**CREDIT CARD FRAUD DETECTION WITH AUTOENCODER**

PROJECT REPORT

**NAME REG. NO. ROLL NO**

VIKASH KUMAR (11803824) 26

AMAN KUMAR (11803761) 25

ABHISHEK YADAV (11803754) 12



*Department of Intelligent Systems*

*School of Computer Science Engineering*

*Lovely Professional University, Jalandhar*

**Student Declaration**

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we shall take full responsibility for it.

VIKASH KUMAR

ROLL NO: 26

AMAN VERMA

ROLL NO: 25

ABHISHEK YADAV

ROLL NO: 12

PLACE: JALANDHAR, PUNJAB

DATE: 13/04/2020

**BONAFIDE CERTIFICATE**

Certified that this project report “ *CREDIT CARD FRAUD DETECTION USING AUTOENCODER* ” is the bonafide work of” *VIKASH KUMAR, AMAN KUMAR VERMA, ABHISHEK YADAV* ” who carried out the project work under my supervision.

**<<Signature of the Supervisor>>**

**<<Name of supervisor>>**

**<<Academic Designation>>**

**<<ID of Supervisor>>**

**<<Department of Supervisor>>**

**TABLE OF CONTENTS**

**TITLE PAGE NO.**

**1. Introduction and objective …………………………... ................ ……………………………….. 4**

**1.1 Introduction………………………….......................................... …… …………………………4**

**1.2 Objective........................................................................................................... 4**

**2. Autoencoders…………………. …………………………... ................ ……………………………….. 5**

**2.1 About………..………………………….......................................... …… …………………………5**

**2.1.1 Anomaly Detection using Autoencoders..................................………………………6**

**2.2 Algorithm….………………………….......................................... …… …………………………6**

**2.2.1 Data..................................……………………………………………………………………….……6**

**2.2.2 Import libraries and load the data…………………………………………7**

**2.2.3 Exploration……………………………………….…………………………………………8**

**2.2.4 Amount Statistics……………………………….………………………………………10**

**2.2.5 Graphical Representation………………....……………………………………….11**

**2.2.6 Training the Model……….………………....………………………………………..14**

**2.2.7 Building the Model……...………………....………………………………………..14**

**2.2.8 Reconstruction Error…….………………....……………………………………….16**

**2.2.9 Reconstruction Error without Froud……..…………………………………..17**

**2.2.10 Reconstruction Error with Froud……..……………………………………..18**

**2.2.11 Prediction………...…….………………....………………………………………….19**

**2.2.12 The confusion Matrix...……………....………………………………………….20**

**3 Conclusion………………………………….…………………………………………….21**

**4 Technology Used………………………………….……………………………………..21**

**5 References………………………………….…………………………………………….21**

**1 Introduction and Objective**

1.1 Introduction

It’s an era of digitalization. Online transactions have got a boom. Though there are many modes of transactions available, credit cards are frequently used to make large transactions. With the ease and other economic perks, there is a surge in the number of credit card transactions too. When an unauthorized person uses the credit card of a person for transactions, it’s credit card fraud. Because fraud does not happen often it is a difficult task to recognize fraudulent transactions.

In this report we will propose a possible method to recognise credit card fraud detection using autoencoders. An anomaly is a data point that is sufficiently different from the other data points. Because fraud happens rarely it can be considered an anomaly and thus anomaly detection can be used to detect fraud.

