

Examen de la práctica 1: KNN y selección de atributos

Aprendizaje Automático II, 2025-2026

3 octubre de 2025

1. Cree un fichero `KNNRegression.py` con una clase `KNNRegressor` que implemente el algoritmo KNN para problemas de regresión. La clase debe de contar con los mismos métodos que la clase `KNNClassifier` de la práctica 1, pero adaptados a problemas de regresión. En particular, el método `predict` debe devolver la media de las etiquetas de los k vecinos más cercanos.
2. Creen un fichero `PearsonmRMR.py` con la clase `PearsonmRMR` donde la relevancia de un atributo se medirá mediante la correlación de Pearson entre el atributo y la variable objetivo, y la redundancia entre dos atributos mediante la correlación de Pearson entre ambos atributos. Para ello puede utilizar la función `numpy.corrcoef`, que devuelve la matriz de correlación de Pearson entre los vectores proporcionados. En particular, si se le pasan dos vectores x e y , `numpy.corrcoef(x, y)` devuelve una matriz 2×2 cuya diagonal son unos y los elementos fuera de la diagonal corresponden al coeficiente de correlación de Pearson entre x e y .

```
numpy.corrcoef(x, y=None)

import numpy as np

# Dos variables con correlación positiva moderada
x = np.array([10, 20, 30, 40, 50])
y = np.array([12, 24, 33, 47, 55])

# Dos variables casi sin correlación
z = np.array([7, 1, 9, 3, 8])

print("Correlación x-y:")
```

```

print(np.corrcoef(x, y))

print("\nCorrelación x-z:")
print(np.corrcoef(x, z))

# imprime:
Correlación x-y:
[[1.          0.99717646]
 [0.99717646 1.          ]]

Correlación x-z:
[[ 1.          -0.24019223]
 [-0.24019223  1.          ]]

```

3. Utilice las clases creadas para construir un modelo de predicción para el problema de diabetes de Sklearn. Cumpla además los siguientes requisitos:
 - a) Solo puede trabajar con los 5 atributos más relevantes según PearsonRMR.
 - b) Deberá de construirse un pipeline que incluya la normalización de los datos, selección de atributos y el modelo de KNN regresión.
 - c) Utilice validación cruzada con 3 folds para elegir el mejor valor de k entre los valores $\{1, 3, 5, 7, 15\}$.
 - d) Deberá reportar el error cuadrático medio (MSE) estimado.

Puede leer el dataset de diabetes con:

```

from sklearn.datasets import load_diabetes
data = load_diabetes()

```