Machine Learning Project

Samin Basir

I. INTRODUCTION

During this semester we learned many different machine learning techniques that can used for either classification, regression, or both. The goal of this project is to show case and apply what we have leamed in order identity the proper model to best fit our different datasets. . This project will analyze problems, design a machine learning solution, implement ML algorithms, and evaluate them on three data sets (one for classification and one for regression from Small Data Sets and one data set either classification or regression from Large Data Sets).

II. CLASSIFICATION DATASET

A. *Credit card fraud: The Dataset*

Since this is a classification dataset, it is important to choose what type of method to use. The Credit Card Fraud Detection seems to be unbalanced dataset with a dimension of [284807, 31]. This dataset provides a mixture of non-fraud and fraud transactions. As the dataset is processed. The value of non-fraud comes to 284315 transactions while the value of fraud is 492 transactions.

B. *Data Processing*

The dataset determines fraud and non-fraud as 1 or 0, respectively, via the "Class" column, so *X* will become our data that is not "Class" while *y* will become our data that is and only is "Class." Next, I split our *X* and *y* into two subsets: training and testing data. Then, I applied feature scaling which will normalize our training and testing data. With this, we can continue to train our classification techniques.

C. *Credit card fraud Logistics Regression*

The first algorithm is Logistic Regression which predicts a binary outcome based on a set of independent variables. The sets were trained, fitted, and predicted onto the Logistics Regression model. Next, the confusion matrix is computed and displayed as shown in Figure 1.

With this matrix, we can determine if our model best fits this dataset. We can calculate the recall and precision score which is important due to the binary outcome of our fraud and non-fraud transactions.

However, since this dataset is heavily imbalanced, we will not receive a proper value as it is heavily biased towards our non-fraud cases due to its extremely large value. The values come out to recall: 65% and precision: 87%. This results in false positives being more biased towards the model, which is what is not desired. So, this logistic regression is not fit for the goal.

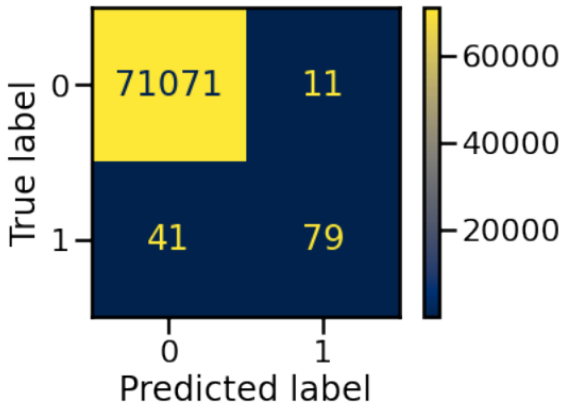

Figure. 1. Logistic Regression Confusion Matrix

D. *Logistics Regression with RandomOverSampler*

Imblearn was applied to fix the issue of imbalanced dataset and to remove any bias towards the minority class. In this case, oversampling was used. Oversampling randomly duplicates examples from the minority class and adds them to the training set. We can see the resulting confusion matrix as shown on Figure 2.

Oversampling has improved in terms of recall score being significantly higher than the precision score. The resulting values showed recall being 90% and precision being 6%. The oversampled model does a better job fitting towards the dataset.

It seems that Logistics Regression can do a better job to fitting the data. Next, we will apply under sampling to the Logistics Regression.

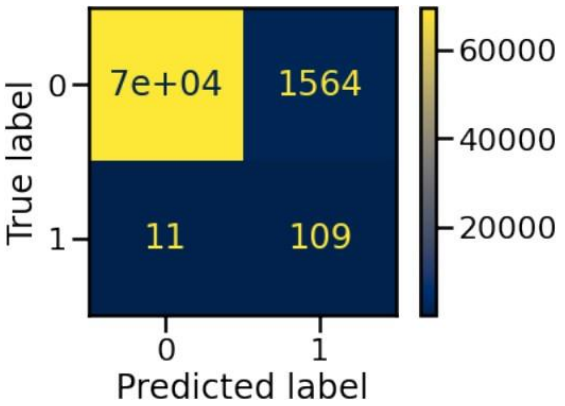

Fig. 2. Over sampled Logistics Regression Confusion Matrix

E. Logistics Regression with RandomUnderSampler

Undersampling is a technique to balance uneven datasets by keeping all of the data in the minority class and decreasing the size of the majority class. It is one of several techniques data scientists can use to extract more accurate information from originally imbalanced datasets. This technique also comes from the same library as oversampling, imblearn. As applied under sampled Logistic Regression, you can see the resulting confusion matrix on Figure 3.

