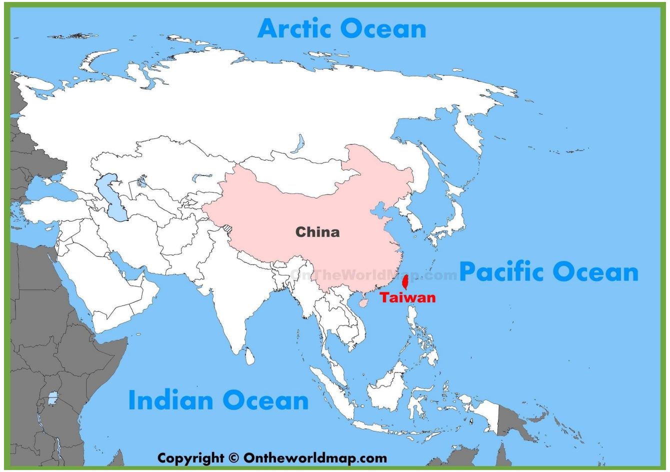
Predicting Credit Card Defaulting

Christopher Bui
Project #3

Background

- Credit Card Defaulting
  - Unable to pay back your credit debt
- 1-3% default
- Circumstances
  - 2005
  - Taiwan in economic downturn



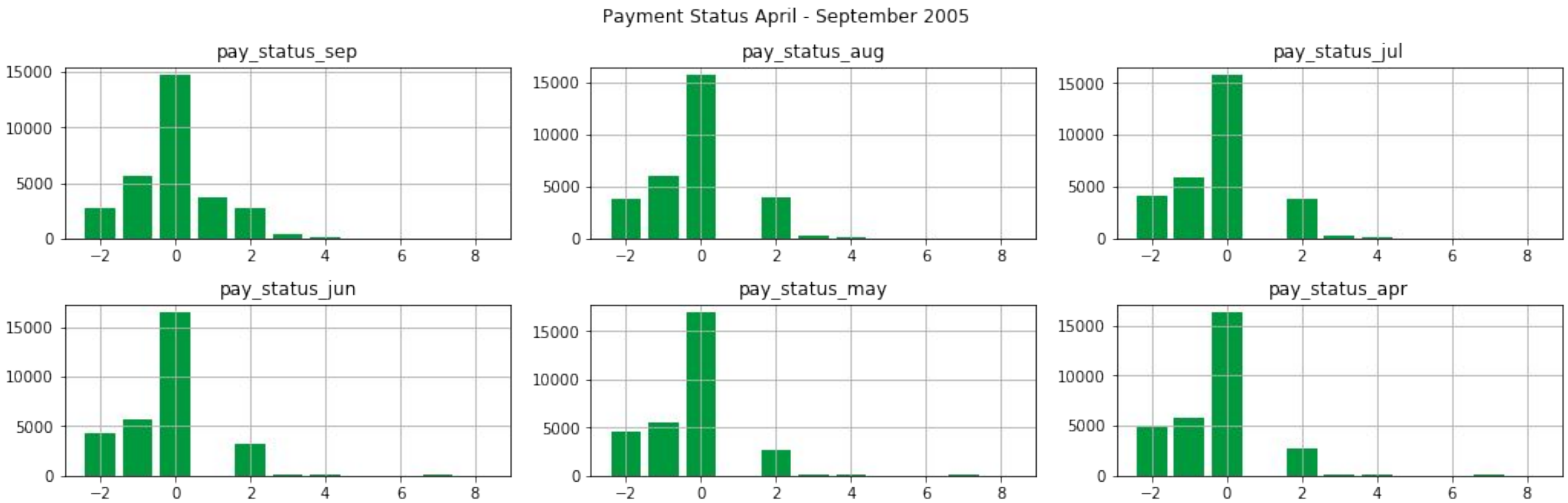
Data

