**SOFTWARE REQUIREMENTS SPECIFICATION**

**An Experimental Study with Imbalanced Classification**

**Approaches for Credit Card Fraud Detection**

**Team Members:**1. Suraj Raj

2. Pradeep Nandagiri

3. John Wesley

4. Pavan Sai Pamujula

**1.0 Introduction**

Now a day the usage of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In this paper, we model the sequence of operations in credit card transaction processing using a Decision tree and Deep Neural Network show how it can be used for the detection of frauds. An both algorithms is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained with sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions. We present detailed experimental results to show the effectiveness of our approach and compare it with other techniques available in the literature.

* 1. **Goals and objectives**

Our project, mainly focused on credit card fraud detection for in real world. Initially I will collect the credit card datasets for trained dataset. Then will provide the user credit card queries for testing data set. After classification process of random forest algorithm using to the already analyzing data set and user provide current dataset. Finally optimizing the accuracy of the result data. Then will apply the processing of some of the attributes provided can find affected fraud detection in viewing the graphical model visualization. The performance of the techniques is evaluated based on accuracy, sensitivity, and specificity, precision. The results indicate about the optimal accuracy for Decision tree are 98.6% respectively.

* 1. **Statement of scope**

In this proposed project we designed a protocol or a model to detect the fraud activity in credit card transactions.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 behavior and pattern of fraudulent transactions. With the rise of machine learning, artificial intelligence, and other relevant fields of information technology, it becomes feasible to automate this process and to save some of the intensive amount of labour that is put into detecting credit card fraud.

* 1. **Software context**

**ANACONDA NAVIGATOR**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

## Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator

**PYTHON**

Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and scientific applications.   
It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.  **Features of Python**   
A simple language which is easier to learn, Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

* **Free and open source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

* **Portability**  
  You can move Python programs from one platform to another, and run it without any changes.   
  It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.
* **Extensible and Embeddable**   
  Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.
* **Large standard libraries to solve common tasks**   
  Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server You can use MySQLdb library using import MySQL db Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.
* **Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating object

**1.4 Major constraints**

we are applying random forest algorithm for classify the credit card dataset. Decision tree is an algorithm for classification and regression. Summarily, it is a collection of decision tree classifiers. Decision tree has advantage over decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature selected from a random subset of the full feature set. Even for large data sets with many features and data instances training is extremely fast in random forest and because each tree is trained independently of the others. The Decision tree algorithm has been found to provide a good estimate of the generalization error and to be resistant to overfitting.

**2.0 Usage scenario**

This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.

**2.1 User profiles**

A **user profile** is a collection of settings and information associated with a user. It contains critical information that is used to identify an individual, such as their name, age, portrait photograph and individual characteristics such as knowledge or expertise

**2.2 Use-cases**

**Diagram

Description automatically generated**

**3.0 Data Model and Description**

This section describes information domain for the software

**3.1 Data Description**

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

**3.1.1 Data objects**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

* Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
* Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.
* Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

**3.1.2 Relationships**

Diagram

Description automatically generated

**3.1.3 Complete data model**

LEVEL 0

LEVEL 1

LEVEL 2

**3.1.4 Data dictionary**

A reference to the data dictionary is provided. The dictionary is maintained in electronic form.

**4.0 Functional Model and Description**

**4.1.** **Description of Major Functions**

There are two categories of supervised learning:

|  |  |  |
| --- | --- | --- |
| **Algorithm Name** | **Description** | **Type** |
| **Linear regression** | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| **Logistic regression** | Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) | Classification |
| **Decision tree** | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made | Regression Classification |
| **Naive Bayes** | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| **Support vector machine** | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| **Random forest** | The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Regression Classification |
| **AdaBoost** | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| **Gradient-boosting trees** | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

**4.1.1 Requirement 1**

* Python
* Anaconda Navigator
* Python built-in modules
* Numpy
* Pandas
* Matplotlib
* Sklearn
* Seaborm

**4.2 Software Interface Description**

**MACHINE LEARNING**

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

**Machine Learning vs. Traditional Programming**

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

**COMPUTER**

**DATA RULES**

**OUTPUT**

**Machine Learning**

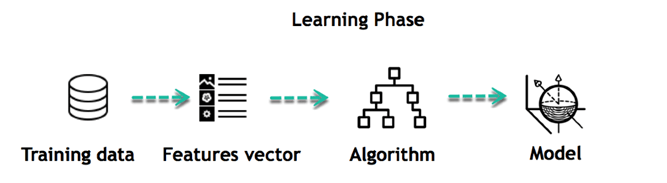
**4.2.1 External machine interfaces**

## How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the learningand inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a feature vector**.** You can think of a feature vector as a subset of data that is used to tackle a problem.

