Credit Card Fraud Detection with Logistic Regression and Support Vector Machine Classifiers

Shadi Bavar [*1] **Matthew Euliano** [*1] **Claire Parisi** [*1]

Abstract

Credit card fraud is a persistent problem that costs consumers billions annually. To address this issue, we develop fraud classification systems using supervised machine learning techniques. Using a dataset of real transaction information, we design logistic regression and support vector machine (SVM) classifiers to predict whether a transaction is fraudulent or not. We find that both approaches suitably classify fraudulent transactions with a high hit-rate.

1. Introduction

With the ubiquity of credit cards as a form of payment and a growing movement towards cashless payment options, protecting credit card accounts from counterfeit charges is now more important than ever. In 2019, fraudulent transactions in European countries amassed to roughly €1.87 billion (European Central Bank, 2021). This rate has only increased year-after-year, so it is imperative to identify and remediate instances of fraud.

Using a publicly available data set of credit card fraud reports from September 2013 by European cardholders, we devise both logistic regression and a support vector machine (SVM) classifier models to approach the binary classification problem of detecting whether a transaction is fraudulent or not. We use linear regression as our baseline model since it is computationally simple and easy to implement. For a more sophisticated method, we chose to use a SVM model. SVMs always converge, are generally fast, have fewer hyperparameters, and do not require a very large data set to learn the decision boundary. While other methods, such as neural networks (NN), tend to have best performance at the limits of computation power (i.e. if there are enough time and resources available), can be arbitrarily constrained in size and dimensionality by the selection of layer number and size, and are fast at classifying new data after learning, they also have many hyperparameters that require searching, are sensitive to initialization and the input data, and can fail to learn the decision boundary if these parameters are not set up properly. SVMs can also perform better than NNs against edge cases. The downside to SVMs are that the computational effort required to train the model scales quadratically with the size of the data set, they are slower than NNs to predict on unseen data and high dimensional problem solutions can generalize poorly (Jeeva, 2018; Luca, 2022a;b). With that, an SVM is a powerful, well-suited approach to this problem. The ultimate goal of these models is to present a solution that can classify instances of fraud with a high hit-rate/recall. In the context of this problem, there is less risk to classifying a normal transaction as fraudulent (false alarm) than there is to predicting a fraudulent transaction as normal (missed detection), so hit-rate is the primary design objective. In this paper, we demonstrate well-performing models for both logistic regression and SVMs. We find that both SVM and logistic regression perform similarly on this dataset.

The remainder of this report is organized as follows: Section 2 introduces the data set, Section 3 describes the data cleaning process, Section 4 talks about the formation of the linear regression models, Section 5 discusses the design of the SVM models, Section 6 reports the results of each model on the data set, and Section 7 concludes the paper.

2. Credit Card Fraud Detection Dataset

The dataset consists of credit card transaction information from Europen cardholders from 2013 and was sourced from Kaggle (Machine Learning Group of Université Libre de Bruxelles). It contains 284,807 labeled samples and has 31 total features. Due to user privacy, the only named features are the time of the transaction and the value. This data set has also already been dimensionality-reduced and is the result of principle component analysis on the original data. One very important characteristic of the data is the skewness or imbalance between the class representation. That is, the majority of the data consists of normal transactions

[*] Equal contribution [1] Department of Electrical and Computer Engineering, Northeastern University, Boston, USA. Correspondence to: Shadi Bavar <bavar.s@northeastern.edu>, Matthew Eulino <euliano.m@northeastern.edu>, Claire Parisi <parisi.cl@northeastern.edu>.

EECE5644: Introduction to Machine Learning and Pattern Recognition, Northeastern University, Summer 1 Semester, 2022.

