Profit Curves
& Imbalanced Classes

Galvanize
Moses Marsh

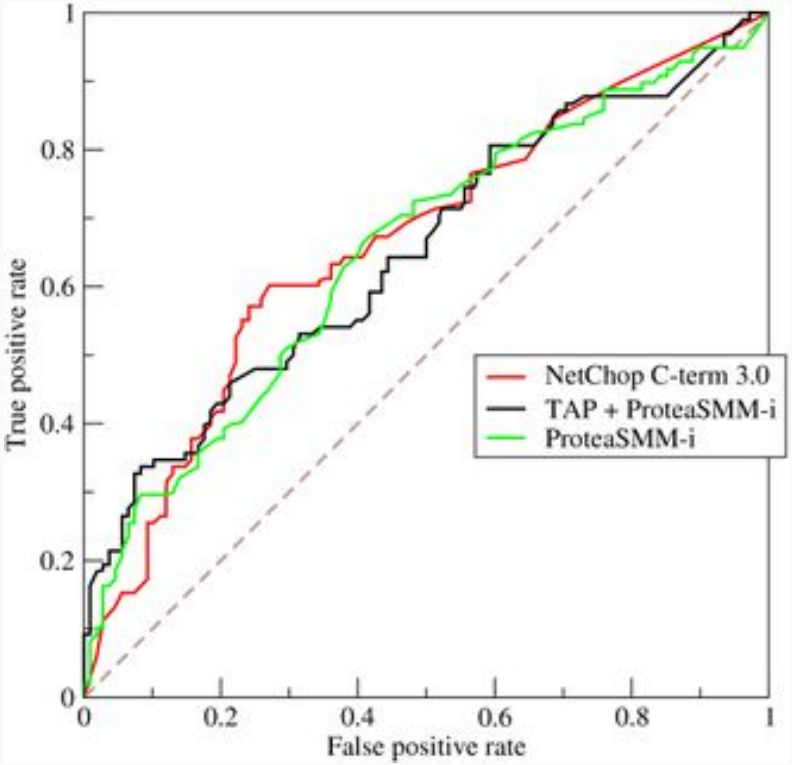
OBJECTIVES: answer the following

- How do we incorporate ***business costs*** into model evaluation?
- What is a ***cost-benefit*** matrix?
- How is a ***profit curve*** constructed?
- What are some issues with ***imbalanced classes***?
- How can the class balance be changed?

Review: ROC Curve, confusion matrix

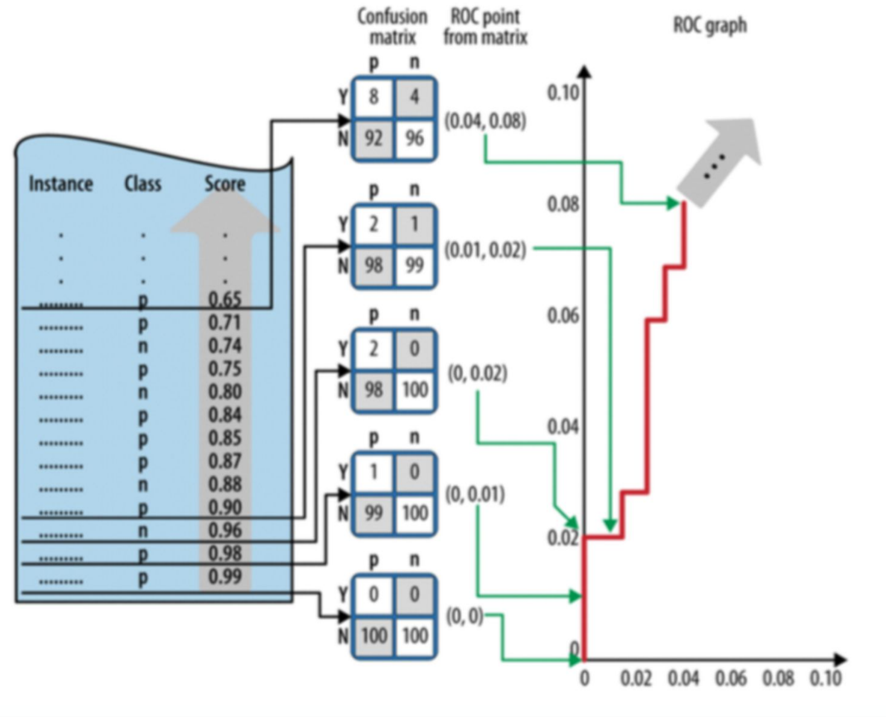
- ROC shows TPR (recall) vs FPR (1 - TNR) at different thresholds
- Confusion matrix:

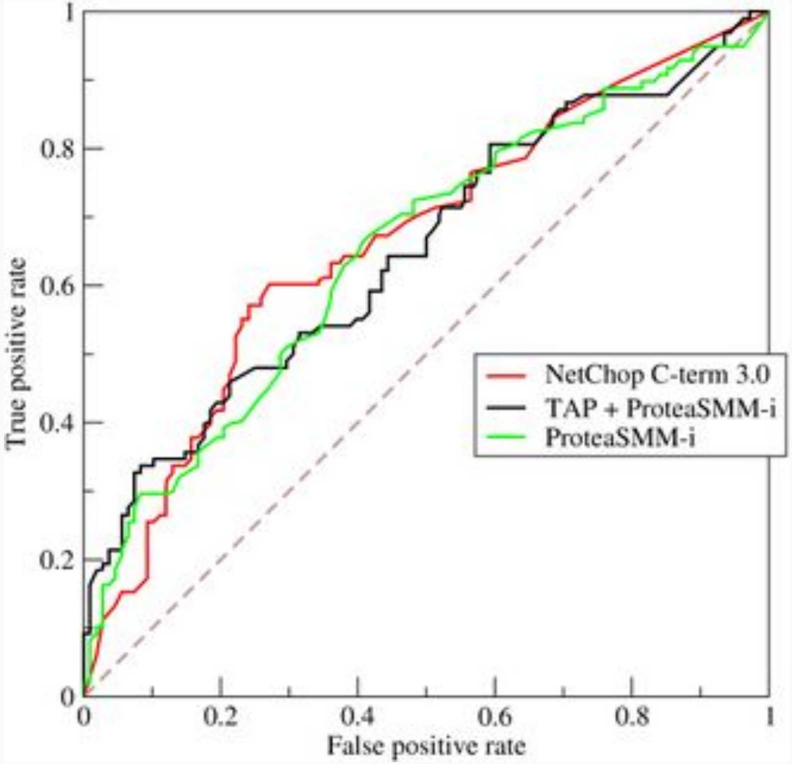
	Pred: Y	**Pred: N**
Actual: y	TP	FN
Actual: n	FP	TN



