CREDIT CARD FRAUD DETECTION USING AUTOENCODERS

END TERM REPORT

by

Dipankar Lama, Madan singh, Kashish Gilhotra, Varun Mishra

Section : K18JF

Roll No(s): 14, 16, 20, 33

Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

April-2020

STUDENT DECLARATION

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

Dipankar Lama
Roll No. : 14

Madan Singh
Roll No. : 16

Kashish Gilhotra
Roll No. : 20

Varun Mishra
Roll No. : 33

Place : Lovely Professional University
Date : 13-April-2020

Table of Content :

S.no	Title	Page No
1.	**BackGround and Objective of Project Assigned …….** 1. About the Fraud ……... 2. Anomaly Detection ……..... 3. Application of Anomaly Detection …..... 4. Machine Learning Approach for Anomaly Detection ….	5 5 6 7 7
2.	**Description …….** 1. Setting up H2O server ……. 2. H2O Module…… 3. Matplotlib Module……. 4. Pylab Module…….. 5. Numpy Module…….. 6. Pandas Module…….. 7. OS Module……. 8. Flow Chart……..	8 8 8 9 10 10 11 11 12
3.	**Roles and Responsibilities ……..** 1. Dipankar Lama ……. 2. Madan Purohit …….. 3. Kashish Gilhotra……. 4. Varun Misra …….	13-14 13 13 13 14
4	**Technology used ……..** 1. H2O python Module …….... 2. What is H2O ……. 3. H2O object System …… 4. Seaborn ……..	14 14 15 16 16-17
5	**SWAT analysis ……..** 1. The Strength of Project…….. 2. Weakness of Project…….. 3. Opportunity of Project……...	18-20 19 19-20 20-21
6	**Screenshots ……...**	21-29

BONAFIDE CERTIFICATE

Certified that this project report "**Credit Card Fraud Detection Using AutoEncoders**" is the bonafide work of Dipankar Lama, Madan Singh, Kashish Gilhotra, Varun Mishra who carried out the project work under my supervision.

BACKGROUND AND OBJECTIVES OF PROJECT ASSIGNED

Fraud is one of the major ethical issues in the credit card industry. The main aims are, firstly, to identify the different types of credit card fraud, and, secondly, to review alternative techniques that have been used in fraud detection. The sub-aim is to present, compare and analyze recently published findings in credit card fraud detection. **Credit card fraud** is an inclusive term for fraud committed using a payment card, such as a credit card or debit card. The purpose may be to obtain goods or services, or to make payment to another account which is controlled by a criminal. The Payment Card Industry Data Security Standard (PCI DSS) is the data security standard created to help businesses process card payments securely and reduce card fraud.

Credit card fraud can be authorised, where the genuine customer themselves processes a payment to another account which is controlled by a criminal, or unauthorised, where the account holder does not provide authorisation for the payment to proceed and the transaction is carried out by a third party. In 2018, unauthorised financial fraud losses across payment cards and remote banking totalled £844.8 million in the United Kingdom. Whereas banks and card companies prevented £1.66 billion in unauthorised fraud in 2018. That is the equivalent to £2 in every £3 of attempted fraud being stopped.

Credit cards are more secure than ever, with regulators, card providers and banks taking considerable time and effort to collaborate with investigators worldwide to ensure fraudsters aren't successful. Cardholders' money is usually protected from scammers with regulations that make the card provider and bank accountable. The technology and security measures behind credit cards are becoming increasingly sophisticated making it harder for fraudsters to steal money.

How Can Payment Card Fraud Occur : There are two kinds of card fraud: card-present fraud (not so common nowadays) and card-not-present fraud (more common). The compromise can occur in a number of ways and can usually occur without the knowledge of the cardholder. The internet

has made database security lapses particularly costly, in some cases, millions of accounts have been compromised.

Stolen cards can be reported quickly by cardholders, but a compromised account's details may be held by a fraudster for months before any theft, making it difficult to identify the source of the compromise. The cardholder may not discover fraudulent use until receiving a statement. Cardholders can mitigate this fraud risk by checking their account frequently to ensure there are not any suspicious or unknown transactions.

When a credit card is lost or stolen, it may be used for illegal purchases until the holder notifies the issuing bank and the bank puts a block on the account. Most banks have free 24-hour telephone numbers to encourage prompt reporting. Still, it is possible for a thief to make unauthorized purchases on a card before the card is canceled.

Frauds in the finance field are very rare to be identified. Because of that, it can do severe damage to the financial field. It is estimated that fraud costs at least $80 billion a year across all lines of insurance. If there is a small possibility of detecting fraudulent activities, that can do a major impact on annual losses. That is why financial companies invest in machine learning as a preemptive approach to tackling fraud.

The benefits of using a machine learning approach are that,

- It helps to find hidden and implicit correlations in data.
- Faster data processing and less manual work
- Automatic detection of possible fraud scenarios.

The best way to detect frauds is anomaly detection.

Anomaly Detection :

In data mining, **anomaly detection** (also **outlier detection**) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.

In particular, in the context of abuse and network intrusion detection, the interesting objects are often not *rare* objects, but unexpected *bursts* in activity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular unsupervised methods) will fail on such data, unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro clusters formed by these patterns.

Three broad categories of anomaly detection techniques exist. **Unsupervised anomaly detection** techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal by looking for instances that seem to fit least to the remainder of the data set. **Supervised anomaly detection** techniques require a data set that has been labeled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherent unbalanced nature of outlier detection). **Semi-supervised anomaly detection** techniques construct a model representing normal behavior from a given *normal* training data set, and then test the likelihood of a test instance to be generated by the learnt model.

Applications of Anomaly Detection :

Anomaly detection is applicable in a variety of domains, such as intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, and detecting ecosystem disturbances. It is often used in preprocessing to remove anomalous data from the dataset. In supervised learning, removing the anomalous data from the dataset often results in a statistically significant increase in accuracy.

Machine learning approaches for Anomaly detection :

- K-Nearest Neighbor
- Autoencoders — Deep neural network
- K-means
- Support Vector Machine
- Naive Bayes

DESCRIPTION

Setup:

We have used H2O as the Machine Learning Platform. H2O is the open source leader in AI and Machine Learning.

Modules used:

1. H2O
2. Matplotlib
3. Pylab
4. Numpy
5. Pandas
6. os

H2O:

This Python module provides access to this H2O JVM, as well as its extensions, object, machine learning algorithm, and modelling support capabilities, such as basic munging and feature generation.

H2O JVM provides a web server so that all communications occur in a socket (specified by an IP address and a port) via a series of REST calls.

There is a single active connection to the H2O JVM at any time, and this handle is stashed out of sight in a singleton instance of H2O connection. In other words, this package does not rely on jython, and there is no direct manipulation of the JVM.

The H2O python module is not intended as a replacement for other popular machine learning framework such as scikit-learn, pylearn2, and their ilk, but is intended to bring H2oO to a wider audience of data and machine learning devotees who work exclusively with Python.

H2O from python is a tool for rapidly turning over models, doing data managing, and building application and building application in a fast, scalable environment without any of the mental anguish about parallelism and distribution of work.

Code Snippet:

```
import h2o
from h2o.estimators.deeplearning import H2OAutoEncoderEstimator
from h2o.estimators.deeplearning improt H2ODeepLearningEstimator
```

Matplotlib:

Matplotlib is a plotting library for the python programming language and its numerical mathematics extension NumPy. It Provides an OO API for embedding plot into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state_machine like OpenGL, designed to closely resemble that of MATLAB, though its use is discouraged.

Code Snippet:

```
import matplotlib.pyplot as plt
```

Pylab:

PyLab is a procedural interface to the Matplotlib object-oriented plotting library. Matplotlib is the whole package; matplotlib.pyplot is a module in Matplotlib; and PyLab is a module that gets installed alongside Matplotlib.

PyLab is a convenience module that imports matplotlib.pyplot (for plotting) and NumPy (for Mathematics and working with arrays) in a single name space. Although many examples use PyLab, it is no longer recommended.

