Credit Card Fraud Imbalanced Dataset: Review, Metrics, Resampling techniques, Data Exploration and Comparing Machine Learning Models

Background:

For the classification problem, a dataset is imbalanced when the classes are disproportionately represented. It is challenging to apply machine learning models because after model fitting, although the accuracy of the model is very high. This model fails to predict the minority class. Most popular imbalanced dataset examples are fraudulent telephone calls, credit card fraud prediction, detection of defects in different scenarios, oil-spill detection, network intrusion detection, rare diseases and cancer prediction. The different ways to deal with the imbalanced data set has been discussed in various literatures [journal articles, blogs] in terms of data-level techniques, algorithm-level methods, and hybrid approaches [1-10].

In this noted work, Hulse et al. [19] presented a comprehensive study of imbalanced data, using 11 learning algorithms with 35 real-world benchmark datasets from different domains. They applied different sampling methods and ML models to derive insights for providing practical guidance to machine learning practitioners working with the imbalanced data. There is no standard approach to handle imbalanced dataset. This study highlighted following conclusions. The different types of sampling work best with different algorithms. Along with the sampling techniques the performance measure affects the results. They observed that for different models, none of the sampling techniques significantly improved the performance when measured by popular metrics. However, when the performance is measured using the threshold-dependent measures, significant improvements have been noted.

Motivation and Project Description:

I found the imbalanced dataset problem interesting and challenging. That's the motivation behind selecting this topic for my first DS project. In real life, imbalanced dataset problem affects different businesses spanning from cybersecurity, financial institutions and industrial or medical diagnosis. The goal of this project is to understand everything about solving this challenging problem with different examples and learning different approaches and provide a guidance to ML newbie learners with all the information for starting the project related to imbalanced datasets. I have carried out a comprehensive review of different blogs and articles. The project is divided in two parts. First, I started with a popular credit fraud dataset to carry out data explorations, learn different metrics and techniques to handle the imbalanced dataset. I created a template notebook with a pipeline to be used in the next part. Second, I am looking for the unsolved dataset to apply learnings from the part 1.

Different Machine Learning Models with and without resampling.
Project Description - Credit card dataset presents transactions that occurred in two days, where there were 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

Imbalanced data handling techniques [6]:

This section summarizes the basic required to work on this project.
1) Collecting more data, if possible.
2) Choosing the right performance metrics – Depending on the business requirement.
3) Resembling the training set**:** Under-sampling – Random under sampling (RUS): TomkekLinks, Cluster Centroids**,** Over-sampling - Random over sampling (ROS), Synthetic minority over-sampling (SMOTE) and SMOTEtomek, adaptive synthetic sampling approach (ADASYN), Selective preprocessing of imbalanced data (SPIDER), Modified synthetic minority oversampling technique (MSMOTE)
5) Ensemble techniques presented in figure 1
6) Penalizing models

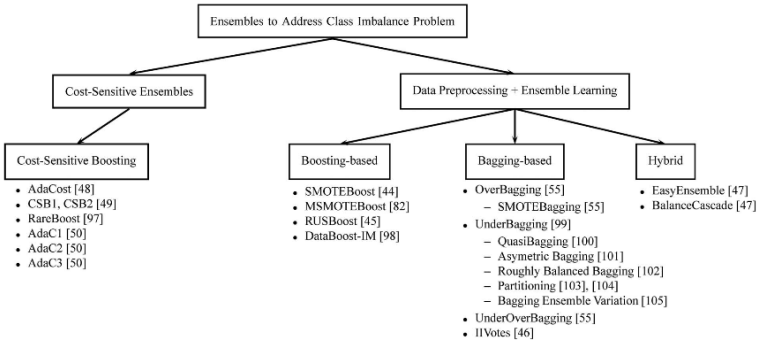


Figure 1 Ensample to address the class imbalance problem [26].

Performance Metrics: Evaluation of a classification algorithm performance with different metrics

Confusion Matrix - a table showing calculated correct predictions and types of incorrect predictions.

Table 1 Confusion Matrix.

		Predicted class	
		Positive Class 1	Negative Class 2
Actual Value	Positive Class 1	True Positive (TP) (Right)	False Negative (FN) (Wrong)
	Negative Class 2	False positive (FP) (Wrong)	True Negative (TN) (Right)

- **True Positives (TP):** Model **predicted** for Positive (Will Have) & they **actually** have the disease.
- **True Negatives (TN):** Model **predicted** for Negative (Will Not Have) & they **actually** don't have the disease.
- **False Positives (FP):** We **predicted** for Positive (Will Have) & they **actually** don't have the disease. (Also known as a "Type I error.")
- **False Negatives (FN):** We **tested** for Negative (Will Not Have) & they **actually** have the disease. (Also known as a "Type II error.")