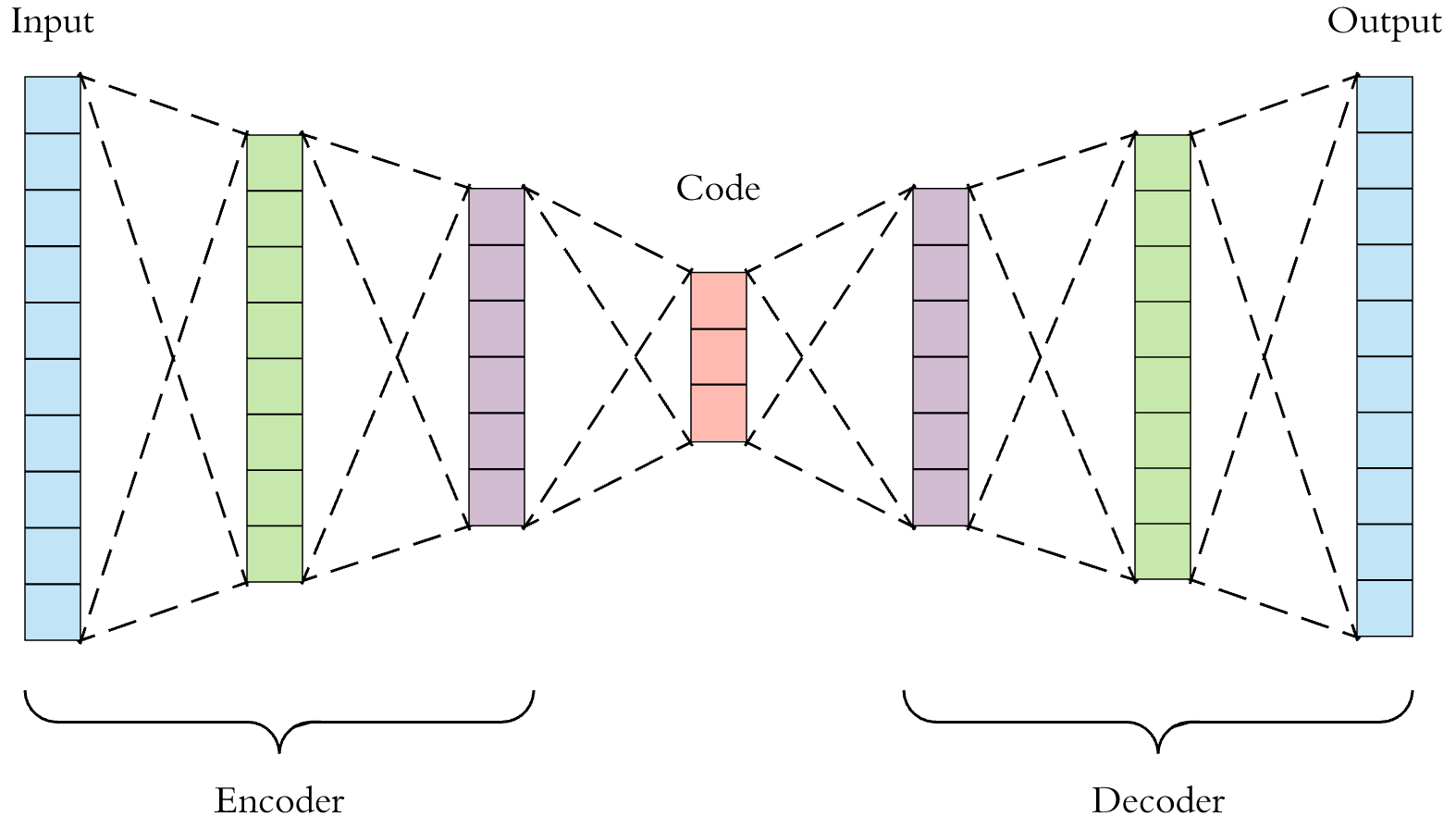
1.2 Objective

The objective of the project is to make a program that detects credit card fraud using autoencoders. The program is trained using certain real datasets. After training, the program becomes appreciably capable of detecting credit card fraud.

**2 Autoencoders**

2.1 About

* 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.
* Autoencoders are a type of neural network that takes an input (e.g. image, dataset), boils that input down to core features, and reverses the process to recreate the input.
* Its network structure consists of three layers.
* First, the input layer consists of m nodes, the same amount of nodes as the dimension of the data.
* The second (middle and hidden) layer has n nodes, where n is the dimension of the encoding.
* The third layer is the output layer and, just as the input layer, it has m nodes.
* Typically autoencoders are used for dimensionality reduction, so the resulting encoding usually has a smaller dimension than the input data .



