CHICAGO CRIME DATA

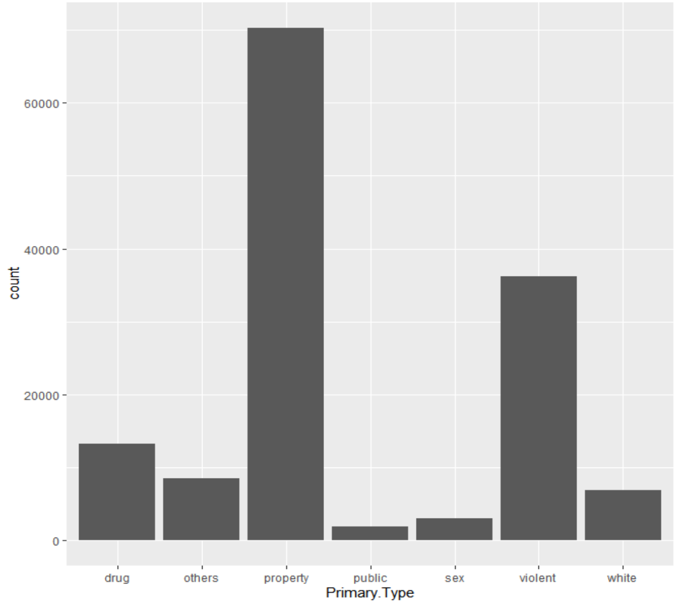
KUBIG
박소현 김효익 조송현 이영신 조규선

Problems with Imbalance in Data

Imbalance in Data – Primary.Type

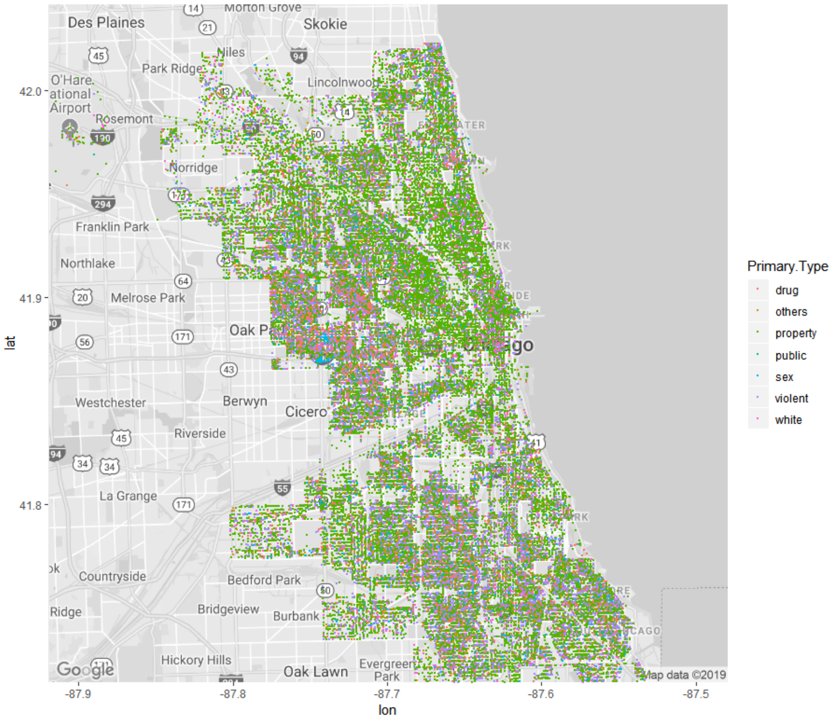
Category	Type of Crime
Violent	"ASSAULT", "BATTERY", "HOMICIDE", "INTIMIDATION", "KIDNAPPING", "CONCEALED CARRY LICENSE VIOLATION", "WEAPONS VIOLATION"
Property	"ARSON", "BURGLARY", "CRIMINAL DAMAGE", "CRIMINAL TRESPASS","MOTOR VEHICLE THEFT", "ROBBERY", "THEFT"
Sex	"CRIM SEXUAL ASSAULT", "OFFENSE INVOLVING CHILDREN", "PROSTITUTION", "SEX OFFENSE", "STALKING"
White	"DECEPTIVE PRACTICE", "GAMBLING"
Public	"INTERFERENCE WITH PUBLIC OFFICER","OBSCENITY", "PUBLIC INDECENCY","PUBLIC PEACE VIOLATION"
Drug	"LIQUOR LAW VIOLATION", "NARCOTICS", "OTHER NARCOTIC VIOLATION"
Others	"NON - CRIMINAL", " NON - CRIMINAL", "OTHER OFFENSE"

```
> table(crime$Primary.Type)

    drug   others property   public      sex  violent    white
   13158     8476    70297     1874     2959    36216     6906
```



Imbalance in Data – Primary.Type

```
chicago_map <- get_map(location=c(lon=-87.7,
lat=41.8781), zoom=11, maptype='roadmap',
color="bw")

ggmap(chicago_map)+
            geom_point(aes(x=Longitude, y=Latitude,
color=Primary.Type), data=coord, size=0.2)
```



Imbalance in Data – Primary.Type

```
crimes <- read.csv("chicago3.csv") %>%
              dplyr::select(Primary.Type, Arrest, Domestic, Ward, Year, time.tag, month, day)
train.index <- sample(nrow(crimes), nrow(crimes)*0.7)
train <- crimes[train.index,]
test <- crimes[-train.index,]
crimes_ctree <- ctree(Primary.Type~., data=crimes)
```

```
> table(predict(crimes_ctree, train), train$Primary.Type)
```

