Vision Artificial. GIEC.

Sistemas de Vision Artificial. GIC.

Miguel Angel Garcia, Juan Manuel Miguel, Sira Palazuelos.

Departamento de Electrónica. Universidad de Alcalá.

Tema 5: ejercicio 02 - k-Nearest Neighbour (kNN)

La base de datos **`hospital`** se ha dividido en dos grupos: 60% para entrenamiento y 40% para test.

Step 1) Carga datos

```
clear all;
close all;

% base de datos
load hospital;
% se utiuliza solo la presión sanguinea
data = [hospital.BloodPressure];

rng(1)
% se obtienen una distribución aleatoria de los índices del tamaño de los
% datos
idx = randperm(length(data))';

% 60% para entrenamiento y 40% para test
distribution_train = 0.6;
idx_train = idx(1:length(data)*distribution_train);
idx_test = idx(length(data)*distribution_train+1:length(data));

% valores y etiquetas para entrenamiento
data_train = data(idx_train,:);
label_train = hospital.Smoker(idx_train); % 'Smoker' = 1, 'Non-Smoker' = 0

tabulate(label_train)
```

```
  Value    Count   Percent
      0       40     66.67%
      1       20     33.33%
```

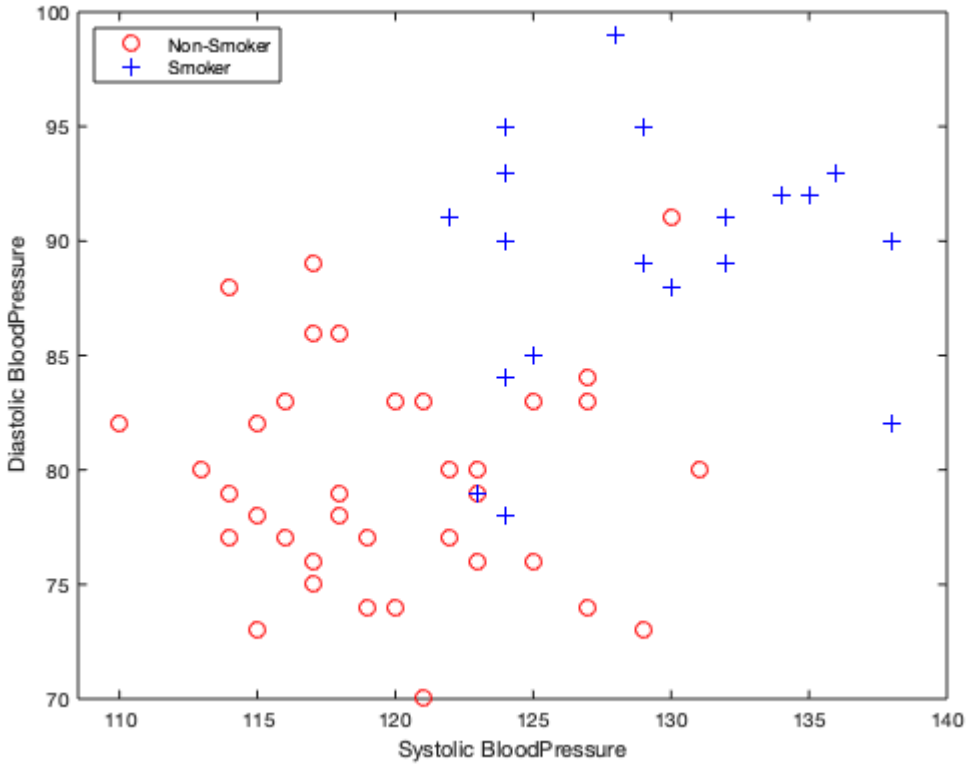
```
% valores y etiquetas para test
data_test = data(idx_test,:);
label_test = hospital.Smoker(idx_test); % 'Smoker' = 1, 'Non-Smoker' = 0

tabulate(label_test)
```

```
  Value    Count   Percent
      0       26     65.00%
```

```
     1        14      35.00%
```

Step 2) Visualiza los datos usando un gráfico de dispersión

```
gscatter(data_train(:,1),data_train(:,2), label_train,'rb','o+',8,'on')
xlabel('Systolic BloodPressure');
ylabel('Diastolic BloodPressure');
legend('Non-Smoker','Smoker', 'Location','northwest')
```



