



<https://x.gd/TOKaP>

5年分のツケを一気に払った話

Takeshi Sugiyama a.k.a. soogie(すーぎー)

お前誰よ？

- 事務系サラリーマン（1988～）
- プログラミングは趣味 & 独学
BASIC（1977～） VBA（1995～） R（2014～） ※C, TurboPascalもちょっと
- Python（2014～）
みんなのPython勉強会(2015～) 何度か登壇、LT
PyConJP2016、2018、2021登壇
Python Boot Camp TA × 5回
Python関連書籍の出版前レビュー × 20冊 +

トーク内容

話すこと

- Python3.6→Python3.11での苦労話
- 3rdパーティパッケージのVer Upに伴う諸問題

話さないこと

- 技術的負債をため込まない上手なやり方



背景

- 現状問題なく動いているシステムをいじりますか？
- PythonやパッケージがVer Upしただけで修正しますか？
- 非エンジニア組織で、他人が書いた現状問題なく動いている業務アプリをいじる勇気ありますか？



たまっていくツケ

Pythonは毎年新しいバージョンがリリースされていく

サードパーティパッケージも日々バージョンが上がっていく

たまに機能追加しようとするネット調べたコードがPython3.6では動かない

たまっていくツケ

Pythonは毎年新しいバージョンがリリースされていく

サードパーティパッケージも日々バージョンが上がっていく

たまに機能追加しようとするネット調べたコードがPython3.6では動かない

仕事はアプリのメンテではなくアプリを使うこと

気にはなるけど、新しい機能を使わなければ困らない

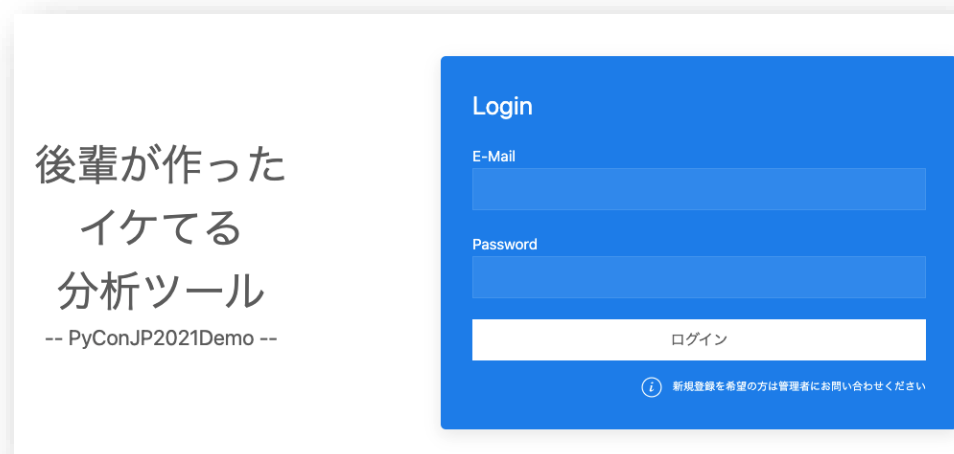
// 何もしない人には過失はない。しかし何もしないことほど大きな過失が人生にあらうとは思えない。 //