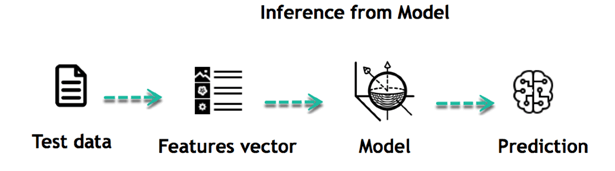
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

#### Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

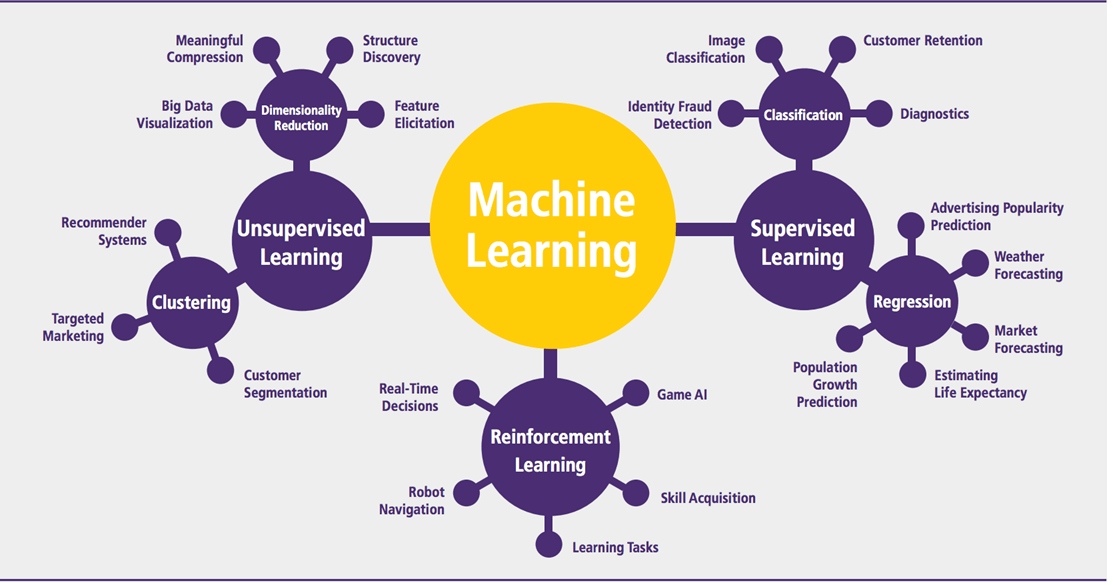


The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

## Machine learning Algorithms and where they are used?



Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

**4.2.2 External system interfaces**.

**FLASK**

**Flask** is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries. It has no [database](https://en.wikipedia.org/wiki/Database) abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

**Features:-**

Development server and [debugger](https://en.wikipedia.org/wiki/Debugger)

Integrated support for [unit testing](https://en.wikipedia.org/wiki/Unit_testing)

[RESTful](https://en.wikipedia.org/wiki/Representational_state_transfer) request dispatching

Uses Jinja templating

Support for secure cookies (client side sessions)

100% [WSGI](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface) 1.0 compliant

[Unicode](https://en.wikipedia.org/wiki/Unicode)-based

Complete documentation

[Google App Engine](https://en.wikipedia.org/wiki/Google_App_Engine) compatibility

Extensions available to extend functionality

**Example:**

The following code shows a simple web application that displays "[Hello World](https://en.wikipedia.org/wiki/Hello_World)!" when visited:

**from** **flask** **import** Flaskapp = Flask(\_\_name\_\_)

@app.route("/")**def** hello() -> str:

**return** "Hello World"

**if** \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=**False**)

**Templates** are files that contain static data as well as placeholders for dynamic data. A template is rendered with specific data to produce a final document. Flask uses the [Jinja](https://jinja.palletsprojects.com/templates/) template library to render templates.

In your application, you will use templates to render [HTML](https://developer.mozilla.org/docs/Web/HTML) which will display in the user’s browser. In Flask, Jinja is configured to autoescape any data that is rendered in HTML templates. This means that it’s safe to render user input; any characters they’ve entered that could mess with the HTML, such as < and > will be escaped with safe values that look the same in the browser but don’t cause unwanted effects.

Jinja looks and behaves mostly like Python. Special delimiters are used to distinguish Jinja syntax from the static data in the template. Anything between {{ and }} is an expression that will be output to the final document. {% and %} denotes a control flow statement like if and for. Unlike Python, blocks are denoted by start and end tags rather than indentation since static text within a block could change indentation.

Each page in the application will have the same basic layout around a different body. Instead of writing the entire HTML structure in each template, each template will extend a base template and override specific sections.



[**g**](https://flask.palletsprojects.com/en/2.1.x/api/#flask.g) is automatically available in templates. Based on if g.user is set (from load\_logged\_in\_user), either the username and a log out link are displayed, or links to register and log in are displayed. **[url\_for()](https://flask.palletsprojects.com/en/2.1.x/api/" \l "flask.url_for)** is also automatically available, and is used to generate URLs to views instead of writing them out manually.

* + 1. **Human interface**

Diagram

Description automatically generated

**5.0 Restrictions, Limitations, and Constraint**

* In this paper a new collative comparison measure that reasonably represents the gains and losses due to fraud detection is proposed.
* A cost sensitive method which is based on Bayes minimum risk is presented using the proposed cost measure.
* Credit card fraud is a criminal offense. It causes severe damage to ﬁnancial institutions and individuals. Therefore, the detection and prevention fraudulent activities are critically important to ﬁnancial institutions. Fraud detection and prevention are costly,time-consuming and labor-intensive tasks.Anumber of signiﬁcant research works have been dedicated to developing innovative solutions to detect different types of fraud. However, these solutions have been proved ineffective. According to Cifa, 33,305 cases of credit card identity fraud were reported between January and June in 2018.