A Mini Project Report on

Fraud Detection and Analysis for Insurance Claim using Machine Learning

Submitted to JNTU HYDERABAD

In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

SHIVAM KUMAR	**(208R1A05N4)**
KYASA VYSHNAVI	**(208R1A05L1)**
PABBOJU MADHU PRASAD	**(218R5A0519)**
GAJULA NITHIN KRISHNA GOUD	**(208R1A05J8)**

Under the Esteemed guidance of

Dr. C.N. RAVI

Professor, Department of CSE

Department of Computer Science & Engineering

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

(2023-2024)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Accredited by NBA,Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad-501 401.

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **“Fraud Detection and Analysis for Insurance Claim using Machine Learning”** is a bonafide work carried out by

SHIVAM KUMAR	**(208R1A05N4)**
KYASA VYSHNAVI	**(208R1A05L1)**
PABBOJU MADHU PRASAD	**(218R5A0519)**
GAJULA NITHIN KRISHNA GOUD	**(208R1A05J8)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Mini Project Coordinator**	**Head of the Department**	**External Examiner**
Dr. C.N. Ravi Professor Department of CSE CMREC, Hyderabad.	**Mr. S. Kiran Kumar** Assistant Professor Department of CSE CMREC, Hyderabad.	**Dr. Sheo Kumar** Professor & HOD Department of CSE CMREC, Hyderabad.	

DECLARATION

This is to certify that the work reported in the present project entitled **"Fraud Detection and Analysis for Insurance Claim using Machine Learning"** is a record of bonafide work done by us in the Department of Computer Science and Engineering, CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

SHIVAM KUMAR (208R1A05N4)
KYASA VYSHNAVI (208R1A05L1)
PABBOJU MADHU PRASAD (218R5A0519)
GAJULA NITHIN KRISHNA GOUD (208R1A05J8)

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Sheo Kumar**, Professor and head of the department of Computer Science and Engineering, **CMR Engineering College** for their constant support.

We are extremely thankful to **Dr. C.N. Ravi**, Internal Guide, Professor of the Department of Computer Science and Engineering, for his constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank **Mr. S. Kiran Kumar**, Mini Project Coordinator, Assistant Professor of the department of Computer Science and Engineering, for his constant support in carrying out the project activities and reviews.

We express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, we are very much thankful to my parents who guided me for every step.

SHIVAM KUMAR (208R1A05N4)

KYASA VYSHNAVI (208R1A05L1)

PABBOJU MADHU PRASAD (218R5A0519)

GAJULA NITHIN KRISHNA GOUD (208R1A05J8)

TABLE OFCONTENTS

CONTENTS **PAGE NO.**

ABSTRACT

Insurance Company working as commercial enterprise from last few years have been experiencing fraud cases for all type of claims. Amount claimed by fraudulent is significantly huge that may causes serious problems, hence along with government, different organization also working to detect and reduce such activities. Such frauds occurred in all areas of insurance claim with high severity such as insurance claimed towards auto sector is fraud that widely claimed and prominent type, which can be done by fake accident claim. So, we aim to develop a project that work on insurance claim data set to detect fraud and fake claims amount. The project Implement machine learning algorithms to build model to label and classify claim. Also, to study comparative study of all machine learning algorithms used for classification using confusion matrix in term soft accuracy, precision, recall etc. For fraudulent transaction validation, machine learning model is built using PySpark Python Library.

LIST OF FIGURES

LIST OF OUTPUT SCREENS

1. INTRODUCTION

Insurance fraud is a claim made for getting improper money and not actual amount of money from insurance company or any other underwriter. Motor and insurance area unit two outstanding segments that have seen spurt in fraud. Frauds is classified from a supply or nature purpose of read. Sources is client, negotiator or internal with the latter two being a lot of essential from control framework purpose of reads. Frauds cowl vary of improper activities that a private might commit so as to attain the favorable outcome from an underwriter. Frauds is classified into nature wise, for example, application, inflation, identity, fabrication, contrived, evoked accidents etc. This could vary from staging incident, misrepresenting matters as well as pertinent members and therefore reason behind finally the extent of injury occurred. Probable things might embrace packing up for a state of affairs that wasn't lined beneath the insurance. Misrepresenting the context of an event. This might embrace transferring blames to the incidents wherever the insured set is accountable, failure to require approved the security measures. Increased impact of the incident .Inflated measure of the loss occurred through the addition of not much related losses or/and attributing inflated price to the increased losses.

1.1 PROBLEM STSTEMENT:

The traditional method for the detecting frauds depends on the event of heuristics around fraud indicators. Supported these, the selection on fraud created is said to occur in either of situations like, in certain things the principles are shown if the case should be interrogated for extra examination. In numerous cases, an inventory would be prepared with scores for various indicators of the occurred fraud. The factors for deciding measures and additionally the thresholds are tested statistically and periodically recalibrated. Associate aggregation and then price of the claim would verify necessity of case to be sent for extra examination. The challenge with above strategies is that they deliberately believe on manual mediation which might end in the next restrictions:

- Inability to perceive the context-specific relationships between the parameters (geography, client section, insurance sales process) which may not mirror the typical picture.
- Constrained to control with the restricted set of notable parameters supported the heuristic knowledge – whereas being aware that a number of the opposite attributes might conjointly influence the decisions.
- Reconstruction of the given model is that the hand operated exercise that need to be conducted sporadically to react dynamic behavior. Also to make sure that the model gives feedback from the examinations. The flexibility to manage this standardization is tougher.
- Incidence of occurrence of fraud is low - generally but 1percent of claims area unit classified.
- Consultations with business specialists point out that there is not a typical model to determine the model exactly similar to the context

1.2 MOTIVATION:

Ideally, businesses ought to obtain the responses to prevent fraud from happening or if that is out of the question, to watch it before important damage is finished at intervals the strategy. In most of the companies, fraud is understood entirely once it happens. Measures are then enforced to forestall it from happening over again. At intervals the given time that they can't resist at different time intervals, but Fraud detection is that the most effective suited issue for removing it from the atmosphere and preventing from continuance once more.

1.3 SIGNIFICANCES OF THE PROBLEM:

Knowing a risk is that the beginning in bar, associated intensive assessment offers the lightness that want. This is typically usually performed exploitation varied techniques, like interviews, surveys, focus teams, feedback conducted anonymously, detailed study of record and analysis to spot traffic pumpers, service users, and subscription scam which are different fraudulent case. The association of Certified Fraud Examiners offers a detailed guide to follow. This can be usually alleged to be a preventive methodology, fraud analysis and detection is associate certain consequence of associate intensive risk evaluation. Recognize and classify threats to fraud in knowledge technology and telecommunications sector stereotypically yield the shape of the chances like:

- Records showing associate degree inflated rates in calls at associate degree surreal time of day to associate degree uncertain location or far-famed fraud location.
- Unusual Dialing patterns showing one variety being referred to as additional of times by external numbers than job out.
- Increased calls created in an exceedingly day than the minute's allotted per day, that might indicate an account has been hacked or shared

1.4 MAJOR CONTRIBUTION:

- To compare machine learning algorithms: LR, XGB, DT, RF and SVM.
- To construct a model that predict transactions could be fraudulent with high accuracy.
- To detect if an insurance claim is fraudulent or not.
- To analyze the performance of fraud detection algorithm

2. LITERATURE SURVEY

Machine learning is usually abbreviated as metric capacity unit. The study of machine learning includes computers with the implicit capability to be trained whereas not being expressly programmed. This capacity unit focuses on the expansion of pc programs that has enough capability to alter, that square measure once unprotected to the new information. Metric capacity unit algorithms square measure generally classified into 3 main divisions that square measure supervised learning, unattended learning and reinforcement learning. Data processing a neighborhood of machine learning has advanced considerably within the current years. Data mining focuses at analyzing the whole data obtained. Furthermore data processing makes an attempt to seek out the realistic patterns in it. On the contrary, within the different of getting the knowledge for world understanding is within the processing applications like machine learning, it uses the knowledge to locate patterns in information and improvise the program actions thereby. Mainly within the supervised machine learning is that the objective of deducing which means from label on the information used for the coaching. The coaching information consists of a group of coaching samples. Just in case of supervised learning, every instance are often a base which incorporates Associate in Nursing input object that's considered the vector and also the output features a worth that acts as an indicator to run the model. A supervised learning rule initially accomplishes a groundwork task from the sample information then tries to construct a short lived perform, therefore it will plot new input vectors. The supervised learning algorithms square measure conspicuously employed in large choice of application areas. Associate in nursing best setting altogether the chance assist the rule to accurately mark the class labels for close instances and therefore a similar aspires supervised learning rule to chop back from the knowledge to the enclosed objects.

- Abhinav Srivastava, A. K. (2008). Credit Card Fraud Detection Using. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, 37 - 48.
- Aisha Abdallah, M. A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 90-113.
- Andrea Dal Pozzolo, G. B. (2018). Credit Card Fraud Detection: A Realistic Modeling. IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, vol 29, NO 8

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

General Performance of Existing System: Different forms of fraud lead to various crimes, however, many cases involve intentional injury to an insured object or the purpose of obtaining assets without payment. It is a well cognizant fact that fraud cases had been evident even from the start of the insurance industry. The discovery of insurance fraud is already a daunting task as not all applications can be severely investigated. The process of detecting fraud in the insurance industry is not only expensive but also time-consuming. The method that is working so far is the computer machine instance. However, the existing technology in the past was pre-programmed, which means that a consistent template was designed to detect fraudulent applications; and if a particular claim fits that figure it will only be identified as illegal, or else it will not be recognized. There are various AI methods with which frauds can be detected. Some techniques are as follows:

- To classify, combine cluster data, and segment, using data mining that can find rules in data and be able to highlight specific patterns, including those related to fraud.
- Professional programs to detect fraud in the form of laws.
- To automatize determining factors of false claims, ML techniques are employed.

➢ **DRAWBACKS OF EXISTING SYSTEM**

- The system is not implemented Convex-NMF based Supervised Spammer Detection with Social Interaction (CNMFSD).
- The system is not implemented any ml classifier for test and train the datasets.

3.2 PROPOSED SYSTEM:

Firstly, data received in batches from the client is validated to check whether data is in the same format as agreed with the client, if not, gets discarded and sent to the archived folder. Then, data transformation is done, in which data format is amended to make it suitable for insertion in the database. In the next stage, this data is exported in the form of CSV and before doing data clustering, data is preprocessed. In the training stage, different models suitable for each cluster are chosen and, hyper-parameter tuning or optimization is done to choose a set of parameters for optimized model learning. Then, the models are saved.

Once the training is completed, now, the model is fit for prediction. Now, data taken from the client for prediction is again checked, and appropriate changes are performed to make it suitable for insertion in the database. Subsequently, after pre-processing of data, it is sent for the process of clustering. Finally, to each cluster, a specific model is allocated. Then, prediction is performed accordingly. The output is exported to a CSV output file

After setting up the Heroku cloud platform, required files needed for pushing the model are added to the model, and the application is pushed on the cloud. Now, the application is ready to get launched. After the application starts, input is taken for training, and predicted output is received in the form of an excel file.

➢ ADVANTAGES OF PROPOSED SYSTEM

- Different models are tested on the dataset once it is obtained and cleaned.

- On the basis of the initial model performance, different features of the model are engineered and tested again.

- Once all the options area unit designed, the model is made and run victimization completely different completely different values and victimization different iteration procedures.

- A predictive model is created that predicts if an insurance claim is fraudulent or not.

- A predictive model is created that predicts if an insurance claim is fraudulent or not.

3.3 FUNCTIONAL REQUIREMENTS:

The Functional requirements for a system describe the functionality or the services that the system is expected to provide. These are the statements of services the system should provide and how the system should react to particular inputs and how the system should behave in particular situation.

User Registration: User Register with their Registration details.

User Login: User Login their account using password Live.

Load Model: Trained or Tested Model will be load.

Predict Output: Output will be predict based on parameters.

3.4 NON-FUNCTIONAL REQUIREMENTS:

The non-functional requirements describe the system constraints.

Performance: The application should have better accuracy and should provide prediction in less time.

Scalability: The system must have the potential to be enlarged to accommodate the growth.

Capability: The capability of the storage should be high so the large amount of data can be stored in order to train the model.

3.5 FEASIBILITY ANALYSIS:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations are involved in the feasibility analysis are:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.5.1 ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.5.2 TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.5.3 SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

- **Processor** - Pentium –IV
- **RAM** - 4 GB (min)
- **Hard Disk** - 20 GB
- **Key Board** - Standard Windows Keyboard
- **Mouse** - Two or Three Button Mouse
- **Monitor** - SVGA

4.2 SOFTWARE REQUIREMENTS:

- **Operating system** - Windows7 Ultimate.
- **Coding Language** - Python.
- **Front end** - Python
- **Back end** - Django ORM
- **Designing** - HTML, CSS, JavaScript
- **Database** - MySQL (vamp server)

5. SOFTWARE DESIGN

The following is the proposed method of the model development:

- Different models are tested on the dataset once it is obtained and cleaned.
- On the basis of the initial model performance, different features of the model are engineered and tested again.
- Once all the options area unit designed, the model is made and run victimization completely different completely different values and victimization different iteration procedures
- A predictive model is created that predicts if an insurance claim is fraudulent or not.
- Binary Classification task takes place which gives answer between YES or NO. This report deals with classification algorithm to detect fraudulent transaction.

The influence of the feature engineering, feature choice parameter modification area unit explored with an aim of achieving superior prophetic performance with superior accuracy. The assorted machine learning techniques area unit utilized in the development of accuracy of detection in unbalanced samples. As a system, the info are divided into 3 completely different segments. These area unit loosely coaching, testing and validation. The algorithmic program is trained on partial set of knowledge and parameters. These area unit later changed on a validation set. This may be studied for evaluation and performance on the particular testing dataset. The high acting models area unit formerly tested with numerous random splits of knowledge. This helps to confirm the consistency in results the approach discussed above comprises of three layers.

