train_test_split

sklearn.model_selection.train_test_split(**arrays*, *test_size=None*, *train_size=None*, *random_state=None*, *shuffle=True*, *stratify=None*) [source]

Split arrays or matrices into random train and test subsets.

Quick utility that wraps input validation, `next(ShuffleSplit().split(X, y))`, and application to input data into a single call for splitting (and optionally subsampling) data into a one-liner.

Read more in the User Guide.

Parameters:

***arrays** : *sequence of indexables with same length / shape[0]*

Allowed inputs are lists, numpy arrays, scipy-sparse matrices or pandas dataframes.

test_size : *float or int, default=None*

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split. If int, represents the absolute number of test samples. If None, the value is set to the complement of the train size. If `train_size` is also None, it will be set to 0.25.

train_size : *float or int, default=None*

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the train split. If int, represents the absolute number of train samples. If None, the value is automatically set to the complement of the test size.

random_state : *int, RandomState instance or None, default=None*

Controls the shuffling applied to the data before applying the split. Pass an int for reproducible output across multiple function calls. See Glossary.

shuffle : *bool, default=True*

Whether or not to shuffle the data before splitting. If shuffle=False then stratify must be None.

stratify : *array-like, default=None*

If not None, data is split in a stratified fashion, using this as the class labels. Read more in the User Guide.

Returns:

splitting : *list, length=2 * len(arrays)*

List containing train-test split of inputs.

> ***Added in version 0.16:*** If the input is sparse, the output will be a `scipy.sparse.csr_matrix`. Else, output type is the same as the input type.

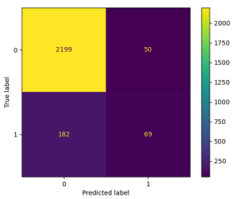
Examples

```
>>> import numpy as np
>>> from sklearn.model_selection import train_test_split
>>> X, y = np.arange(10).reshape((5, 2)), range(5)
>>> X
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
>>> list(y)
[0, 1, 2, 3, 4]
```

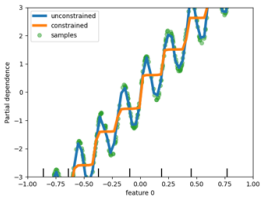
```
>>> X_train, X_test, y_train, y_test = train_test_split(
...     X, y, test_size=0.33, random_state=42)
...
>>> X_train
array([[4, 5],
       [0, 1],
       [6, 7]])
>>> y_train
[2, 0, 3]
>>> X_test
array([[2, 3],
       [8, 9]])
>>> y_test
[1, 4]
```

```
>>> train_test_split(y, shuffle=False)
[[0, 1, 2], [3, 4]]
```

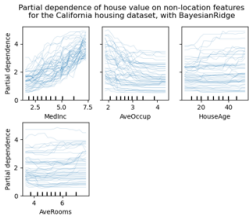
Gallery examples



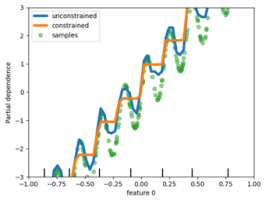
Release Highlights for scikit-learn 1.5



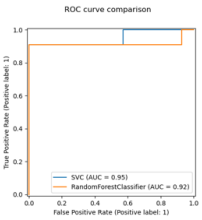
Release Highlights for scikit-learn 1.4



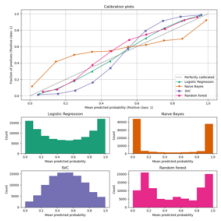
Release Highlights for scikit-learn 0.24



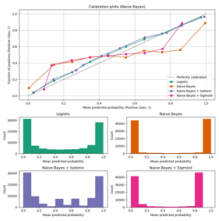
Release Highlights for scikit-learn 0.23



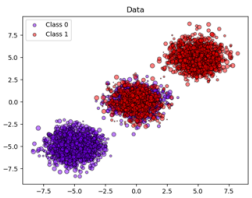
Release Highlights for scikit-learn 0.22



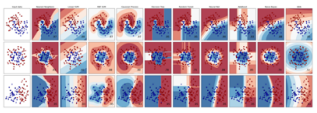
Comparison of Calibration of Classifiers



