SMOTE

SYNTHETIC
MINORITY
OVER-
SAMPLING
TECHNIQUE

AUTHOR: LUCA PERNICE

SMOTE: Synthetic Minority Over-sampling Technique

https://arxiv.org/pdf/1106.1813.pdf

Chawla, Bowyer, Hall & Kegelmeyer

Department of Computer Science and Engineering, ENB 118
University of South Florida
4202 E. Fowler Ave.
Tampa, FL 33620-5399, USA

Kevin W. Bowyer KWB@CSE.ND.EDU
Department of Computer Science and Engineering
384 Fitzpatrick Hall
University of Notre Dame
Notre Dame, IN 46556, USA

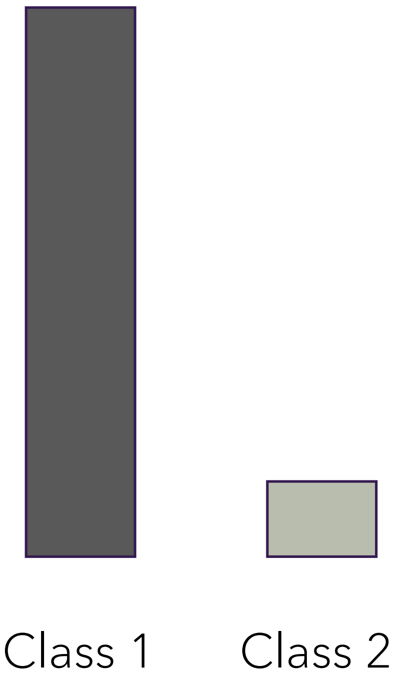
Lawrence O. Hall HALL@CSEE.USF.EDU
Department of Computer Science and Engineering, ENB 118
University of South Florida
4202 E. Fowler Ave.
Tampa, FL 33620-5399, USA

W. Philip Kegelmeyer WPK@CALIFORNIA.SANDIA.GOV
Sandia National Laboratories
Biosystems Research Department, P.O. Box 969, MS 9951
Livermore, CA, 94551-0969, USA

Abstract

An approach to the construction of classifiers from imbalanced datasets is described. A dataset is imbalanced if the classification categories are not approximately equally represented. Often real-world data sets are predominately composed of "normal" examples with only a small percentage of "abnormal" or "interesting" examples. It is also the case that the cost of misclassifying an abnormal (interesting) example as a normal example is often much higher than the cost of the reverse error. Under-sampling of the majority (normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. This paper shows that a combination of our method of over-sampling the minority (abnormal) class and under-sampling the majority (normal) class can achieve better classifier performance (in ROC space) than only under-sampling the majority class. This paper also shows that a combination of our method of over-sampling the minority class and under-sampling the majority class can achieve better classifier performance (in ROC space) than varying the loss ratios in Ripper or class priors in Naive Bayes. Our method of over-sampling the minority class involves creating synthetic minority class examples. Experiments are performed using C4.5, Ripper and a Naive Bayes classifier. The method is evaluated using the area under the Receiver Operating Characteristic curve (AUC) and the ROC convex hull strategy.

Imbalanced dataset



"The cost of misclassifying an abnormal (interesting) example as a normal example is often much higher than the cost of the reverse error."

