Credit Card Fraud Analysis

Team Members:

16BCB0062 (VAASU GUPTA)
16BCB0129 (SIDDHART SHAILENDRA)
17BCB0070(JAI KUMAR)

Final Report submitted for the Project Review of

Course Code: CSE1005 – Software Design & Development

Slot: G1

Professor: Usha K

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

Content

I) Introduction

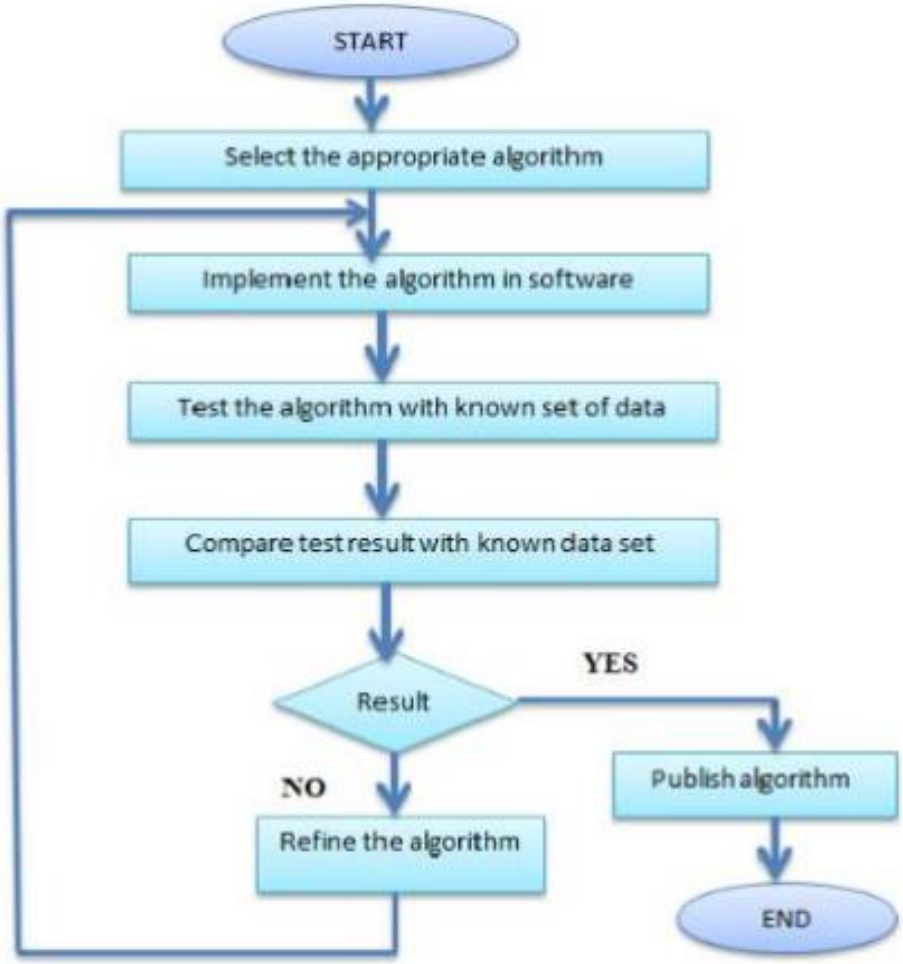
Nowadays when the term fraud comes into a discussion, credit card fraud clicks to mind so far. With the great increase in credit card transactions, credit card fraud has increasing excessively in recent years. Fraud detection includes monitoring of the spending behavior of users/ customers in order to determination, detection, or avoidance of undesirable behavior. As credit card becomes the most prevailing mode of payment for both online as well as regular purchase, fraud relate with it are also accelerating. Fraud detection is concerned with not only capturing the fraudulent events, but also capturing of such activities as quickly as possible. The use of credit cards is common in modern day society. Fraud is a millions dollar business and it is rising every year. Fraud presents significant cost to our economy worldwide. Modern techniques based on Data mining, Machine learning, Sequence Alignment, Fuzzy Logic, Genetic Programming, Artificial Intelligence etc., has been introduced for detecting credit card fraudulent transactions. This paper shows how data mining techniques can be combined successfully to obtain a high fraud coverage combined with a low or high false alarm rate. Along with the great increase in credit card transactions, credit card fraud has become

increasingly rampant in recent years. In Modern day the fraud is one of the major causes of great financial losses, not only for merchants, individual clients are also affected. Fraud is a criminal practice for illegitimate gain of wealth or tampering information. Fraudulent activities are of critical concern because of their severe impact on organizations, communities as well as individuals. Over the last few years, various techniques from different areas such

as data mining, machine learning, and statistics have been proposed to deal with fraudulent activities.

1.1 Objective

Along with the great increase in credit card transactions, credit card fraud has become increasingly rampant in recent years. In Modern day the fraud is one of the major causes of great financial losses, not only for merchants, individual clients are also affected. Fraud is a criminal practice for illegitimate gain of wealth or tampering information. Fraudulent activities are of critical concern because of their severe impact on organizations, communities as well as individuals. Over the last few years, various techniques from different areas such as data mining, machine learning, and statistics have been proposed to deal with fraudulent activities.



Proposed Algorithm for the system

1.2 Motivation

The motivation and opportunity behind credit card fraud are many and varied. Traditional types of fraudulent behaviour such as identity theft relate to family members or people that can easily access individual's mail and personal information and committing fraud either by applying for card or taking over the existing account. Dumpster diving or trashing, where criminals raid rubbish bins to search for credit card details and other sensitive information is becoming

more widespread. Lost or stolen credit cards may also be used fraudulently. Skimming of the magnetic stripe is also still practiced either using highly sophisticated devices embedded in ATM's or POS or using simple hand held skimmers capable of storing magnetic stripe data.

Internet enabled fraud is also growing; phishing attacks continue to harvest credit card users' details and compromised computer with key loggers provide organised criminals with the card details. As the vast majority of all credit card transactions are now authorised and cleared on-line, hacking into the e-payment chain to intercept data can harvest many millions of card details. The e-fraud market has grown. Criminals are now provided with various internet resources to counterfeit credit cards, examples are tipping, custom embossing, decoding machines as well as software such as Creditmaster. A common practise is also that of phishing where fraudulent emails hijacking brand name of banks, credit cards companies, (etc.) are sent aimed at acquiring trickily financial data, account usernames and passwords. Organised crime are normally composed by professional criminals that are setting "carding forums" where it is possible to buy wide-scale global stolen personal and financial information. This practise that leads to the unauthorised use of sensitive information to purchase goods and services often involves thousands and even millions of victims (Peretti, 2008). Indeed credit card fraud is subject to technological enhancement and it is in a continuous evolution.

1.3 Background

Ability of system to automatically learn and improve from experience without being explicitly programmed is called machine learning and it focuses on the development of computer programs that can access data and use it learn for themselves. And classifier can be stated as an algorithm that is used to implement classification especially in concrete implementation, it also refers to a mathematical function implemented by algorithm that will map input data into category. It is an instance of supervised learning i.e. where training set of correctly identified observations is available.

II) Project Description and goals

Product Perspective

Credit is a method of selling goods or services without buyer having cash in mind. A credit card is only an automatic way to offer credit to the consumer. Today every credit cards carries an identifying number that speeds shopping

transactions. In credit card business fraud occurwhen lender is fooled by the borrower offering him/her purchase believing that borrower credit card account will provide payment for this purchase. Ideally no payment will be made. If the payment is made the credit card issuer will reclaim the amount paid. Today with expansion of e-commerce it is on the internet that half of the fraud is conducted. Fraudsters have usually connections with the affected business. Neural networks are also recommended as effective credit card fraud detection methods. The only issue with this method is that all data has to be clustered by the type of account it belongs to. Credit card fraud is a major issue that if not dealt with effectively, it can result in myriad complications. It is vital to try and find ways of detecting the issues and resolving them as soon as they arise. A deviation from such a pattern will then lead to the system classifying it as suspicious transaction and will then be handled accordingly. The proposed system does a job of classifying the transactions into one of the two types: 1. Suspicious Transaction 2. Non-Suspicious Transaction The classification of transaction is done using the following algorithm as depicted in figure 1. The system checks the location and the pattern of spending as the major parameters to decide a spurious transaction. If there is a mismatch in the location or the pattern, the system marks it as suspicious and subjects the transaction through a verification process. The verification could be any process such as alerting the user, calling the user or subject the user to go through another round of clearance such as OTP. If the transaction is valid, then the details are updated for the user. After verification if the system fails to identify the user, the transaction is declined. Since the accuracy depends on location and pattern, it is possible to get false alarms and a valid user may be subjected through verification process. It is also possible that the system could decline a valid transaction.

Product Functions

• User enrolment

In this a new user may have to login into the cloud in order to get access to the GPU

• Continuous User Authentication

The cloud will continuously authenticate an enrolled user. Hence if the cloud encounters any fraud transaction it sends it to the model

• Lock out Impostors

If an unauthorized transaction details is detected by the cloud, then the cloud will immediately reject it and it will get stored in the log.

• Log

A log would be generated for every login or logout for the user and if an unauthorized transaction will be detected then that will be added in the log.

User Classes and Characteristics

Types of users:

• Client

1. Name
2. Age, Gender
3. Address
4. Transaction History
5. Username, Password for Login

• Admin

1. Username, Password for login
2. Access to some features of the database

Operating Environment

• PC/Laptop

• Web Browsers (Chrome/Edge/Firefox)

• Client System

• Server Backend Storage with Mongo

Goals

This system is capable of providing most of the essential features required to detect fraudulent and legitimate transactions. As technology changes, it becomes difficult to track the behaviour and pattern of fraudulent transactions. We have just detected the fraudulent activity but we have not prevented. Preventing known and unknown fraud in real time is not easy but it is feasible. The proposed architecture is basically designed to detect credit card fraud in online payments, and emphasis is made to provide a fraud prevention system to verify a transaction as fraudulent or legitimate. For implementation purposes it is assumed that issuer and acquirer bank is connected to each other. If this system is to be implemented in real time scenario, then exchange of best practices and raising consumer awareness among people can be very helpful in reducing the losses caused by fraudulent transactions. Further enhancement can be done by making this system secure with the use of certificates for both merchant and customer and as technology changes new checks can be added to understand the pattern of fraudulent transactions and to alert the respective card holders and bankers

when fraud activity is identified. The dataset available on day to day processing may become outdated, it is necessary to have updated data for effective fraud behavior identification. To this extent, the incremental approach is necessary in making the system to learn from past as well as present data and capable of handling the both. Fraudster uses different new techniques that are instantaneously growing along with new technology makes it difficult for detection. Also the nature of access pattern may vary from one geographical location 161 to another (such as urban and rural areas) that may result in a false positive detection. In such a case a future enhancement may be based on new multiple models with varying access pattern needs attention to improve the effectiveness. Privacy preserving techniques applied in distributed environment resolves the security related issues preventing private data access.

III) Technical Specification

Algorithms used to predict the fraud is

Logistic Regression

- Linear Discriminant Analysis
- K Nearest Neighbors (KNN)
- Classification Trees
- Support Vector Classifier
- Random Forest Classifier
- HTML, CSS
- JavaScript
- Wammp Server

IV) Design Approach and Details

PROCESS MODEL IDENTIFICATION AND JUSTIFICATION

We are using the iterative and incremental development model for this project which is a combination of both iterative design or iterative method and incremental build model for development.

Model Identification:

As our task requires repeated testing and fine tuning we preferred the Iterative model. Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles and in smaller portions at a time

The reason why we chose this model is that nowadays a lot of frauds are occurring in credit card transactions. New types of frauds are being introduced. So this system is designed to detect such frauds in the credit card transactions. In this model risks are identified very quickly and then the high risk part is dealt with first and improvements are done accordingly

Also Operational product is designed v

Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

Advantages of the Iterative SDLC Model:

- Some working functionality can be developed quickly and early in the life cycle.
- Parallel design and development can be done.
- Progress can be measured.

- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration; and each iteration is an easily managed milestone, High risk part is done first.
- With every iteration, operational product is delivered.
- During the life cycle, software is produced early which facilitates customer evaluation and feedback.

Disadvantages of the Iterative SDLC Model:

- Although cost of change is lesser, but it is not very suitable for changing requirements.
- System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.
- Highly skilled resources are required for risk analysis
- Projects progress is highly dependent upon the risk analysis phase.
- Defining increments may require definition of the complete system.
- More resources may be required.

Why Iterative and Incremental Development Model:

The main cause due to which most of the software development projects fail is the choice of the model. Hence a software development model should be made with great concern.

For example, the Waterfall development paradigm completes the project-wide workproducts of each discipline in one step before moving on to the next discipline in the next step. Business value is delivered all at once, and only at the very end of the project. whereas backtracking is possible in an iterative approach. Comparing the two approaches, some patterns begin to emerge.

- User involvement: In waterfall model, the user is involved in one stage of the model, i.e. requirement. Whereas in the Incremental model, the client is involved at each and every stage.
- Variability: The software is delivered to the user only after all the stages of life cycle is completed. On the other hand, every increment is delivered to the user and after the

approval of user, the developer is allowed to move towards the next module.

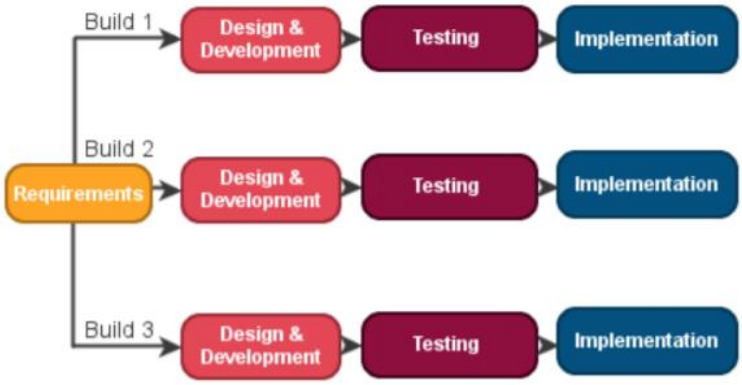
 Human resources: In incremental model less staff is required as compared to waterfall model.

 Time limitation: Operational quality portion is delivered after months while in the Incremental model the operational product is given to the user within a few weeks.

 Project size: Waterfall model is unsuitable for small projects while incremental model is best suitable for small as well as large projects.

In contrast with the V model, the flexibility is less as compared to iterative model, also no prototypes/versions of the software are produced whereas in iterative model through phases new versions are produced which makes it a better option. Also V model does not produce a clear path for problems found during testing phases

Iterative or Incremental Development Model



Iterative or Incremental Development Model

1. Design Overview

System Architecture

This Software Design Specification (SDS) will cover the credit card fraud detection software using Naïve Bayes Classifier

The purpose of using Naïve Bayes Classifier is to ensure that no fraudulent activity occurs during the transaction. It generally compares all the other datasets to detect any fraudulent activity and if any such fraudulent activity occurs it blocks the transaction

We have used multiple architectural styles in this

software. Namely:- 1) Client Server Architecture:

Client-server architecture, is an architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them.

Client/server architecture is also known as a networking computing model or client/server network because all the requests and services are delivered over a network.

- This type of architecture has one or more client computers connected to a central server over a network or internet connection. This system shares computing resources.
- Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service. This is a property called abstraction. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform.

 Many clients can access the server's information simultaneously. Because both client and server computers are considered intelligent devices, the client-server model is completely different from the old "mainframe" model.

Client/server architecture will be an apt backbone of the project, since the software is divided into two parts, namely client and server. The server in this case contains the client details, account information and the transaction details. The client can access and manipulate this data from the server.

2) Rule Based Architecture:

Rule-based systems architecture is used as a way to store and manipulate knowledge to interpret information in a useful way.

Normally, the term 'rule-based system' is applied to systems involving human-crafted or curated rule sets.

A typical rule-based system has four basic components:

- A list of rules or rule base, which is a specific type of knowledge base.

- An inference engine or semantic reasoner, which infers information or takes action based on the interaction of input and the rule base. The interpreter executes a production system program by performing the match-resolve-act cycle.

- Temporary working memory

- A user interface or other connection to the outside world through which input and output signals are received and sent.

This type of system uses knowledge encoded in the form of production rules i.e. if-then rules. The rule has a conditional part on the left hand side and a conclusion or action part on the right hand side. Each rule represents a small chunk of knowledge to the given domain of expertise. When the known facts support the conditions in the rule's left side, the conclusion or action part of the rule is then accepted as known

The software will be developed using a mix of these two types of architectures. The main portal will be built on the basis of client server architecture while the rest of the software will be done based on rule based architecture

1.2 System Interface

- User Interface:

1) The user interface must be aesthetic.
2) The interface should be responsive for all kinds of devices like a PC, tablet, smartphone, etc. 3) One type of user should not have access to the private stored data of other users.

- Hardware Interface:

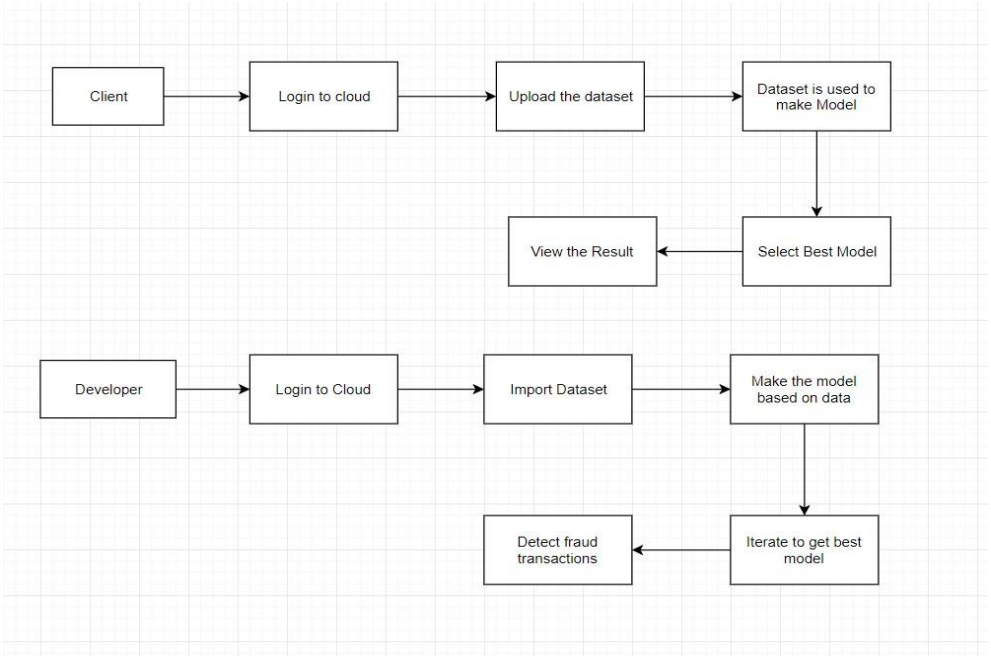
These are the specifications of the system:

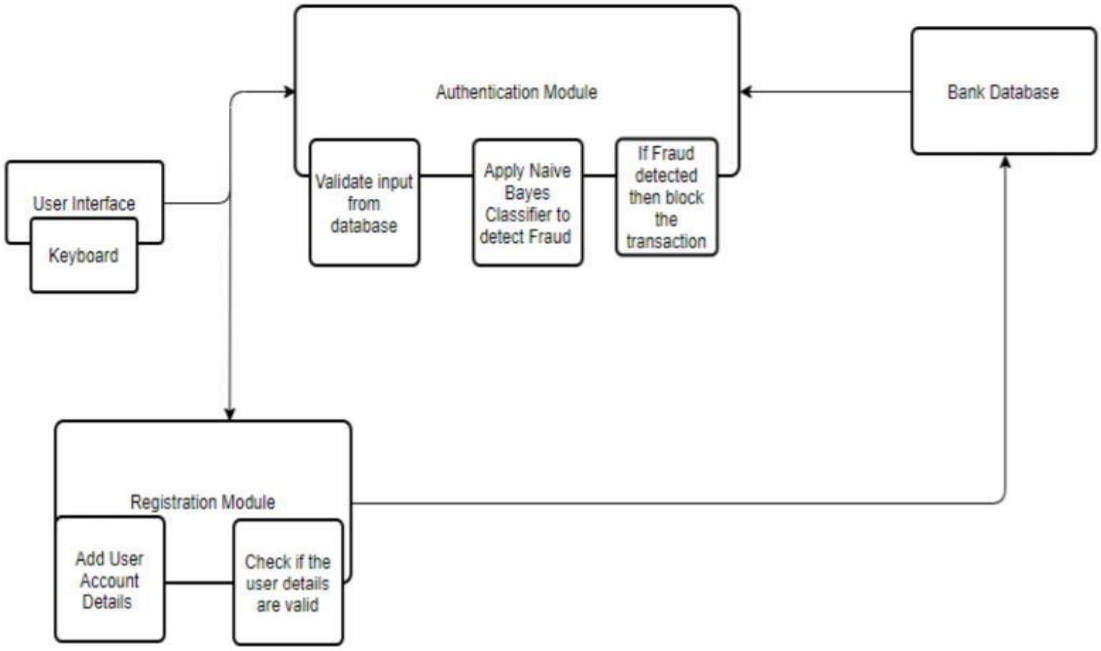
Hardware	Minimum System Requirement
Processor	1.5 GHz Processor speed
Memory	1 GB RAM
Network Connection	The device must be connected using a physical cable or Wi-Fi
Disk Space	500 MB
Keyboard	A fully functioning, well connected keyboard

- Software Interface:

1) An operating system (preferably Windows/Linux) has to be installed on the device.
2) The device needs to have a web browser (preferably Chrome or Firefox) to use the product.
3) There would be cookie and session variable storage on the browser for saving user data, which would be taken permission for, according to the new policy.

1.3 Subsystems





1.4) Subsystem Interfaces

- Login Portal

Here, the user will be asked to enter his username and password for the first step. A connection with the cloud is made and the details of the users is stored in the database. A login for developers is also provided in order to evaluate the fraud.

- Cloud

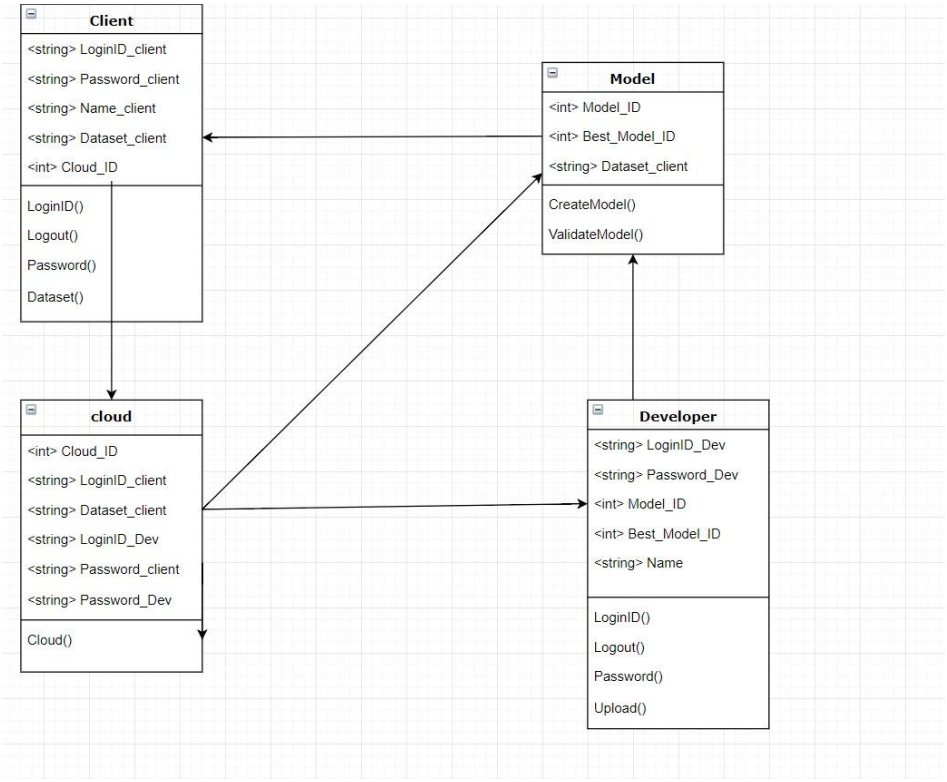
First, when the client logs in, he will have to enter his loginID and password a connection with the cloud server will be made and the user will be allowed to login into the cloud database. Moreover, he can view developer details. The client also has the option to upload the data which they want to be checked for fraudulent transactions.

- Model

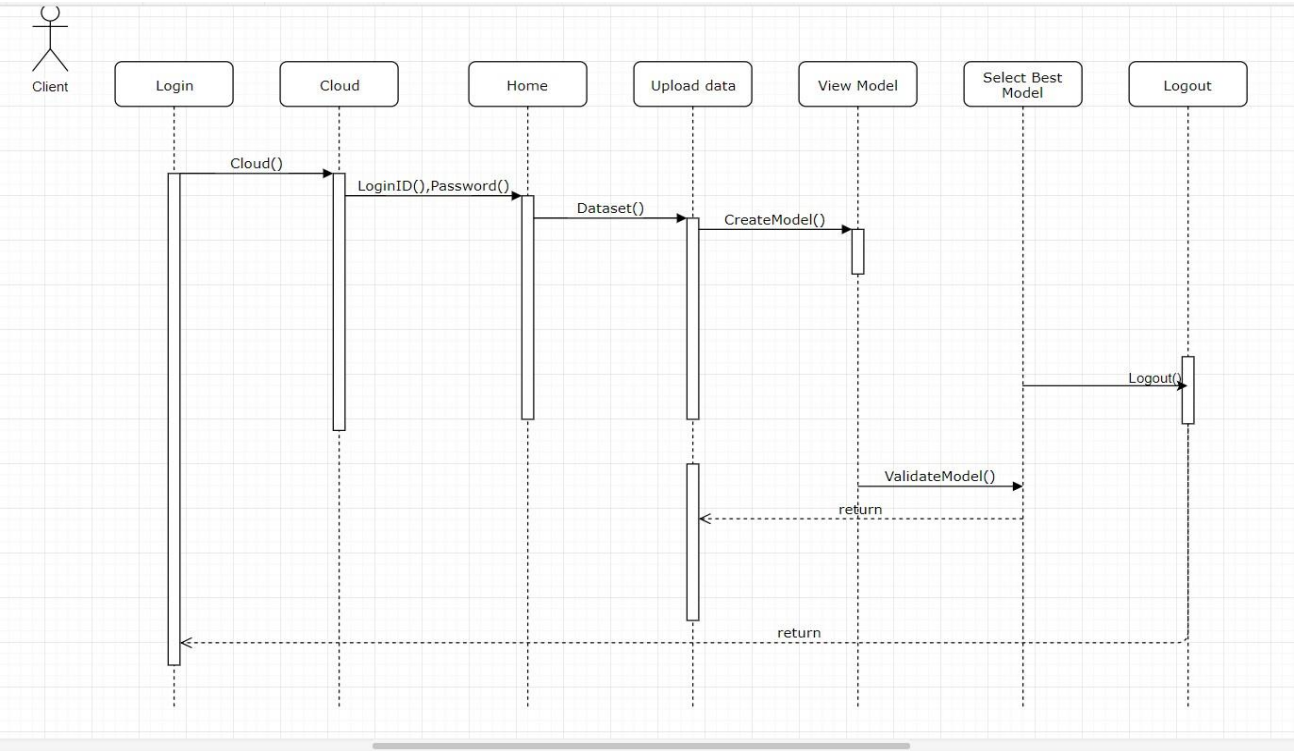
Once the developer gets the data to find the fraudulent transactions he/she has to make model which will suited to the client's demands. The model will be iteration with cross validation after which the client can select the best model possible for their problem.

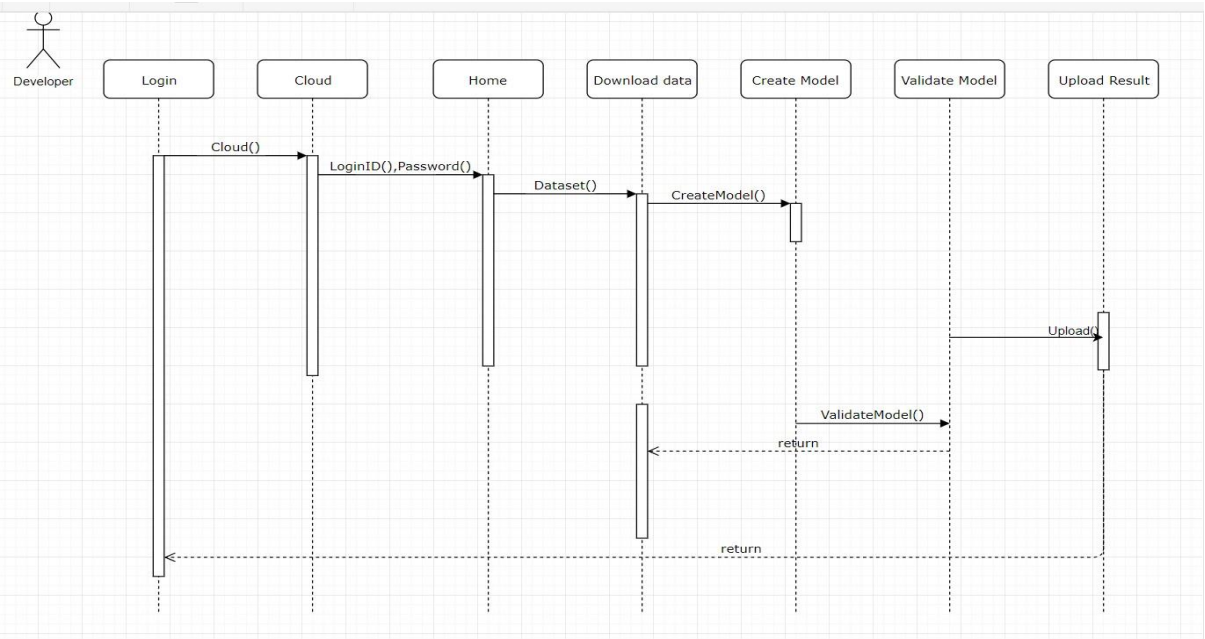
Diagrams

Class Diagram

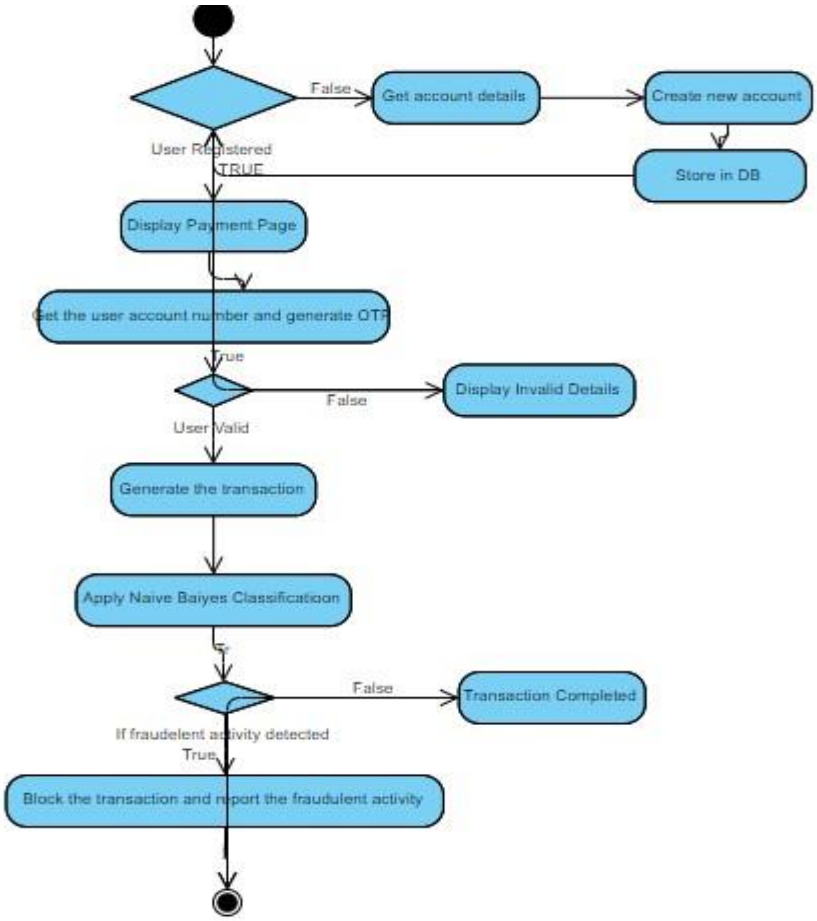


Sequence Diagram

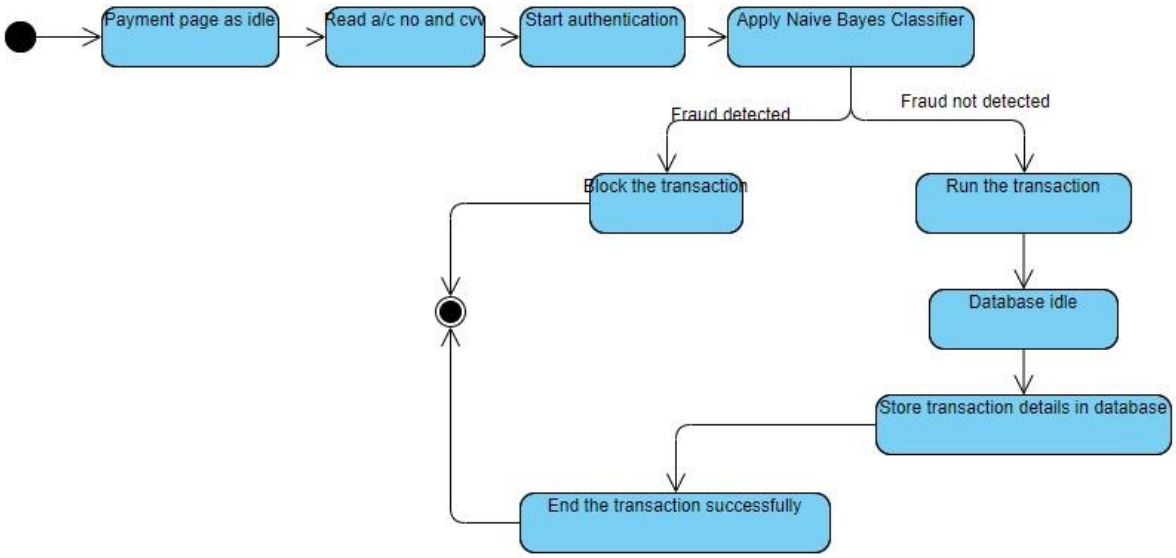




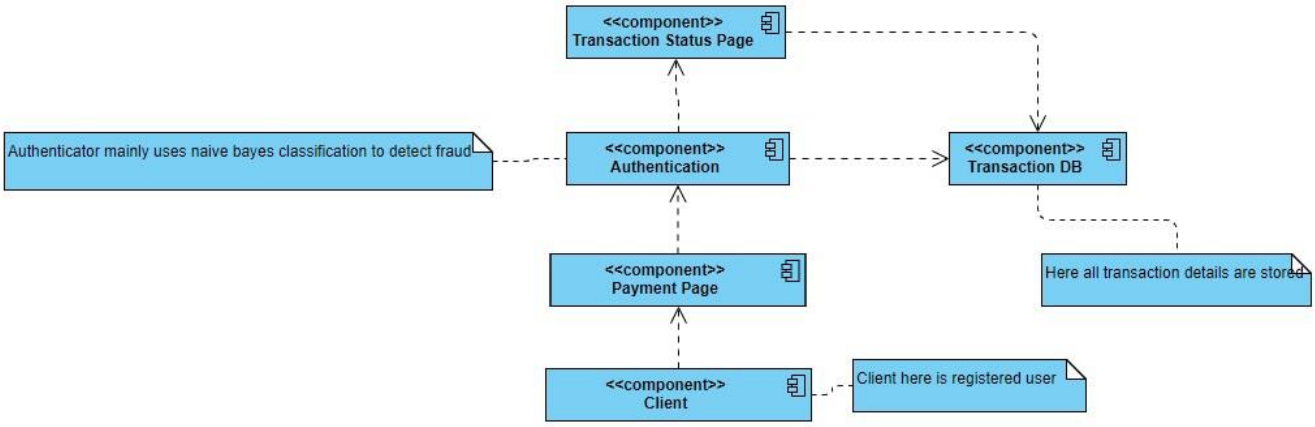
Activity Diagram



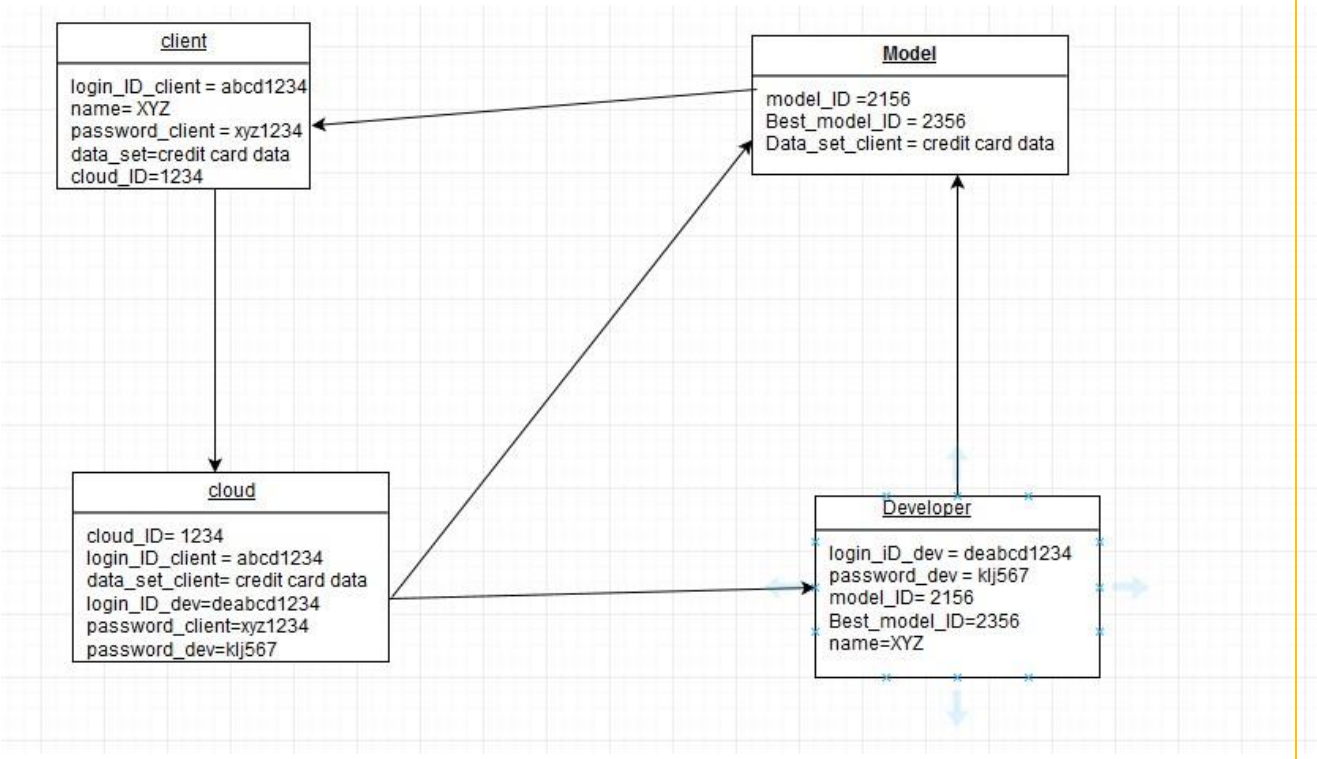
State Chart Diagram



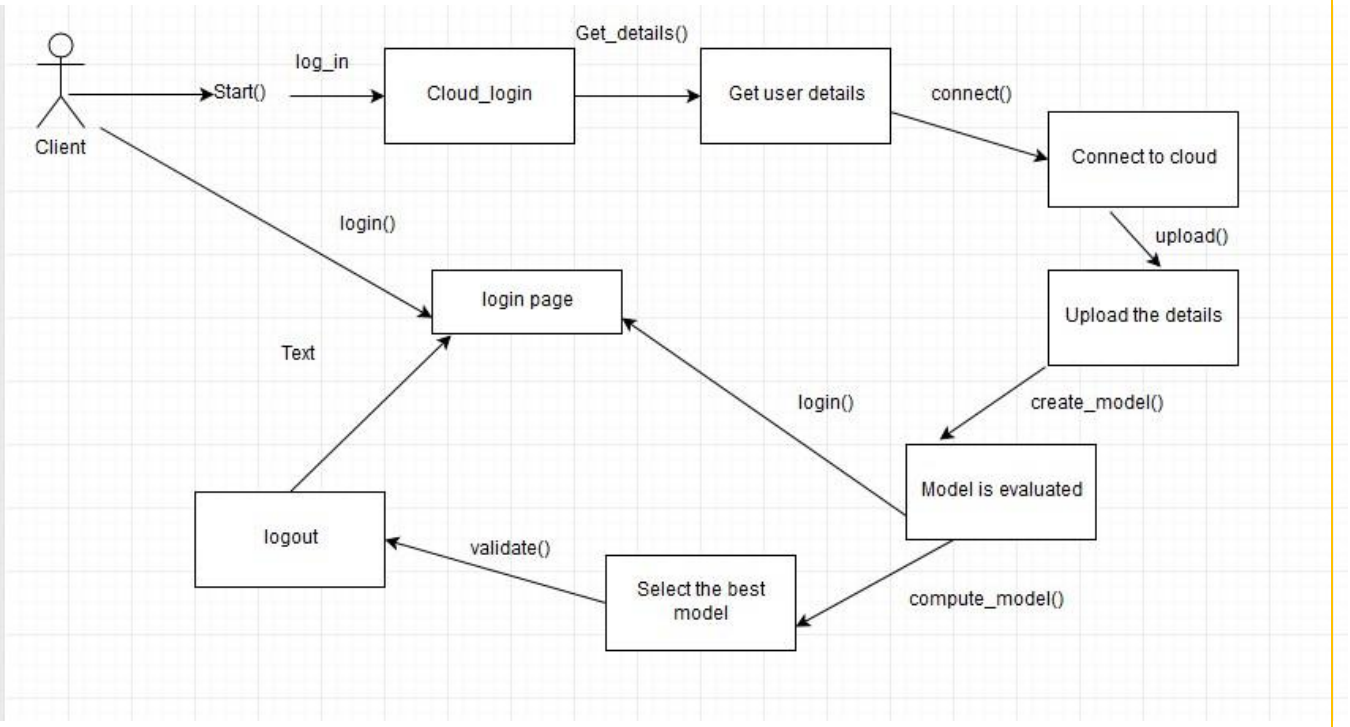
Component Diagram

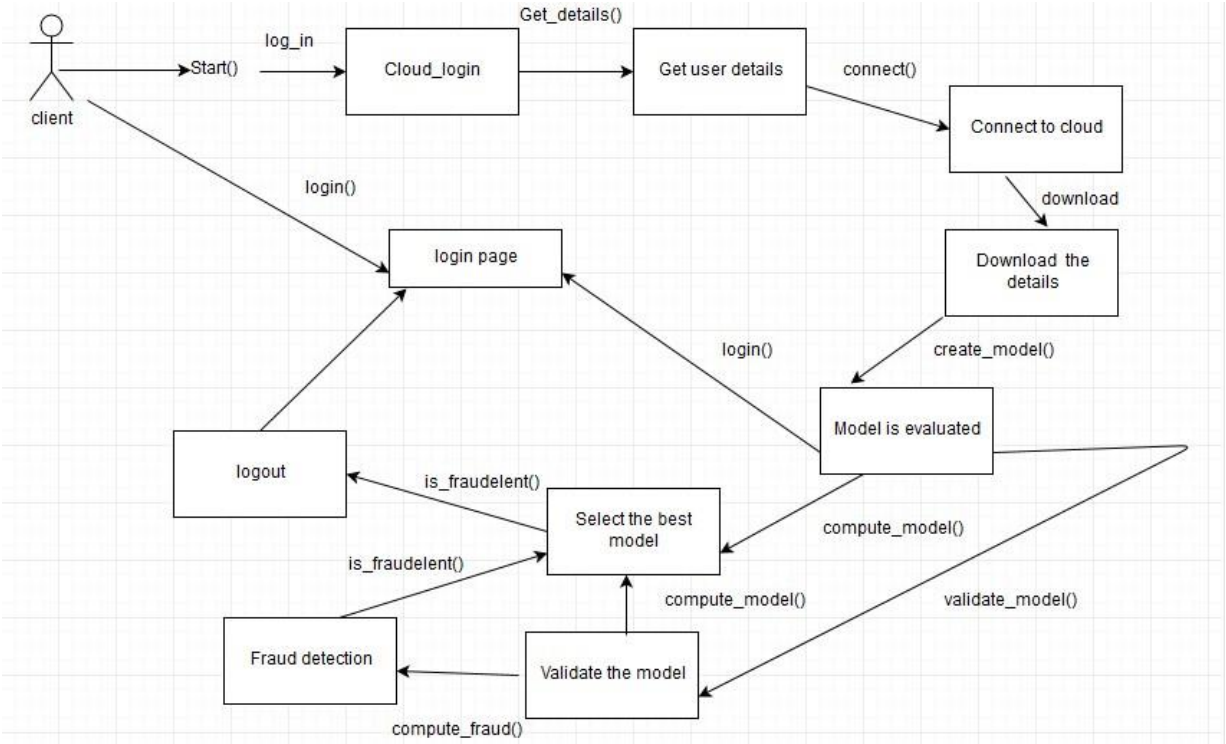


Object Diagram



Communication Diagram




client
Start()
log_in
Cloud_login
Get_details()
Get user details
connect()
Connect to cloud
download
Download the details
login()
login page
login()
create_model()
Model is evaluated
logout
is_fraudelent()
Select the best model
compute_model()
is_fraudelent()
compute_model()
validate_model()
Fraud detection
Validate the model
compute_fraud()

Coding

Module 1 (Login and Registering)

```
<!DOCTYPE HTML>
<html>
<head>
        <title>Fraud Detection</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
        <link rel="stylesheet" type="text/css" href="title.css">
        <link href="https://fonts.googleapis.com/css?family=IBM+Plex+Mono" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Merriweather" rel="stylesheet">

<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
  </script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

  <style>
  body {
background-image: url('im6.jpg');
background-repeat: no-repeat;
background-size: cover ;
}
  div.para{
```

```
    border: 5px solid black;
    width: 70%;
    margin-top:  auto;
    margin-left: auto;
    margin-right: auto;
    margin-bottom: 20px;
    color: black;
    background-color: white;
    padding: 20px;
  }

  div.container1{
            width: 70% ;
            height: 10%;
            margin: auto;

  }
</style>
</head>

<body background="im4.jpg">

            <nav class="navbar navbar-inverse navbar-fixed-top">
            <div class="container">
            <div class="navbar-header">
                        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
```

```
                <a href="#" class="navbar-brand"><span class= "glyphicon glyphicon-piggy-bank"></span> Fraud Detection</a>
            </div>
            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <div class = "nav navbar-nav navbar-right">
                    <li class="Merriwether"><a href="Login.html">LOGIN</a></li>
                    <li class="Merriwether"><a href="Sign Up.html">PATIENT SIGN UP</a></li>
                    <li class="Merriwether"><a href="DSignUp.php">DONOR SIGN UP</a></li>
                </div>
            </div>
            </div>
            </nav>



            </div>

<script src="http://code.jquery.com/jquery-3.3.1.js"
 integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
 crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>
<br><br>
<div class="container1">
  <div id="myCarousel" class="carousel slide" data-ride="carousel">
   <!-- Indicators -->
   <ol class="carousel-indicators">
     <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
     <li data-target="#myCarousel" data-slide-to="1"></li>
     <li data-target="#myCarousel" data-slide-to="2"></li>
   </ol>
```

```
    <!-- Wrapper for slides -->
    <div class="carousel-inner">
      <div class="item active">
        <img src="im1.jpg" alt="Los Angeles" style="width:100%;height: 20%;">
      </div>

      <div class="item">
        <img src="im2.jpg" alt="Chicago" style="width:100%; height: :20%;">
      </div>

      <div class="item">
        <img src="im3.jpg" alt="New york" style="width:100%;height: :20%;">
      </div>
    </div>

    <!-- Left and right controls -->
    <a class="left carousel-control" href="#myCarousel" data-slide="prev">
      <span class="glyphicon glyphicon-chevron-left"></span>
      <span class="sr-only">Previous</span>
    </a>
    <a class="right carousel-control" href="#myCarousel" data-slide="next">
      <span class="glyphicon glyphicon-chevron-right"></span>
      <span class="sr-only">Next</span>
    </a>
  </div>
</div>
<br><br><br>
<div class="para">
  <h2><center>NOTE</center></h2>
  <p><b>
```

Ever since starting my journey into data science, I have been thinking about ways to use data science for good while generating value at the same time.

Thus, when I came across this data set on Kaggle dealing with credit card fraud detection,I was immediately hooked. The data set has 31 features,

28 of which have been anonymized and are labeled V1 through V28. The remaining three features are the time and the amount of the transaction as well

as whether that transaction was fraudulent or not. Before it was uploaded to Kaggle, the anonymized variables had been modified in the form of a PCA

(Principal Component Analysis). Furthermore, there were no missing values in the data set.

This online system is developed on php and supported by an Sql database to store blood and user specific details.

</p>

</body>

</html>

<!DOCTYPE HTML>

<html>

<head>

<title>Sign Up</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="sign.css">

<style>

body {

background-image: url('im6.jpg');

background-repeat: no-repeat;

background-size: cover ;

}

</style>

</head>

<body>

<h1>Customer Details</h1>

```
<div class = "jumbotron">
<form name="user" action="http://localhost:8080/iwp/new2.php" method="POST">
<div class="form-group">
<label for="name">Full Name</label>
<input type="text" id="name" name="name" placeholder="Enter your full name" required class="form-control">
</div>
<div class="form-group">
<label for="email">Email</label>
<input type="email" id="email" name="email" placeholder="Enter a valid email address" required class="form-control">
</div>
<div class="form-group">
<label for="pass">Password</label>
<input type="password" id="pass" name="pass" placeholder="Enter your desired password" pattern=".{8,}" required title="Password must be more than 8 characters long" class="form-control">
</div>
<div class="form-group">
<label for="address">Address</label>
<input type="text" id="add" name="add" placeholder="Enter your current residence area" required class="form-control">
</div>
<div class="form-group">
<label for="number">Phone Number</label>
<input type="number" id="pno" name="pno" placeholder="Enter your current phone number" required class="form-control" onchange="validate1()">
</div>

<div class="form-group">
<label>
Entry Date:</label>
```

```
                                    <input type="date" id="date" name="date" placeholder="Your validity date.." required class="form-control" >
                                    </div>
                                    <button class="btn btn-success">Submit</button>
                            </form>
                    </div>
            </div>
            <script type="text/javascript">
                    function validate1()
                    {
                            var a= document.forms["user"]["pno"].value;
                            var patt = /^\d{10}$/;
                            if(!patt.test(a))
                            {
                                    alert("Phone number not valid.");
                            }
                    }


                    }
            </script>
</body>
</html>
```

Module 2 (Cleaning of Data)

Hide

In [2]:
```
df.describe()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02

Hide

```
# Good No Null Values!
df.isnull().sum().max()
```

```
0
```

Hide

```
df.columns
```

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
```

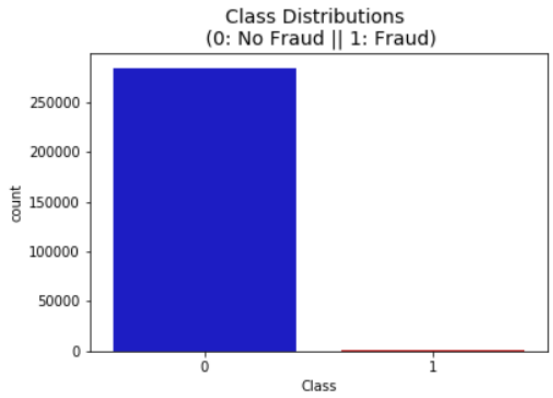
Hide

```
# The classes are heavily skewed we need to solve this issue later.
print('No Frauds', round(df['Class'].value_counts()[0]/len(df) * 100,2), '% of the dataset')
print('Frauds', round(df['Class'].value_counts()[1]/len(df) * 100,2), '% of the dataset')
```

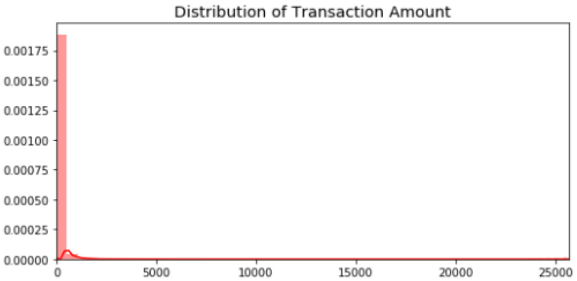
```
No Frauds 99.83 % of the dataset
Frauds 0.17 % of the dataset
```

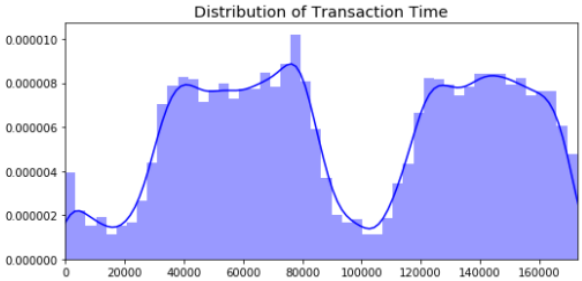
```
colors = ["#0101DF", "#DF0101"]

sns.countplot('Class', data=df, palette=colors)
plt.title('Class Distributions \n (0: No Fraud || 1: Fraud)', fontsize=14)
```

```
Text(0.5,1,'Class Distributions \n (0: No Fraud || 1: Fraud)')
```




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

amount_val = df['Amount'].values
time_val = df['Time'].values

sns.distplot(amount_val, ax=ax[0], color='r')
ax[0].set_title('Distribution of Transaction Amount', fontsize=14)
ax[0].set_xlim([min(amount_val), max(amount_val)])

sns.distplot(time_val, ax=ax[1], color='b')
ax[1].set_title('Distribution of Transaction Time', fontsize=14)
ax[1].set_xlim([min(time_val), max(time_val)])



plt.show()
```





Equally Distributing and Correlating

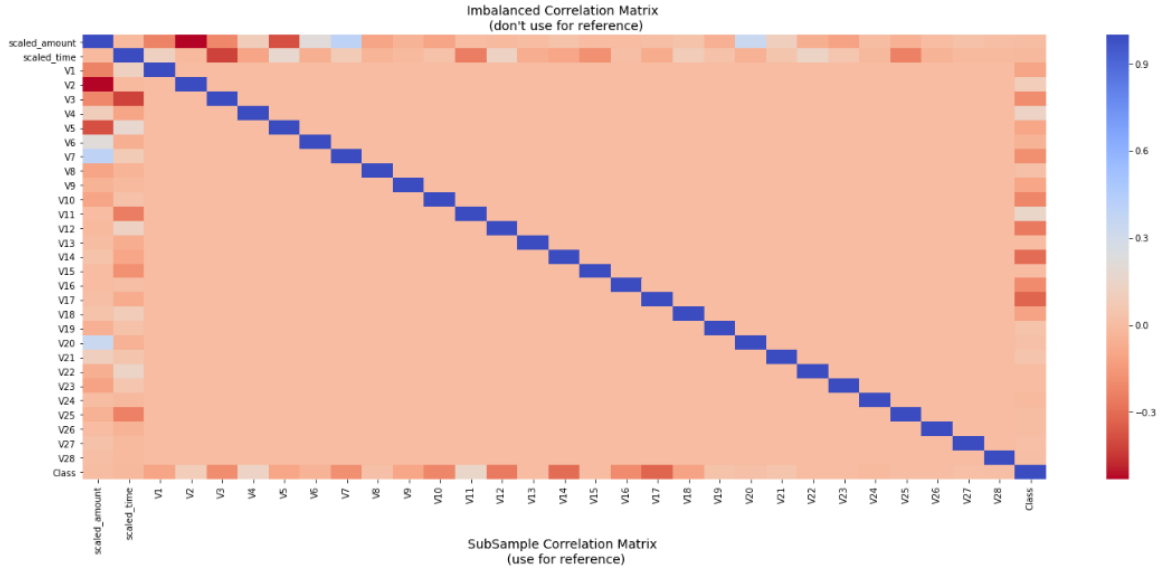
```
print('Distribution of the Classes in the subsample dataset')
print(new_df['Class'].value_counts()/len(new_df))



sns.countplot('Class', data=new_df, palette=colors)
plt.title('Equally Distributed Classes', fontsize=14)
plt.show()
```

```
Distribution of the Classes in the subsample dataset
1    0.5
0    0.5
Name: Class, dtype: float64
```

Correlation Matrices

Correlation matrices are the essence of understanding our data. We want to know if there are features that influence heavily in whether a specific transaction is a fraud. However, it is important that we use the correct dataframe (subsample) in order for us to see which features have a high positive or negative correlation with regards to fraud transactions.

```
f, (ax1, ax2) = plt.subplots(2, 1, figsize=(24,20))

# Entire DataFrame
corr = df.corr()
sns.heatmap(corr, cmap='coolwarm_r', annot_kws={'size':20}, ax=ax1)
ax1.set_title("Imbalanced Correlation Matrix \n (don't use for reference)", fontsize=14)


sub_sample_corr = new_df.corr()
sns.heatmap(sub_sample_corr, cmap='coolwarm_r', annot_kws={'size':20}, ax=ax2)
ax2.set_title('SubSample Correlation Matrix \n (use for reference)', fontsize=14)
plt.show()
```



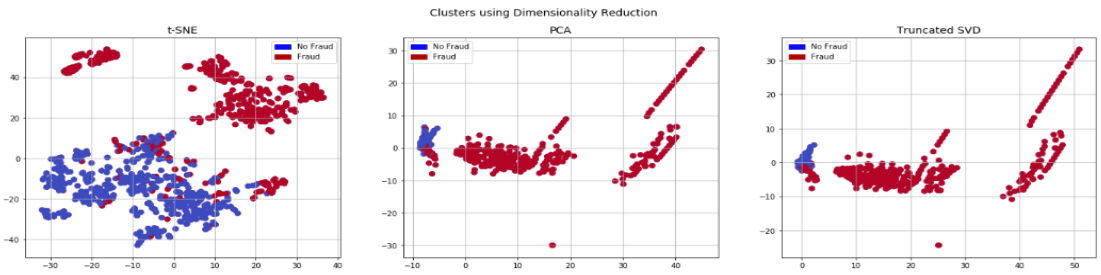
Module 3 **(**Dimensionality Reduction and Clustering)

```
# New_df is from the random undersample data (fewer instances)
X = new_df.drop('Class', axis=1)
y = new_df['Class']


# T-SNE Implementation
t0 = time.time()
X_reduced_tsne = TSNE(n_components=2, random_state=42).fit_transform(X.values)
t1 = time.time()
print("T-SNE took {:.2} s".format(t1 - t0))

# PCA Implementation
t0 = time.time()
X_reduced_pca = PCA(n_components=2, random_state=42).fit_transform(X.values)
t1 = time.time()
print("PCA took {:.2} s".format(t1 - t0))

# TruncatedSVD
t0 = time.time()
X_reduced_svd = TruncatedSVD(n_components=2, algorithm='randomized', random_state=42).fit_trans
form(X.values)
t1 = time.time()
print("Truncated SVD took {:.2} s".format(t1 - t0))
```

```
T-SNE took 6.4 s
PCA took 0.11 s
Truncated SVD took 0.13 s
```

Code

```
ax2.grid(True)

ax2.legend(handles=[blue_patch, red_patch])

# TruncatedSVD scatter plot
ax3.scatter(X_reduced_svd[:,0], X_reduced_svd[:,1], c=(y == 0), cmap='coolwarm', label='No Frau
d', linewidths=2)
ax3.scatter(X_reduced_svd[:,0], X_reduced_svd[:,1], c=(y == 1), cmap='coolwarm', label='Fraud',
linewidths=2)
ax3.set_title('Truncated SVD', fontsize=14)

ax3.grid(True)

ax3.legend(handles=[blue_patch, red_patch])

plt.show()
```



Module 4 (Training Model)

```
X = new_df.drop('Class', axis=1)
y = new_df['Class']
```

Hide

```
# Our data is already scaled we should split our training and test sets
from sklearn.model_selection import train_test_split

# This is explicitly used for undersampling.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Hide

```
# Turn the values into an array for feeding the classification algorithms.
X_train = X_train.values
X_test = X_test.values
y_train = y_train.values
y_test = y_test.values
```

Hide

```
# Let's implement simple classifiers

classifiers = {
    "LogisiticRegression": LogisticRegression(),
    "KNearest": KNeighborsClassifier(),
    "Support Vector Classifier": SVC(),
    "DecisionTreeClassifier": DecisionTreeClassifier()
}
```

```
# Wow our scores are getting even high scores even when applying cross validation.
from sklearn.model_selection import cross_val_score


for key, classifier in classifiers.items():
    classifier.fit(X_train, y_train)
    training_score = cross_val_score(classifier, X_train, y_train, cv=5)
    print("Classifiers: ", classifier.__class__.__name__, "Has a training score of", round(trai
ning_score.mean(), 2) * 100, "% accuracy score")
```

```
Classifiers:  LogisticRegression Has a training score of 94.0 % accuracy score
Classifiers:  KNeighborsClassifier Has a training score of 92.0 % accuracy score
Classifiers:  SVC Has a training score of 92.0 % accuracy score
Classifiers:  DecisionTreeClassifier Has a training score of 91.0 % accuracy score
```

Hide

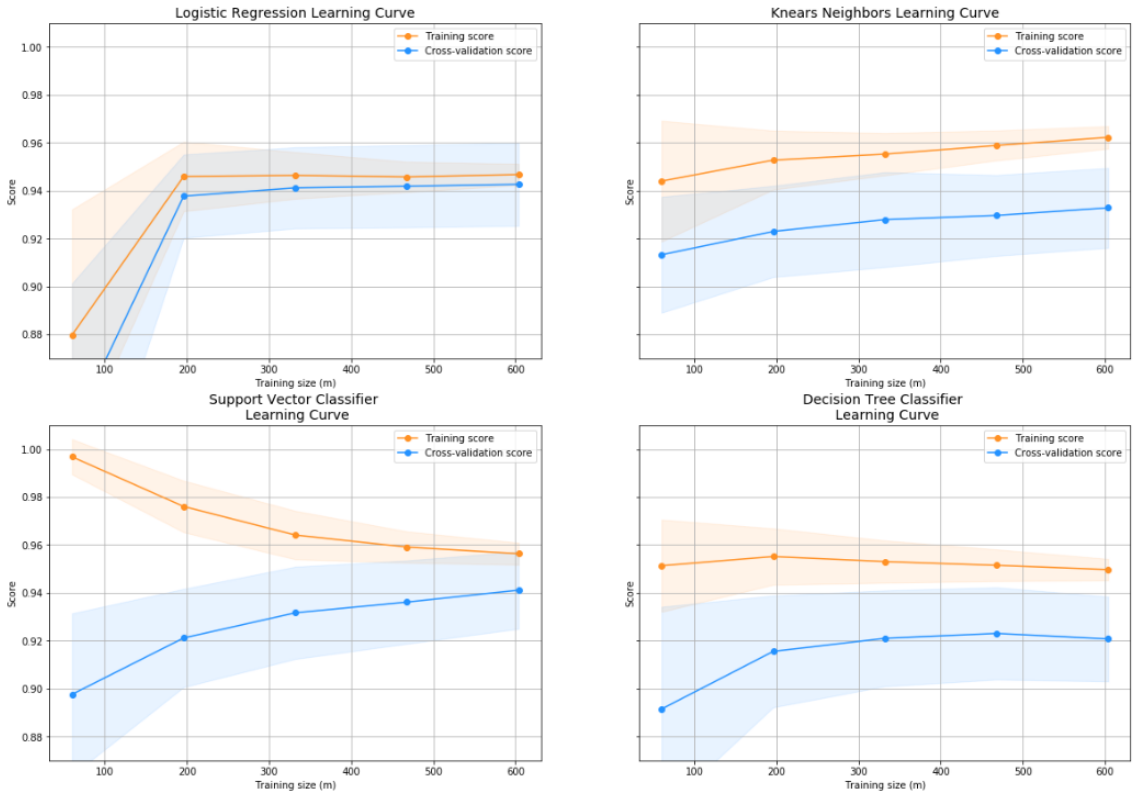
```
# Use GridSearchCV to find the best parameters.
from sklearn.model_selection import GridSearchCV


# Logistic Regression
log_reg_params = {"penalty": ['l1', 'l2'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}



grid_log_reg = GridSearchCV(LogisticRegression(), log_reg_params)
grid_log_reg.fit(X_train, y_train)
# We automatically get the logistic regression with the best parameters.
log_reg = grid_log_reg.best_estimator_

knears_params = {"n_neighbors": list(range(2,5,1)), 'algorithm': ['auto', 'ball_tree', 'kd_tre
e', 'brute']}
```

```
log_reg_score = cross_val_score(log_reg, X_train, y_train, cv=5)
print('Logistic Regression Cross Validation Score: ', round(log_reg_score.mean() * 100, 2).asty
pe(str) + '%')


knears_score = cross_val_score(knears_neighbors, X_train, y_train, cv=5)
print('Knears Neighbors Cross Validation Score', round(knears_score.mean() * 100, 2).astype(str
) + '%')

svc_score = cross_val_score(svc, X_train, y_train, cv=5)
print('Support Vector Classifier Cross Validation Score', round(svc_score.mean() * 100, 2).asty
pe(str) + '%')

tree_score = cross_val_score(tree_clf, X_train, y_train, cv=5)
print('DecisionTree Classifier Cross Validation Score', round(tree_score.mean() * 100, 2).astyp
e(str) + '%')
```

```
Logistic Regression Cross Validation Score:  94.31%
Knears Neighbors Cross Validation Score 92.46%
Support Vector Classifier Cross Validation Score 94.44%
DecisionTree Classifier Cross Validation Score 92.19%
```

```
cv = ShuffleSplit(n_splits=100, test_size=0.2, random_state=42)
plot_learning_curve(log_reg, knears_neighbors, svc, tree_clf, X_train, y_train, (0.87, 1.01), c
v=cv, n_jobs=4)
```

```
<module 'matplotlib.pyplot' from '/opt/conda/lib/python3.6/site-packages/matplotlib/pyplot.py'>
```



Module 5 (Testing Model)

1) Random Forest

Define model parameters

Let's set the parameters for the model.

Let's run a model using the training set for training. Then, we will use the validation set for validation.

We will use as validation criterion **GINI**, which formula is **GINI = 2 * (AUC) - 1**, where **AUC** is the **Receiver Operating Characteristic - Area Under Curve (ROC-AUC)**. Number of estimators is set to **100** and number of parallel jobs is set to **4**.

We start by initializing the RandomForestClassifier.

```
clf = RandomForestClassifier(n_jobs=NO_JOBS,
                             random_state=RANDOM_STATE,
                             criterion=RFC_METRIC,
                             n_estimators=NUM_ESTIMATORS,
                             verbose=False)
```

Let's train the **RandonForestClassifier** using the **train_df** data and **fit** function.

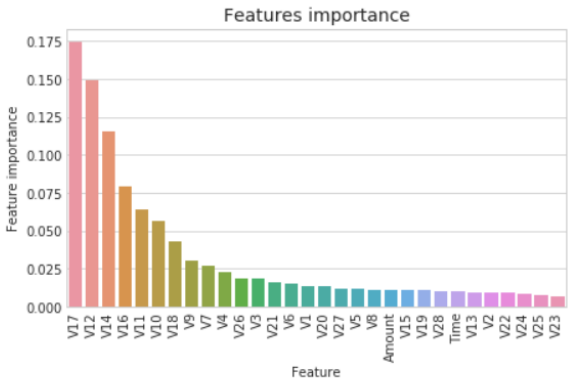
```
clf.fit(train_df[predictors], train_df[target].values)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=4,
            oob_score=False, random_state=2018, verbose=False,
            warm_start=False)
```

Let's now predict the **target** values for the **valid_df** data, using **predict** function.

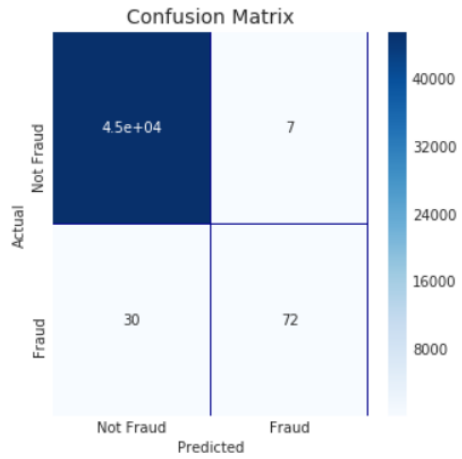
```
preds = clf.predict(valid_df[predictors])
```

2]:
```
tmp = pd.DataFrame({'Feature': predictors, 'Feature importance': clf.feature_importances_})
tmp = tmp.sort_values(by='Feature importance',ascending=False)
plt.figure(figsize = (7,4))
plt.title('Features importance',fontsize=14)
s = sns.barplot(x='Feature',y='Feature importance',data=tmp)
s.set_xticklabels(s.get_xticklabels(),rotation=90)
plt.show()
```



The most important features are **V17**, **V12**, **V14**, **V10**, **V11**, **V16**.

:
```
cm = pd.crosstab(valid_df[target].values, preds, rownames=['Actual'], colnames=['Predicted'])
fig, (ax1) = plt.subplots(ncols=1, figsize=(5,5))
sns.heatmap(cm,
            xticklabels=['Not Fraud', 'Fraud'],
            yticklabels=['Not Fraud', 'Fraud'],
            annot=True,ax=ax1,
            linewidths=.2,linecolor="Darkblue", cmap="Blues")
plt.title('Confusion Matrix', fontsize=14)
plt.show()
```



2) AdaBoost Classifier

```
clf = AdaBoostClassifier(random_state=RANDOM_STATE,
                         algorithm='SAMME.R',
                         learning_rate=0.8,
                             n_estimators=NUM_ESTIMATORS)
```

Fit the model

Let's fit the model.

```
clf.fit(train_df[predictors], train_df[target].values)
```

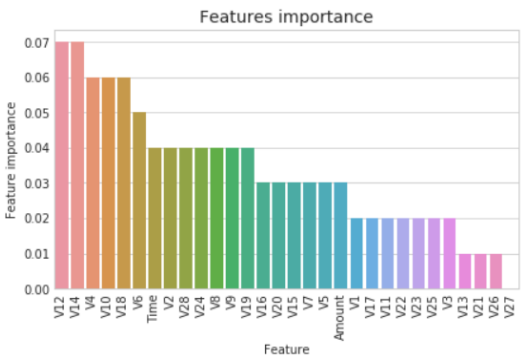
```
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=0.8, n_estimators=100, random_state=2018)
```

Predict the target values

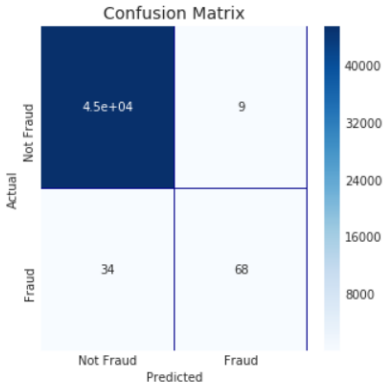
Let's now predict the **target** values for the **valid_df** data, using predict function.

```
preds = clf.predict(valid_df[predictors])
```

In [28]:
```
tmp = pd.DataFrame({'Feature': predictors, 'Feature importance': clf.feature_importances_})
tmp = tmp.sort_values(by='Feature importance',ascending=False)
plt.figure(figsize = (7,4))
plt.title('Features importance',fontsize=14)
s = sns.barplot(x='Feature',y='Feature importance',data=tmp)
s.set_xticklabels(s.get_xticklabels(),rotation=90)
plt.show()
```



In [29]:
```
cm = pd.crosstab(valid_df[target].values, preds, rownames=['Actual'], colnames=['Predicted'])
fig, (ax1) = plt.subplots(ncols=1, figsize=(5,5))
sns.heatmap(cm,
            xticklabels=['Not Fraud', 'Fraud'],
            yticklabels=['Not Fraud', 'Fraud'],
            annot=True,ax=ax1,
            linewidths=.2,linecolor="Darkblue", cmap="Blues")
plt.title('Confusion Matrix', fontsize=14)
plt.show()
```



3) XGBoost

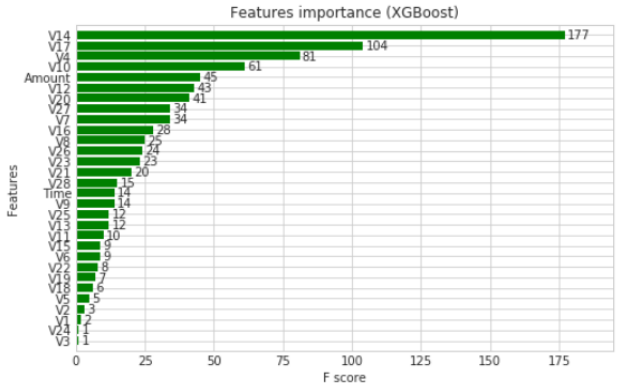
```
# Prepare the train and valid datasets
dtrain = xgb.DMatrix(train_df[predictors], train_df[target].values)
dvalid = xgb.DMatrix(valid_df[predictors], valid_df[target].values)
dtest = xgb.DMatrix(test_df[predictors], test_df[target].values)

#What to monitor (in this case, **train** and **valid**)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

# Set xgboost parameters
params = {}
params['objective'] = 'binary:logistic'
params['eta'] = 0.039
params['silent'] = True
params['max_depth'] = 2
params['subsample'] = 0.8
params['colsample_bytree'] = 0.9
params['eval_metric'] = 'auc'
params['random_state'] = RANDOM_STATE
```

```
model = xgb.train(params,
                  dtrain,
                  MAX_ROUNDS,
                  watchlist,
                  early_stopping_rounds=EARLY_STOP,
                  maximize=True,
                  verbose_eval=VERBOSE_EVAL)
```

```
[0]     train-auc:0.887798      valid-auc:0.85275
Multiple eval metrics have been passed: 'valid-auc' will be used for early stopping.

Will train until valid-auc hasn't improved in 50 rounds.
[50]    train-auc:0.939463      valid-auc:0.882001
[100]   train-auc:0.952123      valid-auc:0.88984
[150]   train-auc:0.977384      valid-auc:0.964654
[200]   train-auc:0.989424      valid-auc:0.981679
[250]   train-auc:0.994246      valid-auc:0.984177
Stopping. Best iteration:
[241]   train-auc:0.993507      valid-auc:0.984726
```

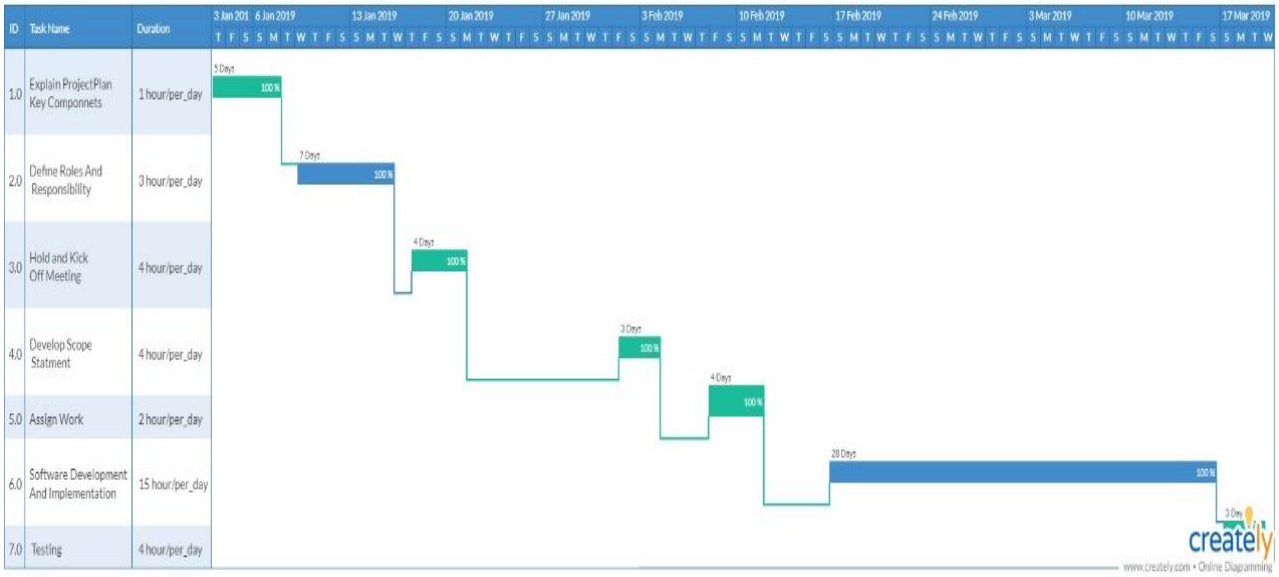
[39]:

```
fig, (ax) = plt.subplots(ncols=1, figsize=(8,5))
xgb.plot_importance(model, height=0.8, title="Features importance (XGBoost)", ax=ax, color="gre
en")
plt.show()
```



V) Schedule, Tasks & Milestones

Schedule



Task

- Project planning
- Defining roles
- Deciding scope
- Getting and analysing the dataset for fraud detection
- Generating models
- Making dataset work according to the models
- Applying algorithm
- Getting desired results
- Making a front end and connecting it to the back end
- Testing

Milestones

Milestone	Milestone Goal
Concept approval	Feasibility studies and basic system concepts have been approved by management and the project is authorized to proceed to detailed requirements definition.
Requirements review	Requirements specifications are complete, correct, approved and suitable for input to design.
Preliminary design review	The architectural design satisfies all product requirements, is approved and is suitable for input into the detailed design process.
Critical design review	Detailed designs fully implement the system architecture, are approved and are suitable for input into the development of code.
Test plan review	Test plans are adequate for the testing of all product features, are approved and are suitable for input to the development of test cases and test procedures.
Test readiness review	Developed and unit tested software has been passed by the test team and is suitable for input into integration testing.
System test review	The software product has passed system testing and is suitable for input into acceptance testing.
Operational readiness review	The software product has passed acceptance testing and is suitable for deployment in its target production environment.
Product operational	The software is in use in its target operational environment.

VI) Project Demonstration

Module-1:

```
<div class="container">

    <h1 class="my-4">Welcome to Credit Card Fraud Detetction System</h1>

    <!-- Marketing Icons Section -->
    <div class="row">
      <div class="col-lg-4 mb-4">
        <div class="card">
          <h4 class="card-header">Detecting Fruadulent Activities in Transaction</h4>


    </div>



  </div>
        </div>
        </div>
```

2)Registration Page

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Credit Card Fraud Detection System| Register user</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
    <link href="css/modern-business.css" rel="stylesheet">
    <style>
    .navbar-toggler {
        z-index: 1;
    }

    @media (max-width: 576px) {
        nav > .container {
            width: 100%;
        }
    }
    </style>
        <style>
    .errorWrap {
    padding: 10px;
    margin: 0 0 20px 0;
    background: #fff;
    border-left: 4px solid #dd3d36;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
.succWrap{
    padding: 10px;
```

```
    margin: 0 0 20px 0;
    background: #fff;
    border-left: 4px solid #5cb85c;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
    </style>



</head>

<body>

<?php include('includes/header.php');?>

    <!-- Page Content -->
    <div class="container">

        <!-- Page Heading/Breadcrumbs -->
        <h1 class="mt-4 mb-3">Get yourself registered</small></h1>

        <ol class="breadcrumb">
          <li class="breadcrumb-item">
            <a href="index.php">Home</a>
          </li>
          <li class="breadcrumb-item active">Registration</li>
        </ol>
          <?php if($error){?><div class="errorWrap"><strong>ERROR</strong>:<?php echo
htmlentities($error); ?> </div><?php }
      else if($msg){?><div class="succWrap"><strong>SUCCESS</strong>:<?php echo
htmlentities($msg); ?> </div><?php }?>
        <!-- Content Row -->
```

```
<form name="donar" method="post">
<div class="row">
<div class="col-lg-4 mb-4">
<div class="font-italic">Full Name<span style="color:red">*</span></div>
<div><input type="text" name="fullname" class="form-control" required></div>
</div>
<div class="col-lg-4 mb-4">
<div class="font-italic">Mobile Number<span style="color:red">*</span></div>
<div><input type="text" name="mobileno" class="form-control" required></div>
</div>
<div class="col-lg-4 mb-4">
<div class="font-italic">Email Id</div>
<div><input type="email" name="emailid" class="form-control"></div>
</div>
</div>

<div class="row">
<div class="col-lg-4 mb-4">
<div class="font-italic">Age<span style="color:red">*</span></div>
<div><input type="text" name="age" class="form-control" required></div>
</div>


<div class="col-lg-4 mb-4">
<div class="font-italic">Gender<span style="color:red">*</span></div>
<div><select name="gender" class="form-control" required>
<option value="">Select</option>
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
</div>
```

```
</div>



<?php $sql = "SELECT * from  tbluser ";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{            ?>
<option value="<?php echo htmlentities($result->BloodGroup);?>"><?php echo htmlentities($result->BloodGroup);?></option>
<?php }} ?>
</select>
</div>
</div>



<div class="row">
<div class="col-lg-4 mb-4">
<div class="font-italic">Address</div>
<div><textarea class="form-control" name="address"></textarea></div>
</div>

<div class="col-lg-8 mb-4">
<div class="font-italic">Message<span style="color:red">*</span></div>
<div><textarea class="form-control" name="message" required> </textarea></div>
</div>
</div>
```

```
<div class="row">
<div class="col-lg-4 mb-4">
<div><input type="submit" name="submit" class="btn btn-primary" value="submit"
style="cursor:pointer"></div>
</div>




</div>
```

3) Login Page

```
<!doctype html>
<html lang="en" class="no-js">

<head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1,
maximum-scale=1">
        <meta name="description" content="">
        <meta name="author" content="">

        <title>Credit Card Fraud Detection System | Admin Login</title>
        <link rel="stylesheet" href="css/font-awesome.min.css">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">
        <link rel="stylesheet" href="css/bootstrap-social.css">
        <link rel="stylesheet" href="css/bootstrap-select.css">
        <link rel="stylesheet" href="css/fileinput.min.css">
        <link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">
```

```
        <link rel="stylesheet" href="css/style.css">
</head>

<body>

        <div class="login-page bk-img" style="background-image: url(img/banner.jpeg);">
                <div class="form-content">
                        <div class="container">
                                <div class="row">
                                        <div class="col-md-6 col-md-offset-3">
                                                <h1 class="text-center text-bold text-light mt-
4x">Credit Card Fraud Detection System Sign in</h1>
                                                <div class="well row pt-2x pb-3x bk-light">
                                                        <div class="col-md-8 col-md-offset-2">
                                                                <form method="post">

                                                                        <label for="" class="text-
uppercase text-sm">Your Username </label>
                                                                        <input type="text"
placeholder="Username" name="username" class="form-control mb">

                                                                        <label for="" class="text-
uppercase text-sm">Password</label>
                                                                        <input type="password"
placeholder="Password" name="password" class="form-control mb">

                                                                        <button class="btn btn-
primary btn-block" name="login" type="submit">LOGIN</button>

                                                                </form>
                                                        </div>
```

```
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
```

4) Upload CSV file

```
<!doctype html>
<html lang="en" class="no-js">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1,
maximum-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <meta name="theme-color" content="#3e454c">

    <title>CFND| Admin User</title>

    <!-- Font awesome -->
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <!-- Sandstone Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <!-- Bootstrap Datatables -->
    <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">
    <!-- Bootstrap social button library -->
```

```
        <link rel="stylesheet" href="css/bootstrap-social.css">
        <!-- Bootstrap select -->
        <link rel="stylesheet" href="css/bootstrap-select.css">
        <!-- Bootstrap file input -->
        <link rel="stylesheet" href="css/fileinput.min.css">
        <!-- Awesome Bootstrap checkbox -->
        <link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">
        <!-- Admin Stye -->
        <link rel="stylesheet" href="css/style.css">
<style>
                .errorWrap {
    padding: 10px;
    margin: 0 0 20px 0;
    background: #fff;
    border-left: 4px solid #dd3d36;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
.succWrap{
    padding: 10px;
    margin: 0 0 20px 0;
    background: #fff;
    border-left: 4px solid #5cb85c;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
                </style>
<script language="javascript">
function isNumberKey(evt)
    {
```

```
      var charCode = (evt.which) ? evt.which : event.keyCode


      if (charCode > 31 && (charCode < 48 || charCode > 57) && charCode!=46)
        return false;


       return true;
    }
    </script>
</head>


<body>
            <?php include('includes/header.php');?>
            <div class="ts-main-content">
            <?php include('includes/leftbar.php');?>
                        <div class="content-wrapper">
                                    <div class="container-fluid">


                                                <div class="row">
                                                            <div class="col-md-12">


                                                                        <h2 class="page-title">Add User</h2>


                                                                        <div class="row">
                                                                                    <div class="col-md-12">
                                                                                                <div class="panel panel-default">
                                                                                                            <div class="panel-
heading">Basic Info</div>
<?php if($error){?><div class="errorWrap"><strong>ERROR</strong>:<?php echo
htmlentities($error); ?> </div><?php }

                                                                                    else if($msg){?><div
class="succWrap"><strong>SUCCESS</strong>:<?php echo htmlentities($msg); ?> </div><?php }?>
```

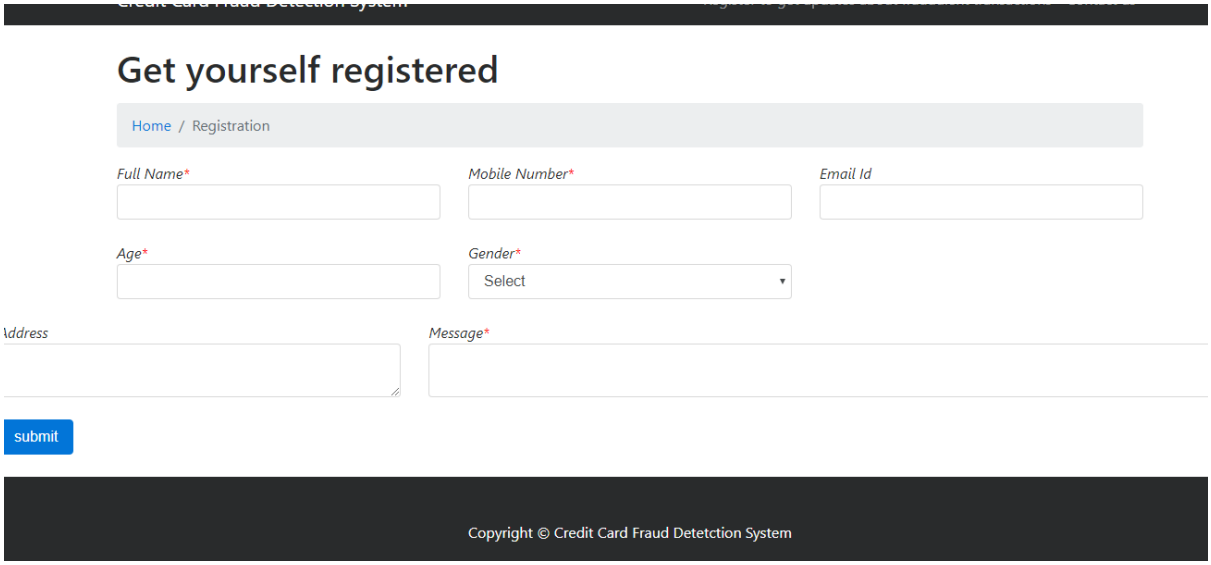
```
                                                    <div class="panel-body">
<form method="post" class="form-horizontal" enctype="multipart/form-data">
<div class="form-group">
<label class="col-sm-2 control-label">Full Name<span style="color:red">*</span></label>
<div class="col-sm-4">
<input type="text" name="fullname" class="form-control" required>
</div>
<label class="col-sm-2 control-label">Mobile No<span style="color:red">*</span></label>
<div class="col-sm-4">
<input type="text" name="mobileno" onKeyPress="return isNumberKey(event)"  maxlength="10" class="form-control" required>
</div>
</div>

<div class="form-group">
<label class="col-sm-2 control-label">Email id </label>
<div class="col-sm-4">
<input type="email" name="emailid" class="form-control">
</div>
<label class="col-sm-2 control-label">Age<span style="color:red">*</span></label>
<div class="col-sm-4">
<input type="text" name="age" class="form-control" required>
</div>
</div>

<div class="form-group">
<label class="col-sm-2 control-label">Gender <span style="color:red">*</span></label>
<div class="col-sm-4">
<select name="gender" class="form-control" required>
<option value="">Select</option>
<option value="Male">Male</option>
```

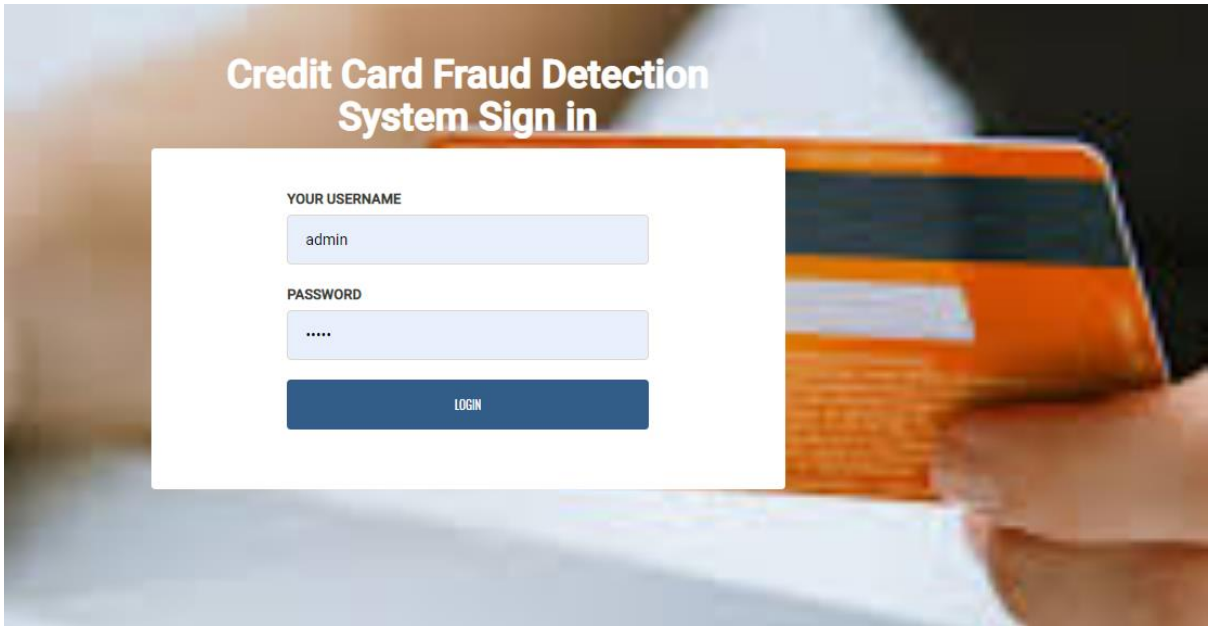
```
<option value="Female">Female</option>
</select>
</div>

</div>



<div class="hr-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label">Address</label>
<div class="col-sm-10">
<textarea class="form-control" name="address" ></textarea>
</div>
</div>

<label class="col-sm-2 control-label">Upload CSV File <span style="color:red">*</span></label>
<table width="600">
<form action="<?php echo $_SERVER["PHP_SELF"]; ?>" method="post" enctype="multipart/form-data">

<tr>
<td width="20%">Select file</td>
<td width="80%"><input type="file" name="file" id="file" /></td>
</tr>

<tr>
<td>Submit</td>
<td><input type="submit" name="submit" /></td>
</tr>
```

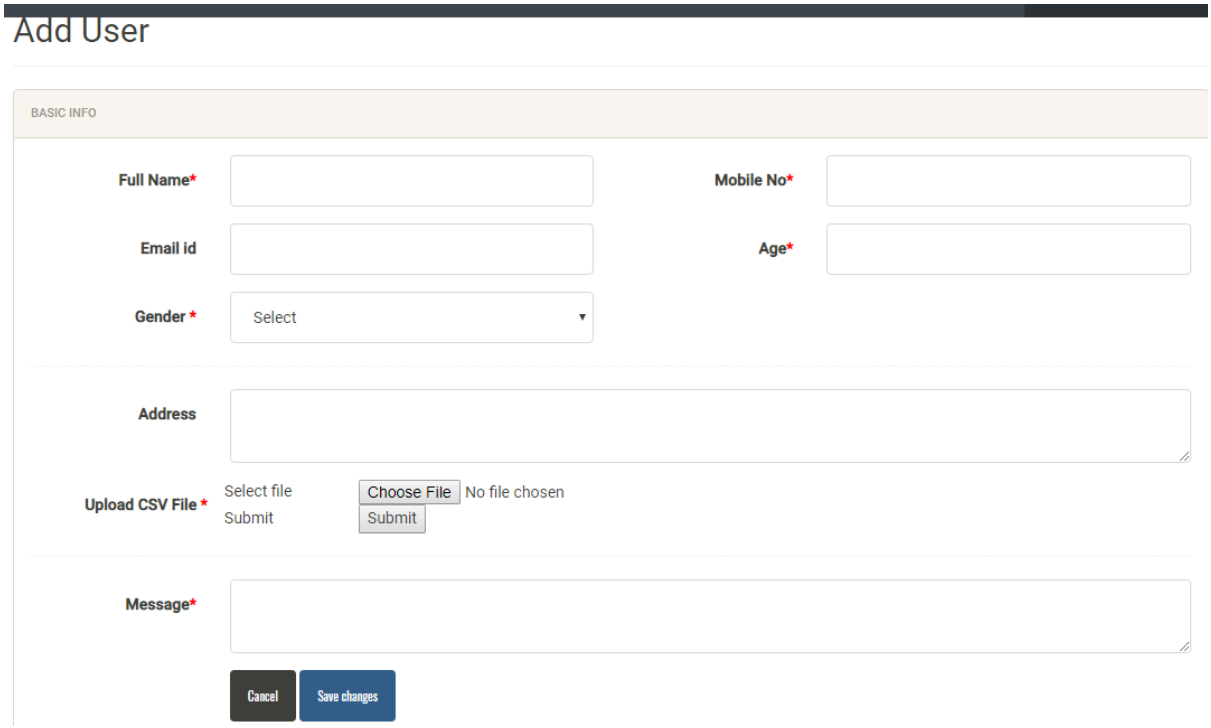
```
</form>
</table>

<?php
if ( isset($_POST["submit"]) ) {

  if ( isset($_FILES["file"])) {

        //if there was an error uploading the file
      if ($_FILES["file"]["error"] > 0) {
        echo "Return Code: " . $_FILES["file"]["error"] . "<br />";

      }
      else {
            //Print file details
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

            //if file already exists
        if (file_exists("upload/" . $_FILES["file"]["name"])) {
       echo $_FILES["file"]["name"] . " already exists. ";
        }
        else {
             //Store file in directory "upload" with the name of "uploaded_file.txt"
       $storagename = "uploaded_file.txt";
       move_uploaded_file($_FILES["file"]["tmp_name"], "upload/" . $storagename);
       echo "Stored in: " . "upload/" . $_FILES["file"]["name"] . "<br />";
       }
      }
```

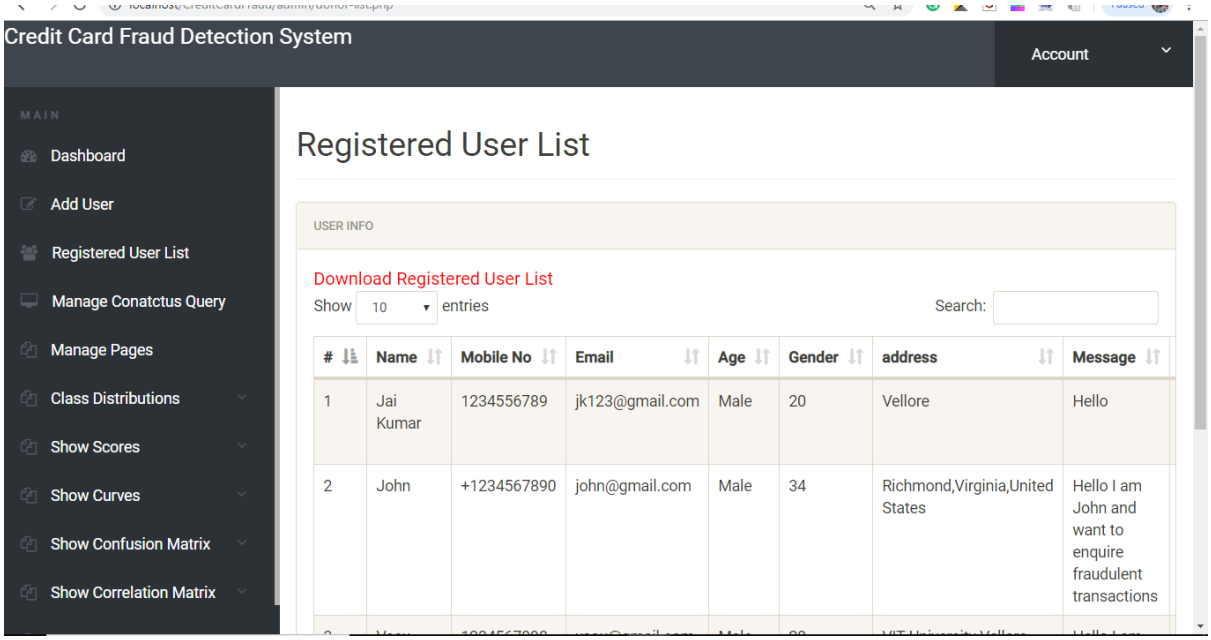
```
    } else {
        echo "No file selected <br />";
    }
}
?>
<div class="hr-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label">Message<span style="color:red">*</span></label>
<div class="col-sm-10">
<textarea class="form-control" name="message" required> </textarea>
</div>
</div>




                                                                    <div
class="form-group">
                                                                        <div
class="col-sm-8 col-sm-offset-2">
        <button class="btn btn-default" type="reset">Cancel</button>
        <button class="btn btn-primary" name="submit" type="submit">Save changes</button>
        </div>
                                                                    </div>

                                                                </form>
                                                            </div>
                                                        </div>
                                                    </div>
```

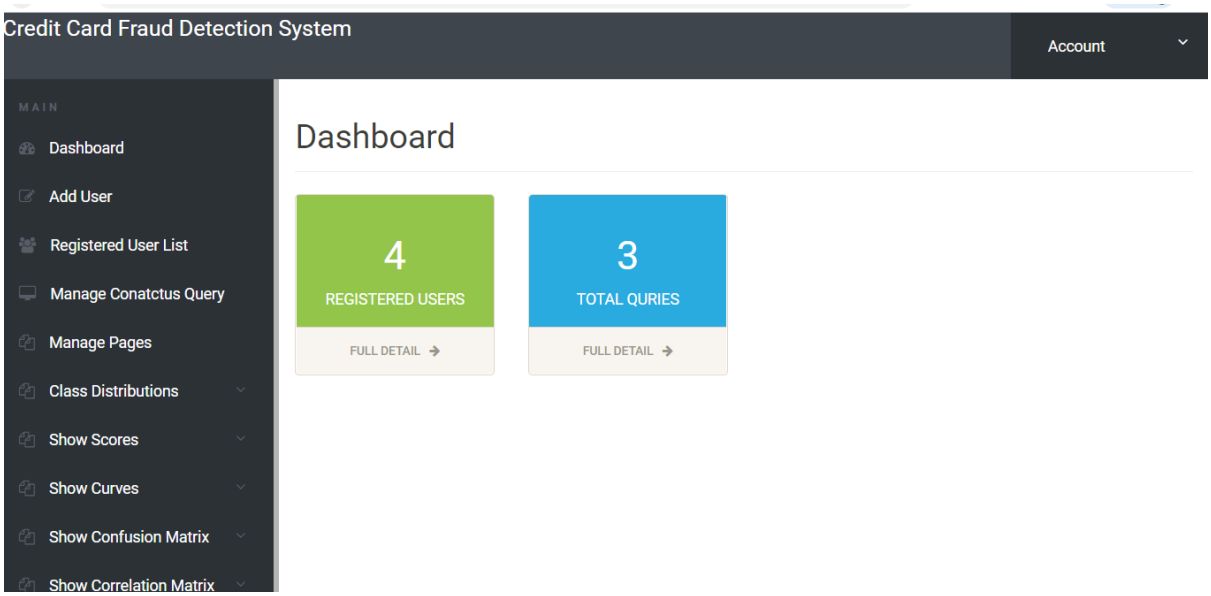
```
                                    </div>
                                </div>


                            </div>
                        </div>
                    </div>
```

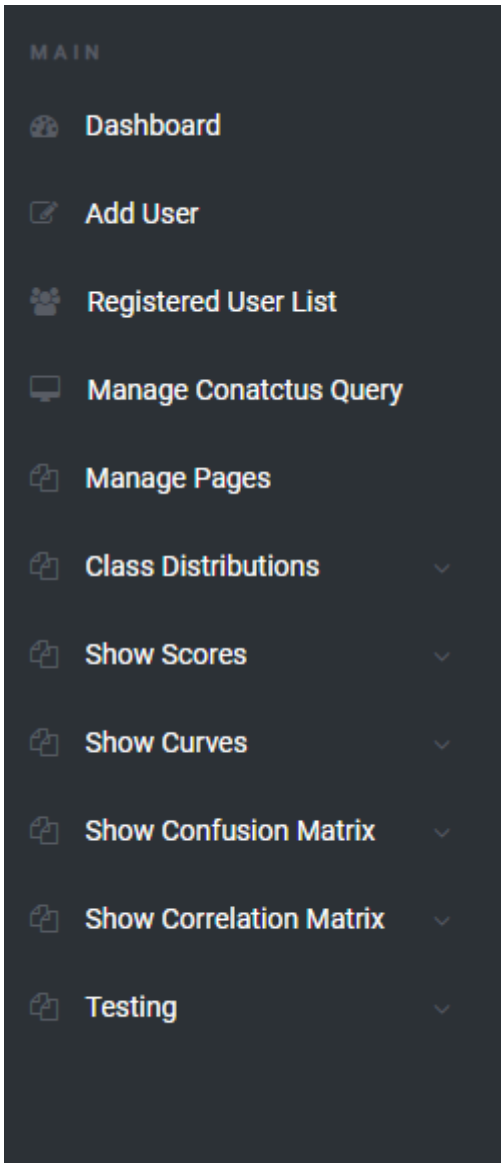

Get yourself registered
Home / Registration
Full Name*
Mobile Number*
Email Id
Age*
Gender*
Select
ddress
Message*
submit
Copyright © Credit Card Fraud Detetction System


Credit Card Fraud Detection System Sign in
YOUR USERNAME
admin
PASSWORD
LOGIN


Add User
BASIC INFO
Full Name*
Mobile No*
Email id
Age*
Gender *
Select
Address
Upload CSV File *
Select file
Choose File
No file chosen
Submit
Submit
Message*
Cancel
Save changes


Credit Card Fraud Detection System
Account
MAIN
Dashboard
Add User
Registered User List
Manage Conatctus Query
Manage Pages
Class Distributions
Show Scores
Show Curves
Show Confusion Matrix
Show Correlation Matrix
Registered User List
USER INFO
Download Registered User List
Show 10 entries
Search:
Name Mobile No Email Age Gender address Message
1 Jai Kumar 1234556789 jk123@gmail.com Male 20 Vellore Hello
2 John +1234567890 john@gmail.com Male 34 Richmond,Virginia,United States Hello I am John and want to enquire fraudulent transactions


Credit Card Fraud Detection System
Account
MAIN
Dashboard
Add User
Registered User List
Manage Conatctus Query
Manage Pages
Class Distributions
Show Scores
Show Curves
Show Confusion Matrix
Show Correlation Matrix
Dashboard
4
REGISTERED USERS
FULL DETAIL
3
TOTAL QURIES
FULL DETAIL


MAIN
Dashboard
Add User
Registered User List
Manage Conatctus Query
Manage Pages
Class Distributions
Show Scores
Show Curves
Show Confusion Matrix
Show Correlation Matrix
Testing

VII) Cost Analysis & Results and Discussion

The key features of the proposed system and the requirement specifications of the proposed system are discussed below.

A. Existing System The Traditional detection method mainly depends on database system and the education of customers, which usually are delayed, inaccurate and not in-time. After that methods based on discrimate analysis and regression analysis are widely used which can detect fraud by credit rate for cardholders and credit card transaction. For a large amount of data it is not efficient.

B. Problem Recognition The high amount of losses due to fraud and the awareness of the relation between loss and the available limit has to be reduced. The fraud has to be deducted in real time and the number of false alert has to be minimized.

C. Proposed System The proposed system overcomes the above mentioned issue in an efficient way. Using genetic algorithm the fraud is detected and thefalse alert is minimized and it produces an optimized result. The fraud is detected based on the customers behavior. A new classification problem which has a variable misclassification cost is introduced. Here the genetic algorithms is made where a set of interval valued parameters are optimized.

Fraud detection is a complex issue that requires a substantial amount of planning before throwing machine learning algorithms at it. Nonetheless, it is also an application of data science and machine learning for the good, which makes sure that the customer's money is safe and not easily tampered with.

Future work will include a comprehensive tuning of the Random Forest algorithm Having a data set with non-anonymized features would make this particularly interesting as outputting the feature importance would enable one to see what specific factors are most important for detecting fraudulent transactions. To be able to test the performance of our algorithms, I first performed an 80/20 train-test split, splitting our balanced data set into two pieces. To avoid overfitting a very common resampling technique of k-fold cross-validation. This simply means that separate your training data into k parts (folds) and then fit your model on k-1 folds before making predictions for the kth hold-out fold. You then repeat this process for every single fold and average the resulting predictions. Owing to such imbalance in data, an algorithm that does not do any feature analysis and predicts all the transactions as non-frauds will also achieve an accuracy of 99.828%. Therefore, accuracy is not a correct measure of efficiency in our case. We need some other standard of correctness while classifying transactions as fraud or non-fraud.

Hence we provide the customer with many different models from which they can choose from to best fit their requirement of what fraud constitutes for them like

We started with **RandomForrestClassifier**, for which we obtained an AUC scode of **0.85** when predicting the target for the test set.

We followed with an **AdaBoostClassifier** model, with lower AUC score (**0.83**) for prediction of the test set target values.

We then experimented with a **XGBoost** model. In this case, se used the validation set for validation of the training model. The best validation score obtained was **0.984**. Then we used the model with the best training step, to predict target value from the test data; the AUC score obtained was **0.974**.

VIII) Summary

Credit is a method of selling goods or services without buyer having cash in mind. A credit card is only an automatic way to offer credit to the consumer. Today every credit cards carries an identifying number that speeds shopping transactions. In credit card business fraud occur when lender is fooled by the borrower offering him/her purchase believing that borrower credit card account will provide payment for this purchase. Ideally no payment will be made. If the payment is made the credit card issuer will reclaim the amount paid. Today with expansion of e-commerce it is on the internet that half of the fraud is conducted. Fraudsters have usually connections with the affected business. Neural networks are also recommended as effective credit card fraud detection methods. The only issue with this method is that all data has to be clustered by the type of account it belongs to. Credit card fraud is a major issue that if not dealt with effectively, it can result in myriad complications. It is vital to try and find ways of detecting the issues and resolving them as soon as they arise. A deviation from such a pattern will then lead to the system classifying it as suspicious transaction and will then be handled accordingly. The proposed system does a job of classifying the transactions into one of the two types: 1. Suspicious Transaction 2. Non-Suspicious Transaction The classification of transaction is done using the following algorithm as depicted in figure 1. The system checks the location and the pattern of spending as the major parameters to decide a spurious transaction. If there is a mismatch in the location or the pattern, the system marks it as suspicious and subjects the transaction through a verification process. The verification could be any process such as alerting the user, calling the user or subject the user to go through another round of clearance such as OTP. If the transaction is valid, then the details are updated for the user. After verification if the system fails to identify the user, the transaction is declined. Since the accuracy depends on location and pattern, it is possible to get false alarms and a valid user may be subjected through verification process. It is also possible that the system could decline a valid transaction.

IX) References

https://www.educba.com/fraud-detection-analytics/

http://www.academia.edu/7379999/Real

Time_Credit_Application_Fraud_Detection_System_Based_On_Data_Mining

https://www.researchgate.net/publication/309638452_Credit_Card_Fraud_Detection_using_Big_Data_Analytics_Use_of_PSOAANN_based_One-Class_Classification