CREDIT CARD FRAUD DETECTION

MAJOR PROJECT – I (20SSP47)

Submitted by

SANJAY K

20MSS040

Under the Guidance of

Dr. R. Umagandhi MCA, M.Phil., Ph.D., NET.

Associate Professor,

In partial fulfilment of the requirements for the award of the degree of

MASTER OF SCIENCE IN SOFTWARE SYSTEMS

(Five Years Integrated Course) of

Bharathiar University

DEPARTMENT OF SOFTWARE SYSTEMS

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous college - Affiliated to Bharathiar University

Accredited with 'A++' grade by NAAC (4th Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Coimbatore - 641 014

NOVEMBER 2023

DEPARTMENT OF SOFTWARE SYSTEMS

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous college - Affiliated to Bharathiar University

Accredited with 'A++' grade by NAAC (4th Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Coimbatore - 641 014

CERTIFICATE

This is to certify that this project work entitled **"Credit Card Fraud Detection"** is a bonafide record of work done by **Mr. Sanjay K (20MSS040)** in partial fulfilment of the requirements for the award of Degree of **Master of Science in Software Systems** (Five years Integrated Course) of Bharathiar University.

Faculty Guide

Head of the Department

Submitted for Viva-Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Mr. Sanjay K (20MSS040)**, hereby declare that this project work entitled "**Credit Card Fraud Detection**" is submitted to **PSG College of Arts & Science, Coimbatore** in partial fulfilment of the requirements for the award of the degree of **Master of Science in Software Systems**, is a record of original work done by me under the supervision and guidance of **Dr. R. Umagandhi MCA, M.Phil., Ph.D., NET.,** Associate Professor, Department of Software Systems, PSG College of Arts & Science, Coimbatore.

This report has not been submitted by me for the award of any other Degree/ Diploma/ Associate ship/ Fellowship or any other similar degree to any other university.

Place: Coimbatore **Sanjay K**

Date: **20MSS040**

ACKNOWLEDGEMENT

My venture stands imperfect without dedicating my gratitude to a few people who have contributed a lot towards the victorious completion for my project work.

I would like to thank **Thiru L. Gopalakrishnan, Managing Trustee, PSG & Sons Charities**, for providing me prospect and surroundings that made the work possible.

I take this opportunity to express my deep sense of gratitude to **Dr. T. Kannaian, Secretary** of PSG College of Arts & Science, Coimbatore for permitting and doing the needful towards the successful completion of this project.

I express my deep sense of gratitude and sincere thanks to our Principal **Dr. D. Brindha, M.Sc., M.Phil., Ph.D.**, **MA (Yoga).,** for her valuable advice and concern on students.

I am very thankful to **Dr. A Anguraj, M.Sc., M.Phil., Ph.D**., Vice Principal (Academics),**Dr. Jayanthi M M.Com., MBA., M.Phil., Ph.D.,** Vice Principal (Student Affairs), Dr**. M Umarani, MBA, M.Phil., Ph.D.**, **Faculty-In-Charge (Student Affairs)**, for their support towards my project.

I sincerely thank Dr. K.V. Rukmani., MCA., M.E., Ph.D., Head of the Department, Department of Software Systems for his whole hearted help to complete this project successfully by giving valuable suggestions. I convey my heartiest and passionate sense of thankfulness to my project guide **Dr. R. Umagandhi MCA, M.Phil., Ph.D., NET., Associate Professor,** Department of Software Systems, for her suggestions which had enabled me to complete the project successfully. I also convey my sincere gratitude to all the faculty members of the Department of Software Systems for their immense support, guidance and suggestions.

This note of acknowledgement will be incomplete without paying my heartful devotion to my parents, my friends and other people, for their blessings, encouragement, financial support and the patience, without which it would have been impossible for me to complete the job.

- **Sanjay K (20MSS040)**

COMPANY CERTIFICATE

WeDesignTech
PRIVATE LIMITED

www.wedesigntech.com
admin@wedesigntech.com

Mr.SANJAY.K [20MSS040]
M.Sc., Software Systems,
PSG College of Arts & Science.

Date : 30.06.2023
Place: Coimbatore

INTERNSHIP CONFIRMATION LETTER

Dear **SANJAY.K,**

We are pleased to inform you that your engagement as **Web Design Trainee** ***(Intern)*** of our company, i.e., ***WEDESIGNTECH PVT LTD,*** has been approved. The terms of your internship with the company will be as follows:

1. You will be working as an Intern under the guidance of Mr. D.Parthiban, and she would be your Reporting Manager too.
2. Intern Period starts from 28th June'2023.
   Working and will be trained in the office environment workspace.
3. You will be monitored, online. Punctual timings at log-in and log-out are expected. You are solemnly responsible for the tasks assigned, and its timely updates.
4. Your performance will be evaluated from time to time and depending upon your performance.

We look forward to associate with you and wish you all the very best in your future endeavor.

Please don't hesitate to contact us at admin@wedesigntech.com *with regard to your internship and formalities.*

WeDesignTech Pvt Ltd Coimbatore 641 014

No. 20/1, Kurunji Nagar,
Civil Aerodrome Post, Goldwins,
Coimbatore - 641 014.

0422 262 5030

INTERNSHIP COMPLETION CERTIFICATE

WeDesignTech
PRIVATE LIMITED

www.wedesigntech.com
admin@wedesigntech.com

Date: 29.09.2023 Place: Coimbatore

TO WHOMSOEVER IT MY CONCERN

This is to certify that **Mr.Sanjay. K (Reg.no.20MSS040)** Fourth year M.Sc (Software System) student of **"PSG College of Arts & Science – Coimbatore"** has successfully completed his Internship Training in our esteemed organization from 28th June 2023 to 29th September 2023.

His performance and conduct were found to be very good,

During this period, he was sincere and regular in attending all the phases of Internship Training Program.

Thanking you,

For WEDESIGNTECH PVT LTD,

Authorized Signatory

No. 20/1, Kurunji Nagar,
Civil Aerodrome Post, Goldwins,
Coimbatore - 641 014.

0422 262 5030

SYNOPSIS

The project entitled "**Credit Card Fraud detection**" in which user can be able to view the findings of the Machine learning Algorithm from their place through the Internet. Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch. And the Machine learning Algorithm can be uploaded to a website enables the users to easily have an access over all available algorithms through internet.
Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 COMPANY PROFILE

WeDesignTech is a website designing and eCommerce website development company in Coimbatore, Tamil Nadu, India. With over a decade of experience in the digital and website development business, they provide the better possible solution to reach your business goals. They carefully listen to every customer's needs and understand the business's potential needs to provide the perfect customized solution to fulfil your needs. They help to couple your vision and goals to focus on the target. They take every project personally and strive to make it a career milestone.

They are a team of highly talented individuals, starting from simple planning to marketing it globally. Everything is created twice, first in mind, next in reality. Their team begins with a powerful brainstorming session and a series of step-by-step processes to make it live.

They have better foolproof methodologies to follow a perfect flow with solid core knowledge. They have over 100+ websites in their portfolio. Be it a simple business website or a large eCommerce store. their team makes your imaginary website come to life. They have a professional support team to guide and help you every step of the way.

COMPANY INFO

ADDRESS: No.20/1, Kurinji Nagar, Goldwins,Coimbatore, Tamil Nadu, India – 641014.

EMAIL: admin@wedesigntech.com

CONTACT: +91 09994061031

1.2 PROJECT OVERVIEW

The prime objective of this project is to recognize fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase.

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. this machine learning model is created, in order to detect the credit card fraud transactions and help the users to be aware of this type scams. this machine learning model is created.

