Profit Curves
& Imbalanced Classes

Galvanize
Moses Marsh

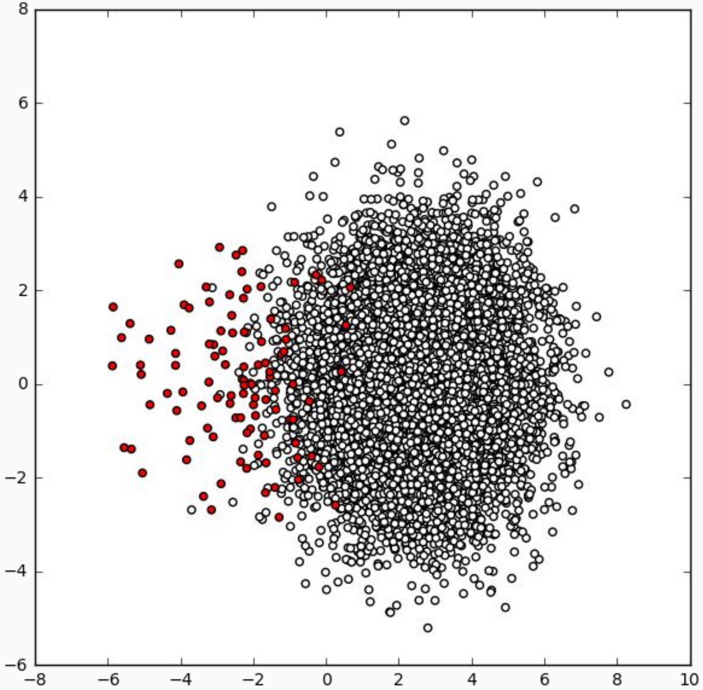
OBJECTIVES

- **Discuss** and give examples of the issues with imbalanced classes.
- **Explain** and **implement** the profit curve method.
- **Explain** cost sensitive learning and how it deals with imbalanced classes.
- **Define**, give examples and relate sampling methods.

Problem Motivation

- Classification datasets can be “imbalanced”.
  - i.e. many observations of one class, few of another
- 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
- Accuracy-driven models will over-predict the majority class.

Imbalanced Classes: failure analysis

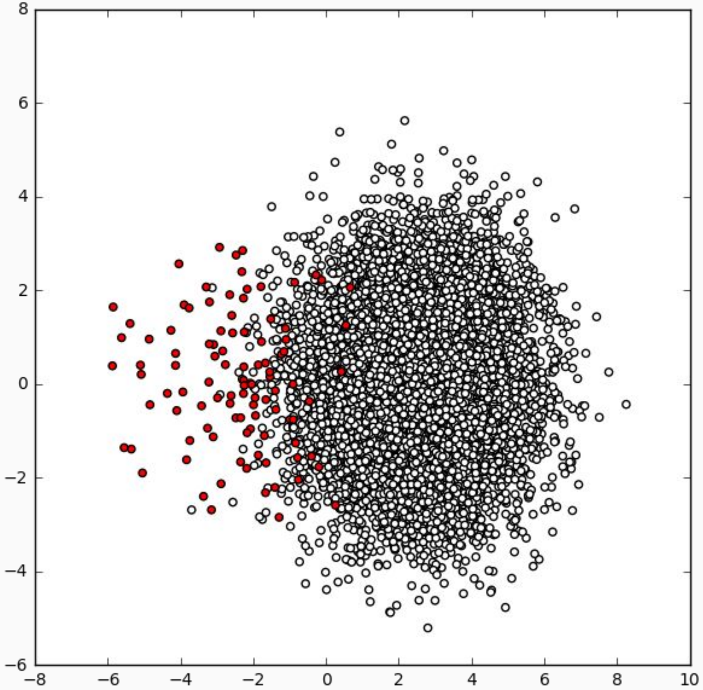


Example : 100 pos, 10000 neg

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

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

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:

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

Sampling:

- Oversampling
- Undersampling
- SMOTE - Synthetic Minority Oversampling TEchnique

QUESTION: how would you pick your favorite matrix ?

