# PREDICTION OF CLAIMS AND FRAUDS PAYMENT USING APP

*Minor project report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**

**By**


| | | |
|---|---|---|
| **K. R. PREM KUMAR** | (20UECS0519) | **(VTU 12021)** |
| **MAHESH SAI CHARAN** | (20UECS0568) | **(VTU 16984)** |
| **GOLLAPALLI JAYANTH** | (20UECS0339) | **(VTU 16923)** |


*Under the guidance of*
*Mrs. PANNEER SELVI. R ,M.E.,*
*ASSISTANT PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**Accredited by NAAC with A++ Grade**
**CHENNAI 600 062, TAMILNADU, INDIA**


**May, 2023**

# PREDICTION OF CLAIMS AND FRAUDS PAYMENT USING APP

*Minor project report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**

**By**


| | | |
|---|---|---|
| **K. R. PREM KUMAR** | (20UECS0519) | **(VTU 12021)** |
| **MAHESH SAI CHARAN** | (20UECS0568) | **(VTU 16984)** |
| **GOLLAPALLI JAYANTH** | (20UECS0339) | **(VTU 16923)** |


*Under the guidance of*
*Mrs. PANNEER SELVI. R, M.E.,*
*ASSISTANT PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**Accredited by NAAC with A++ Grade**
**CHENNAI 600 062, TAMILNADU, INDIA**


**May, 2023**

# CERTIFICATE

It is certified that the work contained in the project report titled ”PREDICTION OF CLAIMS AND FRAUDS PAYMENT USING APP” by ”K. R. PREM KUMAR (20UECS0519), MAHESH SAI CHARAN (20UECS0568), GOLLAPALLI JAYANTH (20UECS0339)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

**Signature of Supervisor**
**Mrs. Panneer Selvi. R**
**Assistant Professor**
**Computer Science & Engineering**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science & Technology**
**May, 2023**

**Signature of Head of the Department**
**Dr. M.S. Muralidhar**
**Associate Professor & HoD**
**Computer Science & Engineering**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science & Technology**
**May, 2023**

**Signature of the Dean**
**Dr. V. Srinivasa Rao**
**Professor & Dean**
**Computer Science & Engineering**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science & Technology**
**May, 2023**

# DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

K. R. PREM KUMAR

Date: / /

(Signature)

MAHESH SAI CHARAN

Date: / /

(Signature)

GOLLAPALLI JAYANTH

Date: / /

# APPROVAL SHEET

This project report entitled PREDICTION OF CLAIMS AND FRAUDS PAYMENT USING APP by K.R.PREM KUMAR (20UECS0519), MAHESH SAI CHARAN (20UECS0568), GOLLAPALLI JAYANTH (20UECS0339) is approved for the degree of B.Tech in Computer Science & Engineering.

**Examiners** **Supervisor**

Mrs. R.Panneer Selvi , M.E.,

**Date: / /**

**Place:**

# ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Mrs. R.PANNEER SELVI, M.E.,** for his/her cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

**K. R. PREM KUMAR** **(20UECS0519)**
**MAHESH SAI CHARAN** **(20UECS0568)**
**GOLLAPALLI JAYANTH** **(20UECS0339)**

# ABSTRACT

Fraudsters have become increasingly sophisticated in their methods of defrauding insurance companies. Insurance companies have to deal with a large number of claims every day, and it is not always possible to manually detect fraudulent claims. To address this challenge, machine learning algorithms and fraud detection techniques can be employed to predict fraudulent claims. The proposes method to predict claims frauds of payment using an app. The first step is to collect data related to claims and payments, which will be used to train machine learning models. Next, the collected data will be analyzed to identify patterns and trends that are indicative of fraudulent claims. Based on the analysis, the machine learning models will be developed and integrated into the app. Systems used for this purpose are based on the application of some methods of Artificial Intelligence, neglect human process analysis and make little use of Visual Analytics (VA) techniques. Using techniques of identification of outliers and conducting analysis through VA to reduce the false positive rate in the identification of fraudulent financial transactions process. Includes a hybrid approach, use of unsupervised outlier detection algorithms. Use of VA with the aim of reducing the incidence of false positives. Logistic regression Algorithm is used for predicting the categorical dependent variable using a given set of independent variables. This algorithm is used for fraud identification. Using Android Studio Software, app is developed and the data set is imported to the App and run the data set to analyze and show the data is fraud or legitimate transaction. The app will enable users to choose file related to claims and payments by importing from mobile, which will then be analyzed by the machine learning models to determine the likelihood of fraud. The app will also provide feedback to users, such as alerts if a claim appears suspicious. The app will be constantly updated with new data to improve the accuracy of the predictions. Overall, this approach can help insurance companies detect fraudulent claims quickly and efficiently, reducing losses and improving customer satisfaction.



**Keywords:Fraud Prevention, Legitimate, Logistic Regression, Machine Learning, Predictive Analysis, Visual Analytics**

# LIST OF FIGURES

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| APP | Application |
| CNN | Convolutional Neural Network |
| DDR | Double Data Rate |
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| ISMS | Information Security Management System |
| IT | Information Technology |
| NLP | Natural Language Processing |
| RAM | Random Access Memory |
| RNN | Recurrent Neural Network |
| SDK | Software Development Kit |
| SAS | Statistical Analysis Software |
| SIM | Subscriber Identity Module |
| SVM | Support Vector Machine |
| UML | Unified Modeling Language |
| VA | Visual Analytics |
| XML | Extensible Markup Language |

# TABLE OF CONTENTS

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

Fraud detection is the process of identifying fraudulent activities or transactions in a given data set. This technique involves training a machine learning model on a data set of known fraudulent and non-fraudulent transactions, so that it can learn the patterns and characteristics that distinguish between them. Fraud detection using machine learning has become increasingly important in recent years, as the volume and complexity of fraudulent activities have grown. Machine learning models can analyze large amounts of data in real time, allowing them to quickly identify suspicious patterns or behavior. This can help organizations to prevent financial losses, protect their customers, and maintain their reputation. Fraud detection using machine learning has many applications across various industries, including finance, insurance, healthcare, and e-commerce.

A fraud detection app using machine learning could be developed to help organizations detect and prevent fraudulent activities in real-time. The app could be designed to integrate with existing systems and workflows, allowing organizations to quickly and easily deploy it. The impact of fraud can be devastating for businesses and individuals. It can result in significant financial losses, legal consequences, and reputational damage. Fraud can take many forms, such as credit card fraud, identity theft, insurance fraud, and money laundering. The growing use of digital technology and online transactions has made it easier for fraudsters to carry out their illegal activities, and as a result, the need for effective fraud detection and prevention mechanisms has become more critical than ever.

Fraud detection is the process of identifying and preventing fraudulent activities by analyzing data patterns and transactions. It involves collecting data from various sources, such as transaction logs, social media, and customer behavior, and using advanced analytical tools to identify any suspicious activities. Fraud detection meth-

ods can vary depending on the industry and the type of fraud being targeted. Still, they typically involve analyzing data using machine learning algorithms, rule-based systems, and anomaly detection.

## 1.2 Aim of the project

The aim of developing an app to predict payment claims frauds is to create a tool that can accurately analyze data and provide insights to help prevent fraudulent claims from being approved. This app would utilize advanced algorithms and machine learning techniques to detect potential fraud patterns and alert the appropriate parties to investigate further. The app could be designed to gather data from various sources, such as claim forms, financial transactions, and social media activity, to build a comprehensive profile of the claimant. This information could then be compared against historical data on previous fraudulent claims to identify potential red flags and suspicious activity.

## 1.3 Project Domain

In the insurance industry. The app aims to prevent and detect fraudulent claims in the insurance payment process. Insurance companies deal with a large number of claims on a daily basis, and it becomes challenging to identify fraudulent claims among the legitimate ones. Fraudulent claims cost insurance companies millions of dollars every year and can have a significant impact on the overall profitability of the company. Therefore, detecting and preventing fraudulent claims is crucial for the success of an insurance company.

In Healthcare industry an app could be developed to predict fraudulent medical claims submitted to insurance companies. In Finance industry an app could be developed to predict fraudulent credit card transactions or other types of financial fraud, in E-commerce industry an app could be developed to predict fraudulent product returns or refund claims. In Telecommunications industry an app could be developed to predict fraudulent billing claims made by customers, In Government agencies an app could be developed to predict fraudulent benefit claims made by citizens.

## 1.4 Scope of the Project

To develop a machine learning-based model that can identify potentially fraudulent insurance claims. The project would involve collecting relevant data sets related to payment frauds in insurance claims and preprocessing them to remove any errors or inconsistencies. The next step would be to select the most relevant features that may be indicative of fraudulent claims and develop a predictive model using machine learning algorithms. Once the model is developed, it would be validated by testing its accuracy and effectiveness on a separate data set of claims. Finally, a mobile application would be developed that uses the predictive model to alert insurance companies about potentially fraudulent claims. The project would aim to reduce instances of payment fraud in insurance claims, thereby improving the overall efficiency of the insurance industry.

To reduce the cyber attacks and digital frauds. To increase fraud detection speed. Make user-Friendly, Responsive interface. The APP should be secured and unique. The App should support all platforms. Less manual work needed for additional verification. To make higher accuracy of fraud detection. To determine whether the customer is claiming fraudulent claim. To ease the task of finding fraud claims. In Banking and finance payment fraud is a growing problem in the banking and finance sector, and a predictive model can be developed to detect fraudulent transactions. Retail and in e-commerce retailers and e-commerce businesses can use predictive models to identify fraudulent transactions, such as stolen credit card information or fake identities used for purchases, in Healthcare the healthcare providers can use predictive models to detect fraudulent insurance claims, such as claims for services that were never provided or claims for unnecessary medical procedures. In Government agencies can use predictive models to identify fraudulent claims for social welfare programs, such as fraudulent claims for unemployment benefits or fraudulent claims for disability benefits, in Telecommunications companies can use predictive models to identify fraudulent phone usage, such as phone hacking or SIM card cloning.

# Chapter 2

# LITERATURE REVIEW

[1] A. Gupta et al., has implemented on the use of machine learning techniques for predicting bank loan approvals and have emphasized the importance of developing accurate loan prediction models for banks to improve their decision-making processes and reduce the risk of loan defaults, provides insights into the use of machine learning algorithms for bank loan prediction and highlights the potential of these algorithms to improve the efficiency and effectiveness of loan approval processes in the banking industry.

[2] Biao Xua et al., are proposed that from a methodological point of view, machine learning based fraud detection can be divided into two categories, conventional methods and deep learning, both of which have significant in terms of the lack of representation learning ability for the former and interpretability for the latter. Proposed a hybrid model that combines deep learning and boosting decision trees to improve the efficiency and accuracy of credit card fraud detection.

[3] Dr.Ahmed Hassan Butt et al., has implemented about the comparison of different algorithm s the performance of Random Forest, Naïve Bayes, K-Nearest Neighbor, Logistic Regression and Multilayer Perceptron. The results show that the Random Forest algorithm outperforms the other algorithms in terms of accuracy and precision. The authors have also discussed the advantages and limitations of the approach and have provided recommendations for future research and have emphasized the importance of real-time detection and prevention of credit card fraud in banks and the need for continuous improvement of fraud detection methods.

[4] Douglas C. Montgomery et al., are proposed in the article for an approach for clustering financial data that integrates cluster detection, optimization, and interpretation. The authors have applied their approach to a dataset of stock market data and have evaluated the performance of their approach using various clustering metrics. Their results show that their integrated approach outperforms traditional clustering

methods in terms of accuracy and interpretability. The authors have emphasized the importance of accurate and interpretable clustering methods for financial data analysis and decision-making, provided insights into the development of new approaches for clustering financial data and highlights the potential of these approaches to improve the efficiency and effectiveness of financial data analysis in the banking and finance industry.

[5] Dr. Jitendra Sheetlani et al., has implemented on the focus of use of machine learning techniques for predicting bank loan approvals and have emphasized the importance of developing accurate loan prediction models for banks to improve their decision-making processes and reduce the risk of loan defaults, provides insights into the use of machine learning algorithms for bank loan prediction and highlights the potential of these algorithms to improve the efficiency and effectiveness of loan approval processes in the banking industry.

