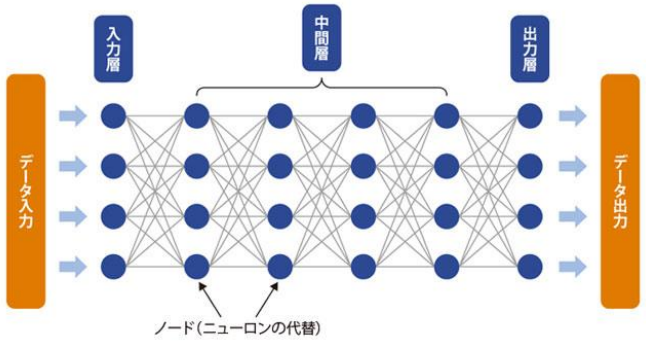
データサイエンス集中講義中級編
～小規模な分子データセットで高度な機械学習モデルを使う方法～

Masaki Open Lab

Agenda

- DXに取り組む材料開発者の悩み：ディープラーニングという“御伽話”
- なぜディープラーニングは“御伽話”と化してしまうのか
- 本日の目標
- 実習
  - Data Augmentation
  - 分子材料をAugmentする方法～分子力学～
  - 実習①：Conformerと3D descriptorの多数生成によるData Augment
  - 実習②：LightGBMをAugmented molecular dataに適用してみる
- まとめと考察

DXに取り組む材料開発者の悩み：ディープラーニングという“御伽話”①



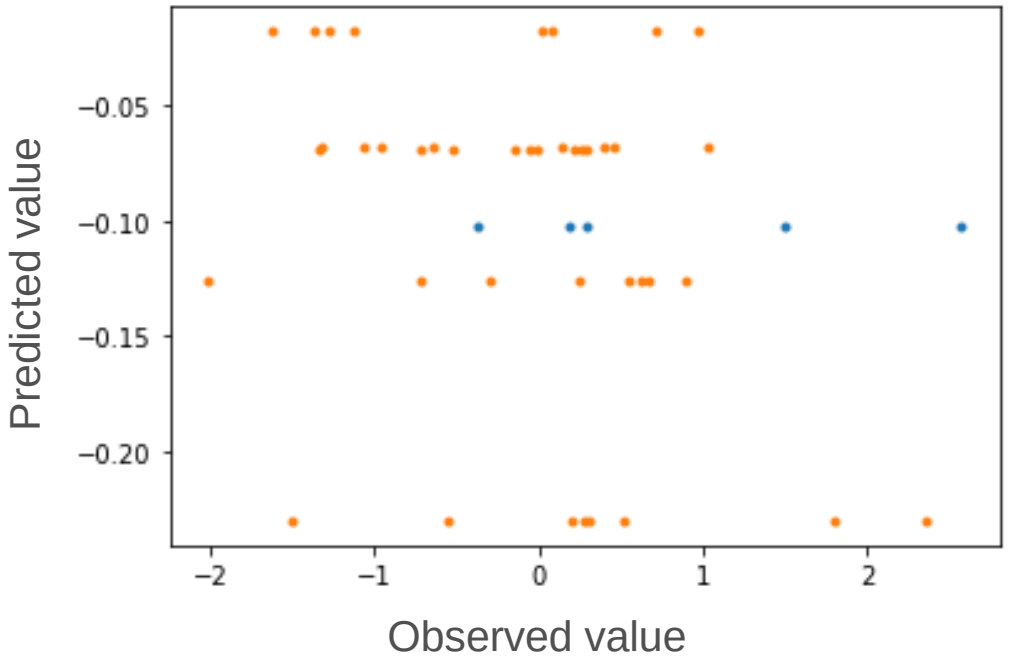
DX

ディープラーニング！ビッグデータ！IoT！
これからは若者の時代だからして、
そういうのはきみ、やっといてくれ
研究もディープラーニングだよ、これからは。

う、うん？　んん？　むむ？

・わが社も材料開発（分子材料設計）に機械学習を適用したい
・そうすれば複雑な事象も、よりいい感じに解析できる＝競争力を稼げる！

なぜディープラーニングは“御伽話”と化してしまうのか

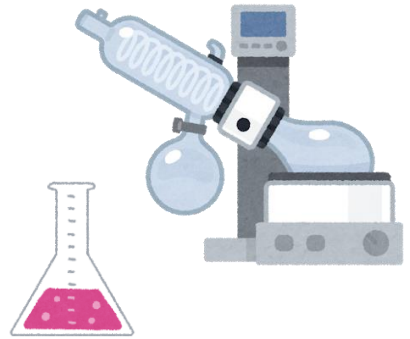


←LightGBMで50点くらいのデータセットを学習させた例
（ポリマのTgが目的変数、説明変数はmordred descriptor）
●交差検証データ　●テストデータ…いやどっちでもいいわそんなん
高度な機械学習モデルってパラメータも膨大なので、
数十点のデータだと正直不安定すぎて使い物にならないことが多い

- 現実問題、材料開発の現場に散在するのは“スモール（< 50点）データ”
- スモールデータの場合、ディープラーニング、GBMなどの高度な機械学習は解析の道具として一般に「不適切」

