

# 需要予測 実践チュートリアル

---

Recruit Restaurant Visitor Forecasting

# Index

---

- ・コンペティション概要
  - ・評価指標
  - ・データ概要
- ・ EDA
- ・データ前処理と特徴量エンジニアリング
- ・モデリング
- ・精度評価

# コンペティション概要

---

# コンペティション概要

---

- ・日本の会社リクルートが開催した, レストランの来客数予測コンペ
- ・ホットペッパーグルメ(hpg)と, Airレジ(air)というリクルートが提供するPOS レジアプリのデータを用いて予測を行う
- ・評価指標はRoot Mean Squared Logarithmic Error(RMSLE)

# 評価指標

---

- ・RMSLEという指標で評価を行う
- ・一般によく用いられるRMSEと比較して以下の特徴がある
  - ・実際の客数より少なく予測した場合, より大きなペナルティを与える
    - 予測を外すことで仕入れや人員が不足する事態は避けたい
  - ・客数の分布にかなり偏りがあるので, 目的変数の分布を正規分布に近づけた いい 意図もあるかも

$$RMSLE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2}$$

# データ概要

---

- ・レストラン, 日付ごとのMulti-Index
- ・hpgとairの予約情報
  - ・予約客数
  - ・予約した時間
  - ・予約が行われた時間
  - ・各予約での来客数
- ・各レストランの市町村情報, 緯度経度, ジャンル(和食, フレンチなど)
- ・曜日や祝日を表すデータ
- ・実際の来客データ(これを予測する)

# EDA

---

# EDAとは

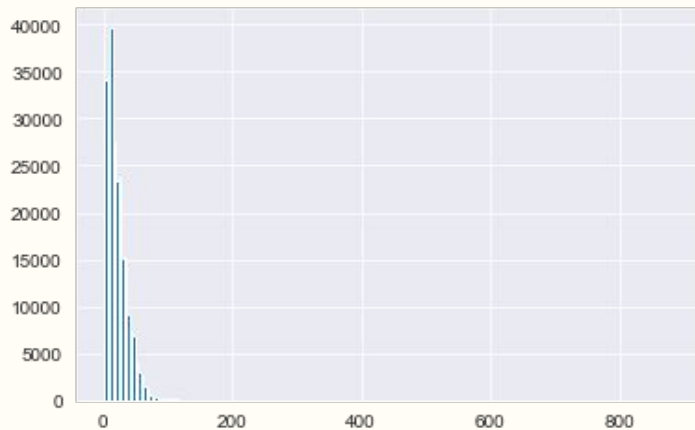
---

- Exploratory Data Analysisの略で、探索的データ分析と訳される
- データを分析し、データと現象の関係を見出すこと
- データから現象の理解を進め、ビジネスに適用
- 機械学習モデルを構築する際のヒントに ← 今回は主にこっち