2.1.1 Anomaly Detection Using Autoencoders

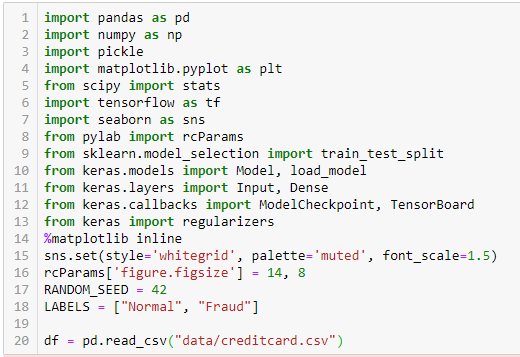
* The autoencoder is trained using a dataset that only contains the non-anomaly points.
* The reconstruction should perform better on normal data than on anomalies since it’s not yet trained with anomaly. So, it now generates output very close to the input.
* We can check whether entered data points are anomaly or not by calculating the error in the reconstructed input used to train the encoder.
* In general, the reconstruction errors in case of anomalies are higher than that in case of the normal data. Thus, an upper bound α (alpha) can be determined for the reconstruction error.
* If reconstruction error[point x(i)] > α => x(i) is anomaly; else it’s not an anomaly.

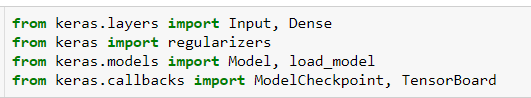
2.2 Algorithm

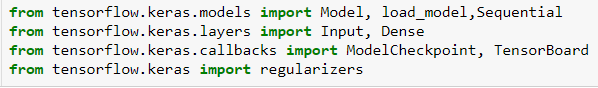
2.2.1 Data

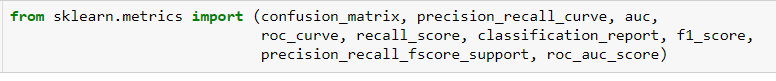
* The dataset has been taken from the data mining website Kaggle.
* It contains transactions of two days. Out of 284,807, fraud transactions are 492.
* Each transaction is either fraudulent or not.
* It features- Time (seconds elapsed between each transaction), Amount (money that was involved in the transaction).
* The data has been transformed using PCA transformation(s) due to privacy reasons.
* The features were normalized and split into a ‘training’ and a ‘test set’.

2.2.2 Import libraries and load the data







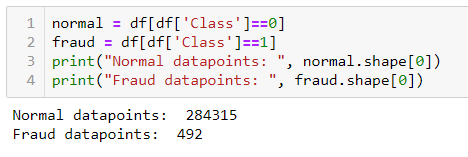


2.2.3 Exploration

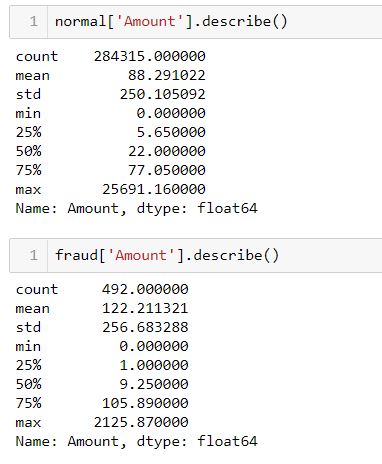
# 

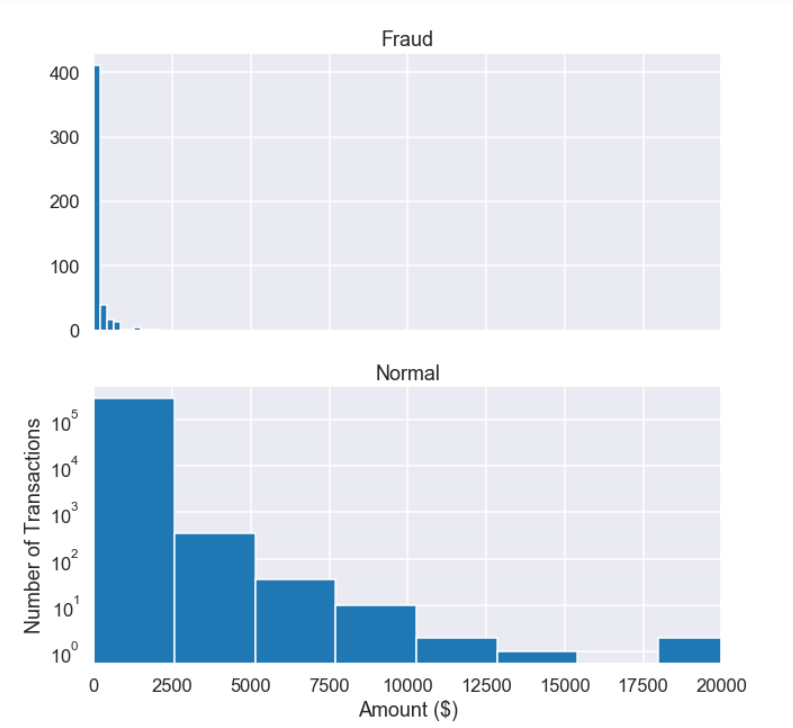
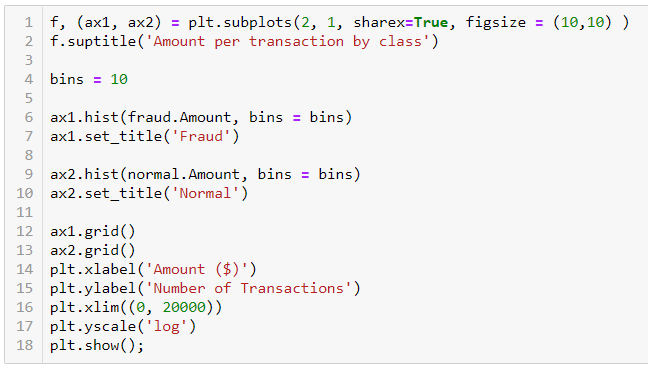


