



Algoritmo de k vecinos más cercanos (KNN)



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Dr. Francisco Arduh
2023

Algunos ejemplos de su uso

◇ Finanzas:

- ◆ Compañías con problemas financieros.
- ◆ Puntaje crediticio.
- ◆ Fraudes.

◇ Agricultura

- ◆ Predicción de clima.
- ◆ Cantidad de agua en el suelo.

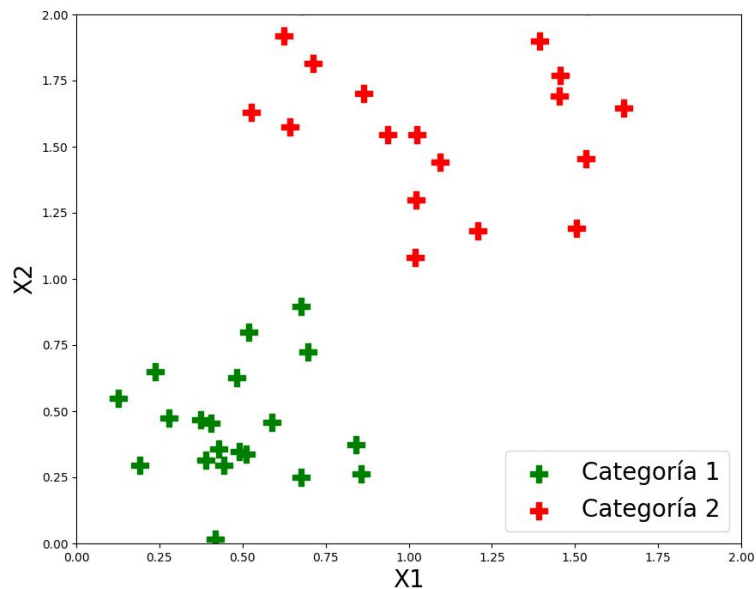
◇ Medicina

- ◆ Riesgo de contraer enfermedades.

