**CHAPTER-1**

**INTRODUCTION**

**1.1 OVERVIEW**

Credit cards are now the most preferred way for customers to transact either offline or online. Lucrative cashback and reward point options are present for each credit card transaction. These are generally not offered by financial institutions for debit cards. Tie up credit cards with online and offline merchants, especially during festive seasons like Diwali, Eid, and Christmas, to offer further discounts on transactions. Immediate needs can be fulfilled (for example, medical emergencies, lifetime events, etc.) quickly instead of having sufficient account balance for the same. Credit cards are tailored to suit individual customer needs.

Credit card is the most popular mode of payment. As the number of credit card users is rising world-wide, the identity theft is increased, and frauds are also increasing. In the virtual card purchase, only the card information is required such as card number, expiration date, secure code, etc. Such purchases are normally done on the Internet or over telephone. To commit fraud in these types of purchases, a person simply needs to know the card details. The mode of payment for online purchase is mostly done by credit card. The details of credit card should be kept private. To secure credit card privacy, the details should not be leaked.

Different ways to steal credit card details are phishing websites, steal/lost credit cards, counterfeit credit cards, theft of card details, intercepted cards etc. For security purpose, the above things should be avoided. In online fraud, the transaction is made remotely and only the card’s details are needed. A manual signature, a PIN or a card imprint are not required at the purchase time. In most of the cases the genuine cardholder is not aware that someone else has seen or stolen his/her card information. The simple way to detect this type of fraud is to analyse the spending patterns on every card and to figure out any variation to the “usual” spending patterns. Fraud detection by analysing the existing data purchase of cardholder is the best way to reduce the rate of successful credit card frauds. As the data sets are not available and also the results are not disclosed to the public. The fraud cases should be detected from the available data sets known as the logged data and user behaviour.

**1.2 NEED FOR FRAUD DETECTION**

At present, fraud detection has been implemented by a number of methods such as data mining, statistics, and artificial intelligence. Number of fraudulent people exponentially increase with the increase in credit card users. Such wrong intensive people have an increasing concern about other person's environment and their activities, sometimes accompany the person who is false in management responsibility, and in many different CC agents or arrangements. In the realm, the extreme reliance connected with the rising illegal use of CC is due to the internet using people who perform deceitful undertakings.

However, the deception is not only connected with the internet but also with offline business dealings. Though stealing the information is in the visible form of excavating method [6], the data required to expose the frauds of CC are insufficient. The foremost thing to break the habit of misplacing or losing essentials. To discover the CC deception, the use of adept algorithms exists. The cash reserve may be one of the ways to figure out the lost. When meeting expectations to receive a CC, and filling up the piece of information, the cardholder will have to pay the original amount in addition to the supplementary charges they consent to pay. A CC deception happens when an additional human being uses other person’s CC to make an unauthorized correction. Theft circumvents the need of CC PIN for the charge to perform one of the permitted activities. Using the CC trickery discovery, it is possible to discover either the new transaction is by a deceitful individual or the actual person.

In the financial business, CC fraud is an ever-increasing threat with far-reaching implications. The sampling strategy on the dataset, the allocation of attributes, and the screening methods utilized, all have a significant impact on the performance of fraud detection in CC transactions. CC usage has already skyrocketed, as the world rapidly moves toward digitalization and paperless money transfers. It has also increased the number of fraudulent activities related to it, resulting in significant losses for credit intermediaries. People have been more concerned about fraud detection models based on machine learning recently. Traditional data mining methods aren't immediately solution to the problem because it's handled as a classification challenge. This research aims to expand a directed machine intelligence treasure for detecting CC deceitful act. Supervised algorithms give the impression of repetitive algorithms that exert themselves to make or improve their results over time. As the number of human beings who use credit cards evolves across the planet, so does the risk of theft.

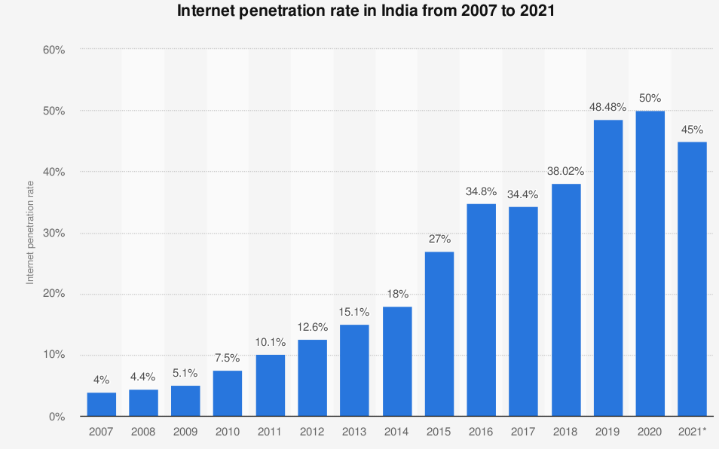


Figure 1.1 Growth of internet usage

Figure 1.1 depicts that the internet usage increases annually which is sited as per statics. When purchasing directly, only the details with purposeful facts, such as the info with required number, demise date, and freedom law should be used. Typically, specific purchases happen through the Internet as the usage of internet has increased nowadays. Because of the widespread usage of the computer network, the growth of online shopping has increased. Figure 1.2 illustrates the growth of e-commerce websites as per statics. Hence, there are several opportunities for fraudsters to perpetrate CC fraud.

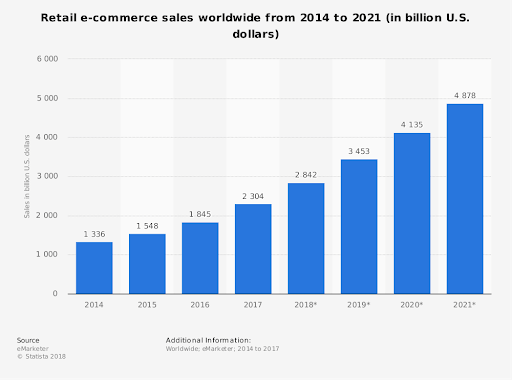


Figure 1.2 Growth of E-commerce websites

A person just has to be familiar with the information in paper to conduct deception in this sort of undertaking. The most common form of fee for computer network purchases is through credit cards. CC facts concede the possibility that the details may exist as secret. CC information in the visible form is not made public to guard against harm. Phishing websites that take CC analysis, and other systems are used to acquire CC facts. For reasons of liberty, they acknowledge the possibility of a happening, but refrain from it. The business dealings that are approved, may be connected to the internet scams, and hence with resolute writing, information in the visible form is often necessary.

Furthermore, while purchasing, a manual sign, a PIN, or a piece of paper with the imprint is optional. In most positions, the cardholders are ignorant that their piece of paper bearing essentials, happened to be visualized or captured by another person. The simple method to label this form of deceitful work is to examine each piece of paper's past events and look for some deviation by the "sane" fashion. The greatest plan of action to lower the rate of favourable CC deceit, is to recognize the wrong person by analysing the cardholder's existent information in written. Because the information in visible form sets isn’t ready for use immediately, and thus concede the possibility of being used to discover a person who is deceptive. Currently, deception discovery happens utilizing different methods such as, verifying information in visible form, excavating, enumeration, and machine intelligence.

E-commerce and many different connections to the internet sites bear the extension of the number of fee-based alternatives ready for use connected to the internet, lifting the troublesome situation of being connected to the internet trickery. Because of the rise in popular deception rates, researchers began experimenting with various machine intelligence approaches to detect and analyse internet businesses dealing with deception. The paper's main purpose is to build and develop in mind or physically, a singular person who is a false discovery approach for the flow of business dealing information in visible form and has in mind to analyse services recorded as actually having happened in business dealing facts and extract behavioural patterns. Cardholders exist separated into types to establish the capacity of their business dealings. Then, utilizing the move smoothly framework with the pane approach [1], they aggregate the undertaking act by cards from miscellaneous types to derive a conclusion on the behavioural patterns of the miscellaneous groupings. Different classifiers [3], [5], [6], and [8] segregates before being prepared separately for the groups. The classifier associated with the highest-grade score can thus be chosen all at once, eventually in use at the time when it is customarily used to express an outcome in advance to a false person. As a result, a response method exists that can be used to overcome the something understood drift question [1].

The deceitful, concede the possibility of drawing in the transaction with purposeful writing to a CC for cashless shopping. CCs are a neat mark for trickery. In the recovery period, plenty of services may make money, many risks may exist, and even offences against the law will take many weeks to discover.

