Random Forests with Scikit-learn

This section will implement Random forests with Scikit-learn for both regression and classification use cases.

Random Forests for Classification

In this code example, we will build a Random forest classification model to predict the species of flowers from the Iris dataset.

```
# import packages
from sklearn.ensemble import RandomForestClassifier
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# load dataset
data = datasets.load_iris()

# separate features and target
X = data.data
y = data.target

# split in train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True)

# create the model
rf_classifier = RandomForestClassifier()

# fit the model on the training set
rf_classifier.fit(X_train, y_train)
```

```
# make predictions on the test set
predictions = rf_classifier.predict(X_test)

# evaluate the model performance using accuracy metric
print("Accuracy: %.2f" % accuracy_score(y_test, predictions))

'Output":
Accuracy: 1.00
```

Random Forests for Regression

In this code example, we will build a Random forest regression model to predict house prices from the Boston house-prices dataset.

```
# import packages
from sklearn.ensemble import RandomForestRegressor
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt

# load dataset
data = datasets.load_boston()

# separate features and target
X = data.data
y = data.target

# split in train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True)
```

```
# create the model
rf_reg = RandomForestRegressor()

# fit the model on the training set
rf_reg.fit(X_train, y_train)

# make predictions on the test set
predictions = rf_reg.predict(X_test)

# evaluate the model performance using the root mean square error metric
print("Root mean squared error: %.2f" % sqrt(mean_squared_error(y_test,
predictions)))
```

```
'Output':
Root mean squared error: 2.96
```

Stochastic Gradient Boosting (SGB)

Boosting involves growing trees in succession using knowledge from the residuals of the previously grown tree. In this case, each successive tree works to improve the model of the previous tree by boosting the areas in which the previous tree did not perform so well without affecting the areas of high performance. By doing this, we iteratively create a model that reduces the residual variance when generalizing to test examples. Boosting is illustrated in Figure 23-4.