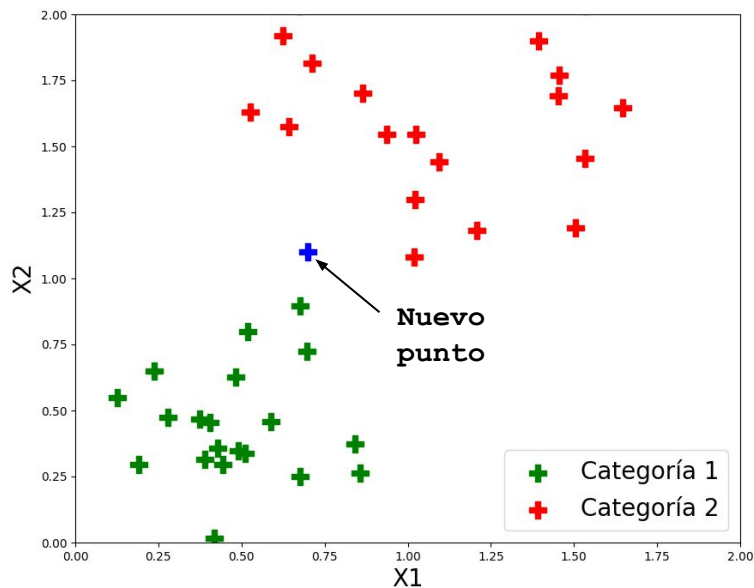


Introducción a KNN

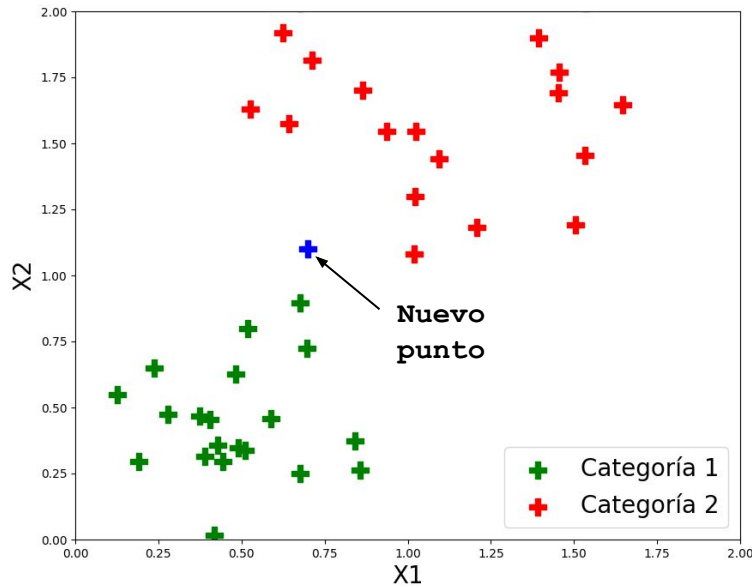
Qué va hacer KNN por nosotros



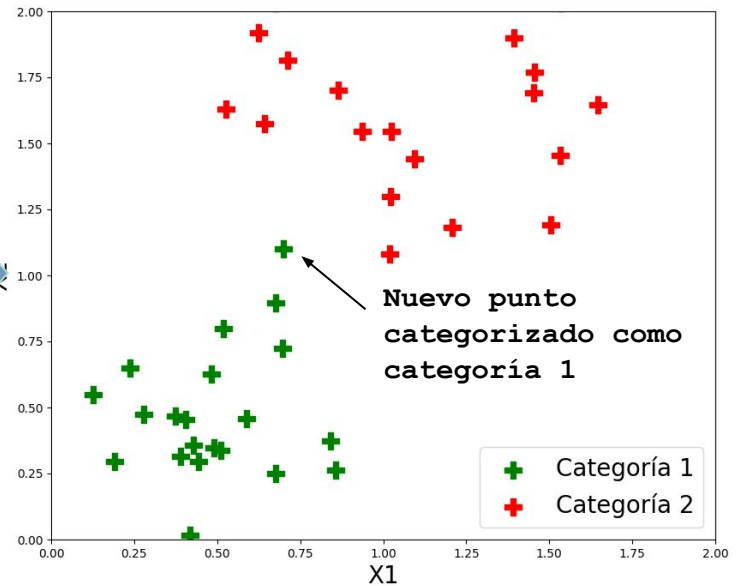
Qué va hacer KNN por nosotros



Qué va hacer KNN por nosotros



KNN





¿Cómo funciona?

Paso 1: Elijo el número “k” de vecinos

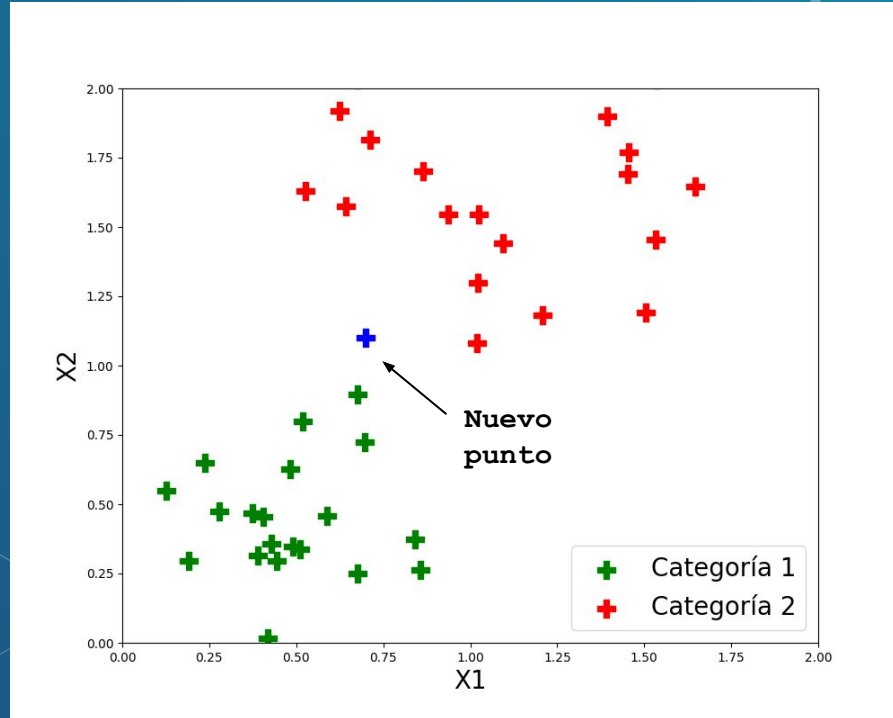
Paso 2: Tomo los k vecinos más cercanos al nuevo punto, de acuerdo a una distancia (euclídea).