[6] Fahd Sabry Esmail et al., are proposed for an approach for clustering financial data that integrates cluster detection, optimization, and interpretation. The authors have applied their approach to a dataset of stock market data and have evaluated the performance of their approach using various clustering metrics. The results show that integrated approach outperforms traditional clustering methods in terms of accuracy and interpretability.

[7] Gokula Krishnan et al., are highlighted the importance of feature selection and data preprocessing techniques in improving the accuracy of credit card fraud detection models. They highlight the need for an effective fraud detection system and discuss the challenges associated with developing such a system

[8] Jafar Nahri Aghdam Ghalejoogh et al., has described a comprehensive review of the current stateof-the-art in credit card fraud detection using machine learning techniques, and offer insights for researchers and practitioners interested in this field

[9] Rakhi Arora et al., has provided a useful reference for researchers and practitioners interested in credit card fraud detection and highlights the importance of developing effective fraud detection systems to prevent financial losses and maintain customer trust.

[10] Srinivasa Rao Dammavalam et al., are explained the main challenges in credit card fraud detection and discuss how machine learning methods can address these challenges and also provide an overview of various machine learning algorithms used in credit card fraud detection.

[11] Subhash P et al., has presented an overview of machine learning algorithms used in credit card fraud detection. The authors provide a comprehensive review of various techniques and methodologies used in recent studies to identify fraudulent transactions in credit card systems.

[12] T. Li et al., has presented a systematic mapping study of credit card fraud detection using machine learning. The authors conduct a literature review of 50 papers published between 2010 and 2020 that apply machine learning techniques to detect credit card fraud.

# Chapter 3

# PROJECT DESCRIPTION

## 3.1 Existing System

SAS Fraud Detection is a software solution that uses advanced analytics and machine learning algorithms to identify and prevent fraudulent activity in insurance claims data. The tool can analyze large amounts of data from multiple sources, including claims data, policyholder information, and external data sources, to identify suspicious patterns and behavior.

One of the key benefits of SAS Fraud Detection is its ability to analyze data in real-time, allowing insurance companies to quickly identify and investigate potentially fraudulent claims. The tool uses machine learning algorithms to learn from past data and detect emerging fraud schemes, providing insurance companies with an additional layer of protection against fraud. Additionally, SAS Fraud Detection provides detailed reporting and visualizations, allowing insurance companies to quickly identify trends and patterns in claims data and take action to prevent future fraud.

**Disadvantages**

1. False positives: One of the major disadvantages of these systems is that they may generate false positives, which means that legitimate claims may be flagged as fraudulent. This can result in delays in processing legitimate claims, which can frustrate policyholders and damage the reputation of insurance companies.
2. Cost: The cost of implementing these systems can be high, especially for smaller insurance companies. This can limit their ability to invest in new technologies and may make it difficult to compete with larger players in the market.
3. Privacy concerns: Collecting and analyzing large amounts of personal data can raise privacy concerns among policyholders. Insurance companies need to ensure that they are complying with data protection regulations and are transparent about how they are using customer data.

4. Complexity: SAS Fraud Detection is a complex tool that requires a high level of technical expertise to use effectively. This can make it difficult for smaller insurance companies with limited IT resources to implement and maintain the tool.
5. Training: Because SAS Fraud Detection is a complex tool, it requires significant training for staff to use it effectively. This can be time-consuming and costly, particularly for companies with limited resources.
6. Integration: SAS Fraud Detection may require integration with other software tools, such as claims management systems, to be effective. This can be a time-consuming and complex process, particularly if the company's existing systems are not compatible with SAS Fraud Detection.

### 3.2 Proposed System

The existing model is based on the techniques of linear regression and visual analytics. The linear regression is applied in order to Gather a large dataset of historical claims and payment transactions, including information such as the type of claim, the amount paid, and any relevant details about the claimant and Remove any incomplete or inaccurate data and prepare the remaining data for analysis by converting categorical variables to numeric data, Once the model has been trained and tested, deploy it as an app that can take in new claims data and predict the likelihood of fraud.

The Visual analytics is used to design with interactive dashboards that allow users to explore data visually, making it easier to identify trends and patterns that may indicate fraud. Users can also interact with the data in real-time, allowing them to quickly adjust the parameters and see the impact on the results. Visual analytics can be used to create data visualizations such as charts, graphs, and heat maps that make it easier to see patterns and relationships in the data. This can help users to identify potential fraud cases more quickly and accurately. An app can be designed to alert users when certain conditions are met, such as when a claim meets certain criteria that indicate potential fraud.

**Advantages**

1. Automation: By automating the process of analyzing claims data, an app can significantly reduce the amount of time and effort required to identify potential fraud cases. This saves resources and reduces the risk of fraudulent claims being missed.
2. Accuracy: Machine learning algorithms such as linear regression are highly accurate at identifying patterns in large datasets. An app that uses these algorithms can therefore provide highly accurate predictions of claims frauds.
3. Scalability: An app can be easily scaled to handle large volumes of claims data. This is important for insurers who may need to process thousands or even millions of claims each year.
4. Cost-effective: Using an app to predict claims frauds of payment is often more cost-effective than hiring additional staff or outsourcing the task to a third-party service provider.
5. Early detection: By detecting fraudulent claims early, an app can prevent payment of fraudulent claims, which can save insurers significant amounts of money.
6. Improved customer service: An app that can quickly identify fraudulent claims can help insurers to pay legitimate claims faster and improve customer satisfaction.

## 3.3 Feasibility Study

### 3.3.1 Economic Feasibility

The economic feasibility of developing an app to predict claims frauds of payment is promising. While the initial development and operational costs can be significant, the potential revenue streams could make the investment worthwhile. Subscription fees and commissions on prevented fraud could generate revenue, and the market demand for such an app is high. Fraudulent claims cost insurance companies and their clients millions of dollars annually, and an app that can accurately predict and prevent these fraudulent claims could save them significant sums of money. Furthermore, the app could also help insurance companies improve their customer experience by reducing the time it takes to process claims, which could lead to increased customer satisfaction and retention.

The success of such an app depends on several factors, including its accuracy, ease of use, and marketing strategy. It is essential to invest in developing an app that provides reliable and comprehensive fraud detection algorithms to ensure customer trust and satisfaction. Moreover, ensuring that the app is user-friendly and accessible can increase its adoption rate and potential revenue streams. Investing in a robust marketing strategy to reach out to the target audience can also help ensure the success of the app in a highly competitive market.

### 3.3.2 Technical Feasibility

Predicting claims frauds of payment using an app is technically feasible. The first step is to gather data on payment transactions, including details such as date, amount, payer, and payee. This data can be collected through various sources, such as electronic payment systems and financial institutions. Once the data is collected, it can be analyzed using machine learning algorithms to identify patterns and anomalies that are indicative of potential fraud. The machine learning models can be trained on historical data of fraudulent transactions and non-fraudulent transactions to learn the characteristics of fraud. The models can then be used to predict the likelihood of a transaction being fraudulent based on the patterns detected in the new data. The predictions can be made in real-time through an app that accesses the data and runs the models on the fly.

To make the app more effective, it is important to continuously monitor and update the machine learning models. This can be done by regularly analyzing the data to identify new patterns of fraud, and incorporating these patterns into the models. The app can also be designed to provide alerts when potentially fraudulent transactions are detected, allowing the user to take action to prevent the fraud. To ensure the security and privacy of the data, appropriate measures such as encryption, access control, and data anonymization can be implemented.

### 3.3.3 Social Feasibility

From a technical perspective, predicting claims frauds of payment using an app is feasible. The app can be designed to utilize machine learning algorithms that can learn from historical data to identify patterns and anomalies indicative of fraud. This requires access to a large dataset of claims, which can be obtained from insurance companies. Data preprocessing techniques such as feature extraction and dimensionality reduction can be applied to clean the data and reduce noise.

The app can also utilize various features such as image recognition, speech recognition, and natural language processing to extract information from different types of claims. The app can be designed to run on mobile devices or web-based platforms, providing flexibility and convenience to users. However, the app's success relies heavily on the accuracy and reliability of the algorithms used, which requires continuous monitoring and improvement. Therefore, regular updates and maintenance of the app's algorithms are crucial to ensure that it remains effective in detecting fraudulent claims.

## 3.4 System Specification

### 3.4.1 Hardware Specification

Operating Systems: Windows 7,8,9,10,11
Hard disk : 20GB
RAM :4GB(DDR4 or DDR5) and above.
Processor: i3(New gen) and Above

### 3.4.2 Software Specification

Operating Systems: Windows 7,8,9,10,11
Forntend: XML
Backend: Python,Kotlin
Frameworks: Pandas,scikit-learn
Database: MySQLite

### 3.4.3 Standards and Policies

**Android Studio**

This standard outlines the requirements for software quality characteristics and metrics. It includes criteria for evaluating the usability, reliability, and security of software, which are all important considerations for Android app development. Organizations that develop Android apps can use this standard to ensure that their apps meet these quality criteria.

**Standard Used:ISO/IEC 25010**

This standard outlines the requirements for an information security management system (ISMS). It provides a framework for managing and protecting sensitive information, including user data collected by Android apps. Organizations that develop Android apps can use this standard to ensure that they have appropriate controls in place to protect user privacy and data.

**Standard Used:ISO/IEC 27001**

# Chapter 4

# METHODOLOGY

## 4.1 General Architecture

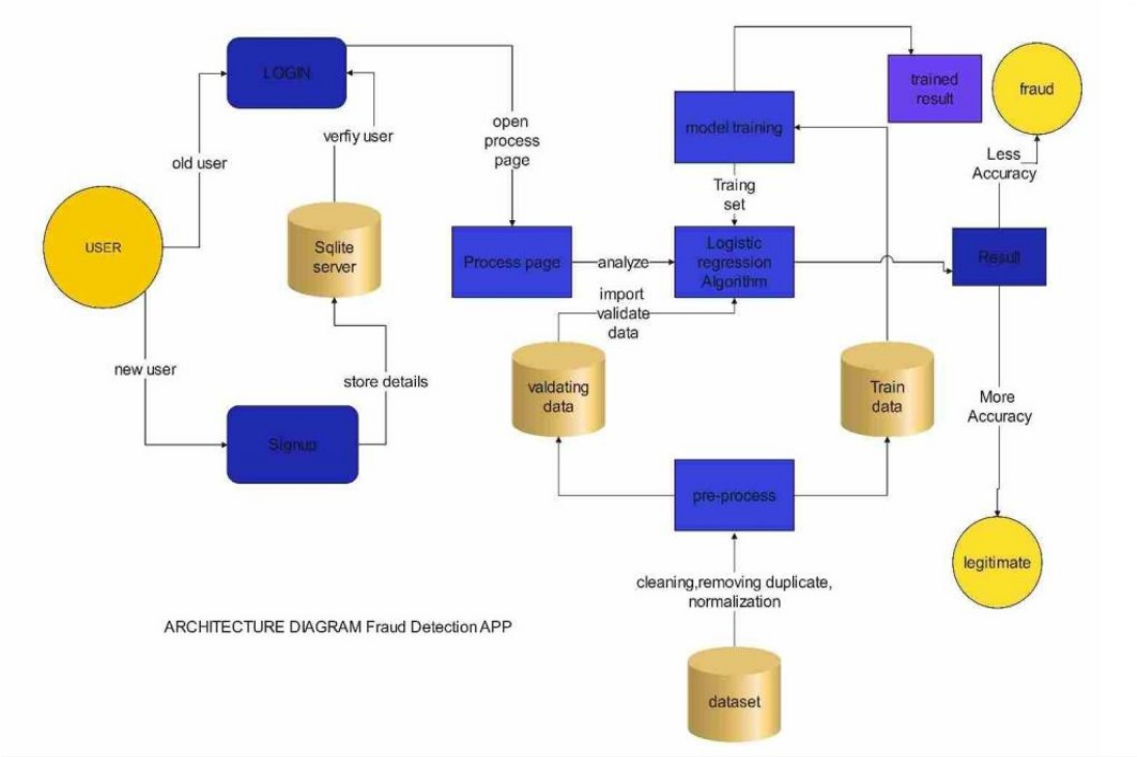



Figure 4.1: **Architecture for prevention of frauds and claims**

### Description

In figure 4.1 is the general architecture of the project which shows the complete working of our model, consisting of data collections, data pre-processing, training and validating the data. Where the inputs are username and password for logging in or signup, the details are stored or retrieved from data base. After importing using logistic regression algorithm it predicts whether the dataset is fraud or legitimate.