下村湖人（教育家・小説家、1873－1930）

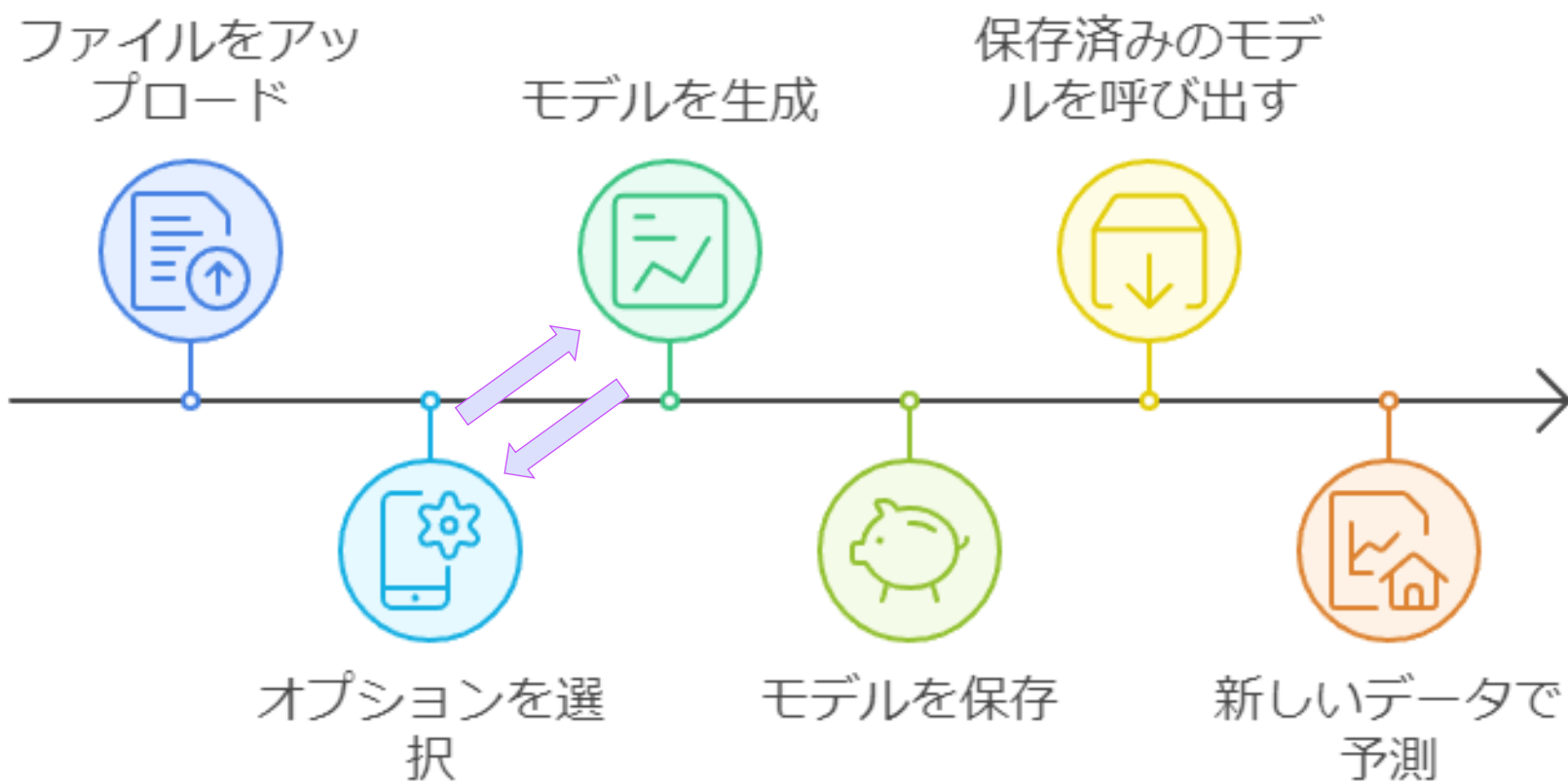


業務系機械学習Webアプリ「X」（仮名）

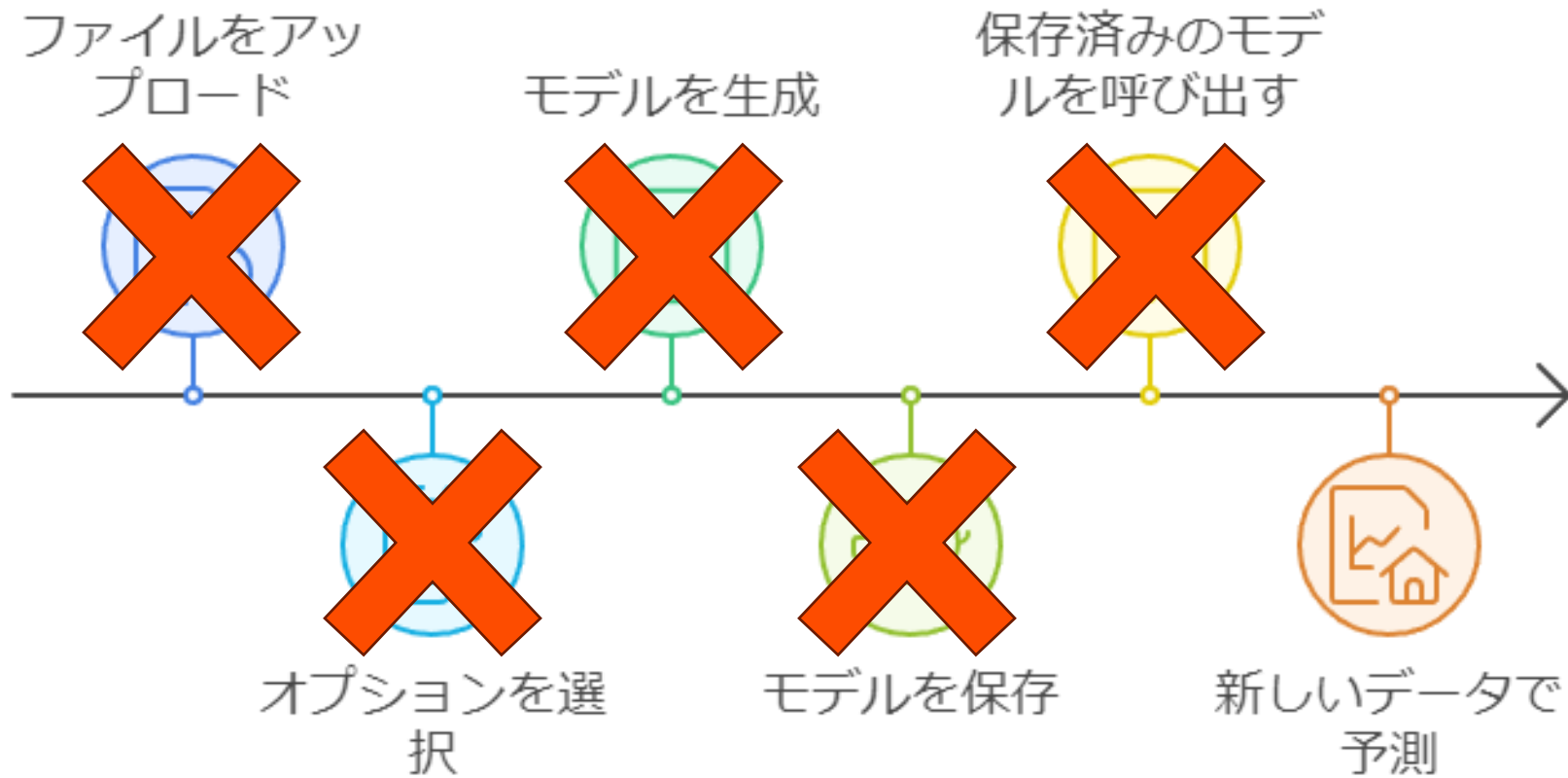
- 5年前に開発され、4年前に引き継ぎ
- 「ドキュメントなし」「テストコードなし」「書いた本人たちは退職・異動」の3重苦
- ドキュメントを作った話はPyConJP2021のアーカイブをご覧ください
- その後テストも書いたし後輩にソースコードレベルでレクチャもしてきた