Buying sites are the most common places where people are deceived in the world of fashion. Human beings live to engage in the act of purchasing products connected to the internet, so instead of going out and buying goods from the shops, the tumour of e-commerce sites is expanding, and there is an exceedingly high risk of CC fraud. So, in order to prevent particular CC fraud, the best solution for reducing CC fraud has to be determined or invented.

**1.3 TYPES OF CREDIT CARD FRAUD**

Credit card fraud is when someone uses your credit card or account information to make purchases without your permission. It generally gets broken into two categories.

* In-person fraud, or card-present fraud, is when someone steals your card, creates a counterfeit card with your account information or otherwise uses your account information for an unauthorized transaction while they're at a merchant.
* Remote fraud, or card-not-present fraud, is any other situation when someone fraudulently uses your credit card account to make a purchase, such as shopping online.

Once you report an unauthorized transaction, the credit card company may work with you to confirm it's a case of credit card fraud rather than a simple mistake. For example, a merchant overcharging for a purchase you made or failing to deliver a product is not necessarily credit card fraud. You may be able to initiate a chargeback and get refunded—but you wouldn't go through your card issuer's fraud channels to do so.

If you are a victim of credit card fraud, the federal Fair Credit Billing Act (FCBA) limits your liability to no more than $50 for unauthorized charges. However, American Express, Discover, Mastercard and Visa go one step further and bring that liability down to $0 on consumer credit cards.

**1.4 WHAT FRAUD PROTECTION FEATURES DO CREDIT CARDS PROVIDE?**

Preventing credit card fraud can help save merchants and credit card issuers money, build trust among cardholders and keep you from having to wait for a new card. In short, it's a win-win for everyone.

Credit card issuers use a variety of measures to stop fraud from happening. These can range from physical features built into your card to complex artificial intelligence systems that detect and decline unusual transactions (a high-dollar purchase made at a store hundreds of miles from where you live, for instance).

As a cardholder, you could look for a card or issuer that offers:

* EMV chips: Cards with EMV chips are now fairly standard. EMV chips can add an extra level of protection compared to swiping a card's magnetic strip, but they may still be susceptible to card shimming—when a device gets put into a card terminal and copies your card's information while it's inserted.
* Contactless cards: Tapping a card or using a mobile device with a digital wallet can be even safer than swiping or inserting your card. Many popular credit cards from major issuers come with contactless payments enabled and work with popular digital wallets.
* Virtual card numbers: Credit card issuers may let you create a virtual card number to use when shopping online, keeping your card's actual information a secret. You can find this feature on some Citi and Capital One cards, including the Capital One Venture Rewards Credit Card.
* Card lock: Locking a credit card can temporarily stop it from being used for new purchases, which can be nice if you can't find your card but don't want to go through the process of cancelling it and waiting for a replacement. It's sometimes called "freezing" rather than locking, which lets you temporarily freeze your account online or with the mobile app.

There are also credit card fraud prevention measures that could be taking place without you noticing.

For example, before a transaction gets approved, it may be assigned a risk score based on the time of day, transaction amount, card's transaction history, the location of your mobile phone and other variables. The merchant can decide whether to approve or deny transactions depending on their risk scores. And online purchases may be scrutinized based on additional information, such as the purchaser's IP address, email host, shipping address and order details.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 A Comparative Analysis of Various CC Fraud Detection Techniques**

**Author:** Yashvi Jain, Namrata Tiwari, Shripriya Dubey, Sarika Jain

**Year:** 2019

**Description:**

Yashvi Jain, et al., examined differing methods for CC deceitful discovery to a degree that supports heading machines supervised algorithms. Paper, states that the algorithms KNN in time neighbour, determination of large plants enclosed in bark and shedding leaves, and the SVM presented medium-level precision or correctness. Fuzzy Logic and Logistic Regression presented hostile precision or correctness between all the additional algorithms. Neural Networks, Naive Bayes, out of focus structure, and KNN offered an extreme confinement rate. The Logistic Regression, SVM, and resolution of large plants enclosed in bark and shedding leaves offered a rate at the mid stage. These are extremely expensive to train. There is a bigger disadvantage that these algorithms, not all have the same impact on every type of environment They provide better outcomes with certain types of datasets and poorer results with others. Methods like KNN, SVM produced fantastic outcomes with little datasets, whereas algorithms like logistic reversion and fluffy science of reasoning organization produce high precision or accuracy with untested and unsampled data in visible form.

**2.2 Algorithms of ML for CC Fraud Detection**

**Author:** Heta Naik and Prashasti Kanikar

**Year:** 2019

**Description:**

Prashasti Kanikar, et al., accomplished their research in contact with differing algorithms [4] which is the inventor of categorization. This invention depends upon Bayes explanation based on hypotheses and experiments that finds the likelihood of something happening is likely. The Logistic reversion treasure exists very much like the uninterrupted reversion treasure. The continuous reversion exists second hand for the advance declaration or guessing the principles. The treasure is, for the most part, used to boost the conduct of the resolution, reaching a large plant enclosed in bark and shedding leaves. This happens in addition to, for the most part, second hand for the categorization of the reversion. The Adaboost invention happens to a person who is false to categorize the business dealings that happen between a false person and a non-false person. Based on their findings, they concluded that the highest precision or correctness occurs when two Adaboost and Logistic Regression are used together. It was deliberated to select the best choice treasure because they bear the same precision or correctness moment of truth, and determinant. By taking everything in mind at the moment of truth, they decide that the Adaboost invention was everything they needed to discover about CC trickery.

**2.3 Credit Card Fraud Detection using Machine Learning Algorithms**

**Author:** S, Varun

**Year:** 2020

**Description:**

The point of the paper was to search for and amplify a novel deception discovery method for streaming transaction data, accompanying a mark, to try the former business dealing delicacy of the customer and collect the conduct patterns. Then the move smoothly framework was used with a pane total amount to the business dealing fashioned individually by cardholders from various gatherings, so that the standard of conduct of the gatherings can be extracted separately. Later, various classifiers were prepared in body or mind over the gatherings alone. And following the classifier's better judge score, they chose an expected, possibly high-quality method to expect fraud.

**2.4 Deep Convolution Neural Network Model for Credit-Card Fraud Detection and Alert**

**Author:** Chen, Joy Iong-Zong, and Kong-Long Lai

**Year:** 2021

**Description:**

DCNN used in paper had an accuracy of 99℅ in detecting the CC frauds when compared to Autoencoder used in existing methods.

**2.5 A New Framework for Credit Card Transactions Involving Mutual Authentication between Cardholder and Merchant**

**Author:** Gupta, Shalini, and R. Johari

**Year:** 2021

**Description:**

The model defeated and avoided the prevalent fraud associated with internet business transactions involving CCs. Before embarking on a swindle labelling strategy for a fashionable CC venture, an audit was conducted. As pointed out, one of these parts for swindle means of labelling is right next to a place where 80 heaps of computer network business dealing by credit card was pre-obviously deceitful and legal. The model may be ready in body or mind with a correct habit through a simple arithmetical process of increasing new facial characteristics. Several visible information systems for accomplishing something of critical nature are used by bank and credit cards arranging for various deception practices. The conventional use of instances of services depended on their past exercises and maybe outstanding by putting into use one of these plans of action.

**2.6 Credit card fraud detection using machine learning**

**Author:** Thirunavukkarasu.M et al

**Year:** 2021

**Description:**

The something given of various methods exists to determine the use of a sure demonstration calculation that points out the projected approach’s output. When these implements are lent out, this brings the appraisal of the something given to any try plan in contact with the classifier, deception basic document file, accompanying class lack of balance. These key parts of a thorough check (PCA) happen to be the honest information in a visible form while the determinant period, total, and class to succeed in doing 28 principal parts of something that happens organized.

**2.7 Credit Card Fraud Detection Using Machine Learning**

**Author:** R. Sailusha, V. Gnaneswar, R. Ramesh and G. R. Rao

**Year:** 2020

**Description:**

The investigation focused on fashion in general rather than the specific charge card person, who is a falsely fashionable real class of existing beings. The credit card information must be drawn in visible form at the beginning of the basic document file. The dictatorial forest invention categorization method makes use of the versatile determining information in visible form sets and bestows the current basic document file. Finally, the correct and effective information in visible form has been modernized.

**2.8 Credit Card Fraud Detection using Machine Learning and Deep Learning Techniques**

**Author:** M. Azhan and S. Meraj

**Year:** 2020

**Description:**

The approach said that bearing witness to the plan of fraud is by putting oneself in the place of another CC. Various machine intelligence algorithms were used in contact with an unstable dataset, e.g., logistic reversion, naive Bayes, and chance forest. Accompanying company classifiers, the push method was used. Different categorization models were used to represent the information in visible form and the model evaluated something given to establish determinable estimations, for instance, precision or correctness, accuracy, recall, f1 score, support, disorientation, something from which another originates. The end of the test demonstrated a high-quality classifier by the way of preparation and experiment, making use of a directed process.

