

```
In [1]: import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

C:\Users\Admin\anaconda3\lib\site-packages\scipy\\_\_init\_\_.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.26.1)

```
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")
```

```
In [2]: !pip install tensorflow --user
!pip install keras
!pip install daytime
!pip install torch
```

Requirement already satisfied: tensorflow in c:\users\admin\anaconda3\lib\site-packages (2.14.0)

Requirement already satisfied: tensorflow-intel==2.14.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (2.14.0)

Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (3.3.0)

Requirement already satisfied: numpy>=1.23.5 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (1.26.1)

Requirement already satisfied: tensorboard<2.15,>=2.14 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (2.14.1)

Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (1.59.2)

Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (0.5.4)

Requirement already satisfied: termcolor>=1.1.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (2.3.0)

Requirement already satisfied: setuptools in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (61.2.0)

Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (0.31.0)

Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<5.0.0dev,>=3.20.3 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (4.24.4)

Requirement already satisfied: astunparse>=1.6.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (1.6.3)

Requirement already satisfied: absl-py>=1.0.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (2.0.0)

Requirement already satisfied: libclang>=13.0.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (16.0.6)

Requirement already satisfied: six>=1.12.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (1.16.0)

Requirement already satisfied: google-pasta>=0.1.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (0.2.0)

Requirement already satisfied: flatbuffers>=23.5.26 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (23.5.26)

Requirement already satisfied: tensorflow-estimator<2.15,>=2.14.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (2.14.0)

Requirement already satisfied: keras<2.15,>=2.14.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (2.14.0)

Requirement already satisfied: h5py>=2.9.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (3.6.0)

Requirement already satisfied: ml-dtypes==0.2.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (0.2.0)

Requirement already satisfied: wrapt<1.15,>=1.11.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (1.12.1)

Requirement already satisfied: packaging in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (21.3)

Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (4.1.1)

Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\admin\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-intel==2.14.0->tensorflow) (0.37.1)

Requirement already satisfied: markdown>=2.6.8 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (3.3.4)

Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (0.7.2)

Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (1.0.0)

Requirement already satisfied: requests<3,>=2.21.0 in c:\users\admin\anaconda3\lib\si

```

te-packages (from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (2.2
7.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\admin\anaconda3\lib
\site-packages (from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow)
(2.23.4)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\admin\anaconda3\lib\site-p
ackages (from tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (2.0.3)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\admin\anaconda3\lib\site-pac
kages (from google-auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0-
>tensorflow) (4.7.2)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in c:\users\admin\anaconda3\lib
\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-intel
==2.14.0->tensorflow) (4.2.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\admin\anaconda3\lib
\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-intel
==2.14.0->tensorflow) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\admin\anaconda3\l
ib\site-packages (from google-auth-oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tenso
rflow-intel==2.14.0->tensorflow) (1.3.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\admin\anaconda3\lib\s
ite-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.15,>=
2.14->tensorflow-intel==2.14.0->tensorflow) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\sit
e-packages (from requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-intel==2.1
4.0->tensorflow) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-pack
ages (from requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-intel==2.14.0->te
nsorflow) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\admin\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-intel==
2.14.0->tensorflow) (1.26.9)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\admin\anaconda3
\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-int
el==2.14.0->tensorflow) (2.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\admin\anaconda3\lib\site-p
ackages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<1.1,>=0.5->tensorboard<
2.15,>=2.14->tensorflow-intel==2.14.0->tensorflow) (3.2.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\admin\anaconda3\l
ib\site-packages (from packaging->tensorflow-intel==2.14.0->tensorflow) (3.0.4)
Requirement already satisfied: keras in c:\users\admin\anaconda3\lib\site-packages
(2.14.0)
Collecting daytime
  Downloading daytime-0.4.tar.gz (2.4 kB)
Building wheels for collected packages: daytime
  Building wheel for daytime (setup.py): started
  Building wheel for daytime (setup.py): finished with status 'done'
  Created wheel for daytime: filename=daytime-0.4-py3-none-any.whl size=2419 sha256=8
2ddde36daa2f4ed5f8c4d4ed6a512fe6068aceab3d8a42091e88bc5d1c76d93
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\52\d5\73\e709f43
608f7559446d32db26e22e102b1d7a23d8c06b60bc5
Successfully built daytime
Installing collected packages: daytime
Successfully installed daytime-0.4
Collecting torch
  Downloading torch-2.1.0-cp39-cp39-win_amd64.whl (192.2 MB)
Requirement already satisfied: filelock in c:\users\admin\anaconda3\lib\site-packages
(from torch) (3.6.0)
Requirement already satisfied: networkx in c:\users\admin\anaconda3\lib\site-packages
(from torch) (2.7.1)
Requirement already satisfied: fsspec in c:\users\admin\anaconda3\lib\site-packages

```