## 4.2 Design Phase

### 4.2.1 Data Flow Diagram

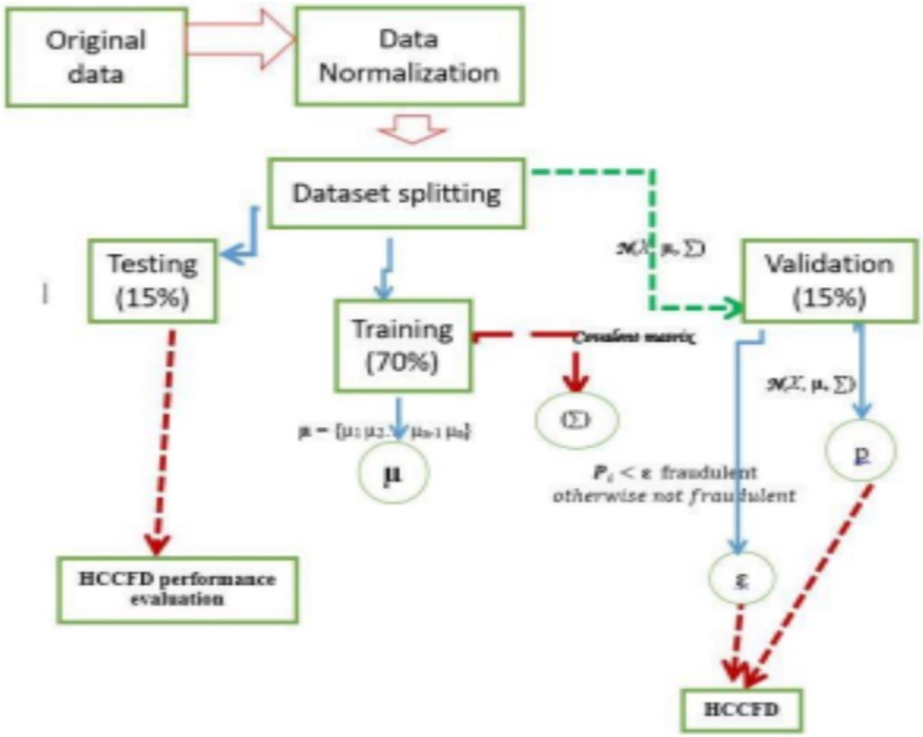



Figure 4.2: **Data flow diagram for prevention of frauds and claims**

## Description

In figure 4.2 The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data is normalized and it is divided into test and train data and test data will increase the performance of logistic regression.

### 4.2.2 Use Case Diagram

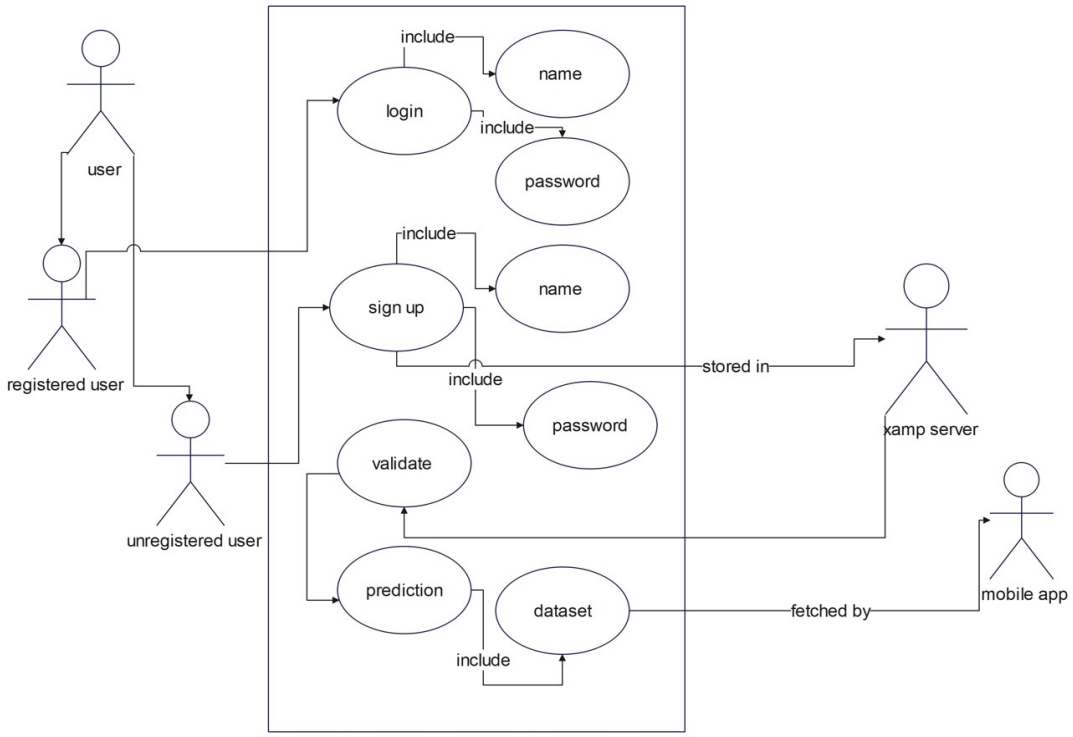



Figure 4.3: **Usecase diagram for prevention of frauds and claims**

## Description

In figure 4.3 consists of actors as user which are classified as registered and unregistered users along with actors xamp server and mobile app which perform certain operations(use cases) as shown in the box(system) between the actors which consists of login with name,passwords as sub-operations signup with name, passwords as sub-operations, along with usecases validate and prediction in the system. A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

### 4.2.3 Class Diagram

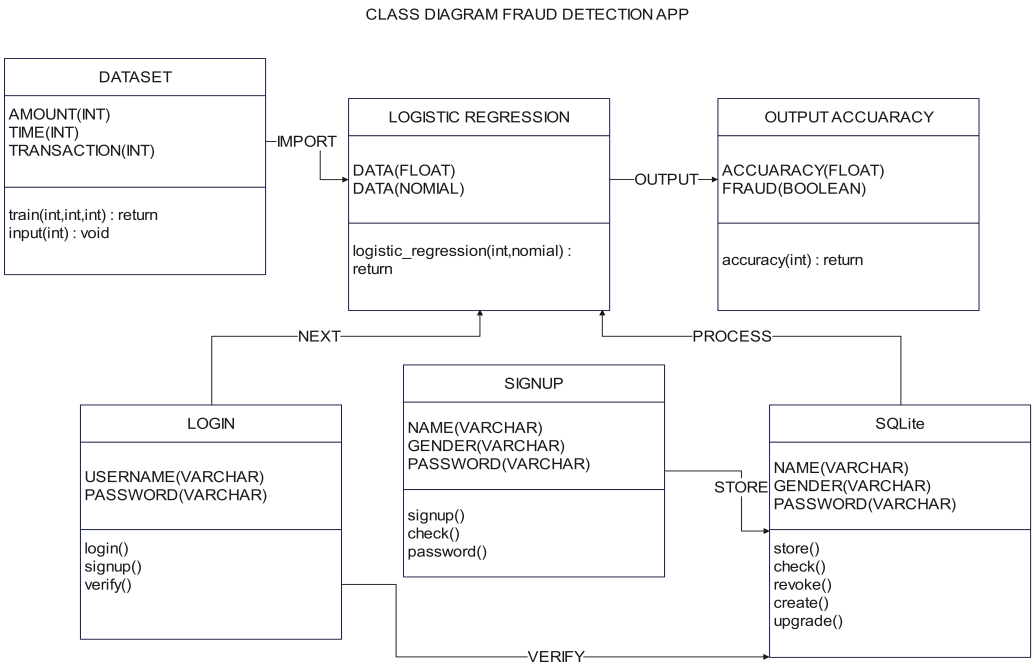



Figure 4.4: **Class diagram for prevention of frauds and claims**

## Description

In figure 4.4 there are 6 classes named dataset, logitic regression program, output accuracy, login, signup, xampserver with each class having its own qualifier eg.username(varchar) etc, and associations such as import, process, output etc. Which are all functions(operations). A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

### 4.2.4 Sequence Diagram

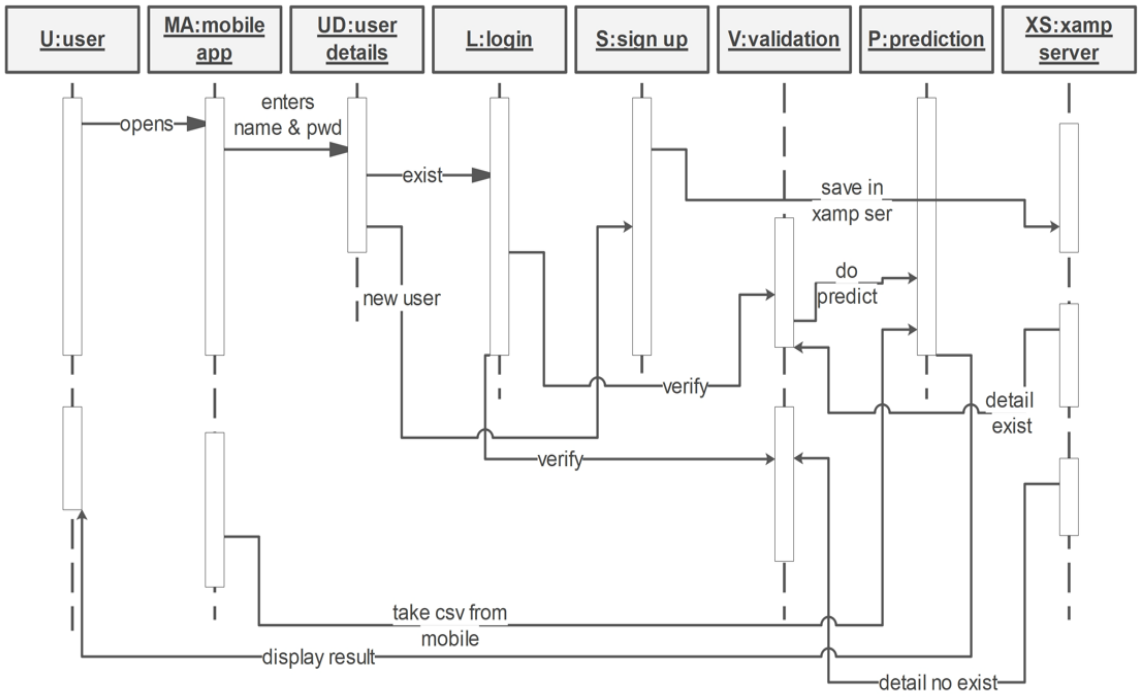



Figure 4.5: **Sequence diagram for prevention of frauds and claims**

## Description

In figure 4.5 sequence diagram consists of with lifelines as user, mobile app, user details, login, signup, validation, prediction, xamp server and a (thin rectangle on a lifeline) represents the period during which an element is performing an operation. The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively, along with call and return message which perform operations to the required lifeline.

