SMOTE: Synthetic Minority Over-sampling Technique

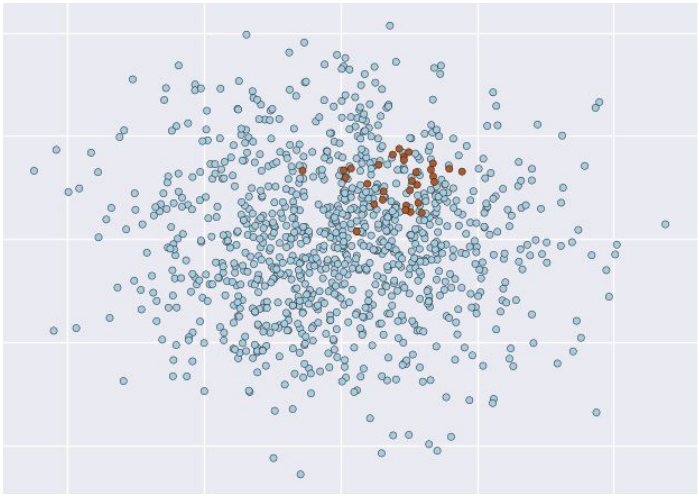
Review of Chawla et al 2002, by the TELUS digital AI team

Why our interest...

- Common problem that we face here at TELUS
  - Online fraudulent orders (~2-3% of all orders)
  - Customer churn (~0.9% of customers)

- Modelling very challenging
  - Interested in modelling and predicting these rare occurrences
  - Cannot rely on model accuracy -> favours majority class
  - Minimize the impact by optimizing model for alternative metrics, e.g. precision, recall, F1 score, AUC ROC
    - But generally doesn't change how model learns

https://www.svds.com/learning-imbalanced-classes/

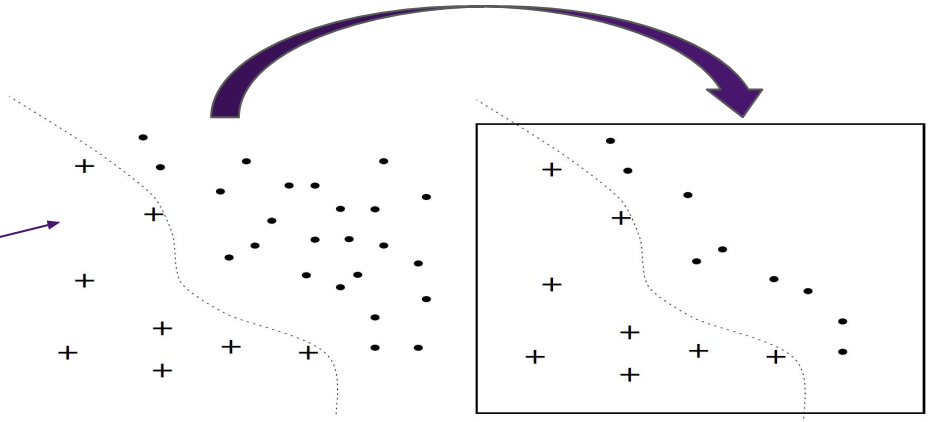


- Inspiration for generating synthetic data
  - Often encounter challenges for using/storing/accessing data due to privacy and security concerns
    - Intend to generate synthetic datasets for testing/training models
    - Mix of continuous and many nominal fields

Alternatives (at that time)

