



Universidad
Rey Juan Carlos

Práctica 3. K-Means, PCA y SVM (I)

Aprendizaje Automático

**Grado en Ingeniería de Robótica
Software**

Curso 2023/2024

Ejercicio 1. Método K-Means.

Se dispone de un conjunto de datos de entrenamiento almacenados en "trainingsetkmeans.csv".

Resuelva los siguientes apartados:

1.1. Realice un script de Python que represente los datos en una figura e indique el número de agrupaciones que observa.

1.2. A continuación genere el modelo K-Means mediante la función de alto nivel de Scikit-Learn, incluyendo el número de agrupaciones que ha observado para detectarlas automáticamente. Posteriormente represente de nuevo los datos en una figura etiquetándolos con la clase a la que pertenece cada uno.

1.3. Calcule la media de distancias euclídeas de cada uno de los datos con respecto al centroide al que pertenecen.

Ejercicio 2. Reducción de dimensionalidad mediante PCA.

Se dispone del conjunto de datos de entrenamiento que están almacenados en "trainingsetPCA.csv". Utilizando la técnica de PCA y otros procesos que considere oportunos justifique con cuántas características se quedaría del set transformado. Razone y justifique su respuesta.



Universidad
Rey Juan Carlos

Práctica 3. K-Means, PCA y SVM (II)

Aprendizaje Automático

**Grado en Ingeniería de Robótica
Software**

Curso 2023/2024

Ejercicio 3. Máquinas de Soporte Vectorial (SVM).

El archivo “trainingsetSVM1.csv” contiene los datos de un conjunto de entrenamiento formado por 3 características de entrada y 2 posibles clases de salida.

A continuación, resuelva los siguientes apartados:

3.1. Realice un script de Python en el que se representen en una figura los datos del set de entrenamiento “trainingsetSVM1.csv” de manera que se distingan las clases entre sí. ¿Qué forma dibujan los datos de entrenamiento?

3.2. A continuación genere el modelo basado en SVM utilizando funciones de alto nivel proporcionadas por *Scikit-Learn*, estableciendo una función Kernel de tipo lineal. Posteriormente, realice un código que represente de forma aproximada el hiperplano de separación óptimo entre clases.

3.3. Para este apartado, realice un script de Python que cargue el segundo set de entrenamiento (“trainingsetSVM2.csv”). Si genera el modelo basado en SVM con una función Kernel de tipo lineal y obtiene el número de errores en la predicción de las entradas del set de entrenamiento utilizando el modelo generado. ¿Por qué el número de errores no es igual a cero?

3.5. Vuelva a generar el modelo utilizando el segundo set de datos de entrenamiento, pero en esta ocasión con una función Kernel polinomial. ¿Qué errores de predicción del conjunto de datos se consiguen variando el grado del polinomio utilizado en el Kernel?