なぜディープラーニングは“御伽話”と化してしまうのか

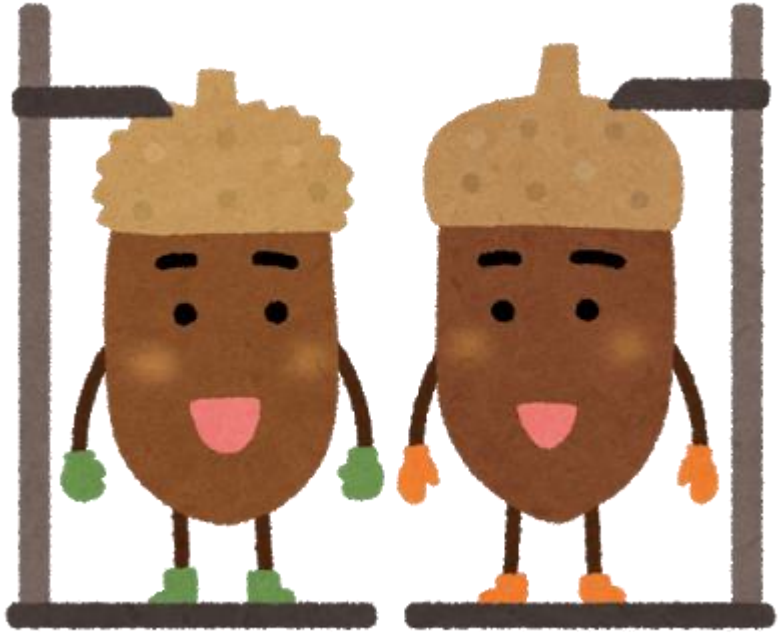
よっしゃ、実験やって合成反応データ増やそうっと。
1日に1件できるから（だいぶ楽観的）、
10000件のデータセットを作るには
大体50年かければいい。よくない。



- 材料開発においてデータ数を増やすには、実験 or 計算科学による代理物性の算出という手法がある
- いずれの手段も、一定の（ものにより相当の）工数を要する
- 現実には、高度な機械学習を実務検討（PoC）する機会は少ない※

※（材料系メーカ管理職向け）もし軽々しく使っている検討見かけたら、ぬか喜びになる前によくよく疑ってください…
うちにそんなビッグデータあったっけ？　過学習じゃないかな大丈夫？　ってジャブ打って部下の反応をみましょう（こんなもんで十分）
回答に納得がいかないようならいったん止めるのが無難です（もちろん納得いくならOK）

なぜディープラーニングは“御伽話”と化してしまうのか



弊社ディープできないんすよー。
御社もそうっすかー。
皆できっこないっすよねーうんうん。

- 高度じゃない機械学習は誰でも使える（例：PLS）ので、
  抜きんでた結果が得られない or 現象を十分に解明できない
- データセットが小さい基礎検討や、ニッチな研究テーマが中々進まない、
  競争に負け、他社に出し抜かれる（ことがありうる）

本日の目標

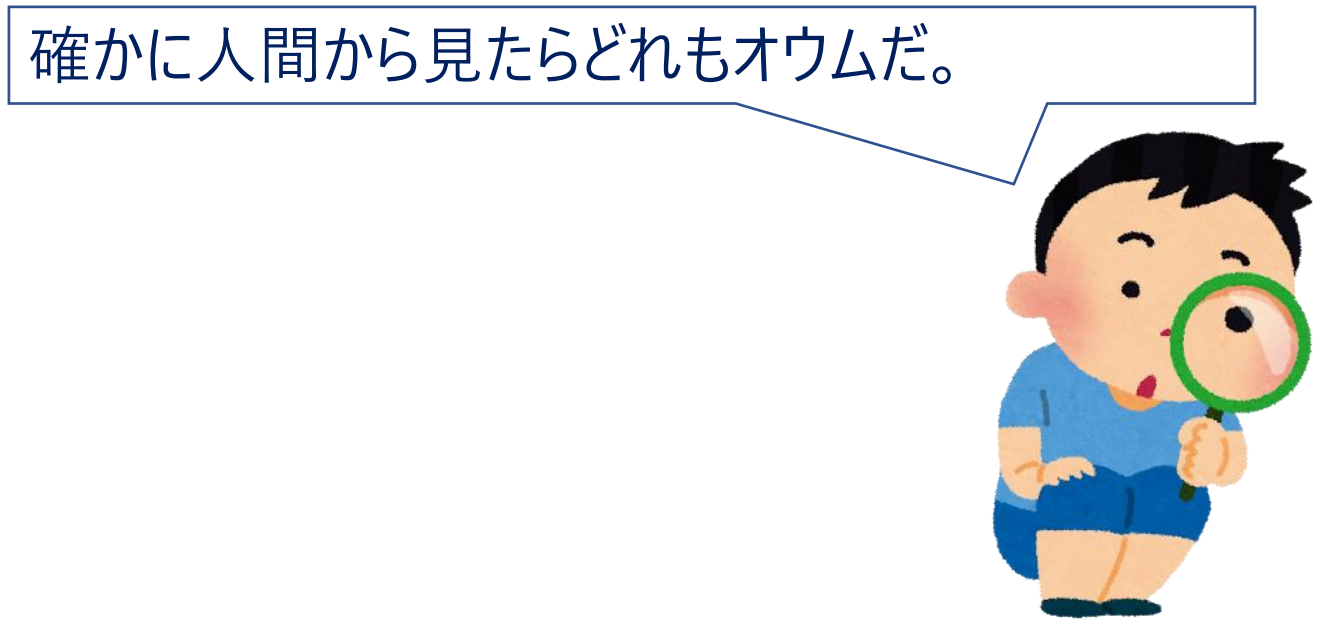
分子材料のスモールデータ解析において
高度な機械学習※1を検討可能※2とする

※1計算環境色々だと思うので、ディープより軽くて成績もよいと評判のLightGBMで示します
Rでディープラーニングモデルも作ったんですが、GPU環境ないと実習にならないじゃないですかやだー
※2あくまで検討できるようになるだけであって、何でもかんでもこれで解けというものではない