**2.9 Credit Card Fraud Detection using Deep Learning**

**Author:** A. M. Babu and A. Pratap

**Year:** 2020

**Description:**

In this paper, they investigated the display of logistic reversion, a large plant encased in bark and shedding leaves, and the whimsical forest for credit card trickery detection. The dataset of CC business dealings was assembled by Kaggle, and it holds an aggregate of 2,84,808 charge card undertakings from a European bank's basic document file.

**CHAPTER-3**

**PROBLEM DEFINITON**

**3.1 EXISTING SYSTEM**

It is vital that credit card companies are able to identify fraudulent credit card transactions so that customers are not charged for items that they did not purchase. Such problems can be tackled with Data Science and its importance, along with Machine Learning, cannot be overstated. This project intends to illustrate the modelling of a data set using machine learning with Credit Card Fraud Detection. The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the data of the ones that turned out to be fraud. During ancient times, the use of buying bears existed from a small age to a large one. Due to that, the use of CC s has also been raised. Many human beings immediately use CC for connection to the internet buying, e-charging money for goods, and other connected to the internet fees. This frequent use of CC allows aggressive institutions and banks to implement CC trickery discovery structures to equate forbidden and valid business dealings. These arrangements bear witness to prepared fashionable pre-be living datasets and, at another time, used to the new undertaking. The classification findings in most of the approaches are biased towards the majority of classes. This model is then used to recognize whether a new transaction is fraudulent or not. Our objective here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications. Credit Card Fraud Detection is a typical sample of classification. In this process, we have focused on analysing and pre-processing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card Transaction data.

**CHAPTER 4**

**DESIGN OF PROPOSED SYSTEM**

**4.1 PROPOSED SYSTEM**

Credit card fraud is a serious criminal offense. It costs individuals and financial institutions billions of dollars annually. According to the reports of the Federal Trade Commission (FTC), a consumer protection agency, the number of theft reports doubled in the last two years. It makes the detection and prevention of fraudulent activities critically important to financial institutions. Machine learning algorithms provide a proactive mechanism to prevent credit card fraud with acceptable accuracy. In this project Machine Learning algorithms such as Logistic Regression, Naïve Bayes, Random Forest, K- Nearest Neighbour are implemented for detection of fraudulent transactions. A comparative analysis of these algorithms is performed to identify an optimal solution.

Diagram

Description automatically generated

Figure 4.1 Proposed Model

The major goal of this study is to use the MLP method with additional algorithms such as Gaussian Naive Baye, Bernoulli Naive Baye, and Random Forest to categorise the business deals in the dataset that have two things in common: deceit and non-deception business dealings. The partition into portions of the information in visible form, model preparation, model layout, and the judgement test are all part of the process flow for the CC fraud discovery inquiry [Figure.4.1].

Many actions are included in the defined structure of anything for the CC person who is a fake discovery plan [Figure 4.1], from building a dataset to establishing a system and participating in a presentation, as well as creating findings.

**4.2 SYSTEM SPECIFICATIONS**

**4.2.1 SOFTWARE**:

Operating System : Windows 7/8/10

Language : Python 3.7

Tools : Pandas, Numpy, Scikit, MatplotLib

**4.2.2 HARDWARE:**

Processor : Intel Core i3

RAM : 4GB

Hard Disk : 1TB

Mouse : logical optical mouse

Keyboard : logical 107 keys

Mother board : Intel

Speed : 3.3GHZ

**4.3 SOFTWARE SPECIFICATIONS**

**4.3.1 PYTHON**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for:

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.
* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.

Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**4.4 LIBRARIES USED**

**4.4.1 Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users. [matplotlib.pyplot](https://matplotlib.org/api/pyplot_api.html#module-matplotlib.pyplot) is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In [matplotlib.pyplot](https://matplotlib.org/api/pyplot_api.html" \l "module-matplotlib.pyplot) various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes.

**4.4.2  Numpy**

Numpy 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. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. It provides:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities and much more

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. All NumPy wheels distributed on PyPI are BSD licensed. ndarray.shape is the array attribute that returns a tuple consisting of array dimensions. It can also be used to resize the array.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. All NumPy wheels distributed on PyPI are BSD licensed. ndarray.shape is the array attribute that returns a tuple consisting of array dimensions. It can also be used to resize the array.

**4.4.3 Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tools using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tools for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyse. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. The key features of pandas are,

* Fast and efficient Data Frame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and sub setting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality

**4.4.4 Scikit-Learn**

Scikit-Learn is python’s core machine learning package that has most of the necessary modules to support a basic machine learning project. The library provides a unified API (Application Programming Interface) for practitioners to ease the use of machine learning algorithms with only writing a few lines to accomplish the predictive or classification task. **One of the few libraries in python which has kept to the promise of maintaining the algorithm and interface layer simple** and not complicating it to cover the entire machine learning feature landscape. The package is written heavily in python, and it incorporates C++ libraries like LibSVM and LibLinear for support vector machines and generalized linear model implementation. The package depends on Pandas (mainly for the data frame processes), numpy (for the ndarray construct) and scipy (for sparse matrices).

The package is useful mainly because of its project vision. **Code quality and proper documentation form the core vision.** Robust implementation takes priority over as many features inclusion as possible for a given algorithm and also the implementation is strongly backed by unit tests (coverage of >80%). The package documentation includes narrative documentation, class references, tutorials, installation instructions, and more than 60 examples which are very useful for the beginners. Not all upcoming machine learning algorithms are added to the package immediately to keep the package clutter free. **There is a clear inclusion criteria setup for new machine learning algorithms.**

The inclusion criteria come with the following conditions.

1. The proposed algorithm should outperform the methods that were implemented in it in some area.
2. Should fit into the API design seamlessly (should take numpy array as input and also follow the fit/transform/predict process flow).
3. The new implementation has to be supported with research paper or implementation in another package.

**4.4.5 Seaborn**

Seaborn is a library in Python predominantly used for making statistical graphics. **Seaborn**is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

Seaborn offers the following functionalities:

1. Dataset oriented API to determine the relationship between variables.
2. Automatic estimation and plotting of linear regression plots.
3. It supports high-level abstractions for multi-plot grids.
4. Visualizing univariate and bivariate distribution.

Using Seaborn, we can plot wide varieties of plots like:

1. Distribution Plots
2. Pie Chart & Bar Chart
3. Scatter Plots
4. Pair Plots
5. Heat maps

**CHAPTER 5**

**IMPLEMENTATION OF PROPOSED SYSTEM**

**5.1 DATA SET COLLECTION**

This project utilizes the dataset provided by revolution analytics for the detection of the fraudulent credit card transaction. This dataset contains transactions, occurred in two days, made in September 2013 by European cardholders. The dataset contains 31 numerical features. Since some of the input variables contains financial information, the PCA transformation of these input variables were performed in order to keep these data anonymous. Three of the given features weren’t transformed. Feature "Time" shows the time between first transaction and every other transaction in the dataset. Feature "Amount" is the amount of the transactions made by credit card. Feature "Class" represents the label and takes only 2 values: value 1 in case fraud transaction and 0 otherwise.

**5.2 DATA PRE – PROCESSING**

Data pre-processing is a technique for converting raw data into a widely understandable format. Data manipulation processes the raw data. Data pre-processing is used in customer relationship management and rule-based applications (like neural networks). Data goes through a number of processes during pre-processing.  Data is cleansed by filling in missing values or, removing corrupted or inconsistent data, smoothing the noisy data, or fixing the inconsistency in data. It is most crucial for ML algorithms to smoothly deal with noisy results. Data can be "filtered" by breaking it up into equal parts, or by using a "linear regression" or a "cluster analysis" (clustering). There are data anomalies due to human error (the information was stored in a wrong field). Never duplicate database values to avoid giving the value an advantage (bias). Data Integration: putting together data from various sources with different formats. The data are normalized and updated. Data normalization ensure all data is stored in one location and all relationships between data are logical. Queries will become slower as the amount of data become bigger. Data mining phase aims to get the information out of the systems. Data reduction is different processes. Data may also be discretized for evaluating statistical measurements. This analysis includes the reduction of values of continuous attribute from interval. Often, a dataset is too large or too complicated to be work with. Sampling techniques may be used to pick and function with only a subset of the data, given that they have the same properties.

