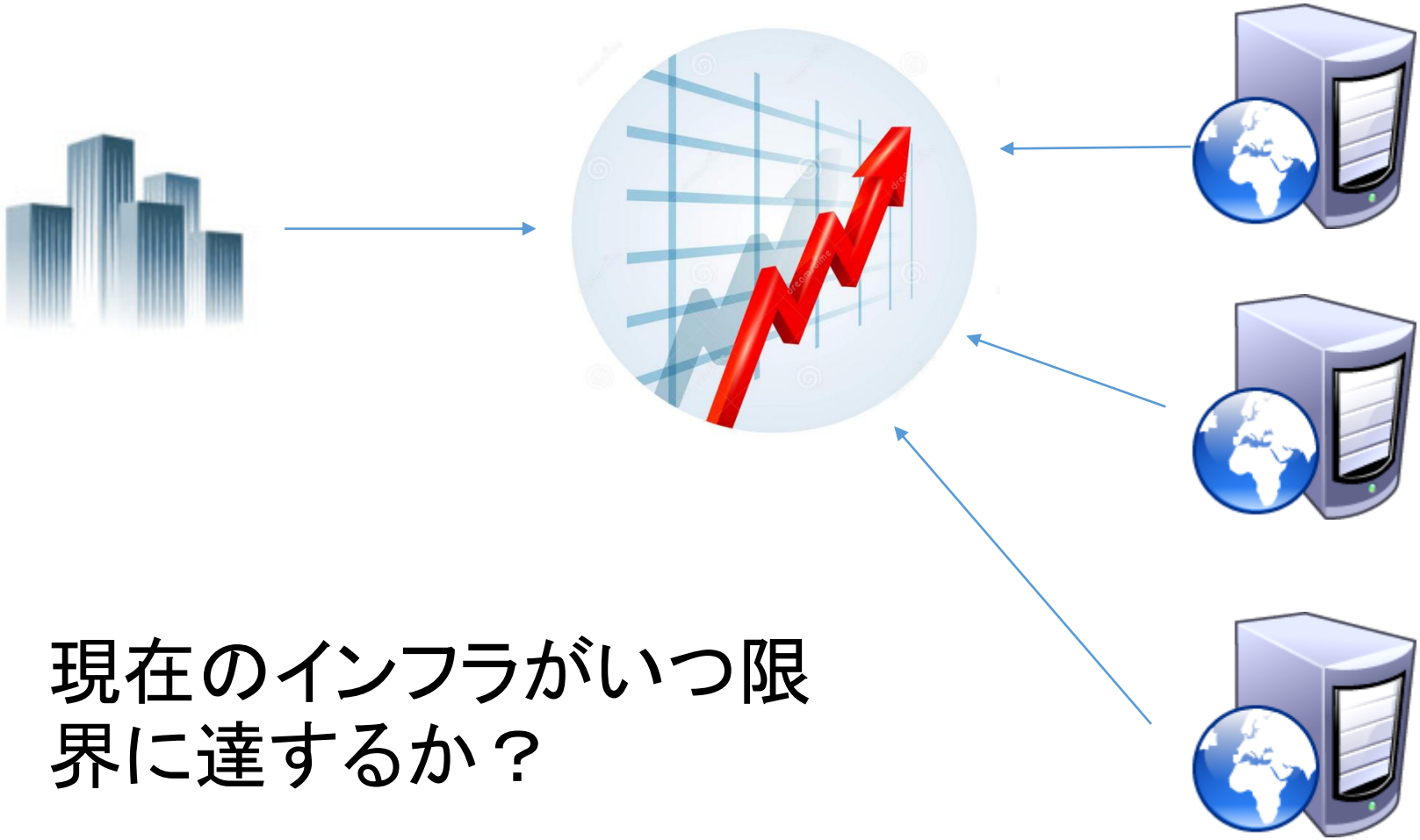
Pythonではじめる機械学習

CAO RUI

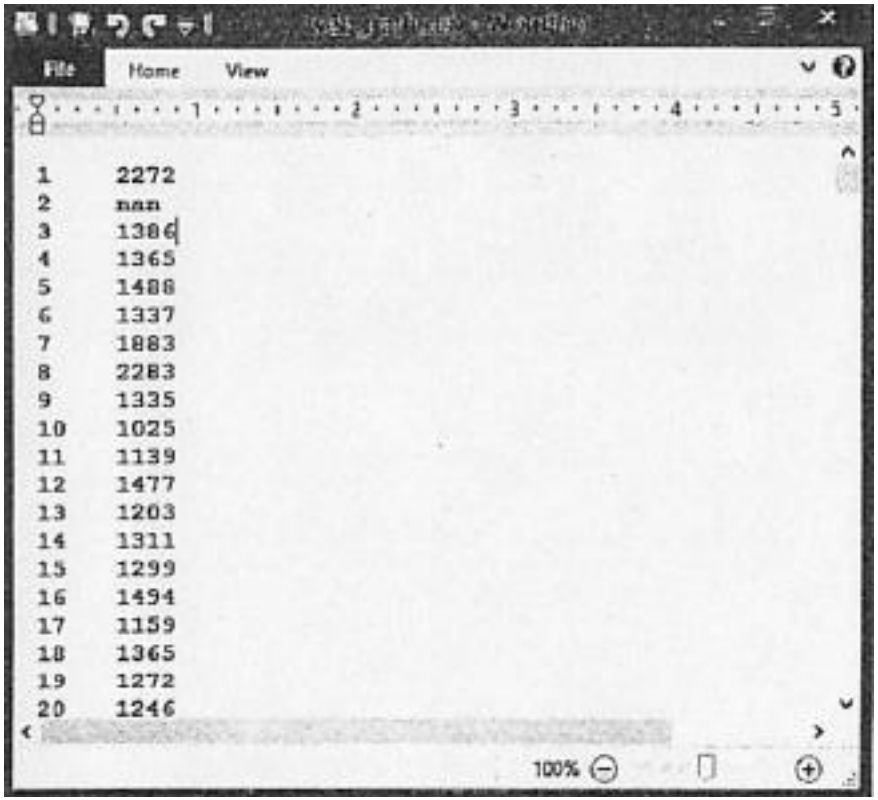
- NumPy
  - 配列を作成する時に使う
- SciPy
  - 数値計算に関するアルゴリズムを満載
- Matplotlib
  - 図を作る

MLAAS



現在のインフラがいつ限界に達するか？

データの前処理とデータ整形

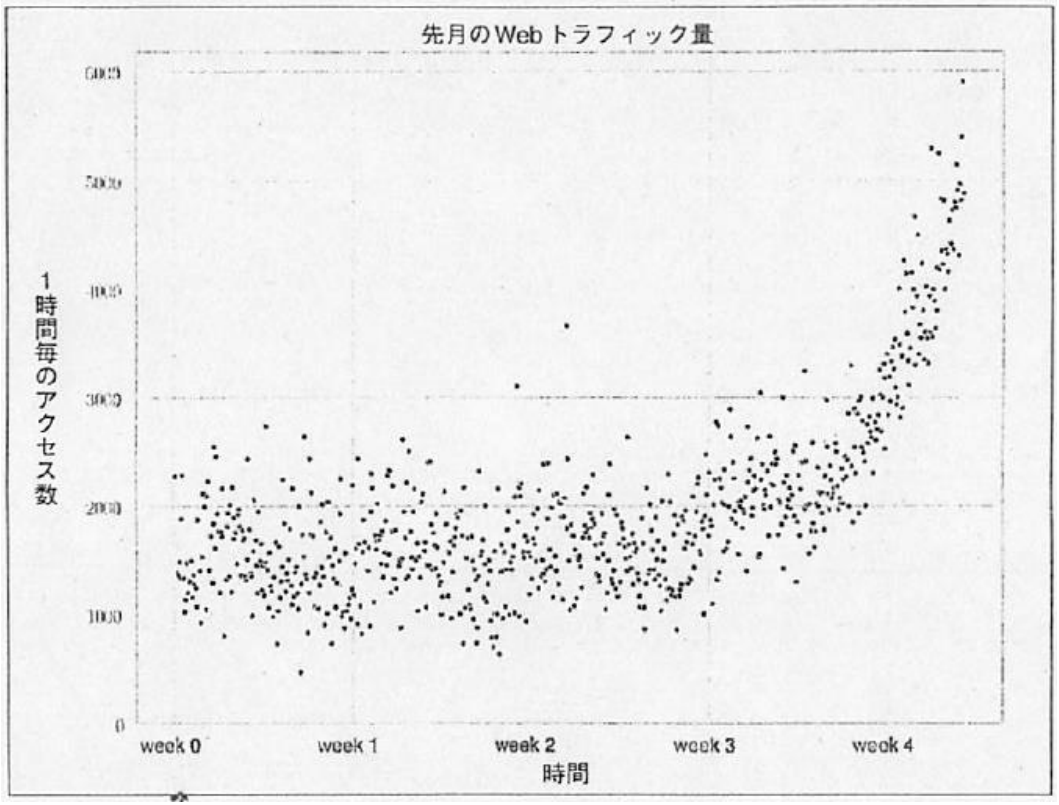



```
>>> print(data[:10])
[[  1.00000000e+00   2.27200000e+03]
 [  2.00000000e+00              nan]
 [  3.00000000e+00   1.38600000e+03]
 [  4.00000000e+00   1.36500000e+03]
 [  5.00000000e+00   1.48800000e+03]
 [  6.00000000e+00   1.33700000e+03]
 [  7.00000000e+00   1.88300000e+03]
 [  8.00000000e+00   2.28300000e+03]
 [  9.00000000e+00   1.33500000e+03]
 [  1.00000000e+01   1.02500000e+03]]
```

```
>>> sp.sum(sp.isnan(y))
8
```

```
x = x[~sp.isnan(y)]
y = y[~sp.isnan(y)]
```