実習の準備～Data Augmentという考え方～

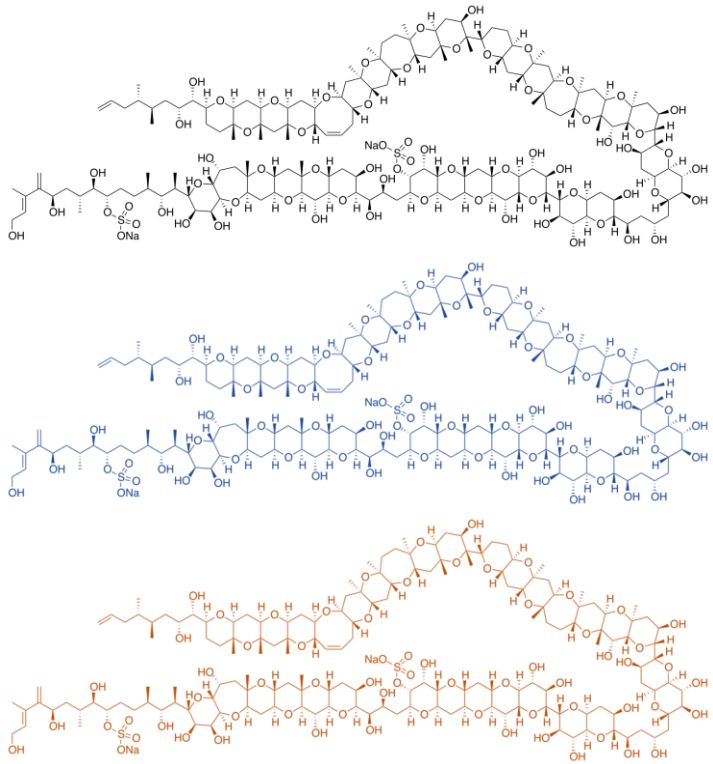
Data Augmentationとは

https://arxiv.org/pdf/1809.06839v1.pdf



・オウムの顔が映った画像を例に挙げる
・画像を白黒化、色調変更、変形…と、元画像から大量の「目的変数（オウム）-説明変数（画像）」というデータをつくる
・この手法を使うと、100枚くらいの元画像でもディープラーニング出来る

分子構造をAugmentするには？



画像と一緒で色をかえてみました！
…それはそれで面白い
（けどもっと簡単な方法がある）

- Data Augmentationを分子構造に適用するには、
  一つの分子から大量の説明変数群を得る方法が必要
- どうやって分子記述子に「揺らぎ」を与えるか？

着想のカギ：分子力学ボーイ（前回の動画）

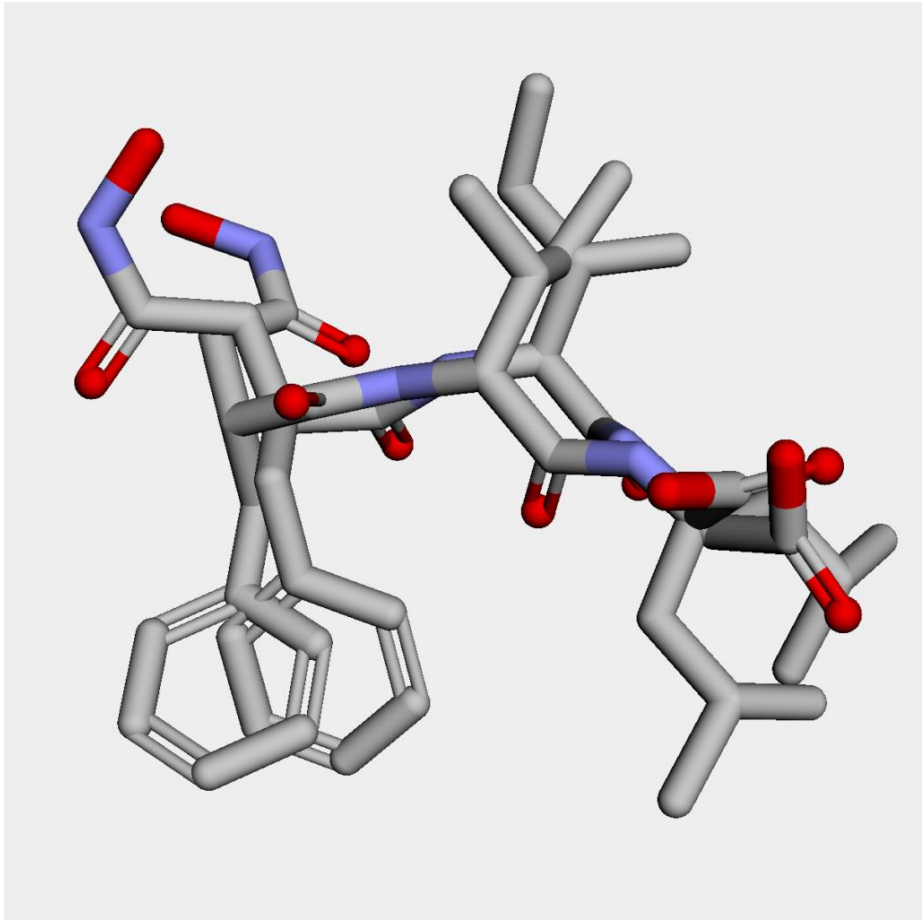
https://future-chem.com/rdkit-conformer/