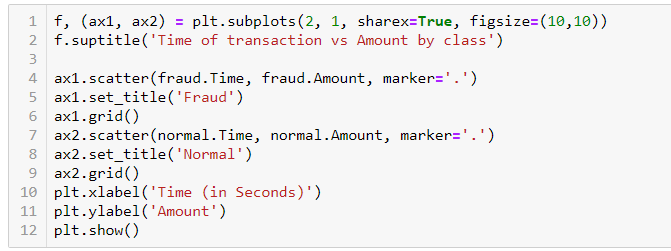
* It’s a highly imbalanced dataset.
* Normal transactions >> Fraudulent Transactions

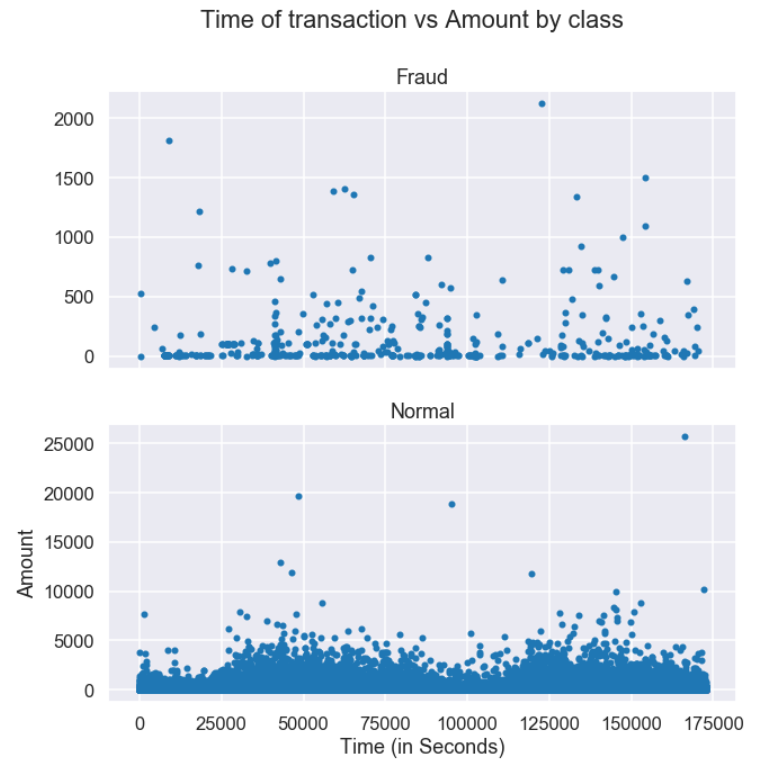


2.2.4 Amount Statistics



2.2.5 Graphical Representation 

\* Check if fraud occurs at certain time only



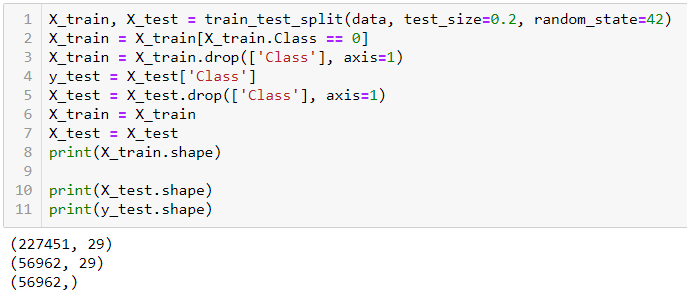
\* It’s evident that there isn’t a fixed time of fraudulent transactions.