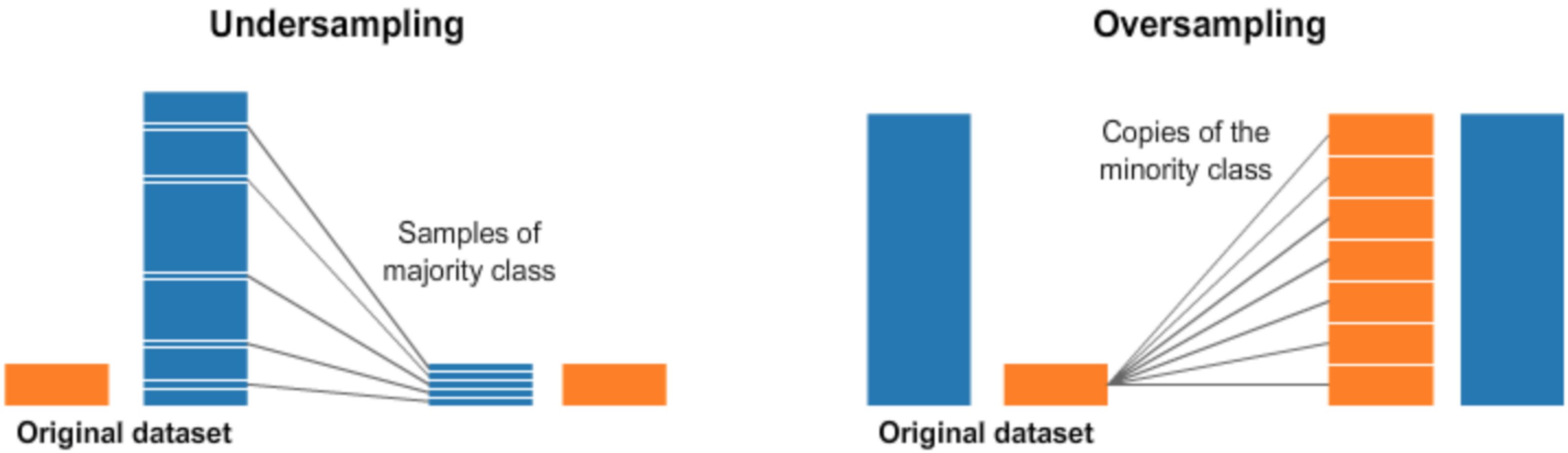
	drug	others	property	public	sex	violent	white
drug	8138	844	3883	884	601	3131	600
others	0	0	0	0	0	0	0
property	694	3210	43132	355	939	11539	4162
public	0	0	0	0	0	0	0
sex	0	0	0	0	0	0	0
violent	230	1947	2265	81	559	10669	57
white	0	0	0	0	0	0	0

```
> table(predict(crimes_ctree, test), test$Primary.Type)
```

	drug	others	property	public	sex	violent	white
drug	3682	366	1651	373	221	1378	252
others	0	0	0	0	0	0	0
property	314	1328	18378	134	401	4942	1799
public	0	0	0	0	0	0	0
sex	0	0	0	0	0	0	0
violent	100	781	988	47	238	4557	36
white	0	0	0	0	0	0	0

