Statistical and Machine Learning Course (MATH 6388) Project

Elizabeth Hope Thomas

Fall 2022

Introduction

This project was completed utilizing concepts that were learned throughout the Statistical and Machine Learning Course. We will perform analysis on two different datasets by creating Machine Learning Models, implementing learning algorithms, and obtaining performance measures. The first dataset will be used to solve the classification problem of credit card fraud. We will then implement Logistic Regression on the Energy Efficiency dataset (the second dataset). In this Course Project we will work to obtain reasonable performance measures on the created Machine Learning models, and we will explain the performance of the trained Machine Learning models; hypothesis/assumptions will also be made.

Dataset 1 (Credit Card Fraud Data) (Classification Problem)

The first dataset that we will be performing analysis on is Dataset 1 which is Credit Card Fraud Detection Data. The Credit Card Fraud Detection Dataset consists of 284,807 unique

credit card transactions over the span of two days with 31 features (ULB). Note that we did not include the Time variable as upon observation of Figure 1 (shown below), we can see that the Time variable is very noisy; I believe that our analysis would be more optimal to not include the Time variable.

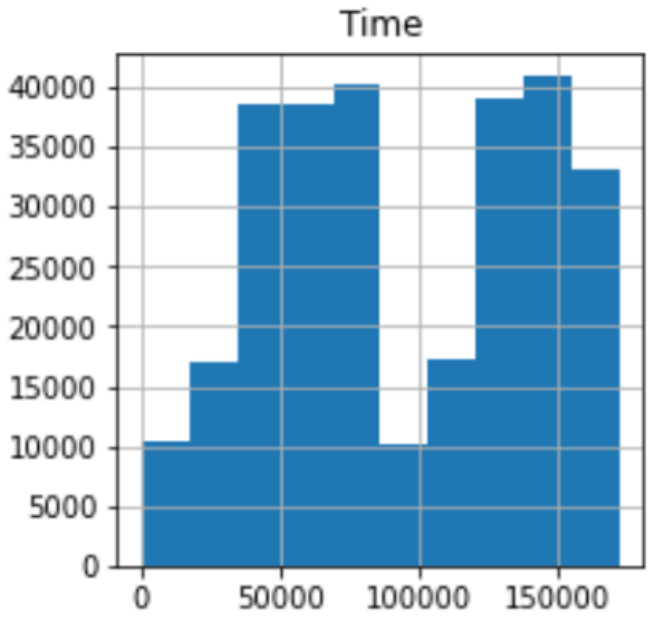


Figure 1. Histogram of the Time Variable

Within the 284,807 credit card transactions, only 492 were fraudulent (0.17% of the available credit card transactions). This implies that we have a very imbalanced dataset.

Luckily there are several machine learning methods that can help us solve this imbalanced dataset problem: (1) Random Under-Sampling (RUS), (2) Synthetic Minority Oversampling Technique (SMOTE), and (3) Random Over-Sampling. We will first solve the issue of the imbalanced dataset problem and then we will create a Machine Learning Model to solve the classification problem of weather a credit card transaction is fraudulent or not.

(1) Random Under-Sampling. We will first implement the Random Under-Sampling Method to solve the imbalanced Credit Card Fraud Data. The Random Under-Sampling Method removes data instances from the majority class (non-Fraudulent transactions) to balance the data. Note that the RUS method is a Prototype Selection algorithm, selecting data instances from the original dataset, S (3. under-sampling). We define $S' \subset S$ such that $|S'| < |S|$ (3. Under-sampling). Also note that under-sampling is considered to be a controlled under-sampling technique, meaning that the user controls the size of S' (3. Under-sampling). When utilizing the 'RandomUnderSampler' function in Python to implement RUS, if we set the replacement argument to 'True' we allow for bootstrapping.

Once the RUS technique was implemented, we obtained 492 fraudulent credit card transactions and 492 non-fraudulent credit card transactions, a very balanced dataset. However, note that the non-fraudulent transactions decreased significantly, losing over 250,000 transactions. On the other hand, credit card companies do not want to spend time investigating non-fraudulent credit card transactions, they are mainly concerned about the fraudulent transactions. After the RUS was utilized, we split the data using the 'train_test_split' function with a Train/Test split of 75%/25%. We then created a Machine Learning model using the 'SGDClassifier' to classify the credit card transactions as fraudulent or not. We obtained an average Cross Validation Score of 0.9363 (shown in Figure 2 denoted by horizontal orange line);

this implies that the SGD classifier classified the transactions well.

