I. INTRODUCTION

In this research, we have two datasets. The first one is about credit card fraud and the second one is about concrete compressive strength. We have imbalanced dataset with credit card fraud dataset. We apply Logistic Regression, Decision Tree Classification, and Neural Networks to predict the transition that is fraud or not fraud. We report the accuracies of 96%, 91%, and 95% respectively. In general, Decision Tree and Neural Network deal with Imbalance Dataset better than Logistic Regression.

We apply Linear Regression, Polynomial Regression, and Neural Network to predict the concrete compressive strength. We report mean squared error values 95, 55, and 44 respectively. Polynomial Regression and Neural Network are good fit for our result, whereas Linear Regression is underfitting.

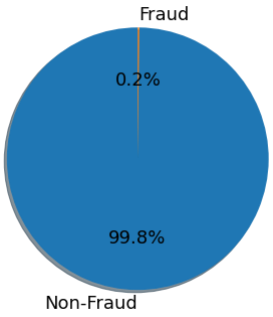
II. CLASSFICATION DATASET: CREDIT CARD FRAUD

A. The Dataset

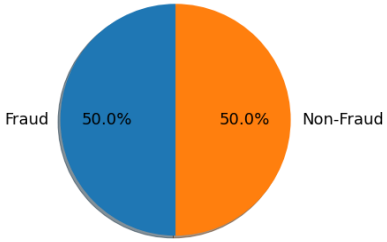
We have a dataset that contains 31 features and 284807 samples. We have the features V1 to V28, Time, Amount and Class. We can use them to predict a transition that is fraud or not by using this dataset. This dataset is downloaded from Kaggle.com.

B. Data Exploration

The first step we need to do is exploring the data to see what we have inside the dataset. We can see this data set is imbalanced. We have number of fraud transition is much larger than the number of non-fraud transition (Figure 1). We have very few non-fraud transitions in our dataset that may make our model does not perform well on new data. Our model is confident when it detects the non-fraud transition because there are very few fraud transitions in our data. Imbalanced data makes our model is not correct as we expect. We will use under resample to balance the number of fraud and non-fraud to 50-50 on training set.

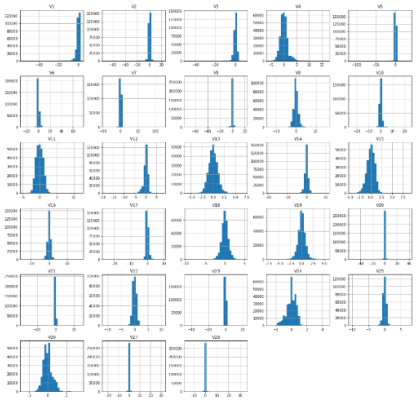


After using under-resampling, we have 50% fraud and 50% non-fraud in our training set. Under-resampling is randomly selecting samples from the majority class, then deleting these samples from the training dataset.



C. Pre-Processing the data.

Feature V1 to V28 are scaled, but Time and Amount are not scale in the dataset. Thus, we need to scale Time and Amount to have the same scale with V1-V28. V1-V28 are normal distributed, so we can use scale Amount and Time to normal distributed.



We need to scale the Time and Amount columns because they have the value very different from V1 to V18 that is scaled before adding to the dataset. V1 to V18 is normal distribution, so we need to scale time and amount to normal distribution to have the same scale. We use StadardScaler from sklearn.preprocessing to do it.

D. Results with Different Models.

1. Logistic Regression Imbalanced Data.

First, we use Logistic Regression for our first model, but we apply the imbalanced data to see what happen if we do not resample the dataset. In this case, we get the accuracy is 99.9% that is very high, but the precision and recall are 86.4% and 58.2% respectively. Precision is the accuracy of our model predicted fraud transitions. Recall is how many transitions our model correctly identifies fraud. With imbalanced data we have a low recall 58.2% that we do not want for our model.

2. Logistic Regression with Balanced Data.

We now try applying Logistic Regression with resample dataset so see if we get better recall. We get the accuracy is 96.3% that is lower than without resampling, but we have higher recall 92.8% recall and 4.2% precision.