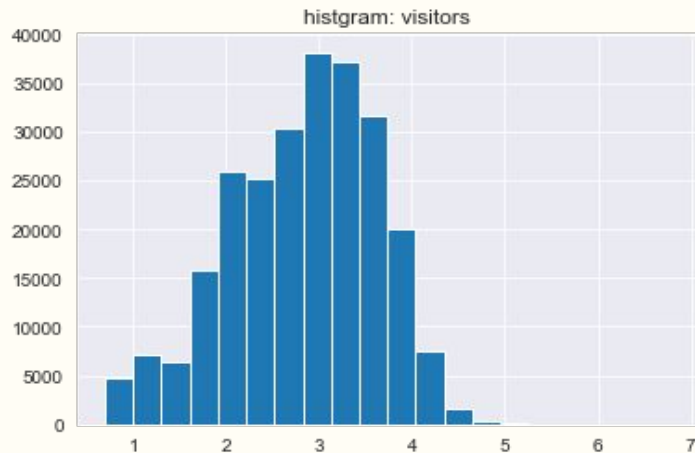


# 目的変数の分布

- ・来客数平均は20人程度だが、100人を超える来客もかなり多い
- ・対数変換 $\log(y+1)$ を行うと、右図のような綺麗な分布に

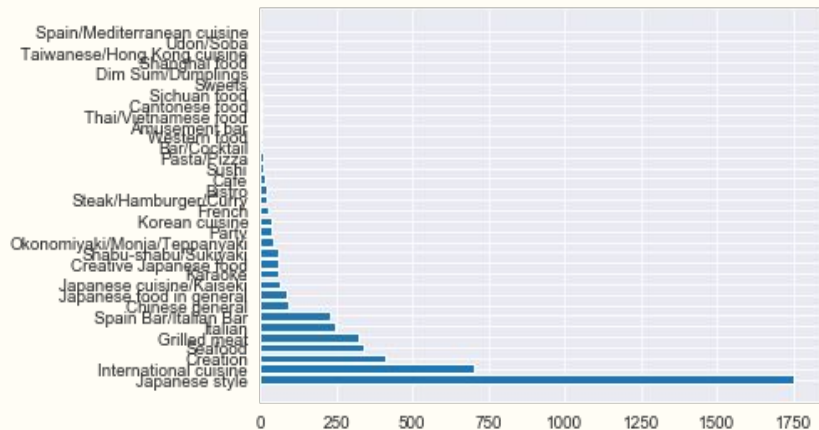
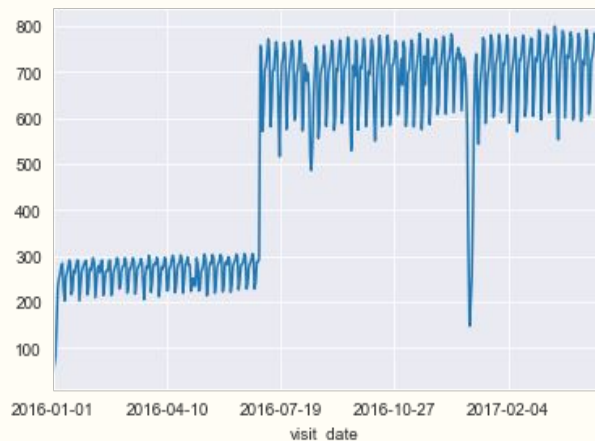


対数変換



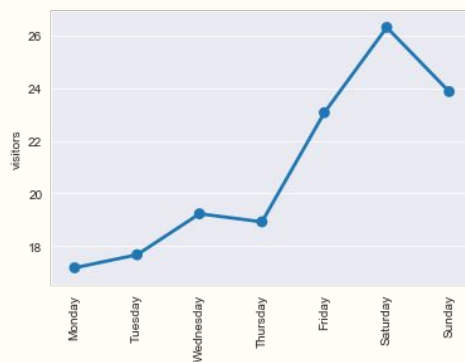
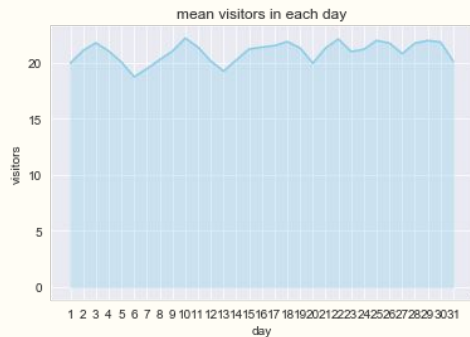
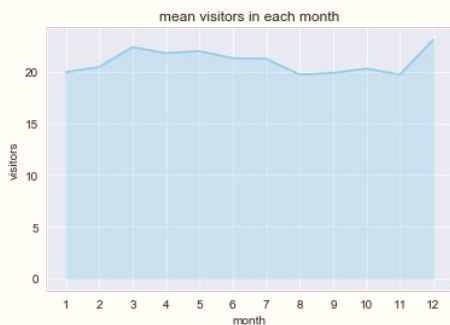
# レストランの情報

- ・日付ごとのレストランの数が、2016年6月ごろに激増している
- ・hpgはairよりも詳細なジャンルを記述しており、出現頻度にかかなりの差がある



# 日付情報

- ・3月や12月の客数が多い(忘年会や送別会など?)
- ・月のはじめよりも, 月の後半の方が客数が多い
- ・曜日ごとでは金土日がやはり客数が増える
- ・平日でも特に火曜日が休みの場合に客数が増加する



# データ前処理と特徴量エンジニアリング

---

# 推薦書籍

---



- ・データ操作のだいたい全てを網羅した本
- ・テーブルのデータ操作，データ型ごとの処理の仕方が記載されている
- ・実践しながら学んでいけばいいが，体系的にまとまっているので目を通してて 損はない



- ・特徴量エンジニアリングの基本的な手法が網羅されている本
- ・テキストデータに関する記載もある
- ・実務やKaggleに取り組んでいくとこの本の言ってる意味がわかるはず

# 特徴量エンジニアリング

---

- ・緯度経度が平均からどれだけ離れているか
- ・日付特徴量をsin, cosでエンコーディング
- ・祝日情報と土日の休日情報, 翌日が休みか否か, 前日が休みか否か
- ・hpgとairのジャンルを結合した, さらに詳細なジャンル
- ・各レストランの直近n日の平均来客数(移動平均)
- ・カテゴリカル変数はCountEncodingとLabelEncoding

# モデリング

---

# 機械学習モデル

---

- ・使用したモデル: CatBoost
- ・Kaggleでよく使用されるモデルは, Gradient Boosting系のモデル
  - ・LightGBM(高精度で高速なので, マスト)
  - ・XGBoost(LightGBMの方が高速で高精度なので, 少し廃れた印象)
  - ・CatBoost(GBDT系の最新モデルで, やや遅いがかなり精度が出る)
- ・LightGBMで試行錯誤を繰り返し, 最終CatBoostやNNなどのアンサンブルを試みるのが一般的



