목차

INDEX

- **Part 1**

  - Introduction
  - Overview of analysis
  - Data description
  - EDA result

- **Part 2**

  - Random Sampling Techniques
  - CV results comparison

- **Part 3**

  - Resampling result
  - Summary

Introduction

- **Why do we need to solve imbalanced data problem?**

Class imbalance usually damages the performance of classifiers.

Thus, it is important to treat data before applying a classifier algorithm.

➔ When there's no treat for imbalanced data, misclassification rate of minor class as major class tends to be high.

➔ In field, the case where we classify minor class as major class is much worse than the case of major class as minor class in terms of loss amount.

- **Business practices (how to deal with imbalanced data)**

Field	Model	Method
Credit Rating Model (Bank)	Logistic regression model	Use whole sample without resampling
	Deep learning model (Multi Layer Perceptron)	Apply under-sampling method i.e. To satisfy default : non-default=1:2, Undersample non-default.
	Tree based model (Random Forest, Gradient Boosting)	Apply undersampling method or Use whole sample w/o re-sampling

Overview of Analysis

Objective

– Compare model performance by random sampling techniques for imbalanced data

– Suggest optimal sampling technique

Design and Procedure

1	Data	Find highly imbalanced data Split data into train and test set in proportion of 7: 3
2	Sampling techniques	Oversampling method : SMOTE, ADASYN Under-sampling method : Tomek link(kind of CNN), NCR
3	Modelling	Random forest (default model w/o parameter tuning) -> In order to remove the impact of model, we apply only one modeling methodology
4	Performance	• Compare confusion matrix, precision, recall, f1-score through cross validation • See the difference in results in test set before and after applying sampling techniques

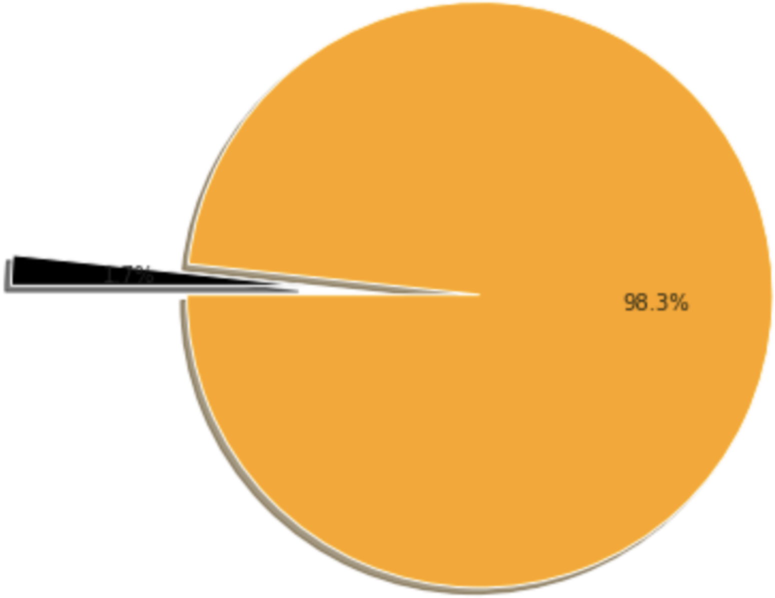
Data description

- **Credit Card Fraud Detection Dataset**

  from https://www.kaggle.com/dark06thunder/credit-card-dataset

- **Target Variable (Fraud/Non-fraud)**

Fraud (422,1.7%)



Non-Fraud (24,712, 98.3%)

Data description

- **Target Variable (Fraud/Non-fraud)**

Fraud means…

There are two types of meanings.

a. A credit card transaction is not a normal payment.
(common definition)

b. A credit card holder is fraudulent(delinquent). (our data)

Hence, fraud detection in our analysis means
detecting a card holder who will be fraudulent in the future.

Data description

- **Features (16 variables)**

No	Col name	Col label	Categorical
1	GENDER	M:Male,F:Female	O
2	CAR	Owns cars or No	O
3	REALITY	Is there a property	O
4	NO_OF_CHILD	Number of Children	
5	INCOME	Anually Income	
6	INCOME_TYPE	Occupation	O
7	EDUCATION_TYPE	Education Level	O
8	FAMILY_TYPE	Marital Status	O
9	HOUSE_TYPE	Way of living	O
~~10~~	~~FLAG_MOBIL~~	~~Is there a mobile phone~~	~~O~~
11	WORK_PHONE	Is there a work phone	O
12	PHONE	Is there a phone	O
13	E_MAIL	Is there a E-mail	O
14	FAMILY SIZE	Number of family members	
15	BEGIN_MONTH	The month of the extracted data is starting point. 0 is current month.	
16	AGE	Age of the Client	
17	YEARS_EMPLOYED	Years of working	

* FLAG_MOBIL is not used because all values are 'YES'

EDA results

- Summary statistics for our data
- Searching outliers using box-plots
- Relation between target and features

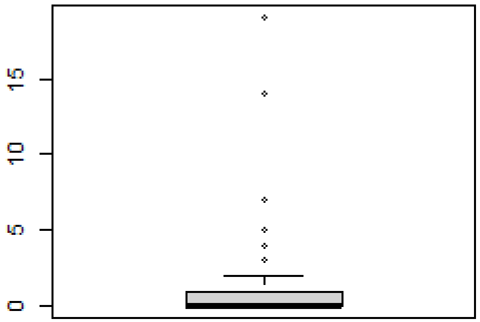
  - using graph
  - using t-test for continuous variables,

    chi-squared test for categorical variables

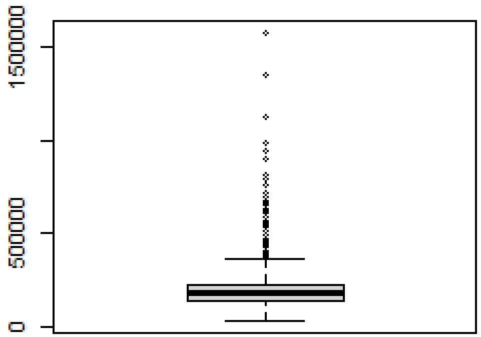
Some outliers and insignificant variables are founded, but we didn’t select (reduce) variables.
This is because we used random forest model for this study which has strength in outlier and feature selection.