Code Snippet:

```
from pylab import rcParams
```

Numpy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Code Snippet:

```
import numpy as np
```

Pandas:

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

Code Snippet:

```
import pandas as pd
```

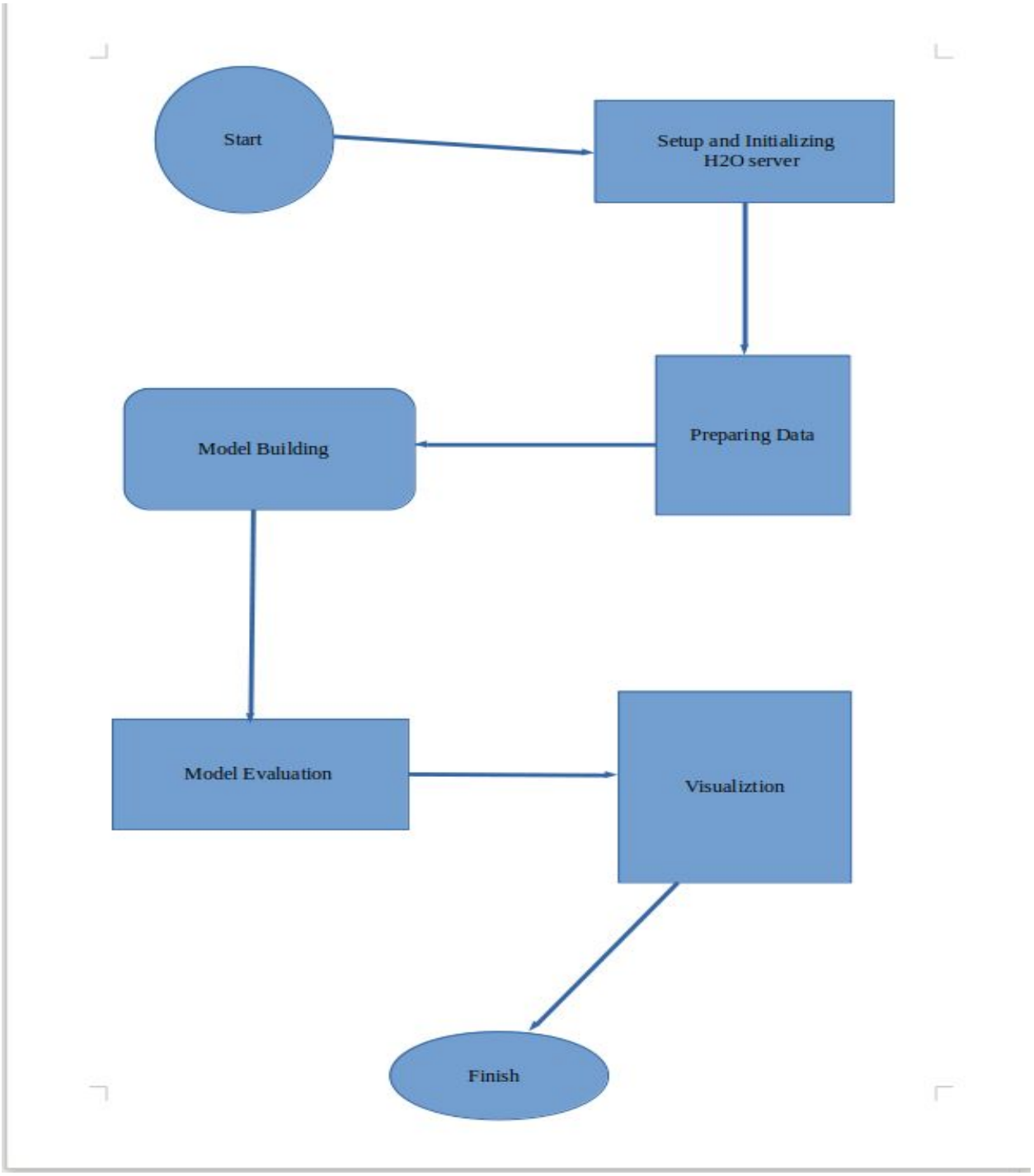
OS:

The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

Code Snippet:

```
import os
```

Flow Chart:


Start
Setup and Initializing H2O server
Preparing Data
Model Building
Model Evaluation
Visualiztion
Finish

ROLES AND RESPONSIBILITIES

1. **Dipankar Lama**

Dipankar writes the code for setup and Exploration of the program. In which he is using H2O as the ML platform, Initialize H2O server using "h20.init()" function. In this initialization, loading of data set using pandas data frame is done by the "pd.read_csv()" function. In Exploration, there is checking for null values in the dataset using "creditData.isnull()" function. In order to process this function it needs to convert the pandas data frame to an H2O data frame. He also contributed to the making of the final report.

2. **Madan Purohit**

Madan writes the code for Preparing data, model building and model evaluation. In preparing data, the time variable is not giving an impact on the model prediction. This can be figured out from data visualization. Before moving on to the training part, it needs to figure out which variables are important and which are not. So it can drop the unwanted variables.

In model building, 4 fully connected hidden layers were chosen with, [14,7,7,14] number of nodes for each layer. First two for the **encoder** and last two for the **decoder**.

In Model Evaluation, there is a special way of analyzing which variables are giving higher impact on the model.

3. **Kashish Gilhotra**

Kashish writes the code for ROC Curve and precision & recall. ROC means Receiver Operating Characteristic. In this there is a curve between true positive rate and false positive rate and the accuracy is 0.9718. In Precision & Recall, Since the data is highly imbalanced, it cannot be measured only by using accuracy. Precision vs Recall was chosen as the matrix for the classification task.

He also contributed to the making of the final report.

4. **Varun Mishra**

Varun writes the code for Confusion Matrix and Classification Report. In Confusion matrix, there is use of seaborn and also uses confusion_matrix() function, heatmap() function, ylabel() and xlabel() function for making confusion matrix, in this there is two class, first is true class and second is predicted class.

In the Classification report there is a report which shows the value of precision, recall, f1-score and support. For making this classification report, it uses classification_report() function.

He also contributed to the making of the final report.

TECHNOLOGY USED

The H2O Python Module :

This Python module provides access to the H2O JVM, as well as its extensions, objects, machine-learning algorithms, and modeling support capabilities, such as basic munging and feature generation.

The H2O JVM provides a web server so that all communication occurs on a socket (specified by an IP address and a port) via a series of REST calls (see connection.py for the REST layer implementation and details).

There is a single active connection to the H2O JVM at any time, and this handle is stashed out of sight in a singleton instance of H2OConnection. In other words, this package does not rely on Jython, and there is no direct manipulation of the JVM.

The H2O Python module is not intended as a replacement for other popular machine learning frameworks such as scikit-learn, pylearn2, and their ilk, but is intended to bring H2O to a wider audience of data and machine learning devotees who work exclusively with Python.

H2O from Python is a tool for rapidly turning over models, doing data munging, and building applications in a fast, scalable environment without any of the mental anguish about parallelism and distribution of work.

What is H2O :

H2O is a Java-based software for data modeling and general computing. The H2O software is many things, but the primary purpose of H2O is as a distributed (many machines), parallel (many CPUs), in memory (several hundred GBs Xmx) processing engine.

There are two levels of parallelism:

- ❏ within node
- ❏ across (or between) nodes

The goal of H2O is to allow simple horizontal scaling to a given problem in order to produce a solution faster. The conceptual paradigm MapReduce (AKA "divide and conquer and combine"), along with a good concurrent application structure, (c.f. jsr166y and NonBlockingHashMap) enable this type of scaling in H2O.

For application developers and data scientists, the gritty details of thread-safety, algorithm parallelism, and node coherence on a network are concealed by simple-to-use REST calls that are all documented here. In addition, H2O is an open-source project under the Apache v2 licence. All of the source code is on github.

For questions, there is an active google group mailing list, or questions can be posted on the H2O community site on Stack Overflow. Our JIRA ticketing system is also open for public use.

Last, but not least, we regularly engage the machine learning community all over the nation with a very busy meetup schedule (so if you're not in The Valley, no sweat, we're probably coming to your area soon!), and finally, we host our very own H2O World conference.

The rest of this document explains a few of the client-server details and the general programming model for interacting with H2O from Python.

The H2O Object System :

H2O uses a distributed key-value store (the "DKV") that contains pointers to the various objects of the H2O ecosystem. Some shared objects are mutable by the client; some shared objects are read-only by the client, but are mutable by H2O (e.g. a model being constructed will change over time); and actions by the client may have side-effects on other clients (multi-tenancy is not a supported model of use, but it is possible for multiple clients to attach to a single H2O cluster).

Briefly, these objects are:

Key: A key is an entry in the DKV that maps to an object in H2O.

Frame: A Frame is a collection of Vec objects. It is a 2D array of elements.

Vec: A Vec is a collection of Chunk objects. It is a 1D array of elements.

Chunk: A Chunk holds a fraction of the BigData. It is a 1D array of elements.

ModelMetrics: A collection of metrics for a given category of model.

Model: A model is an immutable object having predicted and metrics methods.

Job: A Job is a non-blocking task that performs a finite amount of work.

Many of these objects have no meaning to a Python end-user, but to make sense of the objects available in this module it is helpful to understand how these objects map to objects in the JVM. After all, this module is an interface that allows the manipulation.