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 matrix ?

galvanize

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

B	Pred: pos	Pred: neg
Actual: pos	0	[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 matrix ?

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

B	Pred: pos	Pred: neg
Actual: pos	0	[illegible]
[illegible]	[illegible]	980

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

[illegible]	[illegible]	[illegible]
Actual: pos	18	2
Actual: neg	250	730

FORMALIZE COSTS AND BENEFITS

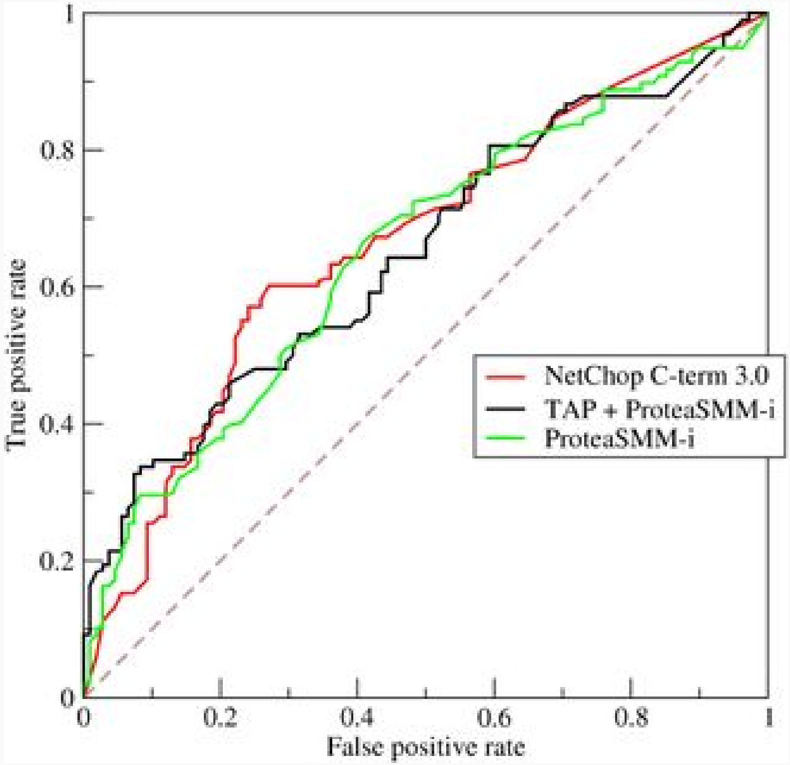
DEFINE A BUS[illegible]SS PROBLEM

Recall the ROC Curve:

- ROC shows FPR = (1-TNR) vs TPR (aka Recall)
- doesn't give preference to one over the other

Q: How to handle unequal error costs?

A: Assign a cost/profit to each type of error/success



Confusion Matrix

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

Confusion Matrix

P = TP+FN = count of actual y
N = FP+TN = count of actual n

VALUES ARE COUNTS

Probability Matrix

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

Confusion Matrix

P = TP+FN = count of actual y
N = FP+TN = count of actual n

VALUES ARE COUNTS

	Pred: Y	Pred: N
Actual: y	p(Y,y)	p(N,y)
Actual: n	p(Y,n)	p(N,n)

Probability Matrix

p(Y,y) = TP / (P + N)
p(Y,n) = FP / (P + N)
p(N,y) = FN / (P + N)
p(N,n) = TN / (P + N)

VALUES ARE PROBABILITIES

Cost-Benefit Matrix

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

Confusion Matrix

P = TP+FN = count of actual y
N = FP+TN = count of actual n

VALUES ARE COUNTS

	Pred: Y	Pred: N
Actual: y	p(Y,y)	p(N,y)
Actual: n	p(Y,n)	p(N,n)

Probability Matrix

p(Y,y) = TP / (P + N)
p(Y,n) = FP / (P + N)
p(N,y) = FN / (P + N)
p(N,n) = TN / (P + N)

VALUES ARE PROBABILITIES

	Pred: Y	Pred: N
Actual: y	b(Y,y)	c(N,y)
Actual: n	c(Y,n)	b(N,n)

Cost-Benefit Matrix

VALUES ARE $$$!