EDA results

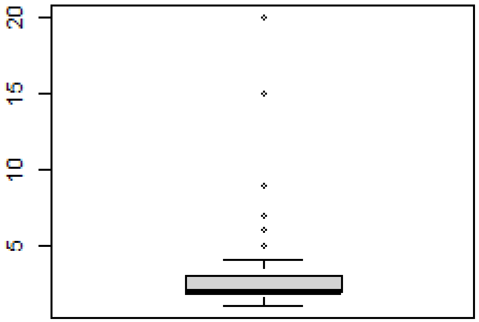
- **Searching outliers using box-plots (continuous varibles)**



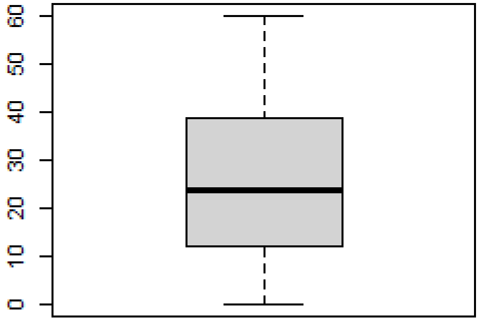
No of Child



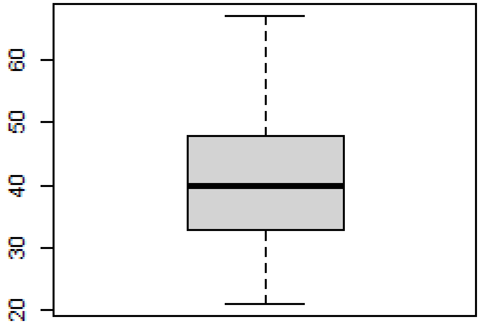
Income



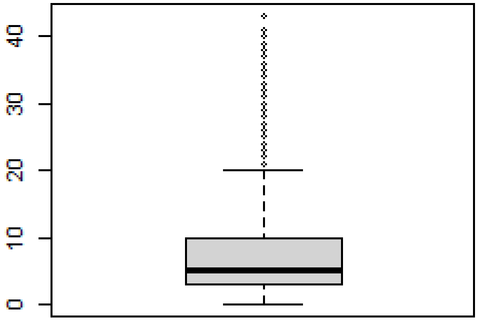
Family_Size



Begin_Month



Age



Years_Employed

EDA results

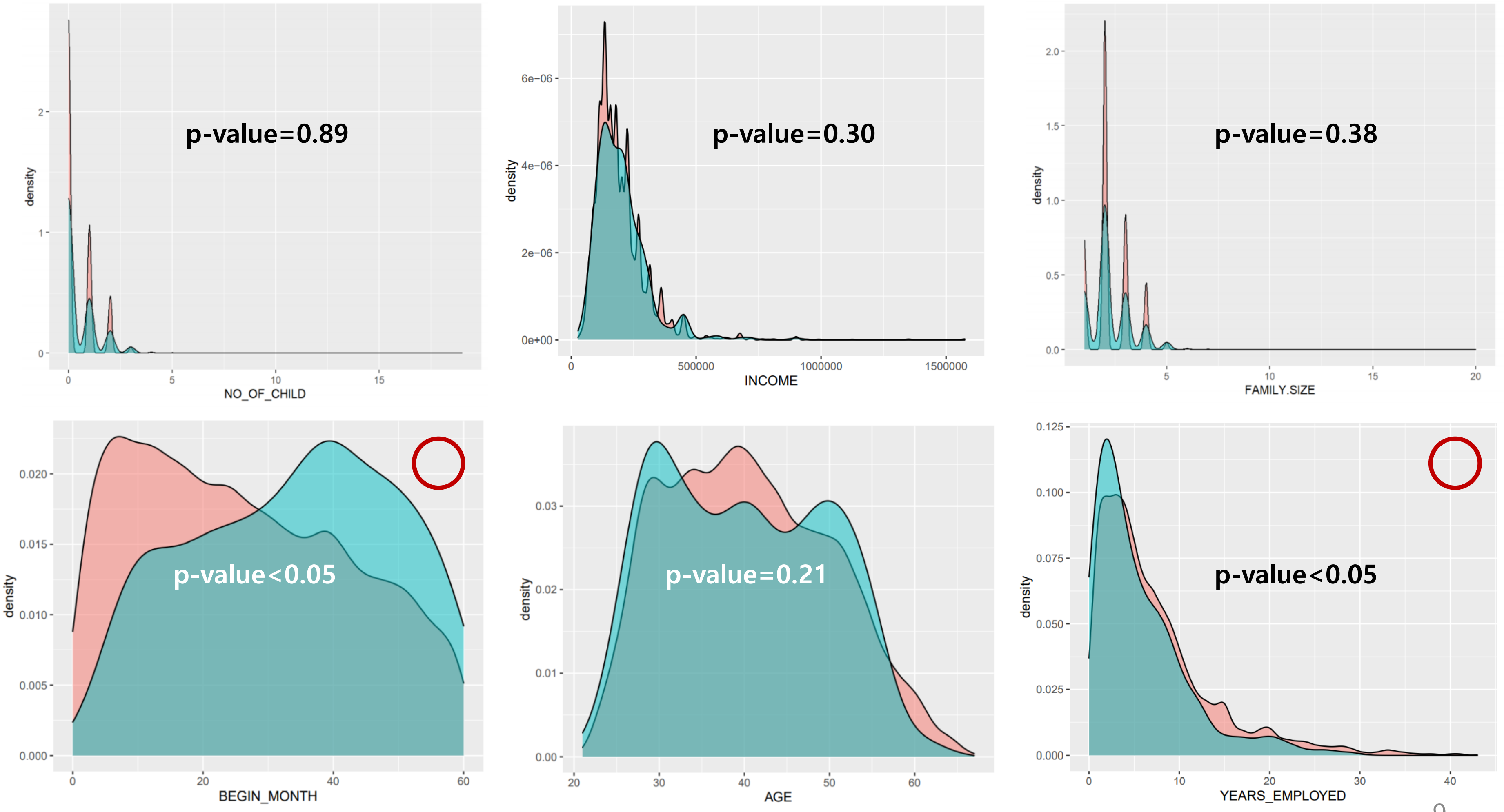
- **Relation between target and feature(continuous)**