Seaborn :

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.Seaborn is a library for

making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

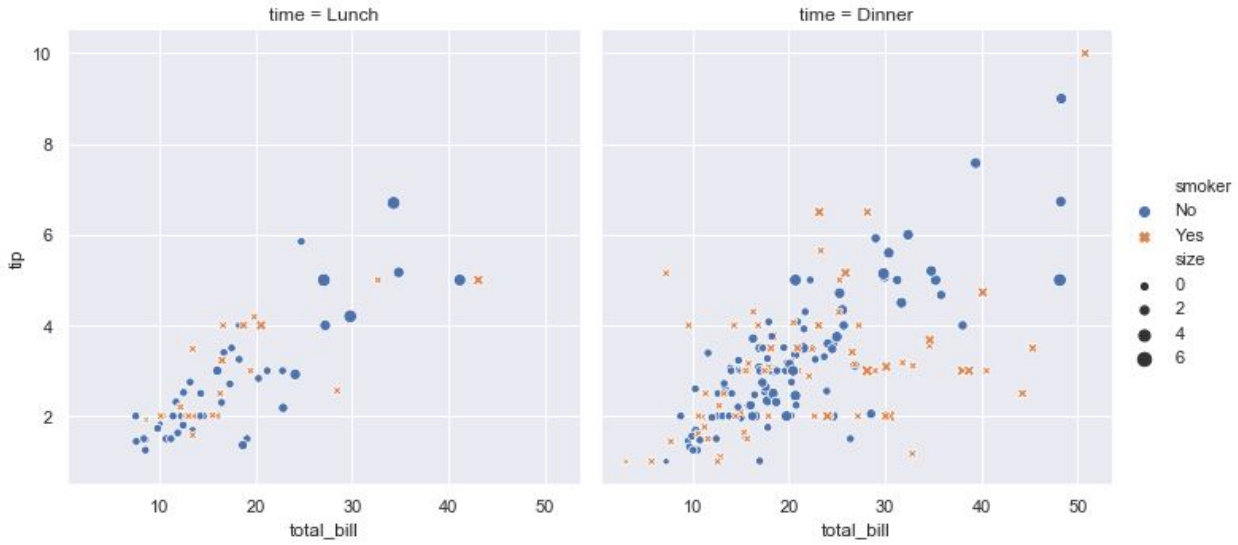
- A dataset-oriented API for examining relationships between multiple variables
- Specialized support for using categorical variables to show observations or aggregate statistics
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
- Automatic estimation and plotting of linear regression models for different kinds dependent variables
- Convenient views onto the overall structure of complex datasets
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing color palettes that faithfully reveal patterns in your data

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

Here's an example of what this means:

```
import seaborn as sns

sns.set()

tips = sns.load_dataset("tips")

sns.relplot(x="total_bill", y="tip", col="time",

        hue="smoker", style="smoker", size="size",
```

data=tips);



SWOT ANALYSIS

SWOT is an acronym of Strengths, Weaknesses, Opportunities and Threats and as these titles suggest it is not purely a method used for controlling areas of planning and risk, but it is also used to highlight areas of the project that could be maximised to the benefit of the whole project or individual areas where some competitive advantage may be gained. It is used to evaluate particular activities of the project in order to optimise their potential as well as to evaluate risks in order to determine the most appropriate way of mitigating those risks.

SWOT analysis is normally performed during the initial project start-up phase so that the elements of the analysis can form the basis of the project plan, but it can also be used later in the project if the project is running into difficulties with scheduling, deliverables or budget and needs to be brought back on track.

For example, if a certain key activity in the project requires new software, a SWOT Analysis can be used to assess the risks and the opportunities of purchasing the software and training staff in its use in order to help with the resource planning.

PERFORMING SWOT ANALYSIS :

The Strengths Of Project :

Increase workplace productivity : Rather than spending hours of manpower on menial, repeatable tasks, employees can configure artificial intelligence to manage it instead. Although we've already used machines on the production lines before, AI allows us to manage a multitude of tasks more efficiently than before.
This is beneficial for all companies. By having technology manage everyday tasks (rather than humans) companies save money. It lowers operations costs and even noncompliance fees. It is Almost Impossible to Make any Prediction regarding Frauds by Humans on the Data set, But it's very Easily and Compatible with this Project

Adopted into many industries : This Technology is adopted by many Credit and Debit Card Companies. AI is now used in a variety of industries, ranging from digital marketing to healthcare. The type and sophistication of the AI needed depend on the task — you'll need less power to automate emails than sorting through a registry of patient information, for example. It's not just for sorting information either; we're also seeing AI used in facial recognition and academic research too.

The Weaknesses Of Project :

The chance to outsmart us : Developers are always pushing to redefine the limits of AI. Right now, it's able to complete a task, learn, and retain information. But maybe, in the future, it'll get

to the point of improving and redesigning without human input. It's this potential reality that makes people remember the robotic overthrowing in the movie *I, Robot*.

The Chance to Give Wrong Outputs : As technology is increasing day by day which leads to formation of new updated modules in that case we have to always change them according to that module which leads to very very much hard work.

Governments are slow on the uptake : Technological experts, like Elon Musk, have warned against artificial intelligence, believing that we need to be smart about how we use it. And how do we use it? This prompts the question of ethics. Is there a line for the ethical use of AI? Bills, regulations, and laws aren't keeping up with the rapid development of technology. Even Congress doesn't fully understand how the internet works, so what hope is there for the ethical use of AI?

Opportunities of Project:

Combining AI with newer forms of tech : Artificial intelligence is connected to other new forms of technology, including machine learning, deep learning, and the Internet of Things (IoT). It'll likely be adopted into programming, enabling developers to reverse problem solve. This allows for enhanced responses to problems, which may benefit other industries, like customer service.

Less strain on employees : And as I said in the strengths section of this SWOT analysis, AI allows us to automate boring, trivial tasks. This is perfect for people who dread taking care of these tasks and would rather focus on the "big picture". Entrepreneurs or startups who have employees wearing a lot of hats and are stretched thin will love AI for this.

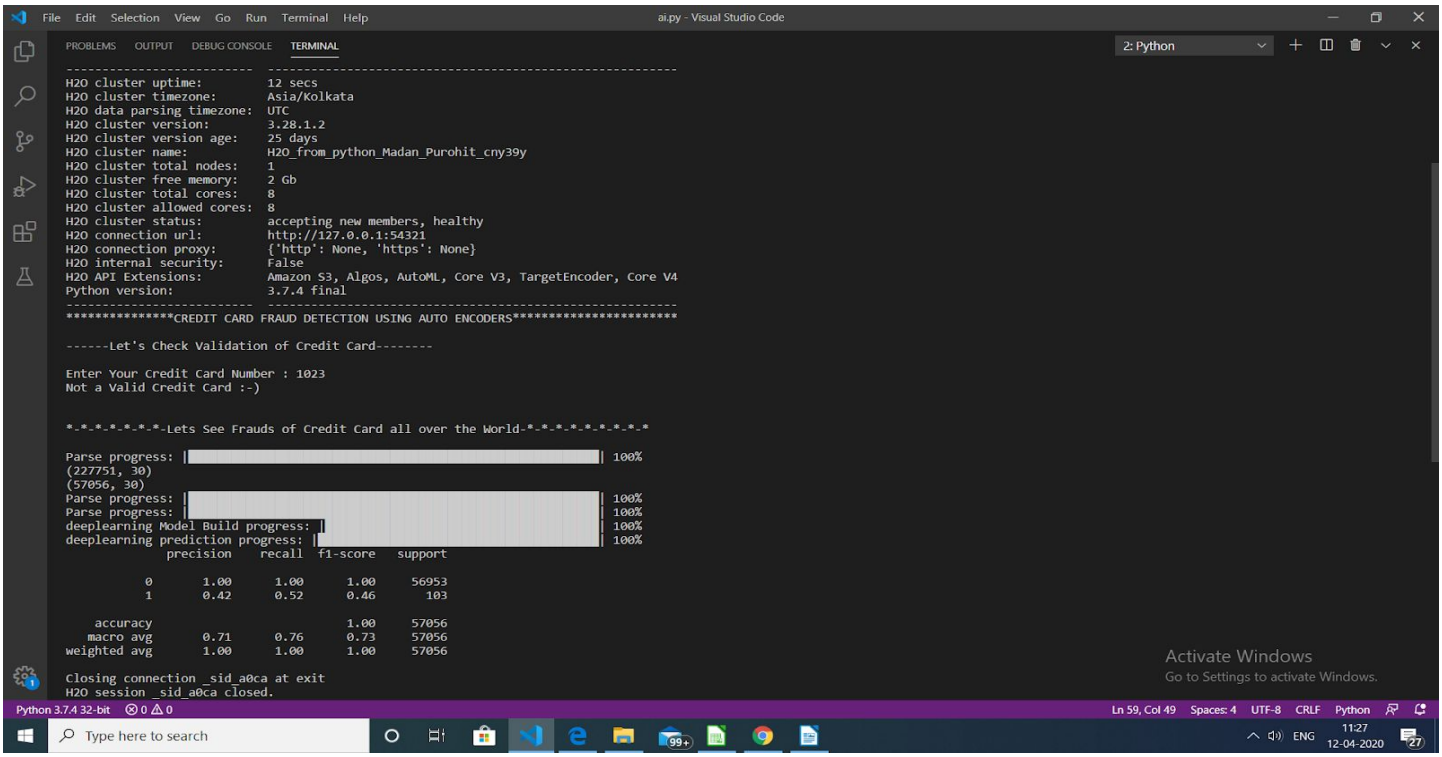
The threats of artificial intelligence :

Job stealing : People believe the adoption of artificial intelligence will lead to job loss. And honestly, this is happening at a small scale. Think about those self-checkouts at Walmart.

There’s several of them and only one or two employees stepping in whenever a customer has a problem.

