Modul 3

# Imbalance Classification

## Data Science Program

Purwadhika
Startup and Coding School

# Outline

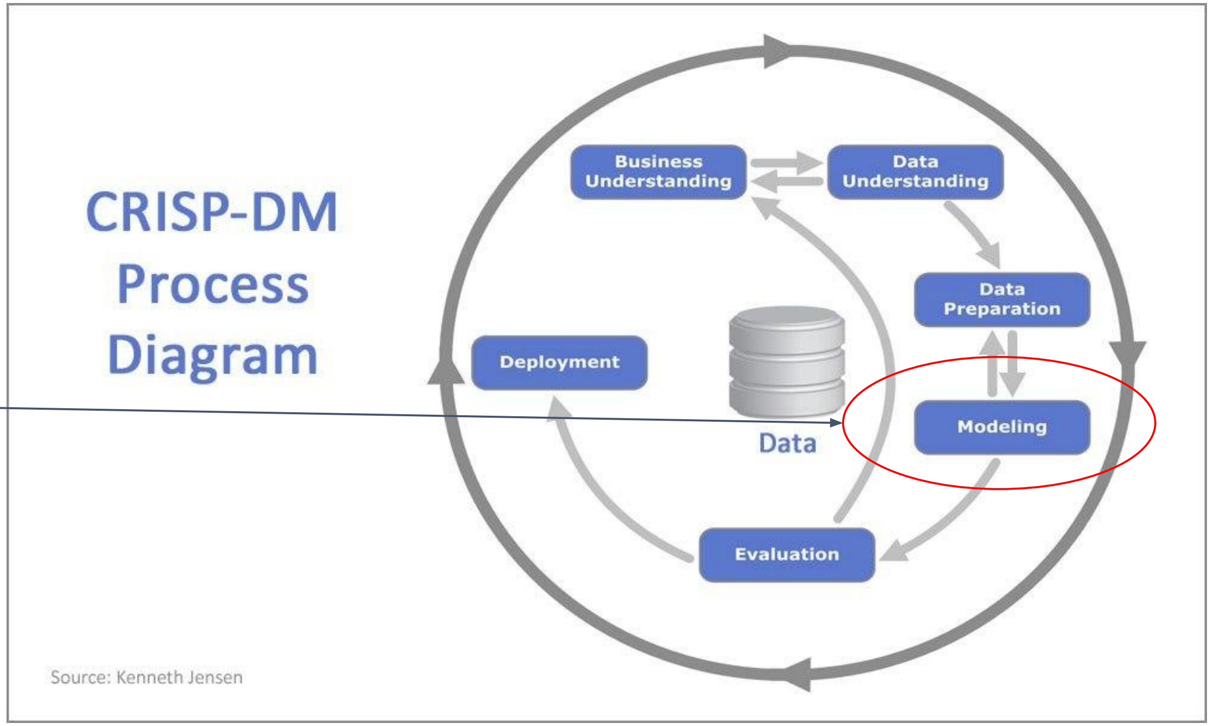
Imbalance Classification

Method:

- data
- resampling
- algorithm based

How to use imbalance method

- metrics
- evaluation method



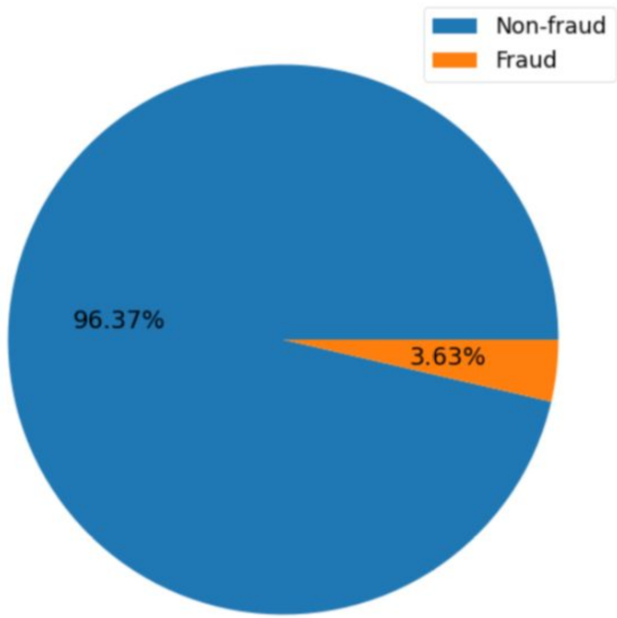

Purwadhika Startup and Coding School

# Imbalance Classification

Purwadhika
Startup and Coding School

# What is imbalance dataset ?

- Imbalance classification, a classification method where the distribution of samples across the classes not equal
- the ratio between negative class and positive class can be below 2:1, 9:1, 95:5, 99:1, etc



