ElasticNet

# library link

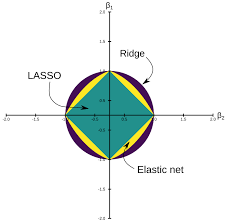
**install :**

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

**github :**

https://github.com/scikit-learn/scikit-learn/blob/7e1e6d09b/sklearn/linear\_model/\_coordinate\_descent.py#L679

# basic description



the elastic net is a regularized regression method that linearly combines the L1 and L2 penalties of the lasso and ridge methods.

# version

* NumPy >= 1.14.6 (pip install numpy)
* Scipy >= 1.1.0 (pip install scipy)
* Joblib >= 0.11 (pip install joblib
* Threadpoolctl >= 2.0.0 (pip install threadpoolctl)
* pandas >= 1.2.4 (pip install pandas)
* matplotlib == 3.22 (pip install matplotlib)

# dataset

* winequality-red.scv:
* The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult: [Web Link] or the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).
* These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.
* P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.  
  Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
* Sources : https://archive.ics.uci.edu/ml/datasets/wine+quality
* or : https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

# code description

* Using Sklearn's ElasticNet, it is a code that predicts total sulfur dioxide by analyzing data in a linear regression method, graphing the predicted and actual values of the test set according to changes in the parameter of ElasticNet (alpha,l1\_ratio), and measured the Mean\_Squared\_error for each graph.

# validation

* Inside the code, the dataset is divided into learning datasets and verification datasets to verify this.

(random\_state = 30)

* Additionally, the fitness between the actual value and the predicted value is evaluated using Sklearn's mean\_squared\_error function.