No more humans working the cashier is a viable future for corporations. This is one example of AI taking over simple human tasks, but also taking away job opportunities. To combat this, the job market will need to evolve. Rather than being replaced, humans will need to work alongside AI. Whether this is a viable future is yet to be determined.

PROJECT SCREENSHOTS :



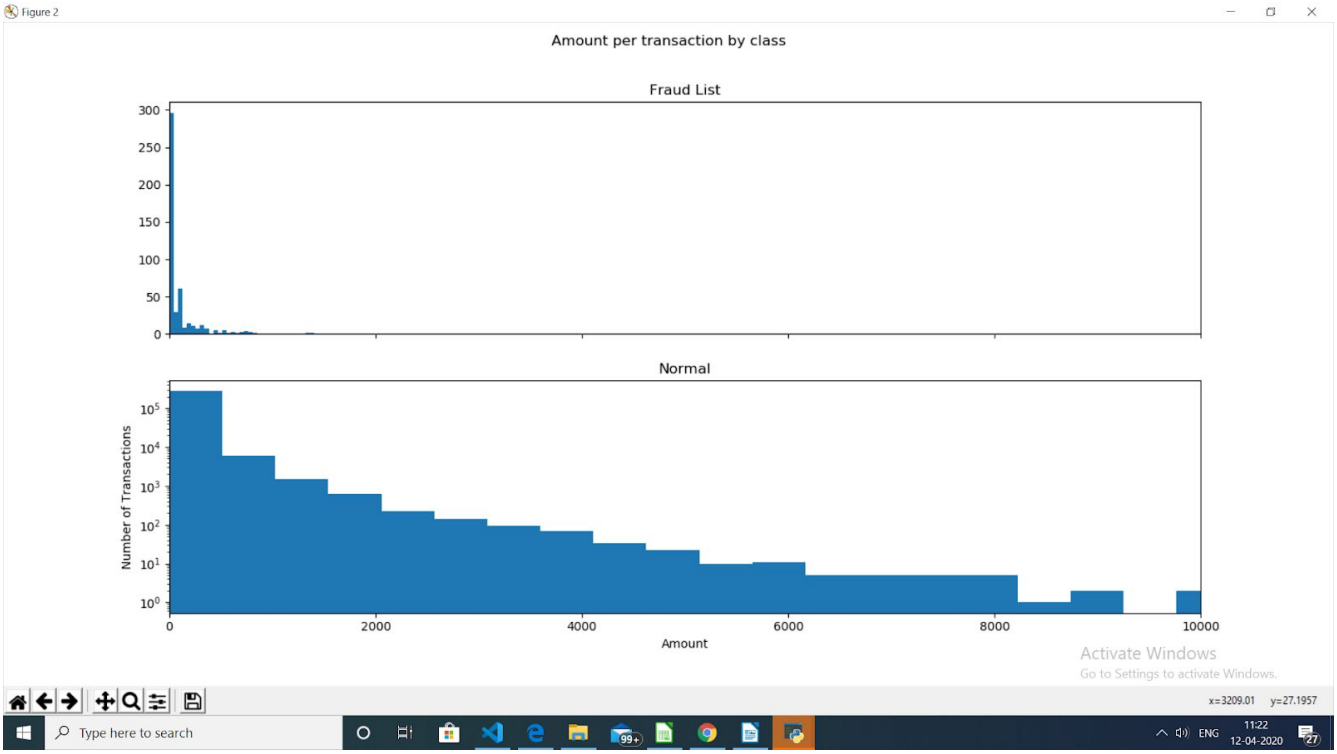


Figure 2
Amount per transaction by class
Fraud List
300
250
200
150
100
50
0
Normal
Number of Transactions
10^5
10^4
10^3
10^2
10^1
10^0
0
2000
4000
6000
8000
10000
Amount
Activate Windows
Go to Settings to activate Windows.
x=3209.01 y=27.1957
Type here to search
ENG
11:22
12-04-2020

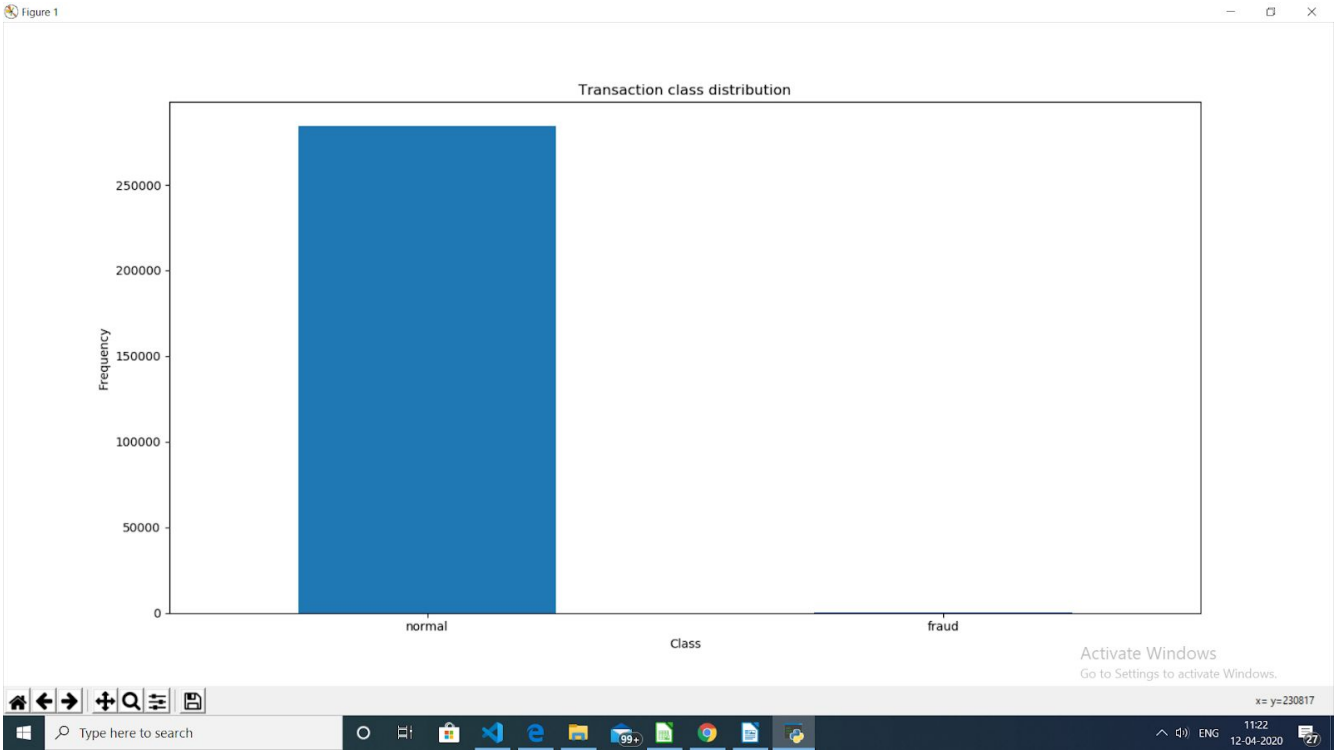


Figure 1
Transaction class distribution
250000
200000
150000
100000
50000
0
Frequency
normal
fraud
Class
Activate Windows
Go to Settings to activate Windows.
x= y=230817
Type here to search
ENG
11:22
12-04-2020