- 23 Features   30,000 Observations
- April - September 2005

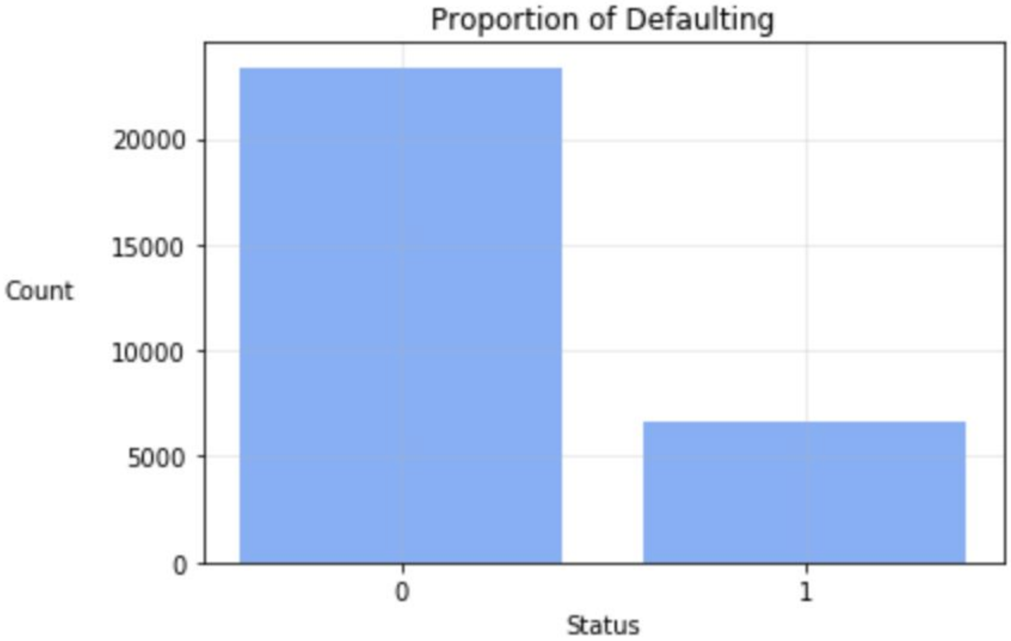
	credit	gender	education	martial_status	age	pay_status_aug	bill_aug	paid_aug	Y
1	627	2	2	1	24	2	97	21	1
2	3767	2	2	2	26	2	54	31	1
3	2825	2	2	2	34	0	440	47	0
4	1569	2	2	1	37	0	1514	63	0
5	1569	1	2	1	57	0	178	1151	0

