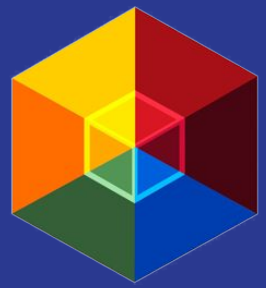
# Handling Imbalanced Dataset

Tim Analisis
Dec 15 2016

KMKLabs - Senayan City



# Introduction

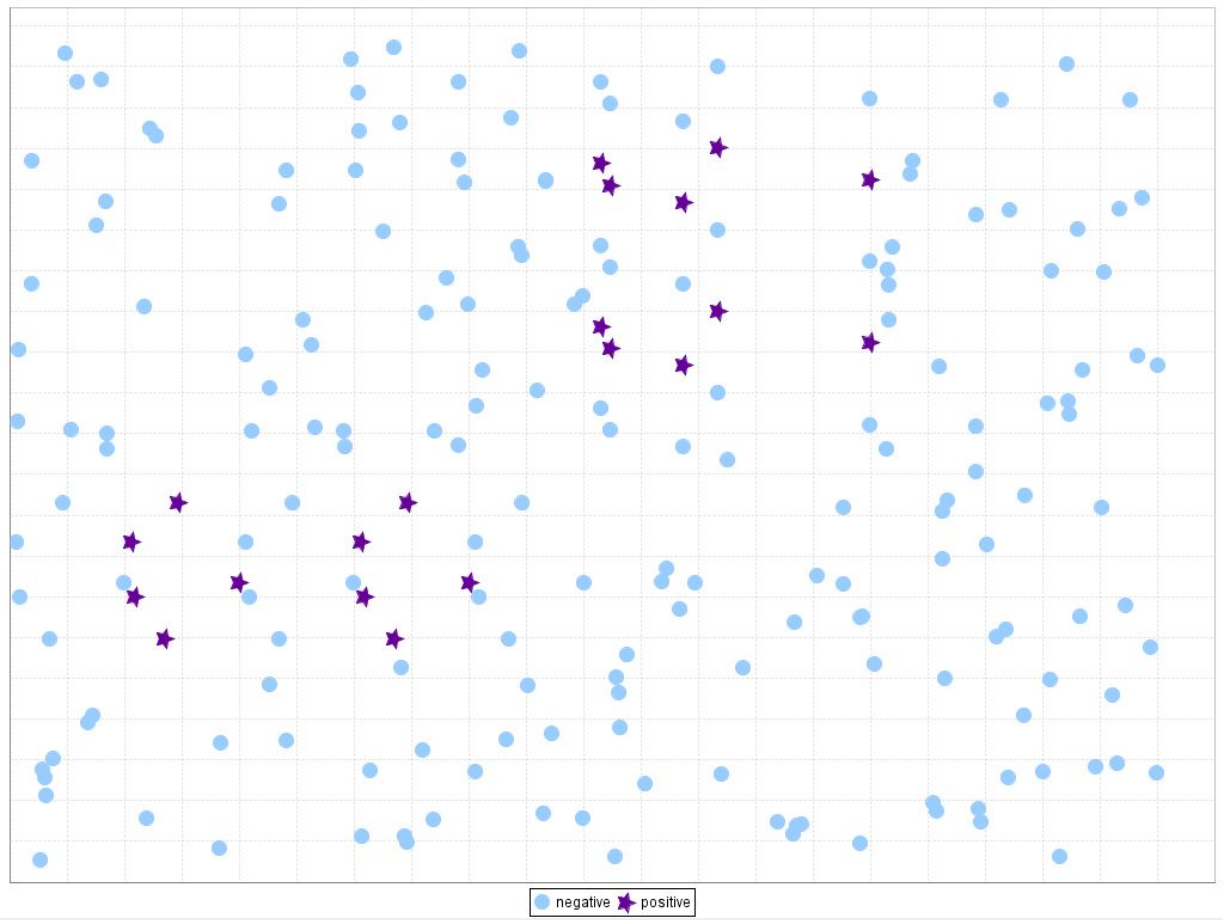
- Background:

→ In real world, a dataset is not always balanced.
→ most classic learning system assume to use the balanced class distribution.