(284,315 samples or 99.83%), while only .17% percent or 492 transactions are examples of fraud. While we do not

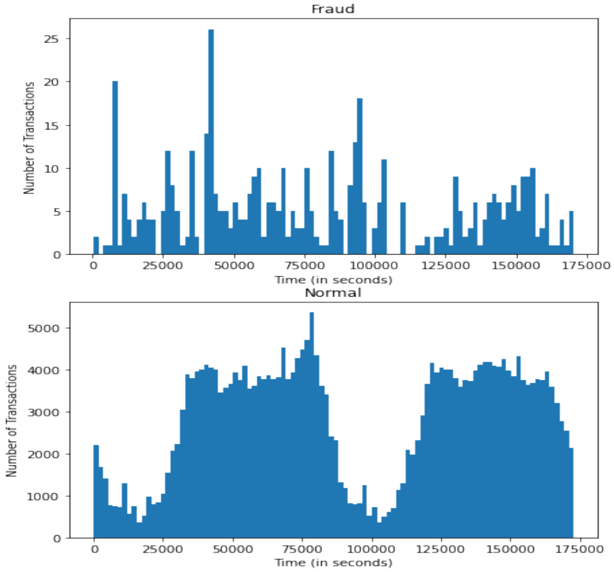

Figure 1. Histogram of fraudulent and normal transactions over time.

know what the majority of the features represent, we can plot the features of time and amount to get a better sense of what the dataset looks like. Figure 1 shows the transactions over time. Looking at the normal transactions, it is clear that there are trends associated with transaction along with time of day; likely, the times with lower transactions are during the night hours. Meanwhile, the fraudulent transactions have less of a time trend.

Plotting the data as a function of amount (Figure 2) we see that all fraudulent transactions are less than €2500, with the majority below €200.

Finally, the plot of the first two features of the data (Figure 3) and their corresponding classes provides a general indication that the problem should be solvable, since the cases of fraud are clustered together. We assume that these features correspond to the first two principal components of the raw data due to their labelling, but this is not explicitly stated in the dataset documentation.

3. Data Pre-processing

Before working with the dataset, we must regularize and condition the data for use. While the unlabeled features are already scaled (since they are a result of PCA), we regularize the time and amount features to match the rest of the dataset. From here, we split our dataset into a training and testing set. When splitting the data into test and training sets, due to the skewness, we performed stratified sampling. This simply ensures that a representative ratio of fraud and not-fraud examples would be contained in both test and training sets. Otherwise, random sampling could result in extremely few or even no cases of fraud in the test set. From the original dataset, 80% of the data is used in the training set and the remaining 20% acts as our testing set.

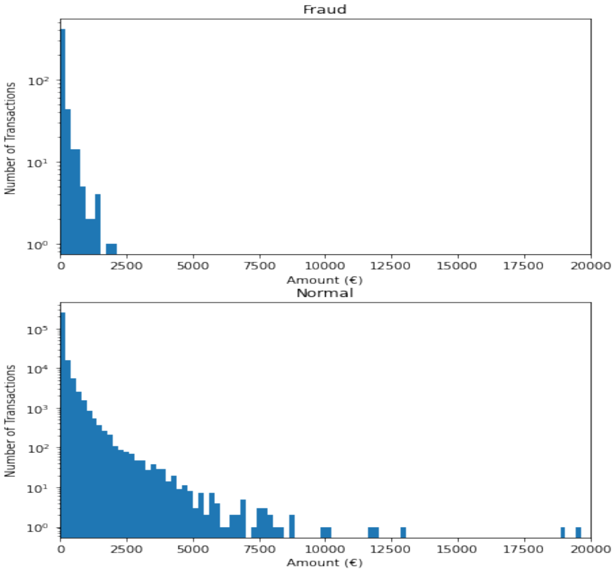

Figure 2. Histogram of fraudulent and normal transaction as a function of the transaction amount. Bin width is approximately €200.

There are a variety of ways to handle imbalanced data which can improve model performance. The two simplest methods are undersampling and oversampling. When undersampling, a random set of samples are chosen from the majority class, equal in number to the size of the less-represented class. The result is a subset of the data set that contains a balanced ratio of both classes. Training on undersampled data has the advantage of running faster and avoiding overfitting, However, there is lossy approach since some of the data is disregarded. When oversampling, samples in the minority class are duplicated until there is an equal balance of samples from each class. The trained model often performs better than undersampling, however, oversampling is prone to overfitting (Kumar, 2020).We decided to handle the skewed fraud detection data with undersampling because of the significant increase in computation speed as well as avoiding overfitting.