In [28]:
```
#最終的な記述子の計算結果（3Dだけで6000以上あります）
desc_dist.head()
```

Out[28]:

	minDPSA1	25%DPSA1	50%DPSA1	75%DPSA1	maxDPSA1	minDPSA1_DGmmff	25%DPSA1_DGm
0	18.724974	20.218300	21.214784	21.890085	23.828661	20.804865	21.897085
1	88.497927	91.945827	93.530543	95.280608	98.153119	91.292615	92.499154
2	-64.624902	-61.960585	-61.224688	-60.366565	-58.461659	-60.648274	-59.911854
3	-24.631658	-23.141817	9.240122	10.204323	10.780084	-25.888754	-25.083464
4	18.814502	20.510317	21.619443	22.827008	24.771458	30.893188	32.378082

5 rows × 6390 columns



・分子力学により得られる3D構造は、原理的に分布を持つ
・分子力学ボーイにて、分布の統計値を含めた記述子とする方法を紹介
→その分布を使えばData Augmentationと同様のことが出来るのでは？

実習①：Conformerと3D descriptorの多数生成によるData Augment

実習①で出来るようになること

- SMILESから複数の力場を使って、3D構造を繰り返し生成する
- 出来た3D構造から、3D記述子を繰り返し生成、目的変数と紐づける

```
In [27]: df_summary_add = pd.concat([dfaugument["Tg"],dfaugument["SMILES"], dfaugument["NAME"], df_summary.drop(['Tg',"SMILES", "NAME"], axis = 1)]
         df_summary_add
         #df_summary_add.to_csv('DataTg_new_3d_augument.csv', index = False)
```

Out[27]:

	Tg	SMILES	NAME	DPSA1	DPSA1_DGmmff	DPSA1_DGuff	DPSA1_ETDG	DPSA1_ETKDG	DPSA1_KDG	DPSA2	...	W
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	60.037749	65.869404	79.163580	60.876532	81.176795	60.924559	409.980429	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	59.710654	61.902463	60.187786	74.160633	78.400091	59.676762	408.272845	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	63.367274	61.554951	63.194006	64.282592	42.071541	91.752191	409.456140	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	62.583285	67.207772	63.385476	83.882075	60.109821	56.720717	409.145870	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	54.831154	68.459728	62.128715	67.109344	81.765365	68.460925	410.704498	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	82.158040	61.817116	59.999777	58.639320	79.204970	51.680038	400.111916	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	81.367478	63.420053	76.094149	59.513060	57.635671	61.893844	397.273826	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	78.634498	62.611159	58.542327	68.615010	63.143502	60.163435	399.590690	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	81.944847	63.270928	36.527172	83.821887	62.359132	90.700038	398.470041	...	
0	279.0	C=CC(=O)OCc1ccccc1	Poly(benzyl acrylate)	69.131535	68.542312	78.234014	67.960711	59.825627	61.336116	411.958747	...	
1	383.0	C=CC(=O)Oc2ccc(c1ccccc1)cc2	Poly(4-biphenyl acrylate)	32.181737	53.001859	37.258761	51.107246	35.611275	33.529972	552.416872	...	
1	383.0	C=CC(=O)Oc2ccc(c1ccccc1)cc2	Poly(4-biphenyl acrylate)	32.668591	52.369964	39.581755	59.707418	36.751147	37.175581	553.935350	...	

←出来上がったAugmented dataの例
一つの分子に対して複数のdataが出力される

データセットと計算環境

https://github.com/Saket-Uoft/Artificial-Intelligence-for-Predicting-Materials-Properties

- Saket et al., “Deep Learning Based Approach to Predict Glass Transition Temperature of Polymers”よりSMILESとガラス転移点のデータセットを引用
- 3D構造の生成はRDKitを使用、力場は6種類
- 繰り返し回数は100回（実習はもう少し軽めにやります）

では、さっそくやってみましょう

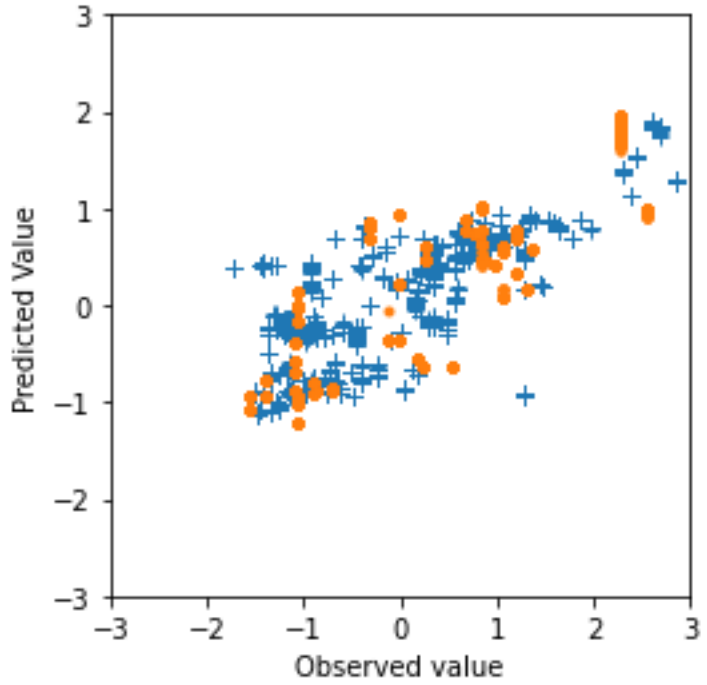
実習①：まとめ

- 分子力学を使うことで、少数のSMILESキーから疑似的にビッグデータを生成すること＝Data Augmentが出来ることを理解した
- 本データを用いれば、高度な機械学習モデルであっても過学習を防ぎ安定した検討が可能であると期待された
- 以降、このAugmented dataを用いて高度な機械学習モデルの検討を行う