Now, our recall is better than the imbalanced data. It is safe if we have more non-fraud transitions are detected as fraud. We have recall 92.8% that is better than 58.1% with imbalanced data. It means we detect more fraud transitions.

Classification Report Logistic Regression with Balanced data				
	Precision	Recall	F1-score	Support
Non-fraud	1.00	0.96	0.98	56864
Fraud	0.04	0.93	0.08	98
Accuracy			0.96	56962
Macro avg	0.52	0.95	0.53	56962
Weighted avg	1.00	0.96	0.98	56962

3. Decision Tree with Imbalanced Data.

We use Decision Tree for our second model. We get 99.9% accuracy for imbalance data, and the precision and recall are 79.1% and 73.5% respectively. Decision Tree give us the same accuracy as Logistic Regression without resampling. However, Decision Tree give us higher recall. Decision Tree is better than Logistic Regression to deal with imbalanced data.

4. Decision Tree with Balanced Data.

We now apply resampling and get 91.1% accuracy that lower than imbalanced data, but we get 1.7% precision and 92.8% recall for our result. We have very low precision and high recall. It is very good for our model because we have can detect more fraud in this case with high recall.

Classification Report Decision Tree with Balanced Data				
	Precision	Recall	F1-score	Support
Non-fraud	1.00	0.99	0.99	56864
Fraud	0.12	0.84	0.21	98
Accuracy			0.99	56962
Macro avg	0.56	0.91	0.60	56962
Weighted avg	1.00	0.99	0.99	56962

5. Neural networks with imbalanced data.

We get 99.9% accuracy with Neural network as well in case with imbalanced dataset. We have 68% recall and 83% precision that is not bad for our result. However, we can balance the dataset and get better result.

6. Neural networks with balanced data.

Now, we get 95.1% accuracy with balanced data. However, we have 88.5% recall and 2.8% precision. This high recall gives us a good chance to predict fraud transition.

Classification Report Neural Network with Balanced Data				
	Precision	Recall	F1-score	Support
Non-fraud	1.00	0.95	0.98	71089
Fraud	0.03	0,88	0.06	113
Accuracy			0.95	71202
Macro avg	0,51	0,92	0.52	71202
Weighted avg	1.00	0.95	0.97	71202

E. Analysis.

1. Logistic Regression.

We use Logistic Regression with C = 1. If we increase the C, we get higher recall, but lower accuracy. If we decrease C, we get higher accuracy, but lower recall. Thus, we choose C = 1 for our Logistic Regression.

We have an imbalanced dataset, so we need to resample it. Before we apply resampling to our dataset, we get the very high accuracy 99.9%, but the precision is high. If we can reduce the precision score and increase recall, we can detect more fraud with high recall.

Confusion Matrix with Imbalanced Data
Logistic Regression

The low recall means that our model does not predict fraud transitions well. The reason is because we have many non-frauds in our dataset, so our model is very confident to detect non-fraud transitions.

When we apply resampling for our dataset. We get better result although we have lower accuracy. We have higher recall 92.8% that is very good for our model, but our precision is lower 4.2%. It is a trade-off between precision and recall. We cannot increase precision and recall at the same time. One increases while another decreases.

Confusion Matrix with balanced Data
Logistic Regression

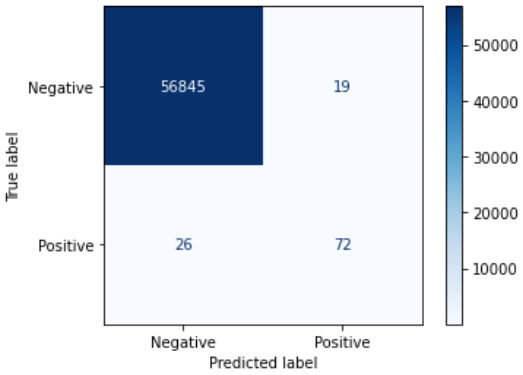
If we have low precision, it does not affect our result. Low precision means we have more false negative cases (non-fraud). However, detecting Fraud is more important for us, so we can have low recall and high precision for trade-off.

Our Logistic Regression is good fit for our data. The training accuracy and validation accuracy are almost the same.

