Tut 4: Hướng dẫn áp dụng Decision Tree và Random Forest Bằng thư viện Scikit-Learn

- **Tác giả: Cao Chánh Dương.**
- **Nhóm nghiên cứu về AI của Trường Đại học Bách Khoa Thành phố Hồ Chí Minh.**

1. Cài đặt thư viện Scikit-Learn

- Tham khảo Tut-0.

2. Áp dụng Decision Tree (trên tập dữ liệu Toy Dataset Iris)

- Đoạn code sau đây sẽ import những thư viện cần thiết:

```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
```

- Trong đó dòng đầu là ta thực hiện import tập dữ liệu Iris vào, dòng sau là để ta import module tree để thực hiện Decision Tree.
- Sau đó ta load tập dữ liệu vào biến và tiến hành chia tập train và test:

```
>>> iris = load_iris()
>>> df = pd.DataFrame(iris.data, columns=iris.feature_names)
>>> from sklearn.model_selection import train_test_split
>>> X_train, X_test, y_train, y_test = train_test_split(df[iris.feature_names], iris.target, test_size=0.5, stratify=iris.target, random_state=123456)
```

- Cuối cùng là sử dụng hàm DecisionTreeClassifier có sẵn trong module tree và fit model dữ liệu vào.

```
>>> rf =  tree.DecisionTreeClassifier(max_depth=5)
>>> rf.fit(X_train, y_train)
>>> predicted = rf.predict(X_test)
```

- Ta làm tương tự nếu ta muốn sử dụng Regressor:

```
>>> rf= tree.DecisionTreeRegressor(max_depth=5)
```

- Sau đó ta đã có thể kiểm tra độ chính xác của việc phân loại:

  >>> accuracy = accuracy_score(y_test, predicted)

- Học viên tham khảo thêm tại link sau:

  http://scikit-learn.org/stable/modules/tree.html#classification

  http://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#sphx-glr-auto-examples-tree-plot-tree-regression-py

3. Áp dụng Random Forest Classifier (trên tập dữ liệu Iris)

- Đầu tiên, ta load Dataset vào biến:

  from sklearn import datasets
  iris = datasets.load_iris()

- Tiến hành chia tập Train và Test (các bạn có thể điều chỉnh lại tỉ lệ nếu muốn):

  from sklearn.model_selection import train_test_split

  df = pd.DataFrame(iris.data, columns=iris.feature_names)

  X_train, X_test, y_train, y_test = train_test_split(df[iris.feature_names], iris.target, test_size=0.5, stratify=iris.target, random_state=123456)

- Tiến hành áp dụng RandomForestClassifier và fit model vào:

  from sklearn.ensemble import RandomForestClassifier

  rf = RandomForestClassifier(n_estimators=100, oob_score=True, random_state=123456)

  rf.fit(X_train, y_train)

- Cuối cùng, ta kiểm tra độ chính xác của model bằng cách tính accuracy:

  from sklearn.metrics import accuracy_score
  predicted = rf.predict(X_test)
  accuracy = accuracy_score(y_test, predicted)

- Học viên có thể tham khảo thêm tại link sau: http://www.blopig.com/blog/2017/07/using-random-forests-in-python-with-scikit-learn/

4. Áp dụng Random Forest Regressor (cho tập dữ liệu Boston):

- Đầu tiên ta import các thư viện cần thiết, load dữ liệu và các feature vào :

```
from sklearn import datasets
import numpy as np
import pandas as pd
boston = datasets.load_boston()

features = pd.DataFrame(boston.data, columns=boston.feature_names)

targets = boston.target
```

- Thực hiện chia các tập dữ liệu:

```
X_train, X_test, y_train, y_test = train_test_split(features, targets,
train_size=0.8, random_state=42)
```

- Tiến hành tạo model:

```
from sklearn.ensemble import RandomForestRegressor

rf=RandomForestRegressor(n_estimators=500,oob_score=True,random_state=
0)

rf.fit(X_train, y_train)
```

- Tới đây, chúng ta đã thực hiện xong và tiến hành kiểm tra độ chính xác của model bằng nhiều cách khác nhau:

```
from sklearn.metrics import r2_score

from scipy.stats import spearmanr, pearsonr
predicted_train = rf.predict(X_train)
predicted_test = rf.predict(X_test)
 test_score = r2_score(y_test, predicted_test)
 spearman = spearmanr(y_test, predicted_test)
```

pearson = pearsonr(y_test, predicted_test)

- Lưu ý: Các bạn có thể chuẩn hóa dữ liệu đầu vào để cho việc thực hiện mang lại kết quả tốt hơn, các bạn tham khảo trong link chi tiết sau nếu muốn cải thiện : **http://www.blopig.com/blog/2017/07/using-random-forests-in-python-with-scikit-learn/**