実習②：LightGBMを
Augmented molecular dataに適用してみる

実習①で出来るようになること

- LightGBMの原理をおおざっぱに理解する
- LightGBMによりTgの学習を行う（Python）
- ハイパーパラメータを最適化し、テストデータに対する予測精度をみる



←LightGBMによる学習結果の例（●交差検証データ　＋テストデータ）
こんな少数データでも、過学習せずモデル構築できます！

データセットと推奨計算環境

- Saket et al., “Deep Learning Based Approach to Predict Glass Transition Temperature of Polymers”よりSMILESとガラス転移点のデータセットを引用
- データ数は351件、論文中においては300点で学習、51点でテスト
- 今回は約70点で学習、280点でテスト（過学習しやすい条件）
- 説明変数は100回Augmentした3D記述子（～230種）と2D記述子（～240種）
- 推奨環境：4コア8スレッドくらいあれば十分※

※もっと貧弱でもいいですがちょっと時間はかかります

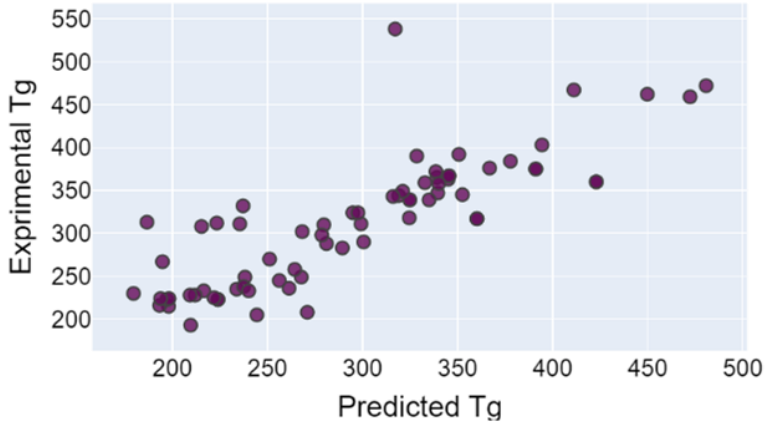
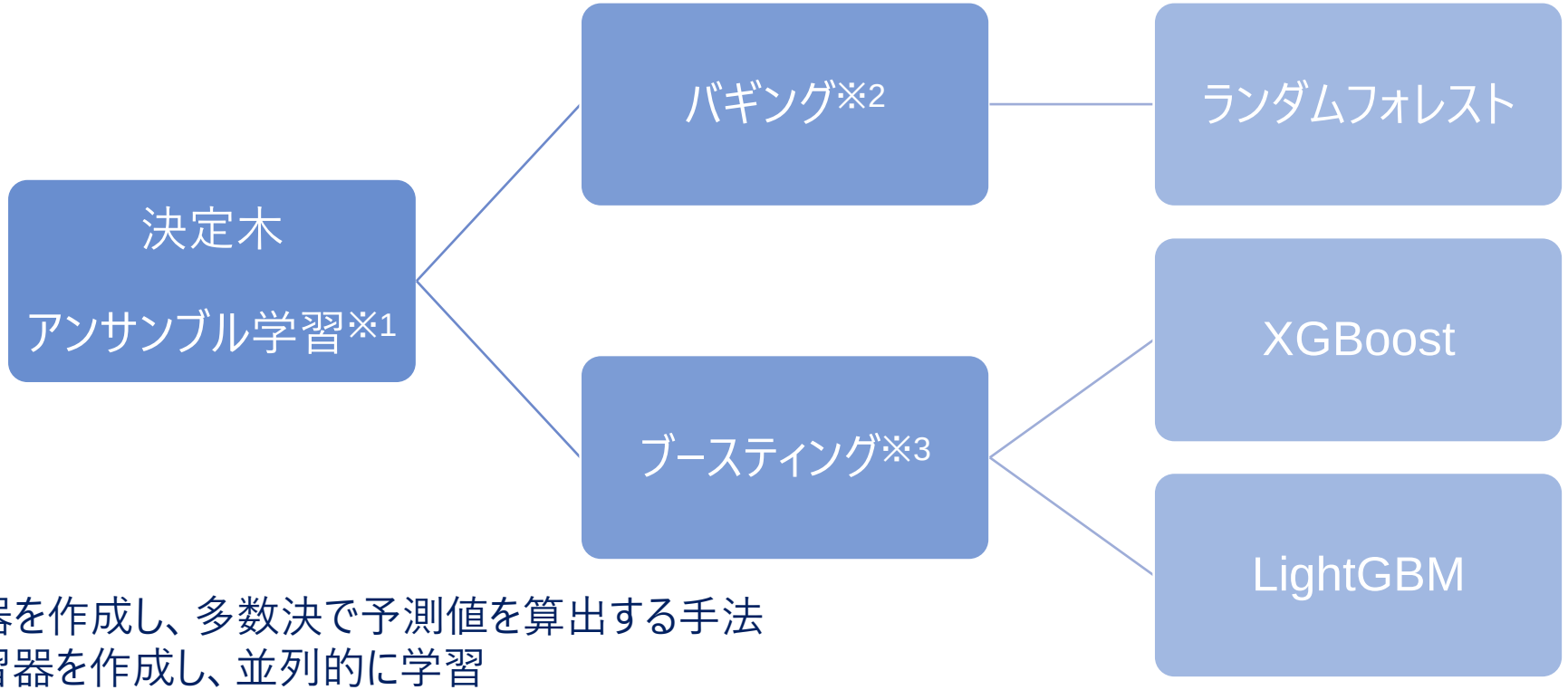


Fig. 5. Real vs Predicted for Test set for CNN

LightGBMの原理



※1：多数の学習器を作成し、多数決で予測値を算出する手法
※2：ランダムに学習器を作成し、並列的に学習
※3：n番目の学習器は、n-1番目の学習器における予測誤差を学習