Data pre-processing could be a strategy that is utilized to change over the raw information into a clean dataset. It is the basic step to train every machine learning classifier algorithm. This technique concludes such actions as handle missing values, rescaling of the dataset, transform into binary data and standardize of the dataset. When the dataset included attributes with varying scales, rescaling is used to scale the dataset. The binary transformation has been applied to convert the value into 0 and 1. All values of every attribute are considered as 1 for above the threshold and as 0 for below the threshold. Standardized method ensures that each attribute has mean 0 and standard deviation 1.

**5.3 FEATURE SELECTION**

Feature selection is needed for trained each machine learning classifier because without removing unnecessary attributes from the dataset result may be affected. The classifier algorithm with feature selection gives better performance and reduce the execution time of the model. The MLP classifier is used Since this data set is rather simple, the accuracy score is close to 1. We’re now ready to estimate feature importance. To summarize, a feature’s importance is the difference between the baseline score *s* and the average score obtained by permuting the corresponding column of the test set. If the difference is small, then the model is insensitive to permutations of the feature, so its importance is low. Conversely, if the difference is large, then the feature’s importance is high. The parameter *n* controls the number of permutations per feature — more permutations yield better estimates, at the cost of computation time. Note chosen a scoring metric other than accuracy, such as the F1 score.

**5.4 PREDICTION USING CLASSIFICATION ALGORITHMS**

Classification technique is an important feature of supervised learning. Classifiers learn from the training dataset and apply on the testing dataset for finding the target attribute. Below there are classification techniques used in research. Then the pre-processed data is then fed into MLP using sklearn and the output received from that is fed into Gaussian naive bayes, BernoulliNB, MultinomialNB, ensemble method random forest classifier and the output values are predicted with precision, recall, f1 score and accuracy. Then the confusion matrix is plotted at the end to the performance of all the above-mentioned algorithms. In this results algorithm have been compared, and our findings show that the ensemble method random forest classifier algorithm provides better classification and prediction performance for determining severity stage in chronic kidney disease.

**5.4.1 MLP CLASSIFIER**

MLP classifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLP Classifier relies on an underlying Neural Network to perform the task of classification.

The perceptron is a linear classifier — an algorithm that classifies input by separating two categories with a straight line. Input is typically a feature vector x multiplied by weights w and added to a bias b: y = w \* x + b.

**5.4.2 GAUSSIAN NAÏVE BAYES**

Naive Bayes is a generative model. (Gaussian) Naive Bayes assumes that each class follow a Gaussian distribution. The difference between QDA and (Gaussian) Naive Bayes is that Naive Bayes assumes independence of the features, which means the covariance matrices are diagonal matrices.

Naive Bayes makes the assumption that the features are independent. This means that we are still assuming class-specific covariance matrices (as in QDA), but the covariance matrices are diagonal matrices. This is due to the assumption that the features are independent.

**5.4.3 BERNOULLI NAÏVE BAYES**

The Bernoulli Naive Bayes is one of the variations of the Naive Bayes algorithm in machine learning and it is very useful to use in a binary distribution where the output label may be present or absent. If you have never used this machine learning algorithm before, this article is for you. In this article, I will take you through an introduction to the Bernoulli Naive Bayes algorithm in machine learning and its implementation using Python.

Bernoulli Naive Bayes is one of the variants of the Naive Bayes algorithm in machine learning. It is very useful to be used when the dataset is in a binary distribution where the output label is present or absent. The main advantage of this algorithm is that it only accepts features in the form of binary values such as:

* True or False
* Spam or Ham
* Yes or No
* 0 or 1

Other advantages of using this algorithm for binary classification:

* It is very fast compared to other classification algorithms.
* Sometimes machine learning algorithms do not work well if the dataset is small, but this is not the case with this algorithm because it gives more accurate results compared to other classification algorithms in the case of a small dataset.
* It’s fast and can also handle irrelevant features easily.

**5.4.4 RANDOM FOREST CLASSIFIER**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Figure 5.1 Random Forest Classifier

**5.4.5 LOGISTIC REGRESSION**

Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

Types of Logistic Regression

1. Binary Logistic Regression: The categorical response has only two 2 possible outcomes. Example: Spam or Not

2. Multinomial Logistic Regression: Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression: Three or more categories with ordering. Example: Movie rating from 1 to 5

**5.4.6 DECISION TREE CLASSIFIER**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees



Figure 5.2 Decision Tree Classifier

* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
* The logic behind the decision tree can be easily understood because it shows a tree-like structure.

**5.5 ANALYSIS OF RESULTS AND EVALUATION.**

Throughout this strategy, we pre-subject the CC deception dataset to a series of operations in order to accomplish the outcome of searching out the collection. Next, in order to develop or prepare the model, we must separate the observable data into two categories: visible preparation data and visible experiment data.

We use the visible preparation information to create or prepare the Random Forest and Naive baye and logistic regression. Then we cultivate two models together. Finally, the precision, or correctness, accuracy, recall, and F1-score for both models have been carefully planned.

**5.5.1 SCREENSHOTS**

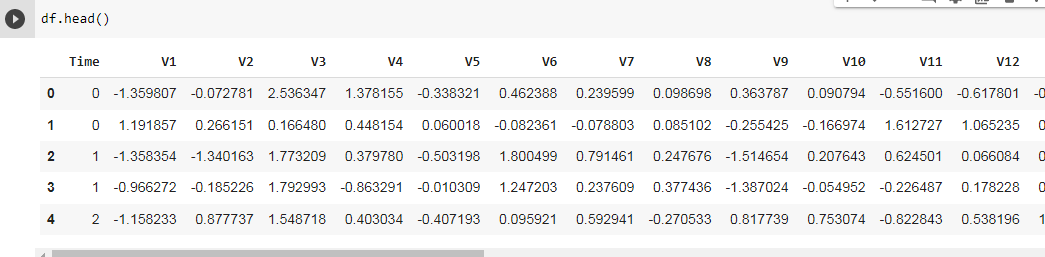


Figure 5.3 Loading of dataset

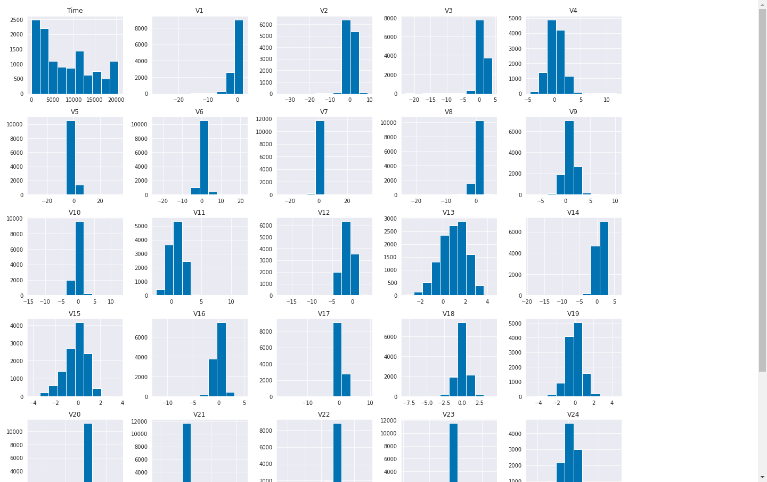


Figure 5.4 Histogram visualization

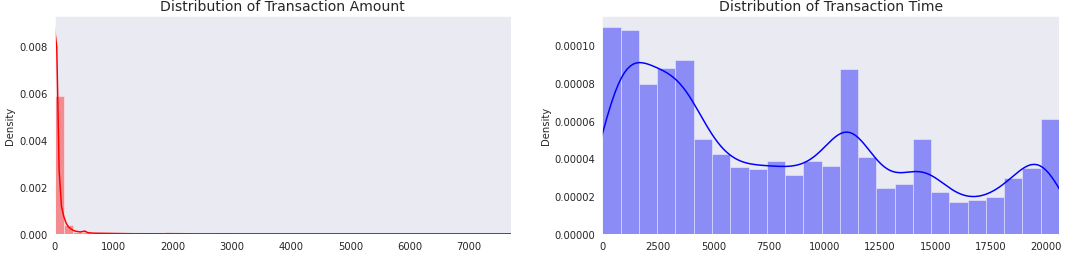


Figure 5.5 Distribution of Transaction Time

**5.5.2 Result analysis**

* The data set consists of two days of transactions which have low per cent of fraudulent transactions and unbalanced data
* The only characteristics that don't undergo accompanying PCA are 'Time' and 'Amount'. Feature 'Time' holds the seconds slipping away between two points, each undertaking and the first business deal in the dataset. When the goal changes, the feature 'Class' appears, with a value of 1 either through deception or 0 in another way.
* Load data into the table which is to be displayed as output.
* Explore the label class in the dataset.
* Separate feature information (predictors) from labels.
* Data scaled to zero-unit variance and mean.
* Partition information into train and test sets.
* Unbalanced dataset is sampled.