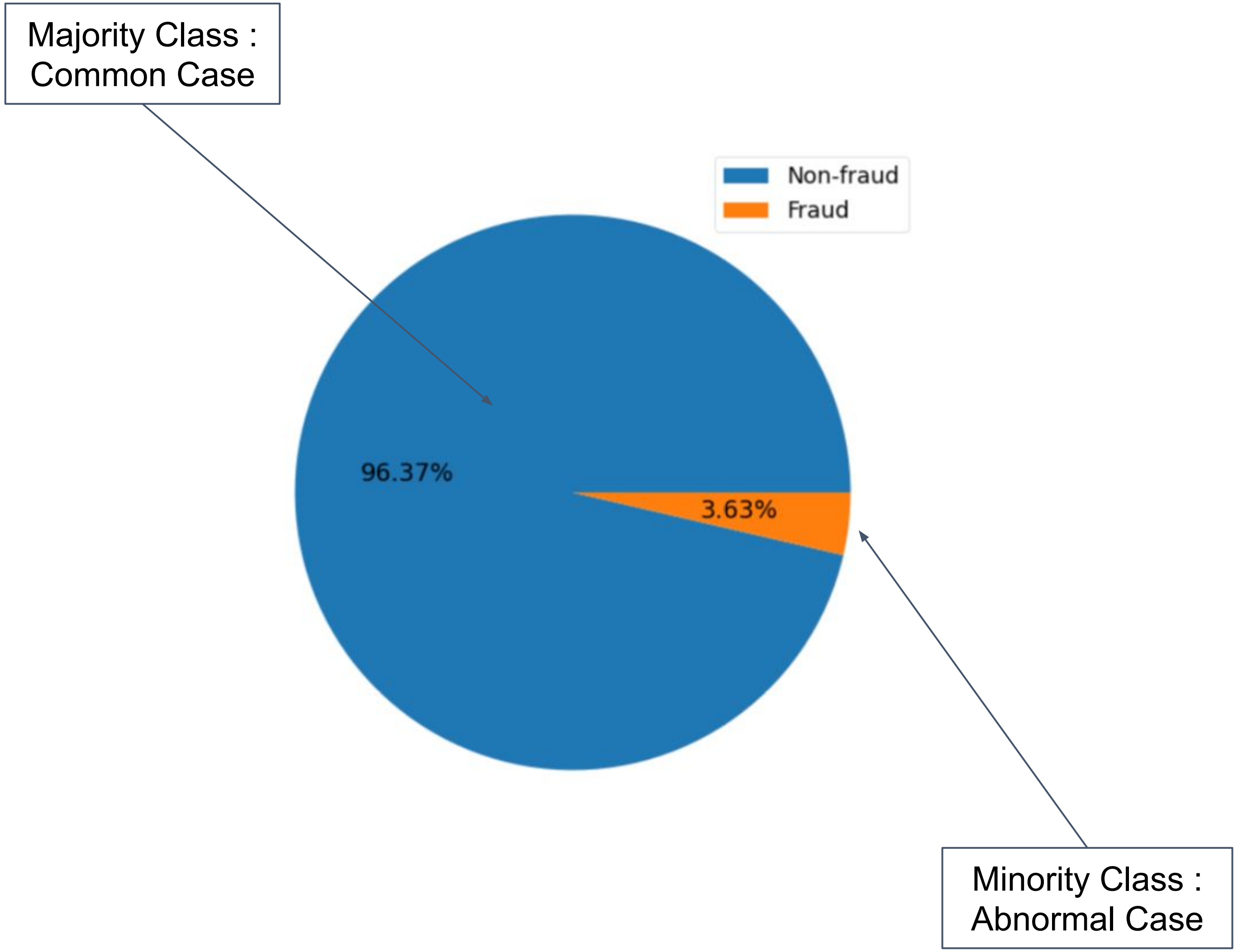

| Actual | Prediction | |
|---|---|---|
| | Non-fraud | Fraud |
| Non-fraud | 5030 | 101 |
| Fraud | 98 | 95 |

Purwadhika
Startup and Coding School

# Cases

Sample cases:

- Fraud detection
- Claim prediction
- Default prediction
- Churn prediction
- Spam Detection
- etc




Purwadhika
Startup and Coding School

# Why should we concern about imbalance classification ?

| Actual | Prediction | |
|---|---|---|
| | Non-fraud | Fraud |
| Non-fraud | 9900 | 0 |
| Fraud | 100 | 0 |

- non-fraud : 9900
  fraud : 100
  ratio : 99:1
- all model prediction is non-fraud
- accuracy = 9900/10000 = 99 %
- but the model fail to detect all the fraud transaction

# Why should we concerned about imbalance classification ?

The algorithm will tend to ignore the minority class while minority class is often important

- those working on fraud detection will focus on identifying the fraudulent transactions rather than on the more common legitimate transactions
- a telecommunications engineer will be far more interested in identifying the equipment about to fail than the equipment that will remain operational
- etc

# Some Solution

Data:

- Collect more data
- Feature Engineering

Resampling:

- Undersampling
- Oversampling
- CNN, NCR, Near Miss
- SMOTE

Algorithm Based :

- Penalized method
- Use certain algorithm

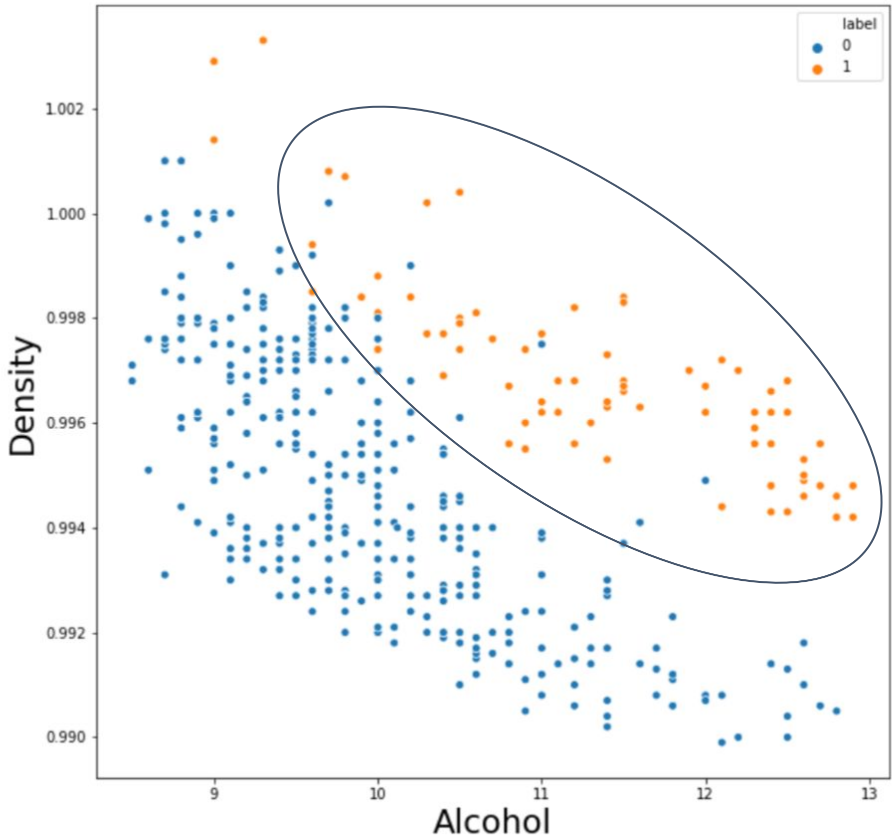
# Data

# Data

- your machine learning is only as good as your data
- if your class is imbalance but the feature is able to separate each classes, there will be no need to use any other balancing technique (resampling and algorithm based)
- You can either :
  - collect more data (row and column)
  - feature engineering

Purwadhika
Startup and Coding School

# Illustration




Dataset Description :

- dataset : white_wine.csv
- imbalanced target : wine quality
  positive class : quality > 6 (18.9%)
  negative class : quality <= 6 (81.1%)
- feature : Density and Alcohol

Task :

- Do Modeling without polynomial features
- Check : recall, precision and f1-score
- Do Modeling without polynomial features
- Check : recall, precision and f1-score