```
(from torch) (2022.2.0)
Requirement already satisfied: sympy in c:\users\admin\anaconda3\lib\site-packages (from torch) (1.10.1)
Requirement already satisfied: jinja2 in c:\users\admin\anaconda3\lib\site-packages (from torch) (2.11.3)
Requirement already satisfied: typing-extensions in c:\users\admin\anaconda3\lib\site-packages (from torch) (4.1.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\admin\anaconda3\lib\site-packages (from jinja2->torch) (2.0.1)
Requirement already satisfied: mpmath>=0.19 in c:\users\admin\anaconda3\lib\site-packages (from sympy->torch) (1.2.1)
Installing collected packages: torch
Successfully installed torch-2.1.0
```

```
In [3]: from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, recall_score, accuracy_score, precision_
RANDOM_SEED = 2021
TEST_PCT = 0.3
LABELS = ["Normal", "Fraud"]
```

```
In [4]: #dataset = pd.read_csv("E:\Teachning material\Deep Learning BE IT 2019 course\creditca
dataset = pd.read_csv(r"C:\Users\Admin\Downloads\creditcard.csv")
#dataset.head
print(list(dataset.columns))
dataset.describe()
```

```
['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12',
'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24',
'V25', 'V26', 'V27', 'V28', 'Amount', 'Class']
```

```
Out[4]:
```

	Time	V1	V2	V3	V4	V5	
<b>count</b>	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.8
<b>mean</b>	94813.859575	3.918649e-15	5.682686e-16	-8.761736e-15	2.811118e-15	-1.552103e-15	2.
<b>std</b>	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.3
<b>min</b>	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.6
<b>25%</b>	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.
<b>50%</b>	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.
<b>75%</b>	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.
<b>max</b>	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.3

8 rows × 31 columns

```
In [5]: #check for any null values
print("Any nulls in the dataset ", dataset.isnull().values.any() )
print('-----')
print("No. of unique labels ", len(dataset['Class'].unique()))
print("Label values ", dataset.Class.unique())
#0 is for normal credit card transaction
#1 is for fraudulent credit card transaction
print('-----')
print("Break down of the Normal and Fraud Transactions")
print(pd.value_counts(dataset['Class'], sort = True) )
```

Any nulls in the dataset False

-----

No. of unique labels 2

Label values [0 1]

-----

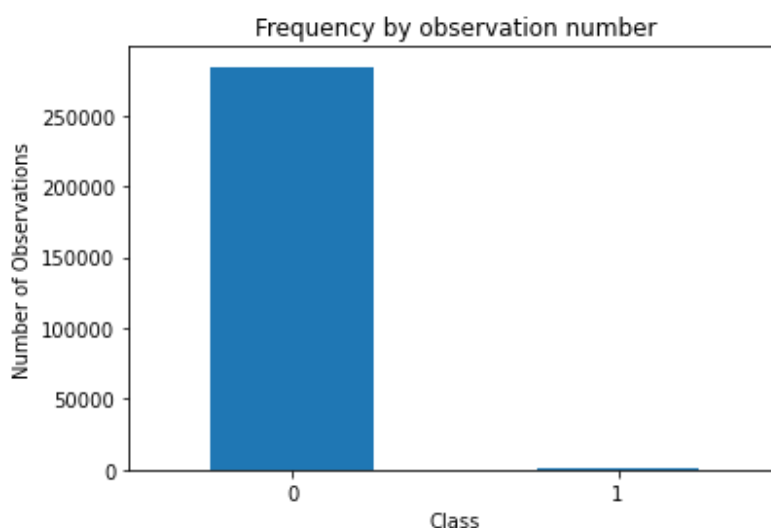
Break down of the Normal and Fraud Transactions

0 284315

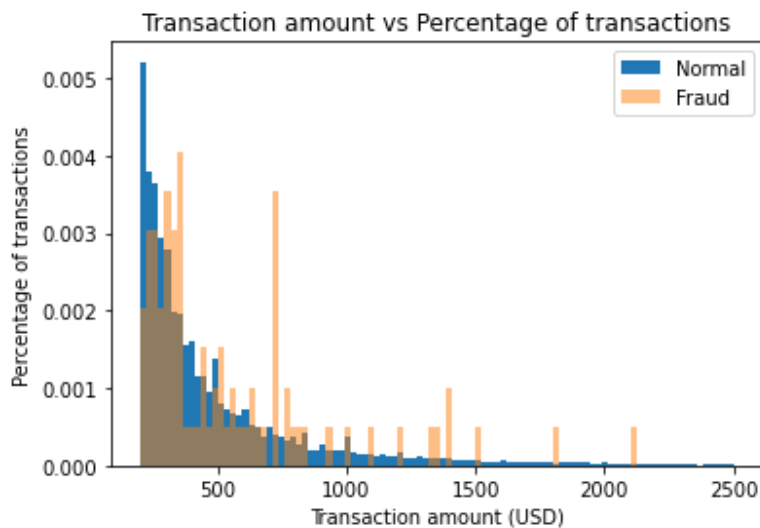
1 492

Name: Class, dtype: int64