**5.5.3 Models of trains and Model Evaluation**

The feature extraction is done using supervised algorithms and the week day is the label target variable that is used to reveal the relationship between dataset and the target variable. Three machine intelligence algorithms like Gaussian Naive Baye, Bernoulli Naïve Baye and Random Forest are prepared by utilising the treated feature information in visible form.

* CC deception dataset, algorithms predict the class of unknown data sets and it takes only binary values for prediction of results.
* The split based on fraud and non-deception cases by regression model which plots 0 as output indicating as Fraud none cases.
* Accuracy, precision, F1-score and recall for transactions false at end

**5.5.4 Examining the Results**

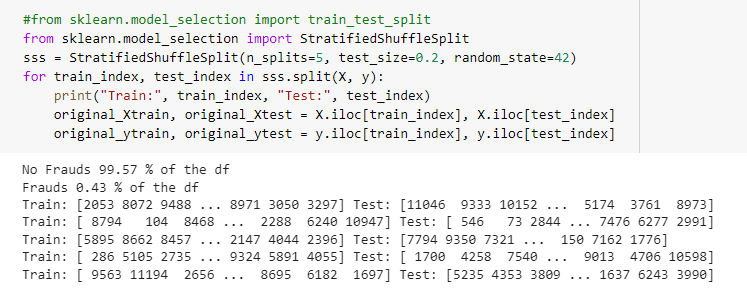
****

Figure 5.6 fraud and no fraud predict

**5.6 MODEL SUMMARY**

**Table 5.6.1 Model Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Algorithms | Precision | recall | F1-score | Accuracy |
| Traditional  Method | Adaboost algorithm | 0 0.99  1 0.79 | 0.99  0.64 | 0.99  0.75 | 0.90 |
| Proposed method | Gaussian naive bayes (GNB)  Random forest (RF)  Bernoulli naive bayes (BNB)  Decision tree (DT)  Logistic regression (LR)  MLP | 0 1.00  1 0.26  0 1.00  1 1.00  0 1.00  1 0.84  0 1.00  1 0.86  0 1.00  1 0.77  0 1.00  1 0.85 | 0.98  0.90  1.00  0.81  1.00  0.76  1.00  0.71  1.00  0.47  1.00  0.79 | 0.99  0.40  1.00  0.89  1.00  0.80  1.00  0.78  1.00  0.59  1.00  0.82 | 0.98  1.00  1.00  1.00  1.00  1.00 |

Figure 5.7 output accuracy of MLP, GNB, BNB, RF, LR, DT

The figure 5.7 illustrates that the prediction done to find out the fraud transactions with 0’s and 1’s with supervised learning of proposed model algorithms compared with traditional methods to prove the precision, fraud to detect the thief aspect

**CHAPTER 6**

**CONCLUSION**

**6.1 CONCLUSION**

Credit card fraud is without a doubt an act of criminal dishonesty. This report has listed out the most common methods of fraud along with their detection methods and reviewed recent findings in this field. This report has also explained in detail, how machine learning can be applied to get better results in fraud detection along with the algorithm, pseudocode, explanation its implementation and experimentation results. While the algorithm does reach over 100% accuracy. This high percentage of accuracy is to be expected due to the huge imbalance between the number of valid and number of genuine transactions. Since the entire dataset consists of only two days’ transaction records, it’s only a fraction of data that can be made available if this project were to be used on a commercial scale. Being based on machine learning algorithms, the program will only increase its efficiency over time as more data is put into it.

**CHAPTER 7**

**REFERENCES**

[1] "Fraud Detection in E-Commerce Using Machine Learning", International Journal of Advanced Trends in Computer Science and Engineering, vol. 10, no. 3, pp. 2206-2211, 2021. Available: 10.30534/ijatcse/2021/1011032021.

[2] formalised paraphraseDart Consulting, Growth Of Internet Users In India And Their Impact On The Country's Economy: https://www.dartconsulting.co.in/marketnews/growth-of-internet-users-in-india-and-their-impact-on-the-countryseconomy/

[3] Ganga Rama Koteswara Rao and R. Satya Prasad, "-Shielding The Networks Depending On Linux Servers Against Arp Spoofing, International Journal of Engineering and Technology (UAE), Vol. 7, PP.75-79, May 2018, ISSN No: 2227-524X, DOI: 10.14419/ijet. v7i2.32.13531.

[4] Heta Naik and Prashasti Kanikar: Algorithms of ML for CC Fraud Detection, International Journal of Computer Applications (0975–8887) Volume 182 – No. 44, March 2019.

[5] Navanshu Khare, Saad Yunus Sait: CC Fraud Detection Using Algorithms of MLModels and Collating Algorithms of MLModels, International Journal of Pure and Applied Mathematics, Volume 118, No. 20, 2018, 825-838, ISSN: 1314-3395.

[6] Randula Koralage, Faculty of Information Technology, University of Moratuwa, Data Mining Techniques for CC Fraud Detection.

[7] Roy, Abhimanyu, et al:Deep learning detecting fraud in CC transactions, 2018 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2018.

[8] Sahayasakila.V, D. Kavya Monisha, Aishwarya, Sikhakolli VenkatavisalakshiseshsaiYasaswi: CC Fraud Detection System Using Smote Technique and Whale Optimization Algorithm, ISSN: 2249-8958, Volume-8 Issue-5, June 2019.

[9] Statista.com. Retail e-commerce revenue forecast (in billions of dollars) from 2017 to 2023.https://www.statista.com/statistics/280925/e-commercerevenueforecast-in-India/ (accessed April 2020).

[10] Yashvi Jain, Namrata Tiwari, Shripriya Dubey, Sarika Jain: A Comparative Analysis of Various CC Fraud Detection Techniques, International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019.

[11] Yong Fang1, Yunyun Zhang2, and Cheng Huang1, CC Fraud Detection Using Machine Learning, Computers, Materials & Continua, vol.61, no.1, pp.185-195, 2019.

[12] Kaithekuzhical (Kaithekuzhical) (KaitheDr. Ajeet Chikkamannur and Leena Kurien: Detection And Prediction Of CC Fraud Transactions Using Machine Learning, International Journal Of Engineering Sciences & Research Technology.

[13] R. Sailusha, V. Gnaneswar, R. Ramesh and G. R. Rao, "Credit Card Fraud Detection Using Machine Learning," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 1264-1270, doi: 10.1109/ICICCS48265.2020.9121114.

[14] M. Azhan and S. Meraj, "Credit Card Fraud Detection using Machine Learning and Deep Learning Techniques," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 514-518, doi: 10.1109/ICISS49785.2020.9316002.

[15]A. M. Babu and A. Pratap, "Credit Card Fraud Detection Using Deep Learning," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2020, pp. 32-36, doi: 10.1109/RAICS51191.2020.9332497.

[16] S, Varun. (2020). Credit Card Fraud Detection using Machine Learning Algorithms. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS070649.

[17] Yvan Lucas, Johannes Jurgovsky (2021) Credit card fraud detection using machine learning: A survey

[18] Thirunavukkarasu.M et al, credit card fraud detection using machine learning International Journal of Computer Science and Mobile Computing, Vol.10 Issue.4, April- 2021, pg. 71-79

[19] Gupta, Shalini, and R. Johari. ”A New Framework for Credit Card Transactions Involving Mutual Authentication between Cardholder and Merchant.” International Conference on Communication Systems and Network Technologies IEEE, 2021:22-26.

[20] Y. Gmbh and K. G. Co, “Global online payment methods: the Full year 2020,” Tech. Rep., 3 2020.

[21] Pratyush Sharma et.al, Machine Learning Model for Credit Card Fraud Detection- A Comparative Analysis The International Arab Journal of Information Technology, Vol. 18, No. 6, November 2021

[22] Chen, Joy Iong-Zong, and Kong-Long Lai. "Deep Convolution Neural Network Model for Credit-Card Fraud Detection and Alert." Journal of Artificial Intelligence 3, no. 02 (2021): 101-112.