* Preparing the data
* Drop the Time column and use the scikit’s StandardScaler on the Amount.
* The scaler removes the mean and scales the values to unit variance.

****

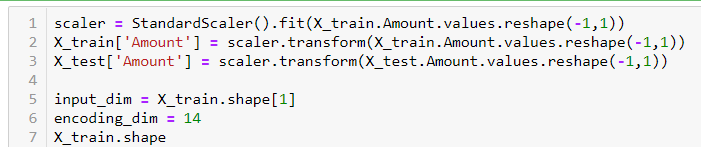
2.2.6 Training the Model

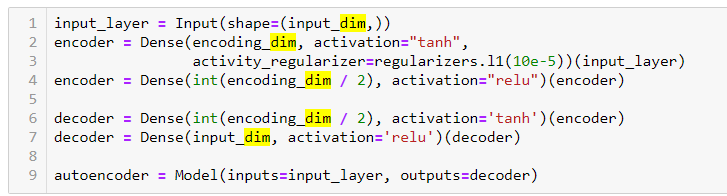
* ⅘ of the total normal transactions were used to train and ⅕ portion of normal transactions plus the fraudulent ones were used while testing.



# .2.2.7 Building the Model

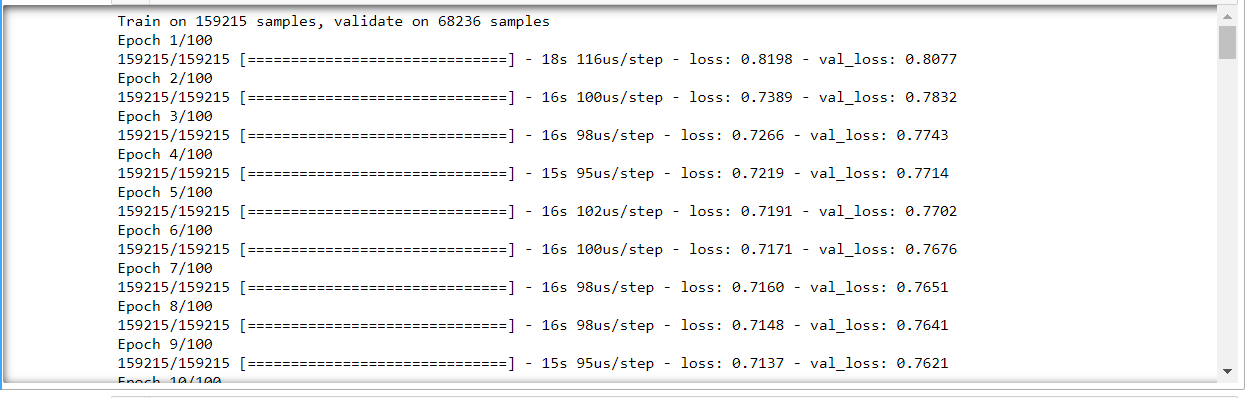
* This Autoencoder uses 4 fully connected layers with 14, 7, 7 and 29 neurons respectively.
* The first two layers are used for our encoder, the last two go for the decoder.
* Additionally, L1 regularization will be used during training.