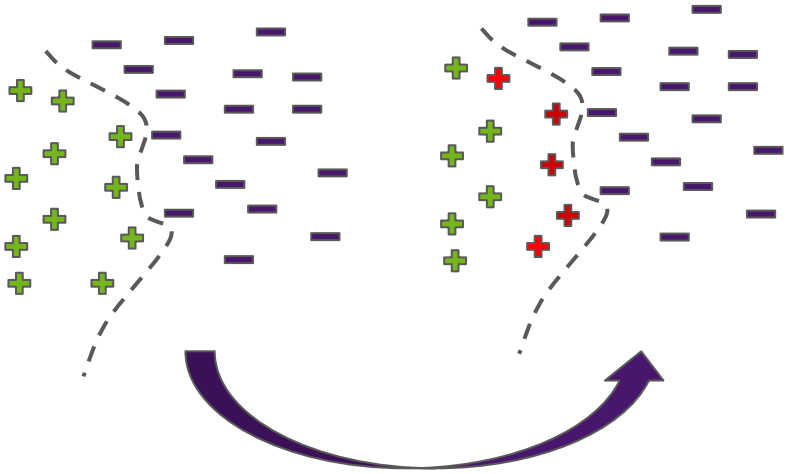
Kubat & Matwin 1998

- Under-sampling
- Selective under-sampling (One-sided selection; *Kubat & Matwin 1997*)
- SHRINK (*Kubat et al 1998*)
  - re-classifies overlapping region as minority class
- Focused resampling (*Japkowicz 2000*)
  - Resampling only the minority class that occurred on the boundary
- Combined over-sampling and under-sampling (e.g. *Ling and Li 1998; Solberg & Solberg 1996*)



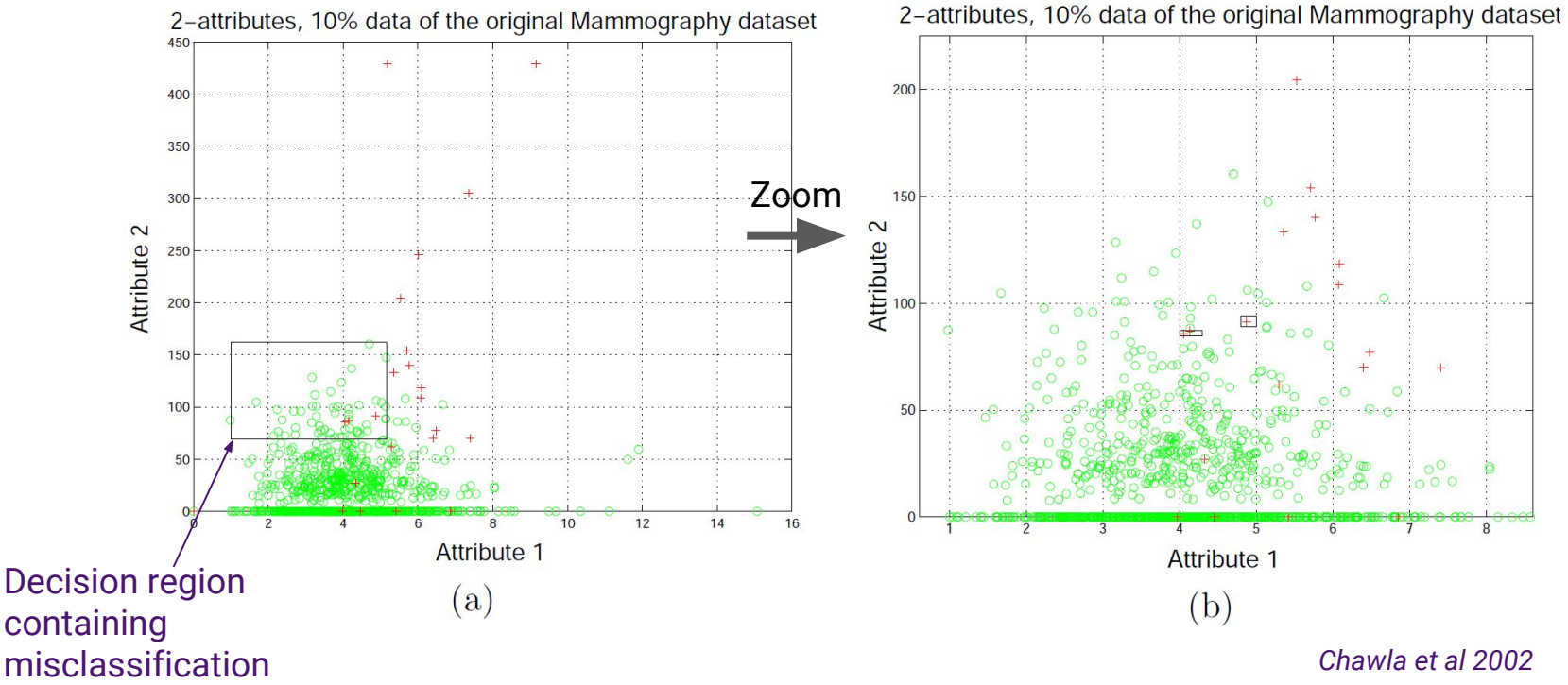
Remove majority examples far from boundary

