SIGNATE
ソニーグループ合同データ分析コンペティション（for Recruiting）

3rd Place Solution

自己紹介

- 名前：yayaya
- 関西学院大学大学院　M2
- 興味：データベース×機械学習(NLP)
- MLコンペが最近の趣味（Kaggle Amexに参戦中）

twitter

今回の結果

■ 実は5位で上位の失格or辞退で繰り上げ3位（賞品が実用的なヘッドフォンになって少し嬉しい）

Private LBの順位

5	yayaya	20.0694954	20.0539604	89	2022-06-02 18:27:05

今回のコンペで難しいと感じた所

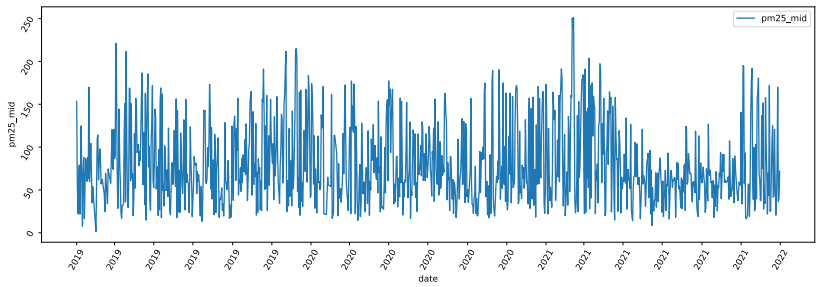
■ 時系列データなのに、過去や未来の特徴量があまり効かない

■ 配布データの特徴量の種類が少ない（大気物質濃度+気象情報は実質9種）

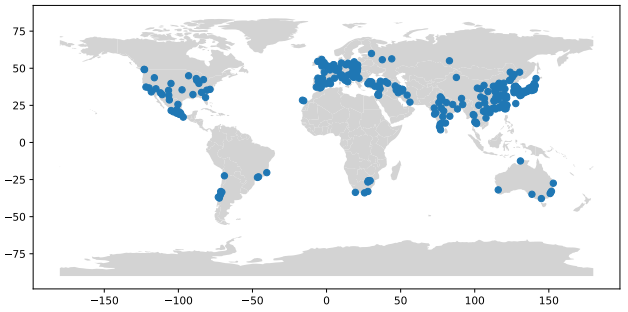
以上を踏まえて今回上位に食い込むには…

➢ targetであるpm25_midをリークをしないように時間的・空間的にうまく集約することがキモのひとつだったと思います

時間



空間



使用したモデルとCV Strategy

モデルは一貫してLightGBMを使用

- Model：LightGBM（seed averagingの結果を提出）
- Split：GroupKfold(group=City, n_splits=10)
  - StratifiedGroupKfold(label=Country，group=City，n_splits=10)でも良かった
- CV：20.54　Public LB：20.06　Private LB：20.05

作成した特徴量

■ **target以外**
- 配布データそのまま(カテゴリ変数はlabel encoding)
- mid min maxの同一特徴量内での差分(ex. ○○_mid - ○○_min)
- 各特徴量のmidをSavitzky-Golay Filteringで平滑化した特徴量，さらに1次微分と2次微分のlag特徴量
- 各特徴量のzero or not
- **CityとCountry単位での観測地の数と観測回数（個人的推し）**
- co，no2，so2の内どれが最大かを表すカテゴリ変数
- City間の距離

■ **targetの集約**
- 各Countryのpm25_midをdate，month，year単位で各種統計量に集約
- **各Cityからの距離がk近傍内にあるCityのpm25_midを各種統計量で集約**
- **各Cityからの一定距離内にあるCityのpm25_midを各種統計量で集約**

□ **targetの真値とtargetの予測値の差分の絶対値をlightgbmで予測し，その予測値を特徴量に**

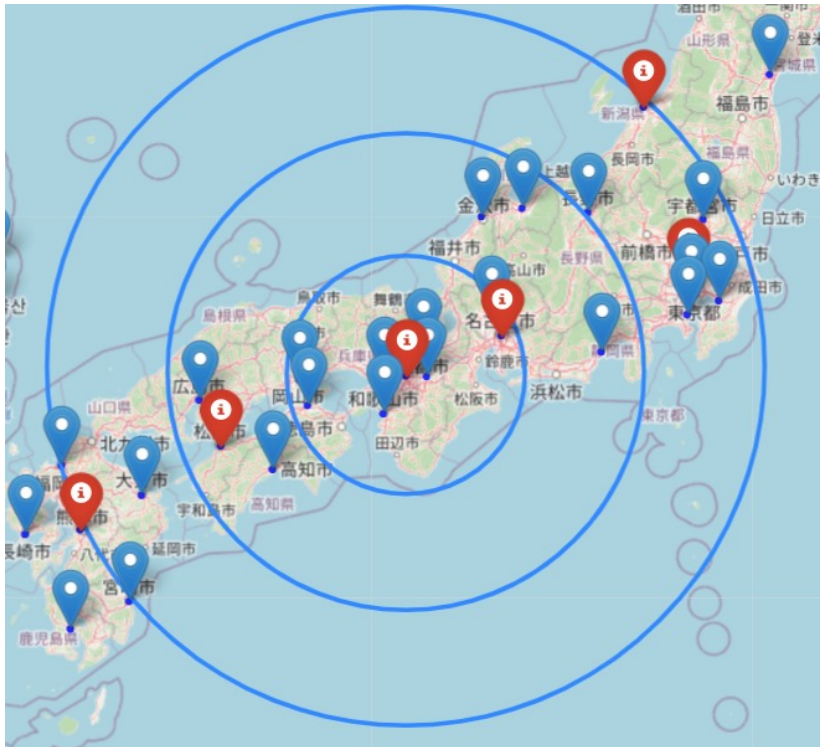
次スライドで一部をもう少し詳しく説明

一定範囲内のtargetを集約

- 空間的に近い都市は似たpm25_mid値であるはず（空間近接性）
  - 2種類の集約方法を採用 -> Private LBが約0.3改善

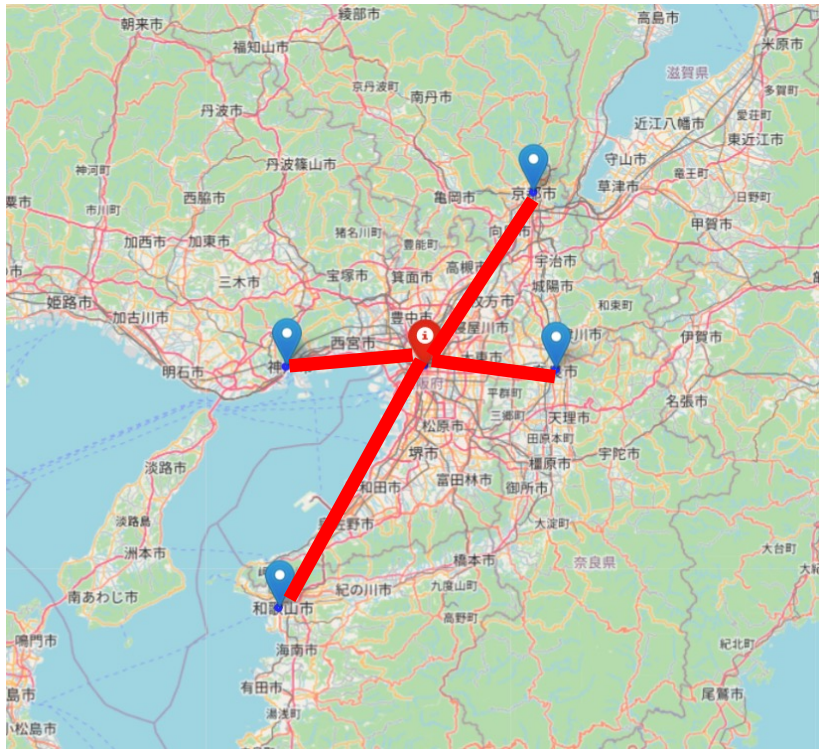
一定距離で集約

（100mile毎に1000mileまで）



近傍で集約

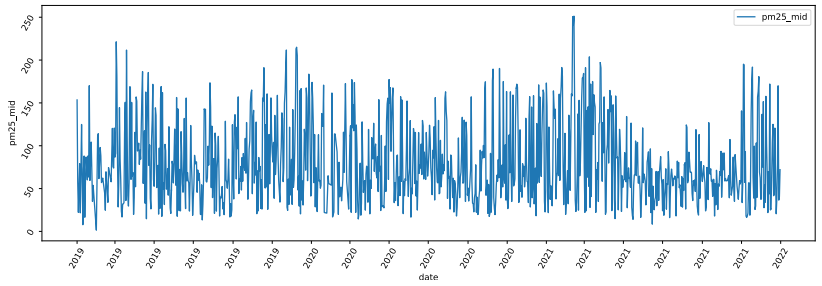
（4近傍から15近傍まで）



targetの差分予測特徴量の作成

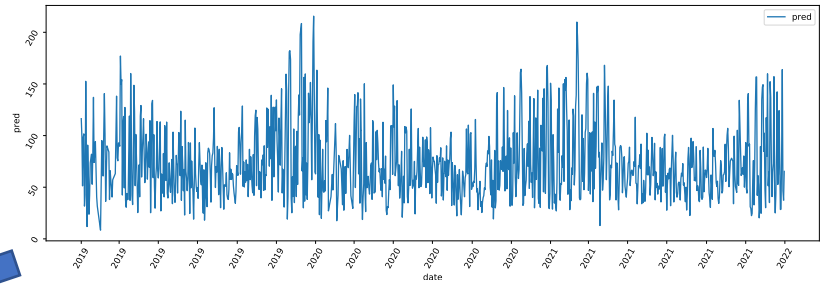
■ 気持ち的にはpm25_midの外れ値度合いを表せる

真のpm25_mid

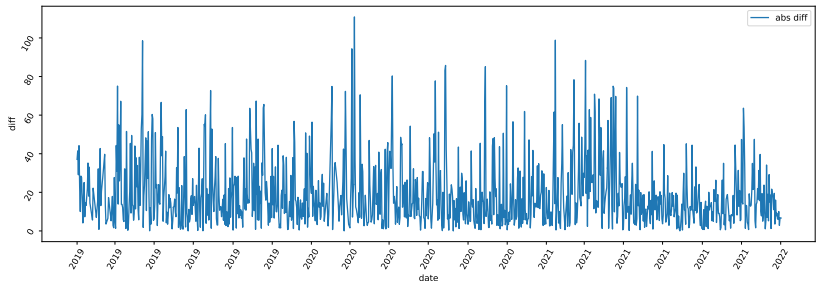


予測

pm25_midの予測値

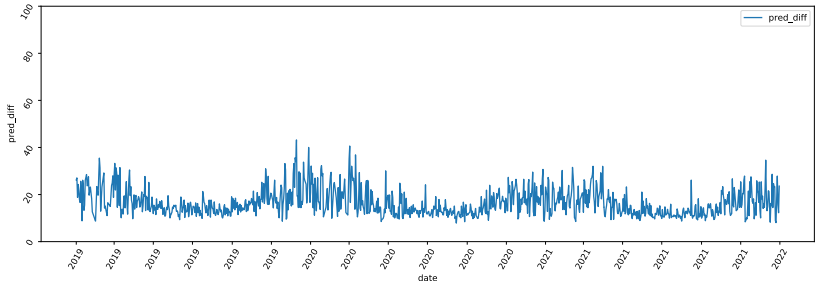


差分の絶対値



予測

差分の絶対値の予測値



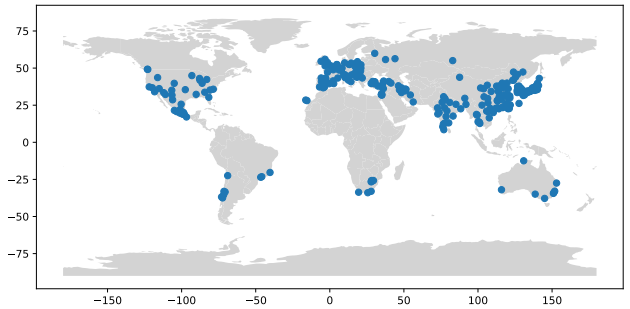
これによりtestにも
差分特徴量が作成できる

Private LB 0.06改善

個人的推し特徴量

City＆Country単位での**観測地の数**と**観測回数**（Private LBが約0.03改善）

観測地マップ



① 国の経済活動が大きいと大気汚染が問題になりがち
② 大気汚染が問題になる国は環境問題を改善するために積極的に観測を行うはず

①はGDPや人口など外部データで考慮できるが、使わずに②も考慮できるのではないかと思い、やってみたら実際少し改善したので個人的に一番テンション上がった

運営さんがデータを加工する段階で人為的に観測地や観測値を削除していた等の場合、仮説が成り立たず、仮説通りに効いたかの真偽は不明…（効いたのでヨシ!!）

Optunaによるハイパラ探索

■ OptunaのLightGBMTunerCVで主要なハイパラを探索

パラメータ	デフォルト値	探索後の値
lambda_l1	0.0	0.0
lambda_l2	0.0	0.0
num_leaves	31	241
feature_fraction	1.0	0.4
bagging_fraction	1.0	0.968
bagging_freq	0	6
min_child_samples	20	20

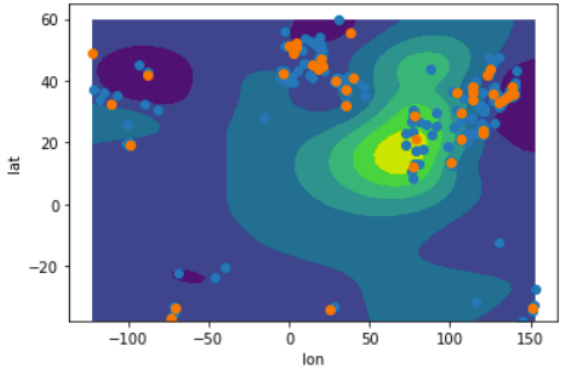
num_leavesを大きくとるとかなり効いた
（RMSEがPrivate LBで0.05改善）

```
gkf = GroupKFold(n_splits=5)
train_data = lgb.Dataset(X_train, y_train)
params = {
    "objective": "rmse",
    "metric":"rmse",
    "verbose":-1,
    "learning_rate": 5e-2,
    "max_depth": 10,
    "random_state": 46,
}

tuner = lgb.LightGBMTunerCV(
    params,
    train_data,
    folds=list(gkf.split(X_train, y_train, groups)),
    categorical_feature=X_train.columns[cat_idxs].tolist(),
    early_stopping_rounds=300,
    verbose_eval=100
)
tuner.run()

print(f'Best score: {tuner.best_score}')
print('Best params:')
print(tuner.best_params)
```