○ 유의성검정 통과 표시

TARGET

🟥 NO

🟦 YES

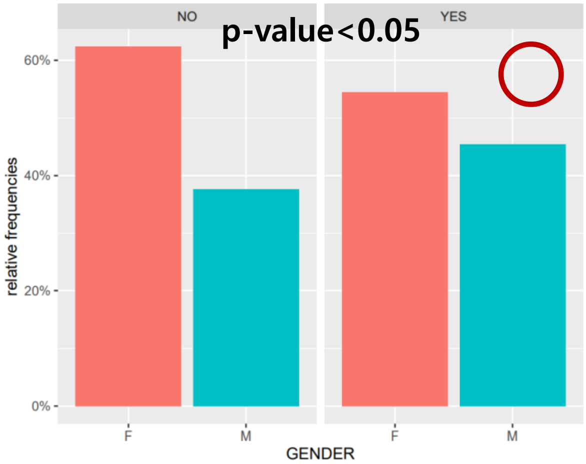


EDA results

▪ Relation between target and feature(categorical)

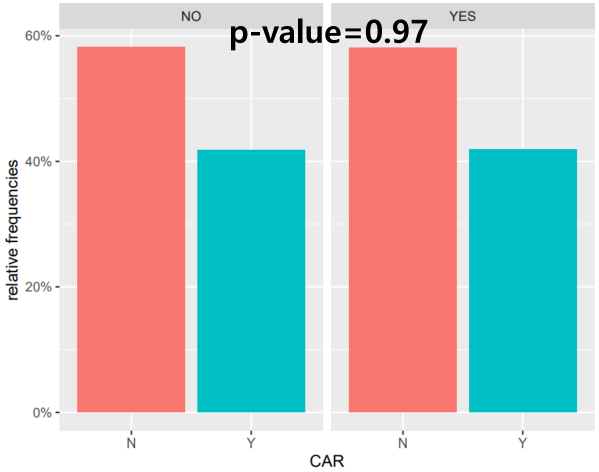
○ 유의성검정 통과 표시

Non-fraud | Fraud



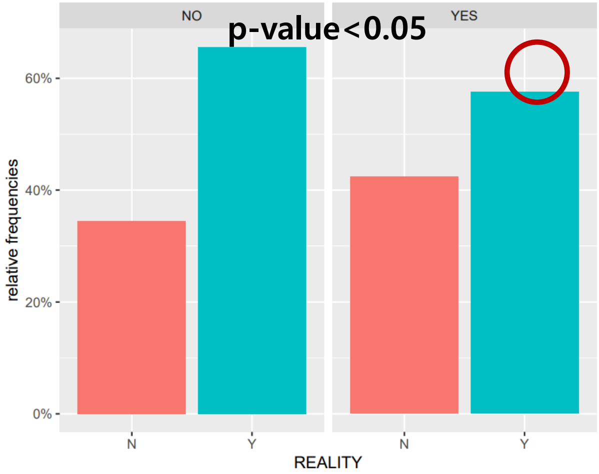
p-value<0.05

Non-fraud | Fraud

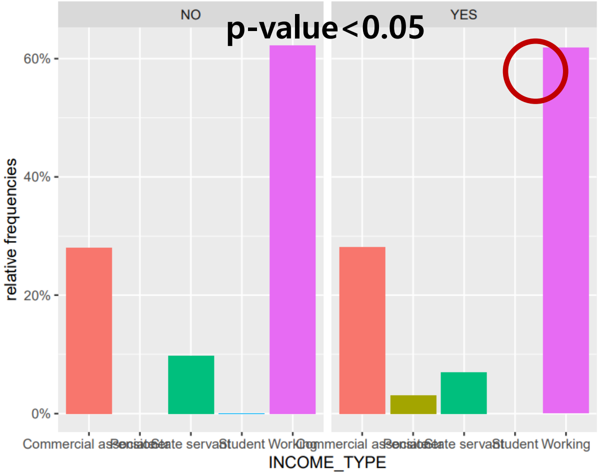


p-value=0.97

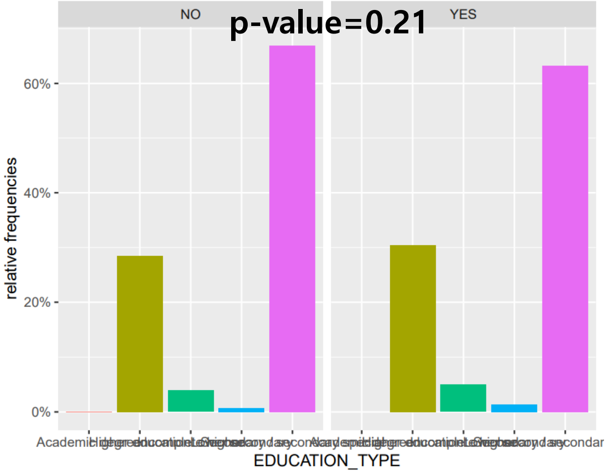