Accuracy - The ratio of correctly predicted examples by the total examples. Overall, how often is the classifier correct? **- TP+TN/(TP+TN+FN+FP)**

Precision - "When it predicts the positive result, how often is it correct?" Accuracy of the positive predictions - TP / (TP + FP)

Recall - "When it is actually the positive result, how often does it predict correctly?" A ratio of positive instances that are correctly detected by the classifier - TP / (TP +FN)

F1 score- harmonic mean of precision and recall - 2 x (precision x recall) / (precision + recall)

ROC curve -more visual way to measure the performance of a binary classifier is the **receiver operating characteristic (ROC) curve**. It is created by plotting the true positive rate (TPR) (or recall) against the false positive rate (FPR),

True Positive Rate or Recall or Sensitivity: When it's actually yes, how often does it predict yes? TP/actual yes = TP/(TP + FN)

False Positive Rate: When it's actually no, how often does it predict yes? FP/actual no = FP/(FP + TN)

AUC: relation between true positive rate and false positive rate - **AUC** stands for *Area under the ROC Curve.* It provides an aggregate measure of performance across all possible classification thresholds.

Significance of the performance measures:

- The higher the **area under the ROC curve** (AUC), the better the classifier. A perfect classifier would have an AUC of 1. Usually, if your model behaves well, you obtain a good classifier by selecting the value of the threshold that gives TPR close to 1 while keeping FPR near 0.
- Prediction Error - Type I (False positive) and Type II (False Negative)
- For a given class, the different combinations of recall and precision have the following meanings:

    - High recalls + high precision: the class is perfectly handled by the model
    - Low recall + high precision: the model can't detect the class well, but is highly treatable when it does
    - High recalls + low precision: the class is well detected, but the model also includes points of other classes in it
    - Low recall + low precision: the class is poorly handled by the model

Different learning Models with and without resampling (SMOTE).

Model details:

- Input - 30 features with 28 PCA transformed features except 'Time' and 'Amount'.
- Output - Feature 'Class' and it takes the value 1 in case of fraud and 0 otherwise.
- Studied the performance metrics – ROC, precision, recall, F1-score and accuracy score and AUC
- The algorithms – Logistic Regression and Random Forest
- Applied Generate synthetic samples -SMOTE to both the algorithms.
- Implemented pipeline to apply different learning models, sampling techniques, grid search etc.

Model results: Summarized using confusion matrix, classification report and ROC curves.

Table 2. With and without SMOTE for the LR and RF model results summarized below using confusion matrix and classification report.

Type of model	Logistic Regression		Random Forest	
Metrics	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE
F-1	0.75598	0.003499	0.8545	0.00597
Recall	0.65833	1.0	0.783	0.7666
Precision	0.8876	0.001752	0.94	0.0030
Confusion Matrix	[71072 10] [41 79]	[2734 68348] [0 120]	[71076 6] [26 94]	[40517 30565] [28 92]
Accuracy	0.9543	0.9857	0.9728	0.9938

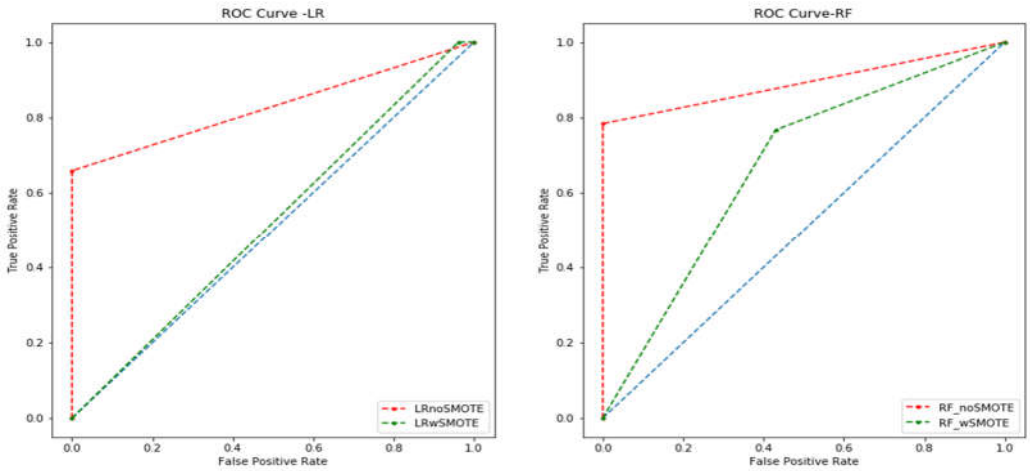
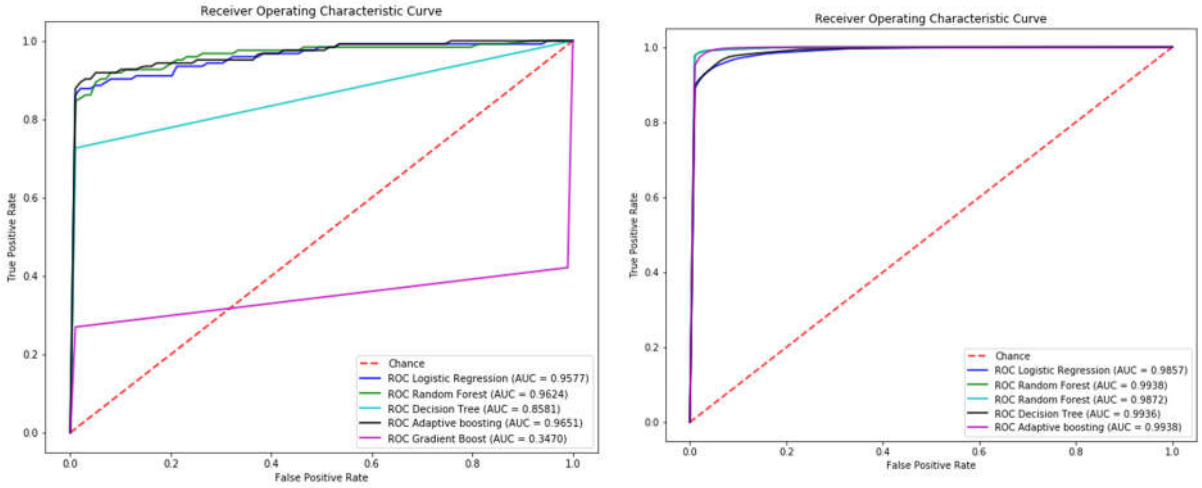