The Machine learning model is created and hosted in the internet using Flask API, Python3, HTML5 and CSS3. First the user has to open the webpage created by using the flask server in their browser of their choice where they can find the overview and description of the Machine Learning model after reading the details the user can find a button which gives access to the user to run the Machine learning model and view the output in their browser along the graphs and diagrams to make the user understand better about the model and the Dataset

1.3 MODULE DESCRIPTION

1.3.1 COLLECTION OF DATASETS

The first module consists of collecting datasets from different sources such as Kaggle, tableau, google research, google cloud, papers with code etc... The dataset is collected in the csv format because it is human readable and easy to edit, it is compact and it is faster to handle which makes it easy to analyse and move forward to the next step without any complications

1.3.2 DATA EXPLORATION

The dataset that has been collected from various sources are explored individually since various sources consists of various types of data i.e. various types of fields are present in various datasets. Those values are explored and listed either manually or it can be done using python commands. Once the explorations are done and the datasets are merged together, so that we can proceed to the next step

1.3.3 DATA PREPROCESSING

In this step the datasets are merged together to form a large dataset so that our machine learning model can be trained to produce results with better accuracy ratio. After merging the datasets, we have check for holes in the dataset such null values, NaN (not a number values), faulty values, uncoordinated values etc… after checking for these possible errors we have find the number of fraud transactions and normal transactions available in out dataset to build our model. We have to find the similarities or connections between values or fields so that we can build our machine learning model

1.3.4 SPLITTING THE DATA INTO TRAINING AND TESTING DATA

In this step the datasets we have collected and created is divided into two segments namely training data and testing data for that we are using train_test_split and standard scalar from scikit-learn and we dividing the dataset into 80% for training purpose and 20% of the dataset for testing purpose

1.3.5 MACHINE LEARNING MODEL BUILDING

In this step we have to choose the machine learning algorithm that are best suited for our problem. In our case we have chosen the Artificial Neural Network (ANN) algorithm to make our machine learning model. We are using keras sequential along with keras metrics from TensorFlow to build our model. The keras metrics is used to find false negatives, false positives, true negatives, true positives etc… and keras sequential is used to perform normalize, dense and dropout operations in the layers

1.3.6 SHOWCASE OUR FINDINGS

In this step we will showcase our findings and working process of the machine learning model (ANN) along with the graphs and diagrams that are created when analysing the datasets when running the machine learning model are displayed in a HTML webpage along with the accuracy ratio

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In existing system, the user cannot access the working process of the machine learning model through a webpage. The existing system does not provide adequate amount of visual representation to help the users understand the problem and dataset. The accuracy provided by the existing system is low compared to the proposed system. The existing system consists of machine learning model that are made complicated and it takes lot of effort to understand.

2.2 PROPOSED SYSTEM

The proposed system can be accessed through online. It will give easy access over any user through online. The whole working process along with the visual representation is available in one place. The user did not need to search of machine learning model at different places. Minimal amount of man power is required as this fully controlled over internet.

This website is available 24x7 to serve users. It is easy and simple to use. This website can make fraud detection easy and simple. With simple changes we can cope up with increasing adaptation among the scammers. The response time taken to provide the output is lesser compared to the existing system.

The user can spend less amount of time trying to figure out where the amount has gone after various transactions. With this website, even a user without machine learning knowledge can access the machine learning model. In this website, the user can get a detailed explanation about the working process and visual representations of the process and dataset.

CHAPTER 3

SYSTEM CONFIGURATION

3.1 HARDWARE SPECIFICATION

COMPONENTS		REQUIREMENTS
PROCESSOR	:	INTEL CORE i5
RAM	:	4 GB RAM
KEY BOARD	:	STANDARD KEYBOARD

3.2 SOFTWARE SPECIFICATION

COMPONENTS		REQUIREMENTS
DEVELOPMENT ENVIRONMENT	:	WINDOWS 7 OR ABOVE
EDITOR	:	VISUAL STUDIO CODE
FRONT END	:	HTML, CSS
BACK END	:	PYTHON, FLASK

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 FRONT END

4.1.1 HTML

Hypertext Markup Language (HTML) is a computer language that makes up most web pages and online applications. It was developed with a motive to create electronic documents called pages displayed on the World Wide Web. A hypertext is a text which is employed to reference other pieces of text, while a markup language is a series of markings that informs web servers about the structure and style of a document. HTML codes are used to design how a browser displays web page elements, such as text, hyperlinks, and media files.

All HTML pages consists of a series of HTML elements, consisting of a set of tags and attributes. HTML elements are the building blocks of a web page. Most tags are contained within less than and greater than angle brackets. Except few tags, all the other tags have their corresponding closing tags. If the closing tags are omitted, the browser applies the effect of the opening tag till the end of the page. A tag tells the web browser where an element begins and ends, whereas an attribute describes the characteristics of an element. A properly written HTML document will be readable to the user, and also conveys the structure of the document, the relationship of its content to each other, and allow the user to link to other pages and sites.

FEATURES OF HTML

HTML has multiple features. Some of them are, simple and ease of use, most browsers support HTML, Free of cost, Compatible with a majority of development tools**,** Easy to edit, Effortless incorporation, etc.,

4.1.2 CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. CSS handles the look and feel part of a web page. It is generally used with HTML to change the style of web pages and user interfaces.

CSS has the potential of handling the design of all web pages at once. It enables the developers to make sure that style elements are applied consistently across the web pages. The accessibility of the website becomes more while using CSS.

FEATURES OF CSS

CSS has several features. Some of them are, it saves time, pages load faster, easy maintenance, superior styles of HTML, multiple device compatibility and so on.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

4.2 BACK END

4.2.1 PYTHON

Python is a simple, general purpose, high level, and object-oriented programming language. Python is a general-purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is an easy-to-learn yet powerful and versatile scripting language, which makes it attractive for Application Development. With its interpreted nature, Python's syntax and dynamic typing make it an ideal language for scripting and rapid application development. Python supports multiple programming patterns, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. It is a multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is dynamically typed, so we can write a=10 to assign an integer value in an integer variable. Python makes development and debugging fast because no compilation step is included in Python development, and the edit-test-debug cycle is very fast.

FEATURES OF PYTHON

Python has many web-based assets, open-source projects, and a vibrant community. Learning the language, working together on projects, and contributing to the Python ecosystem are all made very easy for developers. Because of its straightforward language framework, Python is easier to understand and write code in. This makes it a fantastic programming language for novices. Additionally, it assists seasoned programmers in writing clearer, error-free code. Python is an open-source, cost-free programming language. It is utilized in several sectors and disciplines as a result. In Python, code readability and maintainability are important. As a result, even if the code was developed by someone else, it is easy to understand and adapt by some other developer. Python has many third-party libraries that can be used to make its functionality easier. These libraries cover many domains, for example, web development, scientific computing, data analysis, and more.

4.2.2 FLASK

Flask is a web framework that provides libraries to build lightweight web applications in python. It's having a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