- Definition:

→Imbalanced problem: there is much less examples of one or more class than others.
→ caused by the difficulty or the expensive cost to construct datasets

# Introduction



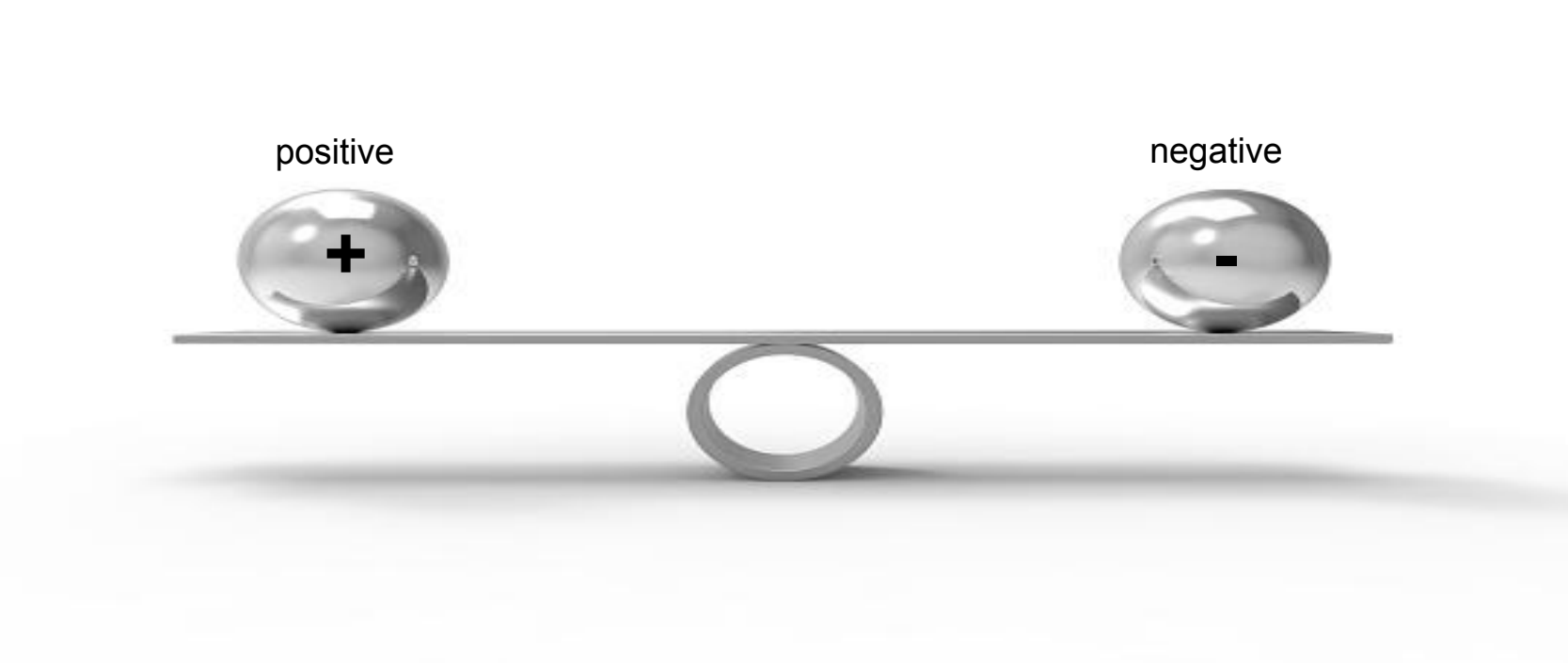

Example of imbalance

→ The star count is too little !!

→ Ratio (pos vs neg)
→ **Ratio = 1 : 95**

# Introduction

Datasets are said to be balanced if there are, approximately, as many positive examples of the concept as there are negative ones.



# Introduction

There exist many domains that do not have a balanced data set.
Examples:

- Helicopter Gearbox Fault Monitoring
- Discrimination between Earthquakes and Nuclear Explosions
- Document Filtering
- Detection of Oil Spills
- Detection of Fraudulent Telephone Calls
- Cancer

Keep in mind that, “Biasanya model kita berfokus kepada class minor,”
Contoh: Fraudulence detection, or Cancer detection

# Problem of Imbalance Dataset

→ Training stages are often biased towards the majority class. (Generalization)

→ because these classifiers attempt to reduce global quantities such as the error rate, not taking the data distribution into consideration.