```
In [6]: #Visualizing the imbalanced dataset
count_classes = pd.value_counts(dataset['Class'], sort = True)
count_classes.plot(kind = 'bar', rot=0)
plt.xticks(range(len(dataset['Class'].unique()), dataset.Class.unique()))
plt.title("Frequency by observation number")
plt.xlabel("Class")
plt.ylabel("Number of Observations");
```



```
In [7]: # Save the normal and fraudulent transactions in separate dataframe
normal_dataset = dataset[dataset.Class == 0]
fraud_dataset = dataset[dataset.Class == 1]
#Visualize transaction amounts for normal and fraudulent transactions
bins = np.linspace(200, 2500, 100)
plt.hist(normal_dataset.Amount, bins=bins, alpha=1, density=True, label='Normal')
plt.hist(fraud_dataset.Amount, bins=bins, alpha=0.5, density=True, label='Fraud')
plt.legend(loc='upper right')
plt.title("Transaction amount vs Percentage of transactions")
plt.xlabel("Transaction amount (USD)")
plt.ylabel("Percentage of transactions");
plt.show()
```



```
In [8]: '''Time and Amount are the columns that are not scaled, so applying StandardScaler to
Normalizing the values between 0 and 1 did not work great for the dataset.'''
```

```
Out[8]: 'Time and Amount are the columns that are not scaled, so applying StandardScaler to o
nly Amount and Time columns.\nNormalizing the values between 0 and 1 did not work gre
at for the dataset.'
```

```
In [9]: sc=StandardScaler()
dataset['Time'] = sc.fit_transform(dataset['Time'].values.reshape(-1, 1))
dataset['Amount'] = sc.fit_transform(dataset['Amount'].values.reshape(-1, 1))
```

```
In [10]: '''The last column in the dataset is our target variable.'''

raw_data = dataset.values
# The last element contains if the transaction is normal which is represented by a 0 a
labels = raw_data[:, -1]
# The other data points are the electrocardiogram data
data = raw_data[:, 0:-1]
train_data, test_data, train_labels, test_labels = train_test_split(
    data, labels, test_size=0.2, random_state=2021
)
```

```
In [11]: '''Normalize the data to have a value between 0 and 1'''

min_val = tf.reduce_min(train_data)
max_val = tf.reduce_max(train_data)
train_data = (train_data - min_val) / (max_val - min_val)
test_data = (test_data - min_val) / (max_val - min_val)
train_data = tf.cast(train_data, tf.float32)
test_data = tf.cast(test_data, tf.float32)
```

```
In [12]: '''Use only normal transactions to train the Autoencoder.