Over-sample minority class close to boundary



Example of poor performance of over-sampling

- Minority over-sampling with replacement doesn't significantly improve class recognition (*Ling & Li 1998*; *Japkowicz 2000*)



Chawla et al 2002

- **Conclusion:** Over-sampling leads to over-fitting

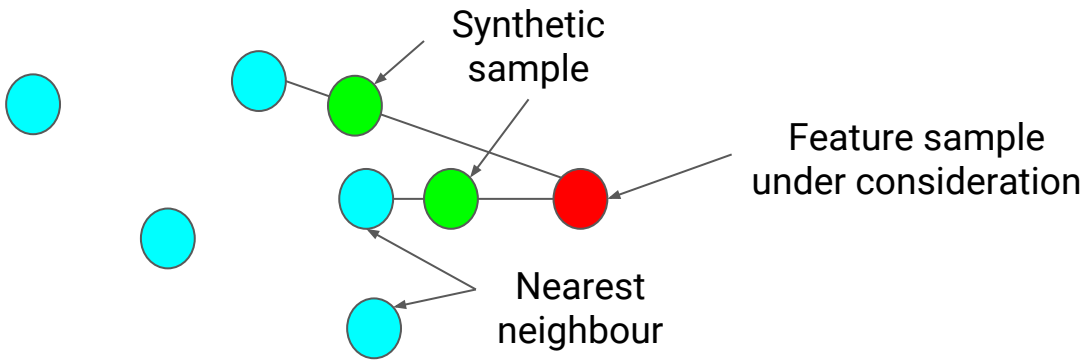
SMOTE algorithm

- Inspired by *Ha & Bunke 1997*
  - Generated new handwritten character recognition samples through rotation and skew transformations
- Create new samples in continuous "feature space"

by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors.

- **Steps**:
  1. Take difference between sample and nearest neighbour
  2. Multiply by random number between 0 and 1
  3. Add this difference to sample to generate new feature along the line segment
  4. Continue on with next nearest neighbour up to user-defined n neighbours

