Scikit Learn

Régression sur le prix des maisons à boston

1) Import de dataset

```
]:
1 from sklearn import datasets
```

```
2]:
1 boston = datasets.load_boston()
2 print(boston["DESCR"])
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------
```

1.1) récupération des X et Y

```
:
1 X = boston["data"]
2 y = boston["target"]
```

2) separer le jeu d'apprentissage du jeu de test

```
:
1 from sklearn.model_selection import train_test_split
```

```
:
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

3) choisir un algo et le tester

Documentation du Random Forest

```
:
1 from sklearn.ensemble import RandomForestRegressor
```

```
]:
1 algo        = RandomForestRegressor()
2 modele      = algo.fit(X_train, y_train)
3 performance = modele.score(X_test, y_test)
```

```
]:
1 performance
```

```
0.8597588038373313
```

4) grille de recherche

Documentation du Random Forest

Documentation grid search

Documentation grid search

```
]:  1  from sklearn.model_selection import GridSearchCV
```

```
]:  1  hyperparameters = {"max_depth" : [1,2,3]}
```

```
]:  1  grille = GridSearchCV(algo, hyperparameters)
```

```
]:  1  grille = grille.fit(X_train, y_train)
```

4.2) résultats

```
]:  1  meilleur_model = grille.best_estimator_
```

```
]:  1  grille.score(X_test, y_test)
```

```
0.8105203810236372
```

4.3) récupérer le modèle pour faire des prédictions

```
]:  1  une_ligne  = X_test[1]
    2  prediction = meilleur_model.predict([une_ligne])
```

```
]:  1  prediction
```

```
array([22.13838832])
```