Without Polynomial Features

performance

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.87 | 0.96 | 0.91 | 106 |
| 1 | 0.69 | 0.38 | 0.49 | 24 |
| accuracy | | | 0.85 | 130 |
| macro avg | 0.78 | 0.67 | 0.70 | 130 |
| weighted avg | 0.84 | 0.85 | 0.84 | 130 |

With Polynomial Features

performance

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.99 | 0.98 | 106 |
| 1 | 0.95 | 0.88 | 0.91 | 24 |
| accuracy | | | 0.97 | 130 |
| macro avg | 0.96 | 0.93 | 0.95 | 130 |
| weighted avg | 0.97 | 0.97 | 0.97 | 130 |

Conclusion :

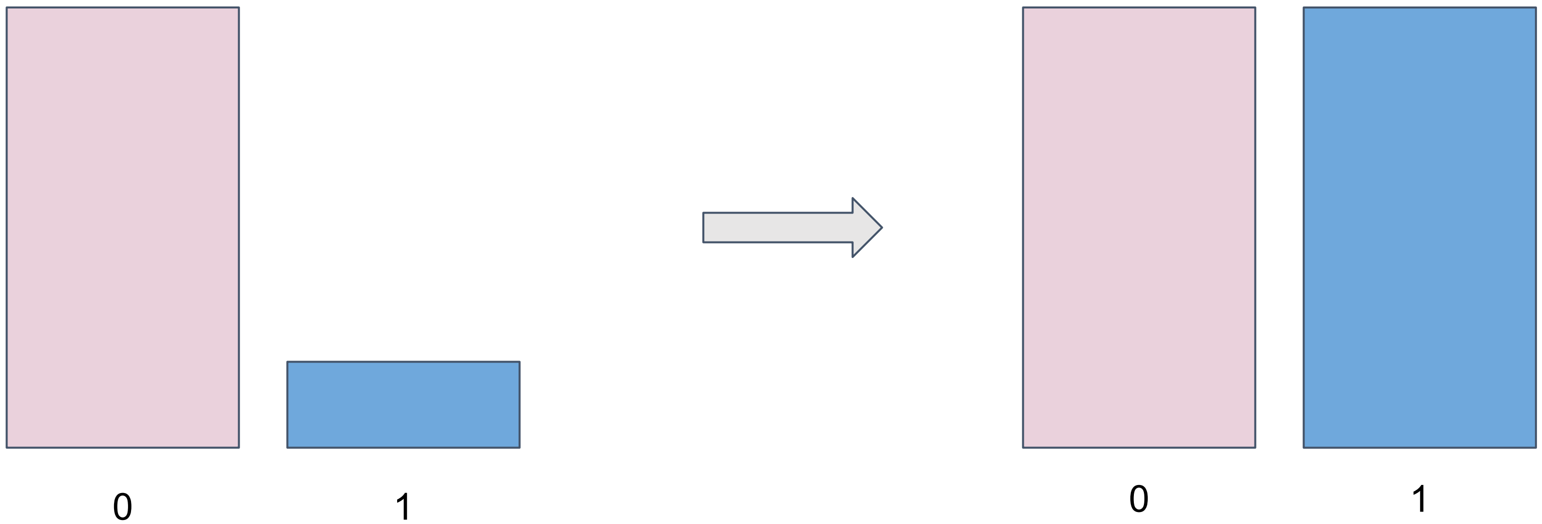
- We can improve performance of imbalance classification by only providing better input

Purwadhika
Startup and Coding School

# Resampling

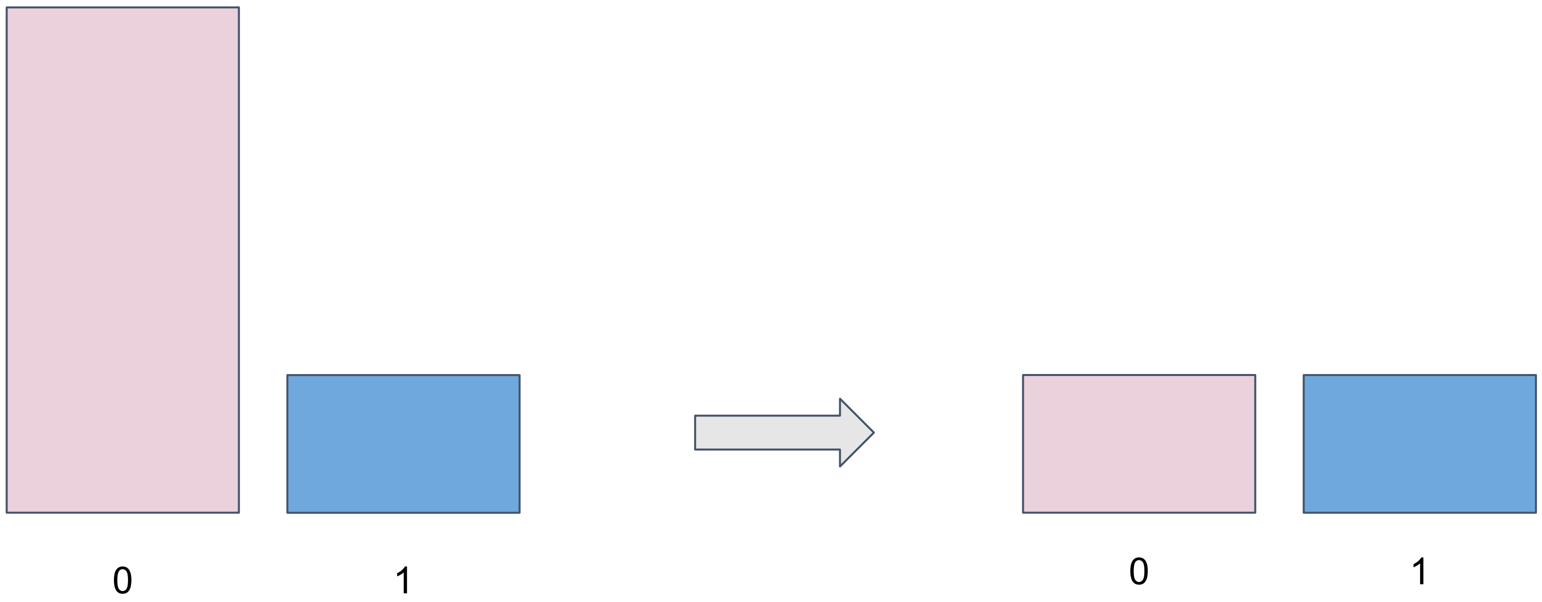
# Resampling

- Creating a dataset that has relatively balanced class distribution
- Method :
  - undersampling
  - oversampling