Non-fraud | Fraud



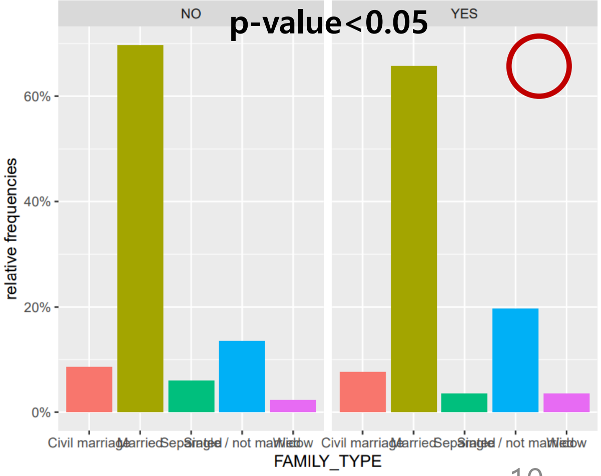
p-value<0.05



p-value<0.05



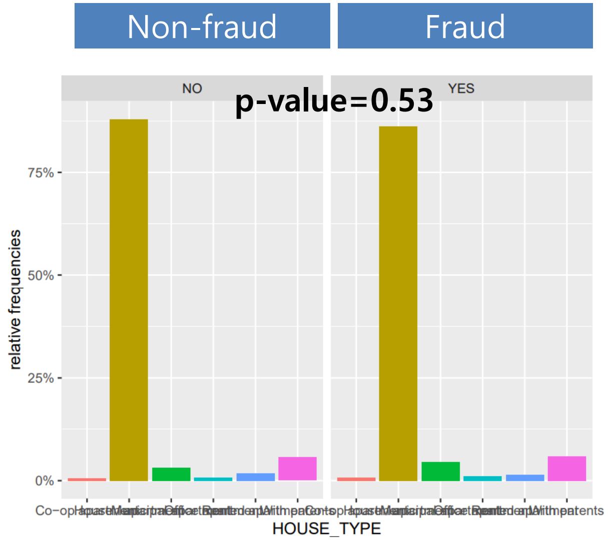
p-value=0.21

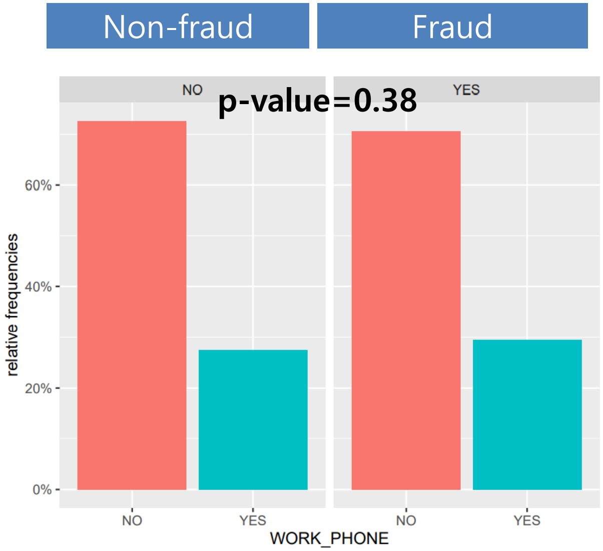


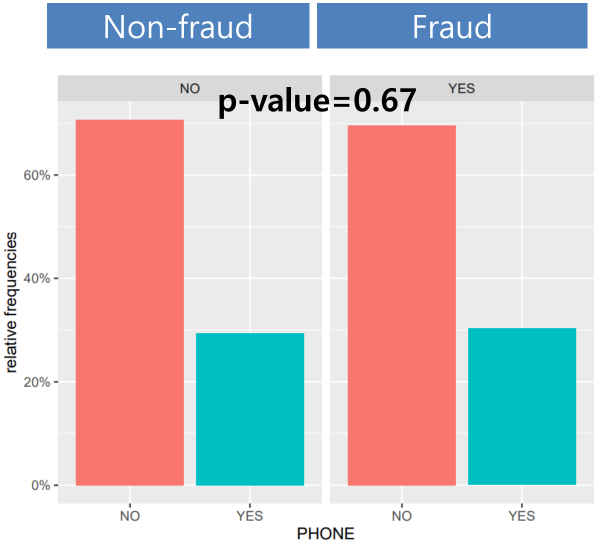
p-value<0.05

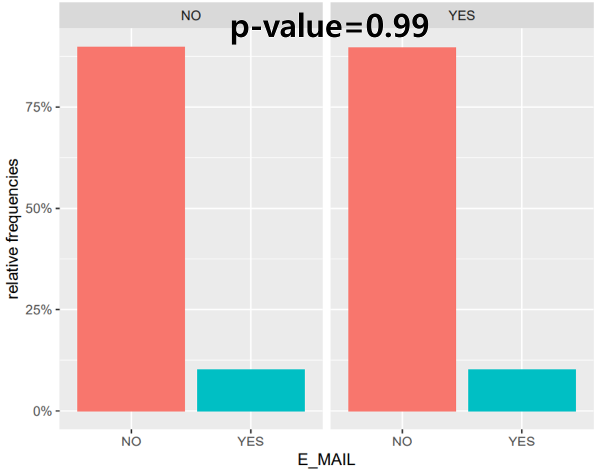
EDA results

▪ Relation between target and feature(categorical)