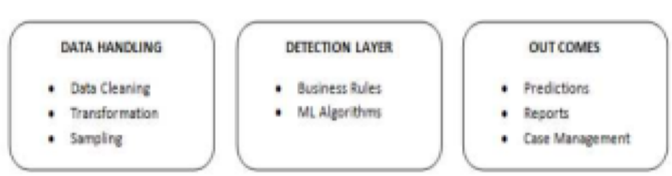


Fig. 1. Layers for Model Building

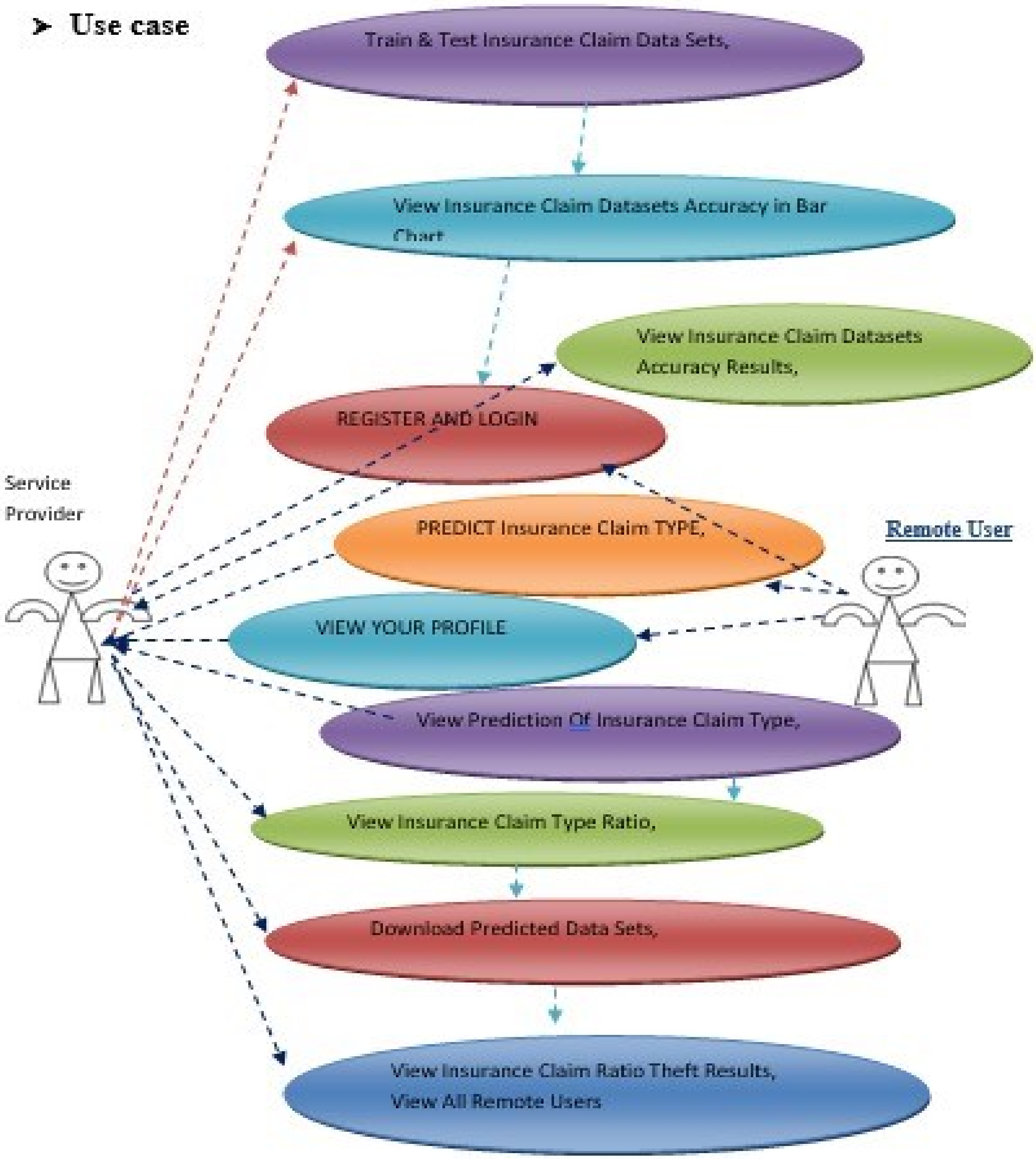
5.1 UNIFIED MODELING LANGUAGE DIAGRAMS:

UML is a method for describing the system architecture in detail using the blue print. UML represents a collection of best engineering practice that has proven successful in the modeling of large and complex systems. The UML is very important parts of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the helps UML helps project teams communicate explore potential designs and validate the architectural design of the software

5.1.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

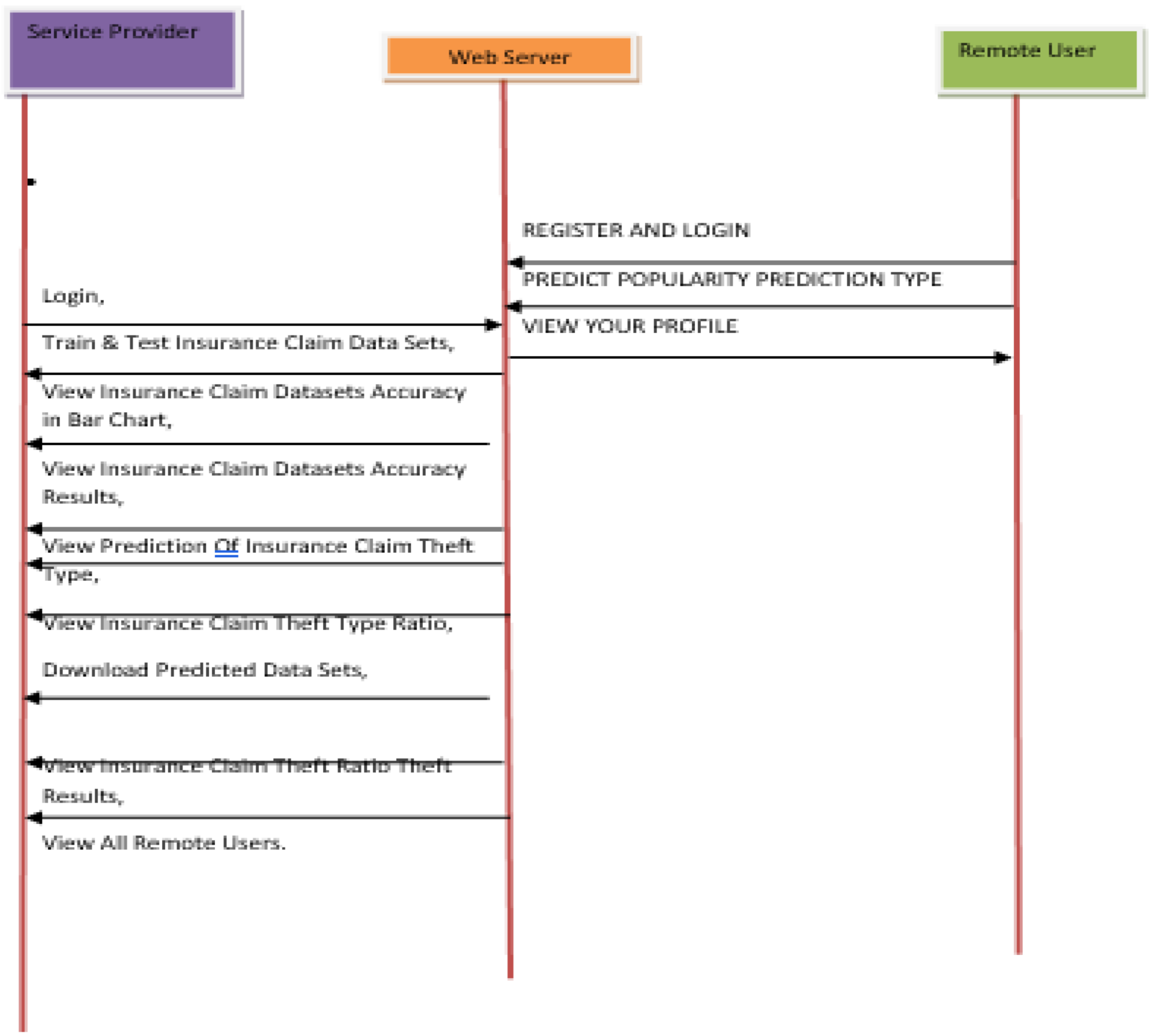
- **Use case**



Train & Test Insurance Claim Data Sets,
View Insurance Claim Datasets Accuracy in Bar Chart
View Insurance Claim Datasets Accuracy Results,
REGISTER AND LOGIN
Service Provider
PREDICT Insurance Claim TYPE,
Remote User
VIEW YOUR PROFILE
View Prediction Of Insurance Claim Type,
View Insurance Claim Type Ratio,
Download Predicted Data Sets,
View Insurance Claim Ratio Theft Results,
View All Remote Users

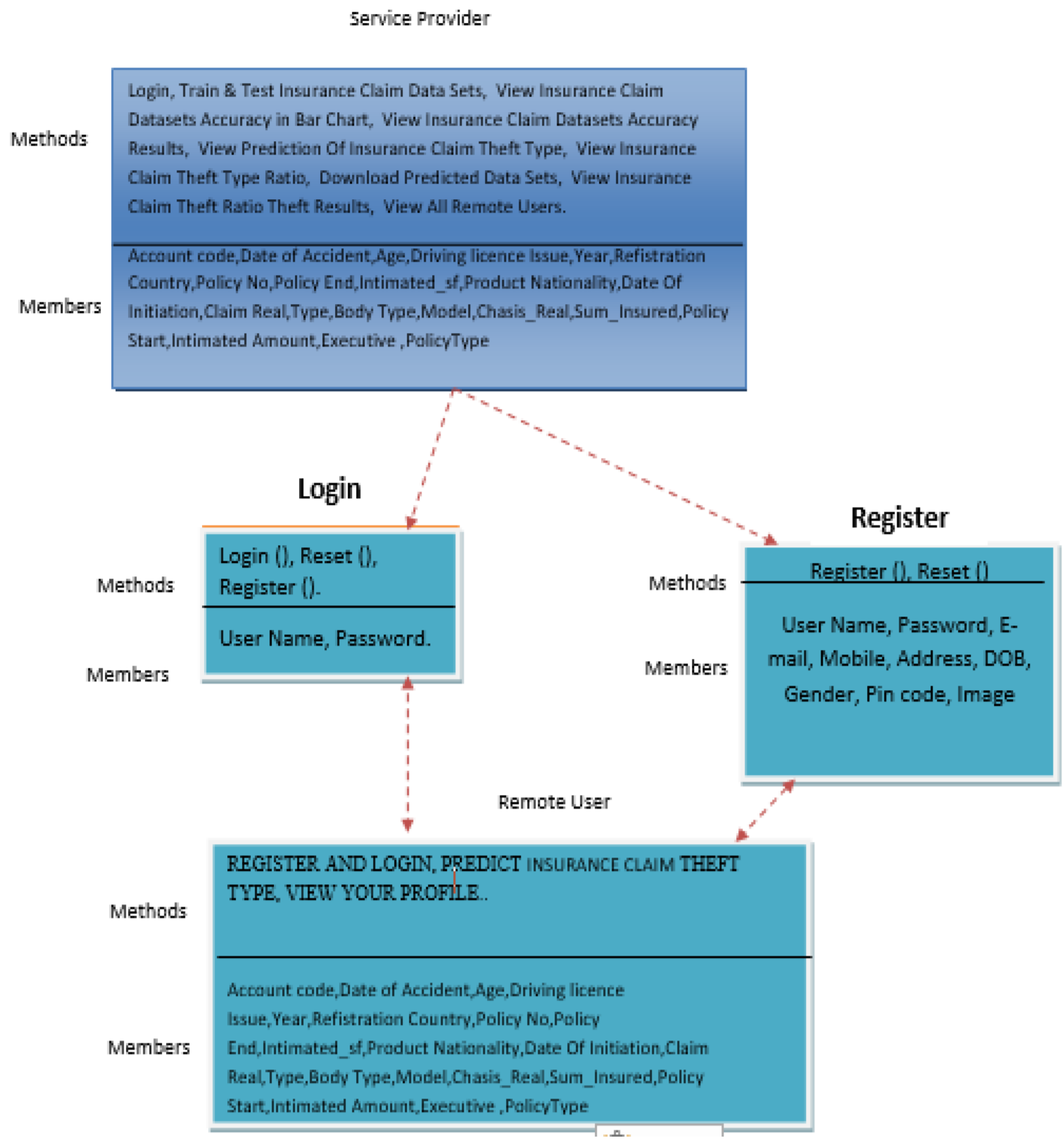
5.1.2 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagram

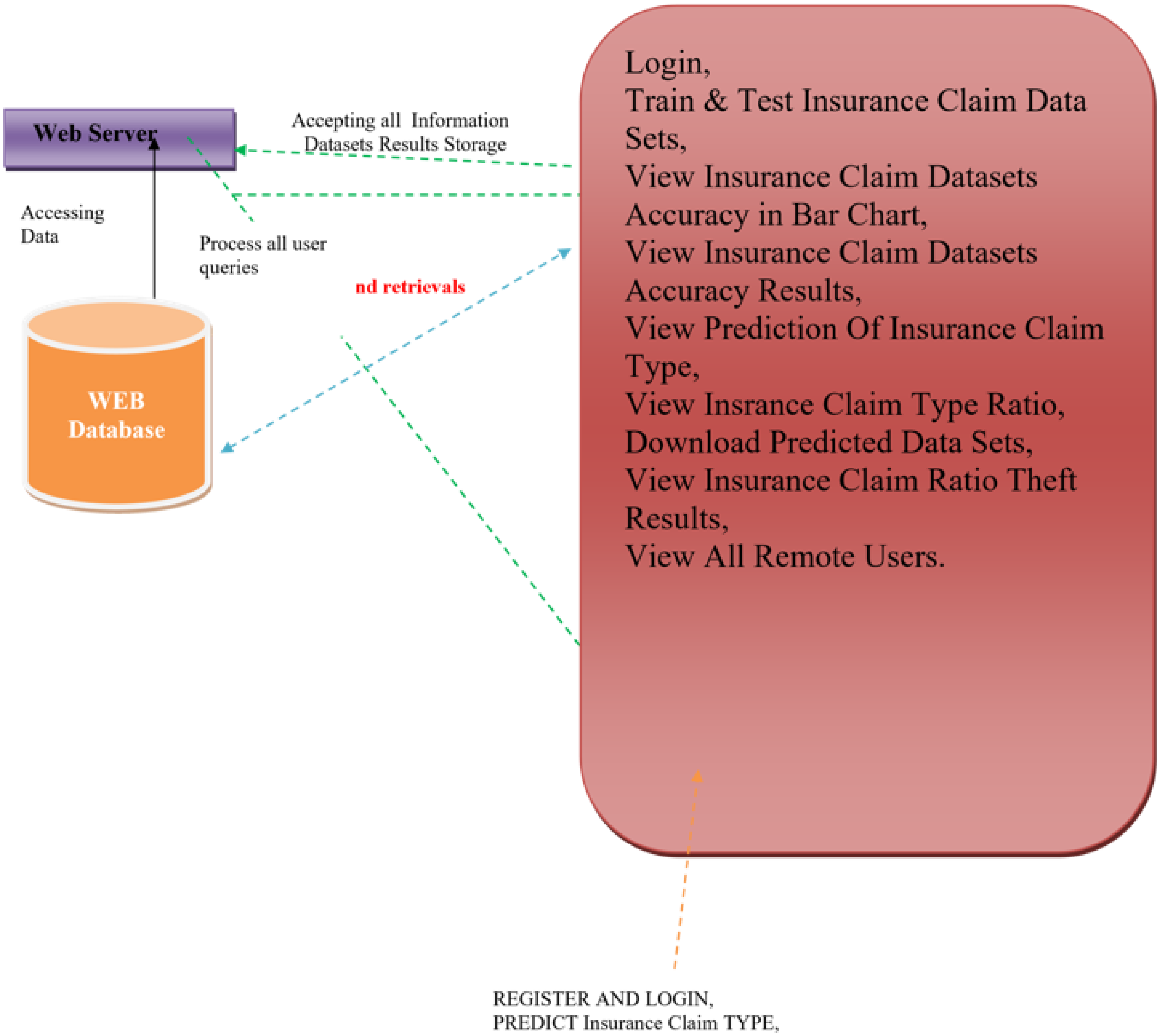


5.1.3 CLASS DIAGRAM:

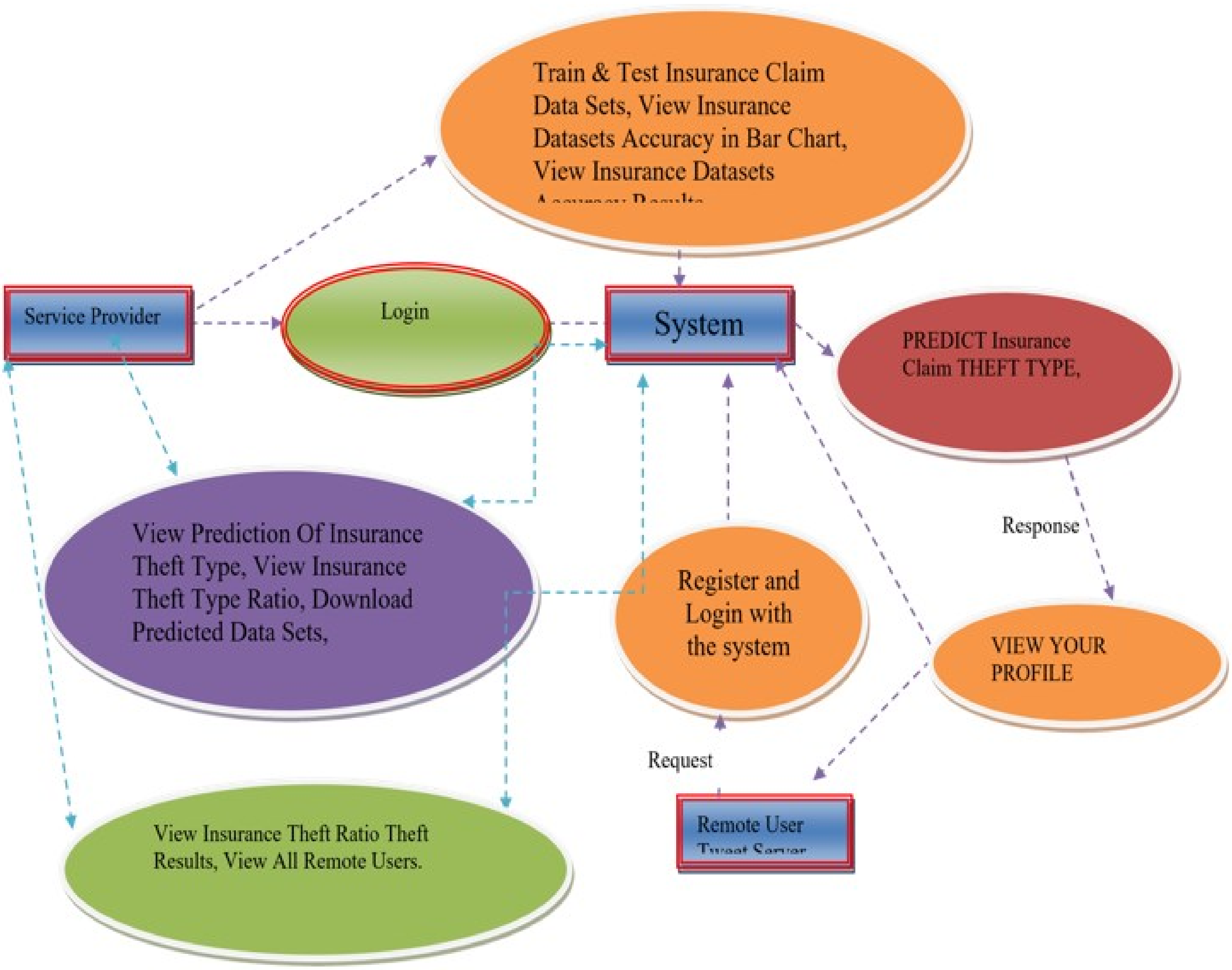
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



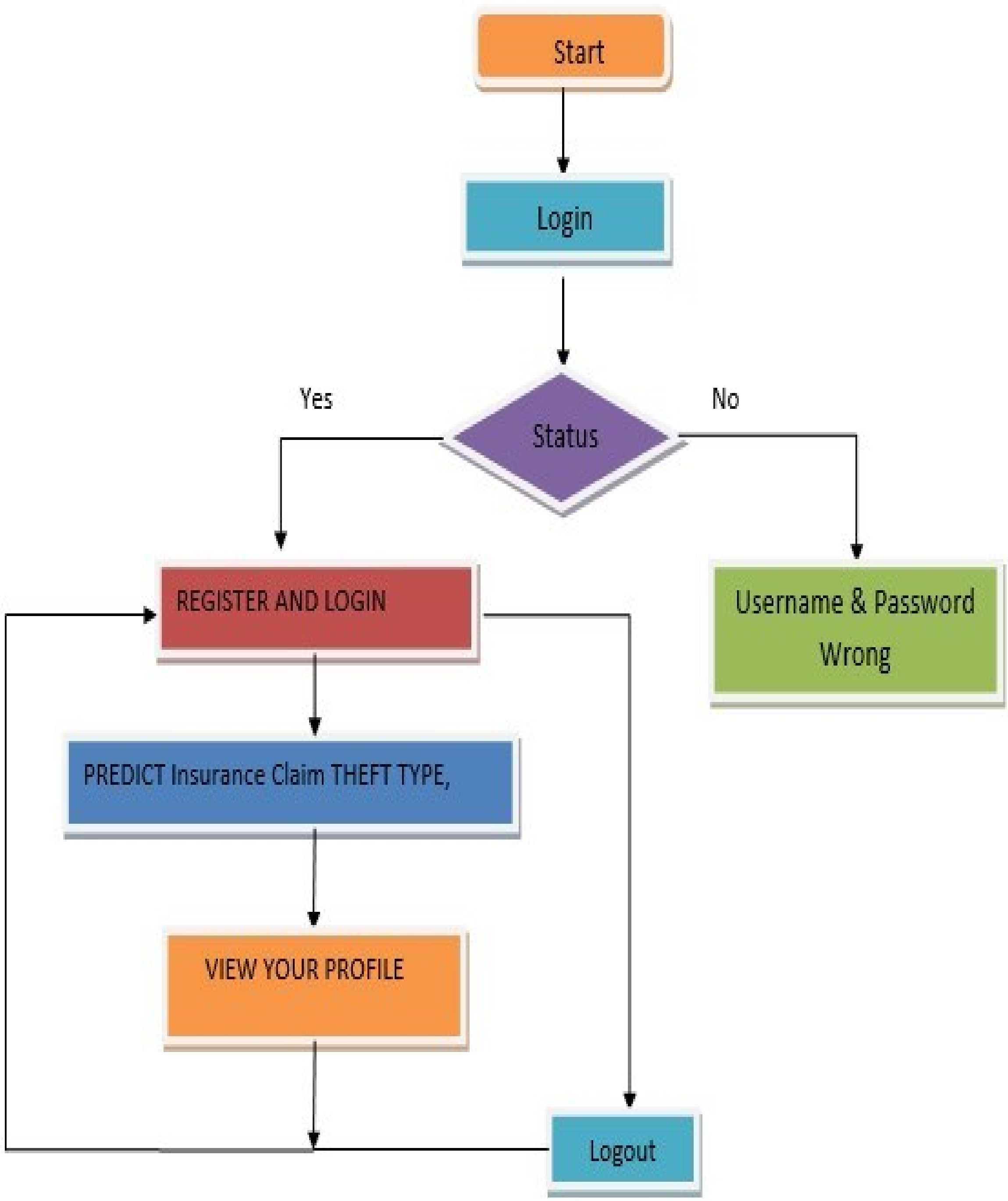
5.1.4 ARCHITECTURE DIAGRAM:



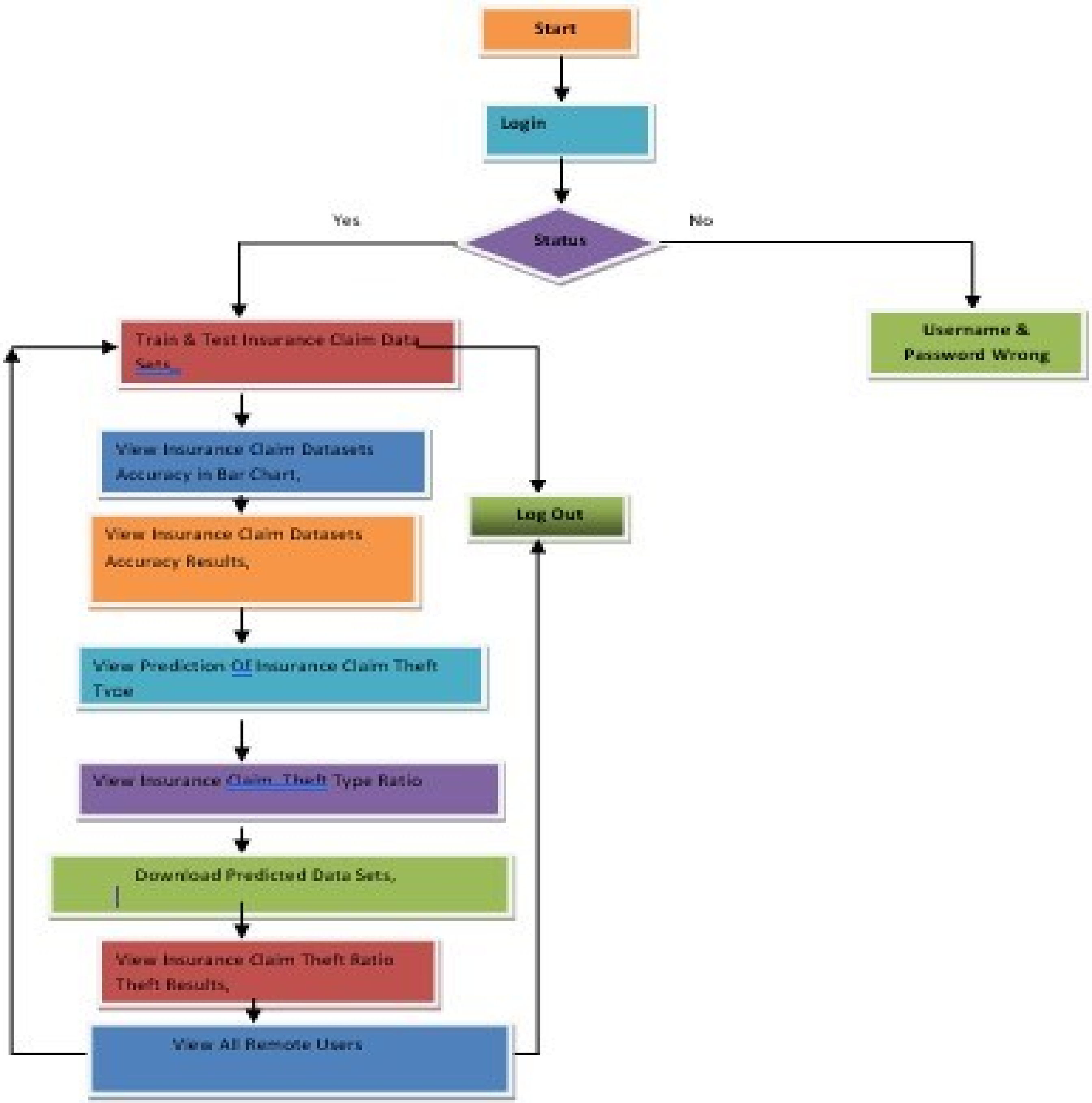
5.1.5 DATA FLOW DIAGRAM:



5.1.6 FLOW CHART DIAGRAM: REMOTE USER:



Flow Chart: Service Provider



6. SYSTEM IMPLEMENTATION

6.1 System Architecture:

Machine learning model is built with different algorithms that is trained by information and data set provided which predict new classification as “fraud” or “not” These algorithms implemented for building model that is trained using historical data and that predict unseen data with most matching features. And then model is tested and validated to evaluate its performance. After the calculations comparison is made.