This model has done a better job in reducing the number of false negatives than oversampled and normal Logistic Regression. Not only this, but the resulting recall score is significantly higher than the precision score. The resulting values is recall: 92% and precision: 4%.

Its seems that this model best fits the given data.

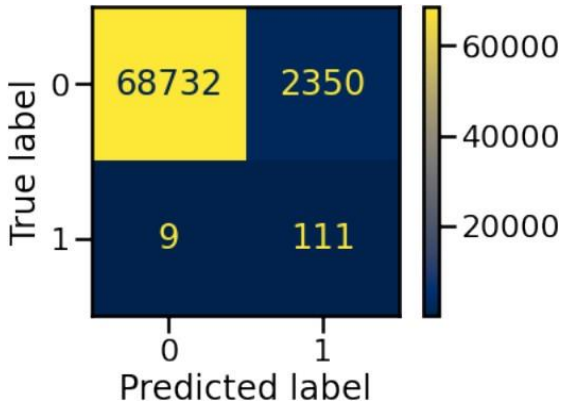

Fig. 3. Under sampled Logistics Regression Confusion Matrix

F. Random Forest

The next algorithm is Random forest which is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned Following the same steps as Logistic Regression, the training and testing sets were trained and fitted with a resulting matrix as seen on Figure 4.

Once again, the imbalanced poses a problem for models that do not resample. Resampling is when we create a newly transformed training set in which examples have a different class distribution.

Since resampling has not been done here, we are resulting in scores that depict in bias of false positives. The recall score turns out to be lower than the precision score. The calculated values come out to recall being 78% and precision being 94%.

In the next few points, we will talk about the recall and precision value through over sampling and under sampling.

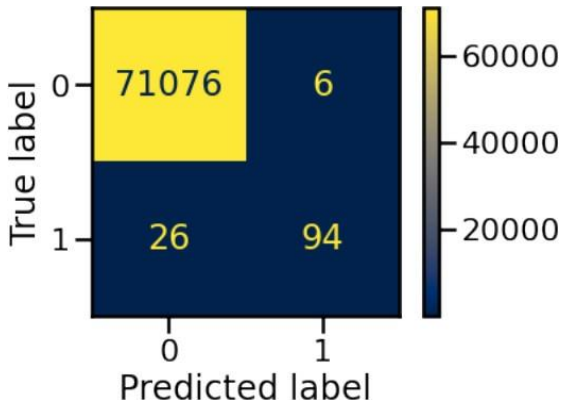

Fig. 4. Random Forest Confusion Matrix

G. Random Forest with RandomOverSampler

The resulting matrix of over sampled Random Forest can be found at Figure 5. Just by looking at the confusion matrix, we can tell that there is a slight improvement of the model with just the use of oversampling. With the false negative value being lower which means there is an improvement, however, this is not enough to deem it efficient.

The resulting recall: 80% is still lower than the resulting precision of 93%. Once again, this has proven that oversampling an extremely imbalanced dataset is not the way to go.

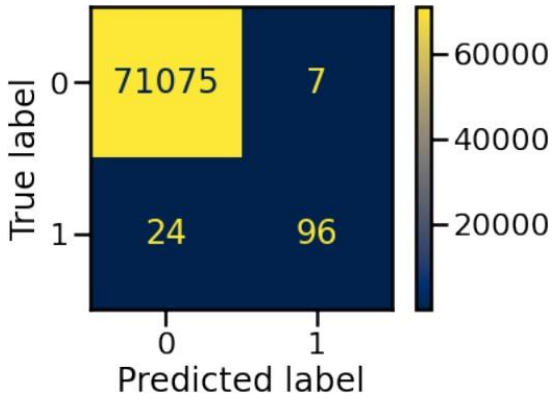

Fig. 5. Over sampled Random Forest Confusion Matrix

H. Random Forest with RandomUnderSampler

The resulting matrix of under sampled Random Forest can be found at Figure 6. By reading the confusion matrix, we notice a much better improvement than was shown with the oversampled Random Forest confusion matrix, Figure 5.