When we draw a learning curve graph, we need to be careful. We need to avoid resample the validation test set during cross validation step. If we do not do it, the data can be leak from test set to training set. We can fit it by using pipe in imbalanced-learn. The pipe helps us to avoid touching test set during cross validation.

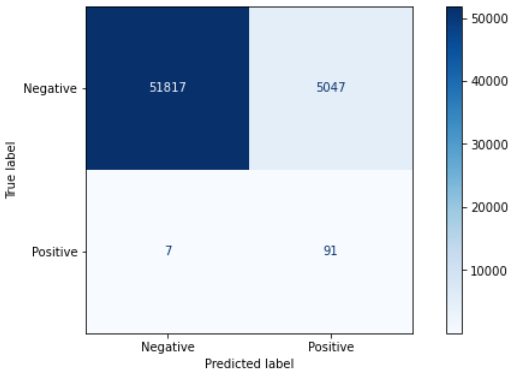
2. Decision Tree.

Decision Tree give us recall 73.5%, but what happens if we apply resampling for our dataset. As showed in previous step, we have better result when we apply under-resample to balance our data.



Confusion Matrix with Imbalanced Data
Decision Tree

We have high recall 92.8% and low precision 1.7% after balancing the dataset. We can detect more Fraud with high recall. After we balance the dataset, we detect more fraud than before 91 compared to 72.



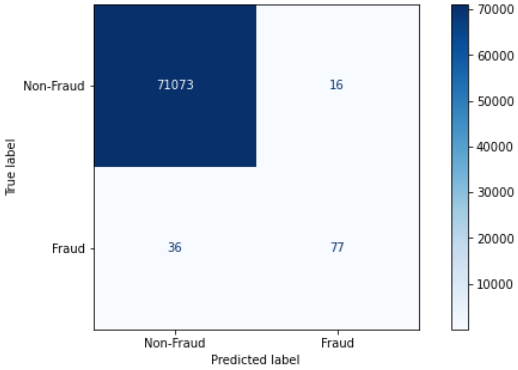
Confusion Matrix with Balanced Data
Decision Tree

Our Decision Tree is good fit for our data. The training accuracy and validation accuracy are almost the same.

3. Neural Networks.

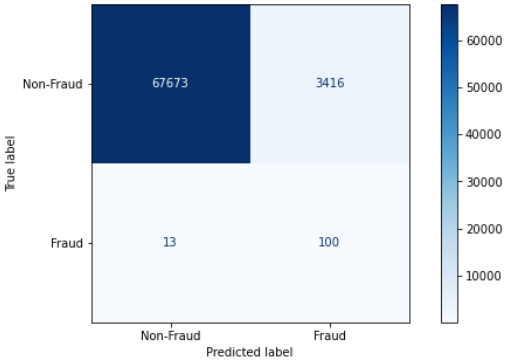
We use the sequential to build for our NN (Neural Network) model. We two hidden layers that have 30 nodes and an output layer has 2 nodes that is fraud and non-fraud. We have total 1,922 params for our NN model.

Neural Network and Decision Tree can handle imbalanced dataset better than Logistic Regression. NN and DT have higher recall than LR.



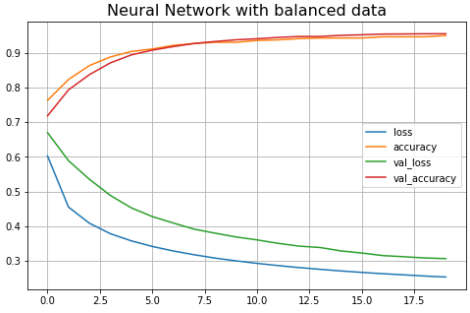
Confusion Matrix with Imbalanced Data Neural Network

If we balance the dataset, we even get better result. We can detect more fraud 100 compared to 77. We have higher recall with 88.5%. It is not bad for our expected.



Confusion Matrix with Balanced Data Neural Network

Our neural network model fit with our dataset. There is no gap between training accuracy and validation accuracy. We use l2 regularization factor of 0.001 to reduce the risk of overfitting.