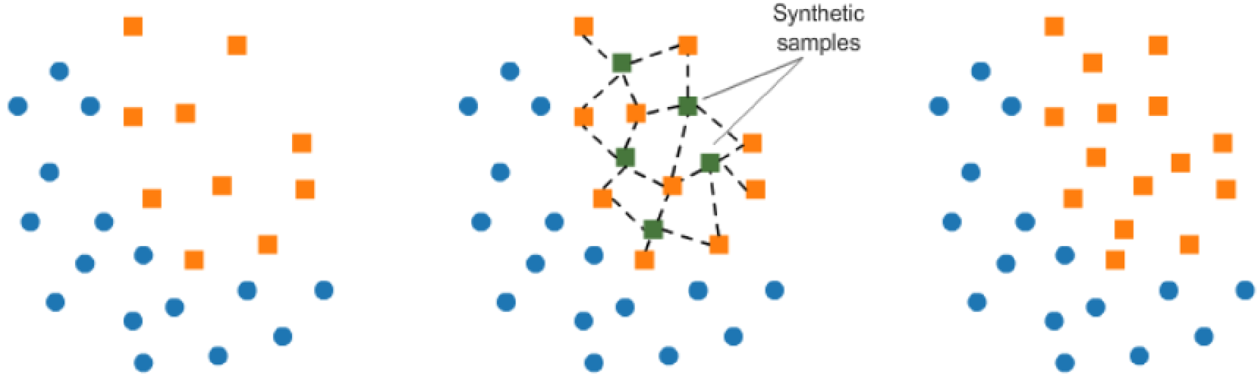






Random Sampling Techniques

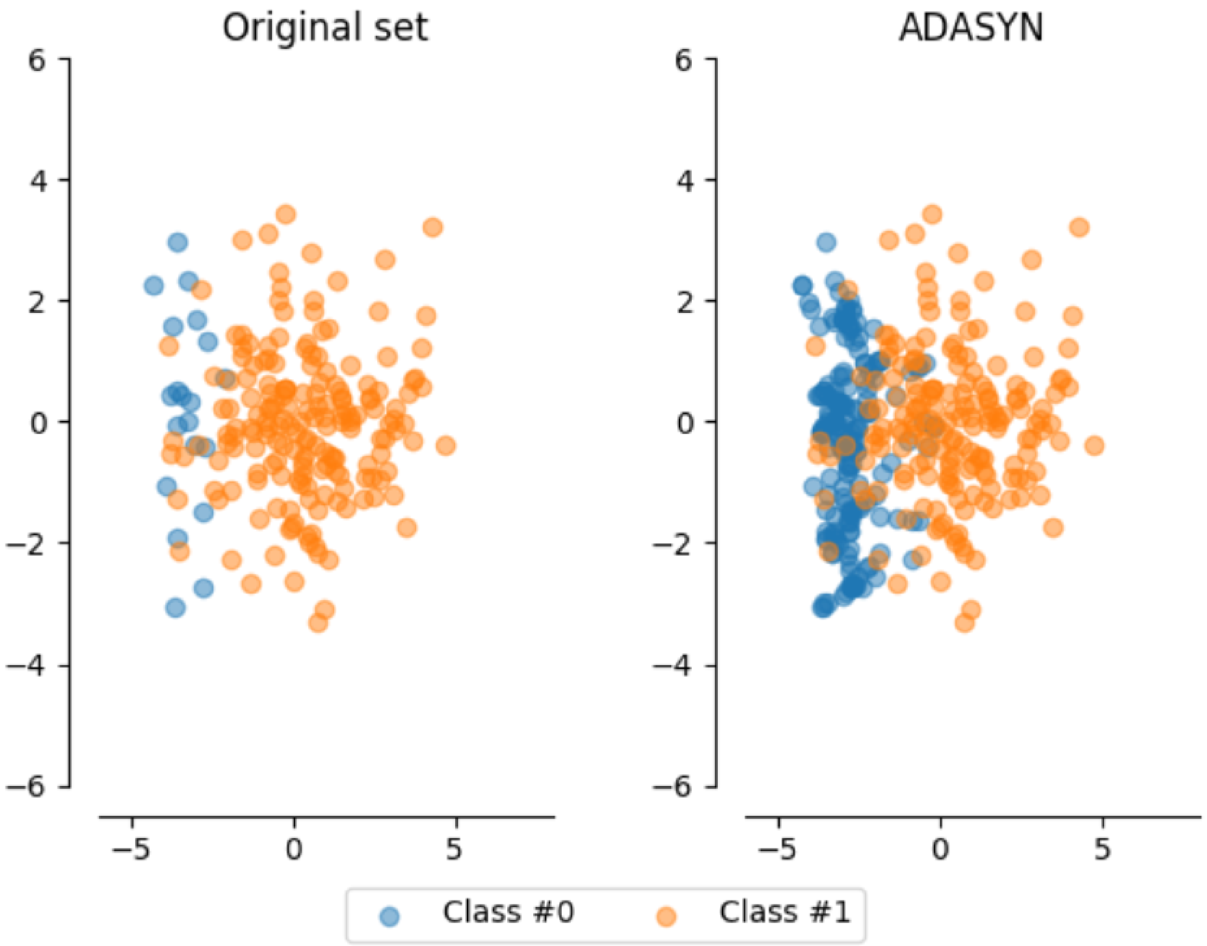
▪ SMOTE algorithm



- ✓ SMOTE stands for Synthetic Minority Over-sampling Technique
- ✓ It is based on the k-nearest neighbor method
- ✓ First, set the percent of the minority class to replicate(say 300% or 500%)
- ✓ For each datapoint in the minority class, choose the k nearest neighbors
- ✓ Then, choose a point randomly out of k nearest neighbors and create a random synthetic datapoint between the original datapoint in the minority class and the chosen nearest neighbor