https://en.wikipedia.org/wiki/Confusion_matrix

Recall the ROC Curve:





Cost-Benefit Matrix

galvanize

- Let’s assign *dollar values* to true positives, false positives, true negatives, and false negatives
- For example, conside a credit card fraud model. If we think a charge is fraudulent, we will call the customer, which costs us \$5. If we catch a fraudulent charge, we save ourselves \$100. Since it still costs us the \$5 to call, we net \$95.

	Pred: Y	**Pred: N**
Actual: y	\$95	\$0
Actual: n	-\$5	\$0

Cost-Benefit Matrix

- Note that these costs are relative to a baseline model that never predicts fraud

	Pred: Y	**Pred: N**
Actual: y	$95	$0
Actual: n	-$5	$0

Expected profit

- Say we have the following confusion matrices. Which model is better?

	Pred: Y	**Pred: N**
Actual: y	114	11
Actual: n	307	818

	Pred: Y	**Pred: N**
Actual: y	47	78
Actual: n	21	1104

Expected profit

galvanize

- All we need to do is multiply the entries of the confusion matrix by their corresponding profits, sum the results, and divide by the total number of data points

	Pred: Y	**Pred: N**
Actual: y	114	11
Actual: n	307	818

Expected profit per transaction:

E(profit) = (114 * 95 + 307 * -5 + 11 * 0 + 818 * 0) / 1250
= 9295 / 1250
= 7.436

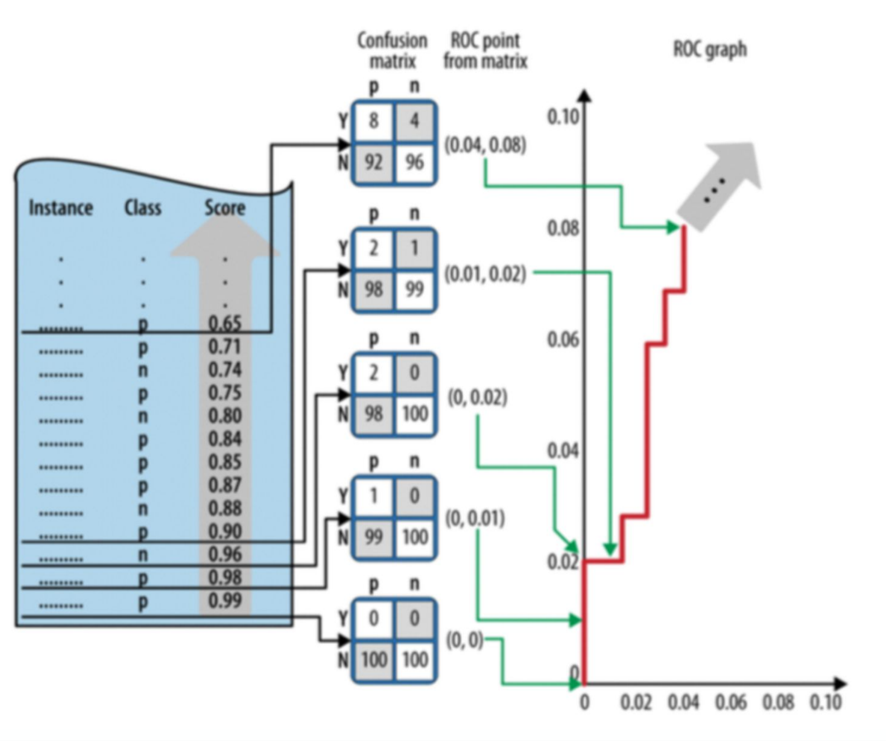
	Pred: Y	**Pred: N**
Actual: y	47	78
Actual: n	21	1104

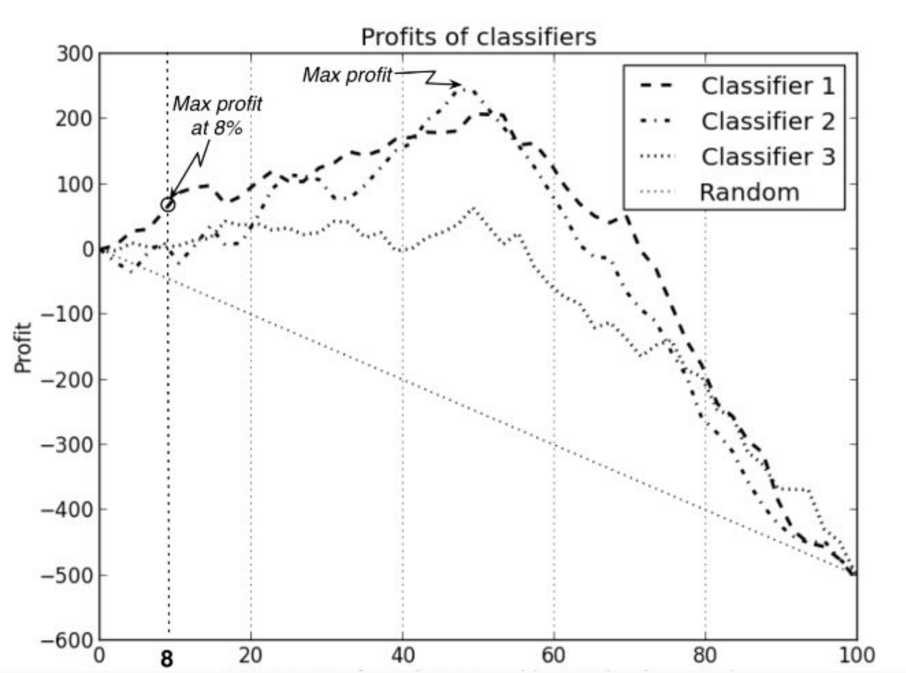
E(profit) = (47 * 95 + 21 * -5 + 78 * 0 + 1104 * 0) / 1250
= 4360 / 1250
= 3.488