いつまでトラフィックにたえるか？

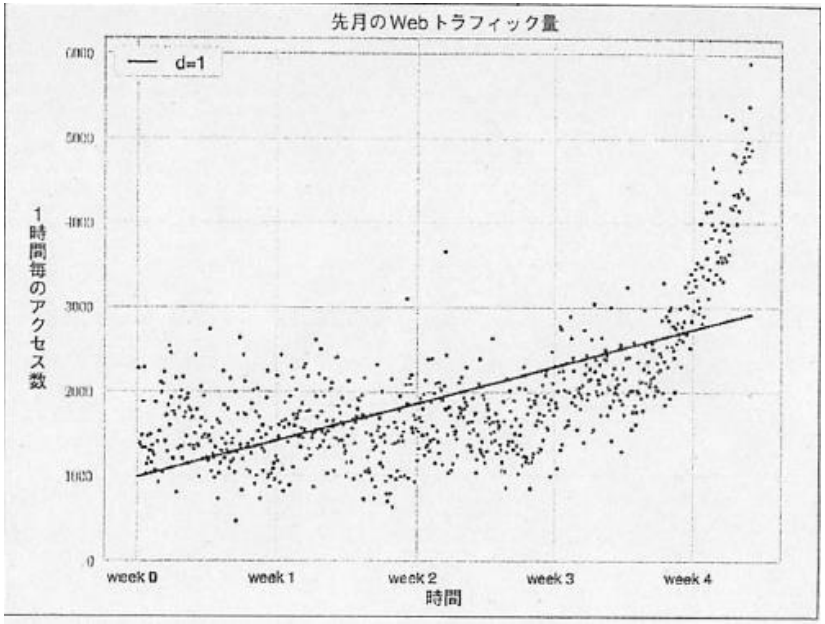


- ノイズの含まれたデータに対して、その背後にある本当のモデルを見つける
- そのモデルを用いて、インフラがバンクする時期を予測する

どのように直線を配置したら近似誤差を最小にできるか？

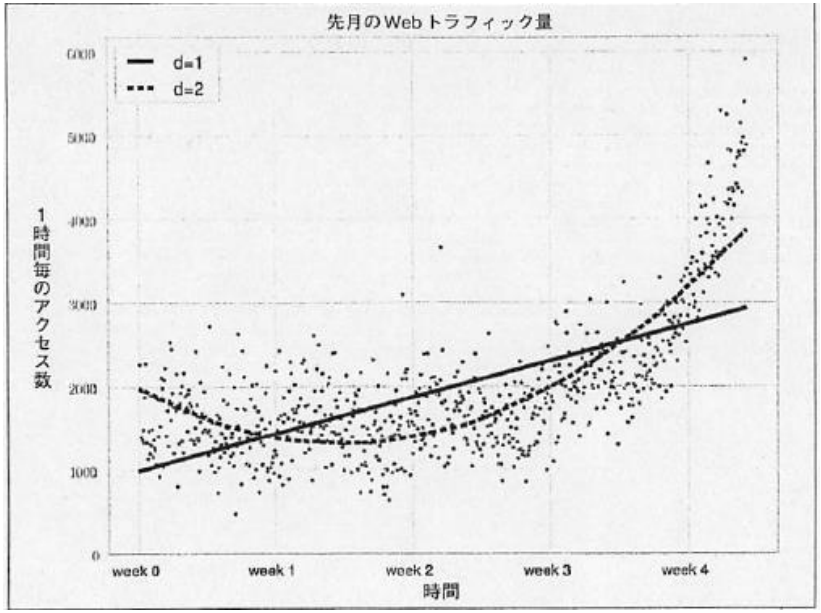
```
fp1, residuals, rank, sv, rcond = sp.polyfit(x, y, 1, full=True)
```