Figure 2. Cross Validation Scores when Utilizing Random Under-Sampling

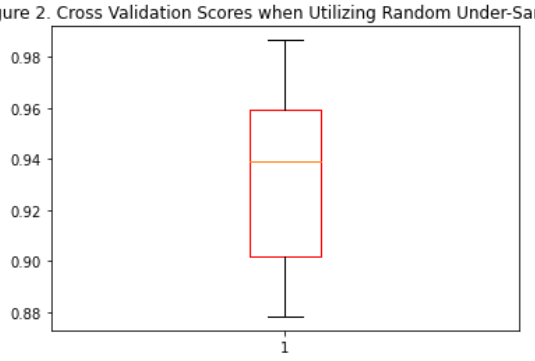
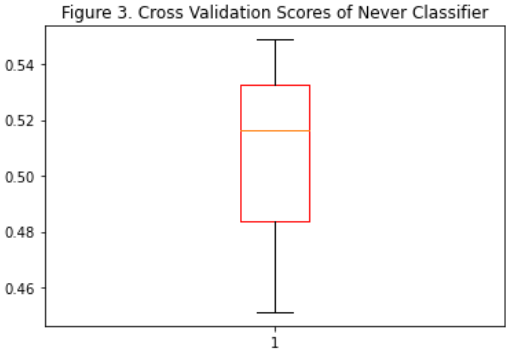


Figure 3. Cross Validation Scores of Never Classifier



The great result was investigated further by defining a classifier that always returns false. In this instance, the accuracy scores were not as "amazing", obtaining an average Cross Validation Score of 0.5054 (shown in Figure 3).

(2) SMOTE. Next, we will implement the Synthetic Minority Oversampling Technique (SMOTE) to solve in the imbalance problem within our credit card dataset. SMOTE will synthesize new examples from existing examples of the fraudulent credit card transactions. Once SMOTE was implemented we obtained 284,315 fraudulent and non-fraudulent transactions, very balanced. Similarly, we implemented a Train/Test split of 75%/25%, and then utilized an SGD Classifier obtaining a mean Cross Validation score of 0.9463 (Shown in Figure 4 by horizontal orange line).

Figure 4.Cross Validation Scores when utilizing SMOTE

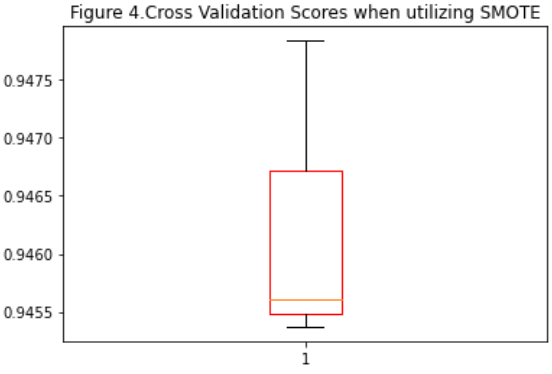
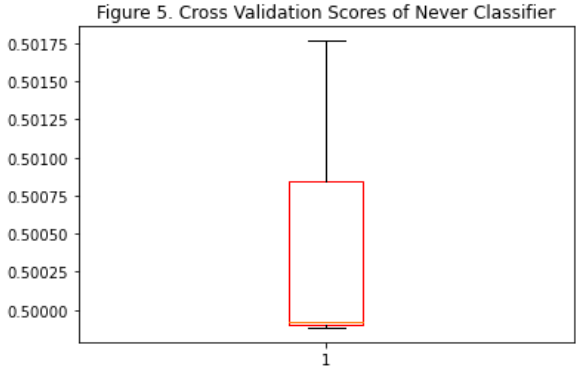


Figure 5. Cross Validation Scores of Never Classifier



These results were once again investigated further by defining the classifier that always returns false to obtain an average Cross Validation Score of 0.5005 (shown in Figure 5); not being as good of a result as previously obtained.

(3) Random Over-Sampling (ROS). Finally, we implement Random Over-Sampling to solve the imbalanced dataset problem obtaining 284,315 fraudulent and non-fraudulent transactions by duplicating data instances from the existing fraudulent transactions. As before we split the data into Training & Testing sets with a 75%/25% split to implement the SGD classifier to then obtain a mean Cross Validation score of 0.9465 (shown in Figure 5). We then investigated these results with the defined Never Classifier to obtain an average Cross Validation score of 0.5005 (shown in Figure 6); not as optimal.

Figure 5. Cross Validation Scores when Utilizing ROS Data

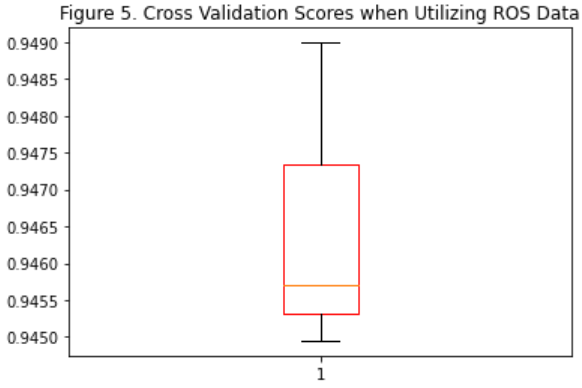
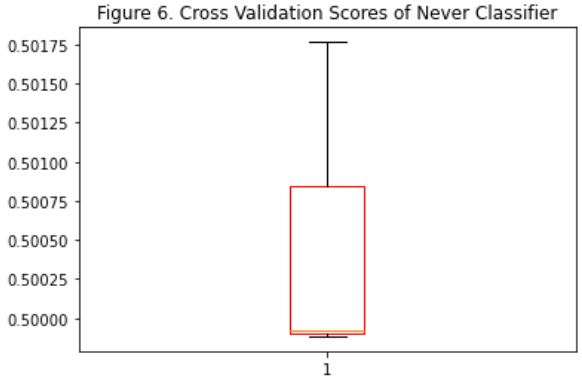