F. Comparison.

Decision Tree and Neural Network deal with imbalanced dataset better than Logistic Regression. Logistic Regression gives us very low recall 58.1% before balancing, but Decision Tree and Neural Network give us better recall although we have imbalanced dataset.

Logistic Regression is our best model for credit card fraud dataset. It gives us 96% accuracy and 92.8% recall. It also gives us 0.97 ROC AUC score. These numbers are highest score of three our models.

Logistic Regression model can fit very well with our model as we see that there is no gap between training and validation set.

Three our models give us good results. They have accuracy over 90% that is better than the random classifier (50%). We have higher recall after we balance our dataset. It means we can detect more fraud transition although we must have lower precision. However, lower precision is not bad. We can leave non-fraud transition and in exchange, we can detect more fraud transition.

II. REGRESSION DATA SET: CONCREATE COMPRESSIVE STRENGTH DATA SET.

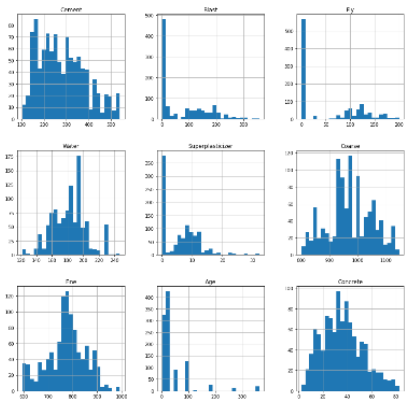
A. The data set.

We have a dataset used to determine the concrete compressive strength. Our dataset has 9 features and 1030 samples. Some of feature: Cement, Blast Furnace Slag, Fly Ash, Water, etc.

B. Pre-Processing the data.

We scale our dataset to normal distribution as we did in the previous dataset about Fraud Detection. Our dataset is almost normal distribution, so we can use

StandardScaler function from sklearn to convert our dataset to normal distribution. Now, all our features have a same scale. The neural network does not work if we do not have inputs with the same scale.



C. Results with Different Models.

1. Linear Regression.

After we apply Linear Regression, we train our dataset and get the mean square score is 96. It seems to be high to be excepted it. Our model may underfitting, so we can try with Polynomial Regress with degree 2 to see our result.

2. Polynomial Regression.

We apply Polynomial Regression and get the mean square error is 55. We now have better result for our model.

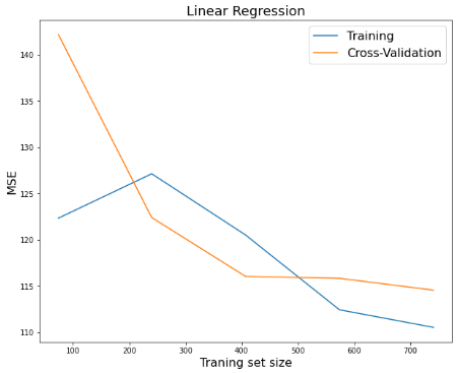
3. Neural Networks

We have the mean square error for this model is 49. It is our best result. We will discuss whether our model is overfitting or not in the next step. If we do not normalize our dataset, the mean square is very high 39381. It seems our NN model does not work if we do not have the data with features that have the same scale.

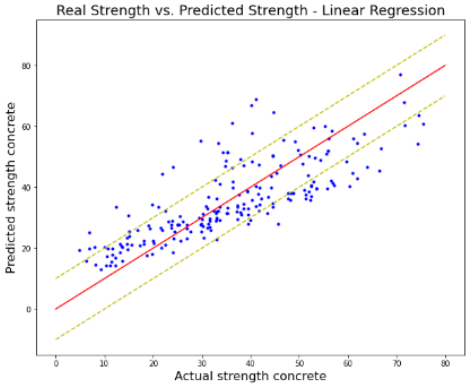
D. Analysis.

1. Linear Regression.

Our Linear Regression Model is underfitting. The mean square error is very high and there is a gap between training and validation.