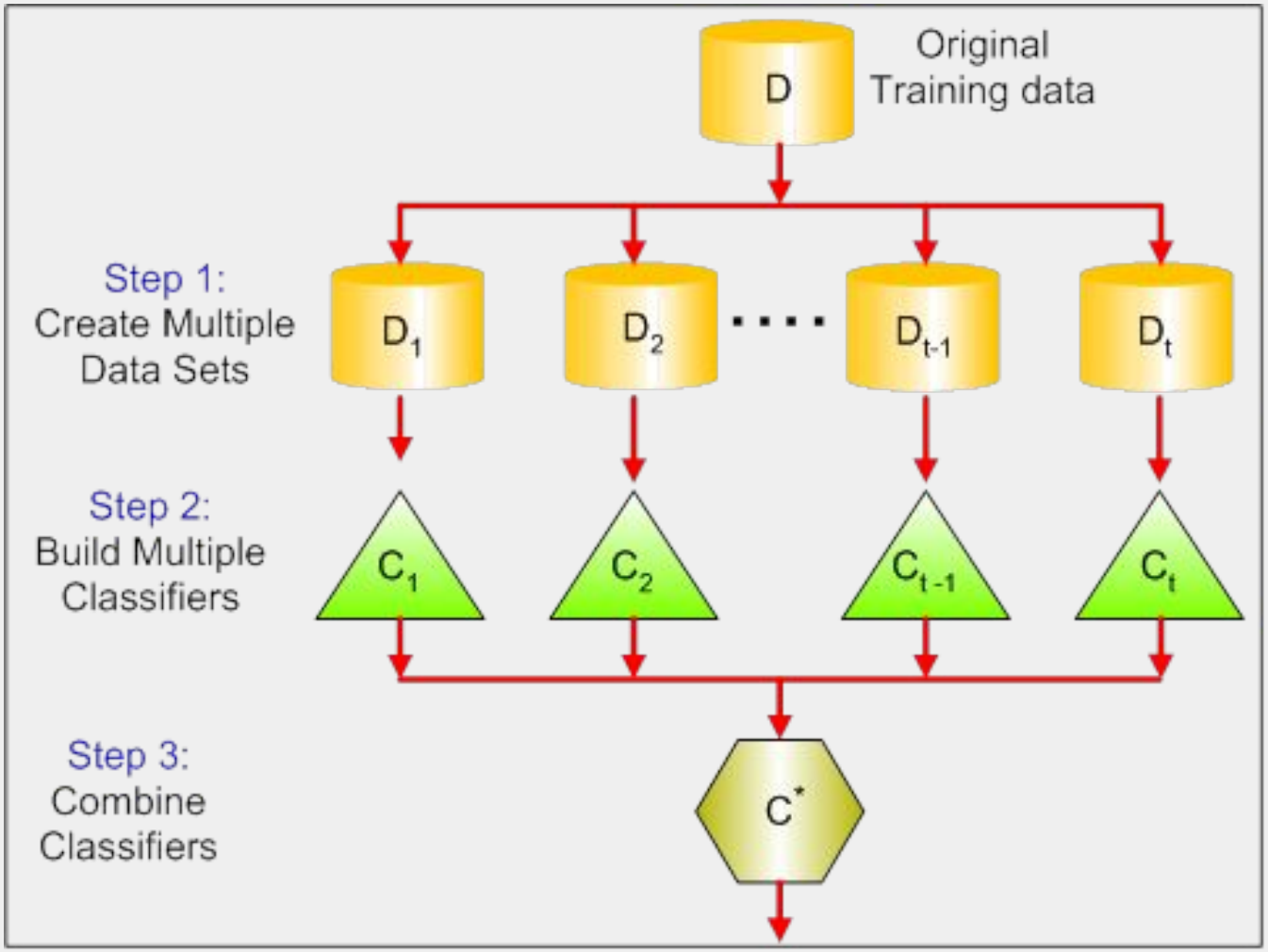
→ As a result examples from the overwhelming class are well-classified whereas examples from the minority class tend to be misclassified.