Tools / Methods

- Matplotlib, Sklearn, Github
- AWS EC2
- Classification Models
    - KNN, CART



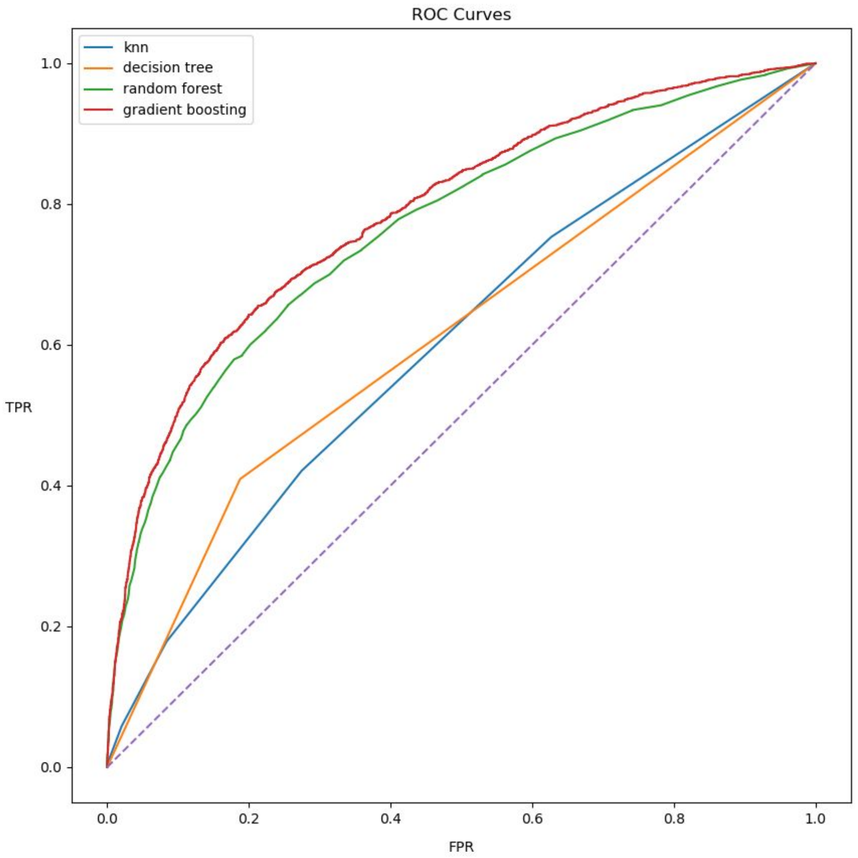
86 % pay early / on time



22% Defaulted

6600 / 30000 Defaulted

Choosing Model

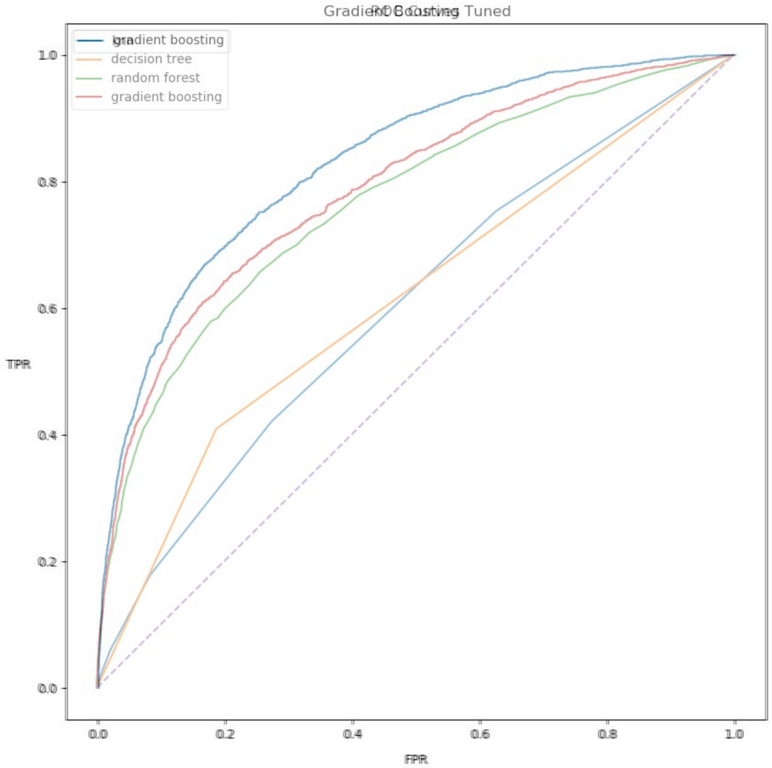


Model	**AUC**
GRAD	0.79
RF	0.77
TREE	0.61
KNN	0.59

Gradient Boosting

- max_depth = **3**, 5, 10 . . .
- n_estimators = 100, 200, **300 . . .**
- max_features = 12, 19, **All . . .**