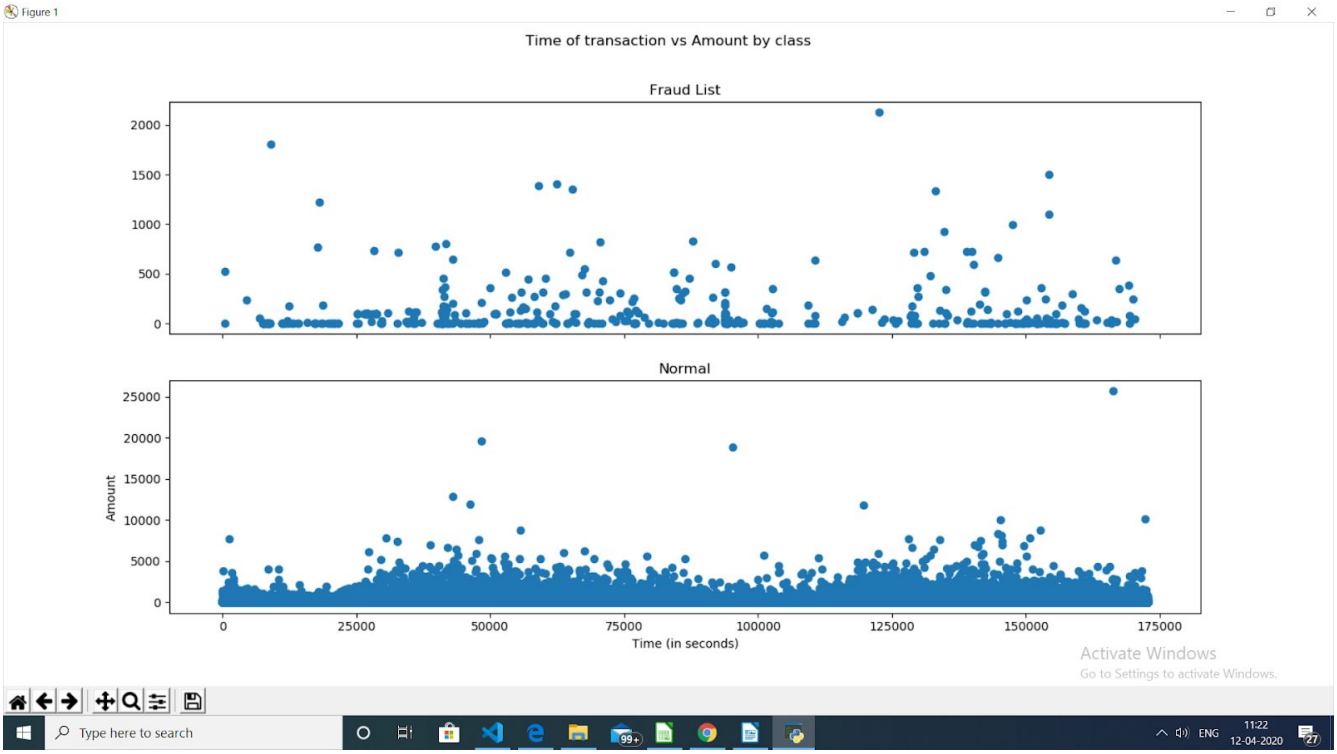


Figure 1
Time of transaction vs Amount by class
Fraud List
Normal
Amount
Time (in seconds)
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
11:22
12-04-2020

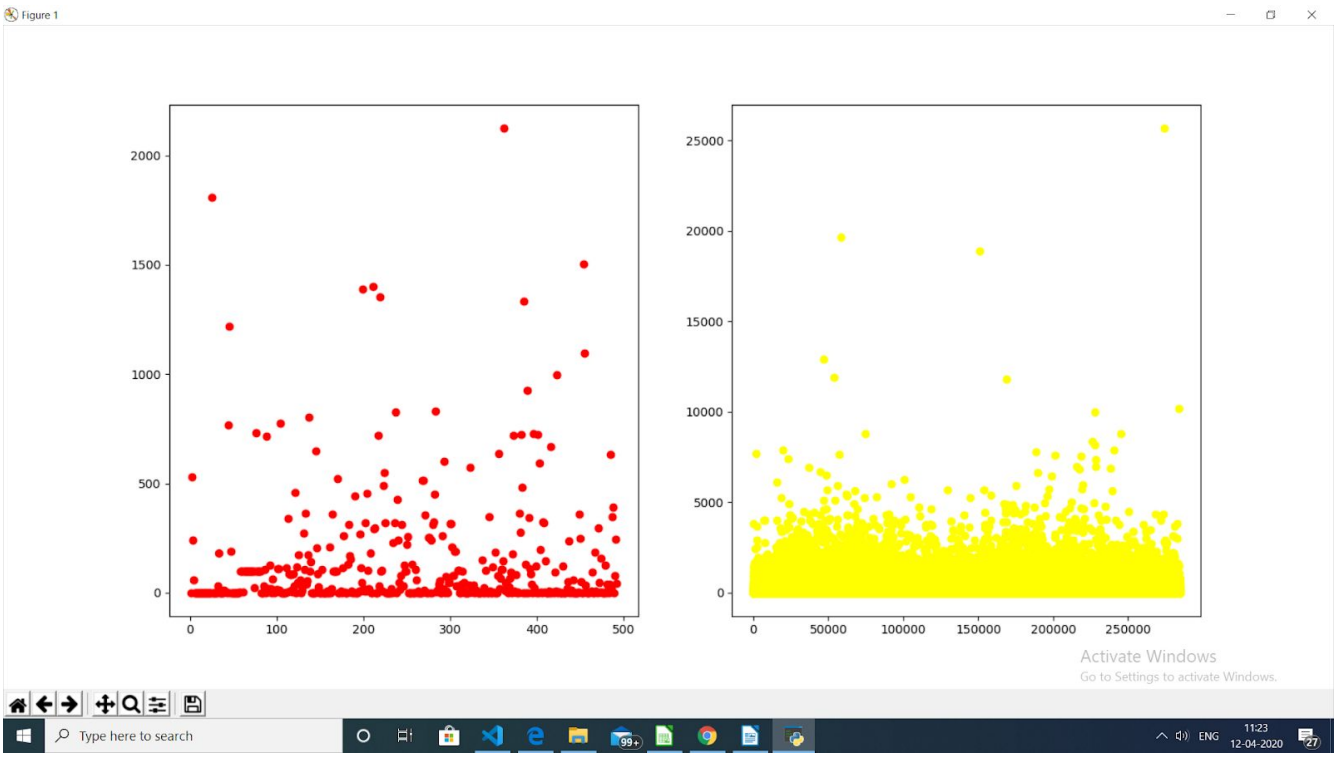


Figure 1
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
11:23
12-04-2020

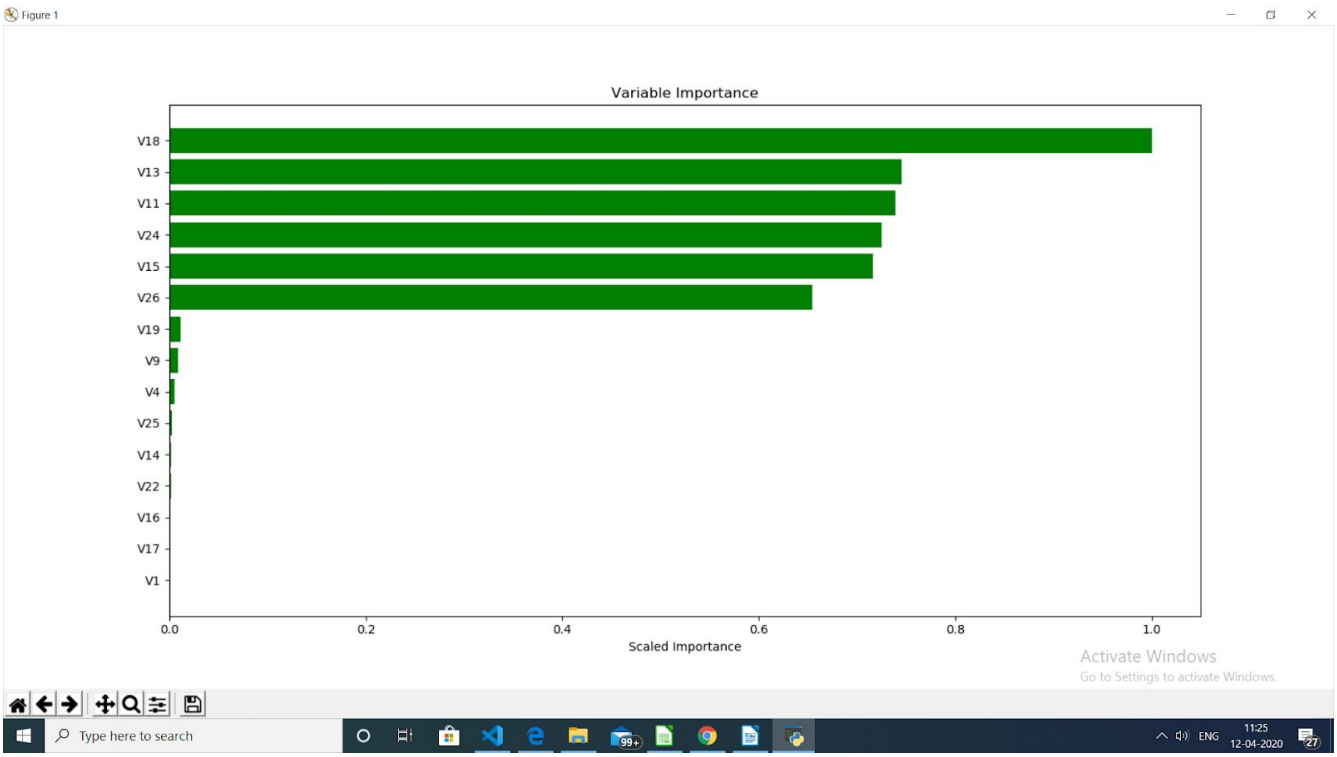


Figure 1
Variable Importance
V18
V13
V11
V24
V15
V26
V19
V9
V4
V25
V14
V22
V16
V17
V1
0.0
0.2
0.4
0.6
0.8
1.0
Scaled Importance
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
11:25
12-04-2020

