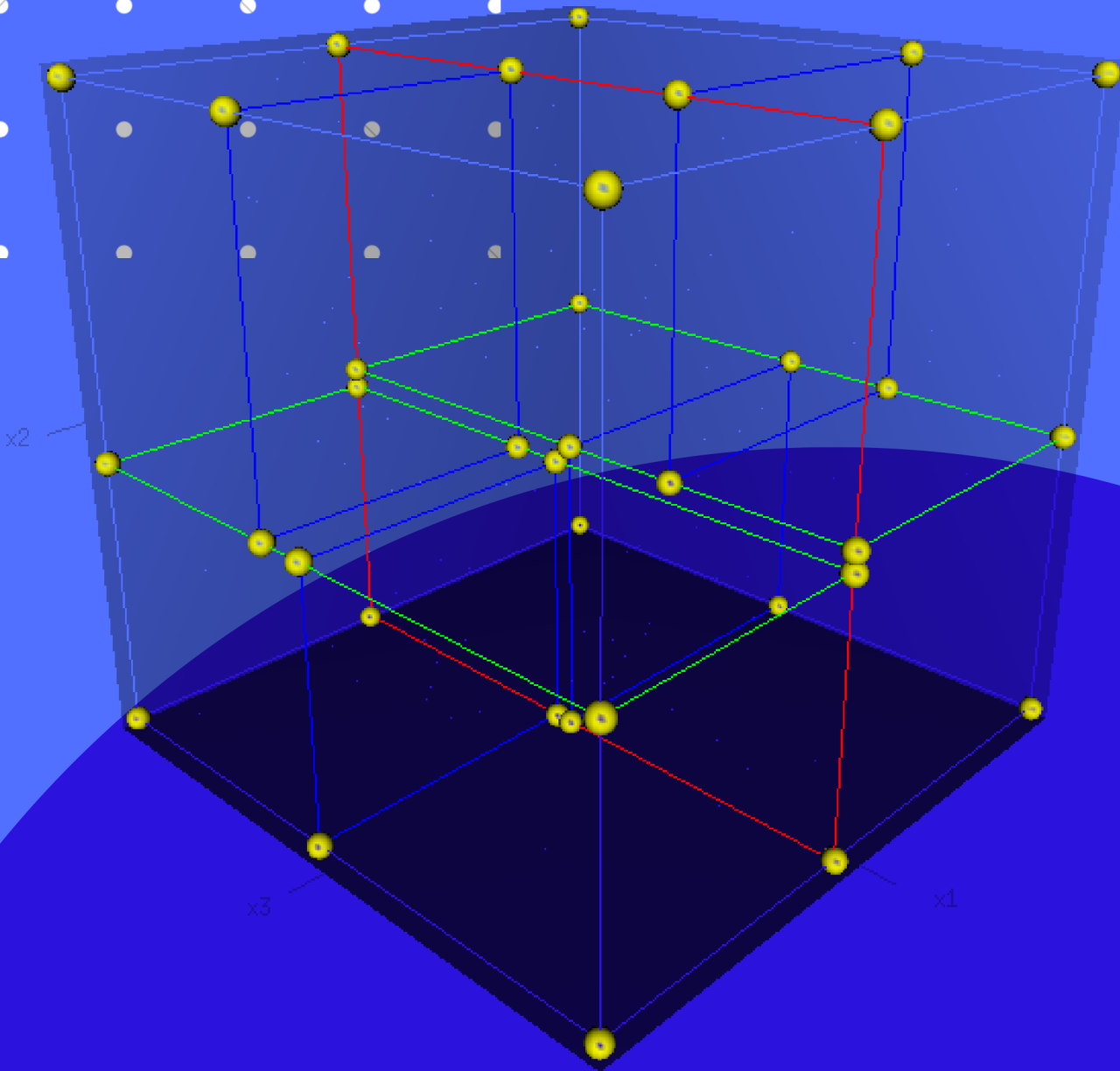


UNIVERSIDAD NACIONAL DE SAN AGUSTIN



IMPLEMENTACION KDTREE Y KNN

ING. JESUS ALPACA RENDON



x x x x

INDICE

KD Tree	03
KNN	04
Dataset	05
Implementacion	06
Resultados	08
Conclusiones	09

KDTREE

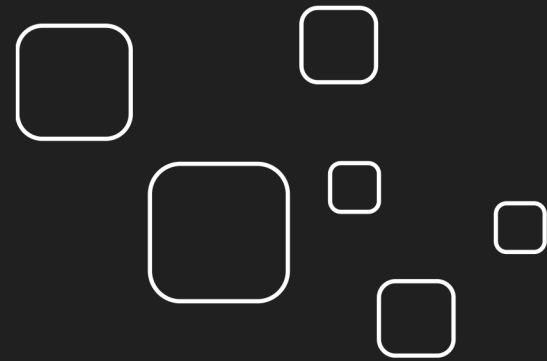
La estructura KD Tree es una estructura multidimensional

