeñas , debemos utilizar leave-one-out.

Método Bootstrap

- Seleccionar N elementos (con reemplazo) y utilizarlos para el entrenamiento
- Los restantes son utilizados para testing
- Repetir el proceso para k kolds

La Estimación del error se obtiene:

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$



Gracias

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ciencia de la Computación

KNN

Cristian López Del Alamo
clopezd@utec.edu.pe
IPRODAM3D - Research group