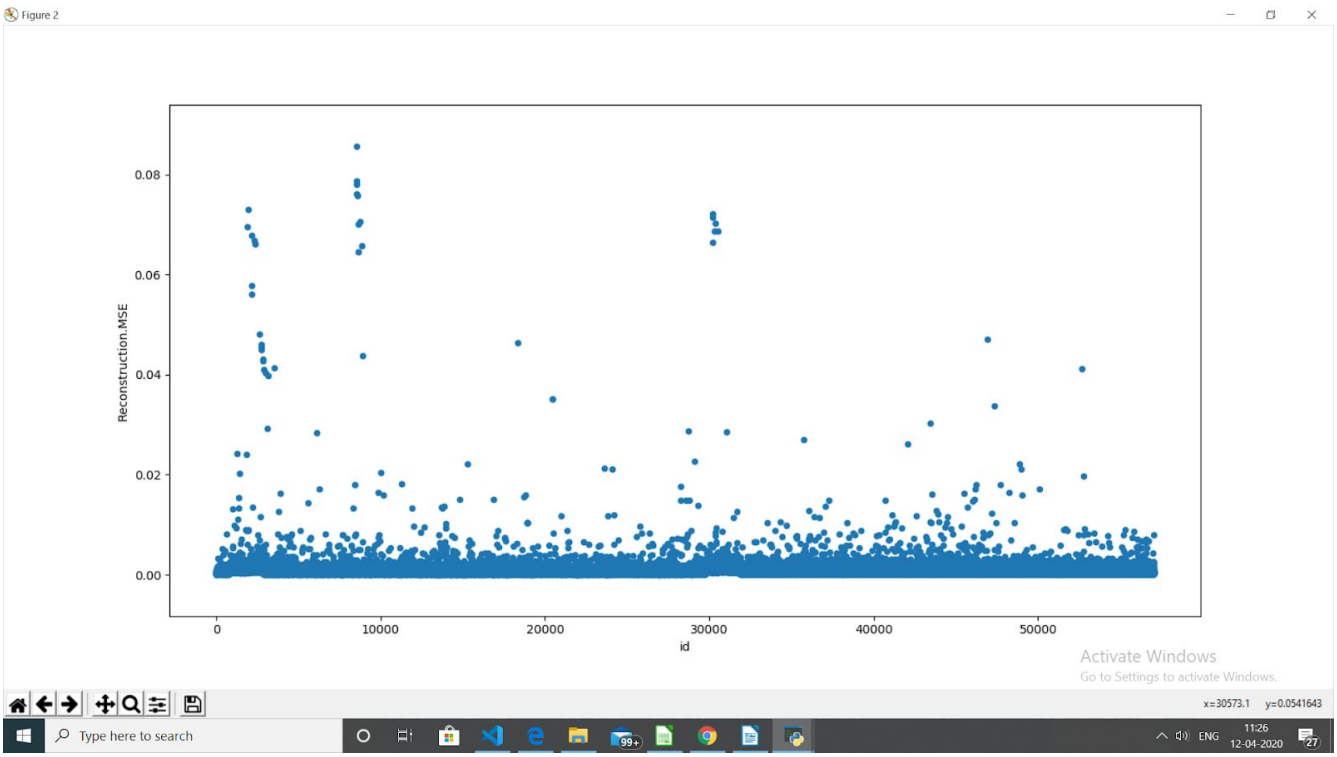


Figure 2
0.08
0.06
0.04
0.02
0.00
Reconstruction.MSE
0
10000
20000
30000
40000
50000
id
Activate Windows
Go to Settings to activate Windows.
x=30573.1 y=0.0541643
Type here to search
ENG
11:26
12-04-2020

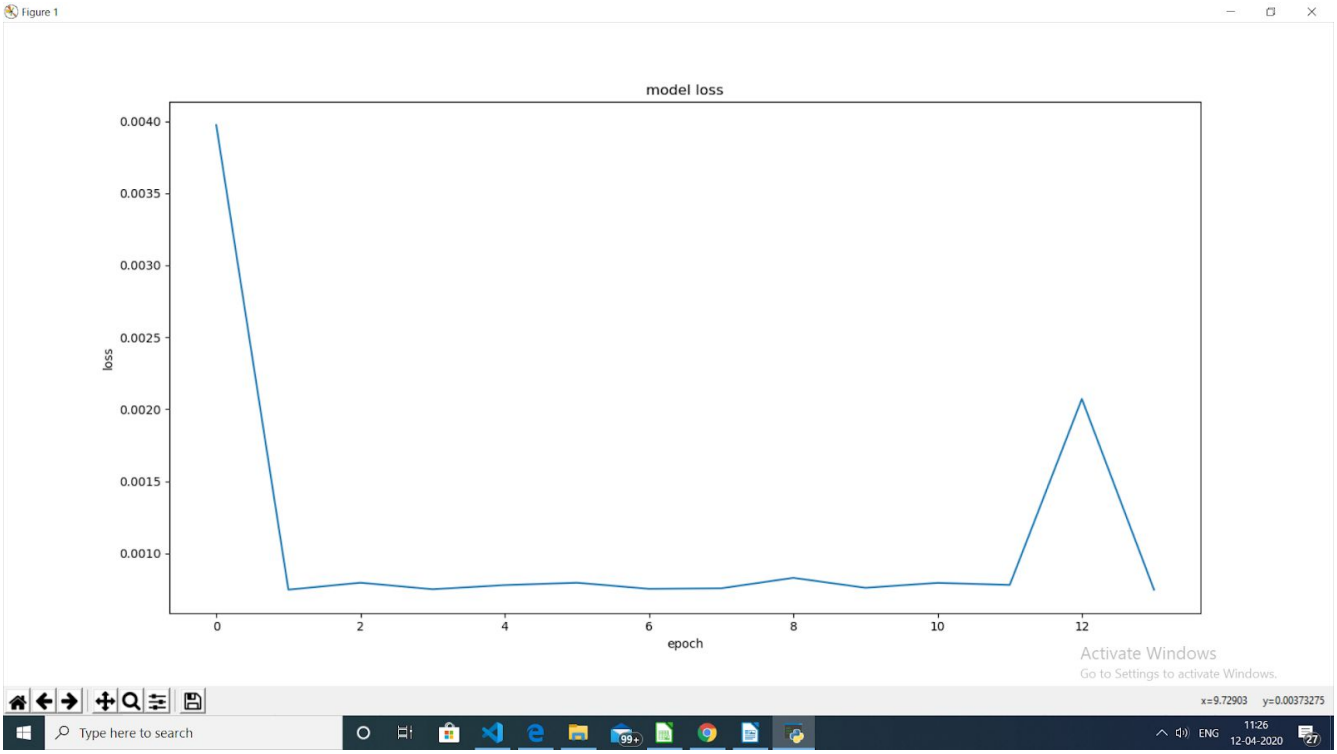


Figure 1
model loss
loss
epoch
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0
2
4
6
8
10
12
Activate Windows
Go to Settings to activate Windows.
x=9.72903 y=0.00373275
Type here to search
ENG
11:26
12-04-2020

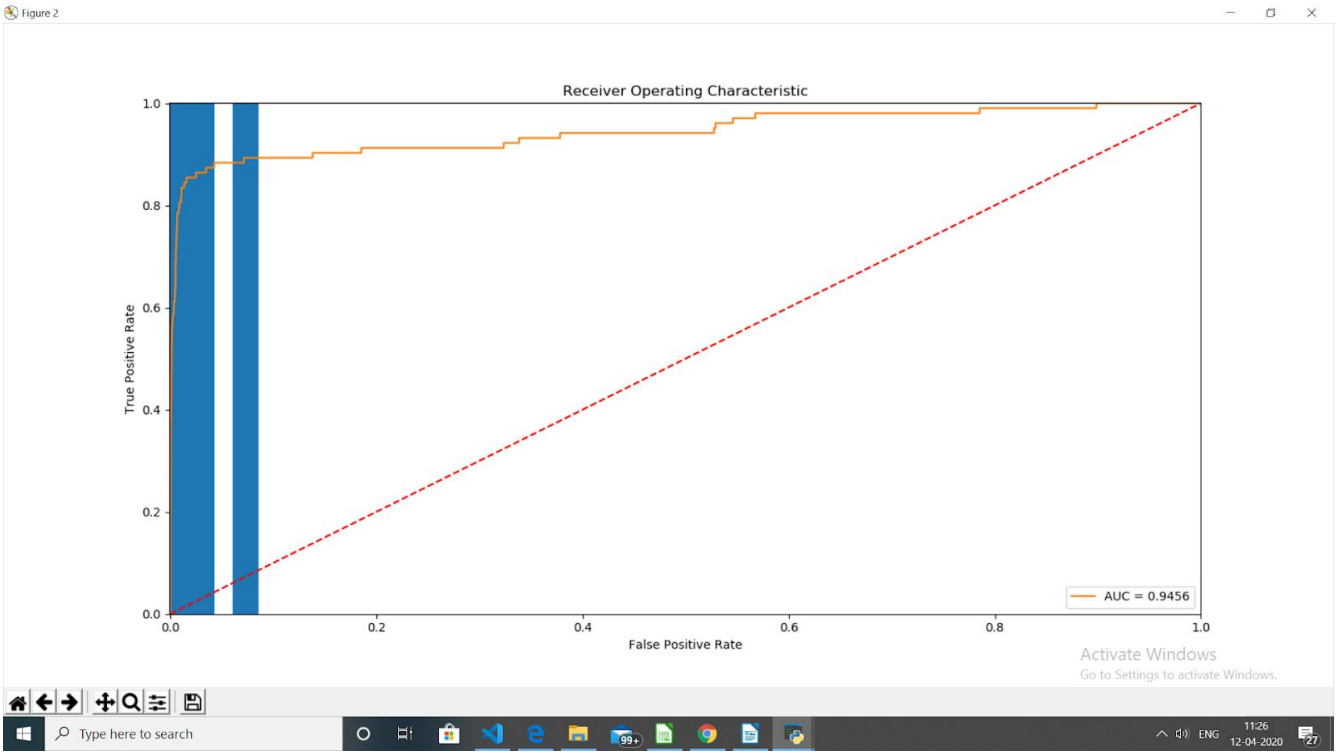


Figure 2
Receiver Operating Characteristic
True Positive Rate
False Positive Rate
AUC = 0.9456
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
11:26
12-04-2020