From Thresholding to Profit Curves

- If we have a model that outputs probabilities, then *every threshold results in a different confusion matrix*
- Then we can assign a net profit per instance (data point) to every threshold
- And plot it!

From Thresholding to Profit Curves



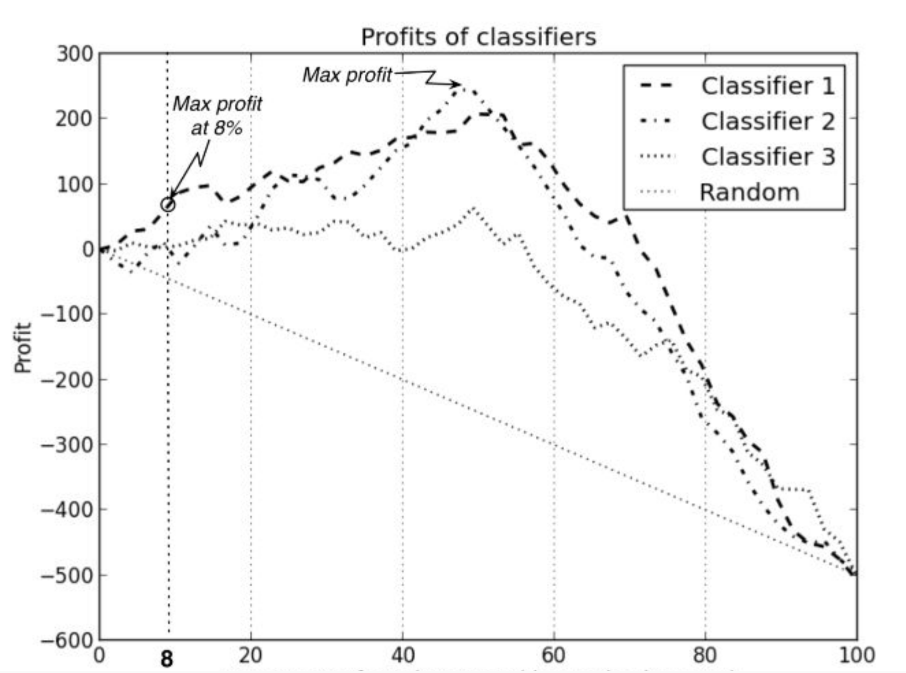


Percent of test instances classified as “positive”

Cost-sensitive Evaluation: comparing classifiers

Profit Curve:

- Same idea as ROC curve but with expected profit
- For each threshold, compute the expected profit



Percent of test instances classified as “positive”

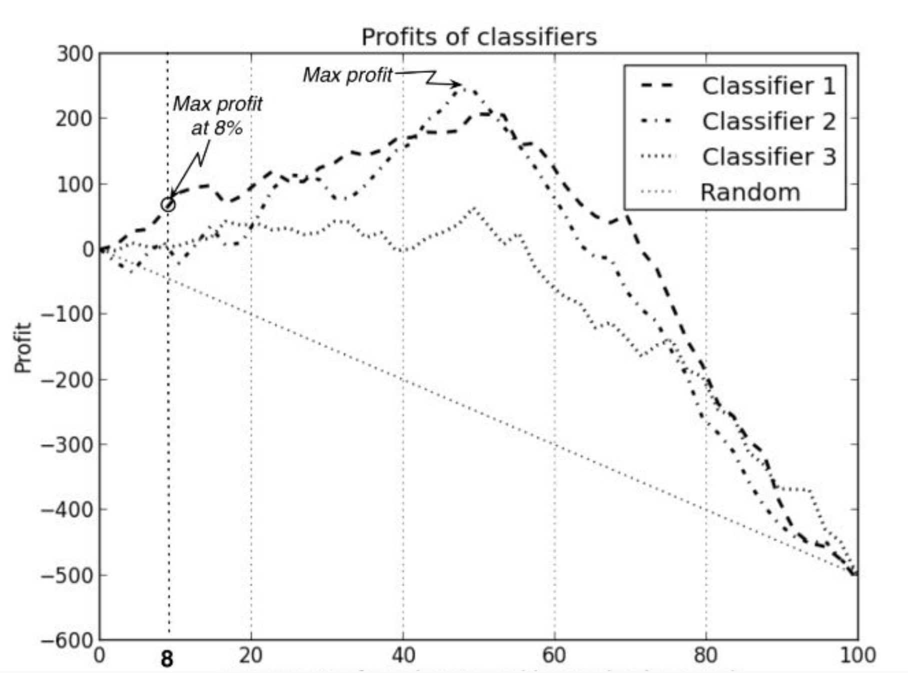
Cost-sensitive Evaluation: comparing classifiers

Profit Curve:

- Same idea as ROC curve but with expected profit
- For each threshold, compute the expected profit

Cost-sensitive evaluation:

- Select threshold with highest expected profit.



