The purpose of this notebook is to experiment with different resampling technique in an imbalanced sample problem and not aimed at finding the best model to predict fraud. This will be followed up with a separate exercise.

Source of data: Kaggle dataset for credit card fraud. The file is too big to store up here, but can be found on this link:

https://www.kaggle.com/mlg-ulb/creditcardfraud

Summary of findings:

I experimented with undersampling, oversampling and custom sampling of the dataset. I aimed to find as many fraud cases as possible without significantly increasing the number of false-positive predictions. It might result in significant loss to the business if ordinary transactions are blocked. Therefore, I focused on precision scoring, however, I included accuracy scoring to provide further insight.

The best results came out with the undersampling technique, although no resampling at all also performed acceptable, in terms of not generating a high number of false positives.

Logistic regression on the undersampled data found 83 out of the 98 fraud cases, and falsely predicted fraud in 51 cases only. The cross-validated precision score on the training data was 1. I found two other models (with different regularization strength) where the model resulted in the same (1) precision score but achieved a bit better result on this particular test set. They both found 80 out of 98 fraud cases but predicted fraud in only 17 or 18 other cases, where the true label was ‘regular’ (not fraud) transactions.

Details of

The data set contains transactions that occurred in 2 days.

It contains only numerical input variables which are the result of a PCA transformation. Which means that I could not make an intuitive analysis of the features. And also, PCA means that correlations between the features are also eliminated.

As a first step, I inspected the data and looked for missing values. The dataset is clean in this sense.

Then, I created a time difference feature based on the available time feature.

I compared the values of the fraudulent and non-fraudulent transactions. The dataset contains more of the low-value fraud transactions, which is in line with my expectation considering credit card transactions.

I then identified the number of transactions and their proportion in each class:

0 284315 (99.83%)

1 492 ( 0.17%)

The dataset is highly imbalanced.

I chose a logistic regression model with the ‘l2’ penalty to compare the effect of different resampling technique.

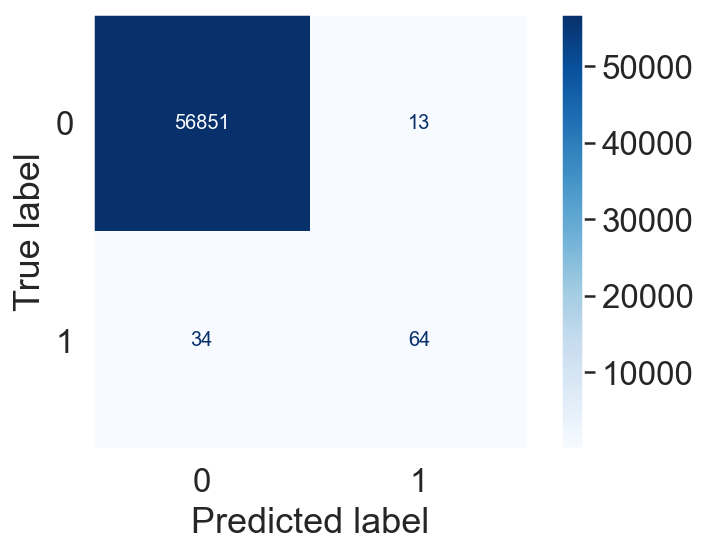
I will compare 5-fold cross-validated scores but will also inspect the confusion matrix on the test set (representing 20% of the data, and includes 98 fraud transactions).

