Analysis of Classification and Regression Models

Adelle Price

University of Colorado Denver

Statistical and Machine Learning, MATH 6388

Overview

Here, I implement and assess various machine-learning models on two separate datasets. The first machine-learning model implementation and assessment involves detecting credit card fraud using simple logistic regression models. I will compare four simple logistic regression models using four metrics: precision, recall, F1 score, and accuracy. The second machine-learning model implementation and assessment involves predicting the heating and cooling load requirements of a building. I will compare 8 different regression models- for the prediction of both the heating and cooling load requirements- and I will evaluate model performance with the metrics mean squared error (MSE) and the coefficient of determination, r^2.

Credit Card Fraud Detection

Data

The data analyzed in the classification portion of the project is a dataset containing credit card transactions by European cardholders. The data was collected in September 2013 over the course of two days. The data contains 28 principal components resulting from PCA of the original credit card transaction data, time (the seconds elapsing between each transaction and the first transaction in the dataset), amount (transaction amount in dollars), and class (equals 1 to indicate fraudulent and equals 0 to indicated not fraudulent). There are 284,807 total transactions; only 492 transactions are truly fraudulent transactions.[1]

Methods

Synthetic Minority Oversampling Technique (SMOTE)

To address the imbalanced ratio of fraudulent to non-fraudulent transactions in the credit card transactions data set, SMOTE was used to synthesize new minority samples for the fraudulent class within the credit card data. SMOTE was implemented using the function *SMOTE* in Python's *imbalanced-learn* library.[2] The steps of the SMOTE Algorithm are shown below[3]:

1. Label the minority class set as A, where each $x \in A$, and calculate the Euclidean distance between each x and every other sample in set A.
2. The sampling rate N is set according to the imbalanced proportion. For each $x \in A$, N samples from are randomly selected from the k-nearest neighbors of x_k and they are placed in the set A_1.
3. For each $x_k \in A_1$, the following formula generates a new sample:

$$x' = x + rand(0,1) * |x - x_k|$$

Logistic Regression

Four binary logistic regression (BLR) models were built where each model outcome classification of 1 indicated a fraudulent transaction and the model outcome classification of 0 indicated a non-fraudulent transaction. The four BLR models are detailed in **Table 1**. Each model was built with a training dataset that account for 80% of the total data. Logistic regression models were built and assessed using Python's *Scikit-learn* library.[4]

Table 1. Binary Logistic Regression Models used to classify transactions as Fraudulent/Non-Fraudulent.

	Predictors Included in Model			
	28 PCs	**Dollar Amount of transaction**	**Time between transactions**	**Data Type (Raw/ After SMOTE Imputation)**
BLR Model 1	X	X	X	Raw Data
BLR Model 2	X	X	X	After Smote Imputation
BLR Model 3	X	X		Raw Data
BLR Model 4	X	X		After Smote Imputation

Each BLR model was assessed with the following metrics: Mean Precision, Mean Recall, Mean F1 score, and Mean Accuracy. Mathematical details about each metric are shown below.

$$Precision = \frac{True\ Positive\ Count}{True\ Positive\ Count + False\ Positive\ Count}$$

$$Recall = \frac{True\ Positive\ Count}{True\ Positive\ Count + False\ Negative\ Count}$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{True\ Positive\ Count + True\ Negative\ Count}{True\ Positive\ Count + False\ Positive\ Count + True\ Negative\ Count + False\ Negative\ Count}$$

The mean value of each metric was calculated from 100 resamples (with replacement) of the original data set (original data set had 287,807 observations for BLR model 1 and 3; original data set had 568,630 observations for BLR model 2 and 4). The size of each resampled test data set accounted for 20% of the size of the original data set.