Percent of test instances classified as “positive”

QUESTION: how would you pick your favorite cost-benefit matrix ?

galvanize

A	Pred: pos	Pred: neg
Actual: pos	$12	$8
Actual: neg	$15	$965

B	Pred: pos	Pred: neg
Actual: pos	$0	$20
Actual: neg	$0	$980

C	Pred: pos	Pred: neg
Actual: pos	$15	$5
Actual: neg	$115	$865

D	Pred: pos	Pred: neg
Actual: pos	$18	$2
Actual: neg	$250	$730

QUESTION: how would you pick your favorite cost-benefit matrix ?

galvanize

A	Pred: pos	Pred: neg
Actual: pos	$12	$8
Actual: neg	$15	$965

B	Pred: pos	Pred: neg
Actual: pos	[illegible]	[illegible]
[illegible]	[illegible]	$980

C	[illegible]	[illegible]
[illegible]	$15	$5
Actual: neg	$115	$865

D	Pred: pos	Pred: neg
Actual: pos	$18	$2
Actual: neg	$250	$730

DEFINE A BUSINESS PROBLEM

QUESTION: how would you pick your favorite cost-benefit matrix ?

galvanize

A	Pred: pos	Pred: neg
Actual: pos	$12	$8
Actual: neg	[illegible]	[illegible]

B	Pred: pos	Pred: neg
Actual: pos	[illegible]	[illegible]
Actual: neg	[illegible]	$980

C	Pred: pos	Pred: neg
Actual: pos	[illegible]	$5
Actual: neg	$115	$865

[illegible]	[illegible]	[illegible]
Actual: pos	$18	[illegible]
Actual: neg	$250	$730

FORMALIZE COSTS AND BENEFITS

DEFINE A BUS[illegible]SS PROBLEM

Cost-Benefit Matrix (example 1)

Prompt: You are building a model to predict if credit card charges are fraudulent.

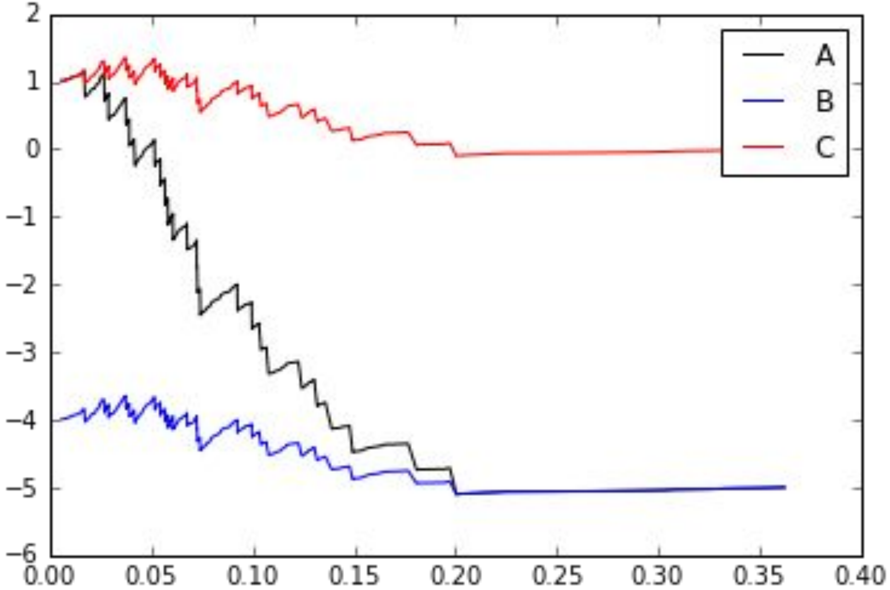
- If we predict a fraudulent charge, we'll call the customer to confirm.
- If you miss a fraudulent charge, it on average costs $100
- Calling someone to confirm if their charge was real costs on average $4

Question: What is an appropriate cost benefit matrix?

A	**Predicted: fraud**	**Predicted: not fraud**
Actual: fraud	$96	-$100
Actual: not fraud	-$4	$0

B	**Predicted: fraud**	**Predicted: not fraud**
Actual: fraud	-$4	-$100
Actual: not fraud	-$4	$0

C	**Predicted: fraud**	**Predicted: not fraud**
Actual: fraud	$96	$0
Actual: not fraud	-$4	$0



A	**Predicted: fraud**	**Predicted: not fraud**
Actual: fraud	$96	-$100
Actual: not fraud	-$4	$0

B	**Predicted: fraud**	**Predicted: not fraud**
Actual: fraud	-$4	-$100
Actual: not fraud	-$4	$0

C	**Predicted: fraud**	**Predicted: not fraud**
Actual: fraud	$96	$0
Actual: not fraud	-$4	$0

Cost-Benefit Matrix (example 2)

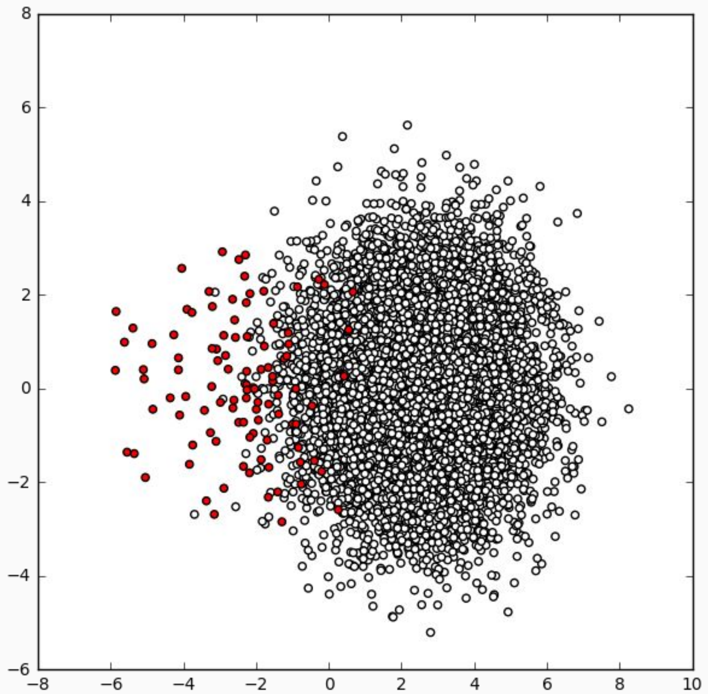
You are building a model to **predict if customers will churn** from your online clothing store.
You'll use your model **to send a promotional email** to users you think are going to churn.