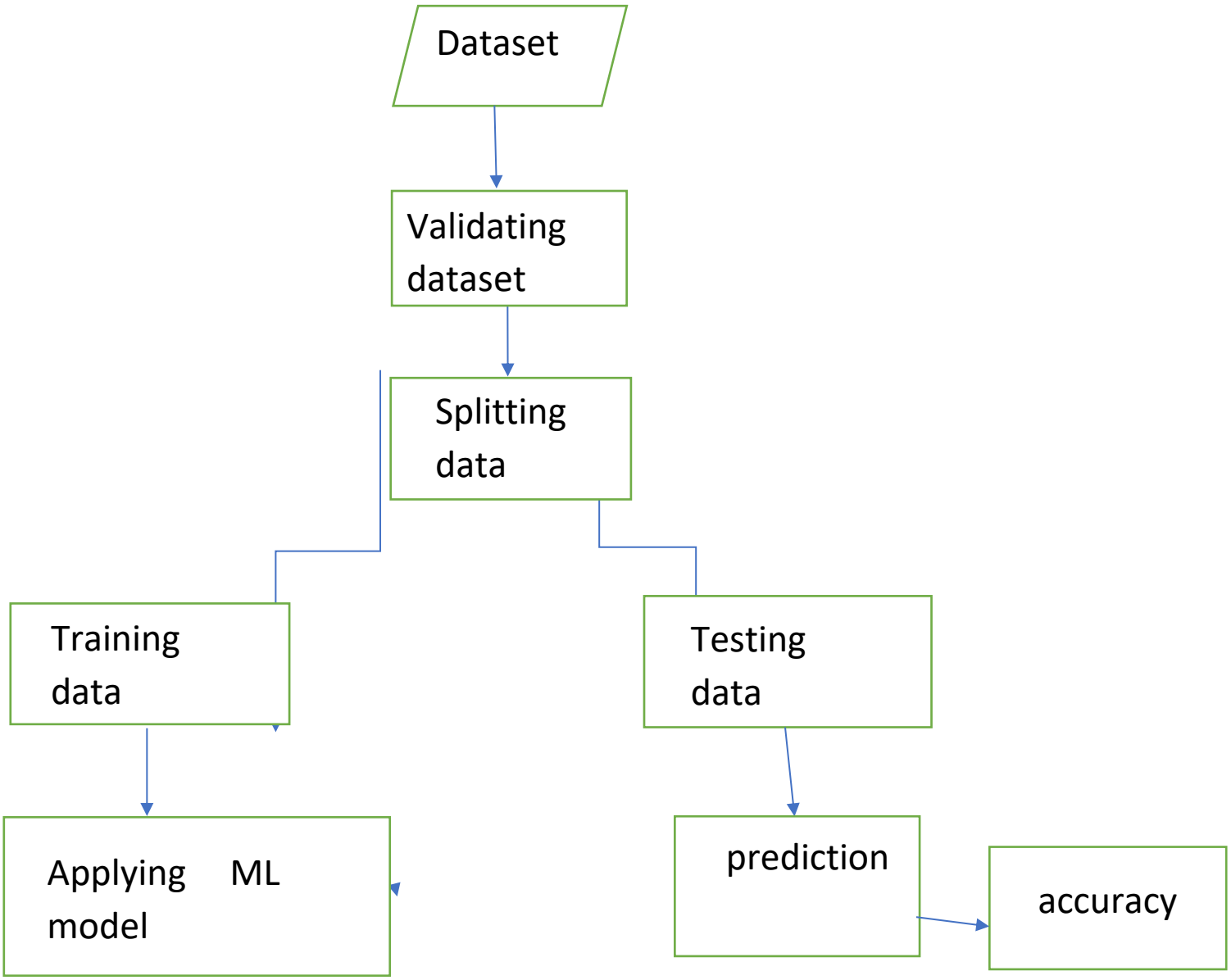
FEATURES OF FLASK

- built-in development server and fast debugger
- integrated support for unit testing
- RESTful request dispatching

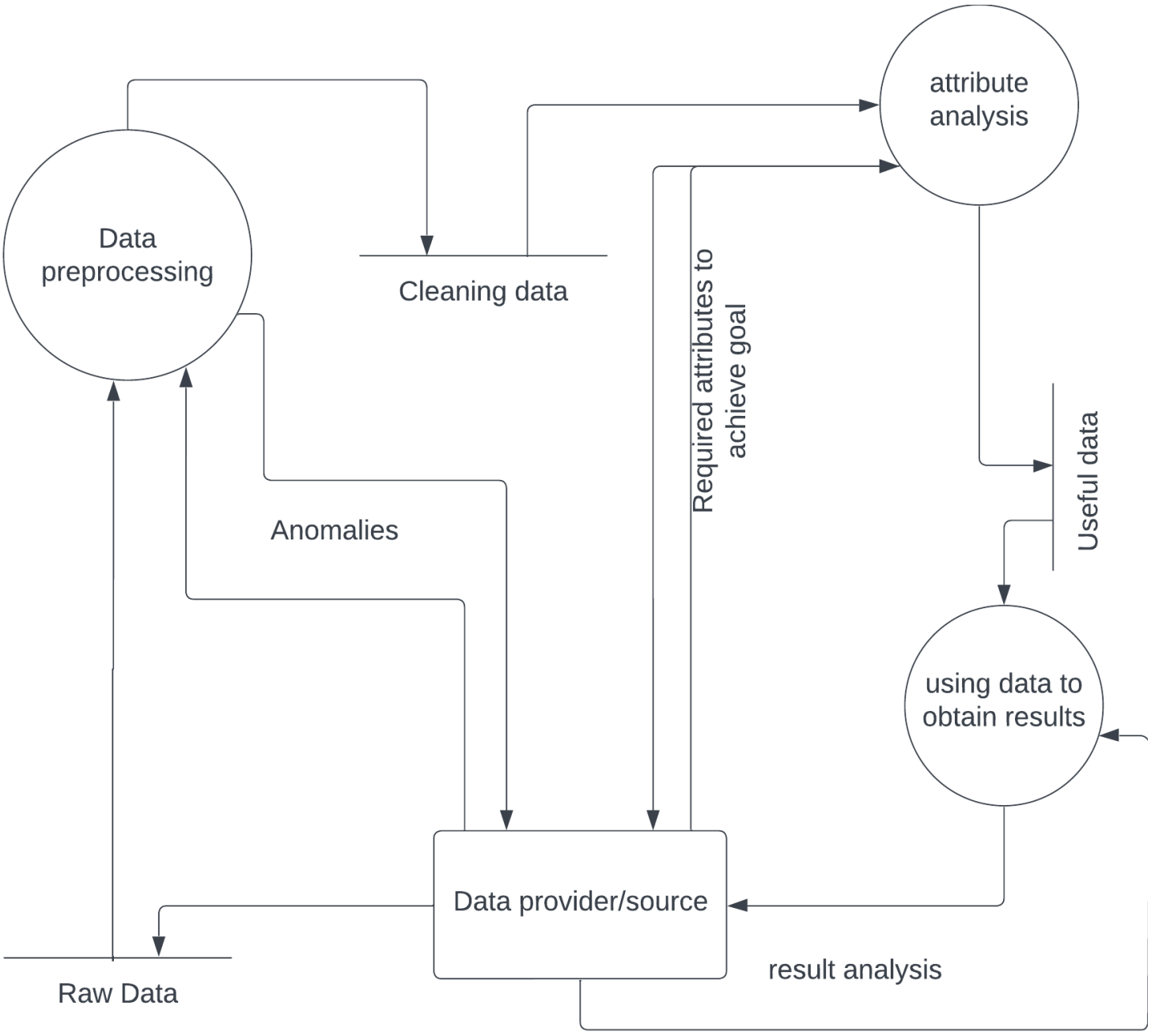
CHAPTER 5

SYSTEM DESIGN

5.1 STRUCTURE DIAGRAM



5.2 PROCESS DIAGRAM



5.4 INPUT DESIGN

Input design lies as the link between the information system and the user. It is the part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system. If the data going into the system are incorrect, then the

processing and output will magnify these errors. The following are the main objectives of input design.

- Achieving the highest possible level of accuracy.
- Ensuring that the input is acceptable and understood by the user.

In this project, the user is required to enter the required dataset in the program manually and , the rest of the process takes place according to the coded functions in the backend.

The data that are to be inserted carefully as this plays a very important role. In order to get the correct output and to achieve good accuracy, the input should be acceptable and understandable by the user.

5.5 OUTPUT DESIGN

Output design plays a very important role in a system. Getting a correct output is a task that has to be concentrated, as a system is validated as a correct one only if it gives the correct output according to the input.

In this project the accuracy ratio plays a vital role. It displays accuracy of the both training and testing model in the webpage

CHAPTER 6

6. SYSTEM IMPLEMENTATION AND TESTING

6.1 SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a system project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of the changeover method.

The first task in implementation is planning deciding on the methods and time scale to be adopted. Once the planning has been completed the major effort is to ensure that the programs in the system is working properly when the staff has been trained, the complete system involving both computer and user can be executed effectively. Thus, the clear plans are prepared for the activities. Implementation means the process of converting a new or a revised system design into an operational one.