Computing the Expected Profit

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

	Pred: Y	Pred: N
Actual: y	p(Y,y)	p(N,y)
Actual: n	p(Y,n)	p(N,n)

	Pred: Y	Pred: N
Actual: y	b(Y,y)	c(N,y)
Actual: n	c(Y,n)	b(N,n)

$$
\begin{aligned}
E[Profit] &= p(Y,y).\,b(Y,y) + p(Y,n).\,c(Y,n) \\
&\quad + p(N,y).\,c(N,y) + p(N,n).\,b(N,n) \\
&= p(Y \mid y).\,p(y).\,b(Y,p) + p(Y \mid n).\,p(n).\,c(Y,n) \\
&\quad + p(N \mid y).\,p(y).\,c(N,y) + p(N \mid n).\,p(n).\,b(N,n) \\
&= p(y).\,[p(Y \mid y).\,b(Y,p) + p(N \mid y).\,c(N,y)] \\
&\quad + p(n)\,[p(Y \mid n).\,c(Y,n) + p(N \mid n).\,b(N,n)]
\end{aligned}
$$

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

Cost-Benefit Matrix (example 2)

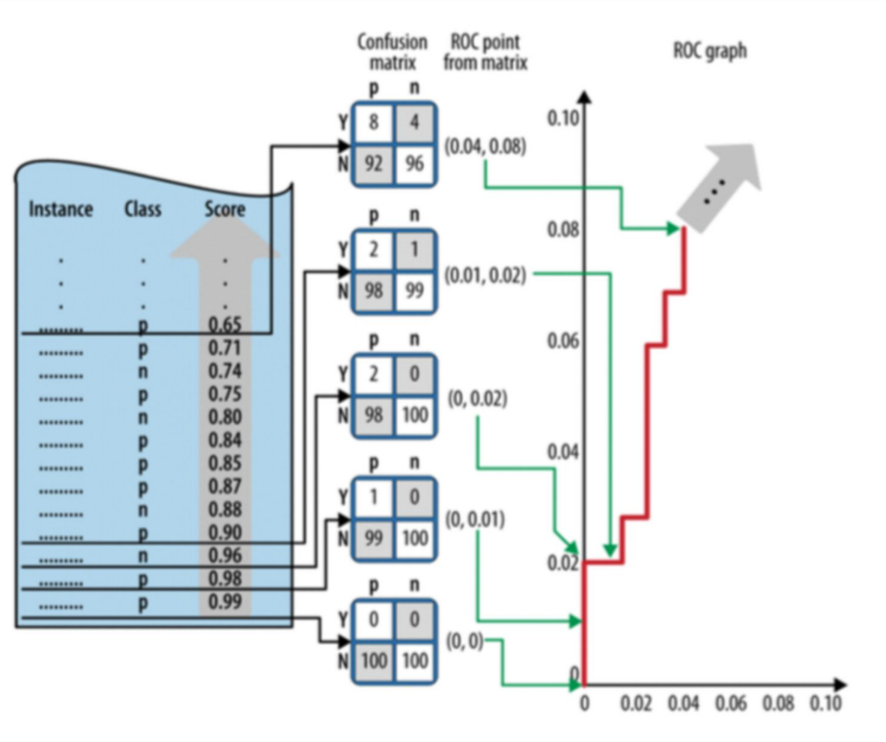
You are building a model to **predict if customers will churn** from your online clothing store.
You'll use your model **to send a promotional email** to users you think are going to churn.