For automobile insurance fraud detection supply regression shows the higher accuracy. Logistic regression evaluates the connection among Y “Label” and also the X “Features” by assessing possibilities employing a supply perform. The model predicts a likelihood that is employed to predict the label category. A supply perform or supply curve may be a common curve with equation:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki} + \varepsilon_i \qquad (1)$$

$$E\left({}^{Y}\!/_{X}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k \qquad (2)$$

Where,

Dependent Variable

Population Y intercept

Population Slope Coefficient

Independent Variable

Random Error term

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

To implement the Logistic Regression using Python, we set the following steps:

- **Data Pre-processing step**: In this step, the data is ready in order that are often employed in code with efficiency. Extraction of the dependent and freelance variables from the given dataset. Then the dataset is split as coaching and checking victimization train test split module from sklearn library. Feature scaling is completed therefore on get correct results of predictions.
- **Fitting Logistic Regression to the Training set**: Logistic Regression category of the sklearn library is employed. Classifier object is made and accustomed work the model to the supply regression Predicting the test result: The model is well trained on the training set, the result is predicted by using test set data.
- **Test accuracy of the result**: Confusion matrix is employed to judge the check accuracy. In this model of fraud detection, the prediction is completed therefore on check if deceitful dealings is claimed as deceitful and the other way around.
- **Visualizing the test set result**: Adjust the model fitting parameters, and repeat tests. Adjust the model fitting parameters, and repeat tests. Adjust the options or machine learning algorithmic program and repeat tests. The Methodology of this project is illustrated in below figure:

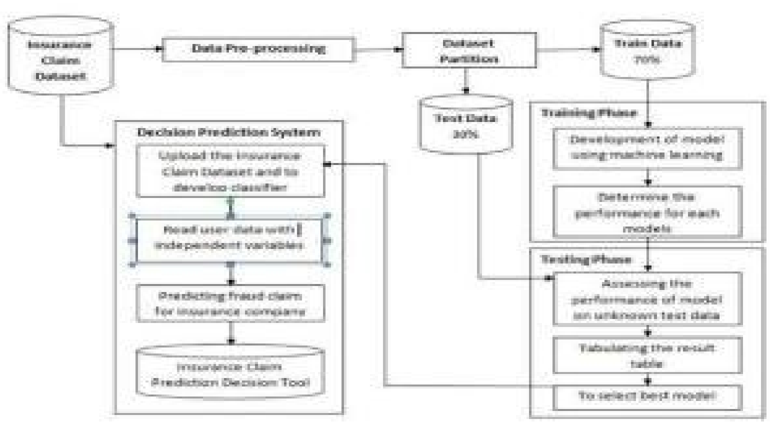


Fig. 2. Prediction Model

6.2 IMPLEMENTATION MODEL:

Pyspark: Fraud detection in car Insurance claims are determined employing a python module PySpark MLlib. It's a machine learning library. It's a wrapper over PySpark Core to try to knowledge analysis exploitation machine-learning algorithms. It works on distributed systems and is ascendable. The implementations of classification, clustering, rectilinear regression, and alternative machine-learning algorithms are found in PySpark ML lib. A repo is employed in PySpark and that they are ready for giant CSV file process in standalone mode. Particularly, the employment of the 'spark.ml' module was favored because the RDD-based MLLIB library goes to be deprecated.

Scikit-Learn: Scikit-learn is the most helpful library for machine learning in Python. The sklearn library contains loads of economical tools for machine learning and applied math modeling as well as classification, regression, clump and spatiality education.

6.3 MODULES:

User

Admin

6.3.1 USER:

User give the live inputs and get the output based on training model.

6.3.2 ADMIN:

Admin Represents analysis, visualization of the Model.

Stores User information.

6.3.3 MACHINE LEARNING:

Machine Learning is one of the booming technologies across the world that enables computers/machines to turn a huge amount of data into predictions. However, these predictions highly depend on the quality of the data, and if we are not using the right data for our model, then it will not generate the expected result. In machine learning projects, we generally divide the original dataset into training data and test data. We train our model over a subset of the original dataset, i.e., the training dataset, and then evaluate whether it can generalize well to the new or unseen dataset or test set. **Therefore, train and test datasets are the two key concepts of machine learning, where the training dataset is used to fit the model, and the test dataset is used to evaluate the model**.

- **Training Dataset**

The **training data is the biggest (in -size) subset of the original dataset, which is used to train or fit the machine learning model**. Firstly, the training data is fed to the ML algorithms, which lets them learn how to make predictions for the given task.

- **Test Dataset**

Once we train the model with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. **The test dataset is another subset of original data, which is independent of the training dataset**. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

- **RANDOM FOREST ALGORITHM**

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 over-fitting.

- **NAIVE BAYES CLASSIFIER ALGORITHM**

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

- **DECISION TREE CLASSIFICATION ALGORITHM**

➢ 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 the sub-trees.

- **SUPPORT VECTOR MACHINE ALGORITHM**

➢ Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

➢ The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyper-plane.

- SVM chooses the extreme points/vectors that help in creating the hyper-plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper-plane.

6.3.4 MODEL SELECTION IN MACHINE LEARNING:

Model selection in machine learning is the process of selecting the best algorithm and model architecture for a specific job or dataset. It entails assessing and contrasting various models to identify the one that best fits the data & produces the best results. Model complexity, data handling capabilities, and generalizability to new examples are all taken into account while choosing a model. Models are evaluated and contrasted using methods like cross-validation, and grid search, as well as indicators like accuracy and mean squared error. Finding a model that balances complexity and performance to produce reliable predictions and strong generalization abilities is the aim of model selection.

6.4 TECHNOLGIES:

6.4.1 PYTHON

6.4.2 FLASK

6.4.1 PYTHON:

Python is a highly interpreted programming language Python provides man GUI development possibilities (Graphical User Interface). flask is, the most frequently used technique of all GUI methods. It's a standard Python interface to the Python Tk GUI toolkit.

Python is the quickest and simplest method for creating GUI apps using Flask outputs. It is a simple job to create a GUI using flask. Python is a common, flexible and popular language of programming.

It is excellent as a first language since it is succinet and simple to understand and also good to use in any programmer's pile because it can be utilized from development of the web to software. It's basic, easy-to-use grammar, making it the ideal language to first learn computer programming.

Most implementations of Python (including C and Python), include a read- eval-print (REPL) loop that enables the user to act as a command-line interpreter that results in sequence and instantaneous intake of instructions. Other shells like as IDLE and Python provide extra features such as auto-completion, session retention and highlighting of syntax.

Interactive mode programming:

Invoking the interpreter without passing a script file as a parameter brings up the following prompt

− $ python

Python 2.4.3 (#1, Nov 11 2010, 13:34:43)

[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information

Type the following text at the Python prompt and press the Enter –

>>> print "Hello, Python!" If you are running new version of Python, then you would need to

use print statement with parenthesis as in print ("Hello, Python!");. However in Python version 2.4.3, this produces the following result

− Hello,

Script mode programming:

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file –

Live Demo print "Hello, Python!" We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

$ python test.py This produces the following result –

Hello, Python! Let us try another way to execute a Python script. Here is the modified test.py file –

Live Demo
#!/usr/bin/python print "Hello, Python!"

We assume that you have Python interpreter available in /usr/bin directory. Now, try to run this program as follows –
$ chmod +x test.py # This is to make file executable
$./test.py
This produces the following result –

Hello, Python!

6.4.2 FLASK WEB FRAMEWORK:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.Unlike the Django framework, Flask is very Pythonic. It’s easy to get started with Flask, because it doesn’t have a huge learning curve.

On top of that it’s very explicit, which increases readability.

It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre- existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools

7. CODING

```
from django.shortcuts import render
from django.contrib import messages
from user.models import Usermodel
import os
import joblib
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(_file_)))
# Create your views here.
def Userlogin(request):
return render(request, "user/Userlogin.html")
def userregister(request):
return render(request, "user/userregister.html")
def userregisterAction(request):
if request.method == 'POST':
name = request.POST.get('uname')
email = request.POST.get('uemail')
password = request.POST.get('upasswd')
phoneno = request.POST.get('uphonenumber')
form1 = Usermodel(name=name, email=email, password=password, phoneno=phoneno,
status='waiting')
form1.save()
messages.success(request, 'Registration Successful')
return render(request, "user/Userlogin.html")
else:
messages.error(request, 'Registration Unsuccessful')
return render(request, "user/userregister.html")
def userloginaction(request):
if request.method == 'POST':
sname = request.POST.get('email')
spasswd = request.POST.get('upasswd')
try:

check = Usermodel.objects.get(email=sname, password=spasswd)
status = check.status
if status == 'activated':
messages.success(request, 'Login Successful')
return render(request, "user/userhome.html")
else:
```

```
messages.error(request, 'Login Unsuccessful')
return render(request, "user/Userlogin.html")
except:
messages.error(request, 'Login Unsuccessful')
return render(request, "user/Userlogin.html")
else:

messages.error(request, 'Login Unsuccessful')
return render(request, "user/userhome.html")
def predict(request):
return render(request, "user/userhome.html")
def predicts(request):
if request.method == 'POST':
l1 = request.POST.get('input1')
l2 = request.POST.get('input2')
l3 = request.POST.get('input3')
l4 = request.POST.get('input4')
l5 = request.POST.get('input5')
l6 = request.POST.get('input6')
l7 = request.POST.get('input7')
l8 = request.POST.get('input8')
l9 = request.POST.get('input9')
l10 = request.POST.get('input10')
l11 = request.POST.get('input11')
loaded_model = joblib.load(os.path.join(BASE_DIR, 'media/rf_model.unknown'))
result = loaded_model.predict([[l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11]])
print(result)
if result == 0:
print('not fraud')
else:

print('fraud')
d = {'pred':result}
return render(request, "user/results.html", d)
else:
return render(request, "user/userhome.html")
def usrlogout(request):
return render(request, "user/userlogin.html")
```

- ***Admin***

#adminbase.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
<title>CyberSercurity</title>
<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
<link rel="stylesheet" href="{% static 'css/styles.min.css' %}">
</head>
<body>
<style>
body {
background-image: url("{% static 'img/bg-1.jpg' %}");
background-size: cover;
background-repeat: no-repeat;
}
</style>
<nav class="navbar navbar-dark navbar-expand-md bg-dark py-3">
<div class="container"><a class="navbar-brand d-flex align-items-center"
href="#"><span>Brand</span></a><button data-bs-toggle="collapse" class="navbar-
toggler" data-bs-target="#navcol-6"><span class="visually-hidden">Toggle
navigation</span><span class="navbar-toggler-icon"></span></button>
<div class="collapse navbar-collapse flex-grow-0 order-md-first" id="navcol-6">
<ul class="navbar-nav me-auto">
<li class="nav-item"><a class="nav-link active" href="{% url 'showusers' %}">Show
users</a></li>
<li class="nav-item"><a class="nav-link active" href="{% url 'ML' %}">Machine
Learning</a></li>
<li class="nav-item"><a class="nav-link active" href="{% url 'logout' %}">Logout</a></li>
</ul>
</div></div>
</nav>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
{%block contents%}
{%endblock%}
</body>
</html>
```

#***adminhome.html***

```
{% extends 'admin/adminbase.html' %}
{% load static %}
{% block contents %}
<section class="">
<div class="container py-4">
<div class="row d-flex justify-content-center py-5">
<div class="col-md-12">
<table border="2px" class="table table-hover table-dark">
<thead>
<tr class="table-danger">
<th scope="col">S.No</th>
<th scope="col">Name</th>
<th scope="col">Mobile</th>
<th scope="col">Email</th>
<th scope="col">Status</th>
<th scope="col">Activate</th>
</tr>
</thead>
<tbody>
{% for i in data %}
<tr scope="row" style="color: BLUE; background-color:#FFFCBB">
<td>{{forloop.counter}}</td>
<td>{{i.name}}</td>
<td>{{i.phoneno}}</td>
<td>{{i.email}}</td>
<td>{{i.status}}</td>
{% if i.status == 'waiting' %}
<td><a class="btn-link" href="/AdminActiveUsers/?uid={{ i.id }}"
style="color:rgb(235, 235, 239)">Activate</a></td>
{% else %}
<td> Activated</td>
{% endif %}
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
</section>
{% endblock %}
```

#adminlogin.html