Figure 2 Comparison of the ROC curve for the LR and RF model without grid search.



(a) without SMOTE (b) With SMOTE

Figure 3 Comparison of the ROC curve for the different models after grid search.

Conclusions:

1) Train split random seed () options affects the results.
2) class_weight= option for the classifier works like resampling.
3) Feature scaling was applied because Time and the amount needed scaling, however, V1-V27 features were within a scale. It also affects results.
4) Grid search helped in obtaining best parameters.
5) Applying learning algorithms and/or resampling techniques showed that depending on the metrics different results are obtained. Defining metrics depending on the business problem to be solved.
6) Credit cards are main source of transactions for any bank customers. Thus, if our fraud detection model blocks non frudelent transaction the customer complaints and customer dissatisfaction will increase. Thus, the metric used for fraud detection should be accuracy of classifying non-fraud cases then the recall score. The cost of mis-classifying actual positive is very high.
7) Random Forest model performed better in both cases with and without applying Synthetic Minority Oversampling Technique (SMOTE).
8) Gradient boosting algorithm performed better only with the SMOTE sampling technique.
9) All advanced models' performance is comparable when applied re-sampling technique with a AUC score 0.99.

References

1. https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
2. https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28
3. https://towardsdatascience.com/dealing-with-imbalanced-classes-in-machine-learning-d43d6fa19d2
4. https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/
5. https://medium.com/james-blogs/handling-imbalanced-data-in-classification-problems-7de598c1059f
6. https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18
7. https://medium.com/@csmoon/6-approaches-to-deal-with-imbalanced-classes-9133f2906e86
8. https://medium.com/@patiladitya81295/dealing-with-imbalance-data-1bacc7d68dff
9. https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb
10. https://medium.com/james-blogs/handling-imbalanced-data-in-classification-problems-7de598c1059f
11. https://medium.com/coinmonks/smote-and-adasyn-handling-imbalanced-data-set-34f5223e167
12. https://www.datacamp.com/community/tutorials/diving-deep-imbalanced-data
13. https://link.springer.com/content/pdf/10.1186%2Fs40537-019-0192-5.pdf
14. https://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-001.pdf
15. https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html
16. https://eprint.iacr.org/2018/476.pdf
17. https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/
18. https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/
19. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.4356&rep=rep1&type=pdf
20. https://imbalancedlearn.readthedocs.io/en/stable/auto_examples/pipeline/plot_pipeline_classification.html#sphx-glr-auto-examples-pipeline-plot-pipeline-classification-py
21. https://link.springer.com/article/10.1186/s40537-018-0151-6
22. https://www.kaggle.com/data/46744
23. https://towardsdatascience.com/working-with-highly-imbalanced-datasets-in-machine-learning-projects-c70c5f2a7b16
24. https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets
25. https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets
26. https://sci2s.ugr.es/keel/pdf/algorithm/articulo/2011-IEEE%20TSMC%20partC-%20GalarFdezBarrenecheaBustinceHerrera.pdf
27. https://sci2s.ugr.es/keel/imbalanced.php
28. https://www.kaggle.com/lct14558/imbalanced-data-why-you-should-not-use-roc-curve