With the work of under sampling, the bias on the dataset has disappeared, and is being evenly distributed. The final recall is 88% and precision is 4%, this determines that this model is the better choice as it removes the bias on the false positives. Along with the

decrease of false negatives and increase to true negatives. So this shows that under sampled Random Forest works great.

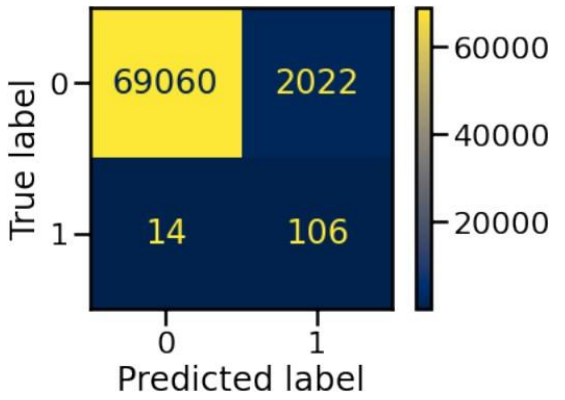

Fig. 6. Under sampled Random Forest Confusion Matrix

I. Neural Network

Neural Network is the third technique used for this classification technique. A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.. The layers can differ, where the first layer is known as the input layer. The inner layers are known as hidden layers and the last layer is known as the output layer. These layers interlock, trained, and fitted on.

When Neural Network is applied to the credit card Dataset, we can see that this technique is being trained on imbalanced data as the recall and precision values differ in the way of false positives being biased and false negatives being ignored.

Over sampling and under sampling is a possible way to fix this issue. As shown in Figure 8 and Figure 9. Based on the result, we can see that the under sampled neural network is the best out of neural networks for this dataset. The resulting recall and precision for under sampled is recall at 90% and precision at 5%.

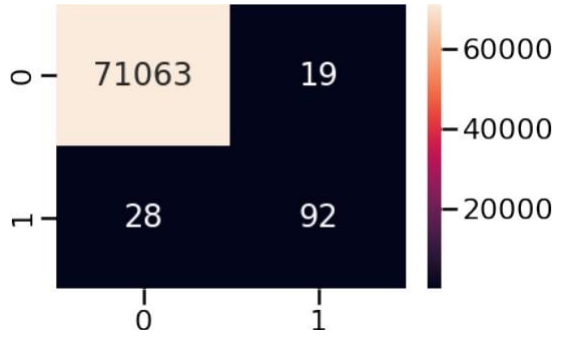

Fig. 7. Neural Network Confusion Matrix

J. Neural Network with RandomOverSampler

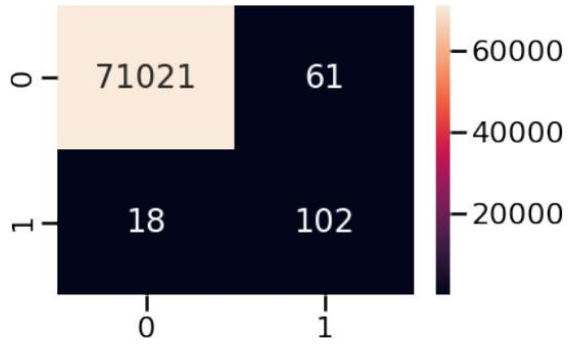

Fig. 8. Over Sampled Neural Network Confusion Matrix

K. Neural Network with RandomUnderSampler

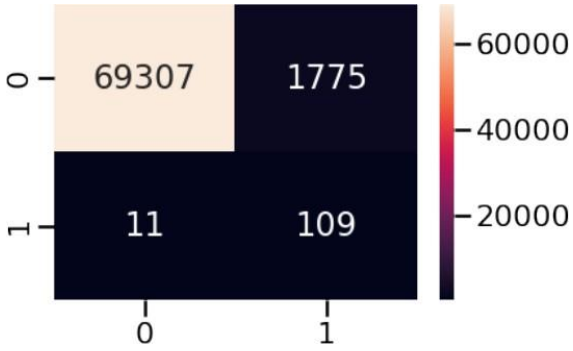

Fig. 9. Under Sampled Neural Network Confusion Matrix

L. Conclusion

In conclusion, based on the data we were able to calculate, we can conclude that under sampled Logistics Regression have the best recall score of 92% with the best precision score of 4%.

