Pipelines

There are many different possible workflows for any given data set when we account for transforms, feature engineering, model selection and model tuning. This means that we need a systematic way to compare these workflow variants. This is where pipelines become so useful. It is the consistency of the three interfaces in sk-learn that allow us make try various pipelines and compare their results as part of the iterative workflow.

```python
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectKBest
from sklearn.metrics import median_absolute_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_boston

## load the boston dataset
boston = load_boston()
X, y = boston['data'], boston['target']
features = boston['feature_names']

## split the data to a training set and a test set
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

## create a pipeline
pipe = Pipeline([("scaler", StandardScaler()),
                 ("featsel", SelectKBest(k=10)),
                 ("rf", RandomForestRegressor(n_estimators=20))])

## train on the training data
pipe.fit(X_train, y_train)

## evaluate the model with the test data
y_pred = pipe.predict(X_test)
print(r'R^2=%.2f, MAE=%.2f'%(r2_score(y_test, y_pred), median_absolute_error(y_test, y_pred)))
```

```
R^2=0.74, MAE=1.54
```

Here we are standardizing the data before selecting the 10 best features according to an ANOVA test. These transformed data are then piped into a random forest regression model. See the SelectKBest class to see the other options that are available as a scoring function. It is worth mentioning that the three scikit-learn interfaces applied with pipelines have had such an impact on the data science workflow that Apache Spark now has similar ML pipelines.

Additional resources

- Cognito: Automated Feature Engineering
- Feature engineering via PCA using Watson Studio Local
- scikit-learn pipeline tutorial example