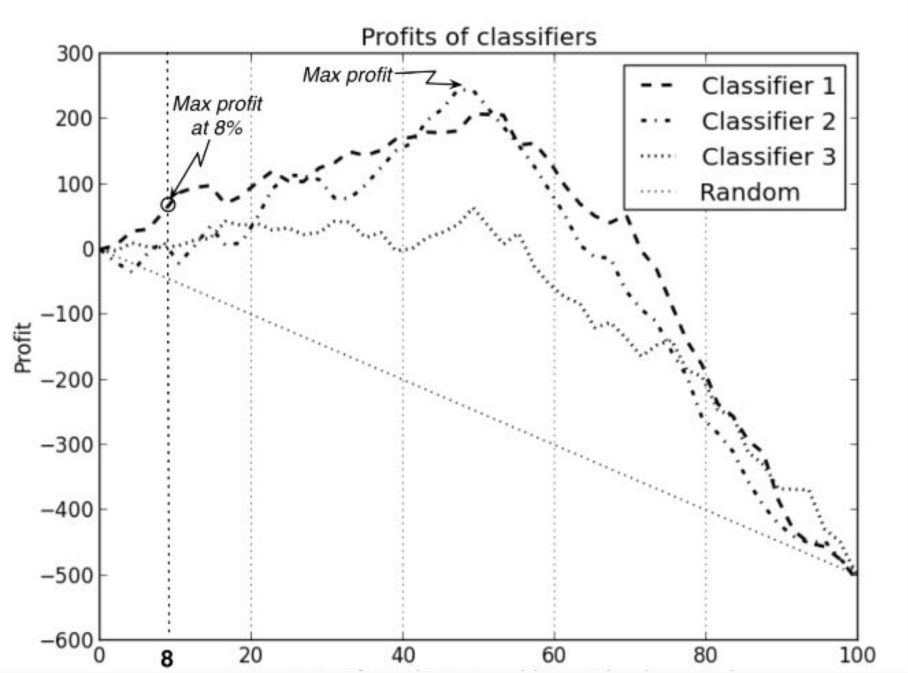
You'd like to use a cost benefit matrix so you can build **profit curves to determine the optimal model**.

- 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	?	?

From Thresholding to Profit Curves

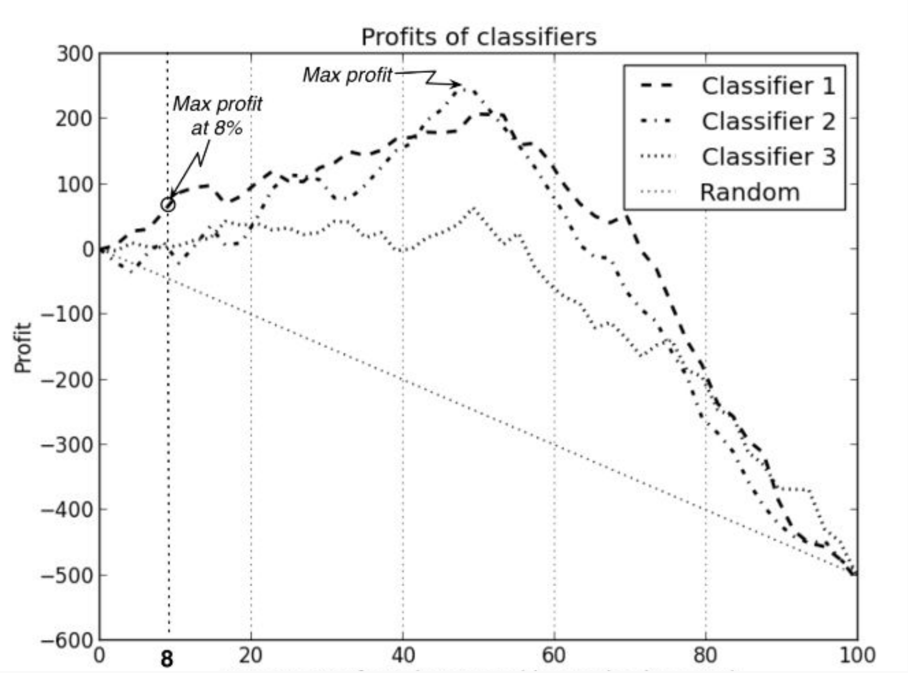




Percent of test instances classified as “positive”

Profit Curve:

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



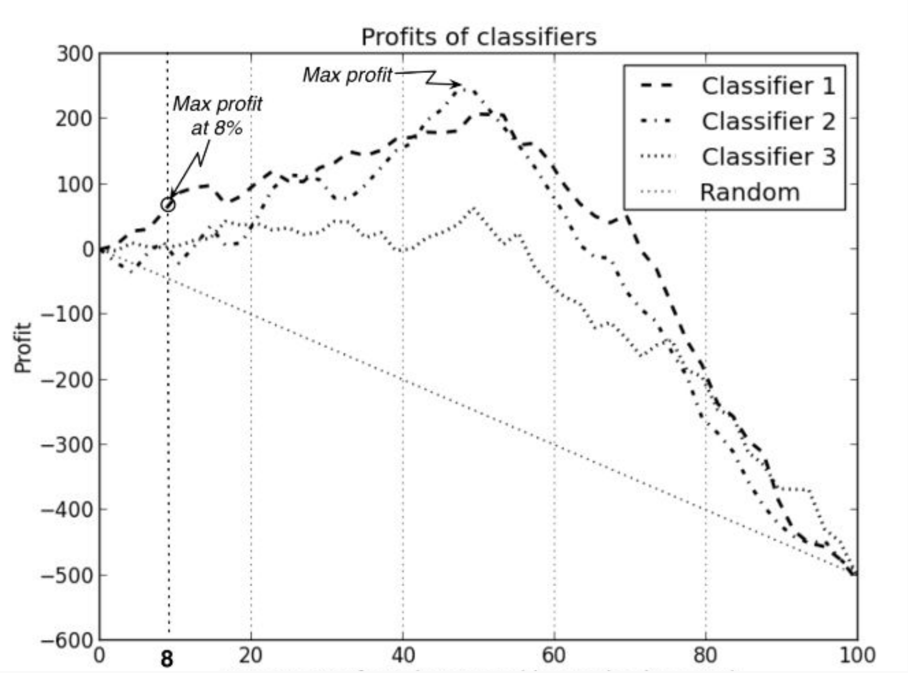
Percent of test instances classified as “positive”

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”

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

- Logistic regression's usual objective function:

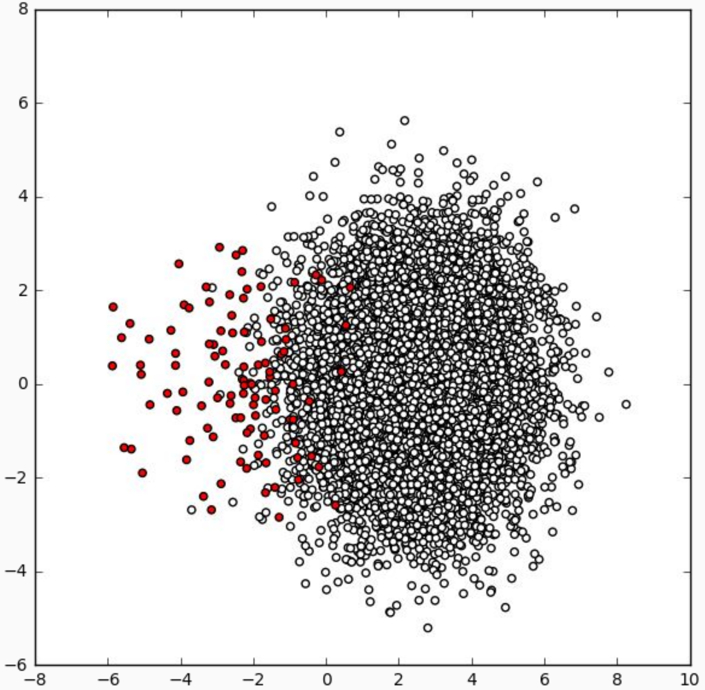
$$\ln p(\vec{y}|X;\theta) = \sum_{i=1}^{n} \left(y_i \ln h_\theta(x_i) + (1-y_i)\ln(1-h_\theta(x_i))\right)$$