1. No modification of the sample

a) Grid Search with ‘accuracy’ scoring

Best score = 0.999

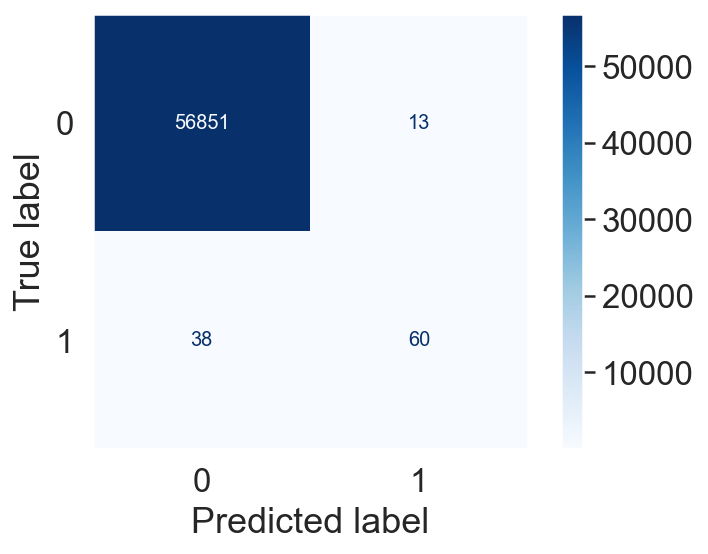
Confusion matrix on the test set



b) Grid Search with ‘precision’ scoring

Best score = 0.894

Confusion matrix on the test set



2 Undersampling

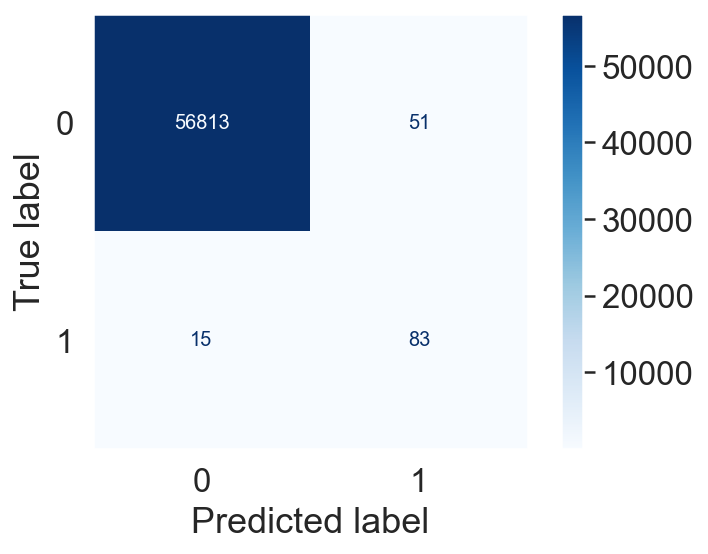
I used the RandomUnderSampler from imblearn with a sampling strategy=1. In this way, I reduced the majority class to the number of minority (fraud) class. This significantly reduced my train set, from 227’845 to 788.

a) the Grid Search with ‘accuracy’ scoring

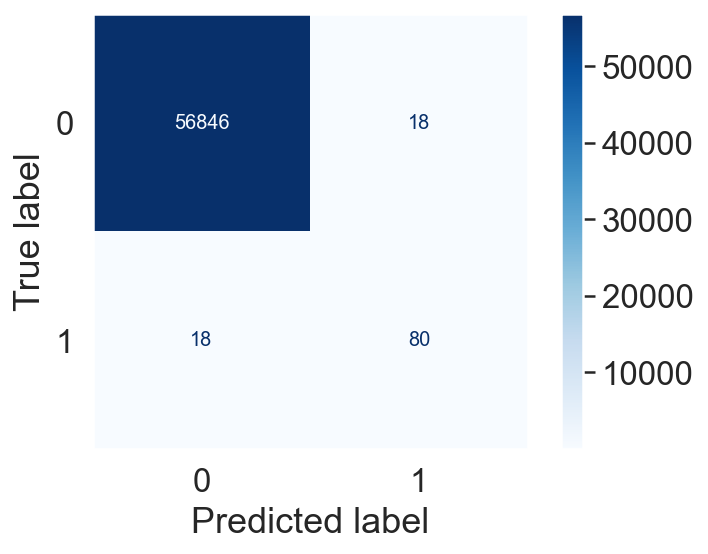
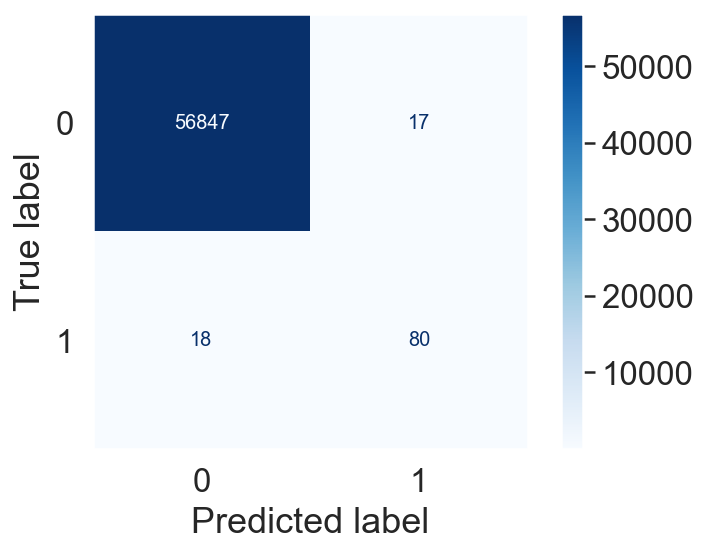
best score = 0.945

b) Grid Search with ‘precision’ scoring

best score = 1.0



I further explored and found another model with a different regularisation parameter (C), which performed better on this particular test set with the following confusion matrix:

3) Oversampling

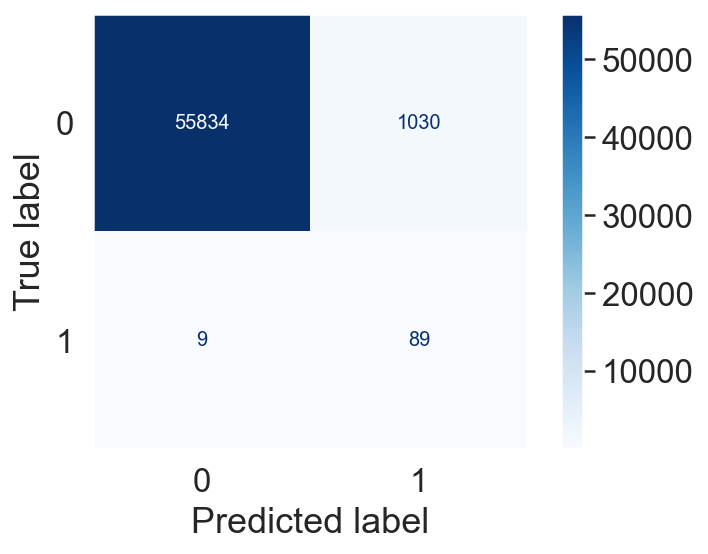
I used the RandomOverSampler from imblearn with a sampling strategy=1. In this way, I increased the minority (fraud) class to the number in the majority class. This significantly increased my train set, from 227’845 to 454’902.

a) the Grid Search with ‘accuracy’ scoring

best score = 0.95

b) Grid Search with ‘precision’ scoring

best score = 0.98



The number of false-positive (model predicted fraud, where in reality the transaction was normal) significantly increased, to 1030.

4) SMOTE

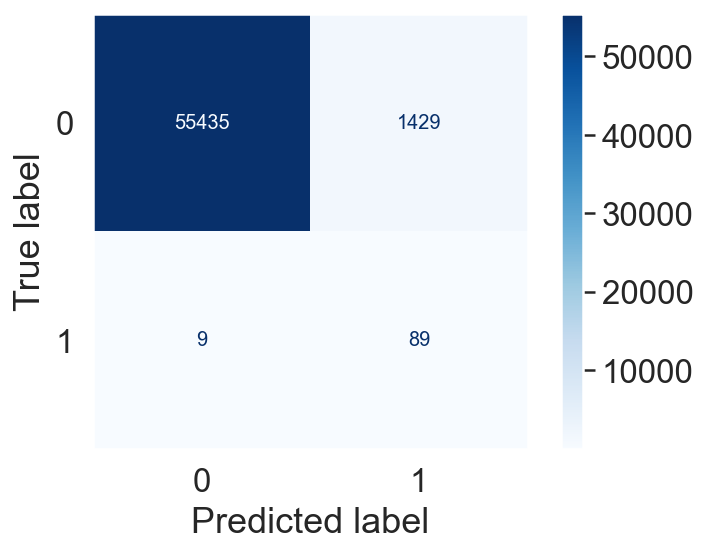
I also experimented with SMOTE, but the results are not worth mentioning here.

5) Custom sampling method

I tried to resample in two steps, first, oversample the minority class to get a representation of cc 10%, then to resample the majority class to match the number of the increased minority class samples. In this way, I got a total number of 25’272 transactions.

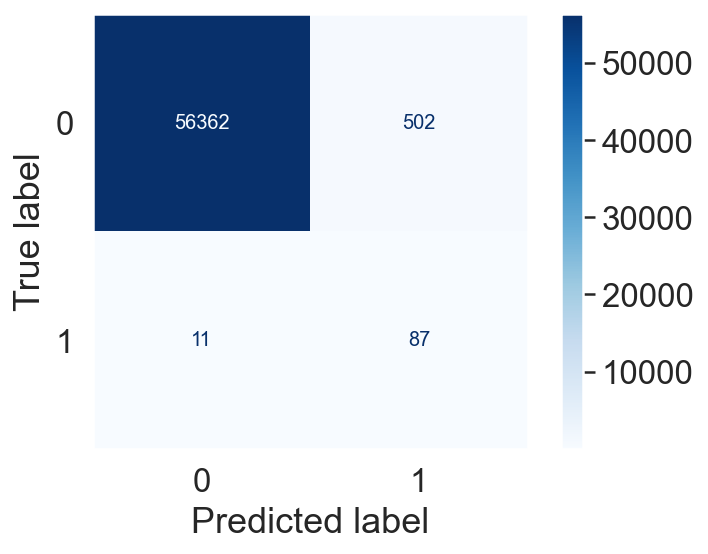
a) the Grid Search with ‘accuracy’ scoring

best score = 0.9496



b) Grid Search with ‘precision’ scoring

best score = 0.99



This custom sampling gave a result with nearly half of false positives and gave found similar true positives on the test set.