Arbol binario que permite el ordenamiento de los datos segun las dimensiones que contenga de manera que cada nodo divide en secciones y va creando nodos izquierdos y derechos, siendo los de la izquierda menores y derecha mayores, iterando la dimensiones que toque en cada rotacion.

```
function kdtree (list of points pointList, int depth)
{
    // Select axis based on depth so that axis cycles through all valid values
    var int axis := depth mod k;

    // Sort point list and choose median as pivot element
    select median by axis from pointList;

    // Create node and construct subtree
    node.location := median;
    node.leftChild := kdtree(points in pointList before median, depth+1);
    node.rightChild := kdtree(points in pointList after median, depth+1);
    return node;
}
```

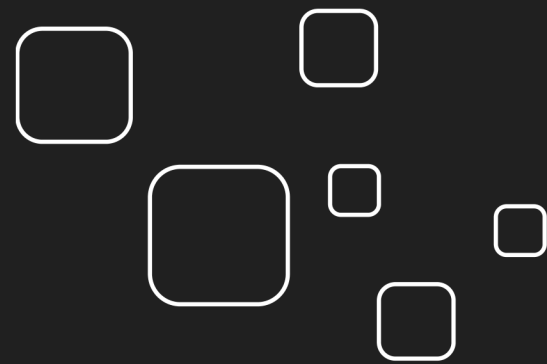


KNN

Algoritmo KNN

Algoritmo que se encarga de encontrar los N vecinos cercanos de un punto segun su distancia, para aplicar una clasificacion por mayoria. Es decir, de 5 vecinos, 3 tienen clasificacion True y 2 clasificacion False, dando como clasificacion al nodo a evaluar que es de clasificacion True.

$$\hat{f}(x) \leftarrow \arg \max_{v \in V} \sum_{i=1}^k [v = f(x_i)]$$





HEART DISEASE CLASSIFICATION DATASET

El presente Dataset contiene registros de enfermedades cardiovasculares las cuales se muestran datos sobre sus diagnósticos, y se comparte con la finalidad de analizar cuales de ellos desembocaron en un ataque cardiovascular

Fuente:

<https://www.kaggle.com/datasets/bharath011/heart-disease-classification-dataset>

Python

Se aplico el lenguaje Python para la realización del presente trabajo, esto por la facilidad que ofrecen ciertas librerías en la medición de métricas y creación de mapas y gráficos visuales para entender la implementación

Estructura KD Tree y algoritmo KNN

Se aplico el algoritmo KNN para obtener los vecinos cercanos con la estructura de KD Tree