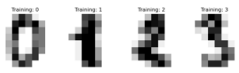
Probability Calibration curves



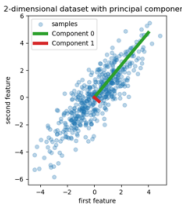
Probability calibration of classifiers



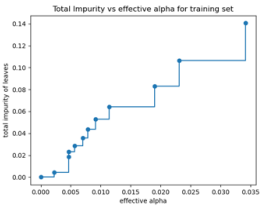
Classifier comparison



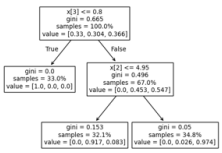
Recognizing hand-written digits



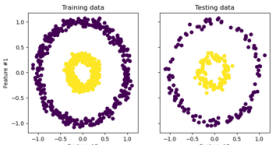
Principal Component Regression vs Partial Least Squares Regression



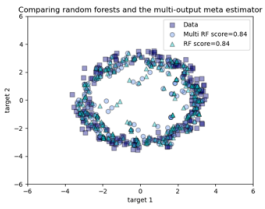
Post pruning decision trees with cost complexity pruning



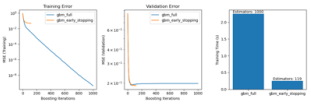
Understanding the decision tree structure



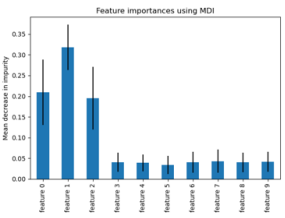
Kernel PCA



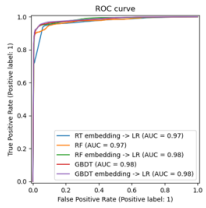
Comparing random forests and the multi-output meta estimator



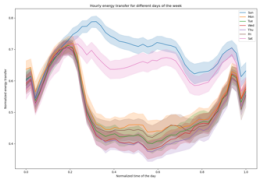
Early stopping in Gradient Boosting



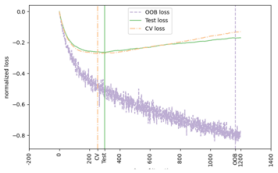
Feature importances with a forest of trees



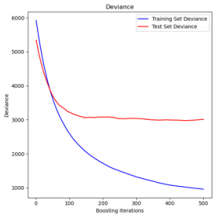
Feature transformations with ensembles of trees



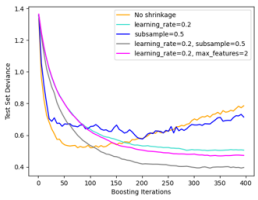
Features in Histogram Gradient Boosting Trees



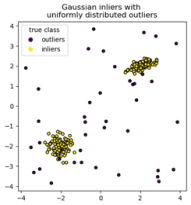
Gradient Boosting Out-of-Bag estimates



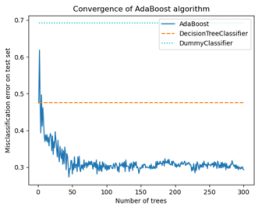
Gradient Boosting regression



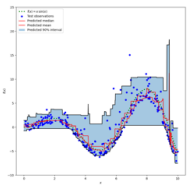
Gradient Boosting regularization



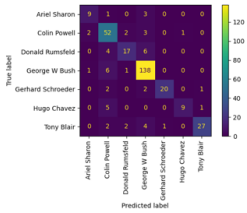
IsolationForest example



Multi-class AdaBoosted Decision Trees

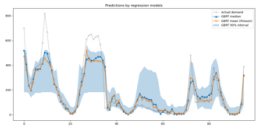


Prediction Intervals for Gradient Boosting Regression

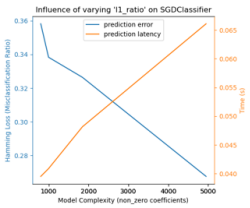


Faces recognition example using eigenfaces and SVMs

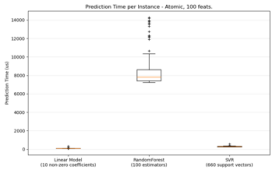
Image denoising using kernel PCA



Lagged features for time series forecasting

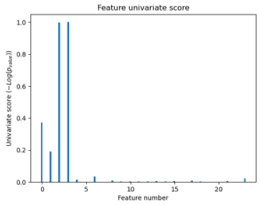


Model Complexity Influence



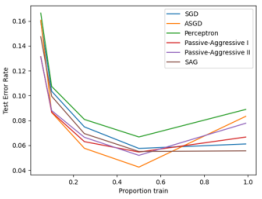
Prediction Latency

Pipeline ANOVA SVM

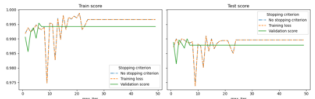


Univariate Feature Selection

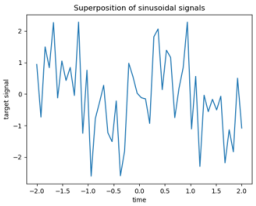
Examples of Using FrozenEstimator



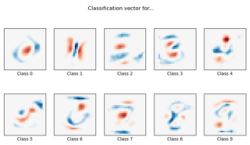
Comparing various online solvers



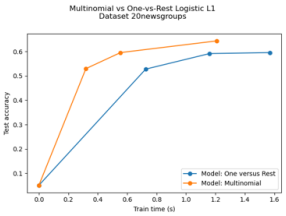
Early stopping of Stochastic Gradient Descent



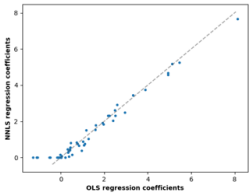
L1-based models for Sparse Signals



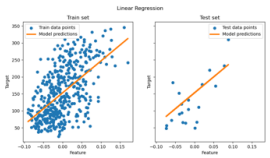
MNIST classification using multinomial logistic + L1



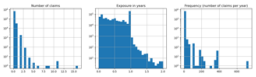
Multiclass sparse logistic regression on 20newgroups



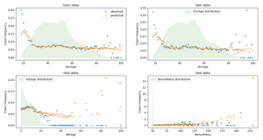
Non-negative least squares



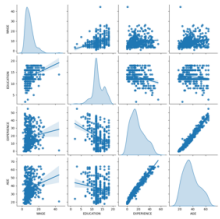
Ordinary Least Squares Example



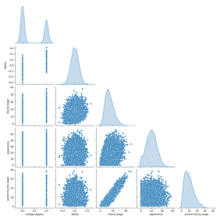
Poisson regression and non-normal loss



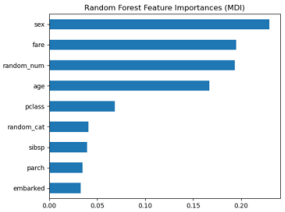
Tweedie regression on insurance claims



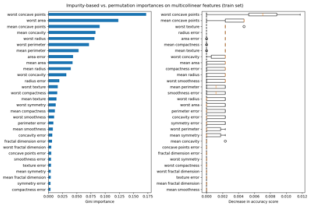
Common pitfalls in the interpretation of coefficients of linear models



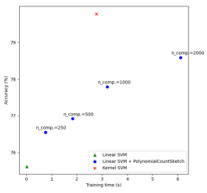
Failure of Machine Learning to infer causal effects



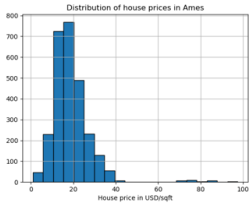
Permutation Importance vs Random Forest



Permutation Importance with Multicollinear or Correlated Features

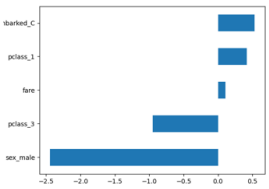


Scalable learning with polynomial kernel approximation

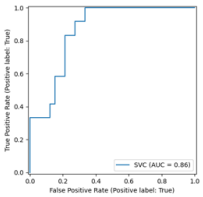


Evaluation of outlier detection estimators

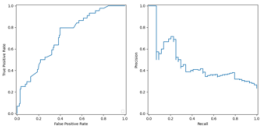
Feature Importance (MDI)



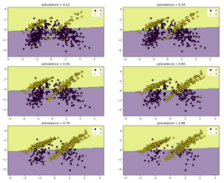
Introducing the set_output API



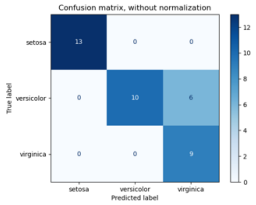
ROC Curve with Visualization API



Visualizations with Display Objects

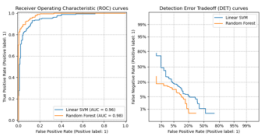


Class Likelihood Ratios to measure classification performance

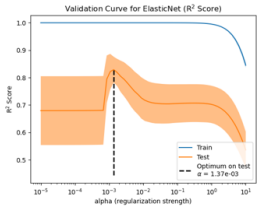


Confusion matrix

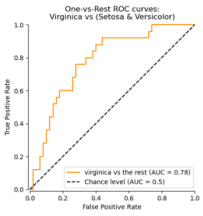
Custom refit strategy of a grid search with cross-validation



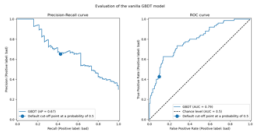
Detection error tradeoff (DET) curve



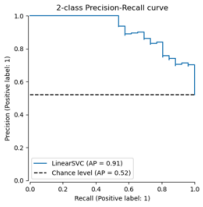
Effect of model regularization on training and test error



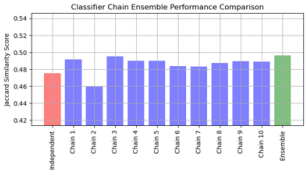
Multiclass Receiver Operating Characteristic (ROC)



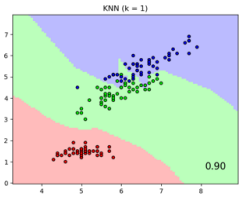
Post-tuning the decision threshold for cost-sensitive learning



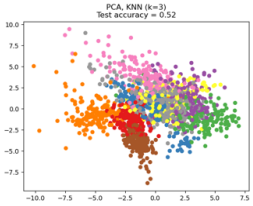
Precision-Recall



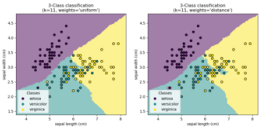
Multilabel classification using a classifier chain



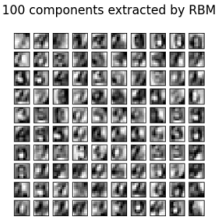
Comparing Nearest Neighbors with and without Neighborhood



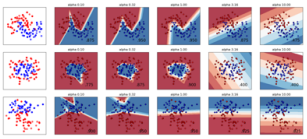
Dimensionality Reduction with Neighborhood



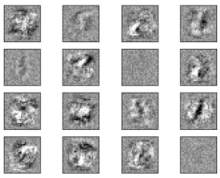
Nearest Neighbors Classification



Restricted Boltzmann Machine features for digit

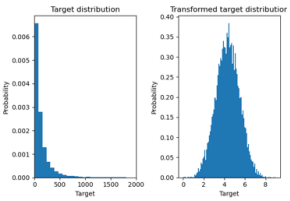


Varying regularization in Multi-layer Perceptron

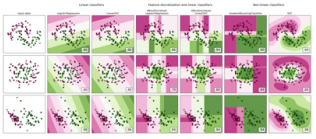


Visualization of MLP weights on MNIST

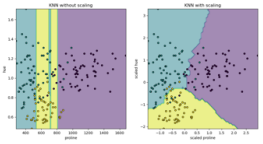
Column Transformer with Mixed Types



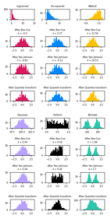
Effect of transforming the targets in regression model



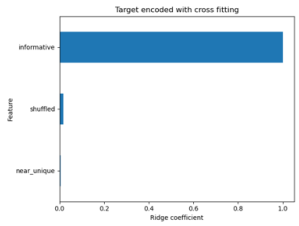
Feature discretization



Importance of Feature Scaling



Map data to a normal distribution



Target Encoder's Internal Cross fitting

Semi-supervised Classification on a Text Dataset

Previous
check_cv

Next
GridSearchCV