	Model	Recall	Precision	F1 Score
0	LogisticsRegression	0.658333	0.877778	0.752381
1	OverSampledLogisticsRegression	0.908333	0.065152	0.121584
2	UnderSampledLogisticsRegression	0.925000	0.045104	0.086013
3	RandomForest	0.783333	0.940000	0.854545
4	OverSampledRandomForest	0.800000	0.932039	0.860987
5	UnderSampledRandomForest	0.883333	0.049812	0.094306
6	NeuralNetwork	0.766667	0.828829	0.796537
7	OverSampledNeuralNetwork	0.850000	0.625767	0.720848
8	UnderSampledNeuralNetwork	0.908333	0.057856	0.108782

Table 1: Credit Card Classification Report

III. REGRESSION DATASET

A. *Energy Efficiency The Dataset*

The regression dataset, Energy Efficiency, is a dataset with a dimension of [768, 10]. The given data looks into assessing the heating load and cooling load requirements of buildings (that is, energy efficiency) as a function of building parameters

Energy analysis using 12 different building shapes simulated in Ecotect has been performed. The buildings differ with respect to the glazing area, the glazing area distribution, and the orientation, amongst other parameters. It can also be used as a multi-class classification problem if the response is rounded to the nearest integer.

RMSE is the root mean square error which will determine the quality of our model's predictions. The R2 score will determine how good our model fits the dataset. A low RMSE and high R2 score will typically mean that the model fits well.

B. *Data Processing*

The *X* will be data that is not "Y2" while *y* will be data that is and only is Y2" Next, we split our *X* and *y* into two subsets: training and testing data. Then, apply feature scaling which will normalize our training and testing data. With this, we can continue to train our regression techniques.

C. *Linear Regression*

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable, however, we will be modelling the relationship between all features.

Once the data processing has been completed, we train, fit, and predict on the Linear Regression model. Next, we create a scatter plot that shows our model's prediction over the true prediction values as shown on Figure 10.

According to the figure, we can see that the RMSE score is 1.95 and a R2 score is 96%. There is no way to improve this model as there is no hyperparameters available to fine tune.

The reason behind this is because Linear Regression is part of the linear model family which usually just models a straight line. Next we can try using another linear model family member that has a few hyperparameters such as the Ridge Regression.

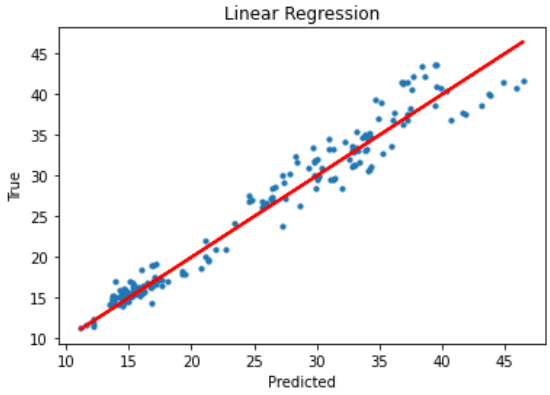

Fig. 10. Linear Regression Scatter Plot

D. *Ridge Regression*

For the Ridge Regression,we can apply grid_search_cv for hyperparameter tuning. However, there was only one hyperparameter that was changeable which is alpha. With my attempt to try and get better RMSE and R2 score resulting in having similar results to Linear Regression.

The results were RMSE: 1.94 and R2: 96%..

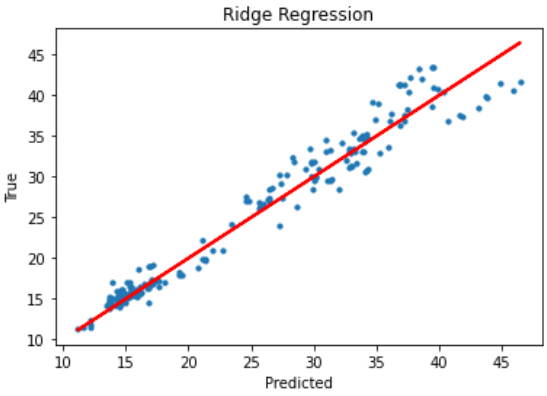

Fig. 11. Ridge Regression Scatter Plot

E. *Support Vector Machine*

Next we can try Support Vector machine. A support vector machine (SVM) is machine learning algorithm that analyzes data for classification and regression analysis. SVM is a supervised learning method that looks at data and sorts it into one of two categories.