Figure 6. Cross Validation Scores of Never Classifier



Overall, the three Machine Learning Methods implemented to resolve the imbalanced Dataset problem performed about the same. Thus, I would recommend (especially to a Credit Card Company) to utilize the Random Under-Sampling technique to solve the imbalanced dataset problem as it removed over 250,000 non-fraudulent transactions from the analysis resulting in us not spending any time/effort analyzing the non-fraudulent transactions, which credit card companies are typically more interested in the fraudulent transactions.

Dataset 2 (Energy Efficiency Data Set) (Linear Regression)

"This study looked into assessing the heating load and cooling load requirements of buildings (that is, energy efficiency) as a function of building parameters" (Energy). The Energy Efficiency Data Set contains 8 feature variables (Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, and Glazing Area Distribution) and 2 response variables (Heating Load and Cooling Load) with 768 data instances. Note that all of the variables here are numerical (continuous), not categorical and do not contain any null values. Our Goal here is to predict the Heating and Cooling Load by utilizing Linear Regression (we will also implement polynomial regression). Furthermore, we will implement the following methods to predict the Heating & Cooling Load: (1) Including the Cooling Load (Y2) variable in the feature matrix to predict the Heating Load (Y1); (2) Excluding the Cooling Load (Y2) variable form the feature matrix to predict the Heating Load (Y1); (3) Including the Heating Load (Y1) variable in the feature matrix to predict the Cooling Load (Y2); (4) Excluding the Heating Load (Y1) variable from the feature matrix to predict the Cooling Load (Y2).

For each of the four methods that we are implementing above we will create the appropriate feature matrix and response variable to then split the data using a Train/Test split of 80%/20% to then perform Linear and Polynomial Regression. Note that all Polynomial Regression performed we used degree 2.

First, we will predict the Heating Load (Y1) by having the Cooling Load (Y2) in the Feature Matrix. Observing Table 1 we can see that we have obtained great results for the Linear Regression, but even better results from implementing Polynomial Regression.

	Mean Squared Error:	Coefficient of Determination:
Linear Regression:	2.72	0.97
Polynomial Regression:	0.47	1.00

Table 1. Mean Squared Error and Coefficient of Determination for Method (1)

Next, we will predict the Heating Load (Y1) by excluding the Cooling Load (Y2) in the Feature Matrix. Studying Table 2, we can observe that the Linear Regression obtained good results; however, the Polynomial obtained the more optimal results in this setting. Also, notice that the results obtained here are not quite as good as the results obtained from including the Cooling Load Variable in the feature matrix.

	Mean Squared Error:	Coefficient of Determination:
Linear Regression:	8.55	0.91
Polynomial Regression:	0.53	0.99

Table 2. Mean Squared Error and Coefficient of Determination for Method (2)

Furthermore, we will predict the Cooling Load (Y2) by including the Heating Load (Y1) in the Feature Matrix. Upon observation of Table 3 we can see that we have once again obtained

a good performance for the Linear Regression, obtaining a bit of a better performance for the Polynomial Regression.

	Mean Squared Error:	Coefficient of Determination:
Linear Regression:	3.78	0.96
Polynomial Regression:	2.23	0.99

Table 3. Mean Squared Error and Coefficient of Determination for Method (3)

Finally, we will predict the Cooling Load (Y2) by excluding the Heating Load (Y1) in the Feature Matrix. We can see from Table 4 that as we have previously seen we obtained better performance measure values when Polynomial Regression of Degree 2 was implemented. We should also notice that we did not obtain as optimal of results as we did when the Heating Load variable was included in the feature matrix.

	Mean Squared Error:	Coefficient of Determination:
Linear Regression:	11.53	0.87
Polynomial Regression:	2.59	0.97

Table 4. Mean Squared Error and Coefficient of Determination for Method (4)

Considering the results obtained after implementing Linear and Polynomial Regression on the four different methods previously discussed to predict the heating and cooling load, we can conclude that Polynomial Regression produced the most optimal results in all four scenarios. Also, both Linear and Polynomial Regression obtained better performance measures when predicting the Heating and Cooling load when the opposing variable is included in the feature matrix (either Heating or Cooling Load).

References

3. under-sampling#. 3. Under-sampling - Version 0.9.1. (n.d.). Retrieved December 3, 2022, from https://imbalanced-learn.org/stable/under_sampling.html

Energy efficiency · MASTER · data science dojo / datasets. Code. (n.d.). Retrieved December 4, 2022, from https://code.datasciencedojo.com/datasciencedojo/datasets/tree/master/Energy%20Efficiency

ULB, M. L. G.-. (2018, March 23). *Credit Card Fraud Detection*. Kaggle. Retrieved December 3, 2022, from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?select=creditcard.csv