6.2 SOFTWARE TESTING

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected.

The process of testing the functionality and correctness of software by running it is called software testing. It is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Software testing is usually performed for two reasons.

- Defect detection

- Reliability estimation

6.2.1 TESTING OBJECTIVES

• Testing is a process of executing a program with the intent of finding an error.

• A good test case is one that has a high probability of finding an as yet undiscovered.

• A successful test is one that uncovers an as yet undiscovered error.

6.2.2 TESTING PRINCIPLES

• All tests should be traceable to customer requirements.

• Tests should be planned large before testing begins.

• Testing should begin “In the Small” and progress towards “In the Large”.

6.2.3 TYPES OF TESTING

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Unit Testing:

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Each module can be tested using the following two strategies:

Black Box Testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. In this testing only the output is checked for correctness.

The logical flow of the data is not checked.

This testing has been uses to find errors in the following categories:

a) Incorrect or missing functions

b) Interface errors

c) Errors in data structure or external database access

d) Performance errors

e) Initialization and termination errors

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

a) Guarantee that all independent paths have been executed.

b) Execute all logical decisions on their true and false sides.

c) Execute all loops at their boundaries and within their operational bounds.

d) Execute internal data structures to ensure their valid

Integrating Testing

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

Acceptance Testing

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

Validation Testing

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed

CHAPTER 7

CONCLUSION & FUTURE RECOMMENDATION

It is hereby concluded that the website and the machine learning model works well and satisfy the users by predicting the fraud transactions with highest accuracy. It has all the required features. This project has a user side which is simple and easy to access. The website is devoid of any issues and errors are properly debugged. This project can be accessed by more than one user once it is hosted on the web.

In the future enhancement each application contains its own advantages and disadvantages. The project has almost covered all the needs and prerequisite. Further refinements and enhancements can be easily done since the coding is mostly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements with extra features like adding other machine learning models, Improving the user interface and user experience will be included. So the application works in more useful manner.

9. BIBLIOGRAPHY

The following books and websites were referred during the analysis and execution phase of the project.

REFERENCE BOOKS:

1. "Python for Data Analysis - Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney was published by O'Reilly Media, Inc
2. "Python Data Science Handbook Essential Tools for Working With Data" is the best book to learn python for data science, written by Jake Vander Plas and released by O'Reilly Media, Inc.
3. Neural Networks and Deep Learning: A Textbook Hardcover – 13 September 2018by Charu C. Aggarwal
4. Flask Web Development, 2nd Edition by Miguel Grinberg Released March 2018by O'Reilly Media, Inc.
5. HTML & CSS: The Complete Reference, Fifth Edition, Thomas A Powell, McGraw-Hill Education; 5th edition (16 January 2010)
6. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web pages,

Eric Freeman, Elisabeth Robson, O'Reilly Media; 2nd edition (September 18, 2012)

REFERENCE WEBSITES:

1. https://stackoverflow.com/
2. https://www.python.org
3. https://www.geeksforgeeks.org/html/
4. https://www.ibm.com/topics/neural-networks
5. https://www.javatpoint.com/flask-tutorial
6. https://www.javatpoint.com/artificial-neural-network

CHAPTER 10

10. APPENDIX

10.1 SCREENSHOTS

HOME PAGE

Major Project - Credit Card Fraud Detection

Description

Credit card fraud is when someone uses another person's credit card or account information to make unauthorized purchases or access funds through cash advances. Credit card fraud doesn't just happen online; it happens in brick-and-mortar stores, too. As a business owner, you can avoid serious headaches – and unwanted publicity – by recognizing potentially fraudulent use of credit cards in your payment environment.

Challenges

The challenge is to recognize fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase.

Main challenges involved in credit card fraud detection are:
Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time. Imbalanced Data i.e most of the transactions (99.8%) are not fraudulent which makes it really hard for detecting the fraudulent ones. Data availability as the data is mostly private. Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported. Adaptive techniques used against the model by the scammers. The model used must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible. We can make the model simple and interpretable so that when the scammer adapts to it with just some changes we can have a new model up and running to deploy.

Modules

- Collection of Dataset
- Checking the Dataset for empty values and Noisy or unwanted data
- Splitting the data into Training data and Test data
- Training the models and choosing the model with highest accuracy raio
- Printing the predicted values with accuracy ratio

Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time. Imbalanced Data i.e most of the transactions (99.8%) are not fraudulent which makes it really hard for detecting the fraudulent ones. Data availability as the data is mostly private. Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported. Adaptive techniques used against the model by the scammers. The model used must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible. We can make the model simple and interpretable so that when the scammer adapts to it with just some changes we can have a new model up and running to deploy.

Modules

- Collection of Dataset
- Checking the Dataset for empty values and Noisy or unwanted data
- Splitting the data into Training data and Test data
- Training the models and choosing the model with highest accuracy raio
- Printing the predicted values with accuracy ratio

Observations

Very few transactions are actually fraudulent (less than 1%). The data set is highly skewed, consisting of 492 frauds in a total of 284,807 observations. This resulted in only 0.172% fraud cases. This skewed set is justified by the low number of fraudulent transactions. The dataset consists of numerical values from the 28 'Principal Component Analysis (PCA)' transformed features, namely V1 to V28 And my predictions is based on two values namely time and amount. There is no missing value in the dataset.

Access my Work By Clicking The Button Below

Access My Work

This Project Was Done By - Sanjay K (20MSS040)

OUTPUT PAGE

Credit Card Fraud Detection

Time V1 V2 V3 V4 V5 V6 ... V24 V25 V26 V27 V28 Amount Class 0 0.0 -1.359807 -0.072781 2.536347 1.378155 -0.338321 0.462388 ... 0.066928 0.128539 -0.189115 0.133558 -0.021053 149.62 0 1 0.0 1.191857 0.266151 0.166480 0.448154 0.060018 -0.082361 ... -0.339846 0.167170 0.125895 -0.008983 0.014724 2.69 0 2 1.0 -1.358354 -1.340163 1.773209 0.379780 -0.503198 1.800499 ... -0.689281 -0.327642 -0.139097 -0.055353 -0.059752 378.66 0 3 1.0 -0.966272 -0.185226 1.792993 -0.863291 -0.010309 1.247203 ... -1.175575 0.647376 -0.221929 0.062723 0.061458 123.50 0 4 2.0 -1.158233 0.877737 1.548718 0.403034 -0.407193 0.095921 ... 0.141267 -0.206010 0.502292 0.219422 0.215153 69.99 0 [5 rows x 31 columns]

Time V1 V2 V3 V4 V5 ... V25 V26 V27 V28 Amount Class count 284807.00 284807.00 284807.00 284807.00 284807.00 284807.00 ... 284807.00 284807.00 284807.00 284807.00 284807.00 284807.00 mean 94813.86 0.00 0.00 -0.00 0.00 0.00 ... 0.00 0.00 -0.00 -0.00 -0.00 88.35 0.00 std 47488.15 1.96 1.65 1.52 1.42 1.38 ... 0.52 0.48 0.40 0.33 250.12 0.04 min 0.00 -56.41 -72.72 -48.33 -5.68 -113.74 ... -10.30 -2.60 -22.57 -15.43 0.00 0.00 25% 54201.50 -0.92 -0.60 -0.89 -0.85 -0.69 ... -0.32 -0.33 -0.07 -0.05 5.60 0.00 50% 84692.00 0.02 0.07 0.18 -0.02 -0.05 ... 0.02 -0.05 0.00 0.01 22.00 0.00 75% 139320.50 1.32 0.80 1.03 0.74 0.61 ... 0.35 0.24 0.09 0.08 77.16 0.00 max 172792.00 2.45 22.06 9.38 16.88 34.80 ... 7.52 3.52 31.61 33.85 25691.16 1.00 [8 rows x 31 columns]

Shape of Fraudulant transactions: (492, 31)

Shape of Non-Fraudulant transactions: (284315, 31)