- 決定木系のアンサンブル学習手法であり、かつブースティングを用いる
- 決定木を層ではなく、葉に準じて成長させる（無駄な分岐を設けない）
- データをヒストグラムにして、大まかな階級で見る（速度優先）

LightGBMの〇と×

- 軽い、速い、複雑系に強い！

※実際Signate出た時も、ディープラーニングを種々検討したあと、アンサンブル学習のためにLightGBM
ちょろっと走らせたら単体のスコアでいきなり新記録が出たので感心するより先に脱力感が出ましたよ私は。速いし。

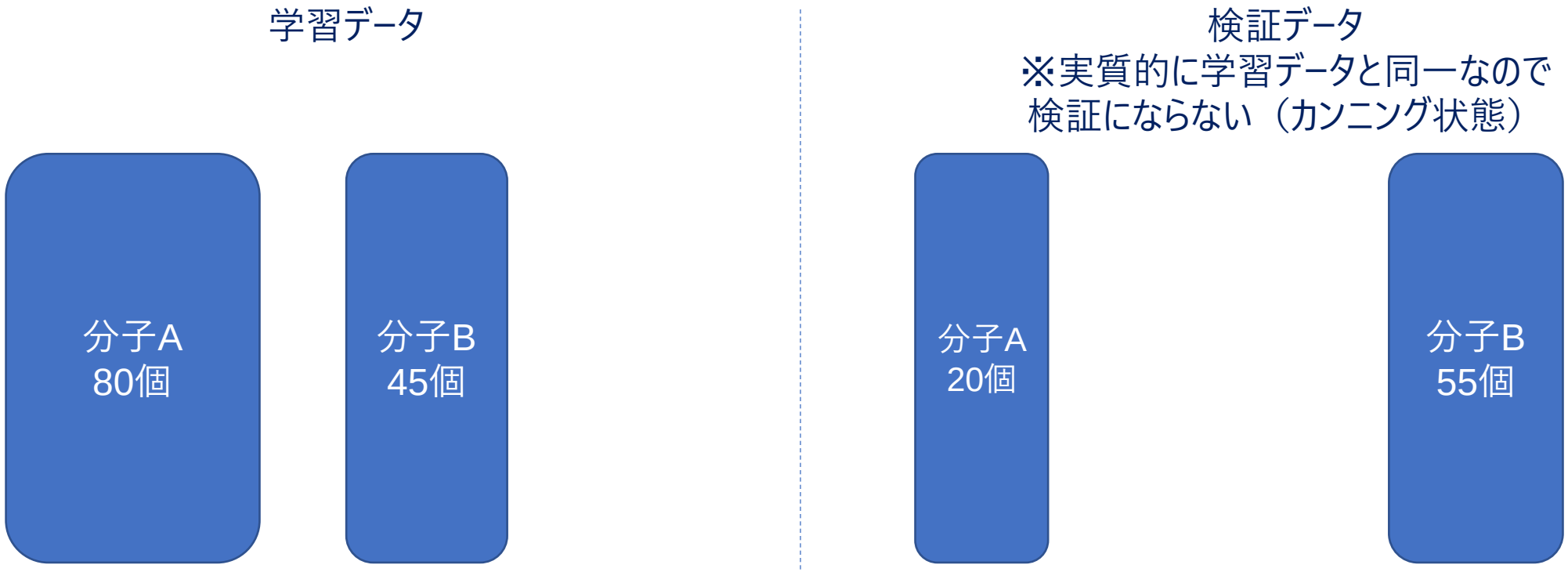
- 過学習しやすい！

ハイパーパラメータの調整

```
{       'num_leaves': int(round(num_le)),
        'objective': 'regression',  #I will check it.
        'max_depth': int(round(max_dep)),
        'learning_rate': lr,
        "boosting_type": "gbdt",
        "subsample_freq": 1,
        "subsample": 0.8,
        "bagging_seed": int(round(bag_seed)),
        "metric": 'mae',
        "verbosity": -1,
        'lambda_l1': 0.8,
        'lambda_l2': 0.2,
        'feature_fraction': 0.6,
        }
```