Paso 3: Entre los k vecinos más cercanos cuento cuantos pertenecen a cada categoría.

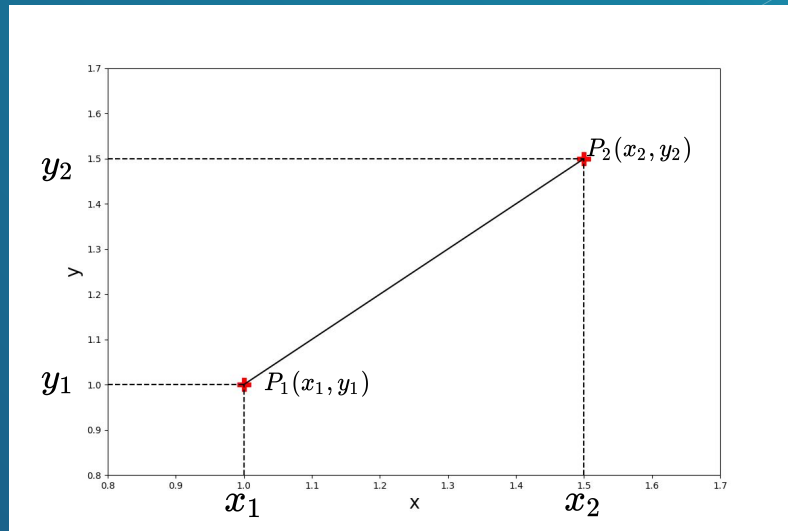
Paso 4: Asigno al nuevo punto la categoría a la categoría más abundante

¿Cómo funciona?