With SVM the results we got were RMSE: 2.48 and an R2 score of 93%, next we try hyperparameter tuning it using grid_search_cv.

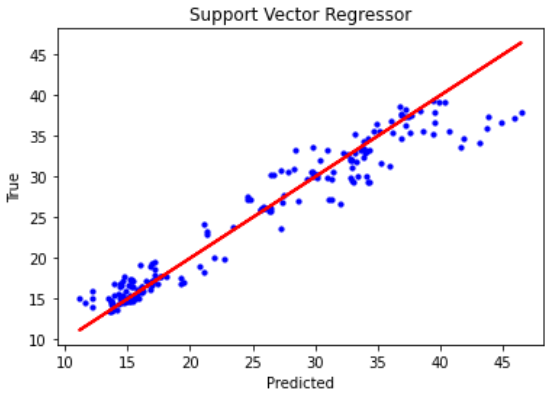

Fig. 12. Support Vector Regressor Scatter Plot

F. *Support Vector Machine (Tuned)*

In this SVM, I applied grid_search_cv and got the best parameters to best fit this dataset. The specific parameters came out to be an "rbf" SVM with a C value of 9, gamma set to auto, and epsilon value of 0.2.

Based on Figure 13, we the resulting values turned out to be RMSE: 1.77 with an R2 score: 96%.

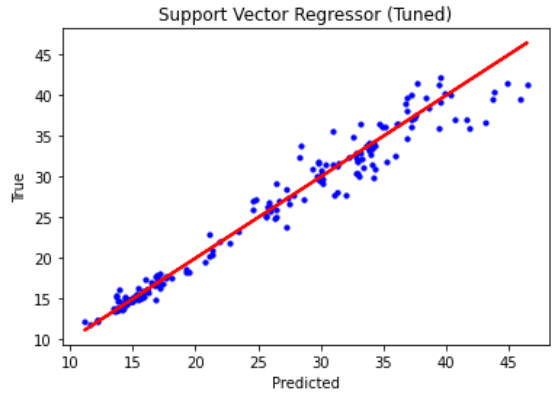

Fig. 13. Support Vector Regressor Scatter Plot (Tuned)

G. *Neural Network*

In neural network, we will not apply grid_search_cv, but instead apply batch normalization, so that we can get better results each batch run. This has definitely improved the score as I was achieve usually high values of RMSE and R2, without it.

According to figure 14, we can see that this model was not able to appropriately fit the model well.

The resulting RMSE is 26.14 with an R2 score of -6.32%

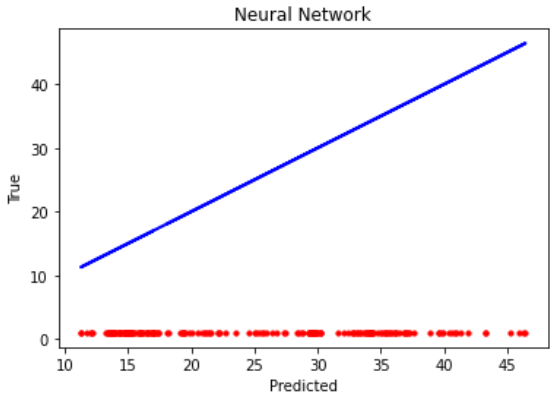

Fig. 14. Neural Network Scatter Plot

H. *Conclusion*

In conclusion with the values we got using Neural Network, we can say this is not the best fit .

The best fitting was hyperparameter tuned SVM it fit the data well

Overall, using different techniques can achieve the best model that is not underfitted as well as not overfitted, but good enough.

	Model	RMSE	R2-Score
0	LinearRegression	1.952846	0.958784
1	RidgeRegression	1.944725	0.959126
2	SupportVectorRegression	2.478816	0.933593
3	TunedSupportVectorRegression	1.770584	0.966119
4	NeuralNetwork	26.146568	-6.322252

Table 2: Credit Card Classification Report

IV. CONCLUSION

This was a great way to apply the skills we had learned throughout this course, applying these skills in a project like this helps improve our knowledge on Machine Language. I had trouble on the Neural Network part and was not able to get the data I sought after because of time constraints. Despite this I have learned a lot from this project.