### 4.2.5 Collaboration diagram

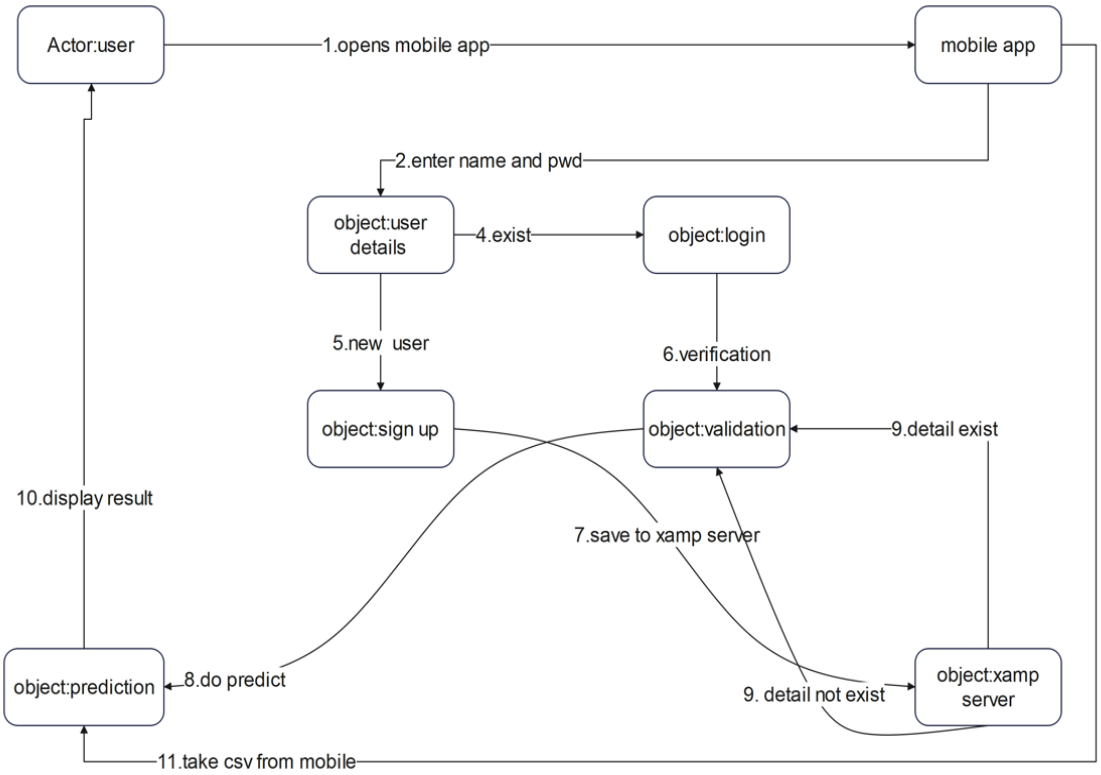



Figure 4.6: **Collaboration diagram for prevention of frauds and claims**

## Description

In figure 4.6 consists of actor user, xamp server and object classes such as mobile app, login, validation, signup, userdetails, prediction, where each object class passes an unique message to other classes, based on user's (actor) operations. The path of arrow displays the operation assigned to each object class, the xamp server stores the user details, the algorithm based is logistic regression

### 4.2.6 Activity Diagram

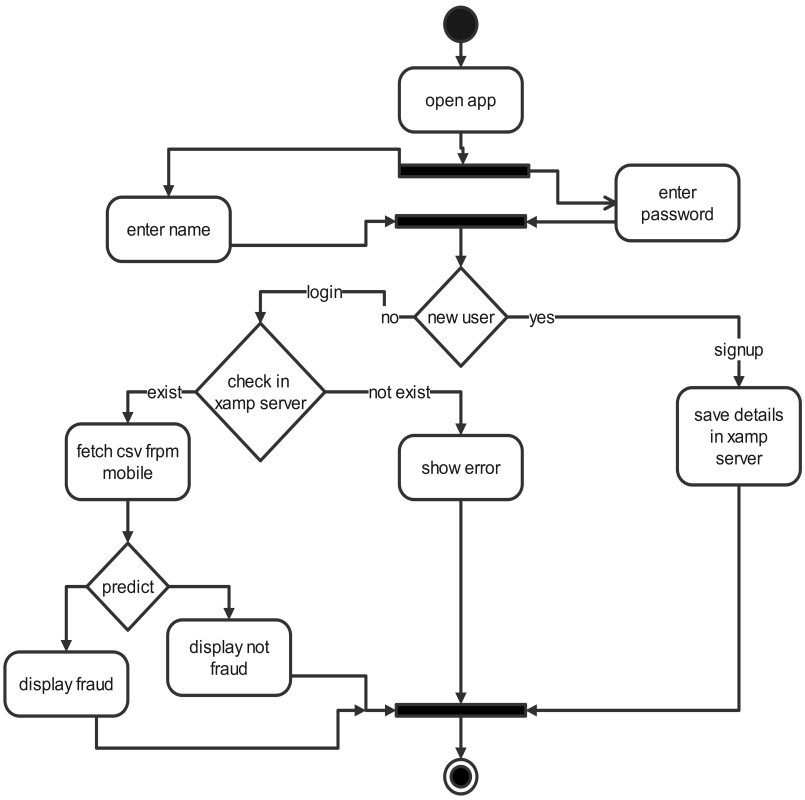



Figure 4.7: **Activity diagram for prevention of frauds and claims**

## Description

In figure 4.7 consists activity diagram where first the user should login or signup. Then the user have to import the dataset and click on run based on accuracy the dataset is legitimate or not.

## 4.3 Algorithm & Pseudo Code

### 4.3.1 Algorithm

Step-1:Start
Step-2:Enter user name and password
Step-3:If login goto Step-4 else if signup goto Step-9
Step-4:If details exist in SQLite server goto Step-5 else goto Step-9

Step-5:Import csv from file manager
Step-6:The dataset will be analysed using logistic regression in python program
Step-7:The dataset will divide the data into trained data and input data analyse the accuracy of the data set using pandas and sklearn frame work in python program
Step-8:Display the output in the activity page
Step-9:Display error saying invalid details goto step-11
Step-10:Write/save details in SQLite server
Step-11:Stop

#### 4.3.2 Pseudo Code

1: Input: Training data
2: Begin
3: For i = 1 to k
4: For each training data instance
5: Set the target value for the regression to $z_i = yi - P(1|dj)[P(1|dj)(1 - P(1|dj))$
6: Initialize the weight of instance$d_j to [P(1|dj) dot (1 - P(1|dj))]$
7: Finalize a $f(j)$ to the data with class value $(Z_j)$ and weight $(w_j)$
8: Classical label decision
9: Assign $(classlabel : 1)$ if P id ¿0.5, otherwise $(classlabel : 2)$
10: End
Psuedo code(fraud detection)
Input:-Traing data
Begin
data=loaddata()
cleandata=preprocessdata(data)
traindata, testdata, trainlabels, testlabels = train(cleandata, labels, testsize, random-state)
model = LogisticRegression()
model.fit(traindata, trainlabels)
predictions = model.predict(testdata)
accuracy = accuracyscore(testlabels, predictions)
precision = precisionscore(testlabels, predictions)
recall = recallscore(testlabels, predictions) f1 = f1score(testlabels, predictions)
suspicioustransactions = []

for i to predictions:
if (predictions[i] == 1):
suspicioustransactions.append(testdata[i])
savemodel(model)
End

## 4.4 Module Description

### 4.4.1 Module1: Frontend Development

To design the front end of APP using Android Studio and to work on login and signup page and collecting dataset. Collect and preprocess data to be used for training and testing the model. This can include cleaning, transforming, and encoding data as needed. Choose a suitable machine learning algorithm for fraud detection, based on the characteristics of the data and the specific use case.

### 4.4.2 Module2: Backend Development

To work on the programme of linear regression to detect whether the given data is fraud or not. Logistic regression is a binary classification algorithm that can be used to predict whether a transaction is fraudulent or not. It is a simple and efficient algorithm that is easy to implement.

### 4.4.3 Module3: Frontend Backend Integration

To integration of python to the android application. To transfer the file location to the python programme. We install a plugin Chaquopy is a Python integration library for Android applications. It allows you to embed Python code in your Android app, so you can take advantage of the extensive libraries available in the Python ecosystem. With Chaquopy, you can use Python to perform complex data processing, machine learning, and scientific computing tasks in your Android app. To use Chaquopy in your Android app, you need to add the Chaquopy dependency to your app's build.gradle file and configure the Python environment using the Chaquopy API.

## 4.5 Steps to execute/run/implement the project

### 4.5.1 Step1: Frontend Development

- Design the each activity of the App using XMl and Android Studio for login , signup, process, home page.
- By using kotlin program try to connect between the Activity by using intent function.
- Pre-Process the data for analyzing by using weka tool.
- Working on Logistic regression algorithm.

### 4.5.2 Step2: Backend Development

- In module two worked on the Logistic Regression, try to train the algorithm with the different datasets and try find Accuracy of the algorithm. Based on the accuracy the fraud is identified.
- Working on the login and signup process by using the Sqlite database in android studio.It is local database.
- Try to import the csv file to the android app and try to find location of the csv file.

### 4.5.3 Step3: Frontend  Backend Integration

- To install the Chaquopy we need add some tags in build gradle ,for installing the current version id of the plugin in top-level build gradle.
- To install the Chaquopy we need add some tags in build gradle ,for installing the plugins for the use of python in android app in module-level build gradle.
- Install python libraries using latest version of pip along with pandas  scikitlearn (sklearn).
- To Creating the project of python in android studio.
- Invoke the python file to kotlin program by the three statements by getting Instance,Module  calling the attribute.

# Chapter 5

# IMPLEMENTATION AND TESTING

## 5.1 Input and Output

### 5.1.1 Input Design

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Transactic | Amount ($ | Time (secc | class | | | |
| 2 | 1 | 10 | 100 | 0 | | | |
| 3 | 2 | 20 | 200 | 0 | | | |
| 4 | 3 | 50 | 300 | 1 | | | |
| 5 | 4 | 15 | 400 | 0 | | | |
| 6 | 5 | 30 | 500 | 1 | | | |
| 7 | 6 | 25 | 600 | 0 | | | |
| 8 | 7 | 12 | 700 | 1 | | | |
| 9 | 8 | 35 | 800 | 0 | | | |
| 10 | 9 | 45 | 900 | 1 | | | |
| 11 | 10 | 18 | 1000 | 0 | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |

Figure 5.1: **Input data for the prevention frauds and claims**

## Description

In figure 5.1 the dataset usually provides an overview of the data contained in the dataset, including the data was collected, any preprocessing or cleanup done, and any relevant information, effects, or variables. The description may include information about the type of data, such as whether it is structured or unstructured, and any limitations or biases that may exist in the data. Additionally, a note can provide important information or sources for the data, as well as instructions data the data has the amount, time, transaction.

### 5.1.2 Output Design

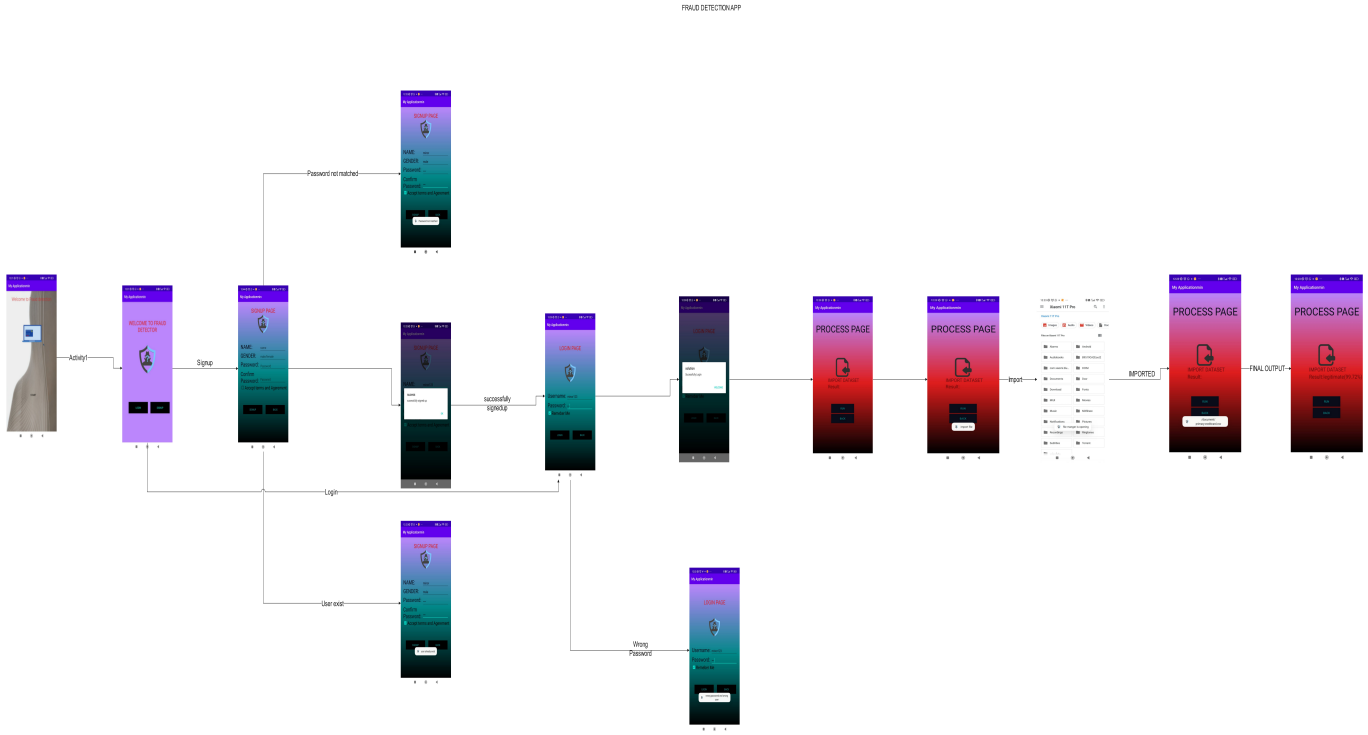



Figure 5.2: **Output data for the prevention of frauds and claims**

## Description

In figure 5.2 shows the output of the analysis of the data and show the weather the data has the legitimate or not by using the accuracy of dataset.If the dataset accuracy is less than it is not legitimate. If the dataset accuracy is more than the dataset is legitimate.