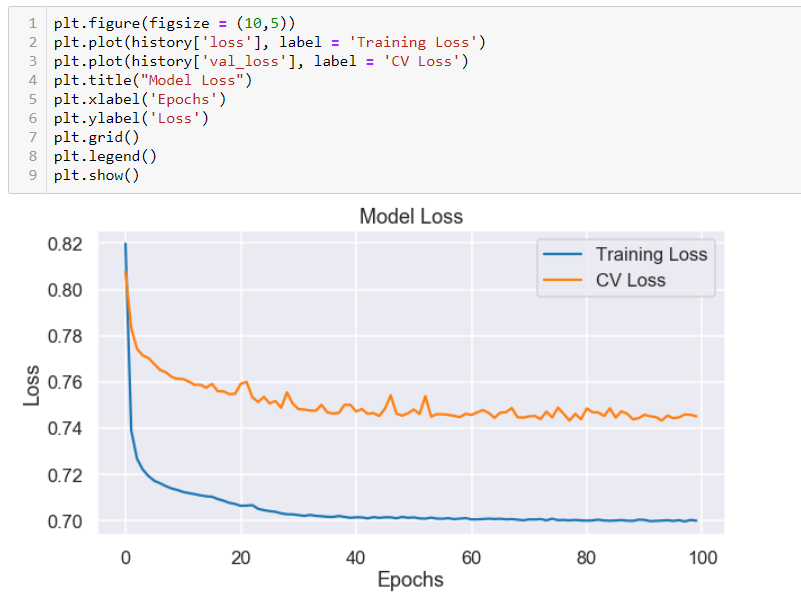




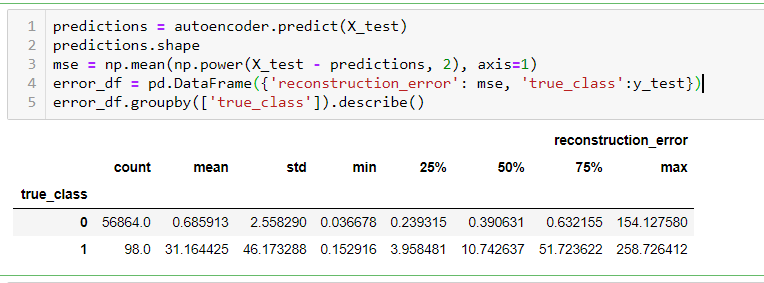
* Training model 100 epochs with a batch size of 32
* Save the best performing model to a file
* ModelCheckpoint provided by Keras is used for the purpose.
* The training progress will be exported in a format that TensorBoard understands.





2.2.8 Reconstruction Error

* The recoreconstruction error on our training and test data seems to converge nicely.
* Following for more details :