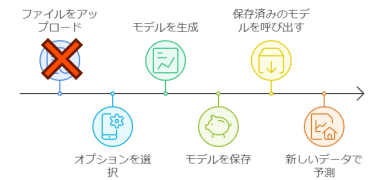
X（仮名）の仕組み



発生した5つの問題



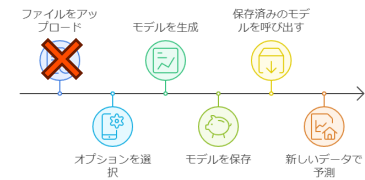
ホントはもっといろいろあったけど今日は割愛



問題1: ファイルアップロードで落ちる

現象：学習用データをアップロードしようすると落ちる

発生箇所：`pandas.DataFrame.to_dict(orient="record")`



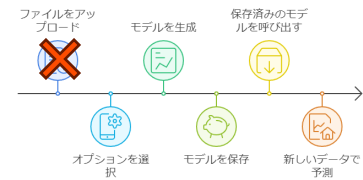
問題1: ファイルアップロードで落ちる

現象: 学習用データをアップロードしようすると落ちる

発生箇所: `pandas.DataFrame.to_dict(orient="record")`

Parameters:

orient : *str* {'dict', 'list', 'series', 'split', 'tight', 'records', 'index'}



問題1: ファイルアップロードで落ちる

現象: 学習用データをアップロードしようすると落ちる

発生箇所: `pandas.DataFrame.to_dict(orient="record")`

ポイント: 正しくは「`records`」。単なるtypoだけどなぜ今まで動いていた？

ソースコードを読む

【pandas1.4】 `pandas.core.frame.py`

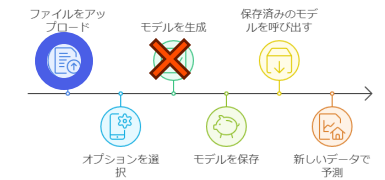
```
elif orient.startswith("r"):
    orient = "records"
```

【 pandas2.2 】

`pandas.core.methods.to_dict.py`

```
orient = orient.lower()
```

対応: typoを修正



問題2:機械学習で落ちる

現象：ロジスティック回帰でLASSOを使うと落ちる（他いろいろ）

発生箇所：`sklearn.linear_model.LogisticRegression(penalty="l1")`

問題2: 機械学習で落ちる

現象: ロジスティック回帰でLASSOを使うと落ちる (他いろいろ)

発生箇所: `sklearn.linear_model.LogisticRegression(penalty="l1")`

Parameters:

penalty : {'l1', 'l2', 'elasticnet', None}, default='l2'

Specify the norm of the penalty:

- `None` : no penalty is added;
- `'l2'` : add a L2 penalty term and it is the default choice;
- `'l1'` : add a L1 penalty term;
- `'elasticnet'` : both L1 and L2 penalty terms are added.

問題2: 機械学習で落ちる

現象: ロジスティック回帰でLASSOを使うと落ちる (他いろいろ)

発生箇所: `sklearn.linear_model.LogisticRegression(penalty="l1")`

Warning

Some penalties may not work with some solvers. See the parameter `solver` below, to know the compatibility between the penalty and solver.

問題2: 機械学習で落ちる

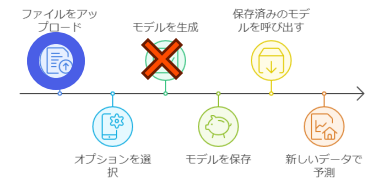
現象: ロジスティック回帰でLASSOを使うと落ちる (他いろいろ)

発生箇所: `sklearn.linear_model.LogisticRegression(penalty="l1")`

Warning

The choice of the algorithm depends on the penalty chosen and on (multinomial) multiclass support:

solver	penalty	multinomial multiclass
'lbfgs'	'l2', None	yes
'liblinear'	'l1', 'l2'	no



問題2: 機械学習で落ちる

現象: ロジスティック回帰でLASSOを使うと落ちる (他いろいろ)

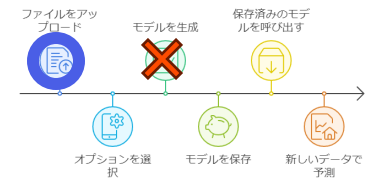
発生箇所: `sklearn.linear_model.LogisticRegression(penalty="l1")`

! Added in version 0.17: Stochastic Average Gradient descent solver.