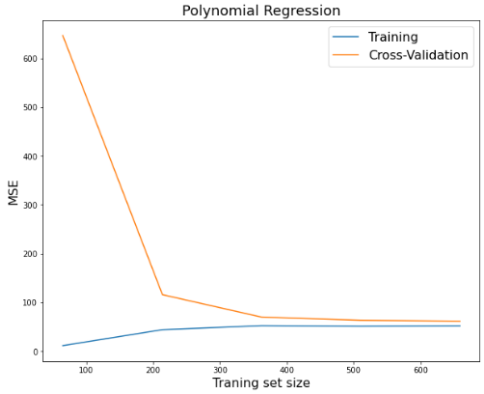
We need predicted values and actual values are the same. Thus, points in graph below should be close to the line y = x.



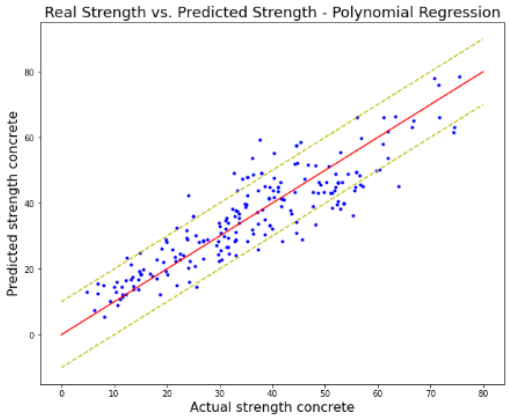
Our Linear Regression model is underfitting, so the points are kind of far from the red line.

2. Polynomial Regression.

When we apply Polynomial Regression with degree 2, we get lower error score. The validation and training error are the almost the same line that means our model fit with our data. Our Polynomial Regression model is less risk to be overfitting or underfitting.



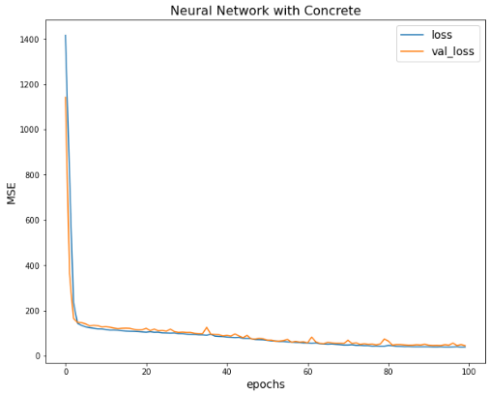
With Polynomial Regression, points in the graph predicted values vs. actual values are close to the red line. It means our model predicts values well.



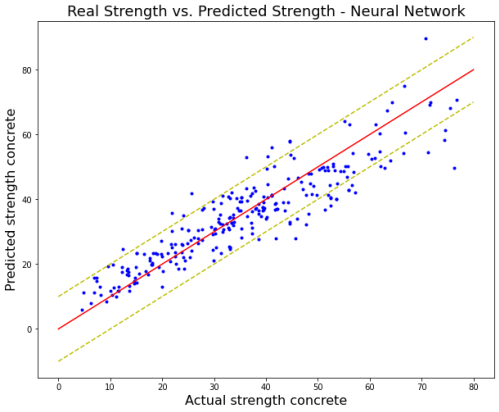
3. Neural Networks

We use two layers for our NN model in this dataset. We use one hidden layer and one output layer to reduce the risk of overfitting. We use 20 nodes for our hidden layer, and 1 node for output layer because we only want to predict a single value that is strength concrete. We do not use activation function as we use in classification from previous dataset. We use mean square error for our loss function. We use 100 epochs for our iteration. If we set higher epochs the mean square error, we can reduce the MSE.

Loss cost and validation cost are almost the same line that means our model is good fit.



There is no gap, and the mean square error is low 48. When we run several times, we may get the lowest MSE is 44 with NN model.



It is presented in the graph. We have points that very close to the red line. It means our model fit very well with our dataset.

E. Comparison.

Linear Regression gives us underfitting model, so the mean square error is high 96. However, we have better models with Polynomial Regression and Neural Network. They have low mean square error 55 and 48 respectively. The neural network is our best model because there is no gap between training and validation error, so it is good fit. Polynomial Regression model is also good, but it has higher chance to have underfitting because there is a gap between training and validation error.