Fraudulant transaction weight: 0.0017994745785028623

Non-Fraudulant transaction weight: 0.9982005254214972

TRAINING: X_train: (159491, 30), y_train: (159491,)

TESTING: X_test: (85443, 30), y_test: (85443,)

VALIDATION: X_validate: (39873, 30), y_validate: (39873,)

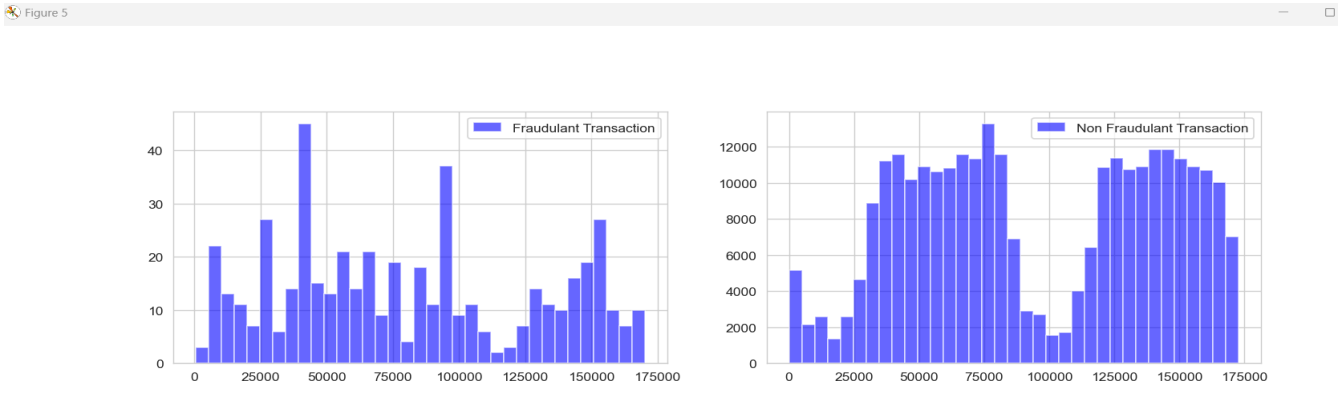
[0.4729713499546051, 23.0, 172.0, 85135.0, 113.0, 0.39649122953414917, 0.8308823704719543]

{'Train Result': {'Accuracy score: 99.75%'}}

{'Test Result': {'Accuracy score: 99.77%'}}

{'ANNs': {'Train': 0.5296167247386759, 'Test': 0.5368171021377672}}

PLOT DIAGRAMS



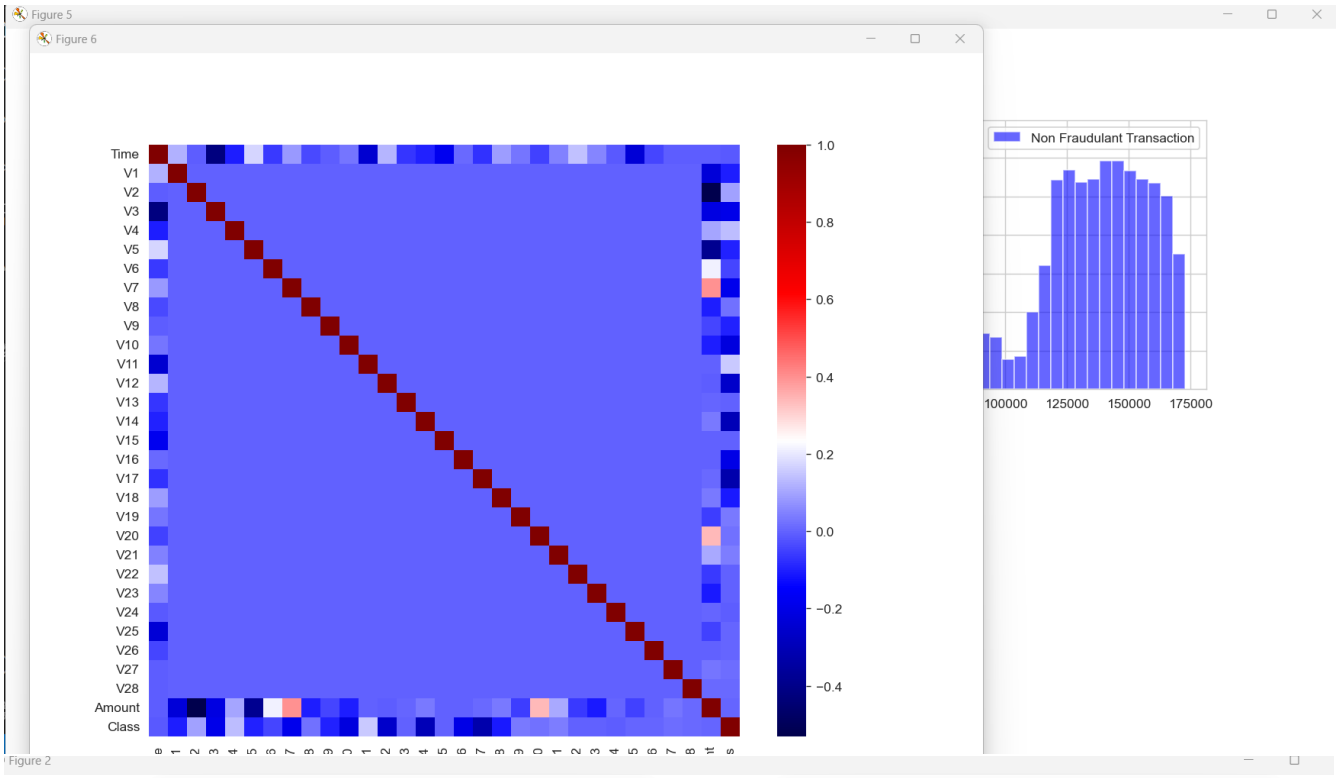


Figure 5
Figure 6
Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
Class
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
Non Fraudulant Transaction
100000
125000
150000
175000
Figure 2

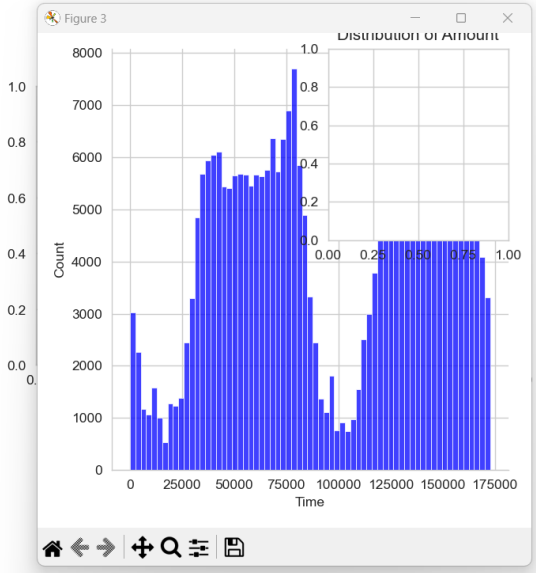


Figure 3
Distribution of Amount
Count
Time
0
25000
50000
75000
100000
125000
150000
175000
1000
2000
3000
4000
5000
6000
7000
8000
0.00
0.25
0.50
0.75
1.00