You'd like to use a cost benefit matrix so you can build **profit curves to determine the optimal model**.

- Customers on average spend **$200/month**.
  Your profit is **10%** of this revenue.
- A promotional email costs on average **$2/customer** and prevents **50%** of users from churning for **6 months**.
- When the promotional email is sent to users who were not going to churn, it annoys **5%** of them and causes them to churn **2 months** earlier than they otherwise would have.

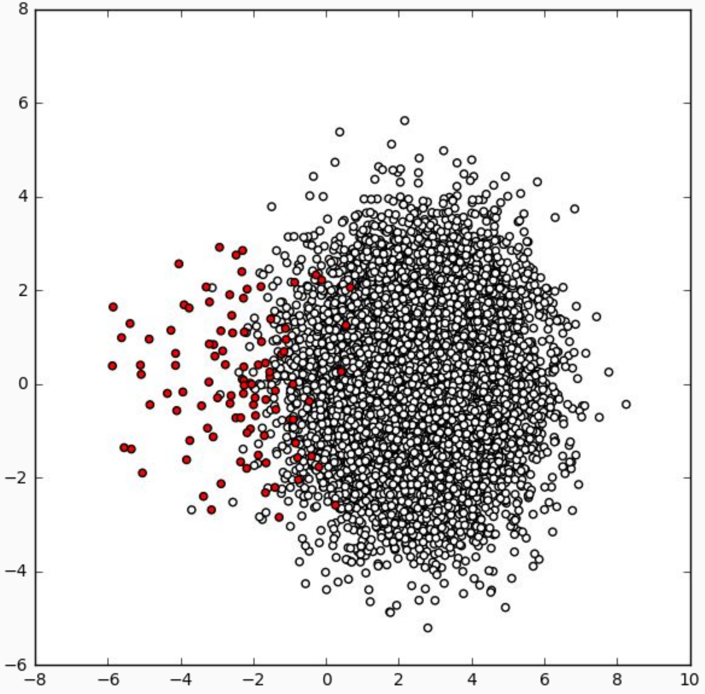
	Predicted: churn	**Predicted: not churn**
Actual: churn	?	?
Actual: Not churn	?	?

Imbalanced Classes



Problem Motivation

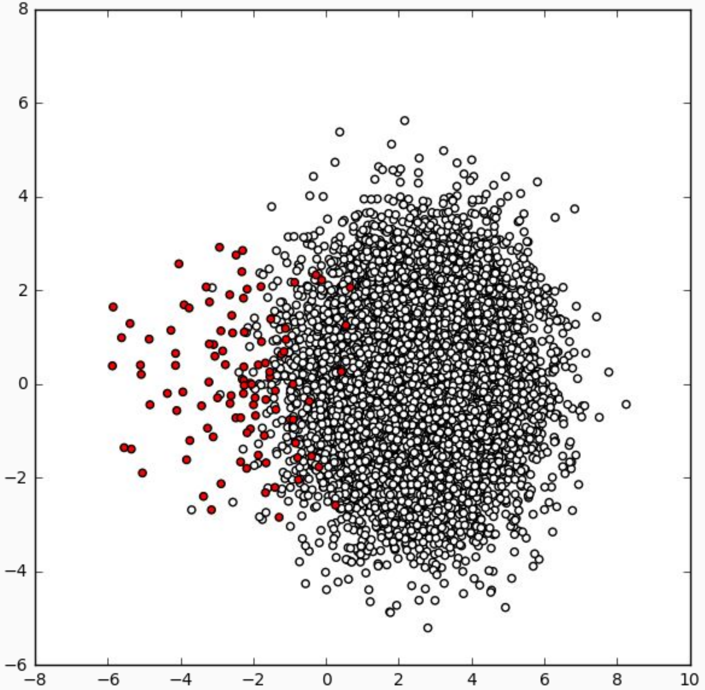
galvanize



Example : 100 pos, 10000 neg

- Classification datasets can be “imbalanced”.
  - i.e. many observations of one class, few of another
- Accuracy-driven models will over-predict the majority class.
- Costs of a false positive is often different from cost of a false negative.
  - e.g. missing fraud can be more costly than screening legitimate activity

Imbalanced Classes: failure analysis

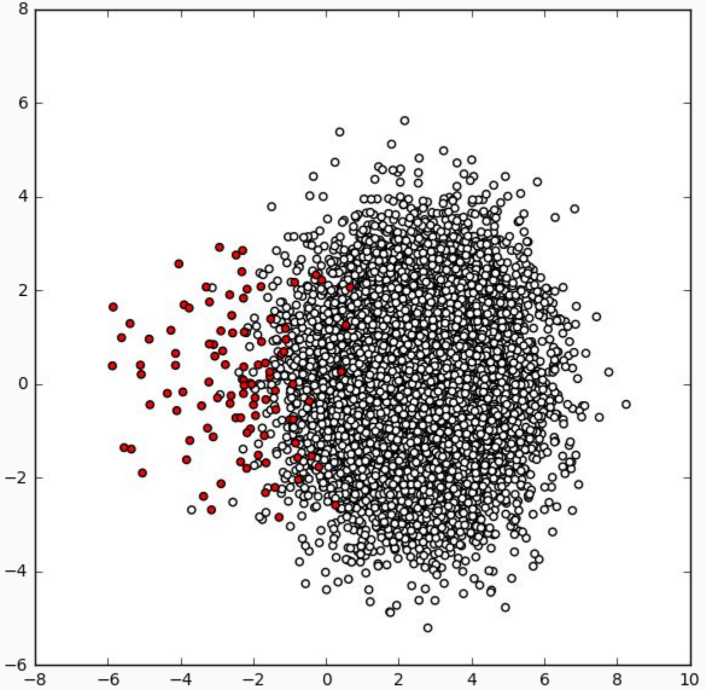


What’s a possible problem during LEARNING (fitting the model) ?