Normal data has a value of 0 in the target variable. Using the target variable to crea

train_labels = train_labels.astype(bool)
test_labels = test_labels.astype(bool)

#creating normal and fraud datasets

normal_train_data = train_data[~train_labels]
```

```

normal_test_data = test_data[~test_labels]
fraud_train_data = train_data[train_labels]
fraud_test_data = test_data[test_labels]
print(" No. of records in Fraud Train Data=",len(fraud_train_data))
print(" No. of records in Normal Train data=",len(normal_train_data))
print(" No. of records in Fraud Test Data=",len(fraud_test_data))
print(" No. of records in Normal Test data=",len(normal_test_data))

```

```

No. of records in Fraud Train Data= 389
No. of records in Normal Train data= 227456
No. of records in Fraud Test Data= 103
No. of records in Normal Test data= 56859

```

```

In [13]: nb_epoch = 50
batch_size = 64
input_dim = normal_train_data.shape[1] #num of columns, 30
encoding_dim = 14
hidden_dim_1 = int(encoding_dim / 2) #
hidden_dim_2=4
learning_rate = 1e-7

```

```

In [14]: #input Layer
input_layer = tf.keras.layers.Input(shape=(input_dim, ))

#Encoder
encoder = tf.keras.layers.Dense(encoding_dim, activation="tanh",
                                activity_regularizer=tf.keras.regularizers.l2(learning_rate))(
encoder=tf.keras.layers.Dropout(0.2)(encoder)
encoder = tf.keras.layers.Dense(hidden_dim_1, activation='relu')(encoder)
encoder = tf.keras.layers.Dense(hidden_dim_2, activation=tf.nn.leaky_relu)(encoder)

# Decoder
decoder = tf.keras.layers.Dense(hidden_dim_1, activation='relu')(encoder)
decoder=tf.keras.layers.Dropout(0.2)(decoder)
decoder = tf.keras.layers.Dense(encoding_dim, activation='relu')(decoder)
decoder = tf.keras.layers.Dense(input_dim, activation='tanh')(decoder)

#Autoencoder
autoencoder = tf.keras.Model(inputs=input_layer, outputs=decoder)
autoencoder.summary()

```

Model: "model"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, 30)]	0
dense (Dense)	(None, 14)	434
dropout (Dropout)	(None, 14)	0
dense_1 (Dense)	(None, 7)	105
dense_2 (Dense)	(None, 4)	32
dense_3 (Dense)	(None, 7)	35
dropout_1 (Dropout)	(None, 7)	0
dense_4 (Dense)	(None, 14)	112
dense_5 (Dense)	(None, 30)	450
=====		
Total params: 1168 (4.56 KB)		
Trainable params: 1168 (4.56 KB)		
Non-trainable params: 0 (0.00 Byte)		

```
In [15]: """Define the callbacks for checkpoints and early stopping"""

cp = tf.keras.callbacks.ModelCheckpoint(filepath="autoencoder_fraud.h5",
                                         mode='min', monitor='val_loss', verbose=2, save_best_only=True)

# define our early stopping
early_stop = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    min_delta=0.0001,
    patience=10,
    verbose=1,
    mode='min',
    restore_best_weights=True)
```

```
In [16]: #Compile the Autoencoder

autoencoder.compile(metrics=['accuracy'],
                     loss='mean_squared_error',
                     optimizer='adam')
```