Data Exploration

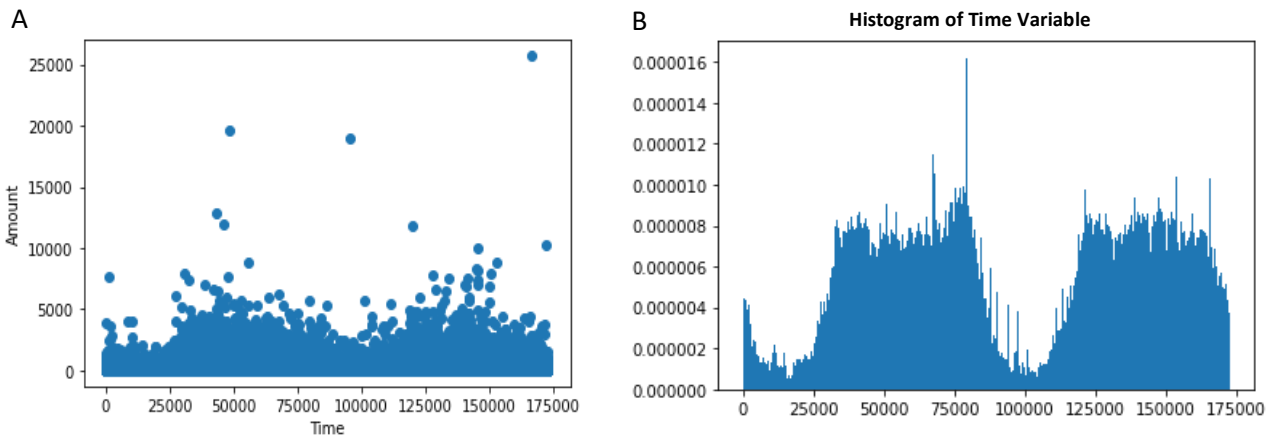


Figure 1. Exploration of Credit Card Fraud Data Amount & Time variables.
A) Time between transactions ("Time") versus dollar amount ("Amount") of each transaction.
B) Histogram showing the distribution of the time between transactions ("Time") variable.

In **Figure 1A** , there is no clear relationship between the Time and Amount variables. Likewise, in **Figure 1B** there are no apparent issues with the distribution of the Time variable; it appears to be continuous and bimodal in distribution. As there is no indication in the documentation for the credit card fraud data regarding if Time is a reliable predictor of the classification of a particular transaction as Fraudulent/Not Fraudulent, this motivates the inclusion of the Time variable in at least one binary logistic regression model assessed here in order to evaluate its contribution to the classification problem.

Results

Data Imputation Using SMOTE

In the raw credit card data (Before applying the SMOTE algorithm), the count of fraudulent transactions accounted for .172% of all transactions. After application of the SMOTE algorithm to the raw credit card data, the count of fraudulent transactions accounted for 50% of the data (**Table 2**). The raw data was used to create BLR Model 1 and 3 while the data after application of the SMOTE algorithm was used to create BLR Model 2 and 4.

Table 2. Counts of Fraudulent Transactions vs. Non-Fraudulent transactions before and after SMOTE.

	Raw Data	After SMOTE
Count of Fraudulent Transactions	492	284315
Count of Non-Fraudulent Transactions	284315	284315

Logistic Regression

BLR Models 1 and 3 (before imputing data with SMOTE) showed the poorest performance here; particularly in the Mean Recall metric (below 65% in for each model). BLR Models 2 and 4 perform well here with Mean Precision, Mean Recall, Mean F score, and Mean Accuracy scores ≥95% for each model. BLR Model 2 (includes Time between transactions as a predictor of the Fraudulent/Not Fraudulent class) outperformed BLR Model 4 (did not include Time between transactions as a predictor of the Fraudulent/Not Fraudulent class) (**Figure 2**).

A

	BLR Model 1	BLR Model 2
Mean Precision	**0.85**	**0.98**
Mean Recall	**0.64**	**0.96**
Mean F1 Score	**0.73**	**0.97**
Mean Accuracy	**0.999**	**0.97**

B

	BLR Model 3	BLR Model 4
Mean Precision	**0.88**	**0.98**
Mean Recall	**0.63**	**0.94**
Mean F1 Score	**0.74**	**0.95**
Mean Accuracy	**0.999**	**0.95**

Figure 2. Performance Metrics of BLR Models.

A) Mean performance (precision, recall, F1 score, and accuracy) of BLR Model 1 and BLR Model 2.