4. Logistic Regression Model

Since the credit card fraud detection problem is a binary classification problem, we can apply binary logistic regression to form a baseline model for identifying fraudulent transactions within the dataset of credit card information. The binary logistic regression classification model,

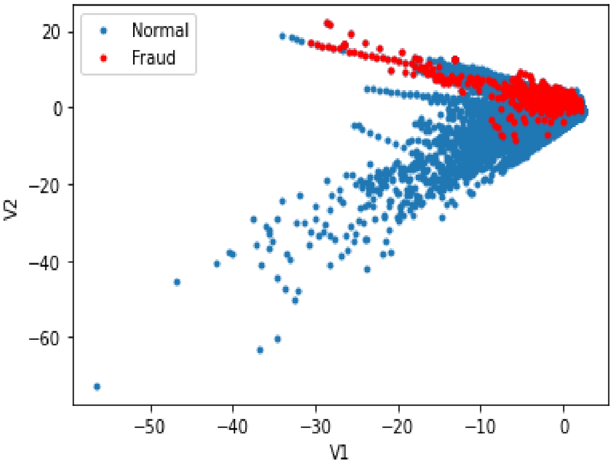

Figure 3. Plot of first two features of the data.

$p(y|\mathbf{x};\theta)$, relates an input, $\mathbf{x} \in \mathcal{R}^D$ of D dimension with $y \in \{0,1\}$ class labels, where θ are the parameters. The model $p(y|\mathbf{x};\theta)$ =Ber$(y|\sigma(\mathbf{w}^T\mathbf{x}+b))$, where $\mathbf{w}$ are the weights and b is the bias (so $\theta = (\mathbf{w}, b)$), and σ is the sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$ (Murphy, 2022). In our case, $\mathbf{x}$ is the credit card data features, and y is the associated label of fraudulent or not fraudulent. The predicted label, $\hat{y} = \sigma(\theta^T\mathbf{x})$, where $\mathbf{x}$ is the augmented input vector to handle the bias term.

Using the logistic regression function from scikit learn,we can easily construct a logisitic regression model for our dataset. The function has a hyper-parameter, C, which is the inverse of regularization strength (Pedregosa et al., 2011). To select the optimal C value, we use 5-fold cross-validation on our training set. The logistic regression model generated using the scikit learn function is optimized using the Limited-memory-Broyden-Fletcher-Goldfarb-Shanno (lmbfgs) solver, which essentially approximates the gradient and updates in a memory-efficient way. After generating our logistic regression model using the training dataset, we can make predictions on the test set and evaluate the relevant performance metrics.

In our case, we generate two logistic regression models for comparison. The first is a basic model using the stratified (imbalanced) training and testing sets. The second model uses the under-sampled, balanced training set to formulate the model. Later, we will see that the pre-processing of our dataset improves the recall metric of the logistic regression model on the test set.

5. Support Vector Machines Model

Support Vector Machines, or SVM, are machine learning models capable of performing linear or nonlinear classification, regression, and outlier detection. For linear classification, SVM splits data by maximizing the margin between classes. This results in outliers not drawing the decision boundary away, so it is fully determined (or ”supported”) by the instances located closest to the boundary. These instances are called the *support vectors*.

For nonlinear datasets, the goal is to find a balance between maximizing the margin and minimizing the margin violations (misclassifications). This is controlled using the C hyperparameter, which adds a penalty to each misclassified point. Thus, decreasing C results in wider margins but more misclassifications.

For imbalanced datasets, the margin favors the majority class because SVM minimizes the total number of misclassifications. This is problematic for detecting the minority class, such as fraudulent transactions in this case. To circumvent this problem, the SVM model can be altered by taking the importance of each class into account. The simplest modification to SVM for imbalanced classification is setting the weight of the C hyperparameter in proportion to the importance of each class. This is known as Penalized/Weighted/Cost-Sensitive SVM.