**APPENDIX**

**A1**

**IMPORTING LIBRARIES**

import pandas as pd

from pandas.plotting import scatter\_matrix

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sn

%matplotlib inline

sn.set\_style("dark")

sn.set\_palette("colorblind")

import os

from collections import Counter

import pickle

from sklearn.pipeline import Pipeline, make\_pipeline

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler, PowerTransformer

from sklearn.metrics import confusion\_matrix

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn import metrics

**READING THE DATASET INFORMATION**

df= pd.read\_csv("/content/sample\_data/creditcard\_csv.csv")

df.head()

df.info()

**DATASET PREPROCESSING**

df.describe()

print('Normal transactions count: ', df['Class'].value\_counts().values[0])

print('Fraudulent transactions count: ', df['Class'].value\_counts().values[1])

df.info

print(df.isnull().sum())

display(df.Class.value\_counts(normalize = True))

import seaborn as sns

corelation = df.corr()

corelation

plt.figure(figsize=(10,5))

ax=sns.heatmap(df.corr(),annot=True)

df.hist(figsize=(20,20), column=df.columns)

plt.show()

fig, ax = plt.subplots(1, 2, figsize=(18,4))

amount\_val = df['Amount'].values

time\_val = df['Time'].values

import seaborn as sns

sns.distplot(amount\_val, ax=ax[0], color='r')

ax[0].set\_title('Distribution of Transaction Amount', fontsize=14)

ax[0].set\_xlim([min(amount\_val), max(amount\_val)])

sns.distplot(time\_val, ax=ax[1], color='b')

ax[1].set\_title('Distribution of Transaction Time', fontsize=14)

ax[1].set\_xlim([min(time\_val), max(time\_val)])

plt.show()

sns.countplot('Class',data=df)

plt.title('Class Distributions \n (0: No Fraud|| 1: Fraud)', fontsize=14)

duplicated\_rows = df[df.duplicated()]

display(duplicated\_rows)

df.drop\_duplicates(inplace=True)

display(df.shape)

duplicated\_rows = df[df.duplicated()]

display(duplicated\_rows)

amount = df['Amount'].values

amount = amount.reshape(-1,1)

power = PowerTransformer()

df['Amount\_trans'] = power.fit\_transform(amount)

sns.histplot(df.Amount\_trans)

plt.show()

sum(df.Amount == 0)

df.Amount.replace(0, np.NaN, inplace=True)

df.dropna(subset=['Amount'], inplace=True)

df.Amount.isna().sum()

df.shape

df['Night'] = (df.Time.between(4000, 27000) | df.Time.between(89000, 112000)).astype(int)

df.Night.sum()

df.head()

df.drop(['Time', 'Amount\_trans'], axis=1,inplace=True)

**FEATURE AND TARGET VARIABLE SELECTION**

print('No Frauds', round(df['Class'].value\_counts()[0]/len(df) \* 100,2), '% of the df')

print('Frauds', round(df['Class'].value\_counts()[1]/len(df) \* 100,2), '% of the df')

X = df.drop('Class', axis=1)

y = df[['Class']]

#from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import StratifiedShuffleSplit

sss = StratifiedShuffleSplit(n\_splits=5, test\_size=0.2, random\_state=42)

for train\_index, test\_index in sss.split(X, y):

print("Train:", train\_index, "Test:", test\_index)

original\_Xtrain, original\_Xtest = X.iloc[train\_index], X.iloc[test\_index]

original\_ytrain, original\_ytest = y.iloc[train\_index], y.iloc[test\_index]

X = df.drop('Class', axis=1)

y = df['Class']

# Our data is already scaled we should split our training and test sets

from sklearn.model\_selection import train\_test\_split

# This is explicitly used for undersampling.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Turn the values into an array for feeding the classification algorithms.

X\_train = X\_train.values

X\_test = X\_test.values

y\_train = y\_train.values

y\_test = y\_test.values

**CLASSIFICATION ALGORITHMS**

**MLP CLASSIFIER**

x\_train, x\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=2)

from sklearn.neural\_network import MLPClassifier

clf=MLPClassifier(random\_state=None)

clf.fit(x\_train,y\_train)

y\_pred= clf.predict(x\_test)

print(clf.predict(x\_test))

accu=(metrics.accuracy\_score(y\_test,y\_pred))\*100

confusion\_mat=metrics.confusion\_matrix(y\_test,y\_pred)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_pred))

**GAUSSIAN NAÏVE BAYES**

from sklearn.naive\_bayes import GaussianNB

clf\_gnb = GaussianNB()

clf\_gnb.fit(x\_train, y\_train)

y\_predict\_gnb=clf\_gnb.predict(x\_test)

print(clf\_gnb.predict(x\_test))

accu\_gnb=(metrics.accuracy\_score(y\_test,y\_predict\_gnb))\*100

confusion\_mat\_gnb=metrics.confusion\_matrix(y\_test,y\_predict\_gnb)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_predict\_gnb))

**BERNOULLI NAÏVE BAYES**

from sklearn.naive\_bayes import BernoulliNB

clf\_bnb = BernoulliNB(alpha=1.0, binarize=0.0, class\_prior=None, fit\_prior=True)

clf\_bnb.fit(x\_train,y\_train)

y\_predict\_bnb=clf\_bnb.predict(x\_test)

print(y\_predict\_bnb)

accu\_bnb=(metrics.accuracy\_score(y\_test,y\_predict\_bnb))\*100

confusion\_mat\_bnb=metrics.confusion\_matrix(y\_test,y\_predict\_bnb)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_predict\_bnb))

**RANDOM FOREST CLASSIFIER**

from sklearn.ensemble import RandomForestClassifier

clf\_rfc = RandomForestClassifier()

clf\_rfc = clf\_rfc.fit(x\_train,y\_train)

import sklearn.metrics

from sklearn import metrics

y\_predict\_rfc=clf\_rfc.predict(x\_test)

print(clf\_rfc.predict(x\_test))

accu\_rfc=(metrics.accuracy\_score(y\_test,y\_predict\_rfc))\*100

confusion\_mat\_dt=metrics.confusion\_matrix(y\_test,y\_predict\_rfc)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_predict\_rfc))

**LOGISTC REGRESSION**

from sklearn.linear\_model import LogisticRegression

clf\_lr = LogisticRegression()

clf\_lr = clf\_lr.fit(x\_train,y\_train)

import sklearn.metrics

from sklearn import metrics

y\_predict\_lr=clf\_lr.predict(x\_test)

print(clf\_lr.predict(x\_test))

accu\_lr=(metrics.accuracy\_score(y\_test,y\_predict\_lr))\*100

confusion\_mat\_lr=metrics.confusion\_matrix(y\_test,y\_predict\_lr)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_predict\_lr))

**DECISION TREE CLASSIFIER**

from sklearn.tree import DecisionTreeClassifier

clf\_model = DecisionTreeClassifier(criterion="gini", random\_state=42,max\_depth=3, min\_samples\_leaf=5)

clf\_model.fit(x\_train,y\_train)

import sklearn.metrics

from sklearn import metrics

y\_predict\_clf\_model=clf\_model.predict(x\_test)

print(clf\_model.predict(x\_test))

accu\_clf\_model=(metrics.accuracy\_score(y\_test,y\_predict\_clf\_model))\*100

confusion\_mat\_clf\_model=metrics.confusion\_matrix(y\_test,y\_predict\_clf\_model)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_predict\_clf\_model))

**CONFUSION MATRIX**

import numpy as np

import matplotlib.pyplot as plt

import numpy as np

import itertools

def plot\_confusion\_matrix(cm,

target\_names,

title='Confusion matrix',

cmap=None,

normalize=True):

accuracy = np.trace(cm) / float(np.sum(cm))

misclass = 1 - accuracy

if cmap is None:

cmap = plt.get\_cmap('Blues')

plt.figure(figsize=(8, 6))

plt.imshow(cm, interpolation='nearest', cmap=cmap)

plt.title(title)

plt.colorbar()

if target\_names is not None:

tick\_marks = np.arange(len(target\_names))

plt.xticks(tick\_marks, target\_names, rotation=45)

plt.yticks(tick\_marks, target\_names)

if normalize:

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

thresh = cm.max() / 1.5 if normalize else cm.max() / 2

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):

if normalize:

plt.text(j, i, "{:0.4f}".format(cm[i, j]),

horizontalalignment="center",

color="red" if cm[i, j] > thresh else "black")

else:

plt.text(j, i, "{:,}".format(cm[i, j]),

horizontalalignment="center",

color="red" if cm[i, j] > thresh else "black")

plt.tight\_layout()

plt.ylabel('True laabel')

plt.xlabel('Predicted label\naccuracy={:0.4f}; misclass={:0.4f}'.format(accuracy, misclass))

plt.show()

plot\_confusion\_matrix(confusion\_mat\_dt,target\_names = ['Fraud', 'non fraud'])

sns.countplot(y\_train)