What’s a possible problem during EVALUATION (scoring the model) ?

Example : 100 pos, 10000 neg

Imbalanced Classes: failure analysis



Example : 100 pos, 10000 neg

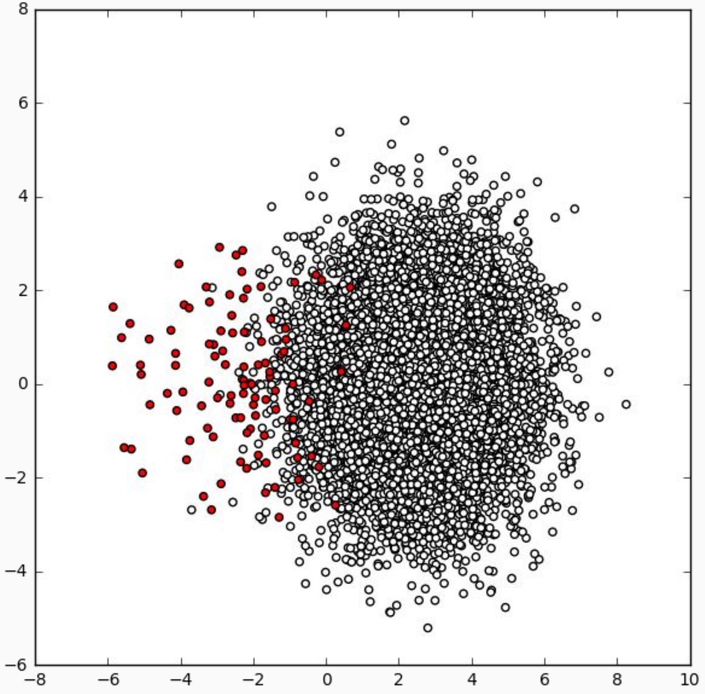
What's a possible problem during LEARNING (fitting the model) ?

The model will overpredict the majority class

What's a possible problem during EVALUATION (scoring the model) ?

False positives and false negatives may have different business costs

Imbalanced Classes: failure analysis



Example : 100 pos, 10000 neg

What's a possible problem during LEARNING (fitting the model) ?

Solution: cost-sensitive learning, oversampling/undersampling

What's a possible problem during EVALUATION (scoring the model) ?

Solution: cost-benefit matrix

Solutions

Cost-sensitive learning & evaluation:

- cost-benefit matrices & profit curves
- modified objective functions

Sampling:

- Oversampling
- Undersampling
- SMOTE - Synthetic Minority Oversampling TEchnique

- Models with explicit objective function can be modified to incorporate classification cost.
  - e.g. logistic regression

- Logistic regression's usual objective function:

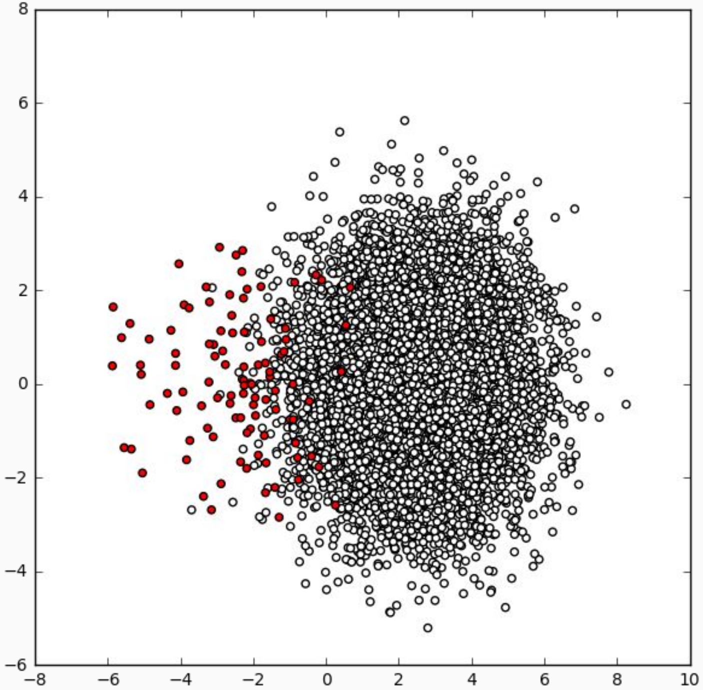
$$\ln p(\vec{y}|X;\theta) = \sum_{i=1}^{n} \left(y_i \ln h_\theta(x_i) + (1 - y_i) \ln(1 - h_\theta(x_i))\right)$$

- New objective function, representing expected cost:

$$J^c(\theta) = \frac{1}{N} \sum_{i=1}^{N} \Big(y_i (h_\theta(X_i) C_{TP_i} + (1 - h_\theta(X_i)) C_{FN_i}) + (1 - y_i)(h_\theta(X_i) C_{FP_i} + (1 - h_\theta(X_i)) C_{TN_i}) \Big).$$

- This will affect optimization.
  - e.g. cost-sensitive logistic regression is not convex!
- Not all models have a cost-sensitive implementation.

Imbalanced Classes: failure analysis



Example : 100 pos, 10000 neg

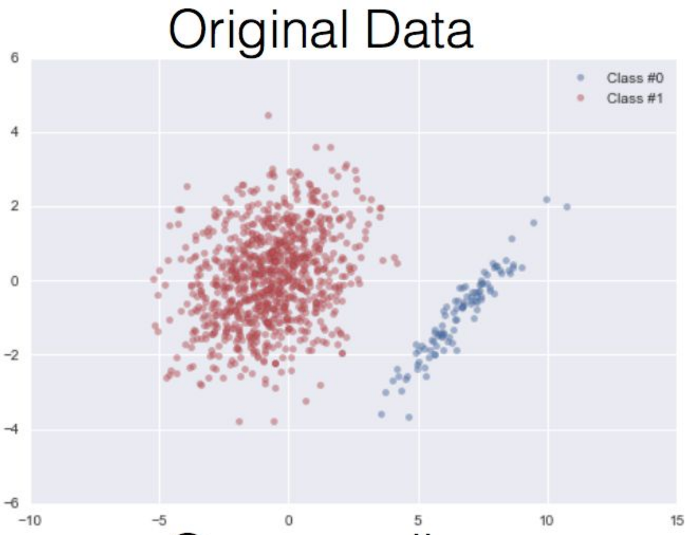
What's a possible problem during LEARNING (fitting the model) ?

