project_b4/reconstruction/machine_learning/energy_regression.py
2021-06-25T18:25+02:00

```python
import numpy as np
from sklearn import (
    ensemble, linear_model, neighbors, svm, tree, naive_bayes,
    gaussian_process, neural_network, dummy)
from sklearn.model_selection import KFold
from sklearn.base import clone
from tqdm import tqdm


def define_model(seed):
    """A helper function to retrieve a model for the energy regression
    exercise. The parameter seed gets used to set the random state of the model
    and ensure reproducible results"""

    # ---------
    # Exercise 19 (Sheet 9):
    # ---------
    # Energy_regression (exercises/energy_regression.py)

    model = ensemble.RandomForestRegressor(random_state=seed)

    return model


def cross_validate_model(X, y, model, seed):
    """This function implements a cross validation on a given model.

    Required return values:
    -----------------------
    predictions: np.array of the same shape as y
        (These are the predictions on the test sets combined into one array)
    true_values: np.array of the same shape as y
        (These are the y's chosen in the different cross validation steps
        combined into one array. This equals y but with different order.)
    models: list of (n_cross_validation_steps) models
        These are used to calculate the feature importances later
    """
    # ---------
    # Exercise 19 (Sheet 9):
    # ---------
    # Energy_regression (exercises/energy_regression.py)

    kf = KFold(n_splits=5, shuffle=True, random_state=seed)

    predictions = []
    true_values = []
    models = []

    for train_index, test_index in kf.split(X):
        X_train, y_train = X.iloc[train_index], y[train_index]
        X_test, y_test = X.iloc[test_index], y[test_index]

        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

        predictions.append(y_pred)
        true_values.append(y_test)
        models.append(model)

    predictions = np.concatenate(predictions)
    true_values = np.concatenate(true_values)

    return predictions, true_values, models
```