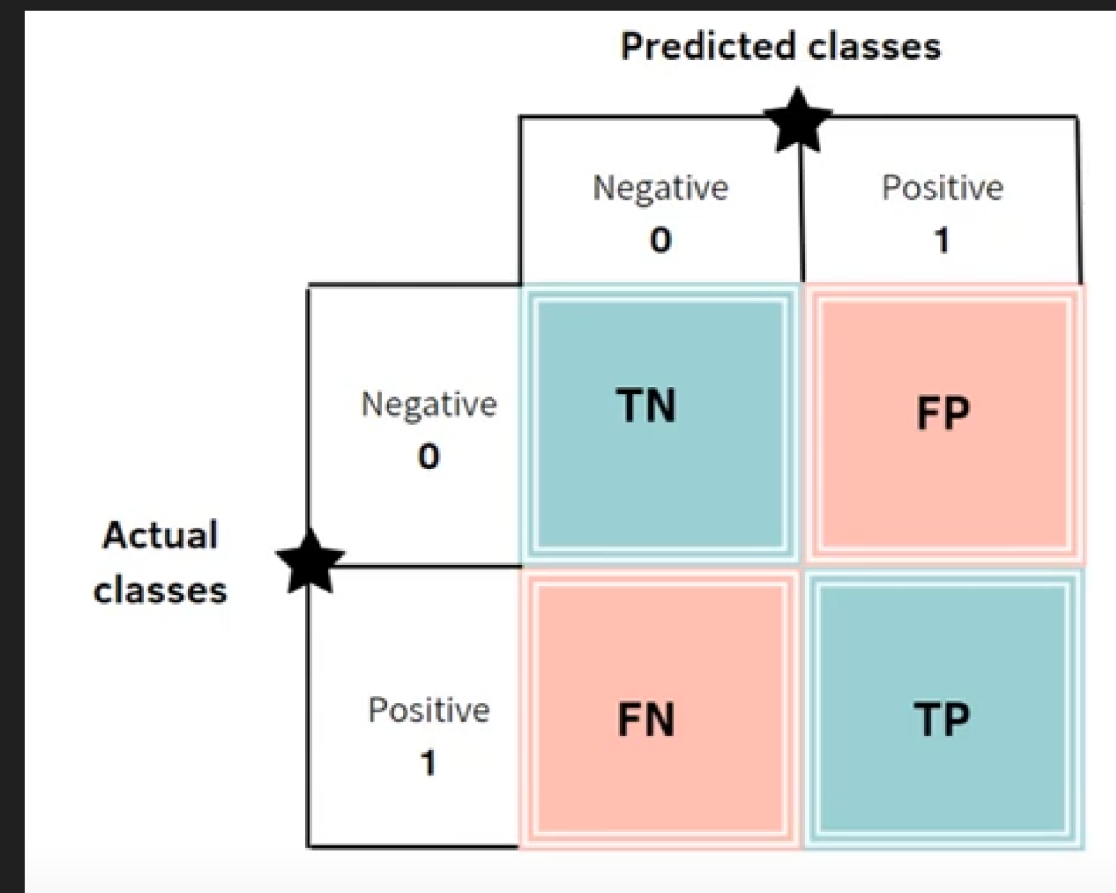
Enlace de Github

https://github.com/Alpha004/AyEDJAAR2023/tree/main/Ejercicio_Final

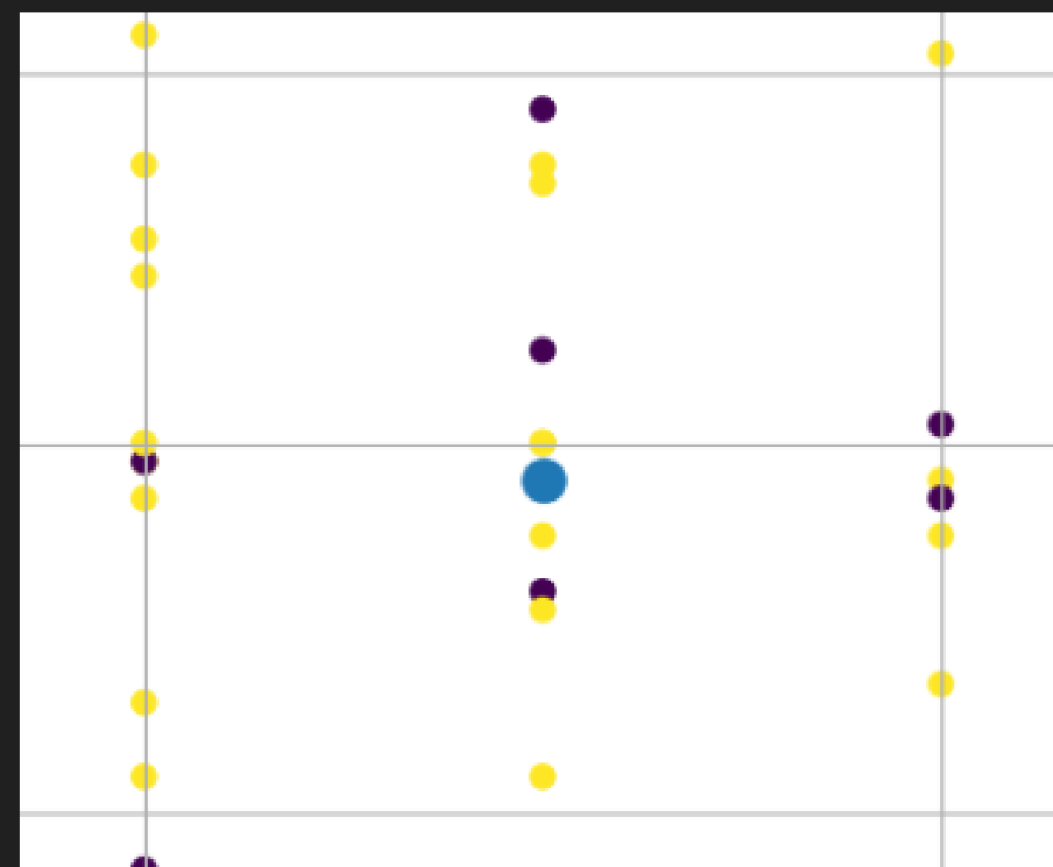
RESULTADOS

Metricas

	precision	recall	f1-score	support
0	0.40	0.65	0.50	136
1	0.56	0.31	0.40	194
accuracy			0.45	330
macro avg	0.48	0.48	0.45	330
weighted avg	0.50	0.45	0.44	330



[[89 47]
[133 61]]



CLASIFICACION DATA:

```
[0 1 1 0 0 0 1 1 0 0 0 0
0 1 0 1 0 0 0 0 0 0 0 0
0 1 0 1 0 1 1 1 1 1 0 0
0 1 1 1 0 1 1 1 0 1 1 1
1 1 1 1 0 0 0 1 0 1 1 1
1 1 1 1 0 0 1 1 0 0 0 1
1 1 1 1 0 0 1 0 1 1 1 0
0 1 1 0 1 1 0 1 1 0 1 1
0 0 1 1 0 0 1 1 0 1 1 1]
```

PREDICTIONS:

```
[0, 0, 0, 1, 1, 1, 0, 1,
0, 0, 1, 1, 1, 1, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 1, 1,
1, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0]
```


CONCLUSIONES

• • • • • • • • • •

- La implementación de KDTree para la resolución del problema es una investigación que denota interés, ya que es una de las estructuras multidimensionales más completas.
- Se debe considerar que la precisión obtenida no es muy aceptable ya que con mucho esfuerzo logra pasar el 50% de precisión, KNN no sería un algoritmo idóneo para clasificar datos.
- Existen otros algoritmos mas fuertes que pueden aplicar una precisión mejor mas aceptable, lo cual se recomienda realizar un proceso similar con otros algoritmos.