For classification via the standard SVM methodology, the penalty weight assigned to misclassifications are the same for every datapoint. Introducing variable weights into the loss function can help to pull the decision boundary away from the fraudulent cases at the expense of a few non-fraudulent misclassifications.

The SVM models were trained on the same undersampled datasets as in logistic regression. The choice of SVM kernel is one of the most important hyperparameter selections when designing an SVM algorithm, each kernel having its own set of parameters to tune. We tested three of the most commonly used kernels: the Gaussian Radial Basis Function (RBF), Linear, and Polynomial kernels.

To find the best combination of hyperparameters, a grid search approach was used on each kernel functions with the following parameters being swept across a range of plausible values: for RBF these are γ, C, class weighting; for linear these are C, class weighting; for polynomial these are γ, degree, C, class weighting.

Where parameter γ tunes the radius of influence of each training sample, such that low values of gamma result in more points being grouped together (Pedregosa et al., 2011).

6. Results

In the case of a heavily skewed classification problem, accuracy and mean squared error are poor performance metrics because the model could simply predict that all unseen data

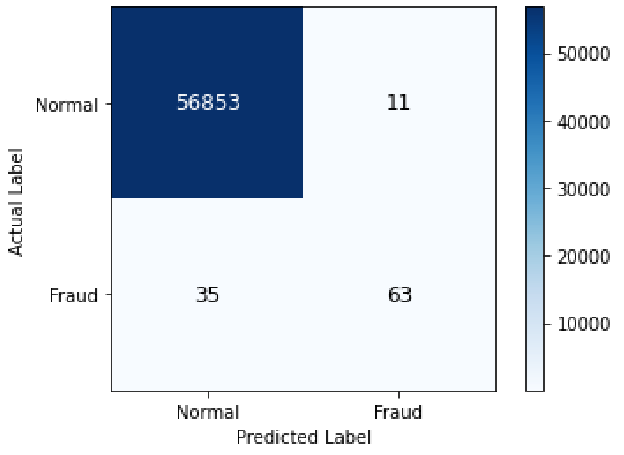

Figure 4. Confusion matrix for baseline logistic regression model.

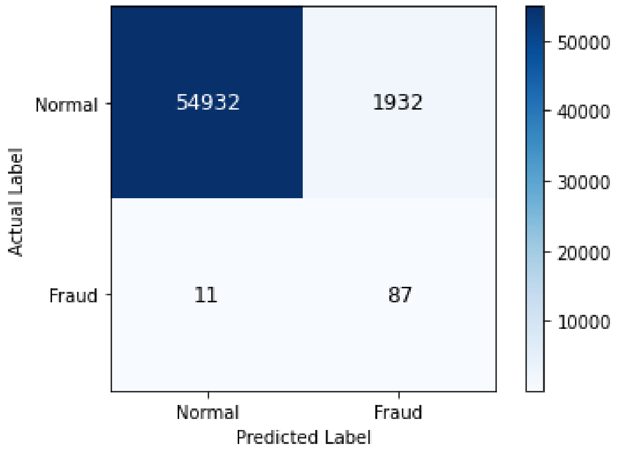

Figure 5. Confusion matrix for logistic regression model trained with the under-sampled, balanced dataset.

belongs to the majority class and would achieve high accuracy. However, this does not identify the few but nontrivial cases in which the samples were, for example, fraudulent. For this reason, the performance metric of most importance to this problem is recall. Recall/ ”hit-rate” describes the fraction of true fraud cases that are correctly identified out of all of the true instances of fraud. In this section, we evaluate each models performance considering recall, but also other metrics such as precision (ratio of correctly detected frauds out of all of the predicted frauds), and F1 score (a harmonic mean of precision and recall, which helps assess the balance between the two metrics). We can also consider the probability of false alarms (pfa) and missed detection (pmd) to help understand further the model trade-offs.