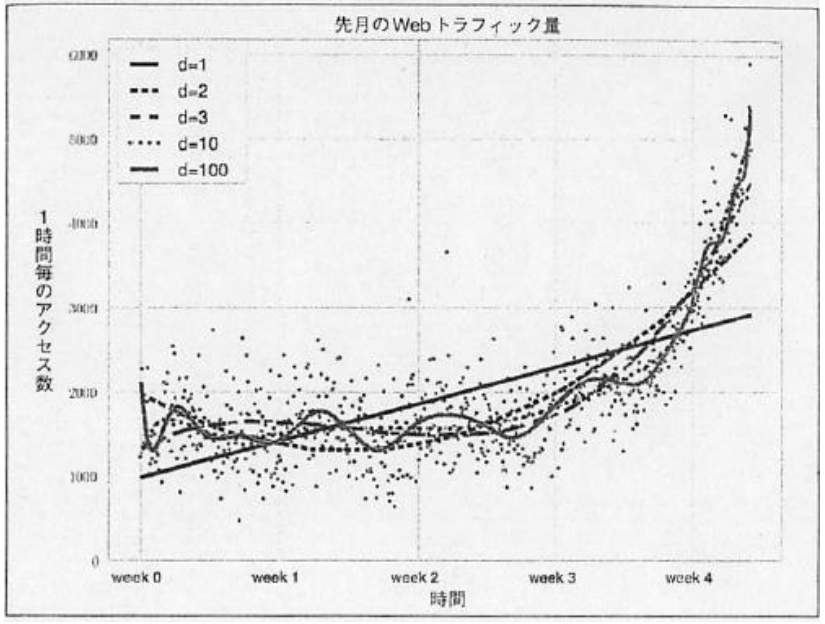
```
f(x) = 2.59619213 * x + 989.02487106.
```




```
>>> f1 = sp.poly1d(fp1)
>>> print(error(f1, x, y))
317389767.34
```

多項式曲線で新しいモデルを生成





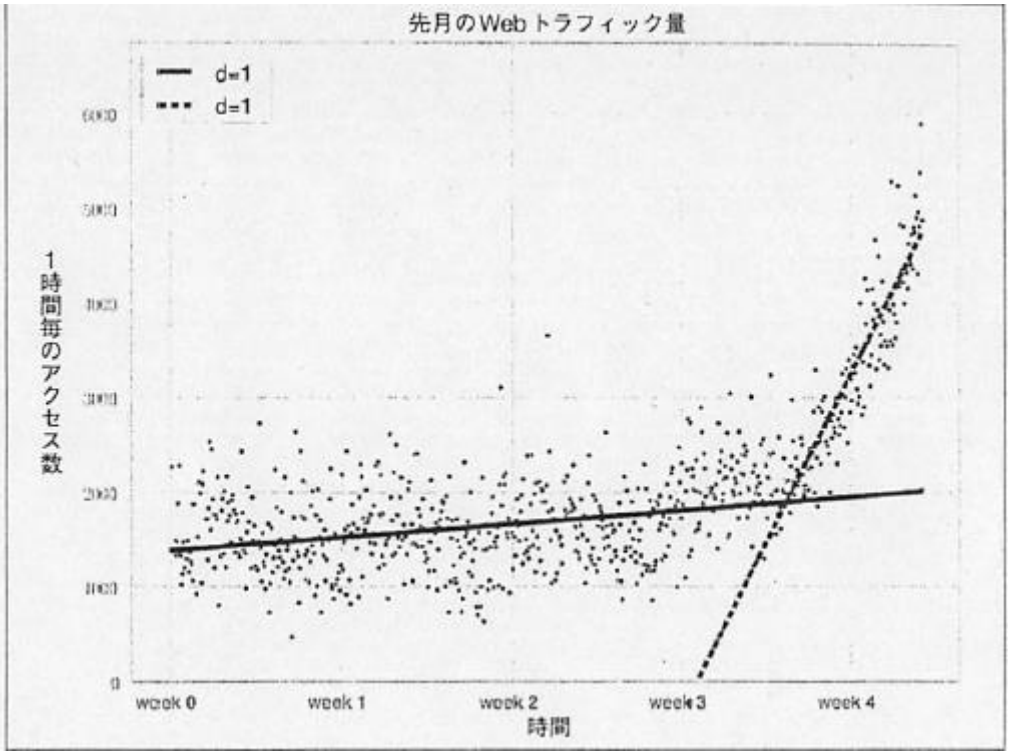
誤差は179,983,507.878

```
Error d=1: 317,389,767.339778
Error d=2: 179,983,507.878179
Error d=3: 139,350,144.031725
Error d=10: 121,942,326.363461
Error d=100: 109,318,004.475556
```