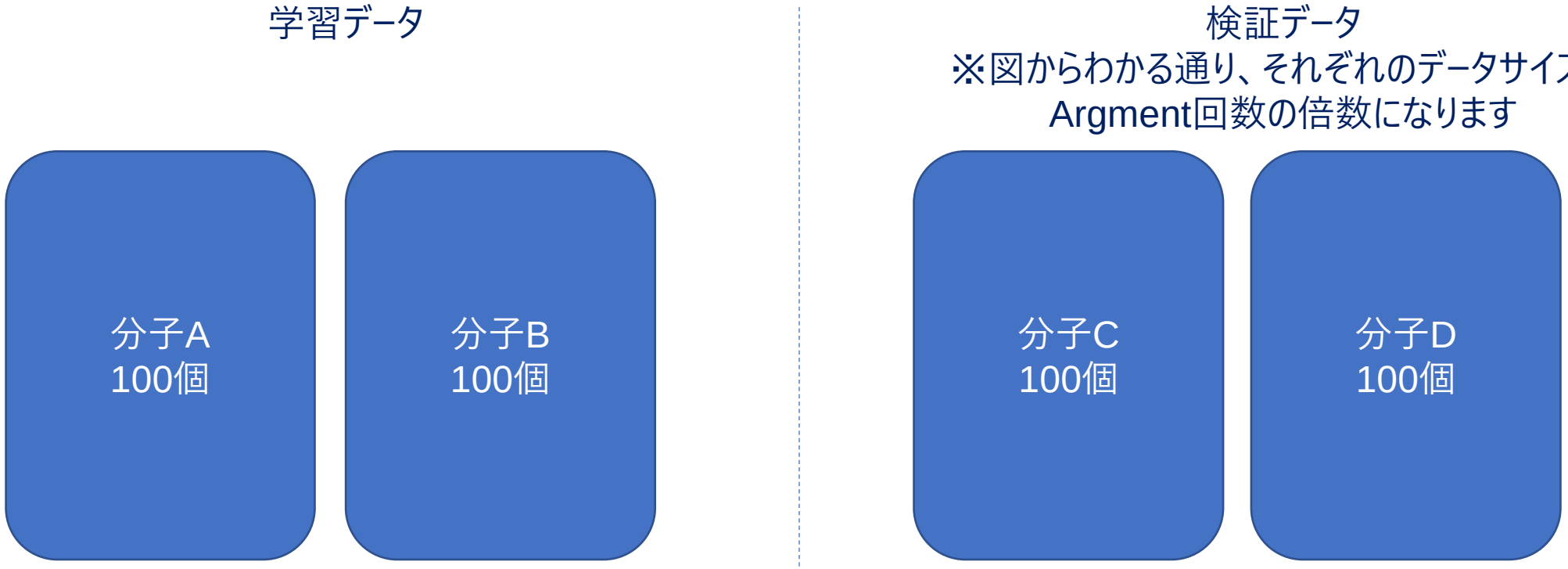
- ハイパーパラメータの調整は、学習データの一部を検証データとし、交差検証（CV）の結果を最適化するのが一般的
- 人力で調整するのは大変なので、普通は各種パッケージを用いる
- 今回はベイズ最適化を利用（Optunaも人気ですが…）

注意点：Augmented Dataの交差検証



- Augmented Dataは通常、目的変数が同値のデータを複数（Argment回数分、今回は100点）含む
- 単にn-fold CVを組むと、学習データと検証データの双方に目的変数が同値のデータが大量に含まれる＝過学習まっしぐら

Group Kfold Cross Validation



・同じグループのデータはひとまとまりとして取り扱う
→学習データと検証データの双方に同じ分子のデータが入ることを防ぐ
（要するにLeakageの防止です）

計算フロー

学習データ/テストデータの読込

モデルと交差検証方法の設定

ベイズ最適化のパラメータ設定
（要するにどれくらい根詰めてやるか）

学習、モデルのパラメータ最適化

最適なモデルの再学習、テストデータによる
予測精度確認

では、さっそくやってみましょう

実習②：まとめ

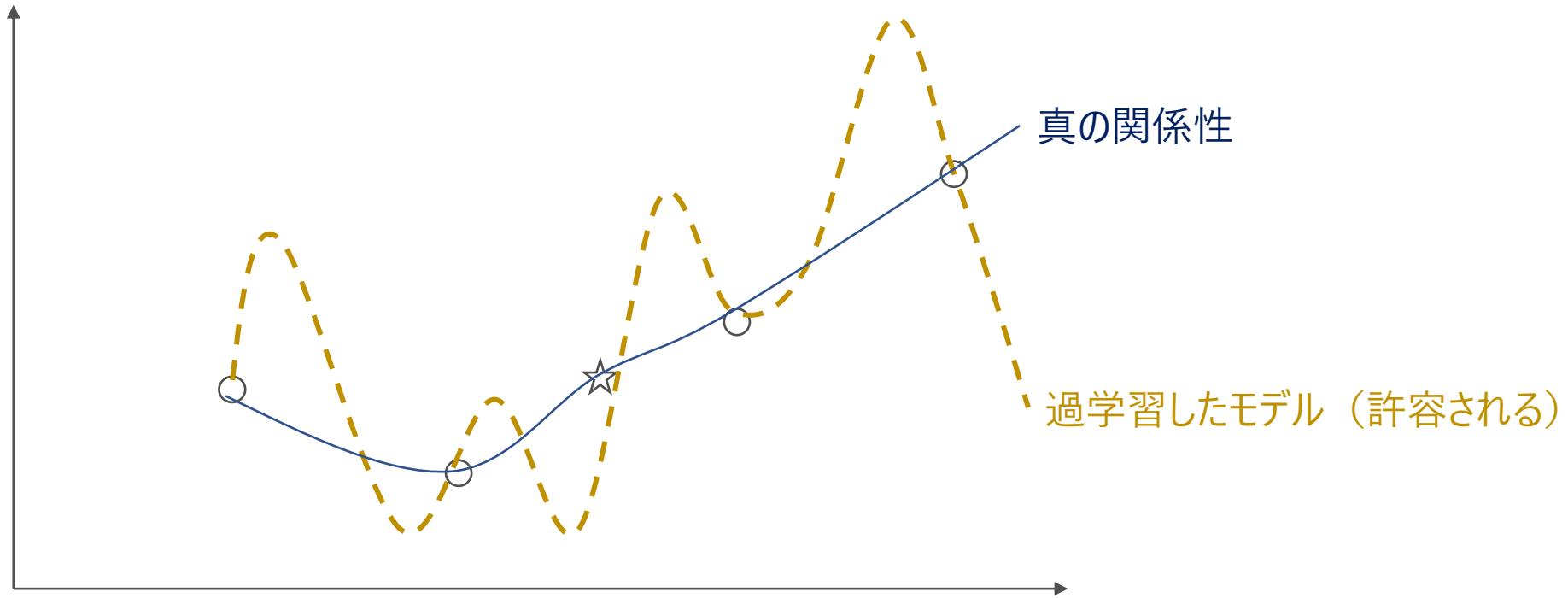
- Kaggleをはじめとする機械学習コンペティションで頻出のLightGBMについて、概要を理解した
- LightGBMを実装し、パラメータの最適化が可能となった
- Augmented Dataの交差検証における注意点を理解して実装した
- 実習①と合わせて、スモールデータに対し安定的にLightGBMを（あくまで一手として）検討することが可能となった