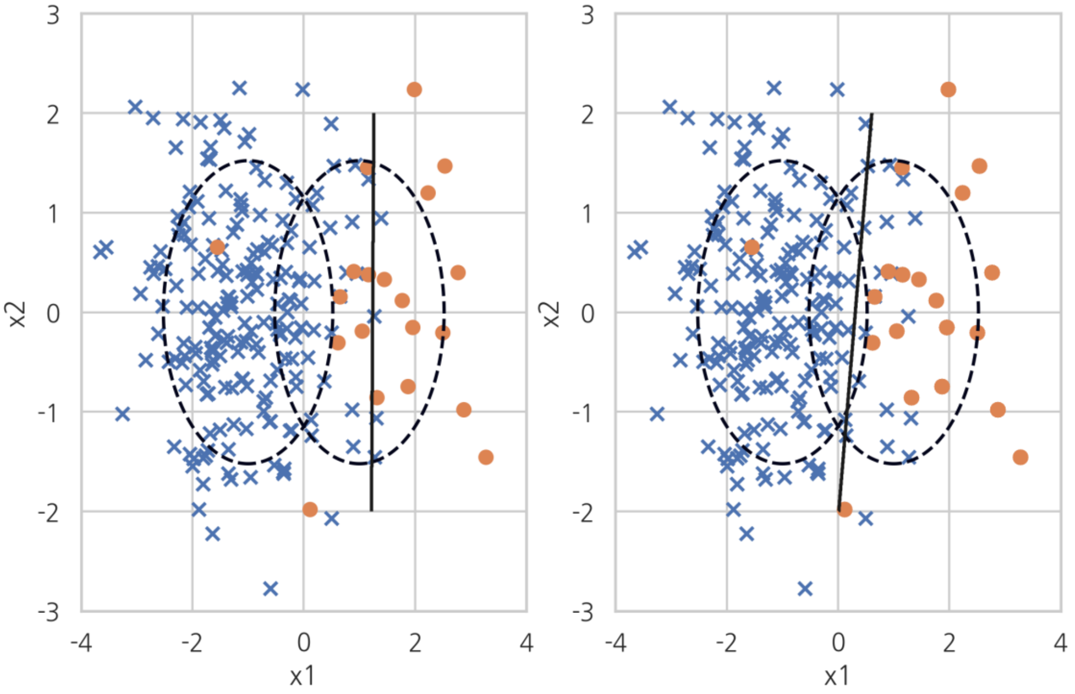
Dealing with Imbalanced Data

Oversampling vs Undersampling

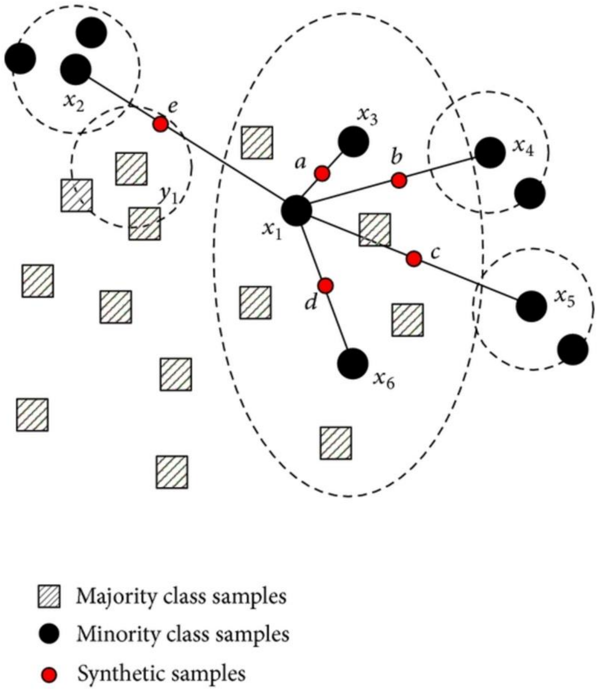


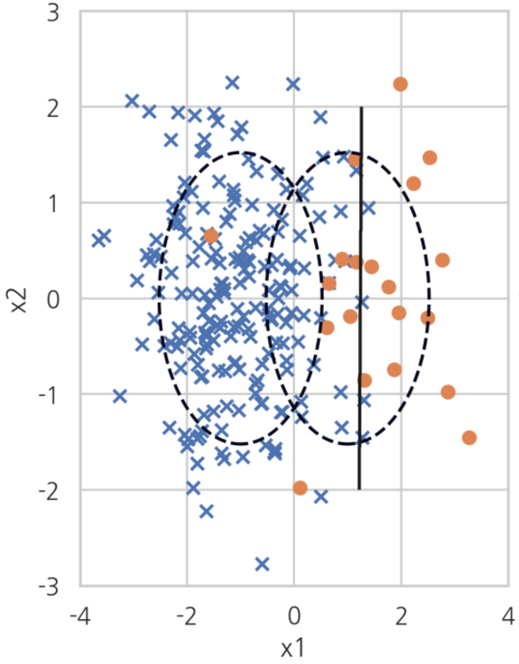
Oversampling

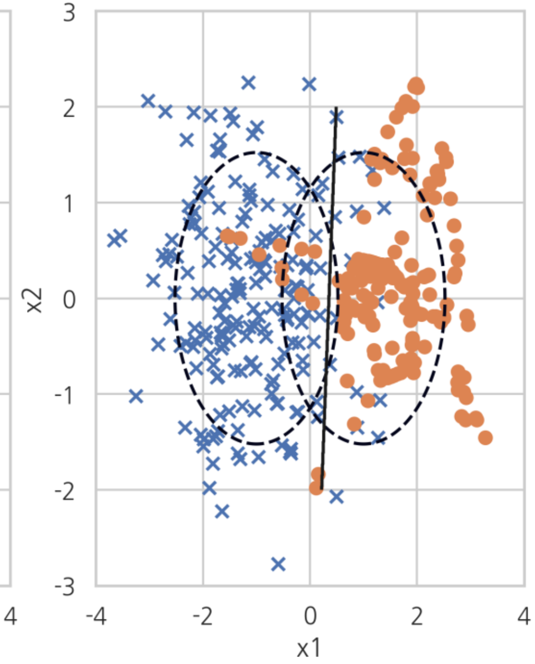
Random Oversampling



SMOTE (Synthetic Minority Over-Sampling Technique)

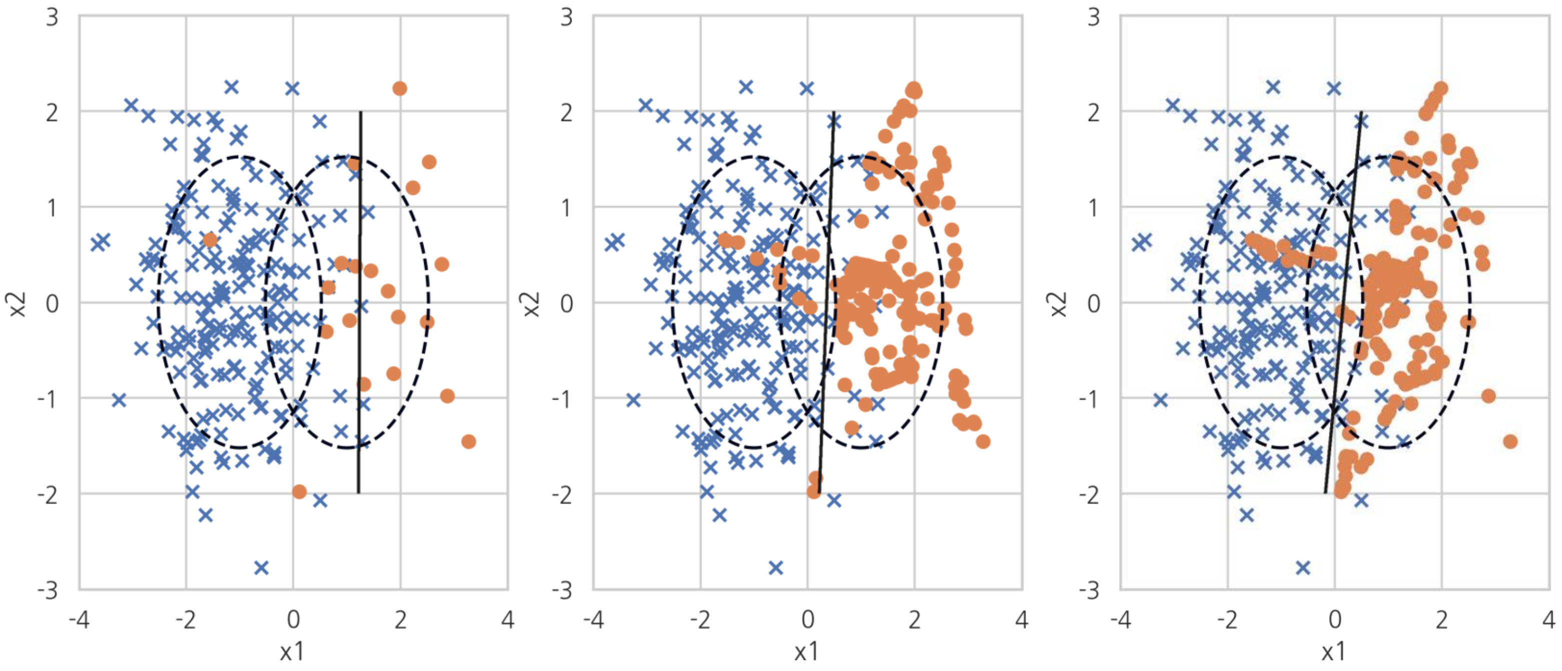






https://arxiv.org/pdf/1106.1813.pdf

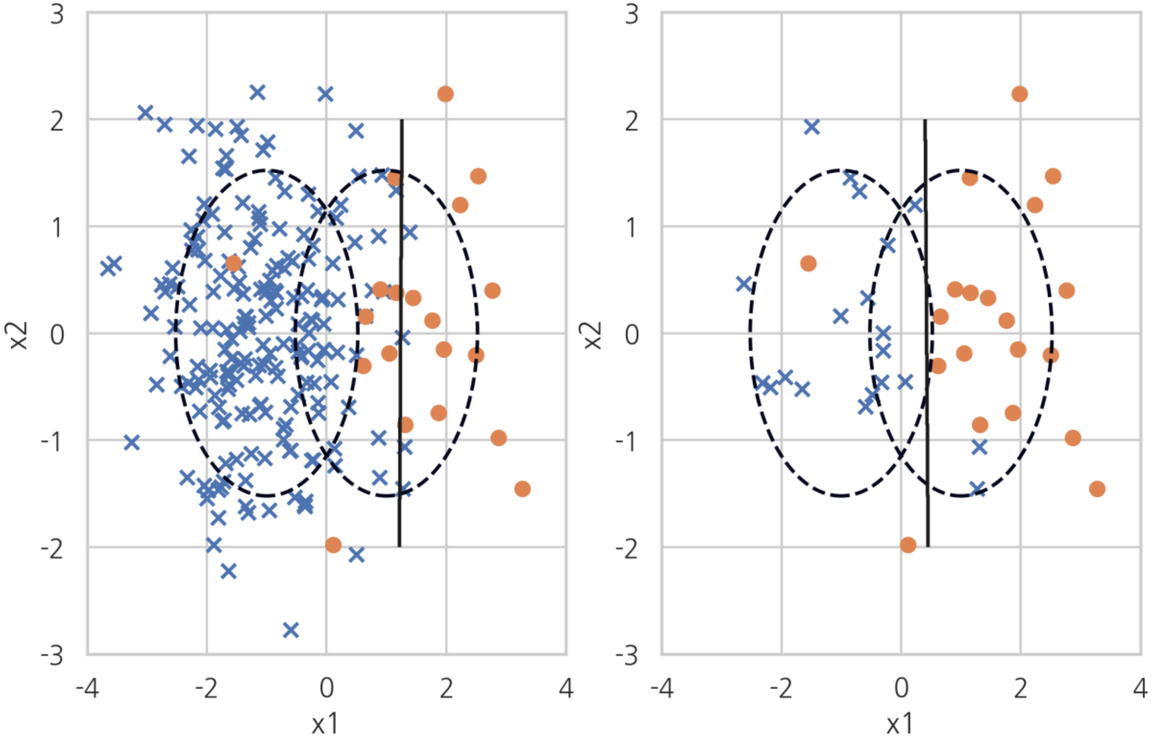
ADASYN (Adaptive Synthetic Sampling)



https://sci2s.ugr.es/keel/pdf/algorithm/congreso/2008-He-ieee.pdf

Undersampling

Random Under Sampling



CNN (Condensed Nearest Neighbor)

$\mathcal{Z} \leftarrow \emptyset$
Repeat
    For all $\boldsymbol{x} \in \mathcal{X}$ (in random order)
        Find $\boldsymbol{x}' \in \mathcal{Z}$ such that $\|\boldsymbol{x} - \boldsymbol{x}'\| = \min_{\boldsymbol{x}^j \in \mathcal{Z}} \|\boldsymbol{x} - \boldsymbol{x}^j\|$
        If class($\boldsymbol{x}$)$\neq$class($\boldsymbol{x}'$) add $\boldsymbol{x}$ to $\mathcal{Z}$
Until $\mathcal{Z}$ does not change

Figure 8.6 Condensed nearest neighbor algorithm.

CNN (Condensed Nearest Neighbor)

CNN model reduction for k-NN classifiers

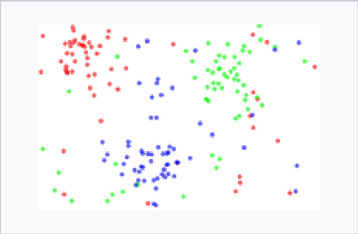


Fig. 1. The dataset.

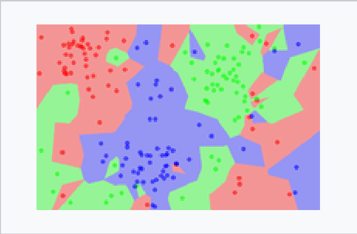


Fig. 2. The 1NN classification map.

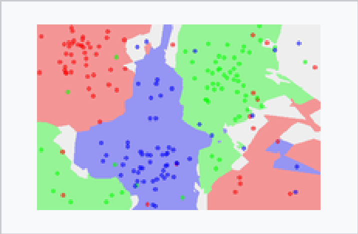


Fig. 3. The 5NN classification map.

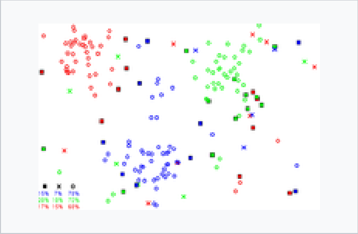


Fig. 4. The CNN reduced dataset.

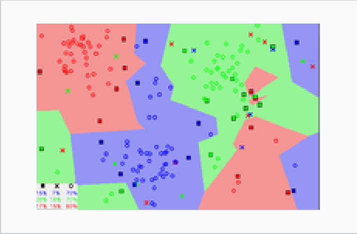
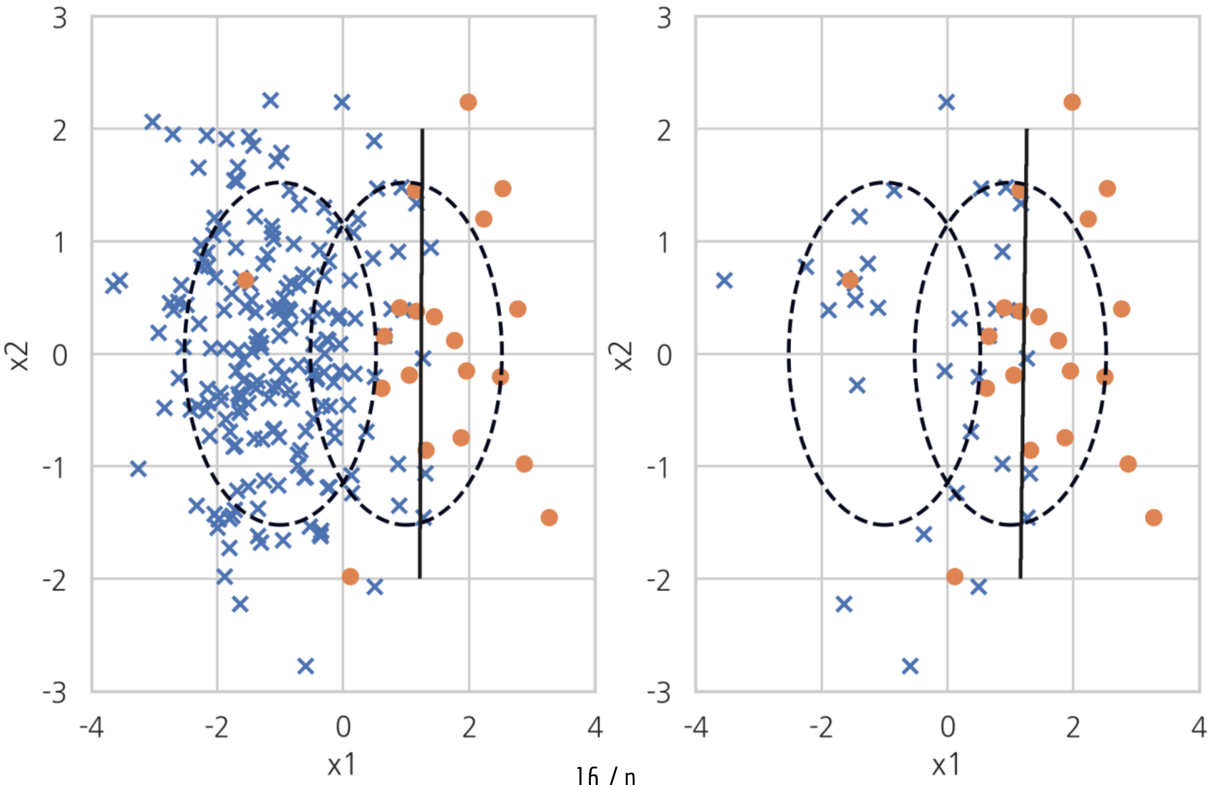


Fig. 5. The 1NN classification map based on the CNN extracted prototypes.

CNN (Condensed Nearest Neighbor)



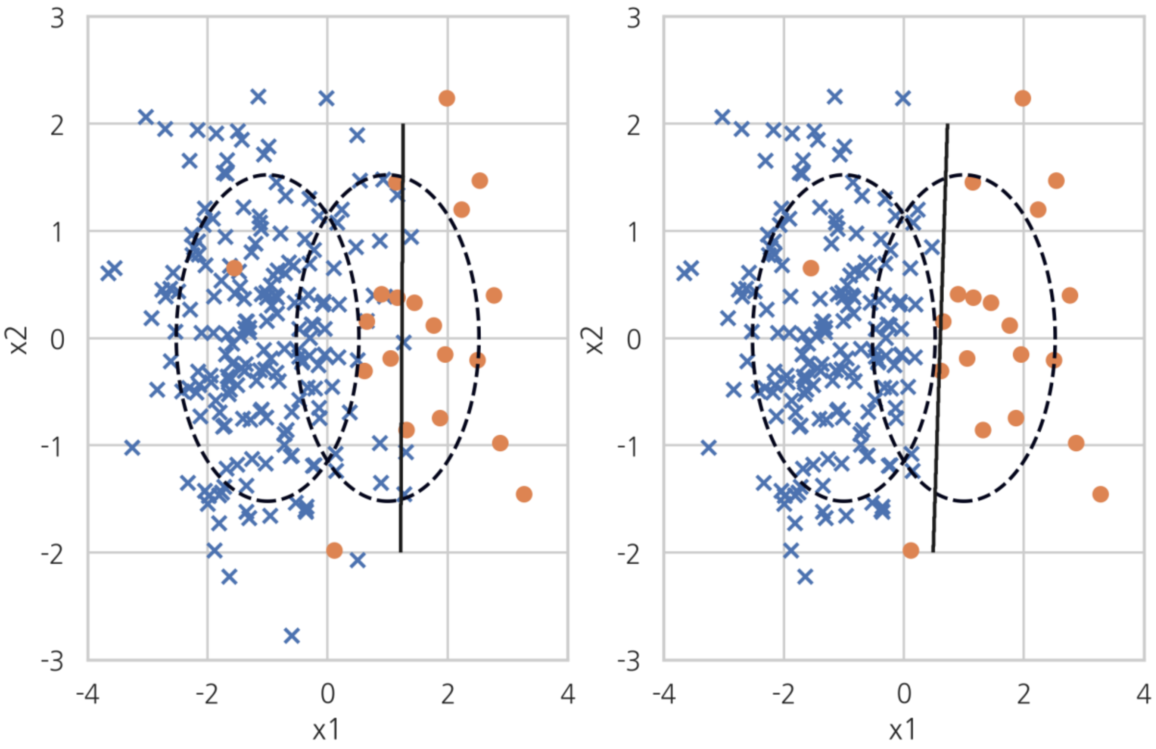
ENN (Edited Nearest Neighbor)

The ENN method proposed by [8], removes the instances of the majority class whose prediction made by KNN method is different from the majority class. So, if an instance $x_i \in N$ has more neighbors of a different class, this instance x_i will be removed. The ENN works according to the steps below:

1. Obtain the *k* nearest neighbors of x_i, $x_i \in N$;
2. x_i will be removed if the number of neighbors from another class is predominant;
3. The process is repeated for every majority instance of the subset *N*.

According to the experiments conducted in [26], the ENN method removes both the noisy examples as borderline examples, providing a smoother decision surface.

ENN (Edited Nearest Neighbor)



NCL(Neighborhood Cleansing Rule)

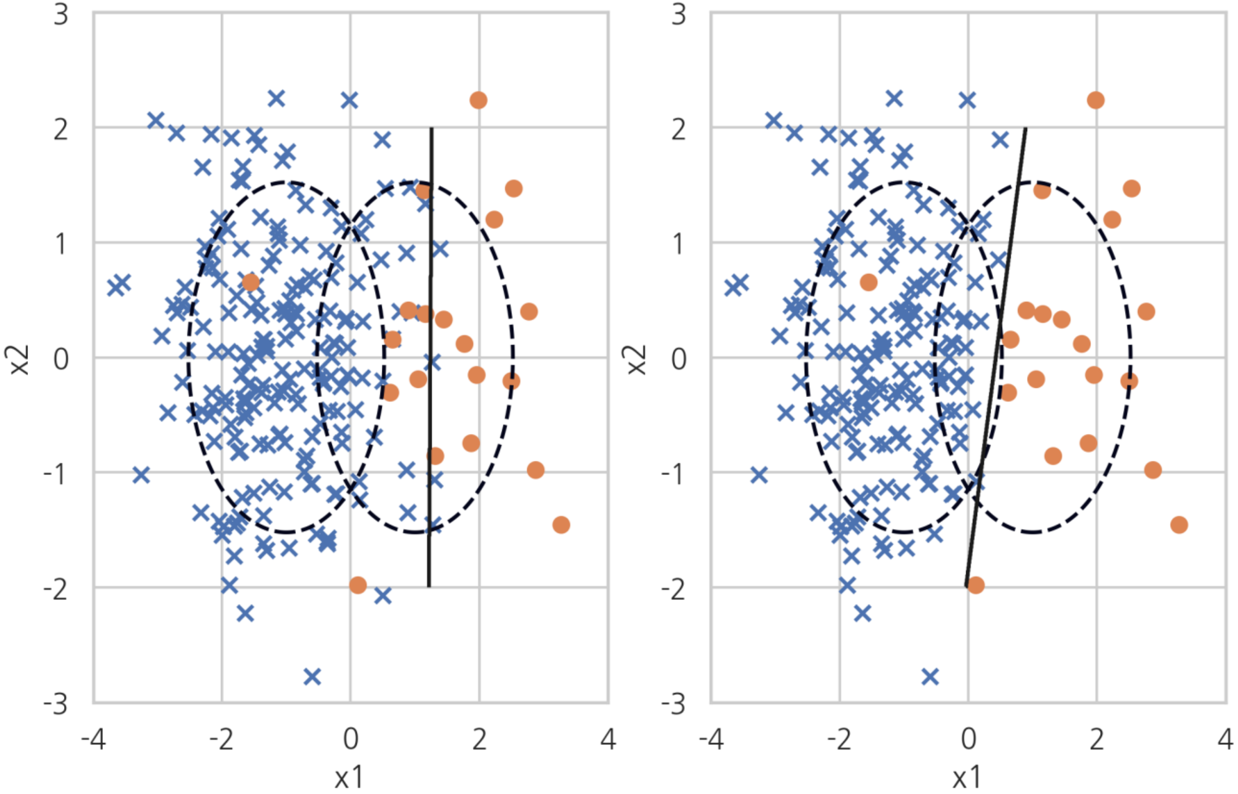
1. Split data T into the class of interest C and the rest of data O.
2. Identify noisy data A_1 in O with the edited nearest neighbor rule.
3. For each class C_i in O

   if ($x \in C_i$ in the 3-nearest neighbors of misclassified $y \in C$)

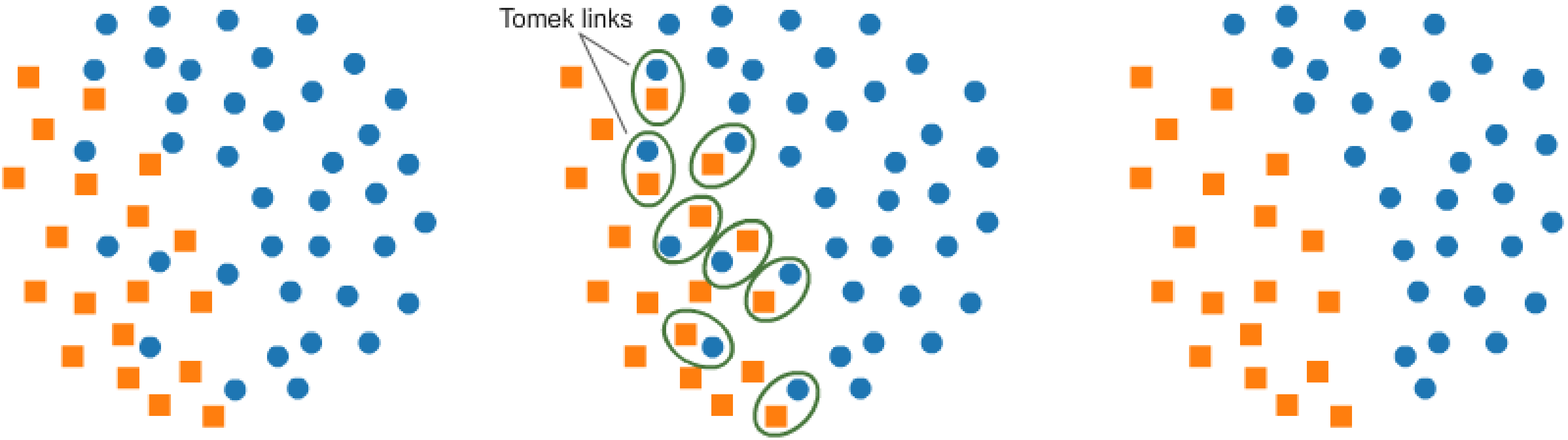
   and ($|C_i| \geq 0.5 \cdot |C|$) then $A_2 = \{x\} \cup A_2$
4. Reduced data $S = T - (A_1 \cup A_2)$

Fig. 1. Neighborhood cleaning rule

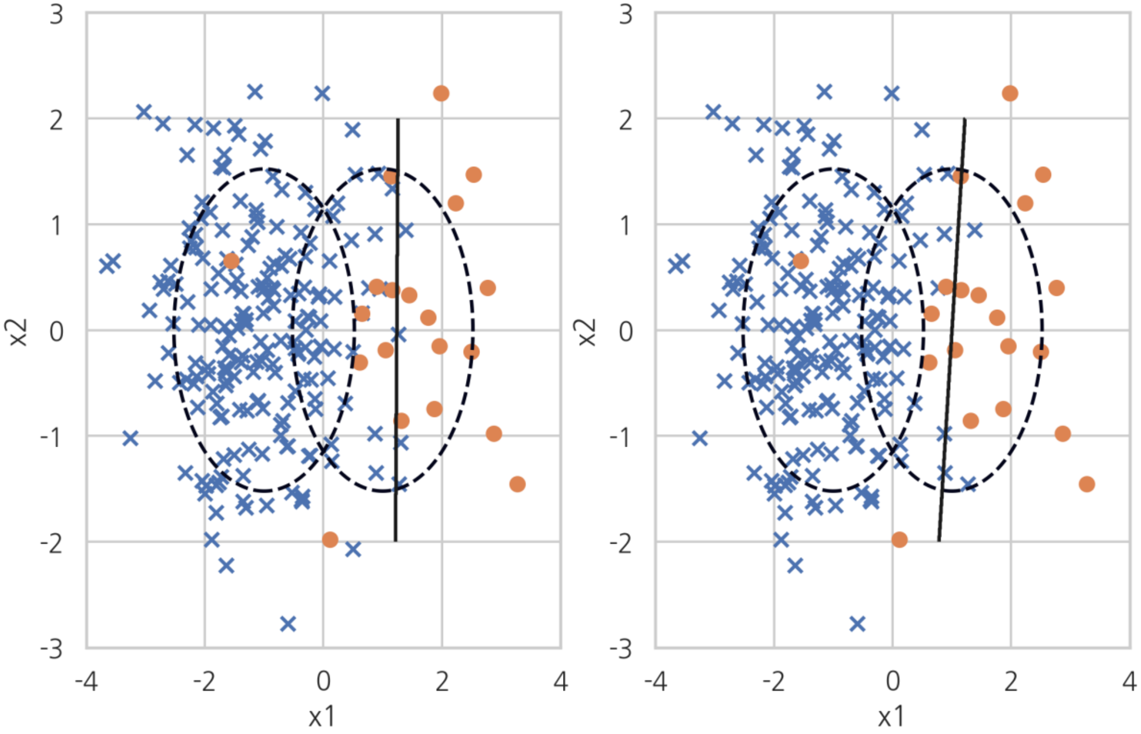
NCL(Neighborhood Cleansing Rule)



Tomek Link Method



Tomek Link Method

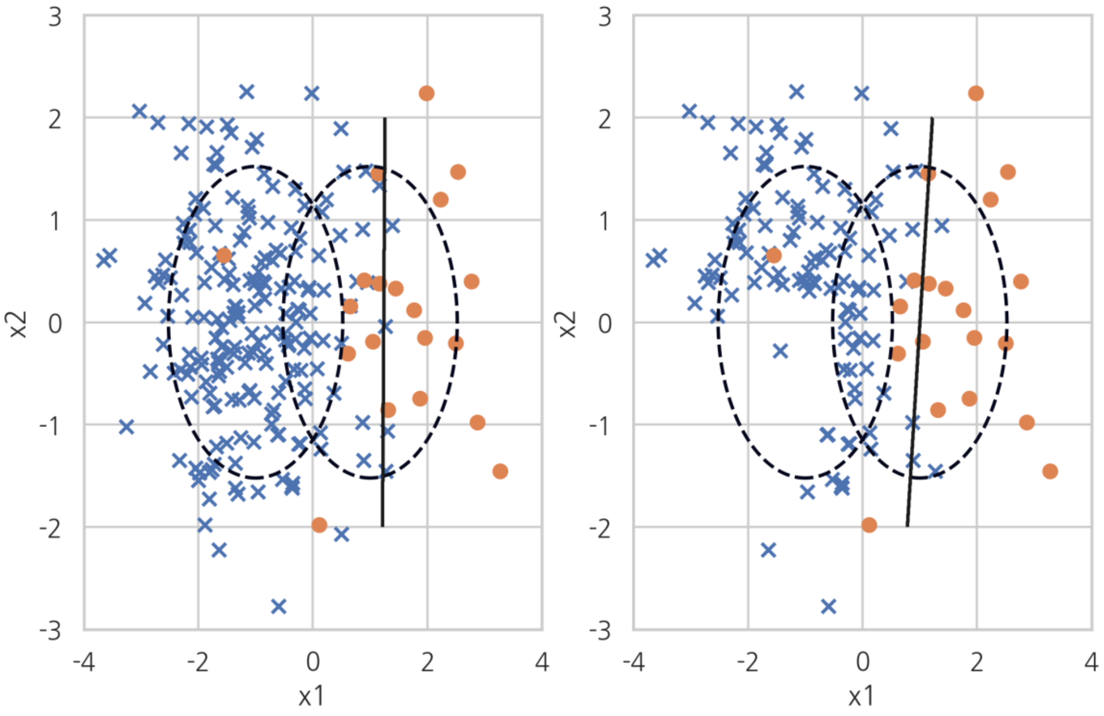


OSS (One Sided Selection)

Table 2: Algorithm for the one-sided selection of examples.

1. Let S be the original training set.
2. Initially, C contains all positive examples from S and one randomly selected negative example.
3. Classify S with the 1-NN rule using the examples in C, and compare the assigned concept labels with the original ones. Move all misclassified examples into C that is now consistent with S while being smaller.
4. Remove from C all negative examples participating in Tomek links. This removes those negative examples that are believed borderline and/or noisy. All positive examples are retained. The resulting set is referred to as T.

OSS (One Sided Selection)



THANK YOU