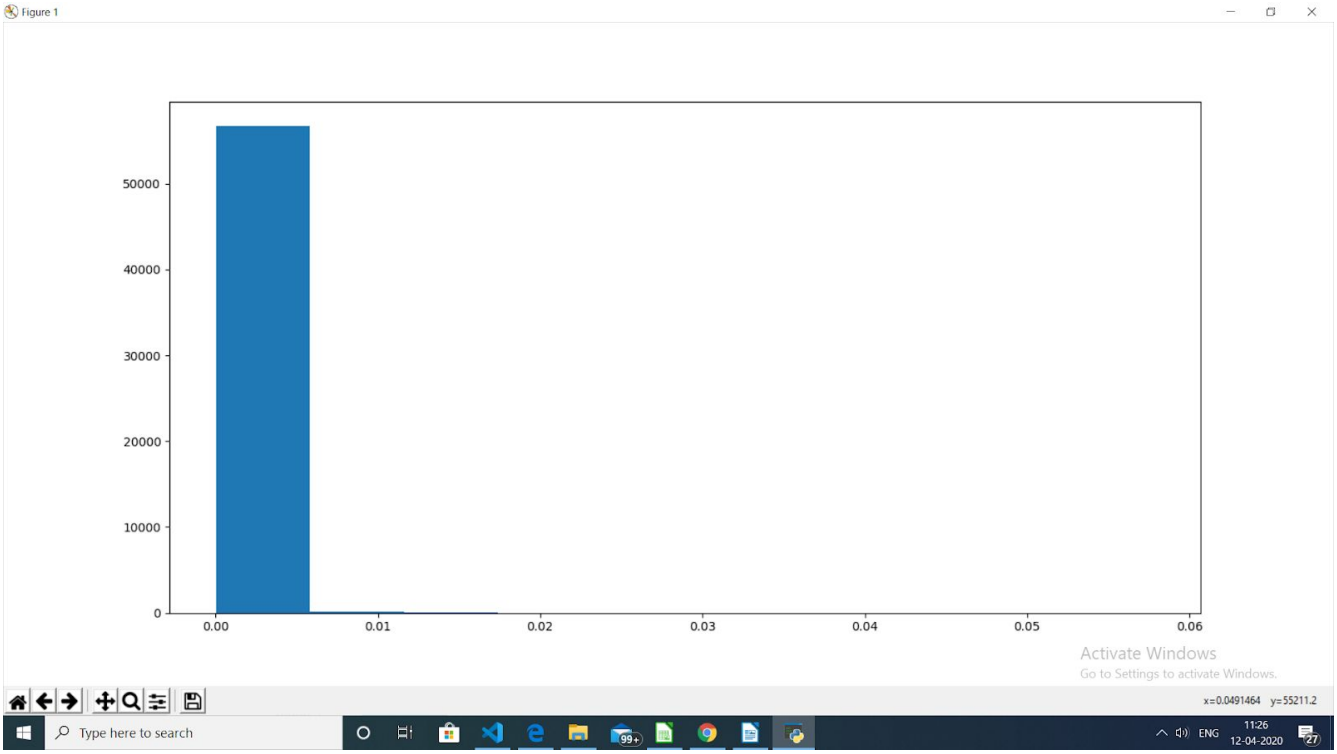


Figure 1
50000
40000
30000
20000
10000
0
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Activate Windows
Go to Settings to activate Windows.
x=0.0491464 y=55211.2
Type here to search
ENG
11:26
12-04-2020

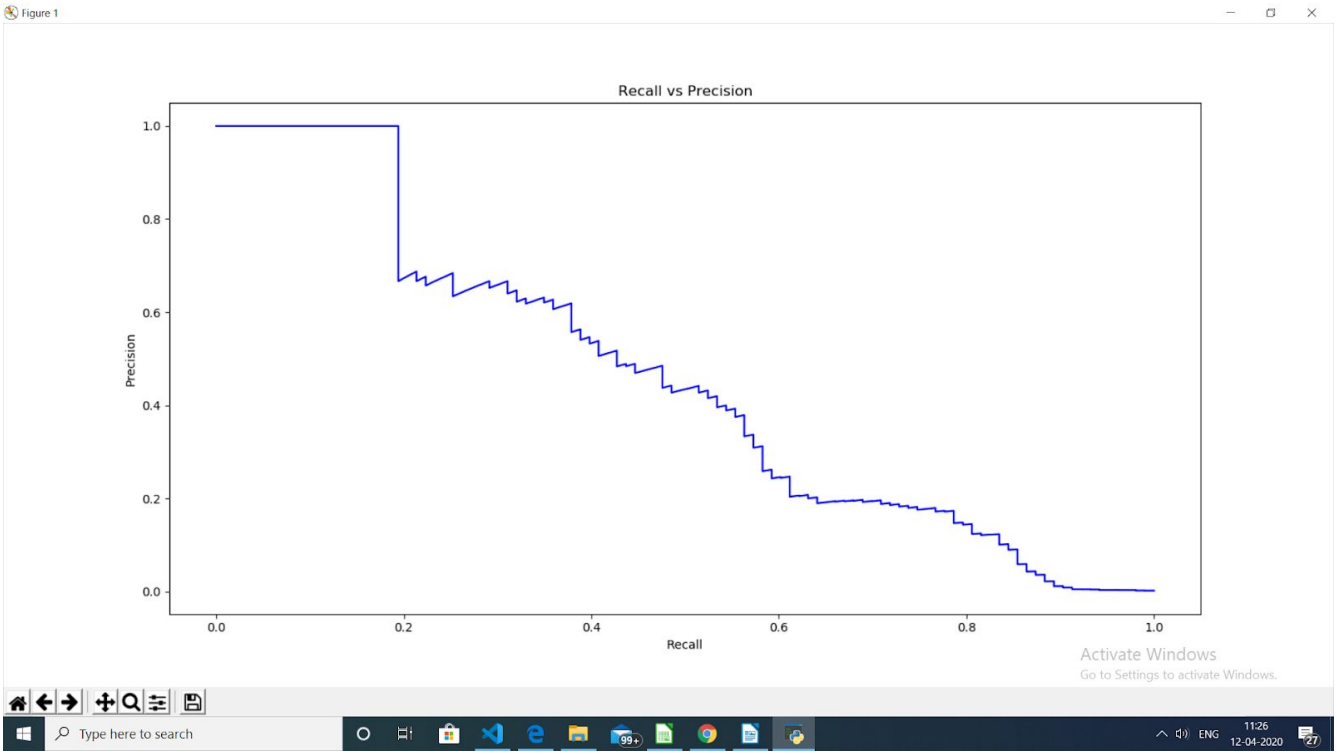


Figure 1
Recall vs Precision
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
11:26
12-04-2020