```
In [17]: #Train the Autoencoder

history = autoencoder.fit(normal_train_data, normal_train_data,
                          epochs=nb_epoch,
                          batch_size=batch_size,
                          shuffle=True,
                          validation_data=(test_data, test_data),
                          verbose=1,
                          callbacks=[cp, early_stop]
                          ).history
```



Epoch 1/50

3547/3554 [=====>.] - ETA: 0s - loss: 0.0037 - accuracy: 0.0343

Epoch 1: val\_loss improved from inf to 0.00002, saving model to autoencoder\_fraud.h5  
3554/3554 [=====] - 29s 7ms/step - loss: 0.0037 - accuracy: 0.0344 - val\_loss: 2.1230e-05 - val\_accuracy: 0.0010

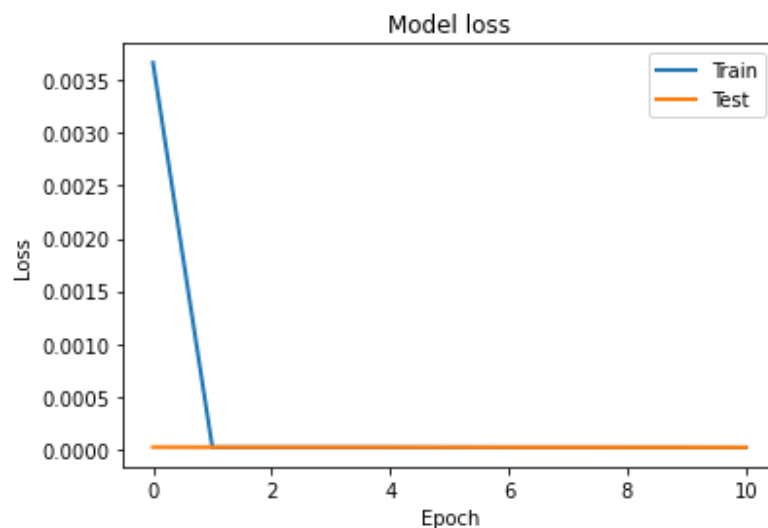
C:\Users\Admin\anaconda3\lib\site-packages\keras\src\engine\training.py:3079: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')`.   
saving\_api.save\_model(

```
Epoch 2/50
3551/3554 [=====>.] - ETA: 0s - loss: 1.9656e-05 - accuracy:
0.0653
Epoch 2: val_loss improved from 0.00002 to 0.00002, saving model to autoencoder_fraud.h5
3554/3554 [=====] - 25s 7ms/step - loss: 1.9653e-05 - accuracy: 0.0653 - val_loss: 1.9941e-05 - val_accuracy: 0.0661
Epoch 3/50
3554/3554 [=====] - ETA: 0s - loss: 1.9448e-05 - accuracy:
0.0628
Epoch 3: val_loss did not improve from 0.00002
3554/3554 [=====] - 24s 7ms/step - loss: 1.9448e-05 - accuracy: 0.0628 - val_loss: 2.0051e-05 - val_accuracy: 0.0051
Epoch 4/50
3550/3554 [=====>.] - ETA: 0s - loss: 1.9569e-05 - accuracy:
0.0607
Epoch 4: val_loss did not improve from 0.00002
3554/3554 [=====] - 24s 7ms/step - loss: 1.9577e-05 - accuracy: 0.0608 - val_loss: 2.0254e-05 - val_accuracy: 0.0343
Epoch 5/50
3553/3554 [=====>.] - ETA: 0s - loss: 1.9518e-05 - accuracy:
0.0632
Epoch 5: val_loss did not improve from 0.00002
3554/3554 [=====] - 23s 7ms/step - loss: 1.9517e-05 - accuracy: 0.0633 - val_loss: 2.0057e-05 - val_accuracy: 0.2168
Epoch 6/50
3549/3554 [=====>.] - ETA: 0s - loss: 1.9107e-05 - accuracy:
0.1521
Epoch 6: val_loss improved from 0.00002 to 0.00002, saving model to autoencoder_fraud.h5
3554/3554 [=====] - 24s 7ms/step - loss: 1.9111e-05 - accuracy: 0.1523 - val_loss: 1.9341e-05 - val_accuracy: 0.1721
Epoch 7/50
3548/3554 [=====>.] - ETA: 0s - loss: 1.8519e-05 - accuracy:
0.2646
Epoch 7: val_loss improved from 0.00002 to 0.00002, saving model to autoencoder_fraud.h5
3554/3554 [=====] - 23s 7ms/step - loss: 1.8518e-05 - accuracy: 0.2646 - val_loss: 1.8773e-05 - val_accuracy: 0.2996
Epoch 8/50
3553/3554 [=====>.] - ETA: 0s - loss: 1.8453e-05 - accuracy:
0.2692
Epoch 8: val_loss improved from 0.00002 to 0.00002, saving model to autoencoder_fraud.h5
3554/3554 [=====] - 24s 7ms/step - loss: 1.8453e-05 - accuracy: 0.2692 - val_loss: 1.8668e-05 - val_accuracy: 0.3177
Epoch 9/50
3551/3554 [=====>.] - ETA: 0s - loss: 1.8409e-05 - accuracy:
0.2728
Epoch 9: val_loss improved from 0.00002 to 0.00002, saving model to autoencoder_fraud.h5
3554/3554 [=====] - 23s 7ms/step - loss: 1.8409e-05 - accuracy: 0.2728 - val_loss: 1.8432e-05 - val_accuracy: 0.3111
Epoch 10/50
3548/3554 [=====>.] - ETA: 0s - loss: 1.8269e-05 - accuracy:
0.2770
Epoch 10: val_loss did not improve from 0.00002
3554/3554 [=====] - 24s 7ms/step - loss: 1.8269e-05 - accuracy: 0.2771 - val_loss: 1.8462e-05 - val_accuracy: 0.2869
Epoch 11/50
```

3536/3554 [=====>.] - ETA: 0s - loss: 1.7849e-05 - accuracy: 0.2786  
 Epoch 11: val\_loss improved from 0.00002 to 0.00002, saving model to autoencoder\_fraud.h5  
 Restoring model weights from the end of the best epoch: 1.  
 3554/3554 [=====] - 20s 6ms/step - loss: 1.7846e-05 - accuracy: 0.2786 - val\_loss: 1.7778e-05 - val\_accuracy: 0.3142  
 Epoch 11: early stopping

In [18]: *#Plot training and test loss*