# Random Undersampling

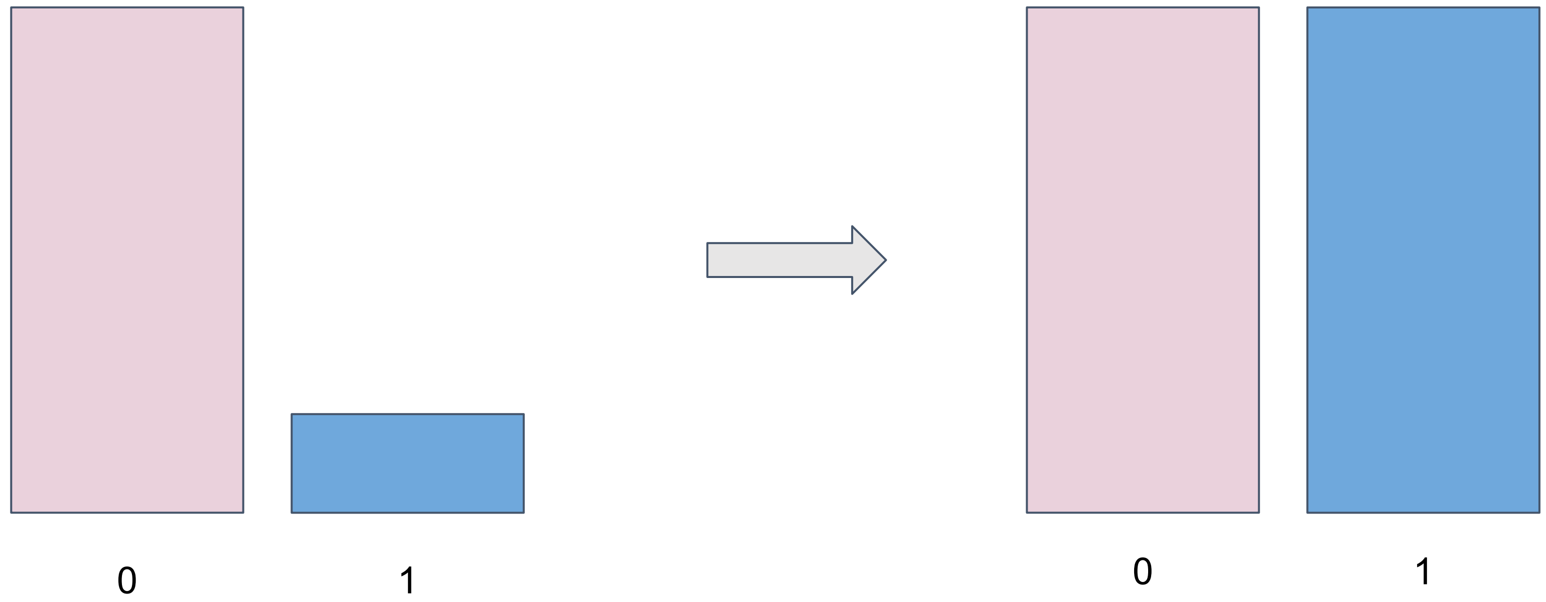
- Discard the majority class randomly until a more balanced distribution is reached




Purwadhika
Startup and Coding School

# Random Oversampling

- Copy and repeat the minority class randomly until a more balanced distribution is reached



# Drawback

| Random Undersampling | Random Oversampling |
| --- | --- |
| - Vast quantity of data are discarded<br>- Loss information<br>- Loss performance | - too many data copied<br>- overfitting<br>- poor generalization |

Purwadhika
Startup and Coding School

# Undersampling Technique

Discard the majority class based on certain criteria

List of technique :

- Condensed nearest neighbour (CNN)
- Neighbour cleaning rule (NCR)
- Near Miss

Aim of these methods is to remove borderline and noisy data in the majority class until a more balanced distribution is reached

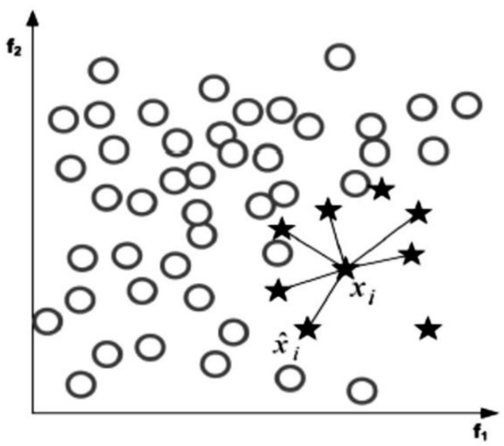
# Oversampling Technique

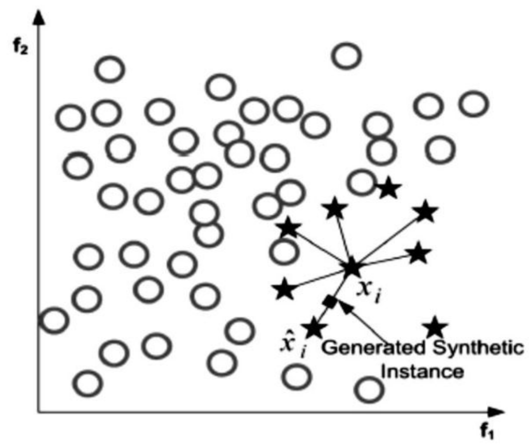
Create synthetic minority data that similar to the real data

Technique :

- Synthetic minority oversampling (SMOTE)

Keep adding synthetic data until a more balanced distribution is reached



# Algorithm Based

# Penalized Models

Make your model paying more attention to the minority class

This method is faster than resampling method

you can use “class_weight” arguments in some scikit learn estimator :

- logistic regression
- decision tree
- random forest
- support vector machine

Purwadhika
Startup and Coding School

# Python Exercise : Imbalance Classification

Analyze data bankloan.csv

- build a logistics regression model
  - target : default
  - features : employ, debtinc, creddebt, othdebt
- Explore the class distribution
- Random state 2020, stratified training 60% validation 20% testing 20%
- Modeling evaluate by f1 score in test set:
  - logistic regression without any treatment
  - logistic regression that optimized by the threshold
  - logistic regression with random undersampling
  - Penalized logistic regression

Purwadhika
Startup and Coding School

# How to Handle Imbalance Problem Properly ?

Two things to note:

- Metrics
- Evaluation Method

# Metrics

Purwadhika
Startup and Coding School

# Metrics We Already Discussed

Metrics that we already discussed can be used to measure imbalance classification problem

- Interested in one of the class only: F1-score
- Interested in the probability (both class are important) : ROC AUC
- Interested in the probability (only one class are important): PR AUC