Before & After

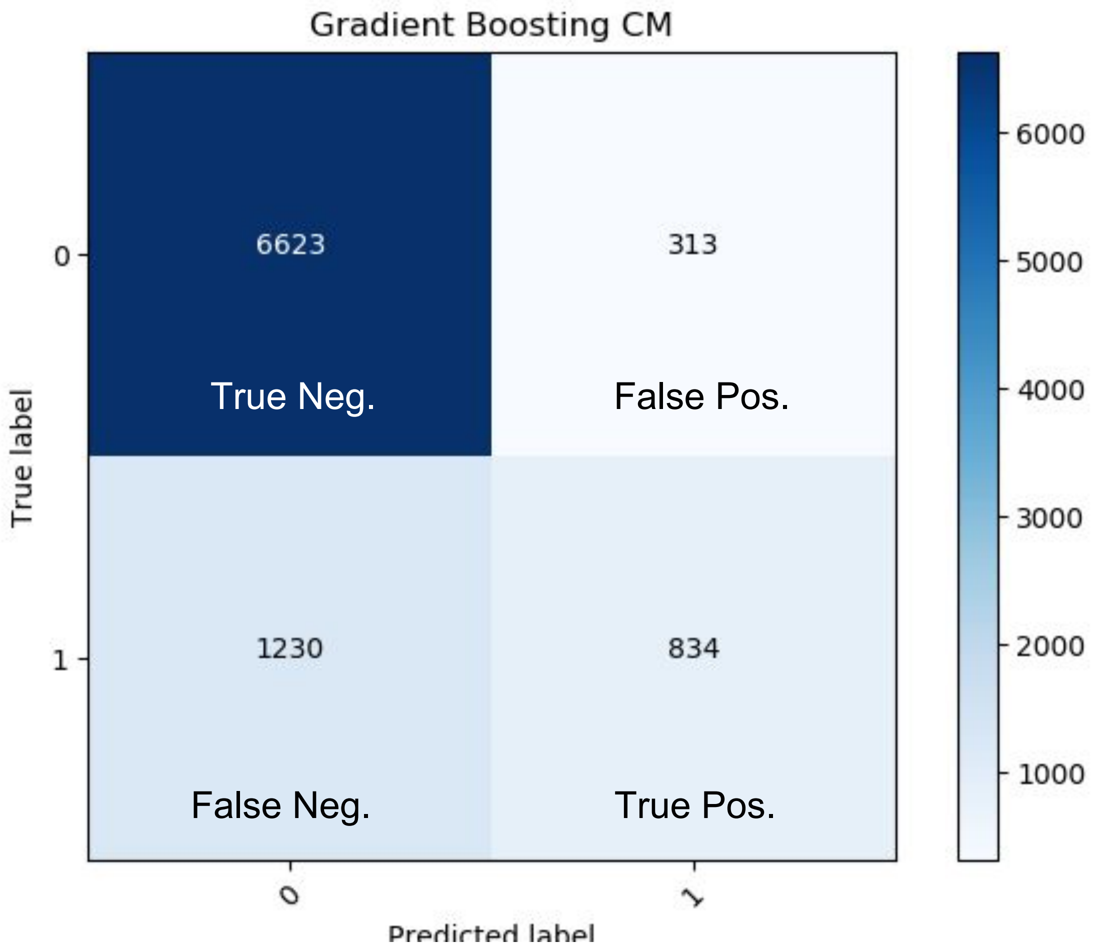


Old AUC: 0.79
New AUC: 0.83

Shows Improvement

Optimal Parameters:

max_depth=3
n_estimators=300
max_features=ALL



Metric	Score
Precision	0.72
Recall	0.40
F1	0.48
Accuracy	0.82
AUC	0.83

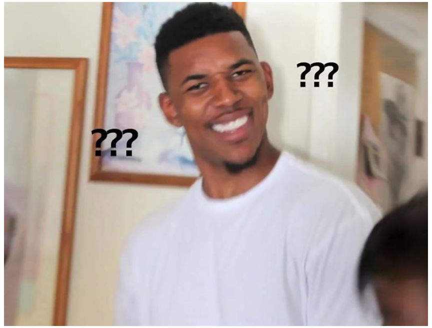
Best Features

- **bill_sep** & **pay_status_sep**   (Most Important Features)
  - Follow up features: bill_aug, bill_jul

- What about earlier months' payment status?
  - Much less important (30% of pay_status_sep)

- Model follows natural intuition to a degree
  - Look at the most recent status of a client (Sept.)

