ランダムフォレストの適用パターン
・パターン名
ランダムフォレストによる数値予測
・目的
回帰タスク
・問題

・適用条件
1. 重要特徴量の数が極端に少なくない
   → ランダムフォレストによる重要特徴量抽出パターンにて確認
2. 数値変数およびカテゴリカル変数のみを持つ
3. Tabularデータである

・適用手順
1. データの読み込み
2. データ必要に応じてデータを整形（NaNの扱いなど）
3. ランダムフォレストモデルへ訓練データを登録
4. ランダムフォレストモデルのパラメータを決定
5. 決定したパラメータでランダムフォレストモデルを学習
6. テストデータへ学習後モデルを適用

・実装上の注意
1. scikit-learnのみ, カテゴリカル変数をone-hot-vectorなどに変換
2. パラメータはグリッドサーチなどで求める
3. グリッドサーチに時間がかかりすぎるときはベイズ最適化などを用いる

ランダムフォレストの適用パターン
・パターン名
ランダムフォレストによる数値予測

・サンプルコード(python)

```
from sklearn.ensamble import RandomForestRegressor
import pandas as pd
import numpy as np
train = pd.read_csv(“sample_train.csv”)
test = pd.read_csv(“sample_test.csv”)
target =  train[“target”].values
train = train.drop([target], axis=1)
model = RandomForestRegressor()
model = model.fit(train,target)
result = model.predict(test)
```

・適用結果
numpy.ndarray([1.2,2.3,…])などのnumpy配列形式のデータが得られる

・理論的背景
相互に相関があるような複数の予測器を学習し, 各予測器での予測結果を出力し, 組み合わせることで
汎化誤差を低減させる。また, 各予測器を学習させる際にはN個ある特徴量のうちM個のみ（M<N）を
用いることで各予測器が同じものになることを防いでいる。このような手法をアンサンブル手法と呼ぶ。
特にランダムフォレストはバギングであり, 他の関連としてはブースティングがある。

・出典
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

・関連するパターン
ランダムフォレストによる変数重要度抽出パターン
ランダムフォレストによる分類パターン

ランダムフォレストの適用パターン
・パターン名
ランダムフォレストによる分類
・目的
分類タスク
・問題

・適用条件
1. 重要特徴量の数が極端に少なくない
   → ランダムフォレストによる重要特徴量抽出パターンにて確認
2. 数値変数およびカテゴリカル変数のみを持つ
3. Tabularデータである
4. ターゲットが数値化されたカテゴリである

・適用手順
1. データの読み込み
2. データ必要に応じてデータを整形（NaNの扱いなど）
3. ランダムフォレストモデルへ訓練データを登録
4. ランダムフォレストモデルのパラメータを決定
5. 決定したパラメータでランダムフォレストモデルを学習
6. テストデータへ学習後モデルを適用

・実装上の注意
1. scikit-learnのみ, カテゴリカル変数をone-hot-vectorなどに変換
2. パラメータはグリッドサーチなどで求める
3. グリッドサーチに時間がかかりすぎるときはベイズ最適化などを用いる

ランダムフォレストの適用パターン
・パターン名
ランダムフォレストによる分類

・サンプルコード(python)

```
from sklearn.ensamble import RandomForestClassifier
import pandas as pd
import numpy as np
train = pd.read_csv(“sample_train.csv”)
test = pd.read_csv(“sample_test.csv”)
target =  train[“target”].values
train = train.drop([target], axis=1)
model = RandomForestRegressor()
model = model.fit(train,target)
result = model.predict(test)
```

・適用結果
numpy.ndarray([2,1,0,…])などのnumpy配列形式のクラス出力が得られる

・理論的背景
相互に相関があるような複数の予測器を学習し, 各予測器での予測結果を出力し, 組み合わせることで
汎化誤差を低減させる。また, 各予測器を学習させる際にはN個ある特徴量のうちM個のみ（M<N）を
用いることで各予測器が同じものになることを防いでいる。このような手法をアンサンブル手法と呼ぶ。
特にランダムフォレストはバギングであり, 他の関連としてはブースティングがある。

・出典
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

・関連するパターン
ランダムフォレストによる変数重要度抽出パターン
ランダムフォレストによる分類パターン

ランダムフォレストの適用パターン
・パターン名
ランダムフォレストによる変数重要度の抽出
・目的
変数重要度の抽出

・問題

・適用条件
1. 数値変数およびカテゴリカル変数のみを持つ
2. Tabularデータである

・適用手順
1. データの読み込み
2. データ必要に応じてデータを整形（NaNの扱いなど）
3. ランダムフォレストモデルへ訓練データを登録
4. ランダムフォレストモデルのパラメータを決定
5. 決定したパラメータでランダムフォレストモデルを学習
6. 学習後のモデルから特徴量の重要度を取り出す

・実装上の注意
1. scikit-learnのみ, カテゴリカル変数をone-hot-vectorなどに変換
2. パラメータはグリッドサーチなどで求める
3. グリッドサーチに時間がかかりすぎるときはベイズ最適化などを用いる

ランダムフォレストの適用パターン
・パターン名
ランダムフォレストによる変数重要度の抽出

・サンプルコード(python)

```
from sklearn.ensamble import RandomForestRegressor
import pandas as pd
import numpy as np
train = pd.read_csv(“sample_train.csv”)
target =  train[“target”].values
train = train.drop([target], axis=1)
model = RandomForestRegressor()
model = model.fit(train,target)
result = model.feature_importances_
```

*RandomForestClassifierでも可能

・適用結果
numpy.ndarray([1.2,2.3,…])などのnumpy配列形式のデータが得られる

・理論的背景
相互に相関があるような複数の予測器を学習し, 各予測器での予測結果を出力し, 組み合わせることで
汎化誤差を低減させる。また, 各予測器を学習させる際にはN個ある特徴量のうちM個のみ（M<N）を
用いることで各予測器が同じものになることを防いでいる。このような手法をアンサンブル手法と呼ぶ。
特にランダムフォレストはバギングであり, 他の関連としてはブースティングがある。

・出典
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

・関連するパターン
ランダムフォレストによる変数重要度抽出パターン
ランダムフォレストによる分類パターン