- 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

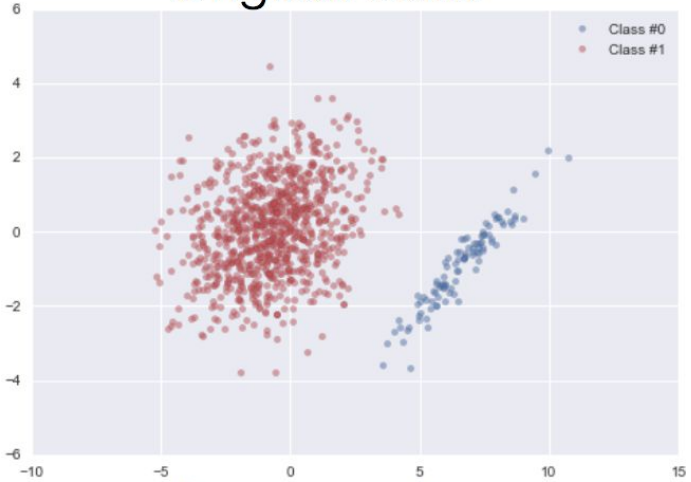
What's a possible problem during LEARNING (fitting the model) ?

Solution: cost-sensitive learning, oversampling/undersampling

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

Solution: cost-benefit matrix

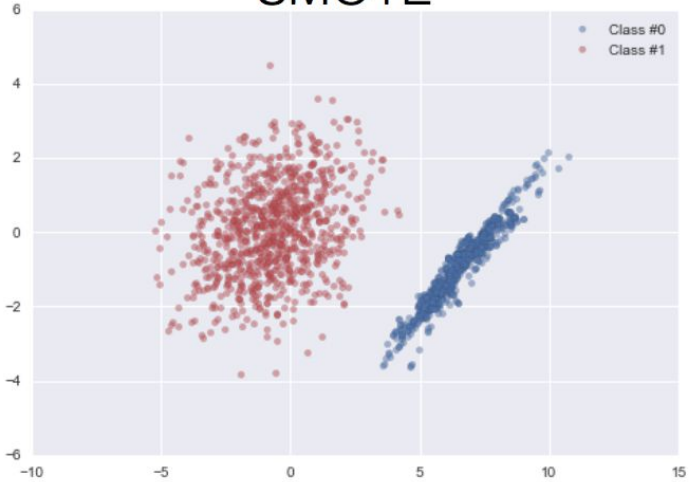
Original Data



Undersampling

Oversampling

SMOTE

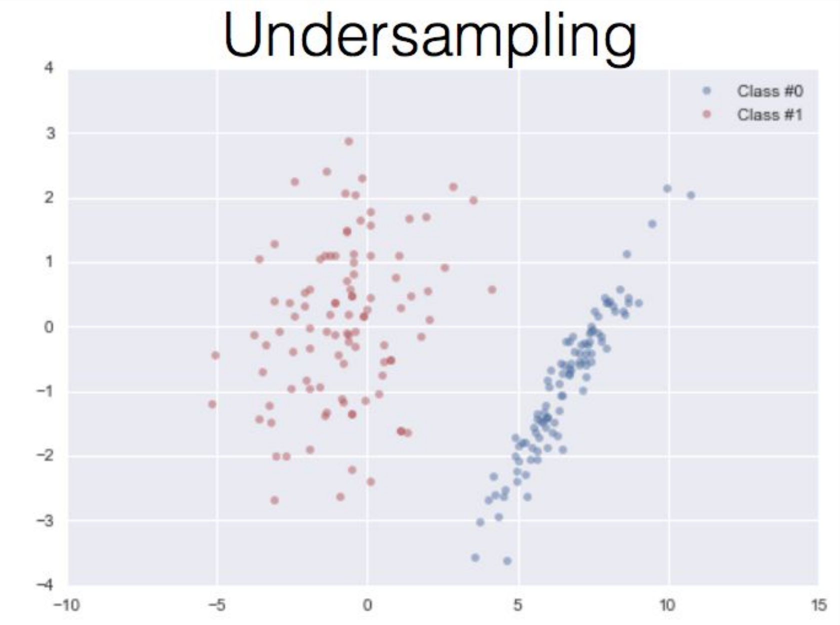


Undersampling

Undersampling randomly discards majority class observations to balance training sample.