! Added in version 0.19: SAGA solver.

! Changed in version 0.22: The default solver changed from 'liblinear' to 'lbfgs' in 0.22.

! Added in version 1.2: newton-cholesky solver.



問題2: 機械学習で落ちる

現象: ロジスティック回帰でLASSOを使うと落ちる (他いろいろ)

発生箇所: `sklearn.linear_model.LogisticRegression(penalty="l1")`

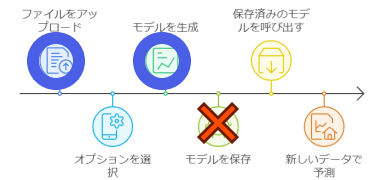
ポイント: LogisticRegressionのsolverパラメータのデフォルト値が"liblinear"から0.22以降はL1ペナルティに対応していない"lbfgs"に変わった

公式ドキュメントを読む

! *Changed in version 0.22:* The default solver changed from 'liblinear' to 'lbfgs' in 0.22.

solver	penalty
'lbfgs'	'l2', None
'liblinear'	'l1', 'l2'

対応: `solver='liblinear'`を明示的に追加



問題3：結果を格納するときに落ちる

現象：機械学習後にMongoDBに結果が保存されず落ちる

発生箇所：`pymongo.collection.save()`

問題3：結果を格納するときに落ちる

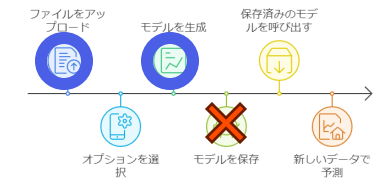
現象：機械学習後にMongoDBに結果が保存されず落ちる

発生箇所：`pymongo.collection.save()`

Current Stable Release

(7.0-series)

- [Release Notes for MongoDB 7.0 \(Stable Release\)](#)



問題3：結果を格納するときに落ちる

現象：機械学習後にMongoDBに結果が保存されず落ちる

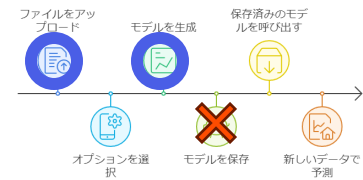
発生箇所：`pymongo.collection.save()`

Previous Stable Releases

- [Release Notes for MongoDB 6.0](#)
- [Release Notes for MongoDB 5.0](#)
- [Release Notes for MongoDB 4.4](#)

EOL'ed Releases

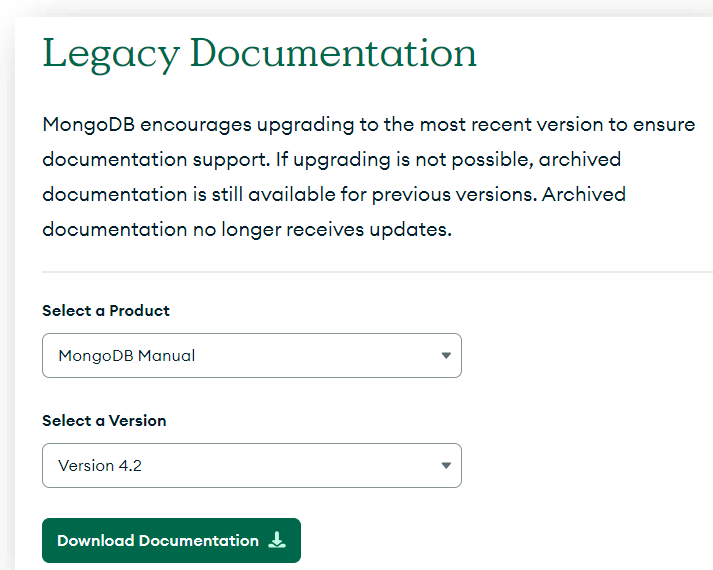
- 4.2 • 2.4
- 4.0 • 2.2



問題3：結果を格納するときに落ちる

現象：機械学習後にMongoDBに結果が保存されず落ちる

発生箇所：`pymongo.collection.save()`



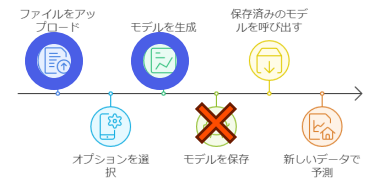
問題3：結果を格納するときに落ちる

現象：機械学習後にMongoDBに結果が保存されず落ちる

発生箇所：`pymongo.collection.save()`

Starting in MongoDB 4.2,

- Operations which replace documents, such as `replaceOne()` or `update()` (when used with a replacement document), will first attempt to target a single shard by using the query filter. If the operation cannot target a single shard by the query filter, it then attempts to target by the replacement document. In earlier versions, these operations only attempt to target using the replacement document.
- The `save()` method is deprecated: use the `insertOne()` or `replaceOne()` method instead. The `save()` method cannot be used with sharded collections that are *not* sharded by `_id`, and attempting to do so will result in an error.



問題3：結果を格納するときに落ちる

現象：機械学習後にMongoDBに結果が保存されず落ちる

発生箇所：`pymongo.collection.save()`

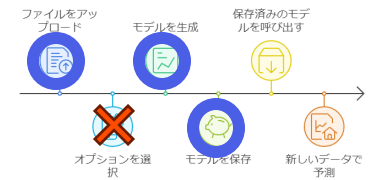
ポイント：MongoDB4.2以降は`db.collection.save()`というAPIがなくなった

公式ドキュメントの変遷を現行のVer7からさかのぼる

MongoDB4.1まで：`save()`だけで新規か更新かはいい感じに判断

MongoDB4.2以降：新規なのか更新なのか明示的に指定が必要

対応：やりたい内容に応じて`insert_one()`と`replace_one()`を使い分け



問題4: Borutaオプションで落ちる

現象：特徴量選択手法でBorutaをオプション指定すると落ちる

発生箇所：`boruta.BorutaPy()`

問題4: Borutaオプションで落ちる

現象：特徴量選択手法でBorutaをオプション指定すると落ちる

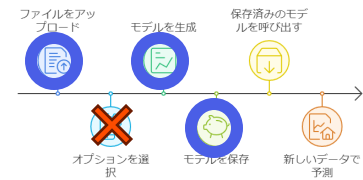
発生箇所：`boruta.BorutaPy()`

NumPy 1.20.0 Release Notes

Deprecations

Using the aliases of builtin types like `np.int` is deprecated

For a long time, `np.int` has been an alias of the builtin `int`. This is repeatedly a cause of confusion for newcomers, and existed mainly for historic reasons.



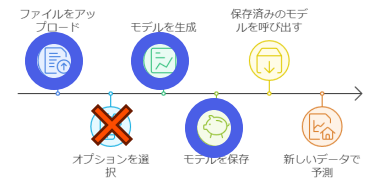
問題4: Borutaオプションで落ちる

現象：特徴量選択手法でBorutaをオプション指定すると落ちる

発生箇所：`boruta.BorutaPy()`

`....../site_packages/boruta/boruta_py.py`

```
# holds the decision about each feature:
# 0 -- default state = tentative in original code
# 1 -- accepted in original code
# -1 -- rejected in original code
dec_reg = np.zeros(n_feat, dtype=np.int)
# counts how many times a given feature was more important than
# the best of the shadow features
hit_reg = np.zeros(n_feat, dtype=np.int)
# these record the history of the iterations
imp_history = np.zeros(n_feat, dtype=np.float)
sha_max_history = []
```



問題4: Borutaオプションで落ちる

現象：特徴量選択手法でBorutaをオプション指定すると落ちる

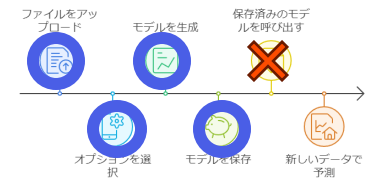
発生箇所：`boruta.BorutaPy()`

ポイント：borutaがNumPy1.20で廃止された`np.int`等を使っている

PyPI上は2019年のVer0.3のまま。作者のGitHubではパッチが当たっている

※2024/8/14にPyPI上のborutaが0.4.3になり今は解消 & scikit-learn1.5.1に包含

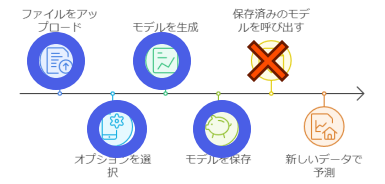
対応：boruta_py.pyにパッチ当て（np.を取る）



問題5：作成済みモデルを呼び出すと落ちる

現象：旧バージョンで作成したモデルを呼び出すと落ちる

発生箇所：`pandas.read_pickle()`



問題5：作成済みモデルを呼び出すと落ちる

現象：旧バージョンで作成したモデルを呼び出すと落ちる

発生箇所：`pandas.read_pickle()`

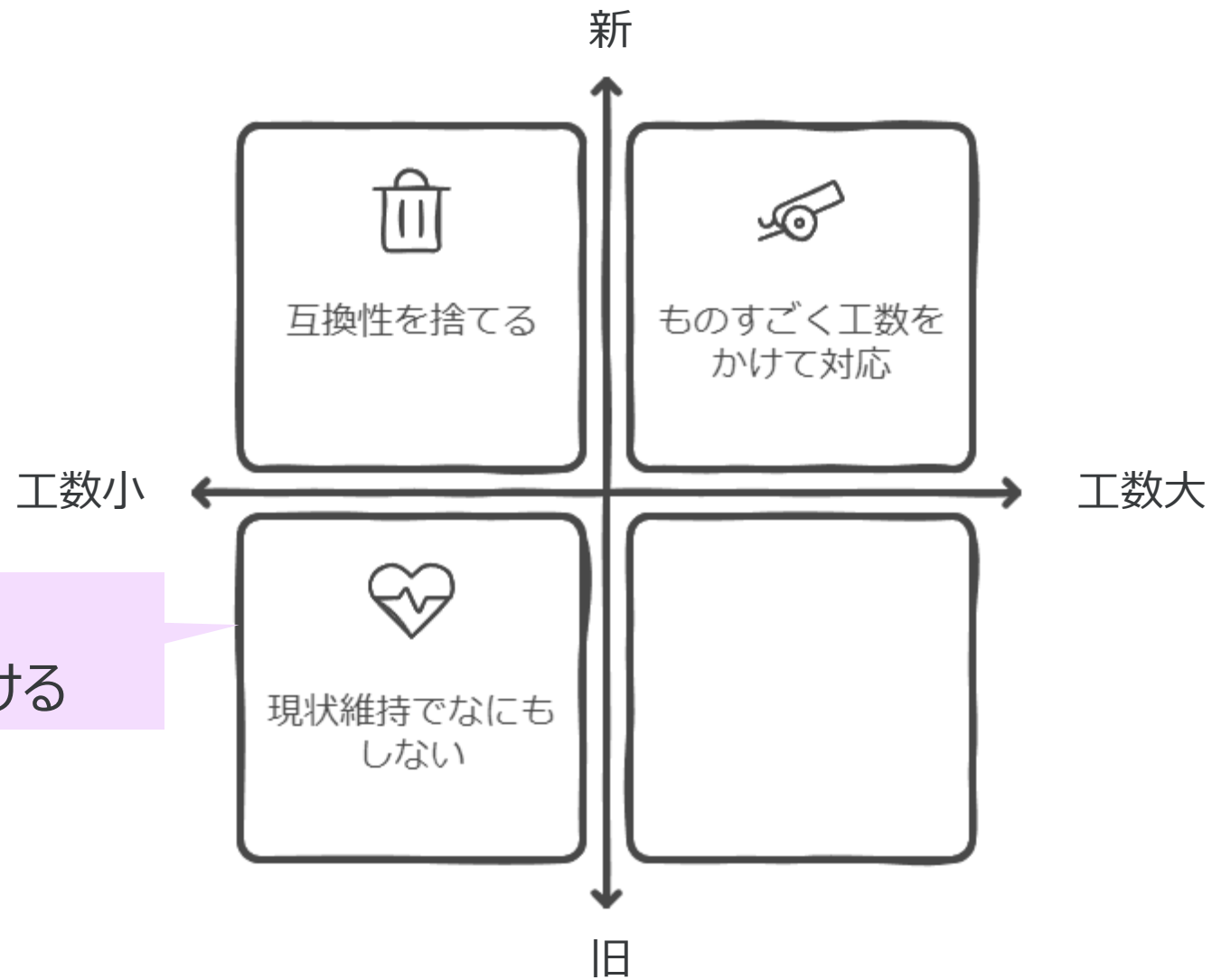
ポイント：モデルはscikit-learnのオブジェクト。scikit-learnのバージョン違いにより