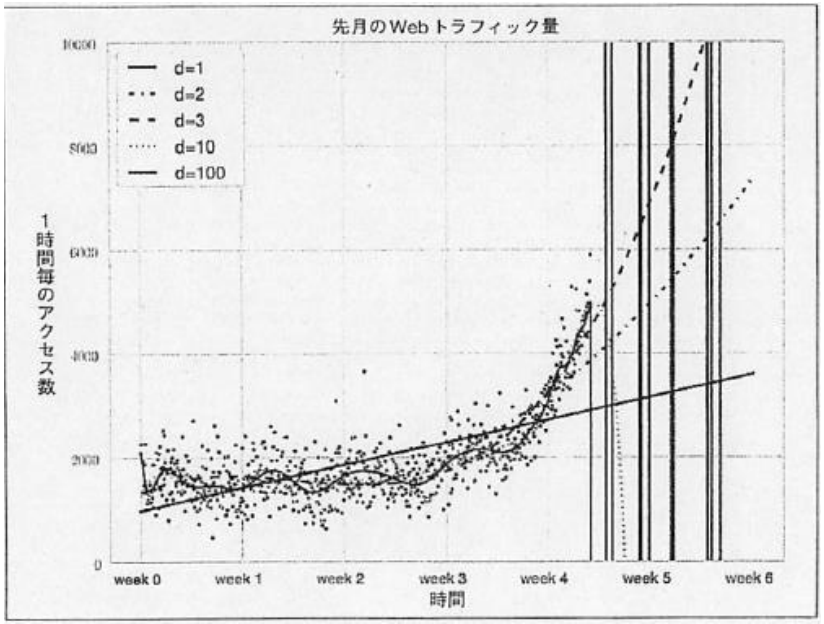
過学習を防ぐための対策

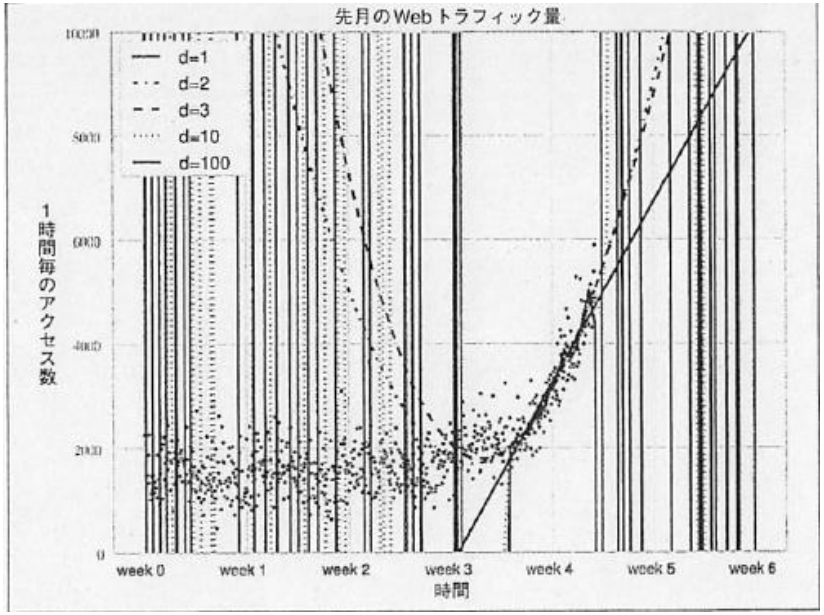
- 多項式モデルからどれか一つを選択する　X
- より複雑なモデル（スプライン曲線など）に切り替える　X
- **データについて別の視点で考え、はじめからやり直す　？**

データを違う視点から眺める



誤差は高次多項式の場合はより依然として大きい　この誤差は信頼できるか






```
Error d=1:   22143941.107618
Error d=2:   19768846.989176
Error d=3:   19766452.361027
Error d=10:  18949339.348539
Error d=100: 16915159.603877
```

訓練データとテストデータ

- 訓練データ

  モデルの学習に使用するデータ

- テストデータ

  検証に使用するデータ

まとめ

- 機械学習の作業において大切なことは、データを理解しデータを扱いやすい形に整形すること

- 正しい評価を行うこと