PRO: Reduces runtime on very large datasets.

CON: Discards potentially important observations.

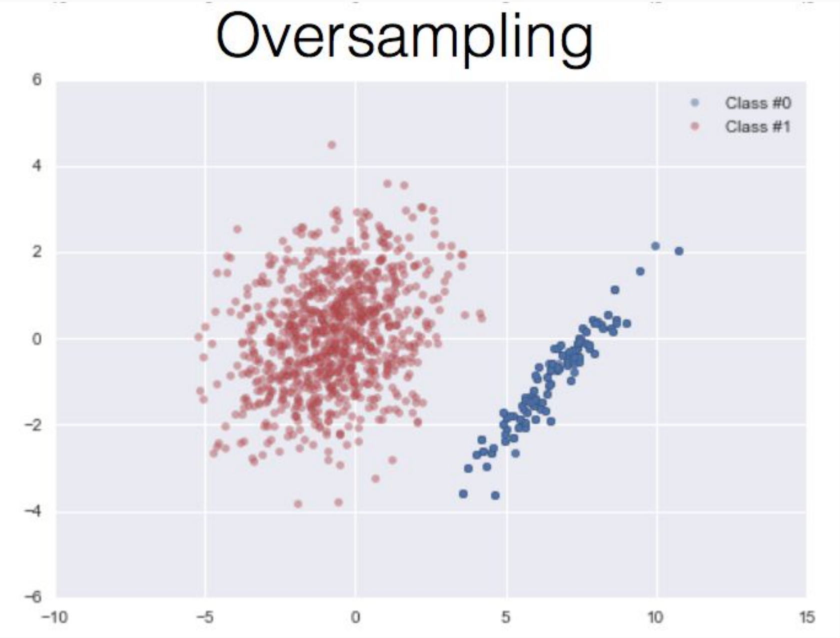


Oversampling

Oversampling replicates observations from minority class to balance training sample.

PRO: Doesn't discard information.

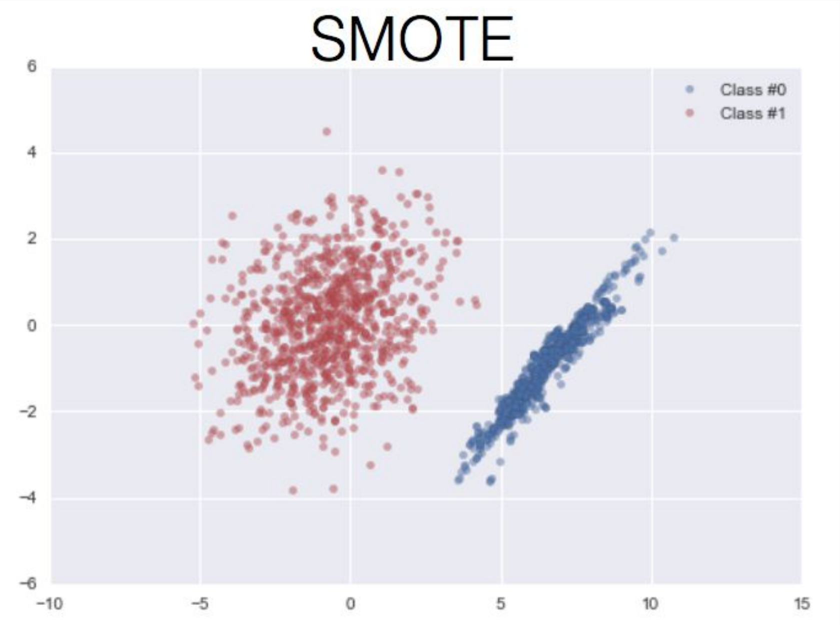
CON: Likely to overfit.



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 is 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