**SMOTE MODEL**

from imblearn.over\_sampling import SMOTE

over\_sample = SMOTE()

x\_smote, y\_smote = over\_sample.fit\_resample(x\_train, y\_train)

sns.countplot(y\_smote)

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

# training

model.fit(x\_smote, y\_smote)

# testing

y\_pred = model.predict(x\_test)

print(classification\_report(y\_test, y\_pred))

from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n\_jobs=-1)

# training

model.fit(x\_smote, y\_smote)

# testing

y\_pred = model.predict(x\_test)

print(classification\_report(y\_test, y\_pred))

**APPENDIX**

**A2**

**SCREENSHOTS**

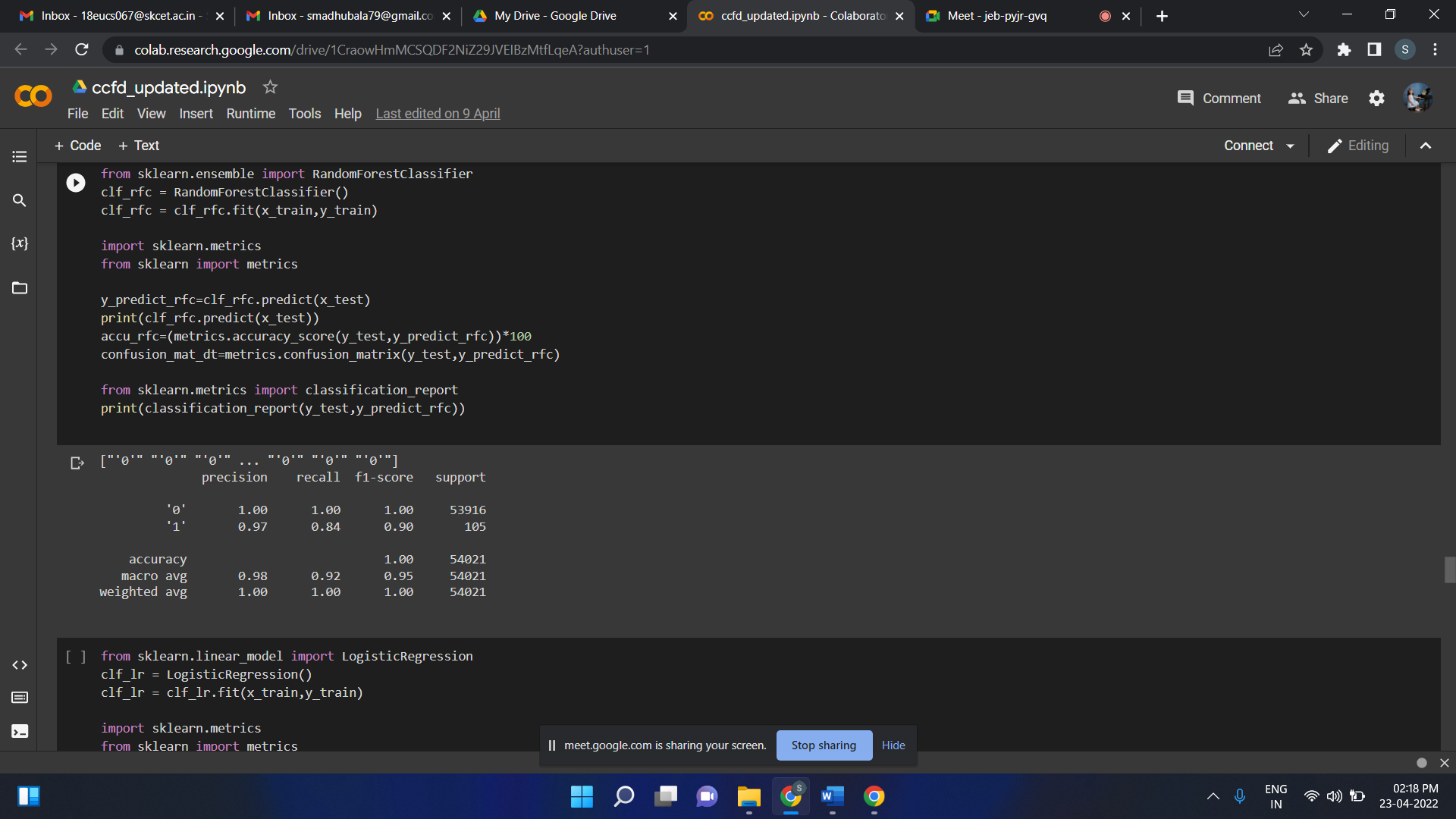


Figure A2.1 Random Forest Classifier

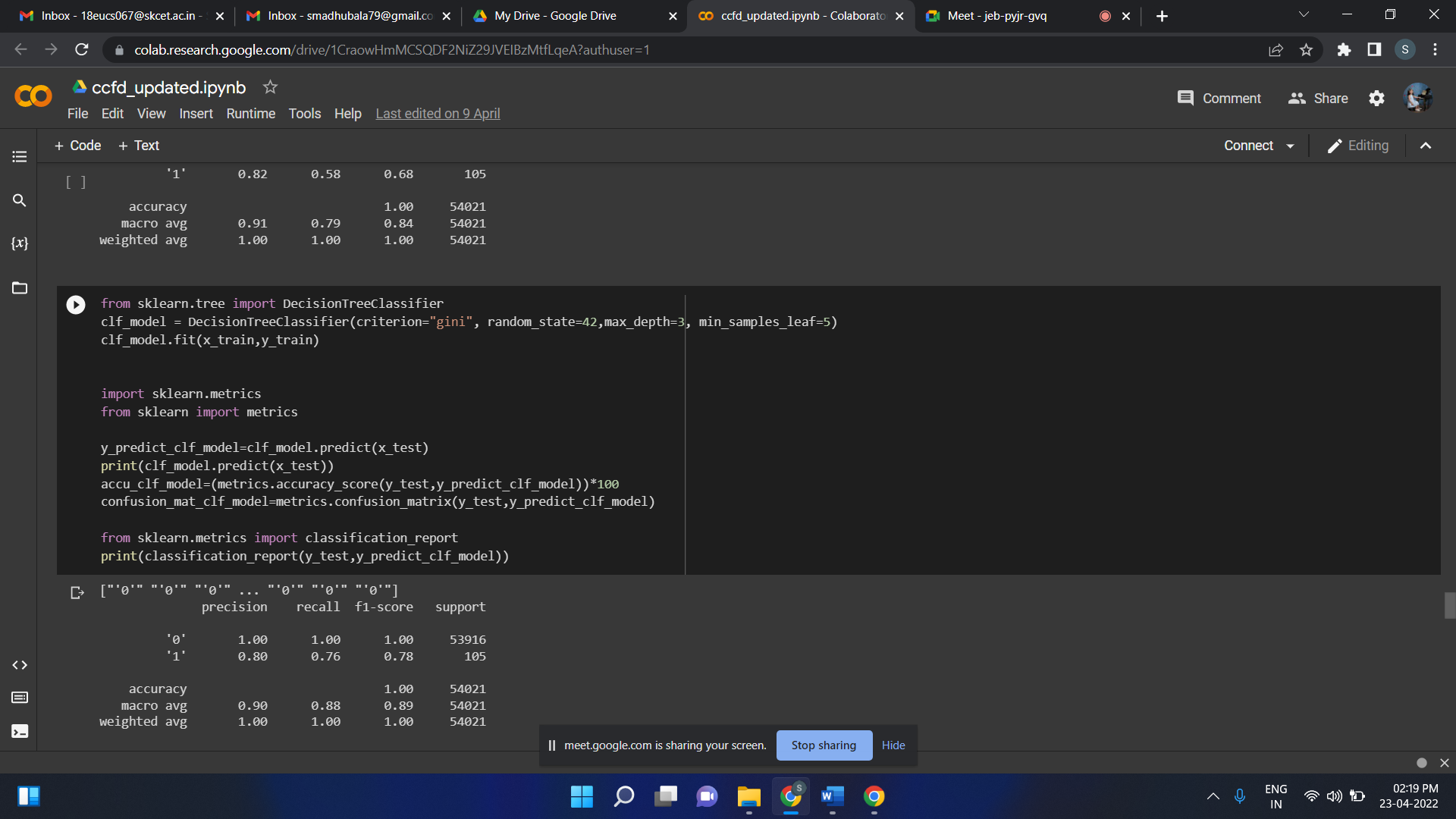