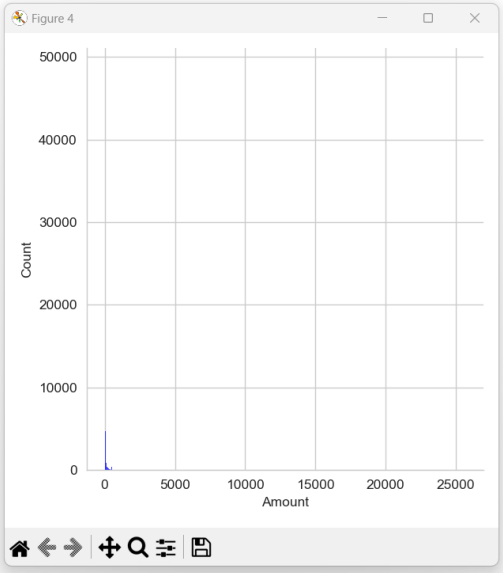


Figure 4
Count
Amount
0
5000
10000
15000
20000
25000
30000
40000
50000

BACKEND SCREENSHOTS

```
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
0
Index(['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',
```

```
d:\Projects\Credit_card_fraud_detection_v1.0\credit_card_fraud_detection_ANN_XGBoost_randomforest\deploy.py:62: UserWarning: Starting a Matplotlib G
UI outside of the main thread will likely fail.
  plt.figure(figsize=(14, 12))
d:\Projects\Credit_card_fraud_detection_v1.0\credit_card_fraud_detection_ANN_XGBoost_randomforest\deploy.py:71: UserWarning: Starting a Matplotlib G
UI outside of the main thread will likely fail.
  plt.figure(figsize=(10,10))
d:\Projects\Credit_card_fraud_detection_v1.0\credit_card_fraud_detection_ANN_XGBoost_randomforest\deploy.py:73: UserWarning: Starting a Matplotlib G
UI outside of the main thread will likely fail.
  plt.show()
Fraudulant transaction weight: 0.0017994745785028623
Non-Fraudulant transaction weight: 0.9982005254214972
TRAINING: X_train: (159491, 30), y_train: (159491,)
_____
VALIDATION: X_validate: (39873, 30), y_validate: (39873,)
_____
TESTING: X_test: (85443, 30), y_test: (85443,)
2023-10-11 20:24:01.157061: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructio
ns in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the
 appropriate compiler flags.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 256)               7936

 batch_normalization (Batch  (None, 256)               1024
 Normalization)

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 256)               65792

 batch_normalization_1 (Bat  (None, 256)               1024
 chNormalization)

 dropout_1 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 256)               65792
```

```
.....] - ETA: 4s - loss: 0.9198 - fn: 5.0000 - fp: 2855.0000 - tn: 3277.0000 - tp: 7.0000 - precision: 0.00 4/78 [>.....
.....] - ETA: 3s - loss: 0.9087 - fn: 5.0000 - fp: 3807.0000 - tn: 4366.0000 - tp: 14.0000 - precision: 0.0 5/78 [>.....
.] - ETA: 3s - loss: 0.9040 - fn: 5.0000 - fp: 4771.0000 - tn: 5448.0000 - tp: 16.0000 - precision: 0.0 7/78 [=>.....] - ETA:
 3s - loss: 0.9008 - fn: 9.0000 - fp: 6711.0000 - tn: 7598.0000 - tp: 18.0000 - precision: 0.0 9/78 [==>.....] - ETA: 3s - los
s: 0.8945 - fn: 13.0000 - fp: 8699.0000 - tn: 9698.0000 - tp: 22.0000 - precision: 0.11/78 [===>.....] - ETA: 3s - loss: 0.8898
 - fn: 16.0000 - fp: 10647.0000 - tn: 11836.0000 - tp: 29.0000 - precision: 13/78 [====>.....] - ETA: 3s - loss: 0.8850 - fn: 20
.0000 - fp: 12594.0000 - tn: 13978.0000 - tp: 32.0000 - precision: 15/78 [====>.....] - ETA: 2s - loss: 0.8806 - fn: 22.0000 - f
p: 14506.0000 - tn: 16155.0000 - tp: 37.0000 - precision: 17/78 [=====>.....] - ETA: 2s - loss: 0.8776 - fn: 23.0000 - fp: 16494.
0000 - tn: 18261.0000 - tp: 38.0000 - precision: 19/78 [======>.....] - ETA: 2s - loss: 0.8731 - fn: 26.0000 - fp: 18486.0000 - tn
: 20357.0000 - tp: 43.0000 - precision: 20/78 [======>.....] - ETA: 2s - loss: 0.8694 - fn: 26.0000 - fp: 19429.0000 - tn: 21456.0
000 - tp: 49.0000 - precision: 21/78 [=======>.....] - ETA: 2s - loss: 0.8673 - fn: 26.0000 - fp: 20419.0000 - tn: 22511.0000 - tp:
 52.0000 - precision: 23/78 [=======>.....] - ETA: 2s - loss: 0.8642 - fn: 27.0000 - fp: 22371.0000 - tn: 24651.0000 - tp: 55.0000
 - precision: 24/78 [========>.....] - ETA: 2s - loss: 0.8616 - fn: 27.0000 - fp: 23345.0000 - tn: 25720.0000 - tp: 60.0000 - precisi
on: 25/78 [========>.....] - ETA: 2s - loss: 0.8606 - fn: 29.0000 - fp: 24308.0000 - tn: 26802.0000 - tp: 61.0000 - precision: 26/78
 [=========>.....] - ETA: 2s - loss: 0.8594 - fn: 29.0000 - fp: 25265.0000 - tn: 27890.0000 - tp: 64.0000 - precision: 27/78 [=======
==>.....] - ETA: 2s - loss: 0.8587 - fn: 29.0000 - fp: 26248.0000 - tn: 28952.0000 - tp: 67.0000 - precision: 28/78 [========>.....
.....] - ETA: 2s - loss: 0.8569 - fn: 30.0000 - fp: 27232.0000 - tn: 30009.0000 - tp: 73.0000 - precision: 29/78 [=========>.....
.....] - ETA: 2s - loss: 0.8556 - fn: 31.0000 - fp: 28211.0000 - tn: 31074.0000 - tp: 76.0000 - precision: 30/78 [=========>.....] -
ETA: 2s - loss: 0.8536 - fn: 31.0000 - fp: 29162.0000 - tn: 32168.0000 - tp: 79.0000 - precision: 32/78 [==========>.....] - ETA: 2s -
 loss: 0.8506 - fn: 33.0000 - fp: 31075.0000 - tn: 34346.0000 - tp: 82.0000 - precision: 34/78 [==========>.....] - ETA: 2s - loss: 0.
8478 - fn: 37.0000 - fp: 32975.0000 - tn: 36530.0000 - tp: 90.0000 - precision: 36/78 [==========>.....] - ETA: 2s - loss: 0.8453 - fn
: 39.0000 - fp: 34881.0000 - tn: 38714.0000 - tp: 94.0000 - precision: 38/78 [==========>.....] - ETA: 1s - loss: 0.8433 - fn: 40.0000
 - fp: 36853.0000 - tn: 40833.0000 - tp: 98.0000 - precision: 40/78 [==========>.....] - ETA: 1s - loss: 0.8402 - fn: 43.0000 - fp: 38
704.0000 - tn: 43068.0000 - tp: 105.0000 - precision:42/78 [==========>.....] - ETA: 1s - loss: 0.8379 - fn: 46.0000 - fp: 40548.0000
 - tn: 45312.0000 - tp: 110.0000 - precision:44/78 [==========>.....] - ETA: 1s - loss: 0.8355 - fn: 48.0000 - fp: 42398.0000 - tn: 475
53.0000 - tp: 113.0000 - precision:46/78 [==========>.....] - ETA: 1s - loss: 0.8324 - fn: 49.0000 - fp: 44236.0000 - tn: 49808.0000 -
 tp: 115.0000 - precision:48/78 [==========>.....] - ETA: 1s - loss: 0.8305 - fn: 49.0000 - fp: 46124.0000 - tn: 52008.0000 - tp: 123.
0000 - precision:49/78 [==========>.....] - ETA: 1s - loss: 0.8295 - fn: 51.0000 - fp: 47041.0000 - tn: 53132.0000 - tp: 128.0000 - pr
ecision:51/78 [==========>.....] - ETA: 1s - loss: 0.8269 - fn: 52.0000 - fp: 48851.0000 - tn: 55412.0000 - tp: 133.0000 - precision:5
3/78 [==========>.....] - ETA: 1s - loss: 0.8239 - fn: 53.0000 - fp: 50630.0000 - tn: 57717.0000 - tp: 144.0000 - precision:55/78 [===
==========>.....] - ETA: 1s - loss: 0.8217 - fn: 54.0000 - fp: 52412.0000 - tn: 60025.0000 - tp: 149.0000 - precision:57/78 [==========
=======>.....] - ETA: 1s - loss: 0.8197 - fn: 55.0000 - fp: 54190.0000 - tn: 62335.0000 - tp: 156.0000 - precision:59/78 [==========
>.....] - ETA: 0s - loss: 0.8174 - fn: 55.0000 - fp: 55983.0000 - tn: 64629.0000 - tp: 165.0000 - precision:61/78 [==========>.....
] - ETA: 0s - loss: 0.8157 - fn: 56.0000 - fp: 57797.0000 - tn: 66906.0000 - tp: 169.0000 - precision:63/78 [==========>.....] - ETA:
0s - loss: 0.8137 - fn: 57.0000 - fp: 59574.0000 - tn: 69221.0000 - tp: 172.0000 - precision:65/78 [==========>.....] - ETA: 0s - loss
: 0.8121 - fn: 57.0000 - fp: 61290.0000 - tn: 71595.0000 - tp: 178.0000 - precision:67/78 [==========>.....] - ETA: 0s - loss: 0.8101
 - fn: 57.0000 - fp: 63083.0000 - tn: 73891.0000 - tp: 185.0000 - precision:69/78 [==========>....] - ETA: 0s - loss: 0.8085 - fn: 58.
0000 - fp: 64873.0000 - tn: 76190.0000 - tp: 191.0000 - precision:71/78 [==========>...] - ETA: 0s - loss: 0.8067 - fn: 58.0000 - fp
: 66643.0000 - tn: 78510.0000 - tp: 197.0000 - precision:73/78 [==========>..] - ETA: 0s - loss: 0.8042 - fn: 59.0000 - fp: 68335.0
000 - tn: 80906.0000 - tp: 204.0000 - precision:75/78 [==========>..] - ETA: 0s - loss: 0.8020 - fn: 62.0000 - fp: 70042.0000 - tn:
```

```
  plt.figure(figsize=(12, 16))
4985/4985 [==============================] - 9s 2ms/step
2671/2671 [==============================] - 5s 2ms/step
Train Result:
================================================
Accuracy Score: 99.75%

------------------------------------------------
Classification Report:
                    0      1  accuracy  macro avg  weighted avg
precision        1.00   0.40      1.00       0.70          1.00
recall           1.00   0.79      1.00       0.90          1.00
f1-score         1.00   0.53      1.00       0.76          1.00
support     159204.00 287.00      1.00  159491.00     159491.00

------------------------------------------------
Confusion Matrix:
 [[158858    346]
 [    59    228]]

Test Result:
================================================
Accuracy Score: 99.77%

------------------------------------------------
Classification Report:
                    0      1  accuracy  macro avg  weighted avg
precision        1.00   0.40      1.00       0.70          1.00
recall           1.00   0.83      1.00       0.91          1.00
f1-score         1.00   0.54      1.00       0.77          1.00
support     85307.00 136.00      1.00   85443.00      85443.00

------------------------------------------------
Confusion Matrix:
 [[85135   172]
 [   23   113]]

127.0.0.1 - - [11/Oct/2023 20:24:42] "GET /predict HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [11/Oct/2023 20:24:42] "GET /predict HTTP/1.1" 200 -
```