## 5.2 Testing

As the project is on bit large scale, we always need testing to make it successful. If each component work properly in all respect and gives desired output for all kind of inputs then project is said to be successful. So the conclusion is-to make the project successful, it needs to be tested.

The testing done here was System Testing checking whether the user requirements were satisfied. The code for the new system has been written completely using python as the coding language, Django as the interface for front-end designing. The

new system has been tested well with the help of the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms has been found to be very much in accordance with the actual flow of data.

## 5.3 Types of Testing

### 5.3.1 Unit testing

Unit testing is an important part of software development that involves testing individually or of code pieces to ensure that the behaves correctly and delivers the expected benefit. The output of the Frontend on the APP analysis of the data and show the weather the data has the legitimate or not by using the accuracy of dataset.If the dataset accuracy is less than it is not legitimate . If the dataset accuracy is more than the dataset is legitimate.

**Input**

```
process.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/backgroun11"
    tools:context=".procees">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="PROCESS PAGE"
        android:textSize="50dp"
        android:textColor="@color/blc"
        android:textAlignment="center"
        android:layout_marginTop="50dp"

        />
 <ImageView
     android:layout_width="match_parent"
```

```
        android:layout_height="100dp"
        android:layout_marginTop="120dp"
        android:src="@drawable/im"
        android:id="@+id/ipl"

    />
  <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="IMPORT DATASET"
        android:textSize="25dp"
        android:layout_marginLeft="100dp"
       />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Result:"
         android:textSize="25dp"
         android:layout_marginLeft="100dp"
         />
    <TextView
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:text=""
         android:textSize="25dp"
         android:layout_marginLeft="1dp"
         android:id="@+id/wri"
         />
</LinearLayout>
    <Button
         android:layout_width="150dp"
         android:layout_height="wrap_content"
         android:layout_marginLeft="120dp"
         android:layout_marginTop="75dp"
         android:text="Run"
         android:background="@color/blc"
         android:textColor="@color/teal_700"
         android:id="@+id/runnn"
         />
    <Button
         android:layout_width="150dp"
         android:layout_height="wrap_content"
         android:layout_marginLeft="120dp"
         android:layout_marginTop="5dp"
         android:text="Back"
         android:background="@color/blc"
```

```
        android:textColor="@color/teal_700"
        android:id="@+id/bm"
        />

</LinearLayout>
Back end
process.kt

package com.minor.myapplicationmin

import android.annotation.SuppressLint
import android.app.Activity
import android.content.Context
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.annotation.RequiresApi
import android.support.v4.app.ActivityCompat
import android.support.v4.content.ContextCompat
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Button
import android.widget.ImageView
import android.widget.Toast
import java.io.File
import java.util.jar.Manifest
import android.net.Uri
import android.provider.DocumentsContract
import android.util.Log;
import android.widget.TextView
import com.chaquo.python.Python
import com.chaquo.python.android.AndroidPlatform
import java.io.BufferedReader
import java.io.InputStreamReader


class procees : AppCompatActivity() {
    private val PICK_FILE_REQUEST = 1
    private var mo=false
    private var locationsd= ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_procees)
        val dkd = findViewById<Button>(R.id.bm)
        val dwf = findViewById<ImageView>(R.id.ipl)
        val nbk = findViewById<Button>(R.id.runnn)

```

```
124         nbk.setOnClickListener {
125
126
127             // Pass the file path to the Python code using Chaquopy
128             if(mo == true) {
129
130
131                 val result = Python.getInstance().getModule("test2").callAttr("wwe",locationsd)
132                 val loe = findViewById<TextView>(R.id.wri)
133                 loe.setText(result.toString())
134             }
135             else{
136                 Toast.makeText(this,"import file",Toast.LENGTH_LONG).show()
137             }
138
139         }
140
141         dkd.setOnClickListener {
142
143             val ite = Intent(this, MainActivity::class.java)
144             startActivity(ite)
145         }
146
147         dwf.setOnClickListener { true
148             mo = true
149             Toast.makeText(this, "file manger is opening", Toast.LENGTH_LONG).show()
150             val intent = Intent(Intent.ACTION_GET_CONTENT)
151             intent.type = "*/*"
152             startActivityForResult(intent, 1)
153         }
154
155     }
156     override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
157         super.onActivityResult(requestCode, resultCode, data)
158         if (resultCode == RESULT_OK) {
159             if (requestCode == 1) {
160                 val uri = data?.data
161                 val path :String = this.filesDir.absolutePath
162
163 //                val file : File = File(path+)
164                 val filePath = uri?.path
165                 if (filePath != null) {
166                     Log.d("data uri ---->",filePath)
167
168                 }
169                 Toast.makeText(this,filePath,Toast.LENGTH_LONG).show()
170                 if (!Python.isStarted()) {
171                     Python.start(AndroidPlatform(this))
172                 }
173                // var locationsd = filePath
```

```
                //val result = Python.getInstance().getModule("test").callAttr("mark",locationsd)
                //val loe = findViewById<TextView>(R.id.wri)
               // loe.setText(result.toString())


                // Pass the file path to the Python code using Chaquopy


            }
        }
    }
}


detection.py
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

# Load data
data = pd.read_csv(r'C:\Users\Prem Kumar\OneDrive\Desktop\minor project\creditcard.csv')

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data.drop('Class', axis=1), data['Class'],
    random_state=0)

# Train a logistic regression model
lr = LogisticRegression(max_iter=10000).fit(X_train, y_train)

# Evaluate the model on the testing set
score = lr.score(X_test, y_test)

if score < 0.5:
    print('The model is worse than guessing! having fraud')
else:
    print(f'legatimate The model accuracy is {score*100:.2f}%')
```

**Test result**


Figure 5.3: **Result FrontEnd**

## Description

In figure 5.3 shows the output of the Frontend on the APP analysis of the data and show the weather the data has the legitimate or not by using the accuracy of dataset.If the dataset accuracy is less than it is not legitimate . If the dataset accuracy is more than the dataset is legitimate.

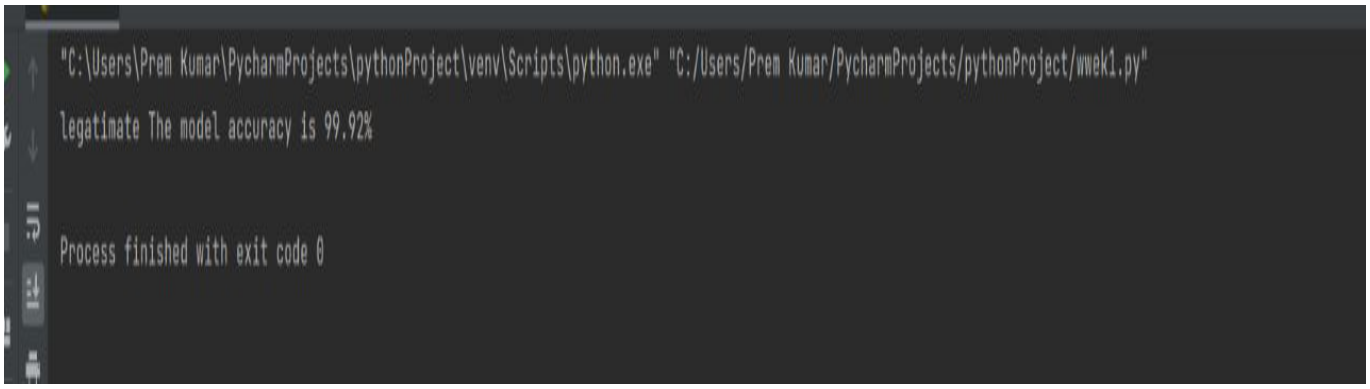



Figure 5.4: **Result for python page**

## Description

In figure 5.4 shows the output of the Backend.The output is shown by the python program.The backend is tested by various datasets given as input to the Logistic Regression program testing various time complexity and accuracy of output.

### 5.3.2 Integration testing

Integration testing is a type of software testing that involves testing how different components or systems interact with each other to ensure that they work correctly and produce the expected results. The output of the Frontend is integrate with the backend and the SQLite database is used for login and signup and analysis is to done useing the python program and show the weather the data has the legitimate or not by using the accuracy of dataset.If the dataset accuracy is less than it is not legitimate . If the dataset accuracy is more than the dataset is legitimate.

**Input**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/background"
    tools:context=".Loginpage">
```

```
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SIGNUP PAGE"
        android:textColor="@color/red"
        android:textSize="30dp"
        android:textAlignment="center"
        android:layout_marginTop="25dp"

        />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:src="@drawable/logo"

        />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <TextView
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:text="NAME:"
            android:textSize="30dp"
            android:layout_marginTop="50dp"
            android:layout_marginLeft="20dp"
            android:textColor="@color/blc"

            />
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="50dp"
            android:hint="name"
            android:id="@+id/namet"
            android:layout_marginRight="20dp"
            android:textColor="@color/blc"
            android:inputType="text"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
```

```
60
61          <TextView
62              android:layout_width="152dp"
63              android:layout_height="wrap_content"
64              android:layout_marginLeft="20dp"
65              android:layout_marginTop="1dp"
66              android:text="Password:"
67              android:textColor="@color/blc"
68              android:textSize="30dp"
69
70              />
71          <EditText
72              android:layout_width="match_parent"
73              android:layout_height="wrap_content"
74              android:layout_marginTop="5dp"
75              android:id="@+id/pet"
76              android:hint="Password"
77              android:layout_marginRight="20dp"
78              android:textColor="@color/blc"
79              android:inputType="textPassword"/>
80      </LinearLayout>
81      <LinearLayout
82          android:layout_width="match_parent"
83          android:layout_height="wrap_content"
84          android:orientation="horizontal" >
85          <TextView
86              android:layout_width="150dp"
87              android:layout_height="wrap_content"
88              android:text="Confirm Password:"
89              android:textSize="30dp"
90              android:layout_marginTop="1dp"
91              android:layout_marginLeft="20dp"
92              android:textColor="@color/blc"
93
94              />
95          <EditText
96              android:layout_width="match_parent"
97              android:layout_height="wrap_content"
98              android:layout_marginTop="25dp"
99              android:id="@+id/cp"
100             android:hint="Password"
101             android:layout_marginRight="20dp"
102             android:textColor="@color/blc"
103             android:inputType="textPassword"/>
104     </LinearLayout>
105     <CheckBox
106         android:layout_width="match_parent"
107         android:layout_height="wrap_content"
108         android:layout_marginLeft="20dp"
109         android:text="Accept terms and Agerement"
```