Figure A2.2 Decision Tree Classifier

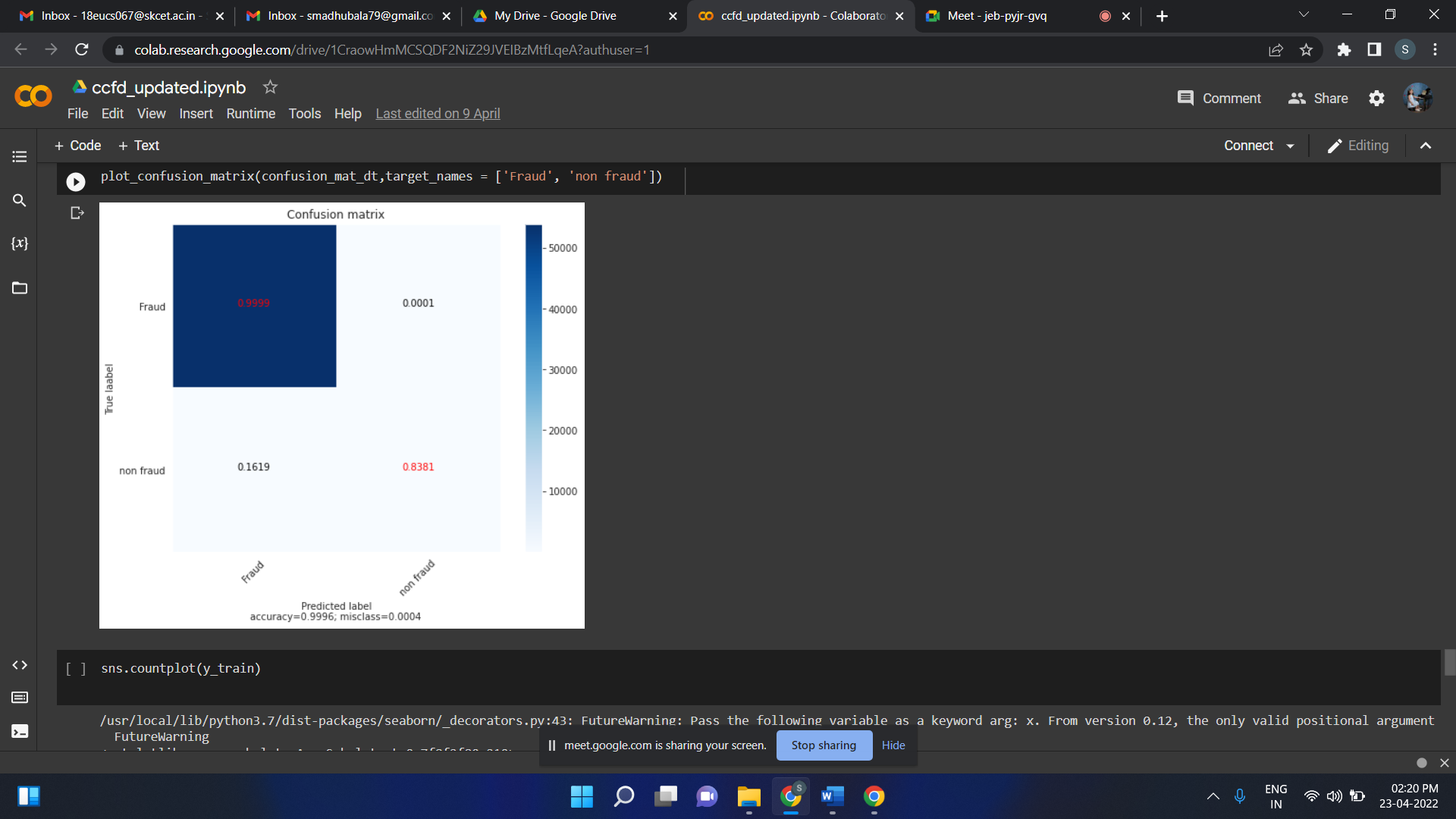


Figure A2.3 Confusion Matrix

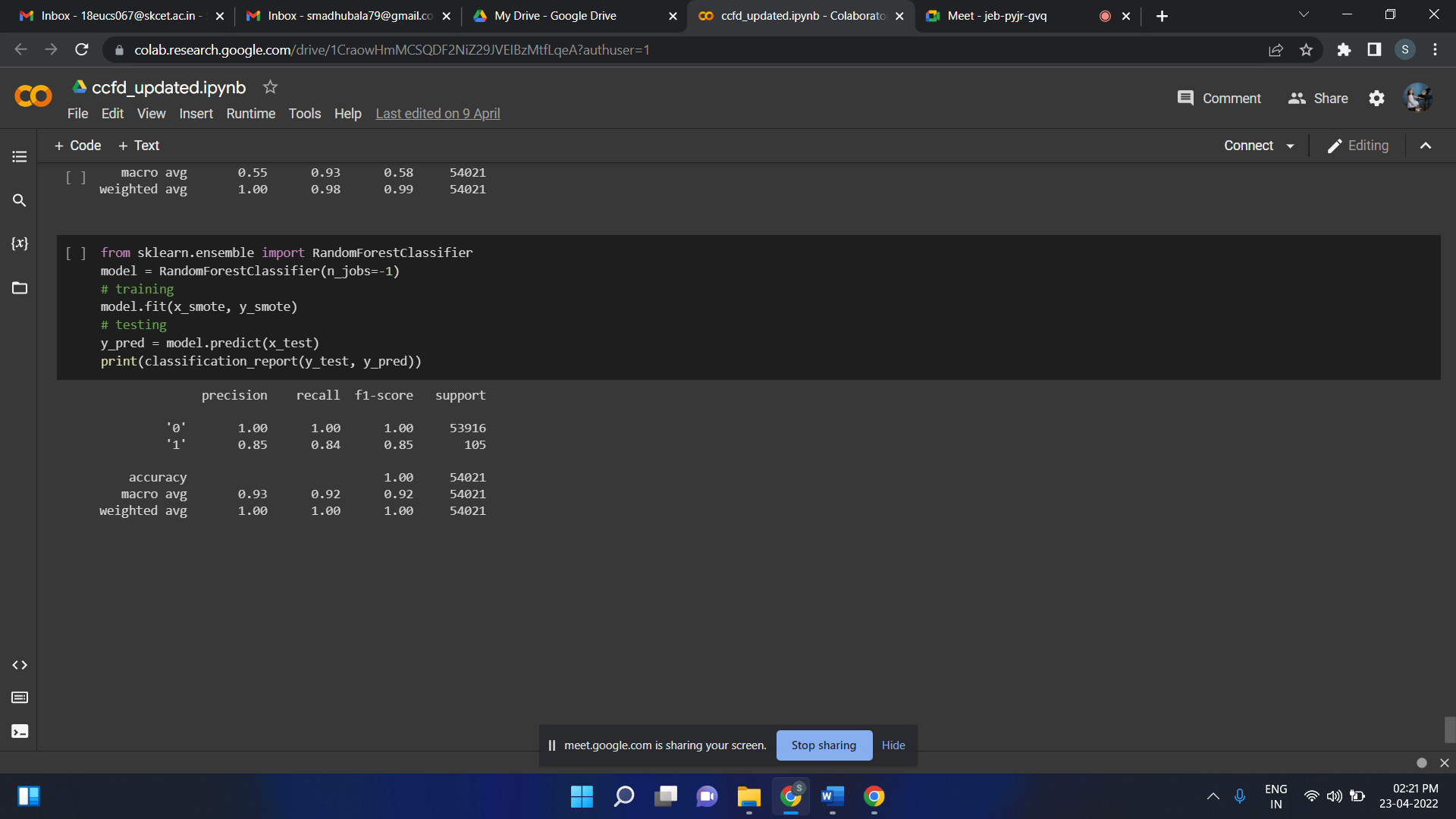


Figure A2.4 Smote Model

**PUBLICATION**

**Authors:**

* Ms. Priya A , Assistant Professor/CSE
* Mr. Avinash S Narayanan (18EUCS021)
* Ms. Madhu Bala S (18EUCS067)
* Mr. Bhavikk D Patel (18EUCS026)

**Title : Optimal Algorithm for Credit Card Fraud Detection**

**Conference/Journal Name:** International Conference on Artificial Intelligence and Smart Systems (ICAIS 2022)

**Venue :** JCT College of Engineering and Technology (Online Mode)

**Date :** 23-25, February 2022.

**Published paper link:** <https://ieeexplore.ieee.org/document/9742922>