```
{% extends 'base.html' %}
{% load static %}
{% block contents %}
<section>
<div class="lgp-hd"></div>
<div class="container login-cont">
<div class="row">
<div class="col-10 col-sm-6 col-md-4 col-lg-4 offset-1 offset-sm-3 offset-md-4 offset-lg-4
login-col"><i class="icon ion-lock-combination"></i>
<br><br><br><br><br><br><br><br>
<form action="/adminloginaction/" class="login-form" method="post">
{% csrf_token %}
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="text" name="uname" required=""
placeholder="Enter Admin ID">
</div>
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="password" name="upasswd"
required="" placeholder="Enter Password">
</div>
<div class="form-group mb-3">
<button class="btn btn-light btn-lg login-btn" type="submit"
value="submit"><strong>Login</strong></button>
</div>
</form>
{% if messages %}
{% for message in messages %}
<font color='white'> {{ message }}</font>
{% endfor %}
{% endif %}
</div>
</div>
</div>
</section>
{% endblock %}
```

#adminml.html

```
{% extends 'admin/adminbase.html' %}
{% load static %}
{% block contents %}
<style>
table {
margin-left: auto;
margin-right: auto;
border-collapse: collapse;
width: 50%;
}
th, td, h2 {
border: 1px solid black;
padding: 8px;
text-align: center;
}
</style>
<h2>Confusion Matrix</h2>
<h2>Accuracy: {{accuracy}}</h2>
<table>
<tr>
<th>s.no</th>
<th>0</th>
<th>1</th>
</tr>
<tr>
<th>0</th>
<td>{{ confusion_matrix.0.0 }}</td>
<td>{{ confusion_matrix.0.1 }}</td>
</tr>
<tr>
<th>1</th>
<td>{{ confusion_matrix.1.0 }}</td>
<td>{{ confusion_matrix.1.1 }}</td>
</tr>

</table>
{% endblock %}
```

❖ ***User***

#results.html

```
{% extends 'user/userbase.html' %}
{% load static %}
{% block contents %}
<style>
center{
text-align: center;
}
</style>
{% if predictions == '0' %}
<div class="center">
<h1><strong>Safe:</strong>He/She is not Fraud</h1>
</div>
{% else %}
<div class="center">
<h1><strong>Warning:</strong>He/She is an Fraud</h1>
</div>
{% endif %}
{% endblock %}
```

#userbase.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
<title>CyberSercurity</title>
<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
<link rel="stylesheet" href="{% static 'css/styles.min.css' %}">
</head>
<body>
<style>
body {
background-image: url("{% static 'img/bg-1.jpg' %}");
background-size: cover;
background-repeat: no-repeat;
}
</style>
<nav class="navbar navbar-dark navbar-expand-md bg-dark py-3">
```

```
<div class="container"><a class="navbar-brand d-flex align-items-center"
href="#"><span>Brand</span></a><button data-bs-toggle="collapse" class="navbar-
toggler" data-bs-target="#navcol-6"><span class="visually-hidden">Toggle
navigation</span><span class="navbar-toggler-icon"></span></button>
<div class="collapse navbar-collapse flex-grow-0 order-md-first" id="navcol-6">
<ul class="navbar-nav me-auto">
<li class="nav-item"><a class="nav-link active" href="{% url 'predict'
%}">Predict</a></li>
<li class="nav-item"><a class="nav-link active" href="{% url 'usrlogout'
%}">Logout</a></li>
</ul>
</div>
</div>
</nav>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
{%block contents%}
{%endblock%}
</body>
</html>
```

#*userhome.html*

```
{% extends 'user/userbase.html' %}
{% load static %}
{% block contents %}
<style>
.form-container {
max-width: 300px;
margin: 0 auto;
}
.form-container label {
color: rgb(207, 20, 20);
}
.form-container input[type="text"],
.form-container input[type="email"],
.form-container input[type="password"] {
display: inline-block;
width: 200px; /* Adjust the width as desired */
padding: 8px;
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 4px;
}
```

```
.form-container input[type="submit"] {
background-color: #4CAF50;
color: white;
padding: 10px 15px;
border: none;
border-radius: 4px;
cursor: pointer;
}
.form-container input[type="submit"]:hover {
background-color: #45a049;
}
.grid-container {
display: grid;
grid-template-columns: repeat(4, 1fr);
grid-gap: 10px;
}
.grid-item {
display: flex;
flex-direction: column;
}
</style>
<br><br>
<div class="form-container">
<form  action="/predicts/" method="POST">
{% csrf_token %}
<div class="grid-container">
<div class="grid-item">
<label for="input1">Age</label>
<input type="text" id="input1" name="input1">
</div>
<div class="grid-item">
<label for="input2">RepNumber</label>
<input type="text" id="input2" name="input2">
</div>
<div class="grid-item">
<label for="input3">WeekOfMonth</label>
<input type="text" id="input3" name="input3">
</div>
<div class="grid-item">
<label for="input4">WeekOfMonthClaimed</label>
<input type="text" id="input4" name="input4">
</div>
```

```
<div class="grid-item">
<label for="input5">DriverRating</label>
<input type="text" id="input5" name="input5">
</div>
<div class="grid-item">
<label for="input1">Year</label>
<input type="text" id="input1" name="input6">
</div>
<div class="grid-item">
<label for="input2">Fault</label>
<input type="text" id="input2" name="input7">
</div>
<div class="grid-item">
<label for="input2">PastNumberOfClaims_1</label>
<input type="text" id="input2" name="input8">
</div>
<div class="grid-item">
<label for="input2">PastNumberOfClaims_2 to 4</label>
<input type="text" id="input2" name="input9">
</div>
<div class="grid-item">
<label for="input2">PastNumberOfClaims_>4</label>
<input type="text" id="input2" name="input10">
</div>
<div class="grid-item">
<label for="input2">PastNumberOfClaims_none</label>
<input type="text" id="input2" name="input11">
</div>
</div>
<input type="submit" value="Submit">
</form>
</div>
{% endblock %}
```

#userlogin.html

```
{% extends 'base.html' %}
{% load static %}
{% block contents %}
<section>
<div class="lgp-hd"></div>
```

```html
<div class="container login-cont">
<div class="row">
<div class="col-10 col-sm-6 col-md-4 col-lg-4 offset-1 offset-sm-3 offset-md-4 offset-lg-4
login-col"><i class="icon ion-lock-combination"></i>
<br><br><br><br><br><br><br><br>
<form action="/userloginaction/" class="login-form" method="post">
{% csrf_token %}
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="text" name="email" required=""
placeholder="Enter User Email">
</div>
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="password" name="upasswd"
required="" placeholder="Enter Password">
</div>
<div class="form-group mb-3">
<button class="btn btn-light btn-lg login-btn" type="submit"
value="submit"><strong>Login</strong></button></div>
{% if messages %}
{% for message in messages %}
<font color='White'> {{ message }}</font>
{% endfor %}
{% endif %}
</form>
<form action="/userregister/" method="post">
{% csrf_token %}
<button class="btn btn-light btn-lg login-btn" type="submit"
value="submit"><strong>Register</strong></button>
</form>
</div>
</div>
</div>
</section>
{% endblock %}
```

#userregister.html

```html
{% extends 'base.html' %}
{% load static %}
{% block contents %}
<section>
<div class="lgp-hd"></div>
```

```
<div class="container login-cont">
<div class="row">
<div class="col-10 col-sm-6 col-md-4 col-lg-4 offset-1 offset-sm-3 offset-md-4 offset-lg-4
login-col"><i class="icon ion-lock-combination"></i>
<br><br><br><br><br><br>

<form action="/userregisterAction/" class="login-form" method="POST">
{% csrf_token %}
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="text" name="uname" required=""
placeholder="Enter Username">
</div>

<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="email" name="uemail"
required="" placeholder="Enter Email">
</div>
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="password" name="upasswd"
required="" placeholder="Enter Password">
</div>
<div class="form-group mb-3">
<input class="form-control form-control-lg lg-frc" type="phonenumber"
name="uphonenumber" required="" placeholder="Enter Phonenumber">
</div>
<div class="form-group mb-3">
<button class="btn btn-light btn-lg login-btn" type="submit"
value="submit"><strong>Register</strong></button>
</div>
</form>
{% if messages %}
{% for message in messages %}
<font color='white'> {{ message }}</font>
{% endfor %}
{% endif %}
</div>
</div>
</div>
</section>

{% endblock %}
```

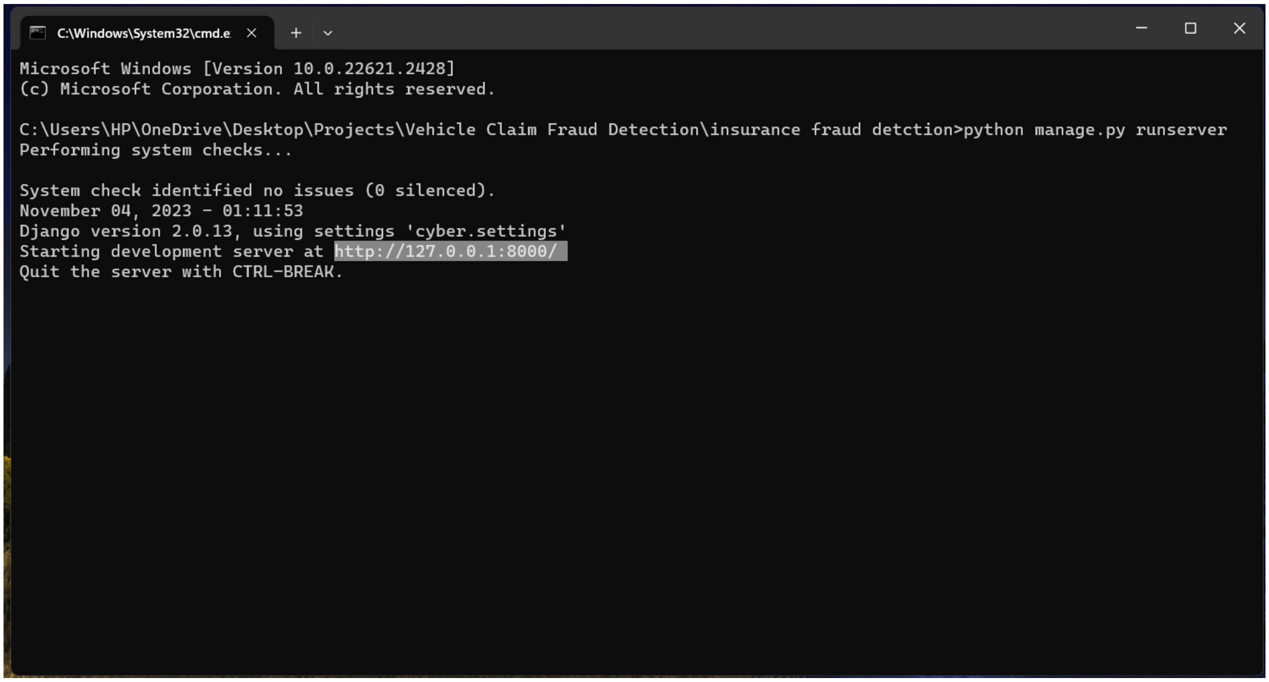
#base.html

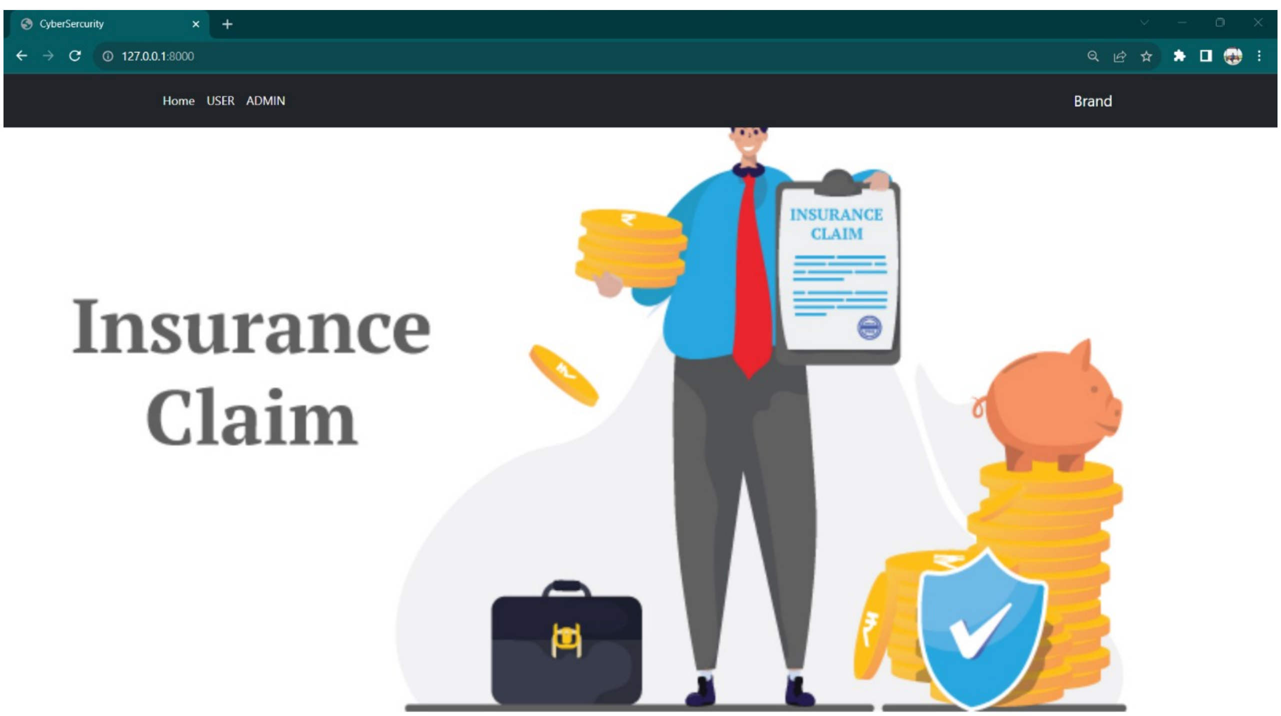
```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-
fit=no">
<title>CyberSercurity</title>
<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
<link rel="stylesheet" href="{% static 'css/styles.min.css' %}">
</head>
<body>
<style>
body {
background-image: url("{% static 'img/bg-1.jpg' %}");
background-size: cover;
background-repeat: no-repeat;
}
</style>
<nav class="navbar navbar-dark navbar-expand-md bg-dark py-3">
<div class="container"><a class="navbar-brand d-flex align-items-center"
href="#"><span>Brand</span></a><button data-bs-toggle="collapse" class="navbar-
toggler" data-bs-target="#navcol-6"><span class="visually-hidden">Toggle
navigation</span><span class="navbar-toggler-icon"></span></button>
<div class="collapse navbar-collapse flex-grow-0 order-md-first" id="navcol-6">
<ul class="navbar-nav me-auto">
<li class="nav-item"><a class="nav-link active" href="{% url 'Home'
%}">Home</a></li>
<li class="nav-item"><a class="nav-link active" href="{% url 'Userlogin'
%}">USER</a></li>
<li class="nav-item"><a class="nav-link active" href="{% url 'adminlogin'
%}">ADMIN</a></li>
</ul>
</div>
</div>
</nav>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
{%block contents%}
{%endblock%}
</body>
</html>
```