10.2 SAMPLE CODE:

depploy.py

```
from flask import Flask,render_template,request
import pickle
import sys
app = Flask(__name__)

@app.route("/")
```

```
def home():
    return render_template("home.html")

@app.route("/predict",methods=["GET"])
def predict():
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    import pickle
    from keras.callbacks import ModelCheckpoint

    sns.set_style("whitegrid")
    data = pd.read_csv("creditcard.csv")
    head = str(data.head())


    #data analysis#
    data.info()
    pd.set_option("display.float", "{:.2f}".format)
    describe = data.describe()


    #checking for missing values#
    print(data.isnull().sum().sum())
    print(data.columns)
    LABELS = ["Normal", "Fraud"]
    count_classes = pd.value_counts(data['Class'], sort = True)
    count_classes.plot(kind = 'bar', rot=0)
    plt.title("Transaction Class Distribution")
    plt.xticks(range(2), LABELS)
```

```
plt.xlabel("Class")
plt.ylabel("Frequency")
print(data.Class.value_counts())
fraud = data[data['Class']==1]
normal = data[data['Class']==0]
print(f"Shape of Fraudulant transactions: {fraud.shape}")
fraudtxn = str(f"Shape of Fraudulant transactions: {fraud.shape}")
print(f"Shape of Non-Fraudulant transactions: {normal.shape}")
normaltxn = str(f"Shape of Non-Fraudulant transactions: {normal.shape}")
print(pd.concat([fraud.Amount.describe(), normal.Amount.describe()], axis=1))
print(pd.concat([fraud.Time.describe(), normal.Time.describe()], axis=1))

# plot the time feature
plt.figure(figsize=(14,10))
plt.subplot(2, 2, 1)
plt.title('Time Distribution (Seconds)')
sns.displot(data['Time'], color='blue')

#plot the amount feature
plt.subplot(2, 2, 2)
plt.title('Distribution of Amount')
sns.displot(data['Amount'],color='blue')

# data[data.Class == 0].Time.hist(bins=35, color='blue', alpha=0.6)
plt.figure(figsize=(14, 12))
plt.subplot(2, 2, 1)
data[data.Class == 1].Time.hist(bins=35, color='blue', alpha=0.6, label="Fraudulant Transaction")
plt.legend()
plt.subplot(2, 2, 2)
data[data.Class == 0].Time.hist(bins=35, color='blue', alpha=0.6, label="Non Fraudulant
Transaction")
```

```
    plt.legend()

    # heatmap to find any high correlations
    plt.figure(figsize=(10,10))
    sns.heatmap(data=data.corr(), cmap="seismic")
    plt.show()

    #data preprocessing
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    scalar = StandardScaler()
    X = data.drop('Class', axis=1)
    y = data.Class
    X_train_v, X_test, y_train_v, y_test = train_test_split(X, y,test_size=0.3, random_state=42)
    X_train, X_validate, y_train, y_validate = train_test_split(X_train_v, y_train_v,test_size=0.2,
random_state=42)
    X_train = scalar.fit_transform(X_train)
    X_validate = scalar.transform(X_validate)
    X_test = scalar.transform(X_test)
    w_p = y_train.value_counts()[0] / len(y_train)
    w_n = y_train.value_counts()[1] / len(y_train)
    print(f"Fraudulant transaction weight: {w_n}")
    fraudwg = str(f"Fraudulant transaction weight: {w_n}")
    print(f"Non-Fraudulant transaction weight: {w_p}")
    normalwg = str(f"Non-Fraudulant transaction weight: {w_p}")

    print(f"TRAINING: X_train: {X_train.shape}, y_train: {y_train.shape}\n{'_'*55}")
    trainx = str(f"TRAINING: X_train: {X_train.shape}, y_train: {y_train.shape}")
    print(f"VALIDATION: X_validate: {X_validate.shape}, y_validate: {y_validate.shape}\n{'_'*50}")
    validx = str(f"VALIDATION: X_validate: {X_validate.shape}, y_validate: {y_validate.shape}")
    print(f"TESTING: X_test: {X_test.shape}, y_test: {y_test.shape}")
```

```
testx = str(f"TESTING: X_test: {X_test.shape}, y_test: {y_test.shape}")

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, f1_score

def print_score(label, prediction, train=True):
    if train:
        clf_report = str(pd.DataFrame(classification_report(label, prediction, output_dict=True)))
        print("Train Result:\n================================================")
        print(f"Accuracy Score: {accuracy_score(label, prediction) * 100:.2f}%")
        print("_____________________________________________")
        print(f"Classification Report:\n{clf_report}")
        print("_____________________________________________")
        print(f"Confusion Matrix: \n {confusion_matrix(y_train, prediction)}\n")

    elif train==False:
        clf_report = str(pd.DataFrame(classification_report(label, prediction, output_dict=True)))
        print("Test Result:\n================================================")
        print(f"Accuracy Score: {accuracy_score(label, prediction) * 100:.2f}%")
        print("_____________________________________________")
        print(f"Classification Report:\n{clf_report}")
        print("_____________________________________________")
        print(f"Confusion Matrix: \n {confusion_matrix(label, prediction)}\n")

#building ANN model#
from tensorflow import keras
model = keras.Sequential([
    keras.layers.Dense(256, activation='relu', input_shape=(X_train.shape[-1],)),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(256, activation='relu'),
    keras.layers.BatchNormalization(),
```

```
        keras.layers.Dropout(0.3),
        keras.layers.Dense(256, activation='relu'),
        keras.layers.BatchNormalization(),
        keras.layers.Dropout(0.3),
        keras.layers.Dense(1, activation='sigmoid'),
    ])
    model.summary()

    METRICS = [
    #     keras.metrics.Accuracy(name='accuracy'),
        keras.metrics.FalseNegatives(name='fn'),
        keras.metrics.FalsePositives(name='fp'),
        keras.metrics.TrueNegatives(name='tn'),
        keras.metrics.TruePositives(name='tp'),
        keras.metrics.Precision(name='precision'),
        keras.metrics.Recall(name='recall')
    ]
    model.compile(optimizer=keras.optimizers.Adam(1e-4), loss='binary_crossentropy',
metrics=METRICS)
    callbacks = ModelCheckpoint(filepath="model.pkl",verbose=0,save_best_only=True)
    class_weight = {0:w_p, 1:w_n}
    r = model.fit(
        X_train, y_train,
        validation_data=(X_validate, y_validate),
        batch_size=2048,
        epochs=3,
    #     class_weight=class_weight,
        callbacks=callbacks,
    )
    score = model.evaluate(X_test, y_test)
    mscore = str(score)
```

```