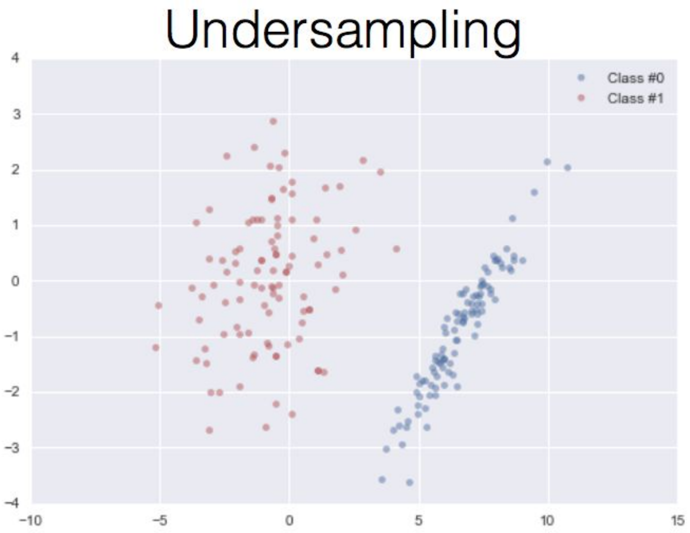
Solution: cost-sensitive learning

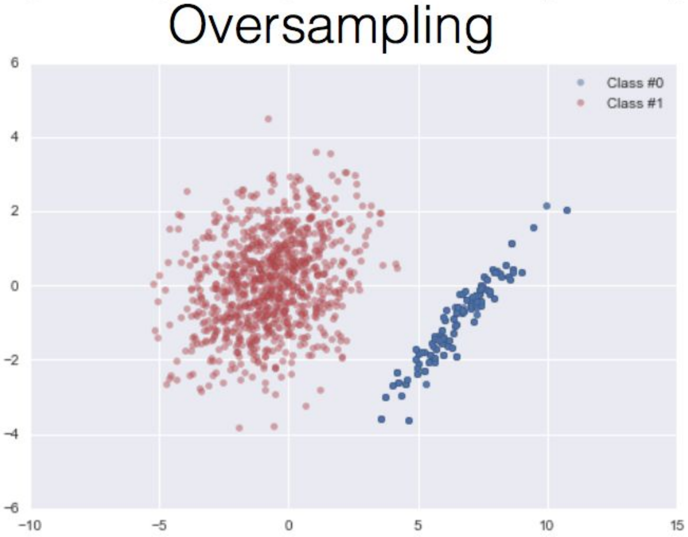
Solution: Train your model on oversampled/undersampled data

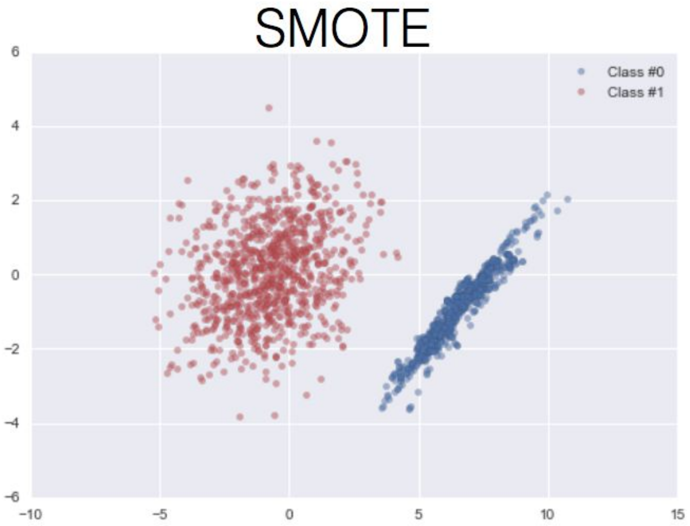
What's a possible problem during EVALUATION (scoring the model) ?

