

TECH Street Pythonのエンジニア勉強会

4/27, 2021

辻 真吾 (@tsjshg)

お前、誰よ

- 辻 真吾（つじ しんご） www.tsjshg.info
 - 東京大学先端科学技術研究センターに勤務
 - みんなのPython勉強会を主宰しています
 - IPA情報処理技術者試験試験委員、Pythonエンジニア認定試験監修など
- 小学生の頃MSX2を買ってもらって以来、コンピュータが好き
- 1994～2000 大学生（修士も）のときはC++
- 2000～2002 ITベンチャー勤務時代はJava
- 2004か5 JavaからPythonに乗り換え（博士課程在学中）

著書



初版は2010年、改訂版は2018年に出版
プログラミング未経験者向けに書いたPythonの入門書



Pythonでデータサイエンスを実践するための本。数学の章を執筆
(2018)

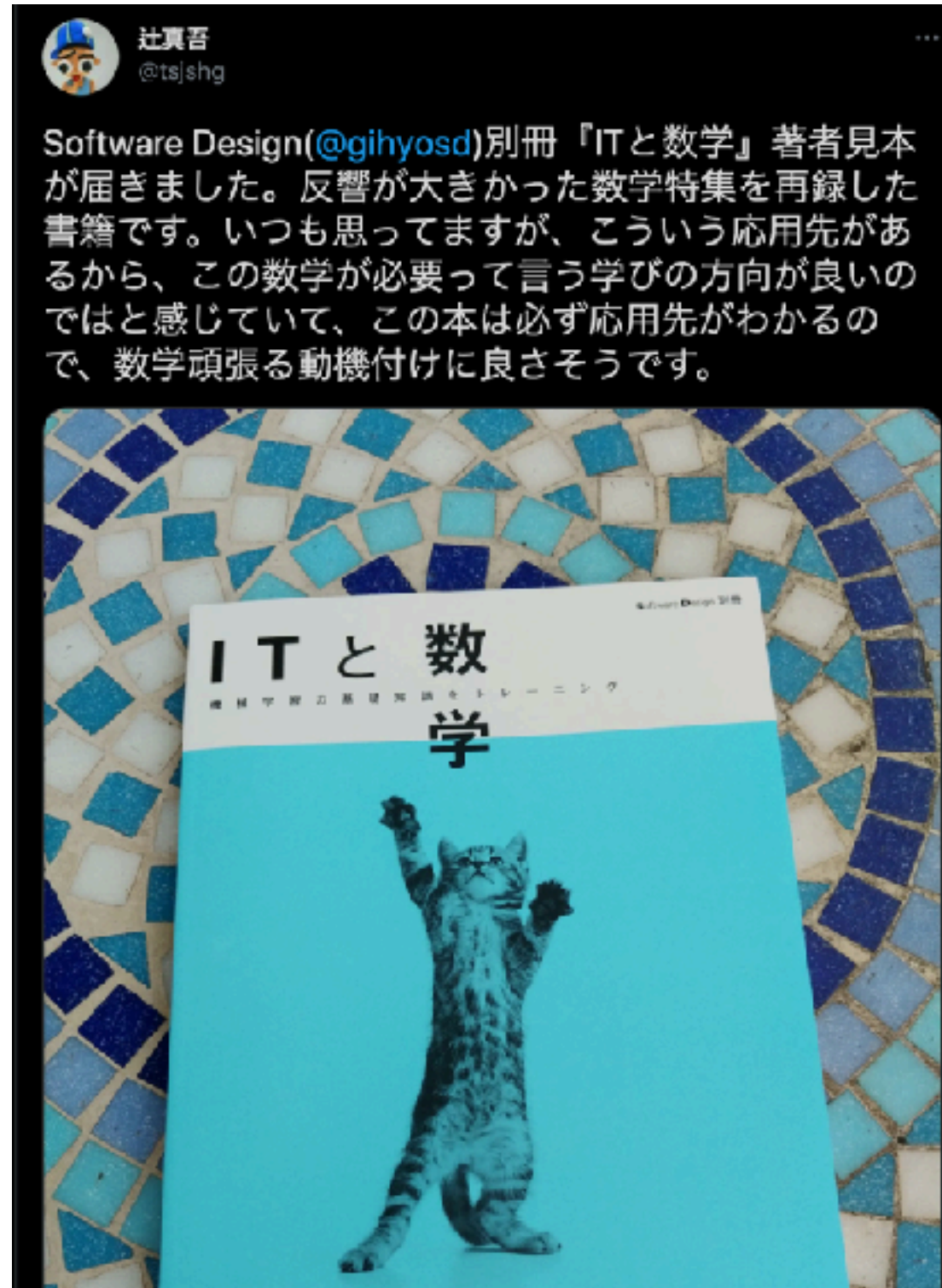


Pythonエンジニア育成推進協会監修で、Pythonのスキルを上げるための本
練習問題が豊富
(2019)



数理・データサイエンス教育強化拠点コンソーシアムのための講談社データサイエンス入門シリーズの1冊
大学初等レベルのアルゴリズムをPythonで学ぶ本
(2019)

4月28日(水)発売（電子版は23日に発売）



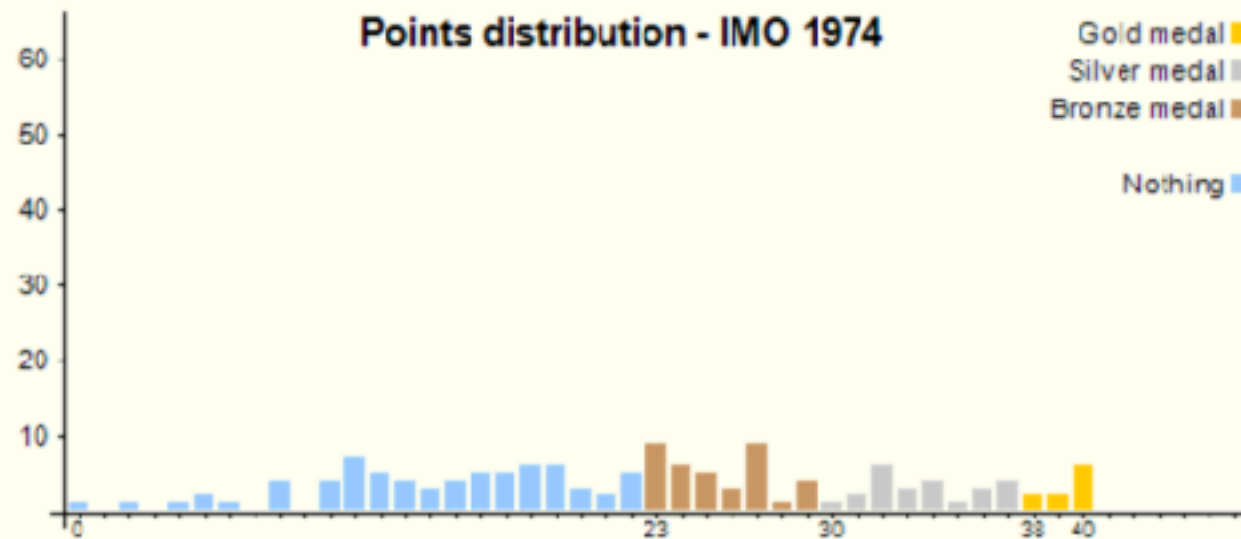
Pythonのトレンド情報

この人は誰でしょう？



◀ 16TH IMO 1974 ▶

COUNTRY RESULTS • INDIVIDUAL RESULTS • STATISTICS



Contestant [♀ ♂] [←]	Country	P1	P2	P3	P4	P5	P6	Total	Rank	Award
Herbert Sinwel	Austria	5	6	8	6	7	8	40	1	Gold medal
Jean Christophe Yoccoz	France	5	6	8	6	7	8	40	1	Gold medal
János Kollár	Hungary	5	6	8	6	7	8	40	1	Gold medal
Adrian Ocneanu	Romania	5	6	8	6	7	8	40	1	Gold medal
Michael Steiner	Sweden	5	6	8	6	7	8	40	1	Gold medal, Special prize
Alexander Grigorjan	Union of Soviet Socialist Republics	5	6	8	6	7	8	40	1	Gold medal
Dimitri Tjukavkin	Union of Soviet Socialist Republics	5	6	8	5	7	8	39	7	Gold medal
Jožef B. Varga	Yugoslavia	5	6	8	6	7	7	39	7	Gold medal
Pavel Kindlmann	Czechoslovakia	5	6	3	6	2	1	23	63	Bronze medal
Georges Welterlin	France	4	0	0	5	7	7	23	63	Bronze medal
Gábor Csuka	Hungary	5	2	1	6	3	6	23	63	Bronze medal
Guido van Rossum	Netherlands	4	6	0	6	7	0	23	63	Bronze medal
Andrzej Turski	Poland	5	5	0	6	7	0	23	63	Bronze medal
Pavle Mladenović	Yugoslavia	5	0	8	5	5	0	23	63	Bronze medal

Pythonの整数型は1つ

```
# 1281桁のメルセンヌ素数  
pow(2, 4253) - 1
```

```
1907970075244390738074680429695291736693569947499401773947418826735289797870050537063  
6804983551490024430349595495070972576218631122414882881192021690454220696074466616936  
4221195289538436845390250168663932838805192055137154390912666527533007309292687539092  
2570433625178573666246999754023754629544902932592333031373306435315565397399219262014  
3860643902007517472302905683827250505157196759460835006340449597766065626902082396082  
5567012344189908927956646011998057988548630107637380993519826582389781888135705408653  
0452196558017580812511640805546090574680282033087187246540810553232158601896113912960  
3047110844314674567196776630892585854727150731156376517100831824864711009761489031356  
2856541784154881743146033909602737947385055355960331855614540900081456378659068370317  
2676969800011877509954910903501084170509179915621679722810701613059725180448720483313  
0638371509485493841573854989460607072258473797817668642213435452698944302835364403718  
7375385397838259511833166416134323695660367676897722287918773420968982326089026150031  
5154241654621113375274311548906663273749214462768335645197767976338755035486650939145  
5648203148224888312702377703966770797655985733335701372734207909906440045574183065432  
0379350833236245819348824064783585692924881021978332974949906122664421376034687815350  
484991
```

メモリが許す限りの大きな整数を扱える

型ヒント (3.5～)

```
def greeting(name: str) -> str:  
    return 'Hello ' + name
```

```
greeting('World')
```

```
'Hello World'
```

型をチェックしてくれているわけではない

```
greeting(3)
```

```
-----  
TypeError                                Traceback (most recent call last)
```

```
<ipython-input-25-a36d78103ab0> in <module>
```

```
----> 1 greeting(3)
```

```
<ipython-input-22-311aff60a9bd> in greeting(name)
```

```
      1 def greeting(name: str) -> str:
```

```
----> 2     return 'Hello ' + name
```

```
TypeError: can only concatenate str (not "int") to str
```

型ヒントについてはこちらがおすすめ

3月 10 みんなのPython勉強会#67
春のPython祭り！

主催：Start Python Club

グループ メンバーです

Start Python Club
Pythonでスタートする人たちの集い。

イベント数 **110回**
メンバー数 **7129**
△

ハッシュタグ： #stapy

フォロー参加者

フォローブックマーク

今から始める型安全 Python

小笠原みつき / @yamitzky
#stapy



<https://startpython.connpass.com/event/201613/>

<https://speakerdeck.com/yamitzky/start-python-type-hints>

dataclass (3.7~)

```
from dataclasses import dataclass
```

```
@dataclass
```

```
class InventoryItem:
```

```
    """Class for keeping track of an item in inventory."""
```

```
    name: str
```

```
    unit_price: float
```

```
    quantity_on_hand: int = 0
```

```
    def total_cost(self) -> float:
```

```
        return self.unit_price * self.quantity_on_hand
```

この2つは型ヒント必須
初期値があれば型は不要

```
item = InventoryItem(3, "test")
```

```
item.total_cost()
```

```
..
```


代入式 別名セイウチ演算子 (3.8～)

```
passwd = 'secrets'  
min_length = 8  
if len(passwd) < min_length:  
    print(f'あと{min_length-len(passwd)}文字足してください。')
```

あと1文字足してください。

こう書けるようになった

```
if (n := len(passwd)) < min_length:  
    print(f'あと{min_length-n}文字足してください。')
```

あと1文字足してください。

代入式 (PEP572) に関する議論に疲れたという理由で、2018年7月グイドさんが、BDFL (Benevolent Dictator For Life) を引退

そもそも=は使えないのが良いところだった

```
if n = len(pwd) < min_length:  
    print(f'あと{min_length-n}文字足してください。')
```

```
File "<ipython-input-38-da9e5c07314c>", line 1  
    if n = len(pwd) < min_length:  
        ^
```

```
SyntaxError: invalid syntax
```

やはり注意が必要 (:=はもっとも優先順位が低い演算子)

```
if n := len(pwd) < min_length:  
    print(f'あと{min_length-n}文字足してください。')
```

```
あと7文字足してください。
```

みんな大好きAnaconda

- Anaconda社が配布するPython
- Python＋独自ツールconda
- 多くの外部ライブラリを同梱（データ解析関連が充実している）
- condaはpipとvenvの機能を持ち合わせるコマンド



Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

2020年9月に規約が改定され完全無料ではなくなった

以下に該当しなければ有償での利用となった模様

<https://www.anaconda.com/terms-of-service> からの引用

To avoid confusion, "commercial activities" are any use of the Repository which is NOT:

- use solely by an individual using for personal, non-business purposes,
- use by a student or employee of an educational institution in connection with educational activities,
- use by an employee or volunteer on behalf of a non-profit institution in connection with the provision of charitable services,
- use by a 501(c)(3) non-profit research institution in connection with non-commercial activities, such as research to address societal needs and global challenges, public policy development, or the advancement of science for the general good, where such work will not result in revenue to the institution, or
- use by entities in common control with each other with fewer than 200 employees in aggregate.

200人以上の従業員がいる組織は対象外

詳しい解説ページ（日本語） → <https://qiita.com/tfukumori/items/f8fc2c53077b234384fc>

環境設定

- 環境に依存したバイナリを含むwheel形式の普及によって、pipのインストールで困ることはなくなってきた
- 標準のPythonだけでOK
 - 仮想環境をvenvで作ってpipでライブラリを管理する
 - みんなのPython勉強会 # 67の動画
 - <https://www.youtube.com/watch?v=S6MHI8OMTRA&t=191s>
- Windowsではランチャーのpyコマンドが便利

ご意見、ご質問受け付けます

この先Pythonに取り組んでいく際におさえておきたいポイント

お題をいただいたので・・・

- ・ 型ヒントの普及
- ・ テスト駆動型開発
- ・ 他の言語との連携
 - ・ Rustなど比較的あたらしいものも
- ・ 人気に陰りがでるのは、いつか？
- ・ （マイナーな意見として）3.9のgraphlib.TopologicalSorter

データサイエンスにおけるlow code, no code

- pandas-profiling
 - 探索的データ分析（EDA: Exploratory Data Analysis）の強い味方
- PyCaret
 - 機械学習の自動化
 - 依存するライブラリが多いので別仮想環境がおすすめ
- PlotlyとDash
 - インタラクティブな可視化とWebアプリ開発

pandas-profiling

```
import pandas as pd
from pandas_profiling import ProfileReport

credit_data = pd.read_csv('BankChurners.csv')

# 1~2分ほどかかります。
profile = ProfileReport(credit_data, title="クレジットカード顧客分析")
profile.to_file("my_report.html")

# 以下のコードでノートブックへの埋め込みも可能
# profile.to_notebook_iframe()
```

Summarize dataset:  36/36 [00:30<00:00, 1.26s/it,
100% Completed]

当日はKaggleからダウンロードしたcsvファイルを使った結果をデモしました

http://www.tsjshg.info/my_report.html

<https://pandas-profiling.github.io/pandas-profiling/docs/master/rtd/index.html>

<https://www.kaggle.com/sakshigoyal7/credit-card-customers>

PyCaret (RのCaretがやってきた)

タイタニックのデータを使って分類モデル作成のセットアップ

```
import seaborn as sns
from pycaret.classification import *

titanic_df = sns.load_dataset("titanic")
clf = setup(titanic_df, target="survived")
```

	Description	Value
0	session_id	6191
1	Target	survived
2	Target Type	Binary
3	Label Encoded	0: 0, 1: 1
4	Original Data	(891, 15)
5	Missing Values	True
6	Numeric Features	2

60項目弱ある

12個のモデルを自動生成

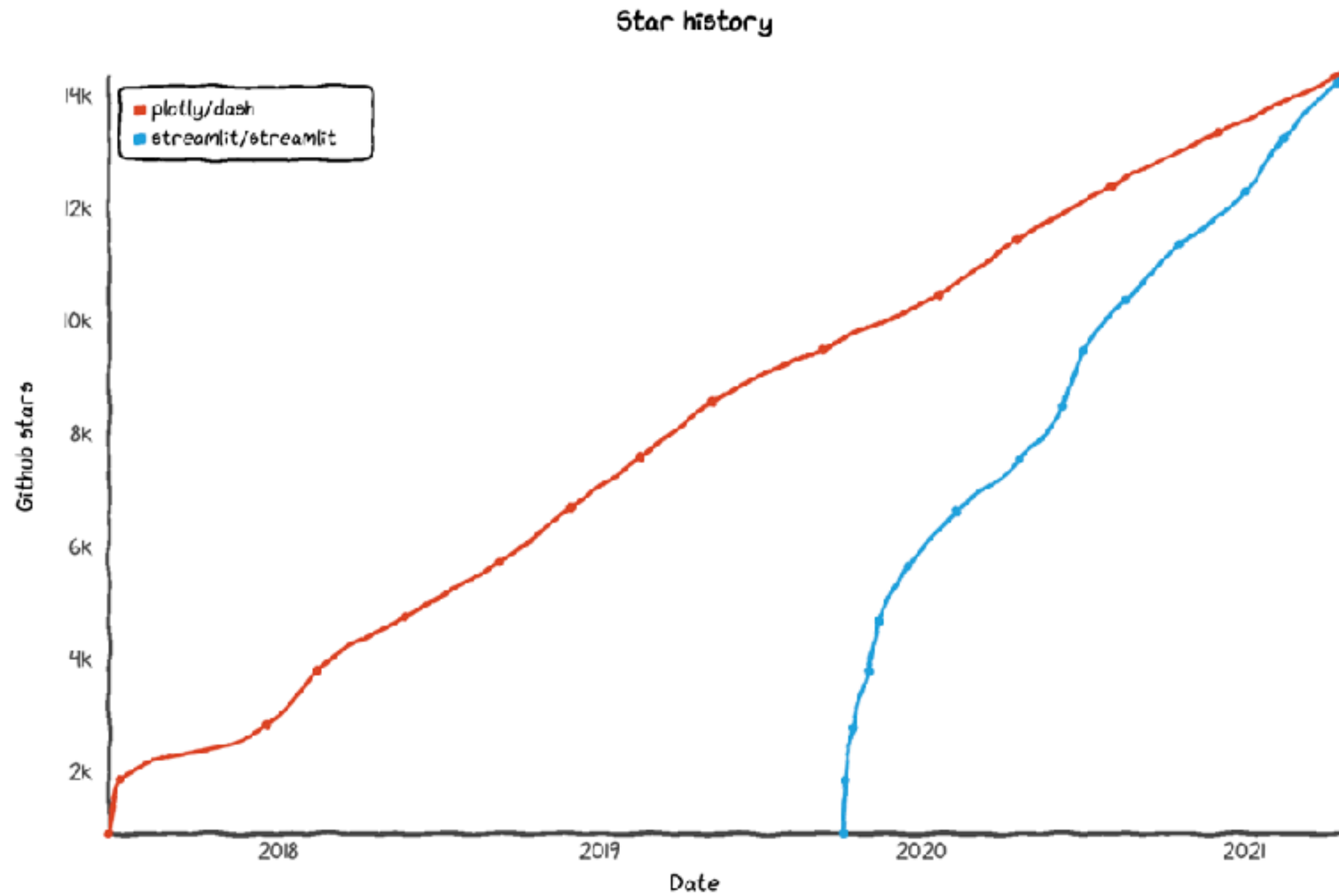
```
best_model = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0740
nb	Naive Bayes	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0060
dt	Decision Tree Classifier	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0060
ridge	Ridge Classifier	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0060
rf	Random Forest Classifier	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0700
ada	Ada Boost Classifier	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0080
	Gradient								

Plotly/Dash

当日は、<https://dash-gallery.plotly.host/dash-tsne/>
こちらを少し改変したアプリを作りデモをしました

Streamlitがきてる



ご意見、ご質問受け付けます