```
        android:textSize="25dp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:layout_width="150dp"
            android:id="@+id/signup"
            android:layout_height="wrap_content"
            android:layout_marginLeft="40dp"
            android:text="signup"
            android:background="@drawable/buttonstyle"
            android:textColor="@color/teal_700"
            android:layout_marginTop="75dp"/>
        <Button
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="25dp"
            android:layout_marginTop="75dp"
            android:text="Back"
            android:background="@drawable/buttonstyle"
            android:textColor="@color/teal_700"
            android:id="@+id/ck"
            />
    </LinearLayout>

</LinearLayout>

process.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/backgroun11"
    tools:context=".procees">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="PROCESS PAGE"
        android:textSize="50dp"
        android:textColor="@color/blc"
        android:textAlignment="center"
        android:layout_marginTop="50dp"

        />
 <ImageView
```

```
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:layout_marginTop="120dp"
        android:src="@drawable/im"
        android:id="@+id/ip1"

      />
    <TextView
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:text="IMPORT DATASET"
         android:textSize="25dp"
         android:layout_marginLeft="100dp"
        />
<LinearLayout
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:orientation="horizontal" >
     <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="Result:"
          android:textSize="25dp"
          android:layout_marginLeft="100dp"
          />
     <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:text=""
          android:textSize="25dp"
          android:layout_marginLeft="1dp"
          android:id="@+id/wri"
          />
</LinearLayout>

          />
     <Button
          android:layout_width="150dp"
          android:layout_height="wrap_content"
          android:layout_marginLeft="120dp"
          android:layout_marginTop="5dp"
          android:text="Back"
          android:background="@color/blc"
          android:textColor="@color/teal_700"
          android:id="@+id/bm"
          />

</LinearLayout>

import pandas as pd
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from io import StringIO
from os.path import dirname, join
def mark(filename):

    data = pd.read_csv(filename)

    X_train, X_test, y_train, y_test = train_test_split(data.drop('Class', axis=1), data['Class'],
        random_state=0)

    lr = LogisticRegression(max_iter=10000).fit(X_train, y_train)

    score = lr.score(X_test, y_test)
    if score < 0.5:
        return 'The model is worse than guessing! and fraud'
    else:
        return  f'legitimate The model accuracy is {score*100:.2f}% and legitimate'
```

**Test result**

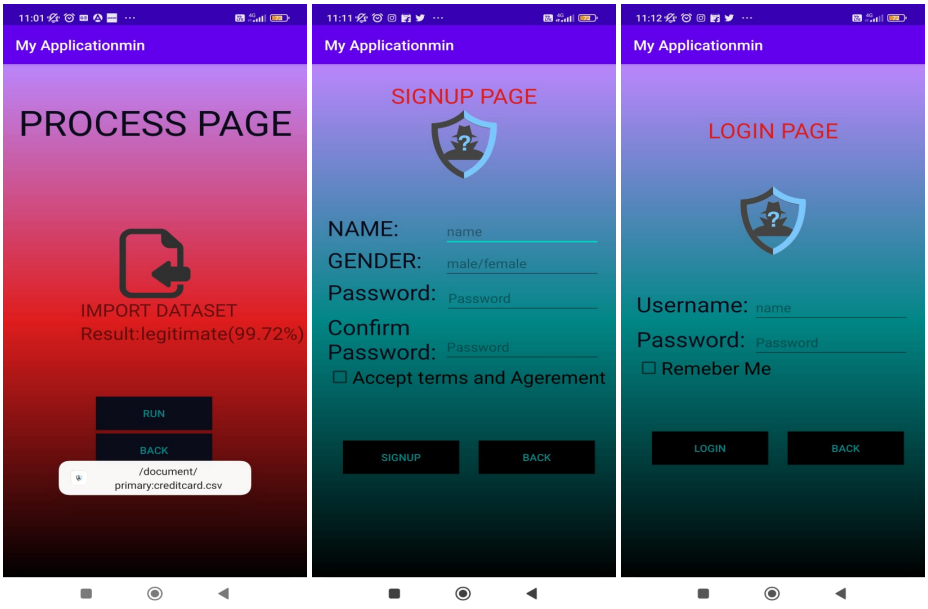



Figure 5.5: **Frontend pages**

### 5.3.3 System testing

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences.Shows the output of the system testing of the project.The APP is build from the android studio.Try to install in different android devices and checking the APP is installing properly or not.The above figure shows the APP is installed successfully and working properly.

**Test Result**

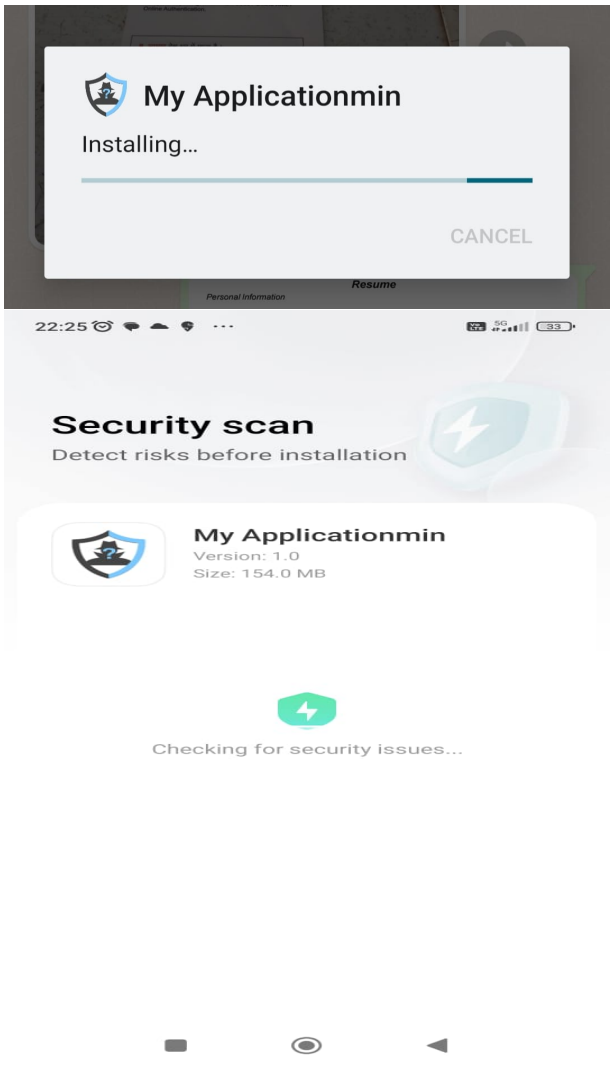



Figure 5.6: **Installing APP to Android Device for System Testing**

# Chapter 6

# RESULTS AND DISCUSSIONS

## 6.1 Efficiency of the Proposed System

The efficiency of using an app to predict claims fraud of payment depends on the accuracy of the algorithms used for fraud detection and the quality of the data used to train those algorithms. Machine learning algorithms can be trained to analyze patterns and detect anomalies in payment transactions, which can help identify potential fraud. For example, if a payment is being made from an unusual location or if the payment amount is significantly higher than usual, the algorithm may flag it as suspicious.

To increase the efficiency of fraud detection, it's important to train the algorithm using high-quality data. This means having access to a large volume of accurate and relevant data about past fraud cases, as well as non-fraudulent payment transactions. Additionally, the algorithm should be continually updated and refined based on new data and feedback. However, it's important to note that no fraud detection algorithm can be hundred percent accurate, and there is always a risk of false positives (flagging legitimate payments as fraud) or false negatives (failing to detect actual fraud). Therefore, it's important to have a human review process in place to investigate any suspicious transactions flagged by the algorithm and to ensure that legitimate claims are not unnecessarily delayed or denied.

## 6.2 Comparison of Existing and Proposed System

**Existing system: (Predictive Analysis)**

In case of credit card fraud detection, the existing system is detecting the fraud after fraud has been happen. Existing system maintain the large amount of data when customer comes to know about inconsistency in transaction, he/she made complaint and then fraud detection system start it working. It first tries to detect that fraud has

actually occur after that it starts to track fraud location and so on. In case of existing system there is no confirmation of recovery of fraud and customers satisfaction.

**Proposed system: (Machine Learning algorithm)**

The proposed system aims to improve upon the existing system by using more advanced machine learning algorithms and techniques. One approach could be to use Logistic Regression or Deep Learning models, which are more complex than other algorithms like decision tree, SVM etc. Regression and can better capture the patterns in the data. Regression, in particular, is an ensemble learning algorithm that combines multiple decision trees to make more accurate predictions. Another approach could be to use unsupervised learning algorithms like Isolation Regression or Local Outlier Factor for anomaly detection. These algorithms can detect outliers or anomalies in the data, which could indicate fraudulent transactions. This approach could be combined with a supervised learning algorithm to improve the accuracy of the predictions.

## 6.3 Sample Code

```
import android.content.DialogInterface
import android.content.Intent
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.support.v7.app.AlertDialog
import android.text.TextUtils
import android.widget.Button
import android.widget.EditText
import android.widget.Toast

class Loginpage : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_loginpage)
        val lo = findViewById<Button>(R.id.lop)
        val ba = findViewById<Button>(R.id.ba)
        val naet = findViewById<EditText>(R.id.unm)
        val pass = findViewById<EditText>(R.id.sin)

         val db = Dbhelper(this)

        ba.setOnClickListener {
            val itb = Intent(this,MainActivity::class.java)
            startActivity(itb)
        }
```

```
        lo.setOnClickListener {
            val pwffs = naet.text.toString()
            val seddf = pass.text.toString()
            if(TextUtils.isEmpty(pwffs) && TextUtils.isEmpty(seddf)){
                Toast.makeText(this,"fill the fields",Toast.LENGTH_SHORT).show()

            }
            else {
                    val khh = db.checkdet(pwffs, seddf)
                    if(khh == true){
                        val mou = AlertDialog.Builder(this)
                        mou.setTitle("solution")
                        mou.setMessage("Sucessfully Login")
                        mou.setPositiveButton("welcome",DialogInterface.OnClickListener(){
                            dialogInterface, i ->
                            val duvre = Intent(this,procees::class.java)
                            startActivity(duvre)
                        })
                        mou.show()


                    }
                else{
                      Toast.makeText(this,"wrong password and wrong user",Toast.LENGTH_SHORT).show()
                    }
            }
        }

    }
}
```

welcome.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Welcome"
    android:background="@drawable/backwelcome"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Welcome to Fraud detection"
        android:fontFamily="sans-serif-light"

    <ImageView
```

```
        android:layout_marginTop="50dp"
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:src="@drawable/img"
        android:id="@+id/sd"
        android:layout_marginLeft="85dp"

        />

    <Button
        android:layout_width="150dp"
        android:layout_height="75dp"
        android:text="START"
        android:layout_marginTop="150dp"
        android:layout_marginLeft="135dp"
        android:background="@drawable/butan"

        android:id="@+id/sf"

        />


</LinearLayout>
```

**Output**

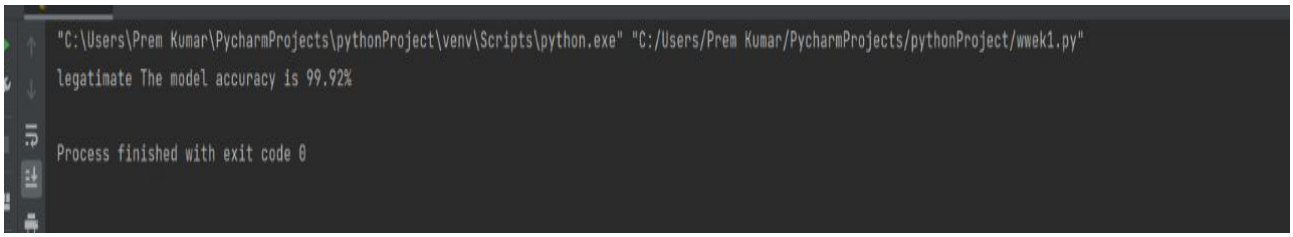



Figure 6.1: **Output(Backend)**

## Description

In figure 6.1 the output is shown by the python program.To find the the dataset is legitimate or fraud is can be find out by the logistic regression.The logistic regression was implemented by the python program using framework pandas and sklearn.The program will read CSV file and give output has the accuracy.If accuracy is more the dataset is legitimate else accuracy is less than it is fraud.