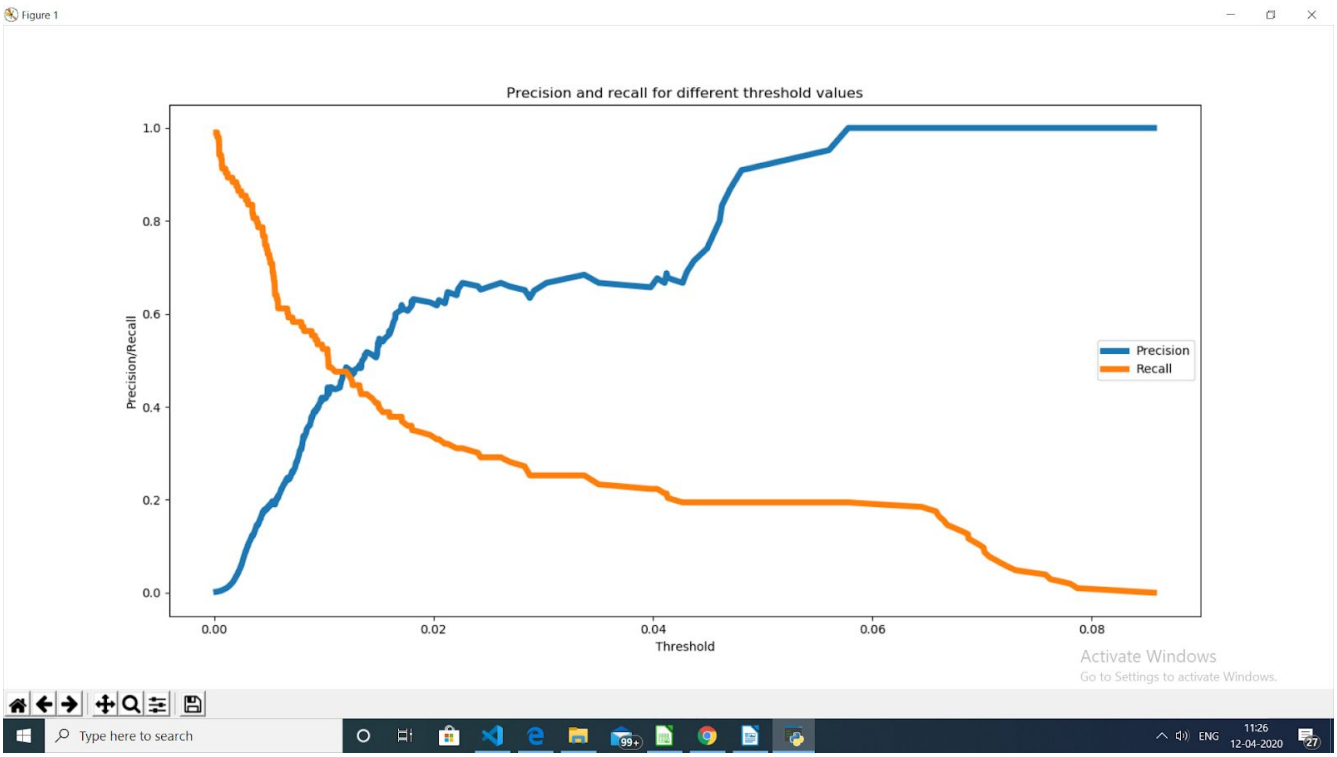


Figure 1
Precision and recall for different threshold values
Precision/Recall
Threshold
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.02
0.04
0.06
0.08
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
11:26
12-04-2020

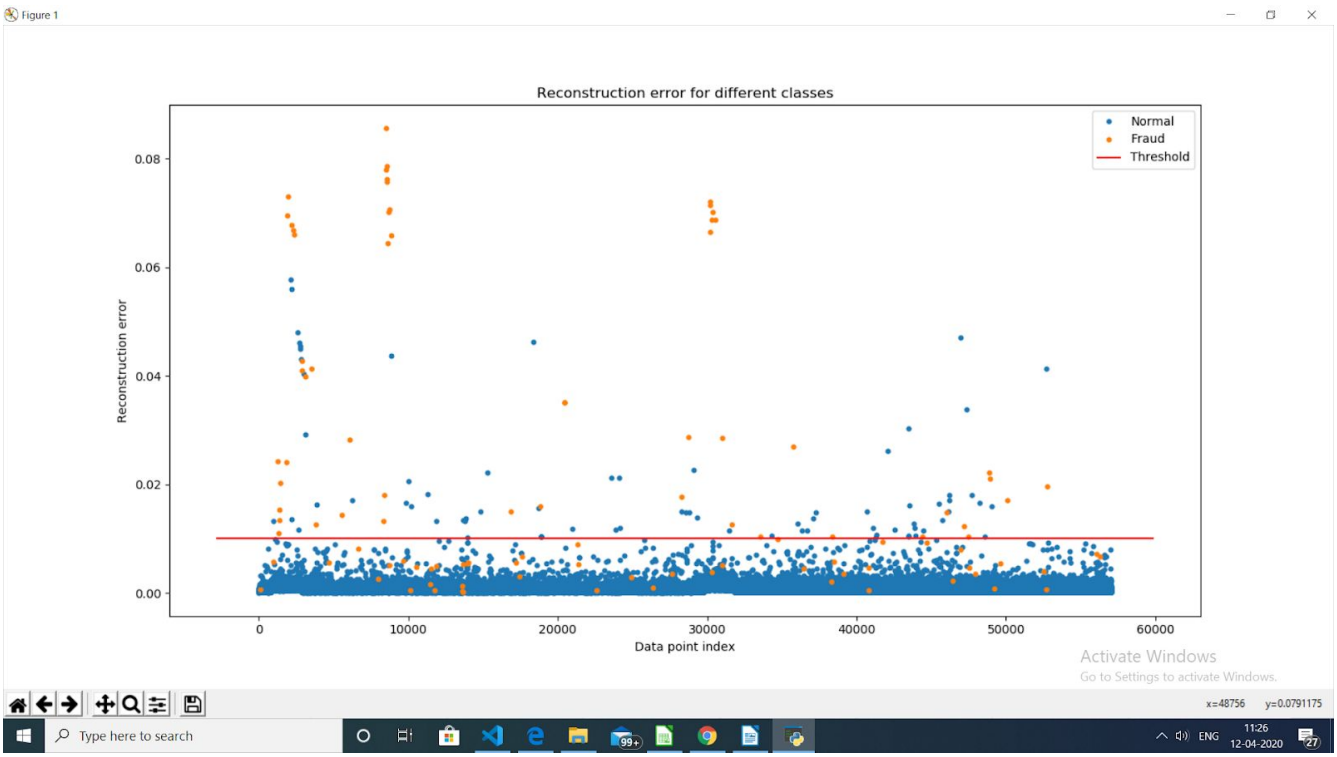
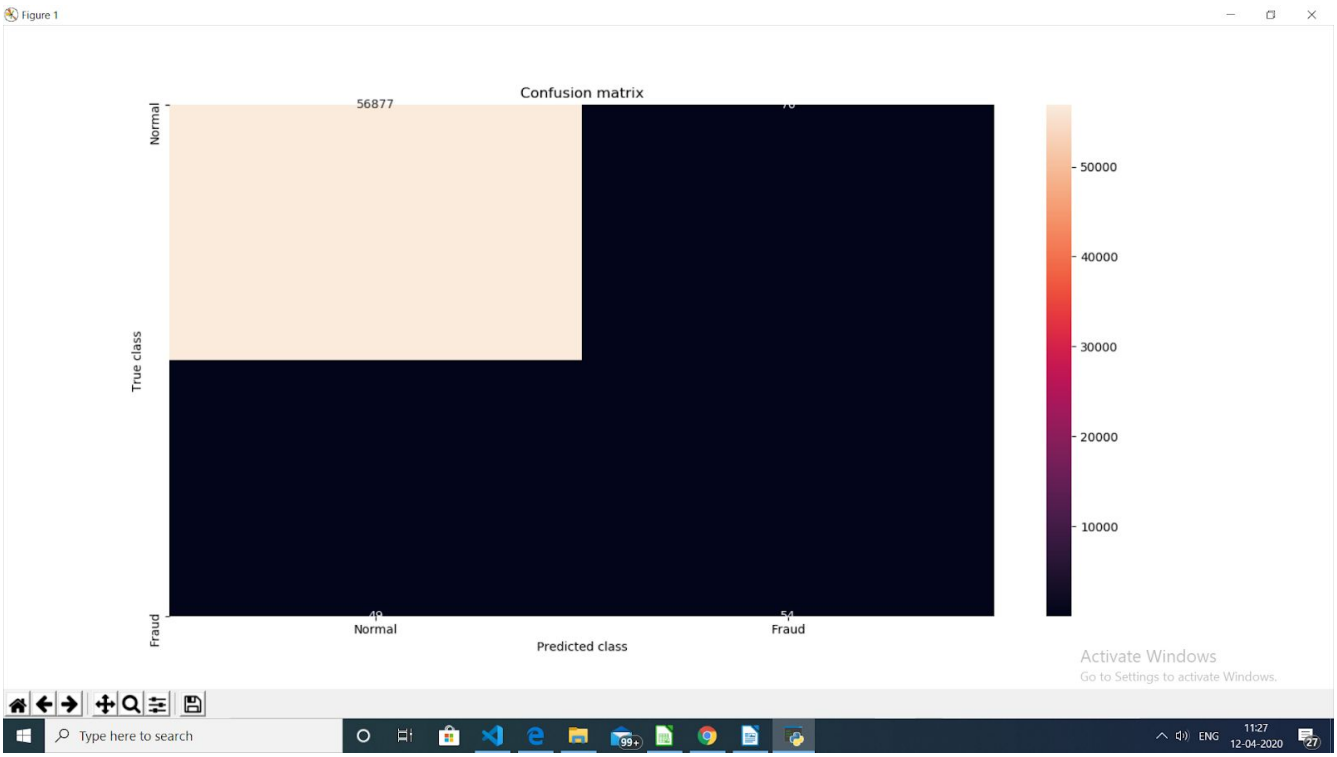


Figure 1
Reconstruction error for different classes
Reconstruction error
Data point index
Normal
Fraud
Threshold
0.08
0.06
0.04
0.02
0.00
0
10000
20000
30000
40000
50000
60000
Activate Windows
Go to Settings to activate Windows.
x=48756
y=0.0791175
Type here to search
ENG
11:26
12-04-2020



Github-Link : https://github.com/Madanpurohit/Credit_Card_Fraud_Detection/tree/master

Data-Set Link : https://www.kaggle.com/mlg-ulb/creditcardfraud

********** END**********