Conclusion: Causes classifier to find larger and less specific decision regions



Experiments

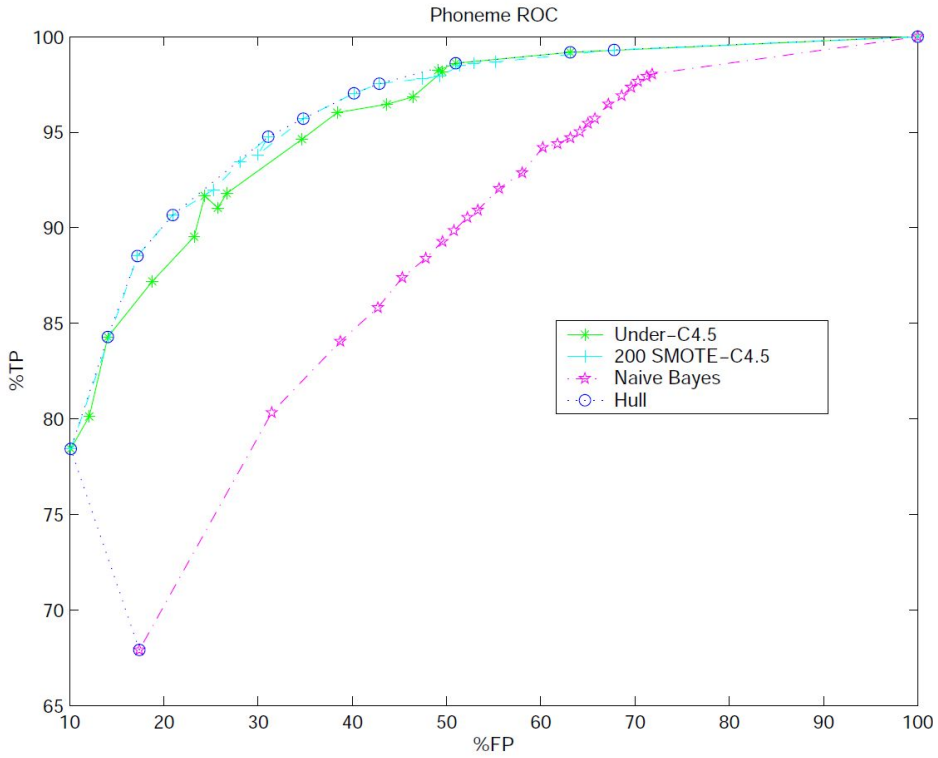
- Combination of under-sampling and 'SMOTING' on different datasets
- Used C4.5 (decision tree classifier; *Quinlan 1992*), Ripper (Rule based learner; *Cohen 1995*), Naive Bayes Classifier

Dataset	Majority Class	Minority Class	Minority ratio
Pima	500	268	0.35
Phoneme	3818	1586	0.29
Adult	37155	11687	0.24
E-state	46869	6351	0.12
Satimage	5809	626	0.10
Forest Cover	35754	2747	0.07
Oil	896	41	0.04
Mammography	10923	260	0.02
Can	435512	8360	0.02

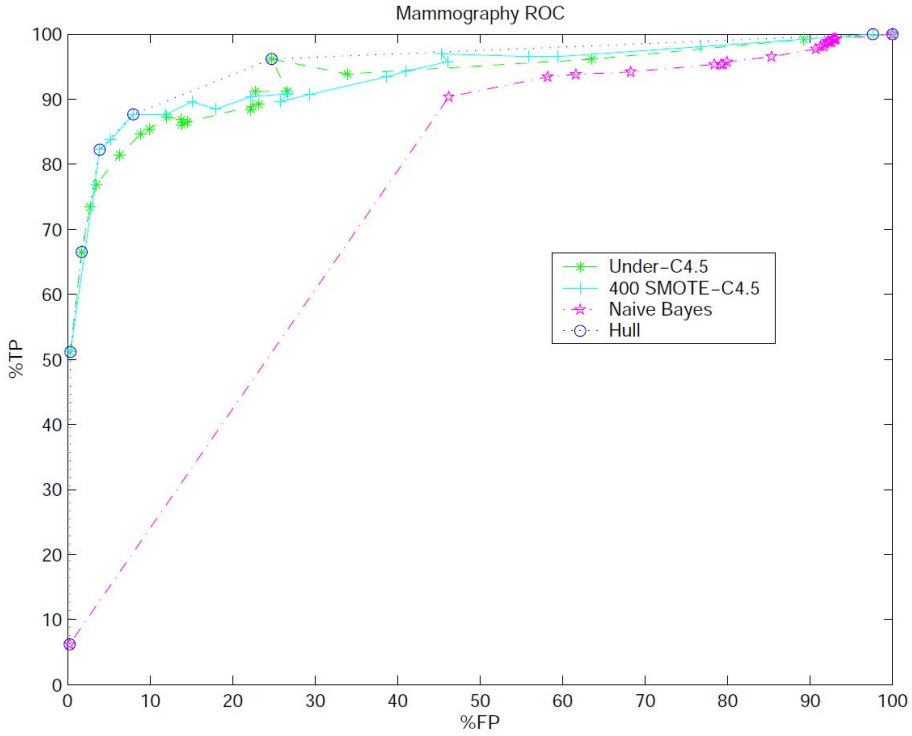
Results

- Generated
  - using 10-fold cross-validation
  - Over-sampled minority class to certain degree and then varied amount of under-sampling of majority class