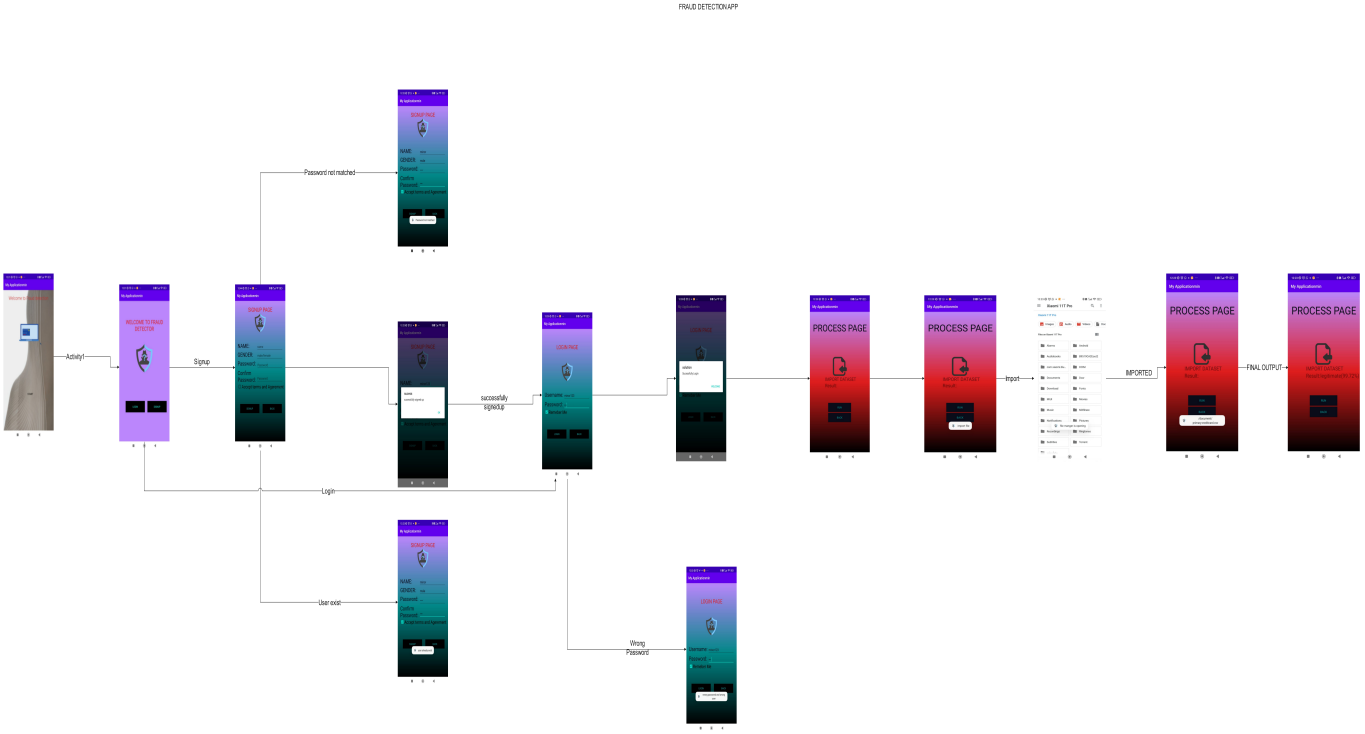



Figure 6.2: **Output(Frontend APP Pages)**

## Description

In figure 6.2 shows the output of the Frontend.The image shows the transaction between the activities of each page in the App. App is open the home page will popup and click the start button and move to the login or signup page.If you already a existing user then login else signup.After signup again login and moves to the process activity and import the file from the file manger and click on run button than the final output will be displayed on the screen.The final output contain accuracy and weather the data is legitimate or not.

# Chapter 7

# CONCLUSION AND FUTURE ENHANCEMENTS

## 7.1 Conclusion

As for the fraud detection app is useful for finding the losses due to fraudulent activities .These apps often utilize advanced technologies like machine learning, data analytics, and artificial intelligence to identify patterns and anomalies in financial transactions that may indicate fraudulent behavior. To ensure the effectiveness of a fraud detection app, it is essential to have accurate and up-to-date data, robust algorithms, and a comprehensive understanding of the type of fraud being targeted. Additionally, regular updates and improvements to the app are necessary to keep up with evolving fraudulent activities. The Logistic regression algorithm is used in this project and it is more efficiency and fast compare to other algorithms like Decision tress, Random forest, Neural networks, SVM etc.

## 7.2 Future Enhancements

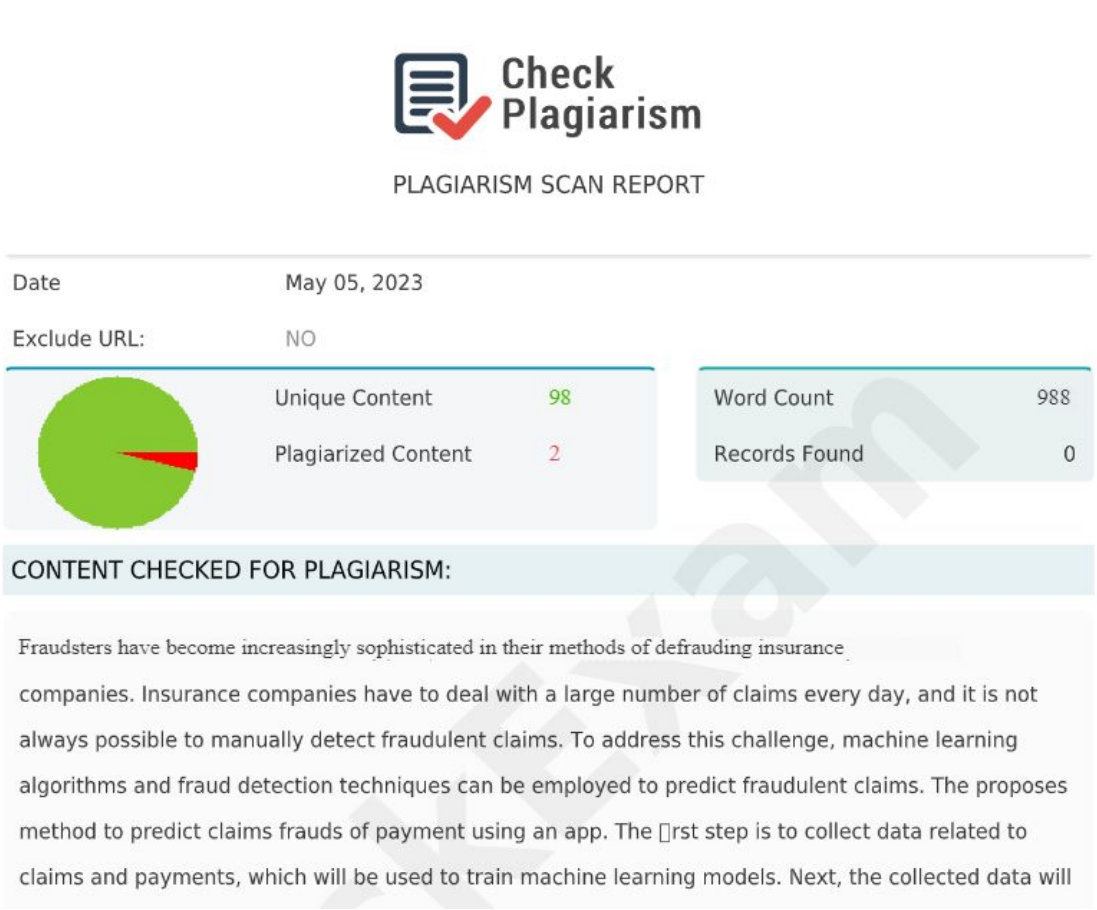
To enhance the ”Prediction of Claims and Frauds Payment Using App,” several future enhancements could be considered. One possible enhancement would be to incorporate machine learning. This would involve creating predictive models that can analyze historical data to identify patterns and predict future instances of claims fraud. Machine learning algorithms could be trained on large datasets of past fraud cases, which would enable them to recognize patterns that are difficult for humans to detect. As the app receives more data, the predictive models could be refined and improved, resulting in more accurate fraud detection.

Another potential enhancement could be to incorporate artificial intelligence (AI) technologies such as natural language processing (NLP) and image recognition. These

technologies could be used to automatically extract data from unstructured sources such as claim documents and images. This would enable the app to analyze large amounts of data quickly and accurately, improving the efficiency and accuracy of the fraud detection process. Additionally, integrating blockchain technology could help to ensure the security and integrity of the data being used to identify fraudulent claims. Blockchain technology can be used to create an immutable record of all transactions, making it nearly impossible for fraudulent claims to be processed. This would help to reduce the incidence of fraud and increase the trust and confidence of users in the app.Overall, these enhancements would help to improve the accuracy and efficiency.

# Chapter 8

# PLAGIARISM REPORT



Check Plagiarism

PLAGIARISM SCAN REPORT

| | |
|---|---|
| Date | May 05, 2023 |
| Exclude URL: | NO |



| | | | |
|---|---|---|---|
| Unique Content | 98 | Word Count | 988 |
| Plagiarized Content | 2 | Records Found | 0 |

CONTENT CHECKED FOR PLAGIARISM:

Fraudsters have become increasingly sophisticated in their methods of defrauding insurance companies. Insurance companies have to deal with a large number of claims every day, and it is not always possible to manually detect fraudulent claims. To address this challenge, machine learning algorithms and fraud detection techniques can be employed to predict fraudulent claims. The proposes method to predict claims frauds of payment using an app. The □rst step is to collect data related to claims and payments, which will be used to train machine learning models. Next, the collected data will

Figure 8.1: **PLAGIARISM REPORT**

# Chapter 9

# SOURCE CODE & POSTER PRESENTATION

## 9.1 Source Code

```
frontend



process.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/backgroun11"
    tools:context=".procees">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="PROCESS PAGE"
        android:textSize="50dp"
        android:textColor="@color/blc"
        android:textAlignment="center"
        android:layout_marginTop="50dp"

        />
 <ImageView
      android:layout_width="match_parent"
      android:layout_height="100dp"
      android:layout_marginTop="120dp"
      android:src="@drawable/im"
      android:id="@+id/ip1"

    />
  <TextView
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="IMPORT DATASET"
        android:textSize="25dp"
        android:layout_marginLeft="100dp"
        />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Result:"
        android:textSize="25dp"
        android:layout_marginLeft="100dp"
        />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="25dp"
        android:layout_marginLeft="1dp"
        android:id="@+id/wri"
        />
</LinearLayout>
    <Button
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="120dp"
        android:layout_marginTop="75dp"
        android:text="Run"
        android:background="@color/blc"
        android:textColor="@color/teal_700"
        android:id="@+id/runnn"
        />
    <Button
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="120dp"
        android:layout_marginTop="5dp"
        android:text="Back"
        android:background="@color/blc"
        android:textColor="@color/teal_700"
        android:id="@+id/bm"
        />

</LinearLayout>
=====================================================

```

process.kt

```
package com.minor.myapplicationmin

import android.annotation.SuppressLint
import android.app.Activity
import android.content.Context
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.annotation.RequiresApi
import android.support.v4.app.ActivityCompat
import android.support.v4.content.ContextCompat
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Button
import android.widget.ImageView
import android.widget.Toast
import java.io.File
import java.util.jar.Manifest
import android.net.Uri
import android.provider.DocumentsContract
import android.util.Log;
import android.widget.TextView
import com.chaquo.python.Python
import com.chaquo.python.android.AndroidPlatform
import java.io.BufferedReader
import java.io.InputStreamReader


class procees : AppCompatActivity() {
    private val PICK_FILE_REQUEST = 1
    private var mo=false
    private var locationsd= ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_procees)
        val dkd = findViewById<Button>(R.id.bm)
        val dwf = findViewById<ImageView>(R.id.ipl)
        val nbk = findViewById<Button>(R.id.runnn)

        nbk.setOnClickListener {


            // Pass the file path to the Python code using Chaquopy
            if(mo == true) {