B) Mean performance (precision, recall, F1 score, and accuracy) of BLR Model 3 and BLR Model 4.

Discussion

Among BLR models 1, 2, 3, and 4; BLR model 2 showed the best performance. BLR model 2 was built from SMOTE imputed data and included the variable Time between transactions as a predictor of the class variable (Fraudulent/Not Fraudulent). This indicates that to detect credit card fraud in the given credit card dataset, imputation with SMOTE should be performed to address the class imbalance problem. Additionally, the Time between transactions variable should be included in the classification model to optimize the model classification of Fraudulent/Not Fraudulent transactions.

There are several limitations of the work shown here. First, the credit card data used to build all four models was collected over a time period of two days. It would be prudent to assess the performance of BLR model 1, 2, 3, and 4 on new credit card data to ensure the metrics reported here for each model hold across a range of data. Second, the credit card data was collected in 2013 and may not reflect current trends in credit card fraud. Lastly, the credit card data was sourced from European card holders and the credit card fraud trends seen in Europe may reflect only those expected in a particular region and may not apply to those seen in other regions or continents.

Heating and Cooling Load Requirement Prediction

Data

The data used for the regression section of the project contains relates to heating and cooling load requirements of buildings. The data contains 8 quantitative predictors: relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution. There are two output variables we are interested in predicting here: building heating load and building cooling load. The data contains 768 observations in total.[5]

Methods

Principal Component Analysis (PCA)

Due to high correlation between several predictors seen in the data (**Figure 3**), PCA was applied to data (after normalization of each column of the raw building energy requirement data), using the *PCA* function in Python's *Scikit-learn* library.[4] The first 5 PCs were chosen to approximate the real data and ensure the predictors in the Linear and Polynomial Regression models were uncorrelated.

Linear and Polynomial Regression

Seven regression models (heating load regression model number indicated by 'HM model#'), with polynomial degrees 1-7, were built with the goal of predicting the heating load requirements of a building (**Table 3**). Seven separate regression models (cooling load regression model number indicated by 'CM model#'), with polynomial degrees 1-7, were built with the goal of predicting the cooling load requirements of a building (**Table 4**). All linear models were built and assessed with Python's *Scikit-learn* library.[4]

Table 3. Regression models built and evaluated for prediction of heating load requirements.

	HM 1	HM 2	HM 3	HM 4	HM 5	HM 6	HM 7
Regression Model Degree	**degree = 1**	**degree = 2**	**degree = 3**	**degree = 4**	**degree = 5**	**degree = 6**	**degree = 7**
Predictors	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**

Table 4. Regression models built and evaluated for prediction of cooling load requirements.

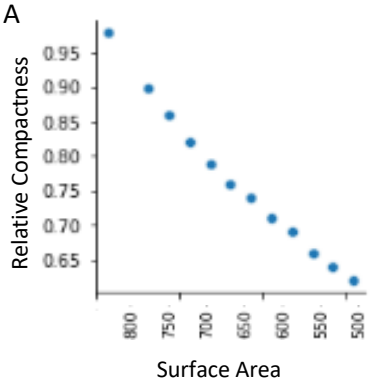
	CM 1	CM 2	CM 3	CM 4	CM 5	CM 6	CM 7
Regression Model Degree	**degree = 1**	**degree = 2**	**degree = 3**	**degree = 4**	**degree = 5**	**degree = 6**	**degree = 7**
Predictors	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**	**PC1, PC2,PC3, PC4,PC5**

The performance of each regression model was assessed with the metrics of Mean Squared Error (MSE) and the coefficient of determination, r^2. To obtain metrics of each model's prediction performance over various datasets, each model was tested on a resampled dataset (with replacement) that accounted for 100% of the size of the original building energy requirement dataset. The mean MSE and mean r^2 of each regression model was calculated over the resampled datasets.