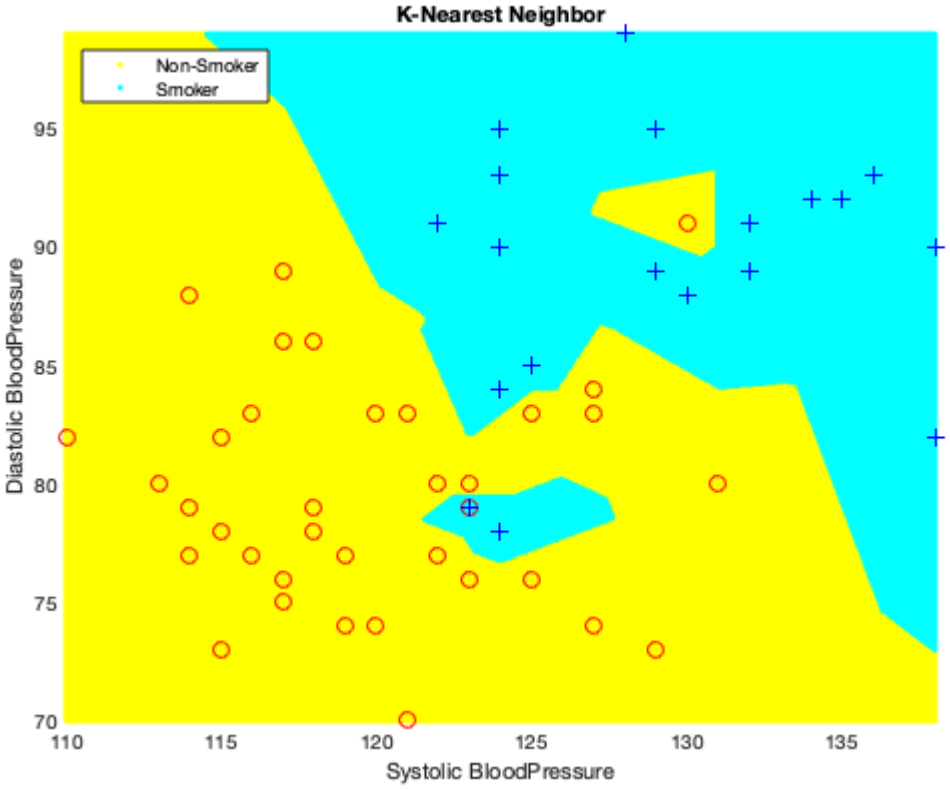
Step 3) Entrenamiento del modelo

```
k = 1;
ClassifierModel = fitcknn(data_train,label_train,'NumNeighbors',k);
```

Step 4) Superficie de decisión

```
% grid
x1range = min(data_train(:,1)):.05:max(data_train(:,1));
x2range = min(data_train(:,2)):.05:max(data_train(:,2));
[xx1, xx2] = meshgrid(x1range,x2range);
XGrid = [xx1(:) xx2(:)];

% predicción
predictions_mesh = predict(ClassifierModel,XGrid);
% Superficie de decisión
gscatter(xx1(:), xx2(:), predictions_mesh,'yc');
```

```
title('K-Nearest Neighbor')
hold on;

% se sobrescriben los datos de entrenamiento
gscatter(data_train(:,1),data_train(:,2), label_train,'rb','o+',8,'on')
xlabel('Systolic BloodPressure');
ylabel('Diastolic BloodPressure');
legend on, axis tight
legend('Non-Smoker','Smoker', 'Location','northwest')
hold off
```



Se pide:

1. Obtenga la predicción del conjunto de datos de test usando la siguiente función: **`predict();`** y dibuje los resultados (predicción y ground truth).
2. Calcule la matriz de confusión del **`ClassifierModel`** utilizando la función **`confusionmat`**.
3. Calcule la especificidad y sensibilidad como: `especificidad = TN/(TN+FP)` y `sensibilidad = TP/(TP+FN)`, indicando qué clase considera como positiva. Nota: Estos parámetros sólo se pueden calcular para clasificadores binarios.
4. Cambie el número **`k`** del vecino más cercano (**`'NumNeighbors'`**) en el paso 3 utilizando los siguientes valores **`K = 1, 5, 10`** y compruebe cómo afecta a los modelos anteriores de ClassifierModel.

5. Entrene de nuevo utilizando el siguiente clasificador: **`fitcnb();`** % un modelo Bayes `naive` y compare los resultados con el kNN.