Paso 1: Elijo el número “k” de vecinos, en particular k=5



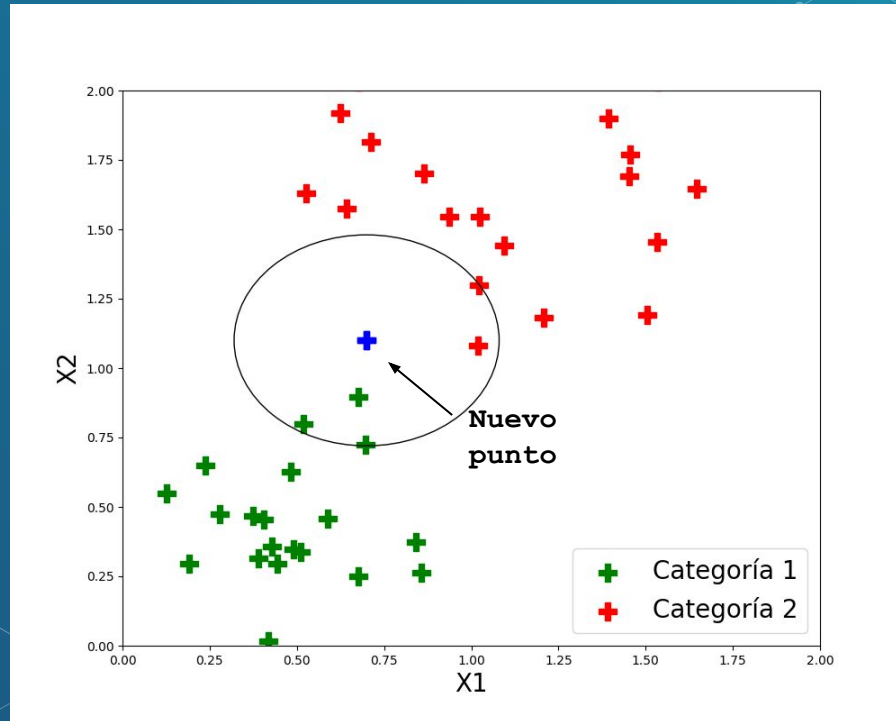
Distancia euclídea



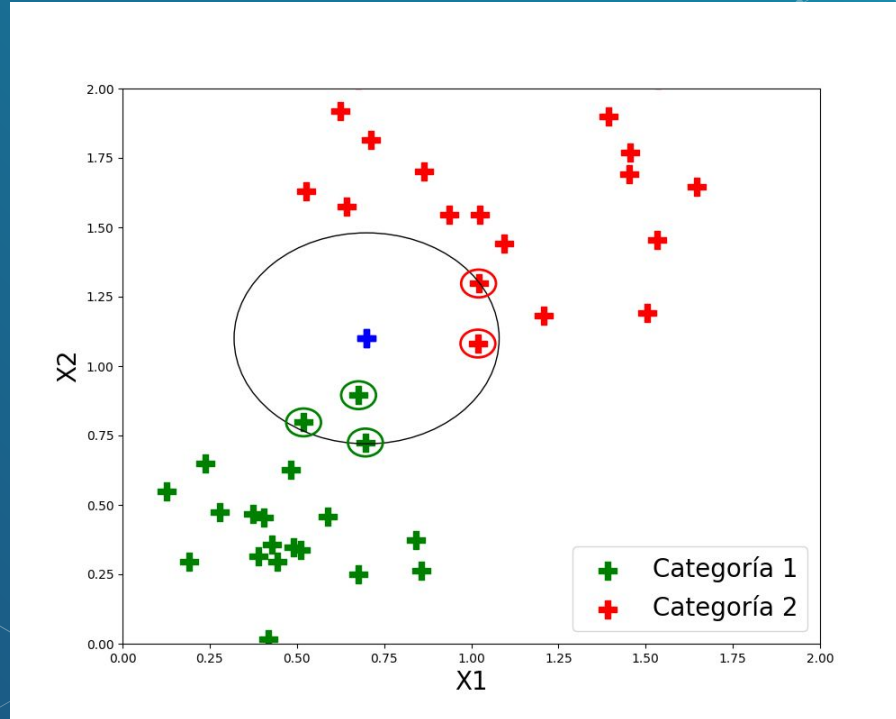
Se define la distancia euclídea d entre P_1 y P_2 como:

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

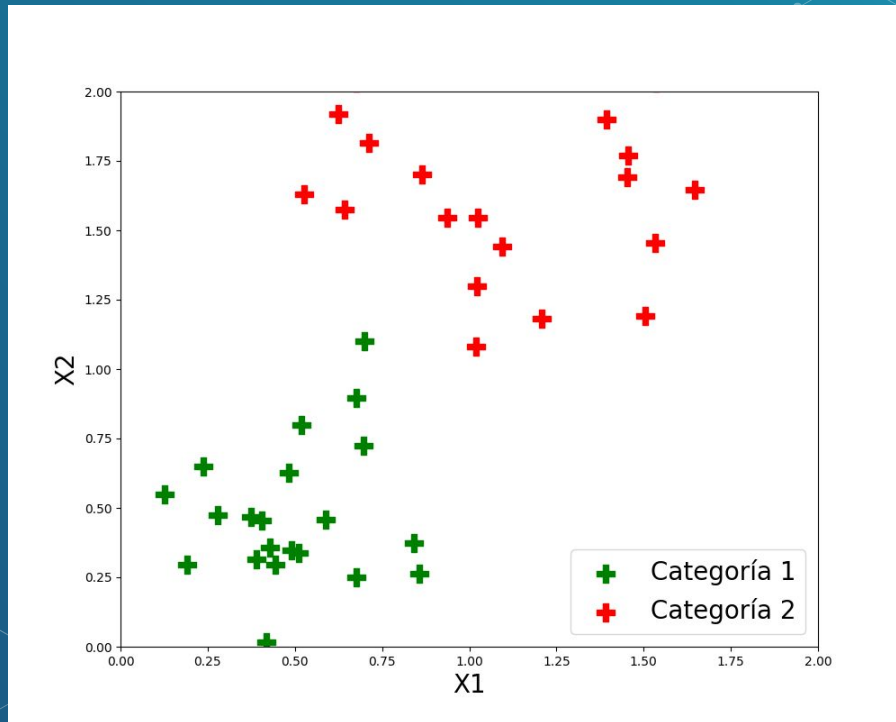
Paso 2: Tomo los 5 vecinos más cercanos al nuevo punto



Paso 3: Entre los 5 vecinos más cercanos cuento cuantos pertenecen a cada categoría.