Data Exploration

Several of the variables in the building energy requirement dataset appear to have high correlation (one example of highly correlated variables is shown in **Figure 3A**). Additionally, though the values of all variables in the building energy requirement data are reported as continuous numerical values, the distribution of some of the continuous variables appear to resemble that of categorical data in which there are identical counts within each continuous variable category (**Figure 3B**). The outcomes Heating Load and Cooling Load also appear to be highly correlated; indicating the regression models that have optimal performance when predicting either heating load or cooling load will likely be similar.

A



B

Surface Area Values	Surface Area Value Counts
514.5	64
563.5	64
588.0	64
612.5	64
637.0	64
661.5	64
686.0	64
710.5	64
735.0	64
759.5	64
784.0	64
808.5	64

C

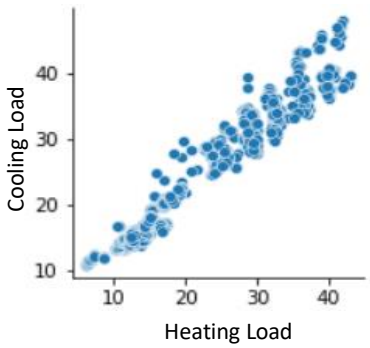


Figure 3. Data Exploration of Building Energy Requirements Data.
A) Scatterplot of Surface Area versus Relative Compactness in the building energy requirement data.
B) Counts of each value for Surface Area in the building energy requirement data.
C) Scatterplot of Heating Load and Cooling Load outcomes in the building energy requirement data.

Results

PCA

After performing PCA on the normalized building energy requirement data, PC 1, PC 2, PC 3, PC 4, and PC 5 were observed to sufficiently explain the variance seen in the normalized building energy requirement dataset and were chosen as the predictors for each regression model (**Figure 4**).

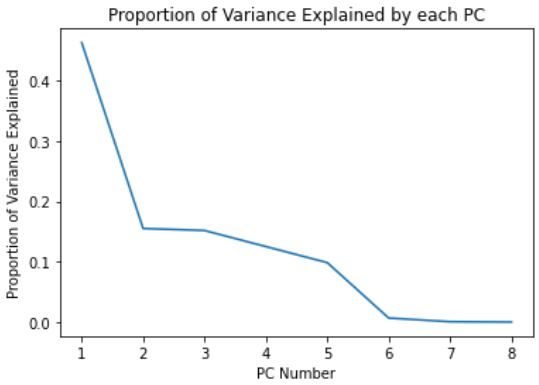


Figure 4. PC number versus Proportion of Variance Explained resulting from the PCA performed on the normalized building energy requirement data.

Linear and Polynomial Regression

Regression models built to predict the heating load requirements of a building with polynomial degrees 4-7 (HM 4-HM 7) show good performance with mean MSE ≤ 1 and mean $r^2 \geq .99$ (**Table 5)**. The models showing optimal performance when predicting heating load requirements are HM 5 and HM 6; both with a mean MSE $< .13$ and mean $r^2 \geq .998$.

Table 5. Metric assessment of regression models built to predict building heating load requirements.

	HM 1	HM 2	HM 3	HM 4	HM 5	HM 6	HM 7
Regression Model	**degree = 1**	**degree = 2**	**degree = 3**	**degree = 4**	**degree = 5**	**degree = 6**	**degree = 7**
r^2	**0.87**	**0.92**	**0.96**	**0.998**	**0.998**	**0.996**	**0.97**
mean r^2 after resampling	**0.88**	**0.93**	**0.97**	**0.998**	**0.999**	**0.999**	**0.994**
MSE	**13.50**	**8.38**	**3.90**	**0.25**	**0.25**	**0.40**	**2.85**
mean MSE after resampling	**11.97**	**6.85**	**2.92**	**0.17**	**0.12**	**0.11**	**0.59**

Regression models built to predict the cooling load requirements of a building with polynomial degrees 4-6 (CM 4-CM 6) show good performance with mean MSE ≤ 1.87 and mean $r^2 \geq .98$ (**Table 6)**. The models showing optimal performance when predicting cooling load requirements are CM 5 and CM 6; both with a mean MSE ≤ 1.19 and mean $r^2 \geq .99$.

Table 6. Metric assessment of regression models built to predict building cooling load requirements.

