In [21]:

```
import numpy as np
from sklearn import datasets
from  sklearn.ensemble import BaggingRegressor
from  sklearn.ensemble import RandomForestRegressor
```

Чтобы BaggingRegressor выбирал случайное подмножество фичей для каждого дерева, достаточно указать параметр max_features

In [22]:

```
clf_bag = BaggingRegressor(max_features=5)
clf_rand = RandomForestRegressor(max_features=5)
```

In [23]:

```
boston = datasets.load_boston()
```

In [28]:

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
```

In [27]:

```
clf_bag.fit(X_train, y_train)
clf_rand.fit(X_train, y_train)
```

Out[27]:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features=5, max_leaf_nodes=None, min_impurity_split=1e-07,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
           oob_score=False, random_state=None, verbose=0, warm_start=False)
```

In [30]:

```
b = cross_val_score(clf_bag, boston.data, boston.target, cv=5,
scoring='neg_mean_squared_error')
r = cross_val_score(clf_rand, boston.data, boston.target, cv=5,
scoring='neg_mean_squared_error')
```

In [32]:

```
b.mean()
```

Out[32]:

```
-31.603481104216151
```

In [33]:

```
r.mean()
```

Out[33]:

```
-22.922560291205588
```

RandomForest работает все-таки лучше.