Solution: cost-benefit matrix


Original Data
Class #0
Class #1
6
4
2
0
-2
-4
-6
-10
-5
0
5
10
15


Undersampling
Class #0
Class #1
4
3
2
1
0
-1
-2
-3
-4
-10
-5
0
5
10
15


Oversampling
Class #0
Class #1
6
4
2
0
-2
-4
-6
-10
-5
0
5
10
15

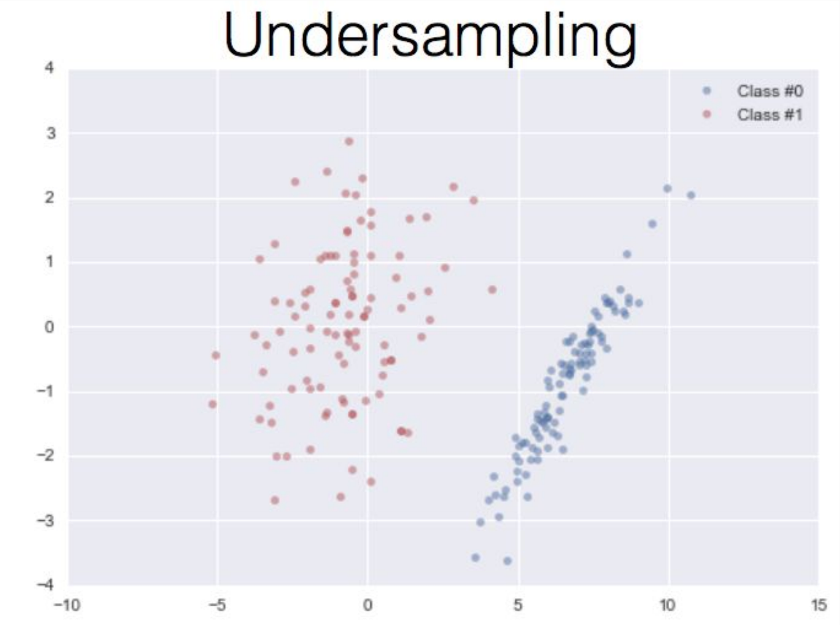

SMOTE
Class #0
Class #1
6
4
2
0
-2
-4
-6
-10
-5
0
5
10
15

Undersampling

Undersampling randomly discards majority class observations to balance training sample.

PRO: Reduces runtime on very large datasets.

CON: Discards potentially important observations.

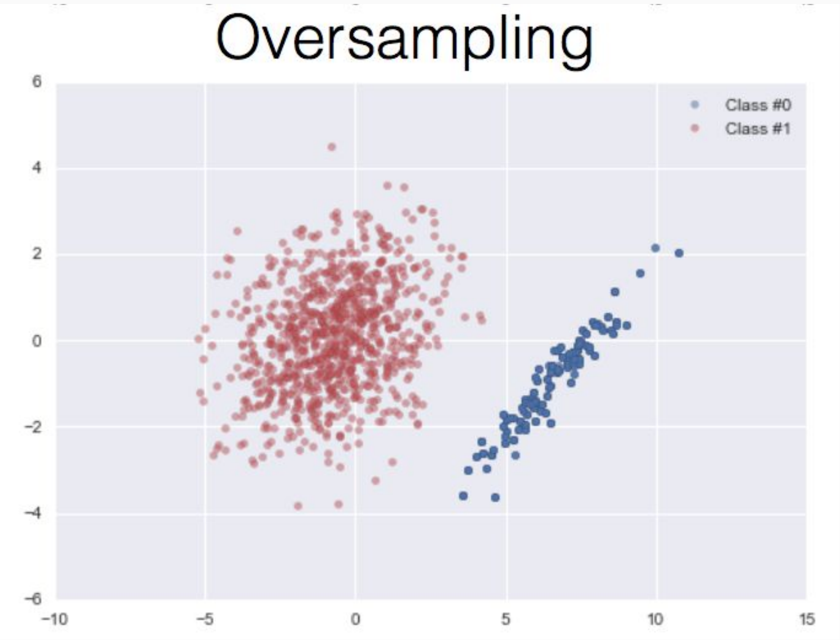


Oversampling

Oversampling replicates observations from minority class to balance training sample.

PRO: Doesn't discard information.

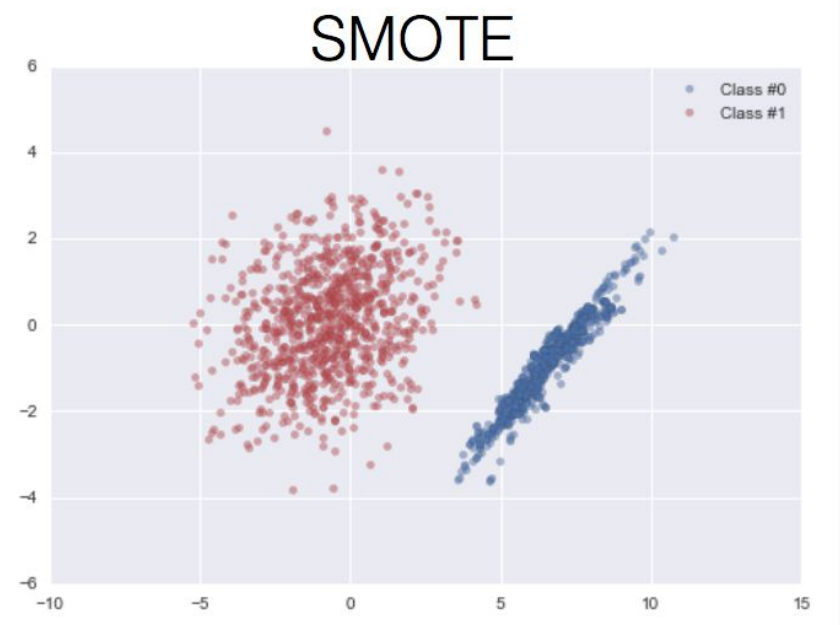
CON: Likely to overfit.



SMOTE - Synthetic Minority Oversampling TEchnique

Generates new observations from minority class.

For each minority class observation and for each feature, randomly generate between it and one of its k-nearest neighbors.



SMOTE pseudocode

```
synthetic_observations = []
while len(synthetic_observations) + len(minority_observations) < target:
    obs = random.choice(minority_observations):
    neighbor = random.choice(kNN(obs, k)) # randomly selected neighbor
    new_observation = {}
    for feature in obs:
        weight = random() # random float between 0 and 1
        new_feature_value = weight*obs[feature] \
                            + (1-weight)*neighbor[feature]
        new_observation[feature] = new_feature_value
    synthetic_observations.append(new_observation)
```

Sampling Techniques

What's the right amount of over-/under-sampling?

- The degree & kind of resampling form another set of hyperparameters to tune
- Mix it up! You may get the best results by both oversampling and undersampling
- Evaluation: ***profit*** if you have a cost-benefit matrix, otherwise ***ROC-AUC*** score, ***F1***, etc.

Cost Sensitivity vs Sampling

- Neither is strictly superior.
- Oversampling tends to work better than undersampling on small datasets.
- Some algorithms don't have an obvious cost-sensitive adaptation, requiring sampling.

See also "Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?" http://storm.cis.fordham.edu/gweiss/papers/dmin07-weiss.pdf