今回改善に寄与しなかった取り組みの一部

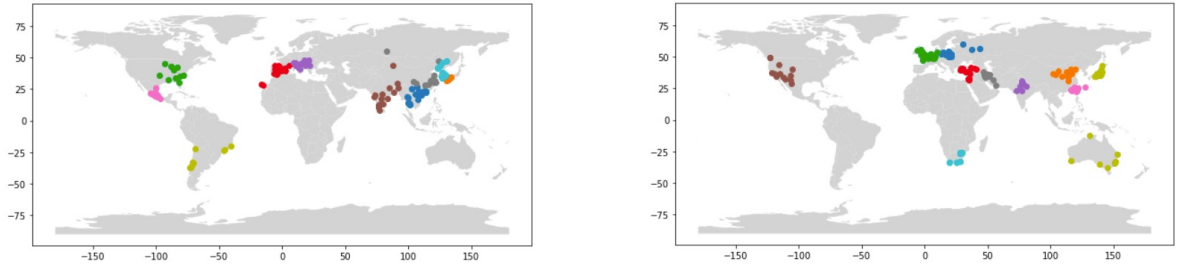
※あくまで自分の環境では効かなかっただけなので無駄という意味ではないです

■ ガウス過程回帰による空間上のtarget分布の推定



■ kmeansで推定したクラスタでtargetを集約（空間類似性）

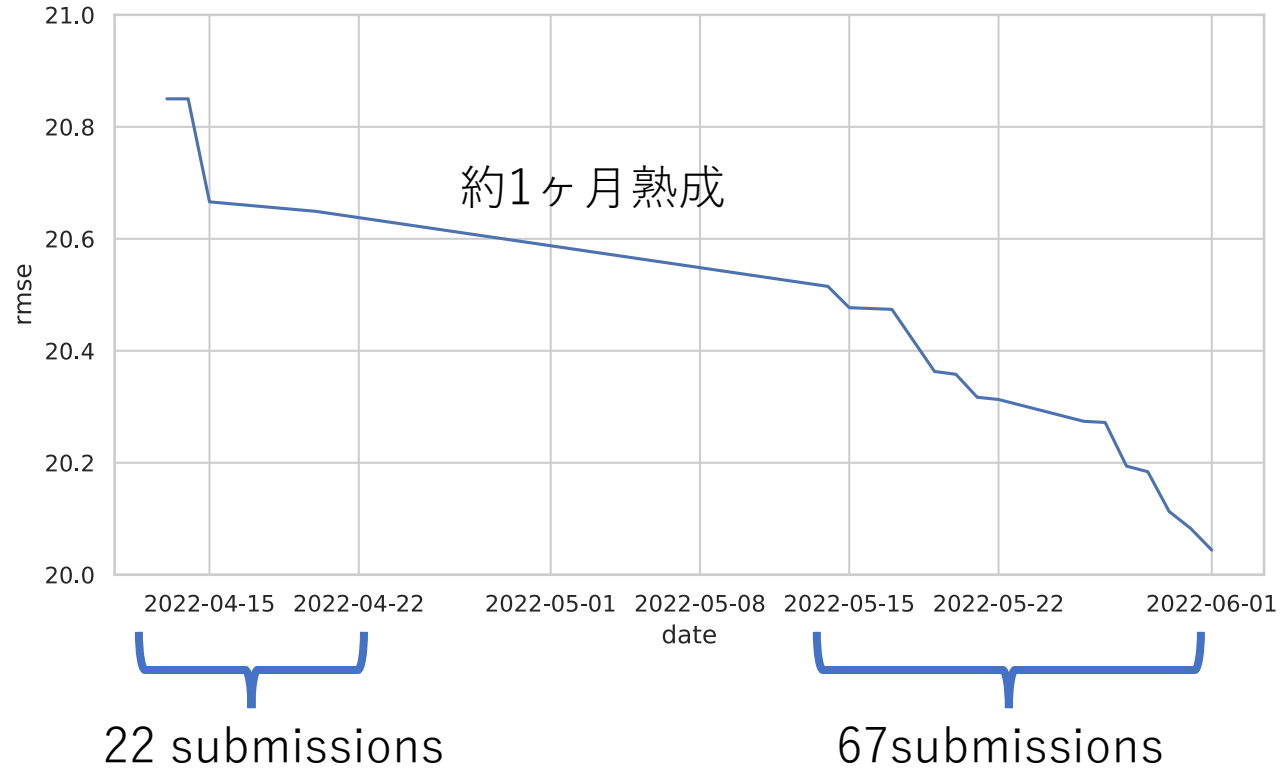
例：クラスタ数20でクラスタリング



■ 一定範囲内のtarget以外の特徴量の集約

Best Private Scoreの推移

コンペ初日4/13 rmse=20.85



6/1 rmse=20.04

22 submissions

67submissions

反省点

■ コンペ終了までTOP1~4位ぐらいまで圧倒的スコアだったので、
軽微な改善を後回しにしていたのが今回の敗因

- Stacking等のモデルアンサンブル
- 外れ値除去等の丁寧な前処理

**最後まで何が起こるか分からないのでやれること些細なことでも
時間があるならやっておくべきという教訓を得た**

最後に

競ってくれた参加者の方々と、面白いコンペの開催と
運営をしてくださったSignateの方々に感謝いたします。

ありがとうございました！！！