Explanation of used python library/necessity/ purpose.

1. from sklearn.model_selection import train_test_split

Used to split arrays or matrices into random train and test subsets

parameters are-

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

test_size float, int or None, optional (default=None) etc.

Return type - List containing train-test split of inputs.

2. from sklearn.neural_network import MLPClassifier

It is multi-layer Perceptron classifier.

Model optimizes the log-loss function using LBFGS or stochastic gradient descent.

3. from sklearn.model_selection import StratifiedKFold

Stratified K-Folds cross-validator

Provides train/test indices to split data in train/test sets.

This cross-validation object is a variation of KFold that returns stratified folds. The folds are made by preserving the percentage of samples for each class.

4. from sklearn.model_selection import GridSearchCV

Exhaustive search over specified parameter values for an estimator.

Important members are fit, predict.

GridSearchCV implements a “fit” and a “score” method. It also implements “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used.

The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

5. from tpot import TPOTClassifier

TPOT is meant to be an assistant that gives an ideas on how to solve a particular machine learning problem by exploring pipeline configurations that we might have never considered, then leaves the fine-tuning to more constrained parameter tuning techniques such as grid search.

6. % matplotlib inline

The % matplotlib inline is an example of predefined magic function in Ipythonand frequently used in interactive environments like jupyter notebook. % matplotlib inline makes your plot outputs appear and be stored within the notebook.

7. from scipy import signal

Signal processing toolbox currently contains some filtering functions, a limited set of filter design tools, and a few B-spline interpolation algorithms for 1- and 2-D data.

8. import pickle

Pickling is the serializing and de-serializing of python objects to a byte stream. Unpicking is the opposite.

9. sklearn.model_selection import cross_val_score

Evaluate a score by cross-validation

10. from sklearn import svm

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using `sklearn.svm.LinearSVC` or `sklearn.linear_model.SGDClassifier` instead, possibly after a `sklearn.kernel_approximation.Nystroem` transformer.

The multiclass support is handled according to a one-vs-one scheme.

11. from sklearn.metrics import precision_recall_fscore_support

Compute precision, recall, F-measure and support for each class

The precision is the ratio `tp / (tp + fp)` `tp` is the number of true positives and `fp` the number of false positive. precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

The recall is the ratio `tp / (tp + fn)` `tp` is the number of true positives and `fn` the number of false negative. The recall is intuitively the ability of the classifier to find all the positive sample.