6.1. Logistic Regression

As a baseline, we train a logistic regression model on the training set which is distributed with the same fraud/non-fraud ratio as the original dataset. In this realization, there are 98 total cases of fraud and 56864 cases of normal transactions. The model classifies the dataset according to the confusion matrix in Figure 4. With this classifier model, we observe a recall of .64, precision of .85. With this model, our probability of false alarm is very low at the expense of having a probability of missed detection of .36.

Using the under-sampled training set (that balances the number of fraudulent and non-fraudulent classes), we can train a new model and observe how it differs from the baseline logistic regression model with no data pre-processing. For the model with the conditioned data, we observe the results summarized in the confusion matrix in Figure 5.

With this classifier mode, we observe a recall of $\approx$ .89, precision of .04. This model has a higher probability of false alarm than the baseline, with the benefit of reducing the probability of missed detection threefold (.11). As fraud detection classifier, this operates much better than the baseline model because the hit rate is much higher and less examples of true fraud are missed. Looking at the ROC curve in Figure 6, we can see the performance between the two models. Essentially, the difference between the two models comes down to a decision threshold. Tuning that threshold yields different values for false alarm and true positive rates. The ”improved” model where the data is trained on the balanced training set, yields better results for detecting frauds (better hit rate). The results are summarized in Table 1.

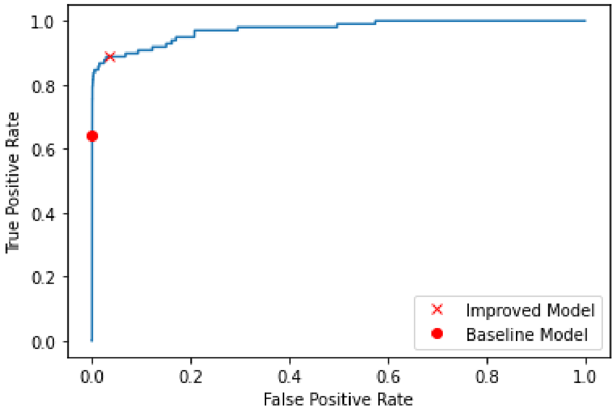

Figure 6. ROC curve for logistic regression models (area under the curve is .97).

While the baseline model provides an overall more balanced model in terms of recall and precision, as shown by its F1 score, the under-sampled model is a much superior fraud detection classifier, since we want the fewest number of missed detections as possible.

Table 1. Logistic Regression Performance Metrics

MODEL	RECALL	PRECISION	F1	PFA	PMD
BASELINE	0.643	0.851	0.733	0.0002	0.357
UNDER-SAMPLED	0.888	0.043	0.082	0.034	0.112

6.2. SVM

The results of SVM hyperparameter selection via grid search are shown in Table 2.

Table 2. SVM Hyperparameter selection results

KERNEL	C	WEIGHT	γ	DEG
LINEAR	1000	1.25	-	-
RBF	10	1.75	0.001	-
POLY	100	1.75	0.01	3

From the confusion matrix in Figure 7 and ROC plotted in Figure 8, there is very little variability in the performance of the models generated using any of the SVM kernels.

This intuits that the preprocessed data and PCA created a dataset that is close to linearly separable whose boundary can be found equally well by several ML algorithms. For the testing set, there is a consistent number of fraud misclassifications (≈ 11) that can be assumed are outliers. The performance metrics for each kernel are summarized in Table 3, which highlights the similarity of all three options.

Overall, for both the logistic regression models and SVM models we see the trade-off of precision and recall; models with better recall tend to have worse precision. As mentioned, for fraud detection, it is preferable to have a higher hit rate at the expense of more false alarms - within reason. The particular example of logistic regression reported above even had slightly fewer false alarms than achieved with SVM, but this can be attributed to variability in training runs. None of the models were able to correctly identify 11 of the cases of fraud, so these would seem to be outliers. We hypothesize that the similarity across models is due to the problem having a linear solution, potentially having been affected by the pre-processing and dimensionality reduction that was performed on the dataset. Both techniques present a reasonable model for fraud detection.