まとめと考察

まとめ

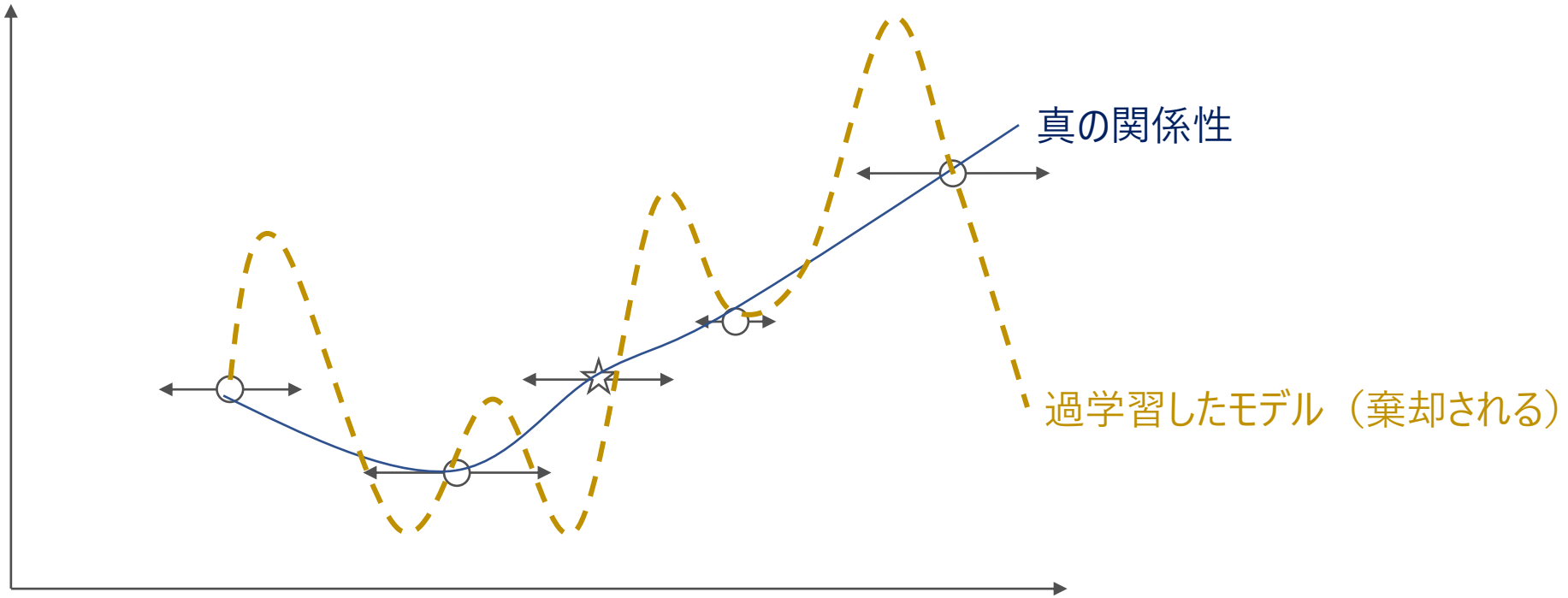
- SMILESキーから生成した3D構造の「分布」を利用して、
  Data augmentを行うことが可能となった
- Data Augmentを行えば、スモールデータにも
  高度な機械学習モデルが検討可能であることを実習により確認した

考察①～Data Augmentって結局なんなん？



- 高度な機械学習モデルは「自由度が高い」
- 交差検証の正答率が高くても、過剰に複雑なモデルが形成されうる
- 特に学習データが少ない＝データ密度が低いほどそうなりやすい

考察①～Data Augmentって結局なんなん？



・Augmented Dataの場合、先ほどの過剰に複雑なモデルは棄却される
（分布の中心にしか合ってないので）

∴ノイズを学習することで、スモールデータでも過学習を防止できる

考察②～なんでもAugment出来るん？

https://jcheminf.biomedcentral.com/articles/10.1186/s13321-020-00420-z

- 工程検査回数の限られたプロセスデータ
  （例：検査は4時間に1回、センサデータは1分に1回得られる）
- 言語（例：翻訳→逆翻訳、類義語への置き換え）
- 画像（例：画像ではなく動画で学習）
- 分子（例：今回提案手法※、一意でないSMILESの利用）

※オリジナル手法だと思って調べたら半年前に論文が出てました…付け焼刃の勉強じゃ世界には勝てないです。

考察③～じゃ全部GBMに投げてしまえ！

- 当たり前ですが、解析をやるときに大事なのは「何を」「どのくらいの精度で」「どの範囲のデータに」「どれくらいの計算負荷で」「どれくらいの可読性で」予測するモデルが欲しい、という要件定義※です

※例えば「決定木系のモデルって原理的に外挿出来ないけどいいの？」って聞かれてちゃんと答えられますか？

- もっとさかのぼれば研究テーマ、開発テーマの目標設定（5W1H）が一番重要です（耳が痛くてたまらねえや）
- 高度な機械学習モデルが選択肢の一つに入るというのはよいことです
- ですがそれはあくまで選択肢であって、目標ではないです
- 目標と違ったところで計算負荷をかけるのは電気代のムダです（CPU使用率は仕事における絶対的KPIではありません…かと言って、スパコン導入したのにだれも使ってないような状況は宜しくないですが）