# Gradient Boosting Decision Tree

---

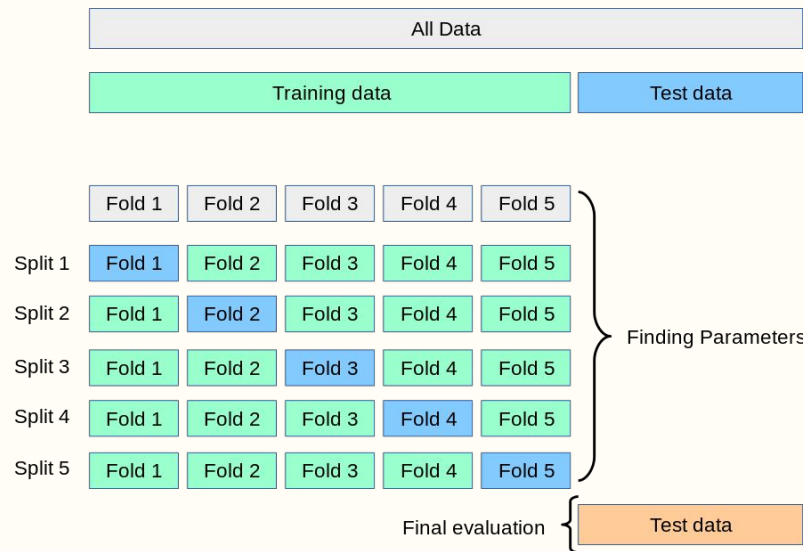
- ・決定木などの弱学習器をいくつも繋いで、ひとつの予測を行うモデル
- ・勾配を利用して学習を進めるのでGradient
- ・テーブルデータではこの派生モデルを使っておけば問題ない

# 精度評価

---

# K-分割交差検証 (K-Fold Cross Validation)

- ・精度評価のために, K-分割交差検証を用いた
- ・学習用データをK個のグループに分割し,  
1個を検証用, K-1個を学習用データとして, 全ての分割で精度をはかり,  
モデルの妥当性を検証する手法

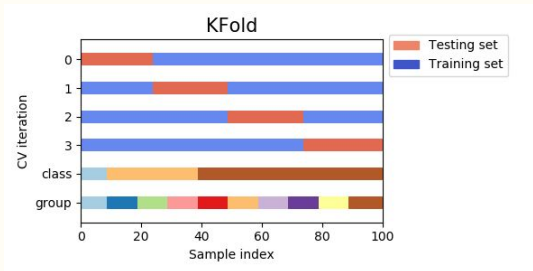


[https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html)

# scikit-learnにおける交差検証

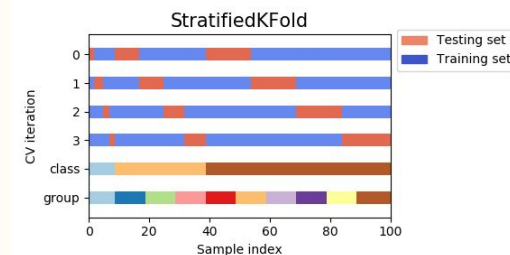
## KFold

- ランダムにK個に分割



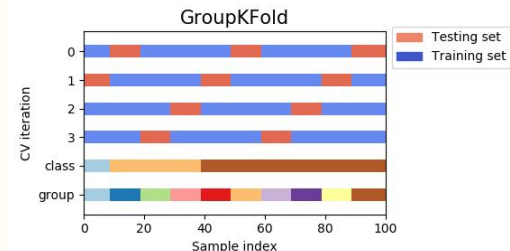
## StratifiedKFold

- K個のクラスの割合がそれぞれ等しくなるように分割



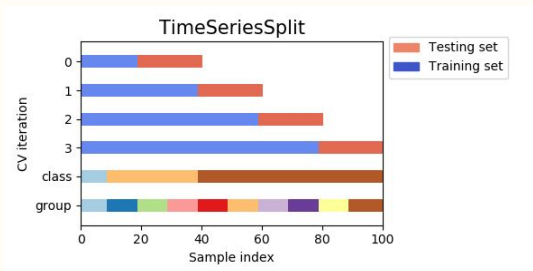
## GroupKFold

- idなど、グループが同じものを同じFoldに入れる分割



## TimeSeriesSplit

- 時系列ごとに分割



[https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_cv\\_indices.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html)

# 精度評価する際の気持ち

---

未知のテストデータに対して、なるべく適合できるように評価したい

- ・まずは信頼できる交差検証(cross validation)を作る
  - ・Adversarial Validationなど、データセットの分割を評価できる手法もある
- ・変更を加えたときに、少ないFoldにスコアがoverfittingしていないか随時      チェックする(経験による職人芸みたいな節がある)

# GroupKFold

---

交差検証にGroupKFoldを用いた

- ・そもそもデータ自体が作為的にレストランを限定して作られているっぽい
- ・本来はこのモデルで全てのレストランにおいて適用できて欲しい
- ・特徴量重要度(Feature Importance)を計算した結果, `air_store_id`がかなり 高く, 店ごとに分割した方がoverfittingを防げると考えたから

おまけ

---

# お役立ちリンク集

---

- [Profiling Top Kagglers: Bestfitting, Currently #1 in the World](#)
- [Feature Engineering](#)
- [Winning solutions of kaggle competitions](#)
- [Kaggle Ensembling Guide MLWave](#)