Chawla et al 2002



Original minority class ratio 0.35



Original minority class ratio 0.02

Results

Dataset	Under	50 SMOTE	100 SMOTE	200 SMOTE	300 SMOTE	400 SMOTE	500 SMOTE
Pima	7242		**7307**				
Phoneme	8622		8644	**8661**			
Satimage	8900		8957	**8979**	8963	8975	8960
Forest Cover	9807		9832	9834	**9849**	9841	9842
Oil	8524		8523	8368	8161	8339	**8537**
Mammography	9260		9250	9265	9311	**9330**	9304
E-state	6811		6792	**6828**	6784	6788	6779
Can	9535	**9560**	9505	9505	9494	9472	9470

Table 3: AUC's [C4.5 as the base classifier] with the best highlighted in bold.

Alternative forms

SMOTE-NC (nominal continuous)

- Extended version of SMOTE to also handle nominal features
- Modified algorithm
  1. Compute median of standard deviations of all continuous features for minority class
  2. If nominal features differ between sample and its potential nearest neighbours, then median included in Euclidean distance computation as penalty to match typical difference in continuous feature values
  3. Nominal feature given value occuring in the majority of its k-nearest neighbours

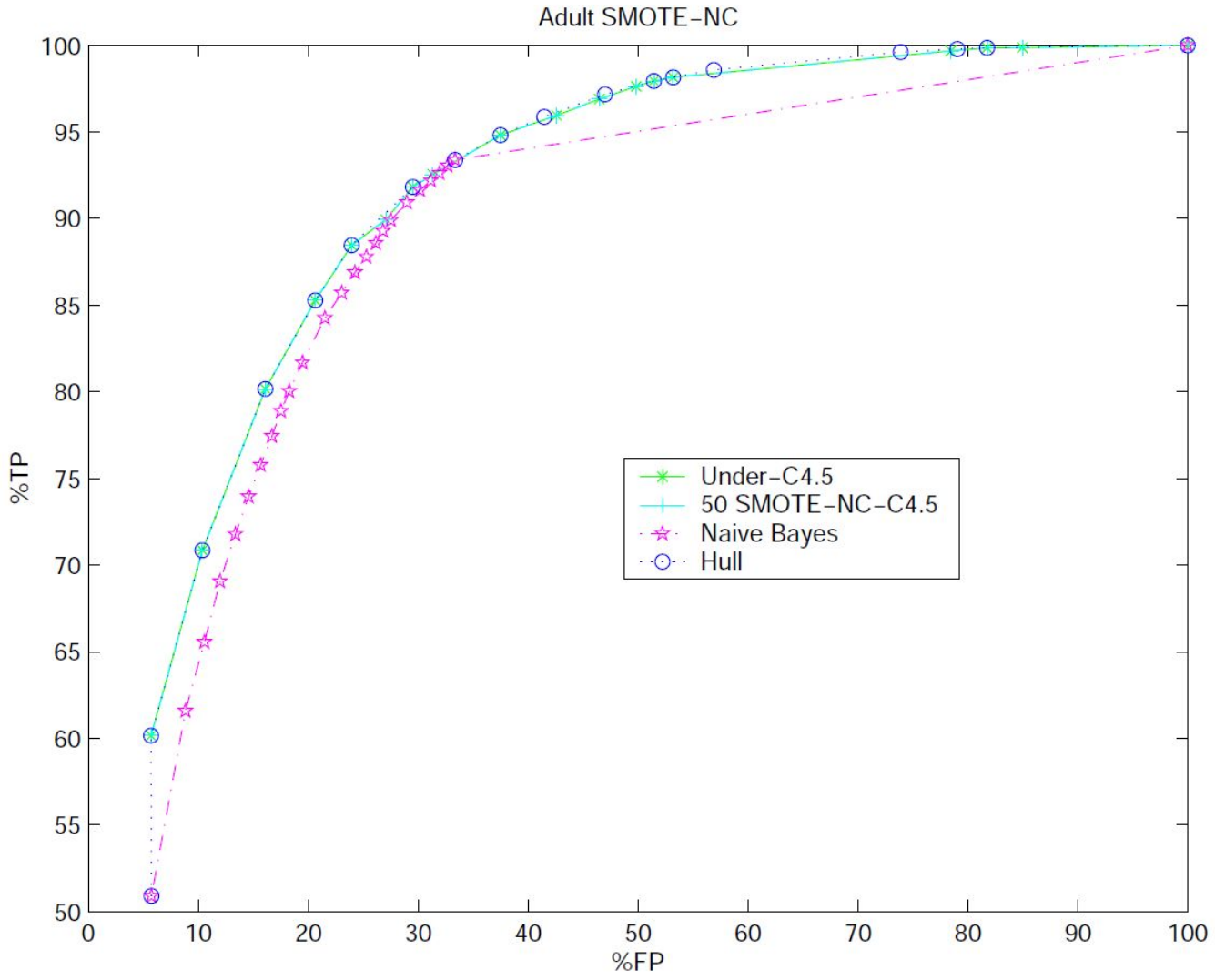
---

F1 = 1 2 3 A B C [Let this be the sample for which we are computing nearest neighbors]
F2 = 4 6 5 A D E
F3 = 3 5 6 A B K

Example

So, Euclidean Distance between F2 and F1 would be:
Eucl = sqrt[$(4\text{-}1)^2$ + $(6\text{-}2)^2$ + $(5\text{-}3)^2$ + Med^2 + Med^2]
Med is the median of the standard deviations of continuous features of the minority class.
The median term is included twice for feature numbers 5: B→D and 6: C→E, which differ for the two feature vectors: F1 and F2.

SMOTE-NC (nominal continuous)



SMOTE-N (nominal)