```

```
136
137                 val result = Python.getInstance().getModule("test2").callAttr("wwe",locationsd)
138                 val loe = findViewById<TextView>(R.id.wri)
139                 loe.setText(result.toString())
140             }
141             else{
142                 Toast.makeText(this,"import file",Toast.LENGTH_LONG).show()
143             }
144
145         }
146
147         dkd.setOnClickListener {
148
149             val ite = Intent(this, MainActivity::class.java)
150             startActivity(ite)
151         }
152
153         dwf.setOnClickListener { true
154             mo = true
155             Toast.makeText(this, "file manger is opening", Toast.LENGTH_LONG).show()
156             val intent = Intent(Intent.ACTION_GET_CONTENT)
157             intent.type = "*/*"
158             startActivityForResult(intent, 1)
159         }
160
161     }
162     override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
163         super.onActivityResult(requestCode, resultCode, data)
164         if (resultCode == RESULT_OK) {
165             if (requestCode == 1) {
166                 val uri = data?.data
167                 val path :String = this.filesDir.absolutePath
168
169 //                val file : File = File(path+)
170                 val filePath = uri?.path
171                 if (filePath != null) {
172                     Log.d("data uri ---->",filePath)
173
174                 }
175                 Toast.makeText(this,filePath,Toast.LENGTH_LONG).show()
176                 if (!Python.isStarted()) {
177                     Python.start(AndroidPlatform(this))
178                 }
179                // var locationsd = filePath
180                 //val result = Python.getInstance().getModule("test").callAttr("mark",locationsd)
181                 //val loe = findViewById<TextView>(R.id.wri)
182                // loe.setText(result.toString())
183
184
185                 // Pass the file path to the Python code using Chaquopy
```

```
186
187
188                 }
189             }
190         }
191 }
192
193
194
195
196
197 ==========================================================================================
198
199
200 DBhelper
201
202 package com.minor.myapplicationmin
203
204 import android.content.ContentValues
205 import android.content.Context
206 import android.database.sqlite.SQLiteDatabase
207 import android.database.sqlite.SQLiteOpenHelper
208
209 class Dbhelper(context:Context):SQLiteOpenHelper(context,"user",null,1) {
210
211     override fun onCreate(db: SQLiteDatabase?) {
212         db?.execSQL("create table userdata(name varchar primary key,gender varchar,password varchar,
                confirmpassword varchar)")
213     }
214
215     override fun onUpgrade(db: SQLiteDatabase?, p1: Int, p2: Int) {
216         db?.execSQL("drop table if exists userdata")
217     }
218     fun insertdata(name:String,gender:String,password:String,confirmpassword:String):Boolean {
219      val db = this.writableDatabase
220      val lo = ContentValues()
221      lo.put("name",name)
222         lo.put("gender",gender)
223         lo.put("password",password)
224         lo.put("confirmpassword",confirmpassword)
225         val res = db.insert("userdata",null,lo)
226         if(res==-1.toLong()){
227             return false
228         }
229         return true
230
231     }
232     fun checkdet(name:String,password:String):Boolean{
233         val db = this.writableDatabase
234         val query = "select * from userdata where name='$name' and password='$password'"
```

```
        val cursor = db.rawQuery(query,null)
        if(cursor.count<=0){
            cursor.close()
            return false
        }
        cursor.close()
        return true

    }
}

================================================================================================
backend python code
================================================================================================

import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from io import StringIO
from os.path import dirname, join
def mark(filename):

    data = pd.read_csv(filename)

    X_train, X_test, y_train, y_test = train_test_split(data.drop('Class', axis=1), data['Class'],
        random_state=0)

    lr = LogisticRegression(max_iter=10000).fit(X_train, y_train)

    score = lr.score(X_test, y_test)
    if score < 0.5:
        return 'The model is worse than guessing! and fraud'
    else:
        return  f'legitimate The model accuracy is {score*100:.2f}% and legitimate'



================================================================================================
```

## 9.2 Poster Presentation

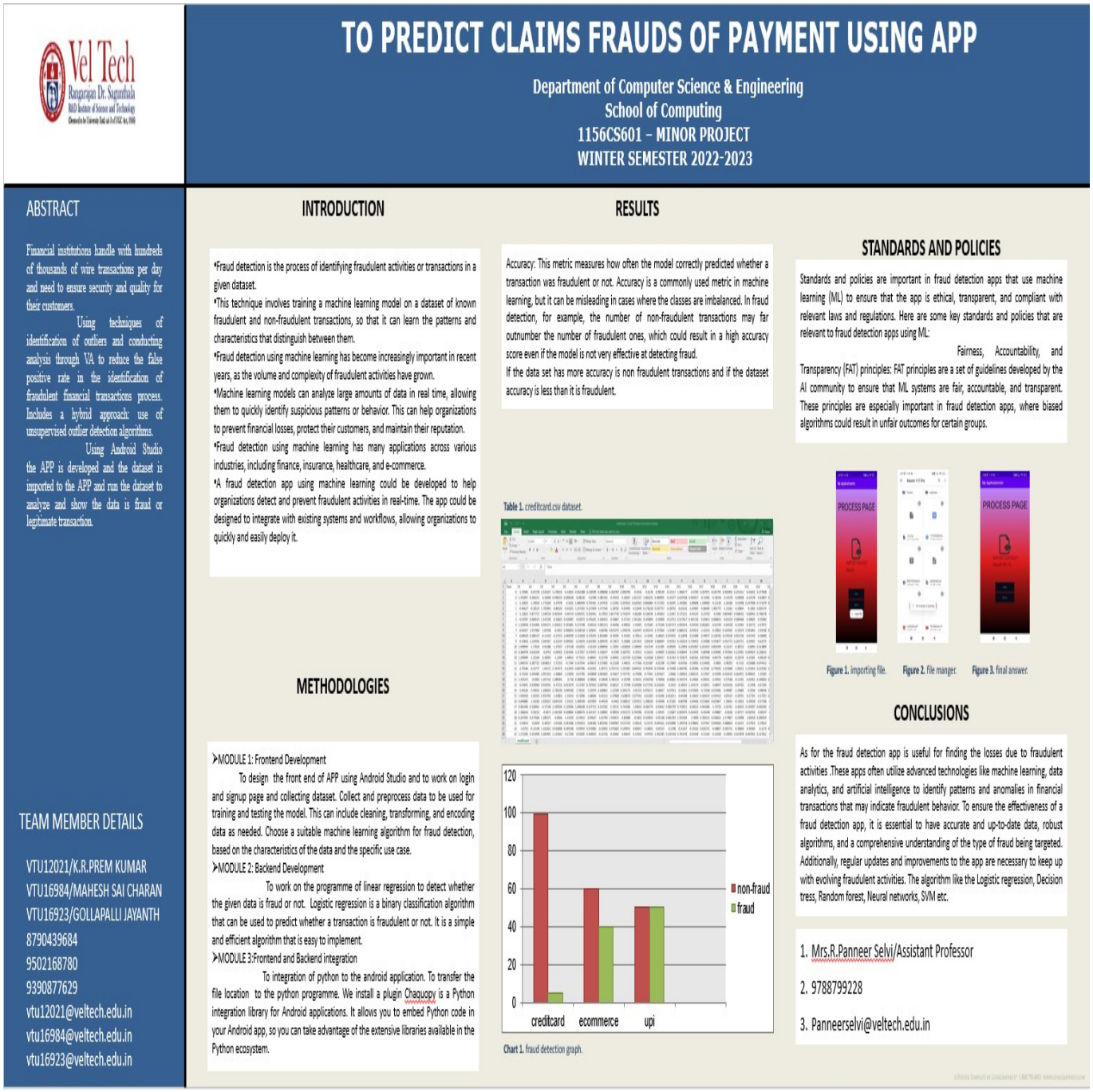



Figure 9.1: **Poster**

# References


[1] A. Gupta; V. Pant; S. Kumar; P. K. Bansal. Bank Loan Prediction System using Machine Learning, Proceedings of the 9th International Conference on System Modeling and Advancement in Research Trends, pp. 423 - 426.2020

[2] Biao Xua , Yao Wang , Xiuwu Liao, Kaidong Wang, "Efficient Fraud Detection using Deep Boosting Decision Trees ," School of Management, Center of Intelligent Decision Making and Machine Learning,Xi'an Jiaotong University, Xi'an, 710049, Shanxi, P.R.China, Volume 5, Issue 1 Feb, 2023.

[3] Dr. Ahmed Hassan Butt, M. Hazeel Ahmed Butt," Credit Card Fraud Detection in Banks using Machine Learning Algorithms ", sst, university of management and technology, university of management technology (umt) c-ii block c 2 phase 1 johar town, lahore, Punjab 54770, Pakistan, 10.14293/S2199-1006.1.SOR-.PPFI7P0.v2,Feb 2023.

[4] Douglas C. Montgomery; Elizabeth A. Peck; and G. Geoffrey Vining. Introduction to Linear Regression Analysis, international statistical Institution, Issue 2,vol 81,318 - 319.2013

[5] Dr. Jitendra Sheetlani, Dr. Harsh Pratap Singh,, Dr. Rajesh Kumar Vishwakarma, Dr. Abdul Razzak Khan Quershi , " A proposed framework of dimensionality reduction techniques to boost credit card fraud classification, IJFANS International journal of food and nutritional sciences,2020.

[6] Fahd Sabry Esmail, Fahad Kamal Alsheref, Amal Elsayed Aboutabl, "Review of Loan Fraud Detection Process in the Banking Sector Using Data Mining Techniques," Faculty of Commerce Business Administration, Department of Business Information Systems, Volume 14.2023

[7] Gokula Krishnan, M.V. Vijaya Saradhi, T.A.Mohana Prakash, K.Gokul Kannan, "Development of Deep Learning based Intelligent Approach for Credit Card Fraud Detection" International Journal on Recent and Innovation Trends in Computing and Communication10(12):133-139,2022.

[8] Jafar Nahri Aghdam Ghalejoogh, Nader Rezaei, Yaghoub Aghdam Mazrae, Rasoul Abdi, " Detecting financial fraud using machine learning techniques", aDepartment of Accounting, Bonab Branch, Islamic Azad University, Bonab, Iran, 2023.

[9] Rakhi Arora, Nitin Dixit and Gaurav Dubey, "Fraud detection of credit cards through machine learning algorithms," Institute of Technology and Management, Gwalior, vol. 08, Issue:08.2023

[10] Srinivasa Rao Dammavalam, and Md Mukheed, "Credit Card Fraud Detection Using Machine Learning," International Journal of Advances in Engineering and Management (IJAEM), Volume 5, Issue 1.2023

[11] Subhash P and K.R. Sumana, "Comparative Analysis of Credit Card Fraud Detection Using Machine Learning and Deep Learning Techniques," International Research Journal of Engineering and Technology (IRJET), vol. 08, Issue:08, Aug 2021

[12] T. Li; G. Kou; Y. Peng; S. Y. Philip; An integrated cluster detection. optimization, and interpretation approach for financial data, IEEE transactions on cybernetics, 52 (12),13848 – 13861.2021

[13] V. Gnaneswar, Sailusha, Ruttala, R. Ramesh, and G. Ramakoteswara Rao, 'Credit Card Fraud Detection Using Machine Learning', In 2020 4th International Conference on Intelligent Computing and Control Systems.IEEE, 2020.

[14] Y. Bao; G. Hilary; B. Ke. Artificial intelligence and fraud detection, Innovative Technology at the Interface of Finance and Operations, pp. 223 – 247.2022

[15] Yunyun Zhang,Yong Fang,Cheng Huang, 'Credit Card Fraud Detection Based on Machine Learning', Computers, Materials Continua CMC, vol.61, no.1, pp.185-195, 2021.