Random Sampling Techniques

- **ADASYN algorithm**



Random Sampling Techniques

▪ ADASYN algorithm

- ✓ ADASYN stands for ADAptive SYNthetic sampling
- ✓ It is also based on the k-nearest neighbors
- ✓ First, for each datapoint in the minor class, find K nearest neighbors and calculate the ratio r_i defined as :

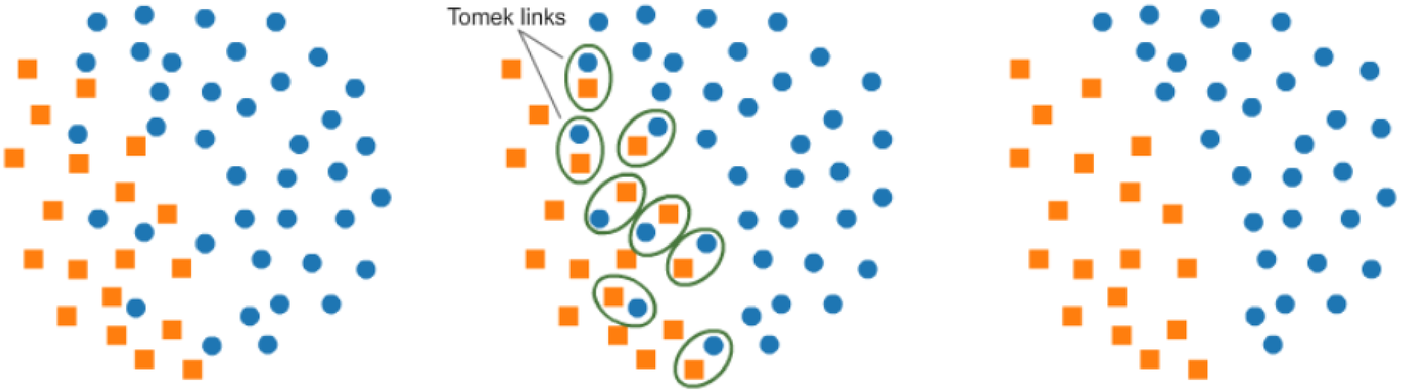
$$r_i = \Delta_i / K$$

where Δ_i is the number of examples in the K nearest neighbors of the datapoint in the K nearest neighbors belonging to the majority class. This acts as a weight.

- ✓ Then, randomly choose another datapoint in the minor class from the K nearest neighbors for the original datapoint.
  Then generate a number of datapoints between the two points in the minor class, depending on the weight.

Random Sampling Techniques

▪ Tomek link



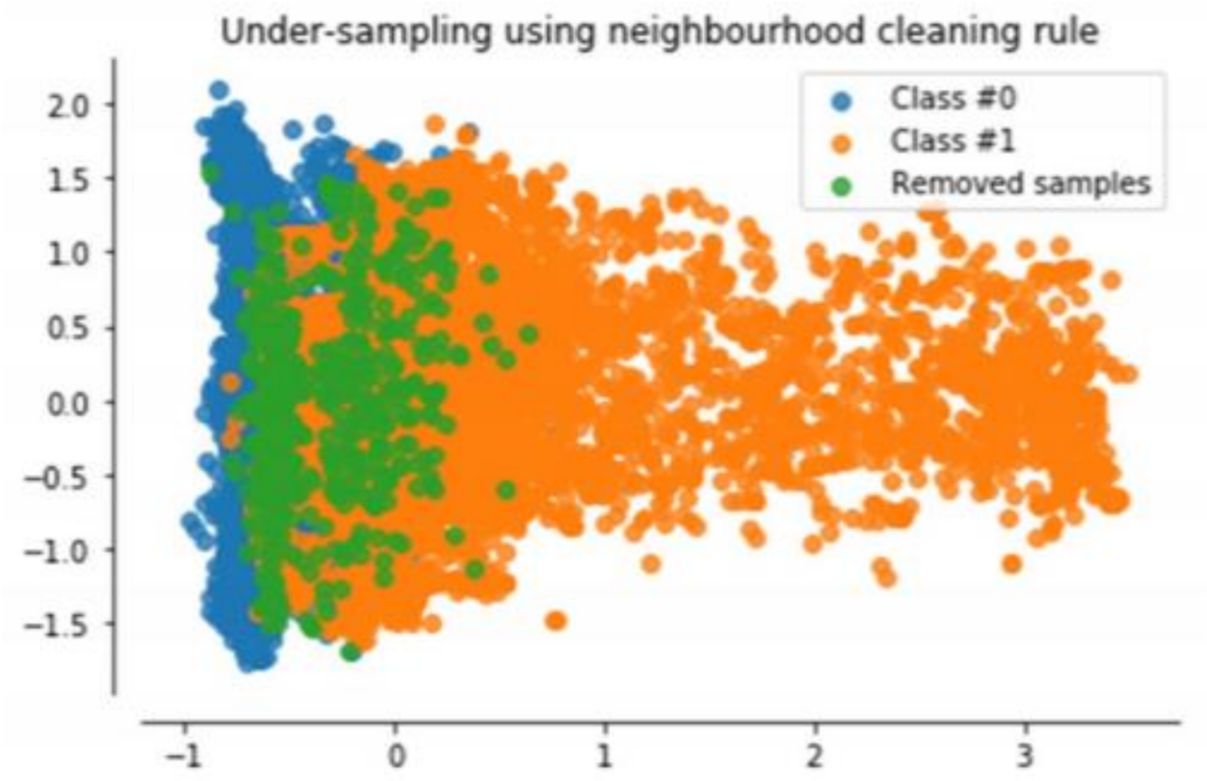
✓ It is one of a modification from CNN(Condensed Nearest Neighbors) undersampling technique developed by Tomek(1976)