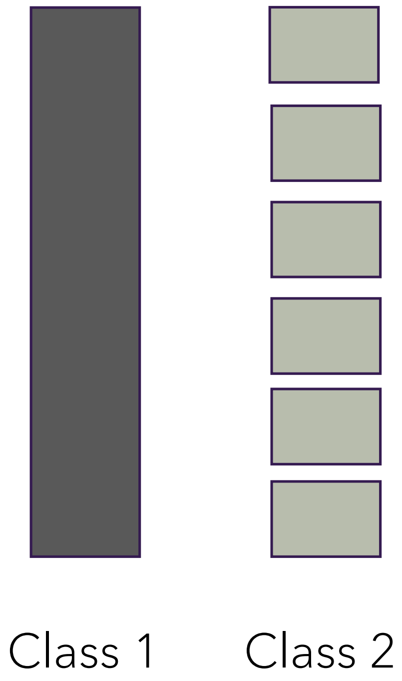
Under - sampling

- Loss of information

Class 1 Class 2

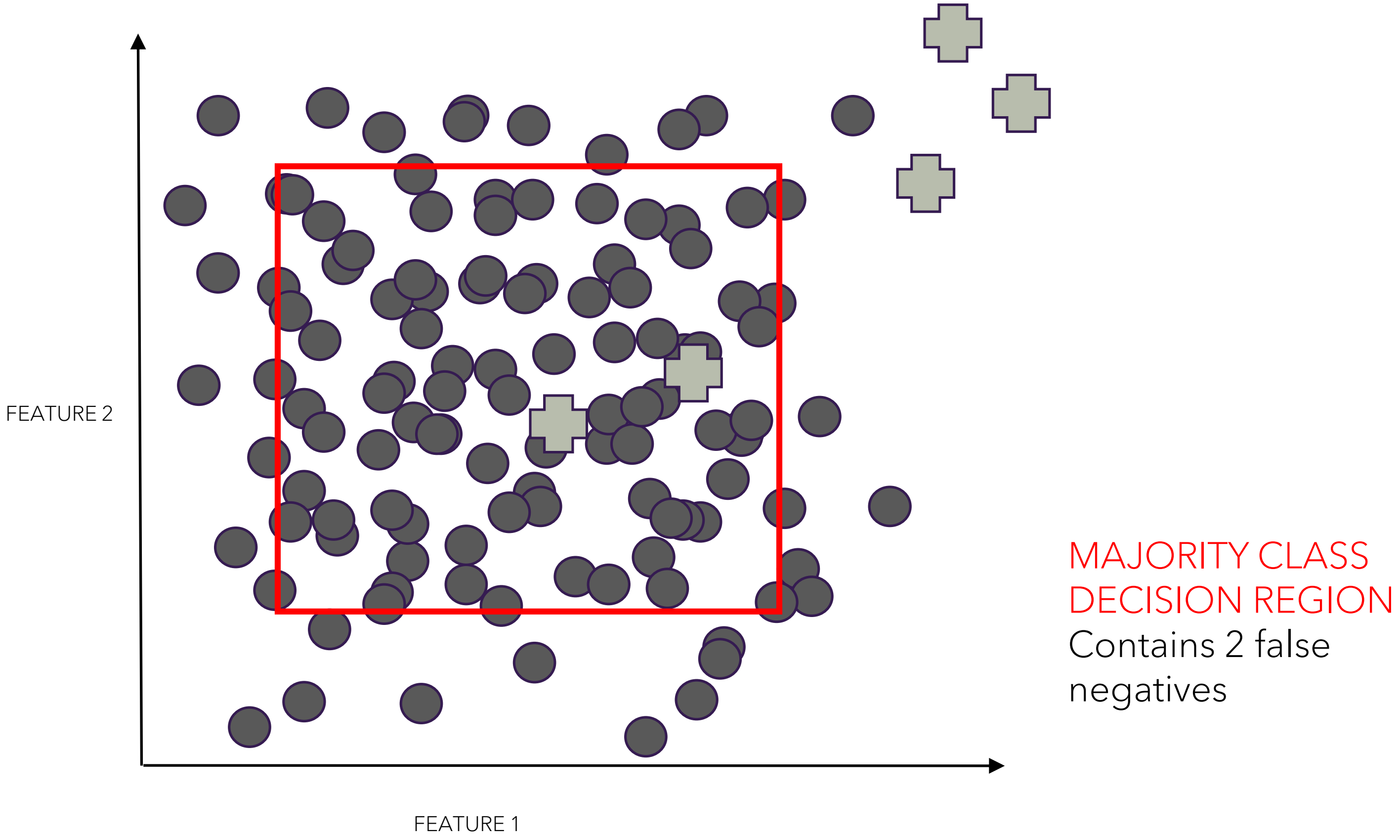
Over - sampling

- Overfitting (with replacement)



DECISION REGION

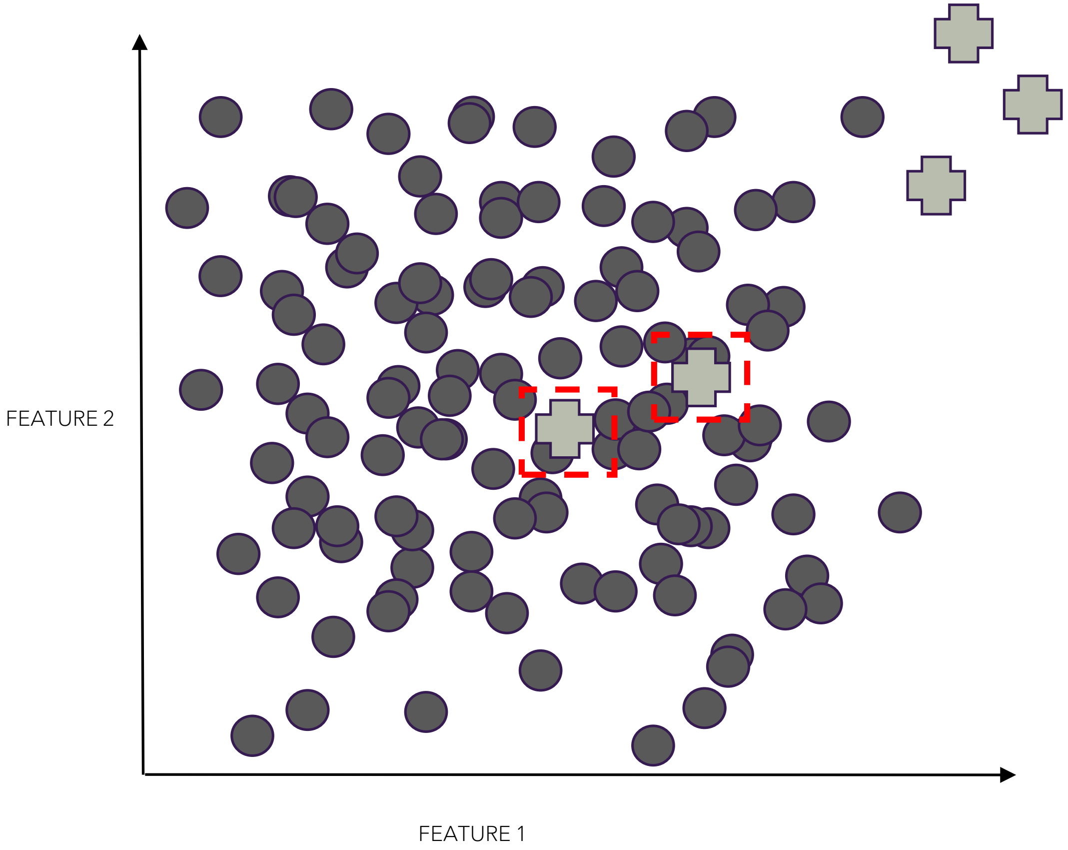
AFTER BUILDING A DECISION TREE



Before
Over-sampling
with replacement

DECISION REGION

AFTER BUILDING A DECISION TREE



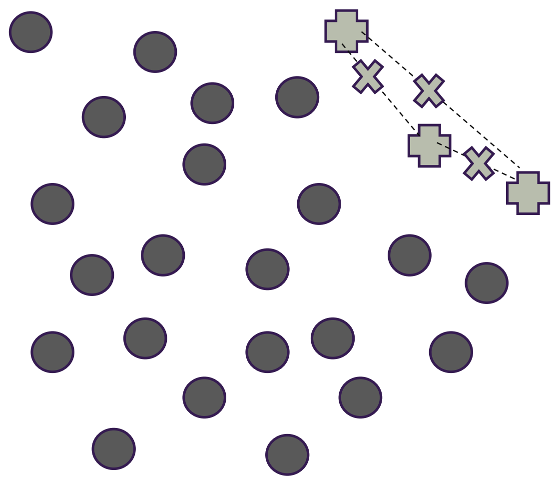
After
Over-sampling
with replacement

NEW MINORITY CLASS
DECISION REGIONS

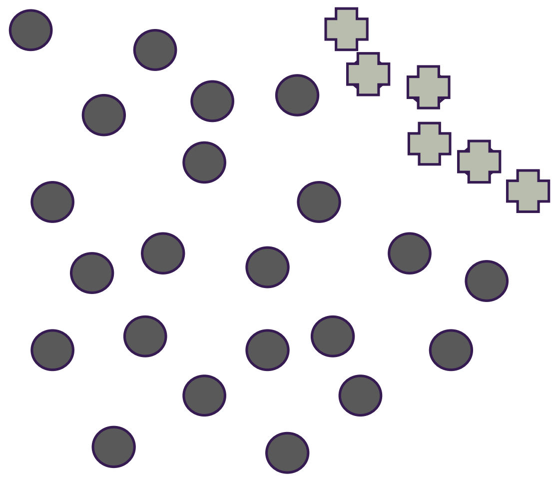
"If we replicate the minority class, the decision region for the minority class becomes very specific and will cause new splits in the decision tree. This will lead to more terminal nodes (leaves) as the learning algorithm tries to learn more and more specific regions of the minority class; in essence, overfitting."

SOLUTION:
ADD PERTURBATION

Generate synthetic data



Generate synthetic data



Algorithm SMOTE(T,N,k)

- Input: Number of minority class samples T;

  Amount of SMOTE N%;

  Number of nearest neighbors K.

- Output: (N/100) • T synthetic minority class samples

Algorithm SMOTE(T,N,k)

```
#If N is less than 100%, randomize the minority class samples as only a
random percent of them will be SMOTEd.
if N<100
      Randomize the T minority class samples
      T = (N/100)*T
      N = 100
N = int(N/100) #The amount of SMOTE is assumed to be in integral
multiplies of 100
```

Algorithm SMOTE(T,N,k)

```
k = Number of nearest neighbors
numattrs = Number of attributes
sample[][]: array for original minority class samples
newindex: keeps a count of number of synthetic samples generated, initialized to 0
#Compute k nearest neighbors for each minority class sample only
for i <- to T
      Compute k nearest neighbors for i, and save the indices in the nnarray
      Populate(N,i,nnarray)
```

Algorithm SMOTE(T,N,k) - Populate(*N,i,nnarray)*

```
#Function to generate the synthetic samples
while N != 0
       Choose a random number between 1 and k, call it nn. This step  chooses
       one of the k nearest neighbors of i.
       for attr <- 1 to numattrs
             Compute: dif = Sample[nnarray[nn]][attr] – Sample[i][attr]
             Compute: gap = random number between 0 and 1
             Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
        newindex++
        N = N – 1
Return
```

TEST
CREDIT CARD FRAUD DETECTION

[link colab]

DATASET

284315 NOT FRAUD

492 FRAUD

30 FEATURES

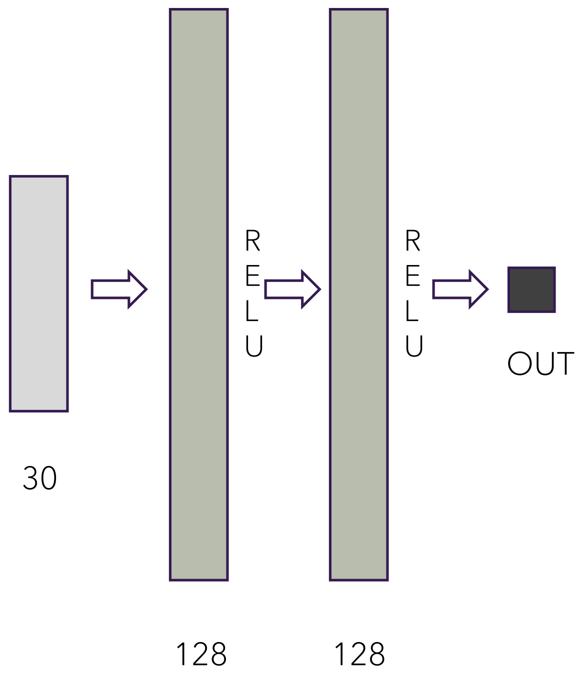
MODEL TRAINING

NEURAL NETWORK

200000 SAMPLES TRAINING SET

20 EPOCHS

0.000001 LEARNING RATE

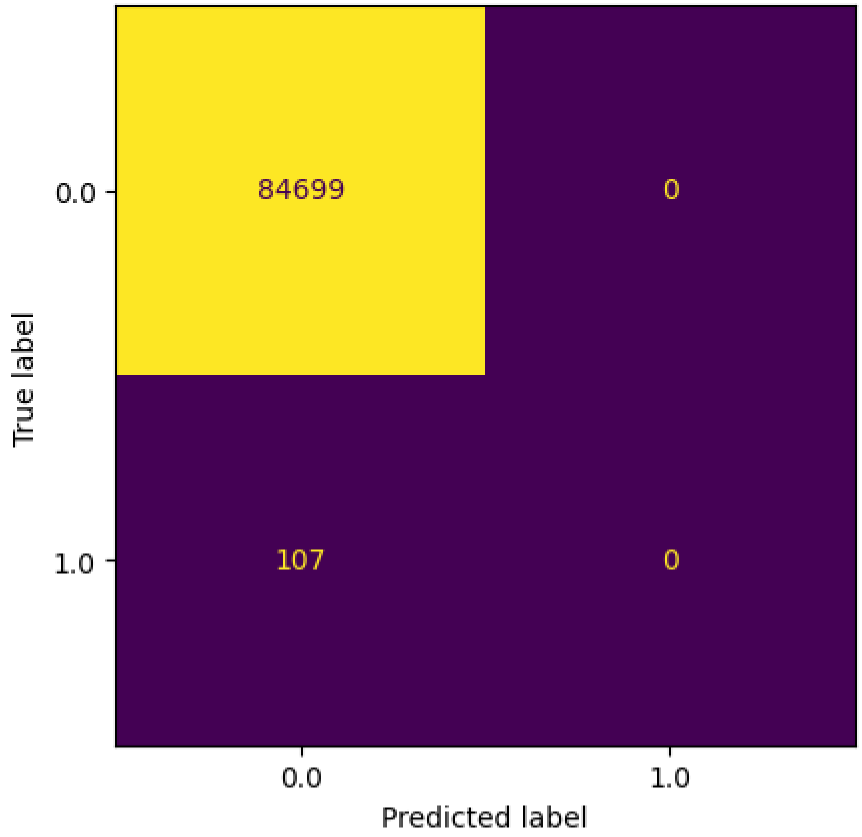


TRAIN DATASET

NOT FRAUD 199615

FRAUD 385

RESULTS
TEST SET



SMOTE OVERSAMPLING (OF TRAIN DATASET ONLY)

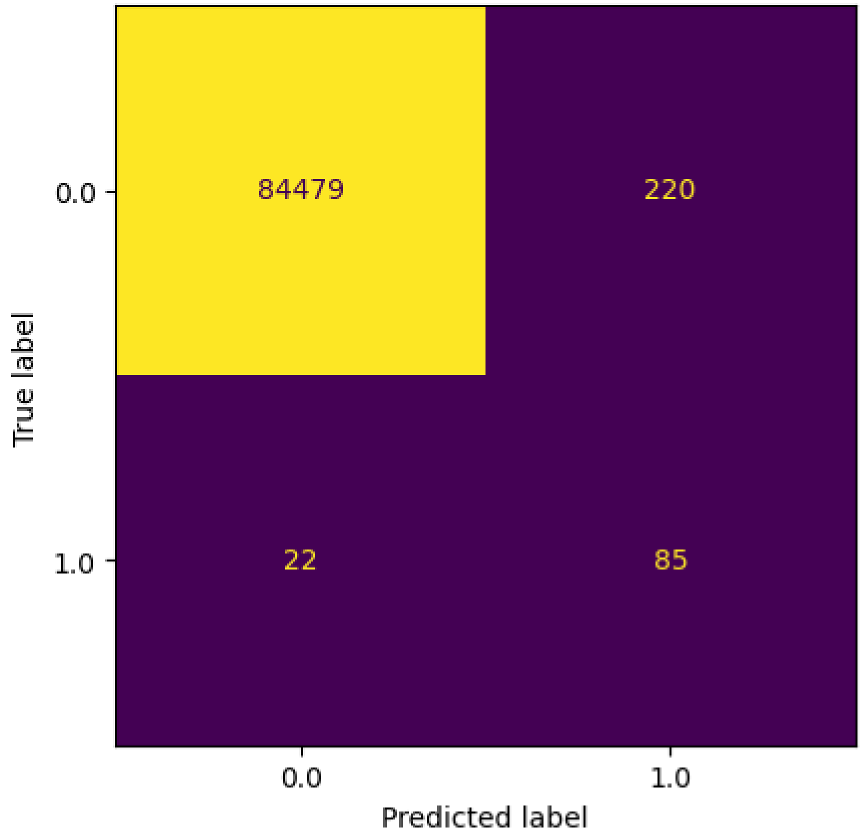
K = 5

NOT FRAUD 199615

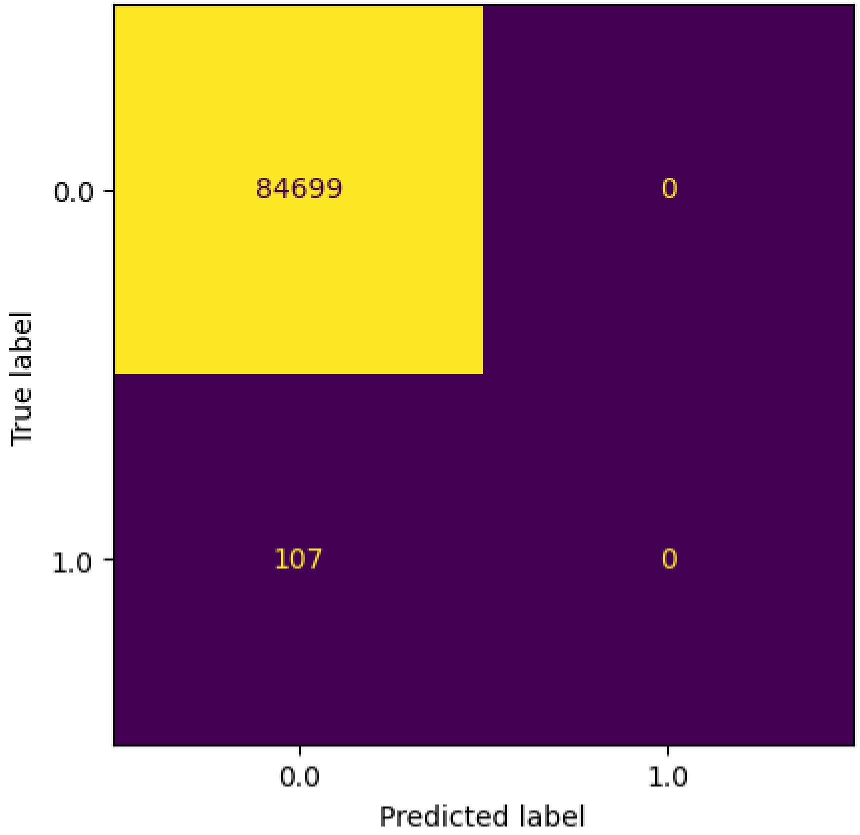
FRAUD 199615 (199230 SYNTHETIC)

RESULTS SMOTE

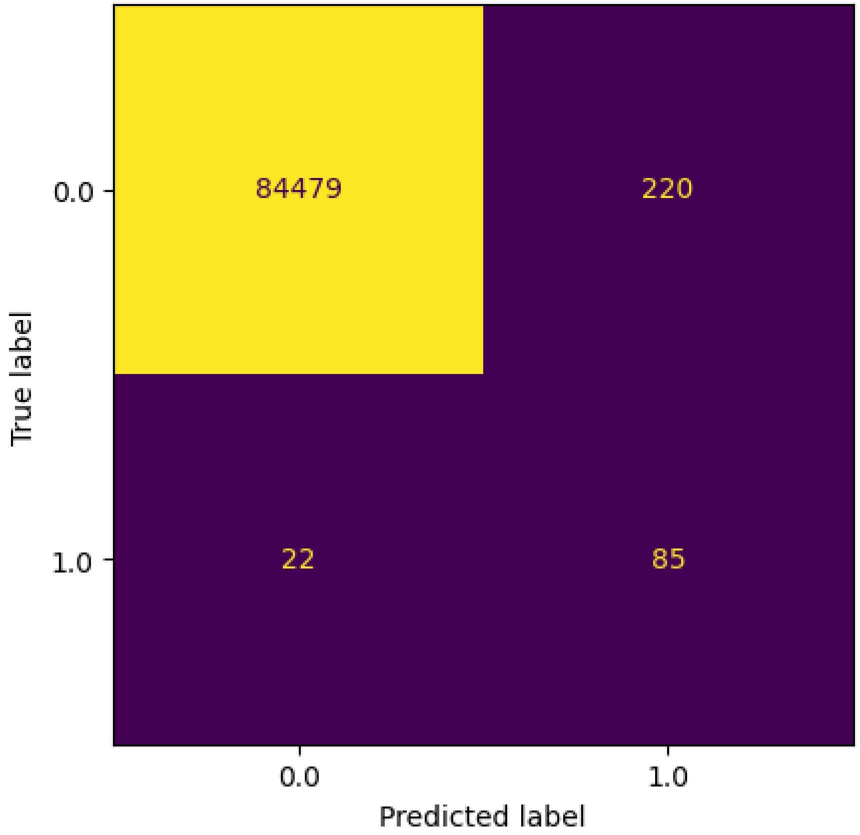
TEST SET



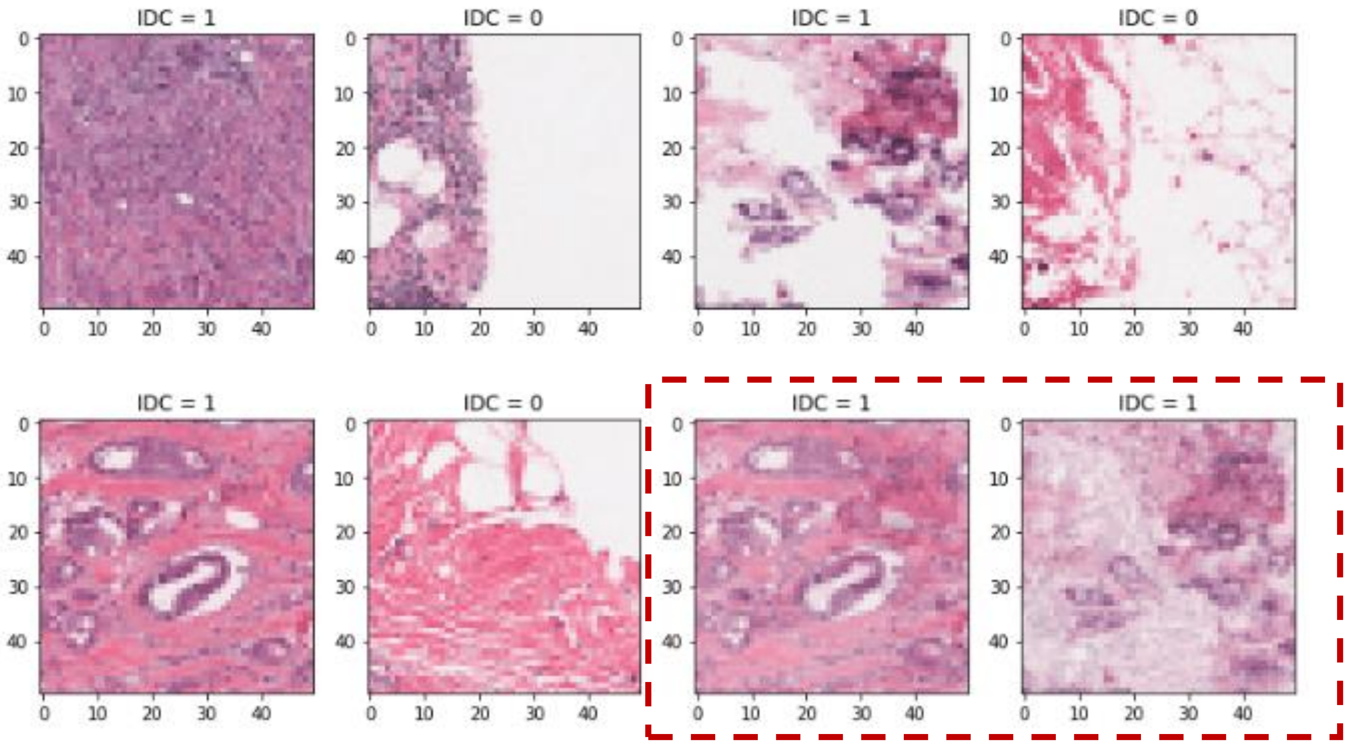
NO SMOTE



SMOTE



OTHER APPLICATIONS



[Imbalanced Histopathological Breast Cancer Image Classification with Convolutional Neural Network]

WHEN SMOTE DOESN'T WORK

HIGH-DIMENSIONAL DATA

HTTPS://BMCBIOINFORMATICS.BIOMEDCENTRAL.COM/ARTICLES/10.1186/1471-2105-14-106

THEORETICAL PROPERTIES OF SMOTE FOT HIGH-DIMENSIONAL DATA

- SMOTE does not change the expected value of the (SMOTE-augmented) minority class and it decreases it's variability

THEORETICAL PROPERTIES OF SMOTE FOT HIGH-DIMENSIONAL DATA

- SMOTE introduces correlation between some samples, but not between variables

THEORETICAL PROPERTIES OF SMOTE FOT HIGH-DIMENSIONAL DATA

- SMOTE modifies the Euclidean distance between test samples and the (SMOTE-augmented) minority class

- SMOTE has hardly any effect on most classifiers trained on high-dimensional data;
- Undersampling or, for some classifiers, cut-off adjustment are preferable to SMOTE for high-dimensional class-prediction tasks.

SOME SMOTE VARIANTS

BORDERLINE-SMOTE

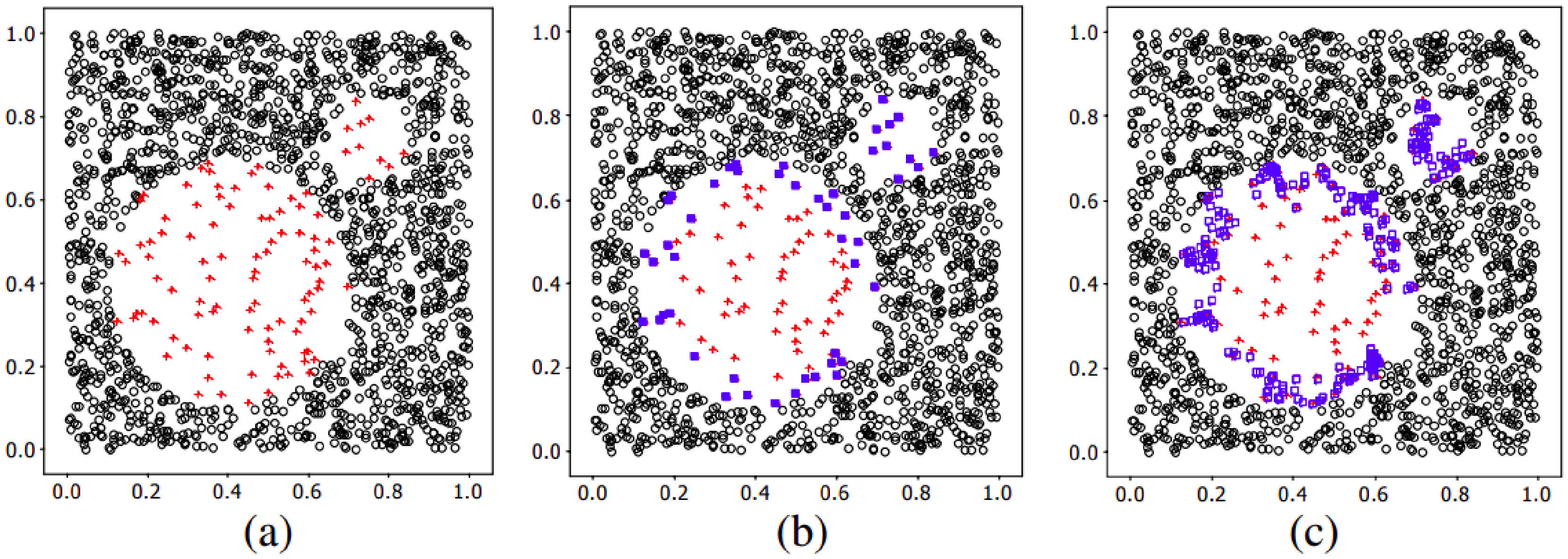
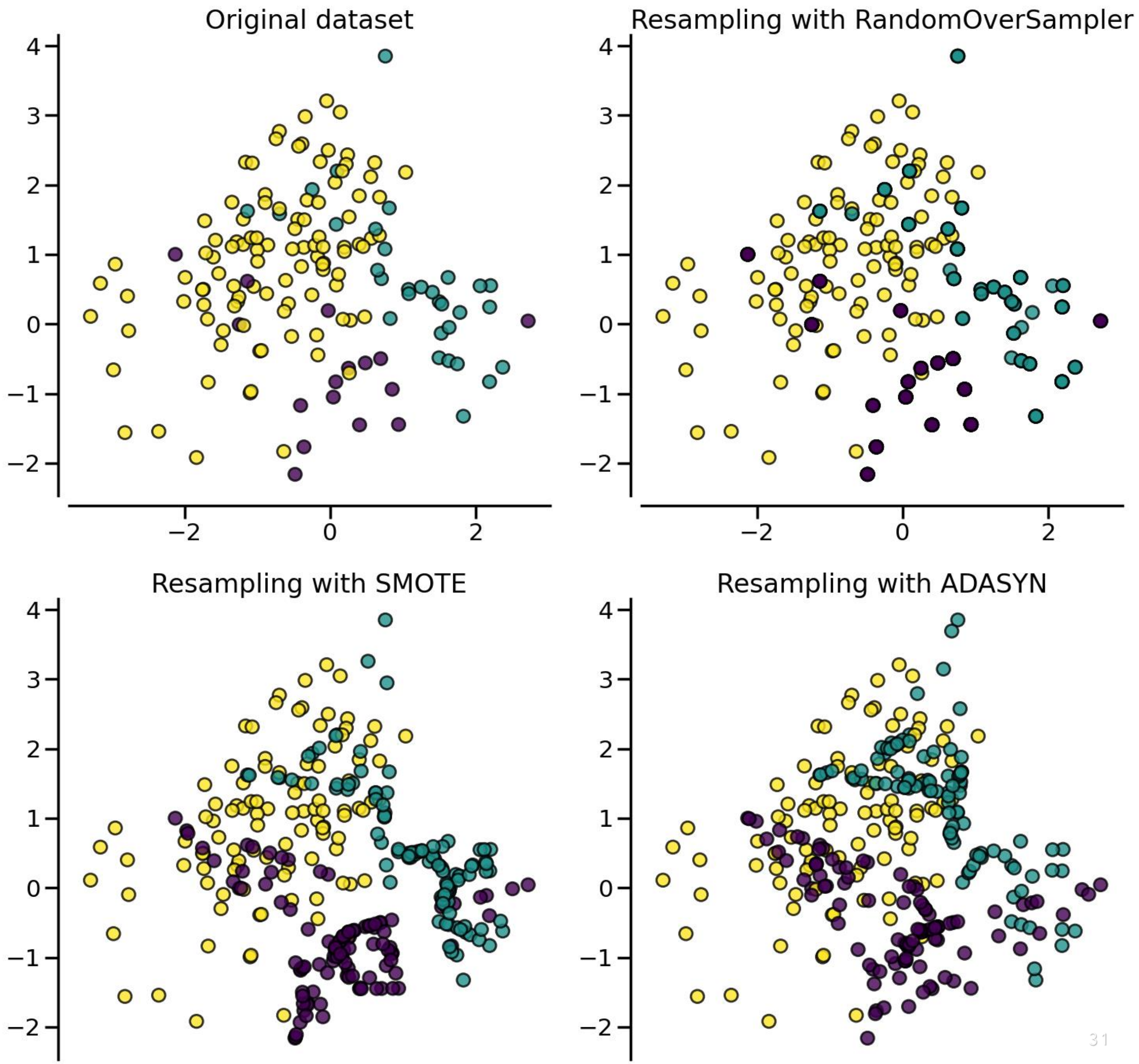


Fig. 1. (a) The original distribution of Circle data set. (b) The borderline minority examples (*solid squares*). (c) The borderline synthetic minority examples (*hollow squares*).

https://sci2s.ugr.es/keel/keel-dataset/pdfs/2005-Han-LNCS.pdf

ADASYN

https://imbalanced-learn.org/stable/auto_examples/over-sampling/plot_comparison_over_sampling.html#sphx-glr-auto-examples-over-sampling-plot-comparison-over-sampling-py



SMOTE-NC

Table 6: Example of nearest neighbor computation for SMOTE-NC.

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
So, Euclidean Distance between F2 and F1 would be:
Eucl = sqrt[$(4-1)^2 + (6-2)^2 + (5-3)^2 + Med^2 + Med^2$] 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 $\rightarrow$ D and 6: C $\rightarrow$ E, which differ for the two feature vectors: F1 and F2.

https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/node15.html