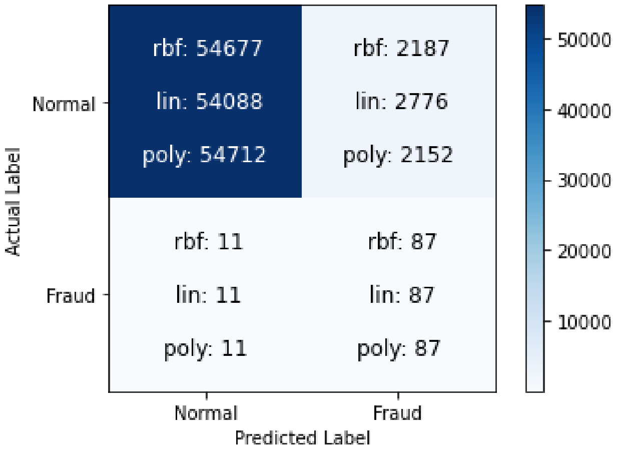

Figure 7. Confusion matrix for SVM kernels.

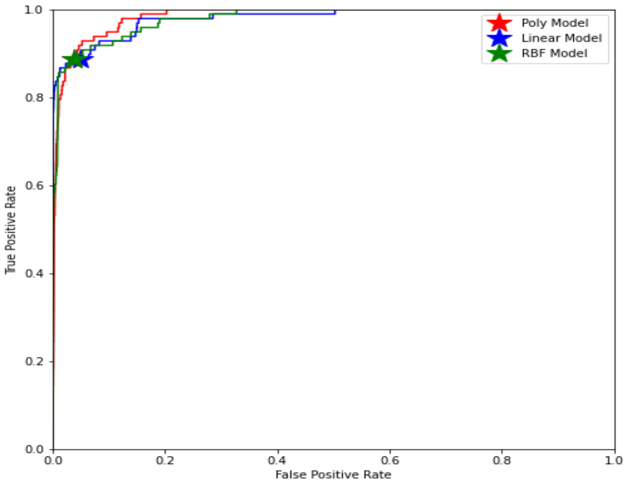

Figure 8. ROC for SVM kernels. Area under the curve is ≈0.98 for all models

Table 3. SVM Performance Metrics

KERNEL	RECALL	PRECISION	F1	PFA	PMD
LINEAR	0.888	0.031	0.058	0.049	0.112
RBF	0.888	0.038	0.073	0.038	0.112
POLY	0.888	0.038	0.074	0.038	0.112

7. Conclusion

In this paper, we present logistic regression and SVM classifier models to detect fraudulent charges on a dataset of credit card transactions. The models presented all perform similarly with approximately 89% recall. In this particular case, the logistic regression model would likely be the preferable option since it is overall simpler and has almost identical performance to the SVM. However, on a different credit card transaction dataset, potentially the SVM may be better suited and perform better since it is a more complex tool for classification.

Appendix

The code for this project can be found at: https://github.com/meuliano/ML_Credit_Card_Fraud_Detection

References

https://www.kaggle.com/code/joparga3/in-depth-skewed-data-classif-93-recall-acc-now. Accessed: 2022-20-06.

European Central Bank. Seventh report on card fraud. https://www.ecb.europa.eu/pub/pdf/cardfraud/ecb.cardfraudreport202110~cac4c418e8.en.pdf, October 2021.

Jeeva, M. The scuffle between two algorithms - neural network vs. support vector machine. https://medium.com/analytics-vidhya/the-scuffle-between-two-algorithms-neural-network-vs-support-vector-machine-16abe0eb4181, September 2018.

Kumar, B. 10 techniques to deal with imbalanced classes in machine learning. https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/, July 2020.

Luca, G. D. Svm vs neural network. https://www.baeldung.com/cs/svm-vs-neural-network, June 2022a.

Luca, G. D. Advantages and disadvantages of neural networks against svms. https://www.baeldung.com/cs/ml-ann-vs-svm, June 2022b.

Machine Learning Group of Université Libre de Bruxelles. Credit card fraud detection. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud. Accessed: 2022-20-06.

Murphy, K. P. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022. URL probml.ai.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.