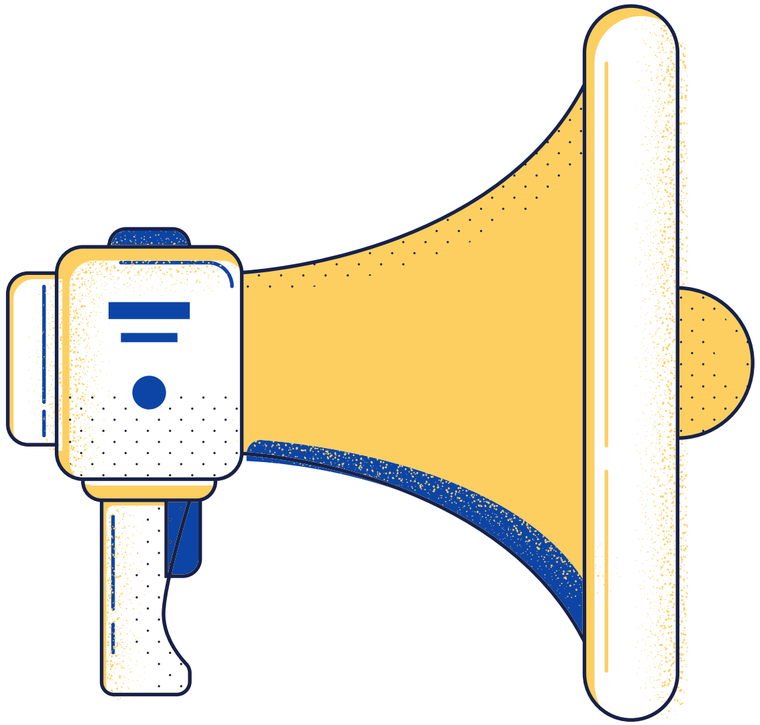
Fraud Detection

BY VIAS AULIA

Case of Imbalanced Dataset



Global losses from fraud were $32.39 billion in 2020 and projected to cost $40.62 billion in 2027*

*https://www.merchantsavvy.co.uk/payment-fraud-statistics/

Topic Outline

01 About Our Data
02 Data Preparation
03 Splitting Data
04 Handling Imbalance Data
05 Modelling

Our Goals

- Looking for best approaches to handling imbalance dataset
- Determine the model we are going to use and decide which one has higher Average Precision
- Understand common mistaked made with imbalanced datasets

01 OUR DATA

Except for the time and amount we don't know what the other columns are about (due to privacy reasons). But according to data source, these unknown columns have been scaled already.

Most of the transactions were Non-Fraud (99.8%) of the time, while Fraud transactions occurs only almost 0.2%.

```
Percentage of each class:
 0    0.998069
1    0.001931
```

01 OUR DATA

There are no Null value in the dataset so we don't need to fillna. But there are duplicate value, so we need to drop these duplicates.

```
Total each class:
 0    197982
1       383


Total each class after drop duplicates:
 0    197288
1       365
```

02

DATA PREPARATION

We want to know if there are features that influence heavily in whether a specific transaction is a fraud. According to heatmap, there are 16 features and 1 target that we will analyse furthermore

03

SPLIT DATA

We use Stratified KFold to split our data, because of our extreme imbalance data, that SKFold will take each class into consideration. As a result, each set will have same distribution of classes, or as close as possible.

"This cross-validation object is a variation of KFold that returns stratified folds. The folds are made by preserving the percentage of samples for each class."

```
for train_index, test_index in skf.split(X,y):
    print('Train:', train_index, 'Test:', test_index)
    X_trainOri, X_testOri = X.iloc[train_index], X.iloc[test_index]
    y_trainOri, y_testOri = y.iloc[train_index], y.iloc[test_index]
```

03

SPLIT DATA

After data already splitted, we scaling data because feature amount and time have not scaled yet.

```
ss = StandardScaler()
X_trainOriscaled = ss.fit_transform(X_trainOri)
X_testOriscaled = ss.transform(X_testOri)
X_trainOriscaled
```

04

HANDLING IMBALANCE DATA

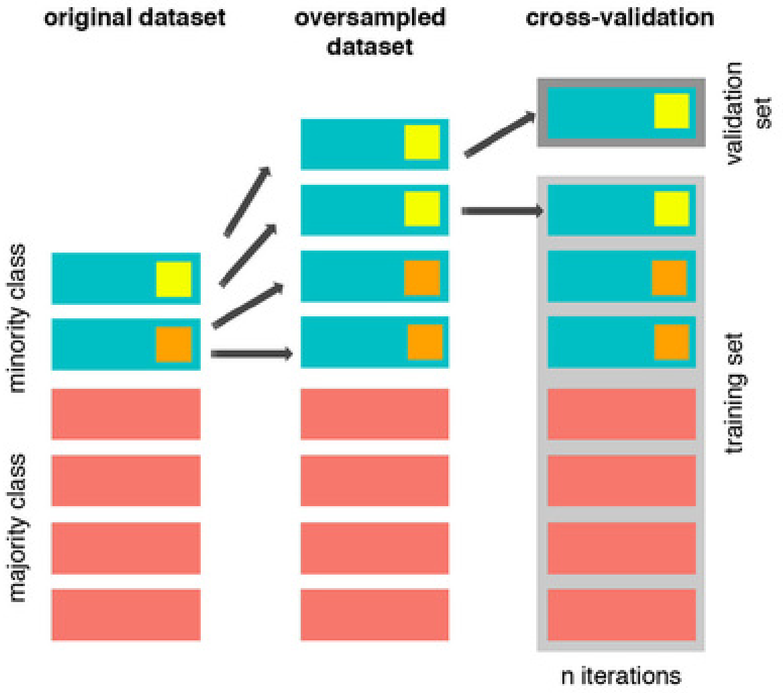
There are number of solutions to the class-imbalance problem, such as undersampling or oversampling. Both comes with plus and minus, despite of that we choose oversampling to handle this imbalance problem.

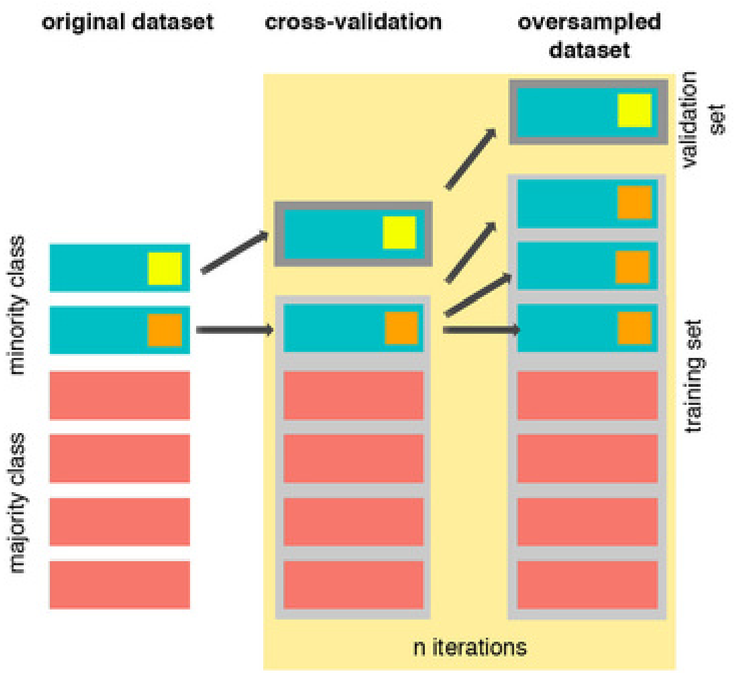
04 —

HANDLING IMBALANCE DATA

Common mistake that was made when trying to oversampling dataset was oversampling before cross validation. But the fact is, it will lead to that so-called "**Data Leakage**". Because these synthetic data due to upsampling will end up both to the training and the validation set.

Instead, we oversampling during cross validation, for this approach we can use pipeline.





04

HANDLING IMBALANCE DATA

For metric, we will use PR Curve as an alternative metric to evaluate the model when the class data is imbalanced. This metric will focus on the model to identify correctly as many positive samples as possible.

Average Precision (AP) is a single number used to summarise a Precision-Recall curve

05

MODELLING

01 Logistic Regression (Baseline)
02 Logistic Regression (Wrong Way Oversampling)
03 Logistic Regression (SMOTE)
04 Random Forest (SMOTE)
05 Decision Tree (SMOTE)
06 XGBoost (SMOTE)

05

MODELLING

Logistic Regression (Baseline)

lLet's create baseline first by Logistic Regression without SMOTE. Baseline result can tell you whether a different model is adding value or not

```
eval(baseline)
```

```
AP Score (Test): 0.7212063421968006
```

MODELLING

Logistic Regression (Wrong Way Oversampling)

We oversample training data first without being done as part of the cross validation

```
X_train_up, y_train_up = SMOTE(random_state=42).fit_resample(X_trainOriscaled,y_trainOri)
y_train_up.mean()
```

```
0.5
```

And from the metric, AP score from this model is greater than the baseline has. But remember, this model is done by wrong way of oversampling, so it's more likely because data leakage on training and test set

```
eval(best_est_naive)
```

```
AP Score (Test): 0.7753211726888084
```

MODELLING

Logistic Regression (With SMOTE)

This time, SMOTE is done as being part of pipeline, so oversampling can be done during cross validation

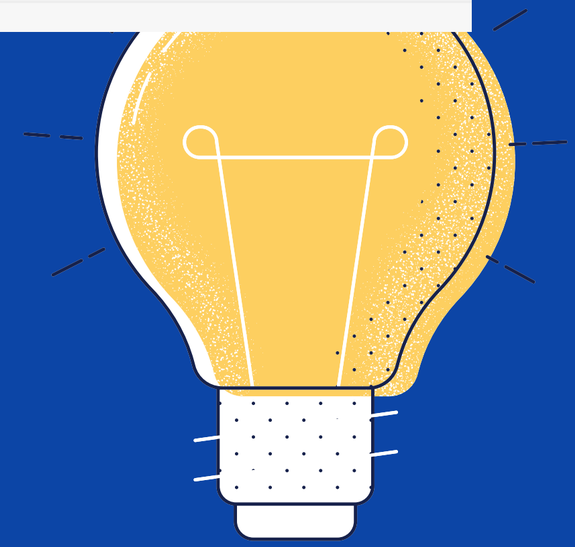
```
skf = StratifiedKFold(n_splits=5, random_state=42,shuffle=True)
sm =  SMOTE(random_state=42)
lr = LogisticRegression(random_state=42)


pipeline = Pipeline([('sampling',sm),('logisticregression',lr)])
param = {'logisticregression__' + key: log_param[key] for key in log_param}
randomcv = RandomizedSearchCV(pipeline, param_distributions=param, scoring='recall', cv=skf, n_iter=10, return_train_score=True, error_score='raise')
randomcv.fit(X_trainOriscaled,y_trainOri)
```

And the result is below

```
eval(best_est)
```

```
AP Score (Test): 0.7753212186386707
```



05

MODELLING

Random Forest (With SMOTE)

Same as before, SMOTE being done as part of Pipeline of Random Forest and the result is below

```
AP Score (Test): 0.636406976620959
```

Decision Tree (With SMOTE)

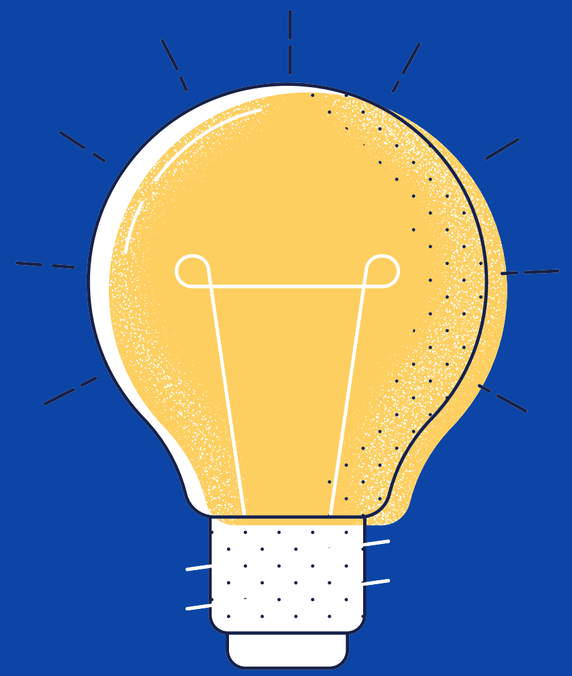
Decision Tree model with SMOTE being done as Pipeline

```
AP Score (Test): 0.4499137530561583
```

XGBoost (With SMOTE)

XGBoost model with SMOTE being done as part of Pipeline

```
AP Score (Test): 0.3746646466372561
```



Conclusion

According to AP Score, Logistic Regression with SMOTE provides best performances than other models. Applying SMOTE before cross validation will lead to data leakage. Instead, apply SMOTE during cross validation, we can use Pipeline to make things easier.

Conclusion

There are number of ways to handling imbalance problem with all pros and cons of different approaches.

So it better to carefully consider whether to use undersampling or oversampling, or instead use other alternative include adjusting the weight of classes or adjusting the decision threshold.

THANK YOU

For all of you who read this far, I do hope this presentation give you some information, or as a refresher or perhaps can be a topic discussion (can't wait!!!).

This project still far from perfect--let alone be good practice, I really appreciate for any of thought, comment, or feedback.

REFERENCES

Classification pr curve

Credit Fraud Dealing With Imbalance Dataset

Fraud Detection Handbook

Average Precision

How to do cross-validation when upsampling data

Kotsiantis, S., Kanellopoulos, D., Pintelas, P. 2006. Handling imbalanced datasets: A review