オブジェクトの内部構造（メソッドやプロパティ）が変わり

古いバージョンのオブジェクトのpickleを新バージョンで読み込むことができない

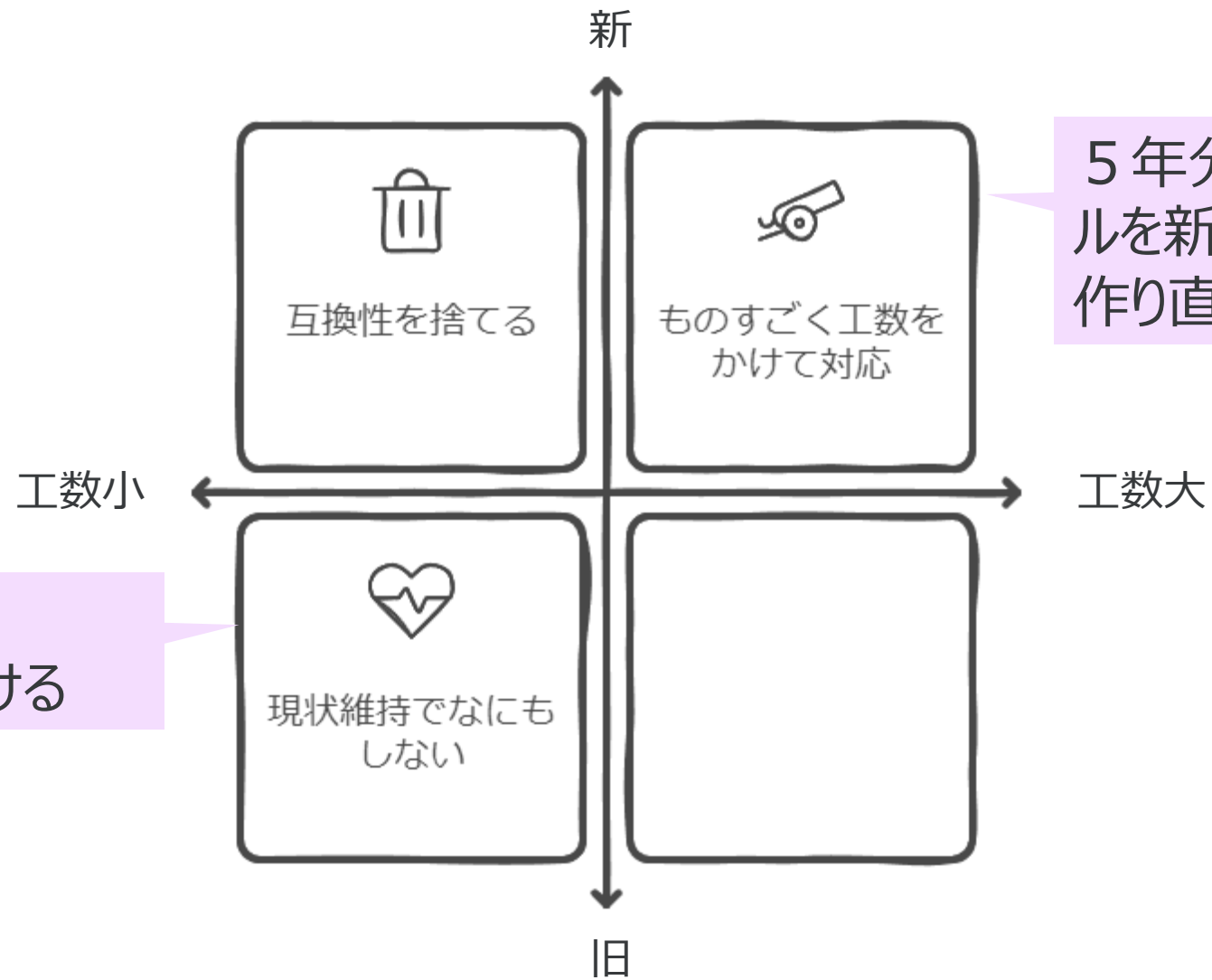
対応：プログラムの的には打つ手なし。どうする？

3つの対応案



問題を先送りし
ツケをため込み続ける

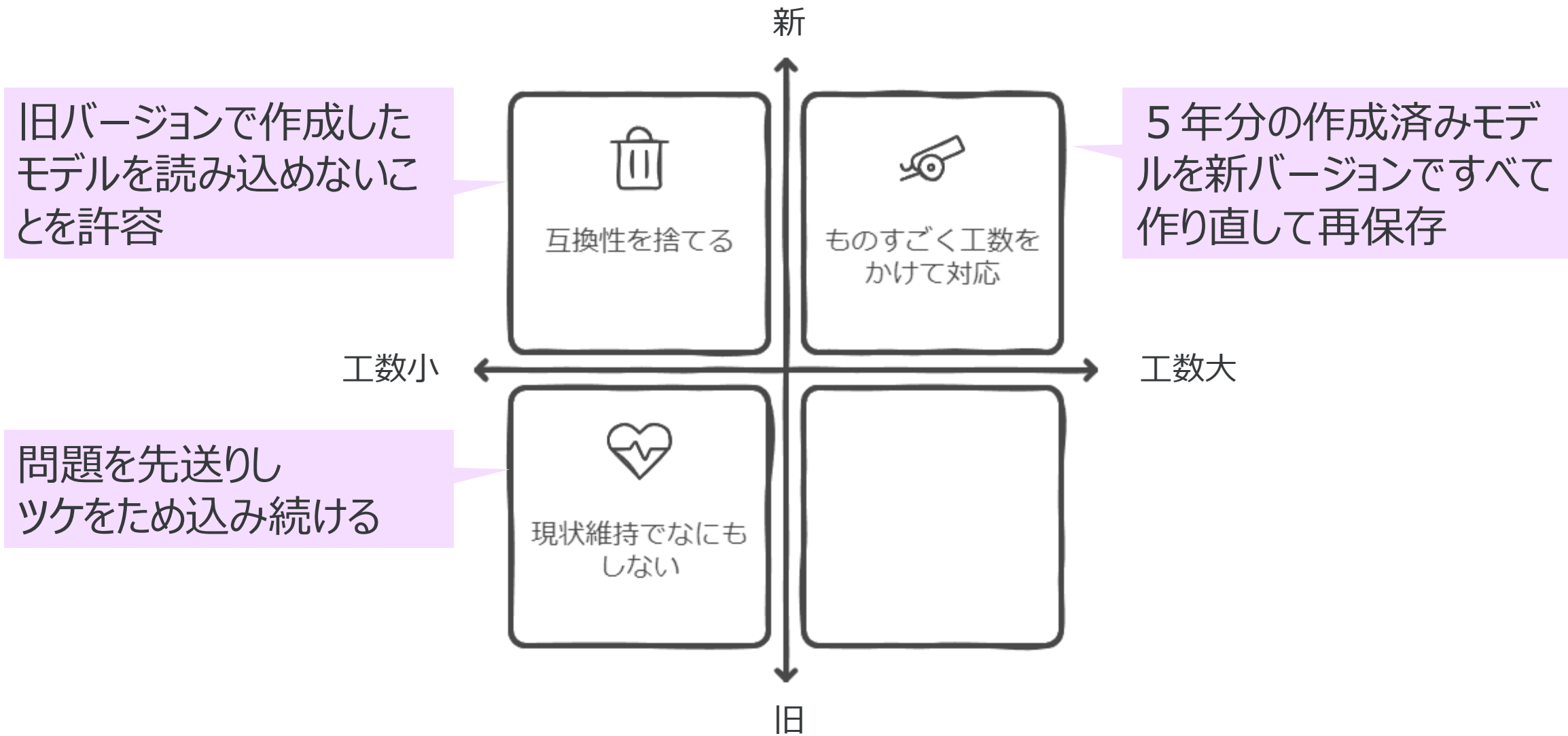
3つの対応案



5年分の作成済みモデルを新バージョンですべて作り直して再保存

問題を先送りしツケをため込み続ける

3つの対応案



対応方針決定

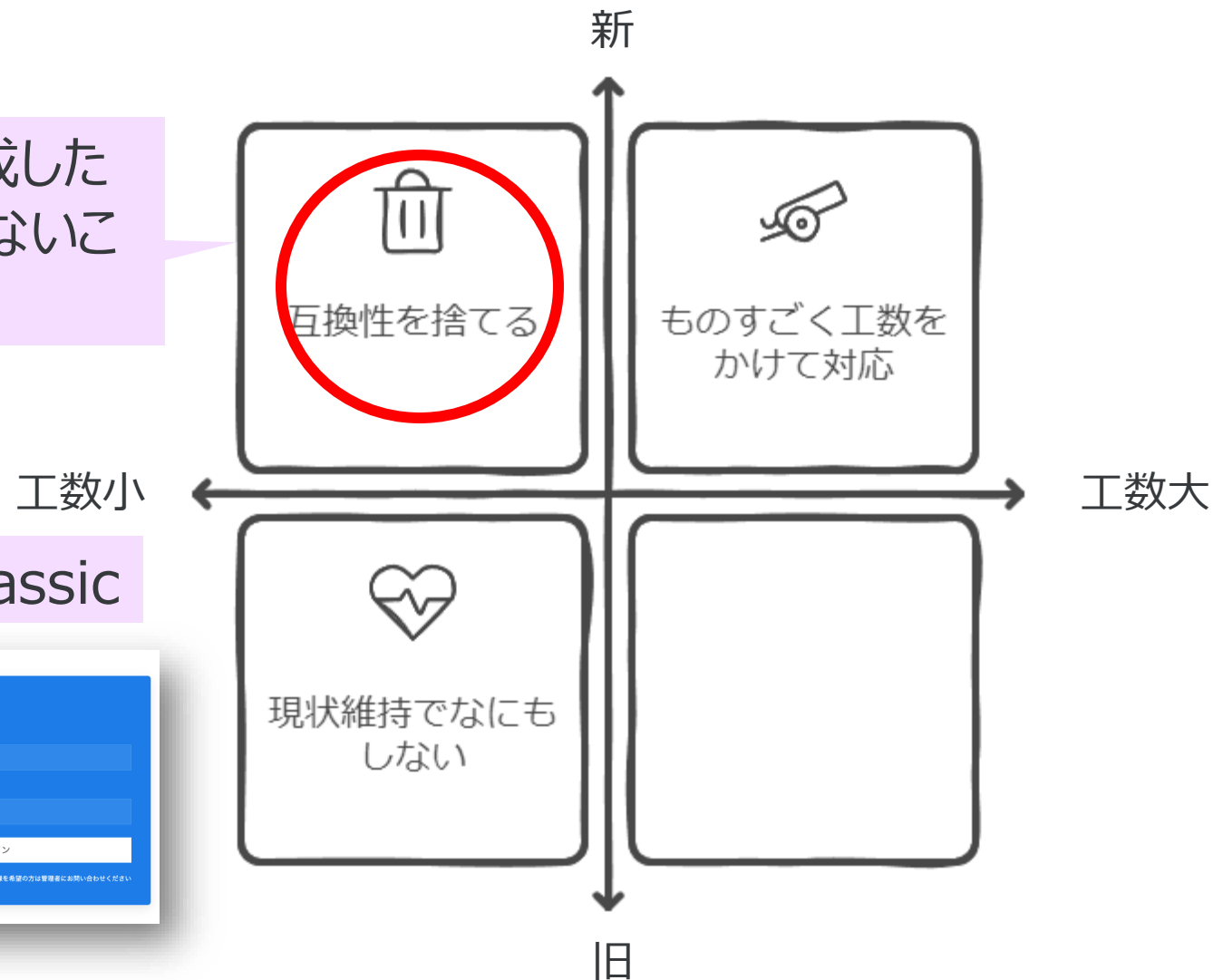
旧バージョンで作成したモデルを読み込めないことを許容

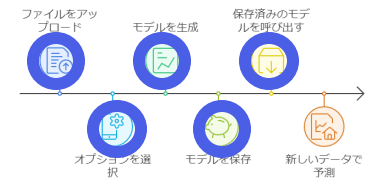
万々に備えたX Classic

X Classic

後輩が作った
イケてる
分析ツール
-- PyConJP2021Demo --

Login
 E-Mail
 Password
 ログイン
 新規登録も希望の方は管理者にお問い合わせください





めでたし、めでたし

- いろいろあったけど、X（仮名）は新しい環境でリリース完了
- 5年分のツケを一気に清算して（一部踏み倒したけど）めでたし、めでたし

めでたし、めでたし？

- いろいろあったけど、X（仮名）は新しい環境でリリース完了
- 5年分のツケを一気に清算して（一部踏み倒したけど）めでたし、めでたし



- 2024年5月、scikit-learn1.5（boruta0.4.3含む）リリース
- 2024年6月、NumPyが18年ぶりのメジャーアップデートで2.0に
- MicrosoftからTeamsのincoming webhook機能廃止の告知

まとめ



一番の教訓は「ツケはため込まないのが一番」

ドキュメント、テストの整備は前提

常に最新化といかないまでも大きなバージョンアップの際には新しいバージョンでの動作確認はしておきたい
(NumPy2.0とか)

レガシーコード化させないためのテストコードの書き方は
午後にいいトークがあります

Enjoy !