Status of the Model



- High Precision…."Ok" Recall

- Only Relatively Small Proportion Defaults
  - Recall = 0.4 → 2640 / 6600 defaulters 👍
  - Accuracy = 0.83

- Livable Results

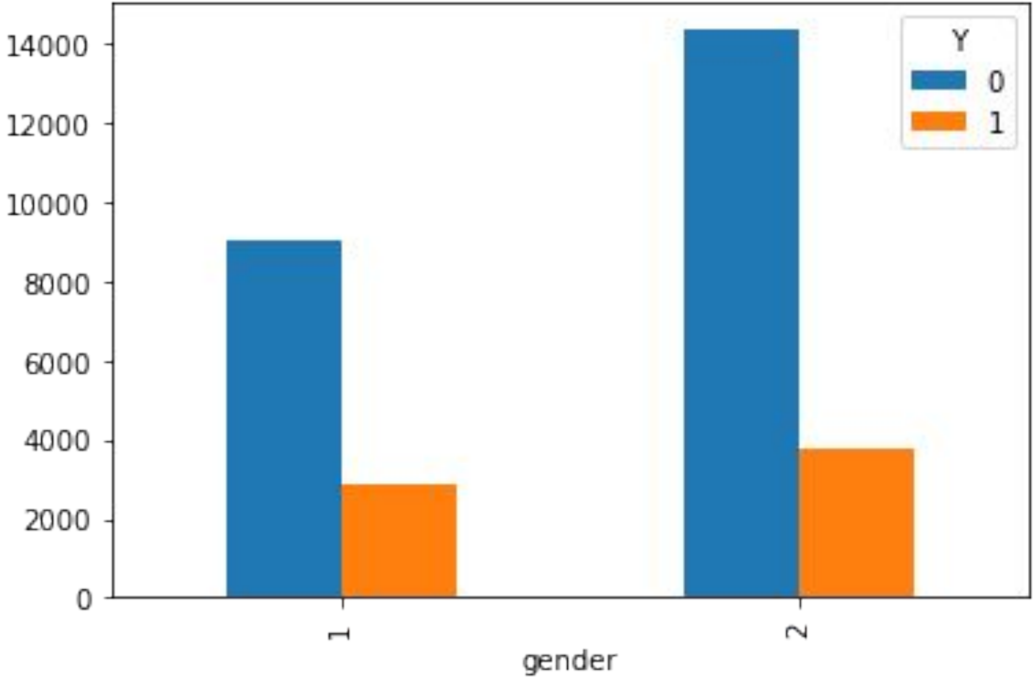
- Try to understand model's relation to the situation

Future Directions

- Model adequate for the situation
    - Enabled promising efforts to collect different types of data & improve model
- Fiscal features of debt, payment status do not guarantee a clear signal of defaulting
- Find a better generalized dataset

END

Defaulting & Gender



Thresholding

> 50% to default	
Metric	**Score**
Precision	0.72
Recall	0.40
F1	0.52
Accuracy	0.83

> 90% to default	
Metric	**Score**
Precision	1.00
Recall	0.006
F1	0.01
Accuracy	0.77

max_depth: 3 max_features: None n_estimators: 250

best_score: 0.4751666776850824

precision: 0.7243195785776997recall: 0.3997093023255814accuracy: 0.8274444444444444auc: 0.8279312993437229

[[6622 314] [1239 825]]

[0.06675498 0.00668585 0.02515955 0.01785101 0.06194356 0.09467239 0.02537071 0.02599007 0.02815825 0.0218619 0.02489656 0.09296947 0.06334256 0.06699342 0.04253396 0.04790901 0.0557756 0.04993573 0.03861271 0.03059331 0.02831242 0.04715636 0.03652061]

*bill_sep & pay_status_sep still most important features*next highest/moderate features were index[12,13] (bill_aug, bill_jul)*pay_status in aug&jul weren't strong features* ***good estimator is your bill amount*** lines up with common sense max_depth: 3 max_features: None n_estimators: 300

best_score: 0.47536320315804237

precision: 0.7271142109851787recall: 0.40406976744186046accuracy: 0.8285555555555556auc: 0.8321877193364805

[[6623 313] [1230 834]]

[0.06932896 0.00588784 0.0243248 0.01754372 0.05821014 0.0824601 0.02436829 0.023781   0.02670914 0.02050335 0.02443363 0.0887392 0.05895782 0.06925458 0.05120494 0.04954589 0.05408405 0.05525376 0.04267801 0.03387458 0.03438822 0.04473817 0.0397298]

*improving n_estimators to 300 keeping depth=3, using all features improved a bit*roc score not as good, but f1 improved, precision & accuracy seem to be steadily improving