plt.figure(figsize=(12, 16))
plt.subplot(4, 2, 1)
plt.plot(r.history['loss'], label='Loss')
plt.plot(r.history['val_loss'], label='val_Loss')
plt.title('Loss Function evolution during training')
plt.legend()
plt.subplot(4, 2, 2)
plt.plot(r.history['fn'], label='fn')
plt.plot(r.history['val_fn'], label='val_fn')
plt.title('Accuracy evolution during training')
plt.legend()
plt.subplot(4, 2, 3)
plt.plot(r.history['precision'], label='precision')
plt.plot(r.history['val_precision'], label='val_precision')
plt.title('Precision evolution during training')
plt.legend()
plt.subplot(4, 2, 4)
plt.plot(r.history['recall'], label='recall')
plt.plot(r.history['val_recall'], label='val_recall')
plt.title('Recall evolution during training')
plt.legend()
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)
print_score(y_train, y_train_pred.round(), train=True)
print_score(y_test, y_test_pred.round(), train=False)
trscore = {
   'Train Result':{
      f'Accuracy score: {accuracy_score(y_train, y_train_pred.round()) * 100:.2f}%'
   }
}
tescore = {
```

```
        'Test Result':{
            f'Accuracy score: {accuracy_score(y_test, y_test_pred.round()) * 100:.2f}%'
        }
    }
    tesscore = str(tescore)
    trsscore = str(trscore)
    result = {
        'ANNs': {
            'Train': f1_score(y_train, y_train_pred.round()),
            'Test': f1_score(y_test, y_test_pred.round()),
        },
    }
    processed_text= str(result)
    return render_template("index.html",head = head,describe = describe,fraudtxn = fraudtxn,normaltxn
=normaltxn,fraudwg = fraudwg,
                           normalwg = normalwg,trainx = trainx,validx = validx,testx = testx,mscore =
mscore,processed_text = processed_text,
                           trsscore = trsscore,tesscore = tesscore)
if __name__ == '__main__':
    app.run(debug=True)
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        h1{text-align: center;}
        footer{text-align: center;}
```

```
        #btn{border: none;
        height: 50px;
    width: 150px;
border-radius: 5px;
background-color: gray;
color: white;
}
    </style>
    <title>ANN Model</title>
</head>
<body>
    <h1 id="h1">Major Project - Credit Card Fraud Detection</h1>
    <hr>
    <h2 id="h2">Description</h2>
    <p id="desc">Credit card fraud is when someone uses another person's credit card or account information to make unauthorized purchases or access funds through cash advances.
        Credit card fraud doesn’t just happen online; it happens in brick-and-mortar stores, too.
        As a business owner, you can avoid serious headaches – and unwanted publicity – by recognizing potentially fraudulent use of credit cards in your payment environment.</p>
      <br>
      <hr>
      <br>
        <h2 id="h2">Challenges</h2>
        <p id="desc2">The challenge is to recognize fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase.</p>
        <p id="desc3"><b>Main challenges involved in credit card fraud detection are:</b><br>
            Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time.
            Imbalanced Data i.e most of the transactions (99.8%) are not fraudulent which makes it really hard for detecting the fraudulent ones.
```

Data availability as the data is mostly private. Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported.

Adaptive techniques used against the model by the scammers. The model used must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible.

We can make the model simple and interpretable so that when the scammer adapts to it with just some changes we can have a new model up and running to deploy.

<hr>

<h2 id="h2">Modules</h2>

Collection of Dataset

Checking the Dataset for empty values and Noisy or unwanted data

Splitting the data into Training data and Test data

Training the models and choosing the model with highest accuracy raio

Printing the predicted values with accuracy ratio

<hr>

<h2 id="h2">Observations</h2>

Very few transactions are actually fraudulent (less than 1%). The data set is highly skewed, consisting of 492 frauds in a total of 284,807 observations. This resulted in only 0.172% fraud cases. This skewed set is justified by the low number of fraudulent transactions.

The dataset consists of numerical values from the 28 ‘Principal Component Analysis (PCA)’ transformed features, namely V1 to V28 And my predictions is based on two values namely time and amount.

There is no missing value in the dataset.


```
        <hr>
        <br>
        <h2 id="work">Access my Work By Clicking The Button Below</h2>
        <form action="/predict" method="GET">
        <button type="submit" id="btn"><b>Access My Work</b></button>
        </form>
        <br>
        <hr>
</body>
<footer>This Project Was Done By - Sanjay K (20MSS040)</footer>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>ANN Model</title>
   <style>
      body{background-color: black;
      color: white;}
      h1{text-align: center;}
   </style>
</head>
<body>
   <h1 class="heading">Credit Card Fraud Detection</h1>
   <br>
   <hr>
   <br>
```

```
    <p id="print1">{{head}}</p>
    <p id="print2">{{describe}}</p>
    <p id="print3">{{fraudtxn}}</p>
    <p id="print4">{{normaltxn}}</p>
    <p id="print5">{{fraudwg}}</p>
    <p id="print6">{{normalwg}}</p>
    <p id="print7">{{trainx}}</p>
    <p id="print8">{{testx}}</p>
    <p id="print9">{{validx}}</p>
    <p id="print10">{{mscore}}</p>
    <p id="print11">{{trsscore}}</p>
    <p id="print12">{{tesscore}}</p>
        {% if processed_text %}
<p id="text">{{ processed_text }}</p>
{% endif %}
<hr>
</body>
</html>
```

SYNOPSIS

The project entitled "**Credit Card Fraud detection**" in which user can be able to view the findings of the Machine learning Algorithm from their place through the Internet. Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch. And the Machine learning Algorithm can be uploaded to a website enables the users to easily have an access over all available algorithms through internet.
Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold.

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity

SANJAY K (20MSS040), Fourth year M.Sc Software systems Batch (2020-2025) in PSG college of Arts & science, Coimbatore.