	CM 1	CM 2	CM 3	CM 4	CM 5	CM 6	CM 7
Regression Model	**degree = 1**	**degree = 2**	**degree = 3**	**degree = 4**	**degree = 5**	**degree = 6**	**degree = 7**
r^2	**0.81**	**0.87**	**0.93**	**0.97**	**0.98**	**0.96**	**0.78**
mean r^2 after resampling	**0.84**	**0.90**	**0.95**	**0.98**	**0.99**	**0.99**	**0.95**
MSE	**17.59**	**12.37**	**3.08**	**3.08**	**2.23**	**3.87**	**20.81**
mean MSE after resampling	**14.05**	**8.94**	**1.86**	**1.86**	**1.11**	**1.18**	**4.30**

Discussion

Among the regression models built to predict heating load requirements of a building, a regression model with polynomial degree 5 or 6, with PC 1-5 as predictors, was shown to perform optimally. Among the regression models to predict cooling load requirements of a building, a regression model with polynomial degree 5 or 6, with PC 1-5 as predictors, was also shown to perform optimally. It was expected for the models showing optimal prediction of heating and cooling load requirements to be highly similar as the outcomes of heating load and cooling load in the original building energy requirements dataset are highly correlated (**Figure 3C**).

There are several limitations of the analysis shown here. First, the dataset provided had only 768 observations. To more accurately assess each regression model's performance, we would ideally test each model on new data or have a larger initial sample size to allow for more diverse dataset resampling. Additionally, the possible values for each variable provided in the initial dataset are reported as continuous variables, but when the counts of each value within each variable are observed, there are clear patterns (highly similar or identical counts) that emerge. To assess the general applicability of the regression models noted to perform optimally here, it would be necessary to apply the models to new, more diverse, building energy requirement datasets.

References

1. Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015
   Dal Pozzolo, Andrea; Caelen, Olivier; Le Borgne, Yann-Ael; Waterschoot, Serge; Bontempi, Gianluca. Learned lessons in credit card fraud detection from a practitioner perspective, Expert systems with applications,41,10,4915-4928,2014, Pergamon
   Dal Pozzolo, Andrea; Boracchi, Giacomo; Caelen, Olivier; Alippi, Cesare; Bontempi, Gianluca. Credit card fraud detection: a realistic modeling and a novel learning strategy, IEEE transactions on neural networks and learning systems,29,8,3784-3797,2018,IEEE
   Dal Pozzolo, Andrea Adaptive Machine learning for credit card fraud detection ULB MLG PhD thesis (supervised by G. Bontempi)
   Carcillo, Fabrizio; Dal Pozzolo, Andrea; Le Borgne, Yann-Aël; Caelen, Olivier; Mazzer, Yannis; Bontempi, Gianluca. Scarff: a scalable framework for streaming credit card fraud detection with Spark, Information fusion,41, 182-194,2018,Elsevier
   Carcillo, Fabrizio; Le Borgne, Yann-Aël; Caelen, Olivier; Bontempi, Gianluca. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization, International Journal of Data Science and Analytics, 5,4,285-300,2018,Springer International Publishing
   Bertrand Lebichot, Yann-Aël Le Borgne, Liyun He, Frederic Oblé, Gianluca Bontempi Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection, INNSBDDL 2019: Recent Advances in Big Data and Deep Learning, pp 78-88, 2019
   Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Frederic Oblé, Gianluca Bontempi Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection Information Sciences, 2019
   Yann-Aël Le Borgne, Gianluca Bontempi Reproducible machine Learning for Credit Card Fraud Detection - Practical Handbook
   Bertrand Lebichot, Gianmarco Paldino, Wissam Siblini, Liyun He, Frederic Oblé, Gianluca Bontempi Incremental learning strategies for credit cards fraud detection, IInternational Journal of Data Science and Analytics
2. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research. 2017;18(17):1–5.
3. *ML: Handling imbalanced data with smote and near miss algorithm in Python* (2022) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/ (Accessed: December 14, 2022).
4. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
5. A. Tsanas, A. Xifara: 'Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools', Energy and Buildings, Vol. 49, pp. 560-567, 2012