

# 第一回レポート課題（回帰）

- Webページのpopularity予測
  - 詳細は割愛（元データがばれるため…）
- 53種類の素性（特徴）からWebニュースの人気（コメント数）を予測
- 説明変数：別ファイル参照
- 目的変数：コメント数（54番目のカラム）

# コンペティション形式

- 学習データ：20000, テストデータ：10000
  - 元データを探すことはしないで。。。

- 評価指標

- Mean absolute error

$$\text{MAE} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} |y_i - \hat{y}_i|$$

- ランキング

- スコアリングサーバ: <http://www.nlab.ci.i.u-tokyo.ac.jp/~nakayama/ds17/report1/index.php>
  - テストサンプルの推定結果を一行ずつ記載したテキストファイルを提出
  - ユーザ名を忘れずに（区別できればなんでもよい。ただし同じ名前をずっと使用すること）
  - 提出した結果は上書きされる。最後のものだけ保存されるので注意。
  - ここまで講義で扱った技術のみを使ってよいランキングと、何でもありのランキングに分けているので間違えないように
- 現在は、提出されたテストデータの推定結果のうち、所定の2000サンプルでスコアリングしている
- 最終的なスコアは、締め切り後に残りのサンプルも合わせて算出

# サンプルプログラム (線形回帰、sample.py)

```
N = 10000
nfeat = 53
```

```
train_data = pd.read_csv("train.csv",header=None)
test_data = pd.read_csv("test.csv",header=None)
```

```
X = np.matrix(train_data)[:,:nfeat]
y = np.matrix(train_data)[:,:nfeat].T #This is the target
```

##NOTE: unlike in previous notebooks, we are using row-wise sample matrix.

```
XTrain = X[:N,:] #use the first N samples for training
yTrain = y[:N]
XVal = X[N,:] #use the rests for validation
yVal = y[:N]
```

```
w = reg.ridgeRegres(XTrain,yTrain,lam=0.1) #linear regression
```

```
yHatTrain = np.dot(XTrain,w)
yHatVal = np.dot(XVal,w)
```

```
print("Training error ", np.mean(np.abs(yTrain - yHatTrain.T)))
print("Validation error ", np.mean(np.abs(yVal - yHatVal.T)))
```

```
yHatTest = np.dot(np.matrix(test_data),w)
np.savetxt('result.txt', yHatTest)
print "Validation error ", np.mean(np.abs(yVal - yHatVal.T))
```

```
yHatTest = np.dot(np.matrix(test_data),w)
np.savetxt('result.txt', yHatTest)
```

```
-7.147297933304066930e+00
-8.962127705469644212e+00
6.437042503287904616e+00
3.995455860717621377e+00
6.223865085716232315e+00
8.268411496238270431e+00
-1.643316765033846494e+00
-1.724821802276007787e+00
1.338253036203899171e+01
4.782330176084697904e+00
...
...
```

(result.txt)

test.csv と同じ順番で、  
各テストサンプルの予  
測値が一行ずつ入っ  
ている。

# 課題詳細

- レポート内容
  - Popularity予測の問題に取り組み、以下の点を中心にA4用紙2~3枚程度にまとめよ。  
(※講義で扱っていない技術を用いても構わないが、その場合free-styleのランキングシステムへ提出すること。)
- 1. 予測性能を向上させるための自分なりの工夫点と結果、考察 (必須)
  - コードを載せる必要はない (実装がポイントの場合は載せてかまわない)
- 2. その他、自由にデータを分析した結果 (+a)
  - 学習データ数と性能、正則化パラメータの関係
  - 各特徴の寄与の分析
- 3. ここまでの講義の感想、要望など (できるだけ)

# 課題詳細

- 評価の方針
  - アイデアや試行錯誤の過程を重視
  - スコアが悪くても、しっかり考察してくれればOK  
(もちろん良くなればプラスに評価しますが)
  - 面白い分析を期待します
- 提出先
  - 以下のアドレスへメールで提出すること（質問もこちらへ）
  - [ds2017@nlab.ci.i.u-tokyo.ac.jp](mailto:ds2017@nlab.ci.i.u-tokyo.ac.jp)
  - 件名は「データサイエンスレポート課題 1」
  - 氏名、学籍番号、所属を忘れずに
- 締め切り
  - 12月8日（金） 23:59
  - スコアリングサーバも同時に締め切り後、テストサンプルでランキング。

# 工夫できそうなところ

- 回帰のオフセット
- 目的変数の性質をよく考える
  - どんな分布をしていそうか？
- 手法・パラメータチューニング
  - 正則化項の入れ方（リッジ回帰）
  - クロスバリデーション
- 誤差の評価
  - L2? L1?
- 特徴の前処理