

# Week 1

takagishi

予測モデル：

- ・製品の総量の予測(数値の予測)なので、回帰モデル。

*f*

- ✓ 線形回帰系
  - 線形回帰
  - Ridge回帰
  - Lasso回帰
  - Elastic Net
- ✓ 決定木(回帰木)系
  - 決定木, ランダムフォレスト, 勾配ブースティング回帰木(GBRT), ...

- ✓ 時系列系
  - AR ... 自己回帰(Auto Regressive)
  - MA ... 移動平均(Moving Average)
  - ARMA ... AR + MA
  - ARIMA ... ARMA + Integrated(和分[積分])
  - SARIMA ... ARIMA + Seasonal(季節変動)

- ✓ ニューラルネット系
  - NN, RNN(LSTM)



今回はこの  
勾配ブースト回帰木  
つかいました。  
Python実装  
Xgboost  
windows anaconda環境の人は  
「conda install -c anaconda py-xgboost」  
でインストール簡単

Xgboostって :

● 元論文

[http://delivery.acm.org/10.1145/2940000/2939785/p785-chen.pdf?ip=203.179.35.3&id=2939785&acc=CHORUS&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&\\_\\_acm\\_\\_=1532530471\\_6d0713472a59a9fb5d642685cbc5c7bf](http://delivery.acm.org/10.1145/2940000/2939785/p785-chen.pdf?ip=203.179.35.3&id=2939785&acc=CHORUS&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&__acm__=1532530471_6d0713472a59a9fb5d642685cbc5c7bf)

● 内部は

<http://kefism.hatenablog.com/entry/2017/06/11/182959>

● 特徴

特徴量のsubsampling  
データのsubsampling

今までに学習したモデルの情報を使って、  
新たなモデルを構築することでデータの学習を進める方法をブースティング(Boosting)と呼び（中略）  
1つ前までの決定木の結果を利用して新たな決定木を作る際に、  
実測値と予測値との誤差が**ある意味で**最小になるように決定木の[アルゴリズム](#)を構築したものがXGBoost

● ハイパーパラメータ

パラメータ名	デフォルト値	パラメータ内容	メモ
eta	0.3	各ステップごとの学習率	低いと過学習抑制に繋がる
max_depth	6	木の深さの上限	深すぎると過学習を引き起こす
min_child_weight	1	葉の重みの下限	下限が大きいかほど単調な木になり過学習抑制
subsample	1	各ステップの決定木の構築に用いるデータの割合	ランダムサンプリングすることで過学習抑制
colsample_bytree	1	各ステップの決定木ごとに用いる特徴量の割合	特徴量間の関係を考慮した形でモデリング可能
lambda	1	L2正則化項の重み	値を大きくすると過学習抑制