

Requisitos:

- Matplotlib

<https://matplotlib.org/users/installing.html>

```
python -mpip install -U pip
```

```
python -mpip install -U matplotlib
```

- Scikit-Learn

<http://scikit-learn.org/stable/install.html>

```
pip install -U scikit-learn
```

- Scipy

<https://scipy.org/install.html>

```
pip install -U scipy
```

Ejercicio 1

Aplicad un análisis PCA a los datos de la PEC (recordad que los datos incluyen valores no válidos):

- ¿Cuántos componentes principales son necesarios para representar el 95% de la varianza de los datos originales?
- Reconstruid el conjunto de datos a partir de los 18 componentes principales (mediante el método *inverse_transform*), y calculad la pérdida de información respecto al conjunto original. Para hacerlo, podéis calcular el promedio de las diferencias elevadas al cuadrado entre cada elemento del conjunto reconstruido y del original. ¿Qué relación tiene este valor respecto a las varianzas acumuladas calculadas en el apartado anterior?
- Visualizar los datos originales (2 primeros atributos) y los datos transformados según sus dos componentes principales utilizando un color diferente para cada clase. Comentad el resultado.

Ejercicio 2

Aplicad el método “multidimensional scaling” (MDS) a los datos. Repetid este proceso 3 veces y mostrad una gráfica en dos dimensiones con colores diferentes para cada clase. Comentad los resultados. ¿Por qué cada gráfica es diferente?

Ejercicio 3

Con los clasificadores de scikit-learn i los parámetros indicados a continuación, obtener el *score*, el *training time* i el *prediction time* cuando se aplican a los datos proporcionados utilizando K fold cross-validation (K=5):

- k Nearest Neighbors (módulo KNeighborsClassifier de sklearn.neighbors): con 3, 4 y 5 vecinos (primer parámetro).
- Linear SVM (módulo SVC de sklearn.svm): *kernel*="linear", *C*=0.025, el resto de parámetros, valor por defecto.
- Decision Tree (módulo DecisionTreeClassifier de sklearn.tree): *max_depth*=5, el resto de parámetros, valor por defecto.
- AdaBoost (módulo AdaBoostClassifier de sklearn.ensemble): parámetros por defecto.
- Gaussian Naive Bayes (módulo GaussianNB de sklearn.naive_bayes): parámetros por defecto.