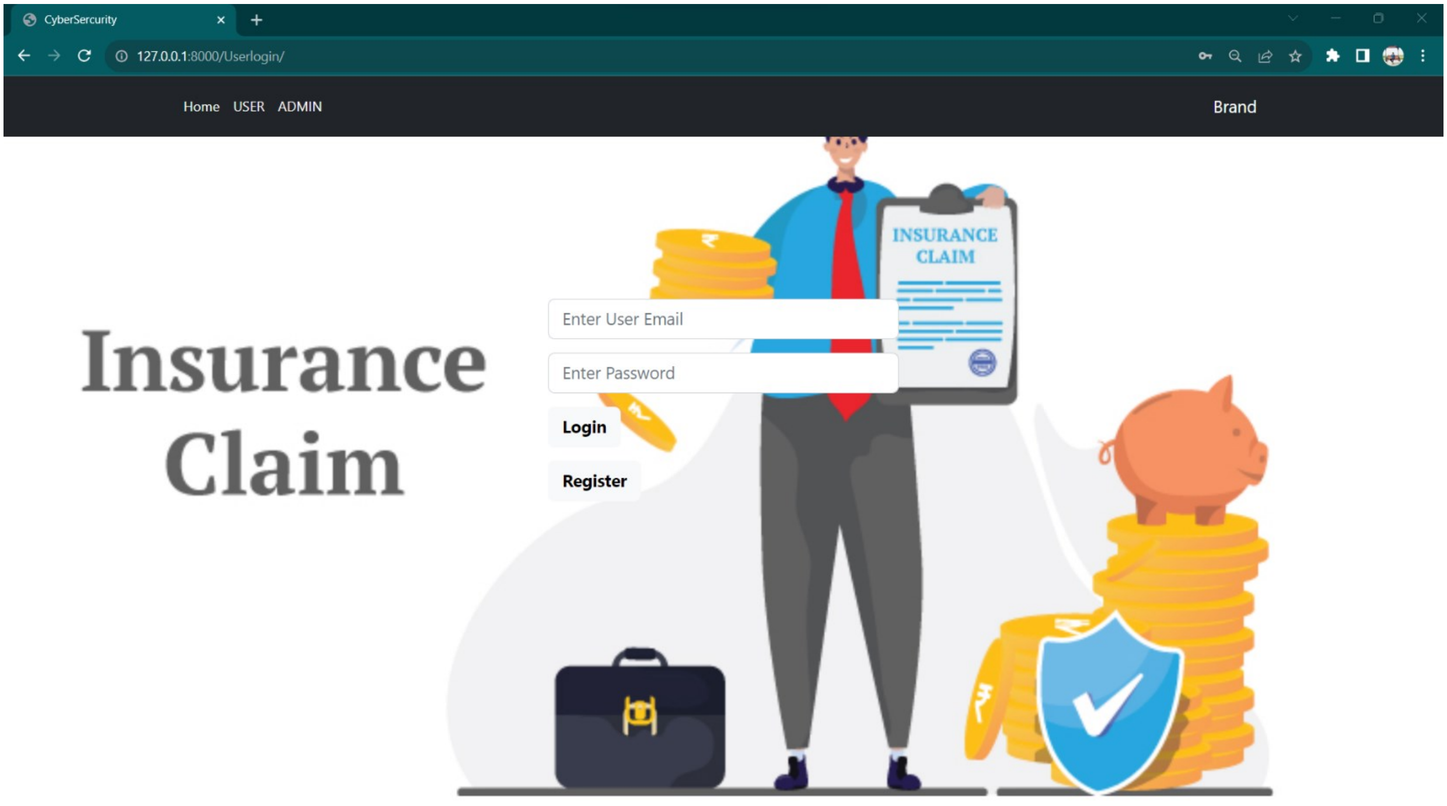
#index.html

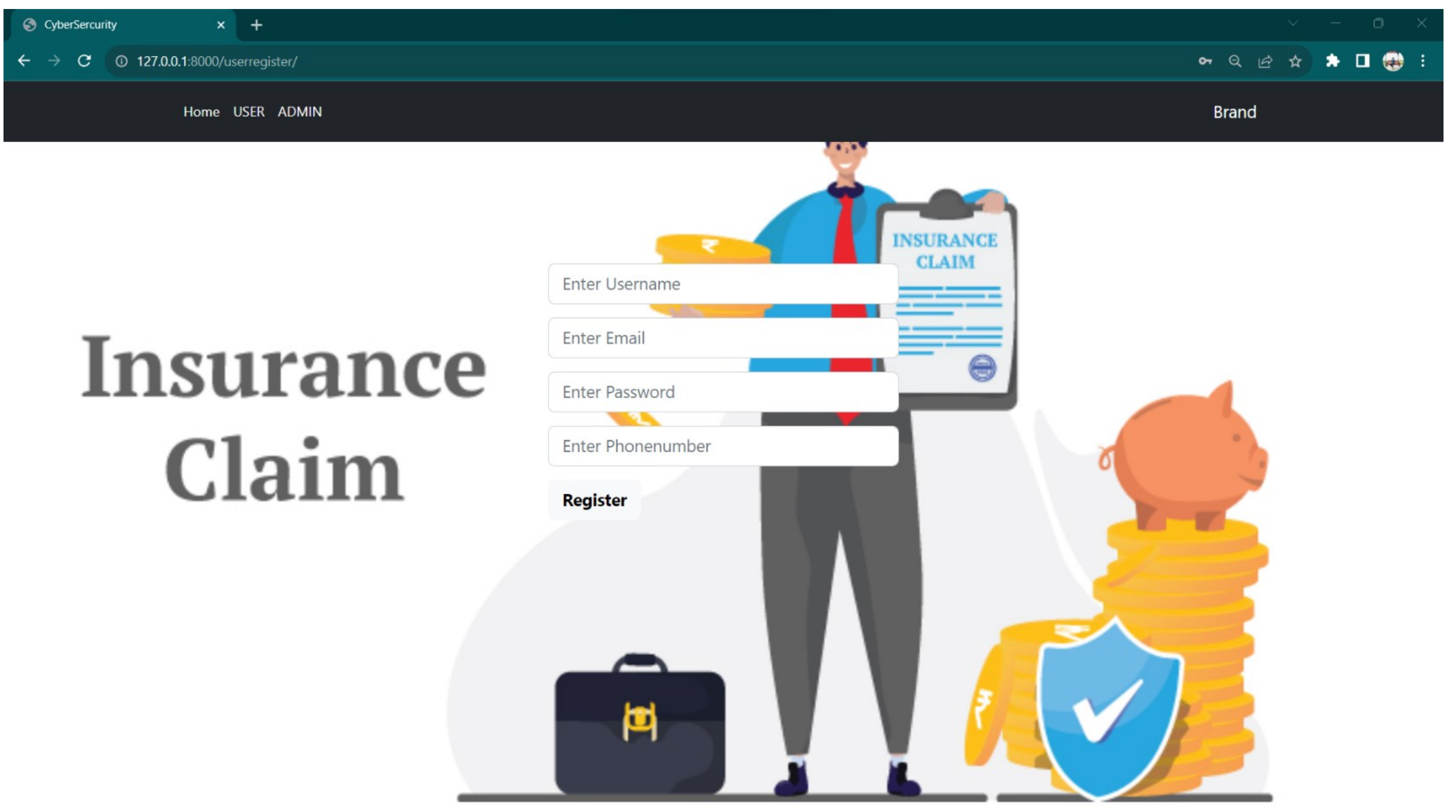
```
{% extends 'base.html' %}
{% load static %}
{% block contents %}
{% endblock %}
```

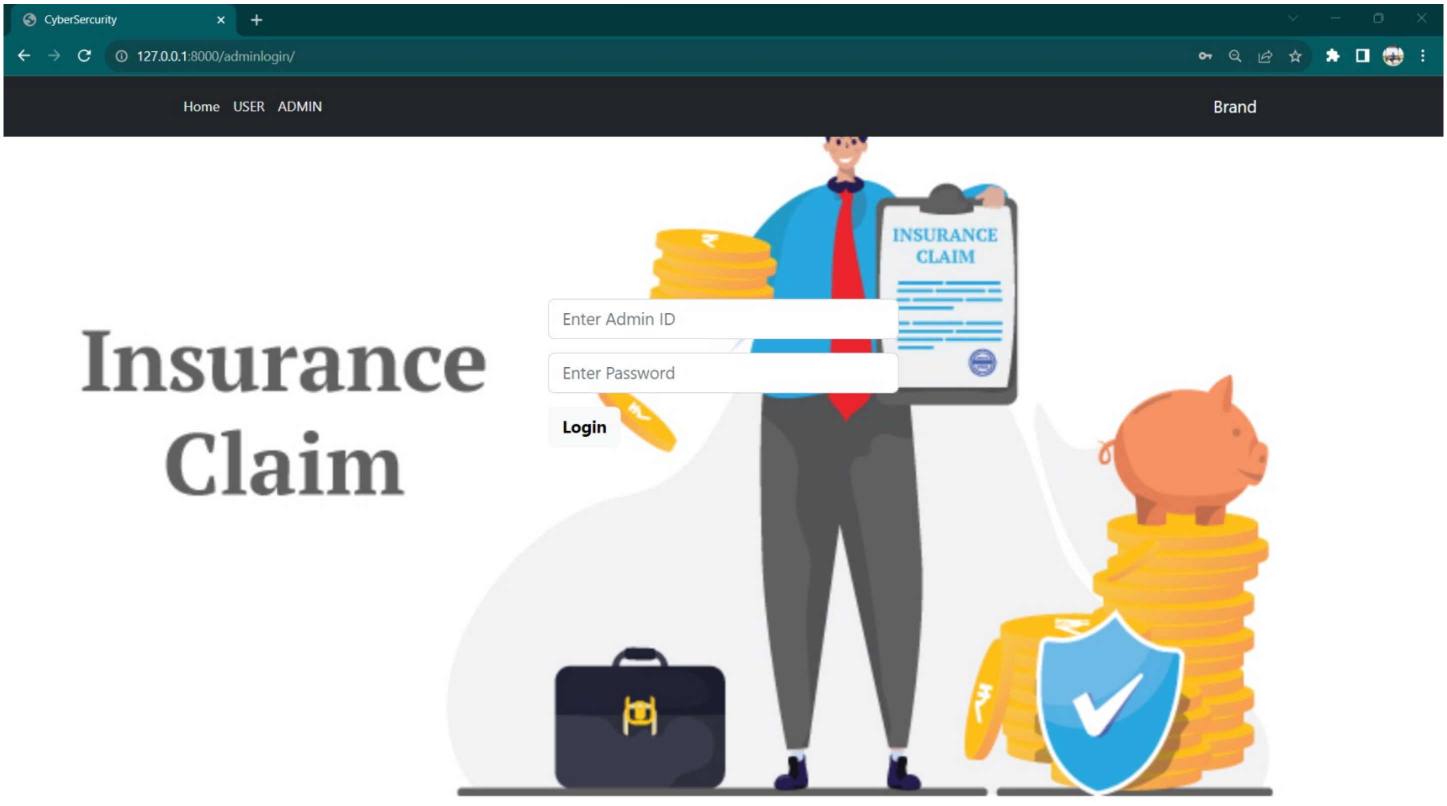
8. OUTPUT SCREENS

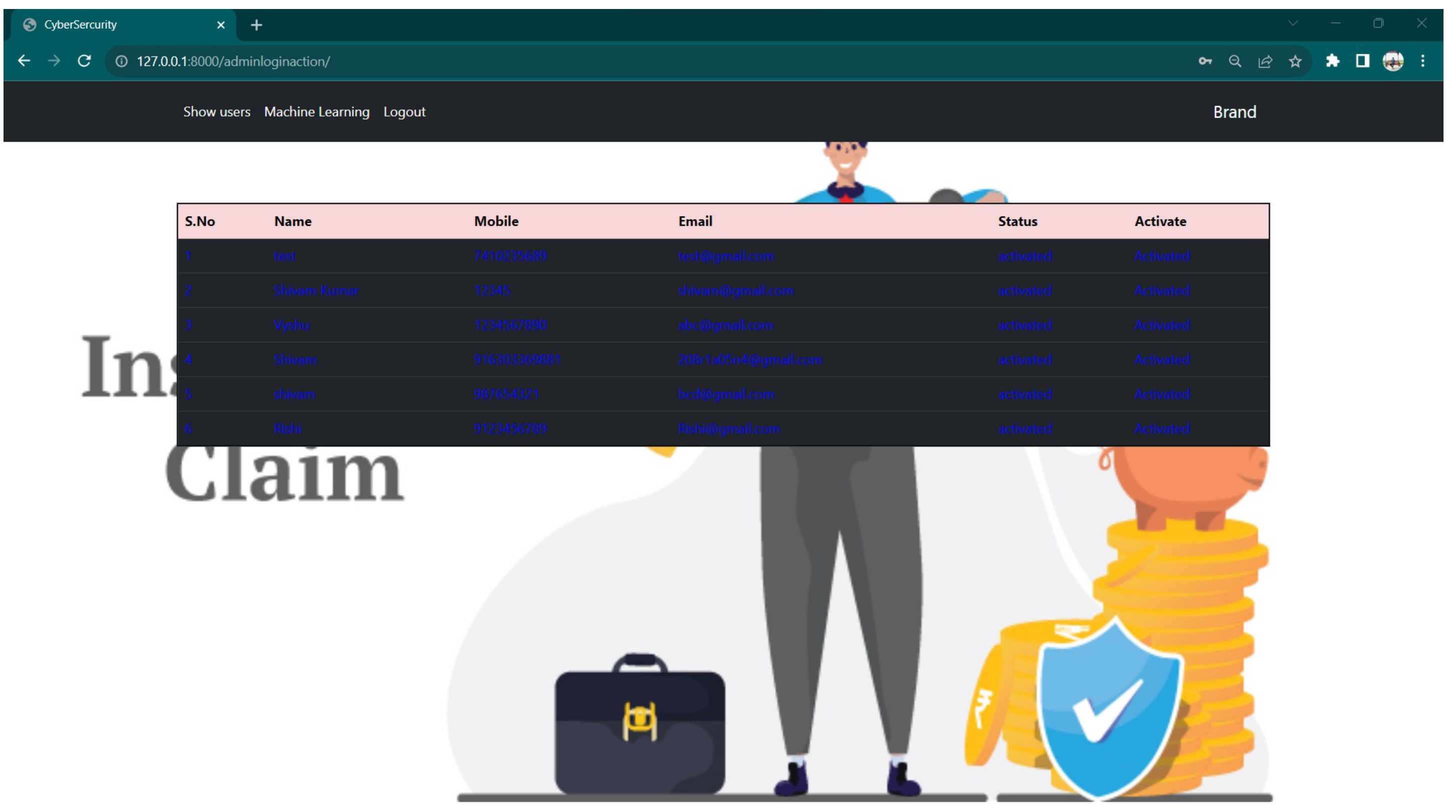


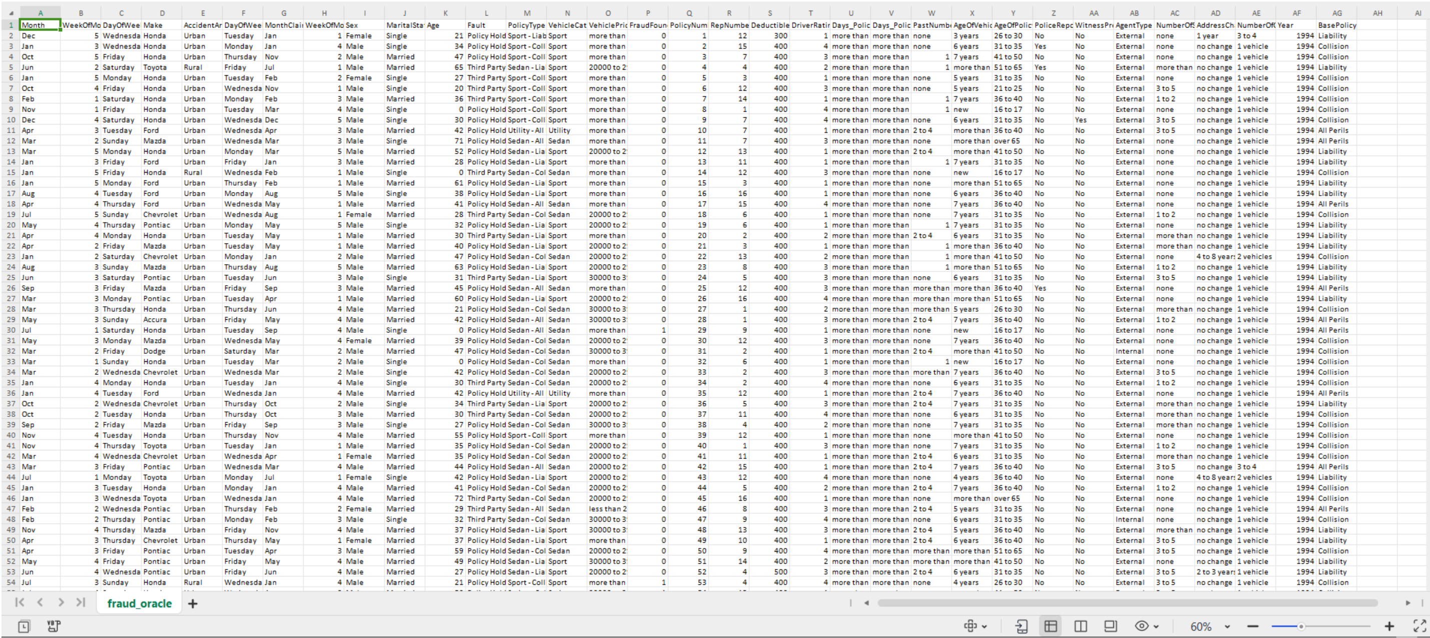



CyberSercurity
127.0.0.1:8000/Userlogin/
Home USER ADMIN
Brand
Insurance Claim
INSURANCE CLAIM
Enter User Email
Enter Password
Login
Register

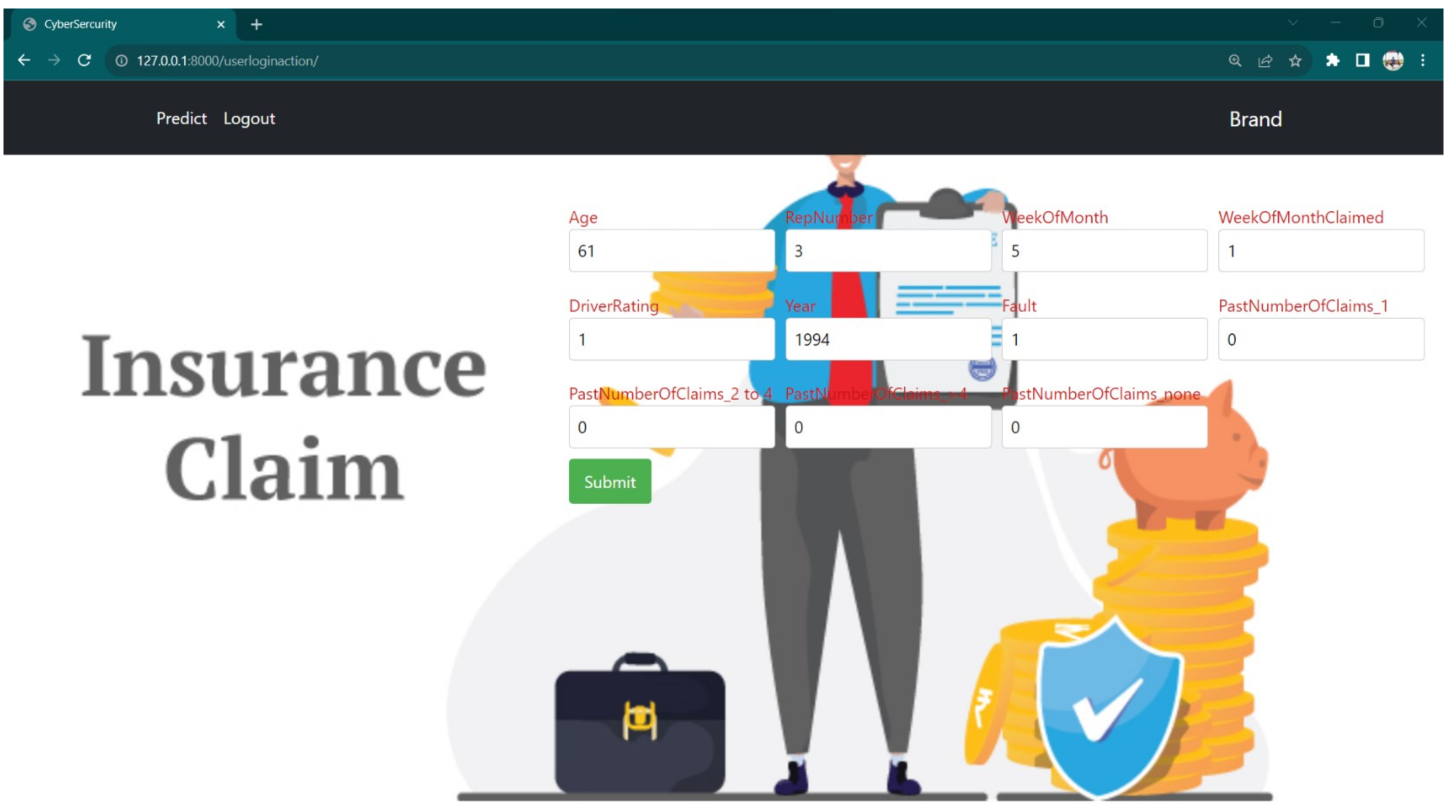

CyberSercurity
127.0.0.1:8000/userregister/
Home USER ADMIN
Brand
Insurance Claim
INSURANCE CLAIM
Enter Username
Enter Email
Enter Password
Enter Phonenumber
Register


CyberSercurity
127.0.0.1:8000/adminlogin/
Home USER ADMIN
Brand
INSURANCE CLAIM
Insurance Claim
Enter Admin ID
Enter Password
Login

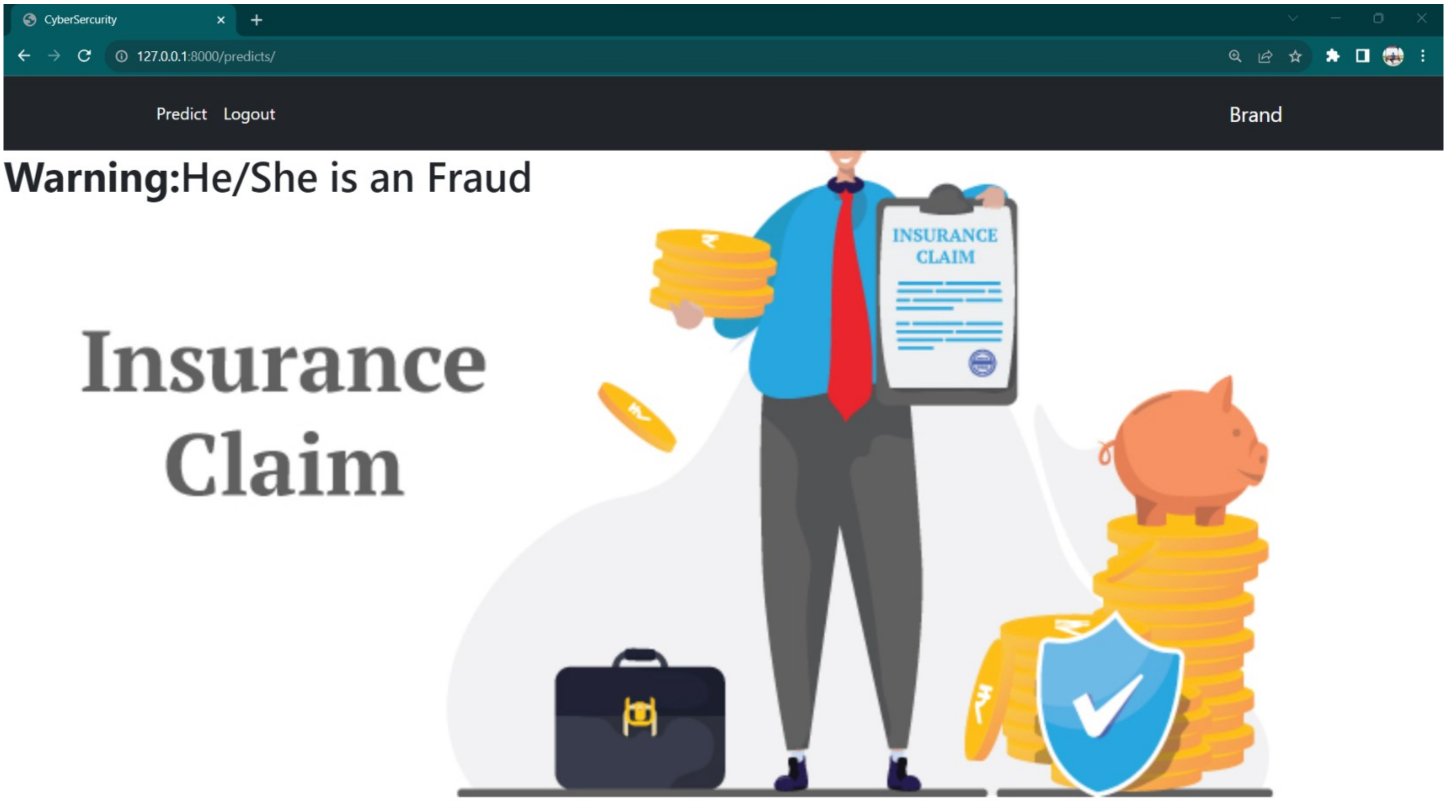

CyberSercurity
127.0.0.1:8000/adminloginaction/
Show users Machine Learning Logout
Brand
S.No Name Mobile Email Status Activate
1 test 7410235689 test@gmail.com activated Activated
2 Shivam Kumar 12345 shivam@gmail.com activated Activated
3 Vyshu 1234567890 abc@gmail.com activated Activated
4 Shivam 916303369881 208r1a05n4@gmail.com activated Activated
5 shivam 987654321 bcd@gmail.com activated Activated
6 Rishi 9123456789 Rishi@gmail.com activated Activated
Claim



fraud_oracle
60%



CyberSercurity
127.0.0.1:8000/userloginaction/
Predict
Logout
Brand
Insurance Claim
Age
61
RepNumber
3
WeekOfMonth
5
WeekOfMonthClaimed
1
DriverRating
1
Year
1994
Fault
1
PastNumberOfClaims_1
0
PastNumberOfClaims_2 to 4
0
PastNumberOfClaims_more than 4
0
PastNumberOfClaims_none
0
Submit


CyberSercurity
127.0.0.1:8000/predicts/
Predict
Logout
Brand
Warning:He/She is an Fraud
Insurance
Claim
INSURANCE
CLAIM

9. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS:

- **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.

Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:
Valid Input: identified classes of valid input must be accepted.
Invalid Input: identified classes of invalid input must be rejected.
Functions: identified functions must be exercised.
Output: identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined. System Test System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

- **White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. Itis used to test areas that cannot be reached from a black box level.

- **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

- **Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- All field entries must work properly.
- All field entries must work properly.

Features to be tested:

- Verify that the entire are of correct format
- No duplicate entire should be allowed
- All the link should take the user to the correct page

- **Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. The task of the integration test is to check that components or software applications, e.g. interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

- **Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

10. EXPERIMENTAL RESULTS

Different measures can be used to evaluate and analyze the Model Performance. Some of the measures used in this project are:

TABLE II. PRECISION ANALYSYS

Model	Recall	Precision	F1 Score
Logistic Regression	79	90	83
XGB	74	89	81
Decision Tree	66	79	71.86
KNN	65	75	68
`Forest Tree	45	77	56

In this analysis, several factors were known which can facilitate to spot for associate degree correct distinction between fraud transactions and non-fraudulent transactions that helps to predict the presence of fraud within the given transactions. Once completely different input datasets are used the Machine Learning models performed at variable performance levels. By considering average F1 score, model rankings are obtained. Higher the F1 score, higher the performance of the model. The analysis indicates that the Adjusted Random Forest formula and changed random below sampling formula provides best performance models.

However, it cannot be assumed that order of prophetical quality would be replicated and might differ for alternative datasets. Once discovered it’s complete that within the dataset samples, the models with datasets that are feature made, performs well.

10.1 TRAINING AND TESTING PHASE:

In this analysis, several factors were known which can facilitate to spot for associate degree correct distinction between fraud transactions and non-fraudulent transactions that helps to predict the presence of fraud within the given transactions. Once completely different input datasets are used the Machine Learning models performed at variable performance levels. By considering average F1 score, model rankings are obtained. Higher the F1 score, higher the performance of the model. The analysis indicates that the Adjusted Random Forest formula and changed random below sampling formula provides best performance models.

However, it cannot be assumed that order of prophetical quality would be replicated and might differ for alternative datasets. Once discovered it's complete that within the dataset samples, the models with datasets that are feature made, performs well. Obtained during training and testing phase. Depending on various features the trends are analyzed and hence used to decide the best model among the various Machine Learning classifiers. The following figures gives some of the graphical representation of the results.

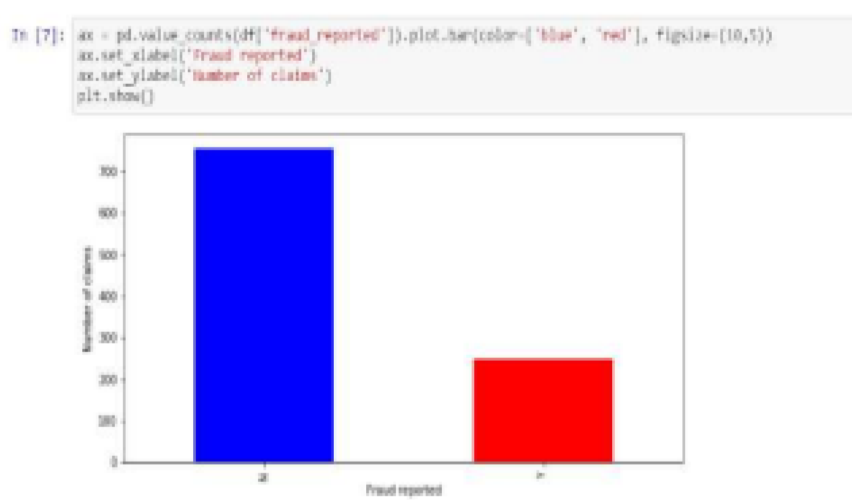


Fig. 3. Fraud Reported

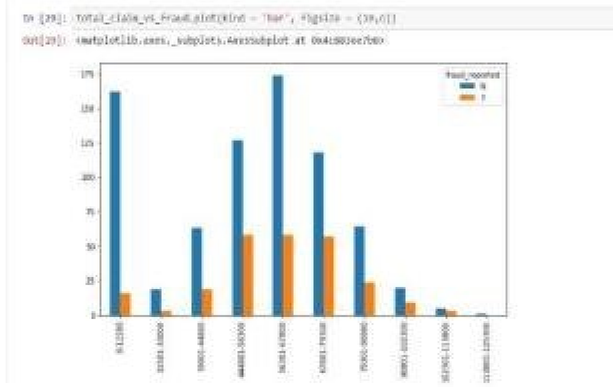
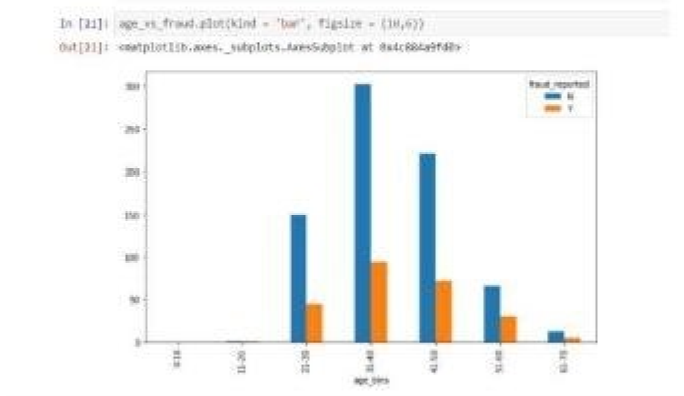


Fig. 4. Total Claim

B. Feature Selection



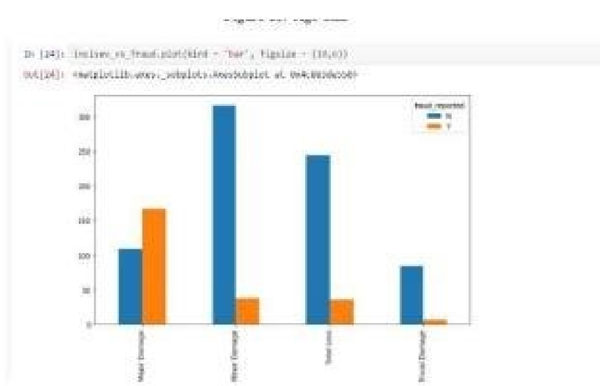


Fig. 6. Incident Sevirty

The following figures gives idea about feature selection. Here we drop column insured education level, insured occupation, authorities contacted since they have very high unique values which will lead to higher number of independent states.

```
In [16]: data[['policy_state', 'insured_education_level', 'insured_occupation','incident_type', 'collision_type', 'incident_severity', 'authorities_contacted']].describe()
```

Out[16]:

	policy_state	insured_education_level	insured_occupation	incident_type	collision_type	incident_severity	authorities_contacted
count	1000	1000	1000	1000	1000	1000	1000
unique	3	7	14	4	3	4	5
top	OH	JD	machine-op-inspct	Multi-vehicle Collision	Rear Collision	Minor Damage	Police
freq	352	161	95	419	470	354	292

Fig. 7. Feature Selection

10.2 ONE HOT ENCODING:

The following figures gives idea about converting all categorical values to numerical using One Hot Encoding. On Hot Encoding allows the representation of categorical data to be more expressive. It is required because many machine learning algorithm unable to work and give expected results with categorical data.

Fig. 8. Data Transformation into Numerical Data

Output of the model:

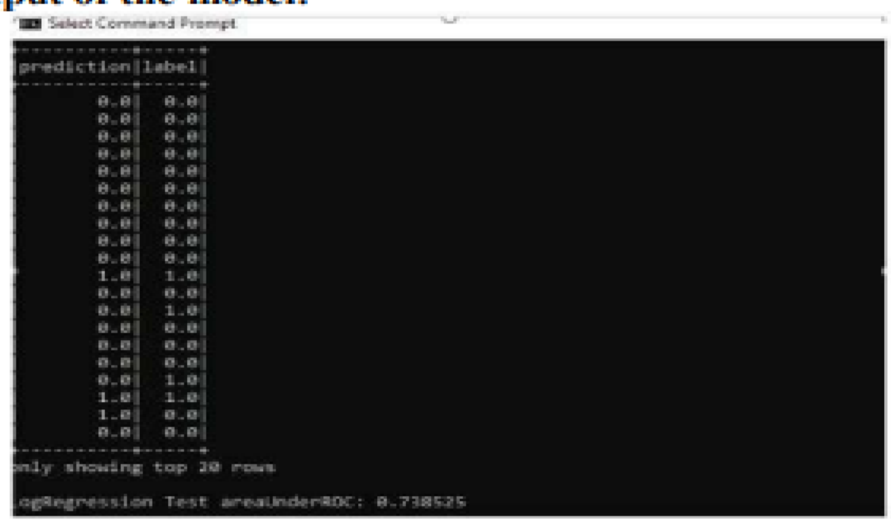


Fig. 9. Ouput for Logistic Regression Model

Input: Auto insurance fraud detection dataset containing 1000 records and 35 features. **Output:** Table comparing the actual results and the predicted results of the model, Where 0.0 stands for ’no fraud’ and 1.0 stands for ’fraud’. • Accuracy of Logistic Regression model.

11. CONCLUSION

The machine learning models that square measure mentioned which square measure applied on these datasets were able to determine most of the fallacious cases with low false positive rate which suggests with cheap exactness. Certain knowledge sets had severe challenges around data quality, resulting in comparatively poor levels of prediction. Given inherent characteristics of varied datasets, it would not be sensible to outline optimum algorithmic techniques or use feature engineering process for a lot of higher performance. The models would then be used for specific business context and user priorities. This helps loss management units to specialize in a replacement fraud situations and then guaranteeing that models square measure adapting to spot them. However, it might be cheap to counsel that supported the model performance on back-testing and talent to spot new frauds, the set of models work the cheap suite to use within the space of the insurance claims fraud detection.

12. REFERENCE

1. **Title:** Fraudulent Insurance Claims Detection Using Machine Learning
   Authors: Abhijeet Urunkar, Amruta Khot, Rashmi Bhat, Nandinee Mudegol
   Publication: 2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)
   Year: 2022

2. **Title:** Sequence Embeddings Help Detect Insurance Fraud
   Authors: Ivan Fursov, Alexander Gusev, Vladimir Iglovikov, Nikita Kolesnikov, Vladislav Nikonov **Publication:** IEEE Access
   Year: 2022

3. **Title:** Block Chain Application in Insurance Services: A Systematic Review of the Evidence **Authors:** Anjali Shetty, A.D. Shetty, R.Y. Pai, R.R. Rao, Roopa Bhandary, Jagadish Shetty, Sandeep Nayak, T. Keerthi Dinesh, Kevin John Dsouza
   Publication: SAGE Open
   Year: 2022

4. **Title:** Fuzzy clustering using salp swarm algorithm for automobile insurance fraud detection **Authors:** S. S. Majhi, S. Bhattachharya, R. Pradhan, S. Biswal
   Publication: Journal of Intelligent & Fuzzy Systems
   Year: 2019

5. **Title:** Model for Credit Card Fraud Detection using Machine Learning Algorithm
   Authors: P. Singh, V. Chauhan, S. Singh, P. Agarwal, S. Agrawal
   Publication: 2021 International Conference on Technological Advancements and Innovations (ICTAI)
   Year: 2021

6. **Title:** Comparative Analysis of ML Algorithm QDA, LR and SVM for Credit Card Fraud Detection Dataset
   Authors: Naveen and B. Diwan

Publication: Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)
Year: 2020

7. **Title:** Predictive Analytics For Insurance Fraud Detection - Wipro
Authors: Wipro Insurance Analytics
Publication: Wipro Thought Paper
Year: 2022

8. **Title:** Comparative Analysis of Machine Learning Techniques for Detecting Insurance Claims Fraud
Authors: Naveen and B. Diwan
Publication: Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)
Year: 2020

9. **Title:** Fraud Detection in Healthcare Insurance Claims Using Machine Learning
Authors: MDPI
Publication: MDPI
Year: 2023

10. **Title:** Healthcare Insurance Fraud Detection Using Machine Learning Techniques
Authors: Alonazi, Faisal
Publication: Journal of Applied Research in Engineering and Technology
Year: 2020

11. **Title:** Insurance Fraud Detection Using Novel Machine Learning Technique
Authors: Naveen and B. Diwan
Publication: Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)
Year: 2020

12. **Title:** Machine Learning for Insurance Fraud Detection: A Literature Review
Authors: Thaifur, Rahma Dwi; Purwanto, Edy; Wijaya, Andi Wahyudi
Publication: International Journal of Electrical & Computer Engineering (IJECE)
Year: 2021

13. **Title:** A Hybrid Machine Learning Approach for Insurance Fraud Detection
Authors: Li, Meng; Zhang, Jing; Qin, Yuanyuan; Chen, Xiaoling
Publication: IEEE Access
Year: 2022

14. **Title:** Deep Learning for Insurance Fraud Detection: A Review
Authors: Kumar, Ankit; Kumar, Deepak; Mishra, Shubhangi; Kumar, Manish
Publication: IEEE Access
Year: 2023

15. **Title:** Insurance Fraud Detection Using XGBoost and Logistic Regression
Authors: Chen, Jun; Zhang, Xiaodong; Zhang, Jingjing; Zhang, Qi
Publication: IEEE Access
Year: 2022