# 2.2.9 Reconstruction error without froud

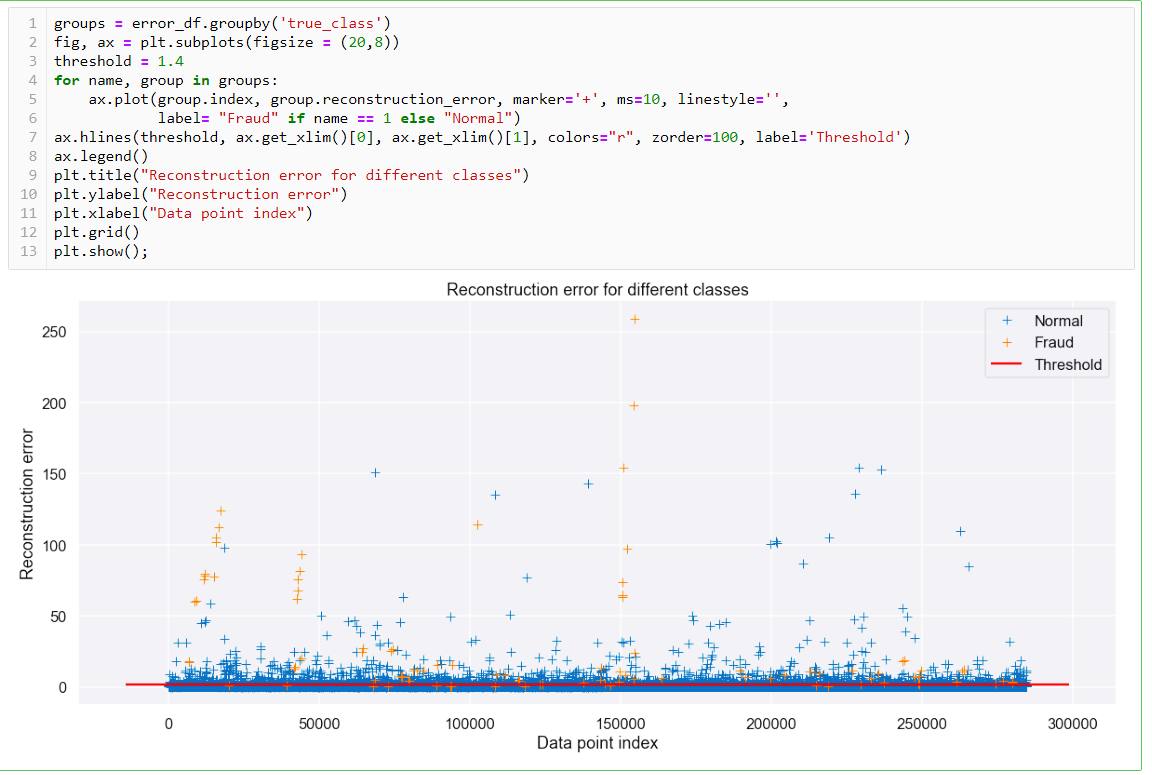


# 2.2.10 Reconstruction error with froud

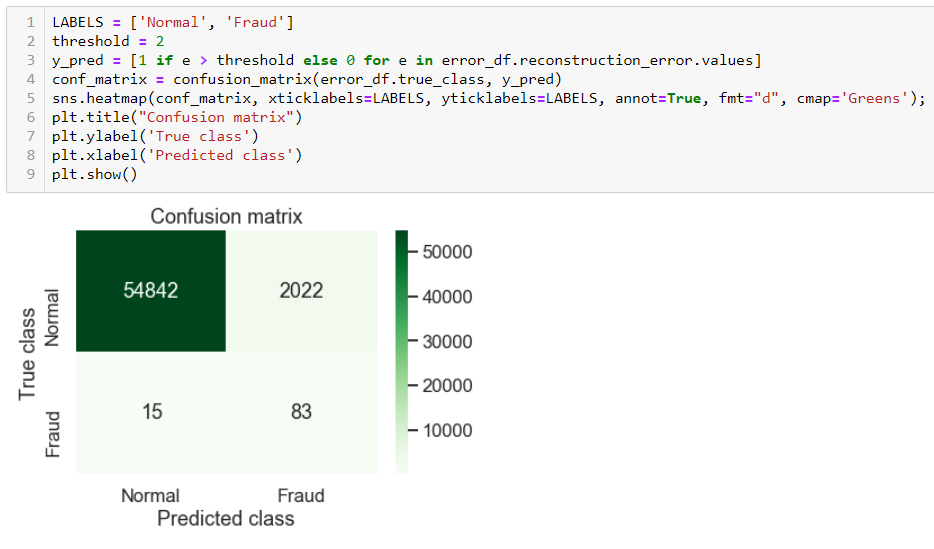
# 

# 2.2.11 Prediction

* In order to predict whether or not a new/unseen transaction is normal or fraudulent, we’ll calculate the reconstruction error from the transaction data itself. If the error is larger than a predefined threshold, we’ll mark it as a fraud (since our model should have a low error on normal transactions)



2.2.12 The Confusion Matrix



* The Model recognizes a lot of fraudulent cases appreciably.
* The threshold value can be changed to see the difference in normal transactions classified as fraudulent.

# 

# 

**3 Conclusion**

* The Autoencoder that is created can reconstruct what non fraudulent transactions look like.
* A huge portion of the taken dataset (80%) was fed to train the model.
* 20% of the dataset was used to test the model.
* To an appreciable extent, it learnt to classify the new examples into normal and fraudulent transactions.

**4 Technology used**

* IDE : jupyter notebook
* Libraries : numpy , pandas , TensorFlow , keras , sklearn, seaborn, matplotlib, scipy, pylab
* Autoencoder

**5 References**

* [Building Autoencoders in Keras](https://blog.keras.io/building-autoencoders-in-keras.html)
* [Stanford tutorial on Autoencoders](http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/)
* [Stacked Autoencoders in TensorFlow](http://cmgreen.io/2016/01/04/tensorflow_deep_autoencoder.html)
* <https://www.youtube.com/watch?v=nTt_ajul8NY&t=154s>
* <https://www.youtube.com/watch?v=1ySn6h2A68I>
* <https://en.wikipedia.org/wiki/Autoencoder>
* <https://www.tensorflow.org/install>
* <https://www.tensorflow.org/learn>