# How to tackle imbalance

1. Algorithm Level
   → Ensemble learning

2. Data Level
   → Manipulating data

# Tackling Imbalance in Algorithm Level



# Tackling Imbalance in Data Level

→ Basic technique:

- Random Over Sampling (ROS)
  Duplicating minority data

- Random Under Sampling (RUS)
  Deleting some majority data

# Tackling Imbalance in Data Level (Cont'd)

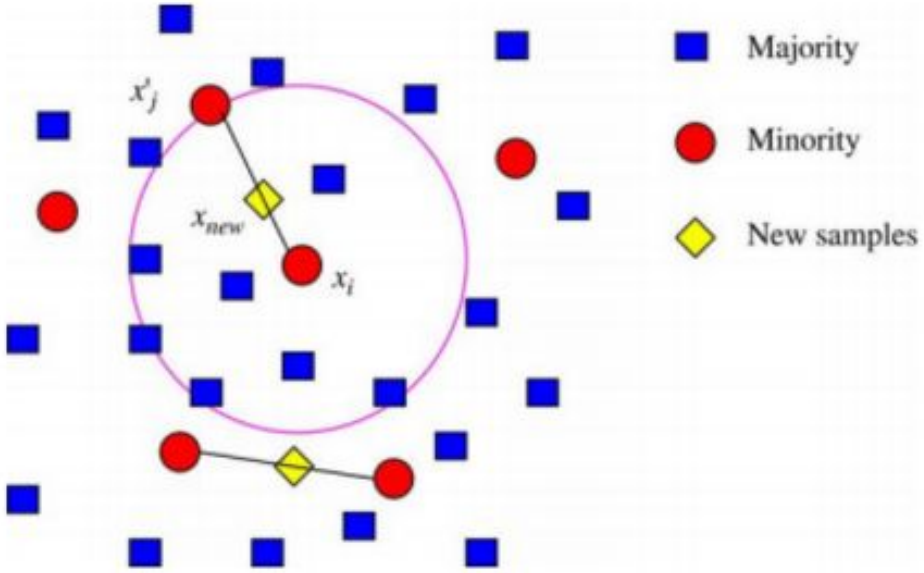
The most famous one: **SMOTE**

- Definition

  - → SMOTE stands for Synthetic Minority Oversampling Technique
  - → is a technique to generate new examples of minority class that is done by interpolating between two examples of minority class that lay together.
  - → utilize Euclidean distance to find the closest neighbor of minority examples.

# SMOTE

❑ Illustration




Source: www.palgrave-journals.com

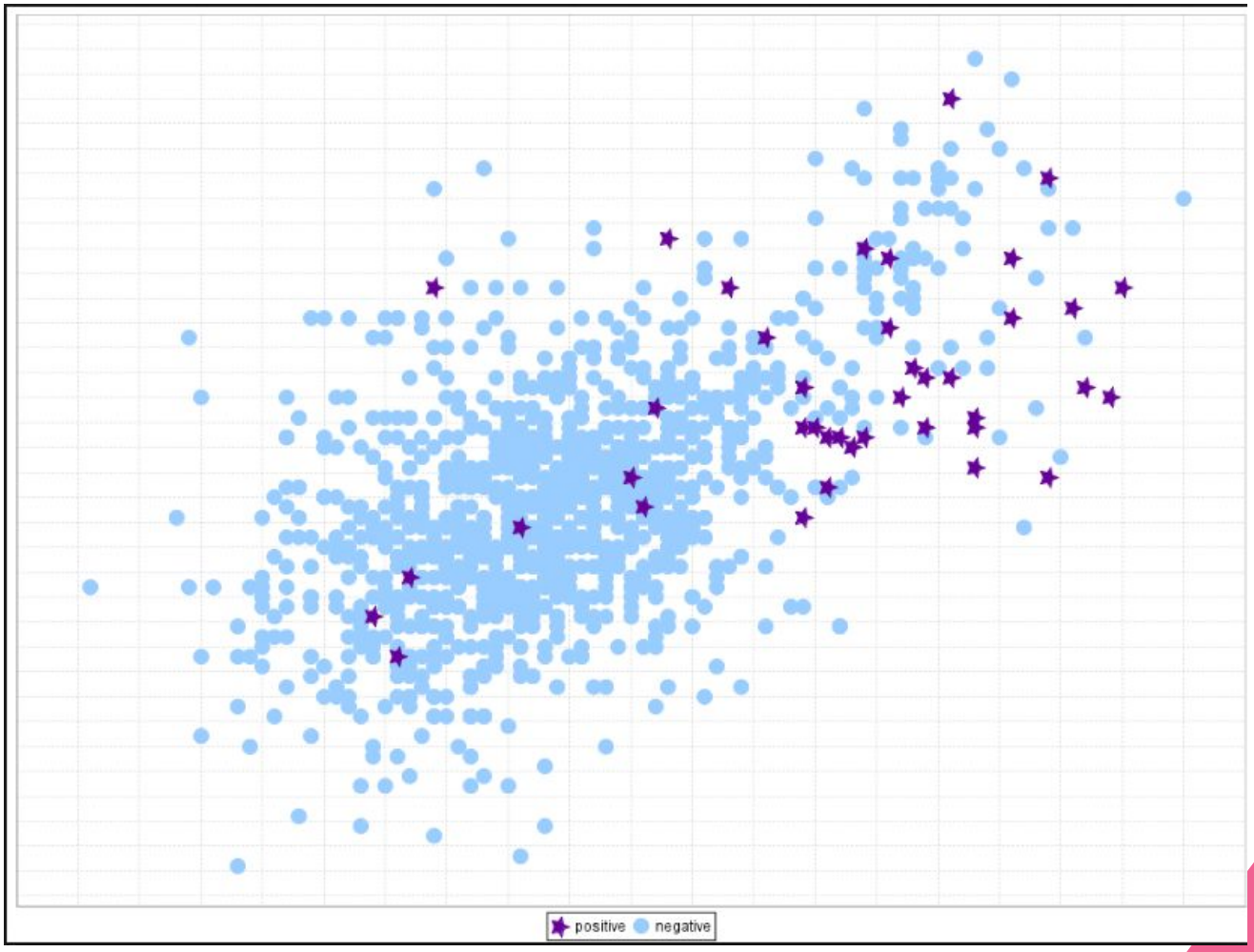
❑ Main Procedure:

→For every minority examples, find its k-nearest neighbors.

→Randomly choose a neighbor to draw a line segment between example in its neighbor.

→ The synthetic data can be created along these line.

# How to measure the performances?




I have dataset with ratio 90% vs 10%

After training,
I obtained accuracy 90%

Is it good result?

# How to measure the performances?

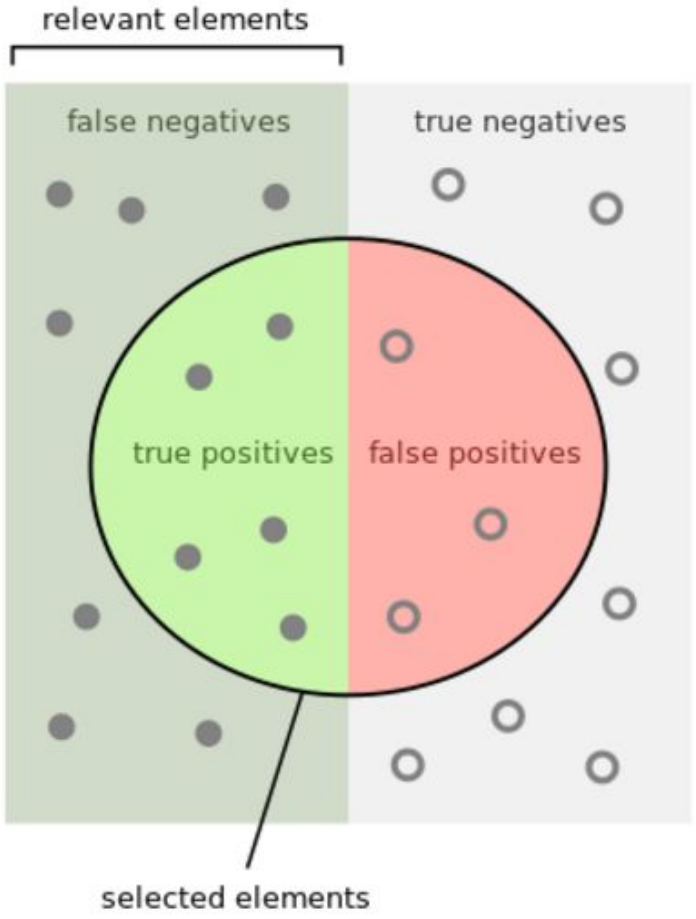
$$Precision = \frac{TP}{TP + FP}$$

$$Recall = Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$F - Measure = \frac{2 * precision * recall}{precision + recall}$$

$$B - Acc = 0.5 * (Specificity + Sensitivity)$$



# SMOTE

## *Installation*

imbalanced-learn is currently available on the PyPi's reporitories and you can install it via *pip*:

```
pip install -U imbalanced-learn
```

The package is release also in Anaconda Cloud platform:

```
conda install -c glemaitre imbalanced-learn
```

thank you :)