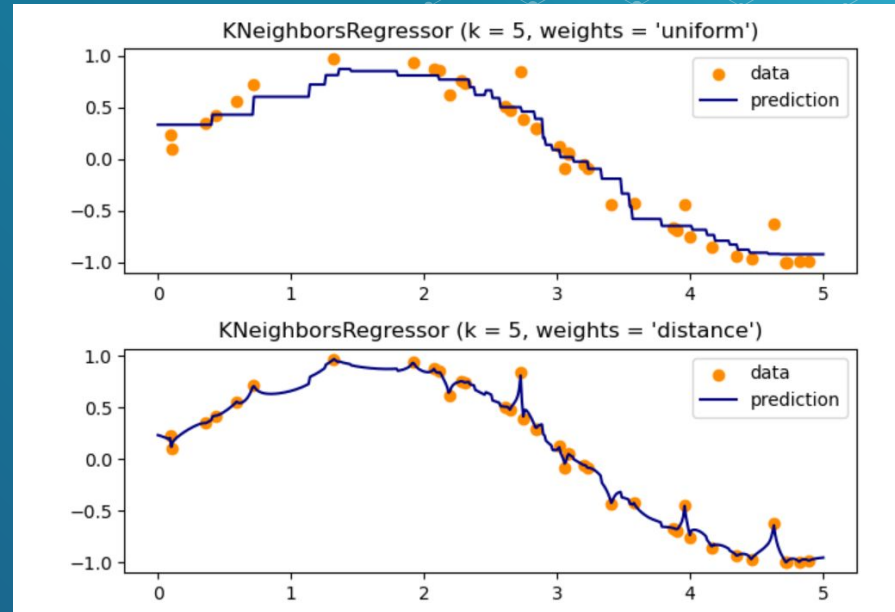


Paso 4: Asigno al nuevo punto la categoría a la categoría más abundante



KNN para regresión

- ◆ Puede ser utilizado como un método de regresión. La salida es computada como el promedio del valor de los k vecinos más cercanos.



Algunas características de KNN

- ◇ Algoritmo supervisado.
- ◇ Aprendizaje basado instancia.
- ◇ KNN es útil para caso de datos no lineales.
- ◇ Algoritmo con pocos hiperparámetros
- ◇ Es un algoritmo “lazy learning”.
 - ◆ Entrenamiento rápido.
 - ◆ Evaluación lenta.
- ◇ Problema de dimensionalidad.
- ◇ Distancia euclídea sensible a las magnitudes.



2

Vamos a su implementación

KNN en scikit-learn (fijado “k”)

Como clasificador:

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)
```

[\[source\]](#)

Como regresor:

```
class sklearn.neighbors.KNeighborsRegressor(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)
```

[\[source\]](#)

KNN en scikit-learn (fijado el radio)

Como clasificador:

```
class sklearn.neighbors.RadiusNeighborsClassifier(radius=1.0, *, weights='uniform', algorithm='auto', leaf_size=30,  
p=2, metric='minkowski', outlier_label=None, metric_params=None, n_jobs=None)
```

[\[source\]](#)

Como regresor:

```
class sklearn.neighbors.RadiusNeighborsRegressor(radius=1.0, *, weights='uniform', algorithm='auto', leaf_size=30, p=2,  
metric='minkowski', metric_params=None, n_jobs=None)
```

[\[source\]](#)

Links útiles

- ◇ Artículo de scikit-learn sobre KNN:
<https://scikit-learn.org/stable/modules/neighbors.html>
- ◇ Algoritmos mencionados:
 - ◆ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>
 - ◆ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsClassifier.html>
 - ◆ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsRegressor.html>
 - ◆ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html>
- ◇ <https://www.ibm.com/es-es/topics/knn#:~:text=El%20algoritmo%20de%20k%20vecinos%20m%C3%A1s%20cercanos%2C%20tambi%C3%A9n%20conocido%20como,un%20punto%20de%20datos%20individual.>