random_forest

January 30, 2019

```python
In [1]: import numpy
        from sklearn.utils import shuffle
        from sklearn.model_selection import train_test_split
        X = numpy.loadtxt("./data/Train/X_train.txt")
        y = numpy.loadtxt("./data/Train/y_train.txt")

In [2]: X, y = shuffle(X, y)
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

In [3]: from models import random_forest
        from sklearn.metrics import accuracy_score
        ls = [5, 10, 50, 100, 561]
        report = {}

In [4]: from sklearn.feature_selection import SelectKBest, mutual_info_classif
        report['mutual_info_classif'] = []
        for l in ls:
            transformer = SelectKBest(mutual_info_classif, k=l)
            X_train_new = transformer.fit_transform(X_train, y_train)
            random_forest_model = random_forest.without_penalty(X_train_new, y_train)
            y_pred = random_forest_model.predict(transformer.transform(X_test))
            score = accuracy_score(y_test, y_pred)
            data={
                'l': l,
                'score': score
            }
            report['mutual_info_classif'].append(data)

In [5]: report['mutual_info_classif']

Out[5]: [{'l': 5, 'score': 0.640926640926641},
         {'l': 10, 'score': 0.675032175032175},
         {'l': 50, 'score': 0.7844272844272844},
         {'l': 100, 'score': 0.9401544401544402},
         {'l': 561, 'score': 0.9601029601029601}]

In [8]: from sklearn.feature_selection import SelectFromModel
        clf = random_forest.without_penalty(X_train, y_train)
```

```
        report['sfm_rf'] = []
        for l in ls:
            transformer = SelectFromModel(clf, prefit=True, max_features=l)
            X_train_new = transformer.transform(X_train)
            random_forest_model = random_forest.without_penalty(X_train_new, y_train)
            y_pred = random_forest_model.predict(transformer.transform(X_test))
            score = accuracy_score(y_test, y_pred)
            data={
                'l': l,
                'score': score
            }
            report['sfm_rf'].append(data)

In [9]: report['sfm_rf']

Out[9]: [{'l': 5, 'score': 0.8211068211068211},
         {'l': 10, 'score': 0.8384813384813384},
         {'l': 50, 'score': 0.9491634491634492},
         {'l': 100, 'score': 0.9536679536679536},
         {'l': 561, 'score': 0.9523809523809523}]
```