✓ Tomek links method uses the rule to select the pair of observation satisfying these conditions :

1) One observation is the nearest neighbor of the other one and vice versa
2) The two observations belong to the different classes
3) Tomek link method eliminates these datapoints linked to this Tomek link in the majority class

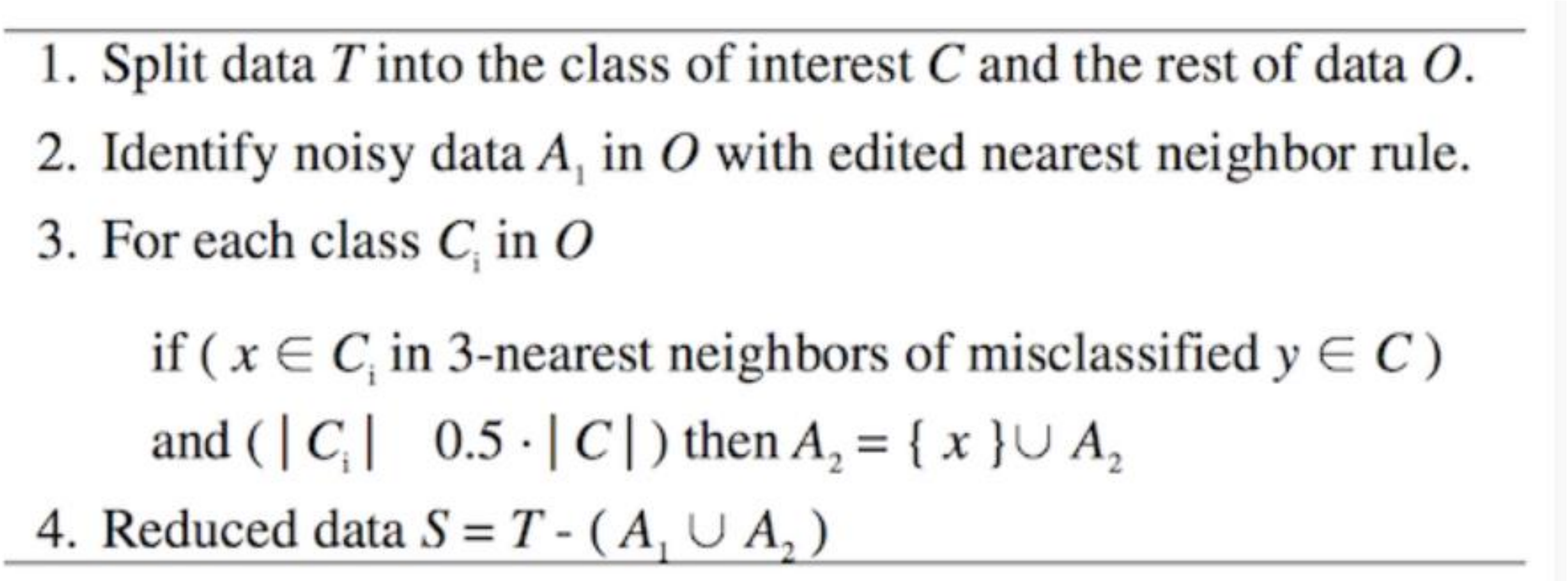
Random Sampling Techniques

- **Neighborhood Cleaning Rule**



Random Sampling Techniques

- **Neighborhood Cleaning Rule**



```
1. Split data $T$ into the class of interest $C$ and the rest of data $O$.
2. Identify noisy data $A_1$ in $O$ with edited nearest neighbor rule.
3. For each class $C_i$ in $O$
       if ( $x \in C_i$ in 3-nearest neighbors of misclassified $y \in C$ )
       and ( $|C_i| \quad 0.5 \cdot |C|$ ) then $A_2 = \{ x \} \cup A_2$
4. Reduced data $S = T - ( A_1 \cup A_2 )$
```

CV results comparison

▪ Data split and some principles

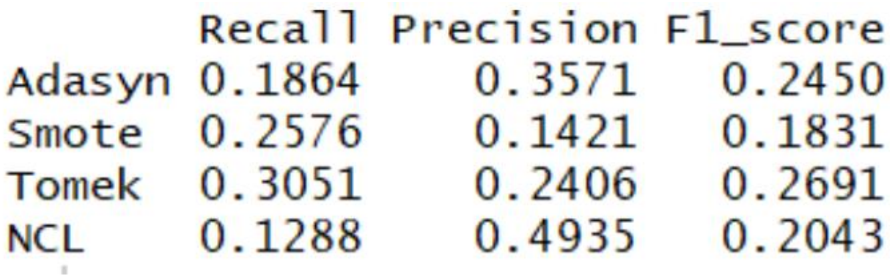
✓ First, we split the raw dataset into the training and test set with proportion 7:3

✓ Then, keeping the proportion with the majority and minority class, we split the training set into 5 sets to implement the 5-fold cross-validation

✓ Since all the random sampling techniques are based on the k-nearest neighbors, we regard all the predictors as numeric values (including dummy variables) and standardize them.