- Extended version of SMOTE to only handle nominal features
- Modified algorithm
  1. Nearest neighbours computed using the modified version of the Value Difference metric (*Standfill & Waltz 1986; Cost & Salzberg 1993*)
     a. Define distance matrix based on overlap of feature values
     b. Distance between two corresponding feature values:

$$\delta(V_1, V_2) = \sum_{i=1}^{n} \left| \frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right|^k$$

Total # of occurrences of feature value V2 for class i

Total # of occurrences of feature value V1

     c. Distance between two feature vectors:

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^{N} \delta(x_i, y_i)^r$$

1 -> Manhattan
2 -> Euclidean

weights; can be ignored or used to increase the weights of samples found closer to majority class to appear further away

  2. Generate new samples using majority vote of k-nearest neighbours

Summary

- Can improve accuracy of classifiers for minority class
- Combination of SMOTE + under-sampling gernerally performs better than only under-sampling
  - Less over-fitting -> more general
- Provides better results than adjusting model parameters for some algorithms (e.g. loss ratio in Ripper; class priors in Naive Bayes)
  - Provides more minority class samples from which to learn

Our take

- Generally provides similar results to over- or under-sampling with challenging decision boundaries

Possible Discussion Points

- Other use cases or areas ripe with class imbalance?
  - Favoured solutions?
- Popular idea, but outdated?
  - Adaptive Synthetic Sampling (*He et al 2008*)
  - Generative Adversarial Networks (e.g. *Douzas & Bacao 2018*)
- A good inspiration for generating synthetic data?
- ...