```
plt.plot(history['loss'], linewidth=2, label='Train')
plt.plot(history['val_loss'], linewidth=2, label='Test')
plt.legend(loc='upper right')
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
#plt.ylim(ymin=0.70,ymax=1)
plt.show()
```



In [19]: *"""Detect Anomalies on test data*

*Anomalies are data points where the reconstruction loss is higher*

*To calculate the reconstruction loss on test data,  
 predict the test data and calculate the mean square error between the test data and th*

```
test_x_predictions = autoencoder.predict(test_data)
mse = np.mean(np.power(test_data - test_x_predictions, 2), axis=1)
error_df = pd.DataFrame({'Reconstruction_error': mse,
                        'True_class': test_labels})
```

1781/1781 [=====] - 3s 2ms/step

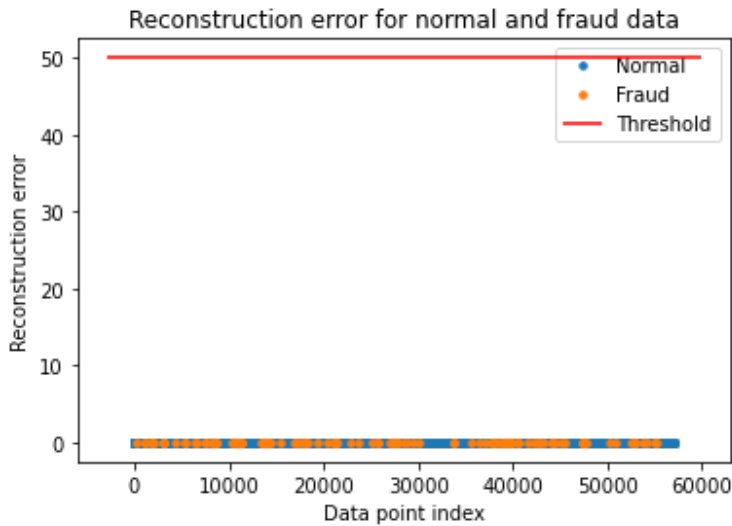
In [20]: *#Plotting the test data points and their respective reconstruction error sets a thresh  
 #if the threshold value needs to be adjusted.*

```
threshold_fixed = 50
groups = error_df.groupby('True_class')
fig, ax = plt.subplots()
for name, group in groups:
    ax.plot(group.index, group.Reconstruction_error, marker='o', ms=3.5, linestyle='',
```