Purwadhika
Startup and Coding School

# Balanced Accuracy

- accuracy is not an effective method
- You are interested in **both classes**
- computes the average of the percentage of positive class instances correctly classified (sensitivity) and the percentage of negative class instances correctly classified (specificity)

$$\text{BalancedAccuracy} = \frac{\text{TP}}{2(\text{TP} + \text{FN}))} + \frac{\text{TN}}{2(\text{TN} + \text{FP}))}$$

# Geometric Means

- G-mean 1 : Focus on **both class**
- Takes into account the relative balance of the classifier’s performance on both the positive and the negative classes

$$G - \text{mean}_1 = \sqrt{\text{sensitivity} \times \text{specificity}}$$

- G-mean 2 : Focus on **one class** only
- Takes into account the relative balance of sensitivity/recall and precision

$$G - \text{mean}_2 = \sqrt{\text{sensitivity} \times \text{precision}}$$

Purwadhika
Startup and Coding School

# F-Measure

- F1-score is the specific version of F-Measure. It's considered precision and recall equally and focused in **one class** only
- F-Measure combine precision and recall in different ways

$$F_\alpha = \frac{(1+\alpha)[\text{precision} \times \text{recall}]}{[\alpha \times \text{precision}] + \text{recall}}$$

- alpha = 1 (F1-score) : combine recall and precision equally

  alpha = 2 (F2-score) : precision twice more important

  alpha = 0.5 (F0.5-score) : recall twice more important

Purwadhika
Startup and Coding School

# Brier Score

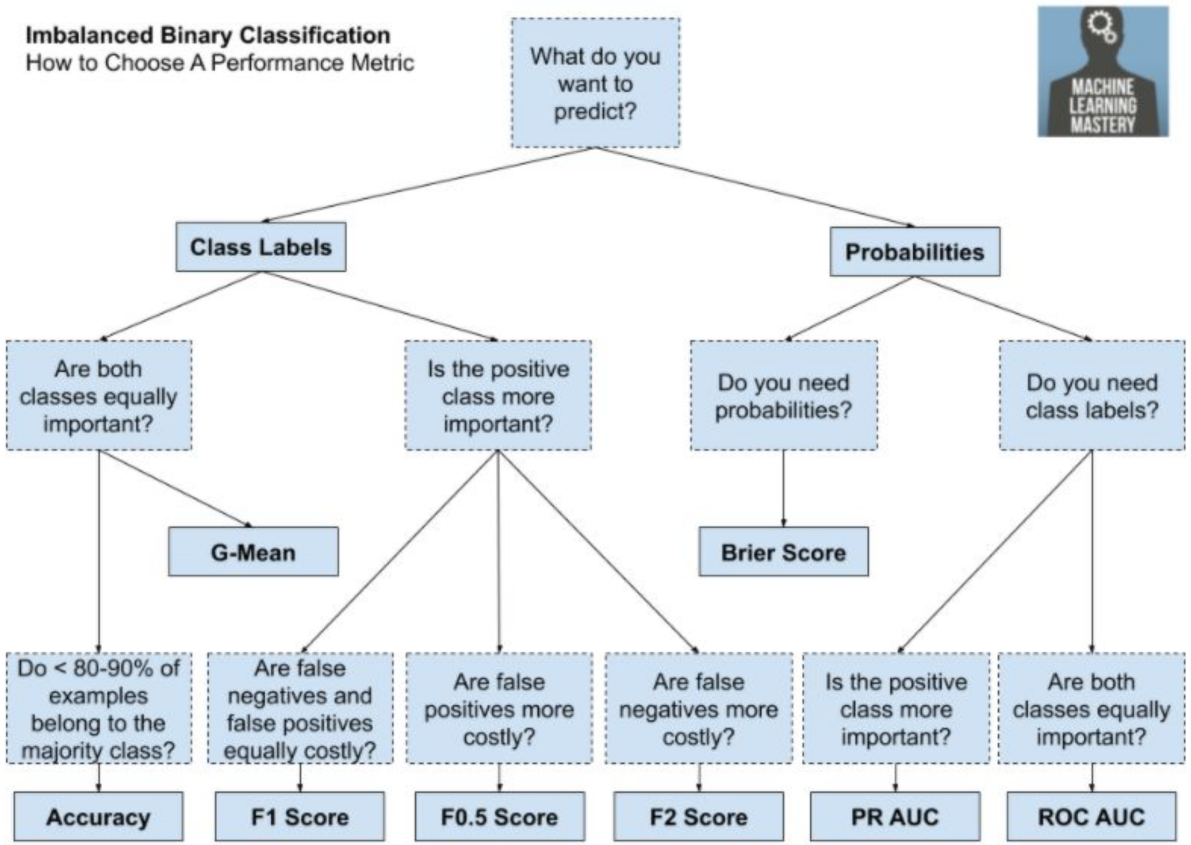
Evaluating a model based on probability:

- No calibration : for parametrics method : i.g. logistic regression
- Need Calibration : for non-parametrics method : i.g. SVM, K-NN, Decision Tree, Random Forest

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_i)^2$$

# Metrics Example Guidance

- wrong metrics may mislead your model from business objective

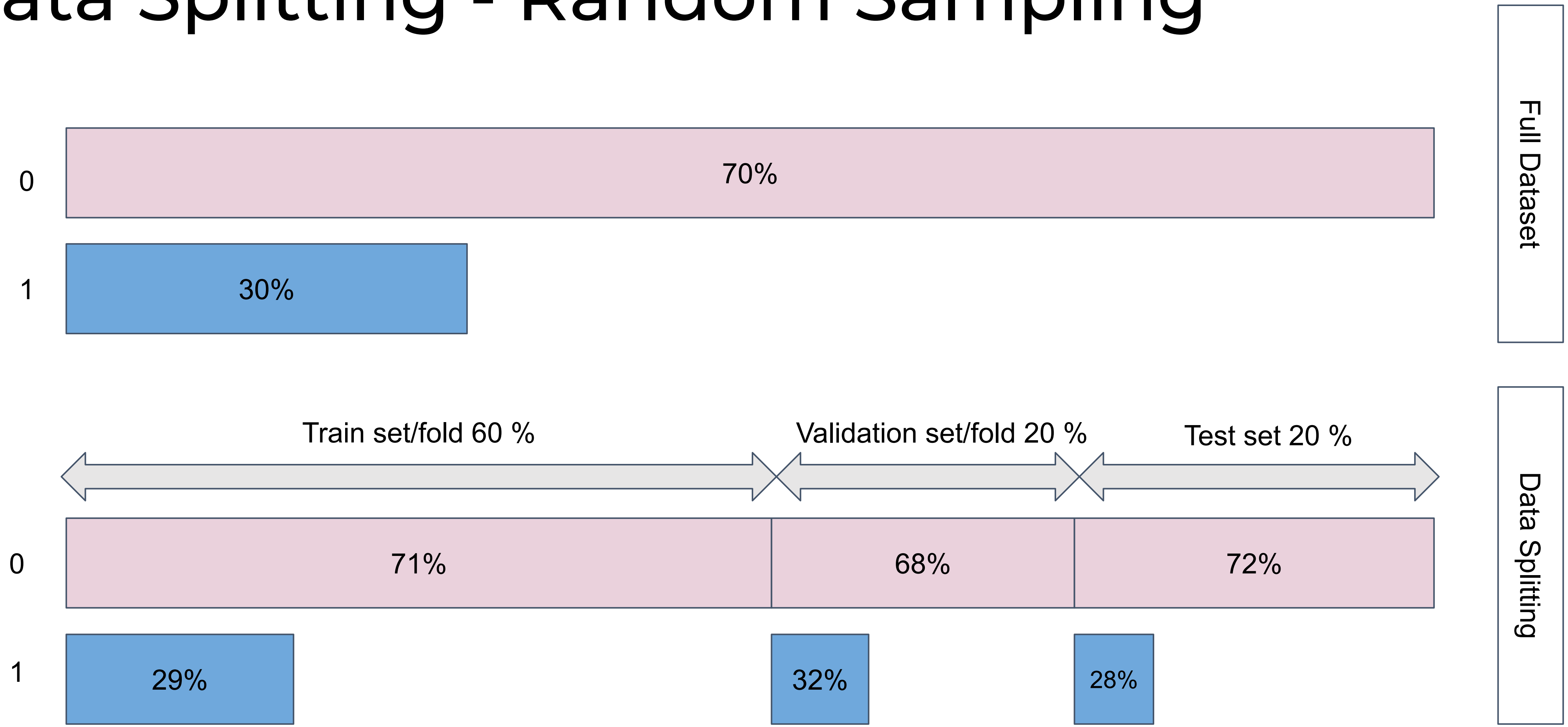



Purwadhika Startup and Coding School

# Evaluation Method

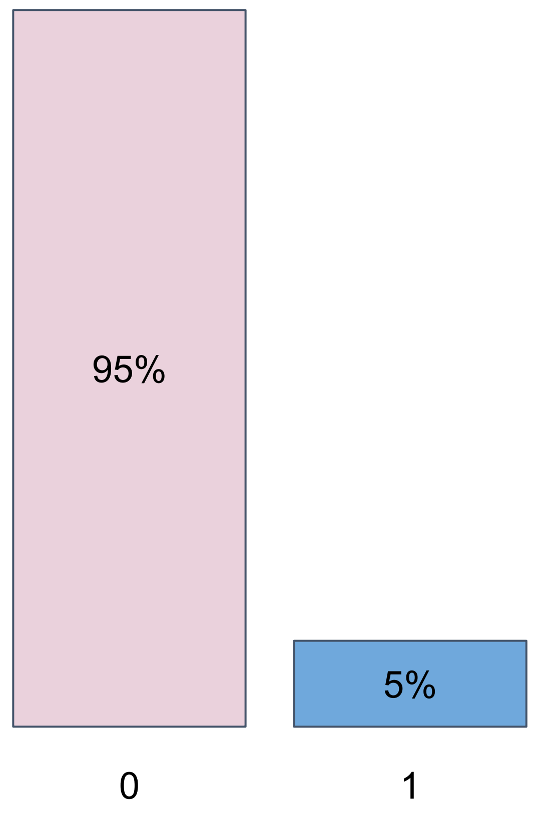
Purwadhika
Startup and Coding School

# Data Splitting - Random Sampling




Not Recommended

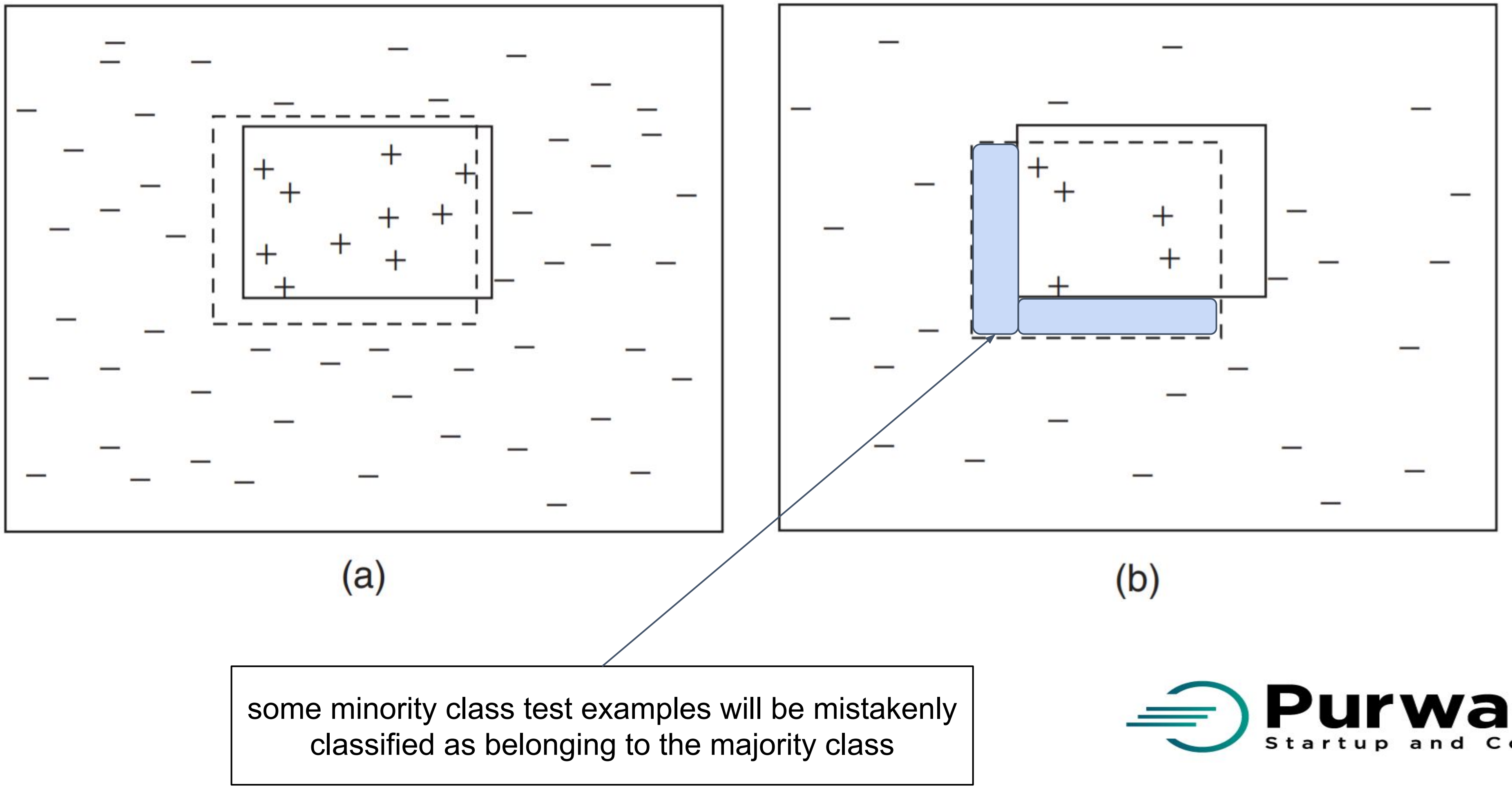
# Possibilities




| The possibilities of training set class distribution | |
|---|---|
| POS | NEG |
| ... | ... |
| 94% | 6% |
| 96% | 4% |
| 97% | 3% |
| ... | ... |
| 100% | 0% |

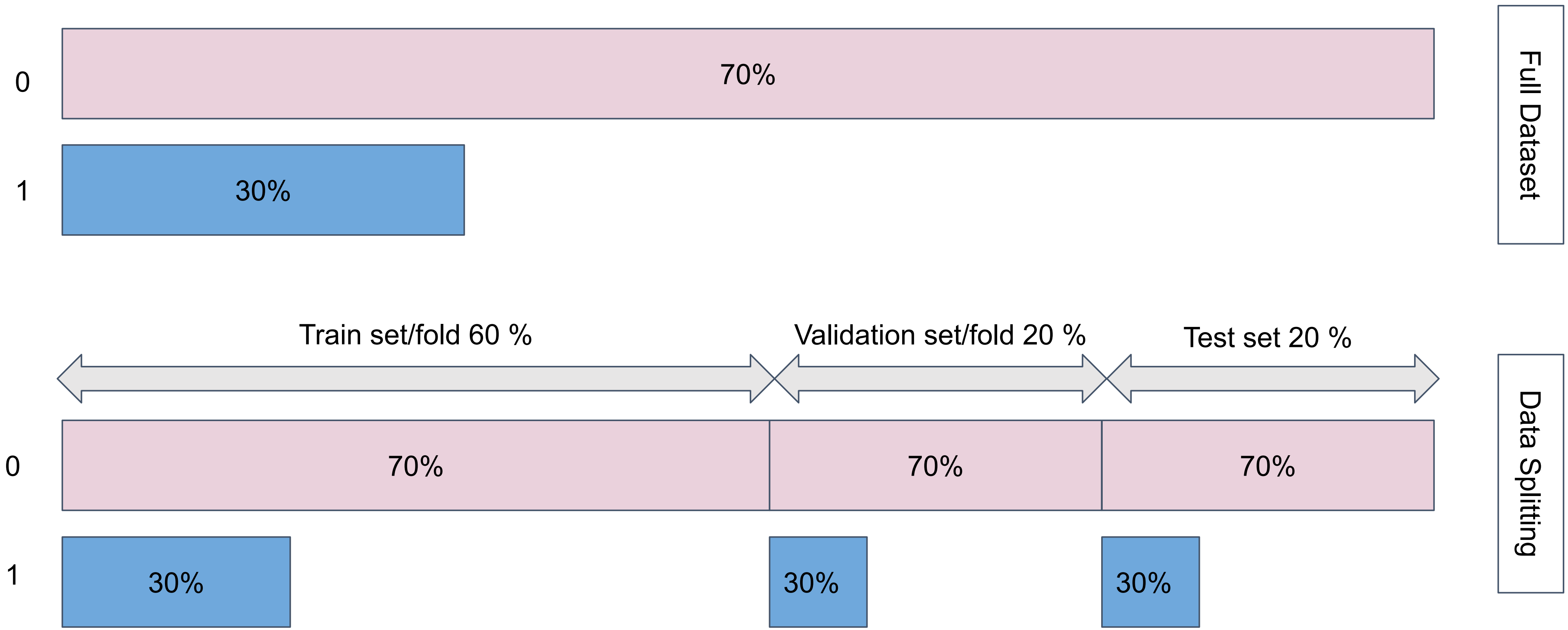
Worst case scenario

# Biased Sample Illustration




Purwadhika
Startup and Coding School

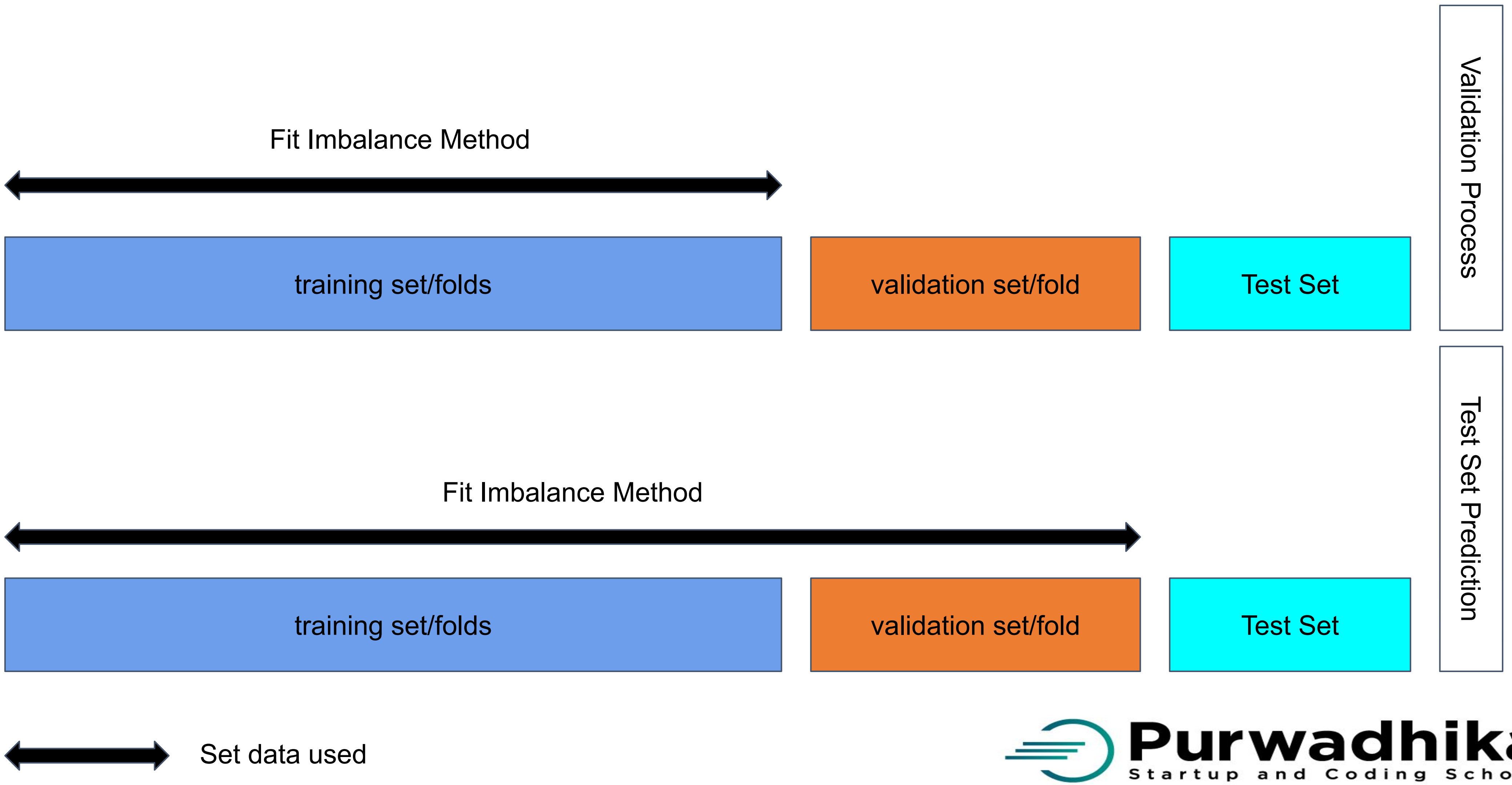
# Data Splitting - Stratified Random Sampling




Highly Recommended

Purwadhika
Startup and Coding School

# Proper Balancing Process



# Python Exercise : Combine Cross Validation with Balancing Method

Analyze data bankloan.csv

- build a logistics regression model
  - target : default
  - features : employ, debtinc, creddebt, othdebt
- Random state 2020, stratified training 60% validation 20% testing 20%
- Modeling evaluate by f1 score using Strat. CV 5 Fold:
  - Penalized logistic regression
  - logistic regression with SMOTE
- Which method is better

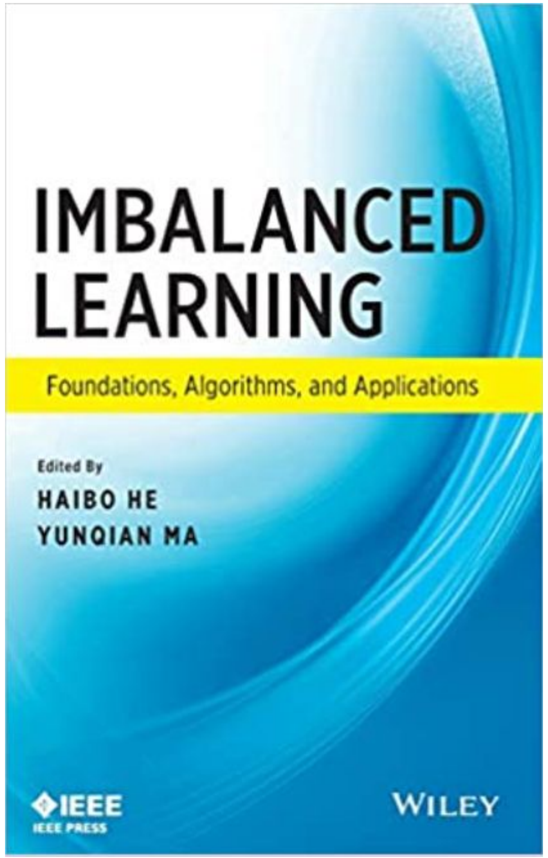
Purwadhika
Startup and Coding School

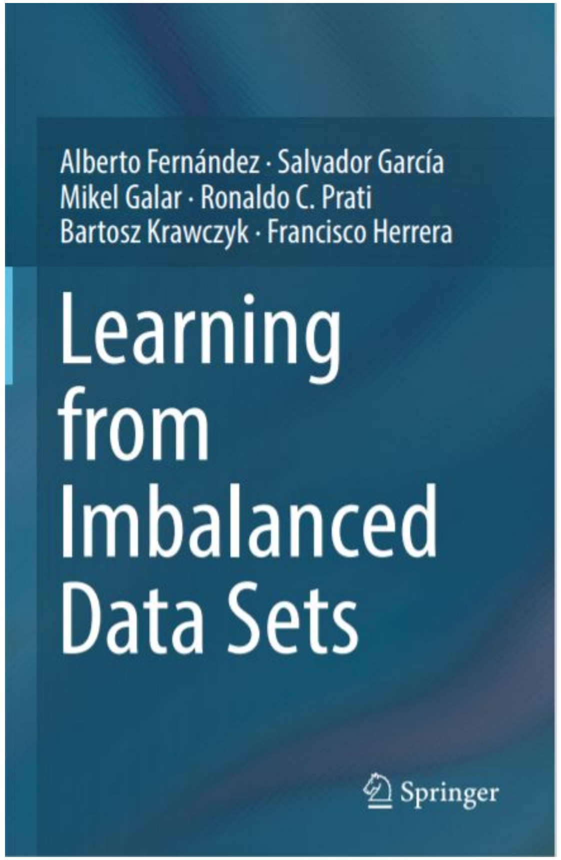
# Python Exercise : Combine Hyperparameter Tuning with Balancing Method

Analyze data bankloan.csv

- build a logistics regression model
  - target : default
  - features : employ, debtinc, creddebt, othdebt
- Random state 2020, ration 80%:20%
- Modeling evaluate by f1 score using Strat. CV 5 Fold:
  - logistic regression with SMOTE optimize the k neighbor
  - optimize c, solver
- Compare the result (before and after)

# References




Purwadhika
Startup and Coding School

# References

https://machinelearningmastery.com/what-is-imbalanced-classification/

https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

https://imbalanced-learn.readthedocs.io/en/stable/api.html

Purwadhika
Startup and Coding School