CV results comparison

- **Performance comparison**



```
        Recall Precision F1_score
Adasyn 0.1864     0.3571   0.2450
Smote  0.2576     0.1421   0.1831
Tomek  0.3051     0.2406   0.2691
NCL    0.1288     0.4935   0.2043
```

✓ Considering the indices of the performance comprehensively, we chose the Tomek sampling as the best one

Resampling result

- **Before : Original train set**

Test set	0	1	
0	7400	117	7517
1	14	10	24
	7414	127	7541

- ✓ Recall = 0.0787
- ✓ Precision = 0.4167
- ✓ F1-score = 0.1325

- **After : Tomek link for whole train set**

Test set	0	1	
0	7357	96	7453
1	57	31	88
	7414	127	7541

- ✓ Recall = 0.2441
- ✓ Precision = 0.3523
- ✓ F1-score = 0.2884

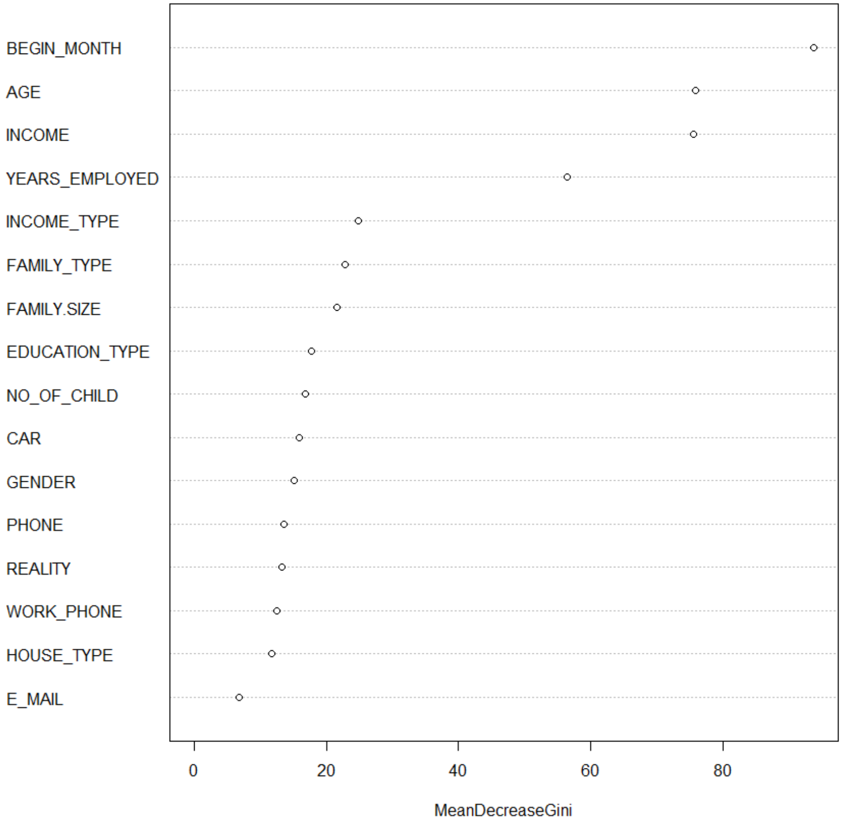
Resampling result

- ✓ Recall = 0.0787
- ✓ Precision = 0.4167
- ✓ F1-score = 0.1325

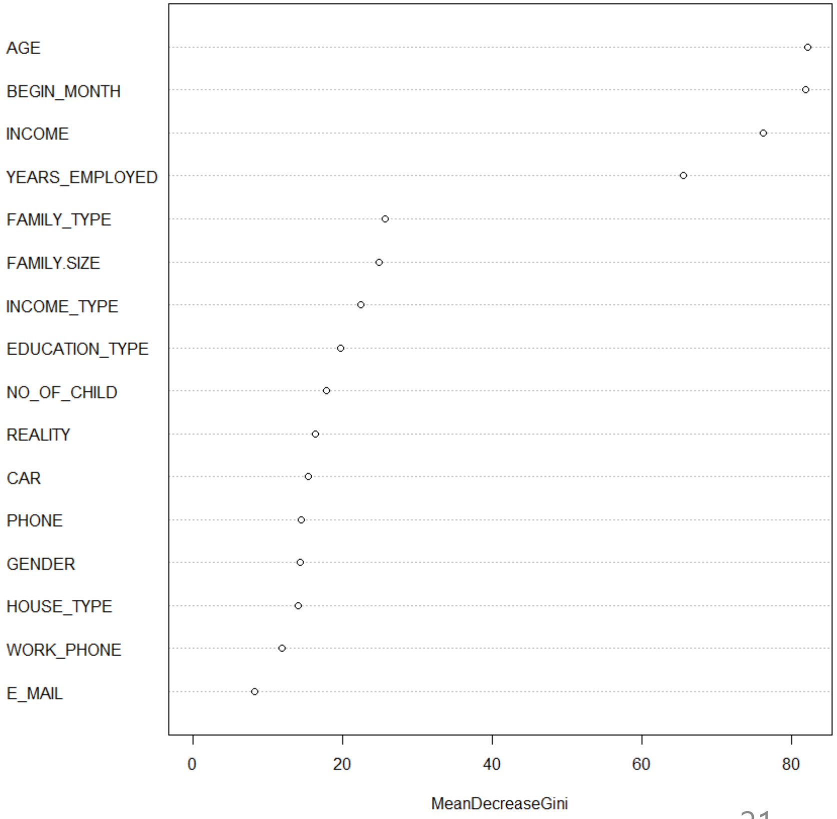
→

- ✓ Recall = 0.2441
- ✓ Precision = 0.3523
- ✓ F1-score = 0.2884

original



Tomek



Summary

- We focused on methods to improve the performance of the classifier using highly imbalanced data
- The results did not fit the purpose of our analysis when we compared each resampling methods based on “accuracy”
- Since it is important to classify true fraud as fraud, we selected ”recall(=sensitivity)” as our performance measure
- Tomek link had the highest recall, which means the true positive rate was higher than other resampling methods
- One can use any appropriate resampling methods that fits the purpose of analysis to improve model quality