

Overview

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

This python library allows users to conduct statistical and machine learning operations on given datasets with minimal work. The library has uses for *classification*, *regression*, *clustering*, *dimensionality reduction*, and *data preprocessing*. These operations can allow you to extract features, normalize data, conduct forecasts, and various other (progressively more advanced) machine learning techniques.

Documentation: http://scikit-learn.org/stable/user_guide.html

Installation

Anaconda typically includes a stable distribution of SKLearn, but if it is not installed; Scikit-learn requires:

- Python (≥ 2.7 or ≥ 3.3),
- NumPy ($\geq 1.8.2$),
- SciPy ($\geq 0.13.3$).

Installation can be completed from the command prompt using *pip* or *conda*.

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

```
conda install scikit-learn
```

Machine Learning Flow

1. Import libraries
2. Load dataset and assign x, y variables
3. Split variables into training and test sets
4. Feature-scale the data if it is highly variable
5. Fit the classifier to the training data
6. Predict the output for the test data
7. Verify accuracy of predictions, and optionally visualize results

Example: Predicting Malignancy of Breast Cancer Sample

Dataset : University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg

breastCancer.csv - <https://goo.gl/4uWY4k>

```

1. """
2. breastCancer.py
3. https://goo.gl/RHsyne
4. Random Forest Classification
5. Predicting Malignancy of Breast Cancer data
6. 2: Benign, 4: Malignant
7. """
8. #      Importing the libraries
9. import pandas as pd
10.
11. #      Importing the dataset
12. dataset = pd.read_csv('breastCancer.csv')
13. X = dataset.iloc[:, 1:10].values
14. y = dataset.iloc[:, 10].values
15.
16. #      Splitting the dataset into the Training set and Test set
17. from sklearn.cross_validation import train_test_split
18. X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
19.
20. #      Fitting Random Forest Classification to the Training set
21. from sklearn.ensemble import RandomForestClassifier
22. classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy')
23. classifier.fit(X_train, y_train)
24.
25. #      Predicting the Test set results
26. y_pred = classifier.predict(X_test)
27.
28. #      Making the Confusion Matrix
29. from sklearn.metrics import confusion_matrix
30. cm = confusion_matrix(y_test, y_pred)
31. print(cm)

```

Confusion matrix:

	Predicted Benign	Predicted Malignant
Actual Benign	109	3 [Type I]
Actual Malignant	3 [Type II]	60

- Only 3 Type-I errors and 3 Type-II errors
- Correct / Total = $\frac{109+60}{109+60+3+3} = 96.57\%$ accuracy.