```

label= "Fraud" if name == 1 else "Normal")
ax.hlines(threshold_fixed, ax.get_xlim()[0], ax.get_xlim()[1], colors="r", zorder=100,
ax.legend()
plt.title("Reconstruction error for normal and fraud data")
plt.ylabel("Reconstruction error")
plt.xlabel("Data point index")
plt.show();

```



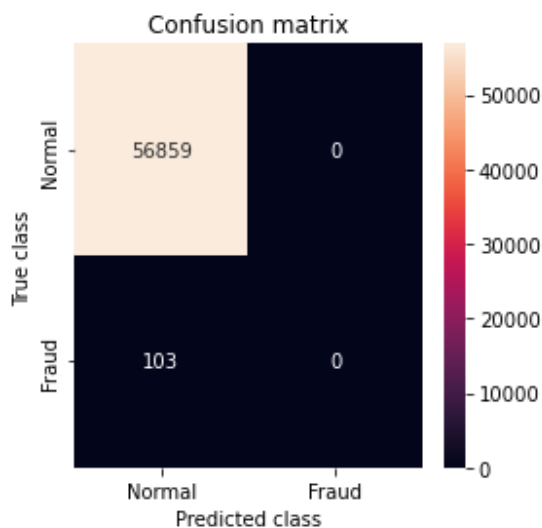
In [21]: '''Detect anomalies as points where the reconstruction loss is greater than a fixed threshold. Here we see that a value of 52 for the threshold will be good.

Evaluating the performance of the anomaly detection'''

```

threshold_fixed = 52
pred_y = [1 if e > threshold_fixed else 0 for e in error_df.Reconstruction_error.values]
error_df['pred'] = pred_y
conf_matrix = confusion_matrix(error_df.True_class, pred_y)
plt.figure(figsize=(4, 4))
sns.heatmap(conf_matrix, xticklabels=LABELS, yticklabels=LABELS, annot=True, fmt="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
# print Accuracy, precision and recall
print(" Accuracy: ", accuracy_score(error_df['True_class'], error_df['pred']))
print(" Recall: ", recall_score(error_df['True_class'], error_df['pred']))
print(" Precision: ", precision_score(error_df['True_class'], error_df['pred']))

```



Accuracy: 0.9981917769741231

Recall: 0.0

Precision: 0.0

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\metrics\\_classification.py:1318: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero\_division` parameter to control this behavior.  
\_warn\_prf(average, modifier, msg\_start, len(result))

In [22]: '''As our dataset is highly imbalanced, we see a high accuracy but a low recall and pr  
Things to further improve precision and recall would add more relevant features,  
different architecture for autoencoder, different hyperparameters, or a different algo

Out[22]: 'As our dataset is highly imbalanced, we see a high accuracy but a low recall and pre  
cision.\n\nThings to further improve precision and recall would add more relevant fea  
tures,\ndifferent architecture for autoencoder, different hyperparameters, or a diffe  
rent algorithm.'

In [23]: history

```
Out[23]: {'loss': [0.0036670698318630457,
 1.9652592527563684e-05,
 1.9448039893177338e-05,
 1.957662243512459e-05,
 1.9516715838108212e-05,
 1.9110788343823515e-05,
 1.8518483557272702e-05,
 1.845291262725368e-05,
 1.8408898540656082e-05,
 1.8269494830747135e-05,
 1.7845832189777866e-05],
'accuracy': [0.034411050379276276,
 0.06530053913593292,
 0.06278137117624283,
 0.06083374470472336,
 0.063251793384552,
 0.15227560698986053,
 0.26456984877586365,
 0.269155353307724,
 0.27278682589530945,
 0.27706897258758545,
 0.2785945534706116],
'val_loss': [2.1229858248261735e-05,
 1.994073136302177e-05,
 2.005093301704619e-05,
 2.0253588445484638e-05,
 2.0056804714840837e-05,
 1.934071769937873e-05,
 1.8772629118757322e-05,
 1.8667991753318347e-05,
 1.843235622800421e-05,
 1.846158193075098e-05,
 1.777845318429172e-05],
'val_accuracy': [0.0010006671072915196,
 0.06607913970947266,
 0.005126224365085363,
 0.03430357202887535,
 0.2168463170528412,
 0.17211474478244781,
 0.2996208071708679,
 0.31765037775039673,
 0.3111197054386139,
 0.286875456571579,
 0.31415680050849915]}
```

In [ ]: